AISE

Questionnaire Specification Language (QSL)

Specification Version 3.1

30 Aug. 2018

Specification URIs

This version:
http://www.aise.ics.saitama-u.ac.jp/qsl/v3_1

Previous version:
http://www.aise.ics.saitama-u.ac.jp/qsl/v3_0

Affiliation:
Advanced Information Systems and Engineering Lab, Saitama University, Japan

QSL Members:
Y. Zhou (shuugen@aise.ics.saitama-u.ac.jp), D3
Z. Wang, M.S. graduated in 2015
D. Matsuura, B.S. graduated in 2016

Author:
Y. Zhou (shuugen@aise.ics.saitama-u.ac.jp), D3

Editor:
Y. Zhou (shuugen@aise.ics.saitama-u.ac.jp), D3

Abstract:
This document describes the background and purpose of Questionnaire Specification Language (QSL), presents the grammar of QSL in XML Schema including the whole elements, attributes, and the structure of it, and shows system template and service template for specifying systems and service of e-questionnaire, e-testing, and e-voting. We devote to continuously improve QSL better and better since 2013 till now.

Status:
This document was last revised on 30 Aug. 2018.
Contents

1 Introduction.................................................................................................................. 5
   1.1 Terminologies........................................................................................................... 5
   1.2 Background............................................................................................................... 5
   1.3 Overview of the Document...................................................................................... 6
   1.4 Changes in this Version.......................................................................................... 6
   1.5 Advantage of using QSL......................................................................................... 7
   1.6 How to use QSL....................................................................................................... 8
2 Requirements................................................................................................................ 9
   2.1 Stakeholders and Their Inconvenience.................................................................... 9
   2.2 Challenge and Scope.............................................................................................. 10
   2.3 Design Philosophy.................................................................................................. 10
   2.4 Documentation Set.................................................................................................. 11
3 QSL Structure................................................................................................................ 12
4 QSL Grammar Outline................................................................................................ 15
   4.1 Structure................................................................................................................ 15
   4.2 Viewing Schemas.................................................................................................... 15
   4.3 Schema Diagrams................................................................................................... 15
   4.4 Namespaces........................................................................................................... 18
   4.5 Conventions........................................................................................................... 18
5 QSL Core Component................................................................................................... 19
   5.1 Overview................................................................................................................ 19
   5.2 Simple Types......................................................................................................... 20
   5.2.1 AddressType........................................................................................................ 20
   5.2.2 AuthType............................................................................................................ 20
   5.2.3 BrowserType......................................................................................................... 20
   5.2.4 ChannelType......................................................................................................... 20
   5.2.5 ClientSideSolutionType....................................................................................... 21
   5.2.6 DatabaseType...................................................................................................... 21
   5.2.7 Data................................................................................................................... 21
   5.2.8 DescriptionType.................................................................................................. 21
   5.2.9 DesktopAppType.................................................................................................. 21
   5.2.10 DeviceType........................................................................................................ 22
   5.2.11 EmailType.......................................................................................................... 22
   5.2.12 Field................................................................................................................ 22
   5.2.13 FormatType....................................................................................................... 22
   5.2.14 FunctionType (Optional)................................................................................... 23
   5.2.15 Gender.............................................................................................................. 23
   5.2.16 Id..................................................................................................................... 23
   5.2.17 LanguageType.................................................................................................... 24
   5.2.18 LimitationType................................................................................................. 24
   5.2.19 Logic................................................................................................................ 24
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.20</td>
<td>MediaType</td>
<td>24</td>
</tr>
<tr>
<td>5.2.21</td>
<td>NameType</td>
<td>24</td>
</tr>
<tr>
<td>5.2.22</td>
<td>OrderType</td>
<td>25</td>
</tr>
<tr>
<td>5.2.23</td>
<td>OSTYPE</td>
<td>25</td>
</tr>
<tr>
<td>5.2.24</td>
<td>PaperType</td>
<td>25</td>
</tr>
<tr>
<td>5.2.25</td>
<td>ParticipantType</td>
<td>25</td>
</tr>
<tr>
<td>5.2.26</td>
<td>PhaseType</td>
<td>25</td>
</tr>
<tr>
<td>5.2.27</td>
<td>QuestionType</td>
<td>25</td>
</tr>
<tr>
<td>5.2.28</td>
<td>ReminderType</td>
<td>26</td>
</tr>
<tr>
<td>5.2.29</td>
<td>ReportType</td>
<td>26</td>
</tr>
<tr>
<td>5.2.30</td>
<td>SoftwareType</td>
<td>26</td>
</tr>
<tr>
<td>5.2.31</td>
<td>SystemType</td>
<td>26</td>
</tr>
<tr>
<td>5.2.32</td>
<td>TelNumType</td>
<td>26</td>
</tr>
<tr>
<td>5.2.33</td>
<td>TimeType</td>
<td>27</td>
</tr>
<tr>
<td>5.2.34</td>
<td>YesNoType</td>
<td>27</td>
</tr>
</tbody>
</table>

6 QSL Schema Descriptions

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Overview</td>
<td>28</td>
</tr>
<tr>
<td>6.2</td>
<td>Relations of Schemas</td>
<td>30</td>
</tr>
<tr>
<td>6.3</td>
<td>100-QSL</td>
<td>31</td>
</tr>
<tr>
<td>6.4</td>
<td>110-System</td>
<td>31</td>
</tr>
<tr>
<td>6.5</td>
<td>120-Service</td>
<td>32</td>
</tr>
<tr>
<td>6.6</td>
<td>210-Phase</td>
<td>33</td>
</tr>
<tr>
<td>6.7</td>
<td>220-Security</td>
<td>34</td>
</tr>
<tr>
<td>6.8</td>
<td>230-Paper</td>
<td>35</td>
</tr>
<tr>
<td>6.9</td>
<td>240-Setting</td>
<td>36</td>
</tr>
<tr>
<td>6.10</td>
<td>250-Environment</td>
<td>37</td>
</tr>
<tr>
<td>6.11</td>
<td>260-Participant</td>
<td>38</td>
</tr>
<tr>
<td>6.12</td>
<td>270-Data</td>
<td>39</td>
</tr>
<tr>
<td>6.13</td>
<td>280-Function</td>
<td>39</td>
</tr>
<tr>
<td>6.14</td>
<td>310-Settingup</td>
<td>40</td>
</tr>
<tr>
<td>6.14.1</td>
<td>AutoSaving</td>
<td>41</td>
</tr>
<tr>
<td>6.14.2</td>
<td>Checking</td>
<td>42</td>
</tr>
<tr>
<td>6.14.3</td>
<td>Distribute</td>
<td>42</td>
</tr>
<tr>
<td>6.14.4</td>
<td>Analyze</td>
<td>43</td>
</tr>
<tr>
<td>6.14.5</td>
<td>Interval</td>
<td>43</td>
</tr>
<tr>
<td>6.15</td>
<td>311-Distributing</td>
<td>44</td>
</tr>
<tr>
<td>6.16</td>
<td>312-Registering</td>
<td>45</td>
</tr>
<tr>
<td>6.17</td>
<td>313-Submitting</td>
<td>47</td>
</tr>
<tr>
<td>6.18</td>
<td>314-Collecting</td>
<td>50</td>
</tr>
<tr>
<td>6.19</td>
<td>315-Analyzing</td>
<td>51</td>
</tr>
<tr>
<td>6.20</td>
<td>316-Counting</td>
<td>52</td>
</tr>
<tr>
<td>6.21</td>
<td>320-Sponsor</td>
<td>54</td>
</tr>
<tr>
<td>6.22</td>
<td>321-Questioner</td>
<td>55</td>
</tr>
<tr>
<td>6.23</td>
<td>322-Respondent</td>
<td>56</td>
</tr>
</tbody>
</table>
6.24 323-Analyst........................................................... 57
6.25 324-Monitor.......................................................... 58
6.26 330-Export............................................................ 59
6.27 331-Import............................................................ 60
6.28 332-Launch........................................................... 61
6.29 333-Stop............................................................... 62
6.30 334-Generate.......................................................... 63
6.31 335-Ping.............................................................. 64
6.32 336-Integrate........................................................ 65
6.33 337-Remind........................................................ 66
6.34 340-Server........................................................... 67
6.35 341-Field............................................................. 68
6.35.1 Fields for Participant............................................. 69
6.35.2 Fields for Paper................................................ 71
6.35.3 Fields for Response.......................................... 73
6.35.4 Fields for Result............................................. 74
6.36 342-Gateway....................................................... 75
6.37 343-Interface...................................................... 76
6.38 344-Device........................................................ 77
6.39 345-Software...................................................... 78
6.40 350-Anonymity.................................................... 79
6.41 351-Authentication............................................. 81
6.41.1 Secrecy............................................................ 82
6.41.2 Token............................................................. 83
6.41.3 Biometric........................................................ 83
6.42 352-Authority...................................................... 84
6.43 353-Seal.............................................................. 85
6.44 354-Channel....................................................... 86
6.45 360-Section........................................................ 87
6.46 361-Question....................................................... 88
6.47 362-Answer......................................................... 90
6.48 363-Description................................................ 92
6.49 364-Media.......................................................... 92
6.50 365-Alignment...................................................... 93
6.51 366-Limitation..................................................... 94
6.52 370-Language...................................................... 95
6.53 371-Time........................................................... 96
6.54 372-Number......................................................... 97
6.55 373-Quota.......................................................... 98
6.56 374-Response...................................................... 99
6.57 375-Result.......................................................... 100
6.58 410-Logic.......................................................... 102
6.59 510-Marking....................................................... 104
6.60 520-Score.......................................................... 105
6.61 530-Sample
6.62 540-Formula
6.63 550-Marker
6.64 610-Auditing
6.65 620-Candidate
6.66 630-Proposer
6.67 640-Auditor

7 Question Types Reference Guide
7.1 Basic Question Types
7.1.1 Multiple Choice
7.1.2 Open-ended Text
7.1.3 Matrix
7.1.4 Ranking
7.2 Possible Combinations of Question Types
7.2.1 Contact Information
7.2.2 Matrix Spreadsheet
7.2.3 Slide and Rating
7.2.4 Rank Order and Drag
7.2.5 Upload File
7.2.6 Sid-by-side Matrix

8 Logic Types Reference Guide
8.1 Basic Logic Types
8.1.1 Skipping
8.1.2 Piping
8.1.3 Extraction
8.1.4 Randomization
8.2 Possible Combinations of Logic Types
8.2.1 Matrix Extraction
8.2.2 Compound Branching
8.2.3 Looping with Text Piping
8.2.4 Quota and Scoring Logic

Appendix A – The Cases for Using QSL
A.1 ENQUETE-BAISE
A.2 A General-Purpose Offline E-Testing Environment
A.3 An E-Voting Application of an Enterprise Information System on WeChat Public Platform

Appendix B – Revision History
1 Introduction

1.1 Terminologies

**QSL Document/ Requirement Specification:** the specification using QSL to describe is written by user who uses QSL apps or other XML editors.

**QSL Specification:** the QSL manual to guide user how to use QSL to specify systems and services of e-questionnaire, e-testing, and e-voting.

**Schema:** QSL grammar for supporting QSL. XML Schema defines these schemas. A schema is the structure of skeletons of requirements.

**Skeleton of Requirements:** a complete list of requirements, or a document of requirements without values. The stakeholders only need to fill in the values to complete the requirement specification.

**Tag:** is used to describe a piece of data for requirements of e-questionnaire, e-testing, and e-voting systems.

**Template:** set of skeletons of requirements.

1.2 Background

AISE members, to design a formalized method for various e-questionnaire, e-testing, and e-voting systems. The mission statement is, in part, to:

“to propose a standard and formalized tool to solve the problems among the stakeholders of various e-questionnaire, e-testing, and e-voting systems …”

The original objective in 2013 was to introduce a uniform and unique method to specify systems involved in the processes of e-questionnaire, e-testing, and e-voting, including data portability. The overall focus today provides a standard that is:

- **Multinational:** Our focus is to have standards that can be adopted globally.
- **Flexible:** Effective across the various questionnaires, testing, and polling regimes.
- **Multilingual:** Flexible enough to accommodate the various languages and dialects and vocabularies.
- **Adaptable:** Resilient enough to support questionnaires, examinations, and elections in both the private and public sectors.
- **Secure:** Able to secure the relevant data from ant attempt, as appropriate to the different requirements of various e-questionnaire, e-testing, and e-voting rules.

Questionnaire Specification Language (QSL) serves as a formalized specification for specifying various e-questionnaire, e-testing, and e-voting systems. QSL provides vocabulary and notation with a standardized, consistent, and exhaustive requirement list for specifying systems and services of e-questionnaire, e-testing, and e-voting.
QSL is a format for portable data. QSL is the precondition to automatically generate e-questionnaire, e-testing, and e-voting. QSL is based on Extensible Markup Language (XML). The grammar of QSL is defined by XML Schema. At present, QSL includes specifications for two templates and their data information:

- **System Template**: security requirements, environment requirements including software and hardware, function requirements during phases of setting up, registering, distributing, submitting, collecting, analyzing, counting, marking, auditing.
- **Service Template**: requirements for paper sheet, settings for paper sheets during phases.
- **System Data**: participant fields for each role, paper fields, and response and result fields.
- **Service Data**: participant information, real responses by respondents, and counts, statistics, and results.

This document and its schemas and structures of two templates represent our best current efforts, knowledge and experience with e-questionnaire, e-testing, and e-voting systems since 2013. It is incumbent on users of this document to identify and requirement gaps, mistakes, inconsistencies or missing data and to propose corrections or enhancement to AISE.

### 1.3 Overview of the Document

To help establish context for the specifications contained in the XML Schema that make up QSL, the first questionnaire specification language for e-questionnaire, e-testing, and e-voting systems. AISE also designed the structure of it. This structure identifies how to combine each elements to obtain various e-questionnaire, e-testing, and e-voting systems. In this document, we present a lot of specifications as templates for references and reusability, and describe how QSL can be used to standardize the data exchanged.

### 1.4 Changes in this Version

The changes from QSL v 3.0 that this new version introduces are as follows:

- Add much more schemas for common requirements of e-questionnaire, e-testing, and e-voting, which are extracted from 200-series schemas in version 3.0;
- Revise some attributes to let 32 requirements QSL can specify.
1.5 Advantage of using QSL

The question usually asked about why any particular standard should be used is “what advantages will it bring me?” In addition to supporting trusted e-questionnaire, e-testing, and e-voting systems, the benefits of adopting QSL are as follows:

Firstly, QSL can specify various systems and services of e-questionnaire, e-testing, and e-voting. It satisfies public demands to unify e-questionnaire, e-testing, and e-voting systems, and promotes the communications of stakeholders among these essentially similar systems. Secondly, QSL can be a unified format for portable data among different e-questionnaire, e-testing, and e-voting systems. It contributes to improve data portability, which right is very important for our modern society and accepted by General Data Protection Regulation since 2016 for providing conveniences for the stakeholders to reuse data. Thirdly, QSL can be the precondition to automatically generate e-questionnaire, e-testing, and e-voting systems. It is conducive to improving efficiency by automating and streamlining stakeholders' work procedures.

For sponsors:
• More choice of products and supporters;
• Clearly mind and unambiguous;
• Supports scalability, transparency, and data reusability;
• Provides basis for make clearly work division, and responsibility;

For executor:
• More choice of survey sheets;
• Clearly work division, and responsibility;
• Supports scalability, transparency, and data reusability;

For respondents:
• Supports trustworthiness of systems;
• Support security of e-questionnaire, e-testing, and e-voting;

For supporters:
• Easily communicate with sponsors and executors;
• Reduced development costs;
• Accommodates future changes more easily;
• Common core but allows local extension;

For evaluators:
• Easily check specifications;
1.6 How to use QSL

As a specification language, QSL has to meet various requirements of systems and services of e-questionnaire, e-testing, and e-voting. Therefore it may need to be tailored for specific scenarios and meet specific rules and practices.

First using the QSL Schemas (XSD files) and an xml-editor to translate it and to build an xml file, that is a QSL template. You can also use a requirement specification or a template, to change it or fill the values by your own mind but do not need to break up the QSL grammar definition.

Secondly, because we have the marked numbers for each schema, you can select which kinds of systems and services you must to specify.

There are two kinds of files, you can download, are listed as follow:

- QSL Templates (System Template and Service Template in XML);
- QSL Documents (Specification specified by QSL in XML);
- QSL Schemas (XSD);

A recommendation of tool for supporting to edit QSL templates, we used XML Spy to design it, and edit it based on templates to create specifications. In future, we are planning to implement a QSL structure editor to edit specification in a QSL-format, and a QSL generator system to automatically generating e-questionnaire, e-testing, and e-voting systems.
2 Requirements

2.1 Stakeholders and Their Inconvenience

There are five kinds of stakeholders around e-questionnaire, e-testing, and e-voting systems, which are:

- **Sponsor** launches activity and usually focuses on the strategic goals, return on investment, as well as the costs and time involved in building and operating the systems;
- **Executor** performs tasks to help carry out the activities after system deployment that perhaps contains questioner, monitor, analyst, etc.;
- **Respondent** is an investigated target to answer a questionnaire, a test, or a vote;
- **Supporter** provides the tech-support services that includes communicator, developer, maintainer, manufacturing engineer, supplier, customer service, etc.;
- **Evaluator** monitors whether the system meets standards, laws, and regulations.

It lacks a unified communication tool shared among the stakeholders of services and systems of e-questionnaire, e-testing, and e-voting that it causes two communication problems occurring among those stakeholders. Firstly, it is not so easy for the stakeholders to define exhaustive requirements of the systems and the services of e-questionnaire, e-testing, and e-voting. For instance, the sponsor, the executor, and the supporter only focus on their own requirements, but have a shallow understanding of the overall requirements. It lacks an exhaustive requirement list to guide and assist the stakeholders to know overall requirements for systems and services of e-questionnaire, e-testing, and e-voting. Secondly, it is difficult to avoid ambiguity for the stakeholders because they do not have standardized terminologies on the systems and the services of e-questionnaire, e-testing, and e-voting. For example, the sponsor use different terminologies for a common requirement of three kinds of services. It may lead the supporter to provide different solutions for the system, after all, the supporter is not the professional expert in the business fields, and the sponsor is not a technical specialist. Moreover, terminologies among the systems and the services of e-questionnaire, e-testing, and e-voting are not unified although the systems and the services have common processes. In another example, the sponsor and the executor use the same terminology for different requirements of services of e-questionnaire, e-testing, and e-voting. It may lead the supporter to provide same solution for the system. It lacks standardized terminologies to unify the common requirements, and to normalize the different requirements for services of e-questionnaire, e-testing, and e-voting and the corresponding systems. To solve the above problems, we proposed QSL.
2.2 Challenge and Scope

Our goal has been to design an Questionnaire Specification Language (QSL) for specifying various e-questionnaire, e-testing, and e-voting systems to all the stakeholders within the event of it. It contains a set of requirements of systems and services specified as a set of XML Schemas during all the phases (state) of the events of e-questionnaire, e-testing, and e-voting.

QSL is meant to assist and enable phases and does not need any changes to traditional methods of executing e-questionnaire, e-testing, and e-voting. The extensibility of QSL makes it possible to adjust to various e-questionnaire, e-testing, and e-voting without affecting the phases. Conceptually QSL simply enable the data portability among stakeholders and the systems in a standardized way.

The solution outlined in this document is non-proprietary and will work as a template for any scenario of e-questionnaire, e-testing, and e-voting using their systems and services for all or part of the process. The objective is to introduce a uniform method to allow the systems to interact with each other.

2.3 Design Philosophy

“How to specify various e-questionnaire, e-testing, and e-voting systems? “ That is the first and the most important question in our mind when we decided to design QSL. Mendeleyev’s periodic table enlightens us, are there some primary elements arranged by some certain rules for the most part, and this arrangement revealed the periodicity of something? The design of QSL is still with our first heart.

According to a lot of investigations of existing e-questionnaire, e-testing, and e-voting systems, the similarities and differences are found.

Firstly, we list six aspects of common requirements as follow:

- **Phase**: setting up, distributing, submitting, collecting, and counting;
- **Security**: consists of authentication, anonymity, and authority;
- **Paper**: question sheet consists of section, question, answers;
- **Setting**: setting for each phase, which set up by questioner;
- **Environment**: consists of server, software, device, database, interface;
- **Participant**: sponsor, questioner, analyst, monitor, and respondent;
- **Data**: consists of participant data, result data, report data, etc.;
- **Validation**: limitation, response required, etc.;
- **Function**: launch, stop launching, distribute, etc.;

---

1 The periodic table is a tabular arrangement of the chemical elements, ordered by their atomic number (number of protons), electron configurations, and recurring chemical properties.

qsl-v3.1-2018

Specifications and Manual AISE Lab, Saitama Univ., Japan
Secondly, we list the different requirements as follow:

- **E-Questionnaire:**
  - Question Type: rating, multiple dimensions matrix, etc.;
  - Logic: extraction, complex piping etc.;

- **E-Testing:**
  - Participant: marker;
  - Phase: marking;
  - Data: sample answer, score;

- **E-Voting:**
  - Phase: auditing;
  - Participant: candidates, auditor, proposal;

After a sketchy analysis, the differences are totally categorized into groups, which are listed in the appendix. Obviously, the list of primary elements is much more elaborated and deliberated, because primary elements are the train of thought of designing QSL (but not the most important of this manual).

### 2.4 Documentation Set

To reach our goal, we defined:

- **QSL Specification:** This document. This introduces the QSL structure and QSL grammar from a complete manual for specifying e-questionnaire, e-testing, and e-voting systems by defining the systematic functions, process, paper sheet, setting, data information, security of content, and describe how uses them encoded as QSL schemas.

- **QSL Schemas:** This consists of a library of XML schemas used in QSL. The XML schemas defined the formal structures of skeletons of requirements.

- **Templates:** sets of schemas for describing requirements of systems and services each e-questionnaire, e-testing, e-voting in two parts (system and service) provide templates that facilitated generation of the main schema structure. This aims to reduce sponsors and executors’ time to specify the service requirements, and reduce supporters’ costs to specify the system requirements and implement and generate real system.
3 QSL Structure

QSL structure is relatively robust, extend and upgrade easily without changing the whole configuration. QSL structure has two layers. In the innermost layer, QSL defines core requirements. Specifying any e-questionnaire, e-testing, and e-voting system must specify all the core (common) requirements. In the outermost layer, there are three isolated ranges, which are different requirements for e-questionnaire, e-testing, and e-voting, respectively.

![QSL Structure](image)

**Fig. 3-1** QSL Structure

QSL is proposed to provide exhaustive requirements of systems and services of e-questionnaire, e-testing, and e-voting. Those requirements are classified into common requirements and different requirements for systems and services of e-questionnaire, e-testing, and e-voting, respectively. We defined skeletons of requirements of systems and services of e-questionnaire, e-testing, and e-voting. In order to provide convenience for the stakeholders to easily use QSL, we propose the sets of skeletons of requirements that we called them as the templates. The templates are used to respectively specify systems and services of e-questionnaire, e-testing, and e-voting. Each template includes core construction and respective constructions. Core construction provides the skeletons of common requirements and document hierarchy structure that helps to define the basic system or service for e-questionnaire, e-testing, and e-voting. Each respective construction provides the skeletons of the different requirement list for e-questionnaire, e-testing, and e-voting, respectively that helps to distinguish e-questionnaire, e-testing, and e-voting. When the stakeholder (usually executor or supporter) makes document to describe requirements, firstly he chooses a template. After that, he fills values in core and single respective construction, and can get the specification easily.

QSL is used to describe various requirements of e-questionnaire, e-testing, and e-voting systems by using tags. We called the document of requirements described by QSL as requirement specification. The requirement specification is divided into system requirement specification and service requirement specification. Before the
stakeholders finish the requirement specification, they should be provided with a complete list of requirements, that is, the skeleton of requirements. The skeleton of requirements is a document of requirements without values. The stakeholders only need to fill in the values to complete the requirement specification. In addition, we defined template, which is a set of skeletons of requirements. We provide the sponsor and the executor with the service template, and the supporter with system template. When they choose a template and fill in the appropriate values, they can get a QSL-format requirement specification easily.

We defined the structure of each skeleton of requirements that is used to describe the valid document structure, i.e., schema. The schemas constrain where are tags in the document, which attributes they have, which kinds of data are filled in, and so on. Considering that there is no tool for proofreading the requirement specification, the stakeholders can only use the existing XML editing tool to import the schemas, convert into a template, fill in the values, and complete the requirement specification. The editing tool helps to check the validation of the document according to the schemas and avoid misoperation when the stakeholders directly edit template. Although the template provides convenience to the stakeholders, the best solution is based on the proofreading that refers to the schemas, to fill in the template.

