PES University, Bengaluru

Department of Computer Science & Engineering

Session: Jan - May 2023

Subject: Object Oriented Analysis and Design with Java

Title: EXAM PORTAL

Team members:

Aishwarya Parthasarathi	***PES1UG20CS523***
Almas Banu	***PES1UG20CS535***
Anusha Achar	***PES1UG20CS539***
Brinda N	***PES1UG20CS542***

Project Description

Our project is an Exam Portal that allows registered users to take quizzes and view their results. The system provides a user-friendly interface for both students and admins. Users can register themselves on the portal and log in to take quizzes on various topics. The portal offers a wide range of categories of quizzes to choose from.

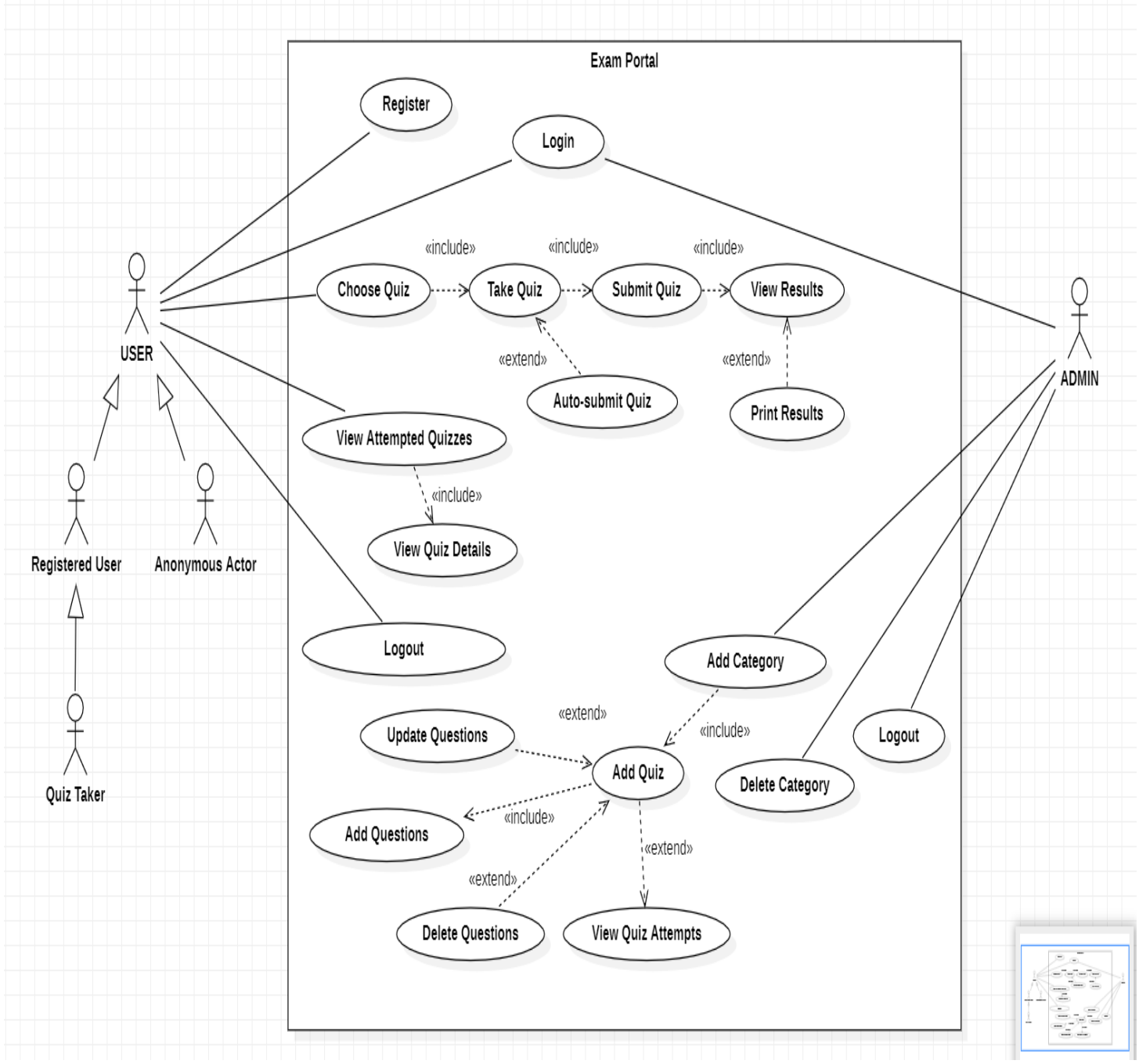
Once logged in, users can select a quiz, take it, and submit their answers before the timer is over. The system also allows users to view their results and provides an option to print them. Users can view their attempted quizzes and their details, such as date and time, category, and score.

