Polish-Japanese Academy of
Information Technology

Quiz App

Project Documentation

Author: Leonid Kim s24900

Tutor: Emil Wcisło, M.E.

Contents

Images and illustration

I. Introduction

Quiz App is and application for educational organizations that improves learning experience. It allows teachers to create and manage quizzes and students to take quizzes and see their results.

I.I Purpose

Improve learning experience by providing automated grading and easy-to-use interface for quizzes.

I.II Scope

Creating a desktop app that is intended to be used by three categories of users: students, teachers, administrators.

I.III Definitions, acronyms and abbreviations:

- Quiz - a short test of knowledge.
- User - anyone who interacts with the app.
- Student – a user that can take quizzes and receive feedback.
- Teacher – a user that can create and manage quizzes.
- Administrator (admin) – a user that can manage system and users.

I.IV Overview

- First section of the document is an introduction describing general information about the project.
- Second section of the document is an overall description of the whole project and showing how the application is structured.
- Third section describes a specific use case.
- Fourth section of the document describes Graphical User Interface.
- Fifth section of the document is dedicated to design decisions.

II Overall description

The Quiz App is a platform designed to facilitate the creation, distribution, and management of quizzes for educational purposes. The app caters to three main types of users: students, teachers, and administrators, each with specific roles and functionalities to ensure a streamlined and effective experience.

II.I Key features and functionalities

User Management:

- Register: New users can create an account to access the app's features.
- Login: Existing users can log into the app to access their personalized dashboards.
- Reset Password: Users can reset their passwords if forgotten, ensuring continuous access to their accounts.
- View Profile: Users can view and manage their profile information.

Student Features:

- Browse Quizzes: Students can browse available quizzes assigned to them or available for practice.
- Take Quiz: Students can participate in quizzes by answering the questions provided.
- Submit Quiz: After completing a quiz, students can submit their responses for evaluation.
- View Quiz Results: Students can view their quiz results, including scores and feedback on their performance.
- Track Progress: Students can track their progress over time, viewing historical performance data to monitor their learning journey.

Teacher Features:

- Assign Students to Quizzes: Teachers can assign quizzes to specific students or groups, tailoring the quiz distribution based on the class or individual needs.
- Create Quiz: Teachers can create new quizzes, designing questions in various formats to assess student knowledge and skills.
- Edit Quiz: Teachers can edit existing quizzes to update content, modify questions, or change quiz settings.
- Delete Quiz: Teachers can delete quizzes that are no longer needed or relevant.

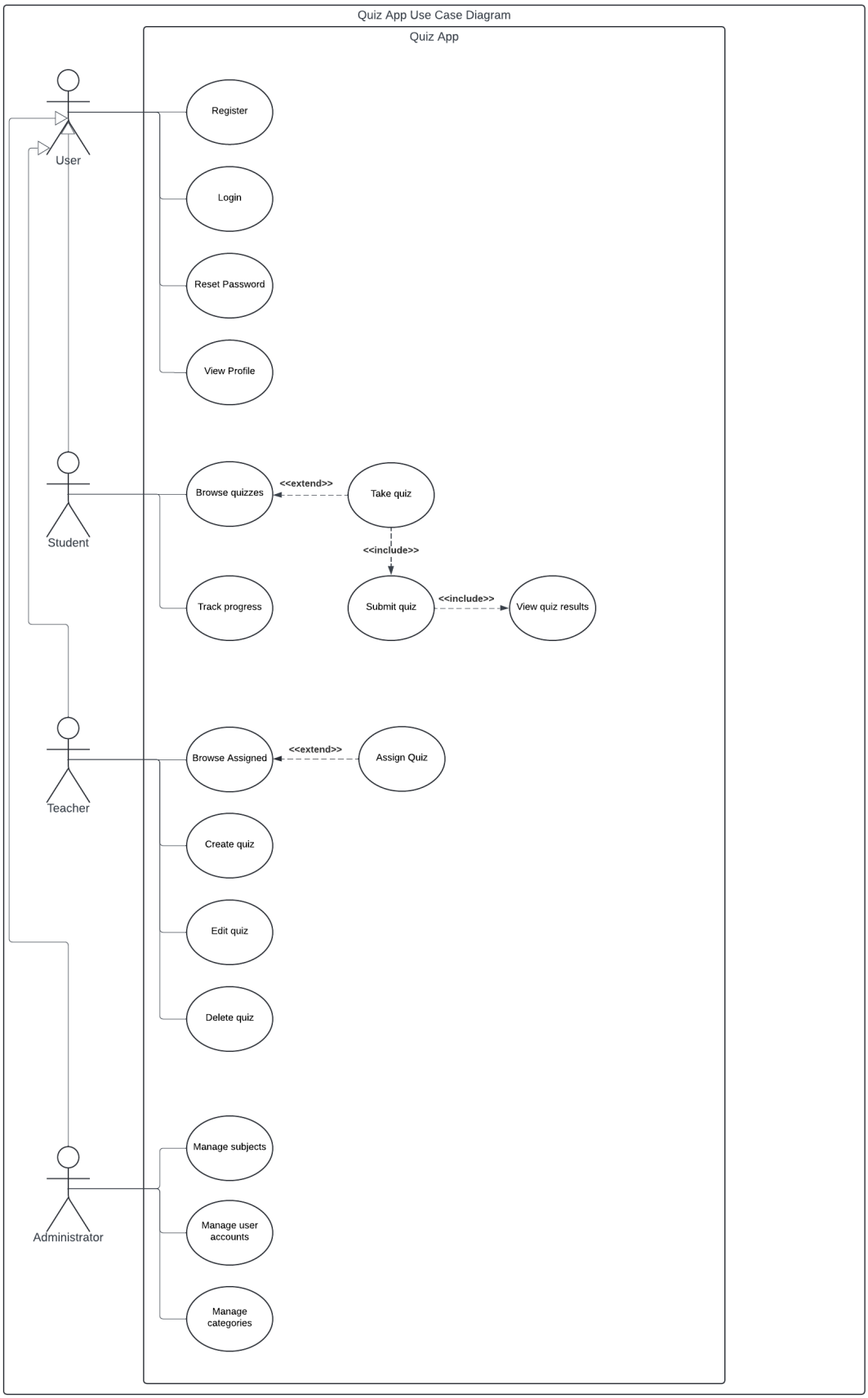
Administrator Features:

- Manage Subjects: Administrators can manage subjects within the app, ensuring that quizzes are categorized appropriately for easy navigation and organization.
- Manage User Accounts: Administrators can create, modify, and delete user accounts, overseeing the user base and maintaining system integrity.
- Manage Categories: Administrators can manage categories to further organize quizzes, subjects, and other content, ensuring a structured and user-friendly experience.