In QSL, each schema provides an intuitive means of requirement navigations for different systems and services for e-questionnaire, e-testing, and e-voting by organizing the corresponding tags in a hierarchical structure. To assist the stakeholders choose the corresponding schemas, we defined each schema and marked with number. Firstly, 100-series schemas are used to declare QSL document and define the structures of system template and service template. The stakeholder can easily get a template with QSL declaration. Secondly, 200-series schemas are used to define the subdivisions of the structures of templates according to the groups of entities we summarized previously, these schemas help the stakeholders clearly understand what kinds of requirements should be specified in a template. Thirdly, 300-series schemas are used to define the structures of skeletons of common requirements. Fourthly, 400-series, 500-series, and 600-series schemas are used to define the structures of skeletons of specific requirements of systems and services for e-questionnaire, e-testing, and e-voting, respectively. Using the marked numbers, the stakeholders can easily distinguish these three kinds of systems and services. Overall, we defined 65 schemas in QSL. There are 44 schemas (300-series schemas) for specifying common requirements, 10 schemas (400-series, 500-series, and 600-series schemas) for specifying specific requirements, 10 schemas (110, 120, and 200-series schemas) for constructing the hierarchy structure of QSL documents, and 1 schema (100 schema) for QSL document declarations.

110 schema is oriented to the supporters and 120 schema is oriented to the executors and the sponsors. The executors and the sponsors choose 120 schema as the structure of service template and they can complete the service requirement specification based
on this schema. The supporters choose 110 schema as the structure of system template, and they refer to the service requirement specification to complete the system requirement specification. The 110 schema and 120 schema must declare QSL document that is based on the 100 schema. The remaining schemas are used to construct the 110 schema and 120 schema. In addition, both 110 schema and 120 schema can be used to provide descriptions to specify data. 110 schema provides the supporters with descriptions for database schema that is defined as 341 schema in QSL. According to 341 schema, the supporters can describe and design database fields. In addition, 120 schema is used to describe participant information, responses, and results. It is desirable to reuse these data.
4 QSL Grammar Outline

4.1 Structure

The Questionnaire Specification Language specification defines a vocabulary and divided messages. Thus common requirements of e-questionnaire, e-testing, and e-voting are defines as elements in the 300-series schemas. The 270 schemas also contain data definition so that be used in data portability. Each message is specified and defined with a separate schema document, and can be combined. As we introduced in pre-chapter, each schema is defined as messages with its unique IDs. According to the combination of messages, users can get the data what they want. Obviously, it is order by a rule we gave to.

4.2 Viewing Schemas

QSL grammar is supplied as xml documents. For viewing the structure of it, we recommend to use an XML editor, such as XML spy. Alternatively, if you read a text document with a graphical display, it is easy to refer to.

4.3 Schema Diagrams

The diagram below represents a simple schema in a XML Spy 2011 style. The root element of an instance described by this schema is the element A. The content model of this element is a sequence of the elements B, D and E. The element B is of complex data type Bstructure. This contains a choice of either element C or element F. Element C is a restriction of another complex data type Cstructure. In this case, the restriction is to forbid the use of the element G (which is defined in Cstructure as optional). The other elements allowed are H, which can appear any number of times (but must appear at least once), and I, which can appear up to three times (or not at all). Element D is optional, and of data type Dstructure. This has a content model requiring all of elements J and K, which are both of type xs:string. Finally, element E is of simple data type Etype, which is restricted from the xs:NMTOKEN data type by only allowing the values ‘yes’ and ‘no’.

It is important to remember that these diagrams do not include any attributes. In this document, these are shown in tables below the diagrams. The full schema is shown below the diagram.
Fig. 4-1  Sample Schema Diagram
4.4 Namespaces