The admin has exclusive rights to manage the quiz portal. The admin can log in using their credentials and can add new quiz categories, questions, view questions, update questions, and delete questions. Additionally, the admin can view who and when these quizzes are attempted by users.

The system also provides robust security features to protect user data and maintain confidentiality. All user data is stored securely and can only be accessed by authorized users.

Our Exam Portal aims to provide an efficient and reliable platform for conducting online quizzes and assessments. It is suitable for educational institutions, training centers, and other organizations that require a platform for conducting online assessments. With our Exam Portal, users can take quizzes from anywhere, anytime, and get instant results, , making learning convenient and fun.

I. Use Case Diagram:



Use Case Specification:

1.

Name: Register

Summary: A new user can register for an account on the Exam Portal.

Actors: Registered User

Preconditions: The user wants to use our exam portal.

Description:

- The user clicks on the "Register" button.
- The Exam Portal displays a registration form.
- The user fills out the form with their personal information and chooses a username and password.
- The user submits the form and the Exam Portal registers their account.

Exceptions:

- If the username is already taken, the Exam Portal displays an error message and prompts the user to choose a different username.

Post-condition: The user is registered with the Exam Portal and can log in with their username and password

2.

Name: Login

Summary: Registered users can log in to the Exam Portal.

Actors: Registered User

Preconditions: The user has registered with the Exam Portal.

Description:

- The user enters their username and password.
- The Exam Portal verifies the login credentials.
- If the credentials are correct, the Exam Portal logs the user in and displays their dashboard.

Exceptions:

- If the username or password is incorrect, the Exam Portal displays an error message and prompts the user to enter their credentials again.

Post-condition: The user is logged in to the Exam Portal.

3.

Name: Take Quiz

Summary: Registered users can take a quiz on the Exam Portal.

Actors: Registered User, Quiz Taker

Preconditions:

- The user is registered with the Exam Portal and has logged in.
- The user has selected a quiz to take.

Description:

- The user clicks on the "Take Quiz" button for the selected quiz.
- The Exam Portal displays the quiz questions and a timer.
- The user selects their answers for each question and submits the quiz before the timer runs out.
- The Exam Portal displays the quiz results, including the user's score and any correct/incorrect answers.
- The user can choose to print their results.

Exceptions:

- If the user does not submit the quiz before the timer runs out, the Exam Portal auto-submits the quiz and displays the results.

Post-conditions:

- The user can view their quiz results, including their score and any correct/incorrect answers.
- The user can view all quizzes they have attempted and their details, including the date and time.

4.

Name: Manage Quizzes

Summary: The admin can manage quizzes on the Exam Portal by adding, updating, or deleting quiz categories and questions.

Actors: Admin

Preconditions: The admin has logged in to the Exam Portal with their username and password.

Description:

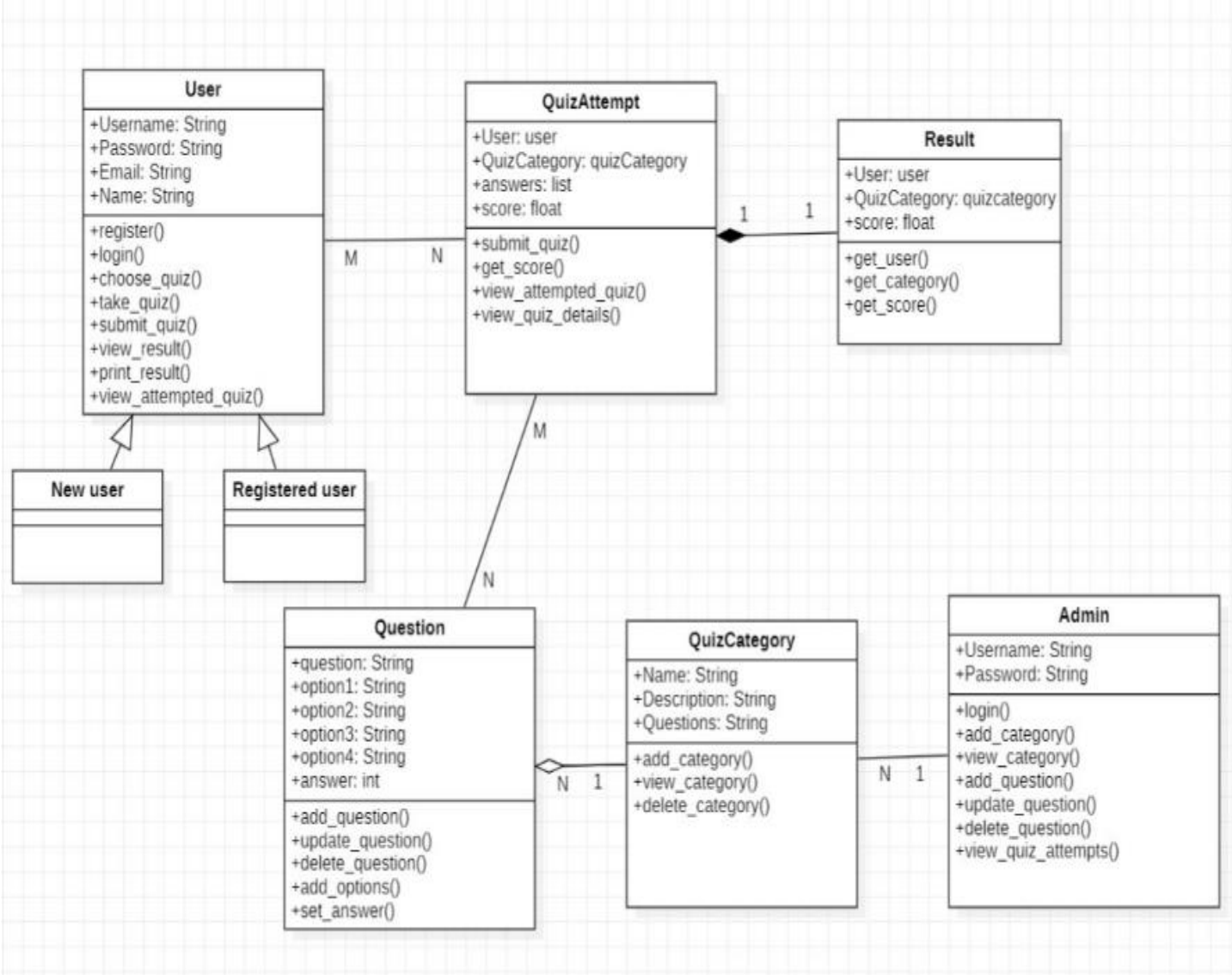
- The admin clicks on the "Manage Quizzes" button.
- The Exam Portal displays a list of quiz categories.
- The admin can select a category to view its questions and options to add, update, or delete questions.
- To add a new question, the admin clicks on the "Add Question" button, enters the question and its answer choices, and saves it to the category.
