

Software Design And Architecture

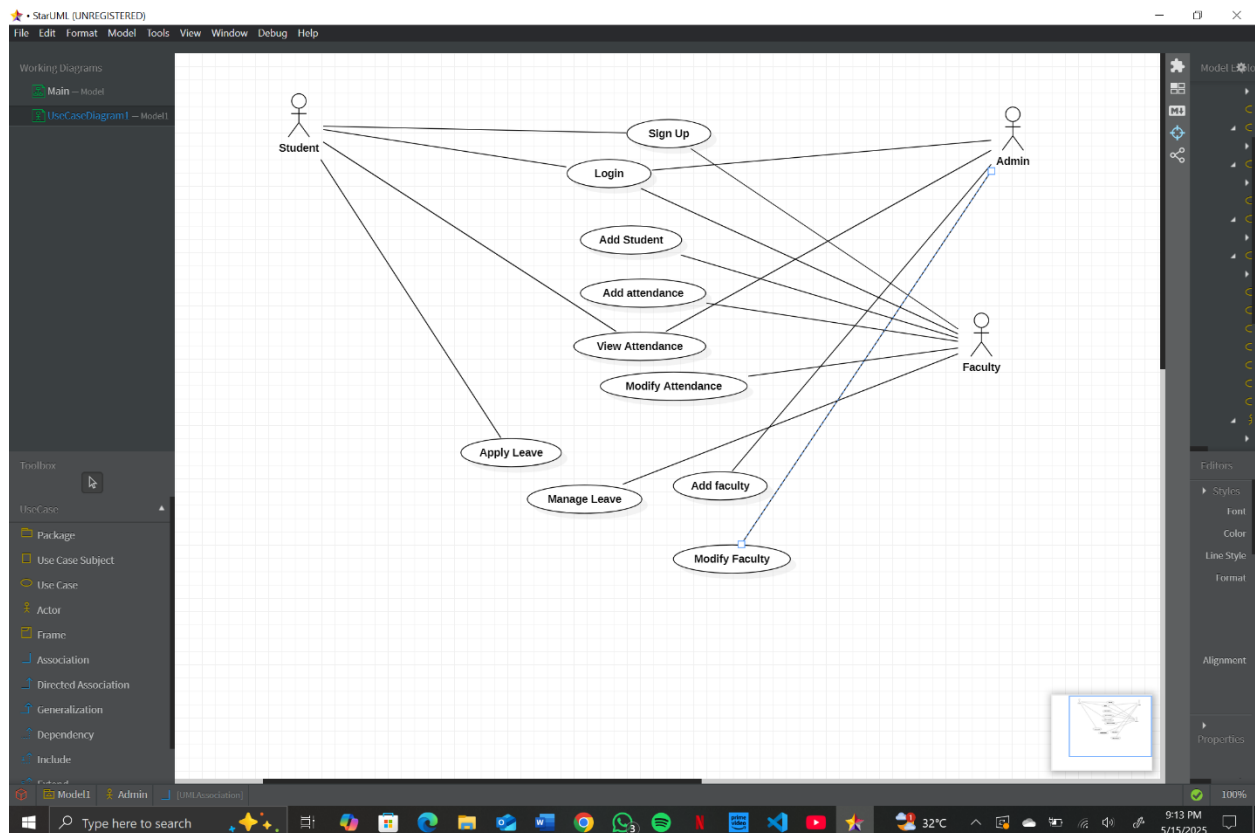
Assignment : 1

Name : Mian Habib Jan

Reg No: SP21-BSE-042

Semester Project: Attendance Management System

Chapter : 1 (Use Case Diagram)



Chapter : 2 (Fully Dressed Use Cases)

Use Case 1: Sign Up

Primary Actor: Student, Faculty, Admin.

Goal in Context: To allow new users to register in the system.

Preconditions: The user does not already have an account.

Trigger: The user navigates to the sign-up page.

Main Success Scenario:

1. The user provides personal details (e.g., name, email, role).
2. The system validates the input.
3. The system creates a new user account and notifies the user of successful registration.

Extensions:

- If the email already exists, the system prompts an error message.

Postconditions: The user is successfully registered and can log in.

Use Case 2: Login

Primary Actor: Student, Faculty, Admin.

Goal in Context: To authenticate and grant access to the system.

Preconditions: The user must have a valid account.

Trigger: The user submits login credentials.

Main Success Scenario:

1. The user enters their username and password.
2. The system verifies credentials.
3. The user is granted access to their respective dashboard.

Extensions:

- If the credentials are invalid, the system displays an error message.

Postconditions: The user is logged in.

Use Case 3: Add Student

Primary Actor: Admin

Goal in Context: To add a new student to the system.

Preconditions: The student is not already in the system.

Trigger: The admin navigates to the "Add Student" page.

Main Success Scenario:

1. The admin enters the students' details.
2. The system validates information.
3. The student is added to the database.

Extensions:

- If details are invalid, the system notifies the admin.

Postconditions: The student is successfully added.

Use Case 4: Add Attendance

Primary Actor: Faculty, Admin

Goal in Context: To record attendance for a class or group.

Preconditions: The class or group must already exist.

Trigger: The user navigates to the "Add Attendance" feature.

Main Success Scenario:

1. The user selects the class/group and the date.
2. The user marks attendance for individuals.
3. The system saves the attendance record.

Extensions:

- If the class does not exist, the system prompts an error.

Postconditions: Attendance is successfully recorded.

Use Case 5: View Attendance

Primary Actor: Student, Faculty, Admin

Goal in Context: To view attendance records.

Preconditions: The system must have attendance data.

Trigger: The user requests to view attendance.

