JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY

BSc. COMPUTER SCIENCE

DESIGN AND IMPLEMENTATION OF COMPUTER APPLICATIONS

MATATU-FLEET MANAGER

GROUP MEMBERS

1. MACHARIA MAURICE - SCT211-0010/2022
2. JOSEPH NGURE - SCT211-0008/2022
3. PEACHES NJENGA -SCT211-0004/2022
4. HILDA MWANGI -SCT211-0026/2022
5. THEURI BONFACE KARUE -SCT211-0573/2022
6. KING’ORI FLORENCE WANGECHI -SCT211-0063/2022

MATATU FLEET MANAGING SYSTEM

INTRODUCTION

In response to the complexities of managing matatu fleets in urban areas, we propose the development of a dedicated desktop application for managing the operation of matatu fleet managers for Bus or Matatu Saccos such as the popular ‘Super-Metro’ and ‘Lopha travelers’ for buses.

The application is specifically for use by a manager who will be positioned at the official matatu stage which is either where the matatu begins or ends its commute.

Our proposal outlines the key features, technical specifications and development timeline associated with this initiative. With the matatu industry's important role in urban transportation, our proposed application is meant to modernize operations and optimize fleet management processes.

This solution aims to enhance efficiency, safety, and profitability by centralizing operations and providing essential tools for scheduling, route planning, renting out vehicles management of staff and vehicles.

POTENTIAL CLIENT USERS

Potential users include

1. Matatu fleet owners
2. Matatu fleet managers

FUNCTIONAL REQUIREMENTS

1. Vehicle Planning: The system should allow for efficient allocation of vehicles based on availability..

2. Expense Management: The system should provide tools to track and manage expenses related to vehicle operations, such as fuel costs, maintenance expenses, and driver and conductor wages.
4. Trip Scheduling: The system should enable the creation and management of trip schedules, including departure times, destinations and arrival times.
5. Route Planning: The system should plan and support the optimization of routes taken by the buses.
6. Renting of Vehicles: The system should allow for the renting out of vehicle to customers per hour, for their personal use and record the duration of the renting out of the vehicle

NON-FUNCTIONAL REQUIREMENTS

1. Performance: The system should be able to handle a large volume of vehicles, trips, and data, providing fast response times and minimizing delays.
2. Security: The system should ensure the confidentiality and integrity of sensitive data, such as financial information and maintenance records and provide for user authentication.
3. Reliability: The system should be highly reliable, minimizing errors, crashes, and downtime to ensure uninterrupted operations.
4. Scalability: The system should be able to scale up or down to accommodate changes in the size of the fleet and the number of trips.
5. Usability: The system should have an intuitive and user-friendly interface, allowing users to easily navigate and perform tasks without extensive training.

SOFTWARE REQUIREMENT

For End-Users:

1.Operating System:

- Windows 10 or later

2.Database:

- Microsoft SQL Server for storing and managing data related to buses, income, maintenance costs, drivers, and salaries.

3. Desktop Application:

- Developed using Visual Basic and Visual Studio.

- Windows Forms application for a user-friendly interface.

4. Authentication:

- User login functionality.

-A password and a username.

For Developers:

1.Integrated Development Environment (IDE):

- Visual Studio for VB.NET development.

2. Version Control:

- Git for version control, hosted on platforms like GitHub

3.Database Development:

-MYSQL workbench for database schema design and management.

4. Dependency Management:

- NuGet for managing third-party libraries and dependencies.

5. Testing:

- Unit testing framework for VB.NET (e.g., MSTest or NUnit).

6. Documentation:

- Use tools like Doxygen or Sandcastle for code documentation.

7.Containerization:

-Use Docker to package the application and its dependencies into containers, providing a consistent environment across different development and deployment stages.

HARDWARE REQUIREMENTS

For End-Users:

1.Processor:

- Dual-core processor or higher.

2.Memory (RAM):

- 4 GB or more.

3. Storage:

- SSD or HDD with sufficient space for the application and database.

4. Display:

- Minimum resolution of 1280x800.

For Developers:

1. Processor:

- Quad-core processor or higher for faster development.

2. Memory (RAM):

- 8 GB or more for smooth development experience.

3. Storage:

- SSD for faster build and compilation times.

4. Development Environment:

- High-resolution monitor(s) for better code readability and multitasking.

AN OVERVIEW OF THE SYSTEM ARCHITECTURE AND KEY COMPONENTS.

Backend Architecture

The relational database management system Microsoft SQL will be used to store various data related to vehicles, routes, schedules, driver, and conductor information.

The server-side components will be developed using the .NET framework. The components handle the core logic and functionality of the system, that is, modules for vehicle registration, route management and vehicle renting. These components communicate with the database to retrieve and update relevant data.

Frontend Architecture

The client-side components, which are responsible for providing a user-friendly interface for the administrators, will be developed using Visual Studio's .NET framework. The administrators will be able to access a mobile application-based dashboard to manage vehicles, routes, and monitor system performance.

Security and Authentication

The system will leverage visual studio’s authentication frameworks to ensure that only authorized individuals have access to the system.

Scalability and Performance:

The system is designed to handle a large number of concurrent users and scale horizontally as the system grows.

