

Project Subject: Online Quiz Website

Dr. Ehsan Alirezai

Course Title: Software Engineering

Team Members:

Abolfazl Hosseini-9931016

Ashkan Shakiba-9931030

Hananeh Montazer-9931101

Contents

1. Introduction

1.1. Purpose	4
1.2. Project Scope.....	4
1.3. Glossary (Requirments)	5
1.4. Overview of Document.....	7

2. Overall Description

2.1. System Environment.....	8
2.2. System Use Case.....	9
2.3. System Process Model	10

3. Architecture

3.1. Description.....	12
3.2. MVC Diagram.....	14
3.3. MVC Use-Case Diagram.....	14

4. Diagrams

4.1. Class Diagram.....	15
4.2. Activity Diagram.....	17
4.3. Sequence Diagram.....	19

4.4. Component Diagram.....	21
4.5. Context Diagram.....	23
4.6. Deployment Diagram.....	25

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 (Requirements)

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

Term	Definition
Architecture Pattern	An architectural pattern is a general, reusable solution to a commonly occurring problem in software architecture within a given context.
Designing Exam	To choosing a difficulty of a exam and number of questions and collecting questions from database
UI/UX	UI refers to the screens, buttons, toggles, icons, and other visual elements that you interact with when using a website. UX refers to the entire interaction you have with a product, including how you feel about the interaction