II.II Workflow

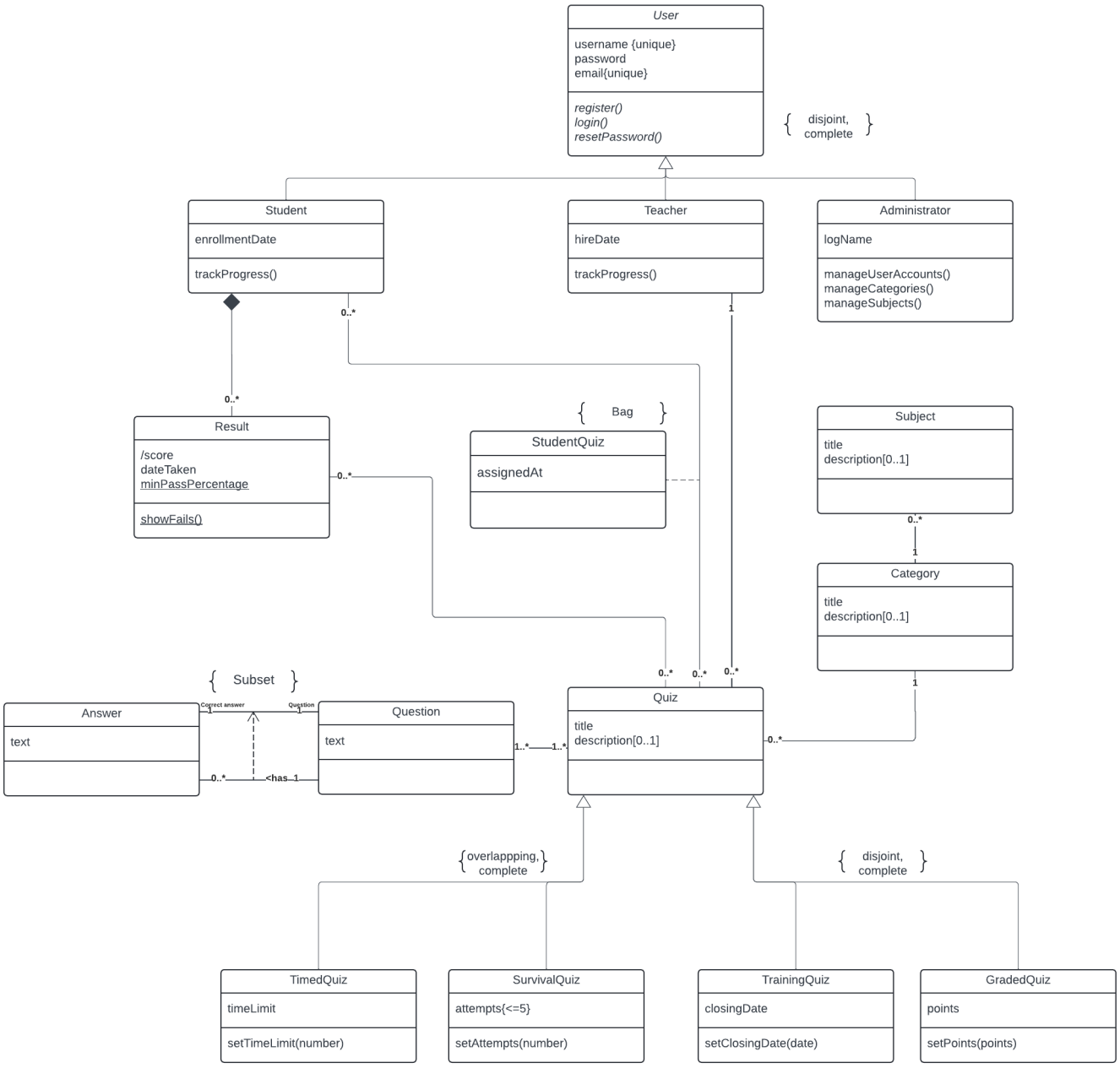
Users start by registering for an account and logging in. This grants them access to the functionalities based on their assigned roles (student, teacher, or administrator). Students browse available quizzes, take them, submit their responses, and view the results. They can also track their progress to see how they are improving. Teachers create quizzes, edit or delete these quizzes as needed. Then they assign them to students. Administrators have overarching control to manage subjects, user accounts, and categories, ensuring the app runs smoothly and remains organized.

II.III Use-Case Model Survey



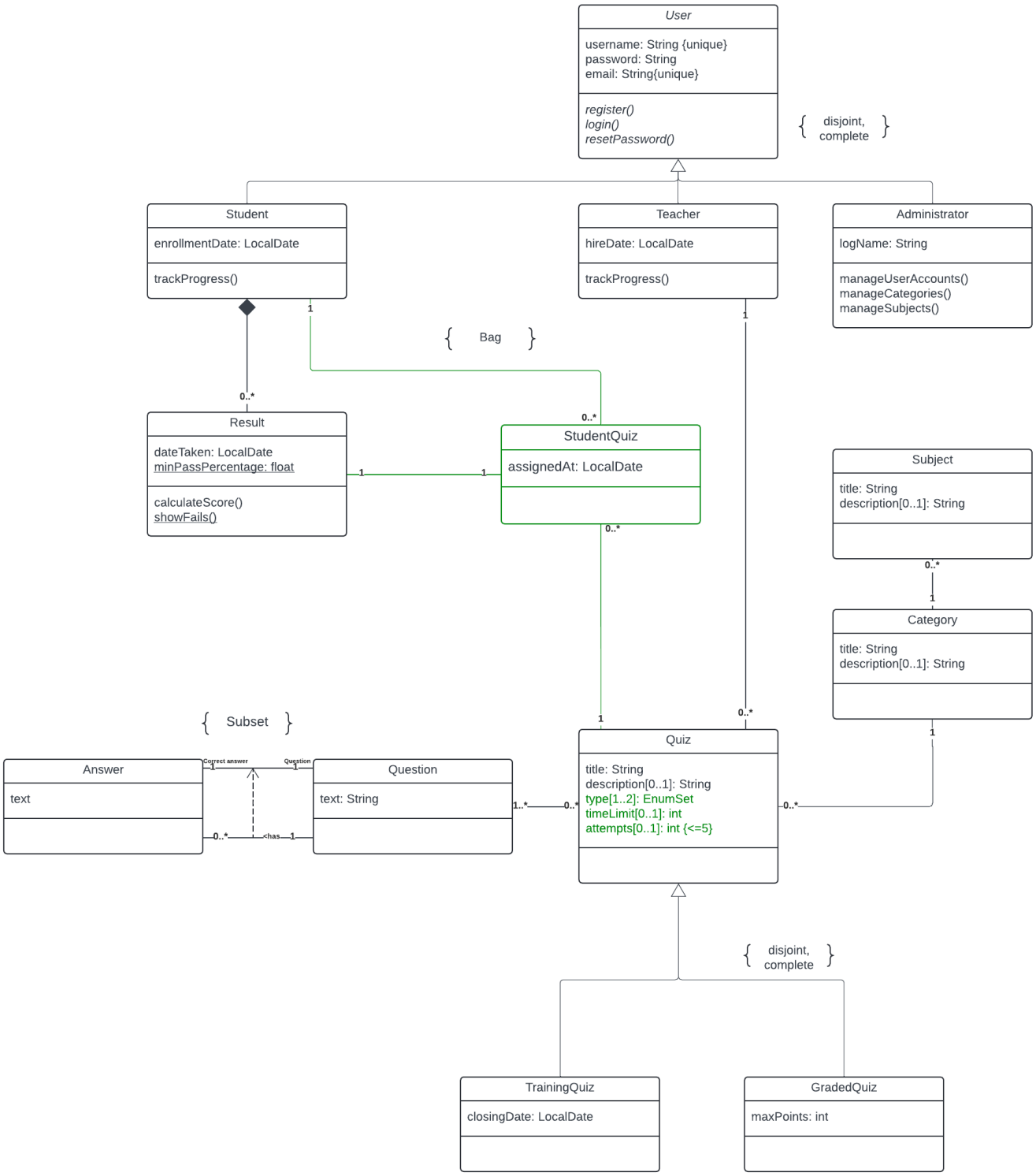
1 Use-case model survey

II.IV Analytical Class Diagram



2Analytical Class Diagram

II.V Design Class Diagram



3 Design Class Diagram

III Take a quiz use case description

III.I Use Case Scenario

Actor: Student

Purpose and context: Student wants to solve an assigned quiz.

