Islamia University Bahawalpur

Submitted to:

Software Engineering
Mam Urooj Fatima
3rd M-(B)

Submitted By:

Muhammad Fahad (1107)

Hassaan Ahmed (1092)

Ayesha Shoaib (1108)

Aqsa Sarwar (1069)

Zaibunisa (1021)

University Bus Management System

Why do we need a University Bus Management System?

It's a busy Monday morning at Bahawalpur Students, Staff, and Teachers are waiting impatiently for the bus, suddenly a bus arrives and everyone rushes toward the bus like a bee, a boy standing aside and praying for everyone not to get hurt. Then a boy enters the screen running to catch the bus it seems like his attendance is not 75% in a particular subject. Few people entered the bus and got the seats, a few kept on standing in the bus and a Few missed the bus and continued watching on their wrist watch for the arrival of the next bus.

1: Introduction:

Considering the flaws in point management of Islamia University Bahawalpur I came up with the idea of managing the buses online through web/ app.

The University Bus Management software will have the following features:

- Arrival Time of Buses
- Departure Time of Buses
- Live Tracking of Buses
- Timetable of the Buses
- Expected time of arrival
- Countdown for the arrival of the Bus
- Number of Buses to pick up the students

In this project a student will log in through his University's ID and just click the time of bus he wants to take from the schedule displayed on the screen, the university will track the number of students who want to take the bus at a specified time and will send the required number of buses to accommodate the students easily.

After a student has opted for the time of the bus he wants to take, he will be shown the countdown for the arrival of the bus and the administration will also send a reminder message just for an alarm 15 minutes before the arrival of the bus.

A student can track his live location through the app which will help the day scholars who will get the bus in between the SP and FP.

This system will not only save the cost of buses as we see that few buses are empty. The university can reduce its cost of transportation by stopping unnecessary buses and along with this it will facilitate the students as well when these unnecessary buses will the adjusted by keeping buses on route during office timings.

1.1 Purpose:

The purpose of the university Bus management system is to facilitate the students by providing them the relevant information which includes the arrival time, departure time, number of coming buses, and the live location of the bus. This will not only facilitate the students but also reduce the fuel consumption which will contribute to cutting the transportation budget as there will be no unnecessary buses on the route.

1.2 Scope:

The best part of the University Bus Management System is that it will be user-friendly and very secure as everyone will log in through their University ID. It can be used by public transportation i.e.

Metro, Trams, etc., as well as some of its features, can also be used by local transportation companies as well to enhance and keep track of their customers.

2: Overall Description:

The Bus Management App leverages location-based services to provide real-time bus tracking and route information to users. Users can install the app, create profiles, and access bus-related services. Admins can use the app to monitor and manage the bus system efficiently.

Use Case Diagram:

