Quizwave

Software Requirements Specification

V2

22.10.15

LAWAL, Muideen Adekunle

IPINLAYE, Olabisi Temidayo

Prepared for

Software Engineering Methodology

Instructor: Saleh H.M (Ph.D.)

2015

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 20.10.2015 |  | LAWAL, Muideen Adekunle |  |
| 07.11.2015 |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |
| --- | --- | --- |
| **Signature** | **Printed Name** | **Date** |
|  | Saleh H.M (Ph.D) | 19.10.2015 |
|  |  |  |
|  |  |  |

**Table of Contents**

Revision History 2

Document Approval 2

1.1 Purpose 5

1.2 Scope 5

1.3 Definitions, Acronyms, and Abbreviations 6

1.4 References 6

1.5 Overview 6

2. General Description 7

2.1 Product Perspective 7

2.2 Product Functions 8

2.3 User Characteristics 8

2.4 General Constraints 9

2.5 Assumptions and Dependencies 9

3. Specific Requirements 9

3.1 External Interface Requirements 9

3.1.1 User Interfaces 10

3.1.2 Hardware Interfaces 10

3.1.3 Software Interfaces 11

3.1.4 Communications Interfaces 11

3.2 Functional Requirements 11

3.2.1 Download App 11

3.2.2 Mobile Application: User Registration 12

3.2.3 Mobile Application: User Login 12

3.2.4 Mobile Application: Profile Page 13

3.2.5 Mobile Application: Start Test 13

3.2.6 Mobile Application: Submit Test 14

3.2.7 Mobile Application: Test Result 14

3.2.8 Mobile Application: Performance Chart 15

3.2.9 Web Portal: Login 15

3.2.10 Web Portal: Upload Question 16

3.2.11 Web Portal: Manage Information 16

3.3 Use Cases 17

3.3.1 Use Case: User Registration 17

3.3.2 Use Case: User Login 17

3.3.3 Use Case: User want to state a test 17

3.3.4 Use Case: User submit test 18

3.3.5 Use Case: User view performance chart 18

3.3.6 Use Case: User view test history 18

3.5 Non-Functional Requirements 18

3.5.1 Performance 18

3.5.2 Security 18

3.5.3 Safety 19

3.6 Inverse Requirements 19

3.7 Design Constraints 19

3.8 Logical Database Requirements 19

4. Analysis Models 20

4.1 Sequence Diagrams 20

4.3 Use Case Diagram 20

4.2 State-Transition Diagrams (STD) 21

A. Appendices 21

A.1 Appendix 1 21

A.2 Appendix 2 21

**1. Introduction**

## 1.1 Purpose

This document describe in details, all the features, functions, and specifications of the requirement for the mobile app under the **Quizwave**. It will outline the constraints, interface and interactions with other application/services. This document will be a starting point in developing the mobile app. The intended audience of this document is our fellow students in Software Engineering (Software Development Management Track) as well as the instructor.

## 1.2 Scope

**Quizwave** is a robust mobile app running on Android OS which provide intuitive user interface, containing pool of lecture videos and questions on different topics and subject areas where registered user (students) can login to stream videos of lectures and take test based on their preferred courses/subjects. Test questions are randomly displayed on the app on the users’ phone. Test result will be displayed after the test is taken in form of report card and will also be synchronized with the web portal, also the result can be printed in pdf. It will help students in monitoring their progression in each course by allowing them to view chart of their performances in those courses.

Moreover, series of lecture videos are displayed on the app and can be streamlined based on different courses offered by the users (students). This will be helpful for some of the students who are absent in class by not missing anything details of the lectures they missed.

User can also share their test result with friend on different online social media and through sms.

More so, questions will be made available in the mobile app through the web-portal by the system administrator who also maintain the system and user can access their test result on the portal as well. The only types of question available in the app will multiple choice or true and false questions.

## 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| SRS | Software Requirements Specification |
| Web Site | A place on the world wide web |
| MMS | Multimedia Messaging Service (picture message) |
| OS | Operating System |
| App | Application |
| Smartphone | Mobile Phone |
| SD card | Secure Digital memory card |

## 1.4 References

[IEEE] The applicable IEEE standards are published in "IEEE Standards Collection",

2001 edition.

[Open Source] http://developer.android.com

[Sommerville] "Software Engineering", version 9.

## 1.5 Overview

This SRS document is intended for project advisor and all individuals participating in this project. The remainder of this document includes three more chapters. It also contains glossary of terms found throughout this document contained in the appendix. Generally, readers will gain understanding of the application features, its functional and non-functional requirements.

