SOFTWARE REQUIREMENTS SPECIFICATION(SRS) FOR SKILLSCORE

Student Activity points Management platform

Prepared by Group no: FN- 8

SUZZANNE SADIQUE **B220564CS**

RUSHDA P P **B220503CS**

RACHEL PAUL **B221138CS**

Table of Contents

1. **Introduction**
   - 1.1 Purpose
   - 1.2 Scope
   - 1.3 Assumptions and dependencies
   - 1.4 Glossary
   - 1.5 References
   - 1.6 Overview of Document
2. **Overall Description**
   - 2.1 System Overview
   - 2.2 User Classes & Characteristics
   - 2.3 Constraints
3. **Functional Requirements Specification**
   - 3.1 Student Features
   - 3.2 Faculty Advisor Features
   - 3.3 Faculty Features
   - 3.4 Notifications
4. **Non-Functional Requirements**
   - 4.1 Security
   - 4.2 Performance
   - 4.3 Scalability
   - 4.4 Usability
   - 4.5 Accessibility
5. **Intended Users**
6. **Use Case Diagram**
7. **Class Diagram**
8. **Database Diagram**
9. **Conclusion**
10. **Group Log**

1. Introduction

1.1 Purpose

The **SkillScore Activity Point Portal** aims to streamline the process of tracking, submitting, and verifying student activity points. It provides role-based access for students, faculty, and faculty advisors to manage event participation records effectively.

1.2 Scope

- Students can submit event participation requests and track their approval status.
- Faculty advisors review and approve/reject student requests with feedback.
- Faculty members manage events and oversee student participation.
- Dashboards provide insights into points, events, and pending actions.

1.3 Assumptions & Dependencies

- The system assumes that all users have valid institutional credentials.
- It relies on a centralized database for request tracking and event management

1. 4 Glossary

- **Activity Points:** Credits awarded to students for participating in extracurricular activities.
- **Faculty Advisor:** A faculty member responsible for reviewing student requests.
- **Dashboard:** A visual summary of user data, including pending requests, approvals, and upcoming events

1.5 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.

1.6 Overview of Document

This document details both the functional and non-functional requirements of the SkillScore system.

.

---

2. Overall Description

2.1 System Overview

The portal allows students to submit participation requests for events, faculty advisors to review and approve/reject these requests, and faculty members to manage event records. The system maintains transparency by tracking total earned points and displaying upcoming events.

2.2 User Classes & Characteristics

- **Student**s: Can submit requests, view events, and track approval status.
- **Faculty Advisors**: Approve or reject submissions, and monitor pending requests, add and manage events.
- **Faculty Members**: Can add and manage events, view pending requests.

2.3 Constraints

- The system must be accessible only via institutional login.
- Event submissions require supporting documents for verification.

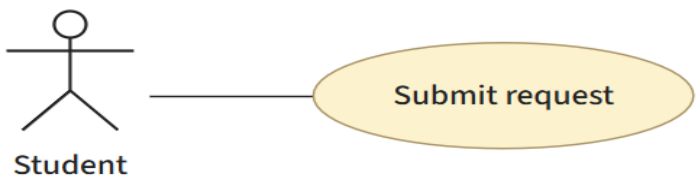
---

3. Functional Requirements

3.1 Student Features

- **Submit activity points requests, along with proof.**

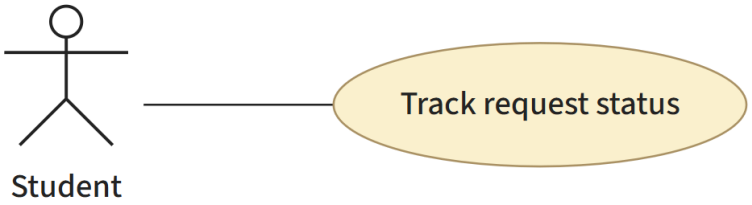
Use case:submit request



Step 1: The student logs in using their institutional credentials.
Step 2: They navigate to the “Submit Activity Request” option in the dashboard.
Step 3: They enter event details such as event name, date, type, and a short description.
Step 4: The student uploads proof of participation (e.g., certificate, photos, etc.).
Step 5: They submit the request for faculty advisor’s approval and additional faculty if needed.