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

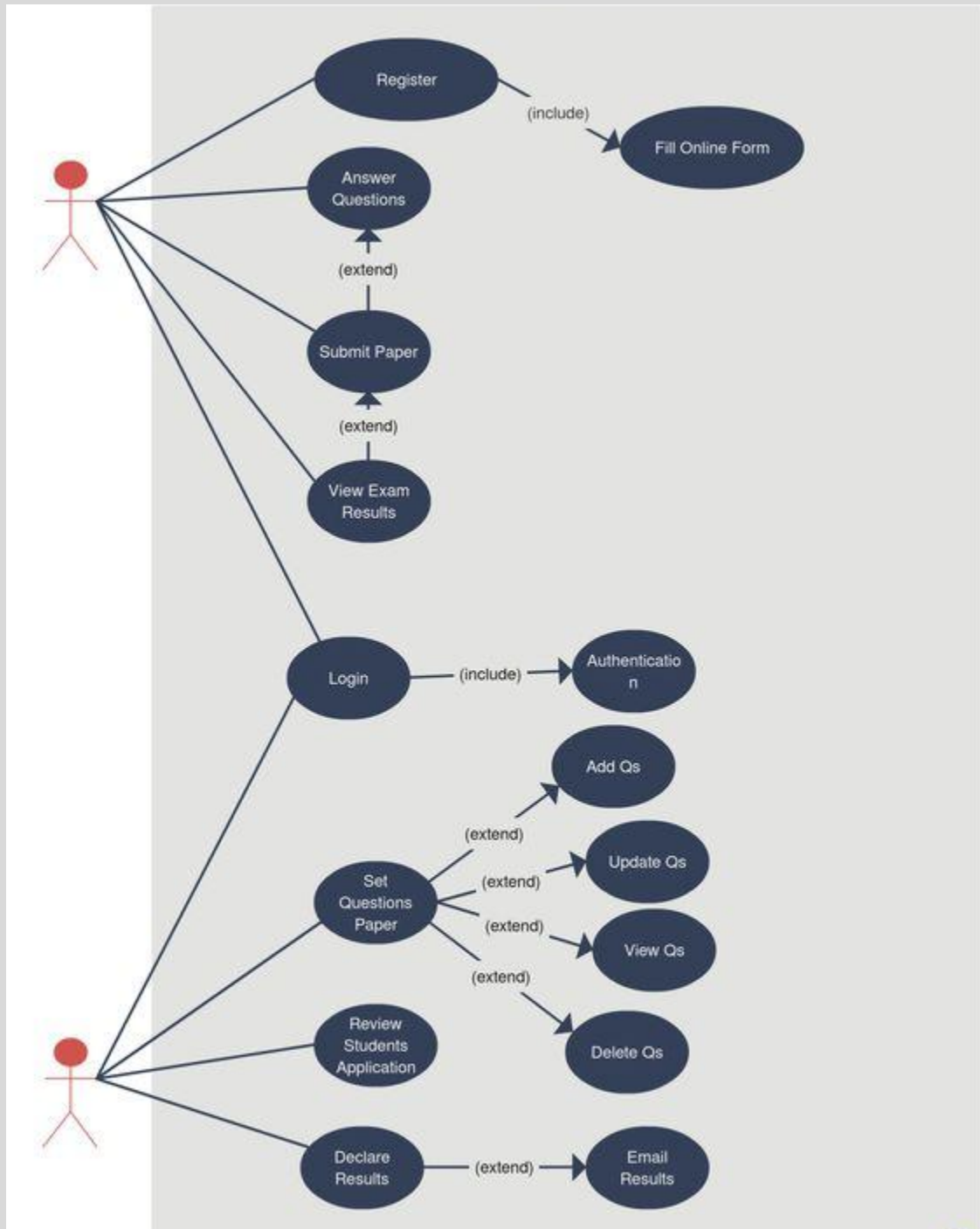
2.1. System Environment

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've used scrum and prototype as our process model.

prototyping is the activity of creating prototypes of software applications, i.e., incomplete versions of the software program being developed. Prototyping has several benefits: the software designer and implementer can get valuable feedback from the users early in the project. The client and the contractor can compare if the software made matches the software specification, according to which the software program is built. It also allows the software engineer some insight into the accuracy of initial project estimates and whether the deadlines and milestones proposed can be successfully met.

Scrum is a framework for project management^[1] commonly used in software development, although it has been used in other fields including research, sales, marketing and advanced technologies.^[2] It is designed for teams of ten or fewer members who break their work into goals that can be completed within time-boxed iterations, called *sprints*. Each sprint is no longer than one month and most commonly lasts two weeks. The scrum team assesses progress in time-boxed daily meetings of 15 minutes or fewer, called daily scrums (a form of stand-up meeting). At the end of the sprint, the team holds two

further meetings: one sprint review intended to demonstrate the work done for stakeholders and solicit feedback, and one sprint retrospective intended to enable the team to reflect and improve..

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

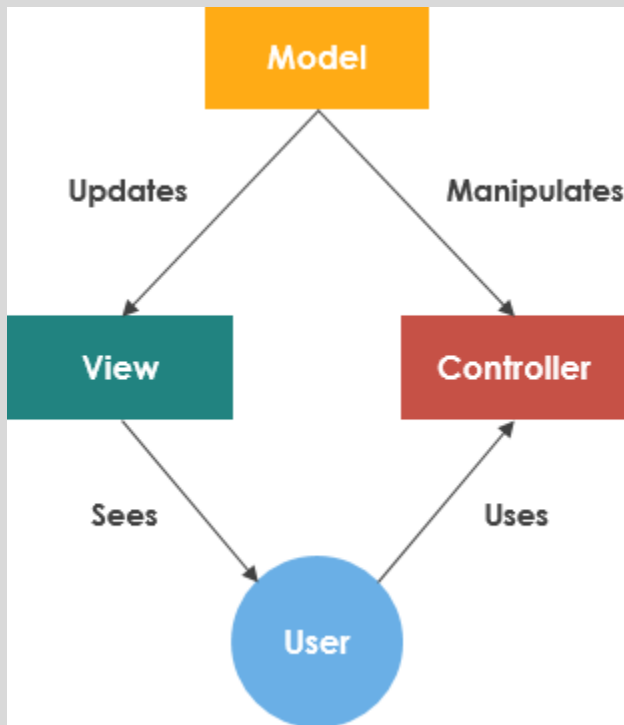
- Model: contains the main capabilities of the application and the data (in fact, the model is responsible for the task for which the application was developed.)**

- 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



4. Diagrams

4.1. Class Diagram

Class Diagram is the blueprint of the system that we are designing. We use it to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide.

We have a teacher part. In this class we save information about teachers like their name, id, courses that he/she teaches, etc. teachers can access every part like questions, quiz, student, class,...

System has an admin that manages the system.

