QGS - Question Generation System from Course Materials

Software Requirements Specification

Project ID: 17-006

Name of the author:

IT14033506 – M.F.F Mohamed

Name of the Supervisor:

Ms. Anjalie Gamage

BSc. Special (Honors) Degree in Information Technology

Date of submission: 02-05-2017

Project Title: Question Generation System from course materials.

Project ID: 17-006

Author Name: A.S.M Nibras

Registration No: IT14058424

Supervisor: ----------
(Miss Anjalie Gamage)

DECLARATION

I hereby certify that the project work entitled “QGS” – Question Generation System submitted to the Sri Lanka Institute of Information Technology is a record of the original work done by me under the guidance of our supervisors Miss Anjalie Gamage and the project work is submitted in the partial fulfillment of the requirement for the award of the degree of BSc. (Honors) degree of Information Technology. The results embodied in this document was not a copy of any document created by any organization, university, any other institute of a previous student project group at SLIIT and was not copied from the internet or any other source.

Name: A.S.M Nibras

Reg.No: IT14058424

Signature: _____

Table of Contents

List of Figures

List of Tables

1 Introduction

This section gives a scope description and overview of this Software Specification Requirement document (SRS) document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1 Purpose

Through this document can be defined as an official statement required by the system developers. This document provides the overall idea and the high-level design of the suggested system.

Software Requirements Specification defines and describes the operations, interfaces, performance, and quality assurance requirements of the Question Generation System. The document also describes the non-functional requirements such as the user interfaces. It also describes the design constraints that are to be considered when the system is to be designed, and other factors necessary to provide a complete and comprehensive description of the requirements for the software. The Software Requirements Specification captures the complete software requirements for the system, or a portion of the system.

In addition, this document acts as a contract between the project team and SLIIT examiners (since there is No client for this research project) to obtain comprehensive description of the requirements for the System being developed.

1.2 Scope

Purpose of the SRS document is to give complete description of the proposed application "QGS" which is a desktop application that can generate suitable questions from text book contents. This document covers the functional requirements, Non-functional requirements, performance requirements, technical requirements, system constraints and research component regarding information extraction, natural language understanding and processing.

Main targeted audience of this document is research team, developing team, supervisors, also be of interest to software engineers building or maintaining the software.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
SRS	Software Requirement Specification
POS Tagging	Part-of-Speech tagging
QGS	Application Name
NLP	Natural Language Processing
IE	Information Extraction
NLTK	Natural Language Tool Kit
DB	Database
Ontology	An ontology is a formal naming and definition of the types, properties, and interrelationships of the entities that really or fundamentally exist for a particular domain of discourse.

Table 1: Definitions, acronyms and abbreviations

1.4 Overview

QGS helps user to preparing questions from text materials. For the explanation purpose, I will take school text book as example. User this example is teacher. Teacher will upload the text materials to the system. The he/she has to specify the chapters separately. System will process with text material and generate all possible questions out of it. These questions will be displayed to the user, then teacher can select the suitable questions and prepare the question paper. Using QGS user has a great opportunity to prepare question paper automatically without any difficulties.

This SRS document include three sections and references & appendices. This chapter is mention the system contains. Further this a chapter introduces the different types of stakeholders and their interaction with the system.

Second section of this document provide an overview of the system under product perspective and user perspective. Mainly in product perspective gives the comparison between

existing products and the QGS (proposed system). Also, this section describes the system interfaces, user interfaces, hardware interfaces, software interfaces, communication interfaces. Also describes memory constraints, system operation and site adaptations requirements are mentioned under this section. Under user perspective this document describes use case diagrams and use case scenarios. Finally, in this section describes user and their knowledge to operate the QGS, system constrains, assumption and the other apportioning of requirements.

Third section of this document describes requirements for the development team. It describes external interface requirements, classes or objects, performance requirements, design constraints, software system attributes (non-functional attributes) and other requirements. Finally, in this document contains the references and appendices.

2 Overall Description

A graphical description of the processes involve in QGS is detailed in Figure 1 below.

