



# seriss

SYNERGIES FOR EUROPE'S  
RESEARCH INFRASTRUCTURES  
IN THE SOCIAL SCIENCES

## Annex 1

### Questionnaire Design and Documentation Tool for the European Social Survey

#### Technical documentation

#### Revision History

Name	Date	Reason For Changes	Version
Tool name changed	July 10 <sup>th</sup> 2019	Conceptual improvement	v1.1

## Contents

1. Introduction	3
1.1 Purpose of the QDDT	3
1.2 Development under SERISS	3
1.3 Current status of the tool	4
1.4 Further information	4
1.5 Definitions, Acronyms and Abbreviations	4
1.6 Acknowledgements	5
2. Software Requirements Specification	6
2.1 Software Perspective	6
2.2. Software functions	7
2.3 User Roles and Characteristics	8
2.4 Operating Environment	10
2.5 Design and Implementation Constraints	10
2.6 User Documentation	11
2.7 Performance Requirements	12
3. Use cases	13
3.1 Overview	15
3.2 Detailed use case diagrams	17
4. System Architecture and Design	2
4.1 System Architecture	2
4.1.1 System Hardware Architecture	2
4.1.2 System Software Architecture	2
4.1.3 File and Database Design	3
4.1.4 Database Management System Files	3
4.1.5 Non-Database Management System Files	3
4.2 Human Machine Interface	4
4.2.1 Inputs	4
4.2.2 Outputs	4
4.3 Detailed Design	5
4.3.1 Hardware Detailed Design	5
4.3.2 Software Detailed Design	5
4.4 External Interfaces	5
4.5 System Integrity Controls	6
5. Software Testing	6
6. Version Description	6
6.1 Installation Instructions	6
6.2 Possible Problems and Known Errors	7

# **1. Introduction**

## **1.1 Purpose of the QDDT**

The aim of the Questionnaire Design and Documentation Tool is to provide an interactive and dynamic web-based tool which can be used to both document and retrieve information on the complex process of designing a cross-national survey questionnaire. It will provide a more streamlined and user friendly replacement for the current paper-based word template used to document the European Social Survey's questionnaire design process.

The QDDT will improve the efficiency of the questionnaire design process. The tool will provide a means of managing the design process, tracking the development of question items over different stages of pre-testing and review and incorporating input from multiple stakeholders. The QDDT should also provide the basis for generating the final field ready questionnaire from the question items developed and recorded within the tool.

The QDDT can also improve the transparency of the questionnaire design process and facilitate knowledge transfer both within the same survey infrastructure across time and across different survey infrastructures. The tool will serve as an archive of survey metadata, including the rationale behind questions and results from pre-testing as well as the final questions developed, providing a complete and searchable record of the questionnaire design process.

The tool is intended to form part of a suite of tools documenting different stages of the survey lifecycle including questionnaire design and translation. The QDDT's underlying database for the tool is structured in line with the internationally coordinated Data Documentation Initiative (DDI). This will make it possible to import and export data from other DDI-compliant tools being developed under SERISS including a Translation Management Tool (TMT) and Question Variable Database (QVDB).

The QDDT is being developed primarily with the European Social Survey in mind. However, it is anticipated that the tool will also be usable by other national and cross-national surveys including SERISS partners SHARE, GGP and EVS and the International Social Survey Programme (ISSP). The tool's reusable database model can also be extended to other questionnaire development and documentation tools.

## **1.2 Development under SERISS**

Development of the QDDT began under the EU funded FP7 project Data Service Infrastructure for the Social Sciences and Humanities (DASISH - GA No: 283846). The original aim for the Synergies for Europe's Research Infrastructures in the Social Sciences (SERISS) project was to test the QDDT prototype developed under DASISH, using the tool to document the end of the ESS Round 8 questionnaire design process and populate the tool with questionnaire items from past ESS rounds (Rounds 4-7). However, in response to recent and ongoing software and computing developments a decision was taken at the start of the SERISS project to redevelop the tool started under DASISH using newer and better performing software (Angular 6, Material Design). The new software should have resulted in a better end product since the new development platform ensures better integration of the tool, is widely used and is also strongly related to other popular frameworks (e.g. Bootstrap).

This documentation sets out the user requirements and technical specification for the QDDT tool as it has been developed under SERISS, building on existing documentation from the DASISH project where relevant.

## 1.3 Current status of the tool

Version 1 of the tool, released in May 2018, incorporates most but not all of the core requirements (see the use case diagram in Section 3 for further details). These functionalities are sufficient to enable researchers at ESS ERIC HQ to document the different stages of the ESS questionnaire development process and to share content with other ESS stakeholders for review/comment. It is possible to enter questionnaire concepts, question items and question constructs, to search for elements entered into the tool, to comment on items in development, to upload supporting background documentation into the tool, to version elements and track their development history and use over time, and to reuse elements across surveys and survey rounds. It is also possible to “publish” subsets of elements for comment/review and to assign these published elements to key development milestones in the design process, for example to identify question items included in pre-testing or piloting. However, it is not yet possible to use the tool to produce a final field -ready questionnaire.

Version 1 of the tool focuses primarily on the needs of internal stakeholders involved in the questionnaire development process. Some additional work may be necessary to develop the user interface for external users. However, key functionalities necessary for an external facing tool – e.g. search capability, publication milestones, different user access rights - are already in place.

Version 1 of the tool currently operates as a standalone tool. The facility to import/export data from the other questionnaire documentation tools (TMT, QVDB) being developed under SERISS has not yet been implemented in practice. However, given that these tools have been developed according to a common metadata model, agreed as part of the DASISH project, interconnectivity and communication between the tools should be possible.

## 1.4 Further information

The current QDDT prototype and source code are available to view open access on GitHub: <https://github.com/DASISH/qddt-client/wiki>

More information on work undertaken as part of the DASISH project can be found here: <https://github.com/DASISH/QDDT>

To learn more about the European Social Survey: [www.europeansocialsurvey.org](http://www.europeansocialsurvey.org)

More information about the Data Documentation Initiative (DDI): <http://www.ddialliance.org/>

## 1.5 Definitions, Acronyms and Abbreviations

CST	ESS Core Scientific Team
ESS	European Social Survey
DDI	Data documentation Initiative
NC	ESS National Coordinator

QDDT	Questionnaire Design and Documentation Tool
QDT	Questionnaire Design Team
QVDB	Question Variable Data Base
TMT	Translation Management Tool

## 1.6 Acknowledgements

The QDDT was conceived as part of the EU funded FP7 project Data Service Infrastructure for the Social Sciences and Humanities (DASISH - GA No: 283846) and has been developed under the Horizon 2020 project Synergies for Europe's Research Infrastructures in the Social Sciences (SERISS - GA No: 654221).

The tool is being developed by NSD – Norwegian Centre for Research Data, home of the ESS Data Archive, in consultation with members of the ESS Core Scientific Team based at ESS ERIC Headquarters.

The following people have been involved in the development of the QDDT:

At ESS ERIC Headquarters, City, University of London: Sarah Butt, Yvette Prestage, Virginia Ros, Sally Widdop

At NSD: Stig Norland, Yong Liu, Dag Øyvind Heradstveit, Håvard Venge Bakkmoen, Benjamin Beuster, Hilde Orten, Knut Kalgraff Skjåk

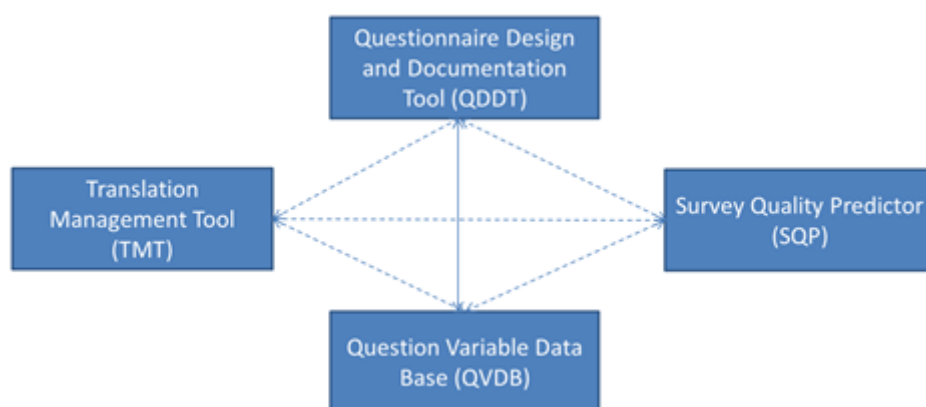
Consultant: Joachim Wackerow

## 2. Software Requirements Specification

### 2.1 Software Perspective

The process of designing a conceptual questionnaire for a scientifically rigorous cross-national survey is a long and complex one.<sup>1</sup> Carefully documenting this process is important not only to ensure the effective realisation of a final questionnaire but also to provide survey users with valuable metadata for future reference. However, many surveys are currently reliant on paper-based tools to document the questionnaire development process. The European Social Survey, for example, relies on a word template which can have as many as 15-20 iterations and run to 150 pages or more in length. The template is time consuming to complete, cumbersome to use when retrieving information, and at risk of information being lost during manual updates.

The QDDT provides a new interactive and dynamic web-based tool which survey stakeholders can use to enter and retrieve detailed metadata about the questionnaire design process. The QDDT is one of a set of tools being developed to document key stages of the survey lifecycle and to provide users with important and insightful metadata about survey questions and their provenance. Two other tools are funded under the DASISH and SERISS projects: a Translation Management Tool (TMT) to document the translation process and a Question Variable Database (QVDB) which will document the variables resulting from the fielded questionnaire. In DASISH work was carried out with the aim is for these tools to communicate efficiently with one another and enable metadata collected at one stage of the survey development process to inform work at a later stage of the development process (see Figure 1). The possibility of incorporating a fourth tool – Survey Quality Predictor (SQP) – already in existence and used by surveys including the ESS to evaluate question quality - into this system should also be explored.