SCHEDULE

Activity	**Estimated start date**	**Estimated date of completion**	**Delivered Product**
Feasibility study	15/2/2024	17/02/2024	Feasibility report
Design	17/2/2024	30/02/2024	Interface and Database design
Programming	21/02/2024	13/03/2022	Source code
Testing	10/03/2024	16/03/2024	Test-reports

STAGES

1. **Feasibility study**

a) Schedule feasibility

-The development of teh application will take 4 weeks for a first working prototype to be delivered

b) Technical feasibility

- The developing team would need proficiency in the use of Visual Studio as an IDE for development, the .NET framework and vb.net programming language.
- The system will be able to be operated by anyone with basic computer literacy

c) Operational Feasibility

- The system is expected to be able to be easily integrated into the normal operation of the matatu fleets and easy to use, therefore received positively by the user.

2. Design

i) ***UI/UX Design***

The UI/UX of our system focuses on providing a user-friendly and intuitive interface for administrators. It aims to enhance the overall user experience by simplifying tasks and improving accessibility.

Design Principles:

- Simplicity: The UI design follows a minimalist approach, ensuring that the interface is clean and uncluttered, making it easy for users to navigate and perform tasks.

- Consistency: A consistent design language, visual basic, is maintained throughout the system, ensuring that users can easily understand and interact with different components.

- Responsiveness: The interface is designed to be responsive across various devices and screen sizes, allowing users to access the system from desktops, tablets, or mobile devices.

User Roles and Workflows:

- Administrators: The admin interface provides a comprehensive dashboard for managing vehicles, routes, schedules, and user accounts. It allows administrators to add, edit, and delete data.

Visual Design:

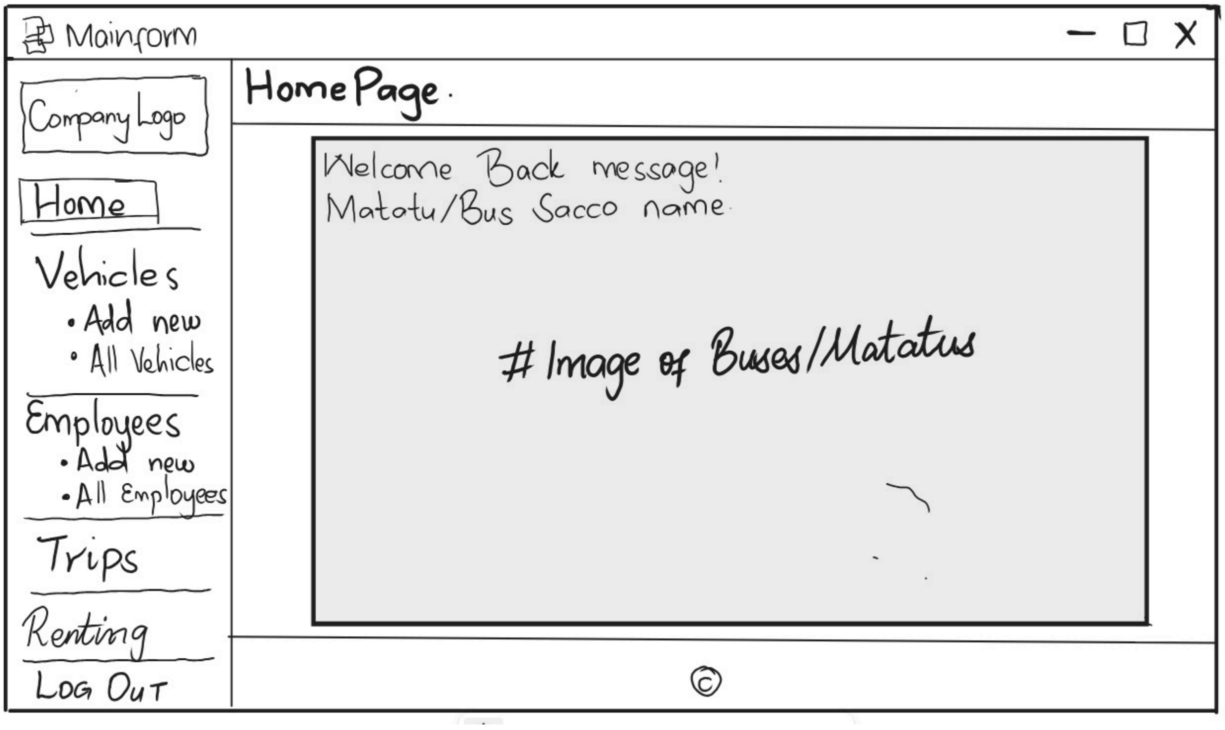
- **Color Scheme**: A visually appealing color palette will be used, with contrasting colors to highlight important elements and ensure readability. The system will use a color scheme that combines gray with white and black. These colors create a clean and minimalist look, which will enhance the overall user experience. Additionally, the use of these neutral colors will help convey a sense of professionalism and reliability. It also provides high contrast, making it easier for users to read and interact with the interface. Overall, this color scheme will contribute to a visually appealing and user-friendly design for the system.