- **View approval status (Pending, Approved, Rejected) with faculty comments.**

Use case: track request status



Step 1: The student logs into the portal and navigates to the **"My Requests"** section.
Step 2: They can view the approval status of each request:
  - **Pending:** Awaiting faculty advisor approval.
  - **Approved:** Request is verified and accepted.
  - **Rejected:** Request is denied with comments from faculty.

Edit Submitted Requests

- **Step 1:** The student navigates to their submitted request while it is still **Pending**.
- **Step 2:** They modify event details or upload new proof.
- **Step 3:** They save and resubmit the updated request.

Resubmit Rejected Requests

- **Step 1:** The student views the faculty advisor's comments on their rejected request.
- **Step 2:** They make necessary corrections and upload revised proof.
- **Step 3:** They resubmit the request for approval.

Track Total Points Earned

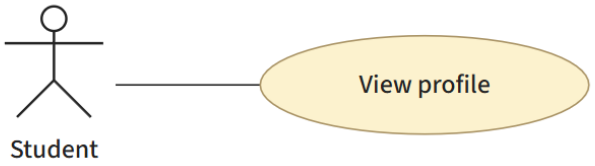
- **Step 1:** The student accesses the **Dashboard**.
- **Step 2:** A **circular progress tracker** visually displays their total points earned.

View Upcoming Events

- **Step 1:** The student clicks on the **"Upcoming Events"** tab.
- **Step 2:** They browse through a list of events where they can earn activity points.

View Profile Details

- **Step 1:** The student clicks on their profile.
- **Step 2:** They see their **personal details, points history, and request status**.



Add Additional Faculty If Required

- **Step 1:** While submitting a request, students see an option to add multiple faculty members.
- **Step 2:** They select additional faculty members responsible for reviewing the request.

3.2 Faculty Advisor Features

View Pending Student Requests

- Step 1: Faculty advisor logs into the portal using institutional credentials.
- Step 2: They navigate to the "Pending Requests" section.
- Step 3: They see a list of all student requests awaiting review.
- Step 4: They can use search and filter options to quickly locate specific requests.

Click on a Request to View Details

- Step 1: Faculty advisor selects a request from the list.
- Step 2: The system displays event details, uploaded proof, and any previous faculty comments.

Approve or Reject Requests

- Step 1: Faculty advisor reviews event details and uploaded proof.
- Step 2: They choose to approve or reject the request.
- Step 3: If rejecting, they enter a reason in the comments field.
- Step 4: The system notifies the student of the decision.

View Students Below Required Activity Points

- Step 1: Faculty advisor navigates to the “Students Below Required Points” section.
- Step 2: They see a list of students who have not met the required activity points.
- Step 3: They can send reminders to students or suggest events for them.

Dashboard Overview

- Step 1: Faculty advisor accesses their dashboard.
- Step 2: The dashboard provides a quick summary of:
    - Number of pending requests.
    - Total students assigned to them.
    - Upcoming events they created.

View All Assigned Students and Their Reports

- Step 1: Faculty advisor selects “Student” from the menu.
- Step 2: They see a list of all students under their supervision.
- Step 3: Clicking a student’s name shows their earned points, pending request and information about the student.

Add Events

- Step 1: Faculty advisor clicks “Add New Event.”
- Step 2: They fill in:
    - Event Name: Leadership Seminar
    - Date: March 10, 2025
    - Type: Leadership & Soft Skills
    - Points: 5
- Step 3: They submit the event, making it available for students to see.

View Events Submitted by Them

- Step 1: Faculty advisor goes to “My Events”.
- Step 2: They see all events they created along with participation data.

3.3 Faculty Features

Approve or Reject Participation Requests

- **Step 1:** Faculty selects a request from the list.
- **Step 2:** The system displays event details, uploaded proof, and any previous faculty comments.
- **Step 3:** Faculty reviews the information and chooses to **approve or reject** the request.
- **Step 4:** If rejected, faculty must provide a reason in the comments section.
- **Step 5:** The system notifies the student about the decision.

