CSN-254

Group:15

DESIGN DOCUMENT

Team Members :

N Suresh Kumar -21114066
Nimmala Sai Bhargav-21114067
Shashi Praneeth -21114089
Shyam Sundar -21114102
Yashwanth krishna -21114093
K P V ANIL -21114048

QUIZZ

28/03/2023

Version 1.0

SUMMARY:

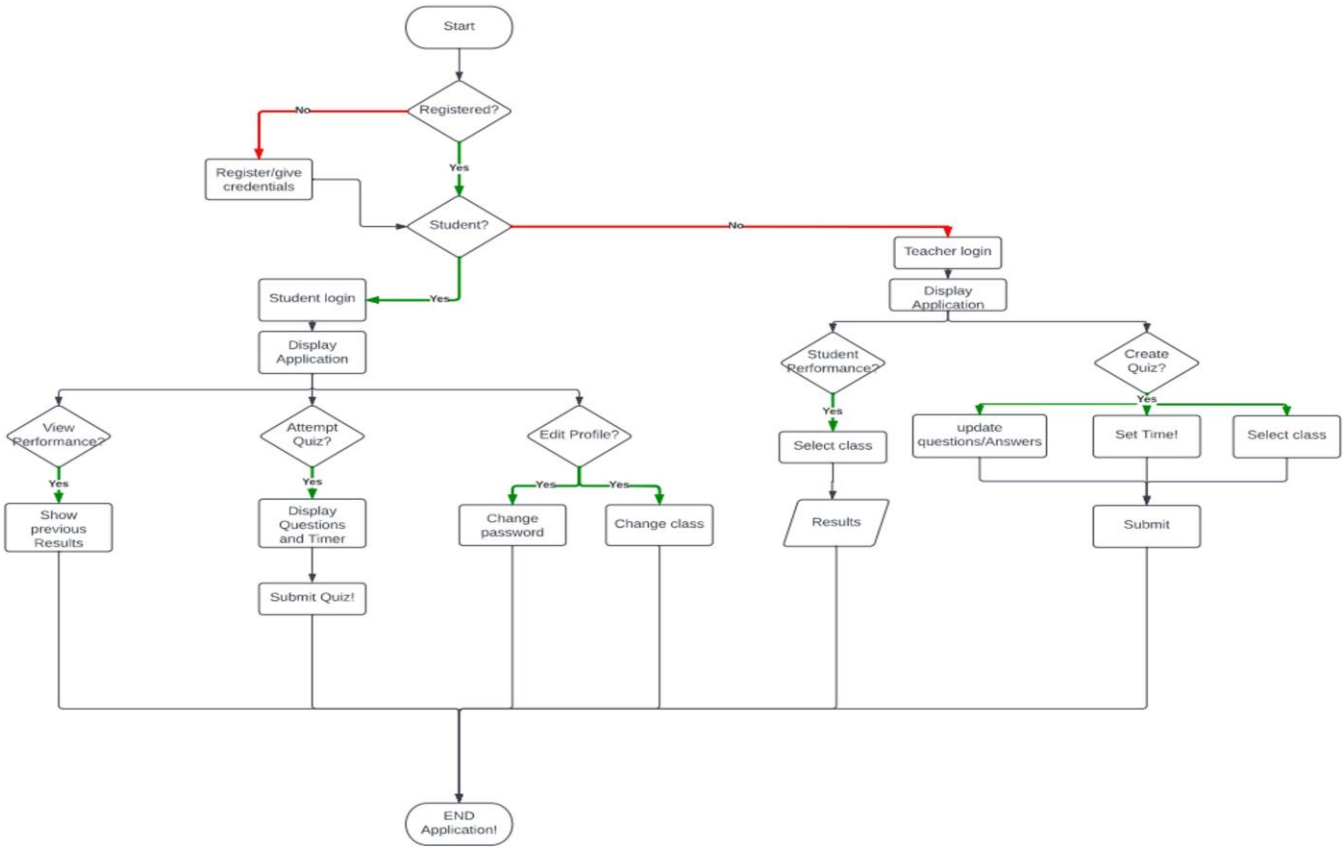
The entire design of our program "QUIZZ" is covered in this document. In this document, we'll go over how our application Is designed in terms of its user interface, underlying implementation, non-functional needs and Also describe the primary goal of our program, assist users in using it, andso forth.

Our application can be used to conduct various types of Quizzes and examsTwo type users are present one would conduct the test and other attemptIt like a teacher and student. It helps to make validation of quiz easy and toStore all results at one place for teacher and to check paper and observing Results over a large range of exams for student.