The QSL schemas are associated with the namespace:
http://www.aise.ics.saitama-u.ac.jp/qsl. This is defined using the prefix qsl. The XML schema namespace http://www.w3.org/2001/XMLSchema is identified by the prefix xs and the XML schema Instance namespace:
http://www.w3.org/2001/XMLSchema-instance by the prefix xsi.
schema version is 3.1. This specification is an English version.
Such messages start with a QSL element, such as:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<QSL
  schemaVersion="3.1"
  xml:lang="en"
  xmlns="http://www.aise.ics.saitama-u.ac.jp/qsl"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
</QSL>
```

4.5 Conventions

Within this specification, the following conventions are used throughout:
• Diagrams are shown as generated by XML Spy 2011 which was also used to generate the schemas and samples. These diagrams show element content, but not attributes;
• Elements and attributes in schemas are identified by partial Xpath expressions. Enough of a path is used to identify the item without putting in a full path.
## 5 QSL Core Component

### 5.1 Overview

The following schema component are defined in the QSL schema:

<table>
<thead>
<tr>
<th>Elements</th>
<th>Simple Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counting</td>
<td>Field</td>
</tr>
<tr>
<td>Data</td>
<td>Formula</td>
</tr>
<tr>
<td>Description</td>
<td>Function</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Question</td>
</tr>
<tr>
<td>Action</td>
<td>Questioner</td>
</tr>
<tr>
<td>Alignment</td>
<td>Quota</td>
</tr>
<tr>
<td>Analyst</td>
<td>Ratio</td>
</tr>
<tr>
<td>Analyze</td>
<td>Generate</td>
</tr>
<tr>
<td>Analyzing</td>
<td>Import</td>
</tr>
<tr>
<td>Anonymity</td>
<td>Integrate</td>
</tr>
<tr>
<td>Answer</td>
<td>Interval</td>
</tr>
<tr>
<td>Audit</td>
<td>Language</td>
</tr>
<tr>
<td>Auditing</td>
<td>Launch</td>
</tr>
<tr>
<td>Auditor</td>
<td>Limitation</td>
</tr>
<tr>
<td>Authenticate</td>
<td>Mark</td>
</tr>
<tr>
<td>Authentication</td>
<td>Media</td>
</tr>
<tr>
<td>Authority</td>
<td>Method</td>
</tr>
<tr>
<td>AutoSaving</td>
<td>Monitor</td>
</tr>
<tr>
<td>Count</td>
<td>Marker</td>
</tr>
<tr>
<td>Biometric</td>
<td>Marking</td>
</tr>
<tr>
<td>Candidate</td>
<td>Number</td>
</tr>
<tr>
<td>Channel</td>
<td>QSL</td>
</tr>
<tr>
<td>Checking</td>
<td>Observe</td>
</tr>
<tr>
<td>Collect</td>
<td>Observe</td>
</tr>
<tr>
<td>Collecting</td>
<td>Paper</td>
</tr>
<tr>
<td>Condition</td>
<td>Participant</td>
</tr>
<tr>
<td>Device</td>
<td>Phase</td>
</tr>
<tr>
<td>Distribute</td>
<td>Ping</td>
</tr>
<tr>
<td>Distributing</td>
<td>Proposer</td>
</tr>
<tr>
<td>Environment</td>
<td>Register</td>
</tr>
<tr>
<td>Export</td>
<td>Registering</td>
</tr>
<tr>
<td>YesNoType</td>
<td></td>
</tr>
<tr>
<td>REMIND</td>
<td>Response</td>
</tr>
<tr>
<td>AuthType</td>
<td>Reply</td>
</tr>
<tr>
<td>BrowserType</td>
<td>Report</td>
</tr>
<tr>
<td>ChannelType</td>
<td>Respondent</td>
</tr>
<tr>
<td>ClientSideSolutionType</td>
<td></td>
</tr>
<tr>
<td>DatabaseType</td>
<td></td>
</tr>
<tr>
<td>DataType</td>
<td></td>
</tr>
<tr>
<td>DescriptionType</td>
<td></td>
</tr>
<tr>
<td>DesktopAppType</td>
<td></td>
</tr>
<tr>
<td>DeviceType</td>
<td></td>
</tr>
<tr>
<td>EmailType</td>
<td></td>
</tr>
<tr>
<td>FieldType</td>
<td></td>
</tr>
<tr>
<td>FormatType</td>
<td></td>
</tr>
<tr>
<td>FunctionType</td>
<td></td>
</tr>
<tr>
<td>GenderType</td>
<td></td>
</tr>
<tr>
<td>IdType</td>
<td></td>
</tr>
<tr>
<td>LanguageType</td>
<td></td>
</tr>
<tr>
<td>LimitationType</td>
<td></td>
</tr>
<tr>
<td>LogicType</td>
<td></td>
</tr>
<tr>
<td>NameType</td>
<td></td>
</tr>
<tr>
<td>OrderType</td>
<td></td>
</tr>
<tr>
<td>OSTYPE</td>
<td></td>
</tr>
<tr>
<td>PaperType</td>
<td></td>
</tr>
<tr>
<td>ParticipantType</td>
<td></td>
</tr>
<tr>
<td>PhaseType</td>
<td></td>
</tr>
<tr>
<td>QuestionType</td>
<td></td>
</tr>
<tr>
<td>ReminderType</td>
<td></td>
</tr>
<tr>
<td>ReportType</td>
<td></td>
</tr>
<tr>
<td>SoftwareType</td>
<td></td>
</tr>
<tr>
<td>SystemType</td>
<td></td>
</tr>
<tr>
<td>TelNumType</td>
<td></td>
</tr>
<tr>
<td>TimeType</td>
<td></td>
</tr>
<tr>
<td>YesNoType</td>
<td></td>
</tr>
</tbody>
</table>
5.2 Simple Types

The simple types are included here with their base types and any restriction applied.

5.2.1 AddressType

xs:token with restriction.
Restriction: xs:maxLength: 255
This type is a simple definition of an address.

5.2.2 AuthType

xs:token with restriction.
Restriction: xs:enumeration:
This type is a simple definition of authentication methods, which can be classified into 3 types.

<table>
<thead>
<tr>
<th>type</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secrecy</td>
<td>password, randomized password</td>
</tr>
<tr>
<td>Token</td>
<td>ID card, job card, library card, roll card</td>
</tr>
<tr>
<td>Biometric</td>
<td>finger prints, iris, face recognition, signature, DNA</td>
</tr>
<tr>
<td>--</td>
<td>others</td>
</tr>
</tbody>
</table>

5.2.3 BrowserType

xs:token with restriction.
Restriction: xs:enumeration: IE, firefox, sarfi, chrome, others
This type is a simple definition of browser. Values are IE, firefox, sarfi, chrome, others.

5.2.4 ChannelType

xs:token with restriction.
Restriction: xs:enumeration: SMS, WAP, digital TV, Internet, Intranet, Kiosk, postal, telephone, digital storage device, paper, fax, email, website, abroad postal, abroad electronic, abroad other, N/A, other
This type exists to hold the possible enumerations for the channel through where a paper and a register-sheet are submitted, a response is collected. SMS is the short
message service (text message). WAP is the wireless access protocol. If other is used, it is assumed that those managing the event will have a common understanding of the channel in use.

5.2.5 ClientSideSolutionType

xs:token with restriction.
Restriction: xs: enumeration: web browser, fat-client, thin-client
This type is a simple definition of browser. Web browser approach is only used to establish the link to run on the submitting server. Fat-client approach is rich of security functionality and cryptographic algorithm that the client side software needs to be installed and executed on the submitting devices. Thin-client approach is a mix of web browser and fat-client that implements a java applet running in the web browser.

5.2.6 DatabaseType

xs:token with restriction.
Restriction: xs: enumeration: Access, SQL Server, DB2, others
This simple type is used to define the system database type.

5.2.7 DataType

xs:token with restriction.
Restriction: xs: enumeration: participant, response, report, setting, paper, paper and setting, field, others
This simple type is used to define the data type for combinations.

5.2.8 DescriptionType

xs:token with restriction.
Restriction: xs: enumeration: header, paragraph, break
This type is a simple definition for description in paper design. This allows that paper, section, question, and answer have header as a title, paragraph as an explanation, and a break for dividing pages. Considering the lightweight arrangement, which questioner wants, it is designed like this.
5.2.9 DesktopAppType

xs:token.
This type is a simple definition for desktop application.

5.2.10 DeviceType

xs:token with restriction.
Restriction: xs: enumeration: PC, PDA, smart phone, tablet, fable, access point, hub, cable, usb, others
This type is a simple definition for client-side device.

5.2.11 EmailType

xs:token with restriction.
Restrictions: xs: maxLength: 129
   xs:pattern: [@]+[@]+[@]+[@]+
This type is a simple definition of an email address, pending a more complete description that is widely accepted in government. It allows any characters except the @ symbol, following by an @ symbol and another set of characters.

5.2.12 FieldType

xs:token with restriction.
Restrictions: xs: enumeration: char, varchar, number, date, int, float, double, decimal, longtext
This type is a simple definition of the field type of database. It is an optional simple type.

5.2.13 FormatType

xs:token with restriction.
Restrictions: xs: enumeration: csv, excel, qsl, xml, word, spss, zip, others
This type is a simple definition of the format type for export and import.
5.2.14 FunctionType (Optional)

xs:token with restriction.
Restrictions: xs: enumeration: gap analysis, trend analysis, comparison analysis, basis analysis, report, answer, submit, monitor, ping, integrate, back, internalStop, authenticate, login, monitor, create, edit, copy, import, export, delete, search, generate tokens, distribute, collect, mark, blind mark
This type is a simple definition of functional actions. Each value is defined as a verb easily taken from literality. The values need to be specially notified are internalStop (stop some action in the midway), blind mark (usually in e-testing for a teacher to mark the responses of anonymous respondents), generate tokens (automatically generate participants (especially respondents) admission ticket in an exam), ping (ping the IP address and the status of respondents), integrate (integrate some divided information like each respondent's response, etc.), monitor the states.

5.2.15 GenderType

xs:token with restriction.
Restriction: xs: enumeration: male, female, unknown
The gender of participant. Options are male, female, or unknown (is not allowed in all contexts).

5.2.16 IdType

xs:NMTOKEN with restrictions.
Restrictions: xs: maxLength: 10
xs: minLength: 4
xs: pattern: prefix +[0-9]*
This type is a simple definition of QSL elements' Id. To distinguish different elements, it allows the pattern value to add prefixes. The representative prefixes concern the PaperType and ParticipantType.

<table>
<thead>
<tr>
<th>Element</th>
<th>Pattern</th>
<th>Element</th>
<th>Pattern</th>
<th>Element</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper</td>
<td>paper[0-9]*</td>
<td>Section</td>
<td>section[0-9]*</td>
<td>Question</td>
<td>q[0-9]*</td>
</tr>
<tr>
<td>Answer</td>
<td>answer[0-9]*</td>
<td>Description</td>
<td>de[0-9]*</td>
<td>Media</td>
<td>me[0-9]*</td>
</tr>
<tr>
<td>Sponsor</td>
<td>sp[0-9]*</td>
<td>Respondent</td>
<td>re[0-9]*</td>
<td>Questioner</td>
<td>qu[0-9]*</td>
</tr>
<tr>
<td>Analyst</td>
<td>an[0-9]*</td>
<td>Marker</td>
<td>ma[0-9]*</td>
<td>Monitor</td>
<td>mo[0-9]*</td>
</tr>
<tr>
<td>Auditor</td>
<td>au[0-9]*</td>
<td>Function</td>
<td>func[0-9]*</td>
<td>Route</td>
<td>route[0-9]*</td>
</tr>
<tr>
<td>Condition</td>
<td>co[0-9]*</td>
<td>Action</td>
<td>ac[0-9]*</td>
<td>Software</td>
<td>sof[0-9]*</td>
</tr>
<tr>
<td>Server</td>
<td>server[0-9]*</td>
<td>Device</td>
<td>Device[0-9]*</td>
<td>Method</td>
<td>me[0-9]*</td>
</tr>
<tr>
<td>Interface</td>
<td>interface[0-9]*</td>
<td>Statistic</td>
<td>st[0-9]*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.2.17 **LanguageType**

xs:language.
Declaration of the type of language used in e-questionnaire, e-voting, and e-testing systems.

5.2.18 **LimitationType**

xs:token with restriction.
Restrictions: xs: enumeration: alphabet, figure
This type is a simple definition of limitation type.

5.2.19 **LogicType**

xs:token with restriction.
Restriction: xs: enumeration: skipping, piping, extraction, randomization
This type is a simple definition of logic type. Its values are basic logic type. As to the combination of them are introduce in [Chapter 8](#) in details. This manual takes a complete chapter to explain it. It is a difficult part in this manual.

5.2.20 **MediaType**

xs:token with restriction.
Restriction: xs: enumeration: video/ogg, video/mp4, audio/ogg, audio/mpeg, img/jpg, img/png, img/gif
This type is a simple definition of media types. It is designed by referring to HTML's tags `<video>` and `<img>`.

5.2.21 **NameType**

xs:token with restrictions.
Restrictions: xs: maxLength: 129
     xs: maxLength: 2
The name of participant and affiliation.
5.2.22 **OrderType**

xs:token with restriction.
Restrictions: xs: enumeration: ascending, descending, random.
This type is a simple definition of order type of settings.

5.2.23 **OSType**

xs:token with restriction.
This type is a simple definition of operating systems (usually defined in software).

5.2.24 **PaperType**

xs:token with restriction.
Restrictions: xs: enumeration: paper, section, question, answer
This type is a simple definition of contents of the main paper structure.

5.2.25 **ParticipantType**

xs:token with restriction.
Restrictions: xs: enumeration: sponsor, questioner, respondent, marker, auditor, analyst, monitor, candidate, proposer.
This type is a simple definition of participant roles.

5.2.26 **PhaseType**

xs:token with restriction.
Restrictions: xs: enumeration: setting up, submitting, registering, collecting, analyzing, marking, auditing, counting.
This type is a simple definition of phase.

5.2.27 **QuestionType**

xs:token with restriction.
Restrictions: xs: enumeration:
This type is a simple definition of question types. It is much more complex because until now there are over 36 question types in the world. Much more detailed information is defined in Chapter 7.

5.2.28 ReminderType

xs:token with restriction.
Restriction: xs: enumeration: password, token type, answer notification, answer finished, close submitting time, declaration time, report
This type is a simple definition to notify what contents.

5.2.29 ReportType

xs:token with restriction.
Restriction: xs: enumeration: basic, gap, trend
This type is a simple definition of analysis types. Basic type is a percentage report. Gap type is the comparison report. Trend type is always used in e-questionnaire for trend report.

5.2.30 SoftwareType

xs:token with restriction.
Restriction: xs: enumeration: database, browser, desktop app, OS

5.2.31 SystemType

xs:token with restriction.
Restriction: xs: enumeration: system
If it need to be divided into 3e systems for extension.

5.2.32 TelNumType

xs:token with restriction.
Restrictions: xs: maxLength: 35
      xs: minLength: 1
Since this must allow for various styles of international telephone number. The pattern has been kept simple. This allows an option plus sign, then between 1 and 35 characters with a combination of digits, brackets, the dash symbol and white space.

### 5.2.33 TimeType

xs:dateTime.

This type abides by the format of XML Schema’s xs:dateTime. The definition of the date and time uses the following format: YYYY-MM-DDThh:mm:ss Z (YYYY stands for year, MM stands for month, DD stands for day, T stands for the start of necessary time, hh stands for hour, mm stands for minute, ss stands for second, Z stands for the time zone identified by UTC).

### 5.2.34 YesNoType

xs:token with restrictions.

Restrictions: xs:enumeration: yes, no

This is a simple enumeration of yes and no and is used for elements and attributes than can only take these values.
# 6 QSL Schema Descriptions

## 6.1 Overview

The following table lists an overview of QSL schemas. Further explanations are given in the following sub paragraphs.

<table>
<thead>
<tr>
<th>Schema Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>QSL-100 QSL</td>
<td>declares QSL documents</td>
</tr>
<tr>
<td>QSL-110 System</td>
<td>provides schema of system</td>
</tr>
<tr>
<td>QSL-120 Service</td>
<td>provides schema of service</td>
</tr>
<tr>
<td>QSL-210 Phase</td>
<td>provides schema for complete phases with fulfilled functions and settings</td>
</tr>
<tr>
<td>QSL-220 Security</td>
<td>provides schema for system security requirements</td>
</tr>
<tr>
<td>QSL-230 Paper</td>
<td>provides schema of a complete paper construction with logic</td>
</tr>
<tr>
<td>QSL-240 Setting</td>
<td>provides schema of settings that questioner sets up before distributing</td>
</tr>
<tr>
<td>QSL-250 Environment</td>
<td>provides schema of server, software, device, network, interface</td>
</tr>
<tr>
<td>QSL-260 Participant</td>
<td>provides schema of participant roles and information</td>
</tr>
<tr>
<td>QSL-270 Function</td>
<td>provides schema of system functions</td>
</tr>
<tr>
<td>QSL-280 Data</td>
<td>provides schema of recorded data of services, and database fields for systems</td>
</tr>
<tr>
<td>QSL-310 SettingUp</td>
<td>provides schema of setting up phase</td>
</tr>
<tr>
<td>QSL-311 Registering</td>
<td>provides schema of registering phase</td>
</tr>
<tr>
<td>QSL-312 Distributing</td>
<td>provides schema of distributing phase</td>
</tr>
<tr>
<td>QSL-313 Submitting</td>
<td>provides schema of submitting phase</td>
</tr>
<tr>
<td>QSL-314 Collecting</td>
<td>provides schema of collecting phase</td>
</tr>
<tr>
<td>QSL-315 Analyzing</td>
<td>provides schema of analyzing phase</td>
</tr>
<tr>
<td>QSL-316 Counting</td>
<td>provides schema of counting phase</td>
</tr>
<tr>
<td>QSL-320 Sponsor</td>
<td>provides schema of participant role of sponsor</td>
</tr>
<tr>
<td>QSL-321 Questioner</td>
<td>provides schema of participant role of questioner</td>
</tr>
<tr>
<td>QSL-322 Respondent</td>
<td>provides schema of participant role of respondent</td>
</tr>
<tr>
<td>QSL-323 Analyst</td>
<td>provides schema of participant role of analyst</td>
</tr>
<tr>
<td>QSL-324 Monitor</td>
<td>provides schema of participant role of monitor</td>
</tr>
<tr>
<td>QSL-330 Export</td>
<td>provides schema of export function</td>
</tr>
<tr>
<td>QSL-331 Import</td>
<td>provides schema of import function</td>
</tr>
<tr>
<td>QSL-332 Launch</td>
<td>provides schema of launch function</td>
</tr>
<tr>
<td>QSL-333 Stop</td>
<td>provides schema of stop launching function</td>
</tr>
<tr>
<td>QSL-334 Generate</td>
<td>provides schema of generate function</td>
</tr>
<tr>
<td>QSL-335 Ping</td>
<td>provides schema of ping IP function</td>
</tr>
<tr>
<td>QSL-336 Integrate</td>
<td>provides schema of integrate function</td>
</tr>
</tbody>
</table>
QSL-337 Remind provides schema of remind function
QSL-340 Server provides schema of server information
QSL-341 Field provides schema of database field for system data
QSL-342 Gateway provides schema of network information
QSL-343 Interface provides schema of interface information
QSL-344 Device provides schema of device information
QSL-345 Software provides schema of software information
QSL-350 Anonymity provides schema of anonymous methods
QSL-351 Authentication provides schema of authentication with methods and channel
QSL-352 Authority provides schema of access control
QSL-353 Seal provides schema of encryption
QSL-354 Channel provides schema of communication channel
QSL-360 Section provides schema of section for a group of questions
QSL-361 Question provides schema of question for a group of answers
QSL-362 Answer provides schema of answer
QSL-363 Description provides schema of description for paper, section, question, and answer
QSL-364 Media provides schema of multi-media
QSL-365 Alignment provides schema of paper arrangement
QSL-366 Limitation provides schema of limitation of text or numbers
QSL-370 Language provides schema of setting for multi-languages
QSL-371 Time provides schema of start time, end time, and time zone
QSL-372 Number provides schema of numbering following orders
QSL-373 Quota provides schema of limitation for rate, quota, or numbers of respondents
QSL-374 Response provides schema of response data
QSL-375 Result provides schema of result data
QSL-410 Logic provides schema of logic of paper, section, question and answer
QSL-510 Marking provides schema of marking phase
QSL-520 Score provides schema of scoring rules
QSL-530 Sample provides schema of sample answers
QSL-540 Formula provides schema of math formula
QSL-550 Marker provides schema of marker and information
QSL-610 Auditing provides schema of auditing phase
QSL-620 Candidate provides schema of candidate and information
QSL-630 Proposer provides schema of proposer information
QSL-640 Auditor provides schema of auditor information
QSL-710 simpleType provides general simpleType definitions for reusing attribute values
6.2 Relations of Schemas
6.3 100-QSL

This schema is used for declaration of a specification, as well as the identifier for QSL structure editor. Defined any specification must define this schema. This schema is based on the extension of complex type QSLStructure.

6.4 110-System

This schema is used for specify the system, which provide services to do e-questionnaire, e-testing, and e-voting. It specifies the functions during each phase, and component information such as server, interface, database, device, etc.
schema is based on the complex type **SystemStructure**.

### 6.5 120-Service

![Service Diagram](image)

**Fig. 6-5-1** Service Diagram

This schema records the service consisting of lots of paper sheets, and settings for papers. Questioners and sponsors can directly use its specifications to define papers to respondents. It is the core component of the question bank for e-questionnaire and e-testing. This schema is based on complex type of **ServiceStructure**.
6.6 210-Phase

This schema is used to specify phases. All the child elements should be described by a sequence order. And the child elements are based on schema 310, 311, 312, 313, 314, 315, and 316. This schema is based on complex type of **PhaseStructure**.

For e-questionnaire, it is by the order of 310, 311, 312, 313, 314, 315, and 316.
For e-testing, it is by the order of 310, 311, 312, 313, 314, 315, 510, and 316.
For e-voting, it is by the order of 310, 311, 312, 313, 314, 315, 316 and 610.
6.7 220-Security

This schema is used to specify system security consisting of schema 350, 351, 352, and 353. This schema is based on the complex type `SecurityStructure`. 

**Fig. 6-7-1  Security Diagram**
6.8 230-Paper

This schema is used to specify paper sheet. There are 3 attributes. This schema is based on the complex type **PaperStructure**.
6.9 240-Setting

This schema is used to specify setting for corresponding paper sheet. It consists of schema 210. This schema is based on the complex type `SettingStructure`.

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier</td>
<td>setting001</td>
</tr>
<tr>
<td></td>
<td>ref</td>
<td>IdType</td>
<td>required</td>
<td>refer to paper</td>
<td>p001</td>
</tr>
</tbody>
</table>
6.10 250-Environment

Fig. 6-10-1 Environment Diagram

This schema is used to specify system environment consisting of schema 340, 342, 343, 344, and 345. This schema is based on the complex type EnvironmentStructure.
6.11 260-Participant

This schema is used to specify a list of participant roles. This schema is based on the complex type DataStructure. But there is a NominationGroup for defining nomination participants containing candidates and its proposers. That means, Proposer element (minOcc 1 and MaxOcc unbounded) is the child elements of Candidate element.
6.12 270-Data

This schema is used to specify data, which is divided into 2 groups for specifying the system data to define the database fields, and for specifying the service data to record the information. The groups are ordered in the choice relations. When the Data tag is the child of System tag, you can choose the schema for SystemDataGroup. Otherwise, you can choose another group. This schema is based on the complex type DataStructure.

6.13 280-Function

This schema is used to specify system functional requirements consisting of schema 210. This schema is based on the complex type FunctionStructure.
### 6.14 310-Settingup

![SettingUp Diagram](image)

<table>
<thead>
<tr>
<th>Element</th>
<th>Group</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>SettingUp</td>
<td>SettingUpFunctionPhaseGroup</td>
<td>belong to 280 schema</td>
</tr>
<tr>
<td></td>
<td>SettingPhaseGroup</td>
<td>belong to 240 schema</td>
</tr>
</tbody>
</table>

Fig. 6-14-1  SettingUp Diagram
This schema is used to specify the phase for preparing an event beginning. This schema is based on the complex type `SettingUpStructure`. Because all the phases are listed in **Function** and **Setting**, we defined each phase has two groups one for function definition, another for setting definition. And the two groups have the choice order relationship. That means, the users choose the set of elements in function group to define function requirements in this phase, and choose the set of elements in setting group to define setting requirements in this phase. The elements in function groups are explained in core schemas (schema 330, 331, 332, 333, 336, and 337) and some elements in setting groups are also defined in core schemas (schema 370, 371, 372, and 373), please refer to the corresponding schema definitions. There are 5 elements are not listed in schema files respectively, because they have the duplicated meanings with other tags.

### 6.14.1 AutoSaving

![AutoSaving Diagram](image)

**Fig. 6-14-2** AutoSaving Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoSaving</td>
<td>enable</td>
<td>YesNoType</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td></td>
<td>scoping</td>
<td>PhaseType</td>
<td>optional</td>
<td>usually submitting phase</td>
</tr>
<tr>
<td></td>
<td>closeTime</td>
<td>xs:nonNegativeInteger</td>
<td>optional</td>
<td>saveTime exclusive</td>
</tr>
<tr>
<td></td>
<td>saveTime</td>
<td>xs:nonNegativeInteger</td>
<td>optional</td>
<td>closeTime exclusive</td>
</tr>
</tbody>
</table>

This schema is used to define to automatically saving the answers when the respondents answering the questions before submitting their responses. The `saveTime` attribute means how many seconds record once. The `closeTime` means when to shut down this function.
6.14.2 Checking

![Checking Diagram](image)

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checking</td>
<td>enable</td>
<td>YesNoType</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td></td>
<td>scoping</td>
<td>PaperType</td>
<td>optional</td>
<td>value: answer</td>
</tr>
</tbody>
</table>

This element defines the spelling checking setting for providing conveniences for respondents to answer questions.

6.14.3 Distribute

![Distribute Diagram](image)

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribute</td>
<td>method</td>
<td>ChannelType</td>
<td>required</td>
<td>method</td>
</tr>
</tbody>
</table>

This schema is used to define the distributing method (channel). The users use this element to define that they distribute the paper to respondents in which kinds of channel. **This value must match with the distribute function during distributing phase.**
6.14.4 Analyze

![Analyze Diagram]

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyze</td>
<td>enable</td>
<td>YesNoType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>ReportType</td>
<td>optional</td>
<td>basic/trend</td>
</tr>
</tbody>
</table>

**Basic analysis**: provides to analyze by machine automatically.

(Removed) **Gap analysis**: provides to analyze for side-by-side matrix question type.

**Trend analysis**: provides a look at data over time for a long-running paper.

6.14.5 Interval

![Interval Diagram]

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval</td>
<td>enable</td>
<td>YesNoType</td>
<td>required</td>
<td>interrupt during the phase</td>
</tr>
<tr>
<td></td>
<td>frequency</td>
<td>xs:nonNegativeInteger &amp; xs:token (unbounded)</td>
<td>optional</td>
<td>if value=0 means enable=no if value=unbounded means no restraint</td>
</tr>
</tbody>
</table>

This schema is used to define the interrupt during the phases. The users use this element to define that they allow the respondent to stop and back to answer. The **frequency** attribute is used to record that the respondents are allow to interrupt how
many times.

### 6.15 311-Distributing

![Distributing Diagram](image1)

**Fig. 6-15-1** Distributing Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Group</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributing</td>
<td>DistributingFunctionPhaseGroup</td>
<td>belong to 280 schema</td>
</tr>
</tbody>
</table>

This schema is used to specify the phase for distributing the paper to respondents or to the web with some access limitation. This schema is based on the complex type **DistributingStructure**. Because all the phases are listed in **Function** and **Setting**, we are aimed to define each phase has two groups one for function definition, another for setting definition. But until now we are not define the settings during this phase. The elements in function groups are explained in core schemas (schema 330, 335, and 337), please refer to the corresponding schema definitions. There is an element that is not listed in schema file, because they have the duplicated meanings with other tags.

![Distribute Diagram](image2)

**Fig. 6-15-2** Distribute Diagram
This schema is used to define the distribute function. The users use this element to define that they distribute the paper to respondents by whom in which kinds of channel. **This value must match with the distribute function setting in setting up phase.** This schema is based on complex type of `DistributeStructure`.

### 6.16 312-Registering

![Registering Diagram](image)

**Fig. 6-16-1 Registering Diagram**

This schema is used to specify the phase for respondent registration. This schema is based on the complex type `RegisteringStructure`. Because all the phases are listed in **Function** and **Setting**, we are aimed to define each phase has two groups one for function definition, another for setting definition. But until now we are not define the settings during this phase. The elements in function groups are explained in core schemas (schema 330, 331, 336, and 337), please refer to the corresponding schema definitions. There are 2 elements that are not listed in schema files, because they have the duplicated meanings with other tags.
This schema is used to define the authenticate function to verify the eligible respondents. This value must match with the Authentication method defined in security requirements.

This schema is used to define the register function. Usually, when the users set everyone can access to the survey, so that there is no need to authenticate but need register respondents information to store in database sometimes.
6.17 313-Submitting

![Diagram of 313-Submitting]

**Fig. 6-17-1** Submitting Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Group</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submitting</td>
<td>SubmittingFunctionPhaseGroup</td>
<td>belong to 280 schema</td>
</tr>
</tbody>
</table>

This schema is used to specify the phase for respondents to submit their responses to store in database. This schema is based on the complex type `SubmittingStructure`. Because all the phases are listed in `Function` and `Setting`, we are aimed to define each phase has two groups one for function definition, another for setting definition. But until now we are not define the settings during this phase. The elements in function groups are explained in core schemas (schema 330, 331, 332, 333, 335, and 336), please refer to the corresponding schema definitions. There are 4 elements that are not listed in schema files, because they have the duplicated meanings with other tags.
### Observe Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observe</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>function Id marked as funcXXX</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>who can do this function</td>
</tr>
<tr>
<td></td>
<td>scope</td>
<td>derived by xs:string</td>
<td>required</td>
<td>which kinds of data will be monitored, value=state</td>
</tr>
</tbody>
</table>

This schema is used to define the monitor function. It is usually used for monitors. The monitors can monitor each respondent’s state such as the completeness of answering paper.

### Save Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>function Id marked as funcXXX</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>who can do this function</td>
</tr>
<tr>
<td></td>
<td>autoSave</td>
<td>YesNoType</td>
<td>required</td>
<td>automatically save the response</td>
</tr>
</tbody>
</table>
This schema is used to define the save function. Usually, when the respondents are answering the questions, the system can automatically save their responses if the questioners set up AutoSaving in Setting.

![Submit Diagram](image)

**Fig. 6-17-4** Submit Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submit</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>function Id marked as funcXXX</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>who can do this function</td>
</tr>
</tbody>
</table>

This schema is used to define the submit function. The system must provide the interface to submit their responses and save in database.

![Upload Diagram](image)

**Fig. 6-17-5** Upload Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upload</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>function Id marked as funcXXX</td>
</tr>
</tbody>
</table>
This schema is used to define the upload function. Some questions request the respondents to upload their responses.

6.18 314-Collecting

This schema is used to specify the phase for collecting responses. This schema is based on the complex type **CollectingStructure**. Because all the phases are listed in **Function** and **Setting**, we are aimed to define each phase has two groups one for function definition, another for setting definition. But until now we are not define the settings during this phase. The elements in function groups are explained in core schemas (schema 330, 335, and 336), please refer to the corresponding schema definitions. As the last child element, **Collect** is not listed in schema file, because they have the duplicated meanings with other tags.
This schema is used to define the collect function. The system should provide interface to respondents to collect data to store in database.

6.19 315-Analyzing

This schema is used to specify the phase for analyzing the collected responses. This schema is based on the complex type **AnalyzingStructure**. Because all the phases are listed in **Function** and **Setting**, we are aimed to define each phase has two groups one for function definition, another for setting definition. But until now we are not define the settings during this phase. The elements in function groups are explained in core schemas (schema 330 and 336), please refer to the corresponding schema definitions. As the last child element, **Analyze** is not listed in schema file, because they have the duplicated meanings with other tags.
This schema is used to define the analyze function. The system should provide interface to analyze response data to get a result.

### 6.20 316-Counting

This schema is used to specify the phase for counting to get the result. This schema is based on the complex type `CountingStructure`. Because all the phases are listed in `Function` and `Setting`. We are aimed to define each phase has two groups one for
function definition, another for setting definition. But until now we are not define the settings during this phase. The elements in function groups are explained in core schemas (schema 330 and 336), please refer to the corresponding schema definitions. As the last child element, Count and Report are not listed in schema files, because they have the duplicated meanings with other tags.

![Count Diagram](image1)

**Fig. 6-20-2** Count Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>function Id marked as funcXXX</td>
</tr>
</tbody>
</table>

This schema is used to define the count function. The system should provide interface to count response data to get a result.

![Report Diagram](image2)

**Fig. 6-20-3** Report Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>function Id marked as funcXXX</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>ReportType</td>
<td>required</td>
<td>basic/ trend</td>
</tr>
</tbody>
</table>

This schema is used to define the report function. The system should provide interface to report the result for normal or in trend.
### 6.21 320-Sponsor

**Fig. 6-21-1** Sponsor Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sponsor</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>sponsor001</td>
</tr>
<tr>
<td>Name</td>
<td>fullName</td>
<td>NameType</td>
<td>required</td>
<td>store in DB should be divided in first name and family name</td>
</tr>
<tr>
<td></td>
<td>usedName</td>
<td>NameType</td>
<td>optional</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>firstName</td>
<td>NameType</td>
<td>optional</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>familName</td>
<td>NameType</td>
<td>optional</td>
<td>-</td>
</tr>
<tr>
<td>Secrecy</td>
<td>type</td>
<td>AuthType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>value</td>
<td>xs:token</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td>Token</td>
<td>type</td>
<td>AuthType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>value</td>
<td>xs:token</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td>Biometrics</td>
<td>type</td>
<td>AuthType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>value</td>
<td>xs:token</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td>IP</td>
<td>-</td>
<td>derived by</td>
<td>required</td>
<td>[0-2][3).[0-9][3).[0-9][3].[0-9][3]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>xs:string</td>
<td></td>
<td>e.g. 10.0.0.52</td>
</tr>
<tr>
<td>Contract</td>
<td>address</td>
<td>AddressType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>tel</td>
<td>TelNumType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>phoneNum</td>
<td>TelNumType</td>
<td>optional</td>
<td>-</td>
</tr>
<tr>
<td>Affiliation</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>aff001</td>
</tr>
</tbody>
</table>
This schema is used to define the sponsor information. The schema is based on the complex type of `ParticipantInfoStructure`. Other roles of participants are also based on this complex type. If the security requirements demand to authenticate, then the role’s information should be recorded authentication method and its values. According to the setting about recording IP addresses, the value should be recorded in `IP` element as its content. In addition, some contract information and affiliation information should be recorded, if the developers defined the fields.

### 6.22 321-Questioner

![Questioner Diagram]

This schema is used to define the questioner information. The schema is based on the complex type of `ParticipantInfoStructure`. 
This schema is used to define the respondent information. The schema is based on the complex type of `ParticipantInfoStructure`.
6.24 323-Analyst

This schema is used to define the analyst information. The schema is based on the complex type of `ParticipantInfoStructure`.
6.25 324-Monitor

This schema is used to define the monitor information. The schema is based on the complex type of `ParticipantInfoStructure`.

![Monitor Diagram]

Fig. 6-25-1  Monitor Diagram
6.26 330-Export

This schema is used to define the export function. A participant can export a kind of data by a format, sometimes by a channel. There are some situations to use this function as following. The channel attribute is set for extension of QSL that is not only for web system but also machine.

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier; value=funcXXX</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>who can do this function, usually questioner</td>
</tr>
<tr>
<td></td>
<td>scope</td>
<td>DataType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>format</td>
<td>FormatType</td>
<td>required</td>
<td>usually .qsl suffix file</td>
</tr>
<tr>
<td></td>
<td>channel</td>
<td>ChannelType</td>
<td>optional</td>
<td>-</td>
</tr>
</tbody>
</table>

This is used to define the export function. A participant can export a kind of data by a format, sometimes by a channel. There are some situations to use this function as following. The channel attribute is set for extension of QSL that is not only for web system but also machine.
6.27 331-Import

This schema is used to define the import function. A participant can import a kind of data by a format, sometimes by a channel.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Phase</th>
<th>Data</th>
<th>Format</th>
<th>Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>questioner</td>
<td>setting</td>
<td>paper and setting, paper, setting</td>
<td>qsl</td>
<td>-</td>
</tr>
<tr>
<td>respondent</td>
<td>submitting</td>
<td>own responses (if allowed by settings)</td>
<td>qsl</td>
<td>-</td>
</tr>
<tr>
<td>analyst</td>
<td>analyzing</td>
<td>response</td>
<td>qsl</td>
<td>-</td>
</tr>
<tr>
<td>marker</td>
<td>marking</td>
<td>response</td>
<td>qsl</td>
<td>-</td>
</tr>
<tr>
<td>auditor</td>
<td>auditing</td>
<td>response, result</td>
<td>qsl</td>
<td>-</td>
</tr>
</tbody>
</table>
### 6.28 332-Launch

**Fig. 6-28-1**  Launch Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier; value=funcXXX</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>who can do this function</td>
</tr>
<tr>
<td></td>
<td>interval</td>
<td>YesNoType</td>
<td>required</td>
<td>match with setting</td>
</tr>
<tr>
<td></td>
<td>frequency</td>
<td>xs:nonNegativeInteger</td>
<td>required</td>
<td>value=0 means interval=no</td>
</tr>
</tbody>
</table>

This schema is used to define the launch function. Usually, a questioner can launch a paper into an active state for respondents. It is related with schema 333 **Stop** function.
6.29 333-Stop

**Fig. 6-29-1  Stop Diagram**

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier; value=funcXXX</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>who can do this function</td>
</tr>
<tr>
<td></td>
<td>goto</td>
<td>PhaseType</td>
<td>optional</td>
<td>when do this action, then go to</td>
</tr>
<tr>
<td></td>
<td>method</td>
<td>derived by xs:string</td>
<td>required</td>
<td>interval</td>
</tr>
<tr>
<td></td>
<td>frequency</td>
<td>xs: nonNegativeInteger</td>
<td>required</td>
<td>value=0 means interval=no</td>
</tr>
<tr>
<td></td>
<td>autoSave</td>
<td>YesNoType</td>