**Figure 2.1 SERISS web-based tools for documenting the survey lifecycle, aims for communication between the tools**

It is intended that questionnaire metadata exported to and stored within the QVDB can be made available via data repositories and/or question banks such as those maintained by CESSDA ERIC.

There are other commercial and open source tools currently available and/or in development which allow for questions to be documented and questionnaires to be produced. These

---

1

include:

- Colectica: The Colectica platform consists of several software tools for viewing, creating, and managing of metadata. Colectica is a commercial software, developed by Algenta Technologies.
- Rogatus QMMS, QBee: Rogatus is an open source toolset currently in development at DIPF with support of GESIS, TBA21 and OPIT. Rogatus consists of different DDI compliant applications (e.g. Questionnaire Builder, Translation Builder, Metadata Builder, Rogatus Portal) to support a multitude of survey processes. The Questionnaire Builder (Qbee) of the Rogatus QMMS is an open-source tool in development for creating paper & pencil as well as computer-based questionnaires for social sciences or educational projects.
- The Questionnaire editor has been developed by GESIS to run surveys on the German national election in 2017 within the GLES project. It enables to create questions, question grids and logical blocks as main components. Answer domains, interviewer instructions and filter statements are managed separately for reuse. The tool is open source and will be further developed to support large studies in general.
- Questionnaire Generator, under developement by Insee, is an open source survey questionnaire generator which takes as input DDI 3.2 files and produces web forms (Xforms format).

However, these tools focus primarily on the documentation and sharing of final questionnaire items rather than tracking in detail the development of questionnaire items from their initial conception through multiple stages of pre-testing and external review. The QDDT will allow for each stage of the questionnaire design process to be documented as well as facilitating the exchange of survey metadata on other stages of the survey lifecycle including translation and variable creation.

The QDDT is being developed primarily with the European Social Survey in mind. However, it is anticipated that the tool will also be usable by other national and cross-national surveys for producing conceptual questionnaires, including SERISS partners SHARE, GGP and EVS and the International Social Survey Programme (ISSP).

## **2.2. Software functions**

The QDDT is intended to:

- Be a tool for survey infrastructures to manage and document the entire questionnaire design process in real time. This requires:
  - Capacity to add, edit, delete and store different questionnaire meta elements (concepts, question items, response scales etc.) to a fine level of granularity;
  - A database structure which allows for question items to be organised by theoretical concepts and provides space to input information on the rationale behind asking certain questions;
  - Ability to version questionnaire elements and to track/compare different versions of the same element over time;
  - Ability for multiple stakeholders to review and comment on elements in development at multiple stages in the design process;

- Capacity to monitor questionnaire development progress and determine elements' status at different stages of the development cycle (e.g. pre-testing, piloting, mainstage);
- Upload of background materials e.g. report on pre-testing results in common file formats (word, pdf).
- Provide an archive of existing questionnaires and questionnaire metadata for exploration and reuse. This requires:
  - Capacity to reuse elements across multiple surveys and/or rounds within the same survey and to track in which surveys and/or studies an element has been used;
  - Boolean field level search in all elements within selected fields;
  - Download and printing of search results/user defined content in human readable format (pdf, other);
  - Programme access to search results in machine readable formats (DDI-XML, DDI-RDF). Work on the DDI export from the tool is however not yet complete.

Further goals for the QDDT are to:

- Support the production of field-ready questionnaires in different modes (e.g. PAPI, CAPI, CAWI) and enable these questionnaires to be exported in human readable (pdf) or machine readable (DDI-XML) format.
- Exchange information with other online tools documenting other stages of the survey lifecycle, including the Translation Management Tool and Question Variable Database, also being developed under SERISS.

The development of these features are ongoing, but needs further funding to be completed.

## 2.3 User Roles and Characteristics

There are two main stakeholder groups for the QDDT.

**Internal stakeholders:** This covers people involved in the questionnaire development process for a particular survey. For the European Social Survey this includes:

- The questionnaire design team at ESS ERIC Headquarters who oversee the questionnaire development process. These users will have primary responsibility for documenting questionnaire development within the QDDT (2-3 people);
- Other members of the ESS Core Scientific Team who contribute to the questionnaire design process including the translation expert and researchers responsible for SQP coding (a further 4-5 people);
- Members of the external Questionnaire Design Teams appointed to develop the rotating modules in each round (4-5 people per module);
- ESS National Coordinators who oversee the implementation of the ESS in participating countries and who comment on the feasibility of fielding the questionnaire in different country contexts (25-30 NCs).

Internal stakeholders will require access to the QDDT at different stages in the questionnaire's development process in order to be able to add or edit content, to view content and to comment on content added by other users. Some internal stakeholders will be heavily involved in the survey questionnaire design process and be interacting regularly with the tool e.g. the questionnaire design team at ESS ERIC HQ. Other stakeholders may be relatively unfamiliar with the ESS questionnaire design process (e.g. members of the external questionnaire design team) or be involved in the questionnaire design process only intermittently (e.g. NCs). The QDDT should be able to meet the needs of both heavy and light users.



In order to effectively manage the questionnaire development process and minimise the risk of multiple users overwriting content or introducing errors and inconsistencies, (at least) two different types of internal user will be distinguished: those with editing rights (i.e. the ability to add/amend questionnaire content within the tool) and those with viewing/commenting rights only.

Access rights for internal stakeholders should be set on a per survey (agency) basis so that, for example, members of the ESS CST can view and comment on the development of the ESS but cannot do the same for other surveys e.g. ISSP which may be developed within the QDDT. Given that the ESS appoints different QDTs in each round some thought should also be given as to whether it is necessary to assign access rights on a per round (study) basis to avoid QDTs viewing internal material from previous rounds.<sup>2</sup>

**External stakeholders:** This covers people not directly involved in the production of a particular survey but who may nevertheless be interested in finding out more about the content of that survey and retrieving metadata on how and why specific question items were developed. This might include:

- Researchers using the survey for substantive analysis and interested in the conceptual background of the items being used in analysis;
- Researchers using the survey for methodological research and interested in the development and pre-testing of particular questionnaire items;
- Teachers and students of questionnaire design;
- Survey designers looking for existing questions to use or adapt for other surveys.

Internal stakeholders should be able to choose what content from the QDDT to make available for external stakeholders. External stakeholders will be able to search for and view this content within the QDDT as well as to export/print the published content for use outside the tool. External publication will take place once the development process for a particular study is completed. Once details of a study have been made available externally, it should no longer be possible to edit the content within the QDDT.

Roles and access rights for the different user groups are summarised in the table below.

**Table 2.1 User Access rights in the QDDT**

	Developer	Internal stakeholder			External stakeholder
Role	Tool admin	Editor	Conceptual editor	Viewer	Guest
e.g. stakeholder	NSD	ESS ERIC HQ	QDT, CST,	NCs	ESS data users
Roles					
Add users	✓	x	x	x	x
<b>WRITE ACCESS</b>					
Write elements: Modules, concepts (upload docs, add, edit, delete, version, publish)	✓	✓	✓	x	x
Write elements: Question items, categories, constructs, missing,	✓	✓	x		x

<sup>2</sup> Unauthorised access to future rounds can be avoided by disabling QDDT access rights following completion of a given round.

domains (upload docs, add, edit, delete, version, publish)				x	
Comment on developments within tool	✓	✓	✓	x	x
<b>READ ACCESS</b>					
View internally published elements and supporting documents for a given survey	✓	✓	✓	✓	x
Search all content for a given survey within QDDT	✓	✓	✓	✓	x
View externally published elements and supporting documents for all surveys	✓	✓	✓	✓	✓
Search all archived content within QDDT	✓	✓	✓	✓	✓
Export content from QDDT in human- readable format (PDF, other), as well as in a machine- actionable structured format (DDI-XML, DDI-RDF).	✓	✓	✓	✓	✓

## 2.4 Operating Environment

The QDDT is designed as a web-based tool. The tool is intended for use on a desktop computer running an up to date web browser. It is not a priority to develop versions for tablets or other mobile devices given that it is envisaged that most users will be in an office environment. The current version of the tool has been developed and tested using Chrome. However, subsequent versions should be compatible with other web browsers including Safari.

## 2.5 Design and Implementation Constraints

As discussed in Section 2.1 above, the QDDT is being developed as one of a suite of tools to document survey metadata at different stages of the survey lifecycle. The tool's development is constrained by the requirement for a common metadata model to enable communication between the tools and the decisions taken regarding this common model as part of the DASISH project. See:

Common metadata model: [Towards a common metadata model for the three DASISH tools.pdf](#)

Versioning of translations: [Versioning of translations in the TMT and the QVDB.pdf](#)

This means that:

- DDI-Lifecycle 3.2 XML will be used as the basis for the exchange of metadata between the tools. *The Data Documentation Initiative (DDI) is an internationally acknowledged metadata specification describing data from the social, behavioral, and economic sciences* (<http://www.ddialliance.org/>);

Further information on the DDI profiles for key elements in v1 of the QDDT can be found here:

Question item: [QDDT master profile QuestionItem v1.xml](#)

Question construct: [QDDT master profile QuestionConstruct v1.xml](#)

Sequence: [QDDT master profile Sequence v1.xml](#)

- The tool relies on rich metadata with a high level of granularity to allow for maximal flexibility in reuse in survey business processes. This means for example that Question Items in the QDDT will contain only the most reusable parts of a question, like the question text and a response domain and not other elements such as instructions which may be questionnaire specific. In order to maximize the reusability of metadata elements metadata elements will be included in other elements by reference rather than inline in the metadata element, if possible. This means for example that a response domain (for example a code list) is included by reference in a question, rather than 'belonging' in the question itself.
- The QDDT employs simple 2-level business versioning to give the user control over the versioning of elements and enable them to keep track of developments over time. A flexible approach to versioning - with a combination of early and late binding and the opportunity to designate items as "work in progress" - has been implemented to take account of the complexity of the questionnaire design process and the variety of workflows that might be encountered.

#### [More on versioning in the QDDT](#)

In line with the conditions attached to EU funding the tool should be developed using open source software and should be made available to end users free of charge. To this end the QDDT uses a MIT License, and all libraries in use, are free of charge for non-commercial entities.