High Level Design:

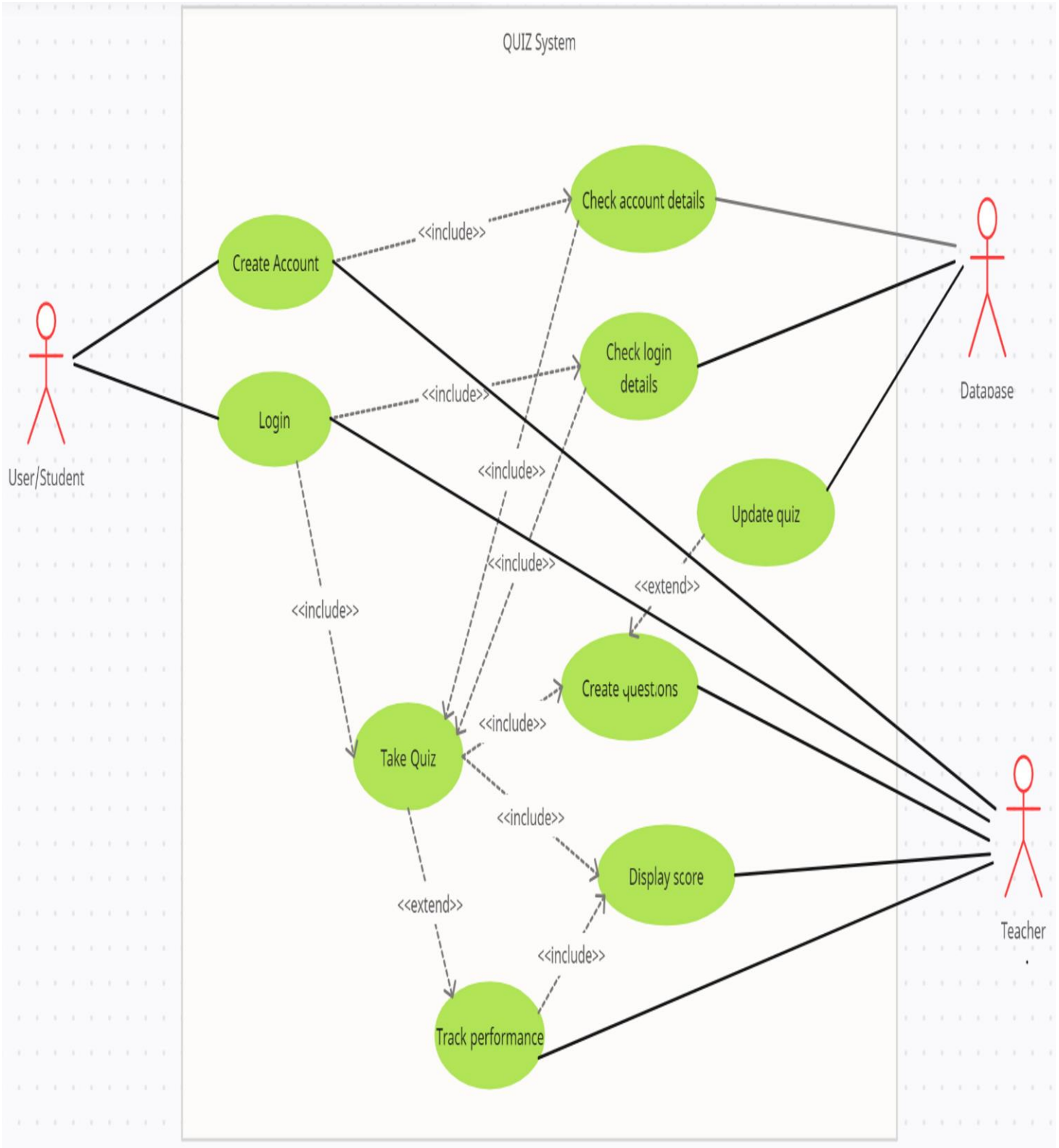
Flow Chart:

The below flow-chart is a representation of our app features in a simple user-understandable manner.



Use Case diagram:

Below is the Use case diagram and its text description for our software.



TEXT DESCRIPTION

U1: ***Register*** *: Using this use case, the user can register into the application by providing the respective details.*

Scenario 1: Mainline sequence

1. *User: Select 'Register' option.*
2. *System: Display Prompt to Select Student/Teacher.*
3. *User: Select Student/Teacher.*
4. *System: Display prompt to enter username, mail and password.*
5. *User: Enter the required details.*
6. *System: Display the user's home page.*

Scenario 2: At step 6 of mainline sequence

6 : System: Displays the message that the login details already exist.

*U2:***Login** *: Using this use case, the user can login into the application by providing the respective details.*

Scenario 1: Mainline sequence

1. *User: Select 'Login' option.*
2. *System: Display Prompt to Select Student/Teacher.*
3. *User: Select Student/Teacher.*
4. *System: Display prompt to enter mail and password.*
5. *User: Enter the required details.*
6. *System: Display the user's home page.*

Scenario 2: At step 6 of mainline sequence

6 : System: Displays the message that the login details are invalid.