<td>required</td>
<td>-</td>
</tr>
</tbody>
</table>

This schema is used to define the stop launching function. Usually, a questioner can stop launching a paper into an inactive state for respondents, so that the respondents cannot answer anymore. The goto attribute is to specify the phase when doing this function. Usually, restarted state is going to the setting up phase, and thrown-away state is going to collecting phase.
6.30 334-Generate

This schema is used to define the generate function. Usually, a respondent login and register, so that he can get a registration ticket, usually in e-testing.
6.31 335-Ping

![Ping Diagram](image)

**Fig. 6-31-1 Ping Diagram**

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ping</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier; value=funcXXX</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>who can do this function; monitor</td>
</tr>
<tr>
<td></td>
<td>scope</td>
<td>DataType</td>
<td>required</td>
<td>IP</td>
</tr>
</tbody>
</table>

This schema is used to define the ping function. Usually, a monitor to monitor the respondents’ IP addresses.
6.32 336-Integrate

![Integrate Diagram]

**Fig. 6-32-1** Integrate Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrate</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier; value=funcXXX</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>who can do this function</td>
</tr>
<tr>
<td></td>
<td>scope</td>
<td>DataType</td>
<td>required</td>
<td>response</td>
</tr>
<tr>
<td></td>
<td>format</td>
<td>FormatType</td>
<td>required</td>
<td>zip</td>
</tr>
</tbody>
</table>

This schema is used to define the integrate function. Usually, an analyst can integrate the responses into a zip or qsl file and export it.
6.33 337-Remind

Remind Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remind</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier; value=funcXXX</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>who can do this function</td>
</tr>
<tr>
<td></td>
<td>value</td>
<td>ReminderType</td>
<td>required</td>
<td>notify data values</td>
</tr>
</tbody>
</table>

This schema is used to define the remind function. The questioner can remind respondents that they have not finished answering or submitting yet with a notification mails.
6.34 340-Server

Fig. 6-34-1 Server Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier</td>
<td>server001</td>
</tr>
<tr>
<td></td>
<td>purpose</td>
<td>PhaseType</td>
<td>required</td>
<td>the server for which phase</td>
<td>-</td>
</tr>
</tbody>
</table>

This schema is used to specify server information for environment. This schema is based on the extension of complex type ServerStructure. We designed it to provide its an interface.
### 6.35 341-Field

#### Field Diagram

![Field Diagram](image)

<table>
<thead>
<tr>
<th>Element</th>
<th>Element group</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParticipantFieldGroup</td>
<td>suggest users to create tables for each kind of roles</td>
<td></td>
</tr>
<tr>
<td>PaperFieldGroup</td>
<td>suggest users to create tables for its child elements</td>
<td></td>
</tr>
<tr>
<td>ResponseFieldGroup</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ResultFieldGroup</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

This schema is used to specify database fields for each table in database. This schema is based on the complex type **FieldStructure**.
6.35.1 Fields for Participant

This group is used to specify the fields for each role of participant in database. According to the phases and different services/systems, the optional elements may be created in database. All the roles have same attributes, so we defined the attribute group named “participantInfoAttriGroup”. We give an example about Sponsor element, others can be referred to this schema as following.
All the roles are used the attributes belonging to this attribute group. Except the **Proposer** element, it must to direct to the **Candidate** (nominee).

**Fig. 6-35-4  Proposer Diagram**
6.35.2 Fields for Paper

This group is used to specify the fields for paper, section, question, answer, and media in database. If the paper is set with media, then the users should create table for media in database. The 5 elements are in a sequence order.

Firstly, we defined the Paper element.

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper</td>
<td>id</td>
<td>FieldType</td>
<td>required</td>
<td>identifier</td>
</tr>
<tr>
<td></td>
<td>title</td>
<td>FieldType</td>
<td>required</td>
<td>paper title</td>
</tr>
<tr>
<td></td>
<td>description</td>
<td>FieldType</td>
<td>optional</td>
<td>paper subtitle</td>
</tr>
</tbody>
</table>
This schema is used to specify fields of paper table in database. Section, Question, and Answer elements refer to Paper.

![Diagram of Section, Question, and Answer](image)

**Fig. 6-35-7** Diagrams of Section, Question, and Answer

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media</td>
<td>id</td>
<td>FieldType</td>
<td>required</td>
<td>identifier</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>FieldType</td>
<td>required</td>
<td>media file suffix</td>
</tr>
<tr>
<td></td>
<td>source</td>
<td>FieldType</td>
<td>optional</td>
<td>media source</td>
</tr>
</tbody>
</table>
This schema is used to specify fields of media table in database.

### 6.35.3 Fields for Response

![Response Field Group Diagram](image)

**Fig. 6-35-9** Response Field Group Diagram

This group is used to specify the fields for respondents’ responses in database. The child element **Response** is created as a table in database.

![Response Diagram](image)

**Fig. 6-35-10** Response Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>id</td>
<td>FieldType</td>
<td>required</td>
<td>identifier</td>
</tr>
<tr>
<td></td>
<td>content</td>
<td>FieldType</td>
<td>required</td>
<td>response content for a question</td>
</tr>
<tr>
<td></td>
<td>respondentId</td>
<td>FieldType</td>
<td>required</td>
<td>respondent identifier</td>
</tr>
<tr>
<td></td>
<td>questionId</td>
<td>FieldType</td>
<td>required</td>
<td>question identifier</td>
</tr>
<tr>
<td></td>
<td>mediaId</td>
<td>FieldType</td>
<td>optional</td>
<td>media identifier</td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>FieldType</td>
<td>optional</td>
<td>it takes how many time to answer this question</td>
</tr>
</tbody>
</table>

This schema is used to specify fields of response table in database. One row data is a respondent to answer a question.
6.35.4 Fields for Result

This group is used to specify the fields for analyzed responses as a result in database. The child element Result is created as a table in database. When the survey are not closed, the real-time calculate the result may record into the database, but read the value of result should show it from calculating responses (not come from the result table, except close the survey can read from result database).

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result</td>
<td>id</td>
<td>FieldType</td>
<td>required</td>
<td>identifier</td>
</tr>
<tr>
<td></td>
<td>content</td>
<td>FieldType</td>
<td>required</td>
<td>response content for a question</td>
</tr>
<tr>
<td></td>
<td>questionId</td>
<td>FieldType</td>
<td>required</td>
<td>question identifier</td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>FieldType</td>
<td>optional</td>
<td>it takes how many time to answer this question</td>
</tr>
</tbody>
</table>

This schema is used to specify fields of result table in database. One row data is a total result for a question.
### 6.36 342-Gateway

![Gateway Diagram](image)

**Fig. 6-36-1** Gateway Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier</td>
</tr>
<tr>
<td></td>
<td>ref</td>
<td>IdType</td>
<td>optional</td>
<td>refer to device identifier</td>
</tr>
<tr>
<td></td>
<td>ip</td>
<td>derived by xs:string</td>
<td>required</td>
<td>IP address value</td>
</tr>
</tbody>
</table>

This schema is used to specify network of environment, which should refer to corresponding devices, and record its IP. We designed it to provide its an interface for extension. Maybe the name of this tag is not correctly (“network” is better). The ip attribute is restricted its value in a mode of `[0-2][3], [0-9][3], [0-9][3], [0-9][3]` to limit. This schema is based on complex type of `GatewayStructure`. 

---

*aqsl-v3.1-2018*

Specifications and Manual

AISE Lab, Saitama Univ., Japan
6.37 343-Interface

![Interface Diagram](image)

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier</td>
</tr>
<tr>
<td></td>
<td>ref</td>
<td>IdType</td>
<td>optional</td>
<td>refer to device/software/server identifier</td>
</tr>
</tbody>
</table>

This schema is used to specify interface of environment, which should refer to corresponding devices, software, or servers. We designed it to provide an interface for extension. This schema is based on complex type of `InterfaceStructure`.
## 6.38 344-Device

![Device Diagram](image)

**Fig. 6-38-1** Device Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute group</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>-</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>type</td>
<td>DeviceType</td>
<td>optional</td>
<td>device type</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>memory</td>
<td>xs:string</td>
<td>optional</td>
<td>8GB~</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cpu</td>
<td>xs:string</td>
<td>optional</td>
<td>inter core i3/5/7/8/10</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>ref</td>
<td>xs:NMToken</td>
<td>optional</td>
<td>refer to software identifier</td>
</tr>
<tr>
<td></td>
<td>access point</td>
<td>encryption</td>
<td>derived by xs:string</td>
<td>optional</td>
<td>WEP/WPA/WPA2/WPA Enterprise/WPA 2 Enterprise/others</td>
</tr>
</tbody>
</table>
This schema is used to specify device of environment. We designed it to provide an interface for extension. It is must to explain that there are 3 attribute groups for different devices. If the type attribute value is PC, then to select attributes in “PC attrGroup.” This schema is based on complex type of **DeviceStructure**.

### 6.39 345-Software

![Software Diagram](image)

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier</td>
</tr>
<tr>
<td></td>
<td>ref</td>
<td>IdType</td>
<td>required</td>
<td>refer to server identifier</td>
</tr>
<tr>
<td></td>
<td>purpose</td>
<td>derived by PhaseType</td>
<td>required</td>
<td>phase/ database/ os</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>optional</td>
<td>-</td>
</tr>
</tbody>
</table>

This schema is used to specify software of environment. We designed it to provide an other interface for extension. This schema is based on complex type of **SoftwareStructure**.
6.40 350-Anonymity

![Anonymity Diagram](image1)

**Fig. 6-40-1** Anonymity Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anonymity</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier for setting</td>
<td>anonymity001</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>usually for respondents</td>
<td>respondent</td>
</tr>
</tbody>
</table>

This schema is used to specify anonymous respondents to prevent any link between unencrypted responses. This schema is based on the complex type `AnonymityStructure`. As to its child element `Method`:

![Method Diagram](image2)

**Fig. 6-40-2** Diagram of Method for Anonymity

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>type</td>
<td>xs:string</td>
<td>required</td>
<td>methods for anonymity</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>phase</td>
<td>PhaseType</td>
<td>required</td>
<td>usually for submitting, and registering phase</td>
<td>submitting/registering</td>
</tr>
</tbody>
</table>

This schema is used to specify anonymous methods. You can use lots of methods for a system. The `Method` type is a `xs:string` enumeration facets, the value is limited as randomized auth token, blinded e-papers, blinded auth token, separation of duty, homomorphic secret, mix net, homomorphic encryption, HSM. As the values’
meaning and the corresponding phase are listed below.

<table>
<thead>
<tr>
<th>Method phase</th>
<th>Method type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting up</td>
<td>randomized auth token</td>
</tr>
<tr>
<td>Submitting</td>
<td>blind e-papers</td>
</tr>
<tr>
<td></td>
<td>blinded auth token</td>
</tr>
<tr>
<td></td>
<td>separation of duty</td>
</tr>
<tr>
<td></td>
<td>homomorphic secret</td>
</tr>
<tr>
<td>Counting</td>
<td>mix net</td>
</tr>
<tr>
<td></td>
<td>homomorphic encryption</td>
</tr>
<tr>
<td></td>
<td>HSM</td>
</tr>
</tbody>
</table>

**Blinded authenticated token**: e-voting system should provide to digitally authenticate a message without knowing the content of the message. The respondent sends a blinded anonymous authentication token (instead of the blinded e-paper) to the validator together with some identification and authentication data. The respondent receives a digital signature from the validator on this blinded token. In the next step, the respondent computes the value for the signed authentication token and sends his data together with his e-paper to the tallier, which accepts the respondent because of the digitally signed authentication token.

**Separation of duty**: a separation of duty approach (also works with at least two submitting servers). One inspecting the right to submit and another one storing the eligible e-papers. The respondent authenticates himself to the first server. In case that he has the right to submit, he receives a random number generated by this first server. This random number is also send to the second server but without any information about the respondent ID. Now the respondent uses this random number to authenticate himself as an eligible respondent to the second submitting server. Again this second server can only check whether an eligible respondent sent the e-paper but not who.

**Mix Net**: a mix net for anonymity as a cryptographic alternative to an anonymous channel. It secures who is communicating with whom and it secures the content of the transferred messages.

**HSM**: a Hardware Security Module (HSM), which is a tamper-resistant or at least tamper-evident hardware component that can securely generate and store long term secrets for use in cryptography. Generally, it is used to generate a digital key pair without revealing the private key. It can be a function that takes as the input the encrypted e-papers and returns as output the decrypted result, while the decrypted responses are not revealed.
6.41 351-Authentication

![Authentication Diagram]

**Fig. 6-41-1  Authentication Diagram**

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier for setting</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

This schema is used to specify authentication to ensure only eligible respondents can submit the responses before storing in database. It can also validate other roles of participants. This schema is based on the complex type `AuthenticationStructure`. There are 3 required elements but in a choice order. There are following possibilities to verify the users' identification.

<table>
<thead>
<tr>
<th>Method</th>
<th>Child Element</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>sth. you known</td>
<td>Secrecy</td>
<td>-</td>
</tr>
<tr>
<td>sth. you have</td>
<td>Token</td>
<td>-</td>
</tr>
<tr>
<td>sth. you are</td>
<td>Biometric</td>
<td>-</td>
</tr>
<tr>
<td>duplicated verification</td>
<td>Secrecy &amp; Token</td>
<td>usually used in e-testing to check the token of respondents and let them login in with their secrecy</td>
</tr>
<tr>
<td></td>
<td>Token &amp; Biometric</td>
<td>usually used in e-voting to double check respondent's token and login by their biometric</td>
</tr>
</tbody>
</table>

**Tips:** You can set lots of duplicated verification by using two elements.
6.41.1 Secrecy

This schema is used to specify secrecy (something you known) for authentication. This schema is based on the complex type SecrecyStructure. There are 2 required elements in a sequence order. There are 2 required child elements.

**Method** element:

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier</td>
<td>method001</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>AuthType</td>
<td>required</td>
<td>method</td>
<td>password</td>
</tr>
<tr>
<td></td>
<td>verifyTimes</td>
<td>xs:nonNegativeInteger</td>
<td>required</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>verifyPhase</td>
<td>PhaseType</td>
<td>required</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>status</td>
<td>xs:token</td>
<td>required</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This schema is used to specify secrecy methods. There are 5 required attributes. The status values are derived by restrictions, which are submitted, unsubmitted, issued, received, rejected, resubmitted, accepted, spoiled, exempted. **Method** type has values such as (for secrecy: password, randomized pwd, for token: ID card, job card, library card, roll card, for biometric: finger prints, iris, face recognition, DNA, others.)
**Channel** element: include schema 354.

### 6.41.2 Token

![Token Diagram](image)

This schema is used to specify token (something you have) for authentication. This schema is based on the complex type `TokenStructure`. There are 2 required elements in a sequence order. There are 2 required child elements. **Method** and **Channel** (schema 354) are included and shared among the **Authentication**.

### 6.41.3 Biometric

![Biometric Diagram](image)

This schema is used to specify biometric (something you are) for authentication. This schema is based on the complex type `TokenStructure`. There are 2 required elements in a sequence order. There are 2 required child elements. **Method** and **Channel** (schema 354) are included and shared among the **Authentication**.
6.42 352-Authority

This schema is used to specify participant who can access control to which phases and which functions, and which data. Except id attribute, the remaining 4 attributes belong to an attribute group named “roughAccessAttrType”. This schema is based on the complex type AuthorityStructure.
6.43 353-Seal

This schema is used to specify to encrypt what kinds of data. This schema is based on the complex type SealStructure. The Method element is as interface for extension.

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seal</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier</td>
<td>-</td>
</tr>
<tr>
<td>Seal</td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>to whom can encrypt</td>
<td>-</td>
</tr>
<tr>
<td>Seal</td>
<td>range</td>
<td>DataType</td>
<td>required</td>
<td>encrypt what</td>
<td>-</td>
</tr>
</tbody>
</table>

Fig. 6-43-1 Seal Diagram
6.44 354-Channel

![Channel Diagram](image)

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel</td>
<td>distribute</td>
<td>ChannelType</td>
<td>optional</td>
<td>method for distributing</td>
</tr>
<tr>
<td></td>
<td>verify</td>
<td>ChannelType</td>
<td>required</td>
<td>method</td>
</tr>
</tbody>
</table>

This schema is used to specify the medium through by which respondent can submit his response.
### 6.45 360-Section

This schema is used to specify section for a paper sheet. This schema is based on the complex type `SectionStructure`. There are 3 attributes listed in table, and a choice model to specify section title (required, no more than 5) and media (optional, no more than 5), and sequence model to specify questions.

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier</td>
</tr>
<tr>
<td></td>
<td>score</td>
<td>xs:positiveInteger</td>
<td>optional</td>
<td>section score for e-testing</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>QuestionType</td>
<td>optional</td>
<td>question types in this section are same or not, usually used for e-testing</td>
</tr>
</tbody>
</table>
6.46 361-Question

Fig. 6-46-1 Question Diagram
<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier; value: q001</td>
</tr>
<tr>
<td>Question</td>
<td>type</td>
<td>QuestionType</td>
<td>required</td>
<td>if section type chosen, then question type is similar with that</td>
</tr>
<tr>
<td></td>
<td>isMandatory</td>
<td>YesNoType</td>
<td>required</td>
<td>* necessary to answer</td>
</tr>
<tr>
<td></td>
<td>score</td>
<td>xs:positiveInteger</td>
<td>optional</td>
<td>question total score</td>
</tr>
<tr>
<td></td>
<td>textAlign</td>
<td>derived by xs:string</td>
<td>optional</td>
<td>alignment for prefix and suffix (alignment relation between description of question and text area of answer)</td>
</tr>
</tbody>
</table>

This schema is used to specify question for a paper sheet. This schema is based on the complex type `QuestionStructure`. There are 5 attributes listed in table, and a choice model to specify question title (required, no more than 5) and media (optional, no more than 5), and sequence model to specify answers.
6.47 362-Answer

Fig. 6-47-1 Answer Diagram
<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>answer id: anXXX</td>
</tr>
<tr>
<td></td>
<td>isDefault</td>
<td>YesNoType</td>
<td>optional</td>
<td>sample answer for e-testing;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>isNAOption &amp; isOtherField exclusive</td>
</tr>
<tr>
<td></td>
<td>isNAOption</td>
<td>YesNoType</td>
<td>optional</td>
<td>isNAOption exclusive</td>
</tr>
<tr>
<td></td>
<td>isOtherField</td>
<td>YesNoType</td>
<td>optional</td>
<td>isOtherField exclusive</td>
</tr>
<tr>
<td></td>
<td>Column.Grid</td>
<td>union:</td>
<td>optional</td>
<td>for 2D matrix</td>
</tr>
<tr>
<td></td>
<td>Row.Grid</td>
<td>xs:nonNegativeInteger</td>
<td>optional</td>
<td>suffix from 0</td>
</tr>
<tr>
<td></td>
<td>X.Grid</td>
<td>veInteger</td>
<td>optional</td>
<td>for 3D matrix</td>
</tr>
<tr>
<td></td>
<td>Y.Grid</td>
<td>&amp;xs:token</td>
<td>optional</td>
<td>(matrix+drop-down list)</td>
</tr>
<tr>
<td></td>
<td>Z.Grid</td>
<td>(enumeration value: *)</td>
<td>optional</td>
<td>suffix from 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>token for piping and</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>extraction logic types</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>weight</td>
<td>optional</td>
<td>weight affects the results</td>
</tr>
<tr>
<td></td>
<td></td>
<td>xs:double</td>
<td>optional</td>
<td>answer score rules</td>
</tr>
<tr>
<td></td>
<td></td>
<td>defaultValue</td>
<td>optional</td>
<td>text and ranking sample</td>
</tr>
<tr>
<td></td>
<td></td>
<td>xs:token</td>
<td>optional</td>
<td>for ranking question type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>derived by</td>
<td>optional</td>
<td>text area size (pixel*pixel)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>xs:string</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>isDragged</td>
<td>YesNoType</td>
<td>optional</td>
<td>provide relationship of hierarchies of data such ad zip code and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>corresponding addresses</td>
</tr>
<tr>
<td></td>
<td>isDynamic2Next</td>
<td>YesNoType</td>
<td>optional</td>
<td></td>
</tr>
</tbody>
</table>

This schema is used to specify an answer. This schema is based on the complex type **AnswerStructure**. **Limitation** helps to limit the text characteristics, and options numbers. **Formula** (schema 540) helps to define the math, chemical, and other interfaces.
6.48 363-Description

Fig. 6-48-1 Description Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>text identifier</td>
<td>d001</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>DescriptionType</td>
<td>required</td>
<td>distinguish title, subtitle, and summary</td>
<td>title</td>
</tr>
<tr>
<td></td>
<td>value</td>
<td>xs:string</td>
<td>required</td>
<td>contents</td>
<td>-</td>
</tr>
</tbody>
</table>

This schema is used to specify contents for paper, section, question, and answer. It has 3 attributes. Usually, the description type chooses title as its value. The description value should be stored in database. This schema is based on the complex type DescriptionStructure.

6.49 364-Media

Fig. 6-49-1 Media Diagram
<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier</td>
<td>m001</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>MediaType</td>
<td>required</td>
<td>media file suffix</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>src</td>
<td>xs:anyURI</td>
<td>required</td>
<td>roots</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>tips</td>
<td>xs:string</td>
<td>required</td>
<td>mouse hovered the media</td>
<td>xxx.jpg</td>
</tr>
</tbody>
</table>

This schema is used to specify contents for paper, section, question, and answer. It has 3 attributes. Usually, the description type chooses title as its value. The description value should be stored in database. This schema is based on the complex type **DescriptionStructure**.

### 6.50 365-Alignment

This specified the arrangement of answer and questions. It is an optional child element of **Question**. Considering questioner could set arrangement of questions and answers (especially for selection question type). These attributes refer to xAML.

#### Alignment Diagram

This specified the arrangement of answer and questions. It is an optional child element of **Question**. Considering questioner could set arrangement of questions and answers (especially for selection question type). These attributes refer to xAML.
6.51 366-Limitation

![Limitation Diagram](image)

**Fig. 6-51-1** Limitation Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>minOccur</td>
<td>xs:nonNegativeInteger</td>
<td>Optional</td>
<td>min and max occur times min=1&amp;&amp;max=1 (only select one) [selection question types]</td>
</tr>
<tr>
<td>Limitation</td>
<td>maxOccur</td>
<td>xs:nonNegativeInteger</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Limitation</td>
<td>minLength</td>
<td>xs:nonNegativeInteger</td>
<td></td>
<td>min and max length for text area [text question types]</td>
</tr>
<tr>
<td>Limitation</td>
<td>maxLength</td>
<td>xs:nonNegativeInteger</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Limitation</td>
<td>format</td>
<td>union: FieldType, xs:anyURI, EmailType, xs:string</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This schema is used to specify limitation for questions and answers.
6.52 370-Language

![Language Diagram](image)

**Fig. 6-52-1** Language Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>original</td>
<td>LanguageType</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td></td>
<td>to</td>
<td>LanguageType</td>
<td>optional</td>
<td>translate to</td>
</tr>
<tr>
<td>scoping</td>
<td></td>
<td>Union: PaperType&amp; DataType&amp; SystemType</td>
<td>optional</td>
<td>if need translate; value: system, paper, response, report</td>
</tr>
</tbody>
</table>

This element defines the original and translation language in which scope.
### 6.53 371-Time

![Diagram of 371-Time](image)

#### Fig. 6-53-1  Time Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>enable</td>
<td>YesNoType</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td></td>
<td>isShowCountdown</td>
<td>YesNoType</td>
<td>optional</td>
<td>fixed value no</td>
</tr>
<tr>
<td></td>
<td>zone</td>
<td>derived by xs:string</td>
<td>required</td>
<td>UTC+9</td>
</tr>
<tr>
<td></td>
<td>start</td>
<td>TimeType</td>
<td>optional</td>
<td>during exclusive</td>
</tr>
<tr>
<td></td>
<td>end</td>
<td>TimeType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>during</td>
<td>xs:double</td>
<td>optional</td>
<td>start/end exclusive</td>
</tr>
<tr>
<td></td>
<td>scoping</td>
<td>union: PhaseType &amp;PaperType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td>paperTimeSettingGroup</td>
<td>sectionId</td>
<td>IdType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>questionId</td>
<td>IdType</td>
<td>optional</td>
<td></td>
</tr>
</tbody>
</table>
### 6.54 372-Number

![Number Diagram](image)

**Fig. 6-54-1  Number Diagram**

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>enable</td>
<td>YesNoType</td>
<td>required</td>
<td>yes means default ordering</td>
</tr>
<tr>
<td></td>
<td>isShowNumbering</td>
<td>YesNoType</td>
<td>optional</td>
<td>number ascending ordering</td>
</tr>
<tr>
<td></td>
<td>order</td>
<td>OrderType</td>
<td>optional</td>
<td>default: ascending</td>
</tr>
<tr>
<td></td>
<td>rule</td>
<td>LimititationType</td>
<td>optional</td>
<td>a-z/ finger</td>
</tr>
<tr>
<td></td>
<td>scoping</td>
<td>union: PaperType &amp; Data</td>
<td>optional</td>
<td>value: paper, section, question, answer, response, report</td>
</tr>
<tr>
<td></td>
<td>startNum</td>
<td>xs:nonNegativeInteger</td>
<td>optional</td>
<td>0~</td>
</tr>
</tbody>
</table>

This schema is used to specify the ordering setting. The `isShowNumbering` attribute is to fixedly shown the fingers by an ascending order.
6.55 373-Quota

Fig. 6-55-1 Quota Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quota</td>
<td>enable</td>
<td>YesNoType</td>
<td>required</td>
<td>limit how much percents of respondents can submit</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>derived by xs:string</td>
<td>optional</td>
<td>rate or number</td>
</tr>
<tr>
<td></td>
<td>scoping</td>
<td>union: ParticipantType &amp;DataType</td>
<td>optional</td>
<td>value: respondent, response</td>
</tr>
<tr>
<td></td>
<td>value</td>
<td>xs:string</td>
<td>optional</td>
<td>-</td>
</tr>
</tbody>
</table>

This schema is used to specify the quota limitation for setting the numbers of respondent to access. There are two kinds of quota types to limit the respondents. It is the `intersection` relation with the authentication methods.
6.56 374-Response

This schema is used to specify response information recorded in database. It has 3 children elements. This schema is based on the complex type `ResponseStructure`. If the users design an e-testing system or service, it should record the score (schema 520) and sample answer (schema 530).
### 6.57 375-Result

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reply</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier</td>
</tr>
<tr>
<td></td>
<td>respondentId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>paperId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>sectionId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>questionId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>answerId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>value</td>
<td>xs:string</td>
<td>required</td>
<td>record value not option no.</td>
</tr>
</tbody>
</table>

**Fig. 6-57-1** Result Diagram
<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistic</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>identifier</td>
</tr>
<tr>
<td></td>
<td>paperId</td>
<td>IdType</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sectionId</td>
<td>IdType</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td></td>
<td>questionId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>answerId</td>
<td>IdType</td>
<td>optional</td>
<td>necessary for ratio</td>
</tr>
<tr>
<td>Ratio</td>
<td>selectedNum</td>
<td>xs:nonNegativeInteger</td>
<td>required</td>
<td>selected&lt;= respondent</td>
</tr>
<tr>
<td></td>
<td>respondentNum</td>
<td>xs:nonNegativeInteger</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>sampleNum</td>
<td>xs:nonNegativeInteger</td>
<td>required</td>
<td>for e-testing; sample&lt;= respondent</td>
</tr>
<tr>
<td></td>
<td>respondentNum</td>
<td>xs:nonNegativeInteger</td>
<td>required</td>
<td></td>
</tr>
</tbody>
</table>

**Result** schema is used to define the report data in details. Its child element **Statistic** used to specify a lots of report style. It determines by setting for result types. This provides the reference to detailed answers identifier, and provides the **Ratio** to completeness and **Accuracy** for e-testing. Both child elements of statistic are in a choice order.
6.58 410-Logic

Fig. 6-58-1 Logic Diagram
<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>LogicType</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>Condition</td>
<td>paperId</td>
<td>IdType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sectionId</td>
<td>IdType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>questionId</td>
<td>IdType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>answerId</td>
<td>IdType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>isChecked</td>
<td>YesNoType</td>
<td>optional</td>
<td>condition if question type selection is selected, then..</td>
</tr>
<tr>
<td></td>
<td>relation</td>
<td>restriction of xs:token</td>
<td>optional</td>
<td>conditions relations: &amp;&amp;,</td>
</tr>
<tr>
<td></td>
<td>score</td>
<td>xs:double</td>
<td>optional</td>
<td>&gt;=score value</td>
</tr>
<tr>
<td>Action</td>
<td>paperId</td>
<td>IdType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sectionId</td>
<td>IdType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>questionId</td>
<td>IdType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>answerId</td>
<td>IdType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paperId</td>
<td>IdType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>isFilled</td>
<td>YesNoType</td>
<td>optional</td>
<td>for jump to conclusion</td>
</tr>
<tr>
<td></td>
<td>descriptionId</td>
<td>IdType</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>to</td>
<td>xs:anyURI&amp; restriction of xs:token pattern: &quot;S('pip[0-9]<em>'), S('ext[0-9]</em>')&quot;</td>
<td>optional</td>
<td>to url or piping! extract content</td>
</tr>
</tbody>
</table>

**Logic** has a child element Route to record the routes of the logic from which **Condition** and go to which **Action**. Much more logic types and the templates for guiding how to use it in details are listed in **Chapter 8**.
### 6.59 510-Marking

This schema is used to specify the phase for mark the responses in e-testing. This schema is based on the complex type `MarkingStructure`. Because all the phases are listed in **Function** and **Setting**, we are aimed to define each phase has two groups one for function definition, another for setting definition. But until now we are not define the settings during this phase. The element in function group is explained in core schemas (schema 330), please refer to the corresponding schema definition. As the last child element, **Mark** is not listed in schema file, because it has the duplicated meanings with other tags.

#### 6.59-1 Marking Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Group</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marking</td>
<td>MarkingFunctionPhaseGroup</td>
<td>belong to 280 schema</td>
</tr>
</tbody>
</table>

This schema is used to define the mark function. The system should provide interface to mark response data to get a score.

#### 6.59-2 Mark Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Use</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>function Id marked as funcXXX</td>
</tr>
<tr>
<td></td>
<td>role</td>
<td>ParticipantType</td>
<td>required</td>
<td>usually “monitor”</td>
</tr>
</tbody>
</table>
### 6.60  520-Score

![Score Diagram](image)

**Fig. 6-60-1** Score Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Used</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>respondentId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>paperId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>sectionId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>questionId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>answerId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>value</td>
<td>union: xs:double&amp; xs:token</td>
<td>required</td>
<td>total score for a paper by a respondent</td>
</tr>
</tbody>
</table>

Schema 520 provides the specification for score data in details. As to the score setting rule is specified in paper sheet.
### 6.61 530-Sample

![Sample Diagram](image)

**Fig. 6-60-1** Sample Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Used</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample</td>
<td>id</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>questionId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>paperId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>sectionId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>questionId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>answerId</td>
<td>IdType</td>
<td>required</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>value</td>
<td>xs:token</td>
<td>required</td>
<td>sample answer</td>
</tr>
</tbody>
</table>

Schema 530 provides the specification for sample answer and its questioner.
6.62 540-Formula

![Formula Diagram](image)

**Fig. 6-62-1** Formula Diagram

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Type</th>
<th>Used</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>type</td>
<td>derived by xs:string</td>
<td>required</td>
<td>math, chemistry, physic, programing, biology, other</td>
</tr>
</tbody>
</table>

Schema 540 provides the skeleton of specification for formula about fields of math, chemistry, physic, programing, biology, other.
6.63 550-Marker

This schema is used to define the marker information. The schema is based on the complex type of `ParticipantInfoStructure`.

6.64 610-Auditing
This schema is used to specify the phase for audit the result with respondent numbers and responses in e-voting. This schema is based on the complex type **AuditingStructure**. Because all the phases are listed in **Function** and **Setting**, we are aimed to define each phase has two groups one for function definition, another for setting definition. But until now we are not define the settings during this phase. The elements in function group are explained in core schemas (schema 330 and 336), please refer to the corresponding schema definitions. As the last child element, **Audit** is not listed in schema file, because it has the duplicated meanings with other tags.

This schema is used to define the audit function. The system should provide interface to audit result is eligible or not.
6.65 620-Candidate

This schema is used to define the candidate information. The schema is based on the complex type of `ParticipantInfoStructure`. In addition, the element has a (the last) child element named Proposer (schema 630).

![Candidate Diagram](image-url)
6.66 630-Proposer

This schema is used to define the Proposer information. The schema is based on the complex type of `ParticipantInfoStructure`.

![Proposer Diagram](image)

Fig. 6-66-1  Proposer Diagram
6.67 640-Auditor

![Auditor Diagram](image)

This schema is used to define the auditor information. The schema is based on the complex type of `ParticipantInfoStructure`.
7 Question Types Reference Guide

The whole things of QSL: QSL structure, QSL Schema, elements, attributes, simple types, and complex types are listed previously. In this chapter, we will see how we can use the important but difficult parts of QSL usage: the question types which determine a complete e-questionnaire, and an e-testing. The reason is that an e-questionnaire or an e-testing is mainly composed of some questions. Although many e-questionnaire and e-testing systems provide widely variety of question types particularly in the various classification methods, we concluded four basic types and aimed at gaining various question types though combination of basic types. We generally talked about the type attribute of question element in previous chapter. Not enough is known about type attribute to use it well. Therefore, we will deeply illustrate the question types and how to use them.

This chapter provides a reference to all of question types as follow:
• Basic question types;
• Possible combinations;

All content and diagrams in this chapter refer to QuestionPro.

7.1 Basic Question Types

Four basic question types are listed as follow:
• Multiple Choice;
• Open-ended Text;
• Matrix;
• Ranking;

7.1.1 Multiple Choice

What is multiple choice question type?
Multiple Choice is tacit recognition as a radio button or a checkbox that that allows the respondents to choose only one or multiple selections from a predefined set of options. In general, there are two types of multiple choice question type, which are single choice and multiple choices. Some screenshots help to deeply understand it.
Fig. 7-1 Screenshots of Radio Button Single Choice

Fig. 7-2 Screenshots of Drop-down Single Choice

Fig. 7-3 Screenshots of Checkbox Multiple Choices

How to distinguish single choice and multiple choices?
With Designing QSL based on combination of primitive elements, both single choice and multiple choices are the multiple choices question type (Drop-down list belongs to single choice). We distinguish both of them by Limitation¹.

How to set up multiple choice question type?

¹Bold contents in this chapter are defined as QSL elements.
Step 1: to choose question type for Question, which has an attribute type and to choose its value “selection”.

Step 2: to distinguish the single choice and multiple choices, and to control it by Limitation.

- Tips 1: As to a radio button single choice type, questioner should choose the Limitation minOccur attribute value as “1”, macOccur attribute value as “1”, which means to restrict respondent to select only one answer.

- Tips 2: As to a checkbox multiple choices type, questioner should choose the Limitation minOccur attribute value more than 1, and macOccur attribute value less than total number of answers, which means to restrict respondent to select multiple answers more than one less than total number of answers.

Step 3: to set up Answer, which has its child element Description, which should be written the content of each answer between start-tag and end-tag of text.

As an example to let you deeply understand how to set up a multiple choice question type (radio button single choice),

