HARAMAYA UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF SOFTWARE ENGINEERING

YEAR III – SEMESTER I

Advanced Programing

Group project Assignment

No.	Name	ID
1	Yabets Tinsae Dube	1040/15
2	Tesfalem Markos	0993/15
3	Abdi Tamene Tekuma	0008/15
4	Yosan Gonfa Merga	1084/15
5	Oliad dandana	1786/15

Feb. 2025

HARAMAYA | OROMIA | ETHIOPIA

Software Requirements Specification

Version 1.0
<<Annotated Version>>

Feb 03, 2025

University Forum

Table of Contents

1.0. Introduction

The University Forum is a Java-based application designed to facilitate academic discussions among students and faculty. Users can post questions, reply to threads, and upvote/downvote responses.
The application integrates various Java concepts, including multithreading, JDBC, REST APIs, and GUI development, ensuring an efficient and user-friendly experience.

1.1. Purpose

The purpose of this document is to outline the functional and non-functional requirements of the University Forum Application. This document will serve as a guide for developers, testers, and stakeholders to ensure the application meets the desired objectives

1.2. Scope of Project

The University Forum Application will provide the following functionalities:

- User registration and authentication.
- Posting questions and creating threads.
- Replying to existing threads.
- Upvoting and downvoting answers.

- Viewing threads and answers in a user-friendly GUI.
- Storing data in a relational database using JDBC.
- Handling multiple concurrent users using multithreading.
- Exposing REST APIs for integration with other systems

1.3. Glossary

Term	Definition
Forum Thread	A discussion topic where users can post questions and replies.
Post	A message created by a user, either as a question or an answer.
Upvote/Downvote	A system to rate responses, increasing or decreasing their visibility.
Multithreading	A Java programming concept that enables concurrent execution of tasks to improve responsiveness.
JDBC (Java Database Connectivity)	An API that allows Java applications to interact with a relational database.
REST API (Representational State Transfer API)	A web service communication standard that allows the system to exchange data over HTTP.
GUI (Graphical User Interface)	The visual interface through which users interact with the forum system.
Authentication	The process of verifying user identity before granting access.
Database	A structured collection of data where forum posts, users, and interactions are stored.

1.4. References

Swing/AWT for GUI Development: Java Swing Tutorial - https://docs.oracle.com/javase/tutorial/uiswing/

