Adama Science And Technology University

Bsc project Documentation

Title: Web Based Virtual Tutorial System

Proposed by: *Getasew Walelign* *a/ur13942/10*

Getasew Tilahun *a/ur14022/10*

Yonatan Demlie *a/ur13944/10*

February 2, 2021

ASTU, Adama, Ethiopia

Table of Contents

Index of Tables

Table of Figures

Chapter 1

1.1 Introduction

In 2006, the Open University of the Netherlands (OUNL) started running trials with a so called virtual classroom. A virtual classroom is a combined set of tools for conducting classroom like sessions live over the Internet. Since 2006,more than 20 pilot projects have been carried out, involving more than 25 tutor sand over 200 students. We intended to use the virtual classroom as a replacement for face-to-face tutoring sessions. We investigated several virtual classroom instruments in order to come to a first selection of promising candidates. Our findings indicate that the virtual classroom is a valuable addition tour educational offerings. In this paper we will describe what a virtual classroom is why we choose to implement it and how we came to our selection of an instrument. A Virtual classroom is a web-based education system for delivering lecture in real time. This systems include communication tools, schedules and also it has rating and reviews. This modern type of learning provides a set of different free skills. It also provides a variety of learning options for its learners, without institutional curricula.

1.2 Background of the project

For a long time teaching and learning process was face to face. Short courses and tutorials are also given face to face. This was the only solution for the past years. But now things are changing, technology changes our life. So we have to use emerging technologies in our day to day activities to make our life better. Since we are students we know the hassle of learning language, getting academic tutor, learning coding and other short courses. If someone wants to take such type of courses either he should live in regional cities or other better towns. Besides the issue of availability there is no convenient time to take such courses. From the time we see this problem we decided to solve this problem by using latest technologies. From that on we are working on this project. And our platform let customers to improve their knowledge on their free time through real time virtual classes. Teachers and those who have knowledge on some field can register on our platform. Our project let the students to take quiz. Most people are busy with their works and they lack courage to learn face to face courses. Our platform let them to learn any courses from the comfort of their home. They can schedule a class at night or anytime they are free and take a course they want by video and text chat. Both teachers and Students can share their screen during the class. Besides this the system will have a white board. This

increases the quality of the learning process. From our survey we understand that there is an on demand need of online courses and the other thing we differ from other platforms is individual learning and RTE(Real Time Engagement).

1.3 Statement of the problem

Tutors, language schools and short course providers give their service to their students in undigitized manner. There are a lot of problems related to undigitized services in the above sectors. Here is the list of the problem we found in existing system:

- Time is not suitable for workers to learn additional skills like language skills.
- Transportation is another problem for tutors to go to students home.
- Tutors pay for brokers to connect with students.
- Female students afraid to have male tutors since this may lead to harassment.
- Language and code schools are not accessible in all parts of Ethiopia.
- Language schools and other short course providers reduce the size of students due to lake of space.

1.4 Justification of the project

The purpose of this project is to make the process to tutoring and short courses available to all parts of Ethiopia. The old system of tutoring is face to face it is not available in every part of Ethiopia. So there should be a platform which lets everybody to take courses and get tutoring service in anytime and from everywhere. Our proposed system let the users to get courses and tutor in real time. There are to type of choices for the users of our platform, individual learning for those who want to take a course individually and group learning for those who want to learn with a group. This system allow the users to get additional knowledge from the comfort of their home. Our platform bring solution for women's since they are busy with household work during the day time, they can take courses during their free and safe time by making a comfortable schedule with the teachers.

1.5 Objectives of the project

1.5.1 General objectives

The general objective of the project is to develop a web based real-time virtual tutorial system for students and teachers.

1.5.2 Specific objectives

Specific objective of the project is:

- Study the existing system
- Gathering and Analyzing the requirements of the proposed system
- Design the user interface and the database of the virtual tutorial classroom
- Implement the system
- Test and deploy the system

1.6 Scope and limitation

1.6.1 Scope of the study

The system allow teachers to give tutorial remotely, give quiz and remote communication via text, audio and video chat with file sharing facility. The system will have online payment method.

The proposed system includes the following features.

- Audio and video conferencing
- Text chat
- Whiteboard for both teachers and students
- Create quiz, conduct and check result as well
- Online payment
- Rate and review teachers
- System Configuration

1.6.2 Limitation of the project

- Cheating prevention during online quiz is very complicated.
- It can’t address all types of courses. Courses which needs laboratory and workshops are unlikely in this platform.
- Due to time constraint our platform don’t support multiple languages.

1.7 Feasibility study

We can analyze the proposed system by taking the three feasibility test whether the system is Economically, Operational or Technically assured or not. From the perspective of those three core points a decision is taken whether to continue or to cancel the project.

1.7.1 Technical feasibility

This project is a web based system. The main technologies that we are going to use is MERN stack. Those technologies are open source and easily available to use and the technical skills can met by our team members. So that our platform is technically feasible.

1.7.2 Operational feasibility

From an operational point of view, the feasibility of this project is high because of the following reasons.

- The technology that required to use this service is easily learnable and user friendly to interact with it.
- The user can use the system anywhere and anytime by using their android devices, tablets and laptop computers by connecting with internet connection.

1.7.3 Economic feasibility

From the perspective of economical feasibility, the feasibility of this project is high and the system is reasonable worthy by current estimates. because of the following reasons.

- The students want to use this system will be charged reasonable price which is lower than they pay for Tutors before this system. This shows everyone can use the service without having to pay additional price than they did before.
- When Users use this system the service will be available at home virtually so this will make them not to waste unnecessary money for transport.

Based on those above three types of feasibility study techniques, We've got developing this system is important and we decided to move forward.

1.8 Significance of the project

The significance of the project is vital, here are some points why we say it is significant:

- A virtual tutorial class provides flexibility in setting study times.
- Provides continuous education, and the ability to study anywhere in the world without being bound by geographical borders.

- Provides the solution for educating students in remote areas.
- Reduces time and costs and provides greater freedom to control your study and work times.
- Provides the possibility of studying while continuing to work.
- It has the possibility of studying and training at any time, as it is an open center around the clock.
- Provides university students with the ability to study many different specializations that they may not find in traditional universities.
- Increases the efficiency of the education and training process.

1.9 Beneficiaries of the project

Beneficiaries of the project are listed below:-

- Students
- Tutors and Teachers
- Language and code schools

1.10 Methodology

Fact Finding Methodologies

Data collection methodologies are methods used to collect different data from different data source (documents,users and organizations etc.)
The following are the data collection methods used for requirement elicitation:-

Primary data source:-
Interview:-we used interview as one of the major data collection method. During interview our team got different necessary information from the department head.
Document analysis:-we have analyzed different documents of the project from our Advisor.
Direct observation:-as part of the learning process and as a student we have gone through the scheduling process.

Secondary data source:-
Internet:-internet aids us to see the available sample on the internet and to download different types of tutorials which help us in doing the project.

Software Development Methodology

For this project we will going to use agile methodology. Agile methodology was developed as a response to growing frustrations with Waterfall and other highly structured, inflexible methodologies.

This approach is designed to accommodate change and the need to produce software faster. Agile values individuals and their relationships and interactions over tools; it features customer collaboration throughout the development process; it responds to change instead of following a set-in-stone plan; and it focuses on presenting working software, rather than documentation.

1.11 Development tools

VS Code: is an open-source code editing program built by Microsoft. It runs everywhere and allows you to do anything from debugging code to inputting Git commands or creating Sass code.

Node.js: is an open-source, cross-platform, back-end JavaScript run time environment.

Express.js: or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License.

MongoDB: is a source-available cross-platform document-oriented database program.

React: React is a free and open-source front-end JavaScript library for building user interfaces based on UI components.

GitHub: is a provider of Internet hosting for software development and version control using Git.

Git: Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development.

1.12 Required resources with cost

Software Resources

Resource	Number of Materials	Price in Birr	Total Price in Birr
Window 10 pro OS	1	Free	
Ubuntu OS	2	Free	
VS code	3	Free	
MongoDB	1	Free	
Microsoft Office	1	Free	
LibreOffice	2	Free	
Payment Gateway	1	Not known	

Table 1 software resources

Hardware

Resource	Number of Materials	Price in Birr	Total Price in Birr
Laptop	3	25000	75000
mouse	3	150	450
Internet cable	3	100	300
Print	200	1	200
Pen	3	12	36
Paper	1 packet	200	200
USB Disk/flash/	2	350	700

Table 2 hardware resources

1.13 Task schedule

Phases	Timeline
Chapter 1	November 12 - December 1
Chapter 2: Description of existing system/ Literature Review	December 2- December 13
Chapter 3: Proposed System	December 14 – December 22
Chapter 4 System Design	December 23 – January 5
Chapter 5: Prototype and Testing Plan	January 6 -January 12
Final Documentation	**January 12 -January 27**

Table 3 task schedule

1.14 Team composition

no	Name	id	email	responsibility
1	Getasew Walelign	A/ur13942/10	getas.walelign@gmail.com	Requirement gathering, analysis, Implementation, Testing
2	Getasew Tilahun	A/ur14022/10	getasewbekahegn@gmail.com	Coordinating, requirement gathering, designing,Implementation.
3	Yonatan Demlie	A/ur13944/10	Yonatan.dem123@gmail.com	Requirement gathering, implementing, Testing.

Table 4 team composition

CHAPTER 2

2. Description of the existing system

Currently tutors and short course providers give their service face to face with their students. Those who want a tutor for their children should connect with a broker to get the tutor. There is also a platform to connect tutors with those who want a tutor. And tutors should go to the students home to give the service. When we come to language school and short course providers their service is limited to some specific towns. Some short course providers register students via call and social medias and some have also online system to register the students but the rest are using manual method till now. Those registered members should go to the location of the service provider to take the courses. Most people are not willing to take such type of courses this is due to the problem of transportation and comfortable time. So it is inevitable that their must be a platform to solve such type of problems. The process of digitization makes it easy to solve the problem.

2.1 Major functions of the existing system

The major function of the existing system is to connect tutors with students,register students for a short course and give face to face courses and tutorials.

Connect tutors with students: the current system connects tutors with students. There are two types of ways to connect tutors with students. The first and the popular one is that tutors search those students by preparing advertising and by plastering the advertisement paper or postcard different physical Advertisement place. The second and less popular way is through a website where tutors are registered and students can deal with them. The tutor gives the service face to face.

Register Students: language schools and short course providers register their students via call and social medias and some of them have their own online registration platforms. But most of language schools still register their students manually by using paper.

Give Tutorial and short course: Tutors,Language schools, Code schools and short course providers give face to face tutorial and course to their students. Either students goes to the location where this type services are given or tutors goes to their home.

2.2 Users of the current system

Home Tutor

A person who wants to give a tutorial finds a student either by using traditional way or by applying online as tutor. Then the tutor deal directly with the people who needs tutorial or he can deal with them by any means either social media or phone call. Then if they agreed, he starts doing his tutor or he will wait until a new student he meet.

Language School

under the this user system there are two actors which are the student and instructor. The student register for language class that what they prefer to learn and they pays for it. After that the instructor can teach those students who full fill the criteria of language class by meeting class. If the student or instructors are not granted as users of a system they will denied to get or give the service respectively.

Code School

System user of this system are the students who applies for coding and programming language class and the instructors who gives the learning teaching process. After the students registered for the service, instructors give formal and practical class as long as they paid for this service they will continue getting served up to achieving completion certificate.

Short Course Providers

This type of user of current system is when a person want to have short time course. The course provider prepares the type and level of package then the user who wants to have that skill of certificate will registered. If the user is eligible for this service they can have it by going to the place where the class or tutorial held on. Else they will be denied to join if they are not eligible.

2.3 Drawback of the current system

The drawback of the current system is not using the advantage of digital world. Besides that services listed above are given in distributed way, if the services related to each other become centralized things will be easier. Additional drawbacks of the current system are listed below:-

- Lack of transportation when tutors move from their home to give face to face service.
- Lack of transportation when students move from their home to take language and short courses.
- Time is not suitable for most students who took language and short courses. Due to this they forced to stop the course.
- It is not accessible everywhere, Such type of services are available only in regional towns.

- Lack of technological advancement in their services.
- Students should connect with brokers to get their tutors.

2.4 Business rule

The document is prepared according to the existed traditional tutor giving and the other one which works to connect the Tutors with students who need this offer. As we tried to collect how the business rule workflow of existing system, we have written down by dividing the two types of service giving system.

Traditional Tutor and Short Course Service

This is the first and traditional way of service providing method which works without need of any system just by following the manual face to face learning and teaching system. According to this type of tutoring service there will be two types of actors which are the student who wants to learn with help and the other one is the tutors(instructor) who wants to give tutorial and get paid. The tutor search those students by preparing advertising and by plastering the advertisement paper or postcard different physical Advertisement place. And then the student who saw this advertisement will contact him if they are looking for instructor. In this way there Is no official way to know the qualification level of instructor just they make a deal and will start giving tutor if they are agreed on payment. And the tutor will get paid at the end of the month since the agreement is usually per month. There is no legal contract between the one who gives the service and the student who get served.

Web based Connect Tutors with Students

This is the second type of tutor system is Haletatutors which works as agent of the students and Tutors by letting the tutor to sign up and wait until the right person get on. This system nothing works more better for this filed just what it does is try to find the right tutor by testing his knowledge about the course that he/she is going to teach and give the customer to start teaching after he passed the requirement. Even though this system is doing better than the former one it doesn’t still change the way of how class is going on.

Code school, language school and Short Course Tutor

This is the third type of system that is working for language schools, Code school and short course providers. On all the three system registration is using office registrar and some have online registration

systems. After the student applied for short time course, code school or language study they pay for course. Coding class and language classes may take long seasons relative to short course package.

Language school after they managed the class schedule, they gives teacher student communication by letting the students to have freedom to speech write and listen language accents by creating different opportunities.

Code school provides better and safe learning and coding environment as well as fast internet connection and latest computers with good performance capacity. By creating team problem solving and coding opportunity they let their students better skilled.

Short course is the third type of school which lets the customer to have what they need within limited time. Since the course is short it doesn’t take that much more resource and time. It lets to fix any type of person’s challenge by giving the course at right time and right place. The businesses rule will be summarized by the following table.

Type	Service	Business Guide	Time
1	• Traditional Tutors	• The student or his family talks which the Tutor in face to face. • Choose his free time for be learn with Tutor. • The class goes by the Tutor going to student's house	• 2:00- 3:00 hr per class • 2 – 3 days per week
2	Web Based Tutor to Student agent(Haleta)	• The student should have registered successfully. • The student should the subject which they need help. • The Tutor Register on the website with his profession. • The system let the Tutor to take online exam. • Haleta let's talented Tutor meet students and class will held on face to face as traditional class. • The user recommended to have good internet connection.	• 1:00 - 3:00 hr per class • 2 – 3 days per week
3	Short Course Code Schools Language Schools	The student get registered using its available way they make payment for course the Tutor gives tutorial for those who paid. The class held on face to face where Tutors choose. Student will receive completion certificate for their skill at the end of the package.	It depends on the type of course. 2 days per week 3 hours per class

Table 5 business rules

CHAPTER 3

3. Proposed System

3.1 Overview

The proposed system is designed to avoid the hassle in traditional way of tutoring and other face to face schools like language schools. The system let students to search their tutors and attend their class in virtual rooms. It also let language and other school owners to create their virtual room to teach their students virtually. The other feature this system consists of is that payment method that allows students to make payment easily.

3.2 Functional requirement

Login : It is where users of the system register, login, logout, create profile, forget password when they come to the system.

Rate and Review Tutor or Company: Students are rated for tutors and companies. and also they evaluate teachers and companies.

Report: It is a way students report. when they complain about teachers and companies.

Text chat: It is a way Students with tutors as well as companies communicate each other before and during class.

Join session: It is a way during class, students and teachers talk in audio and video about making the class more efficient and easy.

White board: It is a board where teachers write and explain during class time. It is also used by students.

Screen share: It is a way for teachers to show files on his computer to students when they are teaching.