U3: ***View Profile*** *: Using this use case, the user can view his personal details.*

Scenario 1: Mainline sequence

1. *User: Select 'View Profile' option.*
2. *System: Displays the personal details of the users.*
3. *User: Selects 'Edit Profile' option.*
4. *System: Displays the info that can be edited.*
5. *User: Modifies the info that is wrong.*
6. *System: Displays a message showing "Updated info successfully".*

Scenario 2: At step 3 of mainline sequence

3. *User: Selects logout option.*
4. *System: Logs out the user out of the application.*

*U4:**Create quiz** : Using this use case , user(teacher/authorities) can create a new quiz for students/staff.*

Scenario 1: Mainline sequence

1.User :Opens the quiz application.
2.System : Displays the login portal.
3. User : Logins by entering the login credentials.
4. System :Displays the teacher portal .
5.User : Selects create quiz option.
6.System : Displays options for adding question of different types like multiple choices, single choice, true/ false type etc, set the time, select the class.
7.User : selects add questions option and adds all the questions, answers and marks for then.
8.System : Ask for save the questions and answers.
9.User :Clicks the save button.
10.User : selects option to set timer.
11 : System : Display a dialogue box to set the timer.
12.User : sets the start time and finish time of the quiz.
13. System : Saves the timer.
14 .User : selects the select the class option.
15.System : shows the place to add the classes.
16.User : Add the classes names for which the quiz being conducted.
17.User : Starts the quiz for students to enter and attempt for the quiz..

*U5:**Attempt quiz** : Using this use case user(student) can take quiz created by his authorities.*

Scenario 1: Mainline sequence

1. *User: clicks attempt quiz*
2. *System: checks whether user is allowed to take quiz and shows a pop up of ready to take quiz!*
3. *User: clicks "yes"*
4. *System: shows quiz portal, timer is on.*
5. *User: choose option and clicks save and next ques.*
6. *System: shows a submit quiz option.*
7. *User: clicks submit button.*
8. *System: pops up "are you sure to end quiz"*
9. *User: clicks Yes*
10. *System: shows a page containing "you have submitted quiz".*

Scenario 2: At step 9 of Mainline Sequence

9. *User: clicks No*
10. *System: Back to quiz portal.*

*U6: **Student Performance** : Using this use case user(Teacher/Authorities) can view Student results of a particular quiz.*

Scenario 1: Mainline Sequence

1. *User: Clicks on Student Performance.*
2. *System: prompts a page with the quizzes User created.*
3. *User: Select particular quiz he want to view results.*
4. *System: Shows the results of the quiz selected.*

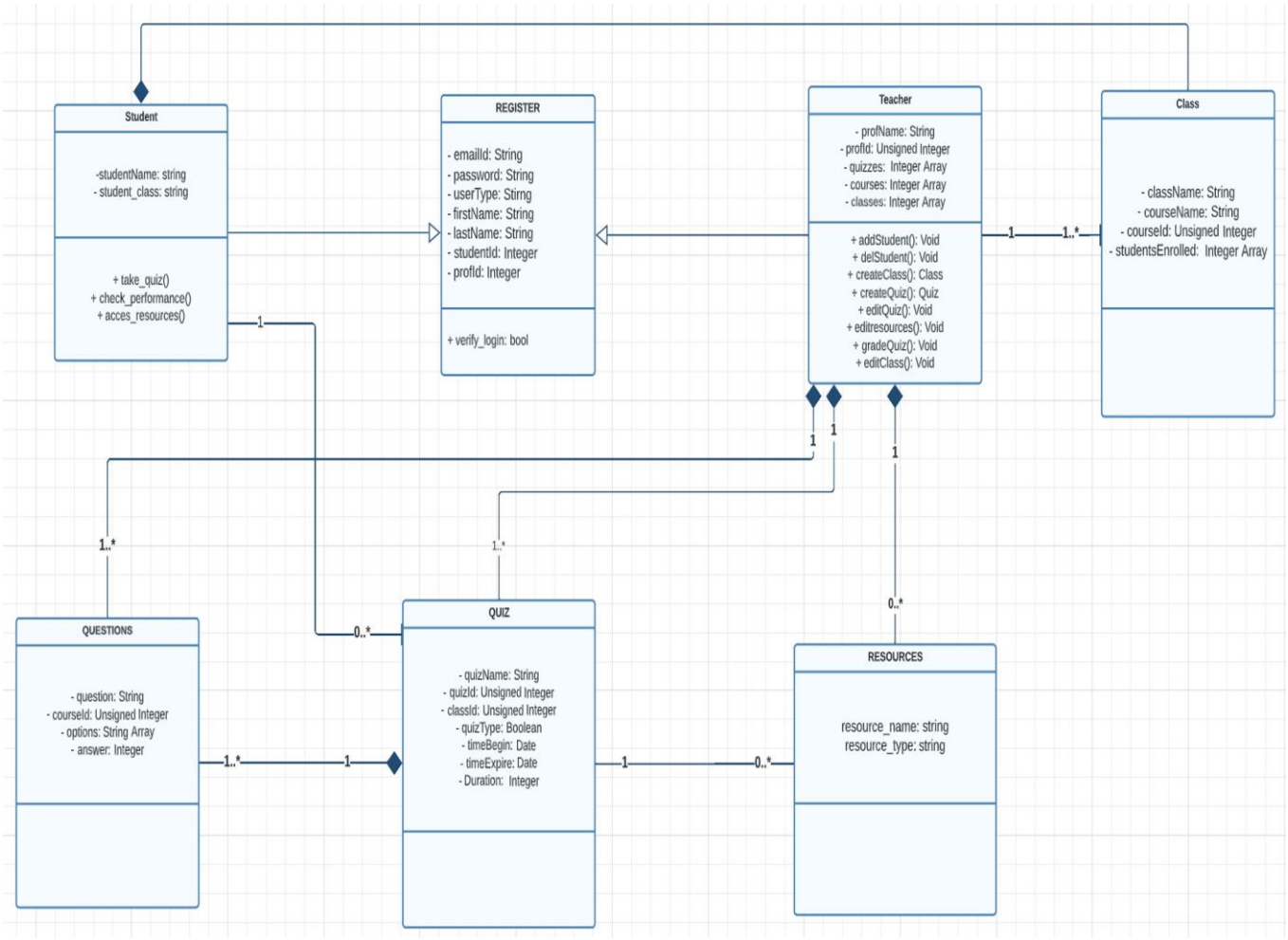
U7: ***Previous results*** *: Using this use case user(student) can view his results of Previous quizzes he attempted.*