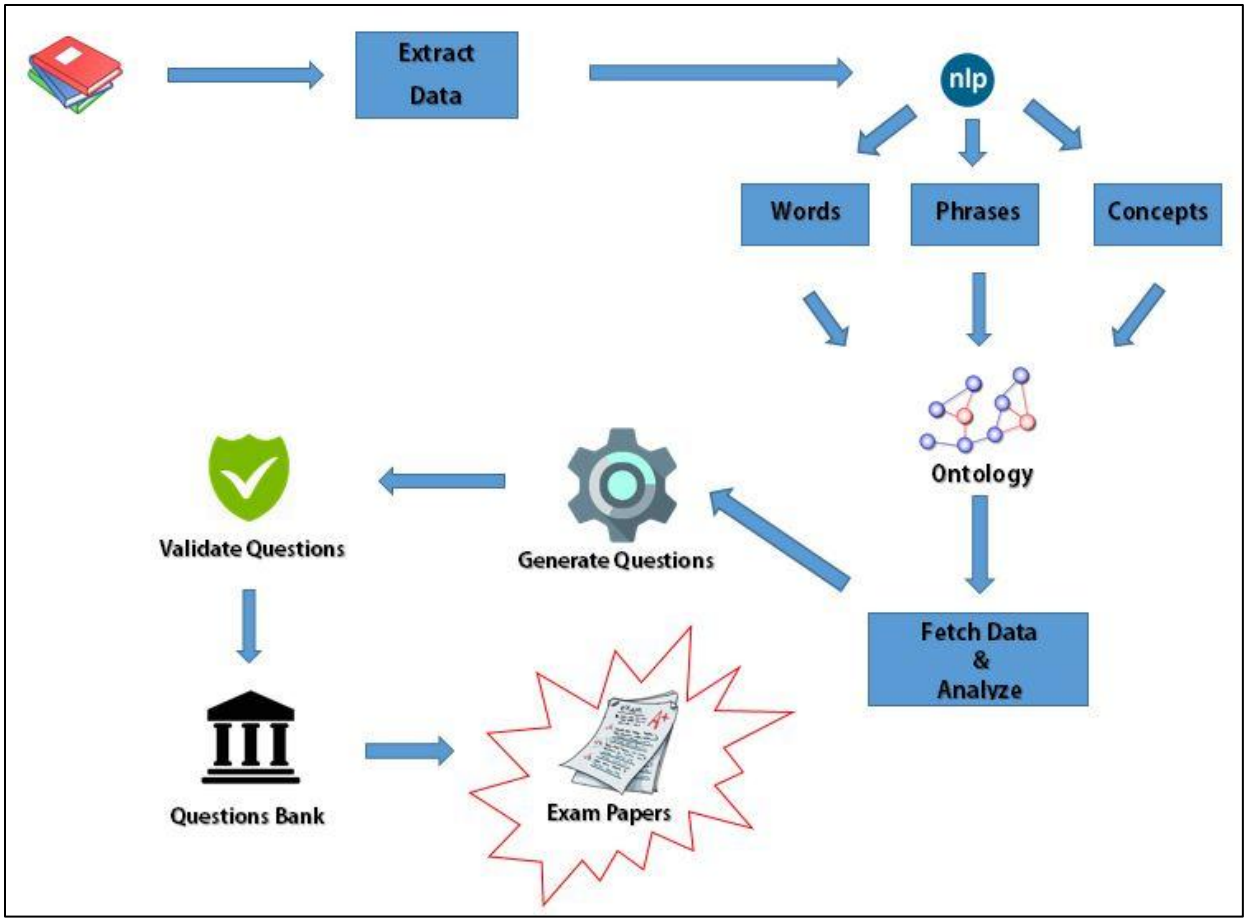

Figure 1:System Diagram

Flow of the function is below

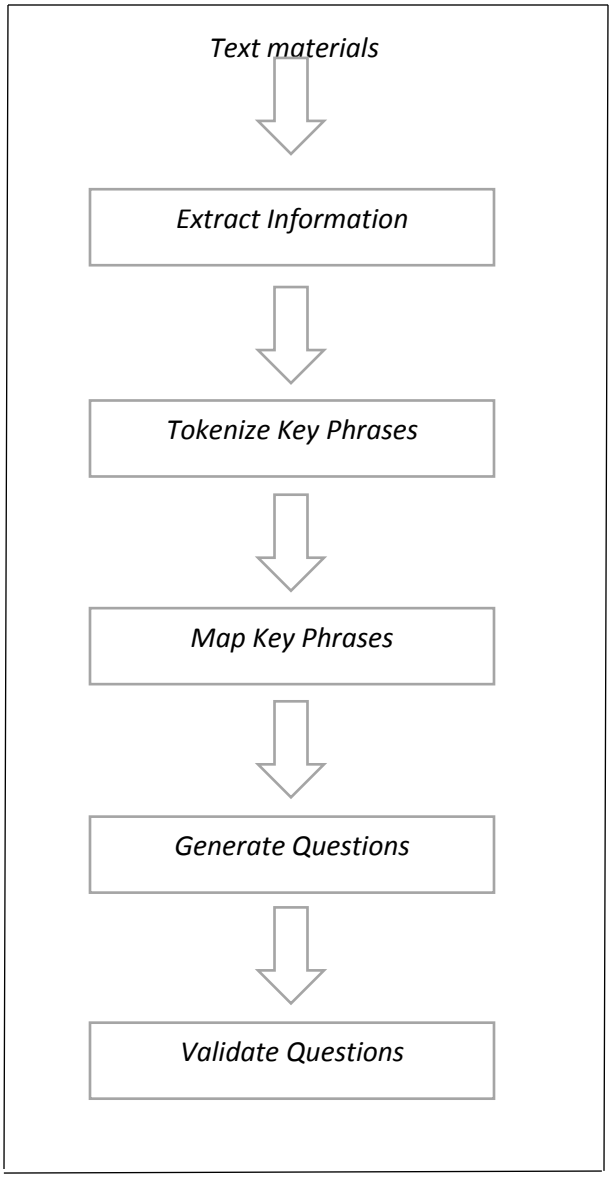

Figure 2.1: Function Flow

QGS system is developed for generating question paper from text materials. For this purpose, end user has to input texts to the system. With sentiment analysis, innovative text analytics and natural language processing(NLP) is employed to extract and classify data. Once the data is extracted, it will then be prepared for analysis. One major component of proposed

system is information extraction(IE). Once information is collected, the desired information the most important part would be to understand the contents of this information.

This where natural language processing comes into play. NLP is a field of computer science and linguistics concerned with the interactions between computers and human – One specific application in NLP that can be used for this purpose is sentiment analysis. It can be used to identify and extract subjective information from the information source collected. Once Information extracted, it will continue to the process of cleaning the extracted information. In this stage, unwanted sentences will be removed before analysis is performed. In addition, any other content that is not deemed relevant to the area of study is also removed from the textual dataset such as includes stop words or words that are not relevant to the course of analysis. When information is cleaned up, there will be two types of sentences. Simple sentences and complex sentences. Complex sentences will be converted to simple sentences[1]. Then using POS tagging method, key phrases will be identified and store them in order to generate questions

2.1 Product perspective

There are some Question Generation projects done by many researches. The main flow of all projects is analyzing the text materials and generating different type of questions such as essay type questions, Multiple Choice Question(MCQ), gap-fill multiple choice questions etc.

Most of the prevailing systems generating questions from text materials but they not filter the important things from the text materials and generate questions, and they not validate the generated questions. In our proposed system, we are extracting the important things from text materials and generate questions from them and also, we validate the generating questions and give percentage for accuracy.

