Software Requirements Specification

for

Quiz Application

Version 1.0 approved

Prepared by Abhilekh Shrivastava

NITK Surathkal

09/01/2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to provide a whole and detailed description of an Online Quiz Application . It will explain the features, interfaces and the functions of the system while providing a black box view of the system itself to the developers, stakeholders and any other concerned party.

1.2 Document Conventions

The document is prepared using Google Docs and is been written according to the standard IEEE template. The font used for the body and headings is Times Roman New . The font size that has been used to type this document is 12 pts and headings sized 14 pts. The headings are made bold in order to distinguish.

1.3 Intended Audience and Reading Suggestions

This documentation is the foundation of the development of the project. It would enable the developers to identify the functional and nonfunctional requirements in order to develop the project accordingly. It would be also be useful for the testers to check if the intended results are achieved by the project as specified in this documentation. The outline of this documentation is as follows:

- Overall Description
- System Features
- External Interface Requirements
- Non functional Requirements

1.4 Product Scope

This software can provide a framework for the process of conducting an arcade style quiz. The same system can be extended to a survey system. The software can be used in educational institutions to save time in paper evaluations. It can aid in testing students and developing their skills. The application can be effectively used by educational institutions and training centres to develop their strategy in examination conduction and to improve efficiency.

1.5 References

- IEEE, "*IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*", IEEE Computer Society, 1998

2. Overall Description

2.1 Product Perspective

The Quiz Application System is a self contained system whose purpose is:

- Test the knowledge of an applicant in any given field.
- To maintain a list of applicants and keep a database of all the users and their scores.

2.2 Product Functions

The idea is to minimize the user involvement in the back end while simultaneously providing a fair and valid ranking system for all who take the quiz.

The Application should allow a user to:

- Create an account so that his scores can be recorded.
- Generate a random set of 10 questions for the user to answer.
- Find out how he has performed with respect to others.
- Change his password for additional security.

The Application should allow the admin to:

- Reset the leaderboards.
- Change the question set.

2.3 User Classes and Characteristics

Users using the Quiz Application need not have expertise working with computers. A general understanding of the English language and an average understanding of a computer system is sufficient.

The Admin, however, needs to have knowledge in database management in addition to the requirements specified above. The admin also requires domain knowledge of the topic on which the quiz is to be conducted.

2.4 Operating Environment

The goal of this application is to provide an automated terminal application that allows a user to take a quiz anywhere using a personal computer and g++ compiler.

2.5 Design and Implementation Constraints

- The website supports English language only.
- Database to be used in form of files only.
- There should be IDE/g++ compiler..

2.6 User Documentation

A well operating application must provide a proper user documentation which would allow the application to be used by a person from any domain. Therefore, in order to help customers get acquainted with all the features of the application, the user documentation will be provided in which details regarding all the functionalities will be given.

2.7 Assumptions and Dependencies

The following are the assumptions made and various dependencies to be taken care of:

- User is expected to know English.
- No email verification for creating new account.

3. External Interface Requirements

3.1 User Interfaces

The basic objective is to provide a simple and user friendly GUI. The various screens that will will be encountered by the user are

- Login screen
- Question screen (during the game)
- Final score screen
- Leaderboard

3.2 Hardware Interfaces

- Requires a mouse and keyboard for inputs
- Monitor for displaying the outputs

3.3 Software Interfaces

- Works on all common Operating systems
- The player module has no direct interaction with the database. A request by the user calls a corresponding appropriate function in the system module.
- User information and questions are stored on a file which can be accessed by the system
- Minimum requirement of 500mb RAM to run the application smoothly

3.4 Communications Interfaces

The system needs no communication from any other device, it uses its own compiler and file system to operate.

4. System Features

The Quiz application has two actors - normal user and admin. Features for both are listed below in detail.

4.1 Create Account

4.1.1 Description and Priority

The quiz application allows users to first login to the application before taking the quiz

4.1.2 Stimulus/Response Sequences

Type 1 in order to choose option Create Account from the menu.

4.1.3 Functional Requirements

Req 1) There should be a username and password for a user.

Req 2) Need to check if username already exists, otherwise create new record in database.

Req 3) Application menu should be displayed in the case of correct credentials and message to choose new username otherwise.

4.2 Login

4.2.1 Description and Priority

The quiz application allows users to login using their preferred usernames and passwords.