Scenario 1: Mainline Sequence

1. *User: Clicks on previous results.*
2. *System: Prompts a page with the quizzes User attempted.*
3. *User: Select particular quiz he want to view results.*
4. *System: Shows the results of the quiz selected.*

CLASS DIAGRAM :

Below is the class diagram of our software.



Classes, Member variables and Member functions:

User Register:

Attributes:

- email Id: String
- password: String
- User Type: String
- First Name: String
- Last Name: String
- Student Id: Integer
- prof Id: Integer

Functions:

+ verify_login: bool

STUDENT:

Attributes:

-Student Name: string
- student_class: string

Functions:

+ take_quiz()
+ check_performance()
+ acces_resources()

TEACHER:

Attributes:

- profName: String
- profId: Integer
- quizzes: Integer Array
- courses:Integer Array
- classes: Integer Array

Functions:

+ addStudent(): Void
+ delStudent(): Void
+ createClass(): Class
+ createQuiz(): Quiz
+ editQuiz(): Void
+ editresources(): Void
+ gradeQuiz(): Void
+ editClass(): Void

CLASS:
Attributes:
- className: String
- courseName: String
- courseId: Unsigned Integer
- studentsEnrolled: Integer Array

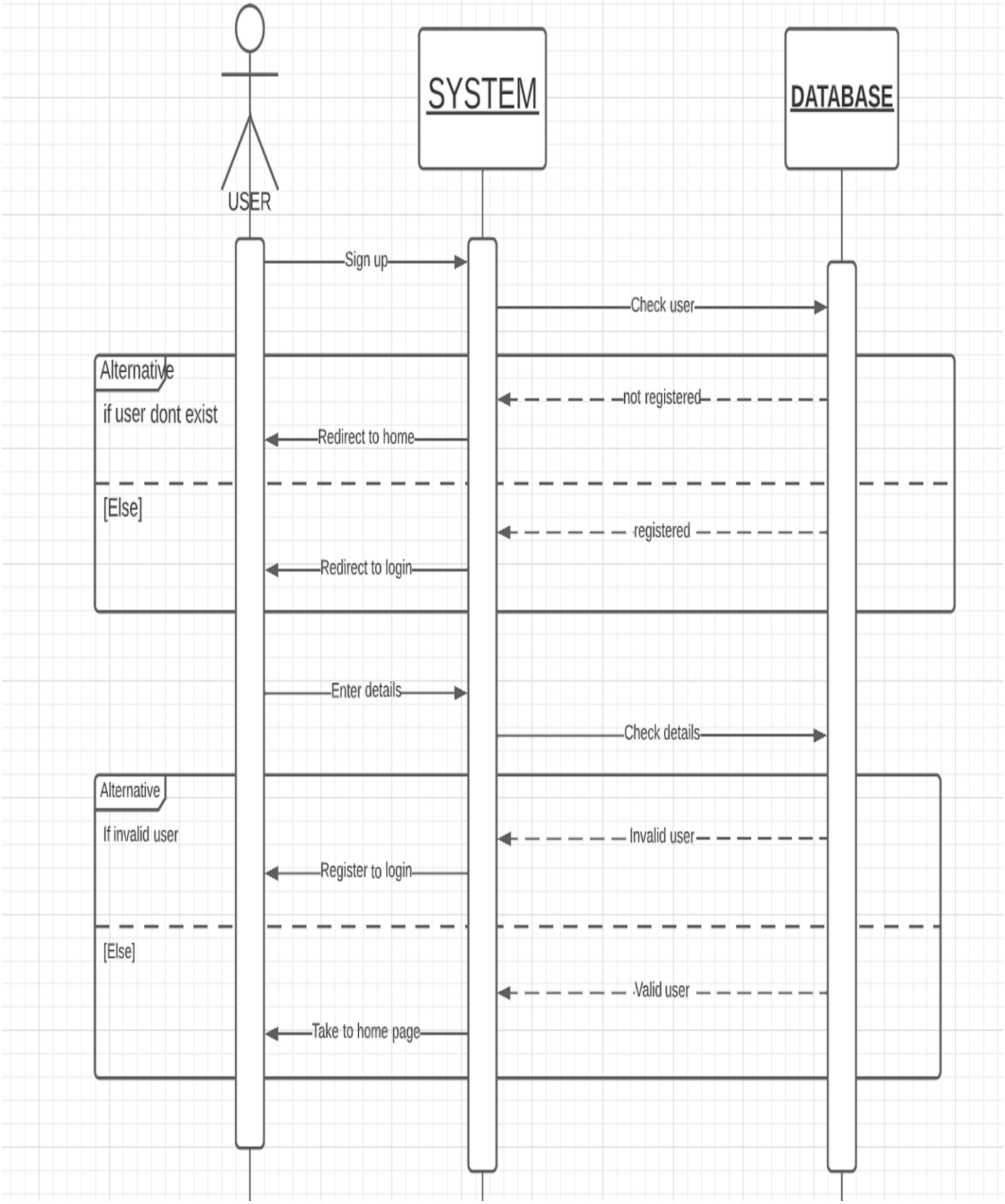
QUIZ:
Attributes:
- quizName: String
- quizId: Unsigned Integer
- classId: Unsigned Integer
- quizType: Boolean
- timeBegin: Date
- timeExpire: Date
- Duration: Integer

