System Analysis & Design Laboratory (Section :- C, Group:- 7)

Software Requirements Specification

MoveMe - Smart Transport Management System for Bangladesh

Motto :- Owner or passenger, your bus problem is over.

Table Of Contents

1. **Abstract ——----------3**
2. **Acknowledgement——----------3**
3. **Introduction——----------4**
   - 3.1. Problem Statement
   - 3.2. Goal
   - 3.3. SDLC
4. **Information Gathering——----------7**
   - 4.1. Information Sources
   - 4.2. Information Gathering Tools
   - 4.3. Benchmark Products
   - 4.4. Survey
5. **System Analysis——----------11**
   - 5.1. Survey Analysis
   - 5.2. Benchmark Gap Analysis
   - 5.3. Feature List Fixation
   - 5.4. Functional & Non-Functional Requirements
   - 5.5. Feasibility Analysis
   - 5.6. SWOT Analysis
6. **System Design——----------17**
   - 6.1. Context Diagram
   - 6.2. Data-Flow Diagram
   - 6.3. Use-Case Diagram
   - 6.4. Use-Case Descriptive Forms
   - 6.5. Activity Diagram
   - 6.6. Swimlane Diagram
   - 6.7. Class Diagram
   - 6.8. CRC Diagram
   - 6.9. Sequence Diagram
   - 6.10. State Diagram
   - 6.11. Entity Relationship Diagram
   - 6.12. Deployment Diagram
7. **UI Design——----------43**
   - 7.1. Different Design Rules
   - 7.2. Our Project UI Design
   - 7.3. Testing
8. **Conclusion——----------54**
9. **Reference——----------55**

1.Abstract

This report outlines the Software Requirements Specification (SRS) for the 'MoveMe' project, aimed at delivering a comprehensive solution for smart transport management. The system is designed to streamline efficient travel management, RFID system, fare collection, and disciplined bus operations while ensuring robust operational oversight. By developing an integrated platform, the project seeks to simplify the transport process and elevate the overall experience for passengers, operators, and administrators alike.

2.Acknowledgement

The 'MoveMe' project has been meticulously developed by a dedicated team known as The Coders. The team members include **Robiul Awoul Robin (ID: 011221564), Md. Sadman Haque (ID: 011221592), Shraboni Akter Mim (ID: 011221126)**. Each member has contributed their expertise and effort to bring this innovative solution to life. This project is proudly presented as a testament to our commitment to creating impactful and efficient transport management systems.

3.Introduction

3.1.Problem Statement

The incidence of road accidents is rising alarmingly, primarily due to poor vehicle parking practices and unauthorized halts at inappropriate locations. This issue is especially prevalent among buses in Bangladesh, where frequent and unnecessary stops contribute to traffic congestion, accidents, and inefficiencies in the transport network. The lack of a centralized system to regulate and monitor these activities exacerbates the problem, making it challenging to ensure discipline and safety in the transportation sector.

3.2.Goal

Our objective is to design and implement an efficient and centralized transport management system that addresses these issues by introducing smart solutions for monitoring and controlling bus operations. By leveraging technology, we aim to streamline the transportation system, reduce road accidents, and promote a safer, more organized, and modernized approach to public transit management.

3.3.SDLC

The Software Development Life Cycle (SDLC) is a systematic and structured process used for developing software applications. It defines a series of steps or phases that guide teams in planning, creating, testing, deploying, and maintaining software systems. SDLC serves as a blueprint to ensure that the software meets the required standards, fulfills user expectations, and is delivered within a stipulated time-frame and budget.

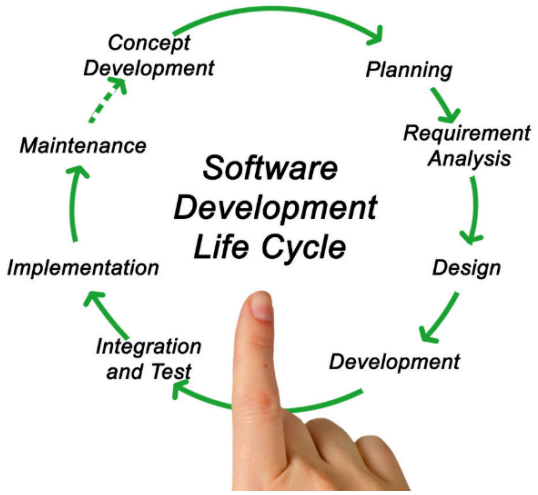


Fig :- Steps of SDLC

Here's a detailed breakdown of each area covered in the report :-

- **Idea Generation :-** This section focuses on brainstorming and conceptualizing the core idea for the project or system. It involves identifying the problem to be solved, understanding the objectives, and exploring various possible solutions. The idea generation phase may include discussions, research, and preliminary feasibility studies to determine the most promising concept.
- **Initial Planning :-** The initial planning phase outlines the scope, goals, and timeline of the project. This includes setting clear objectives, defining the project requirements, and establishing the resources needed, such as personnel, equipment. It also involves identifying key milestones, deliverables, and deadlines to ensure the project stays on track.
- **Information Gathering :-** In this phase, data is collected from various sources to gain a deeper understanding of the project's requirements and constraints. This may involve conducting surveys, interviews, reviewing existing documentation, or analyzing current systems. The gathered

information helps in shaping the design and ensuring that the solution is aligned with user needs and project goals.

- **System Analysis :-** System analysis focuses on examining the existing system (if applicable) or defining the new system's functional and non-functional requirements. This phase includes identifying the inputs, outputs, and processes involved in the system, and evaluating how the system interacts with external entities. The goal is to identify potential problems, inefficiencies, or areas for improvement.
- **System Design :-** The system design phase involves creating detailed blueprints for the system’s architecture, components, and interfaces. It translates the requirements and analysis into a tangible structure, specifying the hardware, software, databases, and network configurations needed for the system. This phase may include creating design diagrams, such as use case diagrams, class diagrams, and deployment diagrams, to represent the system's components and their interactions.

4.Information Gathering

4.1. Information Sources

Internal Sources :-

- **Personal Staff :-** Personal staff assists us by addressing current processes and challenges, providing valuable support in identifying areas for improvement, and helping enhance the overall system's efficiency and functionality.
- **System Documentation :-** System documentation helps us by providing essential resources such as system configurations, design documents, flowcharts, and user manuals, all of which contribute to a comprehensive understanding of the system’s structure and functionality.
- **Report & Transactions :-** Reports and transactions help us by providing crucial insights into revenue reports, service logs, system performance, usage trends, and financial health, enabling us to monitor and optimize the overall operations effectively.

External Sources :-

- **Benchmark Products :-** Benchmark products help us by providing insights into best practices, unique features, functionality, and performance, allowing us to compare and enhance our own product based on industry standards.
- **Professional Journals :-** Professional journals help us stay informed about industry trends, regulations, technology advancements, and provide valuable market and tech insights, keeping us updated on the latest developments and best practices.
- **Government Documents :-** Government documents help by providing essential information on regulations, policies, system compliance, safety standards, and environmental

rules, ensuring that our operations align with legal requirements and industry guidelines.

Combined Sources :-

- **Consultant :-** Consultants help by guiding technology selection and facilitating system integration, ensuring that the chosen solutions are effective and seamlessly incorporated into existing infrastructures.
- **The Users :-** Users help by providing valuable feedback on usability, functionality, and performance through surveys and interviews, offering insights that drive improvements and enhancements to the system.

4.2.Information Gathering Tools

- **Product Reports :-** For product reports, we explore sources such as IEEE Xplore, Emerald Insight, MDPI, and ResearchGate to gather relevant research, insights, and industry developments.
- **Policy Papers :-** For policy papers, we refer to STA for smart transportation solutions and TRB for research and policy papers to stay informed on the latest developments and guidelines.
- **User Report :-** For user reports, we explore "Customer-Centric Design for Transportation" on ResearchGate and "User Experience in Smart Transportation" on SpringerLink to gain insights into user-focused approaches and experience design in transportation systems.
- **Government Documents :-** For government documentation, we explore the Bangladesh Road Transport Authority (BRTA) and the Nagar Poribahan (Urban Transport) system to gather relevant information and ensure

compliance with regulations related to road transportation and urban mobility.

- **Vendors :-** For vendors, we explore resources such as The Transport Technology Suppliers Directory (ITS International) and The Global Transport Technology Vendor List (Smart City Press) to identify key players and innovative solutions in the transport technology sector.
- **Websites :-** For useful websites, we explore TRB Publications, Smart Cities Dive (which covers smart city technology trends), and UITP (a global resource for public transport) to stay updated on the latest developments and innovations in transportation and urban mobility.

4.3.Benchmark Products

Features	sohoj.com BD	Jatri BD	Bd-Ticket	Bus BD	Octopus Hong Kong City Transport	Compass UK City Transport	*Move Me*
Ticketing	✓	✓	✓	✓	✗	✗	✗
Bus Services	✓	✓	✓	✓	✓	✓	✓
Admin Management	✗	✗	✗	✗	✓	✓	✓
Route Information	✓	✓	✗	✓	✓	✓	✓
Online Payments	✓	✓	✓	✓	✓	✓	✓
Real-Time Tracking	✗	✗	✗	✗	✓	✓	✓
Schedule Management	✗	✗	✗	✗	✓	✓	✓
MRT Access	✗	✗	✗	✗	✓	✓	✓
Revenue	✓	✗	✓	✓	✓	✓	✓

Fig :- Benchmark Products

4.4. Survey

- We collected information through face-to-face interviews with bus owners and drivers.
- We collected information from passengers by distributing questionnaires through online Google Forms.

5.System Analysis

5.1.Survey Analysis

- **Analyze Interview with Busowners :-**
  - Understanding the current methods used for tracking and managing revenue across the fleet.
  - Identifying the challenges faced in managing a fleet of buses, including coordination and scheduling.
  - Exploring the potential interest in a subscription-based model for bus services and the key features that would add value.
- **Analyze Interview with Drivers :-**
  - Understanding the drivers' preferences and experiences regarding salary-based payment models.
  - Exploring the feasibility of operating buses without a conductor, and how it impacts the driver's responsibilities.
- **Analyze Questionnaires with Passengers :-**
  - The survey responses indicate strong interest in the MoveMe software, with 80-90% of respondents indicating they would use it if it simplified ticketing, improved travel management, and provided operational transparency.
  - **Bus Location Tracking :-** 90% of respondents consider it "Very important."
  - **System's Impact on Commuting Experience :-** 70% believe it would improve their experience.
  - **Transparency in Fare Management :-** 80% would adopt the app if it helped monitor transportation spending.
  - **RFID Card System :-** Around 65-70% support using it, though some prefer cash or other methods.

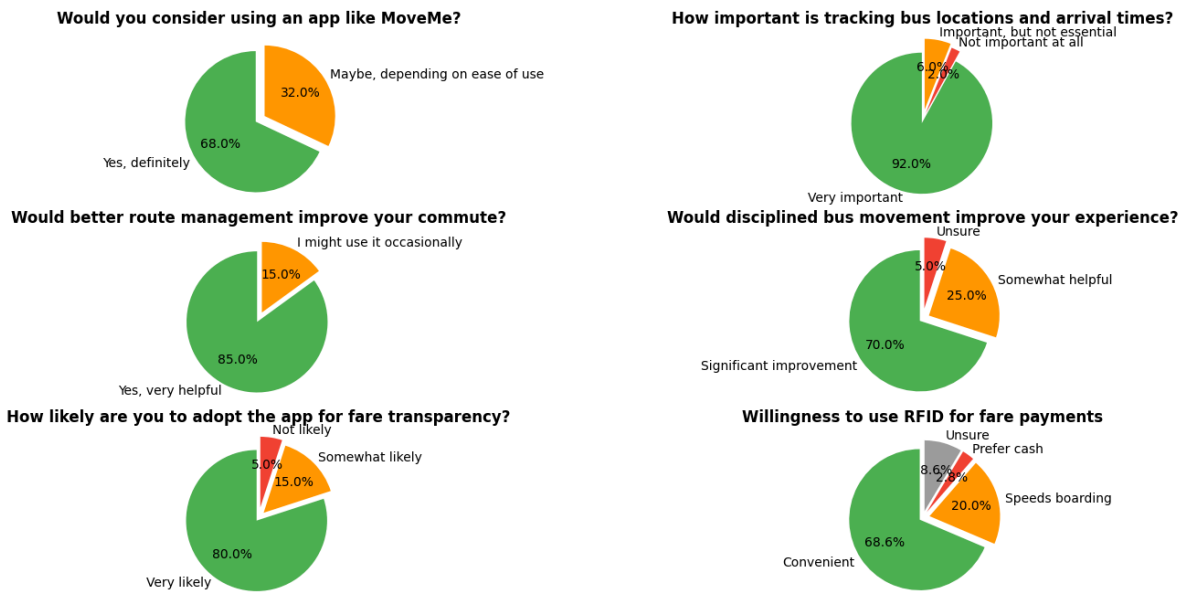


Fig :- Form Analysis

5.2.Benchmark Gap Analysis

- **Ticketing :-** Sohoj.com BD, Jatri BD, Bd-Ticket, and Bus BD provide ticketing services, but Octopus Hong Kong City Transport, Compass UK City Transport, and Move Me do not offer this feature. There is a gap in ticketing functionality for these services.
- **City Bus Services :-** Jatri BD, Compass UK City Transport, and Move Me offer city bus services, while Sohoj.com BD, Bd-Ticket, Bus BD, and Octopus Hong Kong City Transport do not. The gap is in the availability of city bus services for certain platforms.
- **Admin Management :-** Octopus Hong Kong City Transport, Compass UK City Transport, and Move Me provide admin management features, but Sohoj.com BD, Jatri BD, and Bd-Ticket do not. The gap here suggests that these platforms may lack the backend support for administrative control.
- **Online Payments :-** All platforms provide online payment options, so there is no gap in this feature.

