

Module: SRS statements & Analysis Models

Course Code: CSE-434

Course Title: Software Engineering (Sessional)

Team Members

Estiak Ahamed Sazid

ID: 1804051

Karnis Fatema

ID: 1804052

Tahlil Abrar

ID: 1804056

Supervised By

Mir Md. Saki Kowsar

Assistant Professor

Department Of CSE, CUET

Moumita Sen Sarma

Lecturer

Department Of CSE, CUET

Department Of Computer Science and Engineering
Chittagong University of Engineering and Technology
Bangladesh

November, 2023

Contents

1	SRS for Registration and Login in Bus Scheduling System	3
1.1	Introduction	3
1.1.1	Purpose	3
1.1.2	Scope	3
1.2	Overall Description	3
1.2.1	Product Perspective	3
1.2.2	User Classes and Characteristics	3
1.2.3	Operating Environment	3
1.3	Functional Requirements	3
1.3.1	User Registration	3
1.3.2	User Authentication	3
1.3.3	Password Recovery	3
1.3.4	User Authorization and Roles	4
1.4	Non-functional Requirements	4
1.4.1	Performance	4
1.4.2	Security	4
1.4.3	Usability	4
1.5	Constraints	4
1.6	Assumptions and Dependencies	4
1.7	Future Enhancements	4
1.8	Conclusion	4
2	SRS for User Interaction of Bus Information in Bus Scheduling System	4
2.1	Introduction	4
2.1.1	Purpose	4
2.1.2	Scope	5
2.2	Overall Description	5
2.2.1	Product Perspective	5
2.2.2	User Classes and Characteristics	5
2.2.3	Operating Environment	5
2.3	Functional Requirements	5
2.3.1	Bus Schedule Information	5
2.3.2	Bus Route Information	5
2.3.3	Notice board	5
2.3.4	Comment Option	5
2.3.5	Real-time Bus Tracking (Future Enhancements)	5
2.4	Non-functional Requirements	5
2.4.1	Performance	5
2.4.2	Security	6
2.4.3	Usability	6
2.5	Constraints	6
2.6	Assumptions and Dependencies	6
2.7	Future Enhancements	6
2.8	Conclusion	6
3	SRS for Managing Transportation Information by Admin and Database in Bus Scheduling System	6
3.1	Introduction	6
3.1.1	Purpose	6
3.1.2	Scope	6
3.2	Overall Description	6
3.2.1	Product Perspective	6
3.2.2	User Classes and Characteristics	7
3.2.3	Operating Environment	7
3.3	Functional Requirements	7
3.3.1	Transportation Database Management	7
3.3.2	User Interaction Log	7

3.3.3	Bus Schedule Management	7
3.4	Non-functional Requirements	7
3.4.1	Performance	7
3.4.2	Security	7
3.4.3	Usability	7
3.5	Constraints	7
3.6	Assumptions and Dependencies	7
3.7	Future Enhancements	8
3.8	Conclusion	8
4	Use Case Diagram	8
4.1	Use Case Diagram for Registration and Login Module	8
4.2	Use Case Diagram for Interaction with Bus Information Module	9
4.3	Use Case Diagram for Managing Transportaiton Information Module	9
5	Sequence & Swim-lane diagrams	9
6	Activity & State diagrams	10
7	Class & Class Collaboration diagrams	13

1 SRS for Registration and Login in Bus Scheduling System

1.1 Introduction

1.1.1 Purpose

The purpose of this document is to outline the requirements for the Registration and Login module in the Bus Scheduling System at the University. This module is responsible for managing user registration, authentication, and authorization for teachers, students, office staff, and administrators.

1.1.2 Scope

The Registration and Login module will seamlessly integrate with the existing Bus Scheduling System, providing a secure and user-friendly interface for user registration, login, and access control. Users must use a CUET-provided email for registration.

1.2 Overall Description

1.2.1 Product Perspective

The Registration and Login module will closely collaborate with the Interaction of Bus Information, User Interaction with Bus Information, and Management of Transportation Information modules. It relies on a secure database to store and retrieve user-related data efficiently.

1.2.2 User Classes and Characteristics

- **Teachers:** Faculty members using the bus services.
- **Students:** University students using the bus services.
- **Office Staff:** Administrative staff using the bus services.
- **Administrators:** CUET transportation section for managing bus schedules, routes, and system settings.

1.2.3 Operating Environment

The system will be accessible through a web browser, ensuring compatibility with common web technologies and databases.