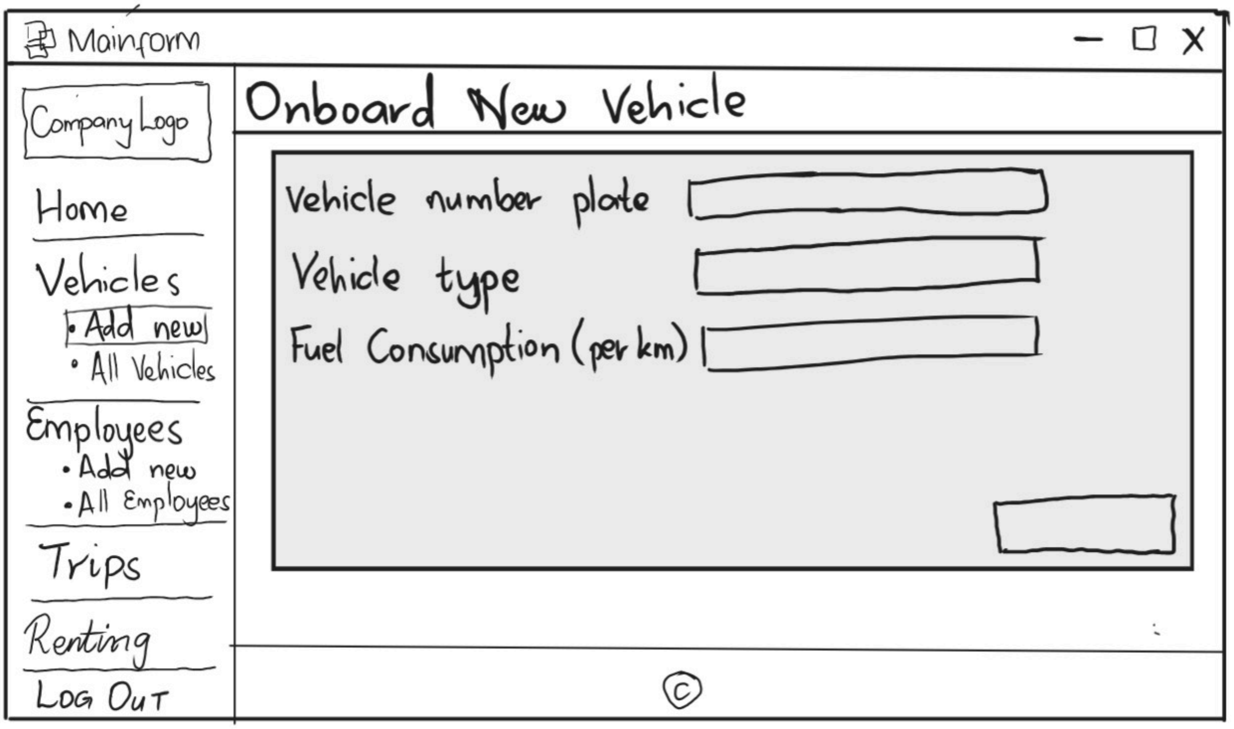
- **Typography**: Clear and legible fonts have been chosen to enhance readability across different devices and screen sizes. A modern and clean sans-serif font like Open Sans will be used for headings and important information; this typeface has a balanced and versatile design, making it suitable for digital interfaces. For body text, the legible serif font to be used will be Georgia. Georgia has a timeless elegance and excellent readability, making it ideal for conveying information in longer text blocks.

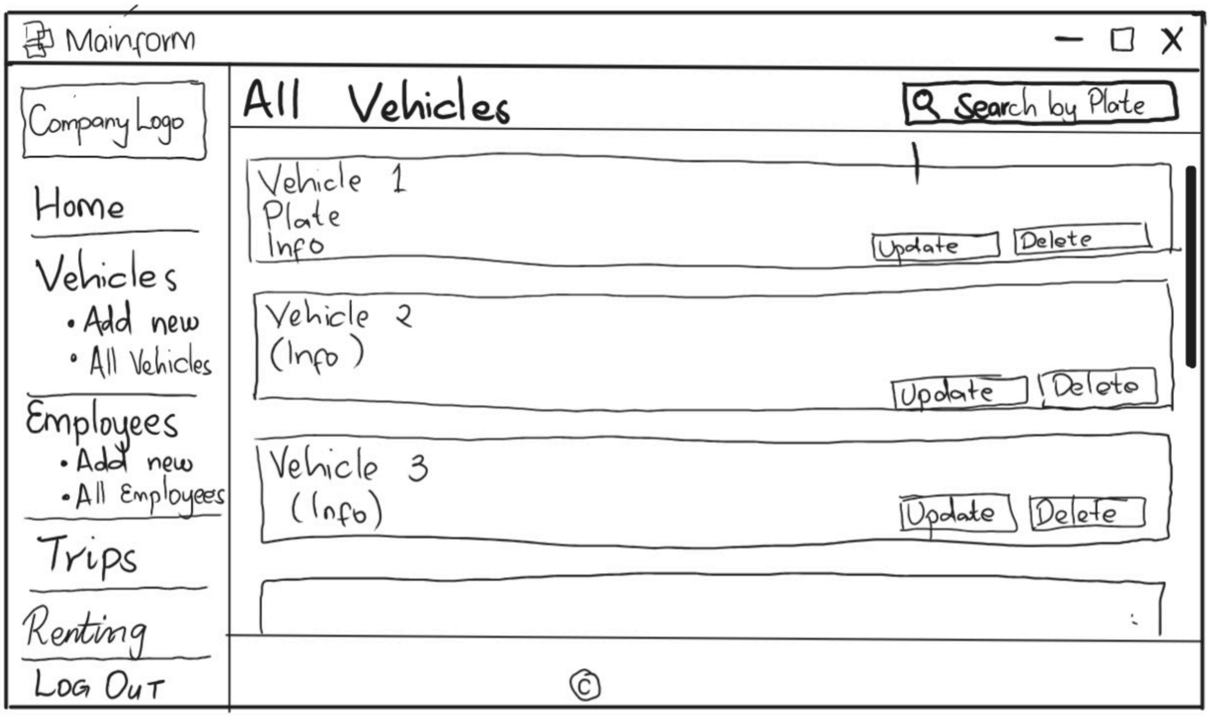
- **Iconography**: Intuitive and recognizable icons will be used to represent different actions and functionalities, aiding users in quickly understanding and interacting with the system.