Dependencies:

Included use cases: Submit quiz

Assumptions and preconditions: The student is logged into his/her account; The system is running and has access to user and quiz data.

Trigger: Browse quizzes UC

Basic flow of events:

1) The student initiates the “Take quiz” use case

2) The system checks if quiz is completed.

3) The system checks if there are questions assigned to the quiz.

4) The system displays quiz instructions (attempts, title, total points, time limit)

5) The student starts the quiz by clicking “Start” button

6) The system presents the first quiz question

7) The student selects an answer and clicks “Next button”

8) The system presents the next question

9) The student selects an answer and clicks “Next” button

10) Repeat points 6-7 until the last question

11) The system displays the last question with “Submit” button underneath

12) The student selects and answers the question

13) The use case ends.

Alternative flows of events:

There are no questions assigned to the selected quiz:

4a) The system displays “There are no questions in this quiz” message

5a) The use case ends

Quiz already completed:

3b) The system detects that the student has already completed the quiz.

4b) The system displays a message: “You have already completed this quiz.”

5b) The use case ends.

Time limit exceeded:

10c) The system detects that the time limit for the quiz has been exceeded.

11c) The system automatically submits the current answers.

12c) The use case ends.

Included Use Case: View Quiz Results UC

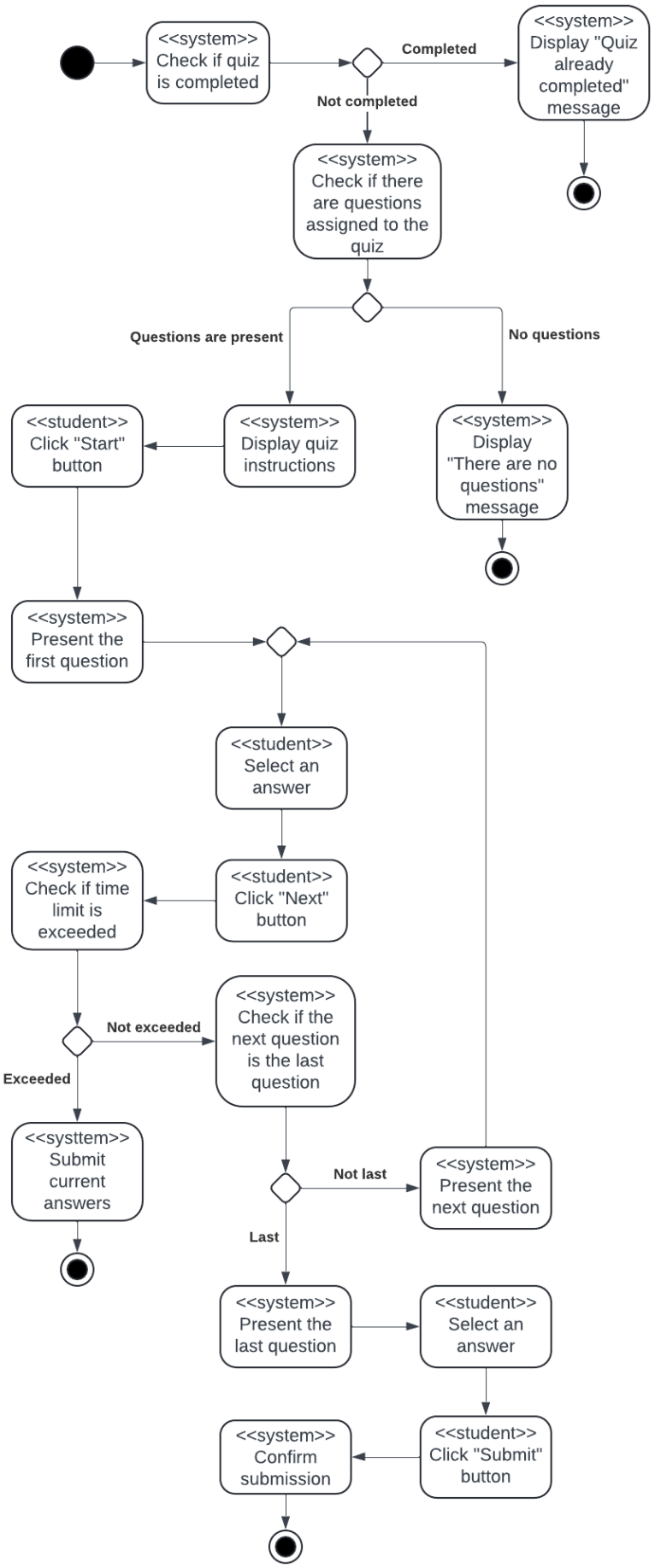
14) The student clicks “Submit” button.

15) The system confirms the submission.

16) The use case ends.

Post-condition: the questions of the quiz are answered, the quiz is submitted.