<https://github.com/DASISH/qddt-client/blob/devel/LICENSE>

## 2.6 User Documentation

The tool should be accompanied by a series of user guides tailored to the different types of internal and external users envisaged. Guidance on some topics such as versioning may be survey specific depending on the policies adopted by individual surveys.

The QDDT ESS User Manual and Conventions Document

## 2.7 Performance Requirements

The QDDT should be able to cope with multiple concurrent users. Although initially intended for use on the ESS only the tool may later be extended for use by other surveys. This may mean that up to 100 internal survey stakeholders may be looking to use the tool to input/edit data at any one time. More significantly, the tool will be open to external stakeholders (i.e. survey users and other researchers) and multiple users may wish to search the database and download information from it at the same time.

The QDDT will need to be able to store a significant amount of information in the form of survey metadata. The ESS alone has developed over 600 individual question items in its first eight rounds with each question item having multiple versions. Users should be able to search the QDDT's database and retrieve information quickly.

### 2.7.1 Software quality attributes

The following attributes for the QDDT are considered important:

Usability: The QDDT will be used by a wide range of different stakeholders with different levels of familiarity with survey questionnaire design and who may interact with the tool only occasionally or even on a one off basis. The tool should be accessible to anyone familiar with web applications.

Reliability: The tool should be reliable and readily available for users to input data or search the database when required. The system may be taken down when maintenance is needed. However, this should be done infrequently. Any problems will be dealt with within normal office hours.

Adaptability: The QDDT is being developed in the first instance as a tool to document the ESS questionnaire design process. However, it should also be possible for other survey infrastructures to use the QDDT to document their questionnaire development. This means that the structure of the tool and the terminology used should be kept as generic as possible.

### 2.7.2 Safety Requirements

The QDDT is intended as a repository of questionnaire metadata over time. Therefore it is essential that data stored within the tool is backed up and will not be lost in the event of a software or hardware failure.

Information backup will be even more important for surveys during the questionnaire development phase when the QDDT is likely to be the only record of questionnaire items being considered.

### 2.7.3 Security Requirements

Access to the QDDT will be via secure login (username and password) with user roles and access rights defined as in Section 2.1.3 above. This is to ensure that unauthorised users cannot edit content within the tool. It also ensures that public access is restricted to final questionnaires only and that unauthorised users cannot view questionnaires still in development and therefore liable to change.

As the QDDT will not be handling personal data there is no need for the tool to run on a secure server.

### 3. Use cases

This section contains a set of use cases for the QDDT, starting with a high level diagram in 3.1 followed by more detailed use cases of specific actions in section 3.2.

The use case elements used in the diagrams are:

#### Actors



There are four main actors: Editors, Conceptual editors, Viewers and Guests. Within the ESS these roles are assigned as follows:

**Editors:** Researchers at ESS ERIC Headquarters, responsible for managing the questionnaire design process.

**Concept editors:** The Questionnaire Design Teams responsible for the rotating modules in each round and other members of the ESS Core Scientific Team (e.g. Translation expert) involved in the questionnaire design process.

**Viewers:** National Coordinators responsible for the implementation of the survey in each participating country and other internal stakeholders.

**Guests:** External stakeholders interested in the ESS questionnaire design process once a round has been completed.

#### Use case

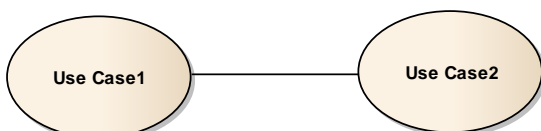


#### Collaboration



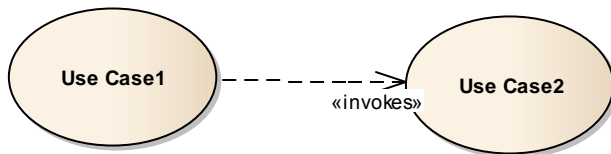
Use case with dotted line: Collaboration i.e. can be performed by more than one type of actor

#### Association



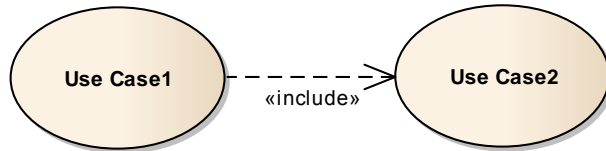
Use case relationship 'Association'

#### Invokes



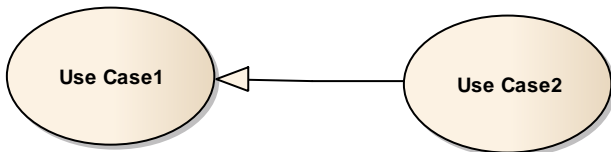
A use case is invoked by another one

**Include**



A use case includes another use case

**Generalize**



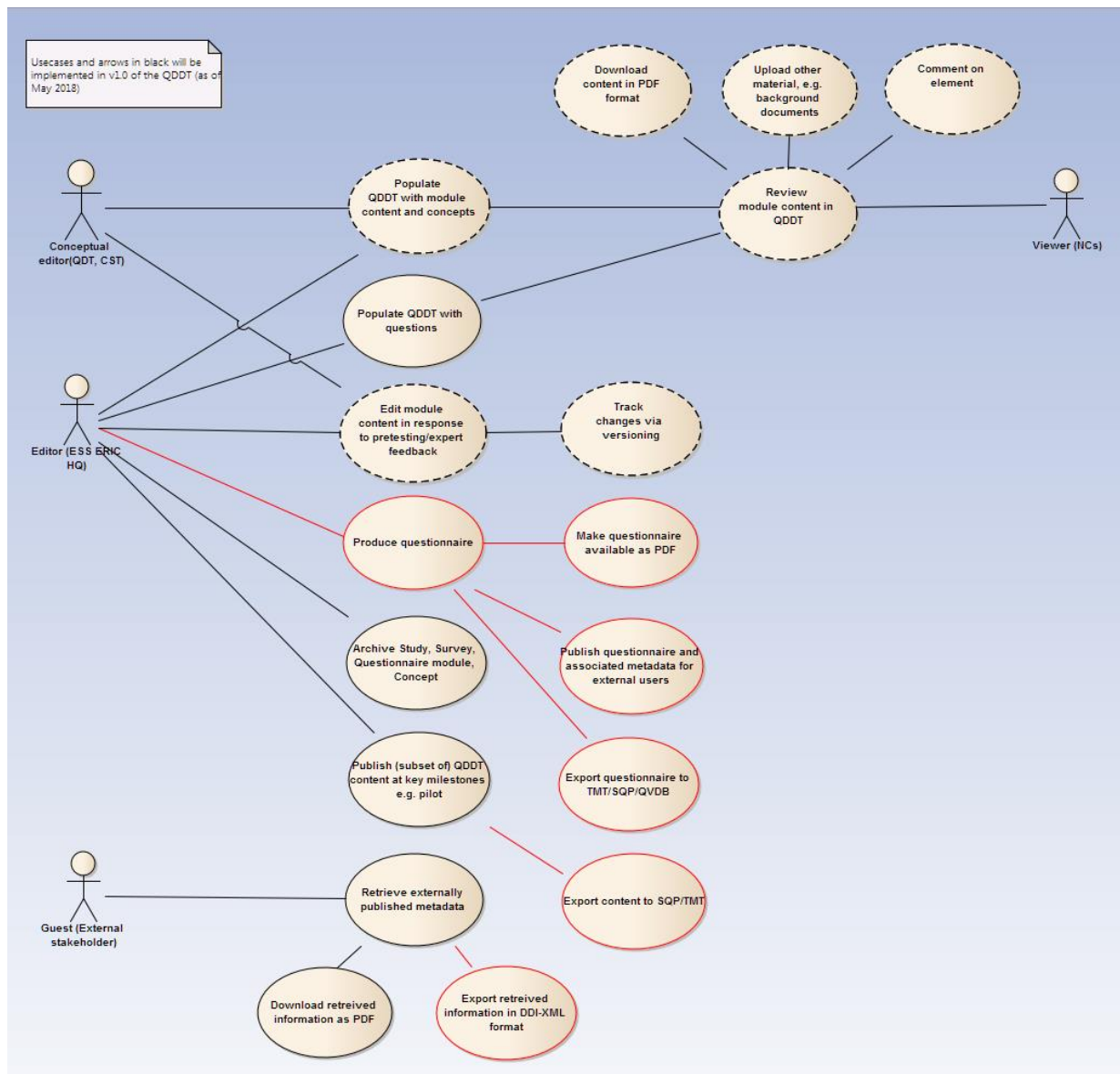
Child use cases are part of an abstract use case

In the diagrams below, use cases shown in black have been implemented in version 1 of the QDDT. Use cases shown in red remain to be implemented.

### 3.1 Overview

The diagram below provides an overview of the QDDT's functionality.