- **Real-Time Tracking :-** Only Octopus Hong Kong City Transport, Compass UK City Transport, and Move Me offer real-time tracking. Sohoj.com BD, Jatri BD, Bd-Ticket, and Bus BD do not, highlighting a gap in real-time tracking functionality for these platforms.
- **MRT Access :-** Octopus Hong Kong City Transport, Compass UK City Transport, and Move Me offer MRT (Mass Rapid Transit) access, while the others do not. This reveals a gap in providing access to rapid transit systems for Sohoj.com BD, Jatri BD, Bd-Ticket, and Bus BD.
- **Revenue :-** Sohoj.com BD, Bd-Ticket, Bus BD, Octopus Hong Kong City Transport, Compass UK City Transport, and Move Me provide revenue management or collection, but Jatri BD does not. This indicates a gap in revenue generation for Jatri BD.

5.3.Feature List Fixation

Features of MoveMe :-

- **Route Management :-** Route Management: MoveMe offers route management, which is critical for planning and optimizing transport routes.
- **City Bus Service :-** MoveMe provides city bus services, catering to urban transport needs.
- **Admin Management :-** MoveMe includes admin management features for controlling and overseeing the transport operations.
- **Real-Time Tracking :-** MoveMe offers real-time tracking, enabling users to track their transportation in real time.
- **MRT Access :-** MoveMe offers access to MRT (Mass Rapid Transit), enabling users to plan their journeys involving rapid transit systems.

- **Revenue :-** MoveMe has a revenue management system to collect and track fares and payments.

5.4.Functional & Non-Functional Requirements

Functional Requirements :-

- **Route Management :-** Ability to manage and optimize transport routes.
- **City Bus Service :-** Provide city bus services with real-time scheduling and tracking.
- **Admin Management :-** Admin panel to oversee and control transport operations.
- **Real-Time Tracking :-** Track buses and MRT services in real-time for users.
- **MRT Access :-** Integration with MRT for seamless planning of journeys involving both bus and rapid transit systems.
- **Revenue Management :-** System to collect, track, and report fare payments.

Non-Functional Requirements :-

- **Performance :-** The system must support high traffic and provide fast real-time updates.
- **Scalability :-** The platform should handle an increasing number of buses, routes, and users.
- **Reliability :-** The system should be reliable and operate without interruptions.
- **Usability :-** The software should have an intuitive interface for both users and administrators.
- **Security :-** Ensure data security for users' personal and payment information.

5.5.Feasibility Analysis

- **Software Requirements :-** The MoveMe software requires four key user roles: the Employee Team, responsible for system management and administrative tasks; Bus-Owners, who manage their fleets, routes, and schedules; Shop-Owners, who collaborate with the platform for ticket sales and services; and Passengers, the end-users who rely on the app for route planning, real-time tracking, schedule management, and ticket purchasing. Each role plays a crucial part in ensuring the smooth operation of the transportation system and enhancing the user experience.
- **Software Team :-** The MoveMe software team includes Frontend Developers (designs the user interface), Backend Developers (handles server-side logic and database management), and a Project Manager (coordinates the team, manages progress, and ensures alignment with business goals).
- **Hardware Requirements :-** The MoveMe system requires an RFID Reader as part of its hardware setup. This device is essential for reading RFID tags on buses, tickets, or other transportation assets, enabling efficient tracking, access control, and payment processing. It ensures smooth interaction between passengers, bus owners, and the system for tasks such as ticket validation and real-time tracking.
- **Hardware Team :-** The MoveMe hardware team focuses on three key components: the RFID System for ticket validation and tracking, the Bus Power System to ensure reliable power for onboard devices, and the Electrical System to manage the bus's electrical infrastructure, supporting systems like GPS, communication tools, and the RFID reader.

5.6.SWOT Analysis

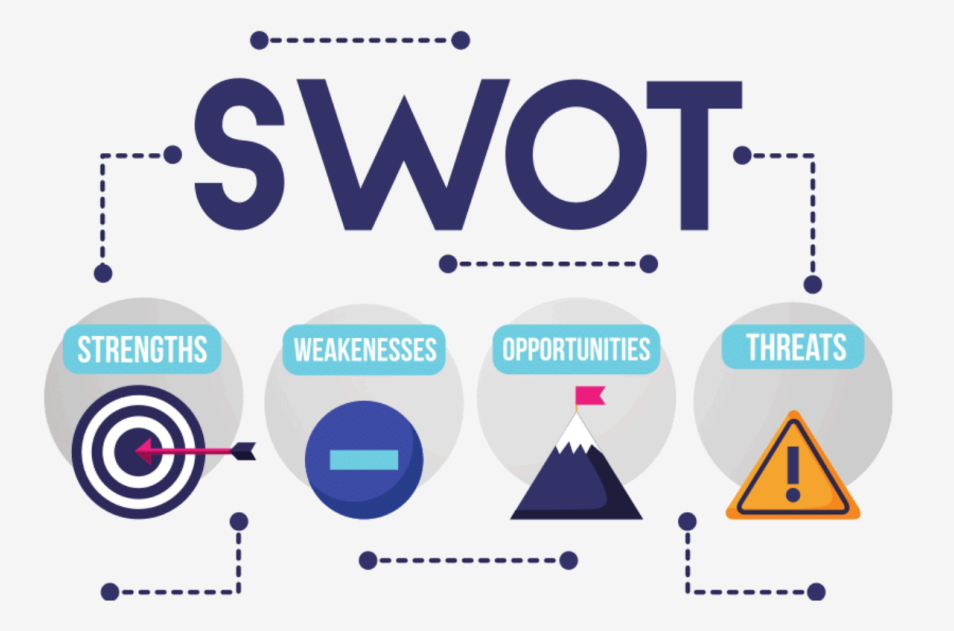


Fig :- SWOT

- **Strength :-** The MoveMe project offers enhanced efficiency, a user-friendly interface, modern technology, a sustainable system, and a scalable design to support future growth and evolving transportation needs.
- **Weaknesses :-** The MoveMe project has potential weaknesses, including high costs, technical dependence, and a learning curve for users and operators adapting to the system.
- **Opportunities :-** The MoveMe project presents opportunities such as growing demand for efficient transportation, government support for smart mobility, data utilization for improved services, strategic partnerships with businesses, and potential for expansion into new markets.
- **Threates :-** The MoveMe project faces threats from strong competition, privacy concerns regarding user data, potential technical failures, and user resistance to adopting new technology.

6.System Design

6.1.Context Diagram

A Context Diagram provides a high-level overview of the system, showing a single process (Process 0) connected to external entities, helping identify the project scope early on and related to Data Flow Diagram Level 0.

- **Steps :-**
  - Label system/process (verb).
  - Identify interacting entities (noun).
  - Group similar entities.
  - Identify major data flows.
- **Rules :-**
  - Processes: Circles, "0" + capitalized name.
  - External entities: Rectangles, all capital letters.
  - Data flows: Line Connection.

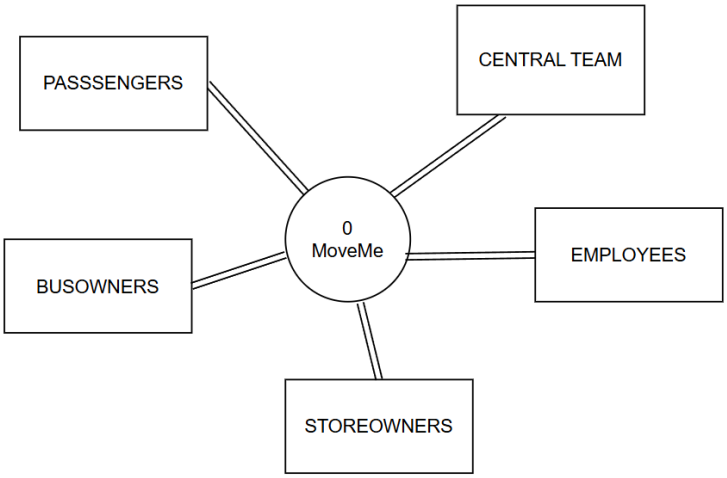


Fig :- Context Diagram (MoveMe)

6.2.Data-Flow Diagram

A Data Flow Diagram (DFD) represents the flow of data between entities, data stores, and processes. It shows the inputs and outputs of each entity and process, without including control flow, decision rules, or loops.

- **Components :-**
  - **Process :-** Changes data and produces an output.
  - **External Entity :-** Outside system that sends/receives data.
  - **Data Store :-** Holds information for later use (for e.g. database).
  - **Data Flow :-** Path data takes between entities, processes, and stores.

- **Notations :-**

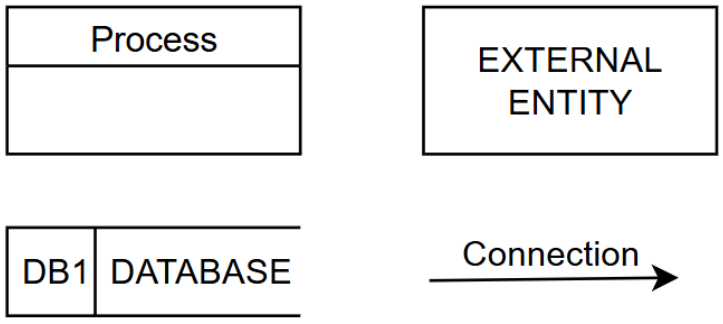


Fig :- DFD Notations

- **Rules :-**
  - Processes and sub-processes must have a name and number, with capitalized first letters.
  - External Entities and Data Stores are in all caps.
  - Each process must have at least one input and one output.

- Each data store must have at least one input and one output.
- Data must pass through a process.
- All processes lead to another process or data store.
- No direct connections: DB to DB, DB to EE, EE to EE.
- DFD starts from top-left and ends at bottom-right.

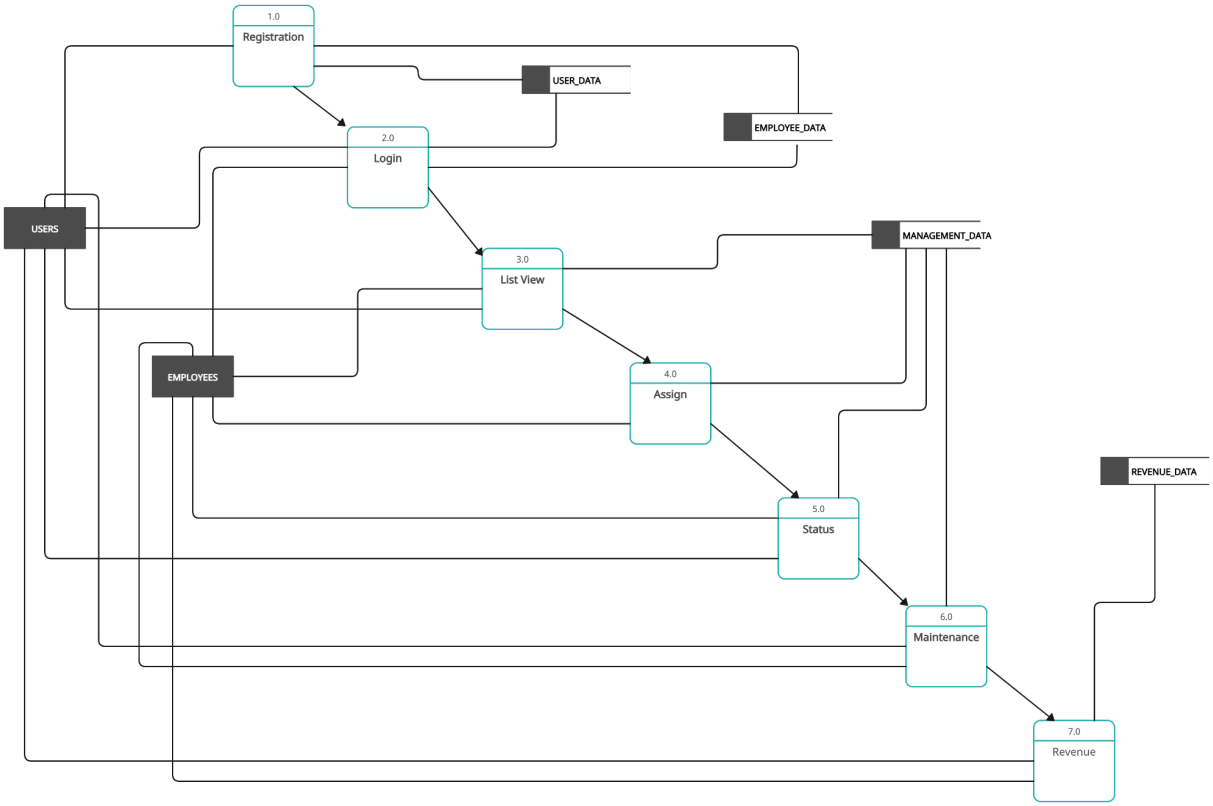


Fig :- Data-Flow Diagram (MoveMe)

6.3.Use-Case Diagram

A Use Case Diagram shows how actors interact with a system to achieve specific goals, focusing on what the system does.