1.5. Overview of Document

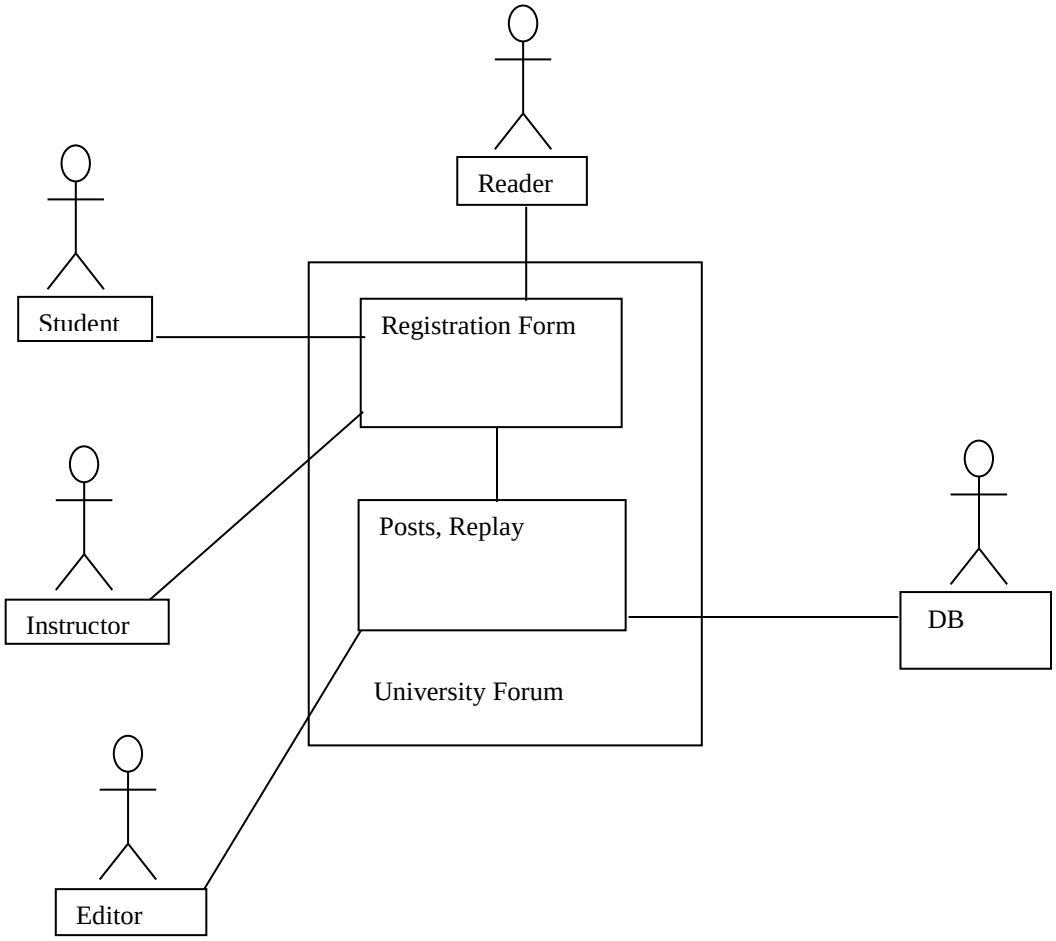
The University Forum Application is a web-based platform designed to facilitate communication and knowledge sharing among students, faculty, and staff. The application will allow users to post questions, reply to threads, and upvote/downvote answers. It will leverage Java technologies such as multithreading, JDBC, REST APIs, and a graphical user interface (GUI) to provide a robust, scalable, and user-friendly solution

.

2.0. Overall Description

2.1 System Environment

This system is very important for the university to perform the task of the web based platform For knowledge sharing and communication among different sectors.



2.2 Hardware Requirements

Server: Minimum 2 GHz CPU, 8 GB RAM, 100 GB storage

Client: Minimum 1 GHz CPU, 4 GB RAM, 20 GB storage

2.3 Software Requirements

Operating System: Windows, macOS, or Linux

Database: MySQL or PostgreSQL

Application Server: Apache Tomcat

Development Environment: IntelliJ IDEA, Eclipse or Apache Netbeans.

Java Version: JDK 11 or higher

APIs: Java RESTful APIs

Frameworks: Spring Boot, JavaFX

2.4 Functional Requirements Specification

2.3 User Characteristics

1. Student Users

Students will use the forum to ask questions, seek help, and engage in discussions related to their courses.

2. Instructor Users

Instructors will use the forum to provide answers, give guidance, and facilitate discussions among students.

2.5 Non-Functional Requirements

1 Performance

The system must handle up to 1,000 simultaneous user connections without performance degradation. Response time for user actions should be less than 2 seconds.

2 Security

User data must be stored securely, following best practices for encryption and data protection.

Authentication mechanisms must prevent unauthorized access.

3 Scalability

The system must be scalable to accommodate future growth in the number of users and forum activities.

4 Usability

The user interface must be intuitive and easy to navigate for both students and instructors.

Clear documentation and user guides must be provided.

5 Maintainability

The code base must be well-documented and follow best practices for code organization and structure.

The system should be easy to update and maintain.

3.0. Requirements Specification

3.1 User Registration and Authentication

Users must be able to register with a username, email, and password.

Users must be able to log in and log out securely.

3.2 Forum Functionality

Post Questions: Users must be able to post questions in various categories.

Reply to Threads: Users must be able to reply to questions and existing threads.