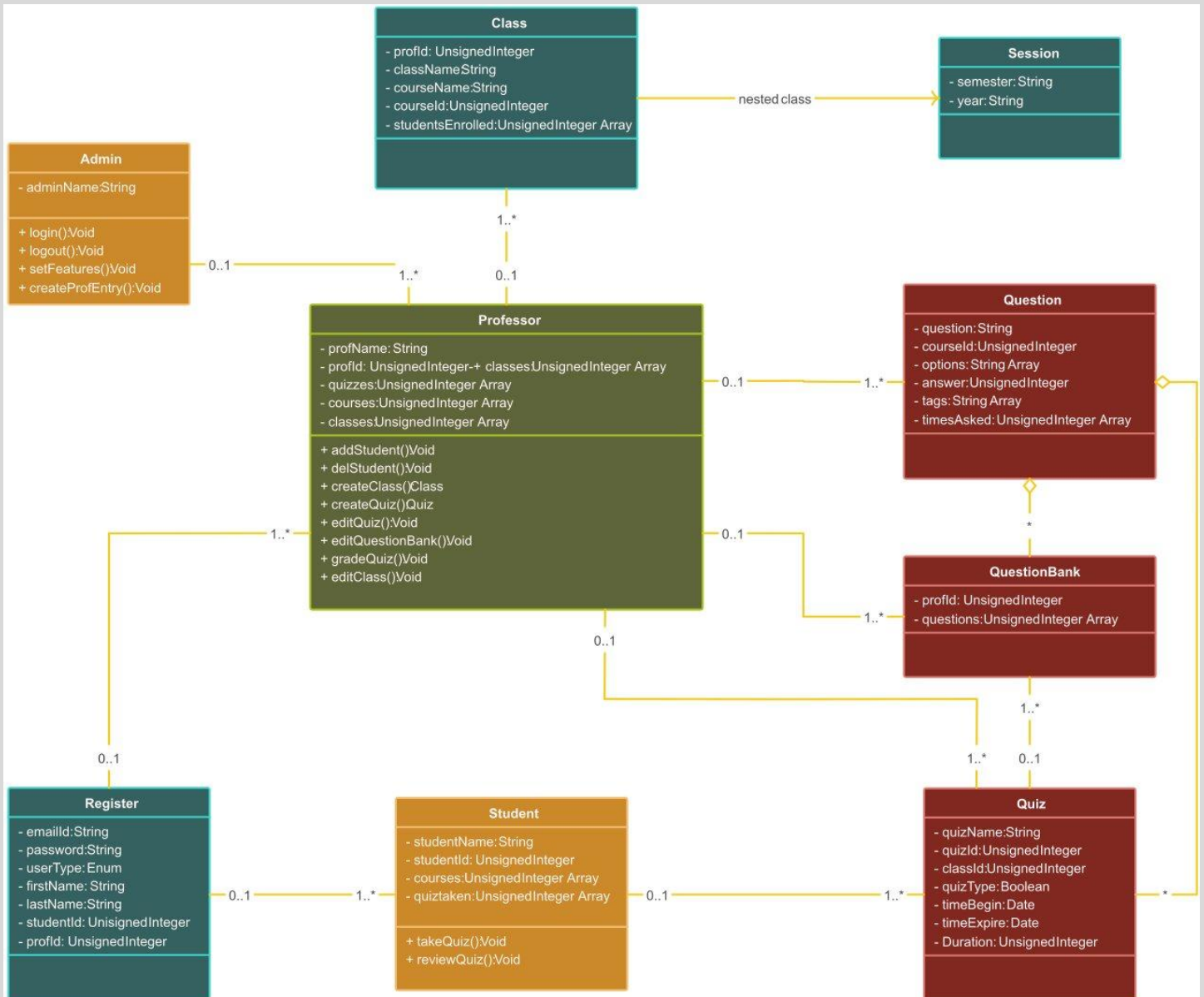
In the register part we save email, password, full name of the registrant and id.

For students we save the courses that they have been examined and the quizzes that have been taken from he/she as well.

Quiz class contains the name of the quiz, its time and its id.

For questions we save the text of each question, the course that it relates to and the answer of each.

Class Diagram



4.2. Activity Diagram

This diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram.