Schedule class: It is where students make appointments with tutors and companies.

Create quiz: It is used by tutors and company to evaluate students. students take the quiz and also see their result.

Payment: It is a way for students to pay for the services they get.

3.3 Non-functional requirement

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system rather than specific behaviors and which has no essential for the system, but it can support and give more quality for the system. Those requirements are listed below.

Usability: the system is user friendly. The new system provides web application user interfaces that are compatible with any browsers. Besides this the system is responsive ,it works perfectly in mobile and tablet devices. The system shall provide the easy access ,The system should is easy to understand, Unauthorized person should not use the system; rather just view the main page and No one can change the password without log in to the system.

Performance: There are no special constraints on the performance. This requirement measures the efficiency, speed, and throughput of the system. The system will make any response on the system within less than 10 seconds depending on the internet/machine speed.

Security: the system must confidentially be controlled from being accessed by Unauthorized users (i.e. each user will have their own roles, the user will Authenticated by his email and password)

Simplicity: the activities should be simple and consistent. The activities will be directed through simple links and simple dialogues. The system will also provide some help through error messages, remarks, and documentations.

Portability: the system should be portable thus capable of running on different platforms, adaptable with other systems, install able on different machines architectures, and replaceable if the need arises.

Accessibility: the system should be accessible at any time since it is needed in every time. It is web based system, therefore internet connection and platform is required to be accessible.

Maintainability: creating new system that can be adjusted easily for adopting new technologies or to fix any defect is called maintainability. The system will be in a way that can handle new or future improvements easily. The database can be exported (as a backup copy) or imported to the system. The system helps the users in reducing usage errors by providing some hints to the users. Moreover, validating the input data before submission make the system more immune against faults.

Error handling: The system should be able to handle the error when the user of the system operates the system.

3.4 System model

System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system. It is about representing a system using some kind of graphical notation, which is now almost always based on notations in the Unified Modeling Language (UML). Models help the analyst to understand the functionality of the system; they are used to communicate with customers.

3.4.1 Scenario

Since a scenario is a tool that is used during requirement analysis to describe a specific use of the proposed system. Using this tool, we will illustrate some interaction of our proposed system.

Scenario 1: Sign up

Participating actors: Students,Tutors and Companies

Flow event:

- Users open the system in their browser.
- The user clicks register button in the navigation bar.
- System displays register form for the user.
- The user enters valid value to the form.
- Click register button.
- The system displays successful message.

Exceptional flow:- if the user doesn’t enter a valid information in the form. The system displays error message.

Scenario 2: Login

Participating actors: Admin,Students,Tutors and Companies

Flow event:

- Users open the system in their browser.
- The user clicks login button in the navigation bar.
- System displays login form for the user.
- The user enters email and password to the login form.
- User clicks login button.
- The system displays login success message and redirect the user to user dashboard.

Exceptional flow:- if the user submit wrong email or password the system displays an error.

Scenario 3: Create Profile

Participating actors: Tutors and Companies

Flow event:

a) The user browse the system and login to the system.
b) Click profile button from the user dashboard.
c) System displays a form for the user.
d) User fills the required information and click submit button.
e) System displays success message to the user.

Exceptional condition:- if the user enters invalid data to the form, the system displays an error message.

Scenario 4: Create room

Participating actors: Tutors and Companies

Flow event:

- The user browse the system and login to the system.
- User dashboard displayed to the user.
- Select create room option from the user dashboard.
- The system creates a room and send a unique room id for the user.

Scenario 5: Join session

Participating actors: Students

Flow event:

- The student browse the system and login to the system.
- Student click the room URL which is send from the teacher or tutor.
- Student pay for the class and the system redirect to the next step.
- The system displays a virtual room to let users to talk in audio, video and text chat.

Exceptional flow:- if the student try to join the wrong session, the system displays a wrong room URL error.

Scenario 6: Screen share

Participating actors: Tutor and Company

Flow event:

The user browse the system and login to the system.

User dashboard displayed to the user.

Select create room option or join an existing room from the user dashboard.

The system displays a virtual room.

From this virtual room tutors and companies can select share screen button to share their screens to the students.

This shared screen will be visible for the students.

Scenario 7: Rating

Participating actors: Students

Flow event:

- The student browse the system and login to the system.
- Student search a tutor or company.
- Student clicks the selected tutor or company.
- From the tutor or company profile click rate button.
- The system displays a popup window for the student.
- The student give rating to the tutor or the company and submit the rating.
- The system remove the popup window after submission of the rating.

Scenario 8: Review

Participating actors: Students

Flow event:

- The student browse the system and login to the system.
- Student search a tutor or company.
- Student clicks the selected tutor or company.
- From the tutor or company profile click review button.
- The system displays a review form for the student.

- The student fills the review form and click submit button.
- The system displays review successful message.

Scenario 9: Create quiz

Participating actors: Tutor and Companies

Flow event:

- Tutor or Company browse the system and login to the system.
- The system displays the user dashboard.
- From the dashboard the user select create quiz option.
- The system displays a form to create the quiz.
- User enters valid values on the form and click submit button.
- The system displays success message.

Exceptional condition:- if the user inters invalid data to the form it displays error message to the user.

Scenario 10: Take quiz

Participating actors: Students

Flow event:

a) The student browse the system and login to the system.
b) System displays the user dashboard.
c) From the dashboard student clicks take quiz option.
d) System displays available quiz.
e) Click the quiz and take the quiz.
f) Once they finish taking the quiz they submit their answers.
g) System displays the result of the quiz.

Exceptional condition:- if the user doesn’t solve all the questions in the quiz, the system tells the user to finish it.

Scenario 11: Check result

Participating actors: Student,Tutors and Companies

Flow event:

a) The user browse the system and login to the system.
b) The system displays the dashboard of the user.
c) From the dashboard select check result option.
d) System displays list of quizzes which are taken before.
e) Students select the quiz to check the result but tutors and companies select the student to check the result.

Scenario 12: Schedule class

Participating actors: Students

Flow event:

a) The Student browse the system and login to the system.
b) Student search a tutor.
c) Student click schedule class button from the tutor profile.
d) System displays a form to schedule the class.
e) Student fill the required data to the form.
f) Click submit button to schedule the class.
g) System displays success message.

Exceptional condition:- if the user enter in valid data to the form, the system shows an error and redirect back to the form.

Scenario 13: Accept schedule

Participating actors: Tutor

Flow event:

a) Tutor browse the system and login to the system.
b) System displays the dashboard of the user.
c) Tutor select schedules option from the dashboard.
d) System displays all schedules of the tutor from students.
e) Tutor accept and reject schedules accordingly.

Scenario 14: Online Payment

Participating actors: Students

Flow event:

a) Student browse the system and login to the system.
b) Student get room URL and click pay button.
c) System redirect to the payment platform option.
d) Student select payment option and pay the price.
e) After completing the payment the payment platform redirects back to the system.
f) System allow the user to get the service.

Exceptional condition:- if the payment information is invalid it displays an error. And the user can't get the service.

Scenario 15: Text Chat

Participating actors: Students,Tutors and Companies

Flow event:

a) User browse the system and login to the system.
b) Search the user account we want to chat.
c) Click send message button.
d) System displays message window for the user.
e) User sends text message to another user by clicking send button.

Scenario 16: Whiteboard

Participating actors: Students,Tutors and Companies

Flow event:

a) The user browse the system and login to the system.
b) User dashboard displayed to the user.
c) Select create room option or join an existing room from the user dashboard.
d) The system displays a virtual room.
e) From this virtual room user selects whiteboard button to write something.
f) This written text will be visible for the users.

Scenario 17: Report

Participating actors: Students

Flow event:

a) Student browse the system and login to the system.
b) Search a tutor and company to apply a report.
c) Click Report button from the user profile.
d) System displays a form to the student.
e) Student fill the form.
f) Submit the report to the system.

Exceptional condition:- if the data is not valid the system will display an error.

3.4.2 Use case model

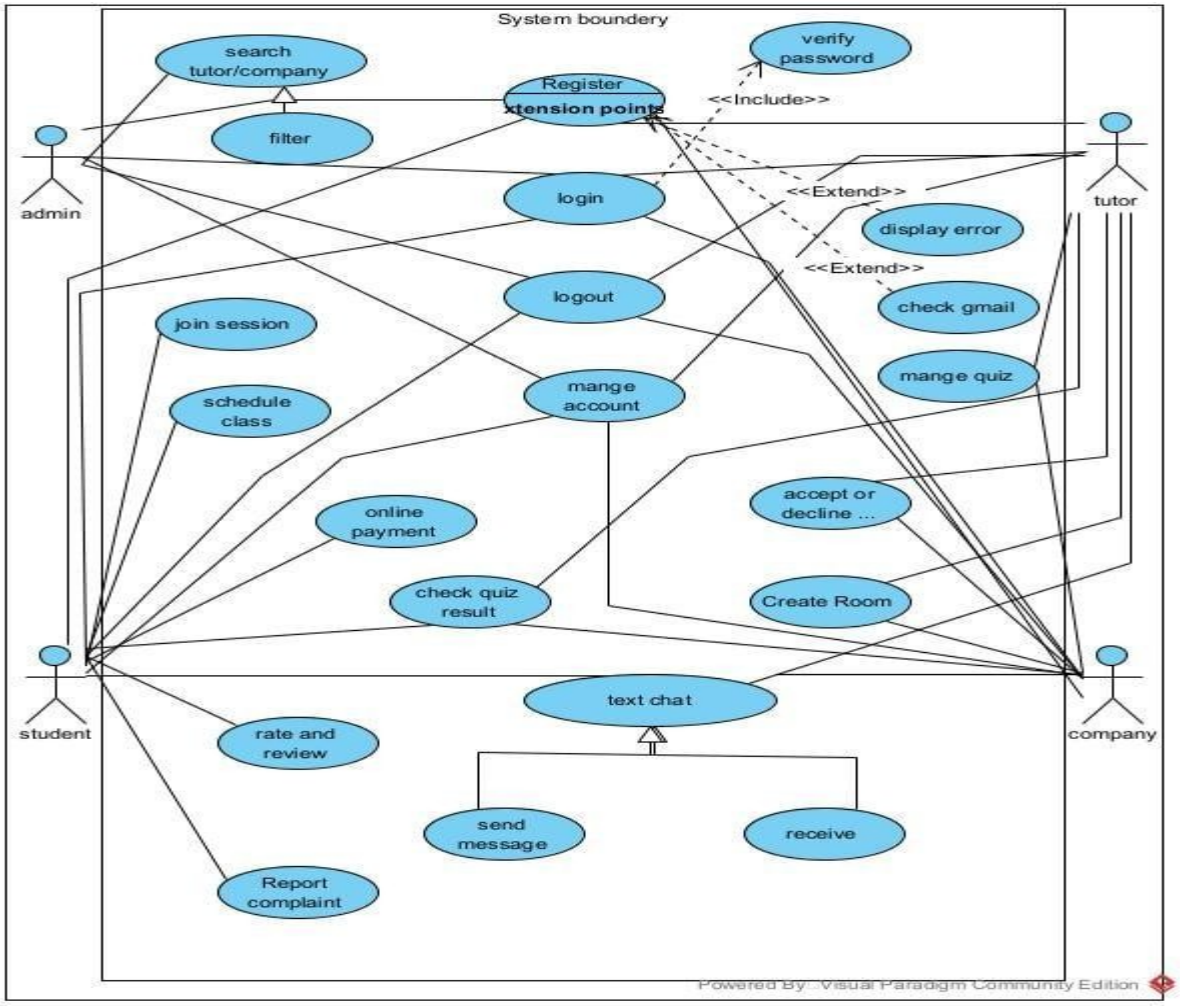


Figure 1: Figure for use case diagram

3.4.3 Use case description

Use Case Name	Login
Use case ID	UC-1
Include	Validation of form inputs, Authentication
Extends	Display Error message
Description	This use case describes how the user login to the system. The user must have a valid email and password to log in.
Participant Actor	Students, Tutor , Company and Admin
Precondition	The user must have an account and remember the username and password before trying to login.
Post condition	The user granted access to get service from the system according to their role or if it fails the authentication, Error message will display on screen and they will try again.
Flow of Actions	1. The user clicks the login button from the landing page. 2. The system displays a login form. 3. The user inputs their user name and password. 4. The user clicks the button login. 5. The system authenticates the user 6.The system allows users to access their privileged tasks.
Alternate Flow of Action	None
Exception	if the user submit wrong email or password the system displays an error.

Table 6 use case description of login

Use Case Name	Register
Use Case ID	UC-2
Include	Validation of Input, check Email and username
Extends	Display error message
Description	It describes how the user registers to the system. The user must complete filling the required input fields like username, email, password, confirm password.
Participant Actors	Company, Students and Tutor
Precondition	The user must have a unique email and username.
Post condition	The user must verify the email. then the account is created.
Flow of Actions	1. The user clicks the register button from the landing page 2. The system displays a register form. 3. The user fills their account information like username, email password... 4. The user clicks the button to register. 5. The system sends a confirmation email to the user email 6. The user confirms the email via the confirmation email 7. The user will be signed in successfully
Alternate Flow of Action	None
Exception	1. If the user enters an existing username or email or both, display an error message and it may go to sign in page. 2. If the user enters invalid information and missing required inputs, display error message

Table 7 sign up

Use Case Name	Forgot Password
Use Case ID	UC -3
Include	Validation of form inputs
Extend	Display Error message
Description	This use case describes how the user able to recover his/her account if he/she forgot either username or password.
Participant Actors	Company, Tutor , Students and admin
Precondition	The user must access either their email or phone that they provided for sign up.
Post condition	The user can log in to the account.
Flow of Actions	1. The system prompts the user to enter either their phone or email 2. The user inputs email or phone 3. If it is correct the user will receive forgot link by email 4 if the user receive the link right time, the can create new password 5. If it is not, display an error message 6. The user redirected to login using a new username or password
Alternate Flow of Action	Ask admin to reset to send their name and email by making phone call admin will check and send by sms if the phone is registered.
Exception	1. If the user enters an incorrect answer, display an error message. 2. If the user enters invalid input (new username and password) display an error message.

Table 8 forgot password

Use Case Name	Account Setting
Use Case ID	UC- 4
Include	Validation of form inputs, View Profile, Check email and username
Extend	Display Error message
Description	This use case describes how the user can change different account information like username, email, password
Participant Actors	Company, Tutor, student and admin
Precondition	The user must be logged in and remember the old password to change a new password
Post condition	Changed Account information
Flow of Actions	1. The user clicks the change account button. 2. The system displays a form. 3. The user change account settings as much as he/she wants 4. If the user changes the email, the system will send a conformation email and the user logs out from the system 5. The user clicks the confirmation email from his/her email 6. The user will log in using his username and password 7. The user clicks the save button and account info will be changed
Alternate Flow of Action	None
Exception	1. If the user enters an existing email or username, display an error message. 2. If the user enters an incorrect old password, display an invalid password message 3. If the user enters invalid input, display an error message.