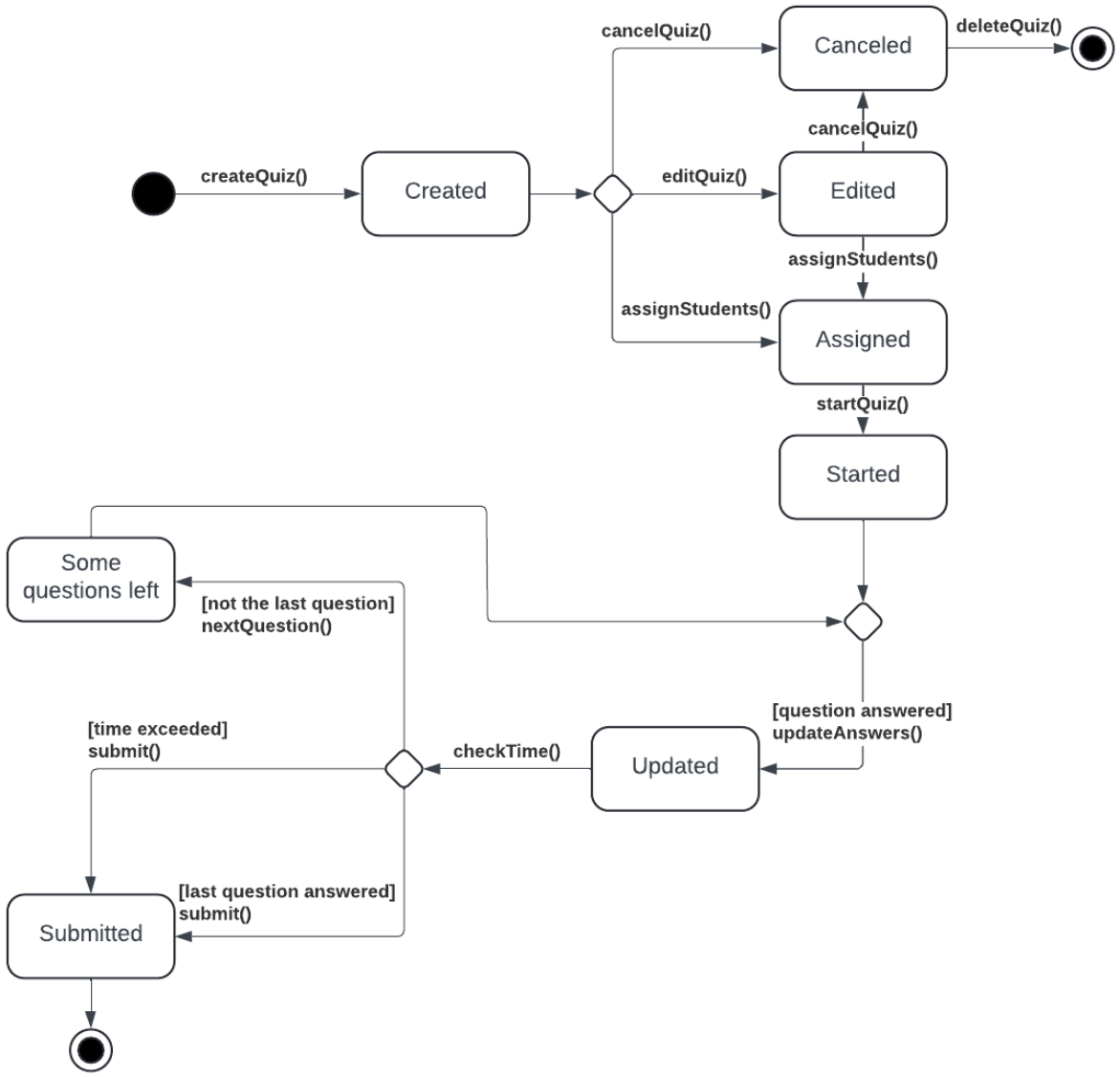
III.II Take a quiz Activity Diagram



4 Take a quiz Activity Diagram

III.III Quiz State Diagram

Since the most important class in the selected use case is “Quiz”, the state diagram is dedicated to it.

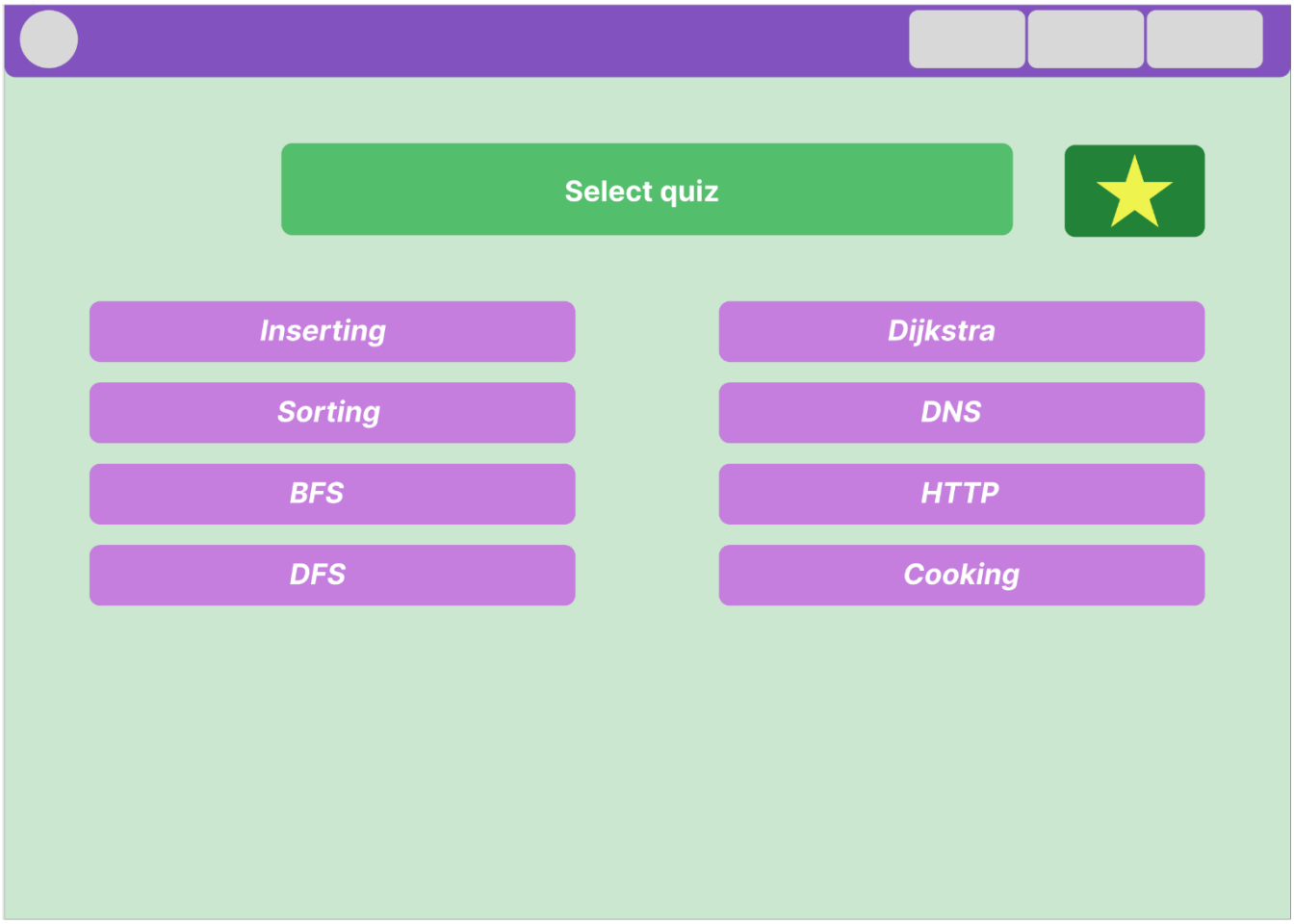


5 Quiz State Diagram

IV The GUI design

The idea behind GUI is simple: make the interface as simple and comprehensible as possible. It was essential to make UI components minimalistic and large enough for user to easily identify and select them.

“Select a quiz” page



6 "Select a quiz" page

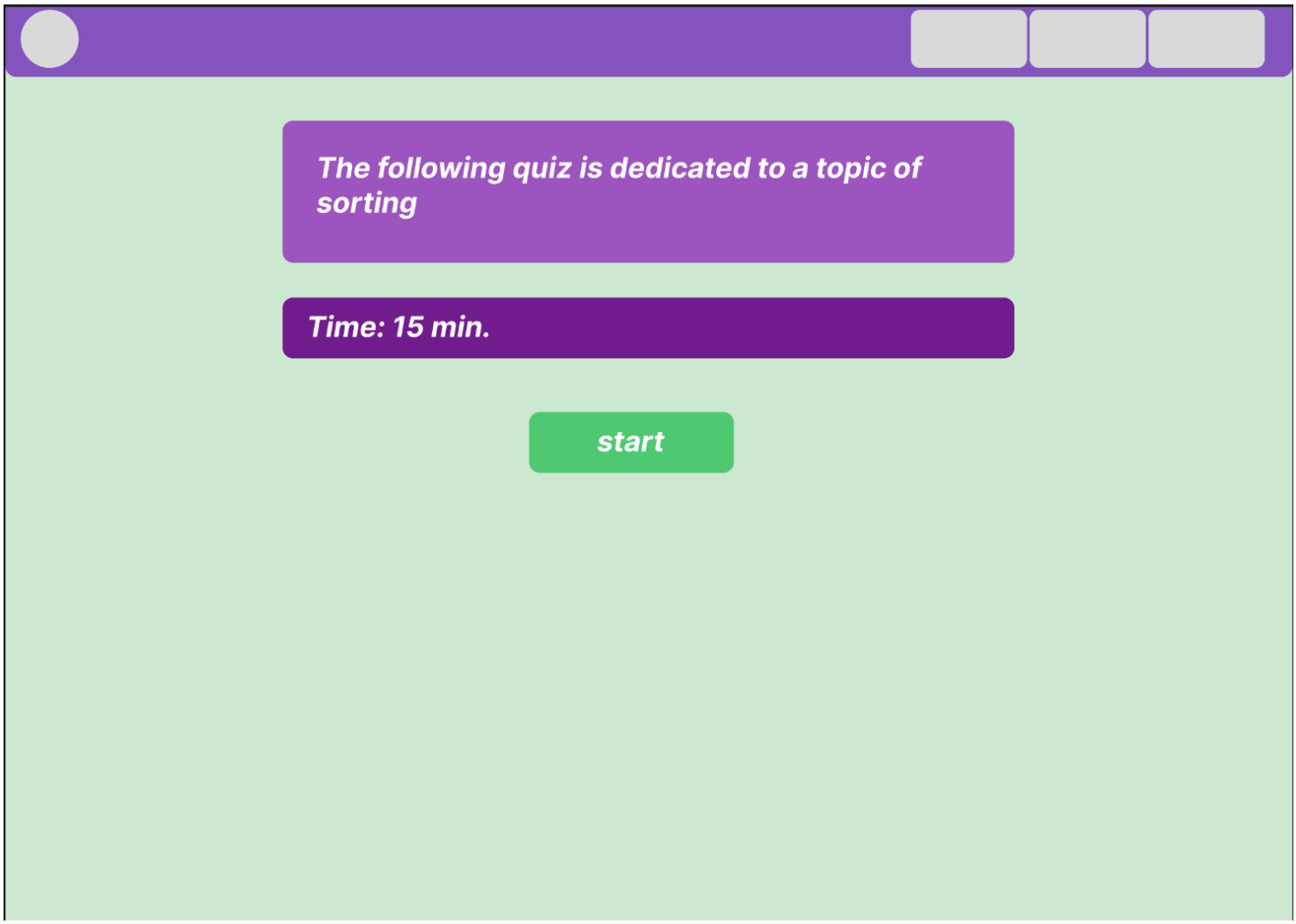
On this page, a student can see all the quizzes that are assigned to him/her. Whenever a mouse pointer floats above an interactive component, the component becomes darker in color. The options are: select a specific quiz or press “star” button to view a list of quizzes that are already completed with their results.