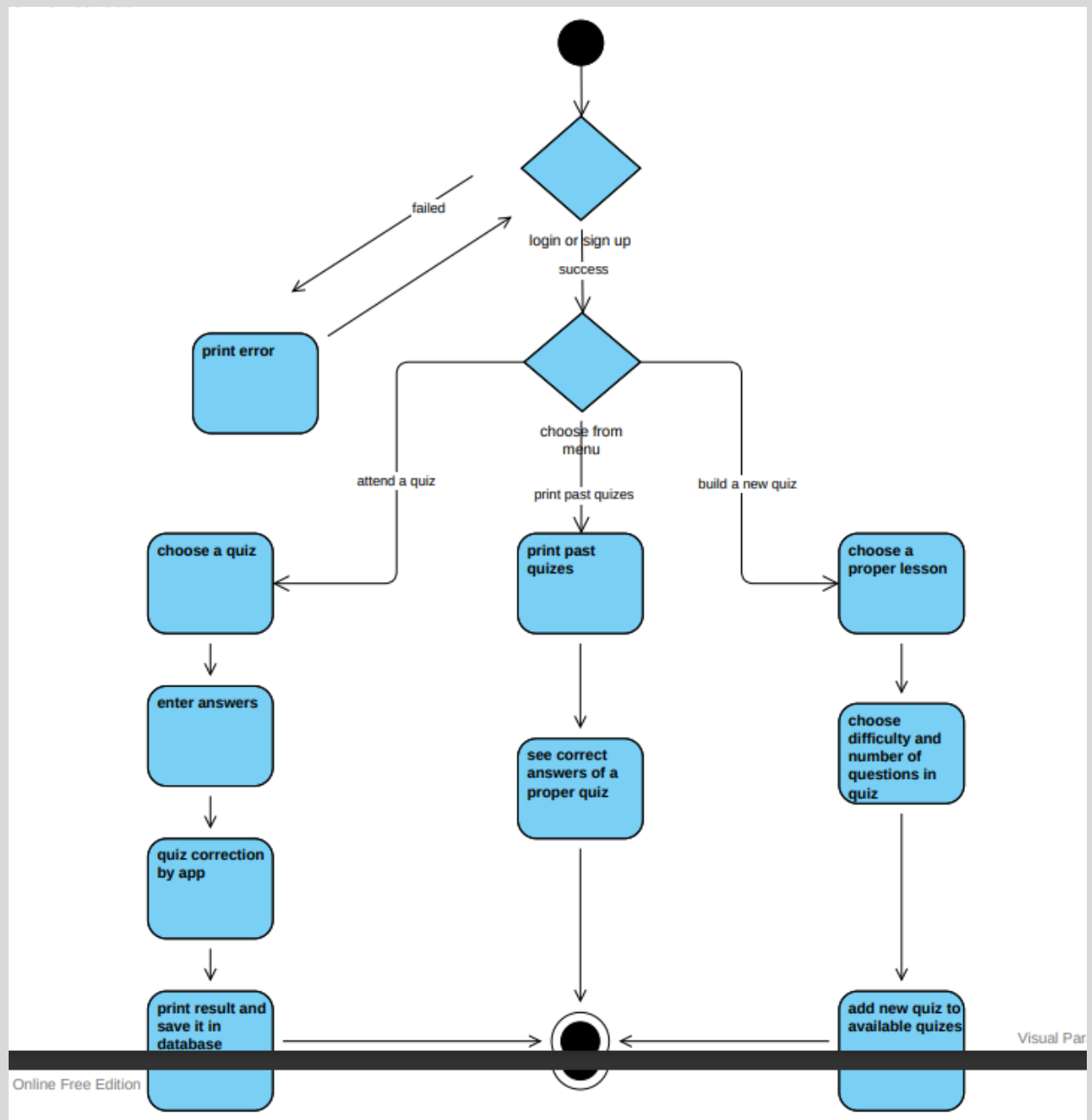
In our system, when a user enters the site, he/she should first sign up or login. Then we have two states. The process can be failed or successful. If it fails it returns to the previous page and he/she should login or sign up again. But if it is successful, the user enters a menu that has three main parts: he/she can participate in a quiz or design a new exam or review past quizzes.

If participating in an exam is chosen, the user should solve the question and enter the answers. Then the quiz will be corrected by the system and it shows the result and saved in the database.

If the user wants to review past exams, he/she can see the correct answers of a chosen quiz.

And at last if somebody wants to design a new quiz, he/she should choose a subject and the difficulty and number of questions in the quiz.