Table 9 account settings

Use Case Name	Search Tutors or Companies
Use Case ID	UC- 5
Include	None
Extend	None
Description	This use case describes how the user can browse Tutors or Company by Course or other parameters
Participant actors	Students , Company , Tutor
Precondition	The user must logged in to use this feature
Post condition	Searches will be retrieved
Flow of Actions	1. The user writes the key he/she wants to search 2. The system displays available Tutors or Companies if they matched to key. 3. If it is not matched, the system displays search not Found.
Alternate Flow of Action	None
Exception	None

Table 10 search

Use Case Name	Filter Tutors or Companies
Use Case ID	UC-6
Include	None
Extend	None
Description	This use case describe how the user can view search response by sorting of based on their query
Participant Actor	Company, Students and Tutor
Precondition	The user must have logged in to see retrieved searches.
Post condition	Searched responses will be available
Flow of Actions	• The user selects the style(sorting,by Name, popularity) he/she wants to view. • Retrieved search will be available for user
Alternate Flow of Action	None
Exception	None

Table 11 filter searches

Use Case Name	View Profile
Use Case ID	UC-7
Include	authentication
Extend	none
Description	This use case describe how the user can see users detail profile
Participant Actors	Students
Precondition	User must retrieve user's by using search query either popular or by name
Post condition	User profile will be showed
Flow of Actions	• The user click on users profile link • the use can show user's detail profile information
Alternate Flow of Action	none
Exception	none

Table 12 view profile

Use Case Name	Rate Tutor or Company
Use Case ID	UC- 8
Include	none
Extend	none
Description	This use case describes how the students can give rating for tutors and companies.
Participant Actor	Students
Precondition	The user must be logged in
Post condition	Will be Rate successfully
Flow of Action	The student browse the system and login to the system. Student search a tutor or company. Student clicks the selected tutor or company. From the tutor or company profile click rate button. The system displays a popup window for the student. The student give rating to the tutor or the company and submit the rating. The system remove the popup window after submission of the rating.
Alternative Action	none
Exception	none

Table 13 rate

Use Case Name	Create Room
Use case ID	UC-9
Include	none
Extend	none
Description	This use case describes how to user can to create virtual class room
Participant Actors	Tutors , Companies
Precondition	The user must be logged in
Post condition	The user will be redirected to class room
Flow of Actions	• The user browser and login to the system • user dashboard displayed to the user • the user select create room option from the dashboard • the system will create the room and unique room will become available to users
Alternative Flow of Actions	none
Exception	none

Table 14 create room

Use Case Name	Join session
Use Case ID	UC-10
Include	Unauthorized user error
Extend	wrong room URL error
Description	This use case describes how a paid user able to join the session
Participant Actor	Students
Precondition	The user must logged in and must paid for the class
Post condition	The user will redirected to virtual room to let users to talk using audio, video and text chat features
Flow of Action	• The student browse the system and log in • the student click the URL which is sent from the teacher or Tutor • the student pay for the class and the system will redirect to the next steep • The user will redirected to virtual room to let users to talk using audio, video and text chat features
Alternative Action	none
Exception	If the student try to join wrong session, there will be displayed wrong room URL error

Table 15 join session(class)

Use Case Name	screen share
Use Case ID	UC- 11
Include	none
Extend	none
Description	This use case describes how the user can share the screens at the time of real time video communication
Participant Actor	Tutor, Company
Precondition	The student should be logged in
Post condition	Shared screen will be available
Flow of Action	• The user browse and log in to the system • user dashboard displays to user • user select create room or join existing room from their dashboard • the system displays a virtual room • tutor and companies can click share screen button to share their screen to the students • finally, the shared screen will be visible for students.
Alternative Action	none
Exception	none

Table 16 screen share

Use Case Name	Review User
Use Case ID	UC-12
Include	none
Extend	none
Description	This use case describes how the user able to review to the admin
Participant Actor	Students
Precondition	The student must be logged in with their account
Post condition	Review submitted successfully message will be displayed
Flow of Action	h) The student browse and login to the system i) the student search a tutor or select company j) student clicks the selected tutor or company k) from the tutor or company profile click for review button l) the system displays a review from the student m) the student fills the review form and click submit button n) the system displays review successful message
Alternative Action	none
Exception	none

Table 17 review user

Use Case Name	Create Quiz
Use Case ID	UC- 13
Include	Validation of form inputs
Extend	Error message
Description	This use case describes how the Create Quiz use case works
Participant Actor	Tutors, Companies
Precondition	The student must be logged in
Post condition	Success message will be displayed
Flow of Action	f) Tutor or Companies log in to the system g) the system displays the user dashboard h) the user select create quiz option i) the system displays a form to create quiz j) user enters a valid values on the form and click submit button k) the system displays success message
Alternative Action	none
Exception	If the user enters invalid data to the form it displays error message to the user.

Table 18 create quiz

Use Case Name	Take Quiz
Use Case ID	UC- 14
Include	none
Extend	Error message
Description	This use case describes how take quiz feature can work
Participant Actor	Students
Precondition	The student must be logged in
Post condition	Quiz result will be displayed
Flow of Action	h) The student browse and logged in to the system i) the system displays user dashboard j) the student click take quiz button k) system display available quiz l) after they finished they click submit buttons m) the system displays their result
Alternative Action	none
Exception	If the user doesn't finished all the quiz questions the system will notify to do all of them

Table 19 take quiz

Use Case Name	Check Result
Use Case ID	UC-15
Include	none
Extend	none
Description	This use case describes how User can check their any quiz results.
Participant Actors	Students, Tutors, Companies
Precondition	The user must be logged in
Post condition	Quiz results is displayed
Flow of Actions	f) The user browse and log in to the system. g) The system displayed user dashboard. h) User clicks Check result option on the dashboard. i) System displays list of quizzes which are taken. j) Students can select the quiz that to check but Tutor and companies select the students to check the result. k) Result is displayed
Alternative Flow of Actions	none
Exception	none

Table 20 check result

Use Case Name	Schedule Class
Use Case ID	UC- 16
Include	Validate time and day
Extend	Error message
Description	This use case describe how the user can scheduling the time to attend the class
Participant Actor	Students
Precondition	A user must Choose his/her favorite Tutor for course to send schedule class request
Post condition	The student will be accepted or rejected
Flow of Action	• The student choose time and days when he/she will available to attend the class • then hit the send request button • the Tutor will check the class schedule request and will give response • either schedule will be rejected or approved message will be sent back to user
Alternative Action	None
Exception	If the user selected invalid date, Invalid input message will be displayed

Table 21 schedule class

Use Case Name	Accept Schedule
Use Case ID	UC-17
Include	none
Extend	none
Description	This use case describes how the use Tutor or the company can check the pending schedule requests.
Participant Actor	Tutor and Companies
Precondition	The user must logged in to their account
Post condition	The user can see all pending schedules
Flow of Action	• The user must click on pending schedule link(icon) • all pending schedule will be sorted down • the tutor or company can approve or reject the schedule by clicking either button
Alternative Action	none
Exception	none

Table 22 accept schedule

Use Case Name	Payment
Use Case ID	UC-18
Include	none
Extend	Insufficient balance error
Description	This use case describes how the student able to commit payment for class that he is going to attend
Participant Actors	Students
Precondition	The student must logged in
Post condition	System redirect the student back to the class screen.
Flow of Action	g) Students log in to the system h) if the schedule is approved, the tutor or company sends a link that redirects the student to payment platform. i) student follow the link and choose payment way form the list j) the student commit payment k) if they paid successfully the transaction committed successfully message will be displayed. l) the system redirect them back to the class screen.
Alternative Flow of Action	f) If the above payment is not working the user will pay manually g) then he must send transaction key with sms or chat message h) admin will check wither the transaction and sender name Is same or not i) if the info is correct he system redirect them to the class screen.
Exception	none

Table 23 payment

Use Case Name	Update
Use Case ID	UC-19
Include	Invalid input
Extend	Error message
Description	This use case describes about how the user able to edit their accounts. Or how Admin can edit other's account
Participant Actors	Admin, Student, Tutor and Company
Precondition	The user must be signed in
Post condition	Update is displayed
Flow of Action	g) Admin opens users profile and click on edit that wanted to be update. User opens their edit profile button. h) The form will be opened i) if user want to change his password, the system ask to fill the old password j) the user click update button k) the system authenticate the user and if the user is authorized, updates will be committed else invalid input will be displayed. l) Updates will be displayed
Alternative Flow of action	none
Exception	If the user uses wrong password as old message error message will be displayed.

Table 24 update profile

Use Case Name	Delete
Use Case ID	UC-20
Include	none
Extend	Error message
Description	This use case describes about how the user can delete their accounts. Or how admin can Delete other's account
Participant Actors	Admin, Student, Tutor and Company
Precondition	The user must be signed in
Post condition	Account will be deleted successfully
Flow of Action	f) The user log in to the system. g) admin search account that needed to be delete. User opens their account setting link. h) the user must insert their current password. i) then the user click the delete button. j) the system will delete the account successfully. k) User will be redirect to homepage of the system and admin will redirect to users list.
Alternative Flow of action	none
Exception	If the user uses wrong password as old message error message will be displayed.

Table 25 delete account

Use Case Name	Log Out
Use Case ID	UC-21
Include	none
Extend	none
Description	This use case describes about how the user can exit from the system
Participant Actors	Admin, Student, Tutor and Company
Precondition	The admin and users must be logged in
Post condition	none
Flow of Action	1. The users clicks log out button 2. The user exit from the system
Alternative Flow of action	none
Exception	none

Table 26 logout

Use Case Name	Create Profile
Use Case ID	UC-22
Include	invalid input
extend	Error message display
Description	This use case describes how the Tutor or Company can fill about their profile data
Participate Actors	Company and Tutor
Precondition	The company or Tutor must have logged in
Post condition	Profile create successfully message will be displayed
Flow of Action	g) The user logs in to his/her account h) user click on create profile button i) the system displays profile form j) user fill the filed with appropriate data k) the user click submit button l) the system save the profile and displayed m) Profile create successfully message will be displayed
Alternative flow of action	none
Exception	If the user fill miss data error message will be displayed

Table 27 create profile

Use Case Name	Report
Use Case ID	UC-23
Include	none
extend	Error message
Description	This use case describes how the user can report for admin while they got something which is wrong
Participant Actors	Students
Precondition	Use must be logged in to the system
Post condition	Successfully reported message will be displayed
Flow of Action	1. The user login to the system 2. the user open tutor or companies profile 3. user click report button 4. the user fill the input 5. the user click submit button 6. successfully reported message is displayed
Alternative Flow of Actions	none
Exception	If the user fills unnecessary character on report form error message is displayed

Table 28 report

Use Case Name	Send Text Message
Use Case ID	UC-24
Include	none
Description	This use case describes how the user can send text message
Participant Actors	Student, Tutor and company
Precondition	The user logged in to the system
Post Condition	The user sent message successfully
Flow of Actions	1. A user click on user profile 2. the user click on message button 3. the user write message on input filed 4. user click send button 5. message sent successfully
Alternative Flow of Actions	none
Exception	none

Table 29 sent text message

Use Case Name	Read Text Message
Use Case ID	UC-25
Include	none
Description	This use case describes how the user can read text message
Participant Actors	Student, Tutor and company
Precondition	The user logged in to the system
Post Condition	The user read message
Flow of Actions	1. The system notify new message 2. the user click on message button 3. message page will opened 4. user click new message sender card 6. message opened successfully
Alternative Flow of Actions	none
Exception	none

Table 30 read text message

3.5 Object model

3.5.1 Data Dictionary

A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmer others who need to refer to them.

Classes	Attributes	Operations	Description
User	_id, firstName , lastName ,role ,email, password createdAt,updatedAt	Login(), Register(), Logout()	Allows the customers of the system to register, login to the system and logout from the system.
Student	_id,userId,Img,createdAt, updatedAt	Create(),Update,Delete()	This object let students to create their profile and delete their account if it is not necessary.
Tutor	_id,userId,Price,experience,academicStatus,phone, img,about,createdAt,updatedAt	Create(),Update,Delete(), View()	This let tutor to create their profile and delete their account if it is not necessary. And let other users to see the profile of tutors.
Company	_id,userId,phone,address, img,about,createdAt,updatedAt	Create(),Update,Delete(), View()	This let companies to create their profile and delete their account if it is not necessary. And let other users to see the profile of company.
Room	_id,userId,title,createdAt, updatedAt	Create(),Update(),Delete()	This object let companies and tutors to create a virtual room.
scheduleClass	_id,userId,studentId,from,to,status,createdAt,updatedAt	Create(),Update(),Delete(),Accept(),decline(),Display(),	This allow users to schedule a virtual class and accept and decline a schedule if they are busy by that time.
Review	_id,studentId,userId,content,createdAt,updatedAt	Write(),view(),update(),delete()	This object let students to give review for tutors and companies.
Rate	_id,studentId,userId,rate, createdAt,updatedAt	Rate(),view(),update(),delete()	This object let students to give rate for tutors and companies.
Quiz	_id,userId,title,description,questions,createdAt,updatedAt	Create(),View(),update(), delete()	Create simple quiz to asses their students.
TakeQuiz	_id,userId,quizId,answewers,createdAt,updatedAt	Submit(),Result(),update()	Students take quiz which are given from their tutors and companies.
Question	_id,quizId,question,choice,answer,createdAt,updatedAt	Add(),Update(),Delete()	Create a question.
Payment	_id,userId,roomId,status, price,createdAt,updatedA	Pay(),show(),	Students pay for the service the want get.

	t	
Message	_id,senderId,recieverId,message,createdAt,updated At	Send(),Read(),Update(), Delete()	Allow users to send and receive messages.
Report	_id,reporterId,userId,report,createdAt,updatedAt	Report(),View()	Users can report to the admin of the system if they don’t get the service they want properly.