1.3 Functional Requirements

1.3.1 User Registration

- The system shall provide a registration form for users to create an account.
- Users must use a CUET-provided email for registration.
- The system shall validate and ensure the uniqueness of email addresses.
- An email will be sent to users for email verification upon successful registration.

1.3.2 User Authentication

- Users must be able to log in using their registered email and password.
- The system shall implement secure password storage techniques.
- Failed login attempts shall be tracked, and users shall be temporarily locked out after a predefined number of unsuccessful attempts.

1.3.3 Password Recovery

- Users shall have the option to reset their password through a password recovery process.
- The system shall send a password reset link to the user's registered email.

1.3.4 User Authorization and Roles

- The system shall support different user roles such as teachers, students, office staff, and administrators.
- Administrators shall have special privileges, including user account management and system configuration.

1.4 Non-functional Requirements

1.4.1 Performance

- The system should support a large number of simultaneous user registrations and logins.
- User authentication and authorization processes should be completed within 3 seconds.

1.4.2 Security

- Passwords must be securely stored using encryption techniques.
- All user interactions must be transmitted over HTTPS.

1.4.3 Usability

- The user interface for registration and login should be intuitive and user-friendly.
- Error messages should be clear and helpful.

1.5 Constraints

- The system must be developed using a technology stack that is compatible with both web browsers and mobile platforms.
- The system must comply with university data protection policies and relevant regulations.

1.6 Assumptions and Dependencies

- Users are assumed to have access to the internet and smartphones.
- The accuracy of user information is dependent on the proper functioning of the registration and authentication processes.

1.7 Future Enhancements

- Integration with CUET email authentication services for seamless registration.
- Implementation of two-factor authentication for enhanced security.

1.8 Conclusion

This Software Requirements Specification delineates the detailed requirements for the Registration and Login module in the Bus Scheduling System at the University. Successful implementation will result in a secure, user-friendly, and efficient user authentication system for the university community.

2 SRS for User Interaction of Bus Information in Bus Scheduling System

2.1 Introduction

2.1.1 Purpose

The purpose of this document is to outline the requirements for the Interaction of Bus Information module in the Bus Scheduling System at the University. This module is responsible for managing and disseminating information related to bus schedules, routes, and real-time updates for university students, faculty, and staff.

2.1.2 Scope

The Interaction of Bus Information module will seamlessly integrate with the existing Bus Scheduling System, providing a user-friendly interface for accessing real-time bus information, routes, and schedules. It will cater to the needs of the user to see the bus information and interact with the information by giving compliments in the comment section.

2.2 Overall Description

2.2.1 Product Perspective

The Interaction of Bus Information module will closely collaborate with the Bus Scheduling and User Interaction with Bus Information modules. It relies on a robust database to store and retrieve bus-related data efficiently.

2.2.2 User Classes and Characteristics

- **Passengers:** University students, faculty, and staff using the bus services.
- **Administrators:** Admin with special privileges for managing bus schedules, routes, and system settings.

2.2.3 Operating Environment

The system will be accessible through web browser, ensuring compatibility with common web technologies and databases.

2.3 Functional Requirements

2.3.1 Bus Schedule Information

- The system shall provide a detailed and up-to-date schedule of bus routes, including departure and arrival times at various stops.
- Users shall be able to view the bus schedule for a specific route and time.

2.3.2 Bus Route Information

- The system shall display comprehensive information about each bus route, including stops, landmarks, and estimated travel times.
- The user can filter bus schedules according to teachers, students and office staff.

2.3.3 Notice board

- The system will display notifications to users about changes in bus schedules, delays, or cancellations.

2.3.4 Comment Option

- The user can comment and give their opinions about the bus schedule.

2.3.5 Real-time Bus Tracking (Future Enhancements)

- The system shall integrate with GPS technology to provide real-time tracking of buses on the designated routes.
- Passengers shall be able to view the current location of the bus.

2.4 Non-functional Requirements

2.4.1 Performance

- The system should handle concurrent user requests for bus information without performance degradation.

2.4.2 Security

- User authentication and authorization mechanisms should be in place to ensure secure access to bus information.
- Data transmission between the server and mobile application should be encrypted using HTTPS.

2.4.3 Usability

- The user interface for accessing bus information should be intuitive and accessible.
- The notice board should be clear and provide relevant information to users.

2.5 Constraints

- The system must be developed using a technology stack that is compatible with both web browsers and mobile platforms.
- The system must comply with university data protection policies and relevant regulations.