- To update an existing question, the admin clicks on the "Update Question" button, selects the question to update, makes the necessary changes, and saves it.
- To delete a question, the admin clicks on the "Delete Question" button, selects the question to delete, and confirms the deletion.
- The admin can also view quiz attempt details, including which user and when quizzes were attempted.

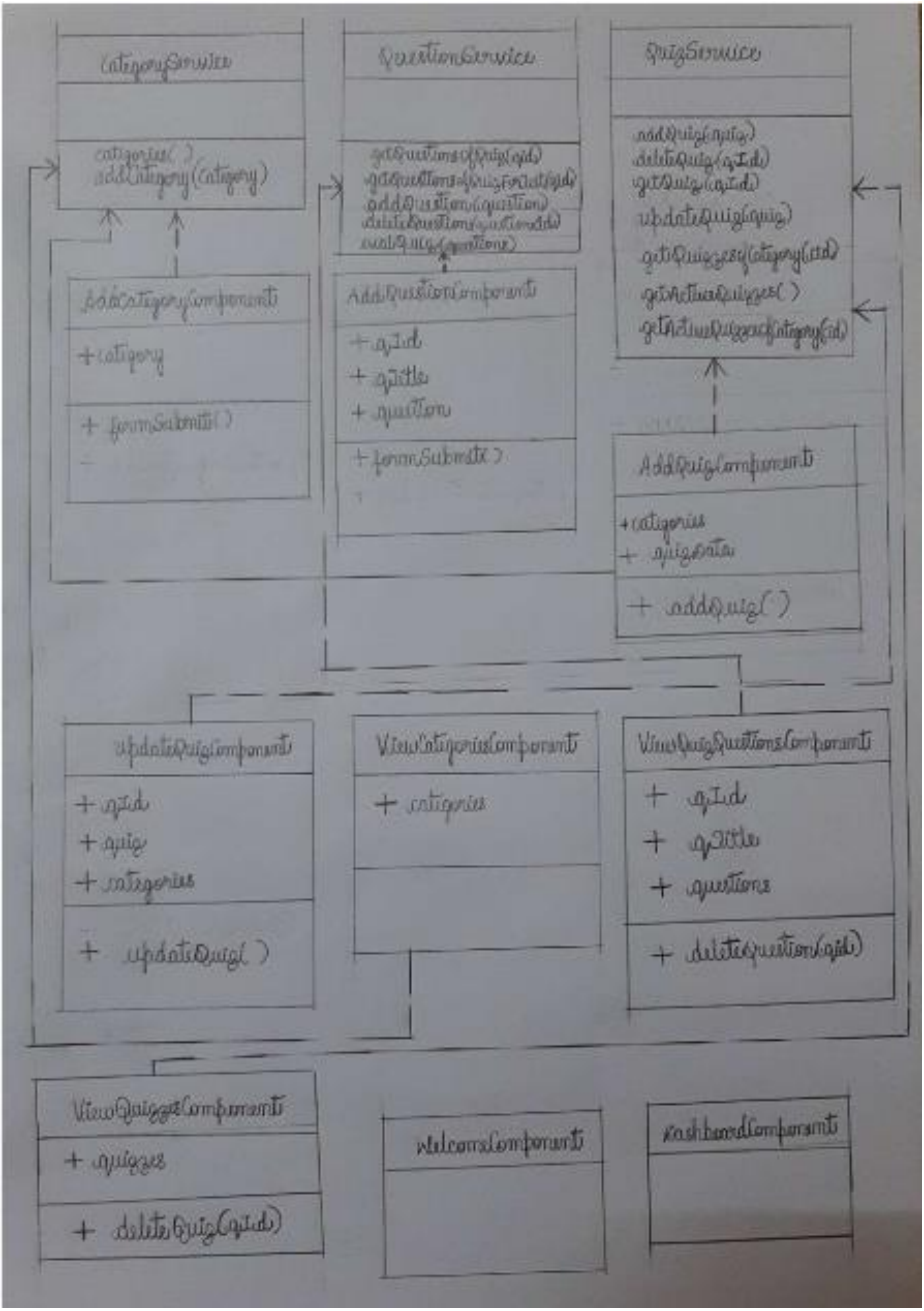
Exceptions:

- If the admin is not authorized to manage quizzes, the Exam Portal displays an error message and does not allow access to the Manage Quizzes feature.

Post-conditions: The quiz categories and questions can be updated or deleted, and quiz attempt details can be viewed.

II. Class Diagram:

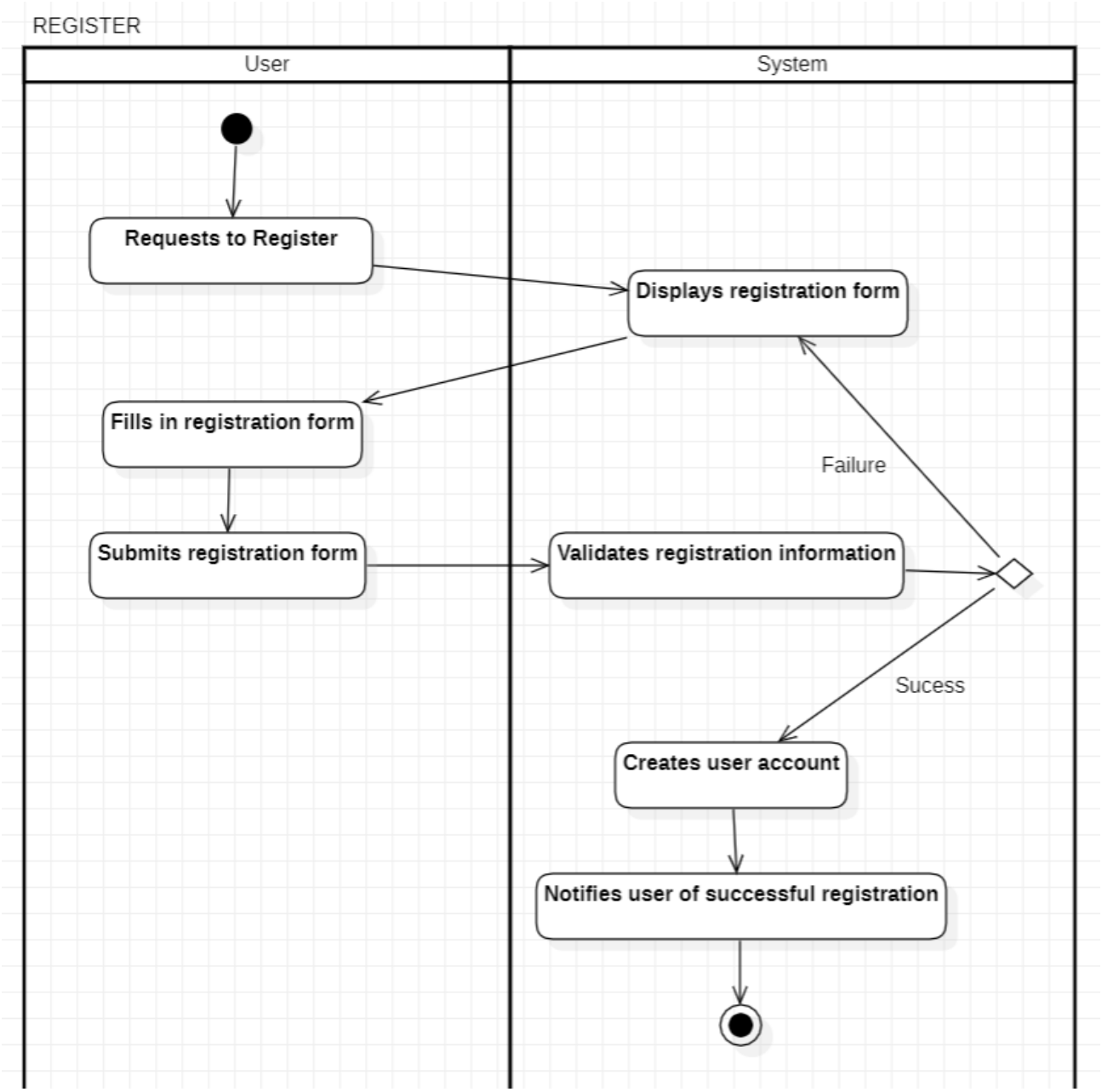




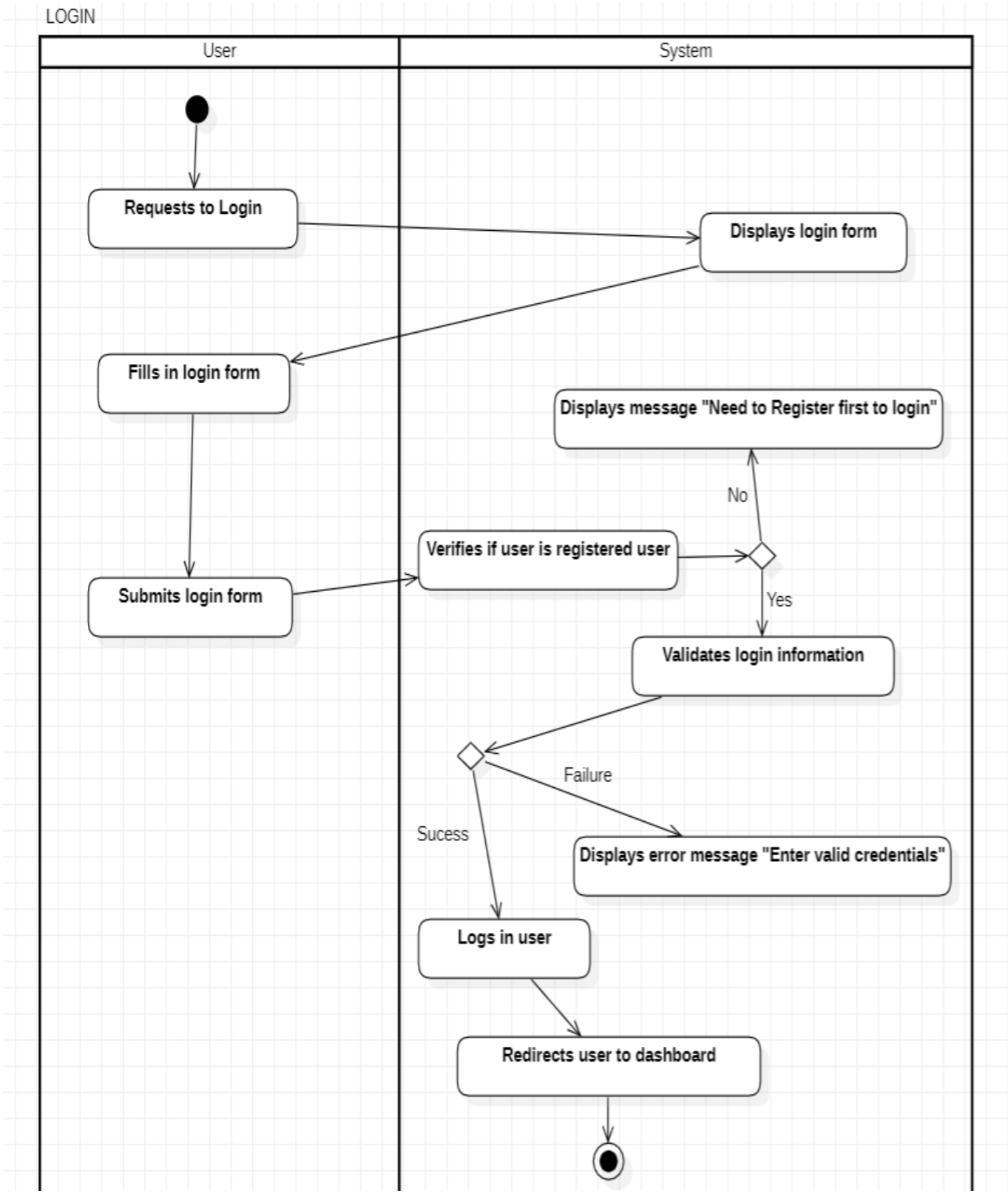
CategoryService
categories( )
addCategory(category)
QuestionService
getQuestionsOfQuiz(qid)
getQuestionsOfQuizForTest(qid)
addQuestion(question)
deleteQuestion(questionId)
evalQuiz(questions)
QuizService
addQuiz(quiz)
deleteQuiz(qId)
getQuiz(qId)
updateQuiz(quiz)
getQuizzesOfCategory(cid)
getActiveQuizzes( )
getActiveQuizzesOfCategory(cid)
AddCategoryComponent
+ category
+ formSubmit( )
AddQuestionComponent
+ qId
+ qTitle
+ question
+ formSubmit( )
AddQuizComponent
+ categories
+ quizData
+ addQuiz( )
UpdateQuizComponent
+ qId
+ quiz
+ categories
+ updateQuiz( )
ViewCategoriesComponent
+ categories
ViewQuizQuestionsComponent
+ qId
+ qTitle
+ questions
+ deleteQuestion(qid)
ViewQuizzesComponent
+ quizzes
+ deleteQuiz(qId)
WelcomeComponent
DashboardComponent