QUESTIONS:
Attributes:
- question: String
- courseId: Unsigned Integer
- options: String Array
- answer: Integer

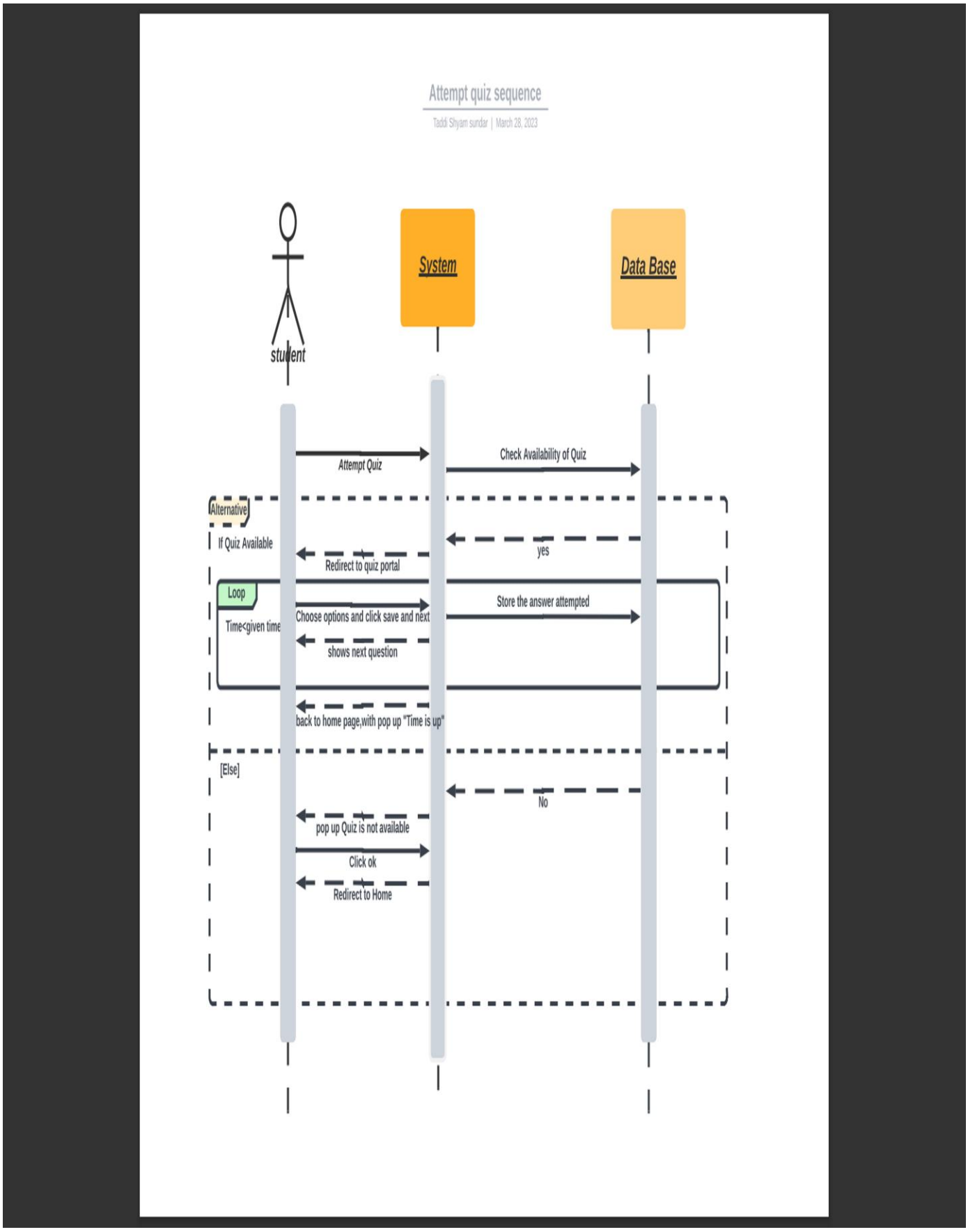
RESOURCES:
Attributes:
resource_name: string
resource_type: string

Sequential Diagrams:

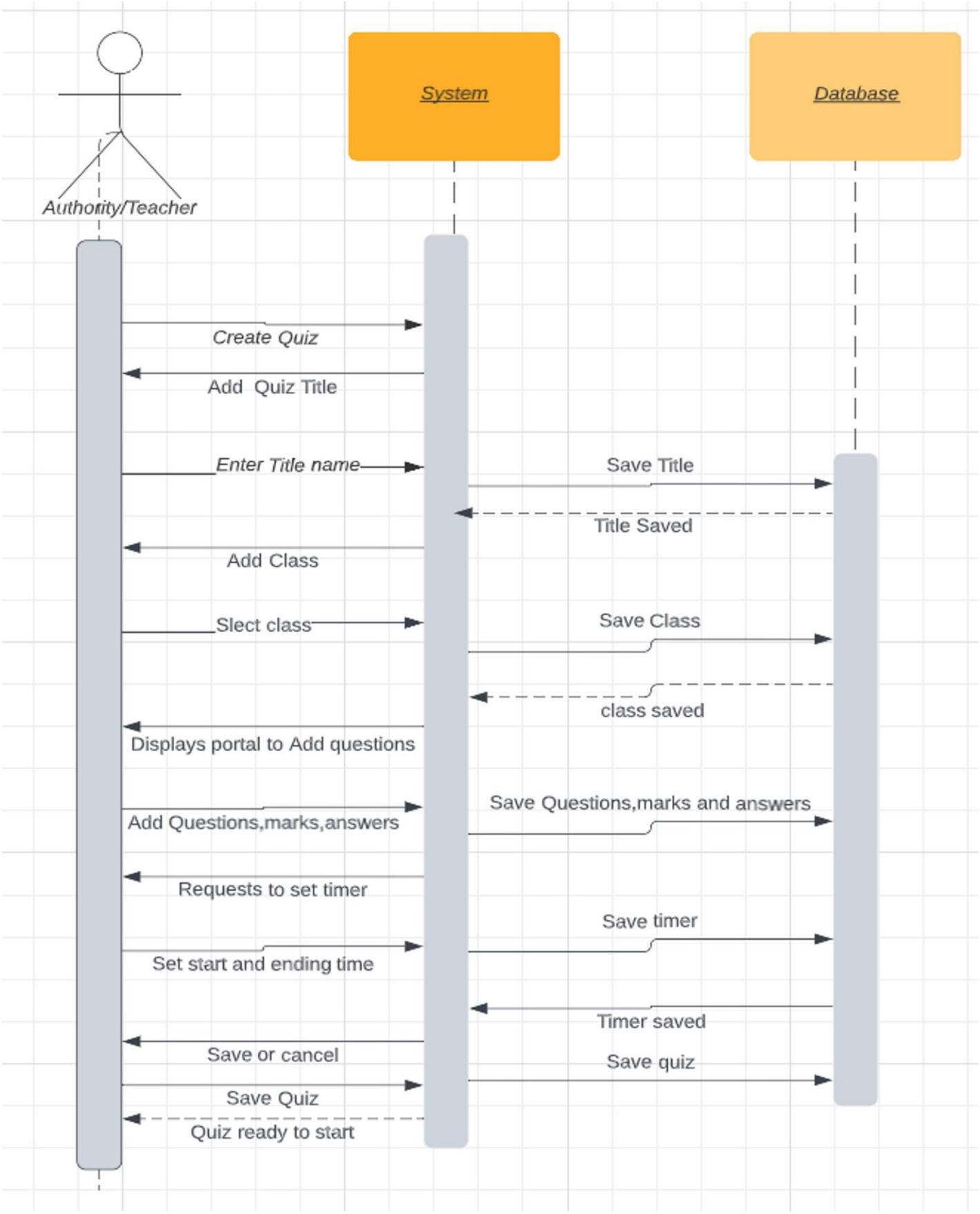
- ***Sequence Diagram-1: Login***



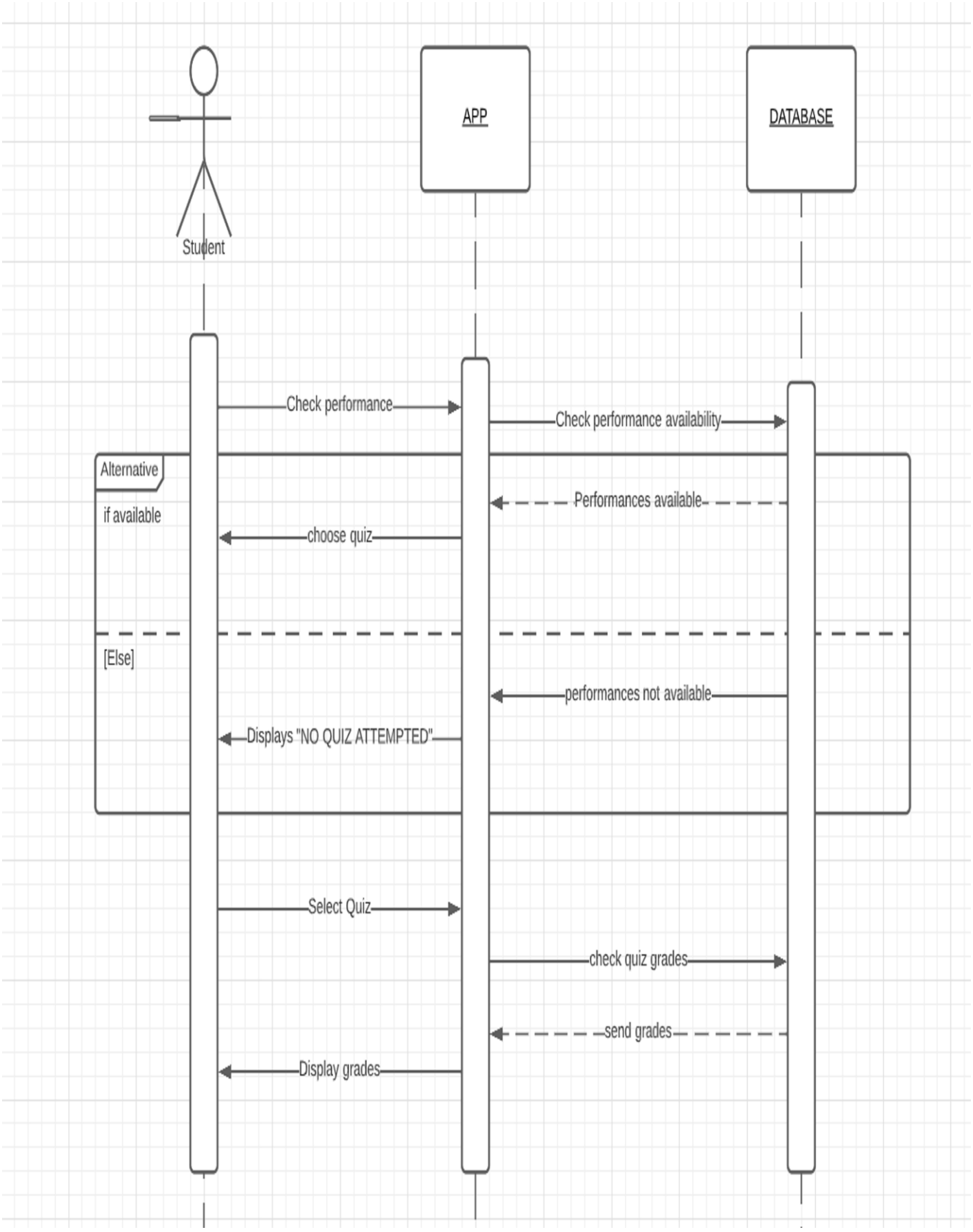
Sequence Diagram-2: Attempt Quiz



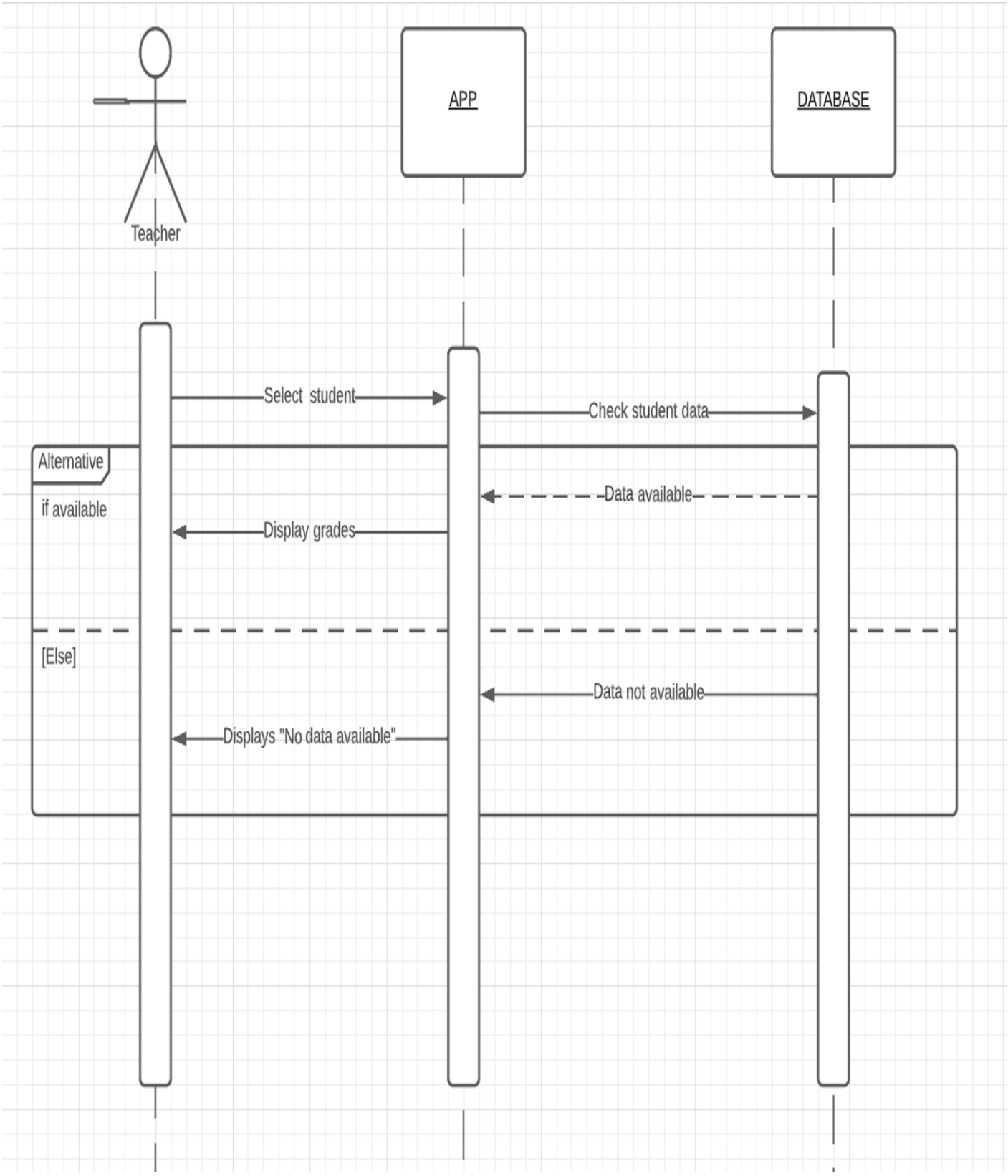
Sequence Diagram-3: Create Quiz



- ***Sequence Diagram-4: Performance (Student)***


Student
APP
DATABASE
Check performance
Check performance availability
Alternative
if available
Performances available
choose quiz
[Else]
performances not available
Displays "NO QUIZ ATTEMPTED"
Select Quiz
check quiz grades
send grades
Display grades

- ***Sequence Diagram-5: Track Performance (Authorities)***



Low Level Design:

User Authentication:

1. User Registration:

. let user select either teacher / student portal and Get the user details from the form.

· If the email ID already exists in the database, then pop up a message showing "account with this email ID already exists".

· If the username already exists in the database, then pop up a message showing "account with this username already exists".

· If both username and email ID are new then create a new object of user class using the details provided by the user and save the data of the user in the database.

· Pop up a message showing "Account created successfully" and redirect to the home page.

2. User Login:

. let user select either teacher / student portal and get the user details from the form.

· Verify username and password entered by user.

· If details entered by the users are correct then the user is redirected to the home page of Quizz App.

· Otherwise, it raises an error and displays "Invalid Credentials!" message and the user are redirected to the login page.

3. Forgot Password:

· If the user forgets the password, he can click on the forgot password.