2.6 Assumptions and Dependencies

- Users are assumed to have access to the internet and smartphones.
- The accuracy of bus information is dependent on the proper functioning of administrators.

2.7 Future Enhancements

- Integration with GPS technology to provide real-time tracking of buses on the designated routes.
- Integration with Smart Bus Cards for contactless boarding.
- Integration with campus events and academic schedules for dynamic adjustments in bus schedules.

2.8 Conclusion

This Software Requirements Specification delineates the detailed requirements for the Interaction of Bus Information module in the Bus Scheduling System at the University. Successful implementation will result in an efficient, reliable, and user-friendly bus information system for the university community.

3 SRS for Managing Transportation Information by Admin and Database in Bus Scheduling System

3.1 Introduction

3.1.1 Purpose

The purpose of this document is to outline the requirements for the Management of Transportation Information by Admin and Database module in the Bus Scheduling System at the university. This module is responsible for overseeing and managing transportation-related data, schedules, routes, and interactions with the database.

3.1.2 Scope

The Management of Transportation Information module will closely collaborate with the existing Bus Scheduling System, providing administrators with tools to manage bus schedules, routes, user interactions, and ensuring the integrity of the transportation database.

3.2 Overall Description

3.2.1 Product Perspective

The Management of Transportation Information module will closely collaborate with the Interaction of Bus Information and User Interaction with Bus Information modules. It relies on a robust database to store and retrieve transportation-related data efficiently.

3.2.2 User Classes and Characteristics

- **Administrators:** CUET transportation section has the ability to manage bus schedules, routes, and system settings.
- **Database Administrators:** Responsible for overseeing and managing the transportation database.

3.2.3 Operating Environment

The system will be accessible through a web browser, ensuring compatibility with common web technologies and databases.

3.3 Functional Requirements

3.3.1 Transportation Database Management

- The system shall store and manage comprehensive data related to bus schedules, routes, and user interactions.
- Administrators shall have the ability to add, edit, or delete bus schedules and routes.

3.3.2 User Interaction Log

- The system shall maintain a log of user interactions with transportation information.
- Administrators shall have the ability to view and analyze user comments and opinions about bus schedules.

3.3.3 Bus Schedule Management

- Administrators shall have the ability to modify bus schedules in response to campus events or academic schedules.
- The system shall provide tools for administrators to efficiently manage and update bus schedules.

3.4 Non-functional Requirements

3.4.1 Performance

- The system should handle concurrent user requests for transportation information without performance degradation.
- Database queries and updates should be completed within an acceptable response time.

3.4.2 Security

- User authentication and authorization mechanisms should be in place to ensure secure access to transportation information.
- Database access and updates should be restricted to authorized personnel.

3.4.3 Usability

- The user interface for administrators should be intuitive and accessible.
- Tools for managing bus schedules and routes should be user-friendly and efficient.

3.5 Constraints

- The system must be developed using a technology stack that is compatible with both web browsers and mobile platforms.
- The system must comply with university data protection policies and relevant regulations.

3.6 Assumptions and Dependencies

- Administrators are assumed to have access to the internet and appropriate permissions for database management.
- The accuracy of transportation information is dependent on the proper functioning of the transportation database.

3.7 Future Enhancements

- Integration with real-time tracking technology for dynamic adjustments in bus schedules.

3.8 Conclusion

This Software Requirements Specification delineates the detailed requirements for the Management of Transportation Information by Admin and Database module in the Bus Scheduling System at the University. Successful implementation will result in an efficient, reliable, and user-friendly transportation management system for the university community.

4 Use Case Diagram

The Use Case Diagram for the CUET Bus System illustrates the interactions between different stakeholders and the system itself with three key use cases.

