Software Requirements Specification

Project Subject: Online Quiz Website

Dr. Ehsan Alirezaei

Course Title: Software Engineering

Team Members:

Abolfazl Hosseini-9931016

Ashkan Shakiba-9931030

Hananeh Montazer-9931101

Contents

1. Introduction

1.1. Purpose

In this document we are going to present a detailed description of the Online Quiz Website.

1.2. Project scope

In this website students and teachers can sign up and create their own accounts to use site features. Teachers can make questions for each field. Put a short summary for their lessons so students can read it and learn the most important notes.

This website is used for teachers to realize how good their students are in that part of lessons that they've tested.

This website can be used for every teacher and its students.

1.3. Glossary

Term	Definition
Teacher	Who can make question or write short summary for his/her lesson
Student	Who is being tested
Database	Collection of all questions in every field with different types of difficulty
Viewer	Everyone who can see the website
Field	Subjects of lessons
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate.
User	Teacher or Student
Sign up	The process of making new account
Log in	Go through your account
Timer	Each test should be finished in certain time and timer will manage that
Test Result	Result will be shown as percentage and student can review questions and see answers below each question

1.4. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It would explain what we use as an agile model in our process model and its advantages.

2. Overall Description

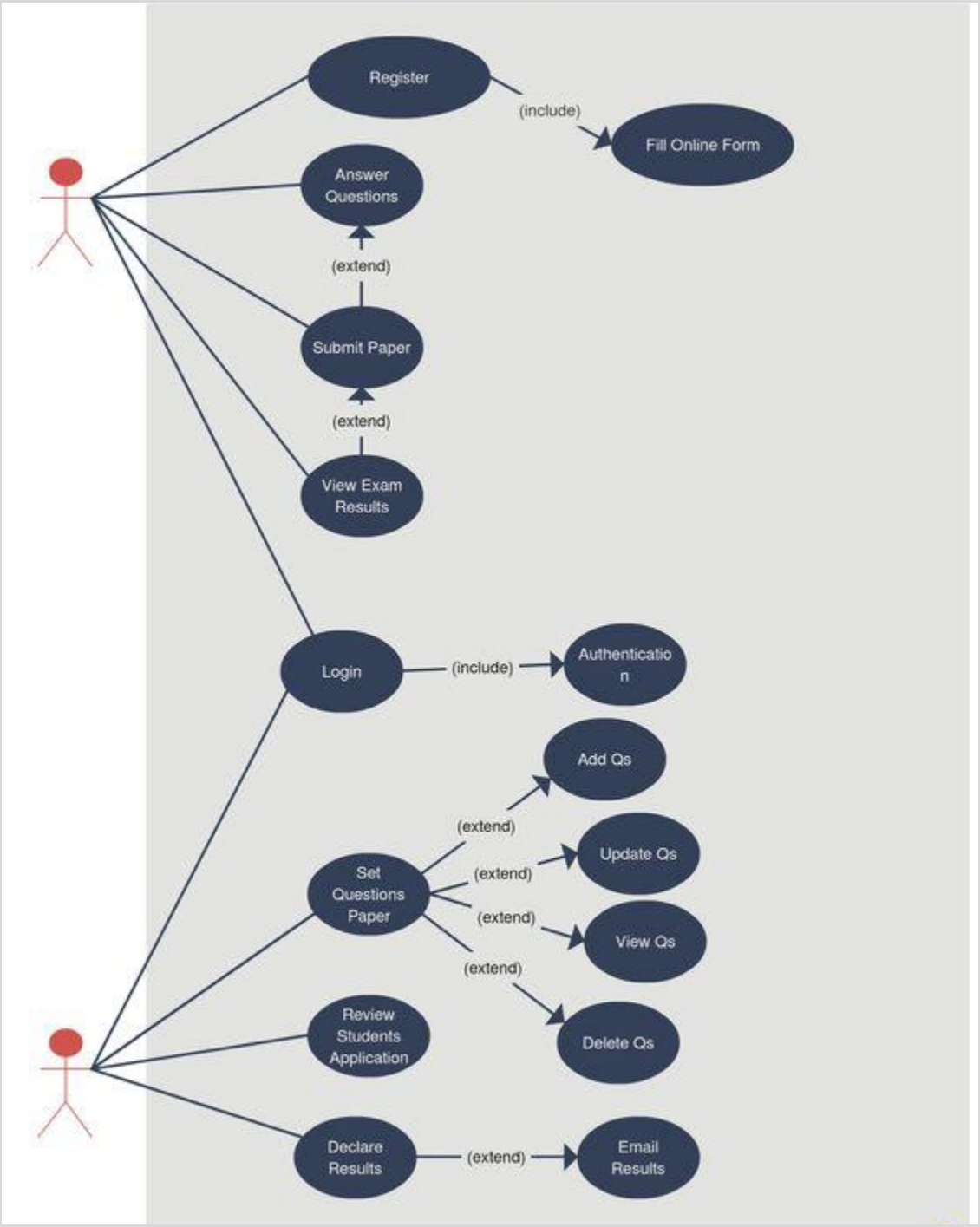
2.1. System Environment

This Online Quiz Website has three active actors and one cooperating system.

Teachers, students and viewers can access the system through the internet. Teachers access the entire system such as questions, answers, results, etc. Students at first access only the questions that have been selected by the teacher and after that access the answers of those questions and her/his test result.

All users such as teachers, students or people who just want to visit the site can access the website using pc, laptop, mobile phone or any device that can connect to the internet.

2.2. System use case



2.3. Process Model

Process model is the mechanism of dividing software development work into distinct phases to improve design, product management, and project management. It is also known as a software development life cycle. We have two general process models: waterfall and agile.