Even though there are similar applications, there is no such exact application as the proposed application. Some sample applications are listed and compared as below.

1) *REVUP: AUTOMATICALLY GENERATING QUESTIONS FROM EDUCATIONAL TEXTS*

RevUp is Automatically generates gap-fill questions from online text. The system analyzes online text and finds most important sentences, then select the main gap-phrases from the selected sentences and choose distractors that are semantically and syntactically similar to the gap-phrase and have contextual fit to the gap-fill question[2].

2) *Question Generation as a Competitive Undergraduate Course Project*

Some groups of students in Carnegie Mellon University(USA) created Question and Answer Generation projects for their under graduate Natural Language Processing course. Each group tried with different techniques of Natural Language Processing(NLP) such as Language Modeling, Part of Speech Tagging, Named Entity Recognition, Parsing etc. And they allowed for students to use any programming language and any existing NLP components to complete their projects.

3) *Automatic Question Generation(AQG)*

The Automatic Question Generation (AQG) approach that generates trigger questions as a form of support for students' learning through writing. The approach first automatically extracts citations from students' compositions together with key content elements. Next, the citations are classified using a rule-based approach and questions are generated based on a set of templates and the content elements. A pilot study using the Bystander Turing Test investigated differences in writers' perception between questions generated by our AQG system and humans (Human Tutor, Lecturer, or Generic Question). It is found that the human evaluators have moderate difficulties distinguishing questions generated by the proposed system from those produced by human (F-score=0.43). Moreover, further results show that our system significantly outscores Generic Question on overall quality measures.

Features	Existing Applications	Proposed Applications
Generating Questions	✓	✓
Allocating Marks	×	✓
Customizing Question Paper	✓ (but limited)	✓
Printout the question paper	Some applications have	✓