```xml
<Question isMandatory="yes" type="selection" id="q0001">
  <Limitation minOccur="1" maxOccur="1"></Limitation>
  <Description type="paragraph" value="Please selected credit card u prefer most" id="de0001"/>
  <Answer id="a0001">
    <Description type="paragraph" value="Visa" id="de0002"/>
  </Answer>
  <Answer id="a0002">
    <Description type="paragraph" value="Visa" id="de0003"/>
  </Answer>
  <Answer id="a0003">
    <Description type="paragraph" value="Visa" id="de0004"/>
  </Answer>
  <Answer id="a0004">
    <Description type="paragraph" value="Visa" id="de0005"/>
  </Answer>
</Question>
```

To distinguish the arrangement of Fig. 7-1 and Fig. 7-2, we can set up Alignment element.

```xml
<Question isMandatory="yes" type="selection" id="q0001">
  <Limitation minOccur="1" maxOccur="1"></Limitation>
  <Description type="paragraph" value="Please selected credit card u prefer most" id="de0001"/>
  <Alignment vertical="center"/>
  <Answer id="a0001">...
```

1 Italics content in this chapter are defined as QSL attributes.
As another example to let you deeply understand how to set up a multiple choice question type (checkbox multiple choices),

<Limitation minOccur="1" maxOccur="4"></Limitation>

<!--at least choose 1 option and at most choose 4 options-->

<Limitation minOccur="3" maxOccur="3"></Limitation>

<!--Exactly choose 3 options-->

To distinguish the arrangement of Fig.7-3, we can set up **Alignment**.

**Attention**: there is no need to set up a drop-down list for a checkbox selection.

<Alignment horizontal="left"/>

There is a special question type named "True-False". The template snippet is shown below.

<Question isMandatory="yes" type="selection" id="q0001">
  <Limitation minOccur="1" maxOccur="1"></Limitation>
  <Description type="paragraph" value="True-False" id="de0001">
    <Description type="paragraph" value="true" id="de0002"/>
  </Description>
  <Answer id="a0001"/>
  <Answer id="a0002" isDefault="yes">
    <!--sample answer-->
    <Description type="paragraph" value="false" id="de0003"/>
  </Answer>
  <Answer id="a0002" isOtherField="a0003" score="-0.5">
    <!--allow respondent to add a other answer, usually in e-questionnaire -->
    <!--if respondent adds other answer, then minus 0.5 score-->
    <Description type="paragraph" value="other" id="de0004"/>
  </Answer>
</Question>

### 7.1.2 Open-ended Text

**What is open-ended text question type?**

**Open-ended Text** is a text type question where respondents can input long text, single row text, numeric text, or an email address.

**How to distinguish long text, single row text, numeric text, email address?**

Because Design of QSL is based on combination of primitive elements, so we proposed that all of these types are the Open-ended Text Question Type. To help understand deeply, Fig. 7-4, Fig. 7-5, Fig. 7-6, and Fig. 7-7 are presented below.
Questioners can use **Limitation** element to restrict.

**How to set up open-ended text question type?**

At first, questioner should choose the question type for the **Question** element, which has an attribute type. Questioner should choose the type attribute value “text”.

**Attention:** If the character text is presented up above the input box, please write in the **Description** element of **Question** element. If the character text is presented in placeholder of the input box, please write in the **Description** element of **Answer** element.

In addition, to set up the **Answer** element, **Answer** element has its child element **Description**.

At last, to distinguish the long text, single row text, numeric text, and email address, questioner can control it through **Limitation** element. **However**, it is different with **Multiple Choice Question Type**, the **Limitation** element is as a child element of **Answer** element.

- As to a long text type (developer should set up the default size as 80px×320px), questioner should choose the limitation attribute **minLength** value and **maxLength** value less than ”255” as default.
- As to a single row text type (developer should set up the default size as 400px, but height cannot be increased), questioner should choose the **Limitation** **minLength** value and **maxLength** value less than ”255” as
• As to a numeric text type (developer should set up the default size as 80px, but height cannot be increased), questioner should choose the **Limitation** attribute **minLength** value and **maxLength** value less than "255" as default.

• As to an email address type (developer should set up the default size as 320px, but height cannot be increased), questioner should choose the **Limitation** attribute. In common, the most attribute value and the **minLength** attribute are not necessary.

• There are 2 **Limitation** locations, first, as the **Question**'s child element, second, as the **Answer**'s child element. The first one in the outer layer to control line numbers. The second one in the inner layer to control character numbers, and the **format** value to control content format.

As an example to let you deeply understand how to set up a long text type,

```xml
<Question isMandatory="yes" type="text" id="q0001">
  <Limitation minLength="0" maxLength="5"/>
  <!--outer layer: limitation control line numbers -->
  <Description type="paragraph" value="Please provide additional information" id="de0001"/>
  <Answer id="a0001">
    <Limitation minLength="5" maxLength="255" format="long text"/>
    <Description type="paragraph" value="" id="de0002"/>
    <!--inner layer: limitation control character numbers -->
  </Answer>
</Question>
```

As an example to let you deeply understand how to set up a single row text type,

```xml
<Question isMandatory="yes" type="text" id="q0001">
  <Limitation minLength="1" maxLength="1"/>
  <!--outer layer: limitation control line numbers -->
  <Description type="paragraph" value="Name:" id="de0001"/>
  <Answer id="a0001">
    <Limitation minLength="2" maxLength="129" format="char"/>
    <Description type="paragraph" value="" id="de0002"/>
    <!--inner layer: limitation control character numbers -->
  </Answer>
</Question>
```

As an example to let you deeply understand how to set up a numeric input type,

```xml
<Question isMandatory="yes" type="text" id="q0001">
  <Limitation minLength="1" maxLength="1"/>
  <!--outer layer: limitation control line numbers -->
  <Description type="paragraph" value="Phone Number:" id="de0001"/>
  <Answer id="a0001">
    <Limitation minLength="5" maxLength="25" format="int"/>
  </Answer>
</Question>
```
As an example to let you deeply understand how to set up an email address type,

```html
<Question isMandatory="yes" type="text" id="q0001">
    <Limitation minLength="1" maxLength="1"/>
    <Description type="paragraph" value="Email:" id="de0001"/>
    <Answer id="a0001">
        <Limitation minLength="10" maxLength="129" format="char"/>
        <Description type="paragraph" value="Email: $a0001" id="de0002"/>
    </Answer>
</Question>
```

There is a special question type named "fill in the blank". The usual presentation in HTML is:

```html
<p>my name is <input type="text">, I am a student</p>
```

The template snippet is shown below. It concerns the piping logic type and its grammar.

```html
<Question isMandatory="yes" type="text" id="q0001">
    <Limitation minLength="1" maxLength="1"/>
    <Description type="paragraph" value="My name is $a0001', I am a student" id="de0001"/>
    <Answer id="a0001">
        <Limitation minLength="2" maxLength="5" format="char"/>
        <Description type="paragraph" value="My name is $a0001', I am a student" id="de0002"/>
    </Answer>
</Question>
```

### 7.1.3 Matrix

**What is matrix question type?**

**Matrix Question Type** is a series of questions that share the same answer choices. There are two types, which are one selection type and many selections type.

**How to distinguish one selection type and many selections type?**

**One Selection Type:** This question can be used when we need to group questions that have the same answer option scale. Consolidated reporting can be done on Matrix questions (see Fig. 7-8).
**Many Selection Type:** Matrix question where respondents can select multiple options (see Fig. 7-9).

**Fig. 7-8** Screenshot of One Selection Type

**Fig. 7-9** Screenshot of Many Selections Type

**How to set up a matrix question type?**
At first, questioner should choose the question type for the `Question` element, which has an attribute “type”. Questioner should choose the type attribute value “matrix”.

In addition, to distinguish the one selection and many selections, questioner can control it through `Limitation` element:

- As to a one selection type, questioner should choose the `Limitation maxOccur` attribute value as “1”, `minOccur` attribute value as ”1”, which means to restrict respondent to select only one option in each row.
- As to a checkbox multiple choices type, questioner should define the `Limitation maxOccur` attribute value not less than 1, `minOccur` attribute value less than total number of answers, which means to restrict respondent to select multiple answers more than one less than total number of options in each row.

At last, to set up the `Answer` element, `Answer` element has its child element `Description`, which should be written the `value`. In addition, its attributes `Column.Grid` and `Row.Grid` to control the descriptions of column and row.

As an example to let you deeply understand how to set up one selection type,

```xml
<Question isMandatory="yes" type="matrix" id="q0001">
  <Limitation minOccur="1" maxOccur="1"/>
  <Description type="paragraph" value="Please rate the following" id="de0001"/>
  <!--Column-->
  <Answer id="a0001" Column.Grid="0"/>
  <!--Grid begins from 0 -->
```
As an example to let you deeply understand how to set up many selections type, it only need to change the Limitation's attributes.

There is a special example that is a Not Applicable Option.

![Screenshot of N/A Option in Matrix](image)

Fig. 7-10 Screenshot of N/A Option in Matrix

7.1.4 Ranking

What is ranking question type?

Ranking Question Type allows a certain set of brands or products to be ranked based upon a specific attribute or characteristic. In common, some e-questionnaire

```xml
<Description type="paragraph" value="Poor" id="de0002"/>
</Answer>

<Answer id="a0002" Column.Grid="1">
  <Description type="paragraph" value="Below Average" id="de0003"/>
</Answer>
...
</Answer id="a0003" Column.Grid="0">
  <Description type="paragraph" value="Product" id="de0004"/>
</Answer>
<Answer id="a0004" Column.Grid="1">
  <Description type="paragraph" value="Service" id="de0005"/>
</Answer>
<Answer id="a0005" Column.Grid="2">
  <Description type="paragraph" value="Support" id="de0006"/>
</Answer>
</Question>

7.1.4 Ranking

What is ranking question type?

Ranking Question Type allows a certain set of brands or products to be ranked based upon a specific attribute or characteristic. In common, some e-questionnaire
```
systems present rank order type (see Fig. 7-11), which is similar with numeric text type to enter the order number.

![Screenshot of Ranking Question Type](image)

**Fig. 7-11** Screenshot of Ranking Question Type

**How to set up a ranking question type?**
At first, questioner should choose the question type for the Question element, which has an attribute “type”. Questioner should choose the type attribute value “ranking”.

**Attention:** If the value is selected as “ranking” that means the order number cannot be repeated and from 1 to the number of total options. In addition, questioner can use Limitation element to restrict maxOccur and minOccur.

Furthermore, to set up the Answer element, answer element has its child element text, which should be written the content of each answer between start-tag and end-tag of text.

At last, do not forget to give it a restriction for its Alignment and set up vertical.

As an example to let you deeply understand how to set up one selection type,

```xml
<Question isMandatory="yes" type="ranking" id="q0001">
  <Limitation minLength="1" maxLength="5"/>
  <Description type="paragraph" value="Please rank the follow actors:" id="de0001"/>
  <Alignment vertical="stretch"/>
  <Answer id="a0001">
    <Description type="paragraph" value="Johnny Depp" id="de0002"/>
  </Answer>
  <Answer id="a0002">
    <Description type="paragraph" value="Will Smith" id="de0003"/>
  </Answer>
  <Answer id="a0003">
    <Description type="paragraph" value="Leonardo DiCaprio" id="de0004"/>
  </Answer>
  <Answer id="a0004">
    <Description type="paragraph" value="Brad Pitt" id="de0005"/>
  </Answer>
  <Answer id="a0005">
    <Description type="paragraph" value="George Clooney" id="de0006"/>
  </Answer>
</Question>
```
However, there is a difficult that is how to set the sample answer for the e-testing. Even we always set the sample in a Boolean type, but in ranking situation, the sample should be a order result. Therefore,

```xml
<Question isMandatory="yes" type="ranking" id="q0001">
  ...
  <Answer id="a0005" defaultValue="4">
    <Description type="paragraph" value="George Clooney" id="de0006"/>
  </Answer>
</Question>
```

### 7.2 Possible Combinations of Question Types

After we explained the basic question types, we will present the possible combinations of basic question types. In this sub-chapter, we will describe the possible combinations as such examples.

At first, we explain some common possible combination question types.

#### 7.2.1 Contact Information

**What is Contact Information?**

Contact Information adds in a properly formatted and consolidated question asking users for their contact information (see Fig. 7-12). It is similar with open-ended text question type, like a set of text questions.

![Fig. 7-12 Screenshot of Contact Information](image)

**How to set up a contact information question type?**

At first, questioner should choose the question type for the **Question** element, which
has an attribute “type”. Questioner should choose the type attribute value “text”, because it likes a set of open-ended text questions.
In addition, to set up the **Answer** element.
Furthermore, to distinguish the long text, single row text, numeric text and email address, questioner can control it through **Limitation** element. However, it is different with **Multiple Choice Question Type**, the **Limitation** element is as a child element of **Answer** element.
At last, the **Alignment** element is necessary to distinguish each line and arrangement.

As an example to set up a contact information presented in Fig. 7-12,

```
<Question isMandatory="yes" type="text" id="q0001">
  <Limitation minLength="1" maxLength="1"/>
  <Description type="paragraph" value="Contact Information" id="de0001"/>
  <Alignment vertical="stretch"/>
  !--each answer arrangement-->
  <Answer id="a0001">
    <Limitation minLength="1" maxLength="15" format="char"/>
    <Description type="paragraph" value="First Name:" id="de0002"/>
  </Answer>
  <Answer id="a0002">
    <Limitation minLength="1" maxLength="15" format="char"/>
    <Description type="paragraph" value="First Name:" id="de0003"/>
  </Answer>
  <Answer id="a0003">
    <Limitation minLength="5" maxLength="129" format="char"/>
    <Description type="paragraph" value="Address 1:" id="de0004"/>
  </Answer>
  <Answer id="a0004">
    <Limitation minLength="5" maxLength="129" format="char"/>
    <Description type="paragraph" value="Address 2:" id="de0005"/>
  </Answer>
  ...<Question>
```

### 7.2.2 Matrix Spreadsheet

**What is matrix spreadsheet?**
**Matrix Spreadsheet** is a multiple questions in spreadsheet style with text input boxes (see **Fig. 7-2-2-1**). It is the combination of open-ended text type and matrix. In QSL, we sort it as one of the matrix (easy to remember because of the name of it).
How to set up a matrix spreadsheet?

```xml
<Question isMandatory="yes" type="matrix" id="q0001">
  <Limitation minLength="1" maxLength="5" format="int"/>
  <Description type="paragraph" value="Please rate the following" id="de0001"/>
  <Answer id="a0001" Column.Grid="0">
    <Description type="paragraph" value="Quarter 1" id="de0002"/>
  </Answer>
  <Answer id="a0002" Column.Grid="1">
    <Description type="paragraph" value="Quarter 2" id="de0003"/>
  </Answer>
  <Answer id="a0003" Column.Grid="2">
    <Description type="paragraph" value="Quarter 3" id="de0004"/>
  </Answer>
  <Answer id="a0004" Column.Grid="3">
    <Description type="paragraph" value="Quarter 4" id="de0005"/>
  </Answer>
  <Answer id="a0005" Row.Grid="0">
    <Description type="paragraph" value="Product 1" id="de0006"/>
  </Answer>
  <Answer id="a0006" Row.Grid="1">
    <Description type="paragraph" value="Product 2" id="de0007"/>
  </Answer>
  <Answer id="a0007" Row.Grid="2">
    <Description type="paragraph" value="Product 3" id="de0008"/>
  </Answer>
</Question>
```

![Screenshot of Matrix Spreadsheet](image-url)

**Fig. 7-13** Screenshot of Matrix Spreadsheet
7.2.3 Slide and Rating

![Screenshot of Star Rating](image1)

Fig. 7-14 Screenshot of Star Rating

![Screenshot of Smiley Rating](image2)

Fig. 7-15 Screenshot of Smiley Rating

![Screenshot of Slide](image3)

Fig. 7-16 Screenshot of Slide

![Screenshot of NPS](image4)

Fig. 7-17 Screenshot of NPS

These 4 kinds of question types are similar. From the respective of basic question type, we consider them as single selection. They have different presentations. However, QSL does not consider the front-end presentation. Star Rating can be designed as a single selection question with 5 answer options, which are weights from 1 to 5. Smiley Rating can be designed as a single selection question with 5 answer options and their corresponding images. As to Slide question, it also can be designed as a single selection with a 100 options. In this situation, questioner just need to specify the limitation \textit{maxOccur} as 100, and omits the answer values and ids.
7.2.4 Rank Order and Drag

This is based on the single selection (drop-down list), but concerns exclusive when choose an option, the others cannot choose the selected options. As the prototype, It likes a matrix: grid row (skiing, snowboarding, biking), and grid column (1, 2, 3), and mutually exclude.

![Screenshot of Rank Order](image)

**Fig. 7-18** Screenshot of Rank Order

This is based on ranking question type. Drag is a presentation, which QSL does not concern until now.

```xml
<Question isMandatory="yes" type="ranking" id="q0001">
  <!--question type: ranking-->
  <Description type="paragraph" value="Please rank..." id="de0001"/>
  <Answer id="a0001">
    <Description type="paragraph" value="Skiing" id="de0002"/>
  </Answer>
  <Answer id="a0002">
    <Description type="paragraph" value="Snowboarding" id="de0003"/>
  </Answer>
</Question>
```

**Fig. 7-19** Screenshot of Drag Object

This is based on ranking question type. Drag is a presentation, which QSL does not concern until now.
This image refers to QTI. It is based on ranking (unrepeated options), but this question concerns the image and location. The questioner can mark the locations on the map and let respondent to choose. It is a extension of match item or Connect the Point question type.

Fig. 7-20 Image of Drag Target

The International Air Transport Association assigns three-letter codes to identify airports worldwide. For example, London Heathrow has code LHR.

Some of the labels on the following diagram are missing: can you identify the correct three-letter codes for the unlabelled airports?
LOW-COST FLYING

Frizz, a new low cost airline, already operates a service connecting Manchester and Edinburgh but has recently opened two new routes: a service between London and Edinburgh and one between London and Manchester.

Mark the airline's new routes on the airport map:

Drag the markers by their ends to connect the appropriate points on the image.

Fig. 7-21 Image of Connect the Point

7.2.5 Upload File

Drag your file here

or

Browse

Fig. 7-22 Screenshot of Upload File

Question isMandatory="yes" type="upload" id="q0001">
<!--question type: upload-->
<Description type="paragraph" value="Drag your file here" id="de0001="/>
<Answer id="a0001"/>

<Description type="paragraph" value="" id="de0002="/>
<!--omit value -->
</Answer>
</Question>
7.2.6 Sid-by-side Matrix

This is a 3D-matrix question type. X.Grid (Importance, Satisfaction), Y.Grid (1,2,3,4,5), Z.Grid (Customer Service, Product Package, On-Time Arrivals).