Activity Diagram

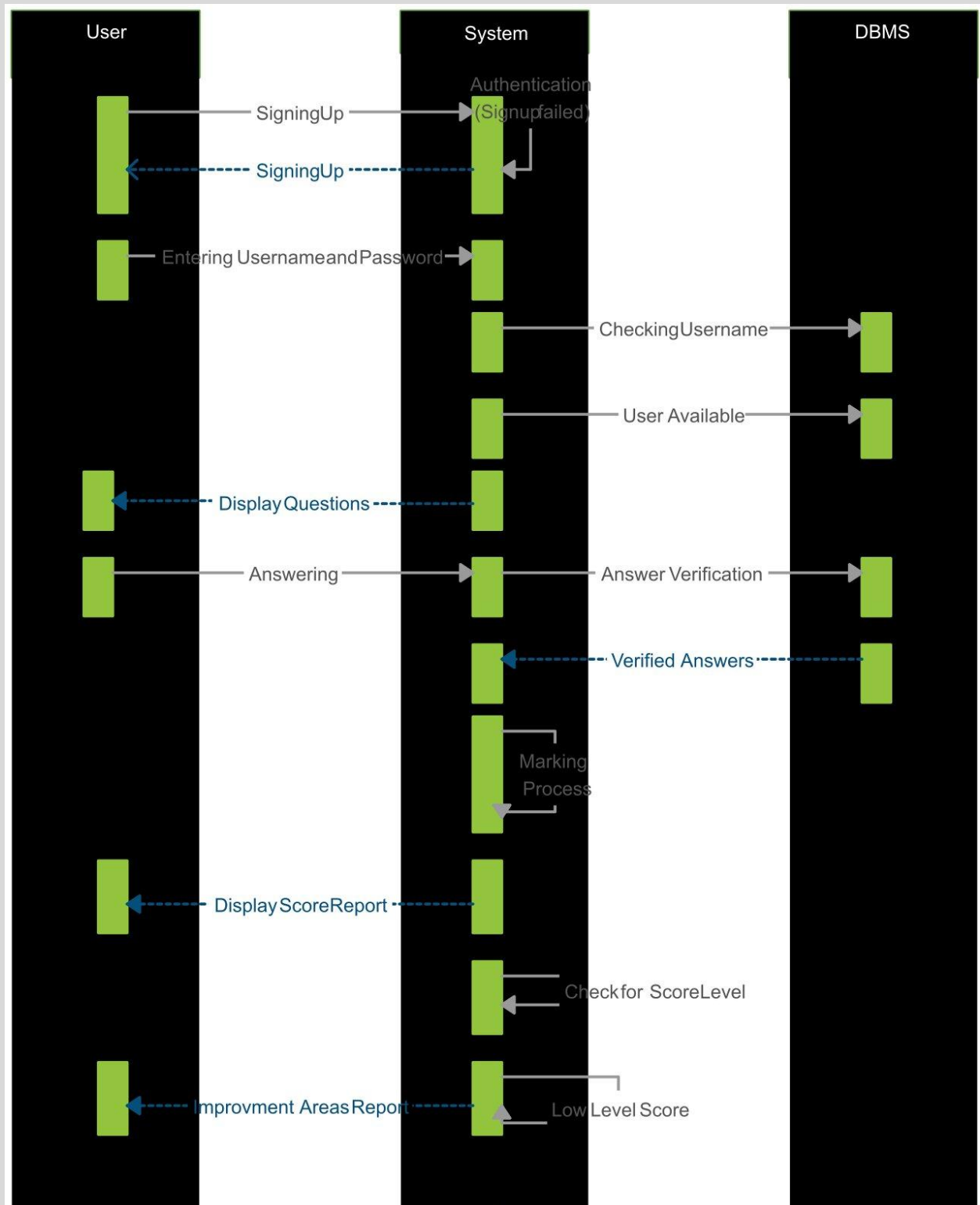


4.3. Sequence Diagram

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction.

At first the user sends a request to the system for registering and the system responds after the authentication process. When registration is done, he/she can choose a specific username and password. System has to check the username with DB and if it's available and no one before chose it, the user can sign up successfully and questions will be shown to him/her. Then the user answers the questions and they will be checked with correct answers that are in DB and then the score of the exam will be calculated and saved in the database.