Main Success Scenario:

1. The user selects the desired filters (e.g., date range, class).
2. The system displays the attendance records.

Extensions:

- If no records are found, the system notifies the user.

Postconditions: The user successfully views attendance.

Use Case 6: Modify Attendance

Primary Actor: Faculty, Admin

Goal in Context: To edit an existing attendance record.

Preconditions: There must be an existing attendance record to modify.

Trigger: The user navigates to the "Modify Attendance" feature.

Main Success Scenario:

1. The user selects the record to edit.
2. The user makes the necessary changes.
3. The system updates the attendance record.

Extensions:

If the record does not exist, the system notifies the user.

Postconditions: Attendance record is updated.

Use Case 7: Apply Leave

Primary Actor: Student

Goal in Context: To allow a student to request leave.

Preconditions: The user must be logged in.

Trigger: The user selects the "Apply Leave" option.

Main Success Scenario:

1. The user fills out a leave request form with relevant details.
2. The system sends the request to the faculty or admin for approval.

Extensions:

- If required details are missing, the system prompts an error.

Postconditions: Leave request is submitted successfully.

Use Case 8: Manage Leave

Primary Actor: Faculty, Admin

Goal in Context: To review and approve/reject leave requests.

Preconditions: There must be leave requests in the system.

Trigger: The user navigates to the "Manage Leave" page.

Main Success Scenario:

1. The user views pending leave requests.
2. The user approves or rejects a request.
3. The system updates the leave status.

Extensions:

- If no requests are pending, the system notifies the user.

Postconditions: Leave request is processed.

Use Case 9: Add Faculty

Primary Actor: Admin

Goal in Context: To add a new faculty member to the system.

Preconditions: The faculty member must not already exist in the system.

Trigger: The admin navigates to the "Add Faculty" page.

Main Success Scenario:

1. The admin enters the faculty details.
2. The system validates the data.
3. The faculty member is added to the system.

Extensions:

- If details are invalid, the system prompts an error.

Postconditions: Faculty member is successfully added.

Use Case 10: Modify Faculty

Primary Actor: Admin

Goal in Context: To update faculty details in the system.

Preconditions: The faculty member must already exist in the system.

Trigger: The admin selects the "Modify Faculty" option.

Main Success Scenario:

1. The admin selects the faculty record to edit.
2. The admin updates the necessary details.
3. The system saves the changes.

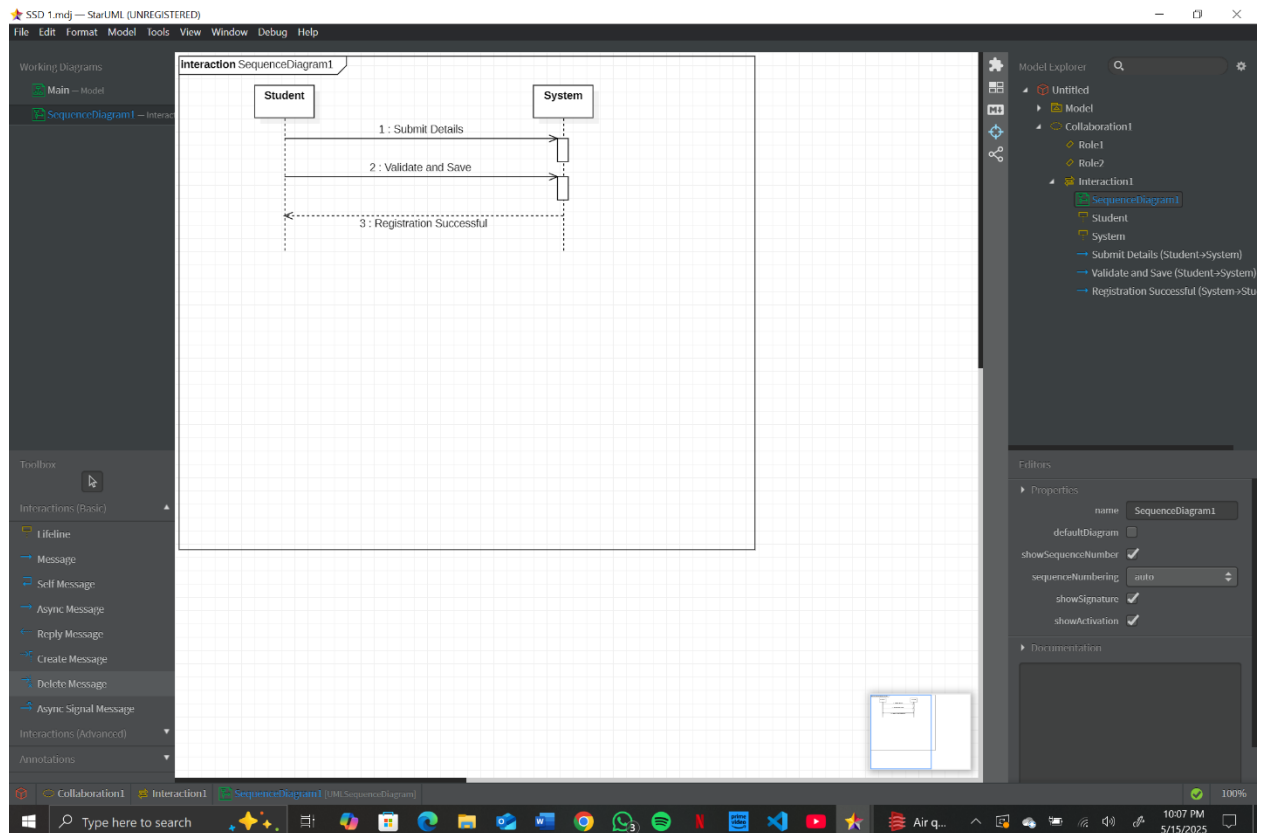
Extensions:

- If the faculty record does not exist, the system notifies the admin.