Figure 3.1 High level use case diagram



**Table 3.1 High level use cases, where to find details**

Use case name	Further details in this diagram	Comment
Populate QDDT with module content and concepts	3.2 Create new survey	
Populate QDDT with questions	3.4 Question item 3.5 Response domain 3.6 Categories	
Edit module content in response to pretesting/export feedback	3.3 Edit survey	
Track changes via versioning	3.9 Versioning	
Produce questionnaire	3.7 Question construct	Question construct is the question as in the instrument. The first step in the instrument design
Archive Study, Survey, Questionnaire module, Concept	3.10 Archiving	
Publish (subset of) QDDT content at key milestones e.g. pilot	3.8 Publication	
Publish (subset of) QDDT content at key milestones e.g. pilot	3.8 Publication	



## 3.2 Detailed use case diagrams

### 3.2.1 Survey, study, questionnaire module, concept

Figure 3.2 Use case diagram for “Create new survey”

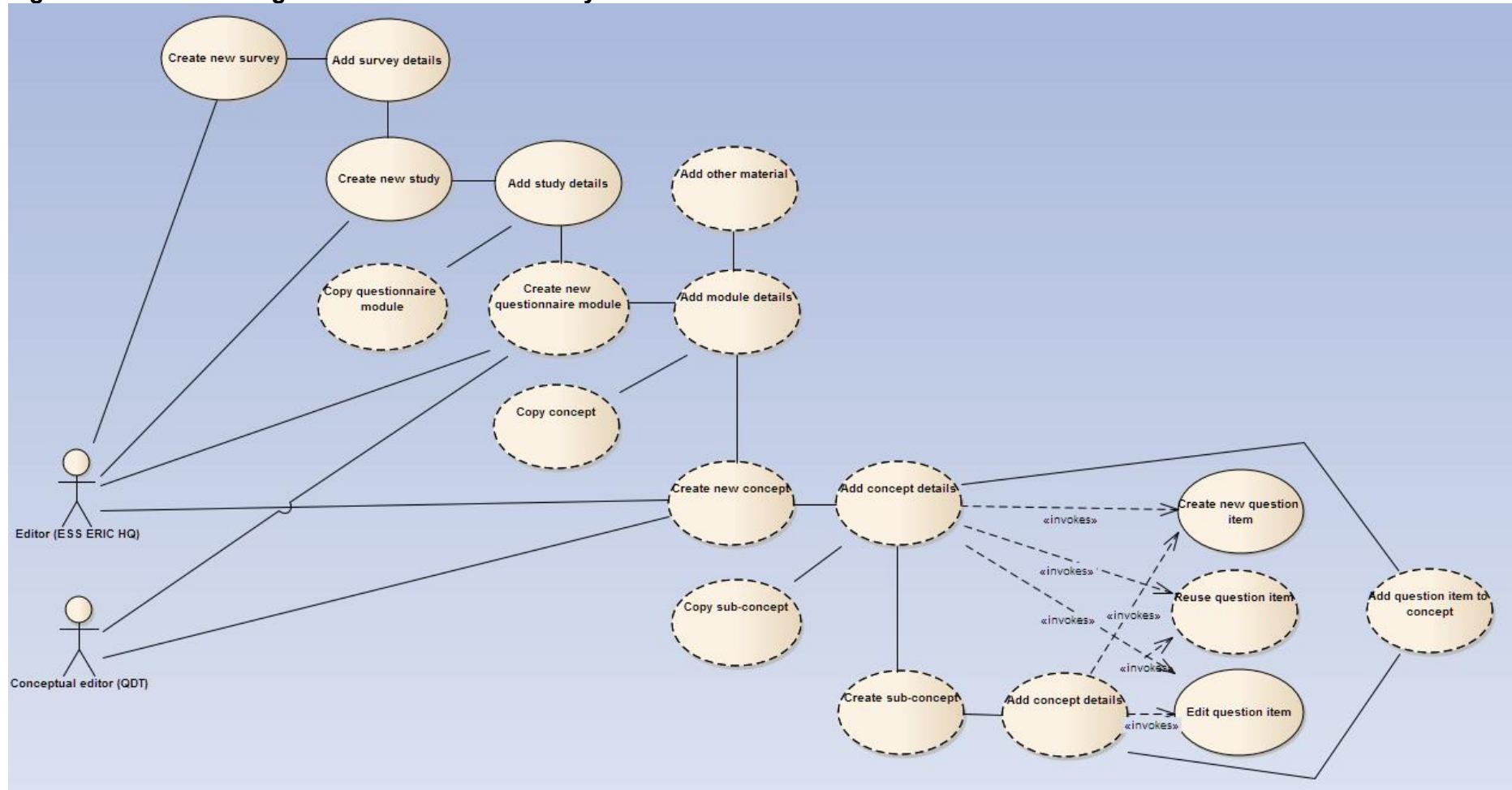
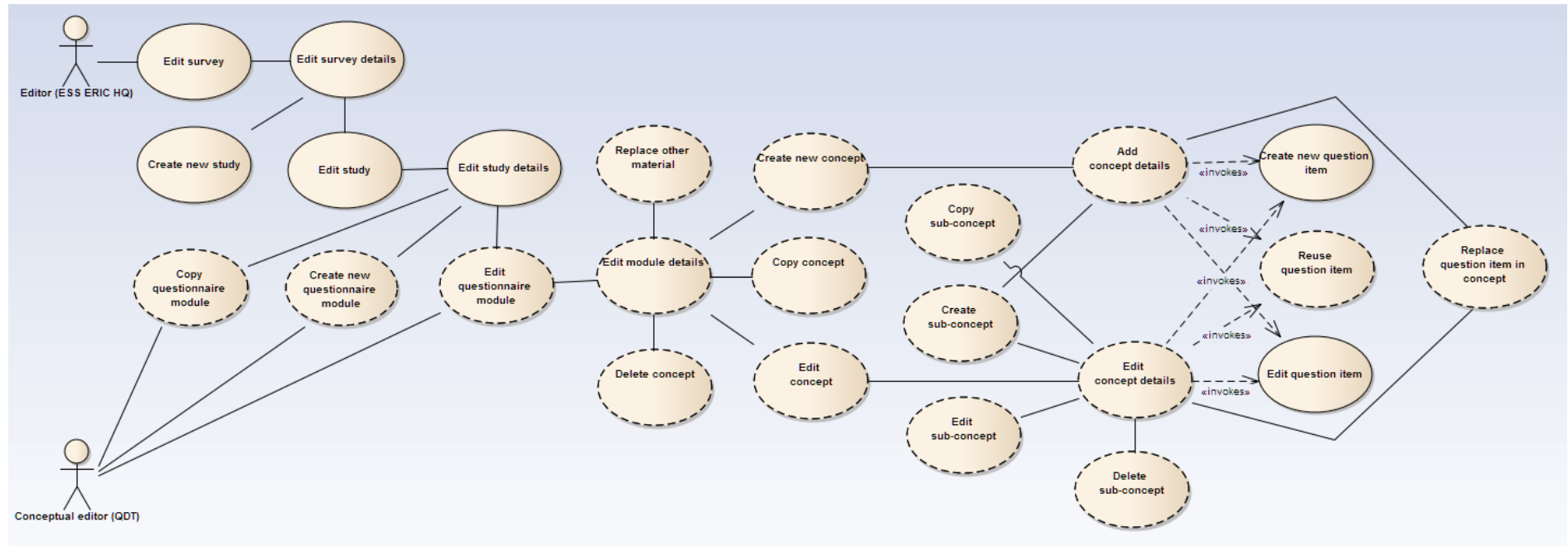


Figure 3.3 Use case diagram for “Edit survey”



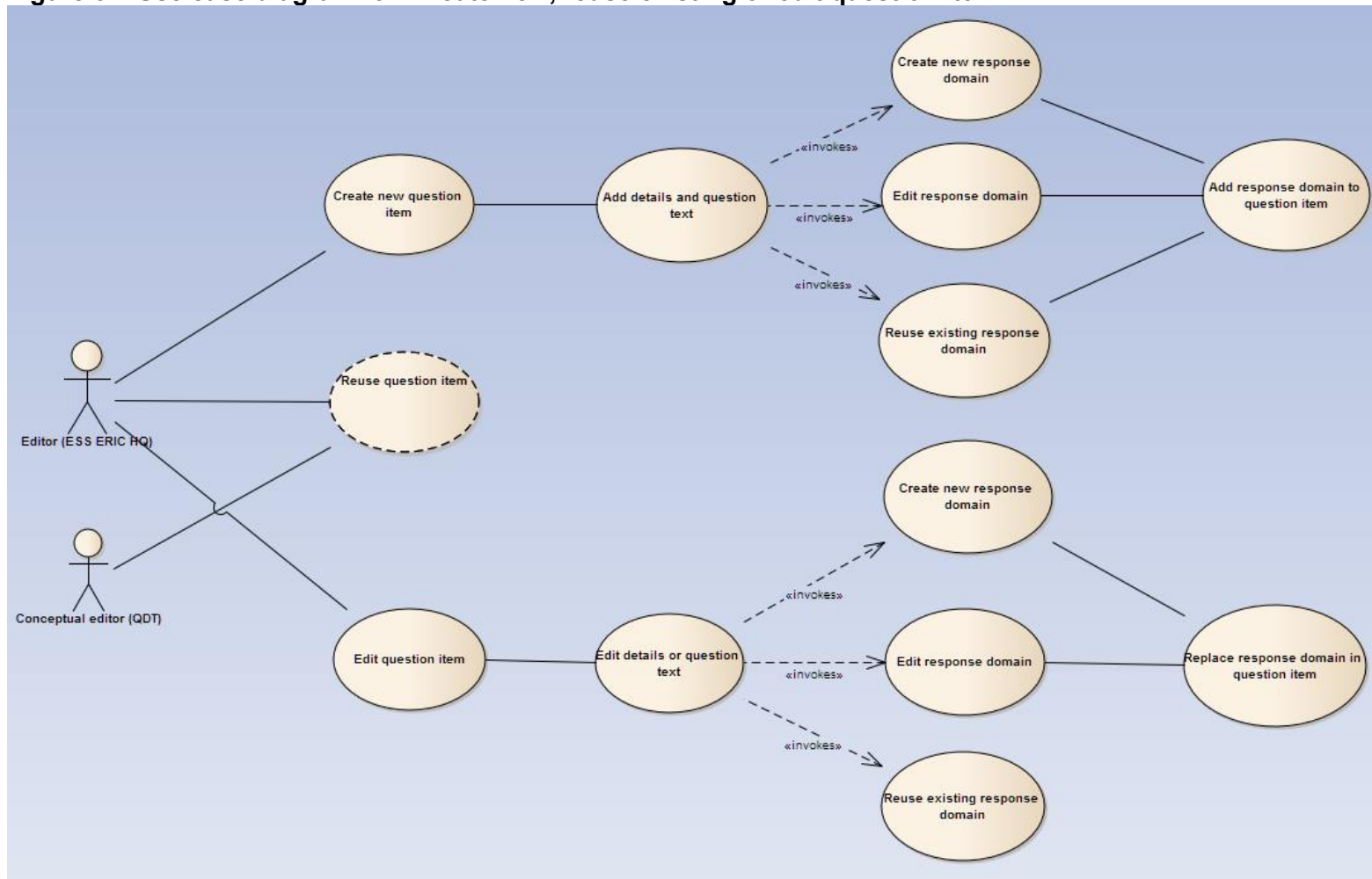
**Table 3.2 Survey, study, questionnaire module, concept diagram – use case name and ID number**