· System asks the user to enter the email ID or username through which he has registered.

· Now the system sends an OTP to his registered email ID, with the help of this the user can reset the password.

4. View Profile:

. On the Home Page if the user needs to see his profile , he can click on "View profile" option .

. It displays the personal details of the users.

. There will be a 'Edit Profile' option which on selecting Displays the info that can be edited.

. After updating the info , Displays a message showing "Updated info successfully".

. There will be a 'Logout' option which Logs out the user out of the application and directs to registration page.

*5. **Create quiz:***

. Using this option, user(teacher/authorities) can create a new quiz for students/staff.

. After selecting this option , it displays options for adding question of different types like multiple choices, single choice, true/ false type etc , set the time, select the class.

. Saves the quiz with a fixed timer.

*6. **Attempt quiz:***

. Using this option in students home page , user(student) can take quiz created by his teacher.

. On clicking 'attempt quiz' checks whether user is allowed to take quiz and shows a pop up of ready to take quiz!.

. And the quiz starts after clicking 'yes' showing the time left and showing each question .

. After finishing , can click on 'submit' option then shows "you have submitted quiz" or after completion of time it automatically submits.

*7. **Quiz Performance :***

. Using this option for a particular quiz , the teacher can see the overall performance of all the students attempted the quiz.

*8. **Previous results:***

