

## Document Information

Category	Information
Project	CQF-QMT
Document	Requirements Specification
Document Version	1.0
Status	In Development
Author(s)	James Wilson, Paul Dudley, Basil Khan
Approver(s)	Linda Steedman
Issue Date	October, 2008
Document Location	
Distribution	Restricted

## Definition of Terms, Acronyms and Abbreviations

This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.

Term	Description
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformations
HTML	Hypertext Markup Language
DTD	Document Type Definition
QTI	Question And Test Interoperability

## Table of Contents

<b>1.</b>	<b>Introduction</b>	<b>3</b>
1.1	Aims and Objectives	3
<b>2.</b>	<b>Technologies</b>	<b>4</b>
2.1	XML	4
2.2	XSD and DTD	4
2.3	PHP and Related Technologies	4
2.4	PHP with XML and XSLT	4
2.5.	Question and Test Interoperability Specification	4
<b>3.</b>	<b>System Requirement Specification</b>	<b>6</b>
3.1	System Overview	6
3.2	User Requirements	6
3.3	Use Case Diagrams	6
3.4	Design Goals	7
3.5	Dependencies	8
3.6	Activity Diagrams	8
3.7	Interface Design	9
<b>4.</b>	<b>Appendices</b>	<b>10</b>
	Appendix A – Class Diagram	<b>Error!</b>
	<b>Bookmark not defined.</b>	

## Introduction

### Aims and Objectives

Initially the project's primary aim was to specify a system which supports the calculation of a SCQF level when supplied with a CQF/SCQF-compliant repository of questions which make use of Bloom's Taxonomy (BT).

However, initial research and prototyping identified significant challenges in developing a software solution to achieve this objective (see the Final Report for further information).

This original objective was modified to specify a system which supports the identification of the Blooms domain and level, along with the SCQF generic outcome, associated with assessments when supplied with a CQF/SCQF-compliant repository of questions which make use of Bloom's Taxonomy (BT).

To achieve the above mentioned aim, the following objectives should be fulfilled.

- Provide a tool to parse and analyze questions from repositories.
- Build a repository of mappings that can help determine and achieve aim of the project.
- Provide the tool and its outputs in open source format so it could be used easily in future extensions.
- Evaluation and testing of the project with initial requirements.

## Technologies

### XML

The usage of XML is becoming prevalent in data-driven processes. Systems including sales reporting applications, purchasing applications, inventory control systems and content management systems are based on XML for data integration and information exchange between various modules.

### XSD and DTD

Both Document Type Definition (DTD) and XML Schemas (XSD) are common methods for defining XML data models.

DTD is the older of the two methods used in XML data modeling. It defines the elements that may be included in your document, types of attributes these elements can have and the indication of parent and child relationships between these elements. A DTD may also be defined inside an XML document in which case it may be called an inline DTD.

XML Schema documents are XML documents which refer to a XML Schema Namespace and can have their own DTD. The main advantage of using a XSD over a DTD is that they provide an object oriented approach towards defining the format of a XML document. They include most basic programming types such as string, integer etc. The creator of the XML Schema document can then use these core types to create complex types for defining an element and its attributes.

### PHP and Related Technologies

PHP (PHP: Hypertext Preprocessor) is a scripting language developed by Zend. PHP is an open source technology which means the API provided can be used without having to acquire any licenses. Additionally, PHP has been deployed and tested extensively on both Windows and Linux platforms. The main reasons, therefore, for choosing PHP for this project are its platform independent code, open source license, and the wealth of XML API support and resources available.

Using PHP implies that the application will use a client-server architecture. An application server such as Apache or Microsoft IIS will provide the service on behalf of another piece of software, i.e. the client. The server's main responsibility is to respond to requests; in this case as it will be a web server the protocol for communication that will be used is Hyper Text Transfer Protocol (HTTP).

### PHP with XML and XSLT

PHP provides a dedicated API to parse information in XML documents. PHP also provides support for using XSLT functions through the Sablotron library.

PHP is capable of parsing an XML document by using a built-in API or with DOM object model support. PHP allows the programmer to create an XML parser by calling the function `xml_create_parser()` which returns a parser reference. This reference can be used to set various options, element handlers and character data handler, as well as specifying any callback functions required. Callback functions are functions which are called when an event occurs in the system. Two callback functions are specified in element handler, one is triggered when a start tag is found by the parser and the second when an end tag is found by the parser. The character data handler allows the programmer to identify a callback function which will handle data between a start and end tag.