III. Activity Diagram:

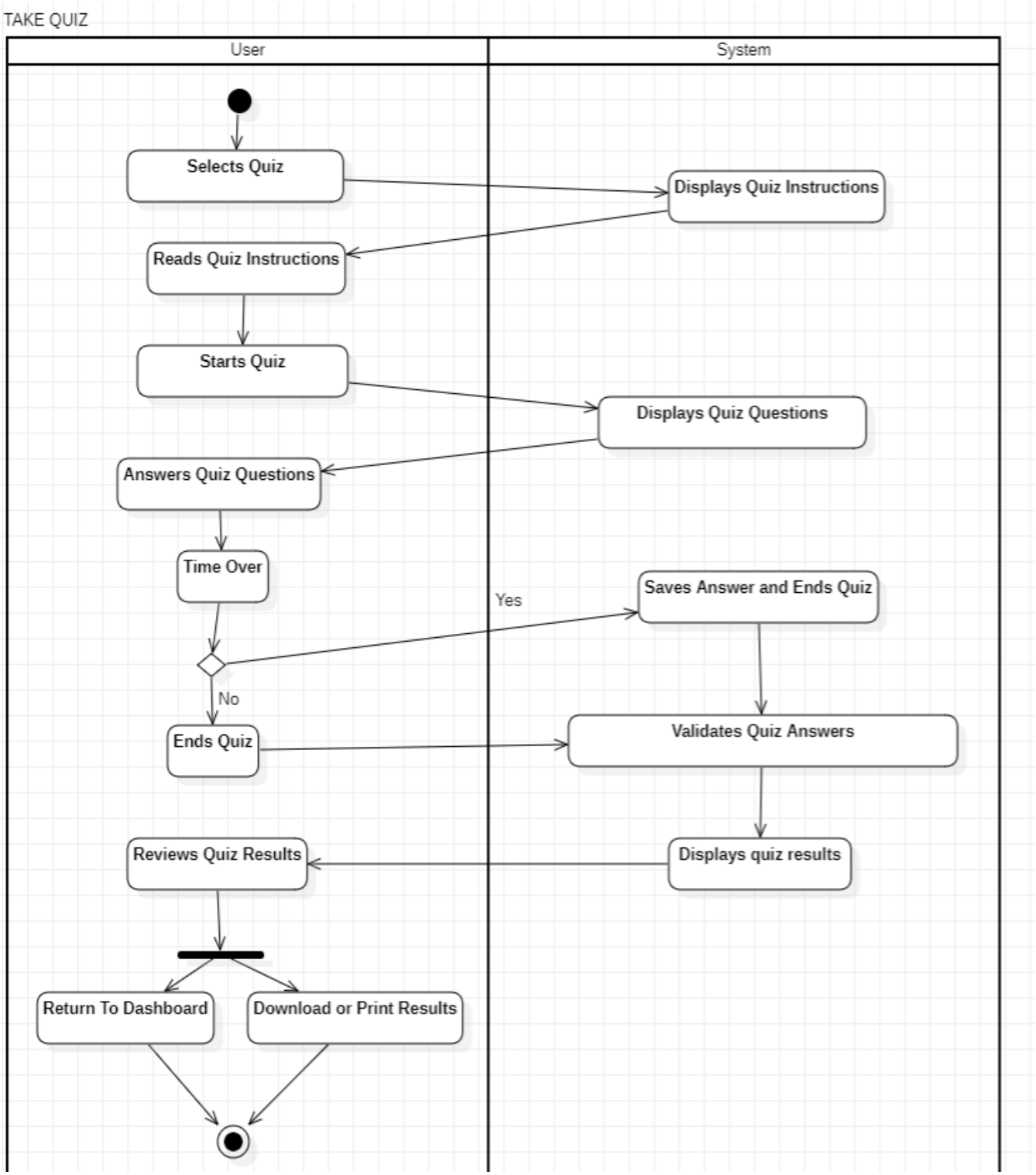
Name: Register



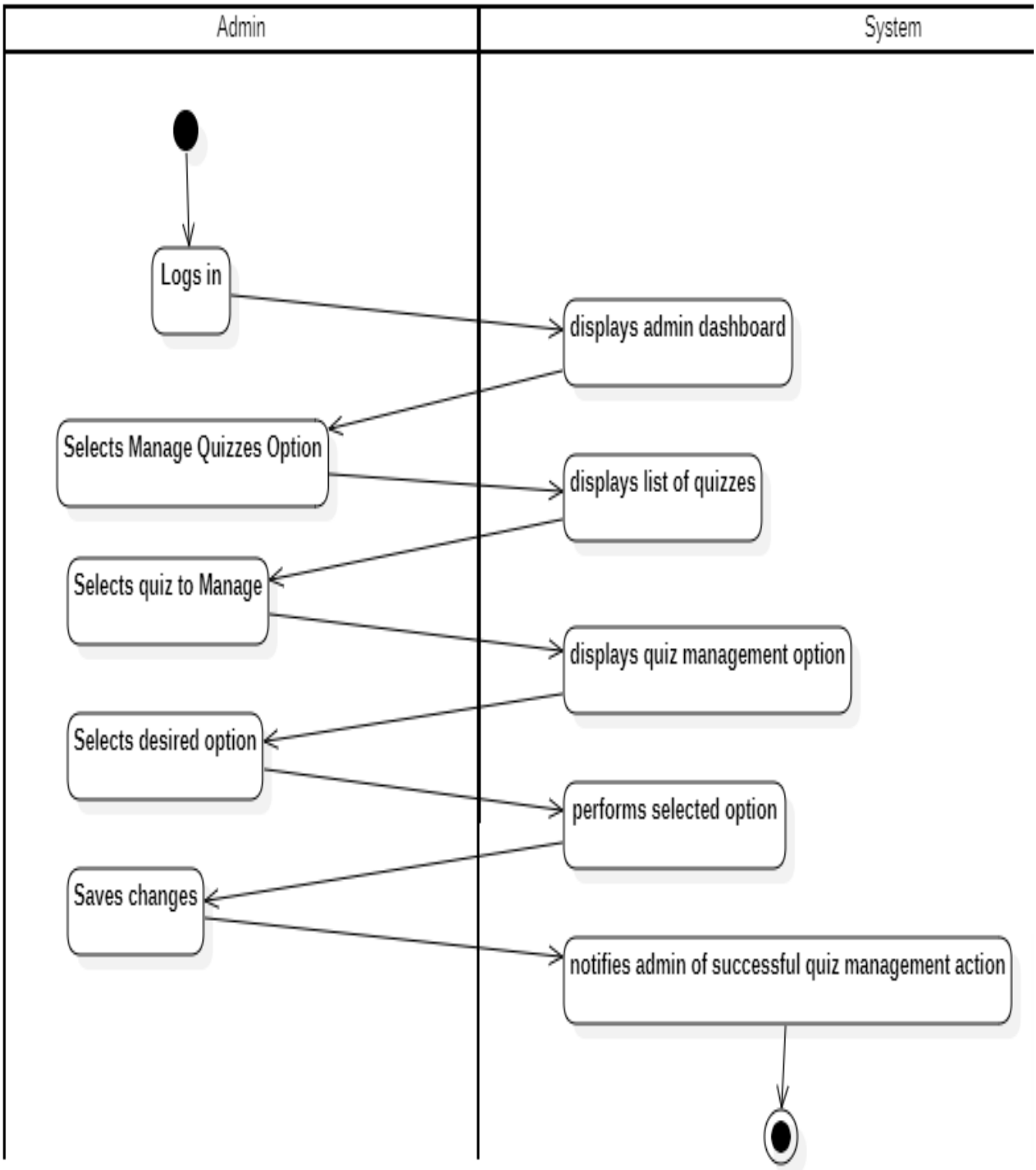
Name: Login


LOGIN
User
System
Requests to Login
Displays login form
Fills in login form
Displays message "Need to Register first to login"
No
Verifies if user is registered user
Submits login form
Yes
Validates login information
Failure
Sucess
Displays error message "Enter valid credentials"
Logs in user
Redirects user to dashboard