Use case name	ID Number	Further details in this diagram	Comment
Create new survey	1		
Add survey details	1.1		
Create new study	1.2		
Add study details	1.2.1		
Copy questionnaire module	1.3		
Create new questionnaire module	1.4		
Add module details	1.4.1		
Add other material	1.4.2		
Copy concept	1.5		
Create new concept	1.6		
Add concept details	1.6.1		
Copy sub-concept	1.6.2		
Create sub-concept	1.6.3		
Create new question item	9	3.4 Question item	
Reuse question item	10	3.4 Question item	
Edit question item	11	3.4 Question item	
Add question item to concept	1.6.4		
Edit survey	2		
Edit survey details	2.1		
Create new study	1.2	3.2 Create new survey	
Edit study	2.2		
Edit study details	2.2.1		
Copy questionnaire module	1.3	3.2 Create new survey	
Create new questionnaire module	1.4	3.2 Create new survey	
Edit questionnaire module	2.3		
Edit module details	2.3.1		
Replace other material	2.3.2		
Copy concept	1.5	3.2 Create new survey	
Create new concept	1.6	3.2 Create new survey	
Add concept details	1.6.1	3.2 Create new survey	
Edit concept	2.4		
Edit concept details	2.4.1		
Delete concept	2.5		
Copy sub-concept	1.6.2	3.2 Create new survey	

Create sub-concept	1.6.3	3.2 Create new survey	
Edit sub-concept	1.6.4	3.2 Create new survey	
Delete sub-concept	1.6.5	3.2 Create new survey	
Create new question item	9	3.4 Question item	
Reuse question item	10	3.4 Question item	
Edit question item	11	3.4 Question item	
Replace question item in concept	2.4.2		

### 3.2.2 Question items

Figure 3.4 Use case diagram for 'Create new, reuse existing or edit question item'

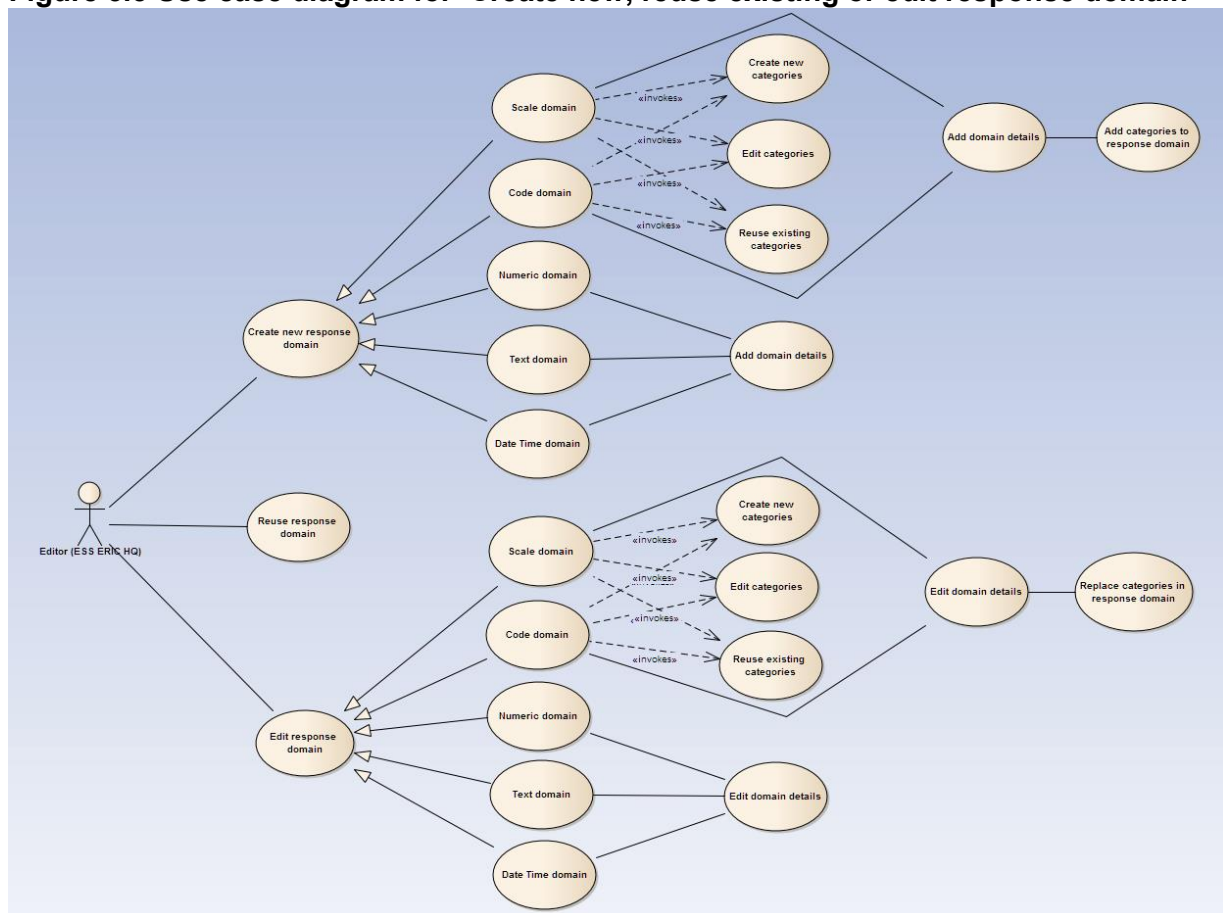


**Table 3.3 Question items diagram - use case names and ID numbers**

Usecase name	ID Number	Further details in this diagram	Comment
Create new question item	9		
Add details and question text	9.1		
Create new response domain	6	3.5 Response domain	
Reuse existing response domain	7	3.5 Response domain	
Edit response domain	8	3.5 Response domain	
Add response domain to question item	9.2		
Reuse question item	10		
Edit question item	11		
Edit details or question text	11.1		
Create new response domain	6	3.5 Response domain	
Reuse existing response domain	7	3.5 Response domain	
Edit response domain	8	3.5 Response domain	
Replace response domain in question item	11.2		

### 3.2.3 Response domains

Figure 3.5 Use case diagram for 'Create new, reuse existing or edit response domain'



**Table 3.4 Response domains diagram - use case names and ID number**

Usecase name	ID Number	Further details in this diagram	Comment
Create new response domain	6		
Scale domain	6a		
Code domain	6b		
Create new categories	3	3.6 Categories	Applies to 6a and 6b
Reuse existing categories	4	3.6 Categories	Applies to 6a and 6b
Edit categories	5	3.6 Categories	Applies to 6a and 6b
Numeric domain	6c		
Text domain	6d		
DateTime domain	6e		
Add domain details	6.1		
Add categories to response domain	6.2		Applies to 6a and 6b
Reuse response domain	7		
Edit response domain	8		
Scale domain	6a		
Code domain	6b		
Create new category	3	3.6 Categories	Applies to 6a and 6b
Reuse existing categories	4	3.6 Categories	Applies to 6a and 6b
Edit category	5	3.6 Categories	Applies to 6a and 6b
Numeric domain	6c		
Text domain	6d		
DateTime domain	6e		
Edit domain details	8.1		
Replace categories in response domain	8.2		Applies to 6a and 6b



### 3.2.4 Categories

Figure 3.6 Use case diagram for 'Create new, reuse existing or edit categories'