Sequence Diagram



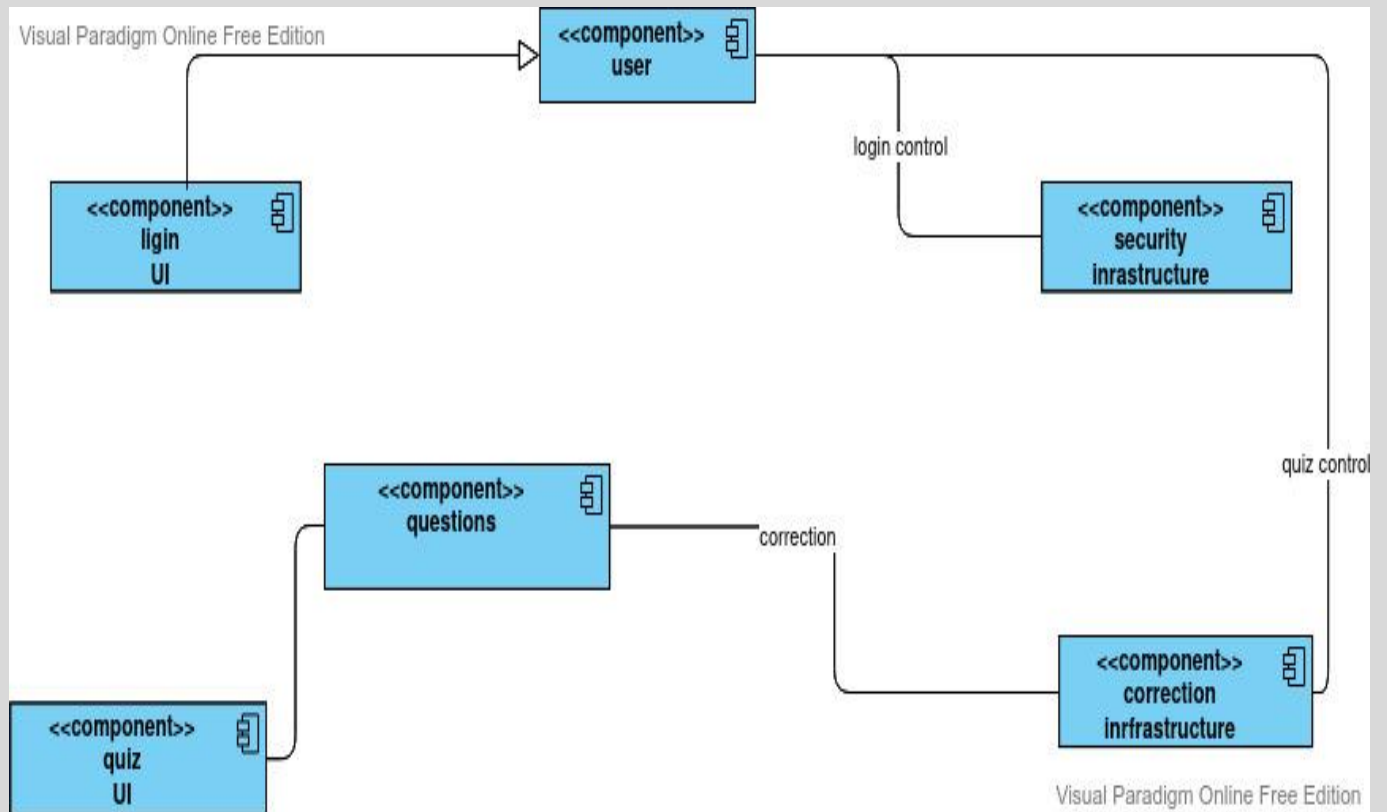
4.4. Component Diagram

Component diagrams are essentially class diagrams that focus on a system's components that are often used to model the static implementation view of a system.

In this diagram the login part is a user interface and through it the user can enter in his/her account. Also security controls the login process and if someone wants to enter your account without permission, it prevents it.

Quiz part is the interface too, and it relates to the question component that contains information about exams and its questions and to checking the quizzes we go to the correction component.