- **Notations :-**

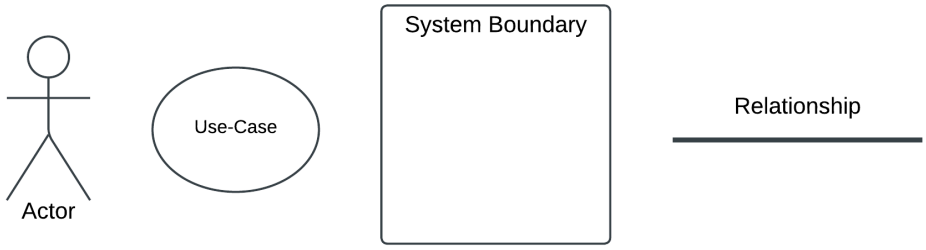


Fig :- Use-Case Notations

- **Relationships :-**
  - **Include :-** The Include Relationship shows one use case including the functionality of another, represented by a dashed arrow.
  - **Extend :-** The Extend Relationship shows a use case being extended by another under specific conditions, represented by a dashed arrow with "extend."

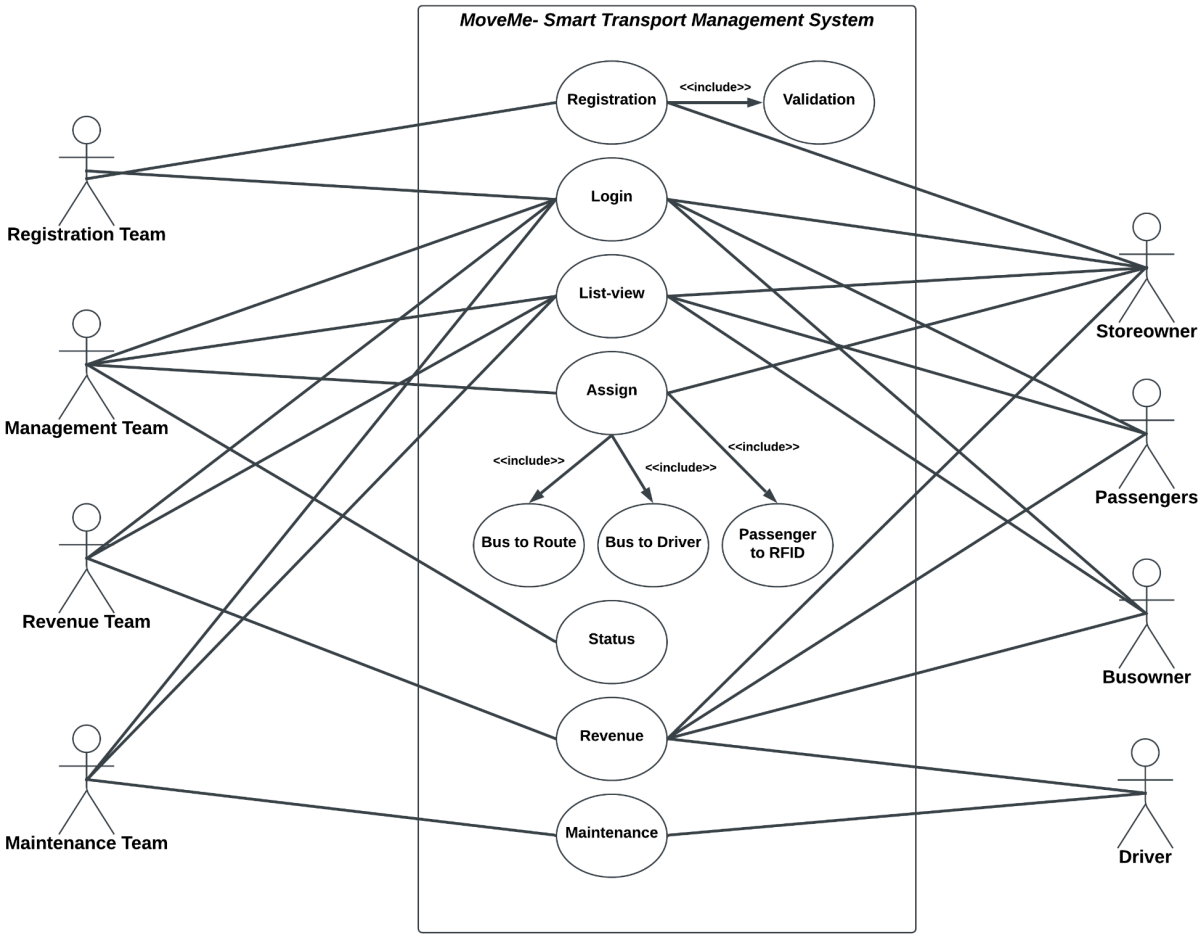


Fig :- Use-Case Diagram (MoveMe)

6.4.Use-Case Descriptive Forms (2 Use-Cases)

Use-Case :- Assign - Bus to Route

1. **Name :-** Assign Bus to Route.
2. **Description :-** The bus management team can assign a route to a specific bus.
3. **Stakeholders & Interests :-**
    a. **Management Team :-** The bus management team can assign a bus to a specific route.
    b. **Busowners :-** Bus owners can view the status of their buses, including which bus is assigned to which route.
4. **Primary Actor :-** Management Team.
5. **Pre-conditions :-**
    a. Busowners must be registered.
    b. Buses must be registered.
6. **Success Scenario :-**
    a. The management team can view buses successfully.
    b. The management team can manage routes successfully.
    c. The management team can assign routes to buses successfully.
    d. Buses can successfully move to their assigned routes.
7. **Alternative Scenario :-**
    a. The central team will temporarily manage buses.
    b. The central team will temporarily manage routes.
    c. The central team will assign buses to routes.
    d. If a system failure occurs, the system will undergo maintenance.
8. **Post-conditions :-**
    a. Passengers can travel by that bus.
    b. The fare will be successfully collected.
    c. Revenue will be generated.

Use-Case :- Revenue

1. **Name :-** Revenue.
2. **Description :-** Revenue is generated through several key sources: fare collection from passengers for the transportation service; employee payments, which may include salaries, wages, and performance incentives tied to service operations; bus owners' earnings from leasing or operating buses within the system, often based on a percentage of fare revenue; and store owners' profits from retail spaces within transportation hubs, where they assign RFID and services to passengers. These components collectively contribute to the overall financial performance of the transportation and retail ecosystem.
3. **Stakeholders & Interests :-**
    a. **Passengers :-** Track their transportation expenses.
    b. **Busowners :-** Monitor their profits from bus operations.
    c. **Storeowners :-** Track profits from retail sales at transportation hubs.
    d. **Revenue Team :-** Oversee and manage all revenue processes.
    e. **Employees :-** Ensure timely salary and wage payments.
4. **Primary Actor :-** Revenue Team.
5. **Pre-conditions :-**
    a. Must ensure that all buses are properly registered within the system.
    b. Each bus should be assigned to specific routes for operation.
    c. Passengers must press their RFID cards to facilitate fare collection during boarding.

6. **Success Scenario :-**
    a. The RFID Reader functions correctly and successfully reads passengers' RFID cards.
    b. The fare is collected from passengers without any issues.
    c. The fare is added to the database accurately and promptly.
    d. The Revenue Team can successfully track and monitor the collected fare records in the system.
7. **Alternative Scenario :-**
    a. If the RFID Reader is not working, it should be repaired promptly to resume fare collection.
    b. If the fare is not added to the database, the system database should be repaired to ensure proper data entry.
    c. In case of a system failure, the system will undergo maintenance to restore full functionality.
8. **Post-conditions :-**
    a. Passengers can check and track their transportation expenses.
    b. Busowners and Storeowners can view and assess their respective profits from the system.
    c. Employee payments are processed and completed successfully.

6.5.Activity Diagram

An Activity Diagram visually represents how activities are coordinated to achieve a specific event, similar to a flowchart. It is commonly used to model business workflows and use cases, illustrating the sequence of actions, decisions, and interactions that occur within a process.

- **Components & Notations :-**
  - **Action :-** In an activity diagram, an action represents a task or a piece of work that is performed. This could be anything from a simple operation to a complex process. It's usually depicted as a rounded rectangle.
  - **Control Flow :-** Control flow represents the sequence in which the actions or activities are executed. It shows the direction of the process, often visualized as arrows pointing from one action to another.
  - **Initial Node :-** The initial node marks the start of the activity diagram. It's typically depicted as a filled black circle and shows where the process begins.
  - **Activity Final Node :-** The initial node marks the start of the activity diagram. It's typically depicted as a filled black circle and shows where the process begins.
  - **Decision Node :-** Splits the flow based on a condition (if/else).
  - **Merge Node :-** Combines multiple flows into one.
  - **Fork Node :-** Splits the flow into parallel or concurrent paths.
  - **Join Node :-** Merges parallel flows back into one.

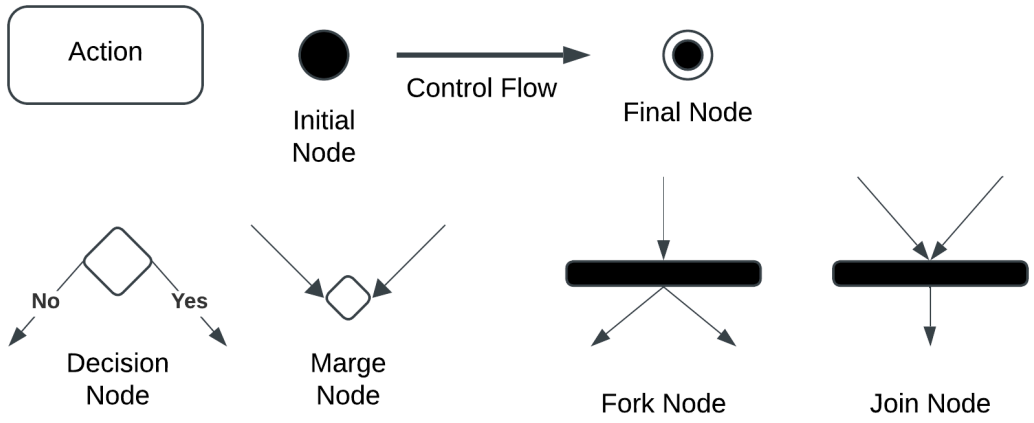


Fig :- Activity Diagram Notations

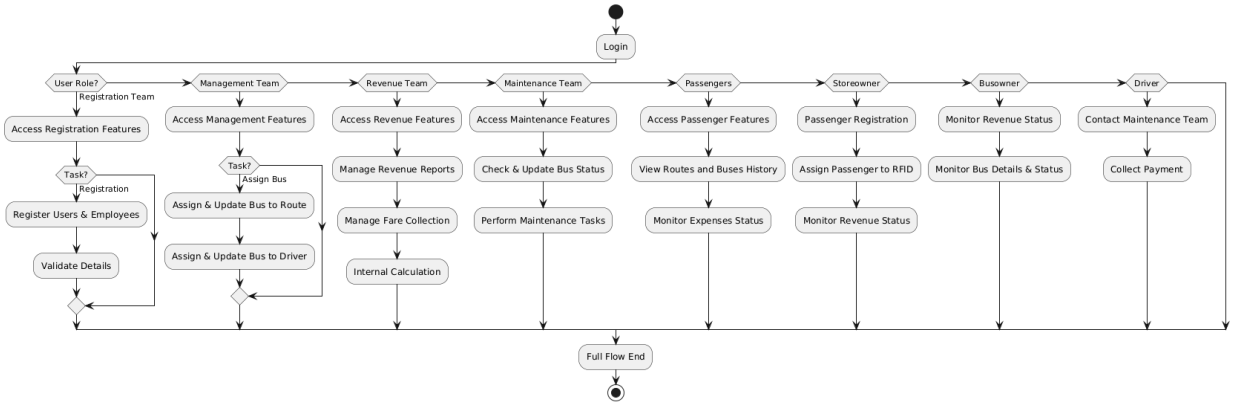


Fig9 :- Activity Diagram (MoveMe)

Activity Diagram of MoveMe Explanation :-

- The flow begins with a login action, where users authenticate themselves into the system.
- After login, users are directed based on their roles. The diagram includes distinct roles.
- Each user role has specific features or tasks they can access.
- Several activities are based on decision-making, such as determining the specific task under a role.
- The diagram concludes with a "Full Flow End," where all activities for the session or process terminate.

6.6.Swimlane Diagram

A Swimlane Diagram groups activities by actor in an activity diagram, visually distinguishing responsibilities among stakeholders. It can be arranged horizontally or vertically.