Table 31: Data dictionary

3.5.2 Class diagram

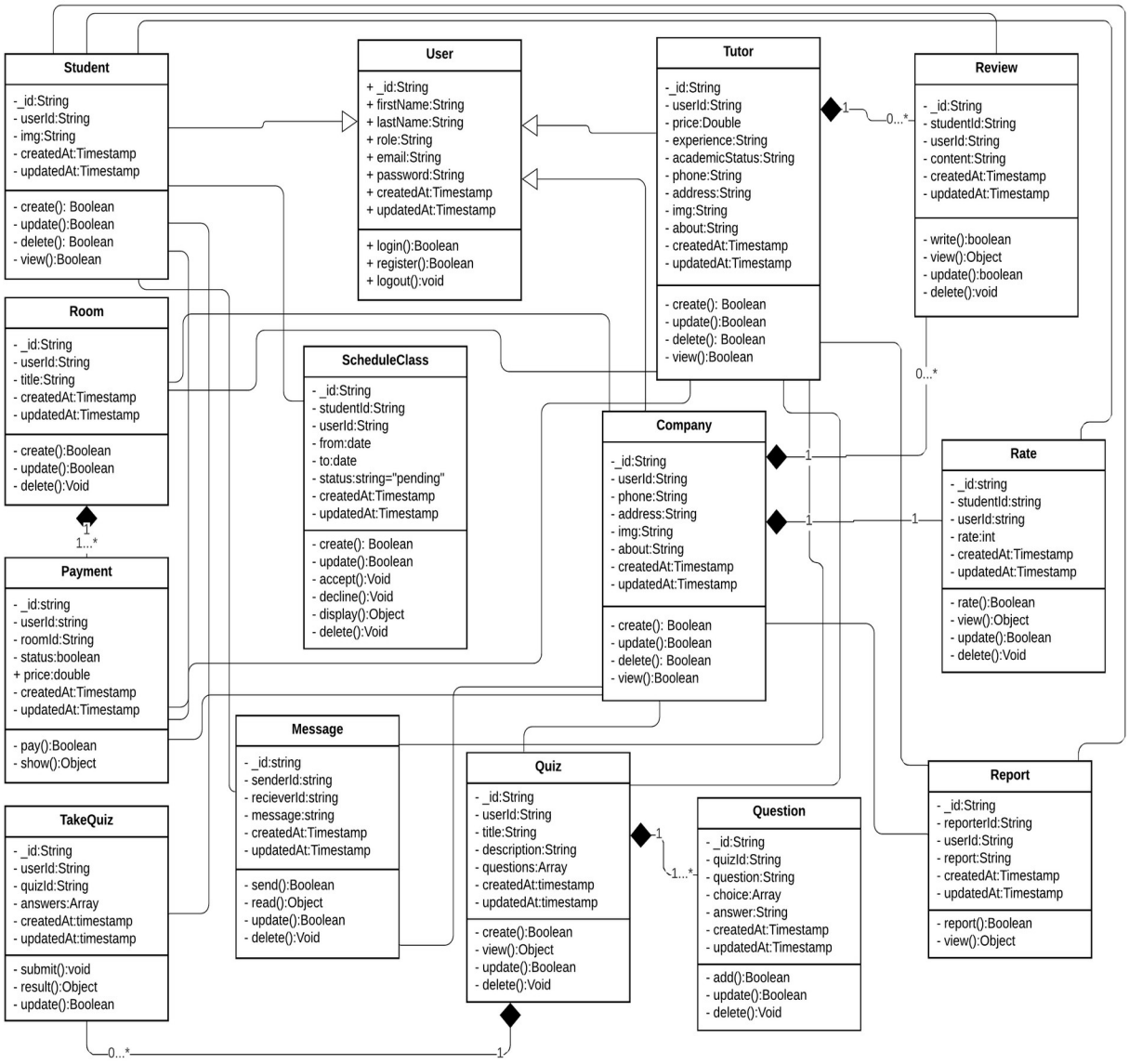


Figure 2: Figure of class diagram

3.6 Dynamic model

3.6.1 Sequence diagram

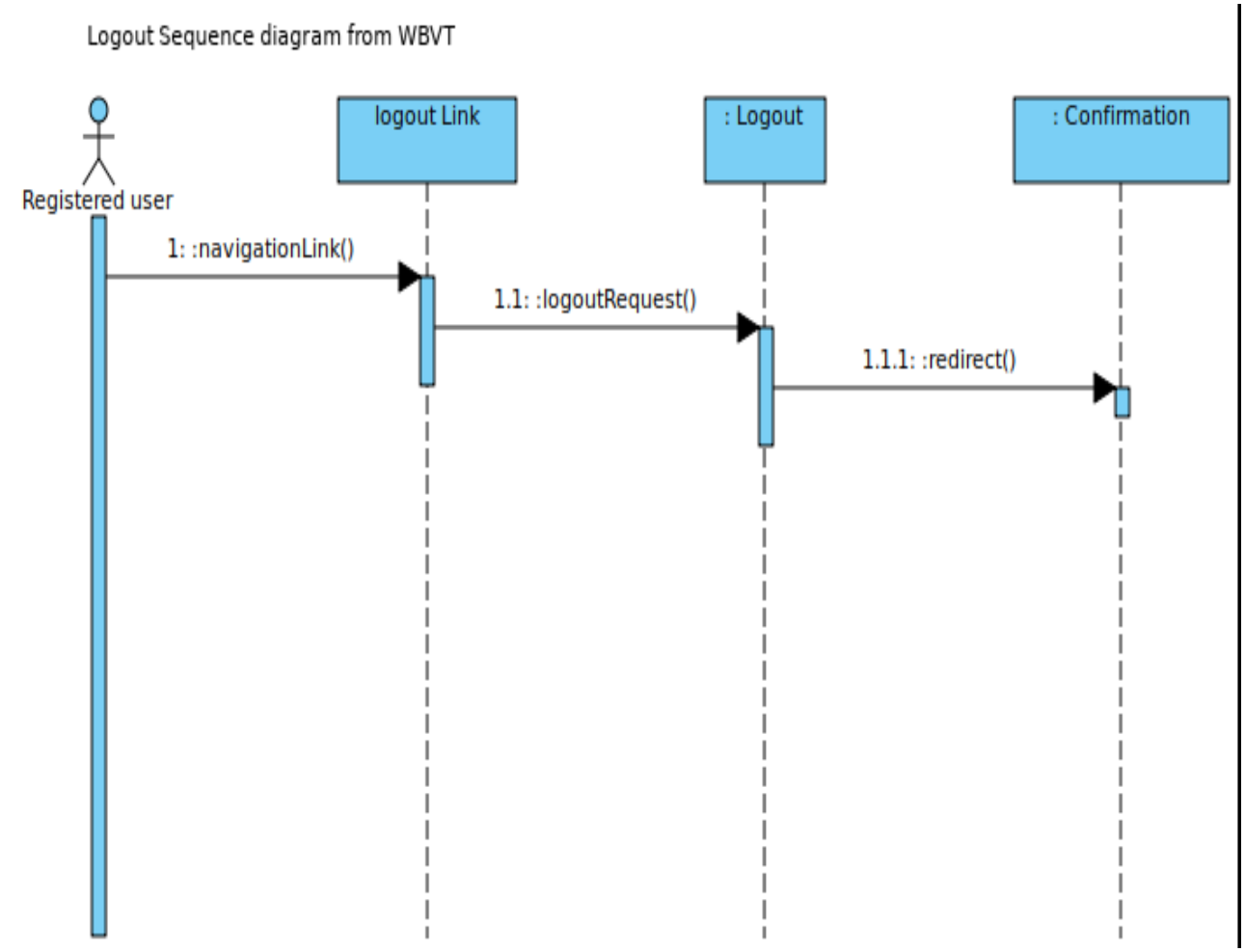


Figure 3: Sequence diagram for login and logout

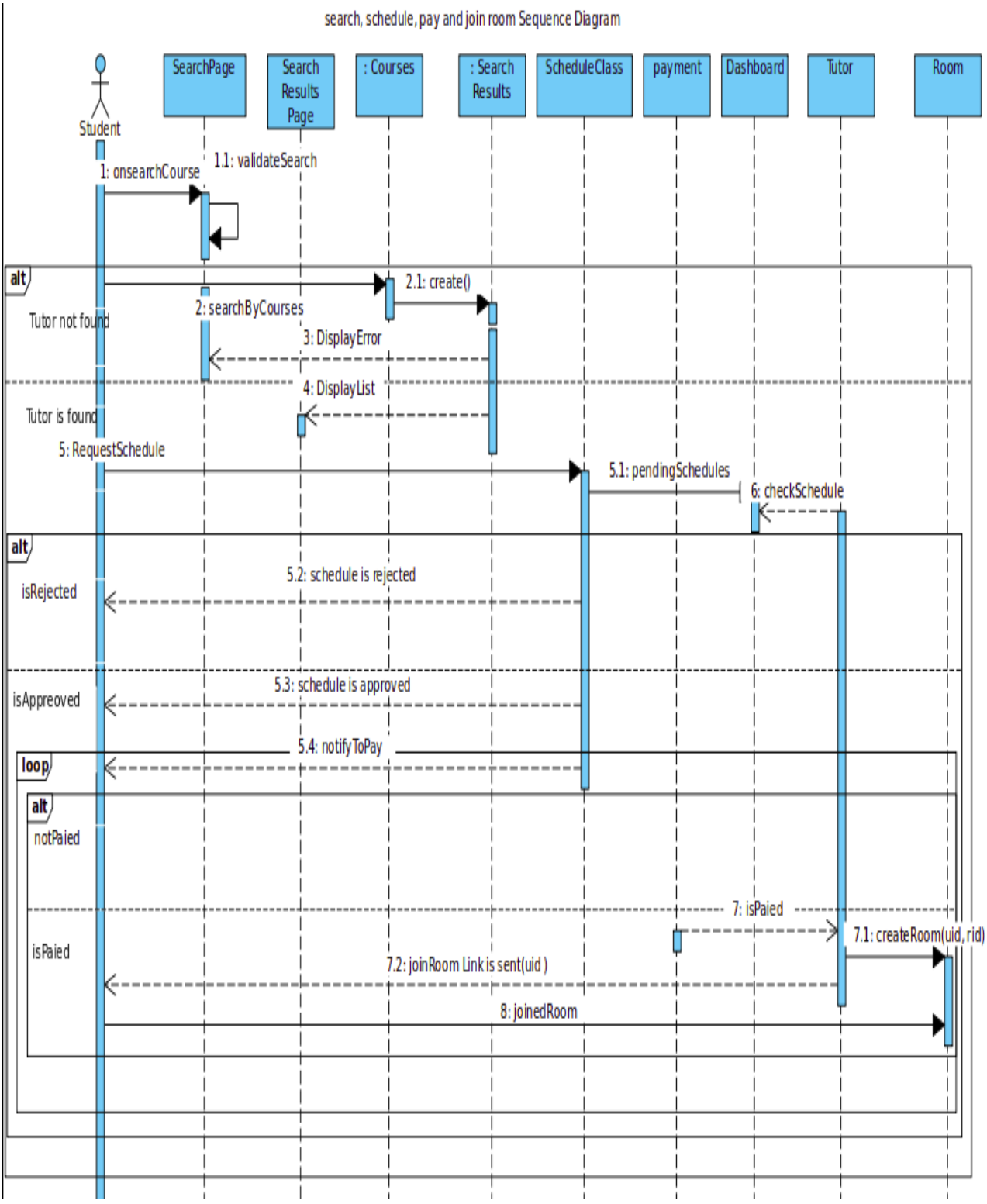


Figure 4: Sequence diagram for schedule class and join room

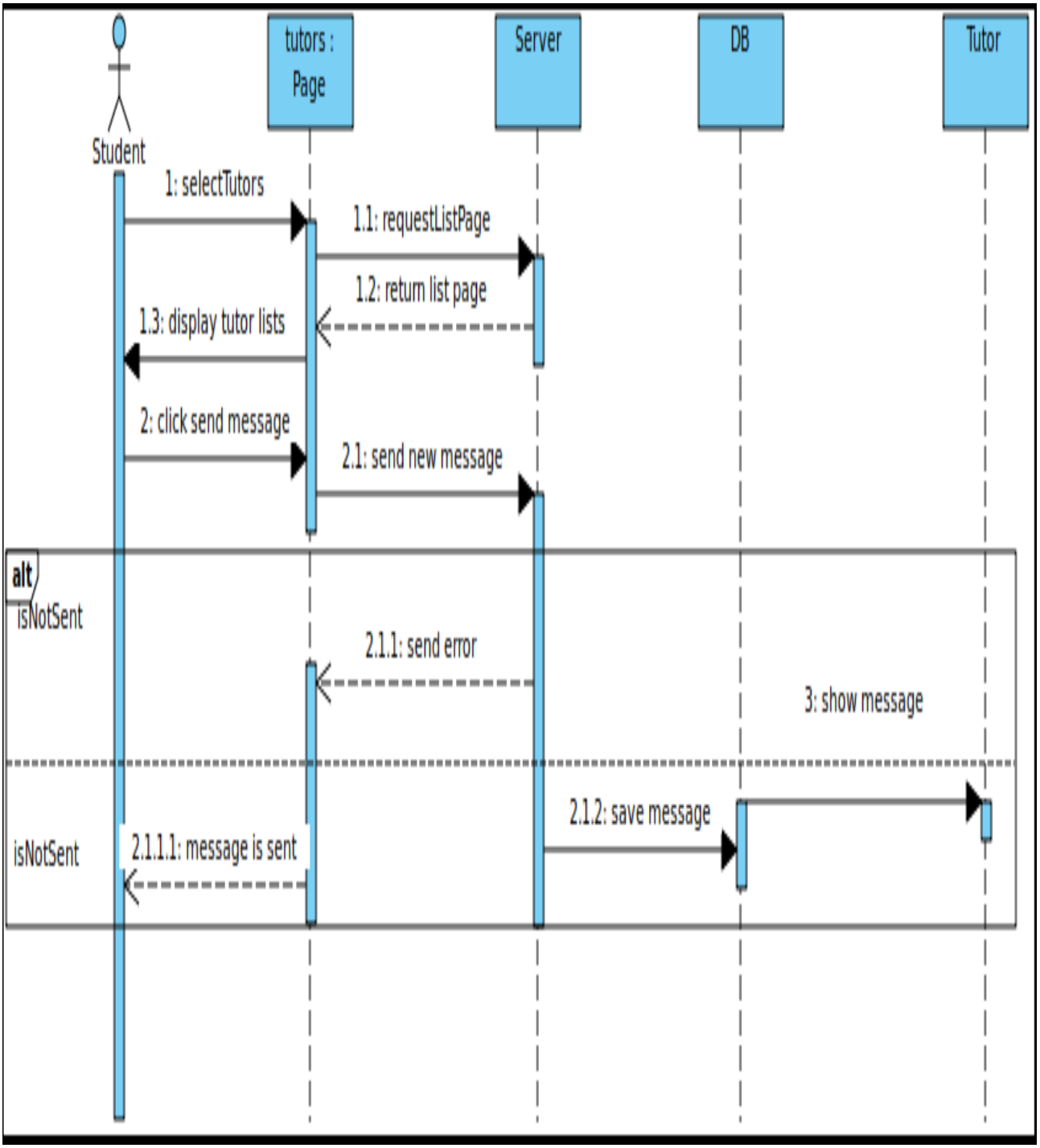


Figure 5: Sequence diagram for text chat

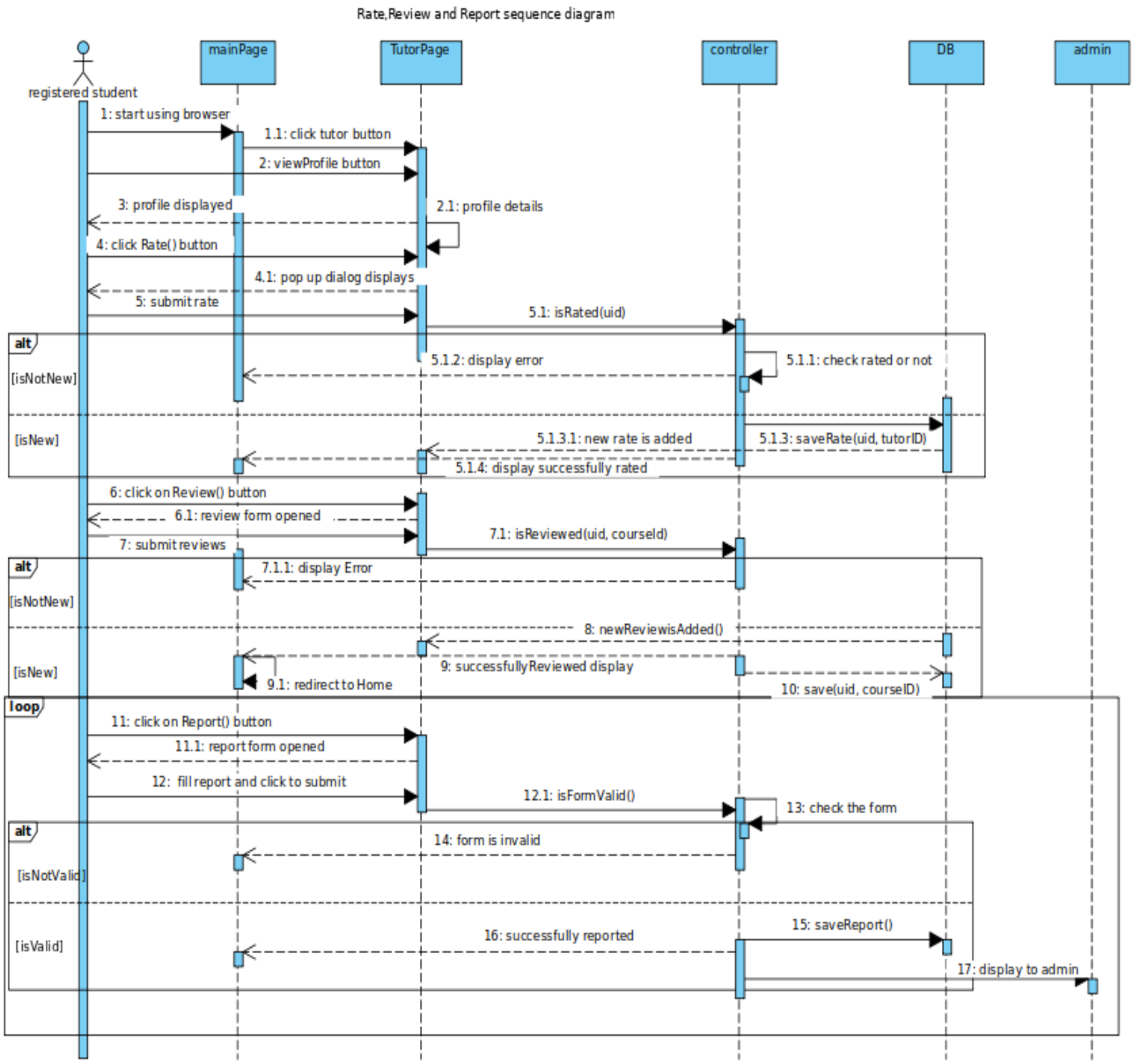


Figure 6: Sequence diagram for rate,review and complaint report

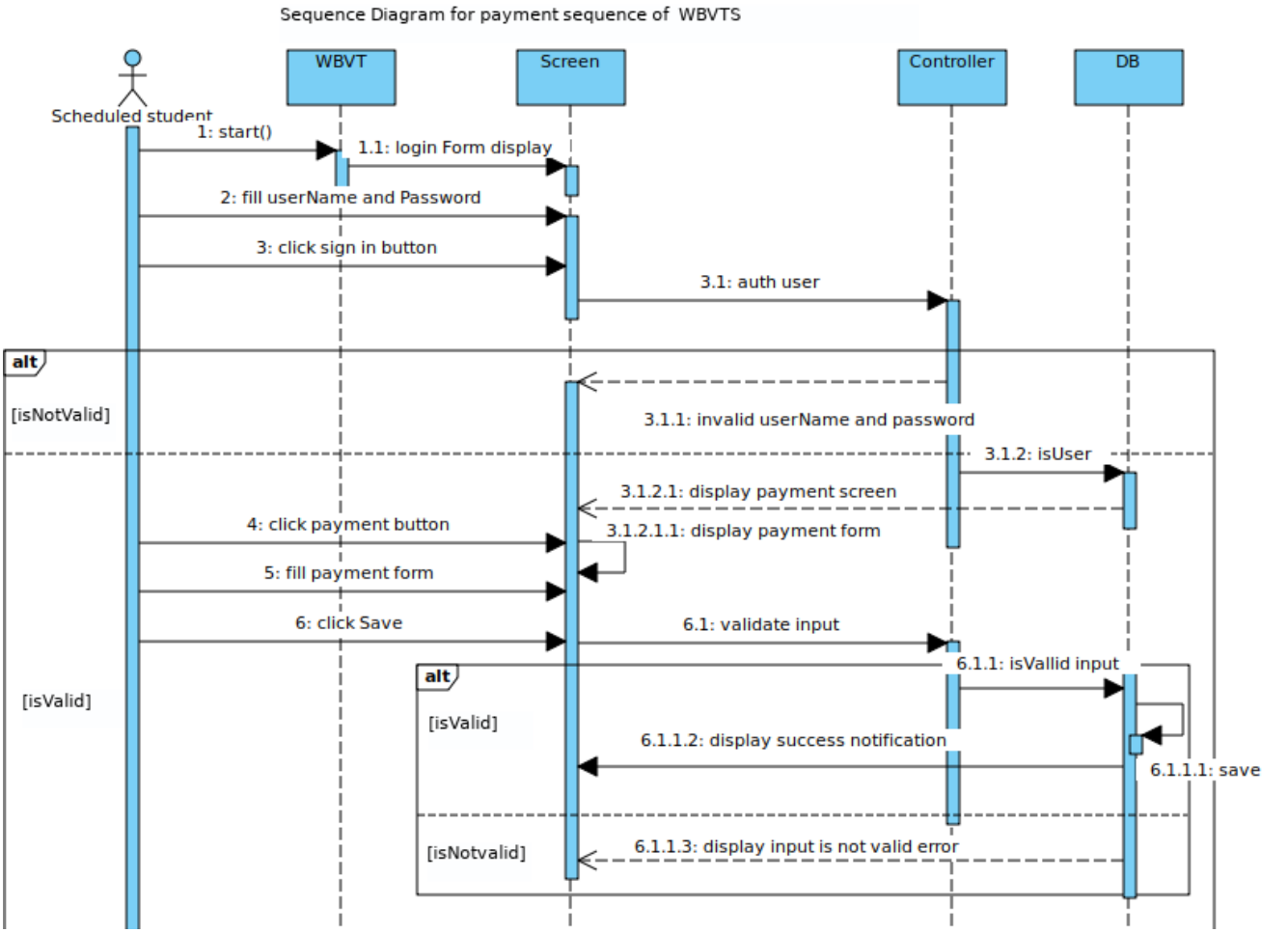


Figure 7: Sequence diagram for online payment

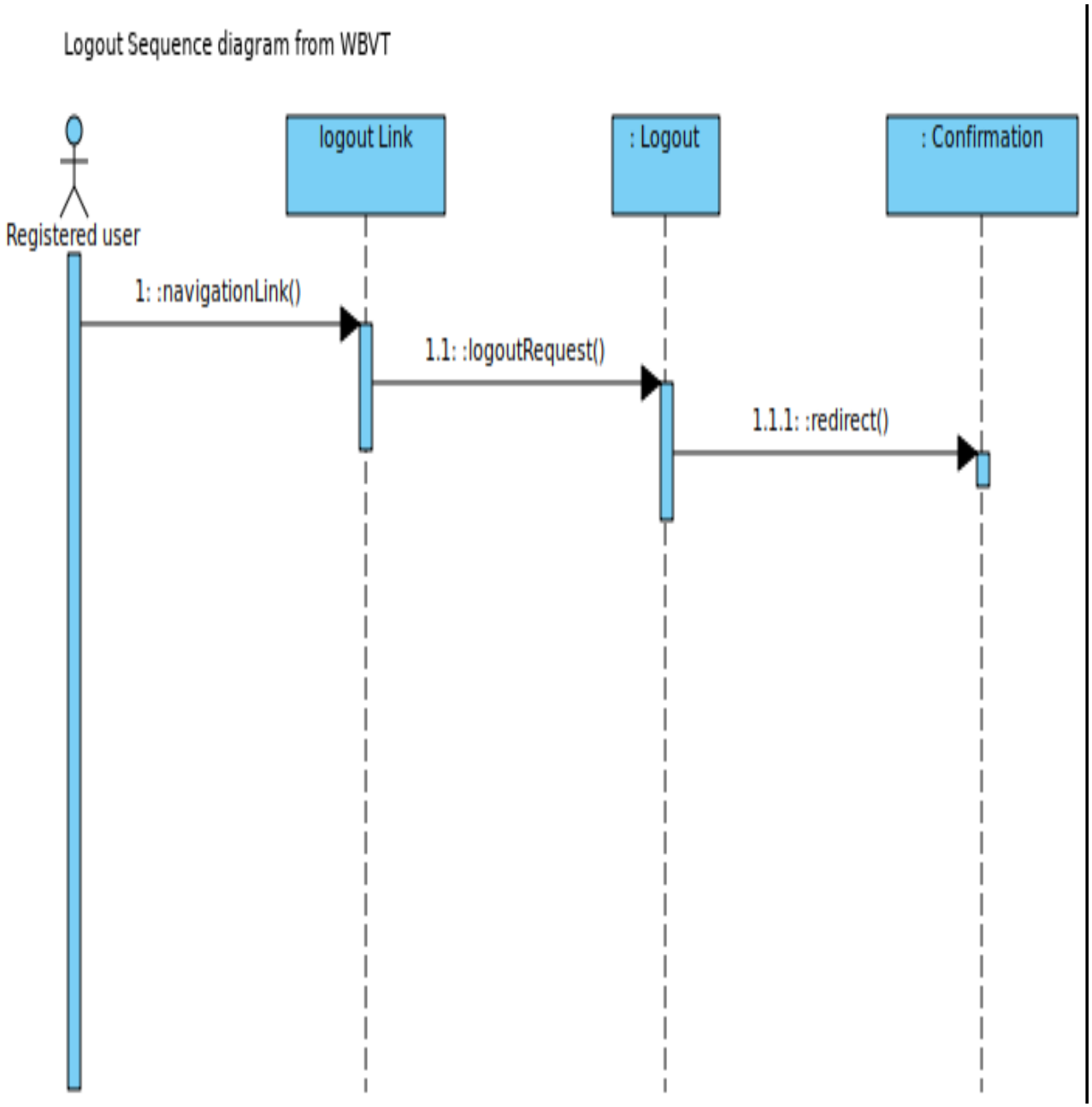


Figure 8: Logout sequence diagram

3.6.2 Activity diagram

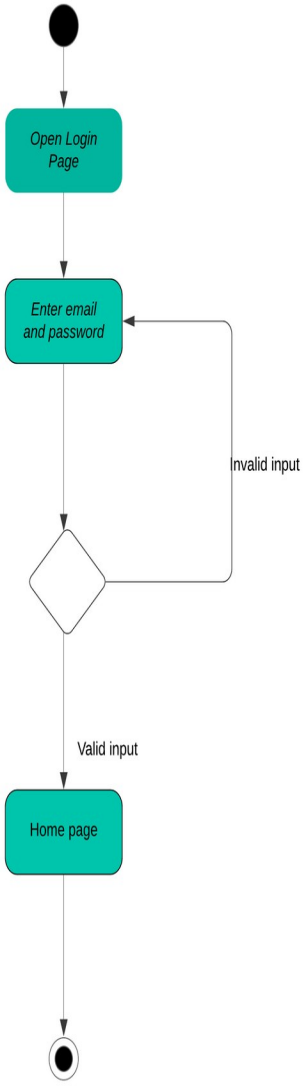


Figure 9: Activity diagram for login

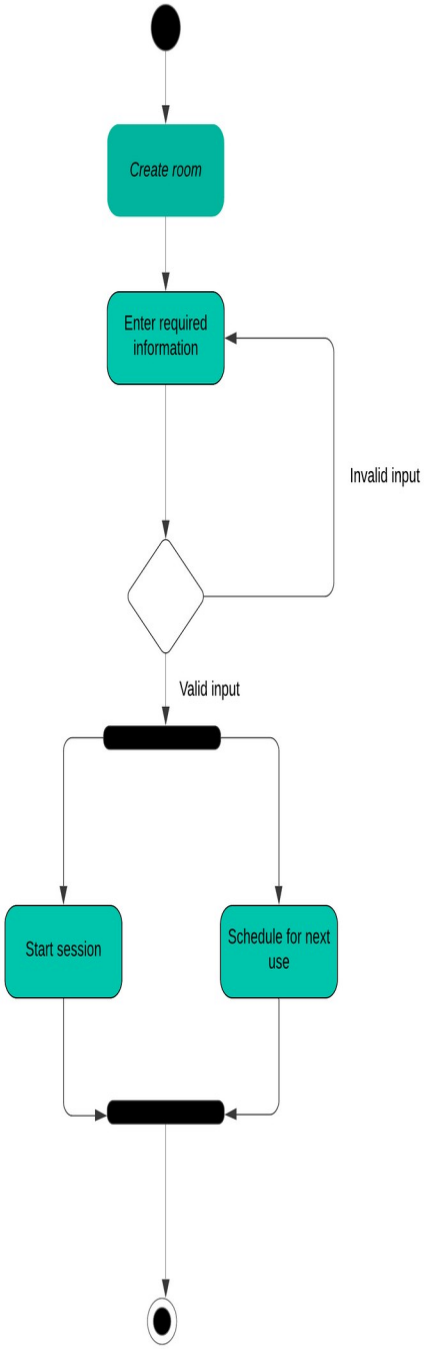


Figure 10: Create room activity diagram

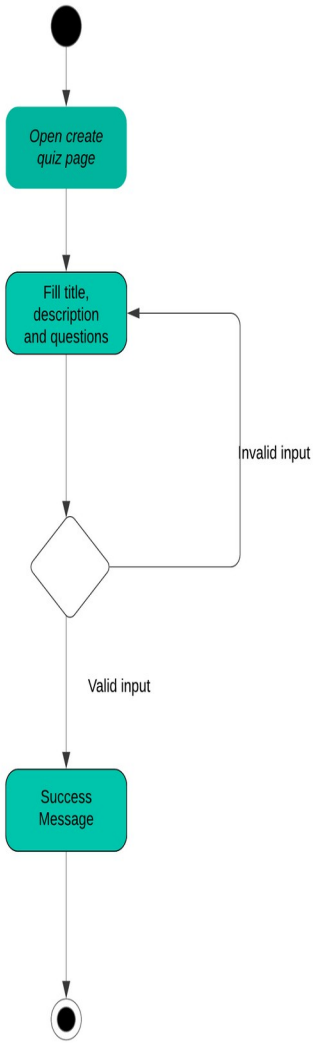


Figure 11: Create quiz activity diagram

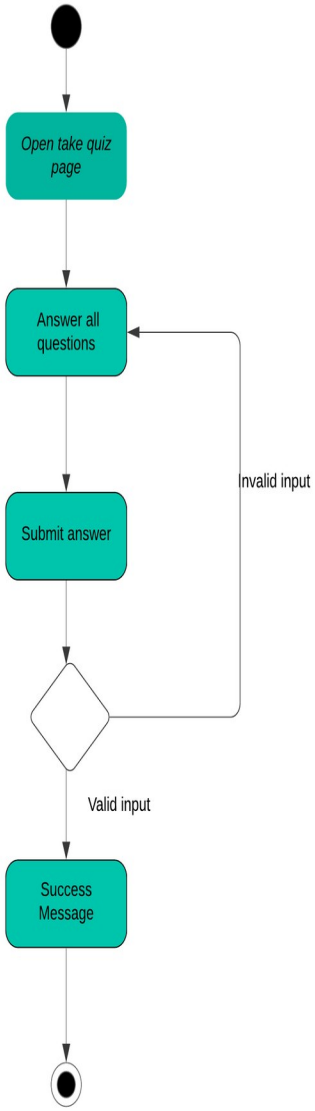


Figure 12: Take quiz activity diagram

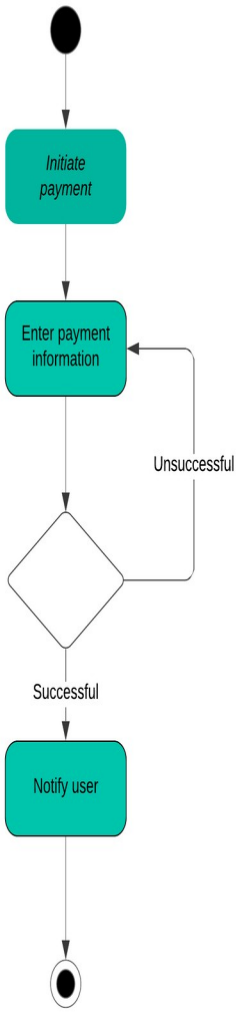


Figure 13: Payment activity diagram

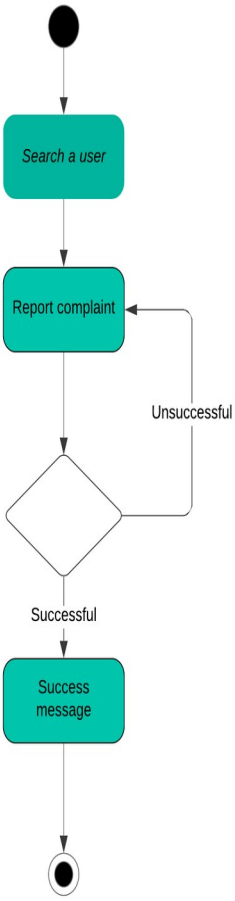


Figure 14: Complaint report activity diagram

3.6.3 State chart diagram

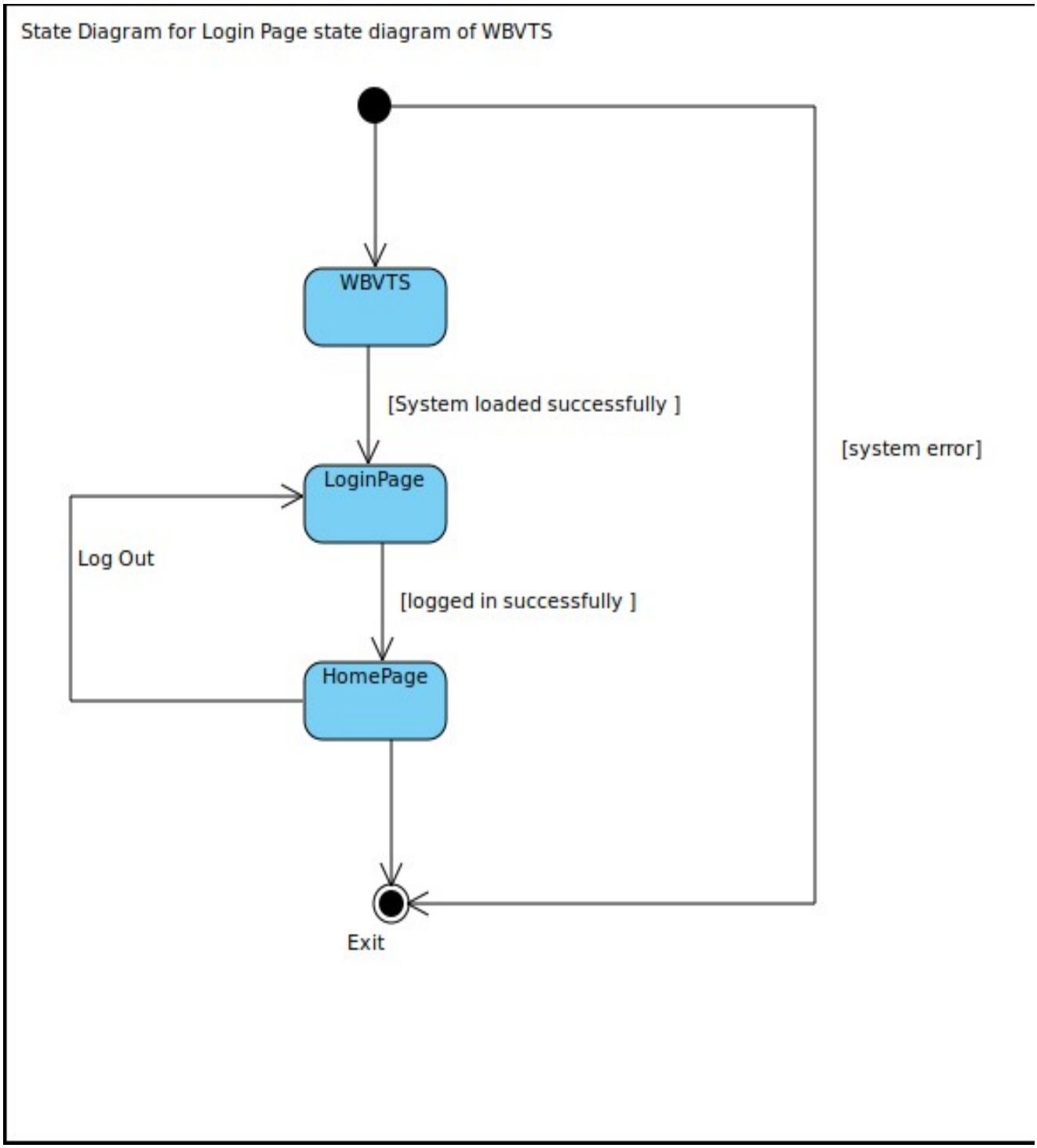


Figure 15: Login state chart diagram

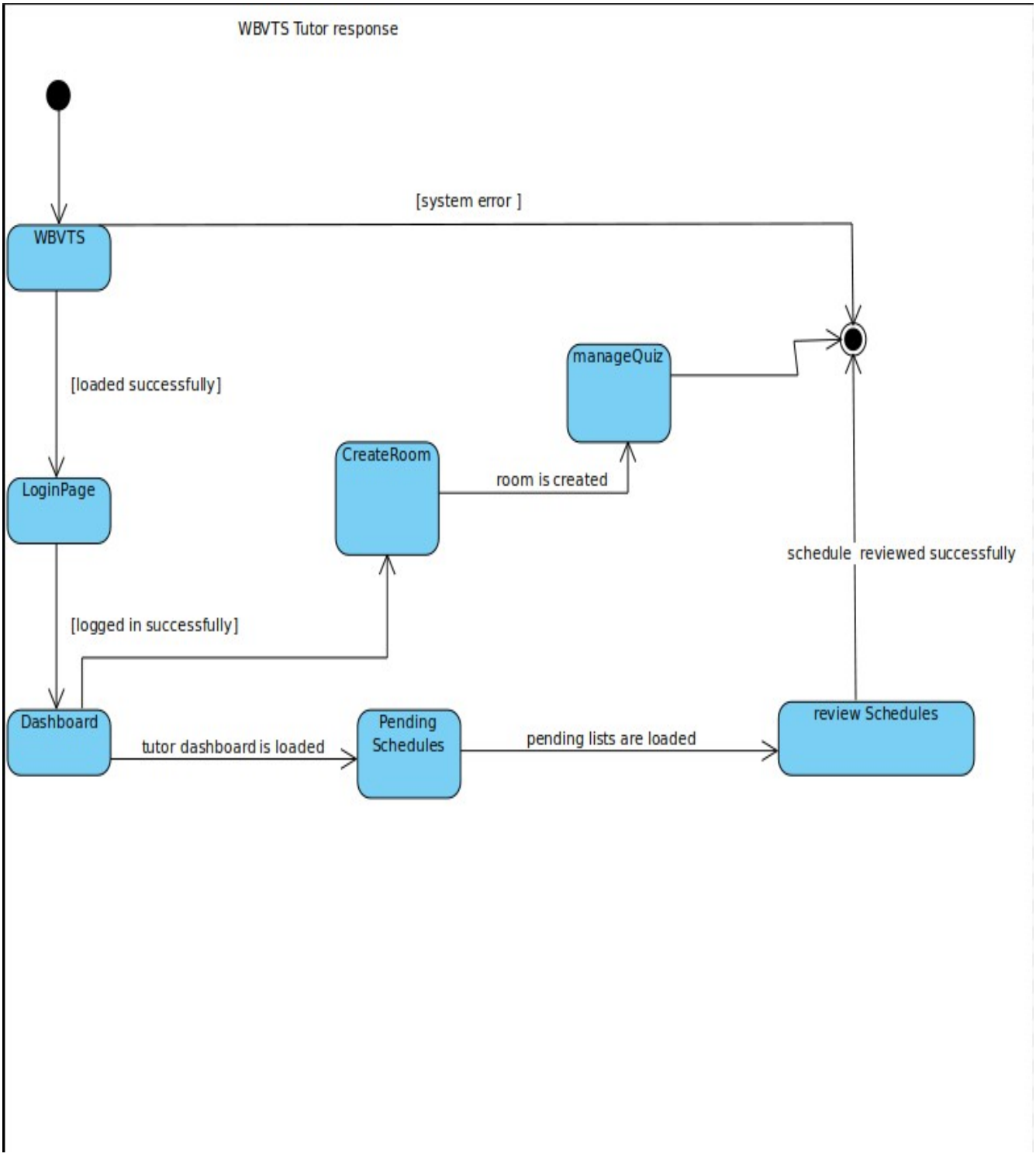