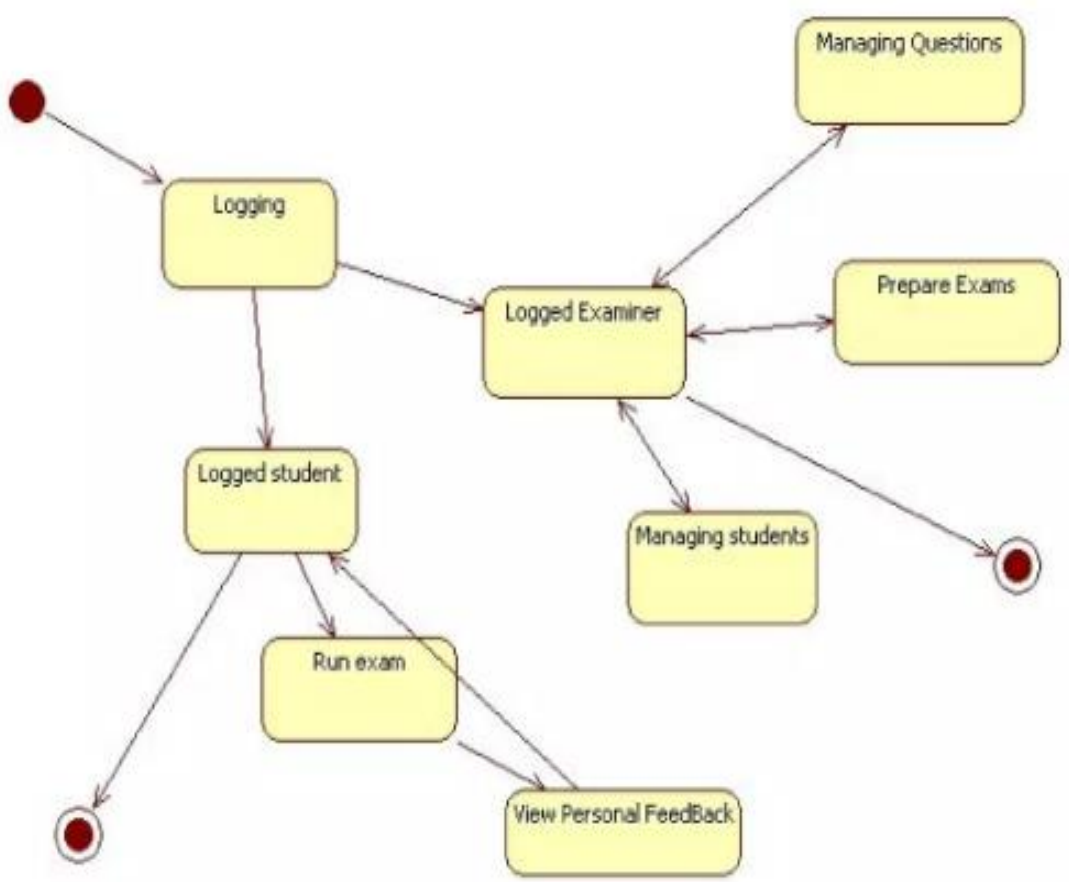
Name: Take Quiz



Name: Manage Quizzes



IV. State Diagram:



- **Architecture Pattern**

Architecture Pattern used by Application is **Model View Controller (MVC)** which separates the application into three interconnected parts: the model (data), the view (user interface), and the controller (business logic).

- **Design Principles:**

1) **Single Responsibility Principle (SRP)**

- ★ The User Class Has A Single Responsibility, Which Is To Represent The Data Of A User.
- ★ The Authority class has a single responsibility of defining the authority string for a user.
- ★ The JwtRequest class has only one responsibility, which is to represent a JWT authentication request.
- ★ The Quiz class has a single responsibility, which is to represent the quiz data model.
- ★ The Question class has a single responsibility, which is to represent a question in an exam.
- ★ The Category class has a single responsibility,

2) **Open/Closed Principle (OCP)**

- The Quiz class is open for extension, but closed for modification.
- The Category class.

3) **Liskov Substitution Principle (LSP)**

The Quiz class can be used wherever the base class Object is expected, without affecting the correctness of the program.

- **Design Pattern**

Domain Model Pattern

The Role class has been implemented as a domain model, which encapsulates the behavior and attributes of a specific concept in the application domain. The class contains fields and methods to manage role-related data.

The Java Persistence API (JPA) design pattern: JPA is a Java-based ORM framework that provides a set of interfaces and annotations to map Java objects to a relational database. The @ManyToOne and @Id annotations in this code are part of the JPA API.(User-Role.java)

The Data Access Object (DAO) pattern:

DAO is a design pattern that separates the database access logic from the business logic of an application.

Object-Relational Mapping (ORM) -

The application is using the JPA (Java Persistence API) to map Java objects to database tables, allowing for seamless integration between the application's Java code and the database.

The class Quiz is annotated with the JPA annotations @Entity, @Id, @GeneratedValue, @Column, @ManyToOne, and @OneToMany. These annotations are used to define the mapping between the Java class Quiz and the relational database table.

Builder design pattern:

The Quiz class could potentially be used with a builder pattern to provide more readable and maintainable code when creating new instances of the Quiz class.

The Question class has a default constructor and getter/setter methods, which can be used to build objects.

Repository pattern:

The Question class does not directly interact with the database. Instead, it is likely used by a repository or service layer to persist and retrieve data.

The Category class acts as a Data Access Object by providing methods to access and manipulate the data stored in the category table.

****