Add and Manage Events

- **Step 1:** Faculty clicks on **“Create New Event”** in the event management section.
- **Step 2:** They fill in details such as:
    - **Event Name:** AI Workshop
    - **Date:** April 15, 2025
    - **Type:** Technical Skills
    - **Points:** 10
- **Step 3:** They submit the event, making it available for students to see and register.

View Events Created

- **Step 1:** Faculty navigates to **“My Events”**.
- **Step 2:** They see a list of all events they have created.

3.4 Notifications

Student Notifications

- **Step 1:** When a student submits a request, they receive a confirmation notification.
- **Step 2:** If the request is approved, they receive an approval notification with updated points.
- **Step 3:** If the request is rejected, they receive a rejection notification along with faculty comments.
- **Step 4:** Students also get reminders for upcoming events.

Faculty Notifications

- **Step 1:** Faculty members receive notifications when new student requests need their approval.
- **Step 2:** They also get reminders about pending requests that have not been reviewed.

---

4. Non-Functional Requirements

4.1 Security

- The system ensures that only authorized users can access respective functionalities using institutional login credentials.
- All data transmissions are encrypted to protect student and faculty information.
- **Example:** A student cannot view another student's pending requests due to role-based access control.

4.2 Performance

- The system must handle concurrent users efficiently without lag.
- Student request submissions and faculty approvals should be processed in under 2 seconds.
- **Example:** Even if 500 students submit activity requests at once, the system should respond instantly.

4.3 Scalability

- The system must support an increasing number of users and events without degradation in performance.
- The database should accommodate expanding records seamlessly.

4.4 Usability

- The platform should have an intuitive and user-friendly interface accessible to students and faculty members with minimal training.
- Simple navigation, clear labels, and guided workflows ensure ease of use.
- **Example:** A first-time student user can submit an activity request without external help.

4.5 Accessibility

- The system must comply with WCAG (Web Content Accessibility Guidelines) to ensure usability for users with disabilities.
- Screen readers and keyboard navigation should be fully supported.

---

5. Intended Audience

5.1 Project Stakeholders

- These individuals oversee the project to ensure it aligns with both functional and non-functional requirements.
- They are responsible for **evaluating system performance**, ensuring the platform meets the defined objectives, and proposing improvements.
- **Example:** A university board member reviews the system's adoption rate to determine whether students are engaging with it effectively.

5.2 Students

- The primary users of the system who submit activity point requests and track their progress.
- They can **view request statuses**, update profiles, and register for events.

5.3 Faculty Advisors and Faculty Members

- These users interact with the system to review and approve or reject student requests.
- They also **manage student records, oversee event participation, and send notifications**.