Chapter 2 give an overview of the app functionalities. It also discusses the constraint, assumptions and allocation of requirements for the development of the mobile app.

Chapter 3 describes the external interfaces to the **Quizwave** app that allow for communication across the app. Also included in this section is the list of specific functions that are included in the **Quizwave**. All performance and logical data requirements are addressed in this section, as well as any design constraints, standards compliance requirements, and software system attributes needed for the **Quizwave** app.

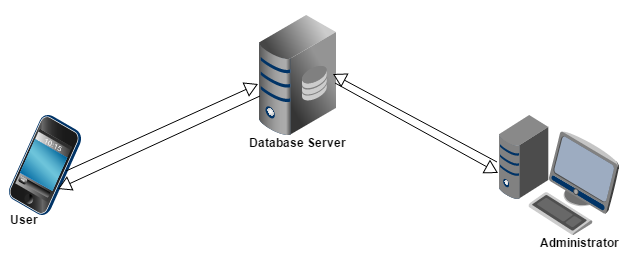
Chapter 4 gives the diagrammatic representation of the app and it core functionalities.

# 2. General Description

Overview of the whole system is described in this section. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will describe the functionalities that this mobile app will provide for its user and what type of user will find this app useful.

## 2.1 Product Perspective

**Quizwave** is an android-based mobile app which allows students to take test on different course/subject areas on their mobile phones. This application is self-contained - it is not build into a pre-existing system but there exist a server-side component (consisting of the web portal) from which the mobile app consume and synchronize services and which will be used for uploading question and answer, managing user information and the system as a whole**.**

**

*Fig 1: Block Diagram for the Quizwave App*

*Source: Author*

Both the mobile application and web portal will communicate with a database in slightly different ways. The mobile application will only use the database to get user data, test question and test result while the web portal will also add and modify data. All of the database communication will go over the Internet.

Test questions is not saved on user phone so as to avoid overloading the users' phone memory while running the app.

## 2.2 Product Functions

This app enable users to be able to take tests on different course of interest. A limited amount of question will be displayed randomly for users to attempt and test result (score card) will be generated for each test the user takes.

On the other hand, the web portal will provide information about the app, provide up-to-date test question for the mobile app and it will also provide information about the system.

## 2.3 User Characteristics

Each type of users that will interact with this system have their own requirements. In general there are two types of users that will interact with the system and are described below;

* User (Student): Student will have to register into the app for login to use this app and take tests. Student will be able to check result of their test as soon as they finishes it, print a report card in pdf format and view a chart of their performance in a particular course. This mobile app requires minimum expertise to use, any Android smartphone user of any background will be able to use this app.
* Administrator: The system administrator uses the web portal to access the system. The administrator uses the portal to upload questions and answer for use on the mobile app and manage user information and the system in general. The system administrator should have server and database experience to fully manage the portal.

## 2.4 General Constraints

The Android OS on the student phone is a constraint. User with Android OS version below 4.4 will not be able to print their report card to pdf.

The Internet connection is also a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

Other constraints such as limited memory and processing power are also worth considering.

Both the web portal and the mobile application will be constrained by the capacity of the database. Since the database is shared between both application it may be forced to queue incoming requests and therefor increase the time it takes to fetch data.

## 2.5 Assumptions and Dependencies

Assumption is made that student has a mobile phone running Android OS.

# 3. Specific Requirements

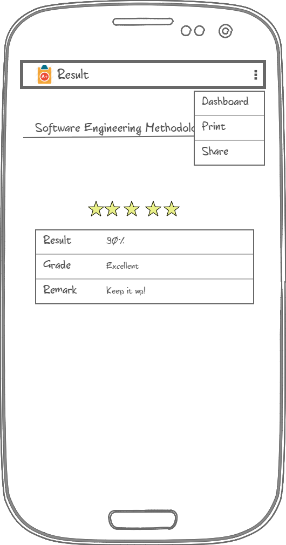
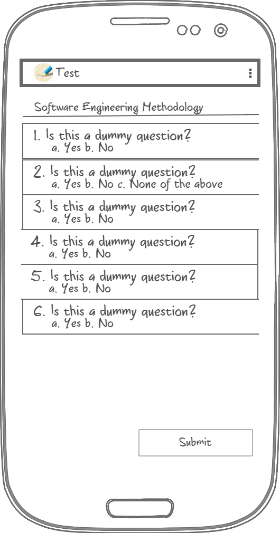
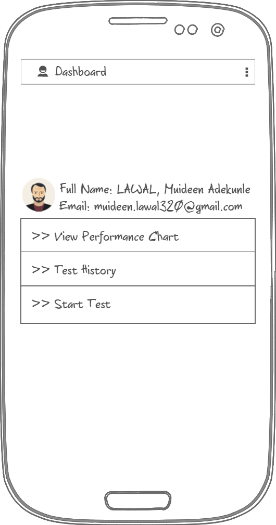
Detail description of the system and all its features is given in this section - thus it contains all of the functional and quality requirements of the system.