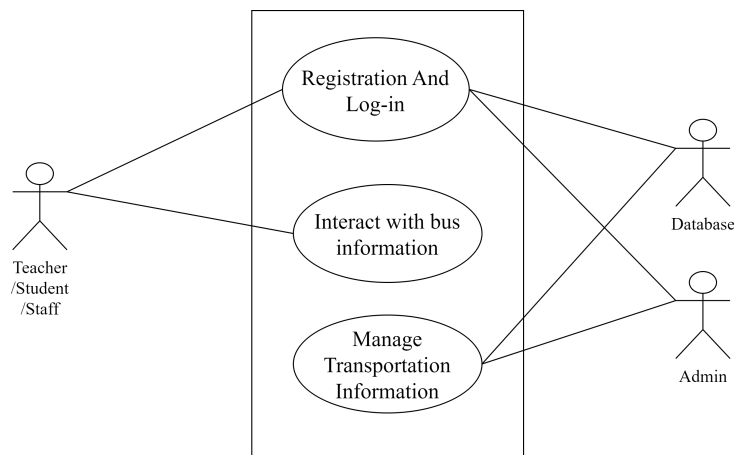


Figure 1: Use Case Diagram for the overall system

4.1 Use Case Diagram for Registration and Login Module

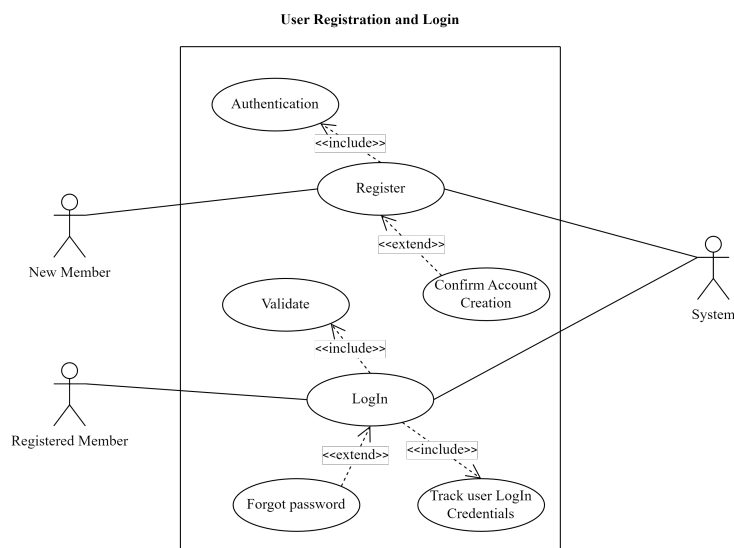


Figure 2: Use Case Diagram for Registration and Log-in system

4.2 Use Case Diagram for Interaction with Bus Information Module

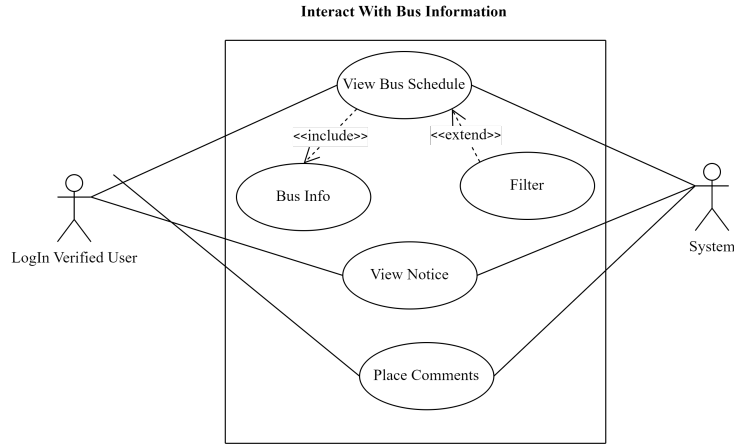


Figure 3: Use Case Diagram for Interaction with bus information

4.3 Use Case Diagram for Managing Transportaiton Information Module

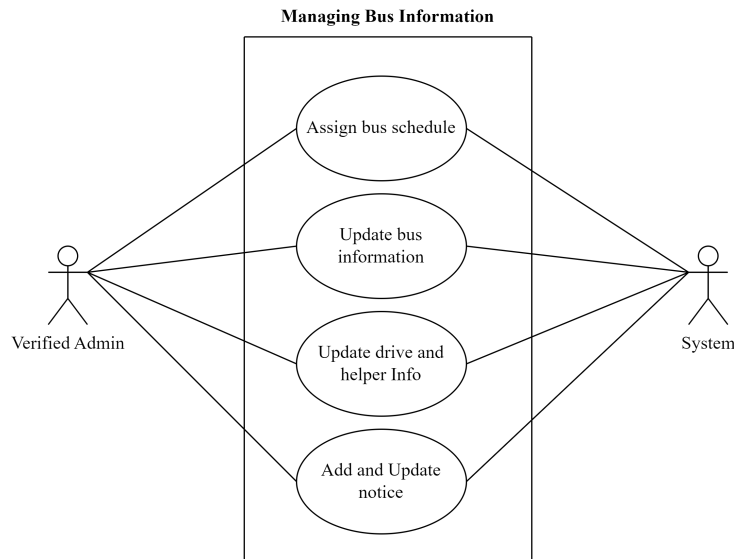


Figure 4: Use Case Diagram for Managing Bus Information

5 Sequence & Swim-lane diagrams

The provided sequence diagram visually encapsulates the dynamic interactions within a bus scheduling system, highlighting key entities such as User, Admin, and Database. Lifelines, representing these entities, delineate the sequence of messages exchanged during various actions. The depicted messages, including register, login, viewScheduleDetails, filterSchedule, comments, viewNotice, manageSchedule, and updateNotice, convey the flow of control and data between the involved entities. This diagram serves as a comprehensive representation of the chronological order of events and communication pathways in the specified use case, offering insights into the dynamic behavior of the bus scheduling system.

The swimlane diagram, a visual organization of system processes into lanes, is a valuable tool for illustrating interactions between "User Authentication," "Bus Schedule Management," and "User Interaction" modules in our system. Each module is assigned a dedicated lane, offering a clear visual depiction

of user movements and system processes. This diagram enhances comprehension of the Bus Scheduling System's dynamic flow and interactions, effectively capturing the roles played by Users, Admin, and the Database.