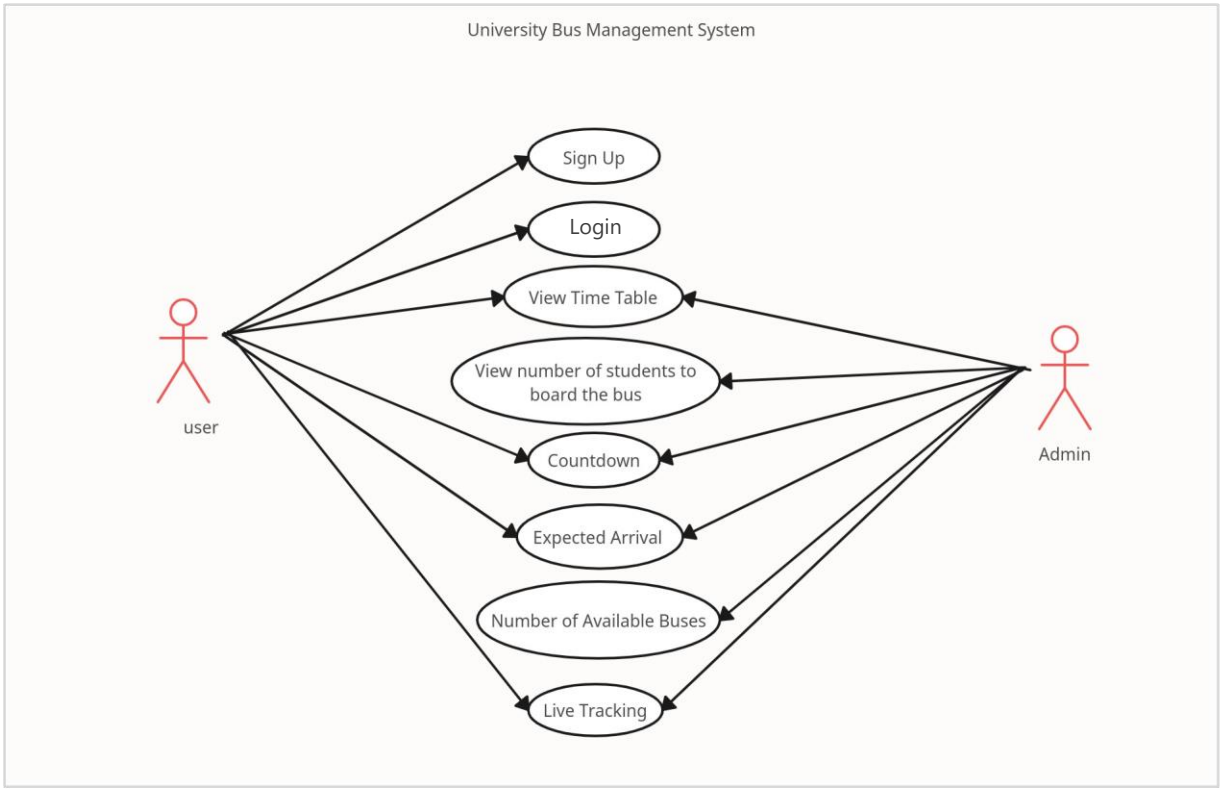

fig:1.1 Use Case Diagram

Use Case Description:

These use case descriptions outline the interactions between users and the "University Bus Management System" app, providing a detailed understanding of the functionality and user interactions. Customize these descriptions as needed based on specific system requirements or features.

Name	Address Code	Description	Actor	Pre-Condition	Post-Condition
Sign Up	SU-01	User registration process to access the system.	Users	User has internet access and a valid university email ID.	User account created and enabled for system access.
Login	LG-01	Authentication process to access the system.	Users	User has a registered account and valid login credentials.	Successful access to the system with user-specific data.
View Time Table	VT-01	Accessing the bus schedule for different routes and times.	Users, Admin	System operational with updated bus schedules.	Display of comprehensive bus timetables for users.
View Students Boarding	VS-01	Observing the number of students boarding a particular bus.	Admin	Real-time data availability from the bus management system.	Real-time count of students boarding a specific bus.
Countdown	CD-01	Displaying a real-time countdown for bus arrivals.	Users, Admin	User has selected a specific bus for their journey.	Real-time countdown displayed for selected bus arrival.
Expected Arrival	EA-01	Providing estimated arrival times for buses.	Users, Admin	Real-time data availability and selected bus information.	Accurate estimation of bus arrival time at specific stops.
Number of Available Buses	NB-01	Determining the count of available buses for service.	Admins	Updated system data on bus availability and allocation.	Number of buses currently available for student pickup.
Live Tracking	LT-01	Tracking the real-time location of university buses.	Users, Admin	System connected to bus fleet tracking devices.	Real-time display of bus locations on a map.

Class Diagram:

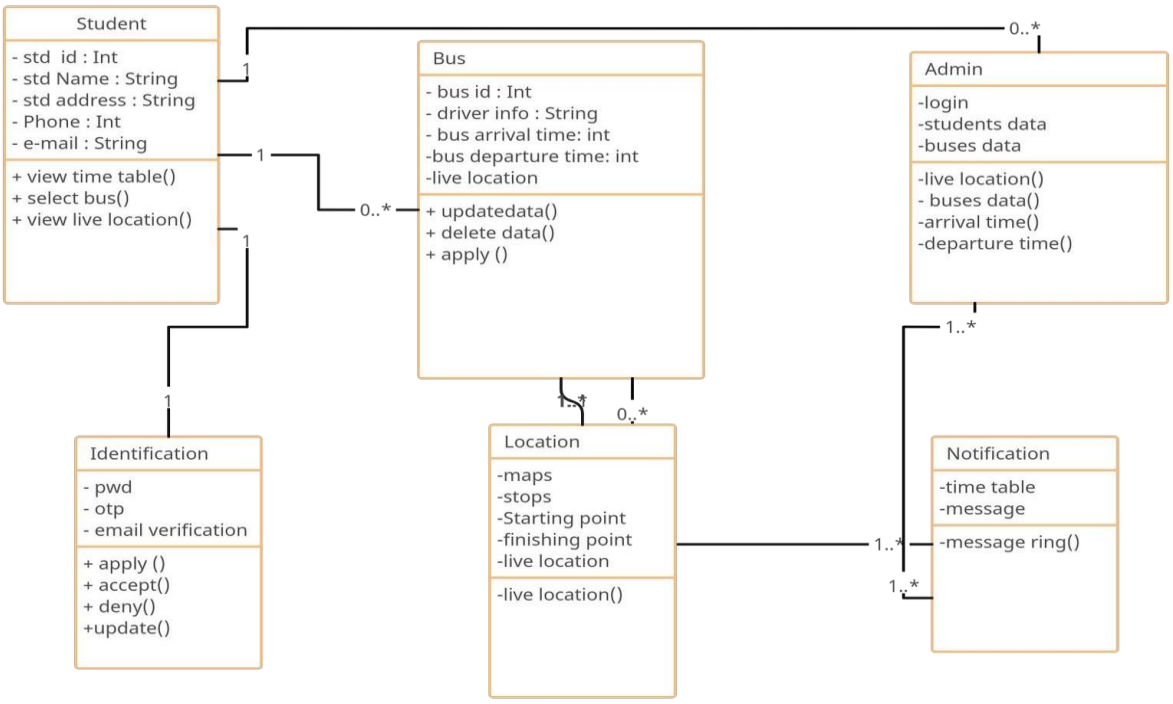

fig: 1.2 Class diagram

3: Functional Requirements:

- Admin
- Login
- Arrival time of the bus
- Departure time of the bus
- Live tracking of the bus
- Timetable of the bus
- Expected time of arrival
- Countdown of the arrival
- Number of buses to pick up the students

3.1 Admin:

- Admin has the access to monitor everything on the app.
- Admin can edit the timetable, monitor the live location, and allow the bus to leave.

3.2 Login:

- In this step the student needs to login to the app through the email provided by the university
- He/she can change the password through the app by using their email, previous password, and name or they can use their email or phone number for the OTP,

3.3 Arrival Time of the Bus:

- The system must provide a list of all bus arrivals, including their expected arrival times.
- Users (students, staff, teachers) should be able to view the arrival times of all buses.: Departure Time of the Bus:
- The system must display the departure times of all buses.
- Users should have access to the bus departure schedule.

3. .5 Live Tracking of the Bus:

- The system must allow users to track the real-time location of university buses on a map.
- Users should be able to view the current position of buses, enabling them to plan their departure accordingly.

3.6 Time Table of the Bus:

- The system should provide a comprehensive bus timetable, detailing the departure and arrival times for all buses on different routes.
- Users should be able to access and search the complete bus timetable.

3.7 Expected Time of the Arrival:

- The system must calculate and display the expected time of arrival for each bus at specific stops or destinations.
- Users should receive accurate estimates of when a bus will reach their location.

3.8 Countdown for the Arrival:

- The system should initiate a countdown timer for users once they select a specific bus for their journey.
- Users should see a real-time countdown to the bus's arrival to help them plan their departure.

3.9 Number of Buses:

- The system must determine the number of buses required to accommodate the number of students who have selected a particular bus.
- The system should send this information to the university's transportation department for bus allocation.

4. Non-Functional Requirements:

4.1 Performance:

- These requirements define how well the system should perform under various conditions. Examples include response times, throughput, and scalability. For instance, a non-functional requirement might specify that the system should respond to user requests within two seconds.

4.2 Reliability:

- These requirements address the system's ability to consistently perform its functions without failures or errors. They may include measures for uptime, availability, and fault tolerance.
- For example, a non-functional requirement might state that the system should have 99.99% uptime.

4.3 Security:

- Security requirements focus on protecting the system and its data from unauthorized access, breaches, and vulnerabilities. They can include authentication, authorization, data encryption, and compliance with security standards.

4.4 Usability:

- These requirements relate to the user-friendliness and user experience of the system. They might specify aspects like user interface design, accessibility, and ease of navigation. For example, a non-functional requirement could state that the system should be accessible to users with disabilities.

4.5 Scalability:

- Scalability requirements describe how the system should handle increased loads or user activity. They may specify the system's ability to accommodate a growing user base or handle more data. For instance, a non-functional requirement might state that the system should scale horizontally to support a doubling of users.

4.6 Maintainability:

- These requirements focus on the ease of maintaining and updating the system over time. They might include code readability, modularity, and documentation standards.

4.7 Portability:

- Portability requirements specify how easily the system can be moved or adapted to different environments or platforms. This includes compatibility with various hardware and operating systems.

4.8 Capacity:

- Capacity requirements define limits or constraints on the system, such as maximum file sizes, concurrent users, or database storage.

4.9 Compliance:

- Compliance requirements ensure that the software adheres to legal and industry-specific regulations and standards. This may include data protection laws, industry-specific certifications, and security standards like ISO 27001.

4.10 Interoperability:

- Interoperability requirements address the system's ability to work seamlessly with other systems or components, such as compatibility with specific browsers, operating systems, or APIs.

Tools Used:

Creately

Draw.io

Adobe Acrobat