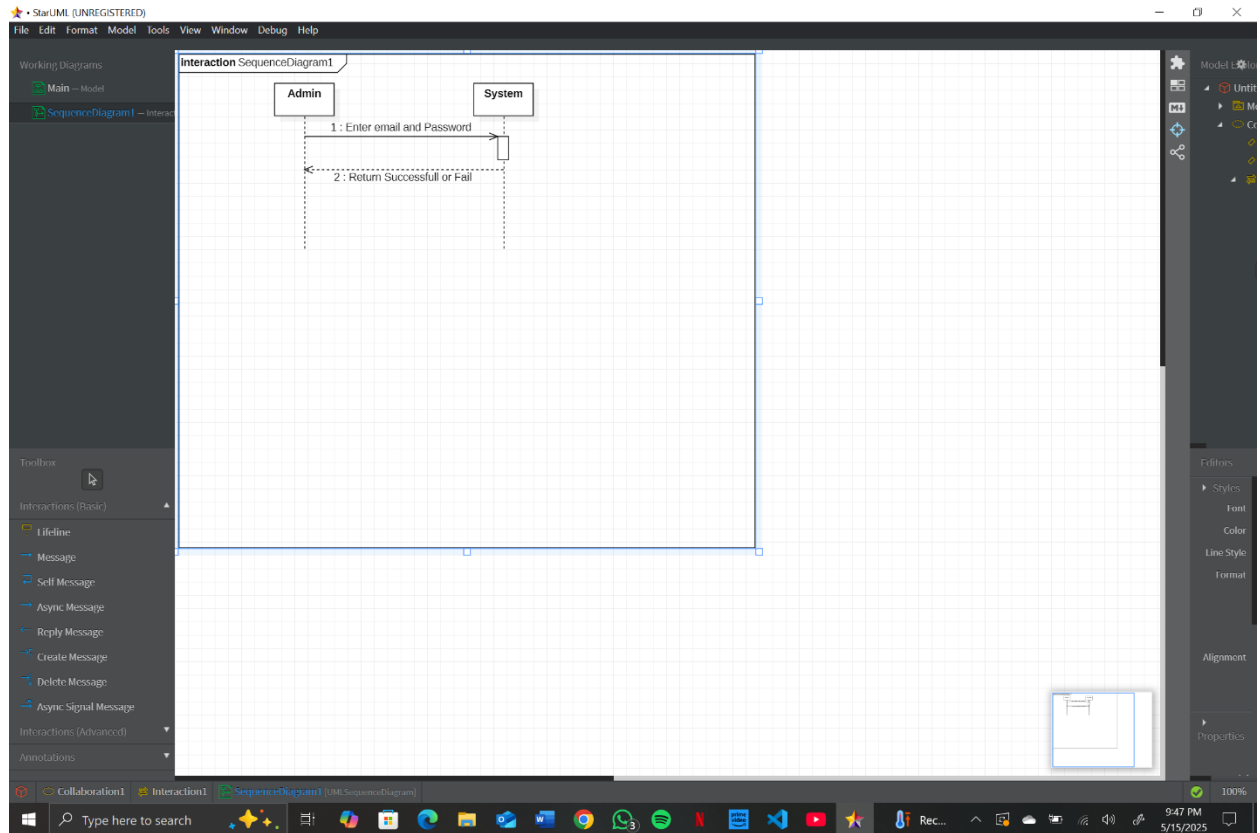
Postconditions: Faculty details are updated.

Chapter : 3 (SSD's)

