VIRTUAL TEST HUB

SOFTWARE DESIGN DOCUMENT

Submitted by

A.ARUN KUMAR (B201031)
V.VAMSHI KRISHNA (B200182)
MD.VASEEM AKRAM (B200927)

Bachelor of Technology

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Software Design Document (SDD)
Virtual Test Hub (VTH)

November 23, 2024

Contents

1 Introduction

1.1 Purpose

The purpose of this SDD is to provide a comprehensive design blueprint for the Virtual Test Hub. This platform facilitates online tests with features for test management, user authentication, and performance reporting. Software Design Document (SDD) of Carpool provides necessary definitions to conceptualize and further formalize design of the software, whose requirements and functionalities were summarized in Software Requirements Specifications (SRS) Report. Aim is to provide guidance to a design which could be easily implemented by any programmer reading this report. The document complies with the IEEE standards (IEEE Std 1016 – 2009).

1.2 Scope

The design includes a modular system built using the MERN stack (MongoDB, Express.js, React, Node.js) with robust integration for test administration, user management, and notification functionalities. This complete SDD will contain the general definition and features of the project, design constraints, the overall system architecture and data architecture, a brief explanation about our current progress and schedule of the project. With the help of UML diagrams, design of the system and subsystems/modules will be explained visually in order to help the programmer to understand all information stated in this document correctly and easily.

1.3 Definitions, Acronyms, Abbreviations

- **SDD**: Software Design Document
- **MERN**: MongoDB, Express.js, React, Node.js
- **VTH**: Virtual Test Hub
- **ER**: Entity Relationship

1.4 References

- Software Requirements Specification (SRS) document for VTH

2 System Overview

2.1 System Architecture

The architecture is based on a client-server model. Users interact with the system via a web application built on React, which communicates with a backend Node.js server to process requests. MongoDB serves as the database for storing user information, test data, and results. The overall system architecture and data architecture, a brief explanation about our current progress and schedule of the project. With the help of UML diagrams, design of the system and subsystems/modules will be explained visually in order to help the programmer to understand all information stated in this document correctly and easily.

2.2 Subsystems

- **Authentication:** Handles secure login, registration, and role management. Authentication is the process of verifying the identity of a user or system before granting access to a resource or service. In the context of your mini project on an online exam portal, authentication ensures that only authorized users—such as students, instructors, or administrators—can access their respective areas and functionalities.

- **Test Management:** Enables instructors to create, modify, and evaluate tests. Test management involves the planning, execution, monitoring, and reporting of software testing activities to ensure the quality of a system or application. For your online exam portal, test management is critical to ensure that features such as authentication, exam creation, and result generation work as expected.

- **Reporting:** Provides performance metrics for students and instructors. Reporting in test management is the process of documenting and communicating the results of testing activities to stakeholders. Effective reporting provides insights into the quality, stability, and readiness of the system.

- **Notifications:** Alerts users about test schedules and reminders. Notifications are essential for keeping users informed about important events or updates. They enhance user experience by delivering timely information, ensuring no critical updates are missed.

3 System Models

The following diagrams illustrate the various models within the system:

3.1 Use Case Diagrams

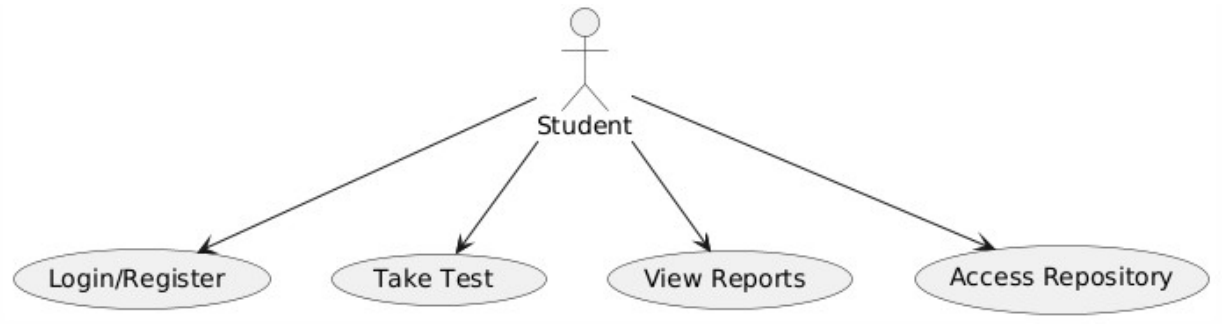

Figure 1: Use Case Diagram for Student Interactions

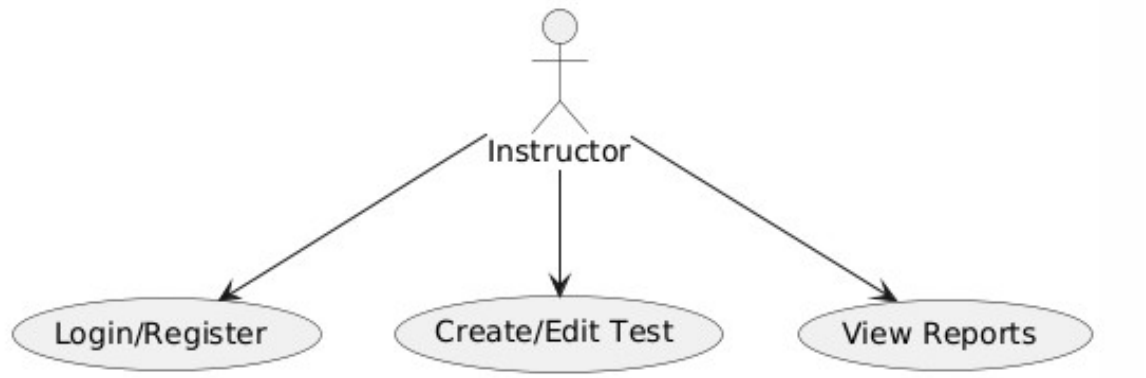

Figure 2: Use Case Diagram for Instructor Interactions

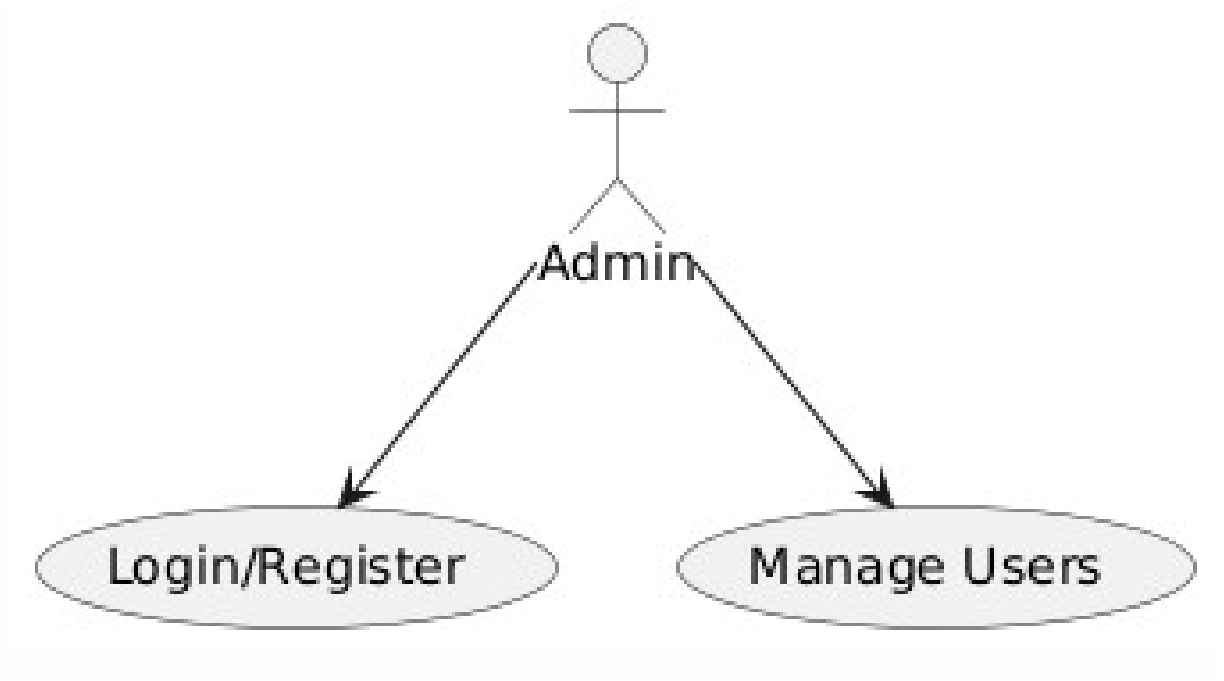

Figure 3: Use Case Diagram for Admin Interactions

3.2 Sequence Diagrams

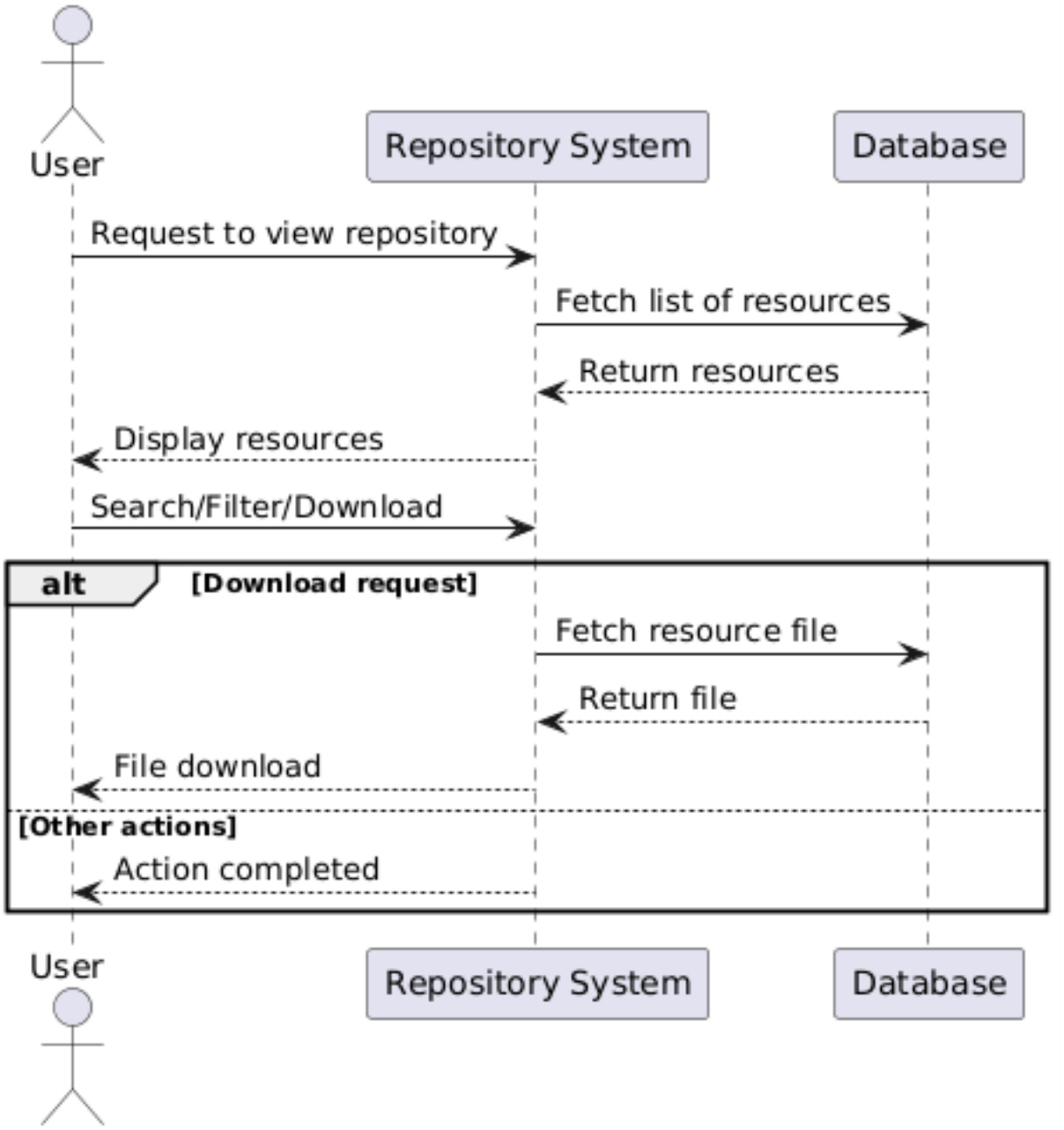

Figure 5: Sequence Diagram for Repository Access

3.3 Class Diagrams