### Question and Test Interoperability Specification

The QTI specification was developed by the IMS global consortium. The specification defines a standard format for the representation of assessment content and results. The purpose of the specification is to facilitate interoperability between systems sharing assessment content, i.e. authoring tools, item banks, test construction tools, learning systems and assessment delivery systems.

The following are the main advantages of QTI:

- It provides a well documented content format for storing and exchanging items independent of the authoring tool used to create them.
- It supports the deployment of item banks across a wide range of learning and assessment delivery systems.
- It provides a well documented content format for storing and exchanging tests independent of the test construction tool used to create them.
- It supports the deployment of items, item banks, and tests from diverse sources in a single learning or assessment delivery system.
- It provides systems with the ability to report test results in a consistent manner.

# System Requirement Specification

## System Overview

CQF-QMT is a system built for the purpose of determining the Blooms domain and level, along with the SCQF generic outcome, associated with assessments. The assessments are to be presented to the system in the form of a valid QTI document. The system parses all available QTI questions and attempts to extract the question stem from them. If successful, a NLP analysis is carried out on the text extracted from the question to identify important verbs. At this stage an attempt is made to achieve a mapping of Blooms Taxonomy with these verbs which could in future versions of the software help to determine the SCQF level.

If the identification of the Blooms domain and level, along with the SCQF generic outcome, is achieved for an assessment, metadata elements are generated. These elements are then attached to the question element in the input XML document and is presented to the user to be downloaded or viewed. If a level is not achieved, the original question node is returned intact.

## User Requirements

The following is a brief list of some of the key user requirements.

- The system should be generic and able to handle most types of QTI documents.
- The system should be able to load QTI assessment items for mapping process.
- The system should be able to map assessment item text against Bloom's Taxonomy.
- The system should generate metadata with appropriate information attached to the assessment item.

## Use Cases Diagrams

Use Case Diagrams are used in software engineering to capture the functional requirements of the system. They identify interaction between actors and the system. Actors may be end users, the system itself, or other systems interacting with the system to achieve some goal.

The following are the Use Cases for the proposed CQF-QMT System.

- Load Question  
The Load Question use case takes a question file from the user and uploads it to the server. The file then is passed to Parse Question case.
- Parse Question  
The Parse Question use case treats the uploaded document as a QTI question and tries to parse it to extract question text.
- Generate Keywords  
The Generate Keywords use case gets the text from Parse Question use case and tries to apply NLP tools to identify verbs. These verbs are treated as keywords in the next process.
- Identify Mappings  
This use case attempts to map keywords generated to identify the correct the Blooms domain and level, along with the SCQF generic outcome, for the question.
- Generate Metadata  
Metadata tags are created with the appropriate information and attached to the QTI document uploaded in the initial stage. The new QTI document with the metadata tags is then available to download for the user.