## 3.1 External Interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

### 3.1.1 User Interfaces

The first time the user opens this app he/she will see the log-in page when he/she opens the application, see Figure 2. If the user has not registered, he/she would see a navigation button to the register page on the login page. User will supply his login details (username and password) to login and after successful registration, the app takes the user to his dashboard (see fig. 4 below) where he can then navigate to the different pages in the app. When user click on “Take Test”, user is prompt to select the course he/she will like to take test for and subsequently the test page will appear (see fig. 5 below) where user will answer questions on the chosen course and after the user submit the test, the result is displayed as in Fig. 6 below.



*Fig 2: Login Fig3: Register Fig4: Dashboard Fig 5: Test Page Fig 6: Result Page*

*Source: Author*

### 3.1.2 Hardware Interfaces

**Quizwave** is a mobile app running on Android mobile phone as such it does not have any direct hardware interface. In theory the application will be able to run by other devices that can emulate the Android, but this will not be a consideration during design.

### 3.1.3 Software Interfaces

The app remotely retrieve data from a MySQL database that is already set up and is the same the web portal connects to. This allow for use both by the student on the mobile phone and the system administrator on the web portal. The mobile app communication to the database is a read-only operation. The only communication will be between the phone and the server housing the database, which will be sending queries and receiving the information back.

### 3.1.4 Communications Interfaces

This mobile app will synchronize specific data with the database server so as to keep test question and user info up-to-date on the app. The communication is handled by the underlying operating system for both the mobile application and the web portal.

## 3.2 Functional Requirements

### 3.2.1 Download App

3.2.1.1 Introduction

The mobile app will be made available on the web portal and the Google play store

3.2.1.2 Inputs

None

3.2.1.3 Processing

Downloading Quizwave app to the phone.

3.2.1.4 Outputs

**Quizwave a**pp

3.2.1.5 Error Handling

Throws exceptions if the phone memory is out of space

### 3.2.2 Mobile Application: User Registration

3.2.2.1 Introduction

When the user launch the application for the first time, he/she would be able to register.

3.2.2.2 Inputs

The user must provide his full name, email and password to be able to register.

3.2.2.3 Processing

The information is sent to the server and stored in the database.

3.2.2.4 Outputs

A message is displayed to the user. “Successful” complete registration, “Unsuccessful” for error in registration data.

3.2.2.5 Error Handling

Throws exceptions if input field is empty (e.g. if full name, email or password field is empty).

### 3.2.3 Mobile Application: User Login

3.2.3.1 Introduction

Given the user have registered, then the user should be able to login to the **Quizwave** app.

3.2.3.2 Inputs

User supply his email as username and password to the username and password field.

3.2.3.3 Processing

Accepts user login detail (username and password) displaying the password in human unreadable form and then validates the username/password against the one in the database server.

3.2.3.4 Outputs

User gain access to the app - his dashboard, after successful login.

3.2.3.5 Error Handling

Throws exceptions if login details is incorrect or input field is empty (e.g. if username or password field is empty).

### 3.2.4 Mobile Application: Profile Page

3.2.4.1 Introduction

A user will have dashboard. This displays the user details (name, email) and links to actions which user can perform is also there.

3.2.4.2 Inputs

None

3.2.4.3 Processing

Retrieve user data from the database server

3.2.4.4 Outputs

Outputs user data and links.

3.2.4.5 Error Handling

Throws exception handling if user internet connection is bad.

### 3.2.5 Mobile Application: Start Test

3.2.5.1 Introduction

Allows user to take test on their preferred course.

3.2.5.2 Inputs

None

3.2.5.3 Processing

Retrieve question for the particular course from the database server.

3.2.5.4 Outputs

Display a list of questions and their options.

3.2.5.5 Error Handling

Throws exceptions handling if no question is found on the database server.

### 3.2.6 Mobile Application: Submit Test

3.2.6.1 Introduction

Here the user submit the question after choosing their answers/when they are done answering the question.

3.2.6.2 Inputs

Questions and answers.

3.2.6.3 Processing

Send the question and answer that the user have chosen to the database server.

3.2.6.4 Outputs

Outputs "Test un/successfully submitted" depending on outcome of function.