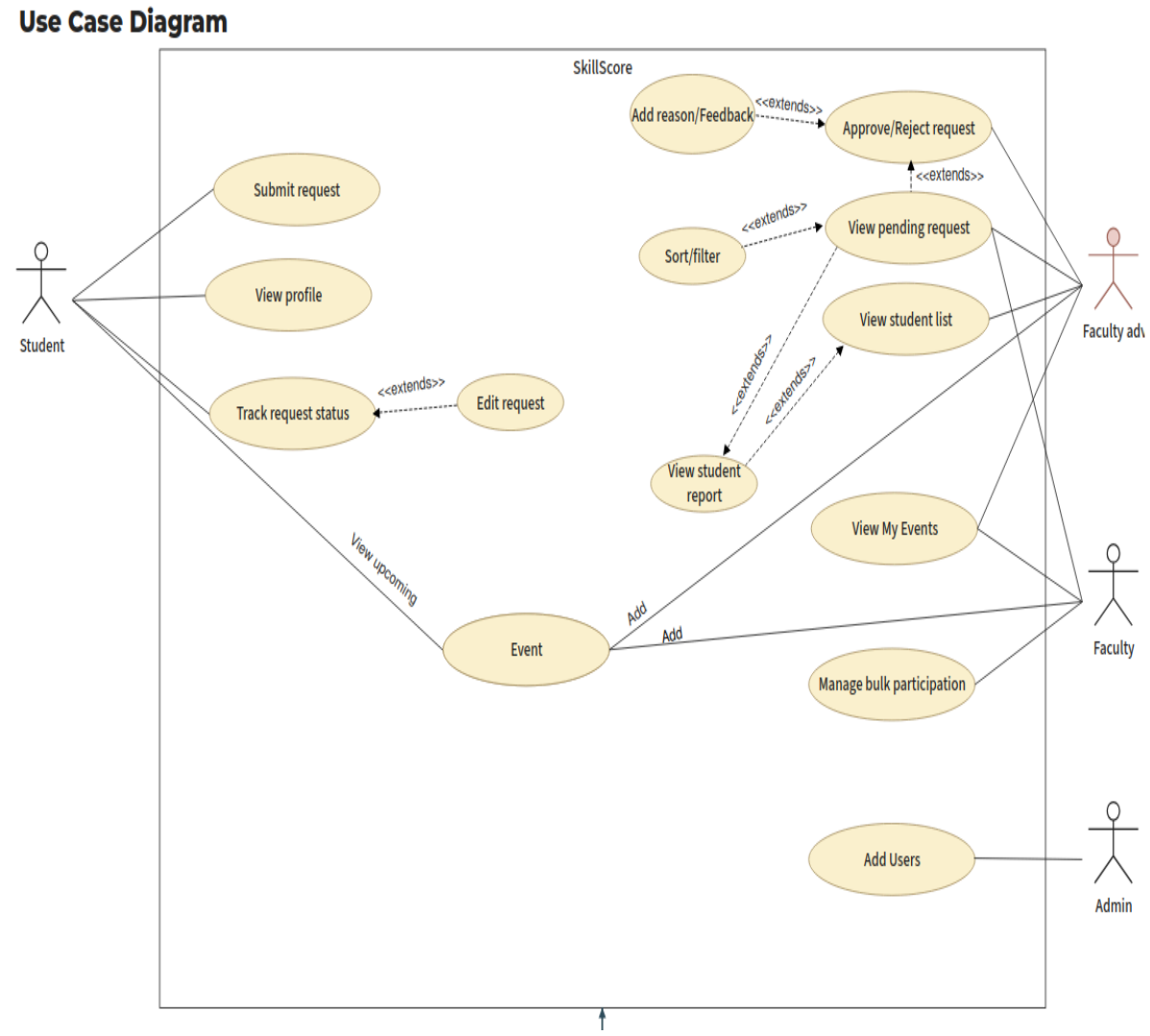
5.4 System Administrators

- Responsible for managing **user accounts, system configurations, and platform maintenance**.
- They ensure security policies are enforced and troubleshoot technical issues.