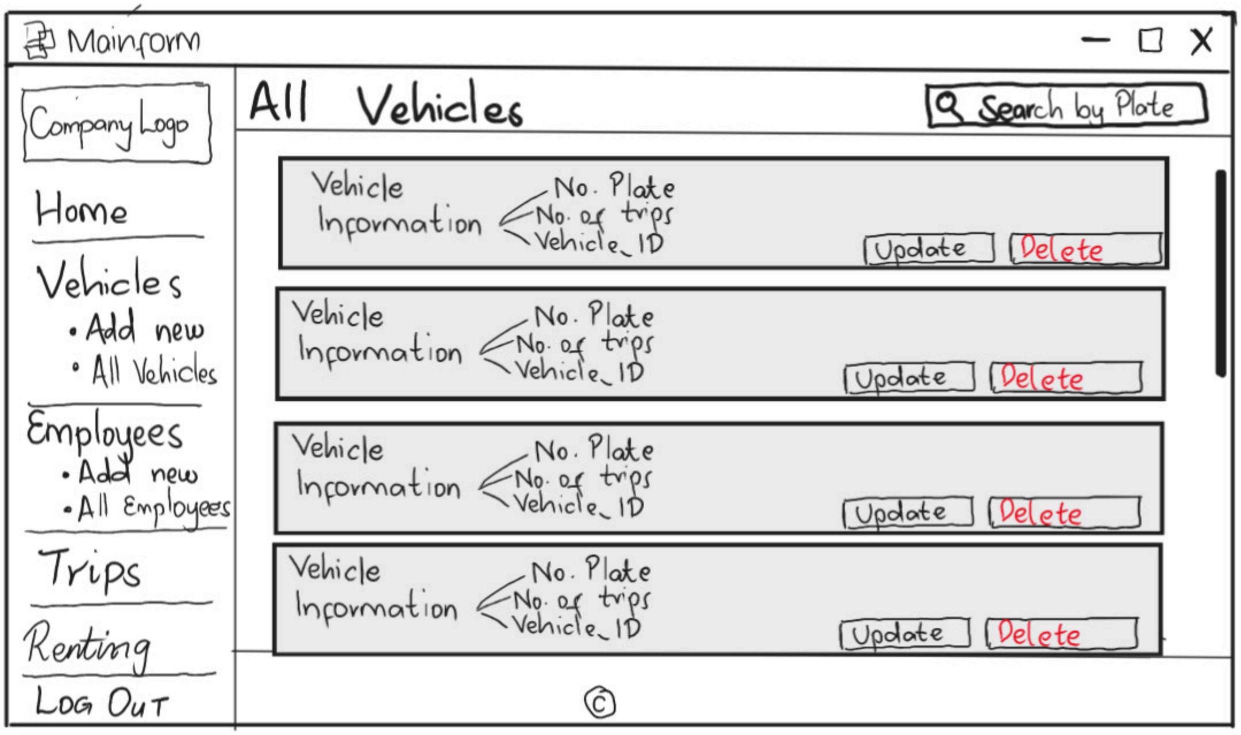
Usability Testing:

The UI/UX design will undergo rigorous usability testing to gather feedback from potential users. This will help identify any usability issues, improve navigation, and enhance the overall user experience.