Table 2: Comparison between QGS and existing applications

2.1.1 System interfaces

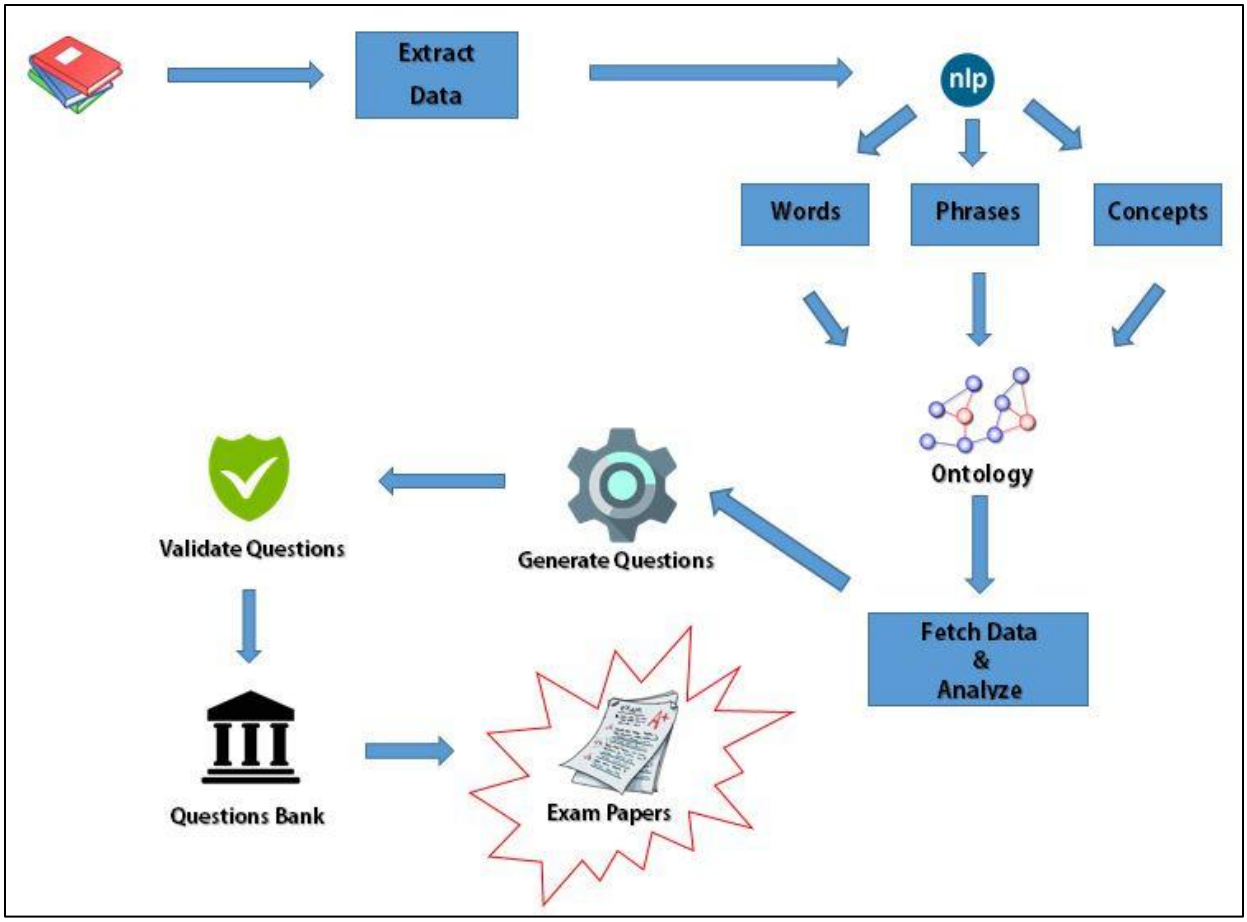

Figure 3: Software Interface

2.1.2 User interfaces

Figure 3: Sign Up

Figure 4: Sign In

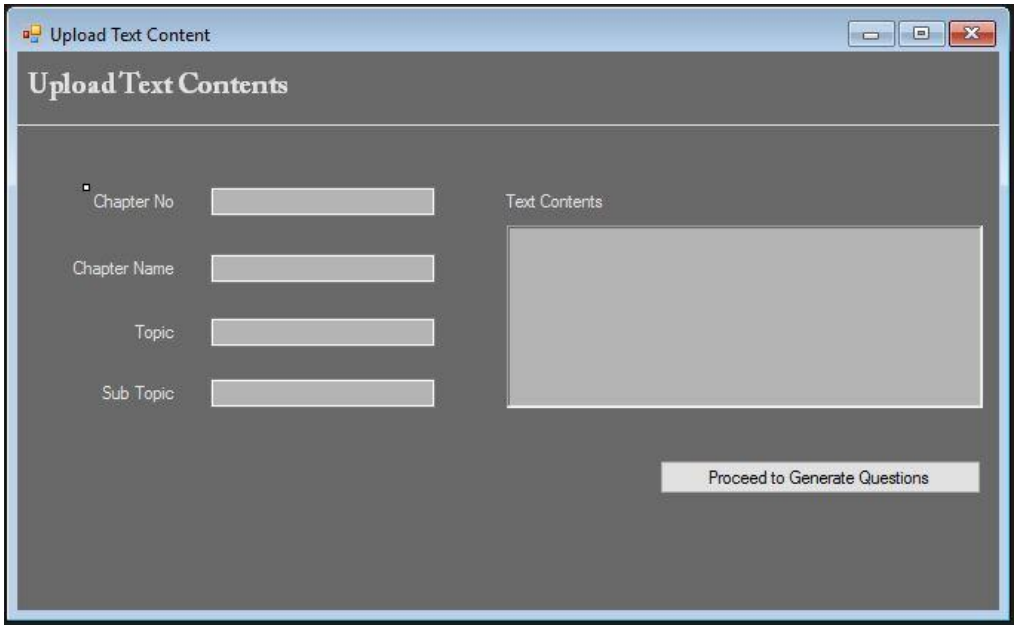

Figure 5: Upload Text Content

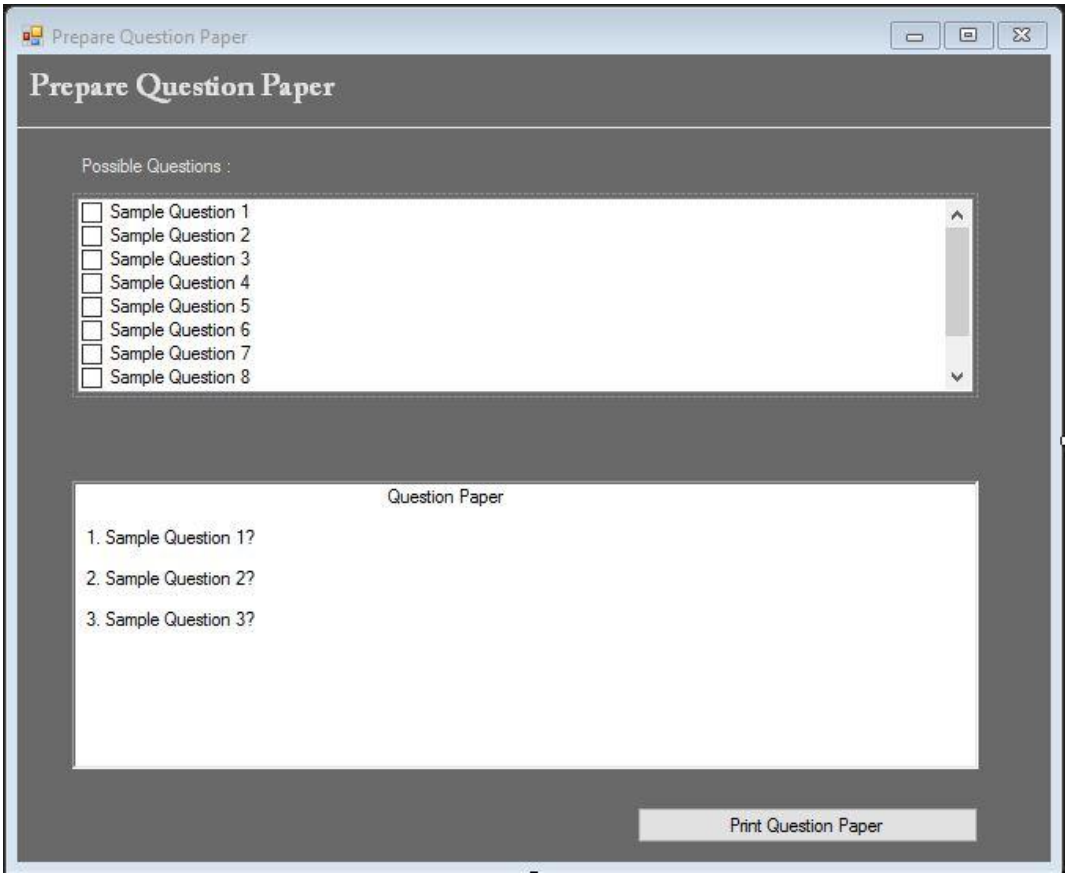

Figure 6: Prepare Question Paper

2.1.3 Hardware interfaces

There are some hardware devices important to run the proposed Question Generation System.

PC: Since this is a windowss based application so user should have a pc with windows 7 or later.

Hard Disk: Minimum 10 GB hard disk space required to run the system.

RAM: Minimum 2+ GB RAM required to run the system.

Keyboard: A keyboard need to enter the paragraph/ text materials to the system.

