UE22CS341A: Software Engineering
Case Study

Unit 1 Deliverable

NAME 1: Sahana Bhat

SRN 1: PES1UG22CS500

NAME 2: Shama Kiran

SRN 2: PES1UG22CS540

1. Introduction

1.1 Purpose

This document specifies the requirements for the Peer Mentoring/Tutoring system. The system allows mentors and mentees to perform various tasks such as managing classes, submitting and reviewing assignments, tracking marks, and communicating through a secure and user-friendly interface. The purpose of this system is to enhance peer-to-peer learning by providing a platform that supports academic collaboration and mentorship.

1.2 Scope

The system is designed to facilitate communication, assignment management, and grade tracking between mentors (teachers) and mentees (students). It aims to provide a consistent user experience and efficient data management. The key functionalities within the scope of this project are:

- User Registration and Profile Management: Both mentors and mentees can sign up for the system. Users can view and edit their profiles, including adding a profile picture and updating personal information. They can also change their passwords as necessary.
- Class Management: Mentors can create and manage classes by adding or removing students. They can assign marks to students for various tasks and assessments. The system allows mentors to update or edit these marks as needed.
- Assignment Management:Mentors can upload assignments that are distributed to all students in their respective classes. Students can download these assignments, complete them, and submit them back through the portal. Once submitted, assignments cannot be modified.

- Notice and Communication System: Mentors can write and distribute notices to all students within their class, ensuring effective communication of important updates and announcements. Students can also view a list of all mentors available on the platform and send messages to any mentor. Mentors can see all incoming messages from students in their inbox.
- Search Functionality: Users can search for specific students or mentors using a search option available at the top of the student and mentor list pages. This feature helps in quickly finding relevant profiles and information.
- Marks and Performance Tracking: Students have access to a section where they can view the marks assigned to them by mentors. Mentors can see the entire history of marks given to each student through their marks profile, allowing for performance tracking and necessary adjustments.
- Session Feedback and Reporting: After interacting with mentors or completing assignments, students can provide feedback. The system can generate reports based on marks, assignment completion, and user feedback to help improve the quality of mentoring and tutoring.

This scope ensures that "Class Manager" functions as a comprehensive tool for peer-to-peer learning, supporting both administrative and academic tasks essential for effective mentoring and tutoring within educational environments.

1.3 Definitions, Acronyms, and Abbreviations

- Class Manager: A web-based peer mentoring/tutoring system where mentors (teachers) and mentees (students) can sign up, manage classes, communicate, and track academic progress.
- Mentor: A peer or senior student responsible for guiding mentees, managing classes, uploading assignments, and providing grades and feedback.
- Mentee: A student who receives guidance from mentors, submits assignments, and tracks their academic progress through the system.
- Assignment: A task or project that mentors upload for mentees to complete and submit back through the Class Manager portal.
- Notice: A message or announcement that mentors can send to all students in their class to communicate important information or updates.
- Inbox: A feature that allows mentors to view messages received from students, facilitating communication within the system.
- Marks Profile: A section where mentors can view and edit the marks they have given to students. Students can also view their marks through this section.
- Session ID: A unique identifier generated when students book a mentoring session or submit an assignment, used to track and manage individual sessions or tasks.
- Profile: A personal information page for each user, including name, contact details, and profile picture. Users can update their profile information and manage settings.

- Feedback: Input provided by students about their mentoring sessions, assignments, or overall experience, used to improve the quality of tutoring services.
- Search Option: A feature that allows users to search for specific mentors or mentees by name or other criteria within the Class Manager system.

This list aligns with the key functionalities and roles within our peer mentoring/tutoring system, providing clear definitions for users and stakeholders.

1.4 References

- https://www.djangoproject.com/
- https://www.w3schools.com/html/
- https://www.w3schools.com/css/
- https://dev.mysql.com/doc/

1.5 Overview

This document outlines the functional and non-functional requirements for the "Class Manager" peer mentoring/tutoring web application. The system enables effective management of mentoring sessions, class assignments, communication, and grade tracking between mentors and students. This document covers system features, interface requirements, and hardware and software specifications.

2. Overall Description

2.1 Functional Requirements

The "Class Manager" system must meet the following functional requirements:

1. User Registration and Authentication: Both mentors and students must be able to sign up for the system using unique credentials. The system should authenticate users to ensure secure access.
2. Profile Management: Users should have the ability to view and edit their profiles, including personal details and profile pictures. Users should also be able to change their passwords if necessary.
3. Class and Student Management: Mentors can create classes and add students to them. They can view and manage student lists within each class.
4. Assignment Management: Mentors can upload assignments for their classes, which students can download. Students should be able to submit completed assignments, with the restriction that submissions cannot be modified once uploaded.
5. Marks and Performance Tracking: Mentors can assign and edit marks for students. Students can view their grades through a dedicated marks section.

6. Notice and Messaging System: Mentors can post notices that will be sent to all students in their class. Both mentors and students can send and receive messages within the system to facilitate communication.
7. Search Functionality: Users should be able to search for specific students or mentors by name or other identifying information.
8. Session Feedback: Students should be able to provide feedback on their mentoring sessions and assignments, which mentors can use to improve the quality of tutoring.
9. Report Generation: The system should be capable of generating reports on student performance, class attendance, and feedback summary for administrative and analytical purposes.
10. User Logout: All users should have the option to securely log out of the system to ensure privacy and security.