- **Components & Notations :-**
  - **Swimlanes & Partitions :-** These are the primary additions in a Swimlane Diagram, grouping activities based on the actor responsible for them, either horizontally or vertically.
  - **Similar to Activity Diagram :-** Other components like activities, flow arrows, and decision nodes remain the same as in an activity diagram, with the swimlanes simply organizing the process by actor.

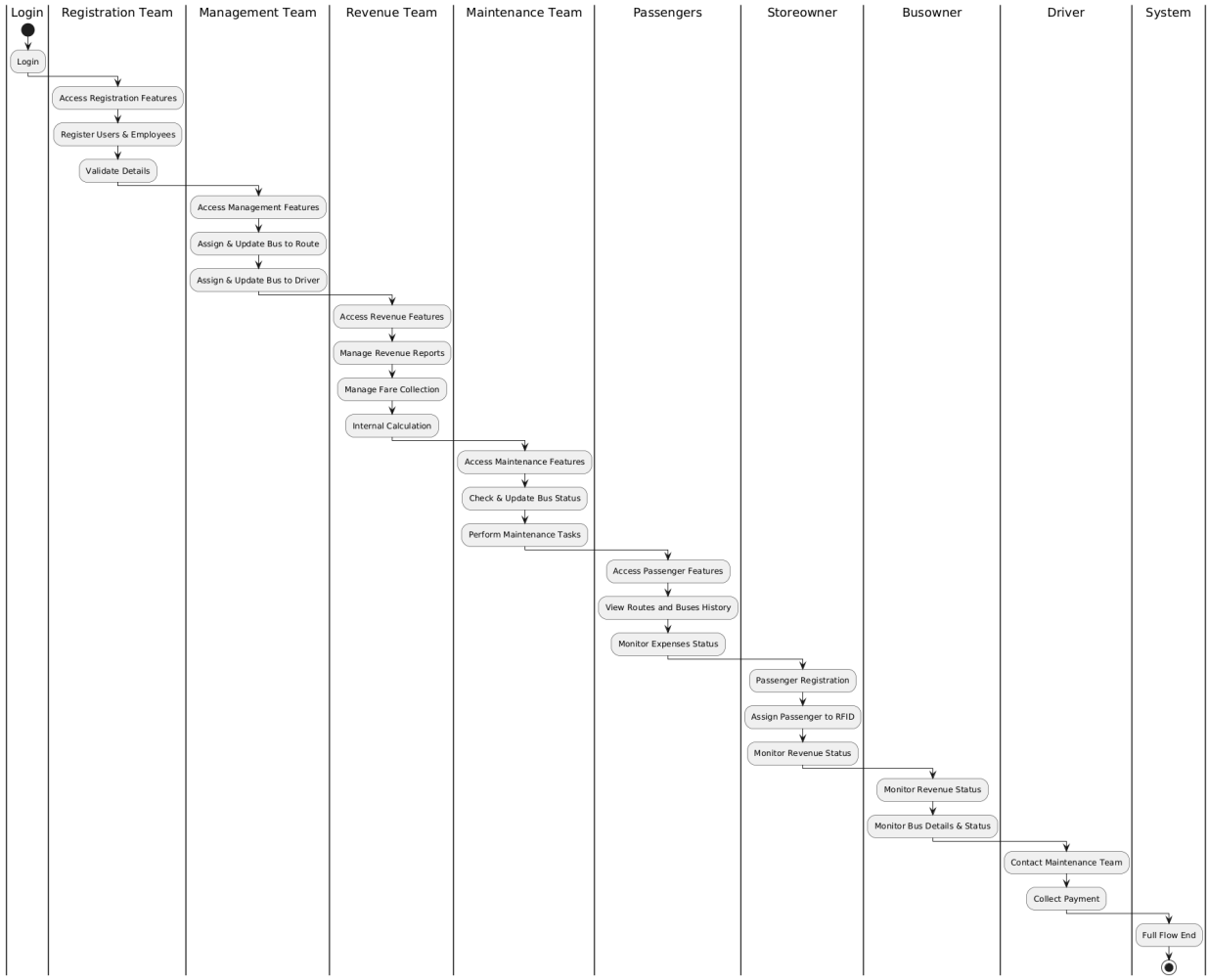


Fig :- Swimlane Diagram (MoveMe)

Swimlane Diagram of MoveMe Explanation :-

- **Login :-** The flow starts with the login action, where users authenticate themselves to access the system. After successful authentication, users are redirected based on their role.
- **User Roles :-** The system defines distinct user roles that determine access to features. These roles include Registration Team, Management Team, Revenue Team, Maintenance Team, Passengers, Storeowners, Bus Owners, and Drivers.
- **Role Based Features :-** Each user role has specific tasks and privileges, such as the Registration Team registering

users and validating details, the Management Team assigning buses to routes, the Revenue Team managing reports and fares, the Maintenance Team checking bus statuses, Passengers viewing routes and bus histories, Storeowners registering passengers and assigning RFID, Bus Owners monitoring revenue and bus details, and Drivers contacting maintenance and collecting

- **Task Decisions :-** The diagram includes decision points where users must choose between different tasks, based on the role-specific needs or situations, like registering a new passenger or assigning a bus.
- **End Flow :-** The swimlane diagram ends at the Full Flow End, marking the conclusion of all activities for the session or process. This signifies the termination of the current tasks and session.

6.7.Class Diagram

A class diagram represents the structure and relationships of classes within a system.

- **Benefits :-**
  - Provides a clear visual structure of the system's classes, attributes, and methods, helping to organize the system's design effectively.
  - Visualizes relationships between classes (e.g., inheritance, associations, dependencies), aiding in understanding how components interact and the impact of changes.
  - Enhances communication among team members (developers, designers, stakeholders) by providing a standardized visual representation that everyone can understand.
  - Acts as a guide for developers during the implementation phase, ensuring the system's design is correctly followed and reducing potential design flaws.
  - Speeds up the development process and ensures that the code structure aligns with the design, reducing the risk of inconsistencies or errors.
- **Notations :-**
  - **Class Name :-** Represents the blueprint or structure of the class.
  - **Attributes :-** Define the data members or properties of the class.
  - **Methods :-** Describe the behavior or functionality that the class can perform.

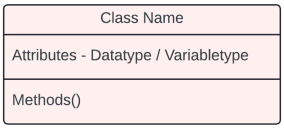


Fig11 :- Class Diagram Notations

- **Relationships :-**
  - **Association :-**
    - Represents a relationship between two classes.
    - Indicates how objects of one class are connected to objects of another class.
    - Depicted as a solid line in a class diagram.
  - **Generalization :-**
    - Subclass inherits the properties and behaviors of Superclass.
    - Shown as a solid line with a hollow arrowhead from subclass to superclass.

Fig12 :- Association

Fig13 :- Generalization / Inheritance

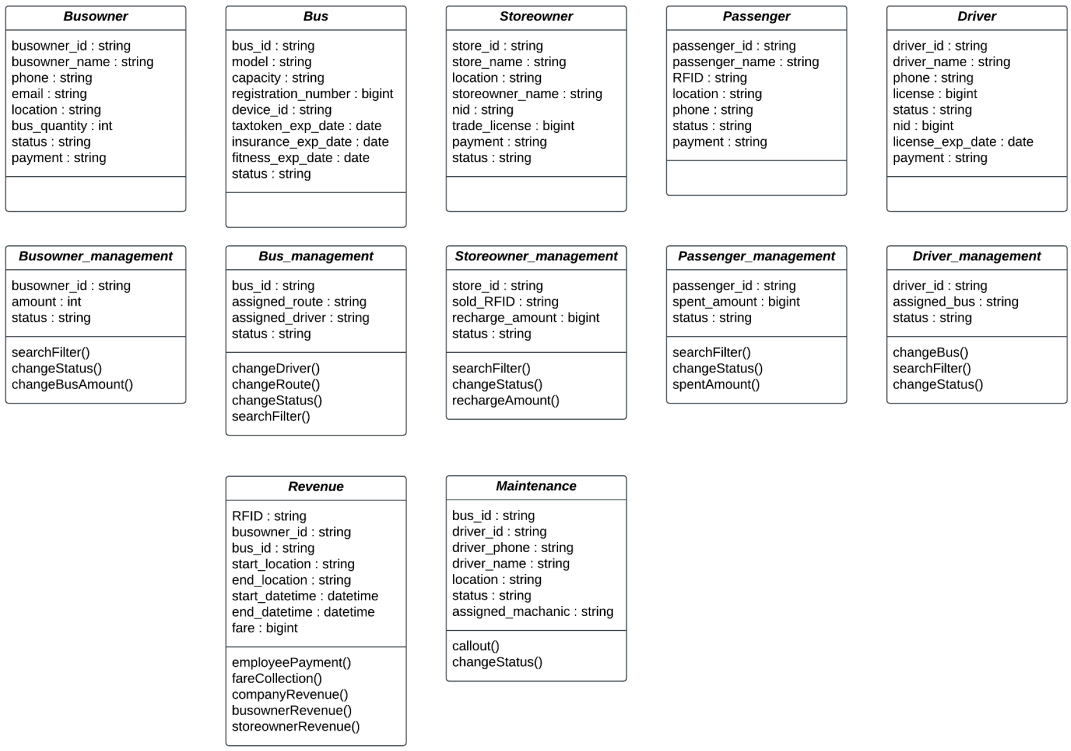


Fig :- All Classes (MoveMe)

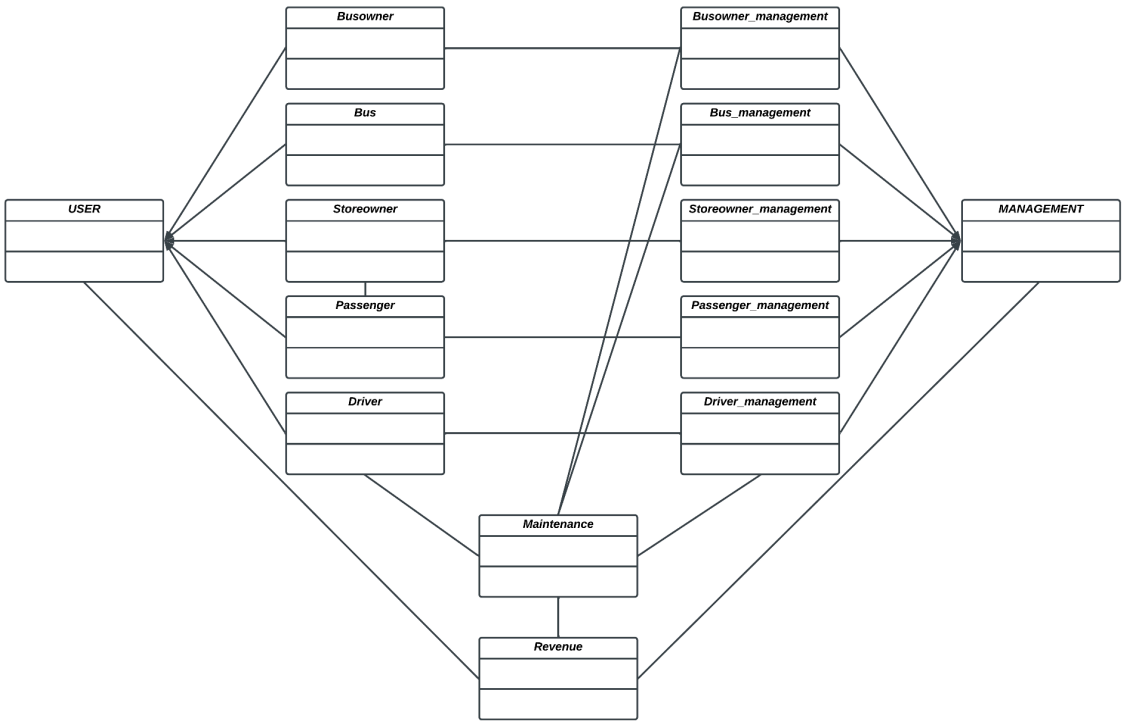


Fig15 :- Class Diagram (MoveMe)

Class Diagram of MoveMe Explanation :-

- User classes are linked to specific management teams, reflecting operational workflows.
- Revenue class is connected to all classes, handling financial data across the system.
- Maintenance class is connected to some particular management class and driver class.