List of completed quizzes

7 List of completed quizzes

As you can see on the image above, a list is displayed right under the “star” button. It takes roughly 40% of the page in width.

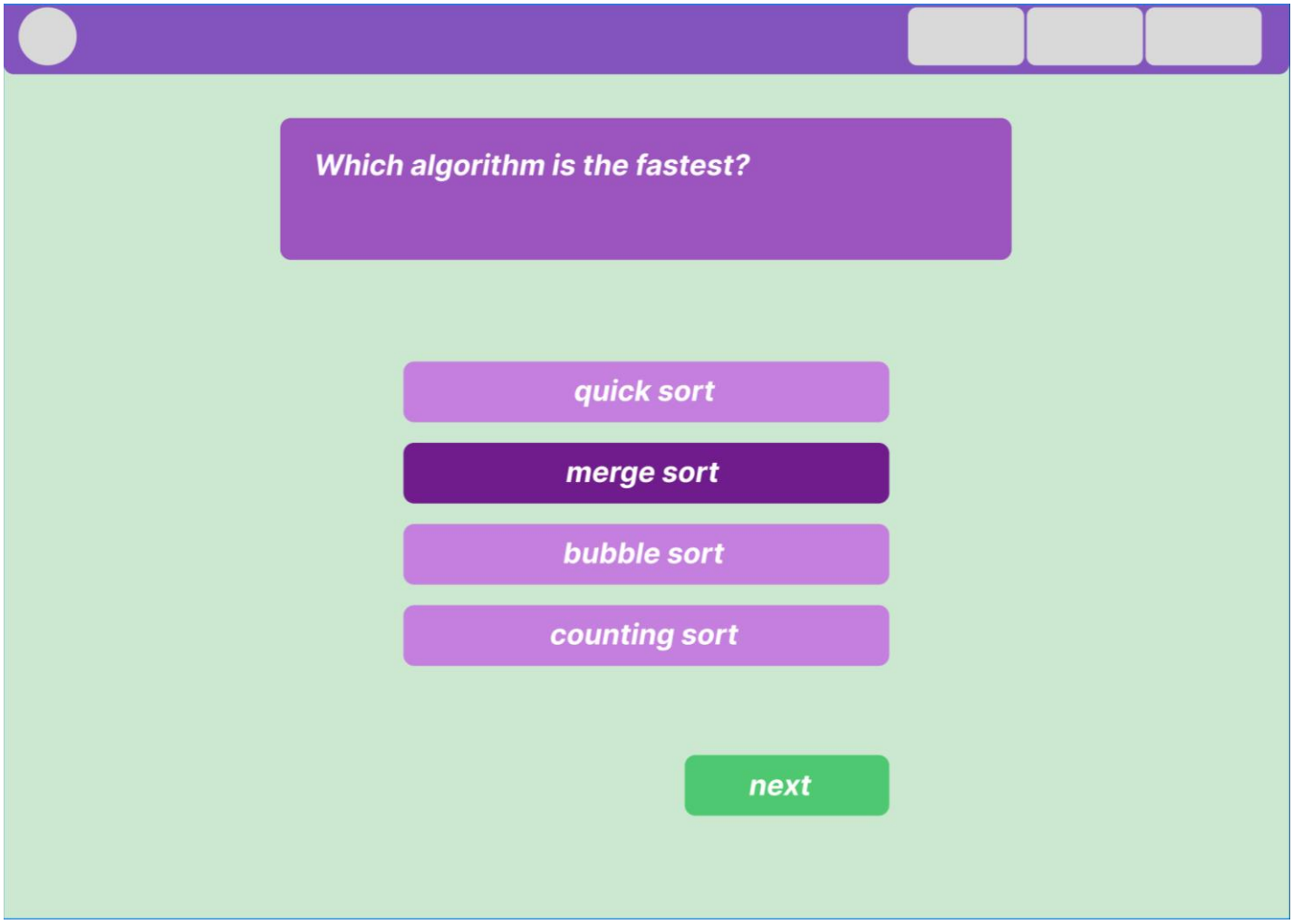
Quiz info, when a quiz is selected



8 Quiz info

The info about the quiz is displayed. If a quiz is timed or has attempt, according information is displayed.