```xml
<Question isMandatory="yes" type="matrix" id="q0001">
    <Limitation minOccur="1" maxOccur="1"/> <!-- single selection -->
    <Description type="paragraph" value="Please rate the following" id="de0001"/>
    <!-- X -->
    <Answer id="a0001" X.Grid="0">
        <Description type="paragraph" value="Importance" id="de0002"/>
    </Answer>
    <Answer id="a0002" X.Grid="1">
        <Description type="paragraph" value="Satisfaction" id="de0003"/>
    </Answer>
    <!-- Y -->
    <Answer id="a0003" Y.Grid="0">
        <Description type="paragraph" value="Customer Service" id="de0004"/>
    </Answer>
    <Answer id="a0004" Y.Grid="0"/>
        <Description type="paragraph" value="Product Package" id="de0005"/>
    </Answer>
    <Answer id="a0005" Y Grid="0">
        <Description type="paragraph" value="On-Time Arrivals" id="de0006"/>
    </Answer>
    <!-- Z -->
    <Answer id="a0006" Z.Grid="0">
        <Description type="paragraph" value="Customer Service" id="de0007"/>
    </Answer>
</Question>
```

Fig. 7-23 Screenshot of SBS-Matrix
8 Logic Types Reference Guide

So far, we have introduced the creation of static e-questionnaires. In this chapter, we will see how to create dynamic e-questionnaires; we also called “intelligent” e-questionnaires to ensure that only relevant questions are displayed to the appropriate respondents.

Previously, we described the logic element and its attribute type at the end of chapter 6. However, only scratching the surface, for this large field, was not enough. For reason, it owes to various manifestations of e-questionnaires.

From all the minor details, many e-questionnaire systems provide it a variety of classifications and definitions. And thereupon we proceeded with the thoughts of trichromatic combination principle, in other words, a multiplicity of separate elements according to some standards. As to uses, this chapter provides a reference to all of logic types (basic logic types and possible combinations) to define them.

All content and diagrams in this chapter refer to QuestionPro®.

8.1 Basic Logic Types

There are four basic logic types:
- Skipping,
- Piping,
- Extraction,
- and Randomization

8.1.1 Skipping

What is skipping?
Skipping gives questioner the ability to create “intelligent” e-questionnaires. For example, it wouldn’t make sense to ask questions about married life to someone who is not married. Using skipping questioner can ensure that only relevant questions are displayed to the appropriate respondents.

How to set up skipping?
- The Logic is the child of Paper, so that questioner must write logic that should be included in tags of paper.
- Logic has a child element Route. Route’s attribute type is written as “skipping”.
- Route’s child elements Condition and Action to control if satisfies conditions and then jump to where.
<Logic>
  <Route type="skipping" id="ro0001">
    <Condition answerId="a0001" isChecked="yes"/>
    <Action questionId="q0007"/>
  </Route>
</Logic>

which means that if selected a0001, then jump to question qu0007.

- Sometimes, between start-tags and the end-tags, records the URL of
destination, as to the to attribute, as the sample value as “url”.

<Logic>
  <Route type="skipping" id="ro0001">
    <Condition answerId="a0001" isChecked="yes"/>
    <Action to="www.url.com"/>
  </Route>
</Logic>

which means that from this answer skipping to a website. There is a situation that
if choose an option then jump to Thank you page usually occurs in
e-questionnaire. <Action descriptionId="de0010"/>

**What is the different skip to options?**

Based on the answer option selected, respondents will be directed to the jump to
location that is set on the survey. Following Skipping To options are available:

- **Papers**: select from any e-questionnaire and e-testing that is permitted to
  enter.
- **Questions**: select from any question that is after the source question.
- **Sections**: can select from any of the section that are after the source
  question.
- **Answers**: usually skip to answer when question types is chosen as “text”.
- **Terminate**: go to finished page, if questioner design its setting that
  respondent can see the result, then show the statistics, otherwise, jump to:
- **Go to Conclusion Page (Description)**: Select this option if questioners
  wish to direct respondents to the Thank You page. Response will be marked
  as a complete response. In other words, select this option if you wish to
  terminate the survey for respondents.
- **Chain URL**: Select this option to take respondents to a different website.

**What is Default Destination skipping?**

The Default Destination skipping is the fallback logic that gets executed when no
other logic gets triggered. If respondents do not choose any option, the default
skipping destination is used when NO OTHER logic gets triggered. This is the
fallback logic that gets executed when no other logic gets triggered.

<Logic>
  <Route type="skipping" id="ro0001">
    <Condition questionId="q0001" isChecked="no"/>
  </Route>
</Logic>
8.1.2 Piping

What is piping?
Piping enables you to carry text from one question to the next depending on the options selected by the respondent. For example if you have a scenario as described below (see Fig. 8-1).

Now questioner would like to replace the text XXXX with the actual product that the respondent had selected for question 1. Piping will enable you to replace XXXX with the appropriate text.

How to set up Piping?
- **Logic** is the schema 310 element. Its child element **Route** to define a route of logic. It is necessary to define route id, and type value as "piping".
- **Route's** child elements: **Condition** and **Action**.
- **Condition** sets the question's reference, and the condition trigger (isChecked). **Action** sets the piping route and the filled location to attribute that is written as dollar mark as braces including piping Ids, underscore and id which is like a function name, such as "$(‘piping Ids’)". We restricted the value of it as "$(‘pip[0-9]’)". Then, in the destination question. As the example,
In this example, at the first question, there are 3 choices using piping logic, whichever be chosen, the question answer text will be piping to next question as one part of next question text.

Can I carry over the Piping Text to multiple questions?
Piping text can only be carried over to the next question. If you wish to carry over text to multiple questions you can get more information in possible combinations of logic types below.

Is there any limitation on the number of characters that can be piped?
Piping text has a limitation of 255 characters (Including Spaces/Special Characters).

8.1.3 Extraction

What is extraction?
Extraction enables questioners to display the choices of a question based on the choices selected for the previous question. It differs from piping. Piping is used to carry out the selected options to fill in some descriptions. However, Extraction is used to carry out the selected options to compare the existing question's texts. For example, (see Fig. 8-2, Fig. 8-3, and Fig. 8-4)
Fig. 8-2  Screenshot of Extraction

The following question without extraction will be displayed as follows:

![Screenshot of Extraction 2](image)

Fig. 8-3  Screenshot of Extraction 2

The following question without extraction will be displayed as follows:

If the respondent selects options **AOL** and **Earthlink** for question 1 then the extracted question (i.e. question 2) will be displayed as follows:

![Screenshot of Extraction 3](image)

Fig. 8-4  Screenshot of Extraction 3

(Only the options selected by the respondent will be displayed on e-questionnaire for respondent).

**How to set up Extraction?**

- **Logic** is the schema 310 element. Its child element **Route** to define a route of logic. It is necessary to define route id, and type value as "extraction".

- **Route's** child elements: **Condition** and **Action**.
• **Condition** sets the question's reference, and the condition trigger (isChecked). **Action** sets the piping route and the filled location to attribute that is written as dollar mark as braces including piping Ids, underscore and id which is like a function name, such as “$(piping Ids)”’. We restricted the value of it as “$('ext[0-9]*')”’. Then, in the destination question.

```xml
<Paper id="p0001" type="questionnaire">
  <Section id="s0001">
    <Question isMandatory="yes" type="selection" id="q0001">
      <Limitation minLength="1" maxLength="4"/>
      <Description type="paragraph" value="Which ISP do you use?" id="de0001"/>
      <Answer id="a0001">
        <Description type="paragraph" value="AOL" id="de0002"/>
      </Answer>
      <Answer id="a0002">
        <Description type="paragraph" value="MSN" id="de0003"/>
      </Answer>
      <Answer id="a0003">
        <Description type="paragraph" value="Earthlink" id="de0004"/>
      </Answer>
      <Answer id="a0004">
        <Description type="paragraph" value="Quest" id="de0005"/>
      </Answer>
    </Question>
    <Question isMandatory="yes" type="matrix" id="q0002">
      <Limitation minOccur="1" maxOccur="1"/>
      <Description type="paragraph" value="Please rate your satisfaction level with:" id="de0006"/>
      <!--Column 0-4-->  
      <Answer id="a0005" Column.Grid="0">
        <Description type="paragraph" value="Very Satisfied" id="de0007"/>
      </Answer>
      <Answer id="a0006" Column.Grid="1">
        <Description type="paragraph" value="Satisfied" id="de0008"/>
      </Answer>
      <Answer id="a0007" Column.Grid="2">
        <Description type="paragraph" value="Neutral" id="de0009"/>
      </Answer>
      <Answer id="a0008" Column.Grid="3">
        <Description type="paragraph" value="Not Satisfied" id="de0010"/>
      </Answer>
      <Answer id="a0009" Column.Grid="4">
        <Description type="paragraph" value="Very Dissatisfied" id="de0011"/>
      </Answer>
    </Question>
  </Section>
</Paper>
```
8.1.4 Randomization

What is randomization?
In order to avoid cheating, some questioners want sections, questions, or answers can be randomized. Randomization enables you to display without ordering in advance.

How to use randomization?
Because randomization does not need to piping to a destination, so if the questioner chooses a randomization type as a attribute value of Route. The reference attribute group support the paper randomization, section randomization, question randomization, and answer randomization.
What is the conflict?
There is a schema 230 Setting. This schema controls the order setting (ascending, descending, and default). If questioner does not set this setting for the same location (paper, section, question, answer), then he can set this logic route.

With which question type does the answer display order works?
- Multiple Choice - Select One, Drop Down, Image Chooser, Select Many, Image Chooser (Select Many)
- Rank Order / Drag N Drop

How to set up Randomization?
- Logic is the schema 310 element. Its child element Route to define a route of logic. It is necessary to define route id, and type value as "randomization".
- Route's child elements: Condition and Action. In randomization logic situation, it does not need Condition.
- Action sets the locations should be randomized.

As the example,

```xml
<Logic>
  <Route type="randomization" id="ro0001">
    <!-- The answers of location: (paper 1 section 1 question 1) should be randomized -->
    <Action paperId="p0001" sectionId="s0001" questionId="q0001"/>
    <!-- The questions of location: (paper 1 section 2) should be randomized -->
    <Action paperId="p0001" sectionId="s0002"/>
  </Route>
  <Route type="randomization" id="ro0002">
    <!-- The sections of location: (paper 2) should be randomized -->
    <Action paperId="p0002"/>
  </Route>
</Logic>
```
8.2 Possible Combinations of Logic Types

After we explained the basic logic types, we will present the possible combinations of basic logic types. In this sub-chapter, we will describe the possible combinations as such examples.

At first, we explain some common possible combination logic types.

8.2.1 Matrix Extraction

What is extraction from matrix question type?
Extracting from a matrix question is a very useful feature, especially for surveys such as (see Fig. 8-5):

![Screenshot of Matrix Extraction](image)

**Fig. 8-5** Screenshot of Matrix Extraction

Now, suppose questioner who would like to ask the respondent why he/she selected Not Satisfied for a particular service provider. Questioner can use Extraction to set this up.

If the respondent selects Not Satisfied for AOL and Qwest for question in Fig. 8-5 then the extracted question (i.e. in Fig. 8-6) will be displayed as follows:

![Screenshot of Matrix Extraction](image)

**Fig. 8-6** Screenshot of Matrix Extraction
In this scenario above, it is similar with the example presented in simple extraction type, just the question type changed. Therefore, the specification is presented below,

```xml
<Paper type="questionnaire" id="p0001">
  <Section id="s0001">
    <!--the first question-->
    <Question isMandatory="yes" type="matrix" id="q0001">
      <Limitation minOccur="1" maxOccur="1"/>
      <!--single choice for matrix each row-->
      <Description type="paragraph" value="How satisfied are you with the following:" id="de0001">
        <Answer id="a0001" Column.Grid="0">
          <Description type="paragraph" value="Not Satisfied" id="de0002"/>
        </Answer>
        <Answer id="a0002" Column.Grid="1">
          <Description type="paragraph" value="Neutral" id="de0003"/>
        </Answer>
        <Answer id="a0003" Column.Grid="2">
          <Description type="paragraph" value="Very Satisfied" id="de0004"/>
        </Answer>
        <Answer id="a0004" Column.Grid="0">
          <Description type="paragraph" value="AOL" id="de0005"/>
        </Answer>
        <Answer id="a0005" Row.Grid="1">
          <Description type="paragraph" value="MSN" id="de0006"/>
        </Answer>
        <Answer id="a0006" Row.Grid="2">
          <Description type="paragraph" value="Earthlink" id="de0007"/>
        </Answer>
        <Answer id="a0007" Row.Grid="3">
          <Description type="paragraph" value="Qwest" id="de0008"/>
        </Answer>
      </Description>
    </Question>
    <!--break to create a new area for the second one-->
    <!--the second question-->
    <Question isMandatory="yes" type="text" id="q0002">
      <Limitation minLength="1" maxLength="1"/>
      <Description type="paragraph" value="Please describe why you are not satisfied with the following:" id="de0010">
        <Answer id="a0008">
          <Limitation minLength="5" maxLength="129"/>
          <Description type="paragraph" value="$(ext0001)" id="de0011"/>
        </Answer>
      </Description>
    </Question>
  </Section>
</Paper>
```
8.2.2 Compound Branching

What is compound branching logic?
With simple branching of skipping type, questioners cannot set up logic based on responses to multiple questions. Also with simple branching, the logic is executed immediately. **Compound Branching** means that with help of compound Logic you can set up branching based on responses to multiple questions in the questionnaire. Sometimes, there is another name called **Delayed Branching** that means questioners can set up Branching Logic based on a Question which was asked much earlier in the questionnaire. But, there are no differences between both of them.

How to use Compound Branching?
In compound branching, we provide logic type “skipping” to jump to other questions, and attribute to connect with relative answers.

Consider this example:
Q1: Gender
Q2: Age
Q3: General question to be answered by everybody
Q4: General question to be answered by everybody
Q5: General question to be answered by everybody

Now after Q5 respondents should branch to different questions based on their selection in Q1 AND Q2
Q6: To be answered only by respondents who selected Gender=Male and Age=18 & above
Q7: To be answered only by respondents who selected Gender=Female and Age=18 & above

Now all respondents should again answer rest of the questions.
Q8: General question for all respondents
Q9: General questions for all respondents.
Specification for setting up this case:

```xml
<Paper type="questionnaire" id="p0001">
  <Section id="s0001">
    <Question isMandatory="yes" type="selection" id="q0001">
      <Limitation minOccur="1" maxOccur="1"/>
      <Description type="paragraph" value="Gender" id="de0001"/>
      <Answer id="a0001">
        <Description type="paragraph" value="F" id="de0002"/>
      </Answer>
      <Answer id="a0002">
        <Description type="paragraph" value="M" id="de0003"/>
      </Answer>
    </Question>
    <Question isMandatory="yes" type="selection" id="q0002">
      <Limitation minOccur="1" maxOccur="1"/>
      <Description type="paragraph" value="Age" id="de0004"/>
      <Answer id="a0003">
        <Description type="paragraph" value="less than 18" id="de0005"/>
      </Answer>
      <Answer id="a0004">
        <Description type="paragraph" value="greater than or equal to 18, less than 60" id="de0006"/>
      </Answer>
      <Answer id="a0005">
        <Description type="paragraph" value="greater than or equal to 60" id="de0007"/>
      </Answer>
    </Question>
    <!-- Question 3-5 omitted -->
    <Question isMandatory="yes" type="text" id="q0006">
      <Limitation minLength="1" maxLength="1"/>
      <Description type="paragraph" value="Reason" id="de0013"/>
      <Answer id="a0010">
        <Description type="paragraph" value="" id="de0014"/>
      </Answer>
    </Question>
    <Question isMandatory="yes" type="text" id="q0007">
      <Limitation minLength="1" maxLength="1"/>
      <Description type="paragraph" value="Reason" id="de0015"/>
      <Answer id="a0011">
        <Description type="paragraph" value="" id="de0016"/>
      </Answer>
    </Question>
  </Section>
</Paper>
```
8.2.3 Looping with Text Piping

What is looping with text piping?
Simply put, considering the following example: Questioners are asking respondents to select credit cards they have and for each credit card, and wish to collect further information using follow up questions (see Fig. 8-7).

If the respondents select the options “Visa” and “American Express”, they should only be asked follow up questions for “Visa” and “American Express” cards. They should skip questions for “Master” and “Discover” cards.
To set this up, first set up the entire survey with all questions (see Fig. 8-7).
Fig. 8-7  Screenshot of Looping with Text Piping

Fig. 8-8  Screenshot of Looping with Text Piping
How to use looping with text piping?

Looping with Text Piping is derived from Piping. Therefore, the logic type value is “piping”. Specification on above example:

```xml
<Paper type="questionnaire" id="p0001">
  <Section id="s0001">
    <Question isMandatory="yes" type="selection" id="q0001">
      <Limitation minOccur="1" maxOccur="4"/>
      <Description type="paragraph" value="What types of credit cards do you have (Select all that apply?)" id="de0001">
        <Alignment vertical="stretch"/>
        <Answer id="a0001">
          <Description type="paragraph" value="Visa" id="de0002"/>
        </Answer>
        <Answer id="a0002">
          <Description type="paragraph" value="Mastercard" id="de0003"/>
        </Answer>
        <Answer id="a0003">
          <Description type="paragraph" value="American Express" id="de0004"/>
        </Answer>
        <Answer id="a0004">
          <Description type="paragraph" value="Discover" id="de0005"/>
        </Answer>
      </Description>
      <Description type="break" value="" id="de0006"/>
      <Description type="break" value="" id="de0006"/>
    </Question>
    <Question isMandatory="yes" type="selection" id="q0002">
      <Limitation minOccur="1" maxOccur="1"/>
      <Description type="paragraph" value="How oftern do you use your $('pip0001')" id="de0007">
        <Answer id="a0005">
          <Description type="paragraph" value="Daily" id="de0006"/>
        </Answer>
        <Answer id="a0006">
          <Description type="paragraph" value="Weekly" id="de0007"/>
        </Answer>
        <Answer id="a0007">
          <Description type="paragraph" value="Monthly" id="de0008"/>
        </Answer>
        <Answer id="a0008">
          <Description type="paragraph" value="Rarely" id="de0009"/>
        </Answer>
      </Description>
    </Question>
    <Question isMandatory="yes" type="selection" id="q0003">
      <Limitation minOccur="1" maxOccur="1"/>
    </Question>
  </Section>
</Paper>
```
8.2.4 Quota and Scoring Logic

Refer to Fig. 8-7. For instance, questioner sets each answer option has different score.