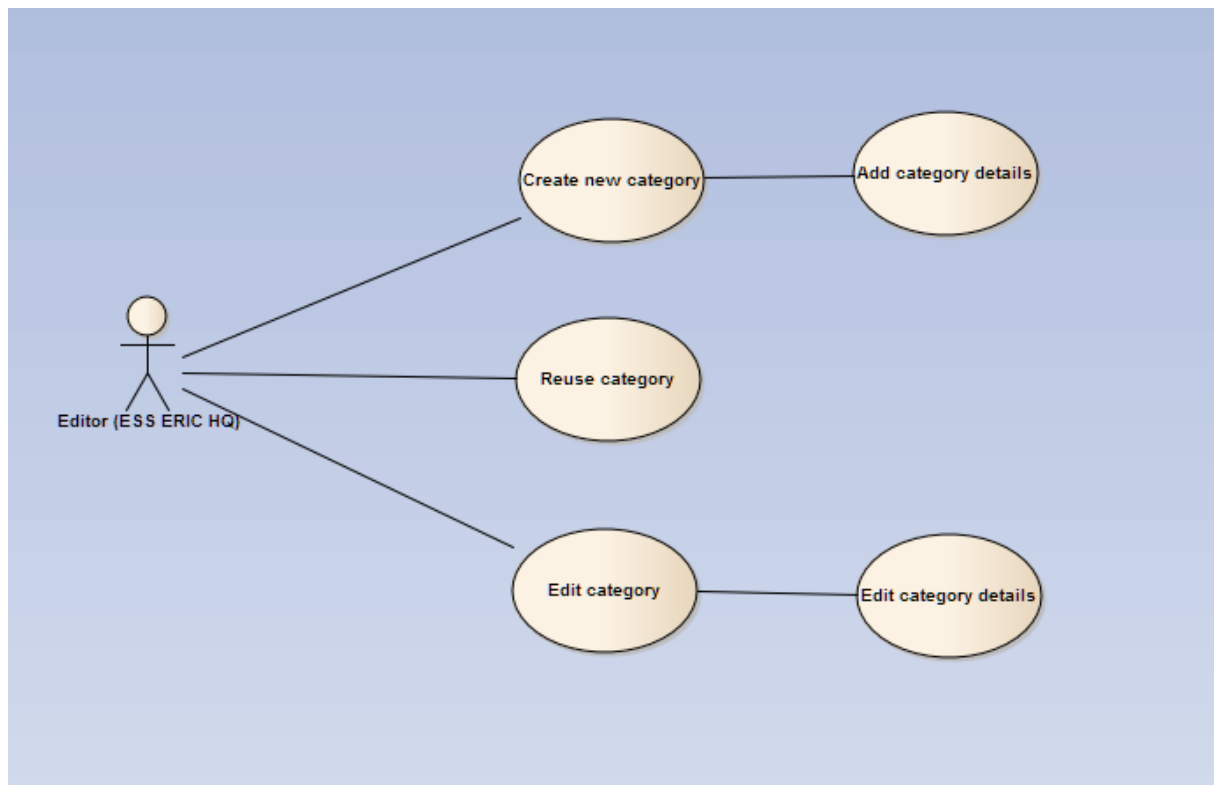


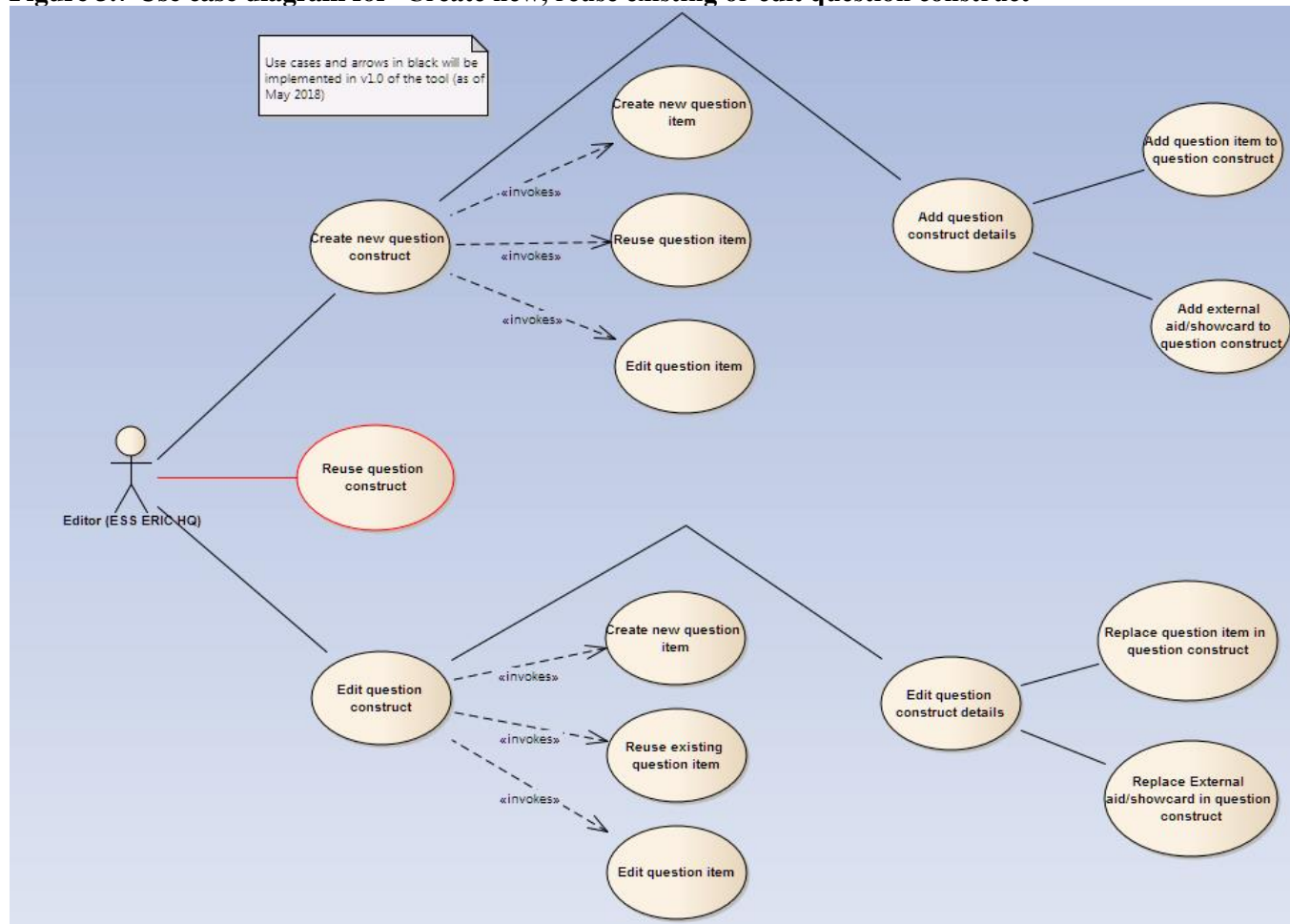
Table 3.5 Categories diagram - use case names and ID numbers

Use case name	ID Number	Further details in this diagram	Comment
Create new category	3		
Add category details	3.1		
Reuse category	4		
Edit category	5		
Edit category details	5.1		

### 3.2.5 Question constructs

Question construct is the question as in the instrument. It is the first step in producing a questionnaire.

**Figure 3.7 Use case diagram for ‘Create new, reuse existing or edit question construct’**

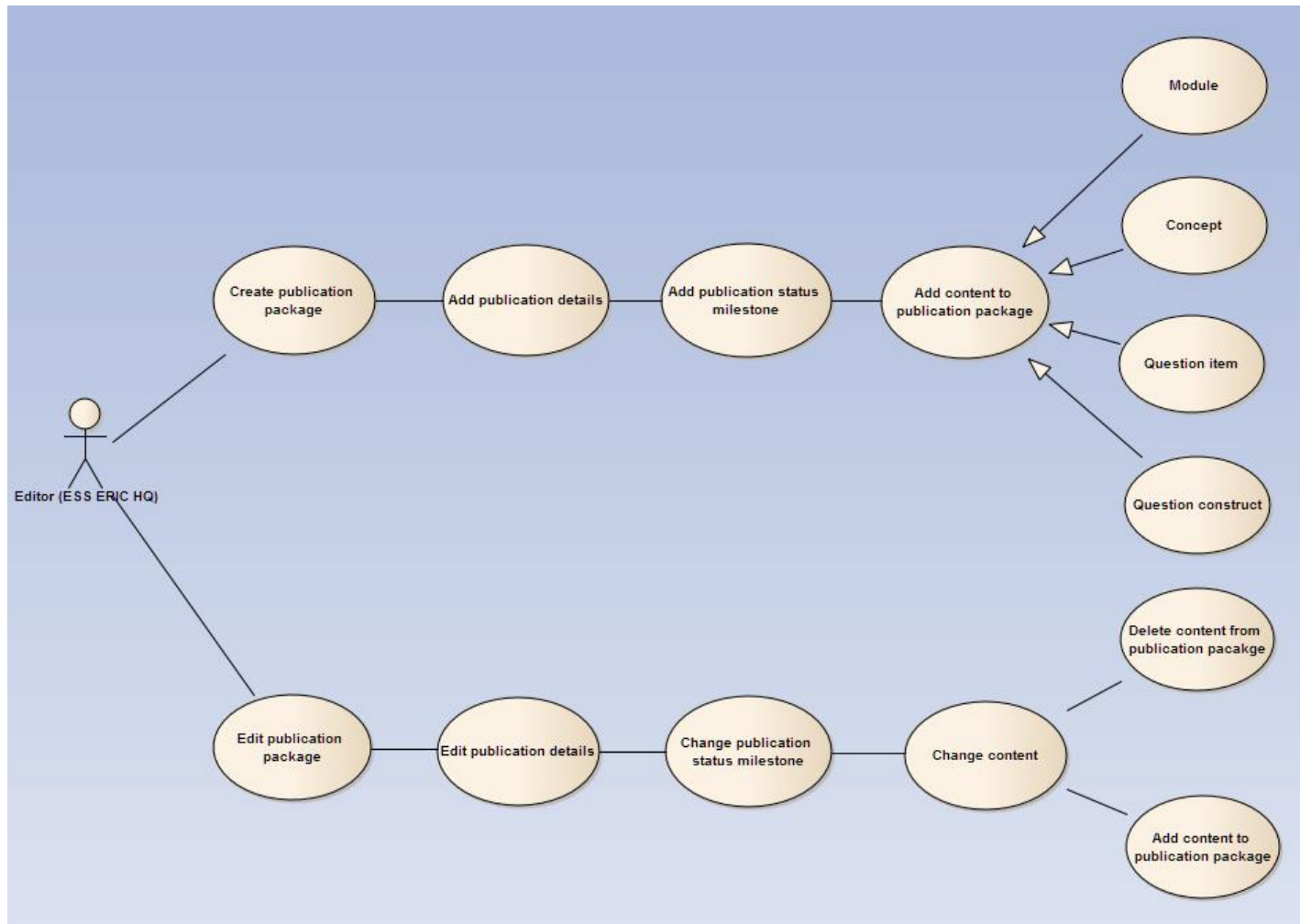


**Table 3.6 Question Construct diagram – use case names and ID numbers**

Usecase name	ID Number	Further details in this diagram	Comment
Create new question construct	12		
Create new question item	9	3.4 Question item	
Reuse question item	10	3.4 Question item	
Edit question item	11	3.4 Question item	
Add question construct details	12.1		
Add question item to question construct	12.2		
Add external aid/show card to Question Construct	12.3		
Reuse question construct	13		Functionality to be implemented in the tool, when Sequence and Instrument will be in place
Edit question construct	14		
Create new question item	9	3.4 Question item	
Reuse question item	10	3.4 Question item	
Edit question item	11	3.4 Question item	
Edit question construct details	14.1		
Replace question item in question construct	14.2		
Replace external aid/show card in question construct	14.3		

### 3.2.6 Publication

Figure 3.8 Use case diagram for 'Publication of questionnaire module, concept, question item and question construct'