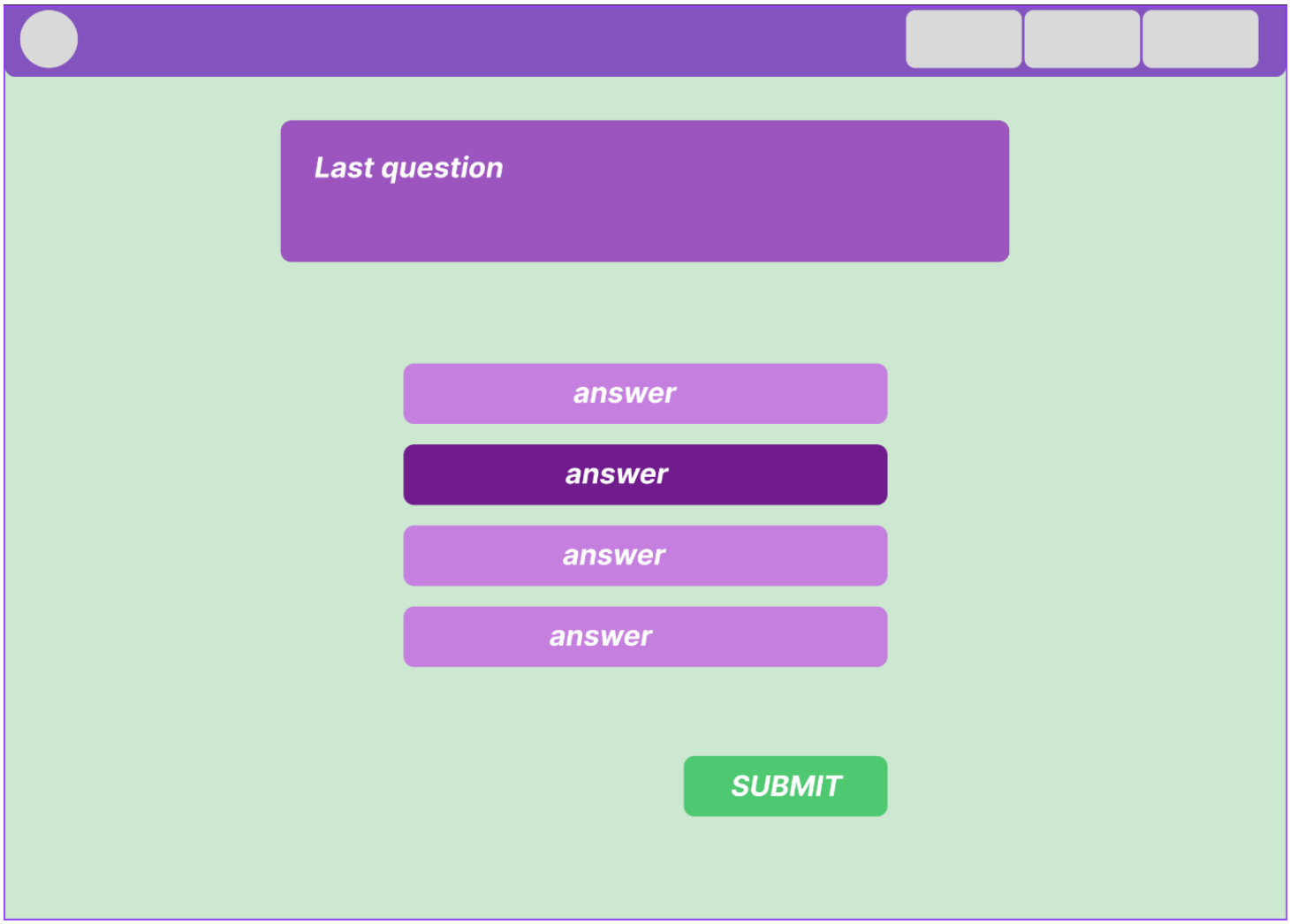
Question of a quiz



9 Question of a quiz

Apart from selecting an answer, a user will have to press “next” button to get to the next question.

Last question



10 Last question

When the last question is presented, a “SUBMIT” button appears.

V The discussion of Design Decisions and the Effect of Dynamic Analysis

- The durability of the app will be achieved via MySQL as a database; Hibernate as an ORM and Java as a language of choice.
- Some constraints will be implemented with annotations, provided by Hibernate, some – through custom validation.
- Due to spatial constraints, get() and set() methods were not included in class diagrams, but they will appear in the implementation to provide safe manipulation of parameters.
- Some classes will include additional methods that are necessary for a proper implementation of desired ideas.
- Overlapping inheritance (Timed quiz, Survival quiz) will be implemented via introducing EnumSet in Quiz class.
- Subset will be validated via custom validator.

Tools used for documentation

Text editor: Microsoft Word 2016

Diagram editor: LucidCharts

GUI design: Figma

VI Corrections

VI.I Use Case Scenario

Actor: Teacher

Purpose and context: Teacher wants to browse assigned quizzes

Dependencies:

Extended Use Case: Assign quiz

Assumptions and preconditions: The teacher is logged into his/her account; The system is running and has access to user and quiz data.

Trigger: Teacher clicks an according button in his menu.

Basic flow of events:

1) The teacher initiates the “Browse Assigned” use case.

2) System displays a list of students.

3) Teacher clicks on a student.

4) System displays a list of quizzes assigned to student.

Extended:

5) Teacher clicks on student choice box.

6) System displays a list of students to select from.

7) Teacher selects a student.

8) Teacher clicks on quiz choice box.

9) System displays a list of quizzes to select from.

10) Teacher selects a quiz.

11) Teacher clicks “Assign” button.

12) System assigns a quiz to student.

13) The use case ends.

Alternative flows:

No students:

2a) The system detects that there are no students available.

3a) The system displays an empty list of students.

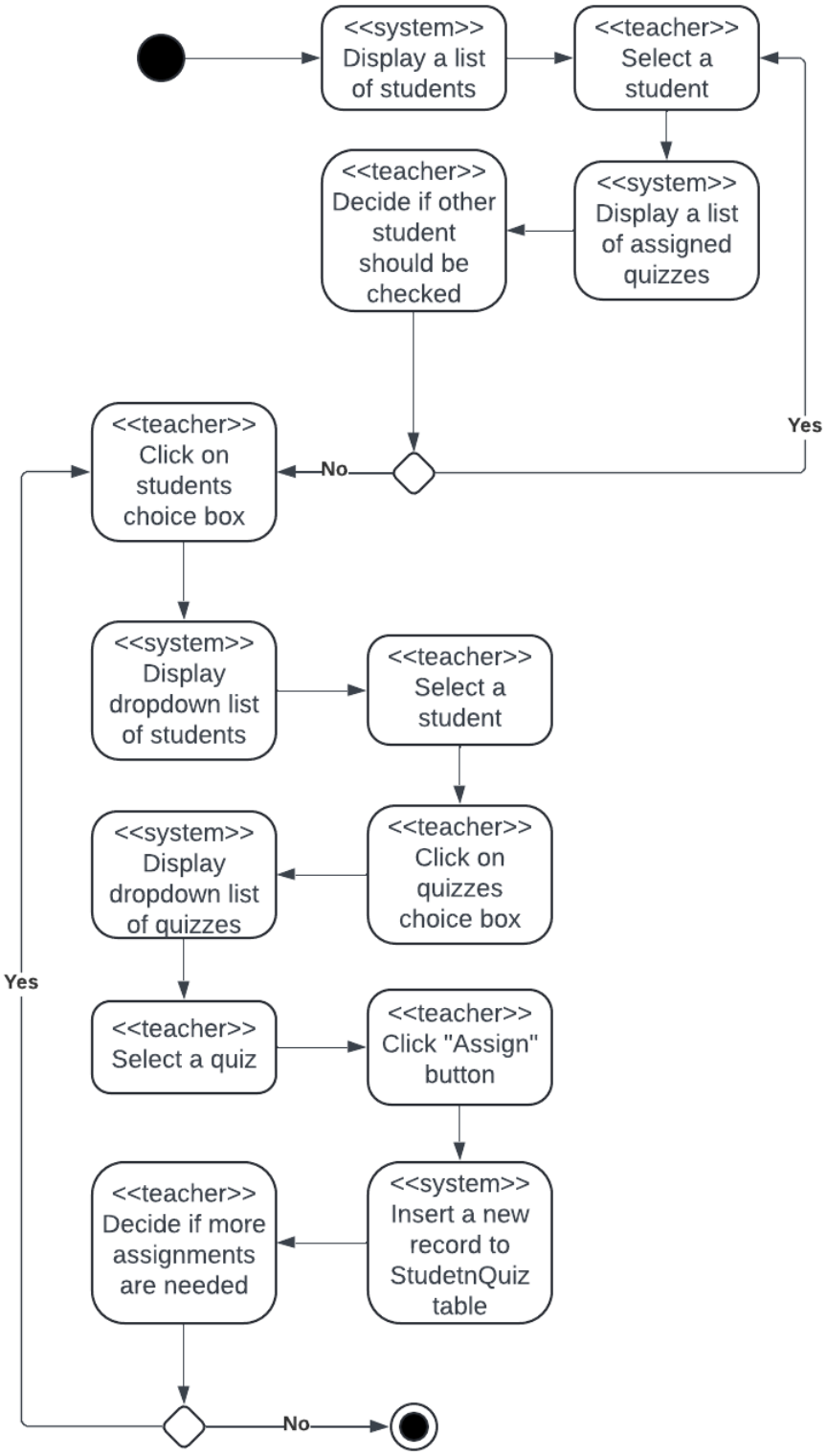
No quizzes assigned to student:

4b) The system detects that no quizzes are assigned to the selected student.