4.2.2 Stimulus/Response Sequences

Keyboard Input for this option.

4.2.3 Functional Requirements

Req 1) Both username and password should be given by the user.
Req 2) The username and password input by the user should match the ones in the database.
Req 3) Application menu should be displayed in case of correct credentials or error .

4.3 Get Rank

4.3.1 Description and Priority

The quiz application allows users to check their rank with respect to other players taking test.

4.3.2 Stimulus/Response Sequences

Keyboard Input for this option.

4.3.3 Functional Requirements

Req 1) The username of the current user.
Req 2) The username should exists in the database and corresponding rank is fetched from it.
Req 3) Rank of the current user.

4.4 Take Test

4.4.1 Description and Priority

The application allows users to take a quiz of 10 questions.

4.4.2 Stimulus/Response Sequences

Keyboard Input for this option.

4.4.3 Functional Requirements

Req 1) The username of the current user
Req 2) There should be 10 questions generated randomly and the user answers should be evaluated.
Req 3) Score of the user.

4.5 Generate Questions

4.5.1 Description and Priority

The Application generates a question from the database.

4.5.2 Stimulus/Response Sequences

Keyboard Input for this option.

4.5.3 Functional Requirements

Req 1) The question number of the question to be generated.
Req 2) The corresponding question to the question number should be retrieved from the database.
Req 3) The question should be displayed as a result.

4.6 Check Answer

4.6.1 Description and Priority

The user's entered answer is compared with the answer in the database.

4.6.2 Stimulus/Response Sequences

Keyboard Input for this option.

4.6.3 Functional Requirements

Req 1) The user should enter their answer and the question number.
Req 2) The answer corresponding to the question number should be retrieved from the database.
Req 3) If the two answers are equal, it should display correct else incorrect.

4.7 Change Password

4.7.1 Description and Priority

The user is allowed to change his password.

4.7.2 Stimulus/Response Sequences

Keyboard Input for this option.

4.7.3 Functional Requirements

Req 1) The username, current password and new password should be given by the user.
Req 2) The username and password are authenticated after which the new password is set
Req 3) If the authentication is successful, a success message is sent else an error message is sent.

4.8 Update Questions

4.8.1 Description and Priority

If the current user is admin, he should be allowed to change the questions database.

4.8.2 Stimulus/Response Sequences

Keyboard Input for this option.

4.8.3 Functional Requirements

Req 1) New questions database
Req 2) If user is authenticated to have admin permission, the questions database should be replaced by the new one.
Req 3) Success or failure message

4.9 Reset Leaderboard

4.9.1 Description and Priority

If current user is admin, he should be allowed to reset the leaderboards and make all players have a score of 0.

4.9.2 Stimulus/Response Sequences

Keyboard Input for this option.

4.9.3 Functional Requirements

Req 1) The user should be an admin
Req 2) If user has admin permission, the players scores in the database needs to be reset to 0 for all players
Req 3) Success or failure message.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Following are performance requirements which define acceptable response times for system functionality.

- The load time for user interface screens shall take no longer than two seconds.
- Selecting any option shall return results within five seconds
- Data in the database should be updated within 2 seconds on creating new account and adding new questions.

5.2 Safety Requirements

All the data stored in form of files should be safe, so that users information and scores are not lost.

5.3 Security Requirements

- Robust login mechanism put in place to ensure security of user accounts
- System is built in a manner to prevent players from gaining access to the database.

5.4 Software Quality Attributes

The Quiz application will be available to all users with an g++ compiler, irrespective of the hardware of the machine they access the application from. It will be easy to use and understand - the Graphical User Interface will be designed such that a new user of any age will be able to quickly grasp the various features of the website.

5.5 Reliability Requirements

- Produces a new set of questions for each test and maintains a consistent difficulty level while producing a new set of questions for a player who is retaking the quiz.
- System allows user to take test only upon authentication.
- Regular database backup.

5.6 Availability Requirements

- Accessible by anyone, anywhere.
- Available on all platforms for easy access

5.7 Maintainability Requirements

- Admin login for administration to perform necessary maintenance activities

5.8 Usability Requirements

- The system is easy to handle and navigates in an understandable way with no delays.

6. Other Requirements

Requires files to hold the information regarding user accounts and authentication information. The files will also hold the questions and their respective answers.

Appendix A: Glossary

g++ compiler - A compiler which can compiler and execute .cpp code .