6.8.CRC Diagram

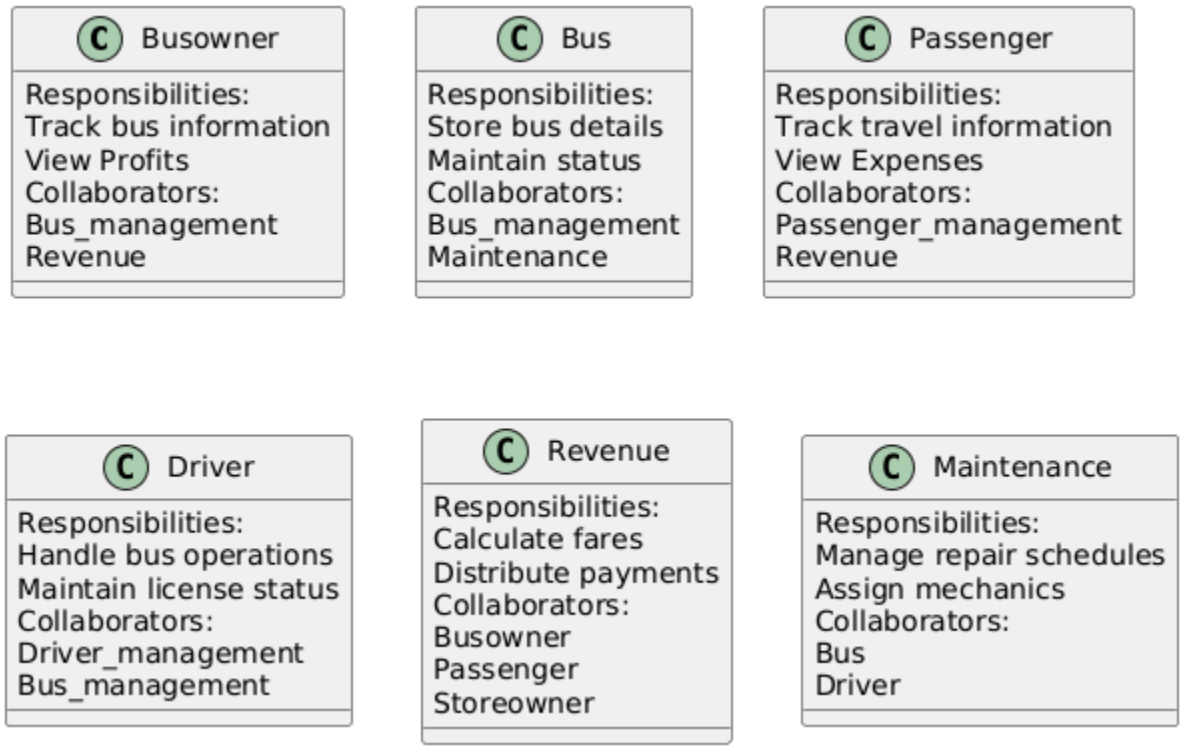


Fig :- CRC Diagram

6.9.Sequence Diagram

A Sequence Diagram illustrates the interactions between objects in a system, showing the order and flow of messages exchanged over time.

- **Benefits :-**
  - Provides a clear visual representation of how the system behaves over time, showing the sequence of interactions between objects.
  - Helps in understanding and refining the system's design and architecture by illustrating the flow of control between components.
  - Assists in clarifying and confirming system requirements by mapping out the interactions needed to meet functional goals.
  - Aids in identifying and resolving issues by visualizing the order of operations, which helps trace errors or unexpected behaviors.
- **Notations :-**

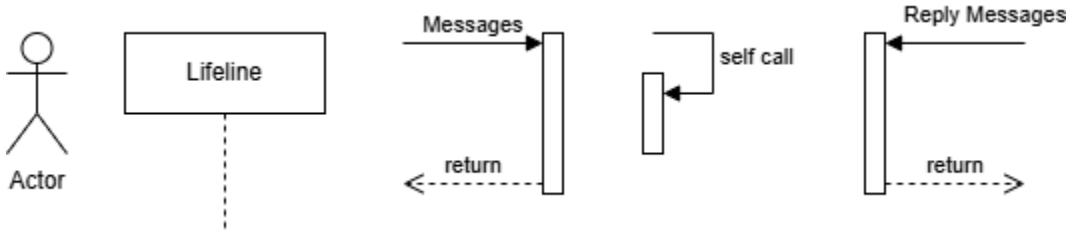


Fig16 :- Sequence Diagram Notations

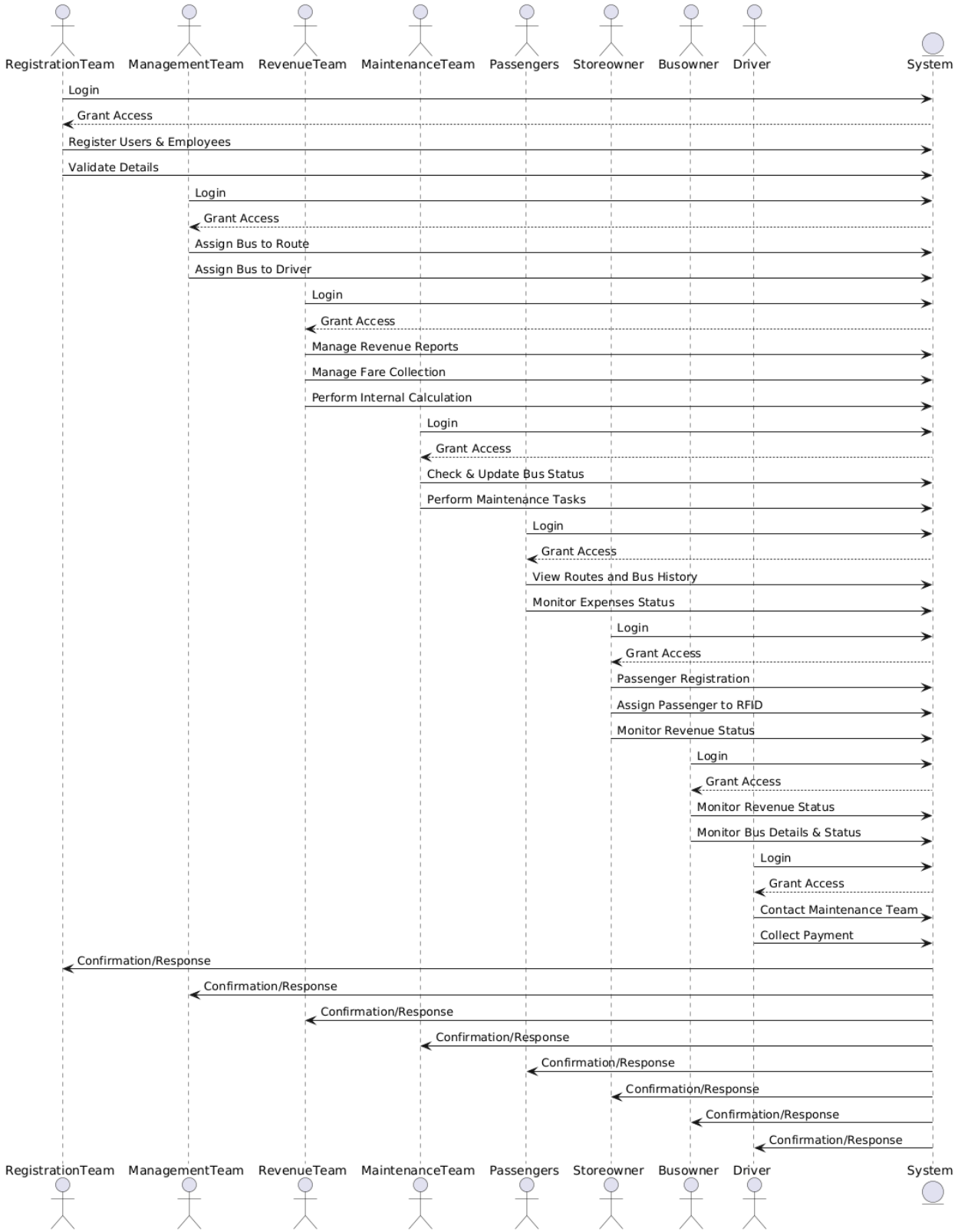


Fig :- Sequence Diagram (MoveMe)

Sequence Diagram of MoveMe Explanation :-

- The Registration Team logs into the system, grants access, registers users and employees, and validates details.
- The Management Team logs in, grants access, assigns buses to routes, and assigns buses to drivers.
- The Revenue Team logs in, manages revenue reports, manages fare collection, and performs internal calculations.
- The Maintenance Team logs in, grants access, checks and updates bus status, and performs maintenance tasks.
- Passengers log in, register as passengers, view routes and bus history, monitor expenses.
- The Store Owner logs in and monitors revenue status.
- The Driver contacts the maintenance team, and collects payments.
- The System processes all interactions and sends confirmation or responses to the respective actors.

6.10.State Diagram

A state diagram models the system's dynamic behavior, showcasing its finite states, transitions, and responses to events over time.

- **Notation :-**
  - Same as an Activity Diagram.
  - Small changes include event transitions between states.

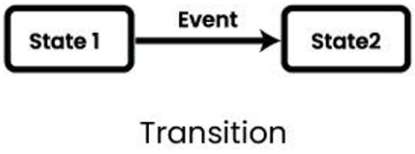


Fig18 :- States & Events

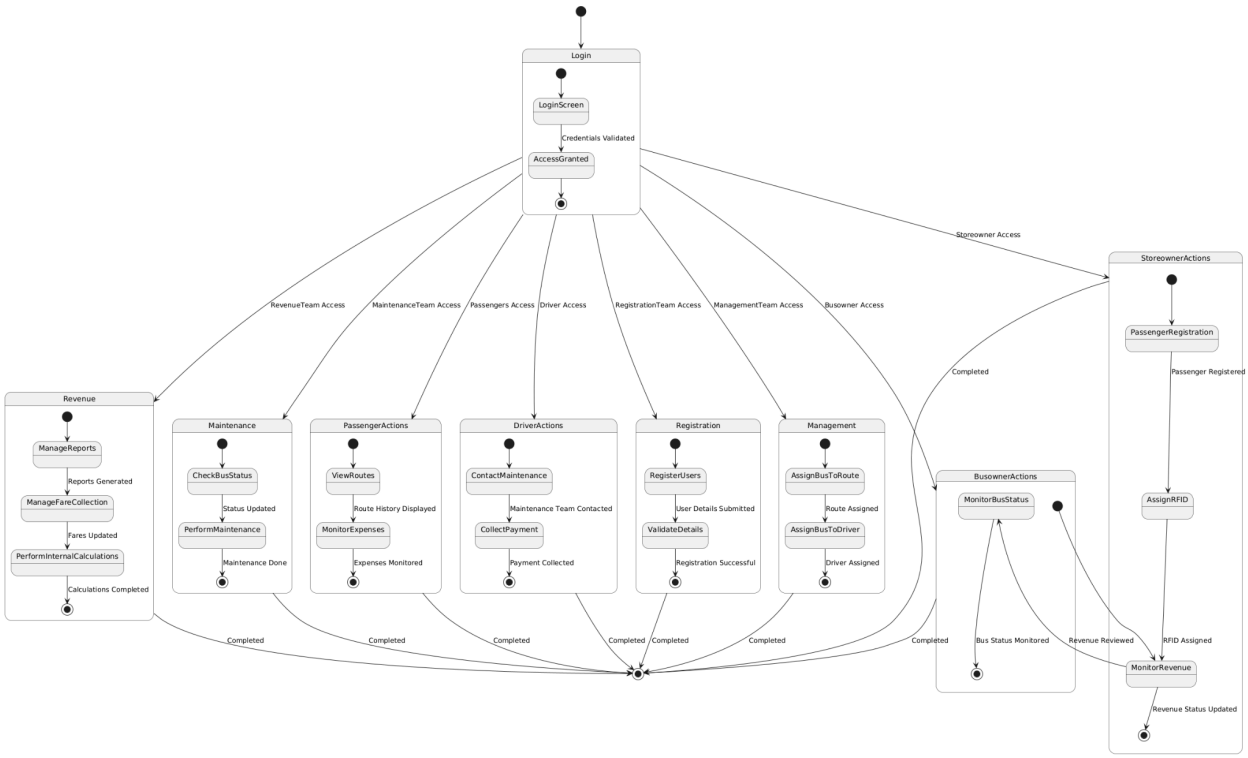


Fig :- State Diagram (MoveMe)

State Diagram of MoveMe explanation :-

The system involves various actions: validating credentials and granting access, generating reports, managing fare collection, and performing calculations. It includes checking bus status, performing maintenance, viewing routes, and monitoring expenses. Drivers can contact maintenance and collect payments, while users are registered, and their details validated. Buses are assigned to routes, and drivers are assigned to buses. Bus owners monitor bus status, and review profit. Store owners register passengers and view their profit.

6.11.Entity Relationship Diagram (Short)

An Entity-Relationship Diagram (ERD) is a visual representation of data that depicts entities, their attributes, and the relationships between them, used to model and design database structures.