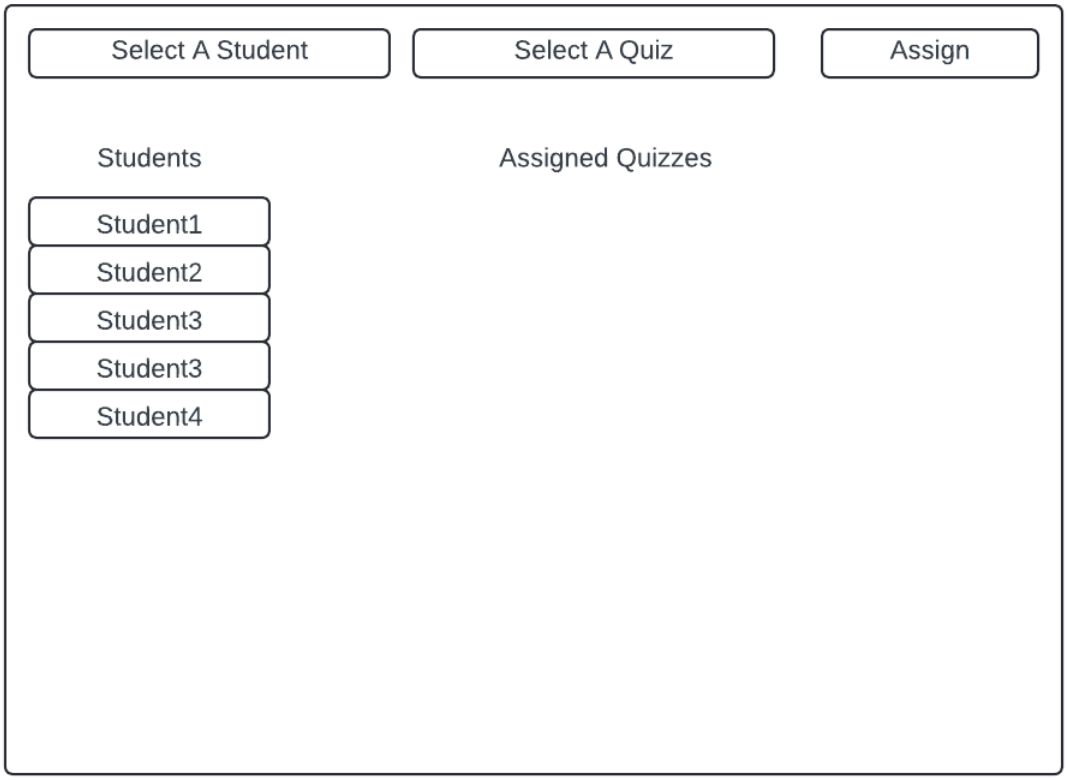
5b) The system displays an empty list of assigned quizzes.

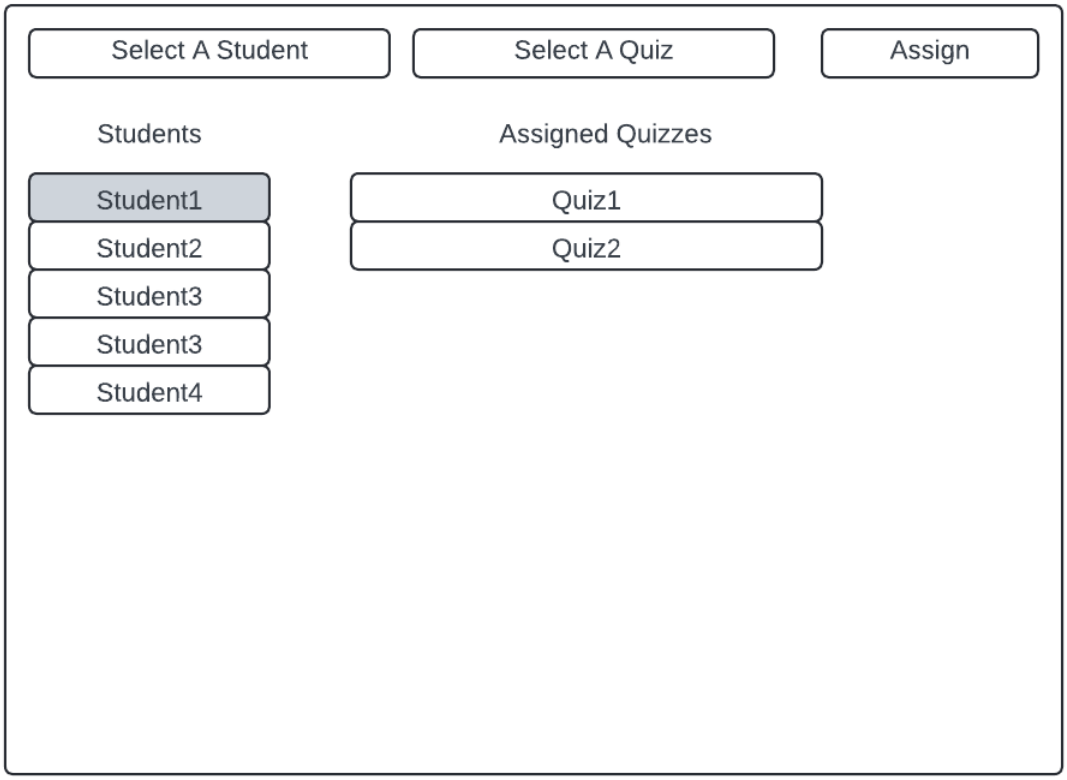
Post-condition: Assigned quizzes are browsed; Another quiz is assigned to a student.