3.2.6.5 Error Handling

Throws exceptions handling if the test was not properly submitted.

### 3.2.7 Mobile Application: Test Result

3.2.7.1 Introduction

When the user clicks "view result" after completing a test, this function shows the test report card.

3.2.7.2 Inputs

None

3.2.7.3 Processing

Accesses the user test score and remarks.

3.2.7.4 Outputs

Outputs users test score and result.

3.2.7.5 Error Handling

Throws exceptions if no result was found for the test.

### 3.2.8 Mobile Application: Performance Chart

3.2.8.1 Introduction

When the user clicks "View Performance Chart", this function is triggered to display user graph of progression in the selected course.

3.2.8.2 Inputs

None

3.2.8.3 Processing

User previous scores in the particular course selected is retrieved from the database server.

### 3.2.9 Mobile Application: Videos/Lectures Streaming

3.2.9.1 Introduction

Allow users to stream videos of selected lectures of their preferred course(s).

3.2.9.2 Inputs

None

3.2.9.3 Processing

Retrieve videos of lectures from the database server.

3.2.9.4 Outputs

Display list of videos of lectures of the preferred course.

3.2.9.5 Error Handling

Throws exceptions if there are no videos of lectures from the database server.

### 3.2.10 Web Portal: Login

3.2.10.1 Introduction

The administrator will be able to the web portal of the Quizwave app.

3.2.10.2 Inputs

The administrator supply his username and password.

3.2.10.3 Processing

Accepts the login detail (username and password) displaying the password in human unreadable form and then validates the username/password against the one in the database server.

3.2.10.4 Outputs

The administrator will gain access to the portal, if the login detail is correct

3.2.10.5 Error Handling

Throws exceptions if login details is incorrect or input field is empty (e.g. if username or password field is empty).

### 3.2.11 Web Portal: Upload Question

3.2.11.1 Introduction

The administrator will be able to upload test question together with their answer to the database.

3.2.11.2 Inputs

Questions, options and answer.

3.2.11.3 Processing

Add questions together with their options and answer to the database.

3.2.11.4 Outputs

Outputs "un/successful insertion" depending on outcome of function

3.2.11.5 Error Handling

Throws exception handling if administrator tries to upload blank question

### 3.2.12 Web Portal: Manage Information

3.2.12.1 Introduction

The administrator will update information on the web portal from time to time

3.2.12.2 Inputs

Contact details (email, phone number), courses name.

3.2.12.3 Processing

Add course name and contact details when necessary to the database.

3.2.12.4 Outputs

Outputs whether update was successfully or not.

3.2.12.5 Error Handling

Throws exception handling if trying to create empty groups or use repeat group titles.

## 3.3 Use Cases

### 3.3.1 Use Case: User Registration

1. The user launches **Quizwave** app
2. The user clicks "Register here" below the login button
3. The app prompts the user to enter the name, email and password
4. The user can now access the app with their details

### 3.3.2 Use Case: User Login

1. The user launches **Quizwave** app
2. The user supply login details
3. The app display users’ profile

### 3.3.3 Use Case: User want to state a test

1. The user launches **Quizwave** app
2. The user clicks "Start a Test"
3. The app prompts the user to choose from a range of courses
4. The user choose a course
5. The app pulls up the test questions!

### 3.3.4 Use Case: User submit test

1. The user click “Submit” after answering the question
2. The app prompt the user to confirm their decision to submit test
3. User made final submission.

### 3.3.5 Use Case: User view performance chart

1. The user click on “Performance Chart” on his dashboard
2. The app prompts the user to choose course the want to check their performance for.
3. The user choose the course
4. The app display a graph of the user results in the course

### 3.3.6 Use Case: User view test history

1. The user click on “Test History” on his dashboard
2. The app display the tests user has taken

## 3.5 Non-Functional Requirements

### 3.5.1 Performance

The main objective of this project is to develop an android app in which student can take test on their mobile phone, thus the main requirement is the users' phone internet connection speed. Stable internet connection will ensure the app load the test questions in time to the app.

### 3.5.2 Security

This app will request access to other app on the user phone only when the user wants to share the result of their test. User data from such apps or their mobile phone will not be stored by the application in any form.

### 3.5.3 Safety

**Quizwave** app will not cause any harm to the users’ phone nor will it affect any other application. However, it is recommended not to use this app while driving or in situations where attention needs to be focused.

## 3.6 Inverse Requirements

This app does not contain lecture notes or videos.

## 3.7 Design Constraints

Internet connections on the users' phone is required to interface between the phone and a database server. This app is developed to only to target the Android OS. Android OS is of different versions and it is important to note that some features (e.g. print to pdf) of the Quizwave app may not be available for users with Android OS below 4.4 (KitKat).