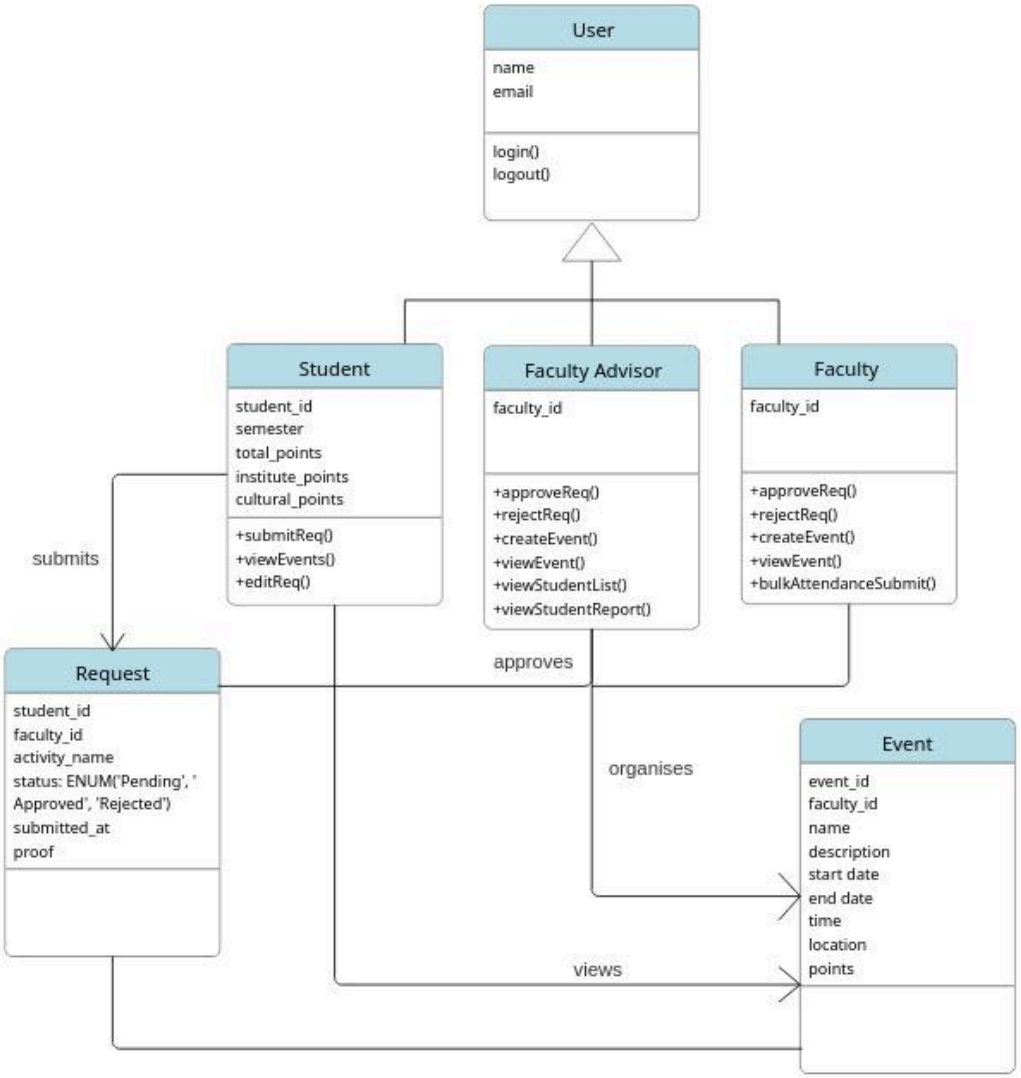
---

6. Use Case diagram

User Roles : Student, Faculty Advisor, Faculty, Admin



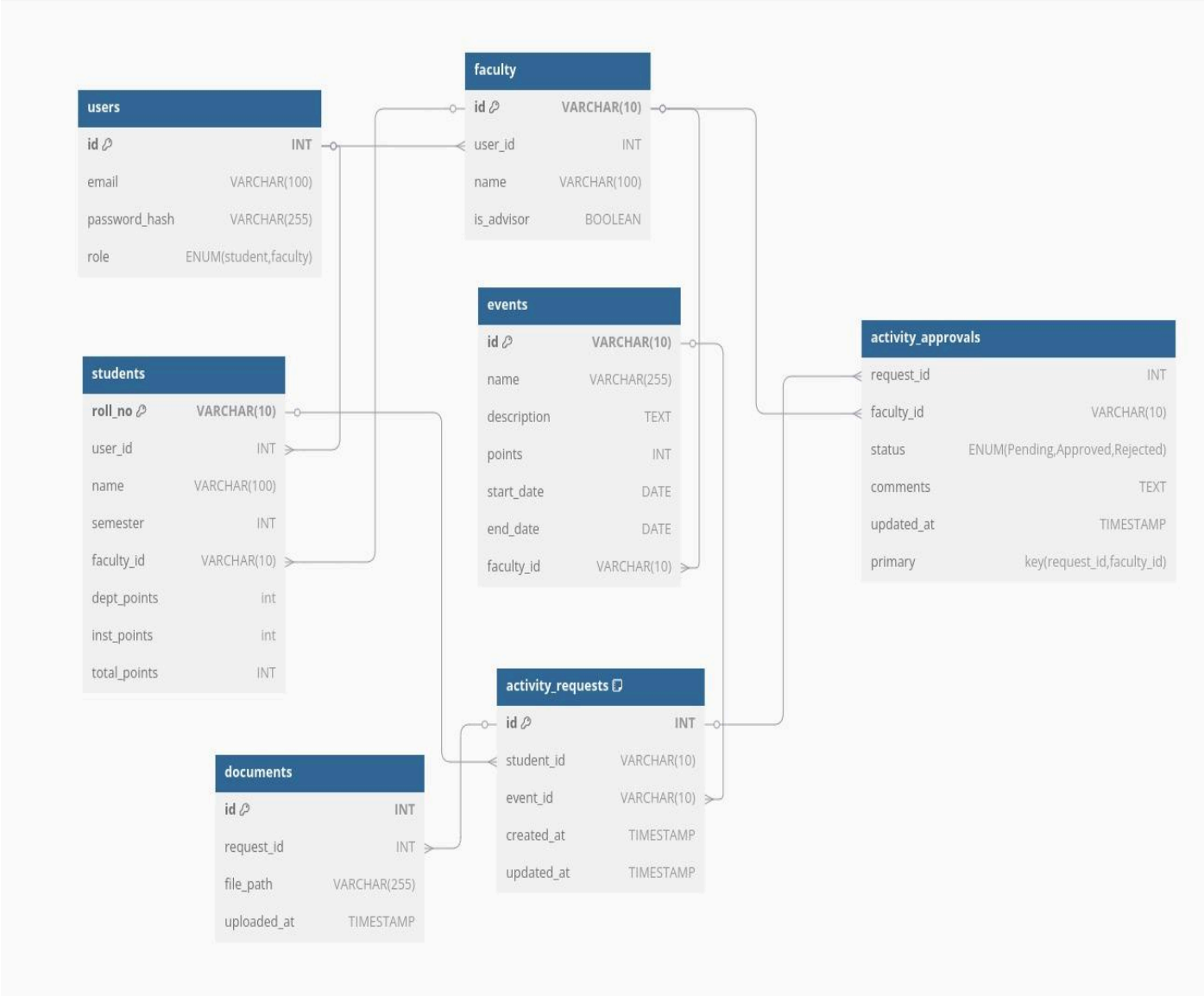
7. Class Diagram



The class diagram illustrates the object-oriented structure of the system, highlighting interactions between different classes.

- **User**: A base class representing both students and faculty, providing login/logout functionality.
- **Student**: A subclass of User, responsible for submitting activity requests and viewing event details.
- **Faculty Advisor & Faculty**: Both extend the faculty role, handling request approvals and event creation.
- **Request**: Represents an activity request submitted by students, pending faculty approval.
- **Event**: Defines details of activities, including name, schedule, and points.
  The diagram ensures clear role-based access control and functional responsibilities within the system.

8. Database Diagram



The database schema represents the structure of the system's relational database. It includes key entities such as users, students, faculty, events, activity_requests, and activity_approvals.

- **Users**: Stores authentication details for students and faculty.
- **Students**: Contains student-specific attributes such as semester and points.
- **Faculty**: Stores faculty details, including their role as advisors.
- **Events**: Represents activities that students can participate in.
- **Activity Requests**: Tracks students' requests for activity approvals.
- **Activity Approvals**: Maintains approval statuses from faculty.
- **Documents**: Stores supporting documents for activity requests.

9. Conclusion

This SRS document provides a structured approach to developing the **SkillShare Activity Point Portal**. It ensures that the system meets both functional and non-functional requirements while aligning with the UI design.

10. Group Log

Meeting Day	Time
16/02/25	6:00 pm -8:00 pm
18/02/25	6:00 pm -8:00 pm
20/02/25	6:00 pm -8:00 pm

NAME	Contributions
Suzzanne sadique	Design ,Functional requirements
Rushda pp	Use case diagram, class diagram
Rachel paul	Non functional requirements, database diagram