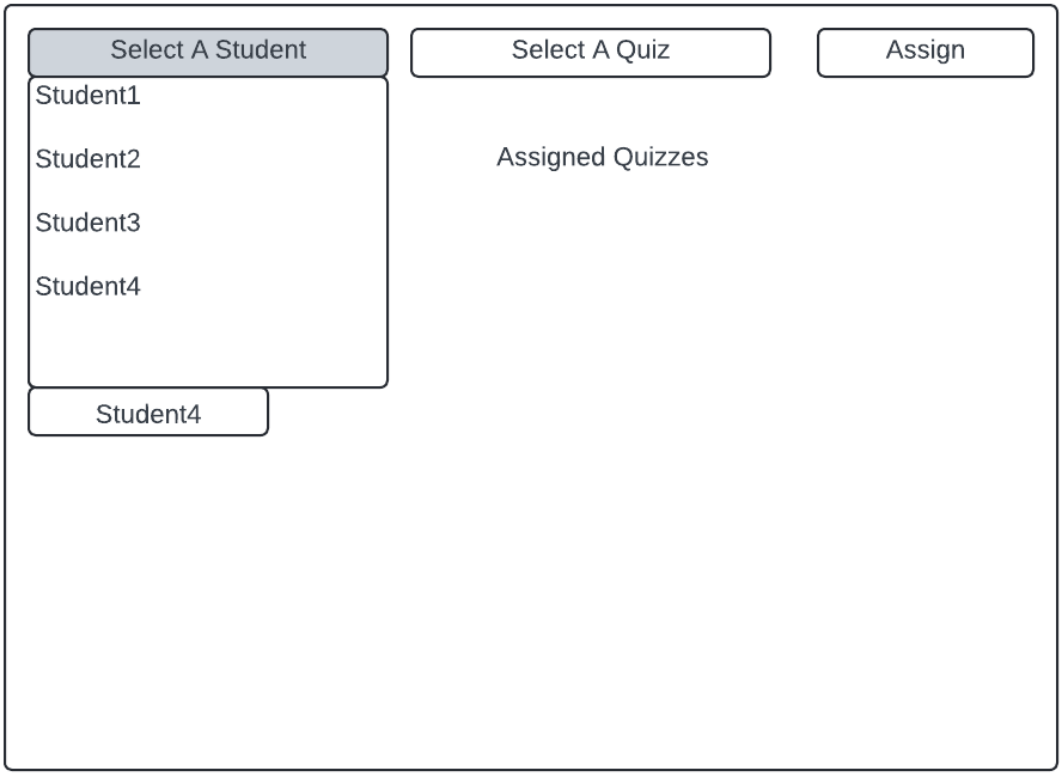
VI.II Activity Diagram

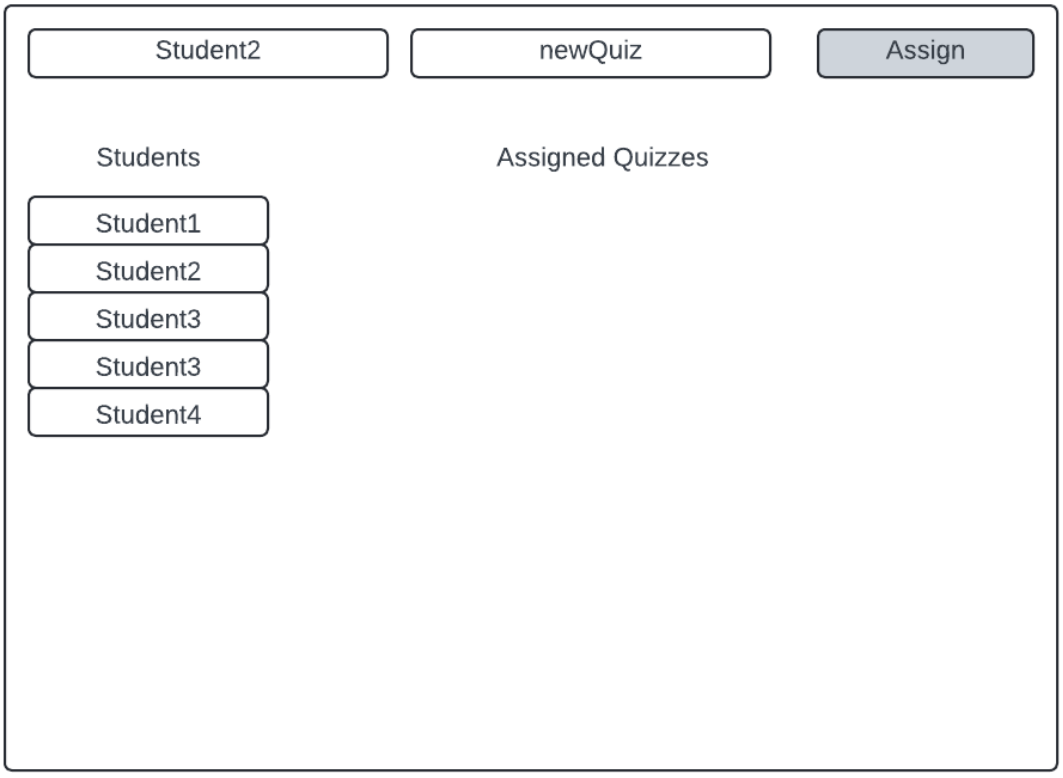


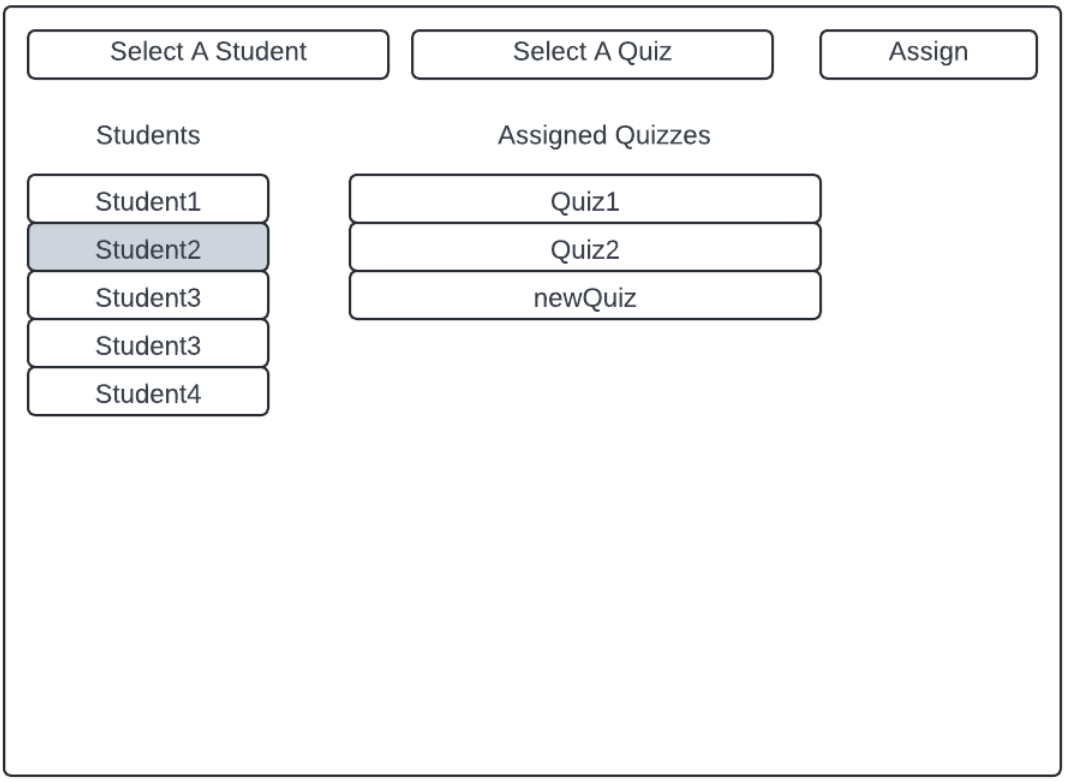
VI.III GUI


Select A Student
Select A Quiz
Assign
Students
Assigned Quizzes
Student1
Student2
Student3
Student3
Student4


Select A Student
Select A Quiz
Assign
Students
Assigned Quizzes
Student1
Student2
Student3
Student3
Student4
Quiz1
Quiz2


Select A Student
Select A Quiz
Assign
Student1
Student2
Student3
Student4
Assigned Quizzes
Student4


Student2
newQuiz
Assign
Students
Assigned Quizzes
Student1
Student2
Student3
Student3
Student4


Select A Student
Select A Quiz
Assign
Students
Assigned Quizzes
Student1
Student2
Student3
Student3
Student4
Quiz1
Quiz2
newQuiz