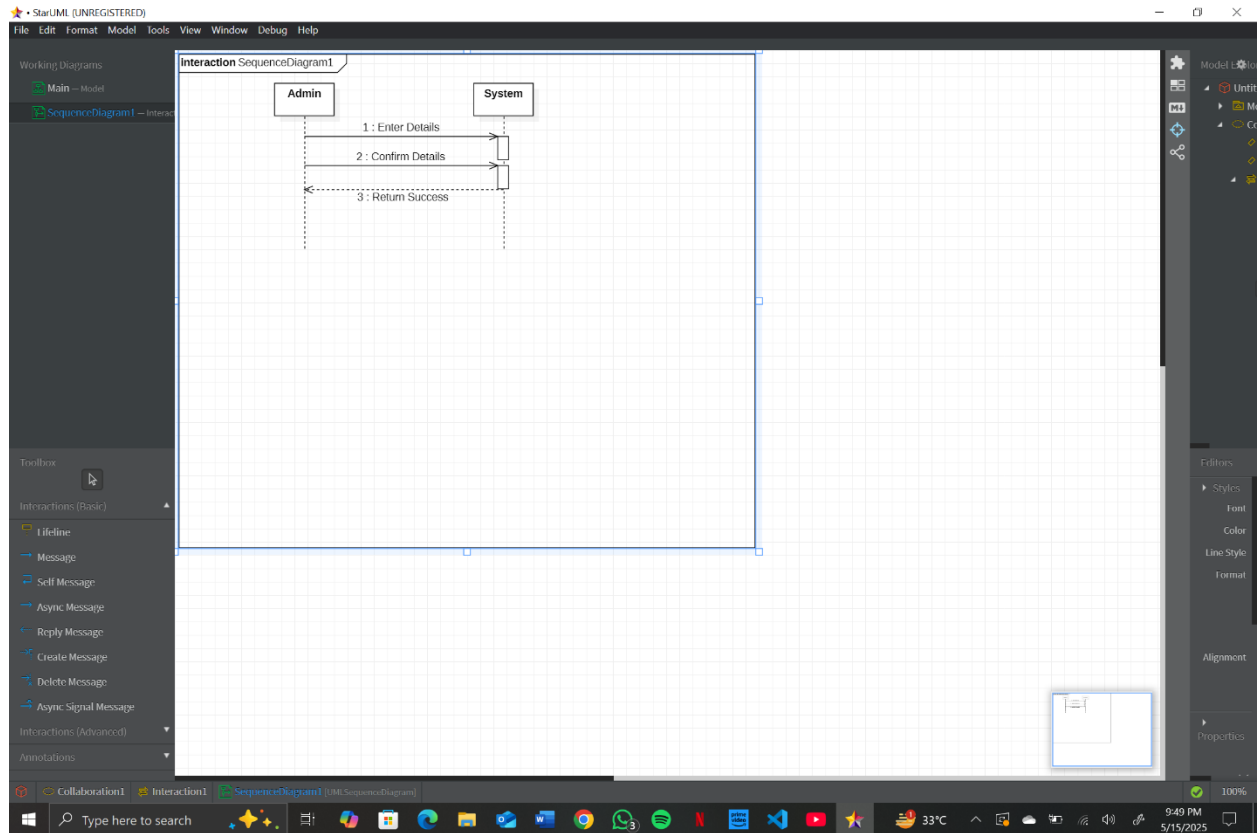
Use Case 1: Sign Up



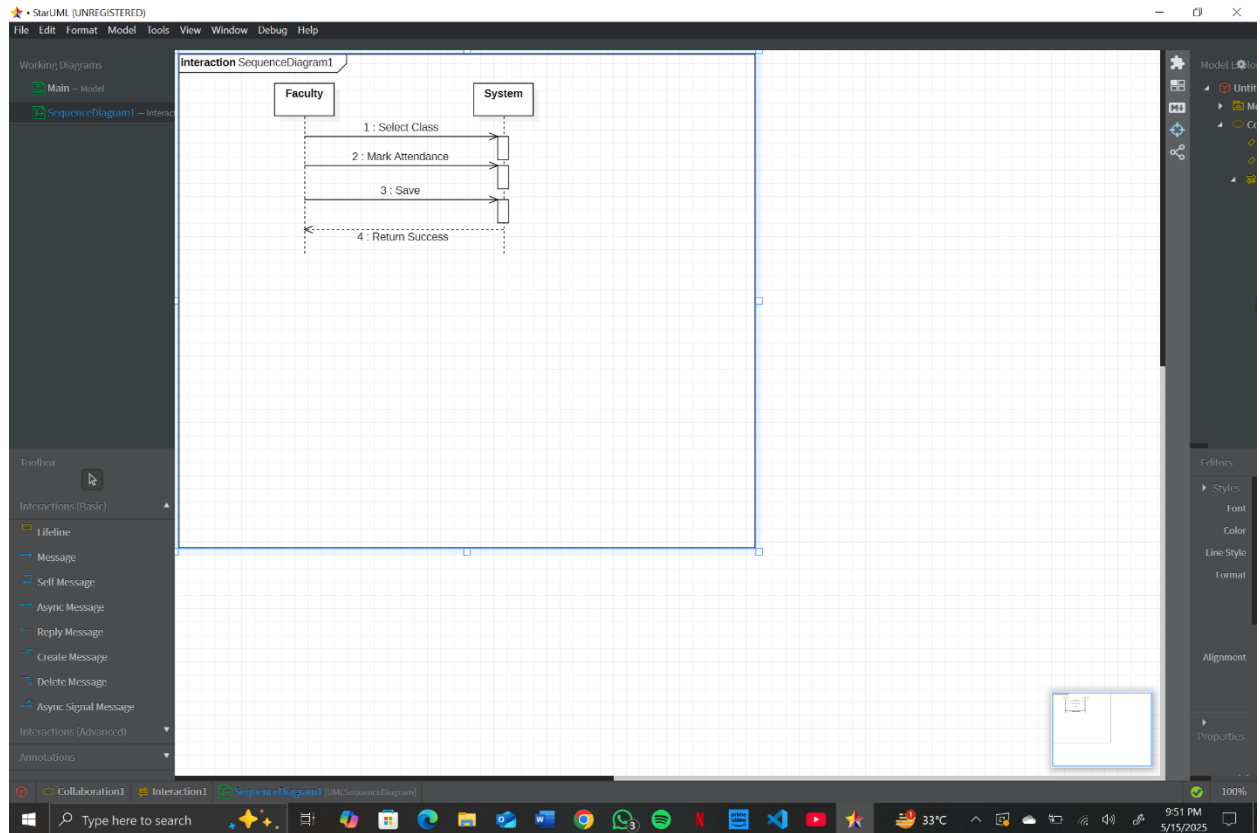
Use Case 2: Login



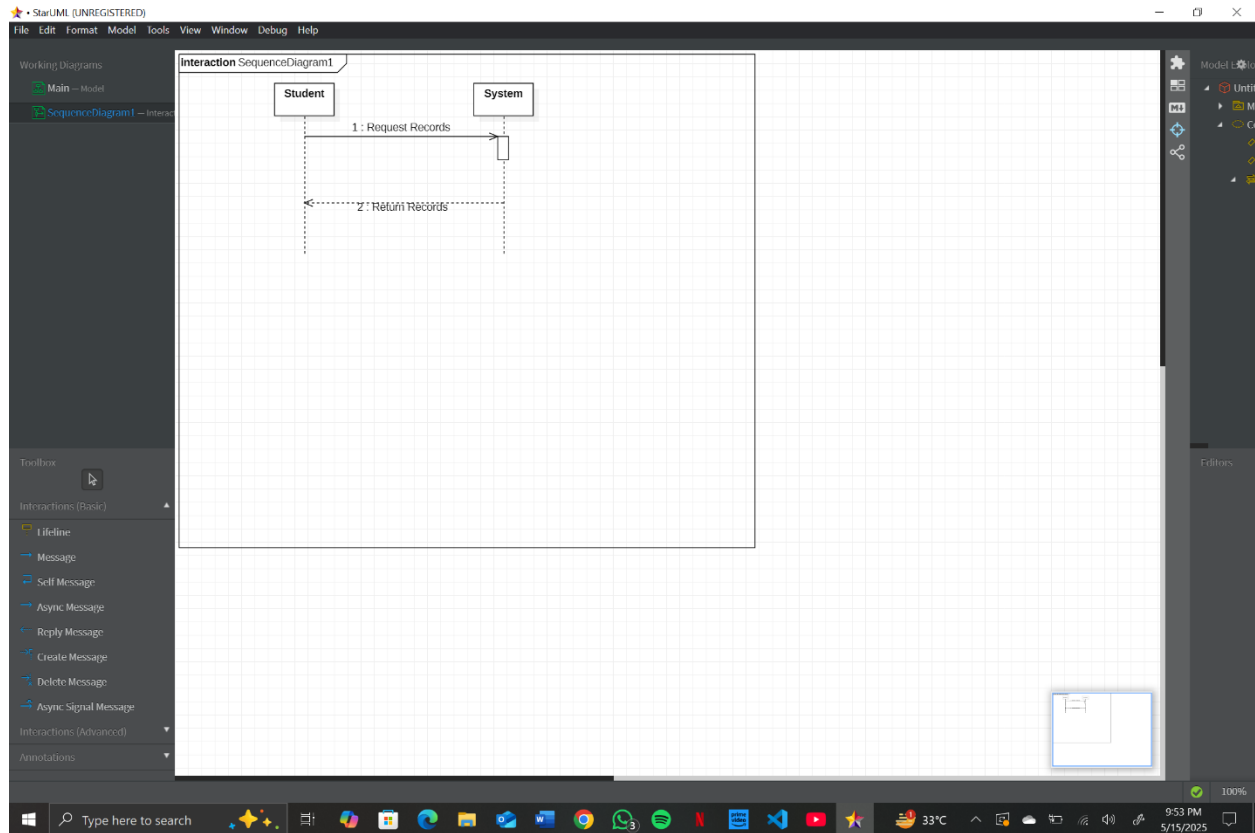