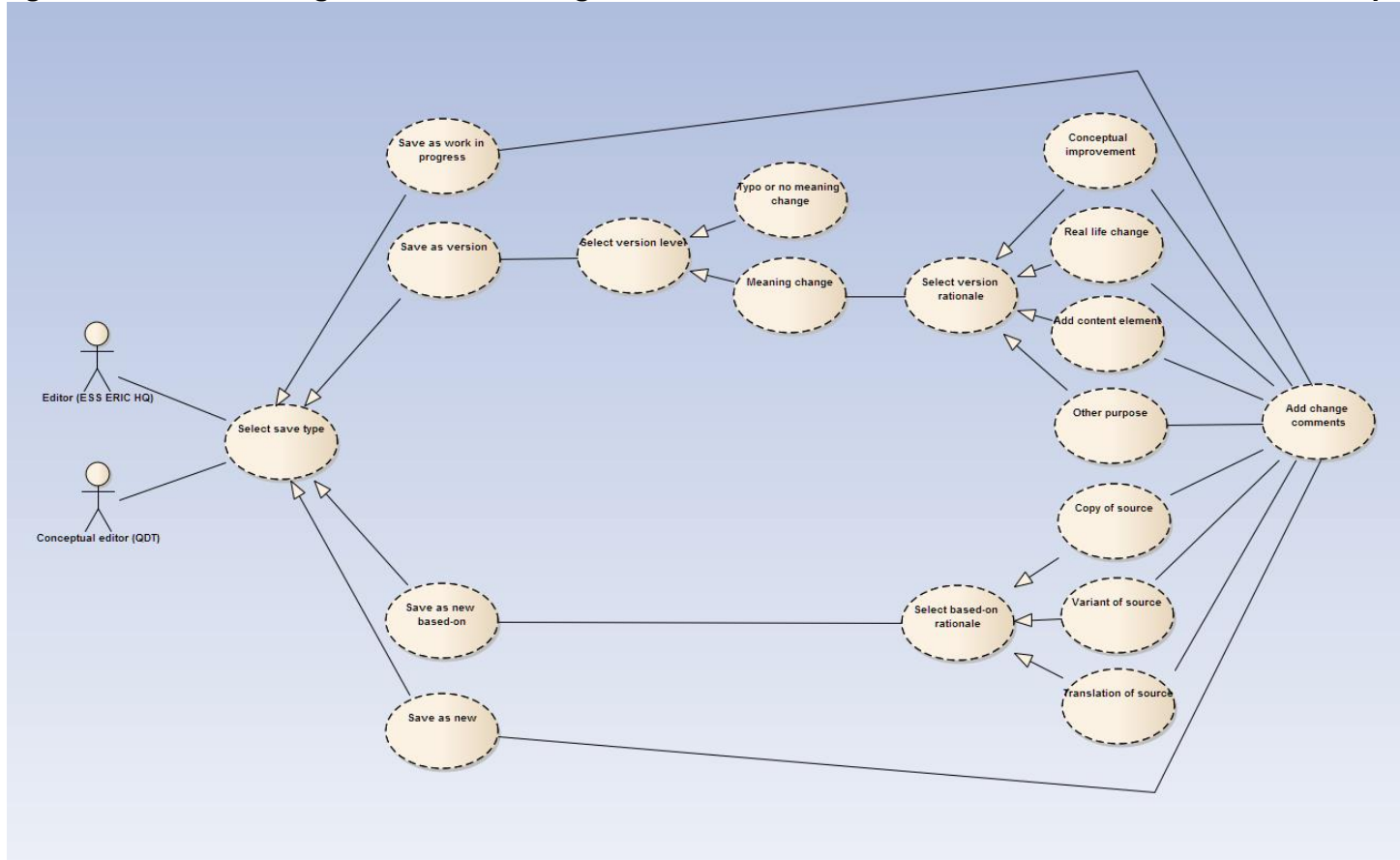
**Table 3.7 Publication diagram – use case names and ID numbers**

Usecase name	ID Number	Further details in this diagram	Comment
Create publication package	15		
Add publication details	15.1		
Add publication status milestone	15.2		
Add content to publication package	15.3		
Module	15.3a		
Concept	15.3b		
Question item	15.3c		
Question construct	15.3d		
Edit publication package	16		
Edit publication details	16.1		
Change publication status milestone	16.2		
Change content	16.3		
Delete content from publication package	16.4		
Add content to publication package	15.3		

### 3.2.7 Versioning

The full diagram applies to elements Category, ResponseDomain, QuestionItem and QuestionConstruct. For elements Survey, Study, Questionnaire module and Concept use cases 'save as new based-on' and 'save as new' and onward do not apply. Conceptual editors can only save Questionnaire modules and Concepts.

**Figure 3.9 Use case diagram for 'Versioning, save as new based-on, save as new and save as work in progress'**



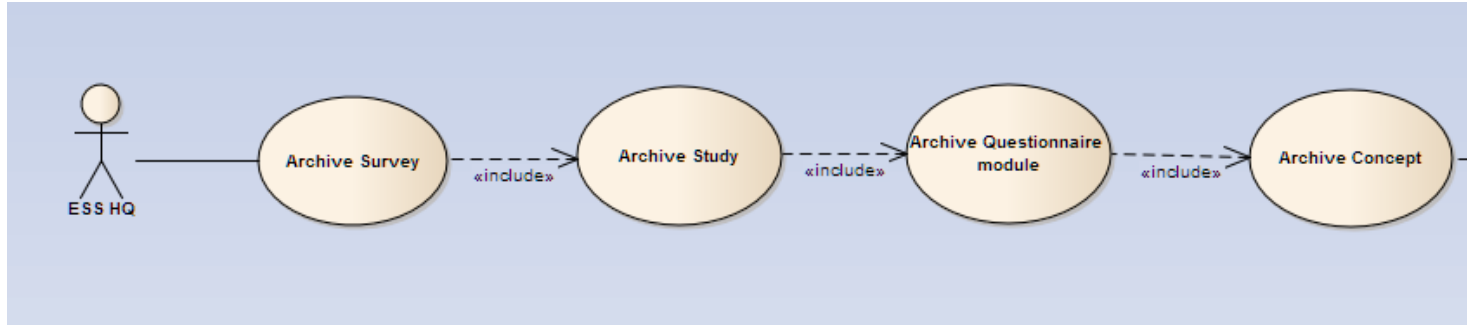
**Table 3.8 Versioning – use case names and ID numbers**

Usecase name	ID Number	Further details in this diagram	Comment
Select change type	17		Presupposes that changes are made to an element. Check comparison to find changes since last business version
Save as work in progress	17a		The save represents an 'in development' revision
Save as version	17b		Business versioning
Select version level	17.1		Applies to 17b
Typo or no meaning change	17.1a		
Meaning change	17.1b		
Select version rationale	17.1.1		Applies to 17.1b
Conceptual improvement	17.1.1a		
Real life change	17.1.1b		
Add content element	17.1.1c		
Other purpose	17.1.1d		
Save as new based-on	17c		The element is saved with a new id, with a based on reference to the element it
Select based-on rationale	17.2		Applies to 17c
Copy of source	17.2a		
Variant of source	17.2b		
Translation of source	17.2c		
Save as new	17d		Applies to categories, missings, response domains, question items, question constructs, and publication packages
Add change comment	17.3		

### 3.2.8 Archiving

**Figure 3.10 Use case diagram for ‘Archiving’**

Archiving prevents an element from being changed. Archiving is possible from different levels and applies to Survey, Study, Questionnaire module, Concept, Sub-concept. It would typically be done once all development work is complete and prior to publication of the study content for external users.



**Table 3.9 Archiving – use case names and ID numbers**

Usecase name	ID Number	Further details in this diagram	Comment
Archive Survey	18		
Archive Study	18.1		
Archive Questionnaire module	18.1.1		
Archive Concept	18.1.1.1		
Archive Sub-concept	18.1.1.1.1		



## 4. System Architecture and Design

### 4.1 System Architecture

The QDDT has a client server architecture. The user accesses the front end user interface (client) using a web browser to input information to and access information from the database stored on a server.

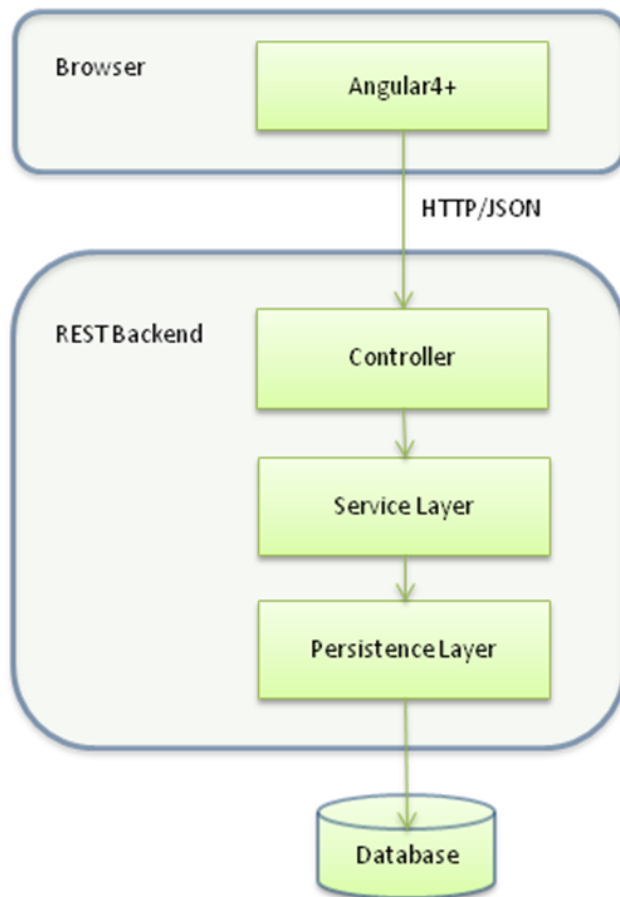
#### 4.1.1 System Hardware Architecture

No specific hardware is required. The QDDT will run on windows and/or linux machines.

#### 4.1.2 System Software Architecture

Angular 6 is used for the client and Spring for the server.