Printer: User need a printer to get the generated questions to handout

2.1.4 Software interfaces

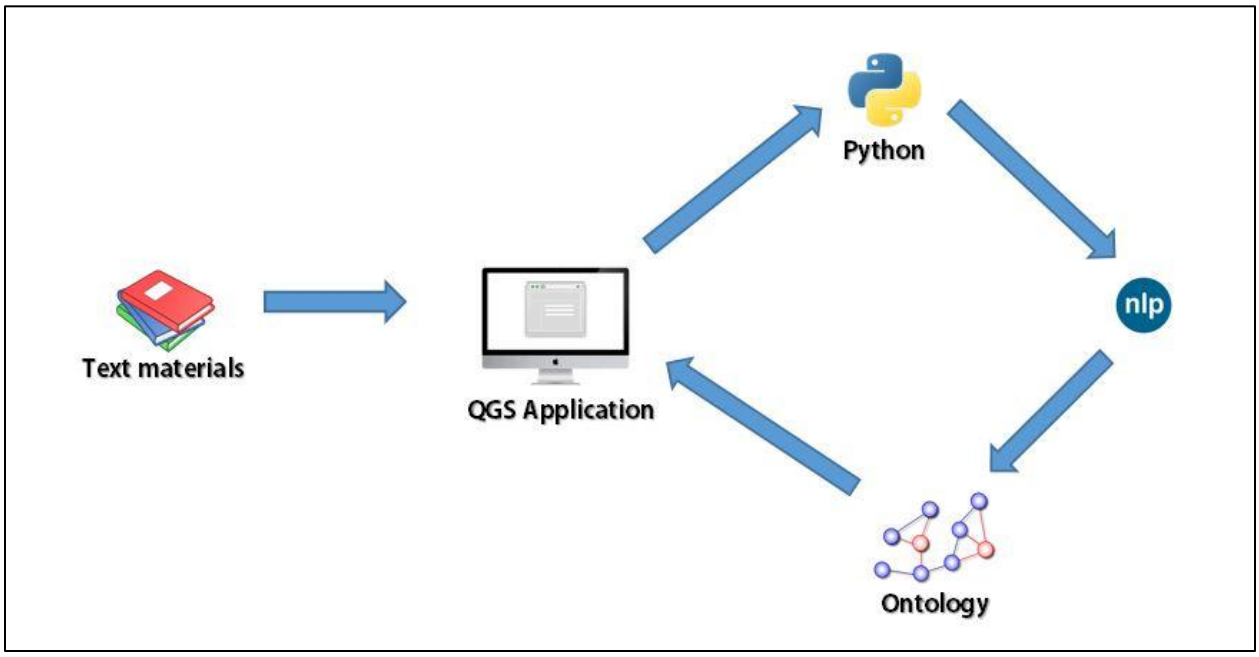

Figure 7: Software Interface

2.1.5 Communication interfaces

Since it is a desktop base application without any network or communication connections, so it does not have any communication interfaces.

2.1.6 Memory constraints

The system should be performed several functions in order. If one function takes much time to complete then the over roll system will get delay. So, each function should complete within few milliseconds. So, to increase the performance, the personal computer must at least require,

- Minimum 10GB of Hard Disk space
- 2GB RAM or higher recommended
- 2GB Cache Memory

2.1.7 Operations

Under this section describes what operations should followed by the users to achieve system functionalities. These operations divide into two main categories such as normal operations and special operations.

Normal operations

I. User wants to install the application.
II. User should register to the system.
III. User want to log in to the system to generate questions.
IV. User should select/brows text materials as input to the system.
V. User should define a limit to generate that number of questions.
VI. Once questions are generated, the user can select the questions by looking the validation/accuracy percentage to print in to the paper.

Special operations

I. System must analyses the text materials which were entered by the user to the system and filtered out the necessary information.
II. Map the relationships between key words and phrases which were extracted from text materials.
III. Generate questions according to the mapping between key words and phrases.
IV. Validate the generated questions and allocate percentage for its accuracy.
V. Print out the user selected questions.