Use Case 3: Add Student



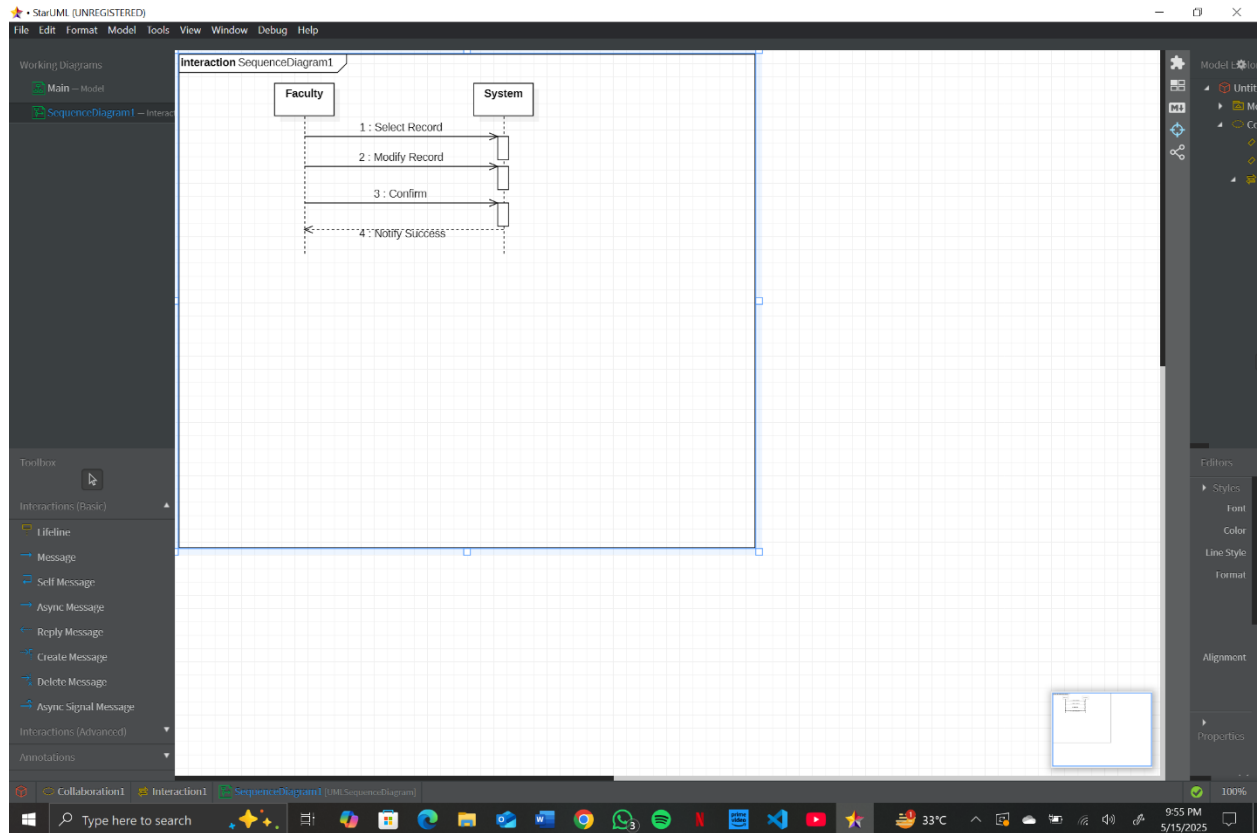
Use Case 4: Add Attendance



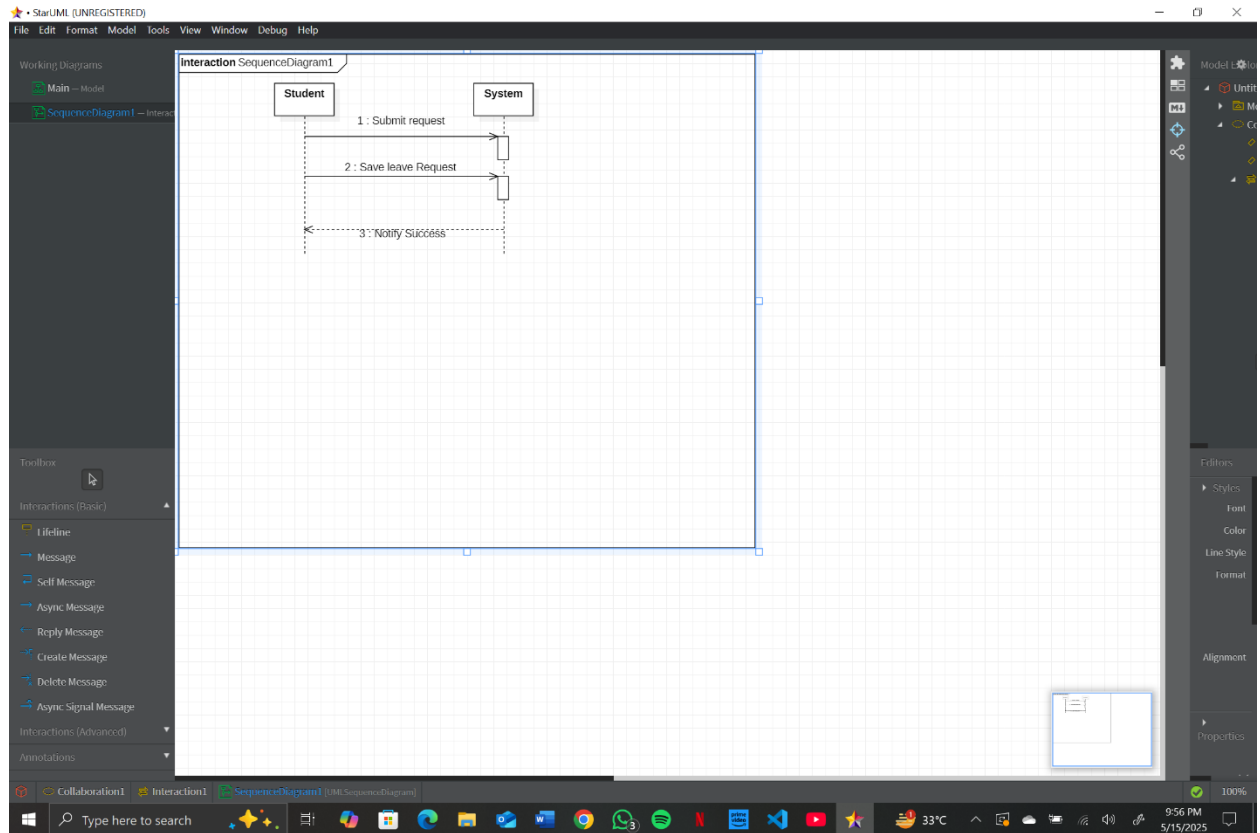