Figure 16: Schedule class state chart diagram

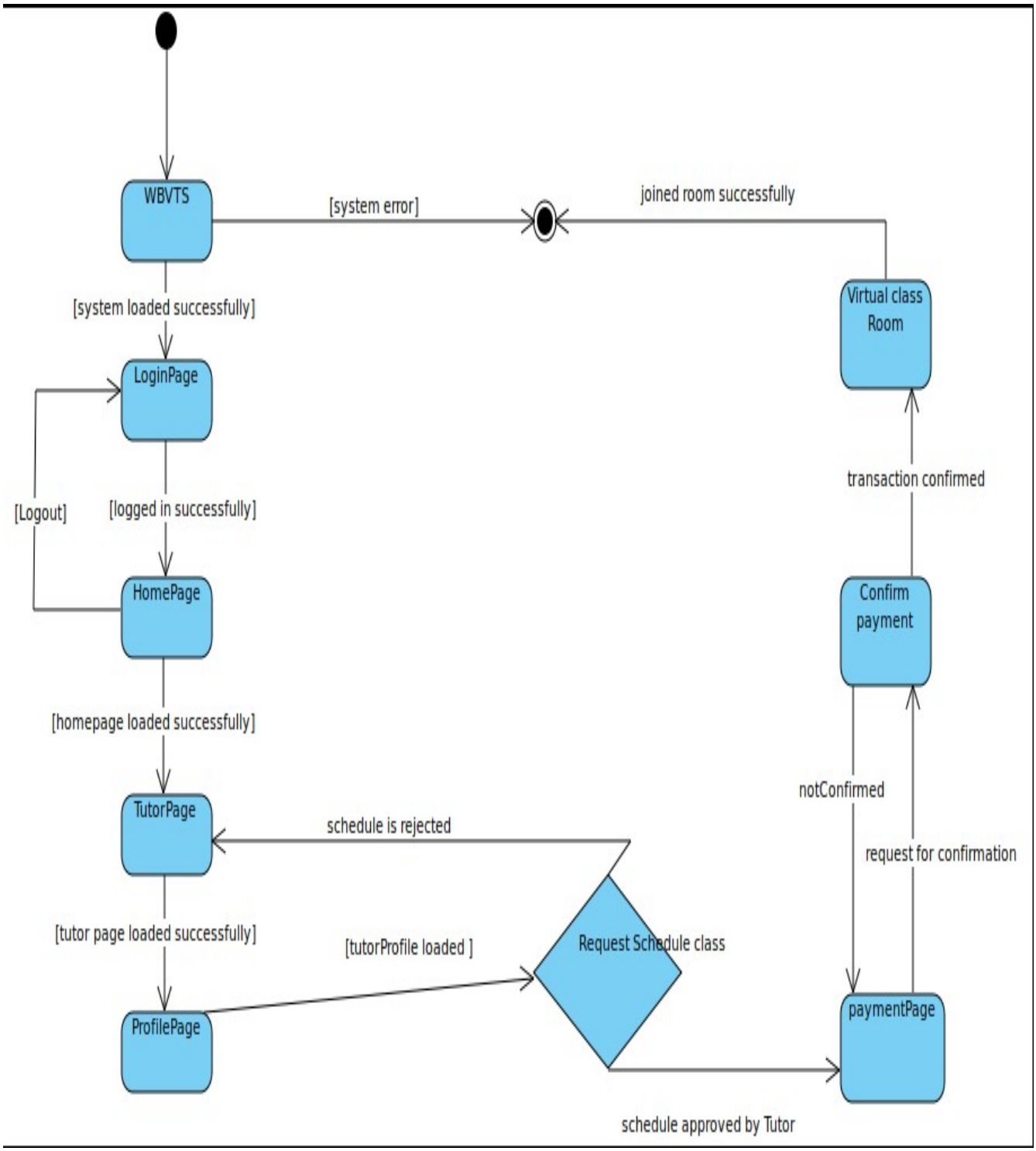


Figure 17: State chart diagram for accept or reject schedule

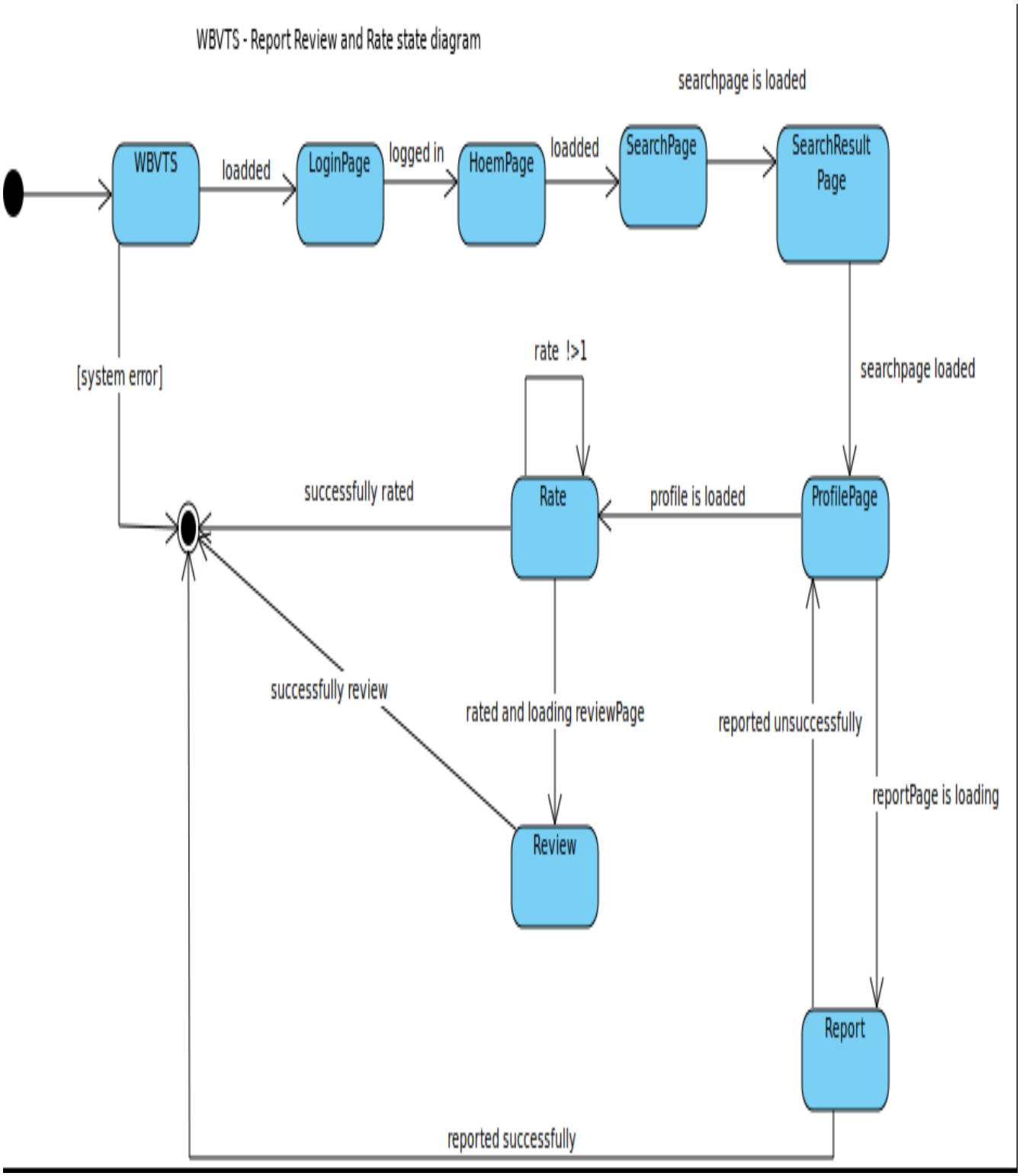


Figure 18: Rate and Review state chart diagram

CHAPTER FOUR

4. System Design

4.1. Overview

This is a system Design document to Virtual Class and Tutorial System. The document includes the design goal, the proposed system design and object design. Systems design is the process of defining elements of a system like modules, architecture, components, and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing, and designing systems that satisfy the specific needs and requirements of a business or organization. In this chapter, we will discuss all related variables of the system we used for modeling to transform into a structure of the system that is defined by a consistent set of rules and definitions.
In this chapter all software requirement specification document is converted into a format that can be implemented and decides how the system will operate. The architecture of the system, Database Design, and User interface design are the main focus of this chapter.

4.1.1 Purpose of the system design

The purpose of this designing is to show the direction how the system is built and to obtain clear and enough information needed to drive the actual implementation of the system. This document describes the design issues of the overall system. The objectives of design are to model the system with high quality so that it could be easy to make a change to it.

4.1.2 Design Goal

Design goals describe the qualities of the system that developers should optimize. The design goals are derived from the non-functional requirements of the RAD.

- **Reliability**: our system is designed as secure against human error, deliberate misuse or machine failure, and in which data will be stored without corruption.
- **Availability**: since the system is an online, it will be accessible through the working hours of the users, unless some problems happened like connection failure, power failure and other.
- **Maintainable**: the system design is easy to maintain and this reduces the system's maintenance costs, which usually represent a high proportion of the total lifetime cost of the system.
- **Secure**: In order to protect the confidentiality of the data, It may be important to build in methods to restrict access to authorized users only. The database can only be accessed the user with user name and password.
- **Performance:** the response time to users request should be as short as possible.
- **Cost-effective**: This is about designing system that delivers the required functionality, ease of use, reliable, secured, etc., to the user in the most cost-effective way. The system provides the service with lower cost than the current system.