We chose an agile model because now our project is small and it is better to use that model.

Because of following points agile model is better to use for us:

- We can deploy software quicker, so customers can get value sooner.
- Waste fewer resources because working on up-to-date tasks
- Adapt better to change and respond faster
- Faster turnaround times
- Detect and fix issues faster

3. Architecture

3.1. Description

Model-View-Controller pattern, abbreviated as MVC, divides the architecture of interactive applications into three parts, which are:

- Model: contains the main capabilities of the application and the data (in fact, the model is responsible for the task for which the application was developed.)
- View: is responsible for displaying the required data and application interface to the user (in some applications, more than one view may be defined for users.)
- Controller: is responsible for handling user input data and establishing communication between model and view.

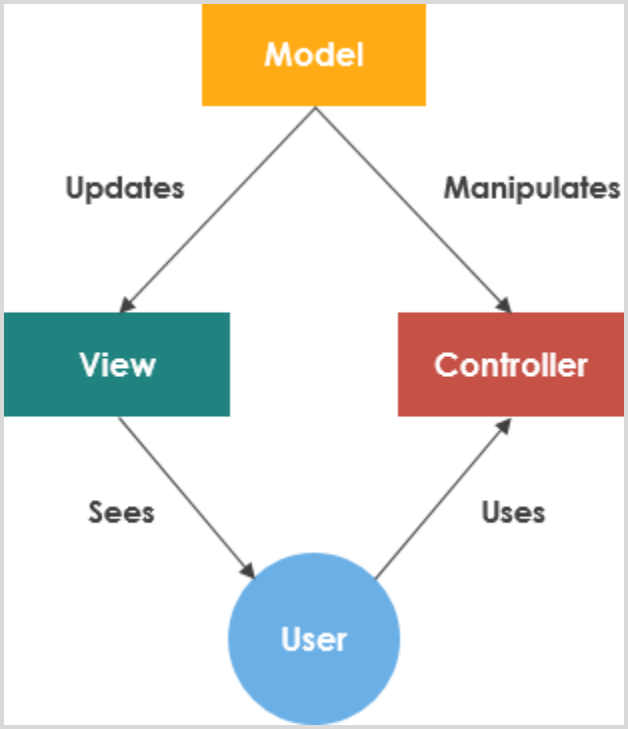
The MVC architectural pattern separates the above components in an application; In other words, this architectural model separates the methods of using data inside the application from the methods of presenting data and receiving them from users, and this problem also reduces the complexity and ease of application development, and provides this opportunity for developers to be able to Reuse application source code in an effective way

- Model: contains the main capabilities of the application and the data (in fact, the model is responsible for the task for which the application was developed.)
- View: is responsible for displaying the required data and application interface to the user (in some applications, more than one view may be defined for users.)

- Controller: is responsible for handling user input data and establishing communication between model and view.

The MVC architectural pattern separates the above components in an application; In other words, this architectural model separates the methods of using data inside the application from the methods of presenting data and receiving them from users, and this problem also reduces the complexity and ease of application development, and provides this opportunity for developers to be able to Reuse application source code in an effective way.

3.2. MVC Diagram



3.3. MVC Use-Case Diagram