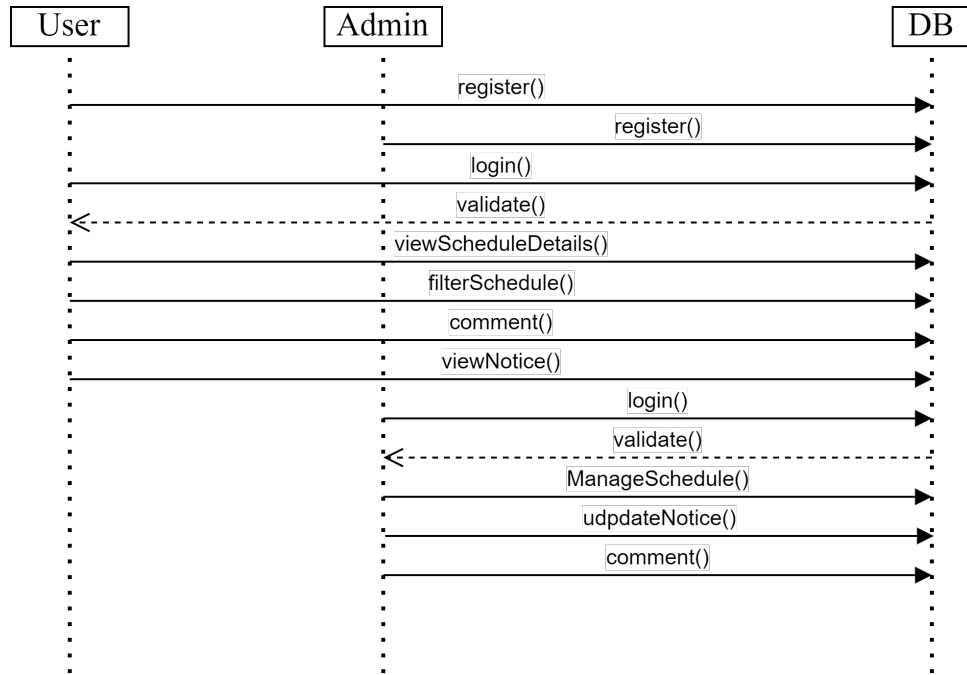


Figure 5: Sequence Diagram

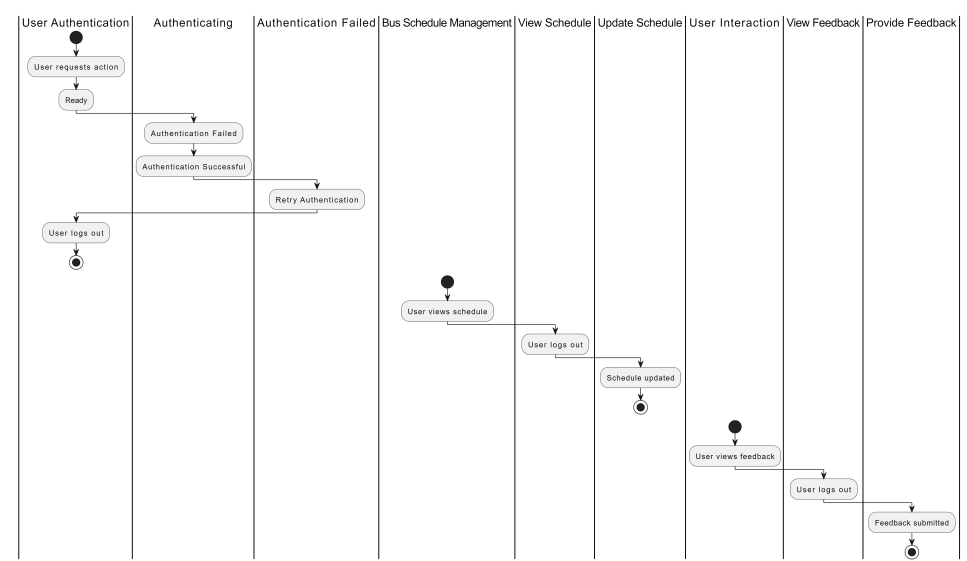


Figure 6: Swim-lane Diagram

6 Activity & State diagrams

Activity diagrams in software engineering are graphical representations that illustrate the dynamic aspects of a system, showcasing the flow of activities and actions. These diagrams utilize various symbols, such as activities, actions, decision nodes, merge nodes, initial and final nodes, as well as fork and join nodes to depict the sequence and relationships between different processes.

The state diagram represents the state transitions and interactions within a bus scheduling system for different user roles. This state diagram provides a visual representation of the flow of user interactions in