Component Diagram

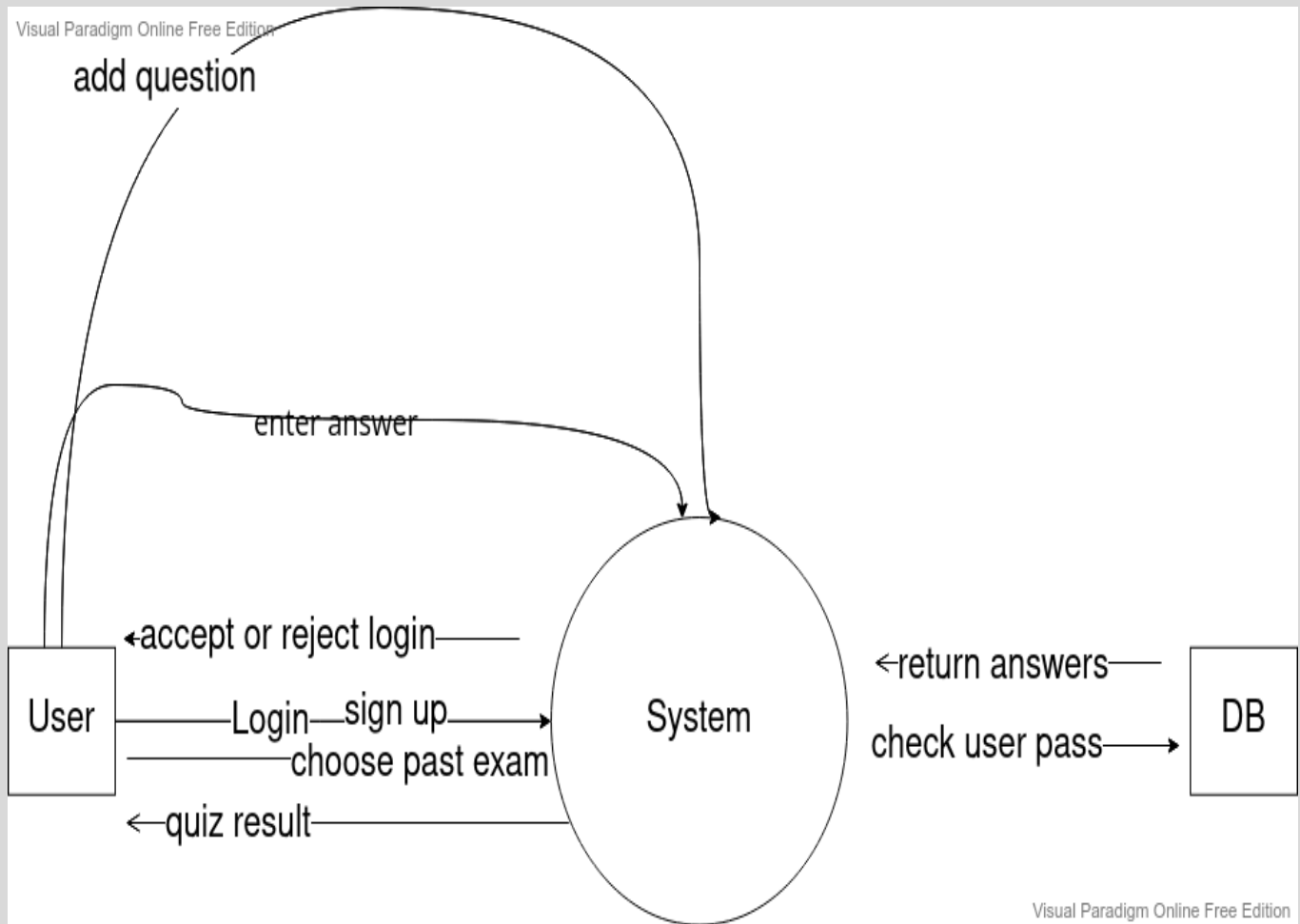


4.5. Context Diagram

A context diagram outlines how external entities interact with an internal software system. It's primarily used to help businesses wrap their heads around the scope of a system. As a result, they can figure out how best to design a new system and its requirements or how to improve an existing system.

If a user wants to sign up or login, he/she should send a request to the system. After that system sends DB a request to check the username and password. DB sends answers to the system about correctness or wrong. Then the system will send accept or reject answers back to the user. If a user wants to review past exams, add questions or enter answers of an exam in progress, he/she should send a request for each process. Then if it is needed, the system sends a request for DB and after receiving DB answer, it will send it to the user.