2.2 Product functions

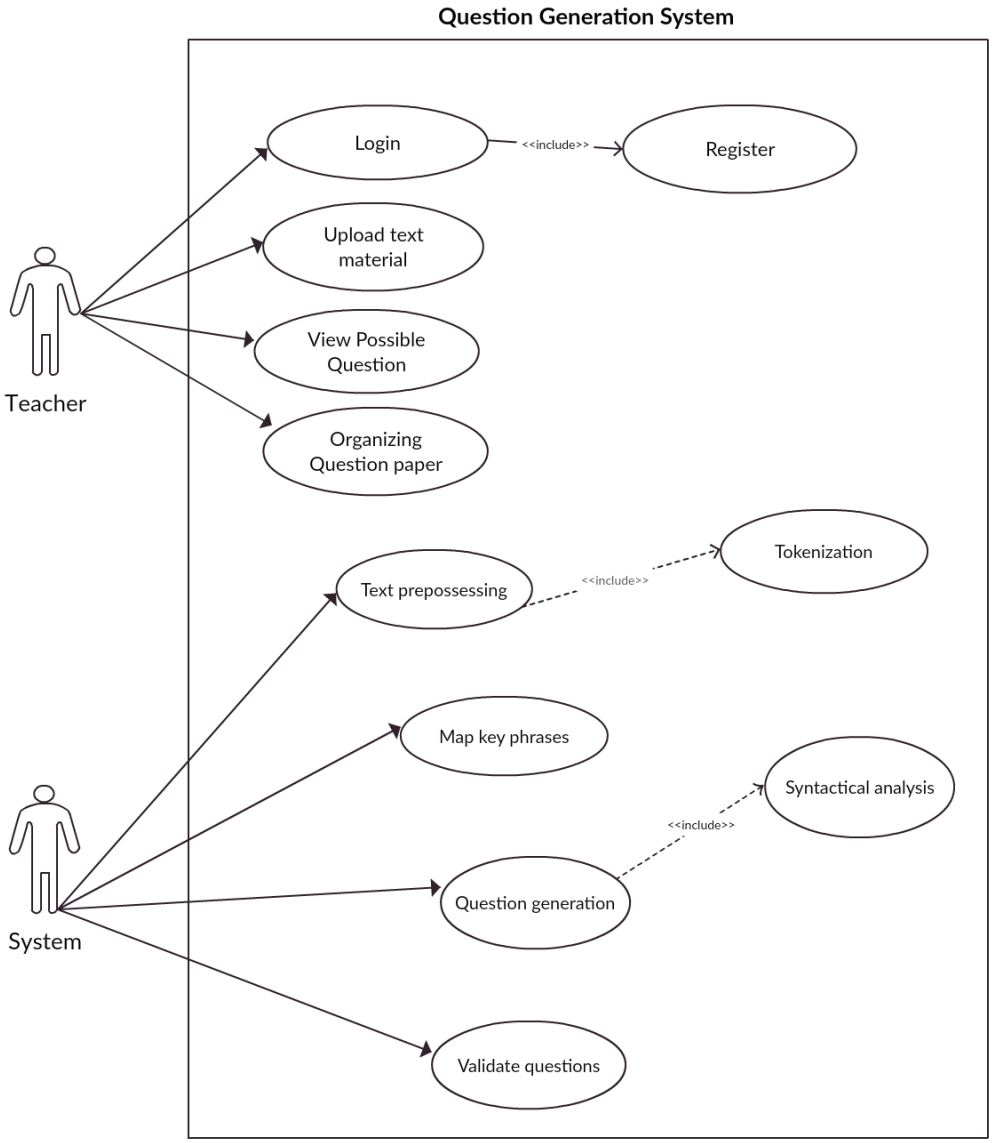

Figure 8: Use Case Diagram for QGS

2.2.1 Use case scenario

Use case ID	01
Use Case	Text Prepossessing
Pre-Condition	User should login to the system with valid credentials.
Primary Actor	System
Main Success scenario	1. Use case begin when user uploaded the text contents. 2. Tokenize text (uploaded text contents) 3. Remove unnecessary text. 4. Convert complex sentence into elementary sentence 5. Identifying key phrases.

Table 3: Use case Scenario

2.3 User characteristics

QGS mainly targeting school/institute teachers who going to prepare exam papers during the examinations. The system will be built in a way that anybody with zero IT literacy can make use of the services provided. Therefore, all interfaces must simple and user-friendly for those users to take advantage from the system.

2.4 Constraints

- System can only analyze text material data written in English language.
- Currently this system will be applicable for some limited inputs. (Eg: Only generating questions for theory basis subjects)
- Natural language processing should have done by using python and NLTK

2.5 Assumptions and dependencies

- Every user has a valid user credentials.
- Lecture contents must be separated according to the chapters.

2.6 Apportioning of requirements

This chapter describe the order of the fulfillment of the requirements of the system. Initially information gathered from text materials and analysis those data using NLTK and Python. Key phrases will be identified and stored them into the ontologies. Identified key phrases will be used to generate questions. Generated questions will be validated and display to the user.

2.6.1 Essential requirements

This requirement should be covered in the first release of the system.

- Cleaning the user input and identifying the key phrases where questions can be generated.
- Ontology creation and mapping the key phrases.
- Generate the questions based on the mapping of key phrases.
- Validate the questions before it displayed to the user.
- Displaying all possible question for a particular text material.
- User has the opportunity to select appropriate question for question paper.
- Allocating marks for each question base on it weight.

2.6.2 Desirable requirements

These requirements are not forced to develop but hope to develop in future releases.

- Developing the system to support multiple languages.
- Developing the system to support complex subjects (So far support only simple theory basis subjects)
- Develop a we based application
- Use more classifier algorithms to give results more accuracy

3 Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

Name of item	Username
Element	Label and Textbox
Description of purpose	Accepts username of the user
Source of input/destination of output	User
Valid range/accuracy/tolerance	Any valid username
Units of measure	-
Data formats	String

Table 4: TextUserName_login

Name of item	Password
Element	Password field
Description of purpose	Accepts password of the user
Source of input/destination of output	User

Valid range/accuracy/tolerance	Exact password for the inserted username
Units of measure	-
Data formats	String

Table 5: TextPassword_login

Name of item	Login
Element	Button
Description of purpose	Triggers validation of login details
Source of input/destination of output	User

Table 6: ButtonLogin_login

Name of item	Chapter name
Element	Label and Textbox
Description of purpose	Accepts Chapter name from User

Source of input/destination of output	User
Valid range/accuracy/tolerance	Appropriate Chapter name
Units of measure	-
Data formats	String

Table 7: TextChapterName_inputData

Name of item	Sub Topic
Element	Label and Textbox
Description of purpose	Accepts Sub topic from User
Source of input/destination of output	User
Valid range/accuracy/tolerance	A sub topic of the chapter
Units of measure	-
Data formats	String

Table 8:TextSubTopic_inputData

3.1.2 Hardware interfaces

It our system it may require considerable processing power as well as speed. There for main hardware resource required is, computer with Windows Operating system (MS Windows 7 or higher) that should contain following requirements.