```xml
<Question isMandatory="yes" type="selection" id="q0001">
  <Limitation minOccur="1" maxOccur="4"/>
  <Description type="paragraph" value="What types of credit cards do you have (Select all that apply?)" id="de0001"/>
  <Alignment vertical="stretch"/>
  <Answer id="a0001* score="2.5"">
    <Description type="paragraph" value="Visa" id="de0002"/>
  </Answer>
  <Answer id="a0002* score="-2 ">
    <Description type="paragraph" value="Mastercard" id="de0003"/>
  </Answer>
  <Answer id="a0003* score="-1">
    <Description type="paragraph" value="American Express" id="de0004"/>
  </Answer>
</Question>
```
The total score of this question is 5 points. But each answer option score is different (2.5 point, -2 point, -1 point, 2.5 point). If questioner sets that if this question reach 3 point then jump to Thank you page.
Appendix A – The Cases for Using QSL

A. 1 ENQUETE-BAISE

ENQUETE-BAISE is a general-purpose e-questionnaire server developing for ubiquitous questionnaire that can be used as a readymade e-questionnaire server component in various web service systems as well as an alone e-questionnaire server with general-purpose for various questionnaires. ENQUETE-BAISE can also be used as an e-testing server and an e-voting server with general-purpose by restricting its general functions and strengthening its security functions. ENQUETE-BAISE has been used since 2007.

Why we mention ENQUETE-BAISE here as a use case? Firstly, it is a general-purpose e-questionnaire server for ubiquitous questionnaire, thus it is able to contain and support services for various e-questionnaires and e-questionnaire systems. Secondly, it belongs to our lab achievement, so we can get the whole information well. That means we can analyze inside and outside, so that specify by QSL very well. Last but not the least, the ideal state of QSL is, implementing a QSL generator system to automatically generate e-questionnaire, e-testing, and e-voting systems. ENQUETE-BAISE is a general-purpose server for supporting e-questionnaire, e-testing, and e-voting, also is the best candidate as the QSL generator systems, owning to its universality and comprehensive.

Fig. A1-1 shows overview of ENQUETE-BAISE for portable data. We implemented and improved ENQUETE-BAISE for data portability. All the data are exchanged in a QSL specification file. A questioner can edit a paper and set up
settings for the paper. The paper data and setting data are saved in the database in real-time. The questioner also can import a QSL-format requirement specification recording paper data and setting data during setting up phase. The questioner can export paper data and setting data by QSL format during the whole phases. The respondents answer questions, and the response data are saved in database during submitting phase. The questioner can export and download the submitted response data. After collecting the responses, all the response data are all integrated and send to analyst to analyze and get the result. ENQUETE-BAISE can automatically analyze and get result. The result data is saved in database and the questioner can export the result data. Sometimes, the questioner can import response data to analyze by ENQUETE-BAISE, and import the result data to record in database.

The paper data, setting data, response data, and result data are specified by QSL. QSL also can specify ENQUETE-BAISE that proves QSL can specify various e-questionnaire, e-testing, and e-voting systems owing to its generality. In addition, QSL is as the format for portable data. Besides, ENQUETE-BAISE is the candidate as the QSL generator system to automatically generate e-questionnaire, e-testing, and e-voting systems.
A. 2 A General-Purpose Offline E-Testing Environment

We implemented a general-purpose offline e-testing environment based on QSL to provide users with offline e-testing service to execute various offline e-testing. This environment has been applied to execute final test of Discrete Mathematic in University of Japan since 2015.

![Offline e-testing environment diagram](image)

The overview of general-purpose offline e-testing environment is illustrated in Fig. B2-1. An editing tool is set into the USB flash memory. A questioner uses the editing tool to specify all participants by QSL. After that, the editing tool will generate the admission ticket for each respondent automatically. The questioner uses the editing tool to prepare exam paper with complex question types and logic, which files are specified by QSL. An offline test server distributes exam paper to respondents through wireless LAN. The environment uses access point to support to test large number of the respondents easily and conveniently. The respondents confirm the exam paper from offline server. During the test, the monitor monitors the connection states of all the respondents through offline server for avoiding online cheating activities, and the closed network will also block accesses from outside to connect to the offline server. After submitting, all the responses will be collected and integrated by offline server as QSL-format files. The offline server distributes and collects offline e-testing. Questioner only need to stick the USB flash memory into a PC, the PC can be the offline server to execute offline e-testing. The marking tool is also set into the USB flash memory. A marker uses the marking tool to mark the collected responses and give a result of the test as a QSL-format file. An analyst uses the marking tool to analyze responses and get the test result automatically. The functions...
of the tools and phases as well as the environments are defined by QSL.

As to the **selling point** of our environment, is implement the constructing server function as a bootable USB flash memory to make the environment can be easily and conveniently constructed even by teachers who are non-specialist of computer.

The usefulness of the general-purpose offline E-testing environment is confirmed in the test. It worked well in the test, although the classroom is large and the number of respondents is many. All of the respondents did the test smoothly from start to the end of the test. All of steps in our test can be easily repeatedly. Besides, all data in the environment were specified by QSL well, so we can use the environment to do various tests in lots of middle schools and universities.

**In general,** QSL used in this situation, shows that QSL can be used as a standardized specification to support a large test and complex question types. The design of our environment is imaginative, but it needs QSL to support it and helps it to clear up what are necessary in the whole environment. QSL can specify all the functions and even the environment. Furthermore, QSL is a practicable data format to support data exchange.
A. 3 An E-Voting Application of an Enterprise Information System on WeChat Public Platform

We developed an e-voting application as a web plugin of an internal enterprise information system, and extended on the WeChat public platform, as we all know that WeChat is one of the most-used smartphone applications in the world. This e-voting application provides services on both Web and WeChat to create and edit e-ballot, to vote and check analyzed results if settings qualified voters to do, to distribute and collect by both e-mail and WeChat, and to make a list of all the drafts, distributed e-ballots as a role of a questioner, and all the submitted and uncommitted e-ballots as a role of a respondent. The e-voting application provides authentication service by the method of secret, and limits the channel on WeChat. WeChat is a platform execute on smartphone and tablet, helping send and reply messages to notify employees who are eligible respondent according to department, branch, or preference even chosen by sponsor or questioner based on internal process.

We specified by QSL to make a detailed requirement list of the e-voting application when we design it. QSL has been used directly. Moreover, all the data are specified by QSL. This e-voting application is used for 377 staffs in a Chinese software company since 2016. Because the company database keeps track of its staffs, departments, and projects for internal use only, we used NAT-DDNS tech to access WeChat, and did test on a WeChat enterprise account to do e-voting. The application is designed to run under Apache and Tomcat. We have made QSL the cornerstone of its development and likely to draw upon QSL.

Comparing with above-mentioned two use cases (even they are published), this is a real application in a real development in a software company; it is not just a research (the development is so-called a project not a product). Under developing, we are effort to implement according to a standardized specification and instructing under the design mind of our product manager. QSL devotes our pre-design stage and API. A so-called project may design under our imagination, but a real product is designed by our team (me is a developer for front end and our team members as developers for after end). In the process of development, we understood that both PM and developers need compromises (PM is too idealistic and has a lot of imagination but does not understand development but, developers have technology but does not understand business field).

Some screenshots of our application during development and test stage are shown as follow:
At first, the application totally divided into three main buttons, which have different functions. 1) to create a new poll, 2) to check the list of poll created by ourselves, 3) to check the lists of poll what we took part in (contains we have already participated
or have not done yet). In addition, some notifications push to us, which want us to poll (we are as the role of the respondents). All the notification will be clicked and jump to the real poll page if you do not poll yet. If you poll for it, it will jump to the result page.

Fig. B3-1  Main buttons & notifications.  Fig. B3-2  Polling sheet.

Secondly, in the creating phase, our application allows you questioner to write poll title; explanation likes some subtitle or information about this poll; some pictures; settings like anonymous respondents, invisible results for respondents, beginning and ending time; respondent ranges; response modes like one page one question, one page multiple questions. As to the question types, there are three kinds types (radio button for single choice, checkbox for multiple choices, and text area for open-ended text). Each question and answer support to add, remove, force or not, choose to write other fields, and add pictures. Each polling sheet supports to edit, immediately publish, and just save.

Thirdly, as to the lists of polling as a questioner or a respondent, both roles are in two kinds of phases, which are before publish or not, and before poll or not. Therefore, as a role of a questioner, the lists show draft list and publish list. And as a role of a respondent, the lists show the stages of being already took part in and before taking part in.
Above-mentioned functions and screenshots are still in the development and testing stages based on WeChat development platform. We used a public WeChat account named “HaveFun” to do develop and test. After testing and before really publishing this application, all the information is private because it belongs to the Star-net company and concerns its departments and staffs.

It is worth mentioning that, the respondent list shows the departments within the company, concerns **security**. Except anonymous function during creating phase, doing an e-voting or polling event by this application, we need to verify the respondents’ identities. With the development of this application, we used to verify the email addresses or telephone numbers based on WeChat platform, because a WeChat account is registered by a real address or a telephone number. All the information about the staffs are stored in the internal database. We were coding to connect and compare these two fields between our database and WeChat account.
based on WeChat development API. If the fields do not match the internal database, the respondent cannot join the company list. What’s more, the eligible respondents only have one chance to poll.

There are some screenshots about respondent lists in a testing stage and a real condition.

**Fig. B3-6** Respondent list in testing stage.

**Fig. B3-7** Respondent list in real.
## Appendix B – Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Contents and Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013.05</td>
<td>v0.0</td>
<td>Prototype</td>
</tr>
<tr>
<td>2014.02</td>
<td>v1.0</td>
<td>First QSL Proposal</td>
</tr>
<tr>
<td>2014.09</td>
<td>v1.3</td>
<td>QSL Foundation in a frame</td>
</tr>
<tr>
<td>2014.11</td>
<td>v1.4</td>
<td>Add some comment, question type, logic type</td>
</tr>
<tr>
<td>2014.12</td>
<td>v1.5</td>
<td>Change some comments, formula, interface, functions</td>
</tr>
<tr>
<td>2015.01</td>
<td>v1.6</td>
<td>Change QSL structure into 3 parts</td>
</tr>
<tr>
<td>2015.05</td>
<td>v1.7</td>
<td>Add some details of phases, security, etc.</td>
</tr>
<tr>
<td>2015.11</td>
<td>v1.8</td>
<td>Change new QSL structure in circle model</td>
</tr>
<tr>
<td>2016.08</td>
<td>v2.0</td>
<td>Extend e-voting elements and attributes</td>
</tr>
<tr>
<td>2017.05</td>
<td>v2.1</td>
<td>Revise some details of e-voting part</td>
</tr>
<tr>
<td>2017.07</td>
<td>v3.0</td>
<td>Desperate the simpleType and ComplexType for reuse</td>
</tr>
<tr>
<td>2018.08</td>
<td>v3.1</td>
<td>Extract primary elements as the 300-series schemas, and cover more than 30 descriptions of requirements.</td>
</tr>
<tr>
<td>2018.09</td>
<td></td>
<td>To be continue…</td>
</tr>
</tbody>
</table>
Requirements for common of systems

- **Setting up**

1. **Authority for sponsor and questioner to set an activity:**
   e-questionnaire, e-testing, and e-voting systems should provide sponsor or questioner the authorities to set an activity.

2. **Check paper state:** e-questionnaire system should provide questioner to check the paper state such as draft, launched, pause, collected, finished.

3. **Client-side software connect with database and server:**
   e-questionnaire, e-testing, and e-voting systems should provide a client-side software connect with database and server to offer service to whom use it (sponsor, executor, respondent).

4. **Copy paper:** e-questionnaire, e-testing, and e-voting systems should provide authorized questioner to copy paper.

5. **Create paper/section/question/answer:** e-questionnaire, e-testing, and e-voting systems should provide questioner to create a paper, sections, questions, or answers.

6. **Delete paper:** e-questionnaire, e-testing, and e-voting systems should provide questioner to delete paper and put it into the trash box (remove permanently or temporarily), sometimes need a password or authority to delete it.
7. **Delete section/ question/ answer:** e-questionnaire, e-testing, and e-voting systems should provide authorized questioner to delete a sections, questions, or answers.

8. **Device for setting up connecting with software:** e-questionnaire, e-testing, and e-voting systems should provide the device for questioner to set up activity connecting with setting up software.

9. **Edit paper/ section/ question/ answer:** e-questionnaire, e-testing, and e-voting systems should provide authorized questioner to edit a paper, sections, questions, or answers.

10. **Executor task authority:** e-questionnaire, e-testing, and e-voting systems should provide executor to authorize their tasks.

11. **Generate token with formats:** e-questionnaire, e-testing, and e-voting systems should provide eligible respondents to generate token with formats.

12. **IP address record and limit:** e-questionnaire, e-testing, and e-voting systems should provide questioner to record and limit the respondents’ IPs.

13. **Launch paper:** e-questionnaire, e-testing, and e-voting systems should provide questioner to launch paper during a range of time.

14. **Launching device for questioner:** e-questionnaire, e-testing, and e-voting systems should provide record information about launching device for questioner.
15. **Paper and setting database:** e-questionnaire, e-testing, and e-voting systems should provide a database for store paper and setting information for a questioner.

16. **Respondent randomized authenticated token:** e-questionnaire, e-testing, and e-voting systems should provide eligible respondent to randomized authenticated tokens.

17. **Search paper** e-questionnaire, e-testing, and e-voting systems should provide authorized questioner to delete paper.

18. **Sponsor and questioner authentication:** e-questionnaire, e-testing, and e-voting systems should provide sponsor or questioner to confirm the roles to set an activity.

19. **Stop launching:** e-questionnaire, e-testing, and e-voting systems should provide questioner, executor, or sponsor (authorized person) to stop launch paper when for emergencies.
Registering

1. **Client-side registration software on server:** e-questionnaire, e-testing, and e-voting systems should provide a client-side software for registration for eligible respondents (mainly) and executor, which is connecting with server for registering.

2. **Executor authentication by secrecy:** e-questionnaire, e-testing, and e-voting systems should provide executor to authenticate their role by secrecy (what they know: password).

3. **Executor authentication by token:** e-questionnaire, e-testing, and e-voting systems should provide executor to authenticate their role by token (what they have: ID, randomized tokens).

4. **Registering server:** e-questionnaire, e-testing, and e-voting systems should provide respondent and executor for registering.

5. **Registration database:** e-questionnaire, e-testing, and e-voting systems should provide a registration database to store registration information of respondent and executor.

6. **Respondent authentication by secrecy:** e-questionnaire, e-testing, and e-voting systems should provide respondent to authenticate whether he is eligible or not by secrecy (what they know: password).

7. **Respondent login:** e-questionnaire, e-testing, and e-voting systems should provide authenticated respondents to login and jump to enter answering page.
Distributing

1. **Distribution by channel:** e-questionnaire, e-testing, and e-voting systems should provide a distribution channel for questioner who launch the paper (SMS, WAP, digital TV, Internet, Intranet, Kiosk, postal, telephone, digital storage device, paper, fax, email, website, abroad postal, abroad electronic, abroad other, N/A, other).

2. **Distribution server:** e-questionnaire, e-testing, and e-voting systems shall provide a distribution server to distribute the paper by different channels.

3. **Distribution software:** e-questionnaire, e-testing, and e-voting systems should provide a distribution software for questioner, sponsor, or other kinds of authorized executor.
● Submitting

1. **Access channel gateway**: e-questionnaire, e-testing, and e-voting systems should provide an access channel gateway to deal with different channel for submitting (technology).

2. **Answer questions**: e-questionnaire, e-testing, and e-voting systems should provide respondent to answer questions and select options.

3. **Back previous and jump next**: e-questionnaire, e-testing, and e-voting systems should provide respondent to back check previous questions, and jump to next, if the questioner allows (setting this function in setting up phase).

4. **Blind e-papers for anonymity**: e-questionnaire, e-testing, and e-voting systems should provide the respondent to send a blinded e-paper to the submit server (usually called validator) together with some identification and authentication information. This validator inspects the respondent’s right to submit. If the respondent has the right to submit, the submitting server digitally signs the blinded e-paper and sends it back to the respondent. The respondent now un-blinds the received data and gets a signed e-paper. The he sends this signed e-paper to a second submitting server (often called tallier). Then the tallier knows that the e-paper was sent by an eligible respondent, but he cannot decide which respondent sent the paper.
5. **Client-side software for submitting:** e-questionnaire, e-testing, and e-voting systems should provide a client-side software for respondent to connect with the submitting server for answer questions, submit answers to server, spoil his response, ensure equality and accuracy of presentation of options on any submit device.

6. **Paper database:** e-questionnaire, e-testing, and e-voting systems should provide a paper database to store the answers of the paper related with each respondent.

7. **Error recovery:** e-questionnaire, e-testing, and e-voting systems should provide error recovery on a submitting server to run a self-check before a resuming is possible. In case of irreversible problems the voting server shall prevent a resuming the submitting phase.

8. **Monitor submitting:** e-questionnaire, e-testing, and e-voting systems should provide monitor to monitor the respondents during submitting phase.

9. **Monitor-oriented software:** e-questionnaire, e-testing, and e-voting systems should provide monitor software to monitor the respondents during submitting phase (prevent cheating).

10. **Ping IP address:** e-questionnaire, e-testing, and e-voting systems should provide monitor to monitor the respondents to ping their IP addresses.
11. **Retrieve question**: e-questionnaire, e-testing, and e-voting systems should provide respondent to retrieve questions during this phase.

12. **Submit response**: e-questionnaire, e-testing, and e-voting systems should provide respondents to submit their responses.

13. **Submitting server**: e-questionnaire, e-testing, and e-voting systems should provide submitting server to
   
a) be capable of producing comprehensive audit data, to indicate to the executor the number of paper submit so far and its current state;

b) store in the e-ballot box all e-papers submit by eligible respondents during phase;

c) warn the executor if they try to close the activity before the final data;

d) not provide any information about the submitting phase except the current state and the number of papers submit so far;

e) run a self-check before a resuming is possible. In case of irreversible problems the voting server shall prevent a resuming the submitting phase;

f) regularly perform automatic self-checks and report the results to the executors, be capable of performing self-checking;

g) provide executor interface to identify and authenticate, start the phase once, resume the phase after exception, malfunctions, and
break-downs, start the counting phase only after having closed
the submitting phase, perform self-checks, check that the server
has been set up correctly, check the current state, read the audit
trails;

h) not reveal the link from the last respondent to his e-paper;
i) remain open a sufficient phase of time to allow for any delay of
data transport;
j) support an adequate number of paper;
Collecting

1. **Collection by channel:** e-questionnaire, e-testing, and e-voting systems should provide a collecting channel for questioner who launch the paper (SMS, WAP, digital TV, Internet, Intranet, Kiosk, postal, telephone, digital storage device, paper, fax, email, website, abroad postal, abroad electronic, abroad other, N/A, other).

2. **Collection server:** e-questionnaire, e-testing, and e-voting systems should provide collection server to collect submitted data.

3. **Collection software:** e-questionnaire, e-testing, and e-voting systems should provide a collection software to collect submitted data to analyst or analyzing phase.
Analyzing

1. **Analysis server**: e-questionnaire, e-testing, and e-voting systems should provide analysis server to analyze the submitted responses, and get the results.

2. **Analysis software on the server**: e-questionnaire, e-testing, and e-voting systems should provide analysis software connecting with the analysis server.

3. **Basic analysis**: e-questionnaire, e-testing, and e-voting systems should provide to analyze the responses of the different question types.

4. **Response database encryption**: e-questionnaire, e-testing, and e-voting systems should provide encrypt the response database away from intruders.
• **Counting**

1. **Counting software:** e-questionnaire, e-testing, and e-voting systems should provide counting software for eligible counters.

2. **Counts scrutiny to get result and statistics:** e-questionnaire, e-testing, and e-voting systems should provide carefully count to get the results and statistics.

3. **Integrate response counting:** e-questionnaire, e-testing, and e-voting systems should provide to integrate response counting and provide export as some formats.
Requirements for e-questionnaire systems

● Setting up

1. **Export paper by formats**: e-questionnaire system should provide questioner to export paper by some formats (usually .xml, .txt, sometimes .csv, .docx, .xsl).

2. **Import paper by formats**: e-questionnaire system should provide questioner to import paper by some formats (usually .xml).

3. **Reminder paper launched**: e-questionnaire system should provide to notice the target respondents that the paper has been launched that you can response it, sometimes also notice them the deadline if the questioner sets it.

4. **Search paper**: e-questionnaire system should provide questioner to search the papers what they have authority to scan or operate. It also provides the respondents to search it on his homepage and the searched papers should be connect to his account (he must have the authority).
• Submitting

1. **Continue answering:** e-questionnaire system should provide respondents to continue answer questions when they are stopping internally. It exists with the internal stop function.

2. **Export response:** e-questionnaire system should provide respondents to export his response if the questioner has already set to allow them to do that. And the export response should be downloaded into the local directory by some formats.

3. **Import response:** e-questionnaire system should provide respondents to import his response if the questioner has already set to allow them to do that. And the import responses should be verified the formats and uploaded form the local directory by the formats.

4. **Integrate response:** e-questionnaire system should provide respondent to intergrade response automatically when he wants to export the response.

5. **Internal stop:** e-questionnaire system should provide respondents to internally stop to answer if the questioner has already set to allow them to do that and the inside the frequency of the stopping. In this function, it lurks a function named **continue answering**.
Analyzing

1. **Analyst-oriented software and device**: e-questionnaire system should provide analyst oriented software and device to analyze the collected responses.
Requirements for e-testing systems

- **Setting up**

1. **Export paper by formats:** e-testing system should provide questioner to export paper by some formats (usually .xml, .txt, sometimes .csv, .docx, .xsl).

2. **Import paper by formats:** e-testing system should provide questioner to import paper by some formats (usually .xml).

3. **Reminder paper launched:** e-testing system should provide to notice the target respondents that the paper has been launched that you can response it, sometimes also notice them the deadline if the questioner sets it.

4. **Search paper:** e-testing system should provide questioner to search the papers what they have authority to scan or operate. It also provides the respondents to search it on his homepage and the searched papers should be connect to his account (he must have the authority).
- **Registering**

1. **Respondent authentication by token**: e-testing system should provide respondents to register on the system by authenticating his ID information (such as ID card, job card, library card).

2. **Executor authentication by token**: e-testing system should provide executors to register on the system by authenticating his ID information (such as ID card, job card, library card).
- **Submitting**

1. **Blinded authenticated token for anonymity:** e-testing system should provide to digitally authenticate a message without knowing the content of the message. The respondent sends a blinded anonymous authentication token (instead of the blinded e-paper) to the validator together with some identification and authentication data. The respondent receives a digital signature from the validator on this blinded token. In the next step, the respondent computes the value for the signed authentication token and sends his data together with his e-paper to the tallier, which accepts the respondent because of the digitally signed authentication token.

2. **Confirm response:** e-testing system should provide a confirmation to the respondent regarding the status of his paper, at least the information that his e-paper has been successfully stored.

3. **Export response:** e-testing system should provide the respondent to export his response for per question if it is allowed.

4. **Import response:** e-testing system should provide the respondent to import his response by some formats (usually .xml, .txt, sometimes .csv, .docx, .xsl) if it is allowed (for uploaded file question type).
5. **Integrate response:** e-testing system should provide respondent to integrate his exported responses by some formats (usually .xml, .txt, sometimes .csv, .docx, .xsl) if it is allowed.

6. **Internal stop:** e-testing system should provide respondent to internal stop and restart to answer (store previous data) if it is set in setting up phase.

7. **Security software:** e-testing system should provide security software to ensure that the right to cast a paper is dictated by the presence of a token.
• Analyzing

1. **Analyst-oriented software and device**: e-testing system should provide an analyst-orientated software and device to analyze collected data (responses).

2. **Trend analysis**: e-testing system should provide a look at data over time for a long-running paper.
Marking

1. **Blink mark response:** e-testing system should provide marker to blink mark responses. It similar with blink analysis. It always mark text question type.

2. **Mark response:** e-testing system should provide marker to mark responses and give the scores.

3. **Marker-oriented software and device:** e-testing system should provide marker-oriented software and device for marker.

4. **Marking server:** e-testing system should provide a marking server to get the response data, identify and authenticate markers, mark the responses and give the score, store the marked data, and send it to counting before deadline.

5. **Marking software:** e-testing system should provide marking software to automatically mark the responses.
- **Counting**

1. **Mix net for anonymity**: e-testing system should provide a mix net for anonymity as a cryptographic alternative to an anonymous channel. It secures who is communicating with whom and it secures the content of the transferred messages.

- **Auditing**

1. **Audit analysis**: e-testing system should provide audit analysis to audit the total numbers of eligible respondents and issued or qualified ballots provided by the security system with the total number reported. It should also provide trusted data provided by the security process and data provided by the submitting process proves that no legitimate papers have been lost by the system. It also proves that there is consistency between the number of eligible respondents and the spoiled, unspoiled and unused votes as recorded by the e-voting system.

2. **Audit eligible respondents**: e-testing system should provide he total numbers of eligible respondents.

3. **Audit report**: e-testing system should provide the auditing report to prove the authenticity of the result.

4. **Auditing server**: e-testing system should provide an (independent) auditing server for collect and count (specially audit) the result.
5. **Auditing software:** e-testing system should provide auditing software for eligible auditor (mainly) and eligible analyst.

6. **Auditor-oriented software and device:** e-testing system should provide an auditor-oriented software and device independently. It close to the security software and auditing analysis software, as well corresponding device for auditor.

7. **Submitting interface:** e-testing system should provide a submit interface to provide the ballot data for analysis and auditing.
Requirements for e-voting systems

- **Setting up**

1. **Launch option nomination:** e-voting system should provide to launch option nomination, which is a process of approving the options to be presented to voters in a referendum. The options can be a straight choice, e.g. YES or NO, to a single question, or can be more complex involving choices to a number of questions and/or preferences of choice.

2. **Launch proposal of option nomination:** e-voting system should provide to launch proposal of option nomination, which is a process of approving nominees as eligible candidates for certain positions in an election. A candidate in this context can be a named individual or a party.

3. **Option database:** e-voting system should provide an option database to store the information for the nominees and the parties.

4. **Option nomination server:** e-voting system should provide a option nomination server to store the candidate information.

5. **Option nomination software:** e-voting system should provide option nomination software.
• Registering

1. **Executor authentication by biometric:** e-voting system should provide to identify and authenticate the executors by fingerprints, iris scans, face recognition, voice, manual signature, and DNA.

2. **Executor authentication by secrecy:** e-voting system should provide to identify and authenticate the executor by password (TAN/knowledge).

3. **Executor authentication by token and biometric:** e-voting system should provide to identify and authenticate the executors by fingerprints, iris scans, face recognition, voice, manual signature, and DNA.

4. **Executor authentication by token:** e-voting system should provide to identify and authenticate the executors by Id card, job card, or library card (ownership).

5. **Respondent authentication by biometric:** e-voting system should provide to identify and authenticate the respondents by fingerprints, iris scans, face recognition, voice, manual signature, and DNA.

6. **Respondent authentication by token and biometric:** e-voting system should provide to identify and authenticate the respondents by both token (ownership) and biometric (property).
7. **Respondent authentication by token:** E-voting system should provide to identify and authenticate the respondents by ID card, job card, or library card (ownership).
• Submitting

1. **Alter message net:** e-voting system should verify the freshness, authenticity, integrity, and format correctness of all messages before processing them.

2. **Auditing interface:** e-voting system should provide an auditing interface to provide the response data for analysis and auditing.

3. **Authenticate the registered respondent:** e-voting system should provide to authenticate and identify the registered respondent whether is the eligible one.

4. **Blinded authenticated token for anonymity:** e-voting system should provide to digitally authenticate a message without knowing the content of the message. The respondent sends a blinded anonymous authentication token (instead of the blinded e-paper) to the validator together with some identification and authentication data. The respondent receives a digital signature from the validator on this blinded token. In the next step, the respondent computes the value for the signed authentication token and sends his data together with his e-paper to the tallier, which accepts the respondent because of the digitally signed authentication token.

5. **Confirm response:** e-voting system should provide a confirmation to the respondent regarding the status of his paper, at least the information that his e-paper has been successfully stored.
6. **Holomorphic secret for anonymity:** e-voting system should provide

7. **Report reason for not voting:** In some jurisdictions, e.g. Australia, where voting is compulsory there is a need to report if an elector has been excused for not voting and what was the reason for the excuse.

8. **Retrieve and amend votes for Pre-ballot box:** e-voting system should provide to retrieve and amend votes before they are counted.

9. **Security software:** e-voting system should provide a security system and sealing mechanism so that trust can be placed in the seal and hence the sealed data. This implies that the seal should be performed as close to the user submission of the vote as technically possible.

10. **Separation of duty for anonymity:** e-voting system should provide a separation of duty approach (also works with at least two submitting servers). One inspecting the right to submit and another one storing the eligible e-papers. The respondent authenticates himself to the first server. In case that he has the right to submit, he receives a random number generated by this first server. This random number is also send to the second sever but without any information about the respondent ID. Now the respondent uses this random number to authenticate himself as an eligible respondent to the second submitting server. Again this second server can only check whether an eligible respondent sent the e-paper but not who.
11. **Store authenticated paper:** e-testing system should store in the e-ballot box only paper cast from eligible respondents. Any other access to the e-ballot box should be denied.

12. **Store first paper:** e-voting system should store in the e-ballot box at the submitting server only one paper per respondent (the first received paper).

13. **Submitting server:** should communicate only with the authentic and unaltered client-side submitting software.
• Counting

1. **Holomorphic encrypts for anonymity:** e-voting system should provide holomorphic encrypts for anonymity.

2. **HSM for anonymity:** e-voting system should provide a Hardware Security Module (HSM), which is a tamper-resistant or at least tamper-evident hardware component that can securely generate and store long term secrets for use in cryptography. Generally, it is used to generate a digital key pair without revealing the private key. It can be a function that takes as the input the encrypted e-papers and returns as output the decrypted result, while the decrypted responses are not revealed.

3. **Mix net for anonymity:** e-voting system should provide a mix net for anonymity as a cryptographic alternative to an anonymous channel. It secures who is communicating with whom and it secures the content of the transferred messages.
• **Auditing**

(Audit is the process by which a legal body consisting of election officers and candidates’ representatives can examine the processes used to collect and count the vote, thereby proving the authenticity of the result.)

1. **Audit analysis:** e-voting system should provide audit analysis to audit the total numbers of eligible respondents and issued or qualified ballots provided by the security system with the total number reported. It should also provide trusted data provided by the security process and data provided by the voting process proves that no legitimate votes have been lost by the voting system. It also proves that there is consistency between the number of eligible voters and the spoiled, unspoiled and unused votes as recorded by the e-voting system.

2. **Audit eligible respondents:** e-voting system should provide the total numbers of eligible respondents.

3. **Audit report:** e-voting system should provide the auditing report to prove the authenticity of the result.

4. **Auditing server:** e-voting system should provide an (independent) auditing server for collect and count (specially audit) the result.

5. **Auditing software:** e-voting system should provide auditing software for eligible auditor (mainly) and eligible analyst.
6. **Auditor-oriented software and device**: e-voting system should provide an auditor-oriented software and device independently. It close to the security software and auditing analysis software, as well corresponding device for auditor.

7. **Security Software**: e-voting system should provide a security system and sealing mechanism so that trust can be placed in the seal and hence the sealed data. This implies that the seal should be performed as close to the user submission of the vote as technically possible.

8. **Submitting interface**: e-voting system should provide a submit interface to provide the ballot data for analysis and auditing.
Requirements for common

- **Paper**

  1. **Answer:** provides a response area with some choices, text area, or others.

  2. **Character design:** provides character design and some css style page design for font.

  3. **Display ordering:** provides to display answer options in the default order, ascending order, descending order, and random order.

  4. **N/A (Not Applicable) option:** provides N/A options that answer will not be factored into the weighted average calculations of question in analyzing. If the respondents using it in a matrix or rating question type, this option will not appear until they select the weights.

  5. **Option arrangement:** provides questioner to arrange the option by horizontal/vertical.

  6. **Other field option:** provides selection question type to choose other options, also can fill the value (single text area).

  7. **Page design:** provides theme design and some css style page design.

  8. **Page:** provides to group the questions to enhance the look of the paper, present with separator sometimes.
9. **Presentation**: provides a message or instruction or heading may also help to visually separate sections.

10. **Question**: provides title and presentation for the question.

11. **Section**: provides to group the questions to enhance the look of the paper.

12. **Sub presentation**: provides a message or instruction or heading may also help to explain the presentation.

13. **Textbox arrangement**: provides text box next to question text.

14. **Title**: provide a message for main information.
• Question Types

1. **Drag and drop**: provides a certain set of options to be ranked based upon a specific numbering and presented by a dynamic drag and drop.

2. **Drop down list select one answer**: provides a long list of options and ask the respondent select any one from it.

3. **Image chooser**: provides options and images used to get the respondents preference. It provides single selection (radio button), multiple selections (check box), and a rating with a layout of the options. It should provide to upload video. In addition, it should give some tips for the image or video.

4. **Introduction text question with acceptance checkbox**: provides a check box prefixed of a text.

5. **Multiple choices select many answers**: provides preceded checkbox for respondent to choose multiple predefined sets of options.

6. **Multiple choices select one answer**: provides a radio button for respondent to choose one of a predefined set of options.

7. **Open ended text for multiple rows text**: provides a comment box for respondent in a multiple rows text. This kind of question types **should** provide the number of rows, and **prefix and suffix** to text box.
8. **Open ended text for single text**: provides a comment box for respondent in a single row text. This kind of question types should provide **prefix and suffix** to text box.

9. **Open ended text for size-field text**: provides a comment box for respondent in a specific size-field text. This kind of question types should provide the height and the width of the input text size, and **prefix and suffix** to text box. Sometimes it also provides text box dynamic drag handle at the bottom right corner and presents the pixel of the text box during dragging.

10. **Rank order**: provides respondent to choose and qualify a set of options based on a specific numbering and presented by drop down list after the options with a numeric list.
• Validation

1. **Limit numbers of answer selected**: provides limitation for respondent to limit the number of options. It represent as selection at most/ exactly/least.

2. **Response required**: provides that respondent must respond (default) the question and presents red asterisk before the question.

3. **Text limit for character number**: provides a limit on the number of characters that can be entered in an open ended text question to set up the character limit. This validation must be applied with question required.

4. **Text limit for character style**: provides a validation for text input boxed to validate email address, phone number, and urls.
• Setting

1. **Anonymity:** provides two kinds of ways, firstly, do not record IP addresses of respondents; secondly, do not record the accounts of the respondents by third-party or the systems.

2. **Activation and deactivation:** provides questioner to control the respondents to access to an activity and terminate data collection.

3. **Anti-ballot box stuffing:** provides to prevent respondent to multiple responses.

4. **Automatic numbering:** provides automatic question, section, numbering for matrix style questions.

5. **Back button:** provides to respondent to allow them to back to the previous question.

6. **Copy papers:** provides to copy entire paper including logic.

7. **Countdown timer:** provides questioner to set a time limit on activity to let the respondents have to complete a paper response within a certain period of time.

8. **Display question numbers:** provides to questioner to select whether to display questioners.

9. **Distribution by email:** provides a link via email.

10. **Distribution for Anonymous:** provides to allow anonymous respondents.

11. **Distribution for Track Responses:** provides to allow co-relating
or linking individual responses to the respondents.

12. **Done button text**: provides the questioner to define the done button for a specific text.

13. **Email and password authentication**: provides to protect offer limited security as respondents may share the password with others, and let respondents to login with email and password.

14. **Exit button**: provides to respondent to allow them to exit to response when they are answering.

15. **Exit url**: provides to respondent to allow them to exit to response and jump to the specific url.

16. **Folders**: provides to be categorized into folders.

17. **IP limitation**: provides the IP limitation for allowing the respondent to multiple responses by record the IP of the respondents.

18. **IP recording**: provides to record the IP addresses of the respondents.

19. **Launch and close time**: provides questioner to set up the launch and close time for activation and deactivation.

20. **Multi-languages**: provides questioner or respondent to design or answer in multi-languages.

21. **Multi-level filtering**: provide to drill down in to data and look at segmented reports.
22. **Next button:** provides to respondent to allow they to see the next question.

23. **Participant ID authentication:** provides to protect offer limited security as respondents may share the password with others, and let respondents to login with participant ID.

24. **Participant statistics:** provides to view the overall statistics. It should provide the viewed count, started count, completed count, drop-out count, validation error count, and competition rate.

25. **Password authentication:** provides all the respondents a same password to access the activity. It shall provide the password including in the email invitation.

26. **Password for email detected automatically:** provides to protect offer limited security as respondents may share the password with others, and let respondents to login with different passwords by email invitations automatically.

27. **Real-time report:** provides a quick overview of the real-tile data.

28. **Real-time response collection:** provides a quick overview of the real-time data.

29. **Reminder emails:** provides to sent email batches for whom have used the track url and sent reminders on the setting time.

30. **Report scheduler:** provides a recurring and automated email reports on a periodic basis.
31. **Response Quota control**: provides the respondents to response and in a limited quota.

32. **Save page and continue later**: provides to save the paper midway and continue from where they left at a later time.

33. **Search and replace**: provides a general search and replace for a questionnaire, testing, or voting.

34. **Spelling checking**: provides to check or validate the response is eligible.

35. **SSL secure link**: provides HTTPS paper link.

36. **SSO**: provides single sign-on to allow to authenticate activity account against with third-party system. It should provide SAML, HMAC-SHA1, and DES encrypted.

37. **Time-based data filters**: provides to filter to segment data based on the time ranges.

38. **Username and password authentication**: provides to protect offer limited security as respondents may share the password with others, and let respondents to login with username and password.
Requirements for e-questionnaire

- **Question Types**

1. **Constant Sum**: provides a collection of ratio data to express the relative value or importance of options. It should provide the total points of the total options.

2. **Contact information**: provides consolidated questions asking respondents for their contact information. It shall provide first name, last name, mail address, phone number, and mailing address. It also should provide the validations for the entries, such as mailing address requires a valid zip code, phoned number requires a valid phone number format, and a email address requires a valid email address.

3. **Data reference**: provides to collect or validate zip code data against “standardized” database.

4. **Date and Time**: provides a set of drop down list of date. It should provide setting for select to display date, time, in 12/24 hour format, whether default to today. And the ranges of the year.

5. **Download file question**: provides respondent to download the file.

6. **Dynamic multi-tier lookup table**: provides to represent hierarchies of data.

7. **Graphic Rating**: provides the multiple-choice question with
different options presented with graphic rate. It provides “thumbs up and down” to select yes and no as the only two options, “star rating” and “face rating” intended to represent sentiments from negative to neutral to positive by 5-point rating scale.

8. **Image hotspot:** provides hotspots selection in an image to let respondent to select.

9. **Likert scale:** provides an opinion list toward any given subject. This kind of question types should provide a scale library (Frequency: {Weekly/ Monthly/ Quarterly/ Annually}, Yes/No: {Yes/ No, Yes /No /Maybe, Yes /No/ NA,}, Rating Scale: {Poor-Excellent, Disagree-Agree, Dissatisfied-Satisfied, 1-10 Rating Scale}, Countries: {All countries}, State/ Provides: {countries list}.

10. **Matrix multiple choices select many answers:** provides a collection of various individual questions together for respondent to select multiple options for a line marked as checkbox.

11. **Matrix multiple choices select one answer:** provides a collection of various individual questions together for respondent to select one option for a line marked as radio button.

12. **Push to social:** provides respondents to send positive feedback on a survey to social networking sites, and collects comments or reasons for feedback.
13. **Rating Slider**: provides to produce question and answer styles by a slider, sometimes shows the points of the options.

14. **Side-by-side matrix**: provides to collect data on 2 dimensions or more dimensions for the same options.
• Validation

1. **Dynamic "parent to child" relationships:** provides to validate the relationship of hierarchies of data, for example, validate a zip code and the corresponding address.

2. **Option quota limit:** provides limitation for quota for each option of the selections.

3. **Response quota limit:** provides limitation for the responses of the paper.

4. **Weighting:** provides a value that gives to each of a number of options to show how important it is compared with the others.
• **Logic**

1. **Automatic redirect:** provides to directly jump to the page of the questioner choice when the respondents finishing answering all the questions. It should provide the finish options, such as urls, link text, and message.

2. **Compound skipping:** provides a combination of multiple options of different questions, if satisfy the options then jump to location. It should provide the criteria name, conditions (question, option, selected), relationship of the conditions (and, or), and setting for logic (if criteria is met jump to, and otherwise jump to).

3. **Extraction question:** provides to display the selected options of a multiple select question as an answer option of the next question.

4. **Looping:** provides to loop option, which allows to dynamically looping through a set of questions base on the response to a multiple-choice question.

5. **Quota control:** provides to control how many respondents you want for each option in a question. In addition, if over limit, the questioner can set a default destination for skipping.

6. **Randomization answer options:** provides respondent to display randomly answer options.

7. **Randomization question:** provides common and block question, and randomly display blocks of questions.
8. **Response quota:** provides a quota for the total number of response of the activity.

9. **Scoring logic:** provides to compute scores or points in real-time. It should to provide the scores for the options, and the branching on score.

10. **Show or hide questions:** provides to show or hide question based on the criteria the questioner defined. It should provide criteria name, conditions (question, option, selected), relationship of the conditions, and setting for logic (if criteria is met, do noting/ show/ hide question).

11. **Skip logic:** provides respondents to answer relevant questions based on their answers. It should provide the default destination to question, terminate paper, chain section, chain paper, thank you page.

12. **Skipping for multiple selections:** provides skipping for multiple selections to corresponding jumping to location, and may also assign piping text for the options.

13. **Text piping:** provides to replace piping text (the options selected by respondent) with the appropriate text to the next question. It should provide the piping text to mark the location of the next question text where to be replaced.
Setting

1. **API**: provides automate customer feedback.

2. **Conjoint analysis**: provides to analyze the results of the conjoint question type.

3. **Consolidate Data**: provides to pair the questions across papers form the dropdowns copy data from a source paper to the current one.

4. **Correlation analysis**: provides a non-parametric measure of correlation, using ranks to calculate the correlation.

5. **Custom variable and mapping**: provides to external variables to connect with email information.

6. **Distribution by live url**: provides a unique link for a paper, and the questioner can post this link to let respondents to response, but the responses collected will be anonymous.

7. **Distribution by SNS integration**: provides to connect with the SNS and share the paper to let the respondents to response.

8. **Dropout analysis**: provides to analyze the drop-out rates for each question.

9. **Email action alerts**: provides to notify immediately when the respondent answers in a particular way to questions.

10. **Gap analysis**: provides to analyze for side-by-side matrix question type.
11. **Heat map analysis**: provides to analyze the results of heat map question type.

12. **Interactive response**: provides to automatic take respondent input without the respondents having to click on continue or submit button on each page.

13. **Multiple criteria segmentation**: provides a mechanism to allow for data segments across multiple questions.

14. **Report sharing**: provides a web link with optional password protection to share the results in public.

15. **SNS authentication**: provides the respondents to connect with SNS count and record the information about UID, full name, locale, gender, link as the data.

16. **Text analysis**: provides to set up text categories with some keywords.

17. **Thank you emails**: provides to sent automatic thank you emails to respondents who complete answering.

18. **Trend analysis**: provides a look at data over time for a long-running paper.

19. **Weighting and balancing**: provides to adjust the data to account for sample bias.
Requirements for e-testing

● Question Types

1. Connect points: provides a question type for connecting points from different groups.

2. Download file question: provides respondent to download the file.

3. Formula: provides a series of letters to support respondent to answer some specific question type for programing, chemical, and mathematics.

4. Image hotspot: provides hotspots selection in an image to let respondent to select.

5. Matrix multiple choices select many answers: provides a collection of various individual questions together for respondent to select multiple options for a line marked as checkbox.

6. Matrix multiple choices select one answer: provides a collection of various individual questions together for respondent to select one option for a line marked as radio button.

7. Upload file question: provides respondents to upload files with their response. It has the limitation for file types and extension.
 Validation

1. **Scoring**: provides a score standard for each options or questions.

 Logic

1. **Automatic redirect**: provides to directly jump to the page of the questioner choice when the respondents finishing answering all the questions. It should provide the finish options, such as urls, link text, and message.

2. **Randomization answer options**: provides respondent to display randomly answer options.

3. **Randomization question**: provides common and block question, and randomly display blocks of questions.

4. **Scoring logic**: provides to compute scores or points in real-time. It should to provide the scores for the options, and the branching on score.

5. **Show or hide questions**: provides to show or hide question based on the criteria the questioner defined. It should provides criteria name, conditions (question, option, selected), relationship of the conditions, and setting for logic (if criteria is met, do noting/ show/ hide question).

6. **Skip logic**: provides respondents to answer relevant questions based on their answers. It should provide the default destination to
question, terminate paper, chain section, chain paper, thank you page.

7. **Skipping for multiple selections:** provides skipping for multiple selections to corresponding jumping to location, and may also assign piping text for the options.
Setting

1. **Conjoint analysis**: provides to analyze the results of the conjoint question type.

2. **Consolidate Data**: provides to pair the questions across papers form the dropdowns copy data from a source paper to the current one.

3. **Correlation analysis**: provides a non-parametric measure of correlation, using ranks to calculate the correlation.

4. **Custom variable and mapping**: provides to external variables to connect with email information.

5. **Distribution by live url**: provides a unique link for a paper, and the questioner can post this link to let respondents to response, but the responses collected will be anonymous.

6. **Distribution by SNS integration**: provides to connect with the SNS and share the paper to let the respondents to response.

7. **Dropout analysis**: provides to analyze the drop-out rates for each question.

8. **Heat map analysis**: provides to analyze the results of heat map question type.

9. **Interactive response**: provides to automatic take respondent input without the respondents having to click on continue or submit button on each page.
10. **Multiple criteria segmentation:** provides a mechanism to allow for data segments across multiple questions.

11. **Report sharing:** provides a web link with optional password protection to share the results in public.

12. **SNS authentication:** provides the respondents to connect with SNS count and record the information about UID, full name, locale, gender, link as the data.

13. **Text analysis:** provides to set up text categories with some keywords.

14. **Thank you emails:** provides to sent automatic thank you emails to respondents who complete answering.
Requirements for e-voting

- **Question Types**

1. **Contact information**: provides consolidated questions asking respondents for their contact information. It shall provide first name, last name, mail address, phone number, and mailing address. It also should provides the validations for the entries, such as mailing address requires a valid zip code, phone number requires a valid phone number format, and a email address require a valid email address.

2. **Data reference**: provides to collect or validate zip code data against “standardized” database.

3. **Dynamic multi-tier lookup table**: provides to represent hierarchies of data.
- **Setting**

1. **Candidate nomination:** provides respondent to nominate or propose candidates.

2. **Election counting:** provides to count the ballots.

3. **Participant authority:** provides to authorize each role of participant (especially executor).

4. **Respondent authentication:** provides to authenticate the respondents by different kinds of methods.

5. **Respondent registration:** provides to register the respondent information.

6. **Vote auditing:** provides to audit the ballots and respondents.

7. **Vote confirmation:** provides to let respondents to confirm their ballots after submitting the ballot (log out cannot).

8. **Voting anonymity:** provides respondent to answer anonymously. It differs from anonymity represented by not recording IP addresses and account information. It should be aligned with respondent registration.
An Example of QSL for a Questionnaire
### Details of a Questionnaire

**Survey Title:** A Questionnaire about Hangzhou City in China  
**Description:** Special Insurance Policies of China Life Ltd.

#### Basic Setting

- **Area:** Nanjing, Beijing  
- **Age:** 20-26  
- **Family Name:** Gao  
- **Sex:** Male  

**Insured Date:** 2016/05/01 — 2016/08/01

#### Questionnaire Sheet

- **Skipping Logic:**  
  - [ ] Skipping  
  - [ ] None  

**Questionnaire Template:** Normal (Selected)

**Q1 to Q3:** Greetings

- **What is your gender?**  
  - [ ] M (Tips: Male, Skipping to: Q3)  
  - [ ] F (Tips: Female, Skipping to: Q5)

**Q4 to Q10:** Multiple Selections

- **What kinds of following ways to travel you will choose?**  
  - [ ] Walking (Tips: Male)  
  - [ ] Bus (Tips: Male)  
  - [ ] Bicycle (Tips: Male)  
  - [ ] Own Car (Tips: Male)  
  - [ ] Tricycle (Tips: Male)  
  - [ ] Train (Tips: Male)
### Details of a Questionnaire

#### Basic Setting

**Survey Title:** A Questionnaire about Hangzhou City in China  
**Description:** Special Insurance Policies of China Life Ltd.

#### Sampling Setting

- **Area:** Nanjing, Beijing
- **Age:** 20 - 26
- **Family Name:** Gao
- **Sex:** Male
- **Insured Date:** 2016/05/01 - 2016/08/01

#### Questionnaire Sheet

- **Skipping Logic:** Skip
- **Questionnaire Template:** Normal (Selected)

#### Q1 to Q3: Greetings

What is your gender?

- **M**  
  - Tips: Male  
  - Skipping to: Q3
- **F**  
  - Tips: Female  
  - Skipping to: Q5

#### Q4 to Q10: Multiple Selections

What kinds of following ways to travel you will choose?

- **Walking**  
  - Tips: Male
- **Bus**  
  - Tips: Male
- **Bicycle**  
  - Tips: Male
- **Own Car**  
  - Tips: Male
- **Tricycle**  
  - Tips: Male
- **Train**  
  - Tips: Male

[Preview]
Basic Setting

Survey Title: A Questionnaire about Hangzhou City in China  Description: Special Insurance Policies of China Life Ltd.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Paper id="paper001" type="questionnaire"
xmlns="org:aise:name:zy:schema:qsl"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <Description id="des001" type="header" value="A Questionnaire about Hangzhou City in China"/>
  <Description id="des002" type="paragraph" value="Special Insurance Policies of China Life Ltd."/>
  <Section id="section001" />  
  <Section>
  <Section id="section002" />
  <Logic>
  <Logic>
  </Logic>
  </Logic>
</Paper>
```
Survey Title: A Questionnaire about Hangzhou City in China