- **Notations :-**

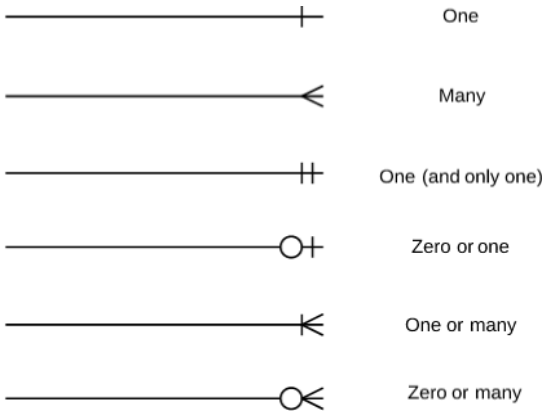


Fig20 :- ERD Notations

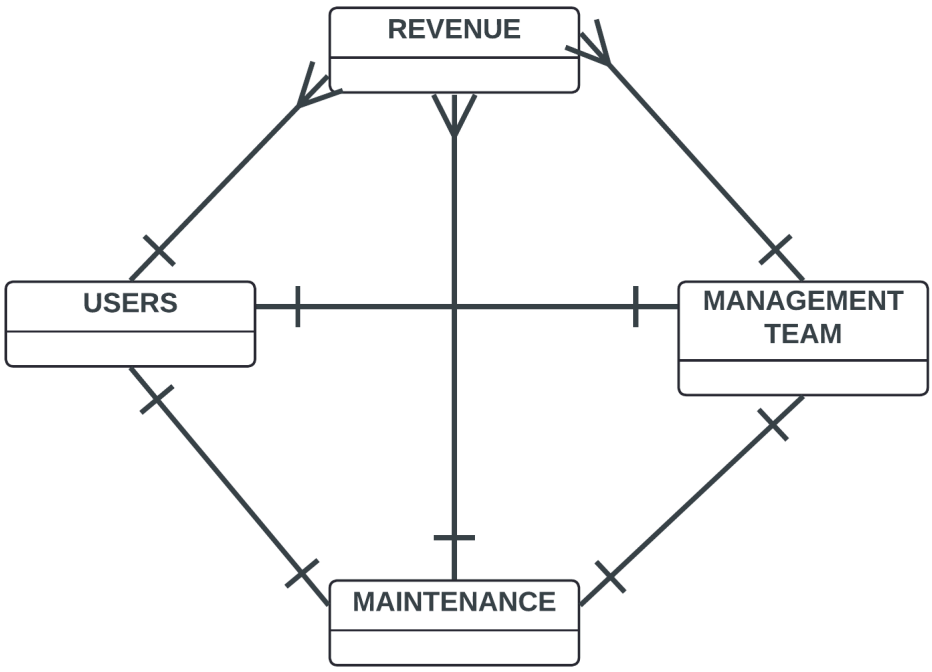


Fig :- Short ERD (MoveMe)

Short ERD of MoveMe explanation :-

- A single management team entity can have multiple associated revenue records.
- A single user can be associated with multiple revenue records, but each revenue record is linked to one specific user.
- Each user is associated with exactly one management team entity, and vice versa.
- Each maintenance team is associated with exactly one user (Driver), and vice versa.
- Each maintenance team is linked to a single management team entity (only at that time).
- A single maintenance team can generate multiple revenue records.

6.12.Deployment Diagram

A deployment diagram visually represents the runtime configuration of processing nodes and the components on them, modeling the physical architecture of an object-oriented system.

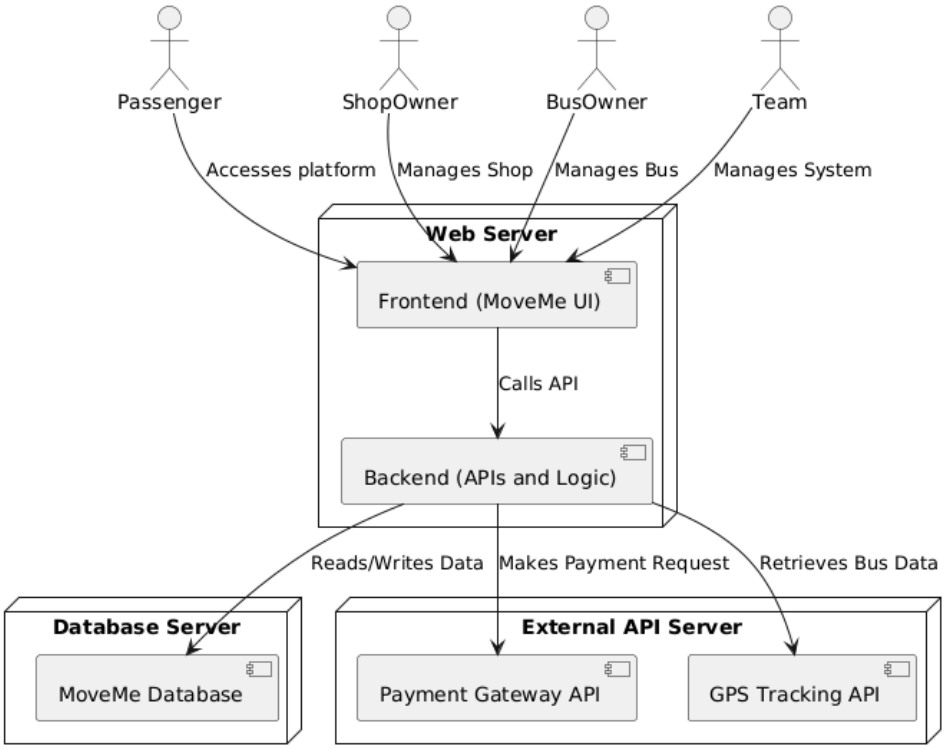


Fig :- Deployment Diagram (MoveMe)

Deployment Diagram of MoveMe Explanation :-

- **Actors :-** Passengers access the platform, ShopOwners manage shops, BusOwners manage buses, and the Team manages the system.
- **Web-server :-** The Frontend provides the user interface for "MoveMe," while the Backend handles APIs and calculation logic.
- **Database-server :-** Stores and manages data in the "MoveMe Database."
- **External API Server :-** The Payment Gateway API processes payments, and the GPS Tracking API tracks bus locations.

7.UI Design

7.1.Different Design Rules

3 Golden Rules of UI Design :-

1. **Place Users In Control :-** Place users in control by ensuring the design is modeless, flexible, interruptible, helpful, forgiving, navigable, accessible, facilitative, preference-oriented, and interactive.
2. **Reduce Users' Memory Load :-** Memory recall can be improved by relieving short-term memory load, relying on recognition, providing visual cues, being forgiving, considering frequency, promoting object-action syntax, using real-world metaphors, enabling user progressive disclosure, and organizing information.
3. **Make the interface consistent :-** Ensure interface consistency by maintaining continuity, consistency within and across products, keeping interaction results consistent, providing aesthetic appeal and integrity, and encouraging exploration.

Shneiderman's 8 Golden Rules of UI Design :-

1. **Strive For Consistency :-** Use the same design patterns, consistent terminology in prompts, menus, and help screens, and uniform colors, layout, capitalization, and fonts throughout the application's workflows.
2. **Seek Universal Usability :-** A good user-interface design ensures that all users, regardless of age or culture, can easily navigate the application and quickly achieve their goals, facilitated by navigation tips and shortcuts.
3. **Offer Informative Feedback :-** The design should keep users informed about the system's status, ensuring they understand the outcome of their actions and next steps, which builds trust in both the product and the brand.

4. **Design Dialog To Yield Closure :-** Ensure the user receives an appropriate message or reward at the end of a process, providing a sense of satisfaction upon task completion.
5. **Prevent Errors :-** Guide users smoothly through the entire process, preventing issues proactively, with good error messages that only appear when necessary.
6. **Permit Easy Reversal Of Actions :-** Relieve user anxiety by allowing actions to be undone, enabling exploration of unfamiliar options with the ability to cancel, go back, or undo/redo actions (e.g., Back, Cancel, Close, Undo, Redo).
7. **Let The Users Be In Control :-** Experienced users value having control over the interface and desire responsiveness to their actions; thus, provide customization options that allow users to use the application in their preferred way.
8. **Reduce Short-Term Memory Load :-** Minimize the amount of information users need to remember by using iconography, visual aids, and consistent item placement, and avoid long sequences of actions to enhance ease of use.

Norman’s 7 Fundamental Design Principles :-

1. **Discoverability :-** Identify the current system state and make clear which actions are possible at any given moment.
2. **Feedback :-** Inform the user that the system is processing their request.
3. **Conceptual Model :-** Offer a clear and concise explanation of how something works, making it easy for users to understand.
4. **Affordance :-** An object's perceived action and its properties should align to create a consistent and intuitive user experience.
5. **Signifiers :-** Clearly communicate where the action should take place, guiding the user to the correct location

6. **Mapping :-** Ensure a clear relationship between controls and their effects on the world, making it intuitive for users to understand the outcome of their actions.
7. **Constraints :-** Break tasks into small steps, providing users with enough information to complete one task at a time.

7.2.Our Project UI Design

Login Interfaces :-

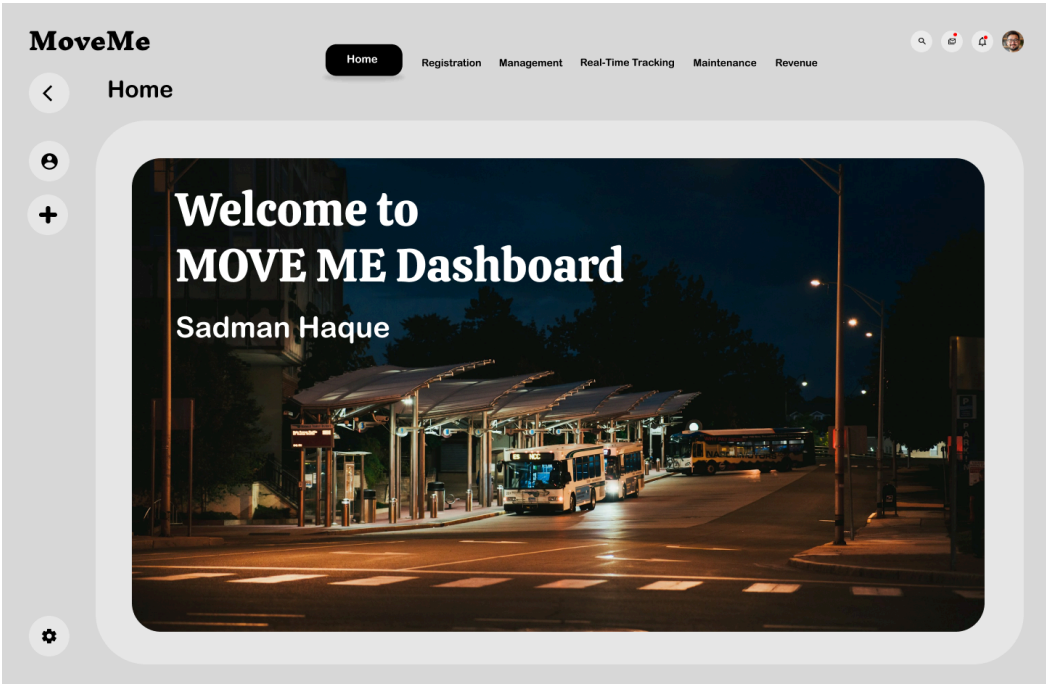
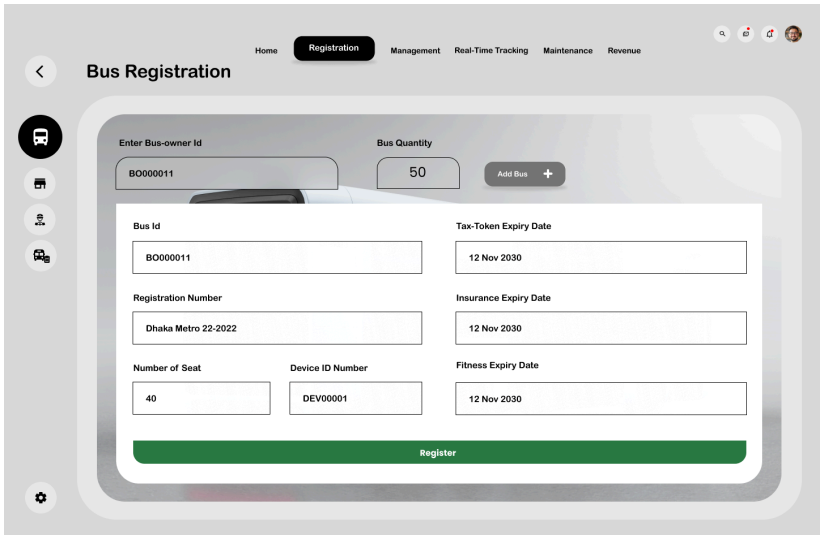
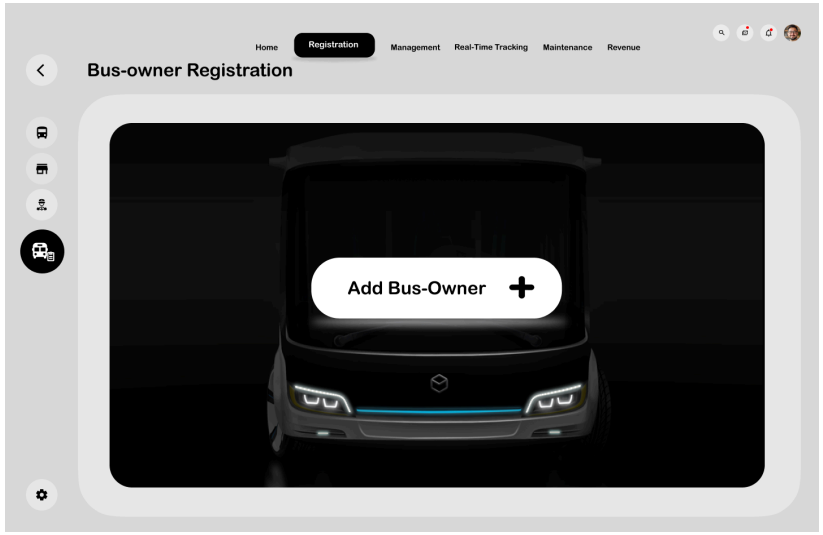
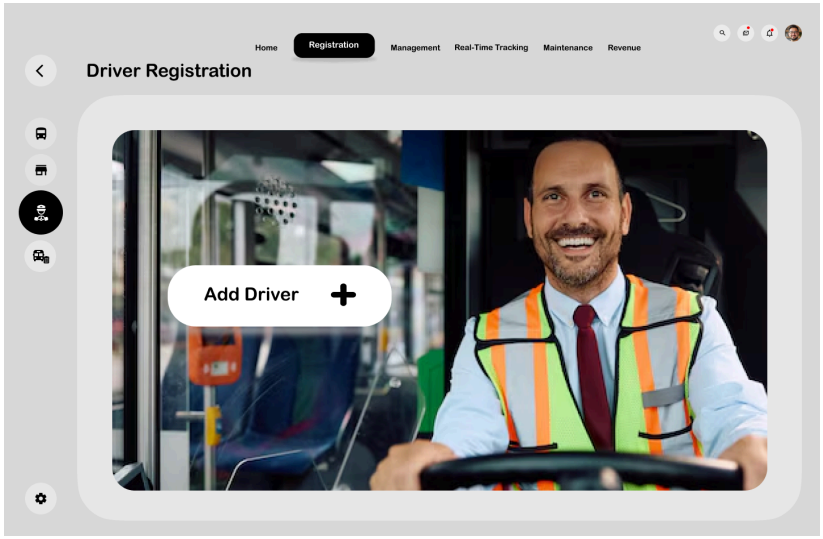