Use Case 5: View Attendance



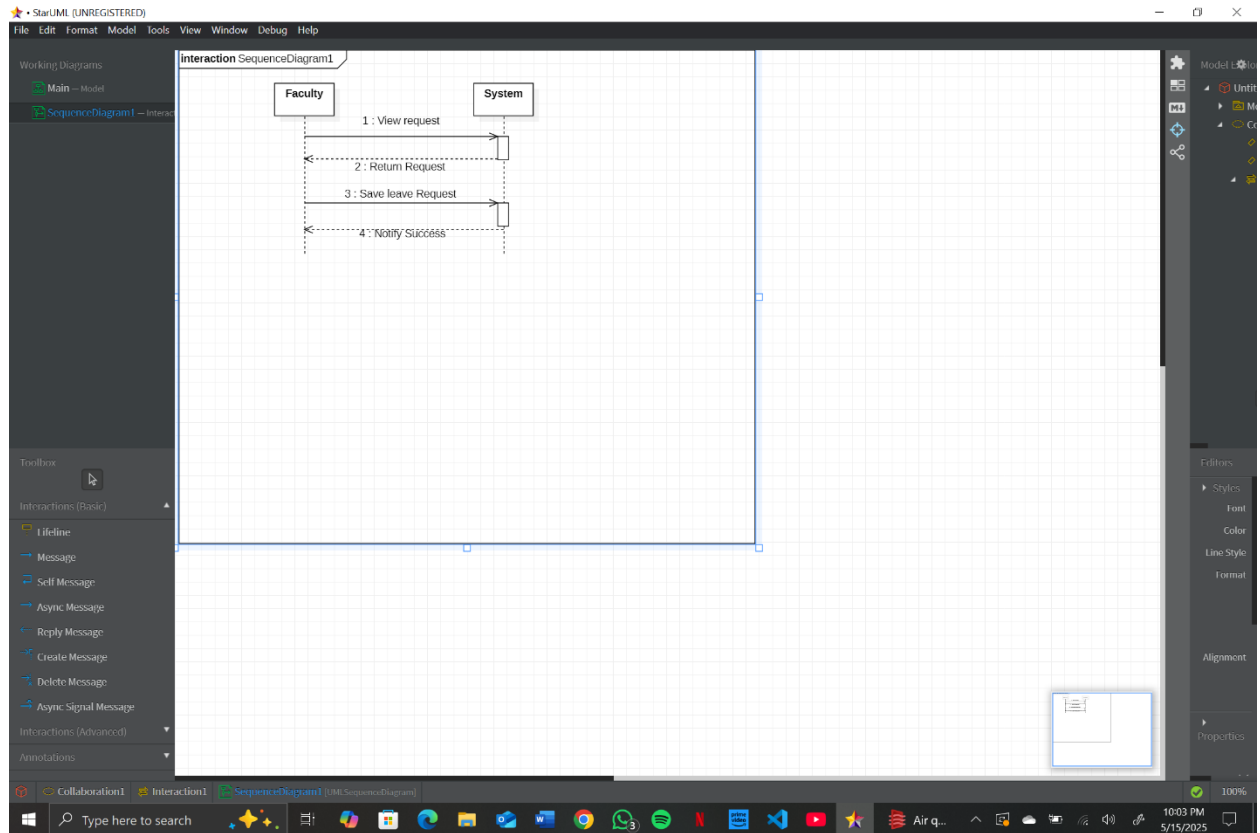
Use Case 6: Modify Attendance



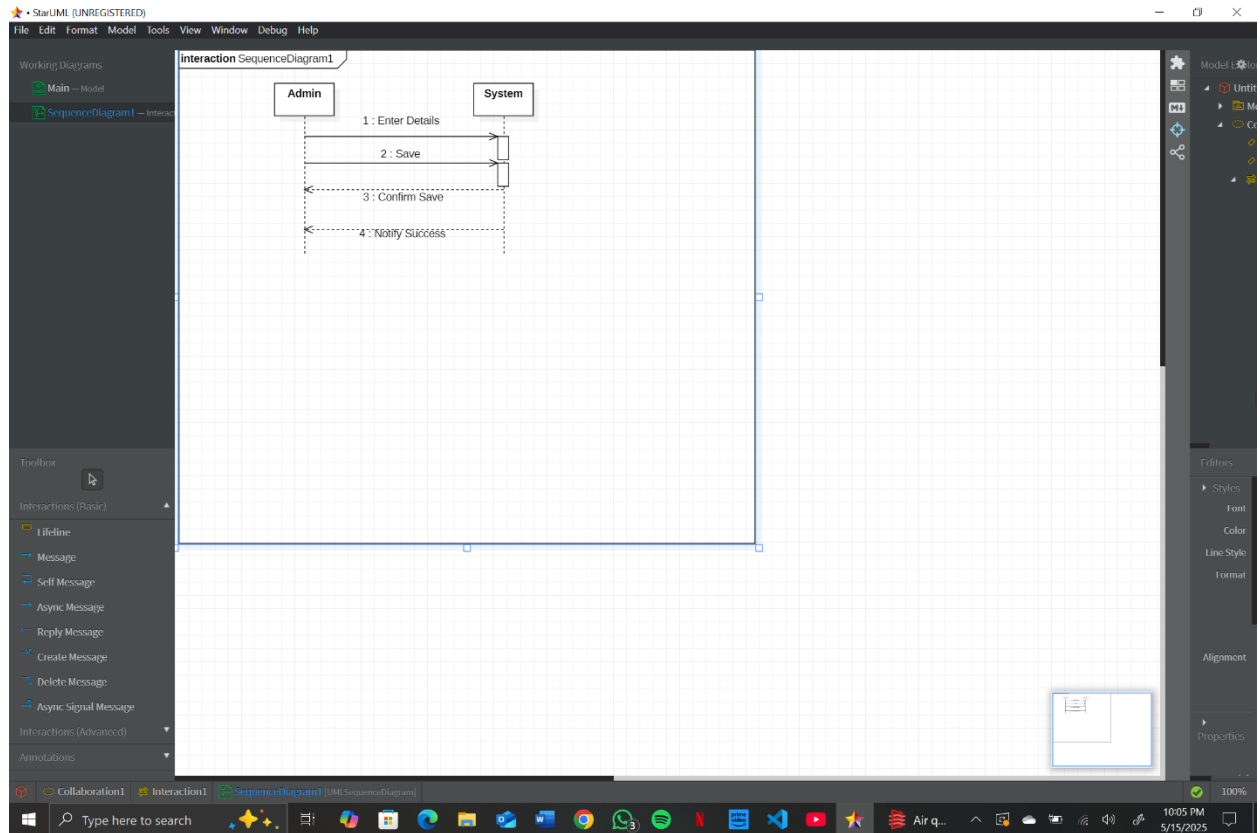
Use Case 7: Apply Leave