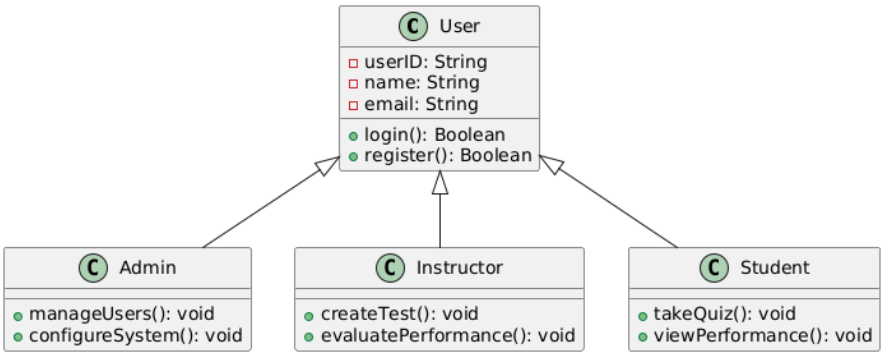

Figure 6:class Diagram for Test and Question Management

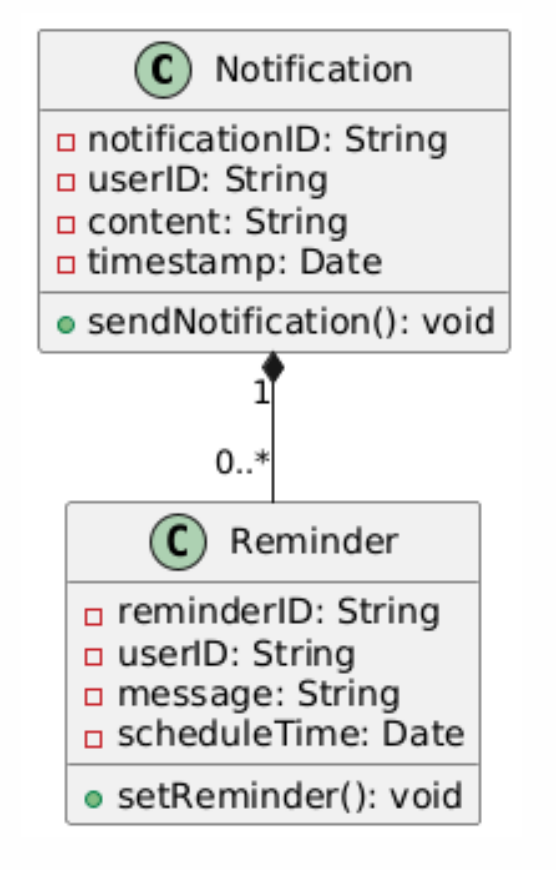

Figure 7:class Diagram for Notifications and Reminders

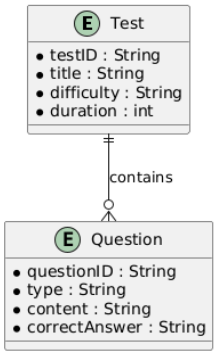

Figure 8: ER Diagram for Notifications and Reminders

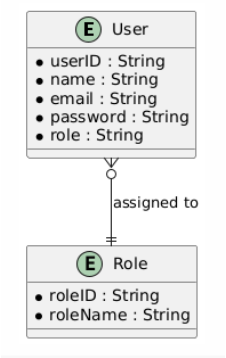

Figure 9: ER Diagram for Notifications and Reminders

4 Design Details

4.1 Module Descriptions

- **Authentication:** Uses JWT-based secure token generation to validate users.
- **Test Management:** Features question randomization and customizable difficulty levels.
- **Reporting:** Generates detailed analytics, including time spent and accuracy metrics.
- **Notifications:** Utilizes a scheduler to send alerts for test-related activities.

5 Deployment Considerations

The application is hosted on a cloud platform to ensure scalability and availability.

- **Frontend**HTML,CSS,JS
- **Backend**MERN,PHP