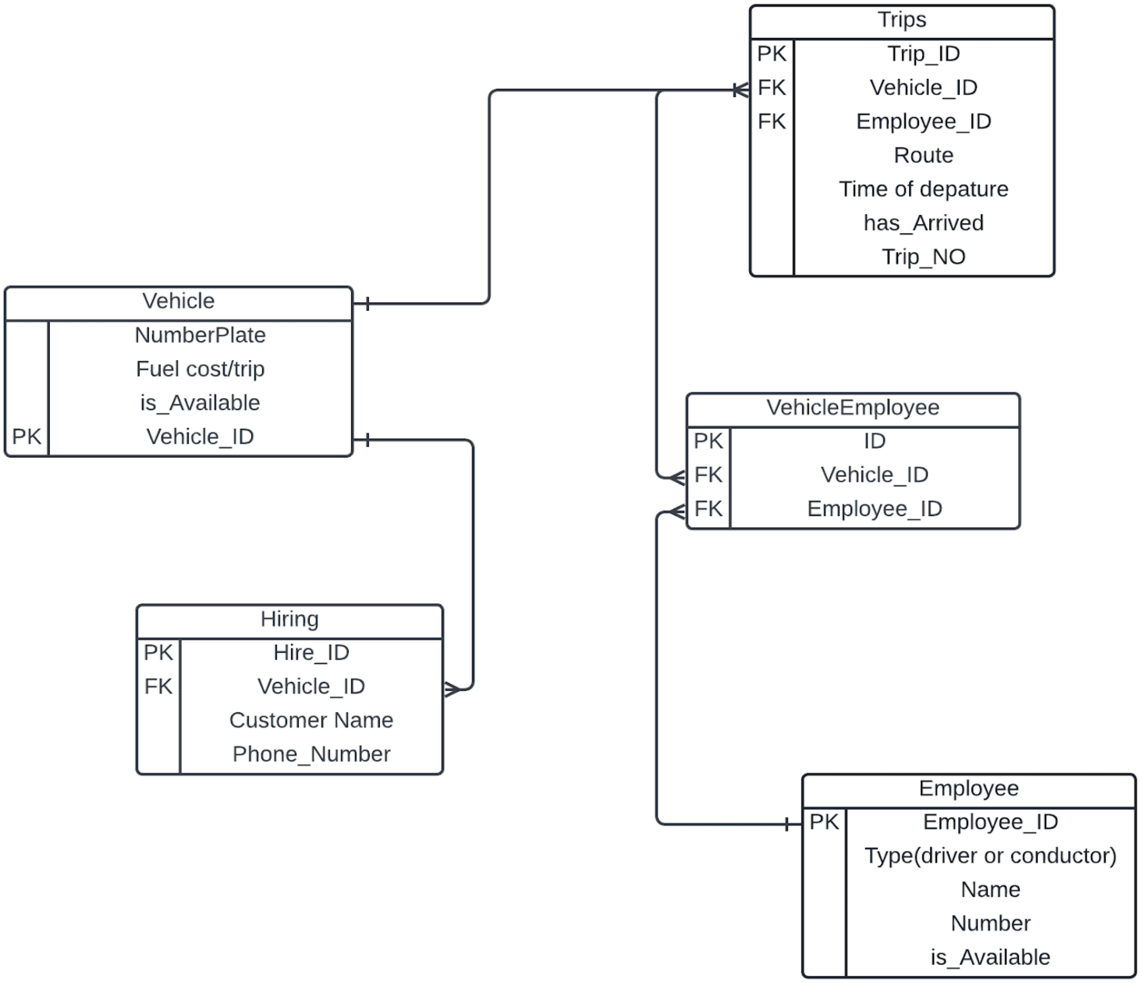



Mainform
Company Logo
Home
Vehicles
• Add new
• All Vehicles
Employees
• Add new
• All Employees
Trips
Renting
LOG OUT
Onboard New Vehicle
Vehicle number plate
Vehicle type
Fuel Consumption (per km)
©


Mainform
Company Logo
Home
Vehicles
• Add new
• All Vehicles
Employees
• Add new
• All Employees
Trips
Renting
LOG OUT
All Vehicles
Search by Plate
Vehicle 1
Plate
Info
Update
Delete
Vehicle 2
(Info)
Update
Delete
Vehicle 3
(Info)
Update
Delete
©


Mainform
Company Logo
Home
Vehicles
• Add new
• All Vehicles
Employees
• Add new
• All Employees
Trips
Renting
LOG OUT
All Vehicles
Search by Plote
Vehicle Information
No. Plate
No. of trips
Vehicle_ID
Update
Delete
Vehicle Information
No. Plate
No. of trips
Vehicle_ID
Update
Delete
Vehicle Information
No. Plate
No. of trips
Vehicle_ID
Update
Delete
Vehicle Information
No. Plate
No. of trips
Vehicle_ID
Update
Delete
©

ii) ***Database Design***



	Trips
PK	Trip_ID
FK	Vehicle_ID
FK	Employee_ID
	Route
	Time of depature
	has_Arrived
	Trip_NO

	Vehicle
	NumberPlate
	Fuel cost/trip
	is_Available
PK	Vehicle_ID

	VehicleEmployee
PK	ID
FK	Vehicle_ID
FK	Employee_ID

	Hiring
PK	Hire_ID
FK	Vehicle_ID
	Customer Name
	Phone_Number

	Employee
PK	Employee_ID
	Type(driver or conductor)
	Name
	Number
	is_Available

- 1. Tables:

  - Create tables for buses/matatus , income, maintenance costs, drivers, and salaries, conductors, schedules.

  - Define appropriate relationships between tables (foreign keys).

  - Use normalization techniques to avoid redundancy.

  2. Stored Procedures:

  - Develop stored procedures for common database operations to enhance performance.

  (**CRUD**) operations.

  3. Indexes:

  - Implement indexes on columns frequently used in queries to optimize database performance.

  (Primary keys)

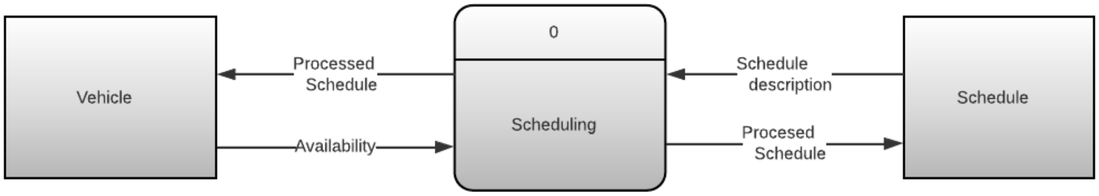
  4. Backup and Recovery:

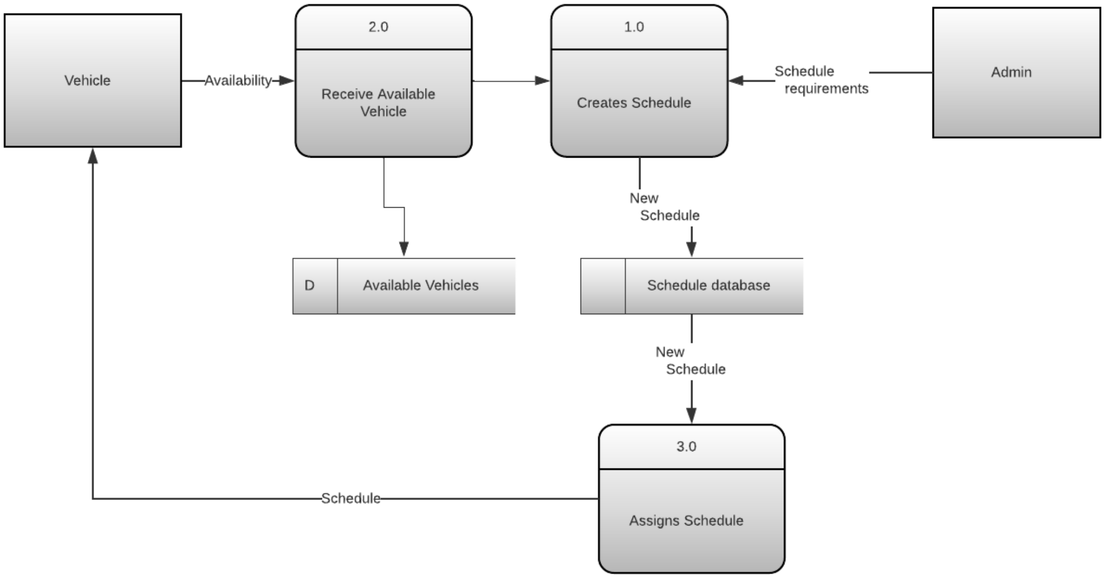
  - Establish regular database backup and recovery procedures

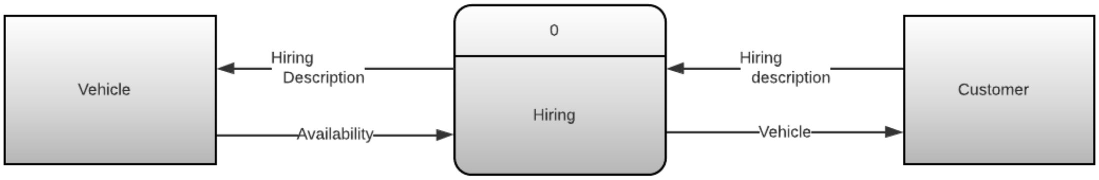
  Below is an Entity Relationship diagram for the proposed database design.