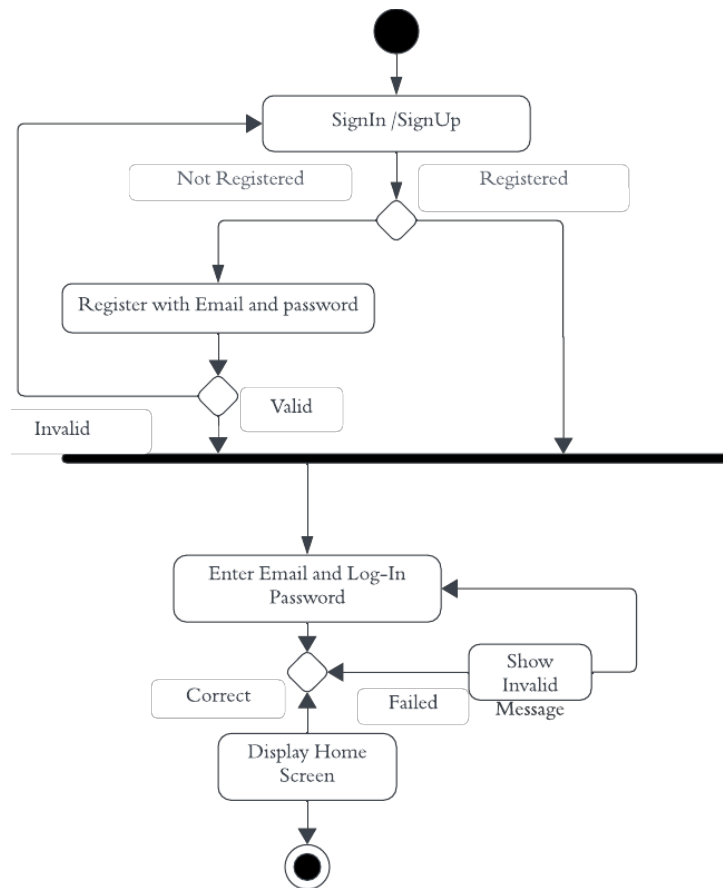


Figure 7: Activity Diagram of Registration and Login

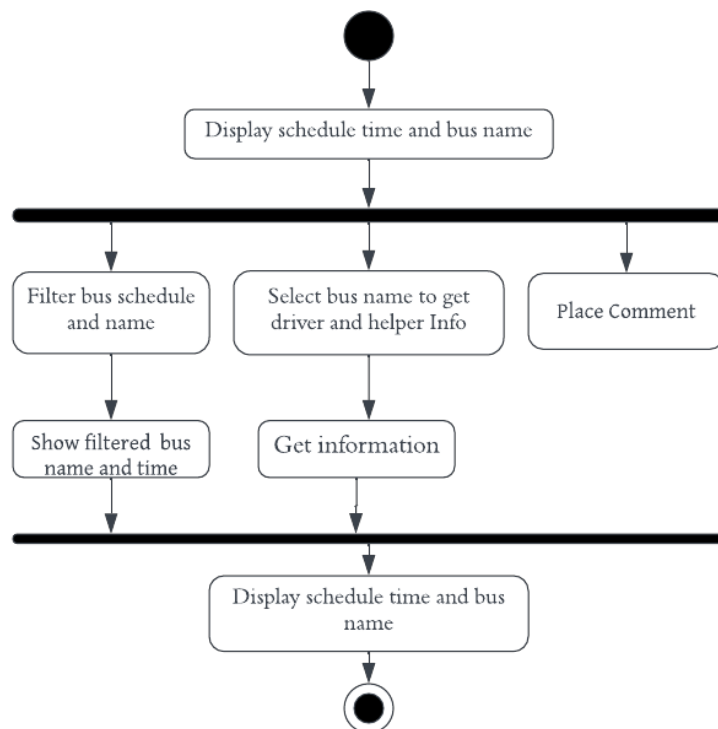


Figure 8: Activity Diagram of Bus Schedule

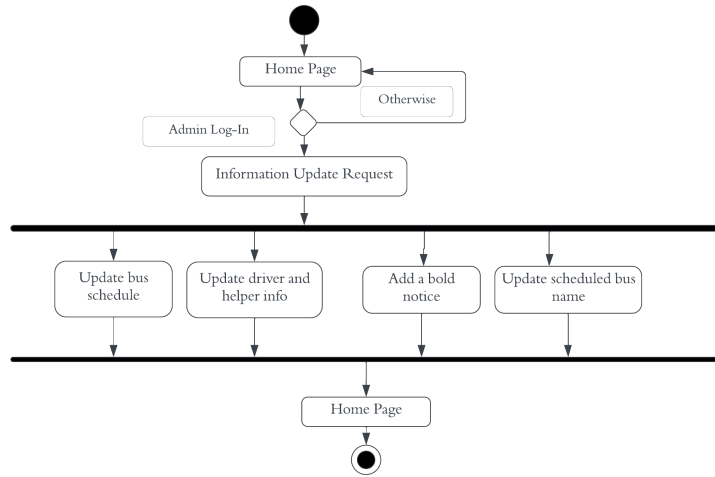


Figure 9: Activity Diagram of Manage Transportation Information

the bus scheduling system, capturing login, role-specific dashboard activities, and logout functionalities.

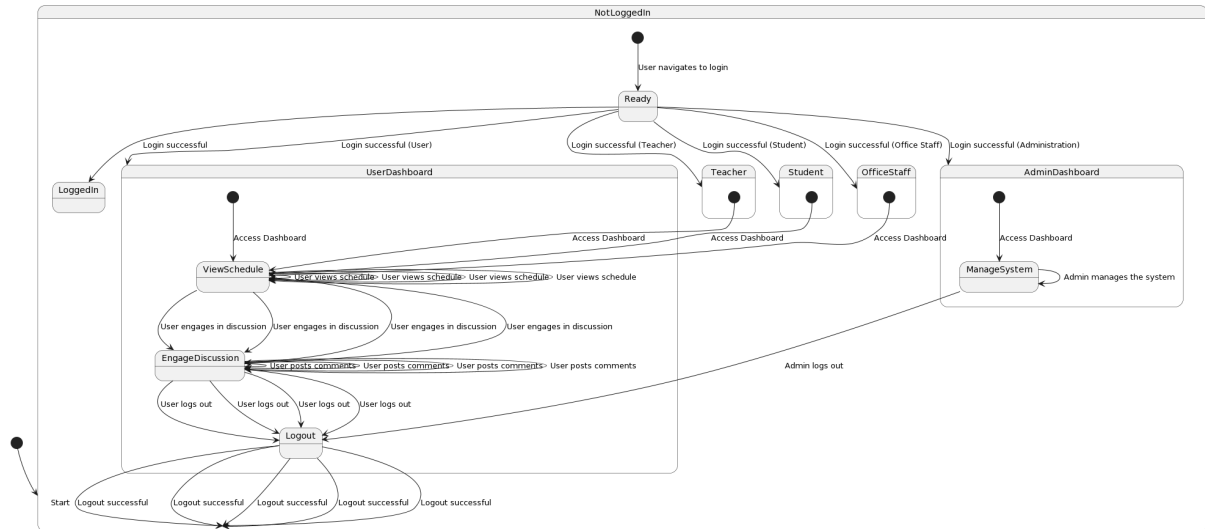


Figure 10: State Diagram of the system

7 Class & Class Collaboration diagrams

This class diagram (Figure 02) outlines the key entities and their relationships in the bus scheduling system, providing a foundation for building the corresponding software architecture. In a bus scheduling system, a class diagram is designed to illustrate the organization and relationships among various entities. The primary classes include 'User', which serves as the parent class, and its subclasses: 'Student', 'Teacher', and 'Office Staff'.