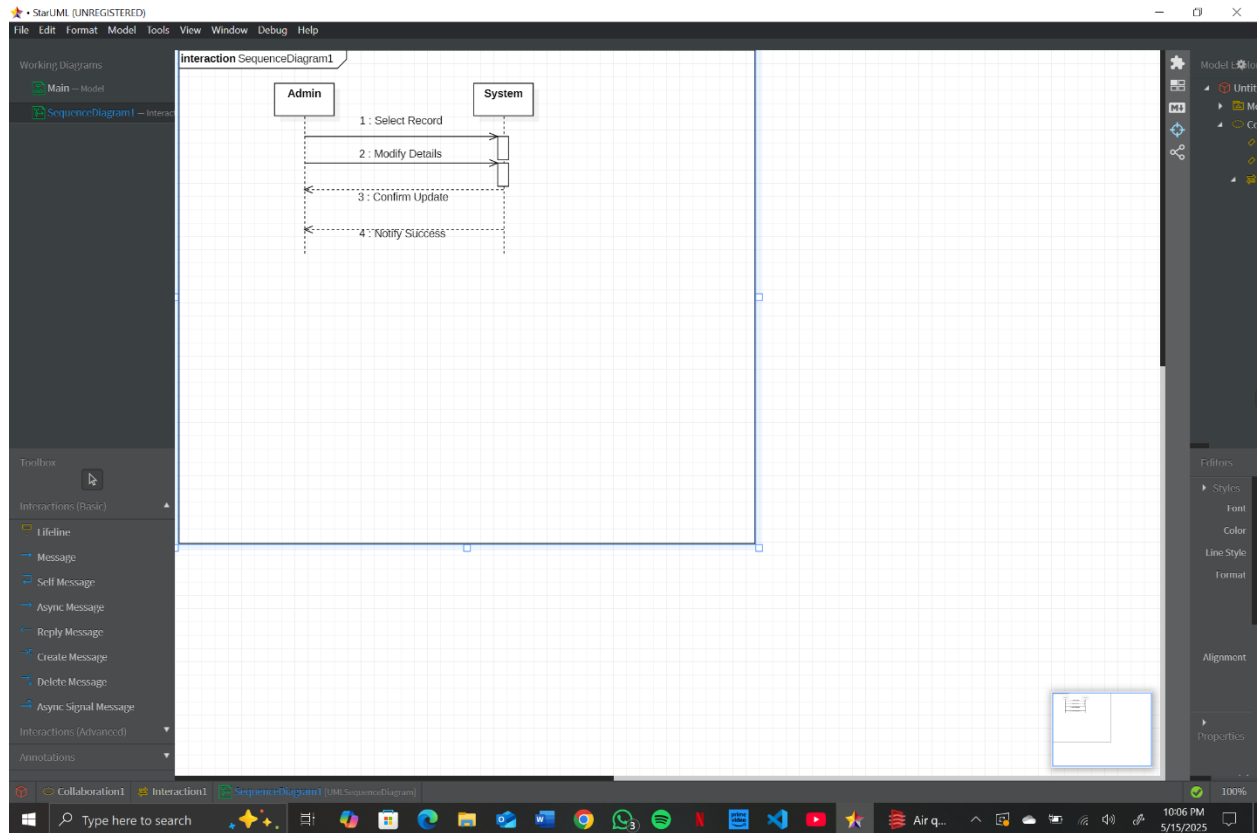
Use Case 8: Manage Leave



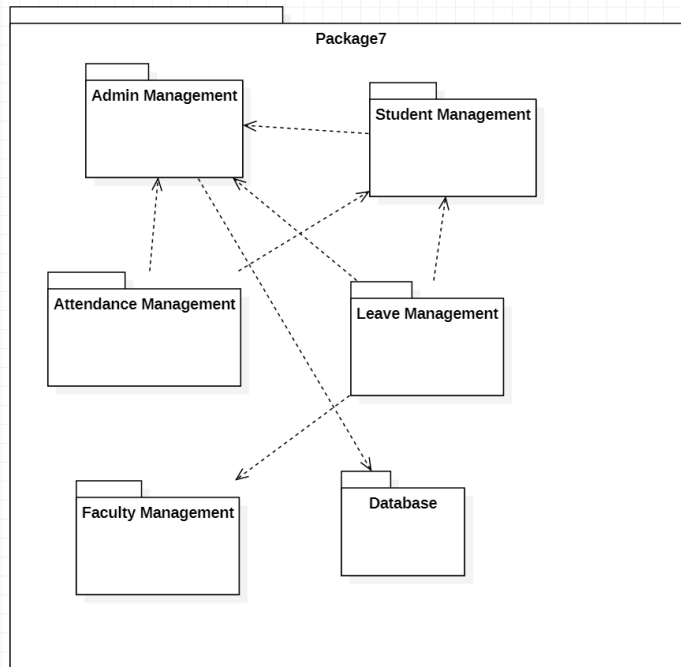
Use Case 9: Add Faculty