- **Scalability** - The system adapts well to increasing data or added features or number of users.

4.2 proposed system architecture

Figure 19: System architecture

4.2.1 System process

System Activity Diagram

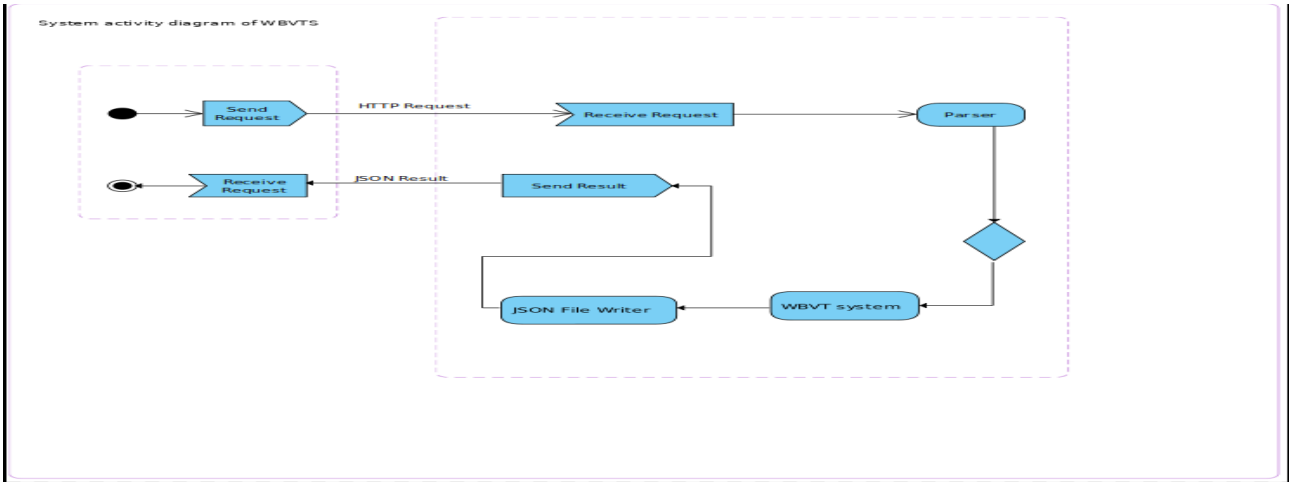


Figure 20: System activity diagram

System Process

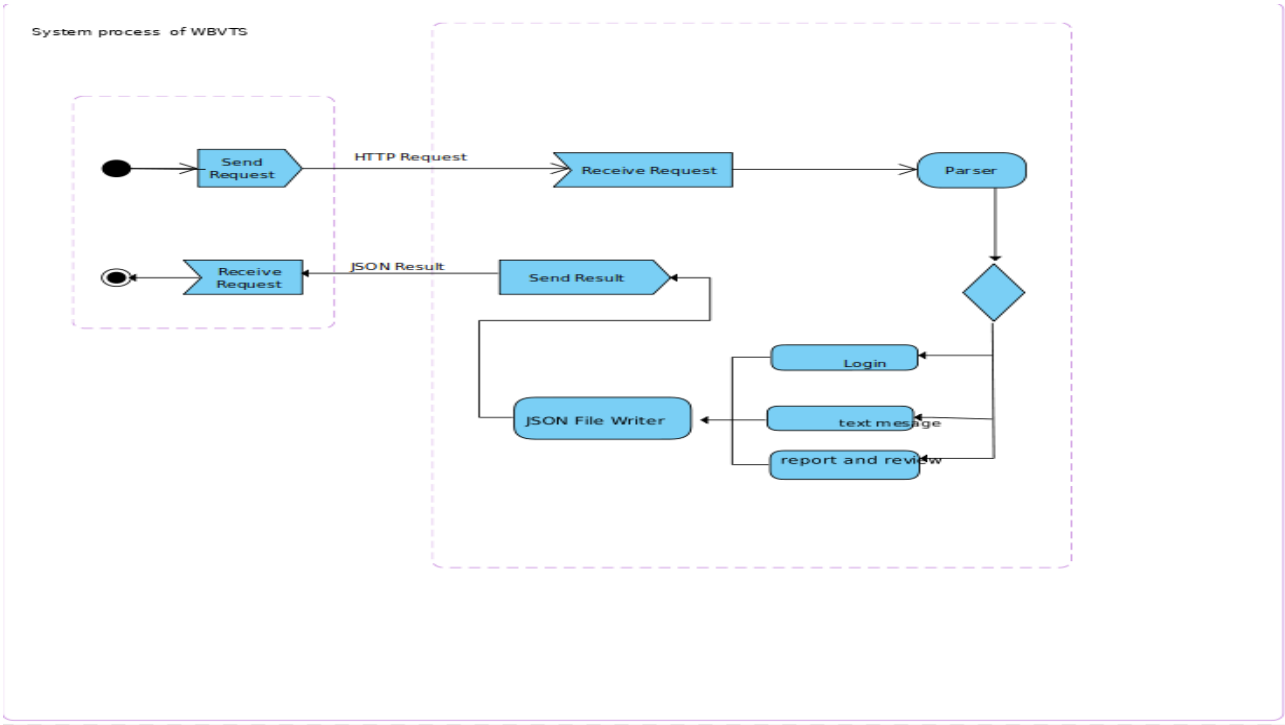


Figure 21: System process

4.2.2 Subsystem decomposition

System decomposition begins by decomposing the system into cohesive, well-defined subsystems. Subsystems are then decomposed into cohesive, well-defined components. Our system have four subsystems Registration,Virtual class room, Quiz and Online payment subsystems.

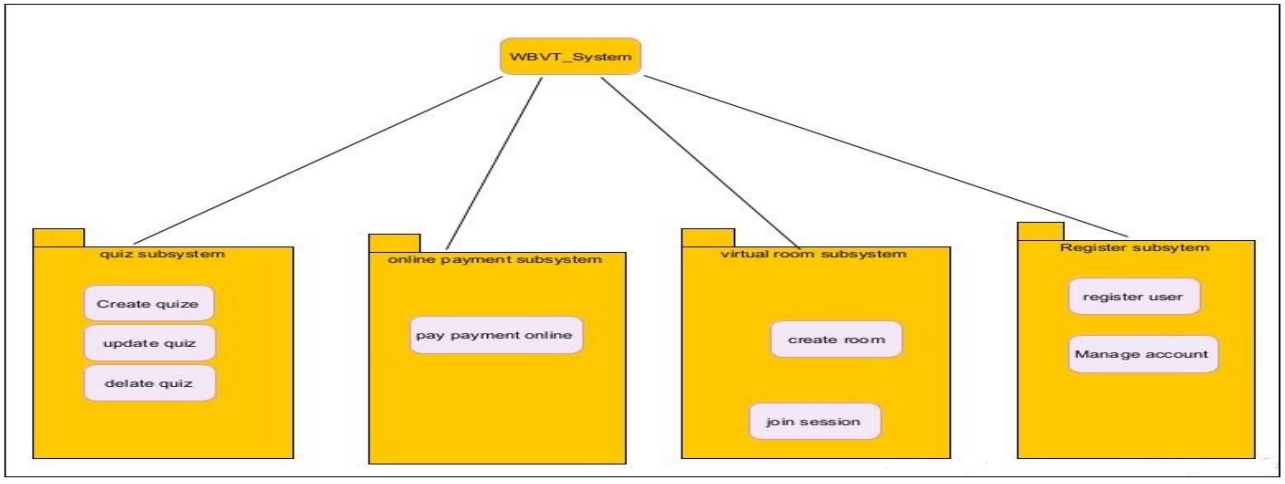


Figure 22: Subsystem decomposition diagram

Subsystem	purpose	class
Quiz Subsystem	Responsible for manage quiz	• Create quiz • Update quiz • Delate quiz
Online payment Subsystem	Responsible for make a payment in online.	Online payment
Virtual Room Subsystem	Responsible for create a link for sending to a student.	• Create room • Join session
Registration Subsystem	Responsible to manage account and register users.	Register user Manage account

Table 32: Subsystem decomposition

4.2.3 Hardware / Software mapping

The following information describes the technologies, techniques and scripting languages used for the development of the web application. Our system will be a web-based system which is available for online users. The system will run over all operating systems. The necessary technologies needed for the web application design, content and presentation semantics are HTML and CSS. JavaScript technologies are used for the client-side scripts. The server-side script is Node.js and the Database Management System is MongoDB. All the components and technologies are running on an Node.js Server. The system consists of three independent components: Web browser, Web Server and Database Server.

4.2.4. Persistent data management

The purpose of this section is to show the mapping of the objects/classes of the system, identified during the analysis stage, in to the corresponding database.