Additionally, the system includes an 'Administration' class representing administrative users responsible for managing the system. Another essential class is 'ScheduleDetail', capturing information about bus schedules, including 'busName', 'driver name', 'date', and 'time'.

The system facilitates communication and information dissemination through the 'Notices' class, which contains 'noticeId', 'noticeBody'. Furthermore, users can engage in discussions through the 'Comments' class, which has 'commentId', 'text', and references to the 'User' posting the comment.

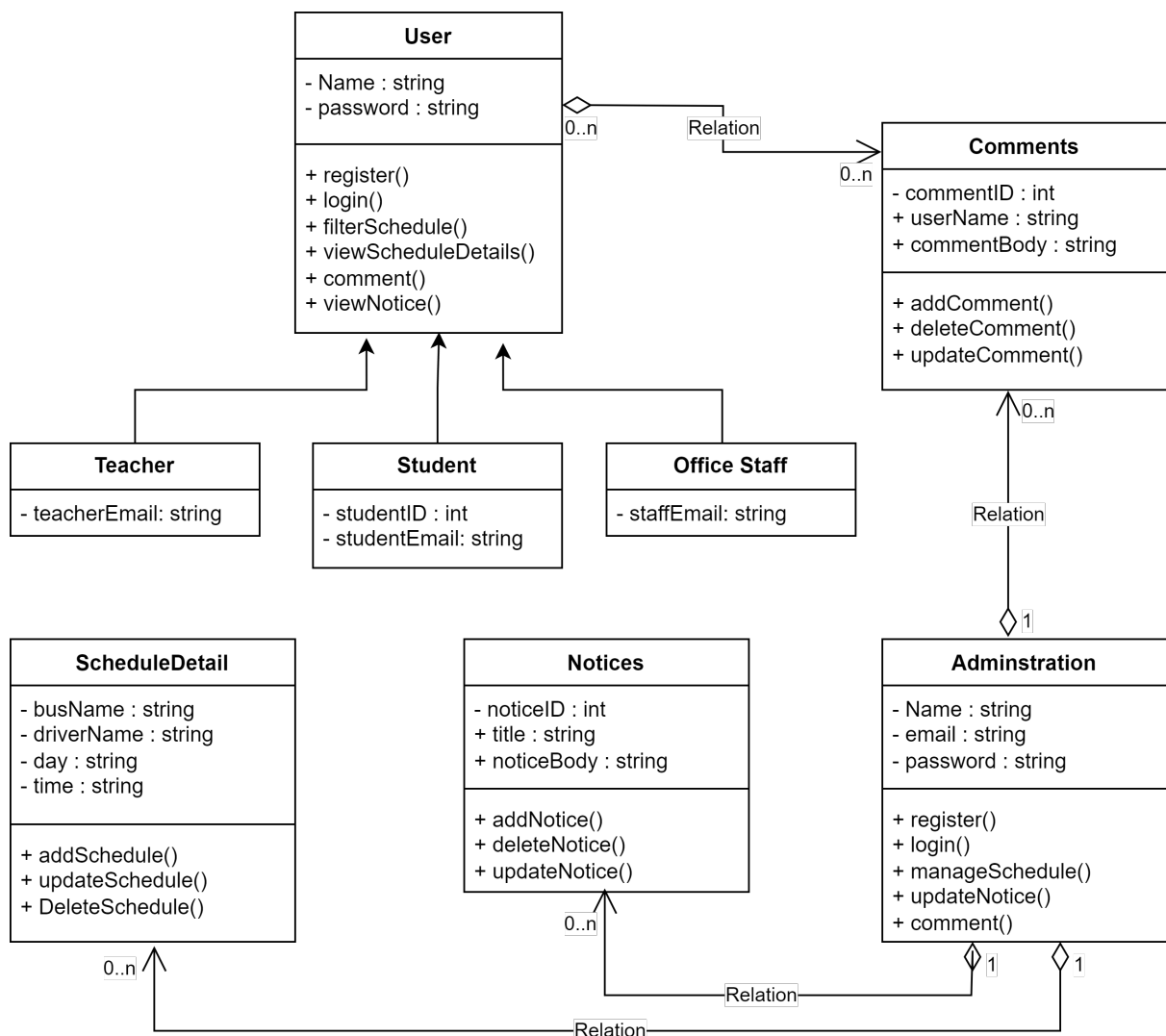


Figure 11: Class Diagram