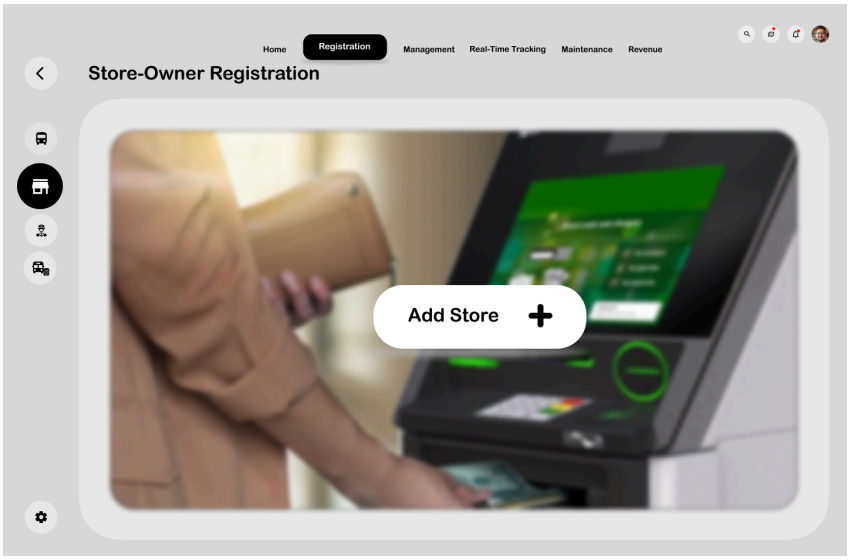
Fig :- Login Interfaces

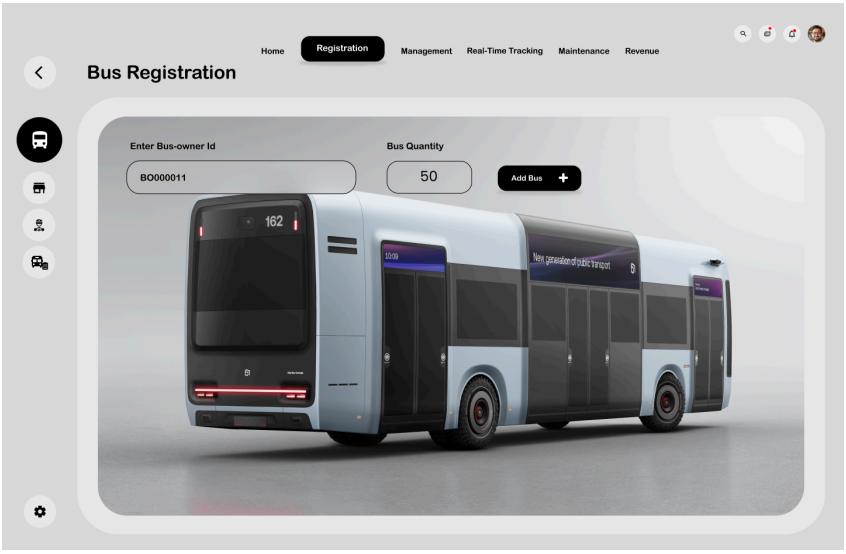
Registration Interfaces :-











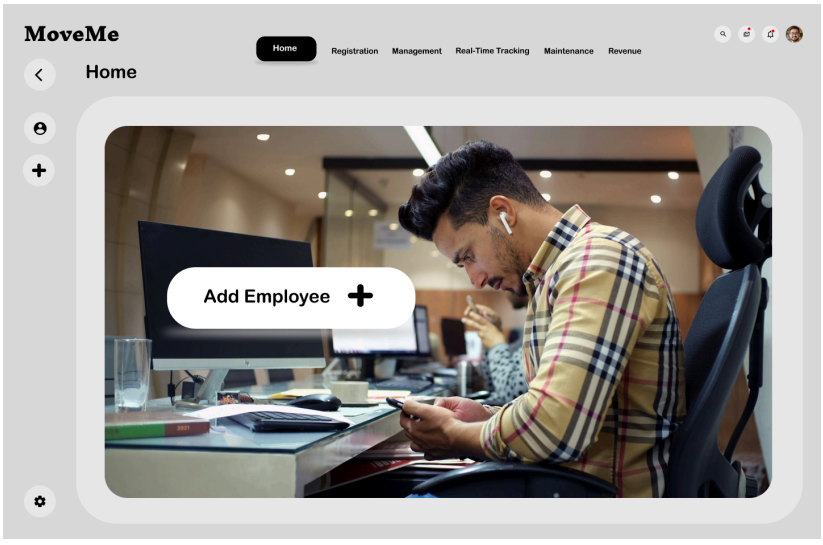


Fig :- Registration Interfaces

Management Interfaces :-

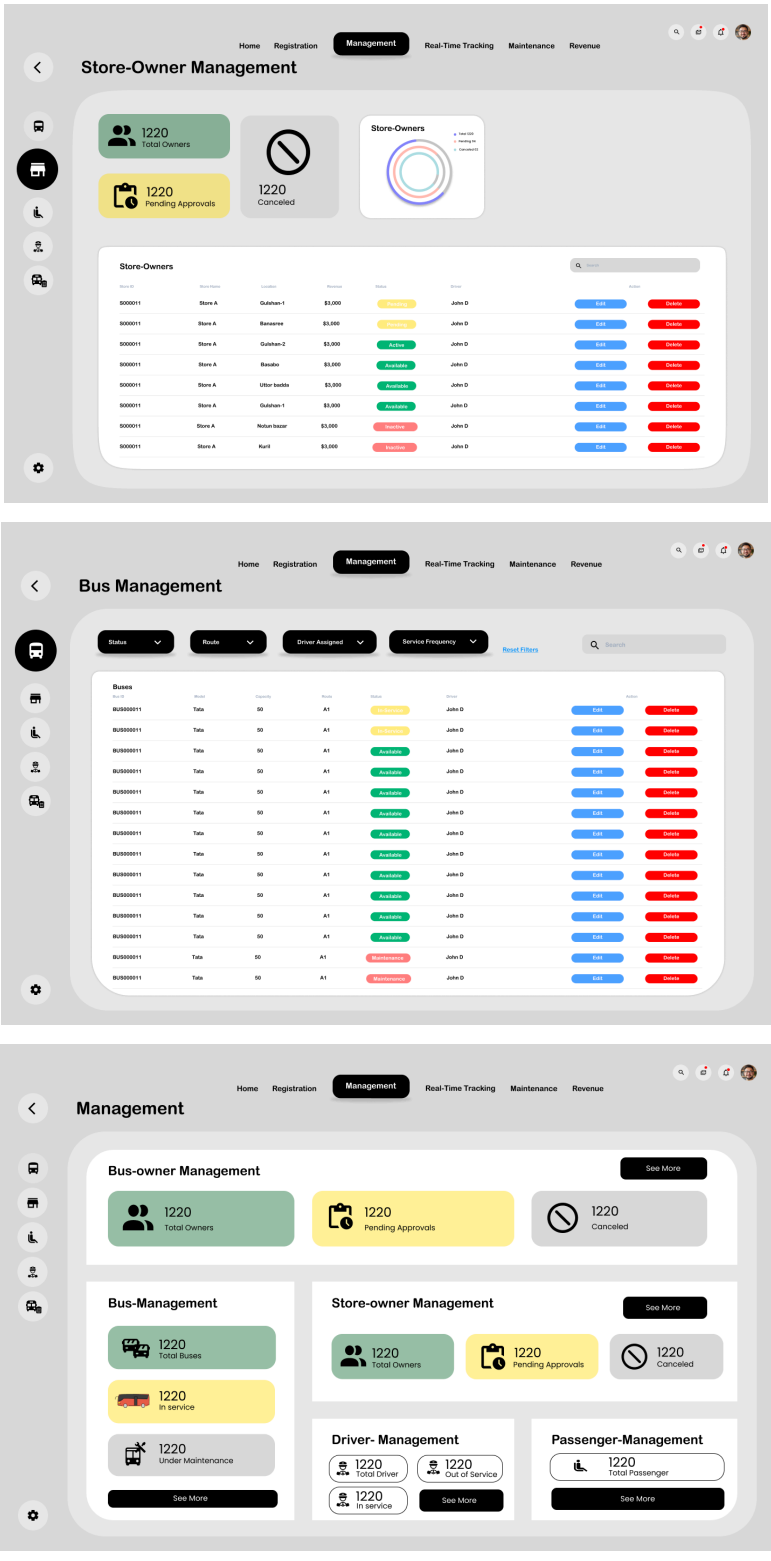
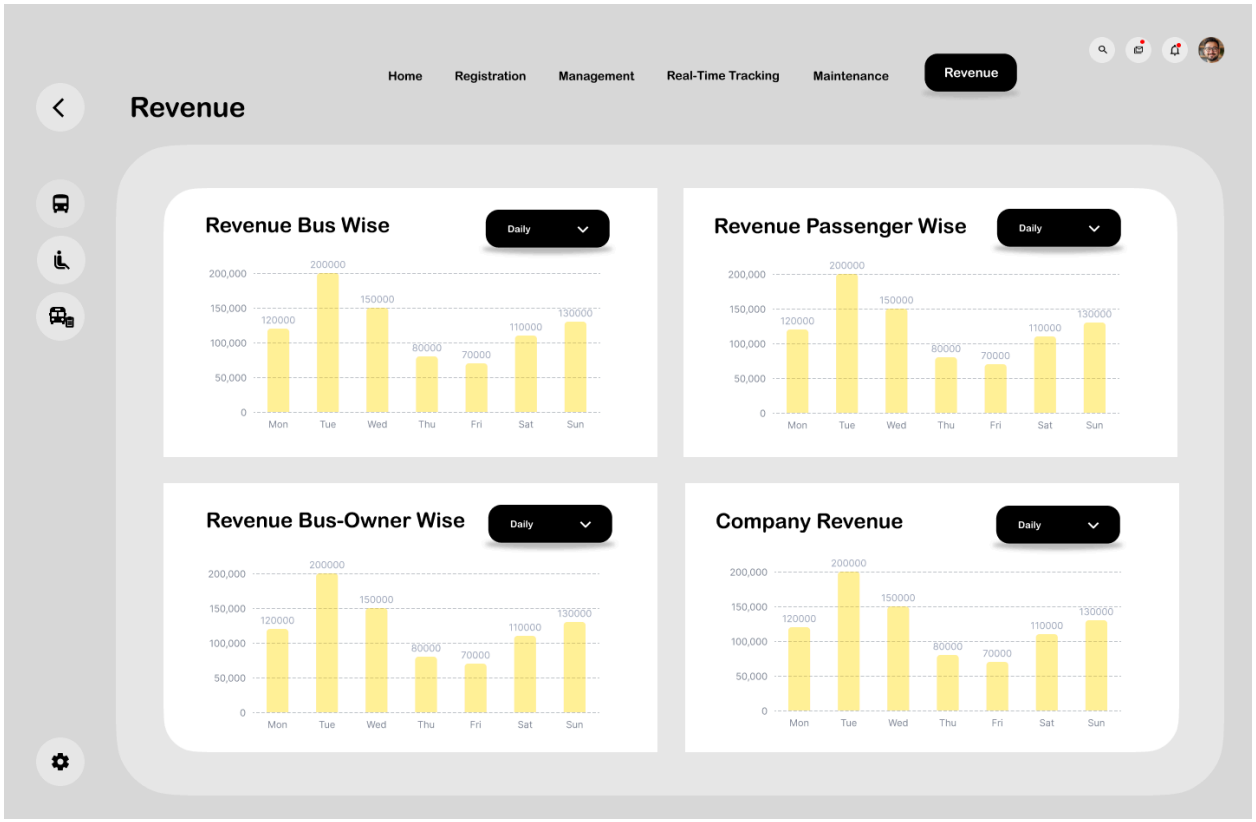