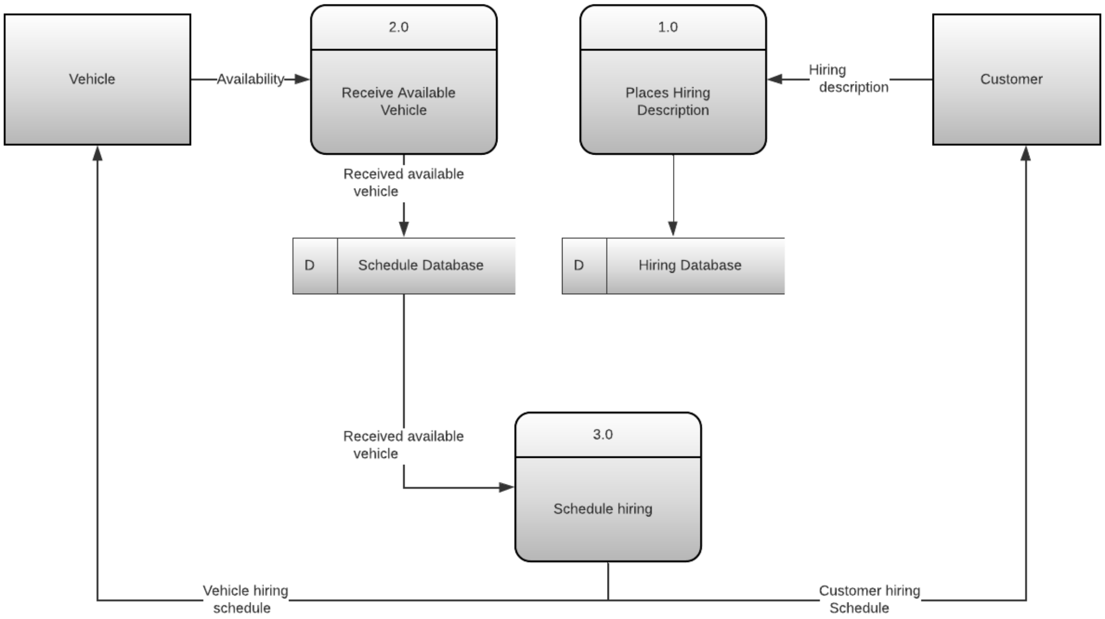
iii) Program Design

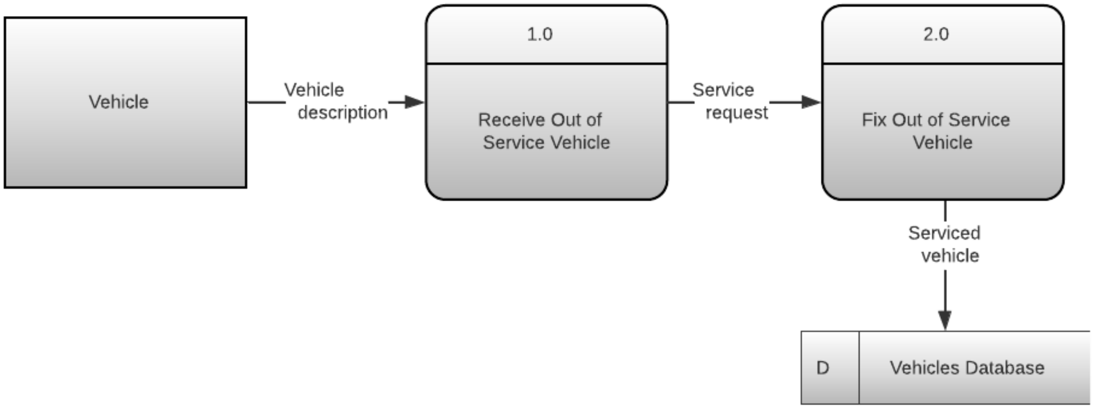
- This includes architecture and logic behind the desktop application's functionality. It will involve determining what programming languages, framework and libraries will be used for development.
- Below is a proposed data-flow diagram, with a context diagram and a level 0 diagram for some proposed processes.


Vehicle
Processed Schedule
Availability
0
Scheduling
Schedule description
Procesed Schedule
Schedule


Vehicle
Availability
2.0
Receive Available Vehicle
1.0
Creates Schedule
Schedule requirements
Admin
D
Available Vehicles
New Schedule
Schedule database
New Schedule
3.0
Assigns Schedule
Schedule

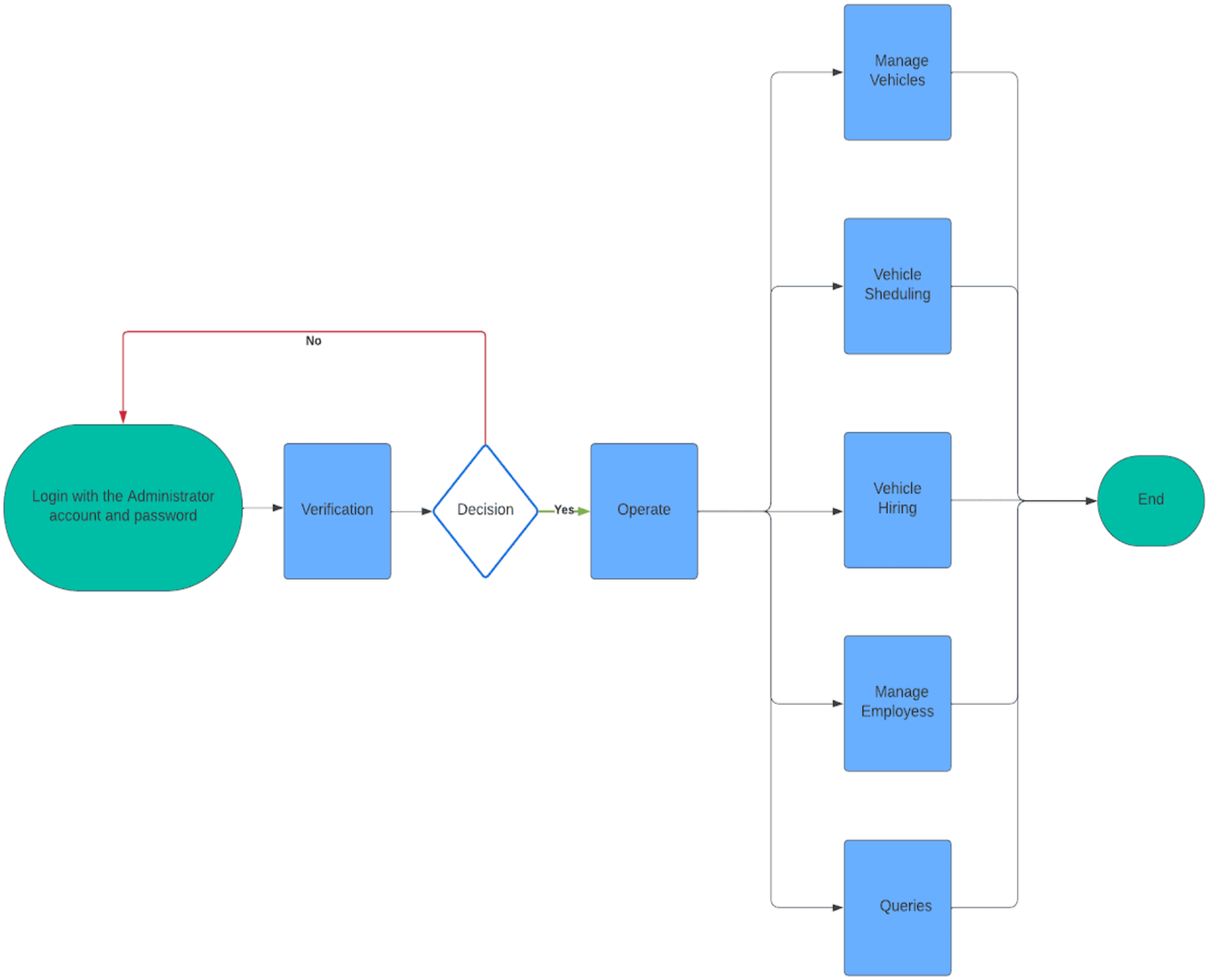

Vehicle
Hiring Description
Availability
0
Hiring
Hiring description
Vehicle
Customer