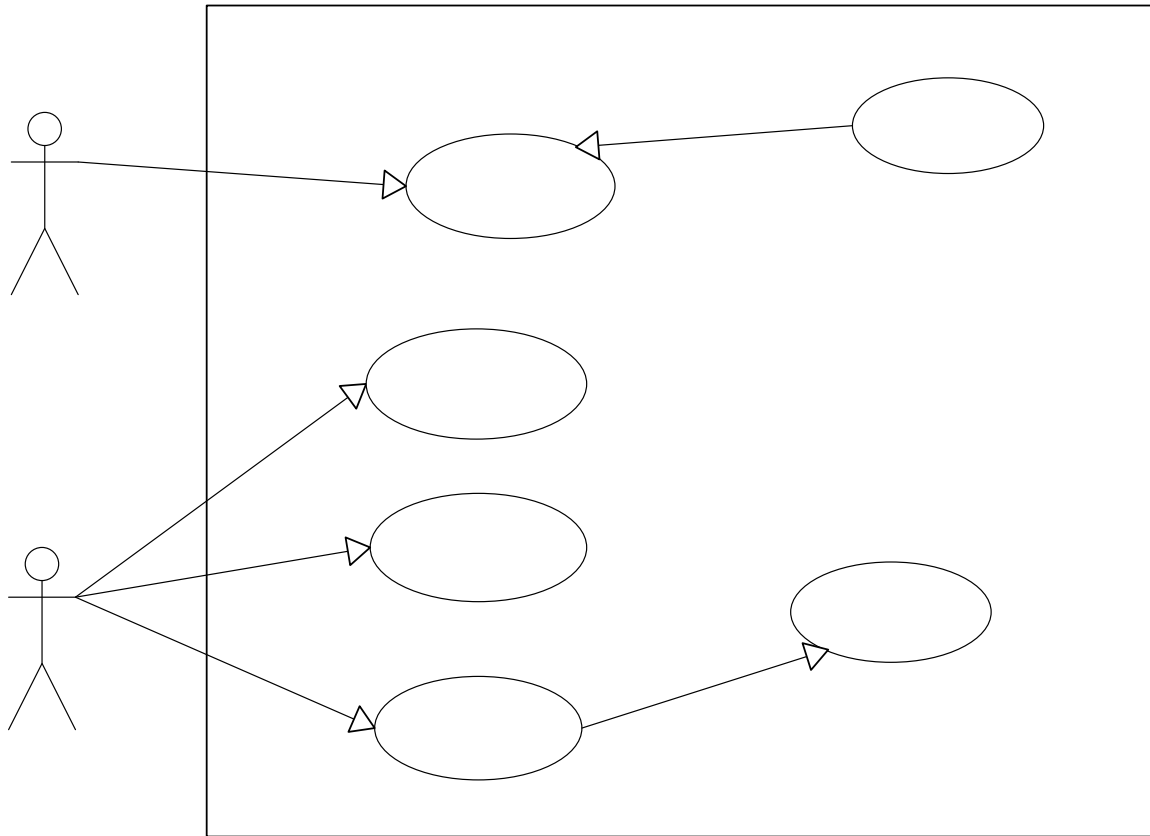


Figure 5.1.1 - Use Case Diagram.

Top Package::User

Design Goals

Reliability

Reliability is an essential aspect of the system because of its potential usage as a component in a larger system. In particular, it could be adapted to form a publically-accessible web service or even as part of a LMS (Learning Management System). In accordance with this, the system has been developed to provide the desired functionality while performing in a predictable and robust manner. Test cases have been conducted using sample QTI documents provided by IMS.

Maintainability

Commenting and documentation of classes and methods has been carried out in order to deliver a manageable final system.

Scalability

The system must be scalable in terms of supporting almost all types of question types. The system has been designed initially with QTI in mind, but it is also modular and scalable enough to introduce a new type of question schema should the need arise in the future.

Generating Keyw

Identifying Mappings

## Performance

Performance and response times are primary concerns and a deliberate and focused attempt has been made to ensure the consistent efficiency of the produced code. In particular, the system must be able to withstand and efficiently process large amounts of input data such as an entire exam or course.

## Dependencies

As the system has been implemented as a web application, it may only be accessed through a compatible browser. The system has been implemented in accordance with the three-tier client-server architecture in which user interface, functional or business logic and data storage are developed on three different levels and maintained as independent components. Therefore, a server computer will be required where the software will be located and the major processing will take place. Multiple clients can then access the system simultaneously from multiple client machines.

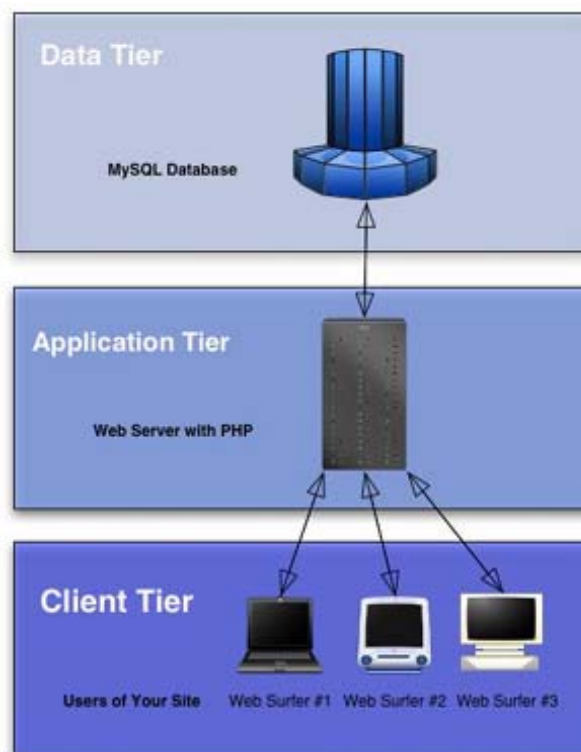


Figure 3.5.1 - The Three-Tier Architecture.