Description: Special Insurance Policies of China Life Ltd.

Sampling Setting

Area: Nanjing, Beijing  Age: 20-26  Family Name: Gao  Sex: Male

Insured Date: 2016/05/01 — 2016/08/01

Questionnaire Sheet

Skipping Logic: ○ Skipping  ○ None  Questionnaire Template: Normal (Selected)

Q1 to Q3: Greetings

What is your gender?

- M  Tips: Male  Skipping to: Q3
- F  Tips: Female  Skipping to: Q5

Q4 to Q10: Multiple Selections

What kinds of following ways to travel you will choose?

- Walking  Tips: Male
- Bus  Tips: Male
- Bicycle  Tips: Male
- Own Car  Tips: Male
- Tricycle  Tips: Male
- Train  Tips: Male
Q1 to Q3: Greetings

What is your gender?

- **M** Tips: Male
- **F** Tips: Female

Skipping to: Q3

Skipping to: Q5

```xml
<Section id="section001">
  <Question id="question001" type="multiple choices" isMandatory="true">
    <Limitation minOccur="1" maxOccur="1"/>
    <Description id="des003" type="header" value="What is your gender?"/>
    <Answer id="answer001"/>
    <Description id="des004" type="header" value="M"/>
    <Description id="des005" type="tips" value="Male"/>
  </Question>
  <Answer id="answer002"/>
  <Question id="question002">
    <Description id="des006" type="header" value="F"/>
    <Description id="des007" type="tips" value="Female"/>
  </Question>
  <Answer id="answer003"/>
</Section>
```
### Details of a Questionnaire

**Basic Setting**

**Survey Title:** A Questionnaire about Hangzhou City in China  
**Description:** Special Insurance Policies of China Life Ltd.

**Sampling Setting**

- **Area:** Nanjing, Beijing  
- **Age:** 20 - 26  
- **Family Name:** Gao  
- **Sex:** Male  
- **Insured Date:** 2016/05/01 - 2016/08/01

**Questionnaire Sheet**

- **Skipping Logic:**  
  - [ ] Skipping  
  - [ ] None  

**Questionnaire Template:** Normal (Selected)

Q1 to Q3: Greetings

- **What is your gender?**  
  - [ ] M  
    - Tips: Male  
  - [ ] F  
    - Tips: Female  

- **Skipping to:** Q3  
- **Skipping to:** Q5

Q4 to Q10: Multiple Selections

- **What kinds of following ways to travel you will choose?**  
  - [ ] Walking  
    - Tips: Male  
  - [ ] Bus  
    - Tips: Male  
  - [ ] Bicycle  
    - Tips: Male  
  - [ ] Own Car  
    - Tips: Male  
  - [ ] Tricycle  
    - Tips: Male  
  - [ ] Train  
    - Tips: Male

**Preview**
### Details of a Questionnaire

**Basic Setting**

**Survey Title**: A Questionnaire about Hangzhou City in China  
**Description**: Special Insurance Policies of China Life Ltd.

**Sampling Setting**

- **Area**: Nanjing, Beijing  
- **Age**: 20 - 26  
- **Family Name**: Gao  
- **Sex**: Male  
- **Insured Date**: 2016/05/01 to 2016/08/01

**Questionnaire Sheet**

**Skipping Logic**:  
- Skipping Logic: Skipping  
- Questionnaire Template: Normal (Selected)

**Q1 to Q3**: Greetings

- **What is your gender?**
  - M - Tips: Male  
  - F - Tips: Female  

**Q4 to Q10**: Multiple Selections

- **What kinds of following ways to travel you will choose?**
  - Walking - Tips: Male  
  - Bus - Tips: Male  
  - Bicycle - Tips: Male  
  - Own Car - Tips: Male  
  - Tricycle - Tips: Male  
  - Train - Tips: Male
What kinds of following ways to travel you will choose?

- Walking
- Bus
- Bicycle
- Own Car
- Tricycle
- Train
An Example of QSL for an E-Voting System
<?xml version="1.0" encoding="UTF-8"?>
<QSL schemaVersion="3.1"
xmlns="org:aise:name:zy:schema:qsl"
xmlns:n1="org:aise:name:zy:schema:qsl"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="org:aise:name:zy:schema:qsl QSLversion3.1/100-qsl-v3.1.xsd">
  <System>
    <Security/>
    <Authentication id="auth001" role="respondent"/>
    <Authority id="access001" role="sponsor" accessRule="job card" type="authenticate" range="registering"/>
  </Security>
  <Environment>
    <Server id="server001" purpose="registering"/>
    <Server id="server002" purpose="registering"/>
    <Server id="server003" purpose="submitting"/>
    <Server id="server004" purpose="submitting"/>
    <Device id="device001" type="PC"/>
    <Device id="device002" type="PC"/>
    <Device id="device003" type="smartphone"/>
    <Software id="soft001" ref="server001" purpose="auditing" role="auditor">
      <Solution type="desktop.app" name="others"/>
    </Software>
  </Environment>
  <Function/>
</QSL>
An Example of QSL for an E-Testing System
General-purpose Offline e-testing environment [Wang et al. 2015]
An Example of QSL (1/2)

<?xml version="1.0" encoding="UTF-8"?>
<QSL version="3.1" xml:lang="en" xmlns="http://www.aise.ics.saitama-u.ac.jp/qsl"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <System>
    <Security/>
    <Environment>
      <Server id="server001" purpose="registering"/>
      <Server id="server002" purpose="submitting"/>
      <Gateway id="ip001" ref="device001" ip="10.0.5.1"/>
      <Interface id="interface001" ref="device003"/>
      <Device id="device001" type="PC" memory="8G" cpu="Intel Core i5" ref="soft001"/>
      <Device id="device002" type="access point" encryption="WPA2-PSK" maxConnection="50"/>
      <Device id="device003" type="usb" edition="3.0" capacity="8G"/>
      <Software id="soft001" ref="server002" purpose="submitting" role="respondent"/>
        <Solution type="browser" name="chrome"/>
      </Software>
      <Software id="soft002" ref="server002" purpose="submitting" role="respondent"/>
        <Solution type="database" name="DB2" edition="8.1.6" data="participant"/>
    </Software>
  </Environment>
</QSL>
### An Example of QSL (2/2)

```xml
<Function>
  <Phase>
    <SettingUp>
      <Import id="func001" role="questioner" scope="paper and setting" format="qsl"/>
    </SettingUp>
    <Distributing>
      <Distribute id="func002" role="questioner" channel="usb"/>
    </Distributing>
    <Registering>
      <Authenticate id="func003" role="respondent" method="token"/>
    </Registering>
    <Submitting>
      <Ping id="func004" role="monitor" scope="ip"/>
      <Observe id="func005" role="monitor" scope="state"/>
      <Reply id="func006" role="respondent"/>
      <Submit id="func007" role="respondent"/>
      <Stop id="func008" role="respondent" method="interval" frequency="1" autoSave="yes"/>
    </Submitting>
    <Collecting>
      <Collect id="func009" channel="usb"/>
      <Integrate id="func010" role="questioner" scope="response" format="zip"/>
    </Collecting>
    <Marking>
      <Mark id="func011" role="marker" scope="response"/>
    </Marking>
  </Phase>
</Function>
</QSL>
```