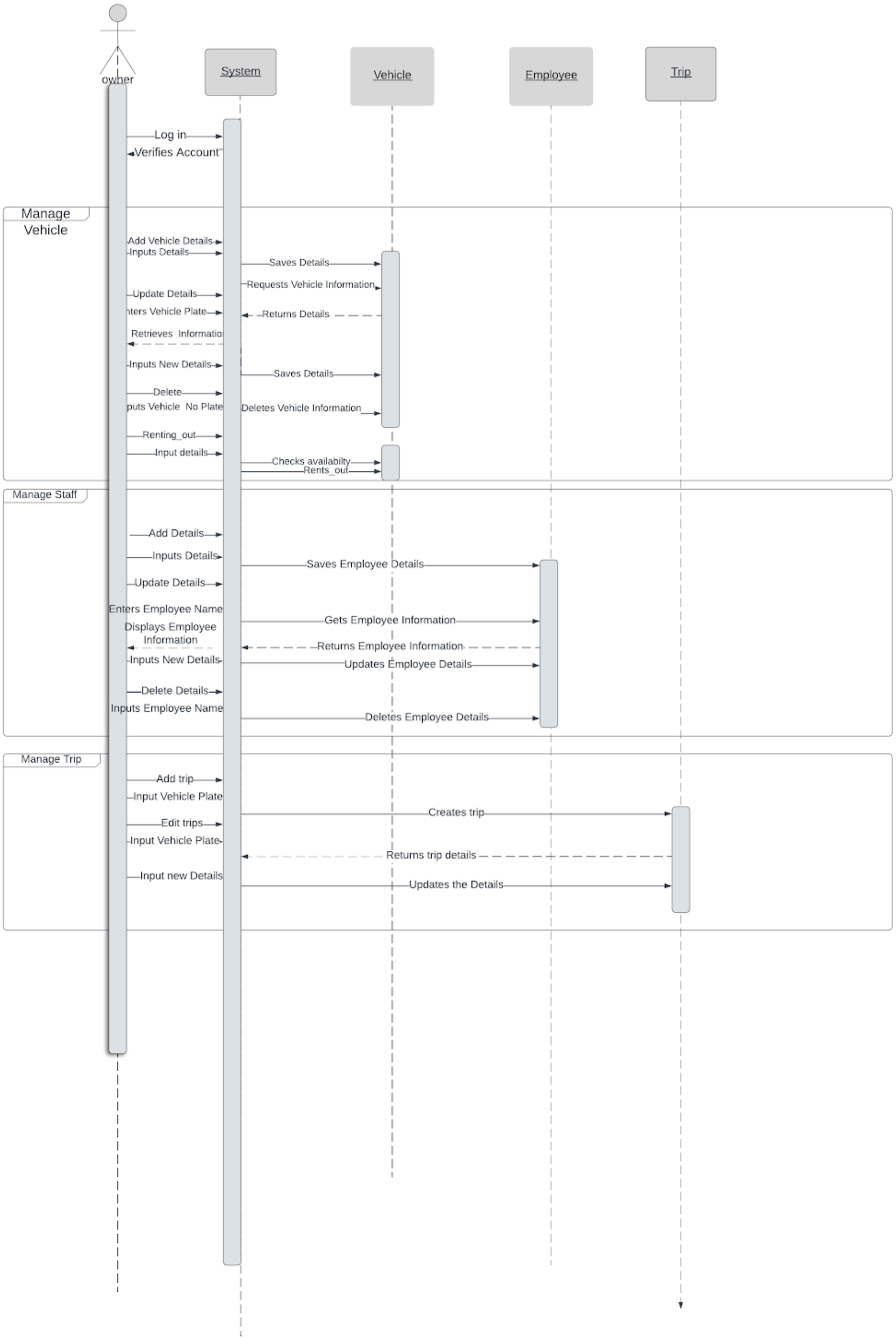
iv)Flow chart

- This will be a visual representation of the step-by-step process/ flow of actions within the application that will help serve as a roadmap as the application is being developed. This is to ensure that all user pathways are optimized for user-experience.



v) Sequence Diagram

- Illustrating the interactions between different components of the application over time, providing insight into the order of actions and data flow within the system.


owner
System
Vehicle
Employee
Trip
Log in
Verifies Account
Manage Vehicle
Add Vehicle Details
Inputs Details
Saves Details
Requests Vehicle Information
Update Details
nters Vehicle Plate
Returns Details
Retrieves Informatio
Inputs New Details
Saves Details
Delete
puts Vehicle No Plate
Deletes Vehicle Information
Renting_out
Input details
Checks availabilty
Rents_out
Manage Staff
Add Details
Inputs Details
Saves Employee Details
Update Details
Enters Employee Name
Gets Employee Information
Displays Employee Information
Returns Employee Information
Inputs New Details
Updates Employee Details
Delete Details
Inputs Employee Name
Deletes Employee Details
Manage Trip
Add trip
Input Vehicle Plate
Creates trip
Edit trips
Input Vehicle Plate
Returns trip details
Input new Details
Updates the Details

3. **Programming**

   The actual coding of the application.

4. **Testing**

   This will be done after completion of the programming stage to test whether the application works as desired to produce a working prototype.