Upvote/Downvote Answers: Users must be able to upvote or downvote answers to questions.

3.3 Multithreading

The server must handle multiple simultaneous user requests using Java multithreading

3.4 Database Connectivity

The application must connect to the database using JDBC for storing and retrieving data.

3.5 REST APIs

The application must expose REST APIs for various functionalities such as posting questions, replying to threads, and voting.

3.6 Graphical User Interface (GUI)

The application must have a user-friendly GUI built with JavaFX, allowing users to interact with the forum easily.

3.7 External Interface Requirements

1.1 User Interfaces

The University Forum Application will provide a Graphical User Interface (GUI) for users to interact with the system. The GUI will be developed using Java Swing or JavaFX and will include the following screens:

1.1.1 Login/Registration Screen

Purpose: Allow users to log in or register for a new account.

Components:

Username and password fields.

"Login" button.

"Register" button to navigate to the registration screen.

Error messages for invalid credentials or registration failures.

11.2 Home Screen

Purpose: Display a list of threads sorted by recent activity or vote count.

Components:

List of threads with titles, descriptions, and timestamps.

"Post Question" button to navigate to the post question screen.

Search bar to filter threads by keywords.

1.1.3 Thread Detail Screen

Purpose: Display the details of a specific thread and its associated replies.

Components:

Thread title and description.

List of replies with usernames, reply text, timestamps, and vote counts.

"Reply" button to post a new reply.

Upvote and downvote buttons for each reply.

1.1.4 Post Question Screen

Purpose: Allow users to post new questions.

Components:

Title and description input fields.

"Post" button to submit the question.

"Cancel" button to return to the home screen.

1.1.5 Reply Screen

Purpose: Allow users to reply to a thread.

Components:

Text area for entering the reply.

"Submit" button to post the reply.

"Cancel" button to return to the thread detail screen.

1.2 Hardware Interfaces

The application will run on standard desktop and laptop systems with the following minimum requirements:

Processor: Intel i3 or equivalent.

RAM: 4 GB.

Storage: 500 MB of free disk space.

Display: 1024x768 resolution or higher.

1.3 Software Interfaces

Operating System: The application will be compatible with Windows, macOS, and Linux.

Database: MySQL will be used for data storage. The application will interact with the database using JDBC.

Web Server: A Java-based server (e.g., Apache Tomcat) will host the REST APIs.

APIs: The application will expose REST APIs for communication between the GUI and the server.

1.4 Communication Interfaces

The application will use HTTP/HTTPS protocols for communication between the client (GUI) and server.

3.8 Detailed Non-Functional Requirements

3.3.1 Logical Structure of the Data

The University Forum Application will store and manage data in a structured and efficient manner using a relational database.

3.3.2 Security

The University Forum Application will implement robust security measures to protect user data and ensure secure interactions.

Authentication

- User passwords shall be stored in the database using secure hashing algorithms (e.g., bcrypt).
- The system shall enforce strong password policies (e.g., minimum length, special characters).
- The system shall implement session management to track authenticated users.
- Authorization
- Only authenticated users shall be allowed to post questions, reply to threads, and vote on replies.
- Users shall only be able to edit or delete".

Data Encryption

- All sensitive data (e.g., passwords) shall be encrypted during transmission using HTTPS.
- Database backups shall be encrypted to prevent unauthorized access.

Input Validation

- The system shall validate all user inputs to prevent SQL injection, XSS, and other attacks.
- The system shall sanitize user inputs before storing them in the database.

Logging and Monitoring

- The system shall log all security-related events (e.g., failed login attempts, unauthorized access attempts).
- The system shall monitor for suspicious activities and alert administrators in real-time.

Conclusion

The University Forum Application will provide a robust platform for students and faculty to share knowledge and collaborate. By leveraging Java technologies such as multithreading, JDBC, REST APIs, and GUI, the application will be scalable, efficient, and user-friendly.