Context Diagram



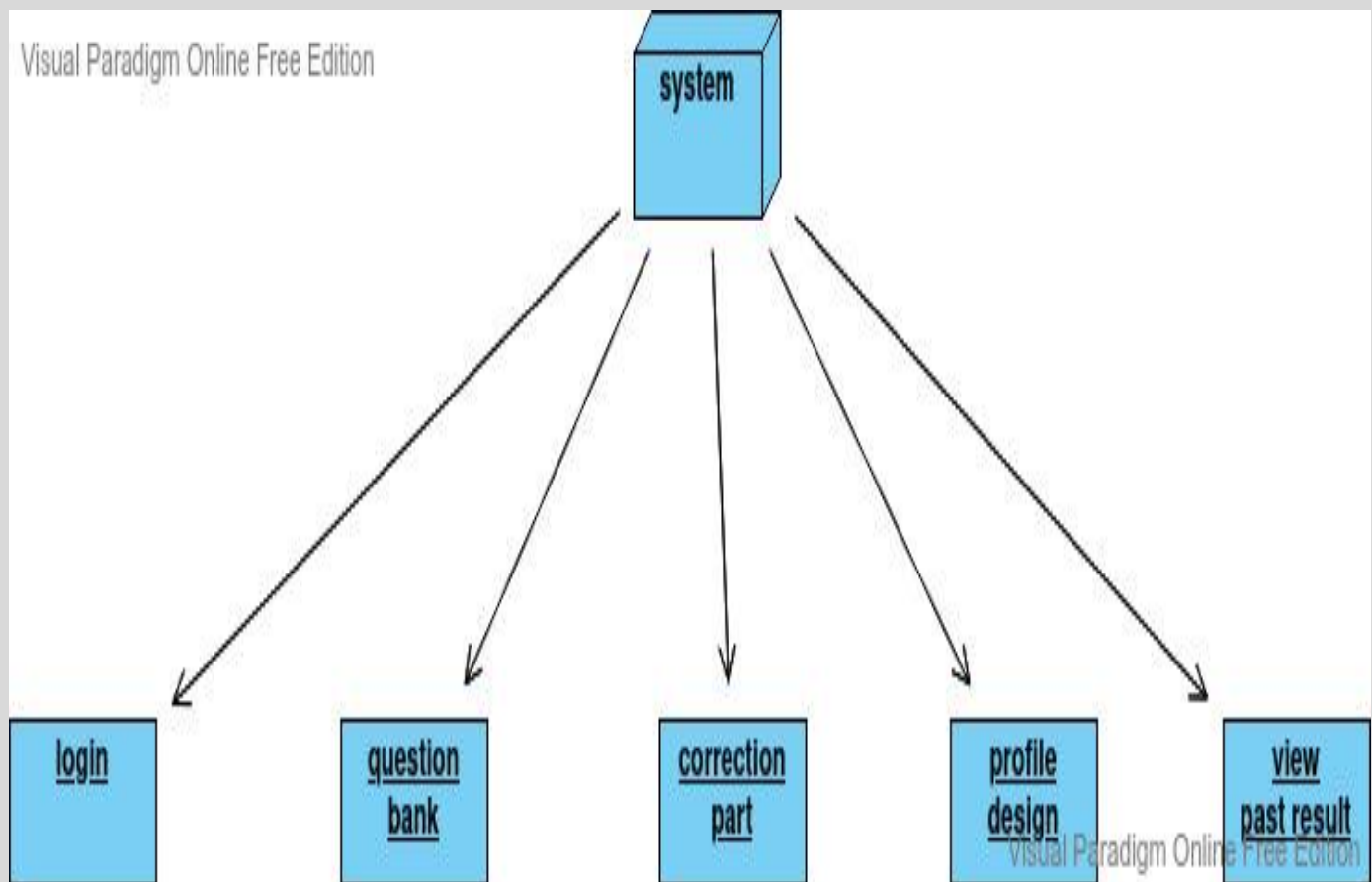
4.6. Deployment Diagram

Deployment diagrams model the physical architecture of a system. They show the relationships between the software and hardware components in the system and the physical distribution of the processing.

Our system has five main parts such as login, question bank, correction part, profile design and view past results.

We have explained each part in the last diagrams.

Deployment Diagram



(The system use-case and the requirements have been written in previous pages.)

In the end, we would like to thank our dear teacher who accompanied us with great patience in this difficult semester and we were able to finish this project.