Figure 23: persistent data management for user

Figure 24: Persistent data management for room

Figure 25: Persistent data management for quiz

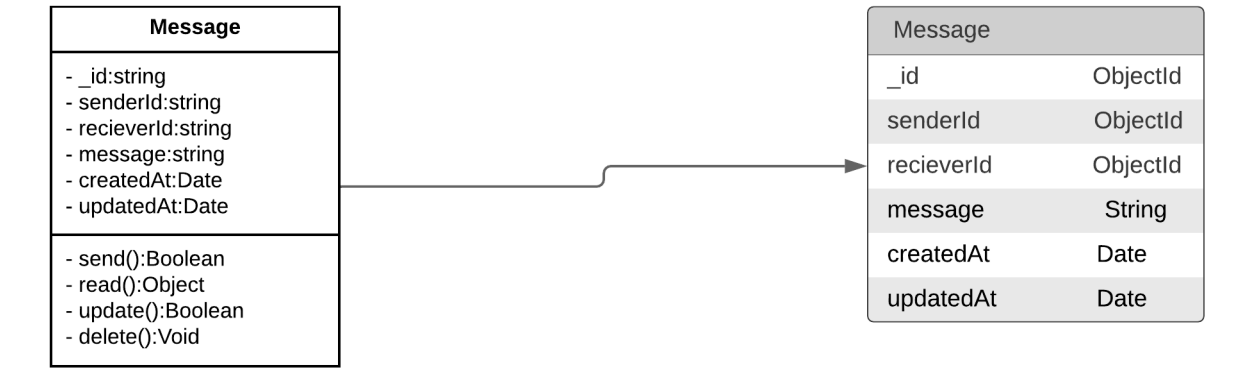


Figure 26: Persistent data management for message

Figure 27: Persistent data management for compliant

Figure 28: Persistent data management for review

Figure 29: Persistent data management for payment

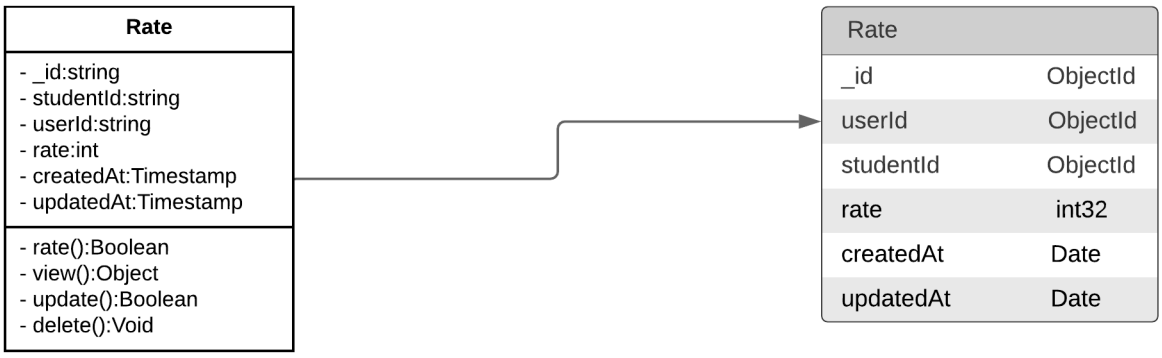


Figure 30: Persistent data management for rate

4.2.5 Component diagram

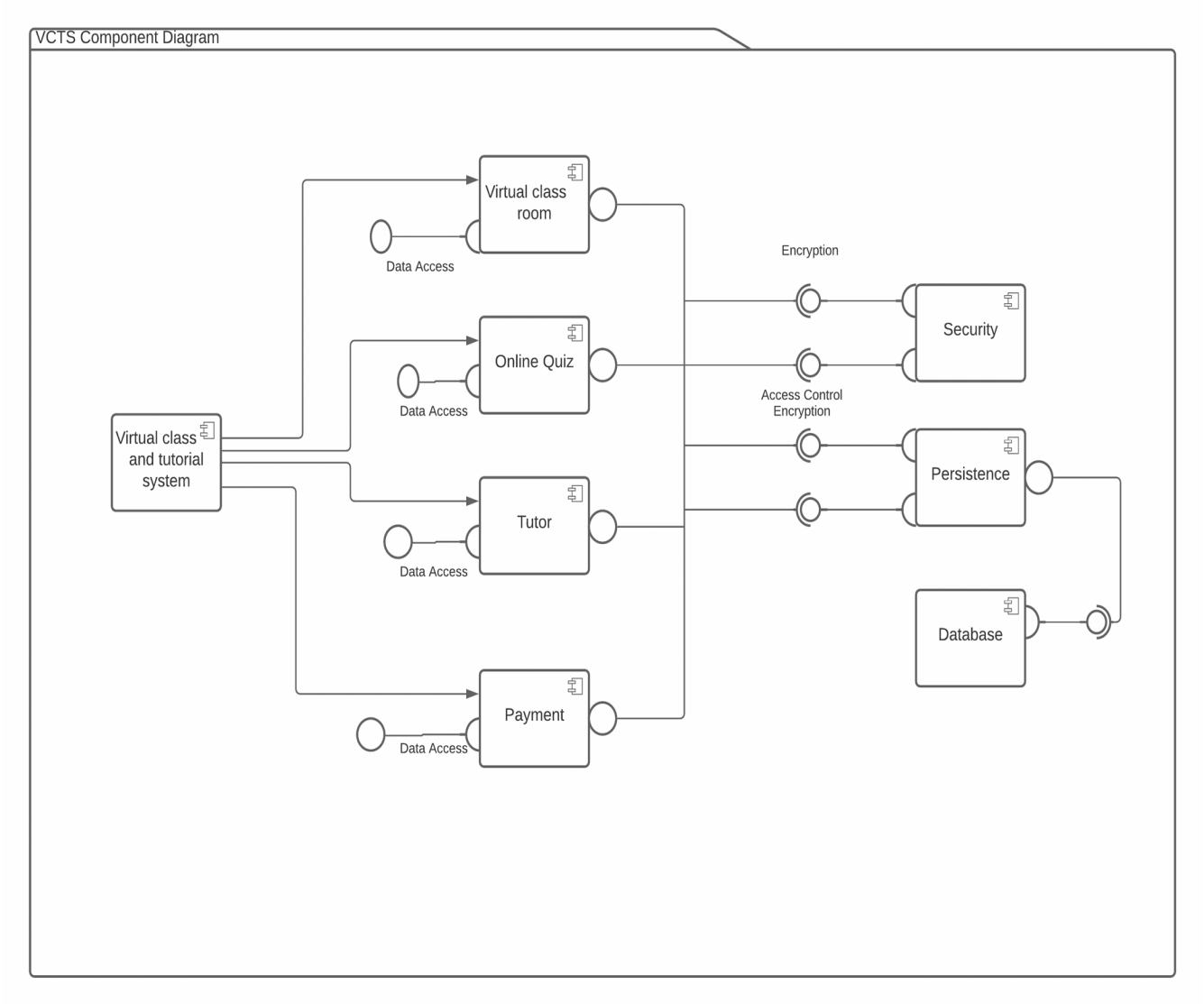


Figure 31: Component diagram

4.2.6 Database design

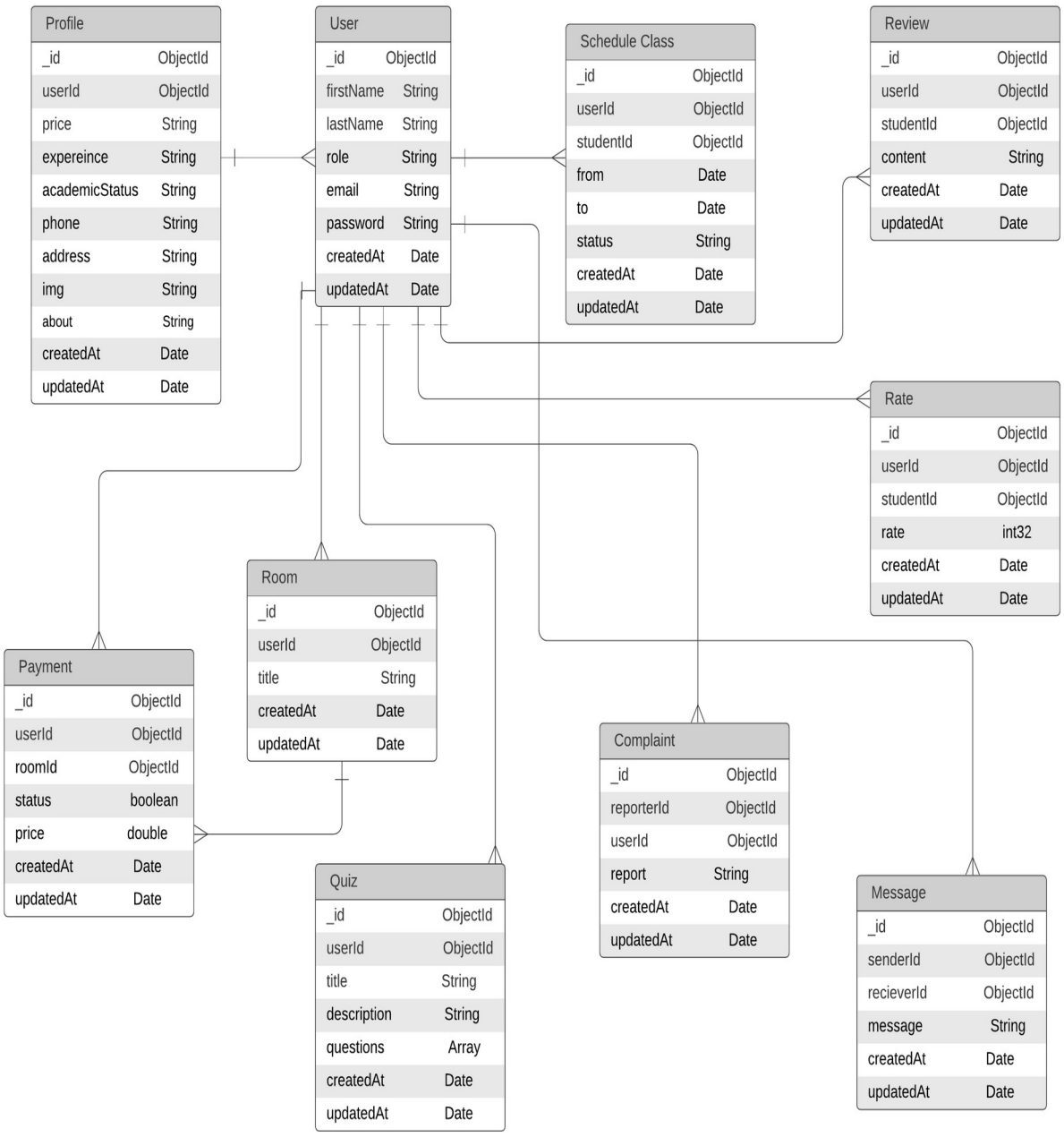


Figure 32: Database deign

4.2.7 Access Control

Actor	Permission to do
Admin	Manage users
	Manage payment
	Read complaints
Tutor	Create profile
	Approve schedules
	Create quiz
	Create room
	View rating
	View review
Company	Create profile
	Create quiz
	Create room
Student	Create profile
	Schedule tutorial
	Take quiz
	Report complaint
	Give rating
	Write review

Table 33: Access control

4.2.8 User interface design

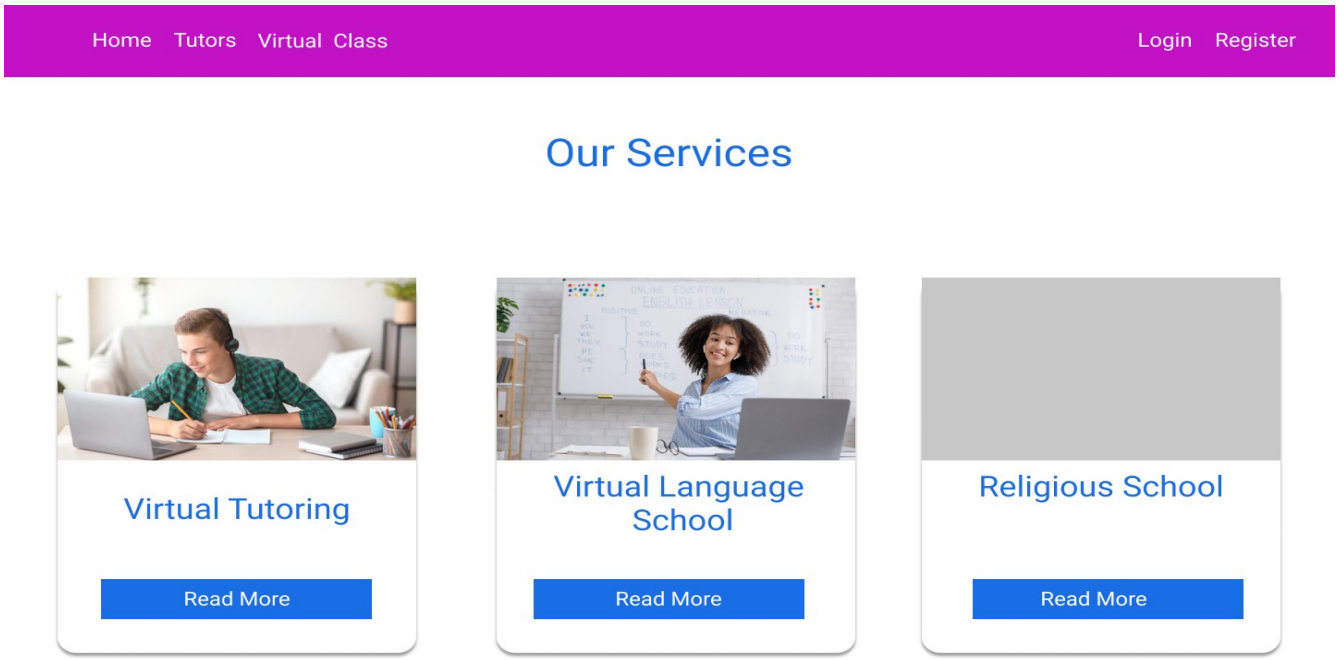
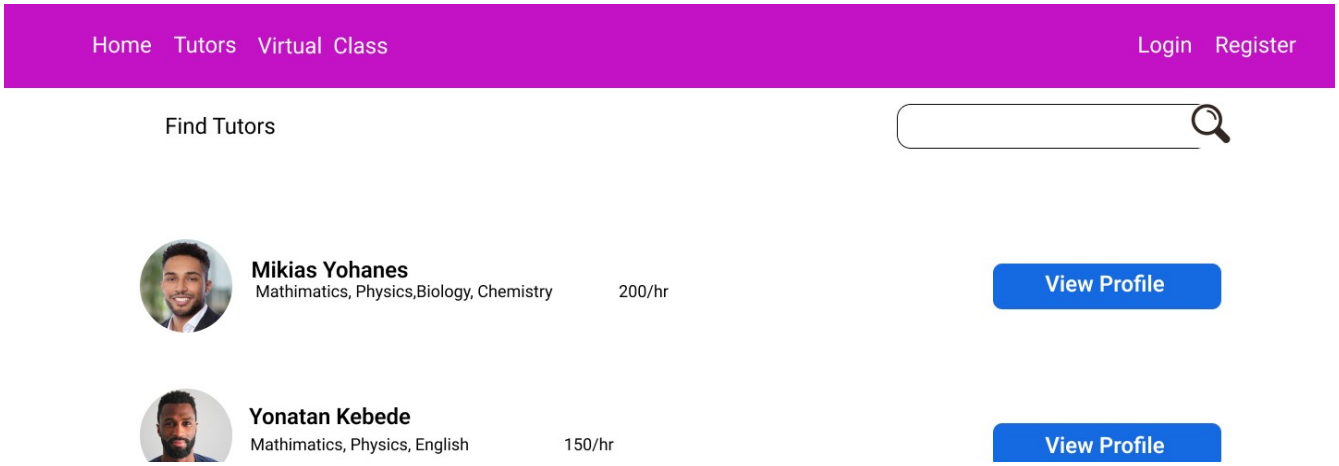


Figure 33: User interface design for home page



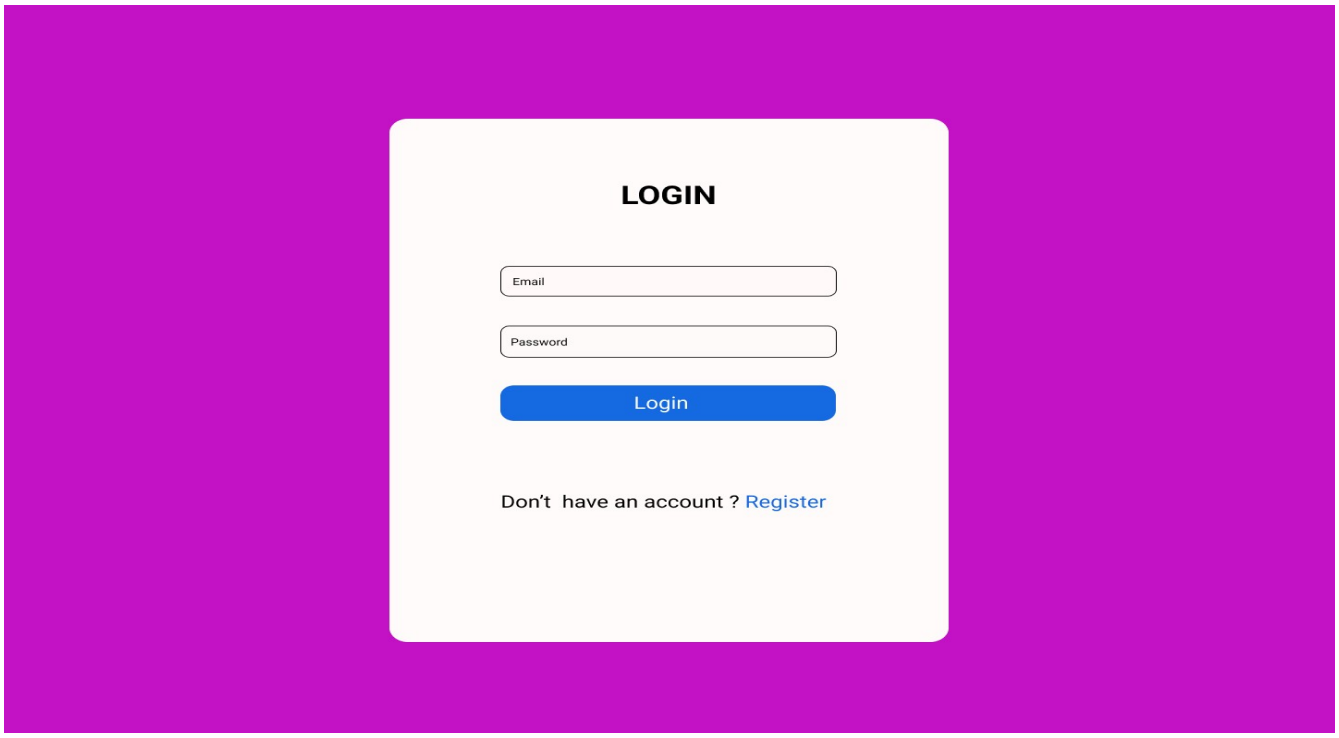


Figure 35: User interface design for login page

5. Chapter Five

5.1 Overview

Implementation is the core process of integrating and developing the system functionality. It includes the development, installation in order to do the actual project work to produce deliverable. The implementation phase involves putting our project plan into action and we are going to do this by implementing both functional and non functional requirements.

5.2 Coding Standards

In order to develop the proposed system, we have used coding standards that forces us to work together uniformly and have consistency throughout our development.
The following are the coding standards we used to develop the proposed system.

Standards for Naming

In order to develop our proposed system, we should follow a common name standard throughout our codes. While we are naming variables, class, modules or methods we should use a proper name standard. The name standards we use are as follows: -

- The name we use should be meaningful and tell why they exist.
- The name should be short and pronounceable.
- Names for variables and methods should be lower case letters.
- Names for classes should start with Upper case.
- Numbers should not be used as naming.

Standards for Commenting

While coding we should also use standards for commenting a portion of our codes. So, the following are the standards we use for commenting while we code: -

- Comments should be short and should be informative.
- Comments should describe why the code is used not how it works.

Standards for Indentations

Node.js is not indentation sensitive. However, it is important to understand the code So, we should use proper indentation and spacing while we code.

Standard for methods and modules

Node.js have a good guide how to use methods and modules starting from naming to how to import them and while we are coding, we would be using the Node.js standards for the methods and modules.

5.3 Development Tools

•**VS Code:** is an open-source code editing program built by Microsoft. It runs everywhere and allows you to do anything from debugging code to inputting Git commands or creating Sass code.

•**Node.js:** is an open-source, cross-platform, back-end JavaScript run time environment.

•**Express.js:** or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License.

•**MongoDB:** is a source-available cross-platform document-oriented database program.

•**React.js:** React is a free and open-source front-end JavaScript library for building user interfaces based on UI components.

•**GitHub:** is a provider of Internet hosting for software development and version control using Git.

•**Git:** Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development.

5.4 Prototype

prototype is a sample system which is similar to the proposed one but it doesn't mean it's fully functional. For instance, on our proposed system we designed a prototype which mainly do only functional the CRUD operations which are Create, Read, Update and Delete. The prototype consists of the frontend of the system which mainly includes the user interface part which is developed using Express and React template.

Event if the prototype is not fully functional, we have developed the database and developed the connection to it through Node.js framework . We used MongoDB database as our database server and we have made the connection tot the node.js framework.

Frontend/ Client Side:- this is part of of the system that the users interact with it. The client side for our prototype is React and Bootstrap templates. Since node.js supports MVC(model view controller) architecture. For node.js architecture we use React and bootstrap as front end where we design the user interface.

Backend/ Server Side :- backend is core part of the system where routing of URLs and busiiness logic takes place. In case, we used Node.js as our backend and MongoDB as database. Node.js is one of the most popular JavaScript framework that enables us to develop fast and clean code.

5.5 Implementation Detail

We will use MERN stack to develop our system. MERN stack is 3-tier architectural pattern. Including the front-end display tier (React.js), application tier(Express.js and Node.js) and the database tier (MongoDB). But MVC is frequently used architecture MVC so we will follow the MVC architecture. We can achieve it by sticking to server-side templating in Node, and just sending the HTML to the browser and treating the browser as the View layer i.e. doing things the way things have always been done forever.

- **Mongoose'** models defines the data part. This is where we will store all of the crucial data our application needs to function.
- **Express & Node.js** does all the functional programming and will be used to write the Business Logic Tier (controller). This tier represents the Application Server that will act as a bridge of communication for the Client and Database. This tier will serve the React components to the user's device and accept HTTP requests from the user and follow with the appropriate response.
- **React** serves as the "**V**" in the **MVC.** Our Client tier (View) will be written in JavaScript, HTML, and CSS, using **ReactJS** as the framework. This level of the architecture is what the user will interact with to access the features of our application**.**