- Minimum 10 GB hard disk space required to run the system.
- Minimum 2+ GB RAM required to run the system.
- Monitor
- keyboard
- Printer

3.1.3 Software interfaces

Windows running operating system needed where it has latest version of Python, mysql and NLTK installed.

3.1.4 Memory constraints

This system is doing many parallel tasks. In order to perform this, must at least require following requirements.

- Minimum 10GB of Hard Disk space
- 2GB RAM or higher recommended
- 2GB Cache Memory

3.2 Classes/Objects

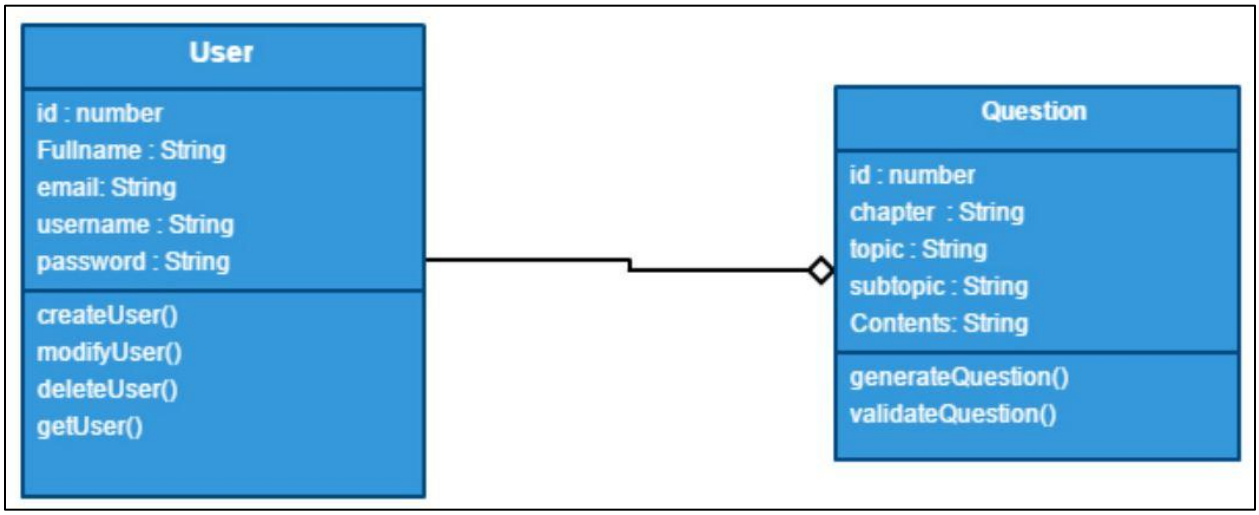

Figure 9: Class Diagram

3.3 Performance requirements

Application is using Natural processing task in python. This takes considerable amount of processing power for personal computer. The minimum hardware requirement is mentioned in section 2.1.3. Other than this, application perform well as much as possible.

3.4 Design constraints

The user interface of the system is designed in a way such that it will grab hold of as many users who wish to use this system. Since most the knowledge in IT literacy varies from person to person. The technical design of the system database will be designed in a very simple and accurate way to avoid maintenance problems. System design should adept the scalability, data formats and cost benefits tradeoff to the system.

Main design constraints are,

- Keep the interfaces simple as much possible.
- Easily understandable architecture.
- Less IT literacy people use the system easily.
- Python and NLTK use for question generation.

3.5 Software system attributes

3.5.1 Reliability

- Data should not be corrupted.
- The system should be tested using different testing metrics to verify the accuracy of generated question.
- Data is not lost while retrieving information from the ontologies.
- Ontology updates should not lead to data loss or redundant data.

3.5.2 Availability

- The application should be available when ever required.

3.5.3 Security

- The application is password protected. Users should access the application by entering their username and password.

3.5.4 Maintainability

- Appropriate coding standard ad naming convention will be used in the system development.
- Proper indentation and code commenting will be used to ensure code understandability.
- SRS will be used while testing the application.

4 References

[1] Noah A. Smith, Michael Heilman, Rebecca Hwa, "Question Generation as a Competitive Undergraduate Course Project" University Pittsburgh, PA, 2008

[2] Revup: "Automatically generating questions from educational texts" [Online]

Available:
https://www.googlesciencefair.com/projects/en/2015/7151ae4ff6b70198aafc08fbee39127ad0913cd407d98d8b596a85c14ed57ba9