Fig :- Management Interfaces

Revenue Interfaces :-



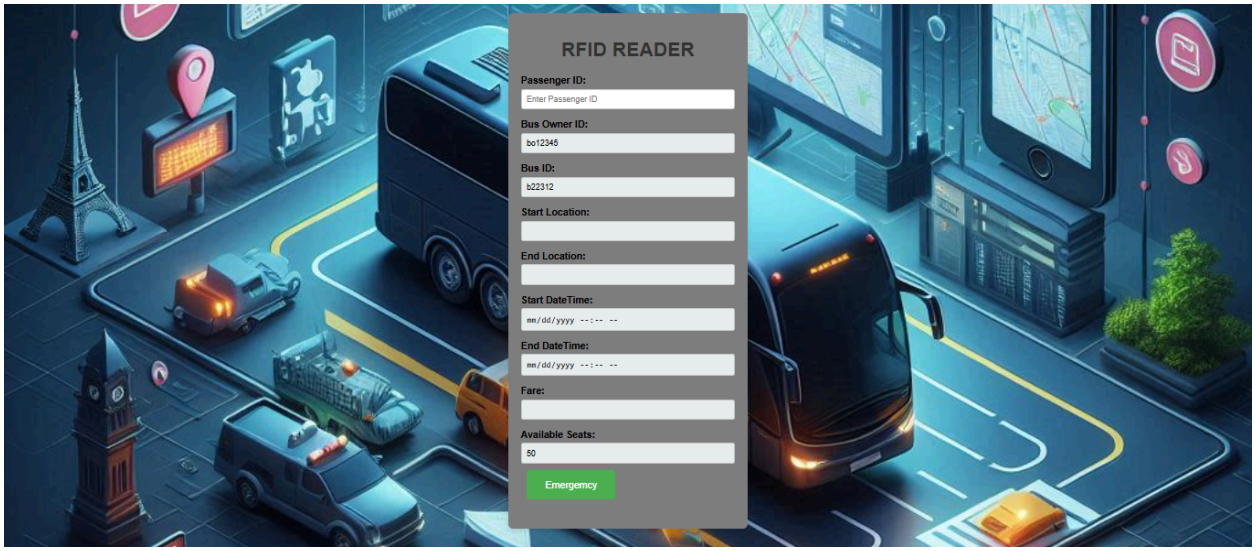
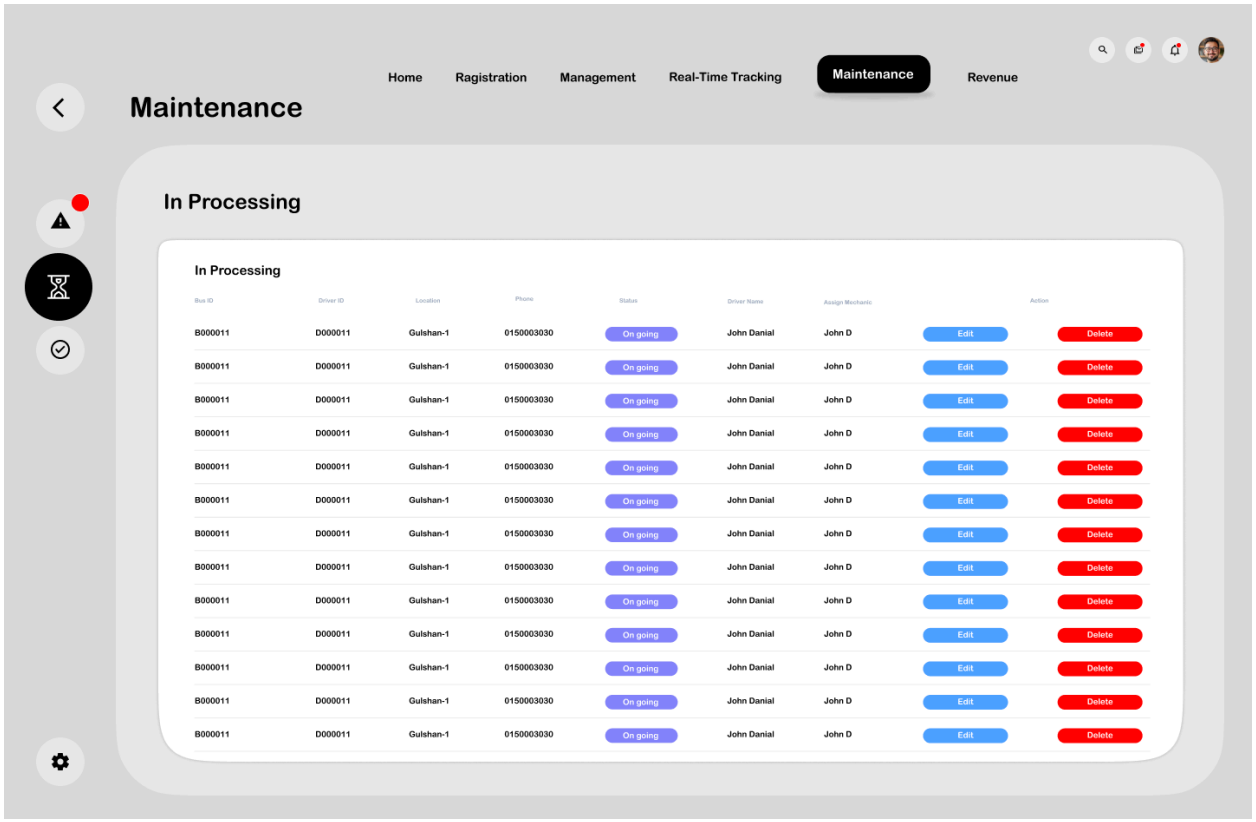


Fig :- Revenue Interfaces

Maintenance Interfaces :-



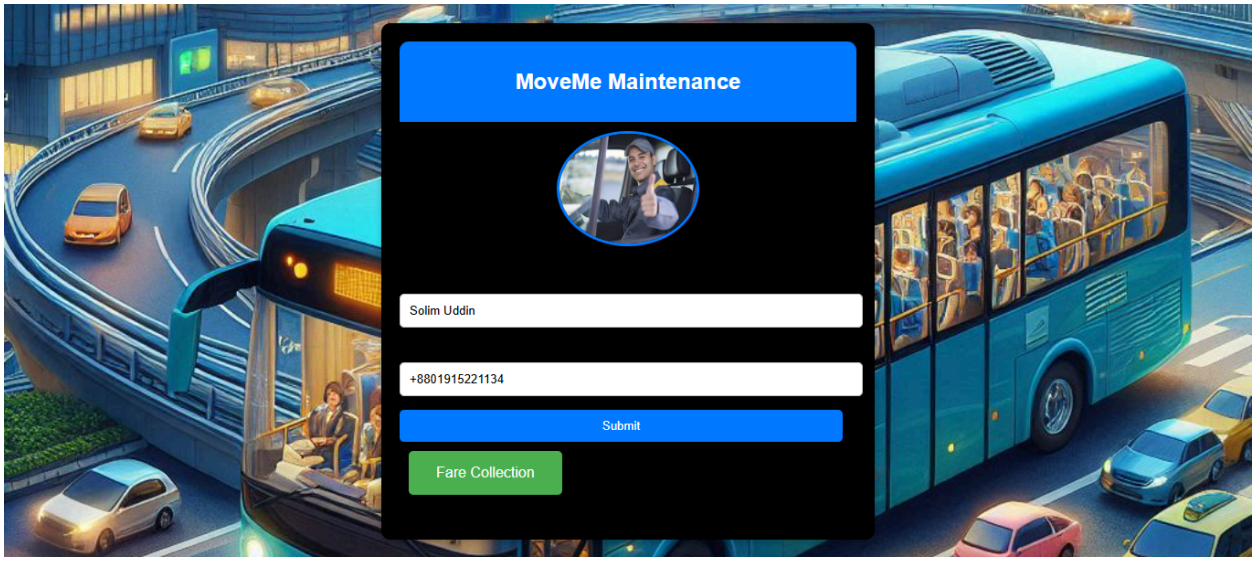


Fig :- Maintenance Interfaces

7.3.Testing

Here, we will explore the various software testing methodologies followed throughout the project.

Unit Testing :-

Focuses on verifying the functionality of individual units or modules of the software independently.

For example, in our project, the Busowner management module can be included in unit testing. After completing the registration of a Busowner, the management team can proceed to register all the buses into the system database.

Integration Testing :-

It involves combining modules incrementally and testing them as a group to ensure they work together seamlessly. This testing checks the compatibility and interaction between the integrated modules to verify that they function correctly as a cohesive unit.

For example, Revenue Management is a part of integration testing because it is divided into two components. One component is integrated with the bus system (RFID Reader), while the other is integrated with the Revenue team. On the bus, the RFID card is used for fare collection during entry and exit. The system is considered complete when the Revenue team can successfully track and manage the collected fare.

System Testing :-

It ensures that the software functions as expected in various environments, such as different operating systems and devices. It verifies the software's compatibility with changing environments and evaluates the overall performance and functionality of the entire system.

For example, our system is specifically designed to operate only on desktop environments. During system testing, we ensure that all functionalities work seamlessly across different desktop configurations, such as various operating systems (e.g., Windows, Linux) and hardware setups.

Performance Testing :-

It evaluates the speed, responsiveness, and overall effectiveness of a software application. It ensures that the software delivers the desired results within a specified timeframe, verifying its ability to handle workloads efficiently under normal and peak conditions.

For example, our system was responsive only on desktop platforms and not on others. It performed quickly because we optimized its operations to run locally on the desktop, ensuring faster processing and responsiveness.

8.Conclusion

"MoveMe" smart transport management system has been developed through a structured and thorough process, guided by the principles of software engineering and detailed documentation. From identifying the problem and defining clear goals to following the Software Development Life Cycle (SDLC), we gathered and analyzed information using various sources and tools, including surveys and benchmark studies, to ensure a comprehensive understanding of user needs. System analysis included survey analysis, benchmark gap analysis, and the fixation of functional and non-functional requirements, supported by feasibility and SWOT analyses. The design phase incorporated multiple diagrams, such as context, data-flow, use-case, activity, and entity-relationship diagrams, along with UI designs adhering to key design principles. This systematic approach ensured that "MoveMe" effectively addresses the identified transportation challenges, meeting both user and system requirements with an innovative and practical solution.

9.References

Theoretical Concepts :-

- https://aws.amazon.com/what-is/sdlc/#:~:text=The%20software%20development%20lifecycle%20(SDLC,expectations%20during%20production%20and%20beyond.
- https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/
- https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_planning.htm
- https://www.onlinemanipal.com/blogs/what-is-benchmarking-analysis
- https://asana.com/resources/swot-analysis
- https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/
- https://www.geeksforgeeks.org/use-case-diagram/
- https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/
- https://en.wikipedia.org/wiki/Class_diagram#:~:text=The%20class%20diagram%20is%20the,be%20used%20for%20data%20modeling.
- https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagrams/
- https://www.geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/
- https://www.lucidchart.com/pages/er-diagrams
- https://medium.com/@sachinrekhi/don-normans-principles-of-interaction-design-51025a2c0f33
- https://capian.co/shneiderman-eight-golden-rules-interface-design

Benchmark Products :-

- https://www.shohoz.com/air-tickets?utm_source=GoogleSearch&utm_medium=SearchAds&utm_campaign=GA_WebTraffic&utm_content=Local&utm_term=MultipleDestination&gad_source=1&gclid=CjwKCAiAkc28BhB0EiwAM001TdPXFIbTVWNenmWSeeARXKHq3G4gbKRXuYPBbUZMn4JVmjUpCQ1MxRoCaGkQAvD_BwE
- https://www.jatri.co/
- https://bdtickets.com/?gad_source=1&gclid=CjwKCAiAkc28BhB0EiwAM001TXGsn5D26-yciGpZky5ikDKp6ee6sYTwXwGyJu8gQaZz3x7q3ACeUBoCLDsQAvD_BwE
- https://busbd.com.bd/
- https://www.compass-travel.co.uk/

Research Papers :-

- https://www.researchgate.net/publication/369980801_Smart_Transportation_An_Overview_of_Technologies_and_Applications
- https://www.sciencedirect.com/topics/computer-science/smart-transportation-system
- https://www.researchgate.net/publication/337928590_Smart_Transportation_Systems
- https://ieeexplore.ieee.org/document/8358540
- https://www.semanticscholar.org/paper/Smart-Transportation-System-Odtalla/c789c4972b2e492980d811abf3320ad489e58296
- https://www.nature.com/articles/s41598-023-50906-7
- https://www.researchgate.net/publication/367439907_IoT-Based_Public_Transport_Management_System
- https://www.mdpi.com/1424-8220/23/8/3880