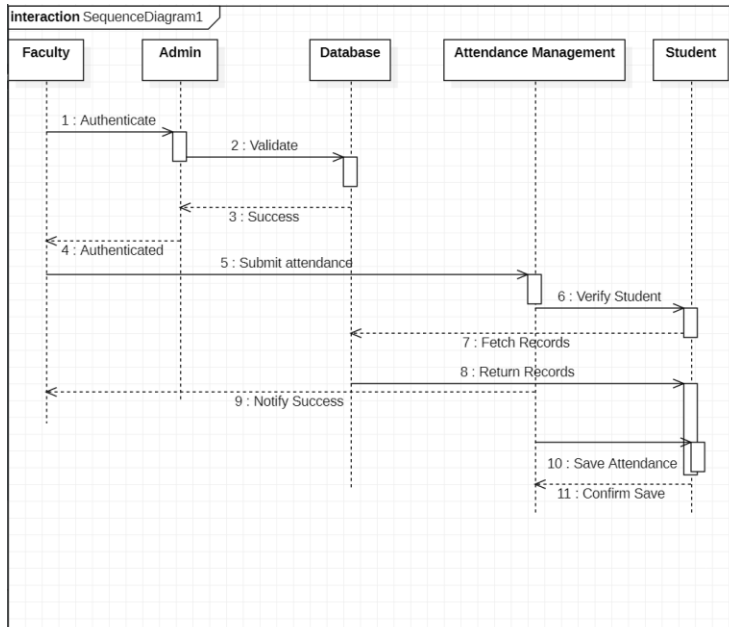
Use Case 10: Modify Faculty



Chapter : 4 (Package Diagram)



Chapter: 5 (Sequence Diagram)



Chapter: 6 (Class Diagram)