## 3.8 Logical Database Requirements

Information supplied to the Quizwave app and the portal is held in a database. This contains information such as user details, questions (with options and answer), as well as relationships that exist between this information. The following is a logical classification of data entities as well as their attribute that may be stored in the database.

* User: ID, Name, email, password
* Questions: ID, Question, Options, Answer, Date, Time
* Test Result: ID, User Details, Question Details, Score, Remark, Date, Time
* Admin: ID, Username, Password

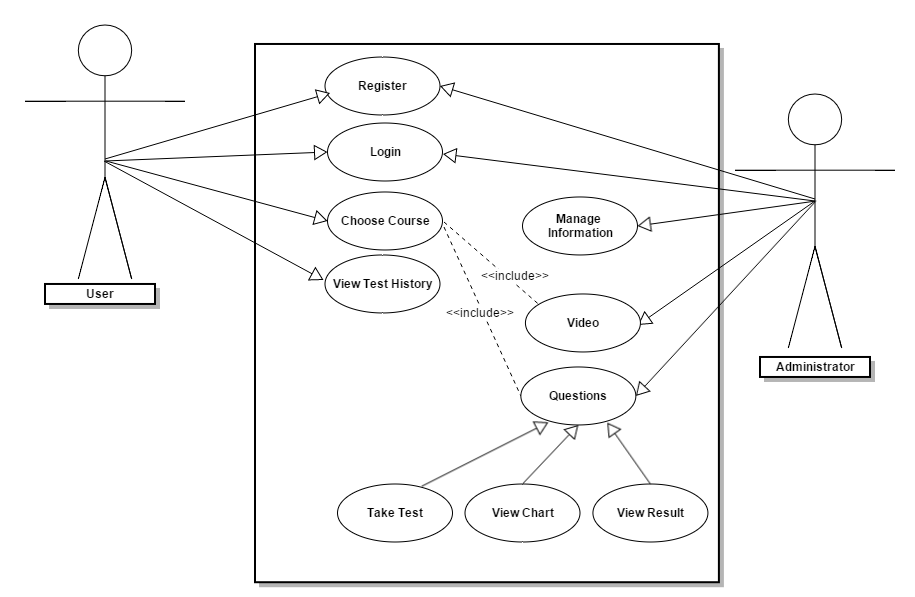
The server will be configured on a Linux platform, and through use of Object-Oriented PHP will allow interaction and processing in conjunction with MySQL database. Processes to be done on the server include: pushing/pulling data, updating data.

# 4. Analysis Models

## 4.1 Sequence Diagrams

Sequence diagram shows the interaction and how the processes operate with one another and in what order. After launching the Quizwave app, the app prompt the user to login. User only login successfully after the login details (username and password) has been found in the database. User can now take test and view their result. Fig 2 below represent the sequence diagram for the **Quizwave** app.

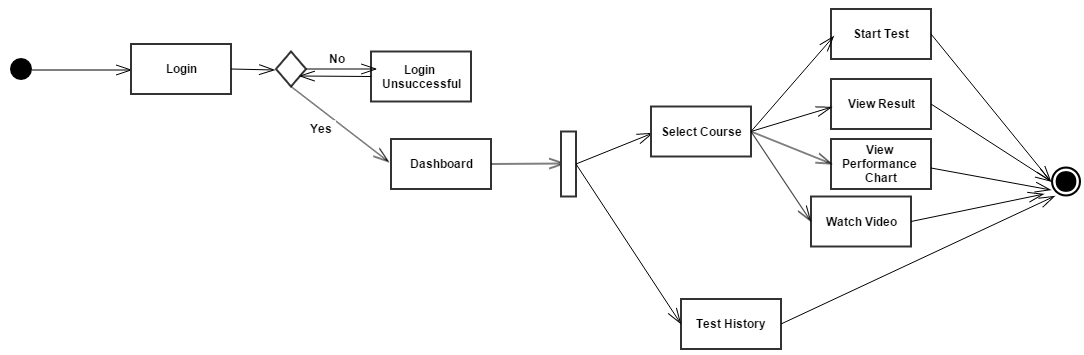
## 4.3 Use Case Diagram



*Fig 3: Use case diagram for the* ***Quizwave*** *app*

*Source: Author*

## 4.2 State-Transition Diagrams (STD)



*Fig 3: State diagram for the* ***Quizwave*** *app*

*Source: Author*

# A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## A.1 Appendix 1

## A.2 Appendix 2