Software Requirements Specification

for

Quiz Application

Version 1.0

Prepared by Asher Everen & Shaun Fernandez

College of Engineering, Pune

10th August 2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this project is to develop a functional and robust application that can provide the user the functionality to create quizzes and to also provide functionality to generate links to participate in the quizzes. The quiz-master will input questions and options and choose the correct answer, also the quiz-master has the option to load questions from an online data-bank of questions. The quiz participant will be able to do the test and see the results at the end. Only the quiz-master gets all the results after a set time period.

1.2 Document Conventions

The font used in the document is “Times” with size “14”. Topic headings have been typed in bold.

1.3 Intended Audience and Reading Suggestions

This SRS is intended mainly for developers, project managers, users and testers. Index page is for everyone. Developers can concentrate on the technology stack being used; product managers and marketing staff can concentrate on the scope and business model of the product and users can focus on UI and robustness of product.

1.4 Product Scope

Schools, Colleges and Companies can use this application as a convenient platform for creating tests and carrying them out on the internet. All sorts of tests can be done through such an application on many domains.

.

2. Overall Description

2.1 Product Perspective

A user can add questions to the test via a web application and use pre existing questions from the database. They can add the answers and pick the correct answer. They can also select the maximum marks they want for a quiz and a quiz with questions adding up to those marks will be generated.

2.2 Product Functions

Through the product, users will be able to perform the following major functions

1) Create a quiz
2) Participate in the quiz
3) View results of a quiz

2.3 User Classes and Characteristics

The main user classes will be participant (student) and quiz-master (teacher). The participant will answer the questions to the quiz. The quiz-master can create a quiz and view results.

2.4 Operating Environment

The product will be operated on a web server. Any web browser can be used to run the product. An Ubuntu server will be used for hosting the application.

2.5 Design and Implementation Constraints

This is a web application, however we do not have access or funds to purchase a domain or server. It will be hosted locally on our machine.

2.6 User Documentation

Help page will be provided to the user if necessary.

2.7 Assumptions and Dependencies

This product will assume the user has a working web browser and a working internet connection to participate and create quizzes.

3. External Interface Requirements

3.1 User Interfaces

A Web Application user interface will be developed.

3.2 Hardware Interfaces

No hardware constraints for this application.

3.3 Software Interfaces

A web application user interface will be developed using the Flask/Django framework in Python.

3.4 Communications Interfaces

The communication between different segments of the system i.e (interface and database) is very essential as they are contingent on each other. Communication between the database and web interface is established using the Flask framework in Python.

4. System Features

4.1 Creating Quiz

4.1.1 Description:

Users will be given a Web Portal GUI to create a quiz. They select the questions and add more questions as necessary.

4.1.2 Response Sequences

A “Create” button will be displayed on the portal, which the user will have to click. After this a response will be generated displaying a link to participate in the quiz.

4.1.3 Functional Requirements

Only content in English will be accepted for this.

4.2 Participate in Quiz

4.2.1 Description:

User has link from quiz-master. They use their account to participate in the quiz.

4.2.2 Response Sequences

The user will have to click on the correct option for each question. After the user has clicked on “Submit” at the end of the quiz, they will be able to see their results. The Quiz-master also has access to the results.

4.2.3 Functional Requirements

System needs to be connected to the internet.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The requirements for the peak performance of this product are as follows:

1. Good network connection with the internet.
2. Python3 programs should be executable.

5.2 Safety Requirements

Not Applicable

5.3 Security Requirements

Results of other users should not be accessible to anyone but that user that took the test and the quiz-master.

5.4 Software Quality Attributes

Adaptability : Can be incorporated into any web browser.

Flexibility : There is a fixed probability of outcome so the software is inflexible.

Maintainability : Debugging can be done easily. Modification and extension of functionality is possible but with limitations.

Reliability : The system should run without interruption using high-uptime server.

Reusability : A Modular programming approach ensures that it is reusable.

Testability : Program is modular and structured which makes testing easy.

5.5 Business Rules

All Individuals can use the web application, since it is user friendly. No specific prerequisite knowledge is required to use the application.

6. Other Requirements

Not Applicable