**Figure 4.1 Overview of system software architecture**

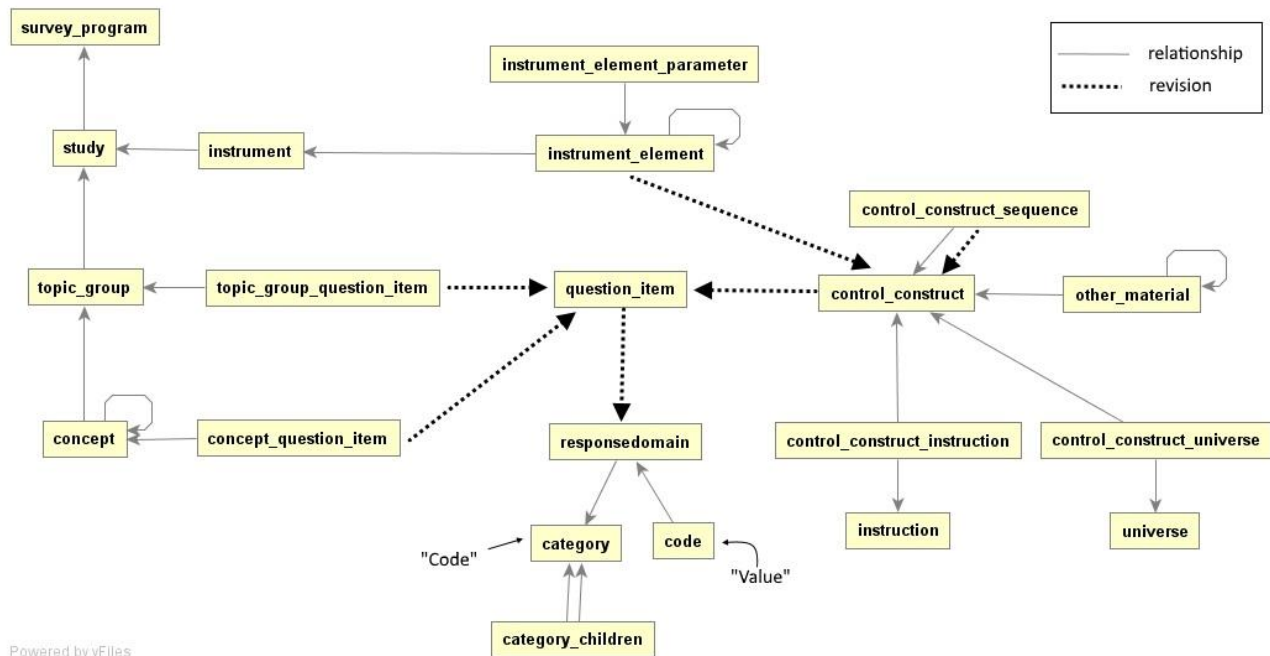


### 4.1.3 File and Database Design

The QDDT database design in compliance with DDI 3.2.

### 4.1.4 Database Management System Files

Figure 4.2 ER diagram of DB



[wiki/General-structure-of-QDDT-DB-model](http://wiki/General-structure-of-QDDT-DB-model)

There are 2 schemas in the QDDT, audit and public (default db name). Public is where the working model resides, Audit keeps all revisions of any entity in the QDDT, and is part of a third part library, thus implementation and diagrams are not shown.

The database will never be huge, in the range of 10s of megabytes, there are quite a few transactions going on pr user, but they are not expensive, and the load while testing is not measurable. The postgres server never exceed 3-4% load, and it has more than 10 db's running there. It is estimated that the application would need at least 100 concurrent user before performance is affected. Query time from the client → server → sql server and back takes between 10 and 50 ms.

### 4.1.5 Non-Database Management System Files

External files can be uploaded to the QDDT user interface. Uploaded files will be saved to a folder structure on the web server.

webapproot/uploads-to-qddt. This path is configurable (application.properties)

qddt.fileroot=/home/deploy/deployment/test/uploads-to-qddt/

## 4.2 Human Machine Interface

### 4.2.1 Inputs

Inputs to the system are made by the user via the web-based front end. Inputs are questionnaire elements as broadly defined by DDI 3.2.

Figure 4.3 QDDT user interface

The top screenshot shows the 'Survey' page in the QDDT interface. The left sidebar contains navigation links: Home, Questions, Categories, Missings, Domains, Constructs, and Publications. The main content area has a teal header with 'Survey' and a 'NEW' button. Below this, a card displays 'European Social Survey' with a version indicator 'Version 1.0' and a '0' in a box. The right sidebar shows 'Survey Toc' with links 'To the top' and 'European Social Survey'.

The bottom screenshot shows the 'QuestionItems' page. The left sidebar is the same. The main content area has a teal header with 'QuestionItems' and a 'NEW' button. Below this is a search bar with the placeholder 'Searches in [name and question and responsename]' and a close button 'X'. A table lists survey items with the following columns: Name, Question Text, ResponseDomain, Modified, Version, and Agency.

Name	Question Text	ResponseDomain	Modified	Version	Agency
djoinctx	Thinking about the amount of income tax you paid in the last 12 months, would yo...	Mixed [Just11 + DKREF]	Tue Jul 31 2018	2.0	int.esseric
djoincjcc	In case you feel your income from social benefit payments s is unjust, what would ...	Mixed [Income + DKREF]	Tue Jul 31 2018	1.0	int.esseric
djoincjcb	In case you feel your income from pensions payments before taxes is unjust, what ...	Mixed [Income + DKREF]	Tue Jul 31 2018	1.0	int.esseric
djoincjcaa	In case you feel your income from work before taxes is unjust, what would be a jus...	Mixed [Income + DKREF]	Tue Jul 31 2018	1.0	int.esseric
djoincocc	In [country], how just are the pension payments of those who have the same emplo...	Mixed [Just11 + DKREF]	Tue Jul 31 2018	1.0	int.esseric
djoincoca	In [country], how just is the gross income of those who have the same occupation ...	Mixed [Just11 + DKREF]	Tue Jul 31 2018	1.0	int.esseric

### 4.2.2 Outputs

Outputs from the system are PDF exports and a ddi xml export.

## 4.3 Detailed Design

### 4.3.1 Hardware Detailed Design

The database server on which the QDDT is installed at NSD is a virtual linux server with 1 cpu ( 6000 bogomips ) and 8 GB ram. (It has about 20 databases running and no load to speak of)  
The webserver is a virtual linux server with 2 cpus (10400 bogomips ) and 8 GB ram.  
Both servers could take at least a 100 fold increase in load, before more ram or faster disks would be required.

The client user interface can be accessed on any device capable of running a modern web-browser.

### 4.3.2 Software Detailed Design

The client (web browser) uses a single page web app created with the Angular framework. This minimizes the workload of the web server, which is only used when a user first enters the site.

Typescript is used as the language together with Angular; they work very well together. This is compiled to javascript code that is run as a website on web server of the customers choice.

The backend (server) is made with the Spring framework, with Java as the language. It is compiled into a Jar file that runs as a service/job/process on a server available from the web. Java runtime (minimum version 1.8.0\_151) is needed on the server

The QDDT is configured to use Postgres as database server (DB), this is configurable, and any available database should work as a backend DB server. This server must be accessible from the backend server.

It is possible to configure the backend including the DB server into a docker image, which would improve security and portability of the QDDT

The frontend is implemented in Angular 6 with Typescript, and has [the following dependencies](#) at the time of publishing v1.0

[Backend](#) is implemented with the [Spring framework](#).

The system is modelled around DDI 3.2, which means there will be nearly a 1:1 correspondence of [DDI](#) and system classes. There are 2 deviations; Category and the Instrument hierarchy. More information on how these elements are programmed can be found here: [wiki/Dev---Responseudomains](#), *the instrument hierarchy is not part of this deliverable.*

A conceptual description of each class is available in the source [code](#).

## 4.4 External Interfaces

The QDDT exports to PDF and as a ddi xml export.

To communicate with the system it is necessary to send and receive html requests with json.

The QDDT does not currently interact with any external interfaces. However, there are plans to ensure connectivity between the QDDT and other tools developed under the SERISS project including the QVDB (see Section 2.1 above).

## 4.5 System Integrity Controls

The system uses Spring security to validate and restrict access to the system. All events that give rise to exceptions are logged in log files on server. Users are logged in and access to the different parts of the system is validated against user access rights. Since the system is fully revision based, every change to the system is stored with information on who made the change and when.

## 5. Software Testing

The QDDT has undergone two rounds of Beta testing by researchers at ESS ERIC HQ, one in June-July 2017 and a second in June-July 2018. On both occasions researchers populated the QDDT with content from the questionnaire development that took place in previous rounds of the European Social Survey.

Both round of Beta testing were retrospective and involved a single user populating the tool with previously agreed content. To fully test the tool's potential requires for it to be used to document questionnaire discussions taking place between multiple stakeholders in 'real time'. The ESS intends to use the QDDT to document the development of the ESS Round 10 questionnaire from the start of the development process.

Further details of the beta testing and resulting outputs can be found in:

Butt, S et al (2018). *Report on the use of QDDT in documenting the development of ESS Round 9* Deliverable 4.1 of the SERISS project funded under the *European Union's Horizon 2020 research and innovation programme* GA No: 654221. Available at: [www.seriss.eu/resources/deliverables](http://www.seriss.eu/resources/deliverables)

## 6. Version Description

The current version of the QDDT, released in May 2018, can be accessed here: <https://github.com/DASISH/qddt-client/wiki>

There are two versions of the QDDT available - an "in development" version which is a work in progress and a "production" version which is the latest stable version of the tool in use by the European Social Survey.

### 6.1 Installation Instructions

*QDDT client is the front end of [QDDT](#) (Questionnaire Design and Documentation Tool). The QDDT client is implemented with [angular6](#). The QDDT API is a bridge between QDDT client and QDDT backend. The backend is a spring-based Java server.*

For instructions on how to install please see: <https://github.com/DASISH/qddt-client/wiki/Dev---Getting-started>

## **6.2 Possible Problems and Known Errors**

A log of any known issues with the QDDT can be viewed here:

<https://github.com/DASISH/qddt-client/issues>