. using this option in students Home Page , the user can see hi/her previous quiz performances.

. on clicking , it Prompts a page with the quizzes User attempted. On Selecting particular quiz he/she wants to view results , it Shows the results of the quiz selected.

Questions Q.

Why write a design document ?

Ans. A design paper outlines our strategy for problem-solving. A design document's primary objective is to increase the software's efficiency through the use of the requirements gathered thus far and input from other people. By outlining the specifics of how the software should be constructed, this paper will be used to help with software development.

What does it generally consist of ?

Ans. Use case models, sequence diagrams, collaboration models, object behaviour models, and other related requirement information are included in the Software Design Document's narrative and graphical documentation of the software design for the project.

What main information does the user/customer get during this document review?

Ans. Customers can learn about the general scope of the software project, its design, and the implementation-stage methods used for functions like LOGIN, View Articles, Share with Friends, Text to Speech, Add Friend, and Accept Request. Flowcharts, class diagrams, use class diagrams, and sequence diagrams are examples of high level design that make it easier for the user to reach the primary functions (features) that the project is capable of

executing. The user can comprehend the software's functional algorithms by looking at the low-level architecture.

What are the tools Used to Create Diagrams?

Ans. The links to the flowcharts and diagrams that we created online using Lucid Chart are given below the diagrams.