## Activity Diagrams

The main activities of the CQF-QMT system can be shown with the help of Activity Diagrams. Activity Diagrams represent the operational step-by-step workflow of each component in a system. Each activity in activity diagrams is represented by a state. States are connected using arrows, with the direction of the arrow indicating the flow of the activity.

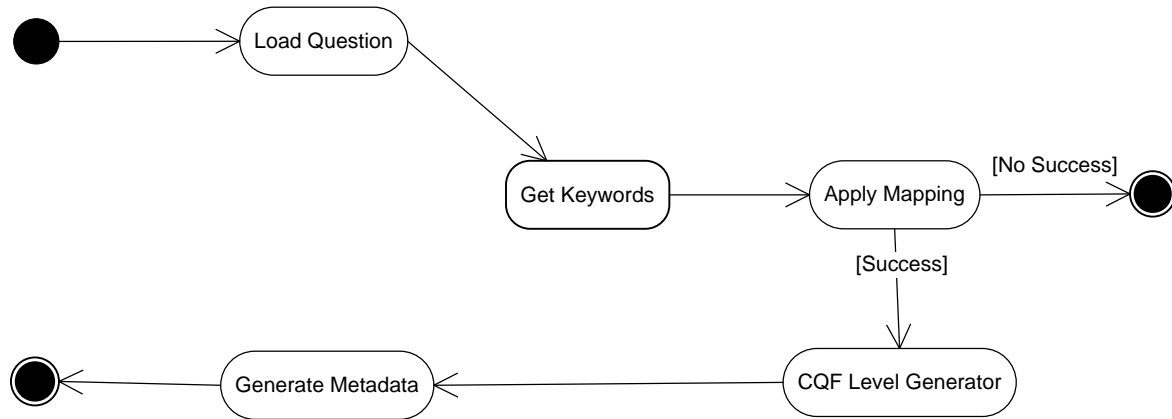


Figure 2.6.1 - Activity Diagram for CQF-QMT.

### Interface Design

The primary aim of the interface is to facilitate the straightforward interaction between the system and the user. Careful consideration of the aesthetic aspect of the interface was carried out as a low priority.

The first interface that is loaded when initially using the system is the CQF-QMT main page (figure 3.7.1 below). From here the user may upload a valid QTI XML document containing one or more questions. The file is transferred from the local computer of the user to the server file system for processing. The system will then attempt to parse the document and will notify the user of any identified questions and verbs.

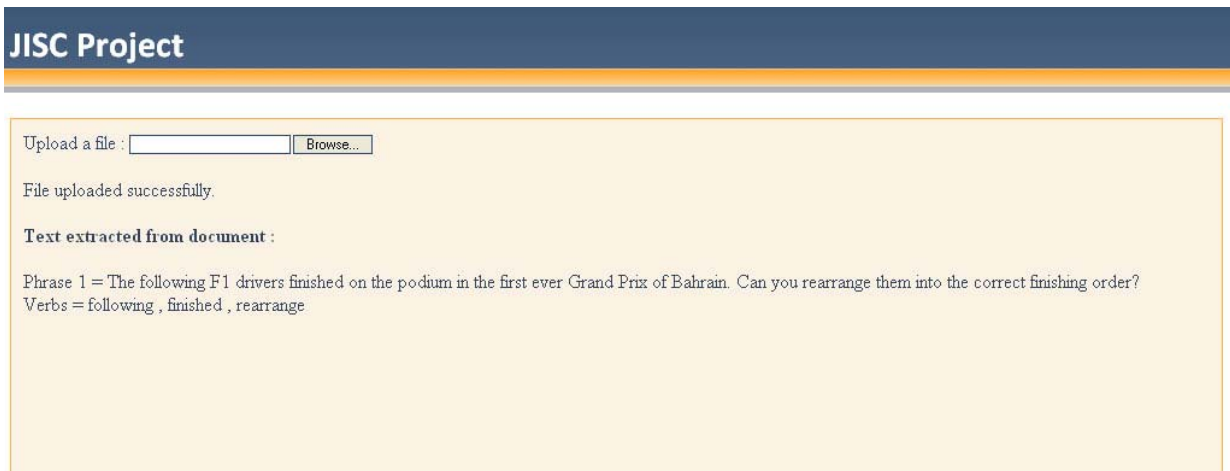


Figure 3.7.1 - The CQF-CMT interface.

## Appendices

### Appendix A – Class Diagram