2.2 Non functional requirements

All application data for the peer tutoring system will be stored in a PostgreSQL database, which must be installed on the host server. The system will be developed using Django, which is compatible with major web servers like Apache or Nginx, and can be run on various operating systems, including Windows, Linux, and macOS. Initially, the application can be hosted on a virtual private server (VPS) or a standard desktop computer capable of running both web and database servers. However, as user traffic grows, it may be necessary to upgrade to a cloud-based hosting solution with scalable infrastructure to ensure optimal performance and reliability. Performance limits will be monitored and adjusted as needed based on regular stress testing to handle increased traffic and maintain system availability.

2.3 Hardware Requirements

- A desktop, laptop, or server with a stable internet connection.
- Minimum 100GB of hard disk space for application data and backups.
- 8GB of RAM to ensure smooth operation and multitasking.
- Compatible with Windows 10, Linux, or macOS operating systems.

2.4 Software Requirements

2.4.1 **Server Side**

Programming Language: Python 3.x

Web Framework: Django 4.x

Web Server: Nginx or Apache

Database: PostgreSQL 13.x

2.4.2 **Client Side**

Languages: HTML5, CSS3, JavaScript

Libraries/Frameworks: Bootstrap 5, jQuery

Browser Compatibility: Supports latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari

Operating System: Compatible with Windows 10, Linux, macOS

2.4.3 **Python**

Python is a versatile, high-level programming language widely used for web development, data analysis, artificial intelligence, and more. It is known for its readability and extensive libraries. In this project, Python will be used for server-side scripting to handle requests, manage database interactions, and implement the logic of the peer tutoring system.

2.4.4 **Django**

Django is a high-level Python web framework that promotes rapid development and clean, pragmatic design. It provides built-in features like user authentication, a robust admin interface, and database management, making it ideal for building scalable web applications such as our peer tutoring platform.

2.4.5 **HTML and CSS**

HTML (HyperText Markup Language) is the standard language for creating web pages, and CSS (Cascading Style Sheets) is used for styling the HTML content. Together, they form the backbone of the web interface, allowing the creation of structured, visually appealing pages. HTML is used for structuring content, while CSS controls the layout, colors, fonts, and other visual aspects.

2.4.6 **JavaScript**

JavaScript is a lightweight, dynamic programming language essential for adding interactivity to web pages. It allows client-side script execution, enabling features like form validation, dynamic content updates, and interactive UI elements without needing to reload the page. JavaScript is integral to enhancing user experience in modern web applications.External Interface Requirements

2.5 User Interfaces

- Web Interface: A responsive and intuitive web interface for both students and teachers, accessible through modern web browsers. The interface will include forms for user registration, login, session booking, assignment submission, and messaging.
- Profile Management: Users can manage their profiles, upload profile pictures, edit personal details, and change passwords through dedicated profile management pages.
- Notifications: A notification system for displaying messages, assignment deadlines, and other important updates directly on the user dashboard.
- Search and Filter: Search and filter options for students and teachers to easily find and interact with one another.

3. Error Handling

- Error Notifications: The system shall provide clear notifications to users in case of errors during interactions, such as failed login attempts or submission issues.
- Error Logging: All errors, including network issues and invalid inputs, shall be logged for later analysis by administrators to improve system reliability and troubleshoot issues.

4. Environmental Requirements

Operational Temperature**:** The system shall be operational within a temperature range of 10°C to 40°C to ensure proper functioning of server hardware and other components..

5. Requirements Traceability Matrix (RTM)

The Requirements Traceability Matrix (RTM) ensures that all requirements for the Class Manager system are covered by design, development, and testing activities. Below is a simplified RTM example for the Class Manager peer mentoring/tutoring system:

```
# Requirements Traceability Matrix for Class Manager

| Requirement ID | Requirement Description | Module | Test Case ID | Verification Method | Priority |
|----------------|-------------------------|--------|--------------|---------------------|----------|
| FR-01 | User Registration and Authentication | User Management | TC-01 | Functional Testing | High |
| FR-02 | Profile Management | User Management | TC-02 | Functional Testing | High |
| FR-03 | Class and Student Management | Admin Management | TC-03 | Functional Testing | High |
| FR-04 | Assignment Management | Assignment Module | TC-04 | Functional Testing | High |
| FR-05 | Marks and Performance Tracking | Grading Module | TC-05 | Functional Testing | High |
| FR-06 | Notice and Messaging System | Communication Module | TC-06 | Functional Testing | Medium |
| FR-07 | Search Functionality | Search Module | TC-07 | Functional Testing | Medium |
| FR-08 | Session Feedback | Feedback Module | TC-08 | Functional Testing | Medium |
| FR-09 | Report Generation | Reporting Module | TC-09 | Functional Testing | Medium |
| FR-10 | User Logout | User Management | TC-10 | Functional Testing | High |
| NFR-01 | Database Compatibility (PostgreSQL) | Database Management | TC-11 | Integration Testing | High |
| NFR-02 | Web Server Compatibility | Server Management | TC-12 | Integration Testing | High |
| NFR-03 | Performance and Scalability | System | TC-13 | Performance Testing | High |
| NFR-04 | Security (Data Protection) | Security Module | TC-14 | Security Testing | High |
| NFR-05 | Browser Compatibility | UI/UX | TC-15 | Compatibility Testing | Medium |
| NFR-06 | Usability (Responsive Design) | UI/UX Design | TC-16 | Usability Testing | Medium |
```