



**AMAL JYOTHI**  
**COLLEGE OF ENGINEERING**  
( A U T O N O M O U S )

# INSTHELPER

20INMCA509 - Mini Project 2

Scrum Master

**Binumon Joseph**

Assistant Professor

Department of Computer Applications

**DEPARTMENT OF  
COMPUTER APPLICATIONS**



**Ajul K Jose**

AJC20MCA-I004

INMCA2020-25 S9

<https://github.com/ajulkjose246/InstHelper>

mail.ajulkjose@gmail.com

# ABSTRACT

---

InstHelper is an advanced platform designed to revolutionize college campus operations through automation and innovative technology integration. The platform includes a Flutter Android application for College Vehicle Management and a web application for the Canteen Management System, providing users with seamless access to essential campus services.

The College Vehicle Management module is a Flutter Android app exclusively accessible to college staff. It allows drivers to efficiently manage vehicle details, such as insurance information, test dates, maintenance schedules, and trip details. Administrators can monitor this data in real-time through a web interface, ensuring the efficient upkeep and operation of college-owned vehicles and tracking trip information for better resource management.

The Canteen Management System is a web application that replaces the existing Loyverse system. This new system integrates directly with the college website (<https://aesajce.in/>), allowing users to book canteen products online. This seamless integration enhances user convenience by centralizing access to both college information and canteen services.

## MODULES:

1. Authentication: Manages user registration, login, and role-based access control.
2. College Vehicle Management: A Flutter Android app accessible only to college staff, allowing drivers to manage vehicle details like insurance, test dates, maintenance schedules, and trip details; administrators can oversee and monitor vehicle and trip information.
3. Canteen Management System: A web application replacing Loyverse, integrated with the college website for easy online booking of canteen products.
4. Admin Dashboard: Provides graphical data representation and real-time updates for administrators.

## USERS:

1. College Vehicle Management System Users:
  - a) Drivers: Manage vehicle details, log trips, and receive notifications for maintenance and insurance.
  - b) Administration: Monitor vehicle information, approve trip requests, generate reports, and communicate with drivers.
  - c) Admin: Oversee all system aspects, manage user accounts, monitor system health, and handle data backup and recovery.

2. Canteen Management System Users:

- a) Students: Browse menu, place and track orders, view order history, and manage payments.
- b) Faculty: Browse menu, place and track orders, view order history, and manage payments.
- c) Canteen Staff: Manage menu items, process orders, track inventory, generate reports, and manage promotions.
- d) Administrators: Monitor system performance, manage users, access financial reports, review audit logs, and handle system integration.
- e) Admin: Oversee all system functionalities, configure system settings, manage backups and recovery, and ensure security.



# REQUIREMENT GATHERING

---

## 1. **Project Overview:**

- InstHelper is designed to revolutionize college campus operations through automation and advanced technology integration. The project focuses on addressing administrative and logistical challenges within the campus, specifically in the areas of College Vehicle Management and Canteen Management. The primary objectives are to streamline operations, enhance user convenience, and provide real-time updates for better resource management and operational efficiency

## 2. **System Scope:**

- The system is proposed as a full-scale implementation for college campuses. It will integrate multiple functionalities into a single platform accessible via a mobile application and an admin web interface

## 3. **Target Audience:**

- Students
- Faculty
- Administrators
- Visitors
- Parents
- Drivers

## 4. **Modules:**

- Authentication: Manages user registration, login, and role-based access control.
- College Vehicle Management: A Flutter Android app accessible only to college staff, allowing drivers to manage vehicle details like insurance, test dates, maintenance schedules, and trip details; administrators can oversee and monitor vehicle and trip information.
- Canteen Management System: A web application replacing Loyverse, integrated with the college website for easy online booking of canteen products.
- Admin Dashboard: Provides graphical data representation and real-time updates for administrators.

**5. User Roles:**

- College Vehicle Management System Users:
  1. Drivers: Manage vehicle details, log trips, and receive notifications for maintenance and insurance.
  2. Administration: Monitor vehicle information, approve trip requests, generate reports, and communicate with drivers.
  3. Admin: Oversee all system aspects, manage user accounts, monitor system health, and handle data backup and recovery.
- Canteen Management System Users:
  1. Students: Browse menu, place and track orders, view order history, and manage payments.
  2. Faculty: Browse menu, place and track orders, view order history, and manage payments.
  3. Canteen Staff: Manage menu items, process orders, track inventory, generate reports, and manage promotions.
  4. Administrators: Monitor system performance, manage users, access financial reports, review audit logs, and handle system integration.
  5. Admin: Oversee all system functionalities, configure system settings, manage backups and recovery, and ensure security.

**6. System Ownership:**

- The system is owned by the academic institution (college/university)

**7. Industry/Domain:**

- Education

**8. Data Collection Contacts:**

- Amal K Jose
  - Role: Assistant Professor
  - Contact: 94964 40324

- Nisha E C
  - Role: AES Software Developer
  - Contact: 82813 41319
- Aleena Joseph
  - Role: AES Software Developer
  - Contact: 8086025320
- Jaison Joseph
  - Role: Driver cum Office Assistant
  - Contact: 9495313766

**9. Questionnaire for Data Collection:**

- What are the primary challenges you face with the current vehicle management system?
- What features would be most valuable in a new vehicle management system for college staff?
- How do you currently manage canteen orders and inventory?
- What issues or limitations have you encountered with the existing canteen management system?
- What key functionalities do you expect from the new canteen management system?
- How important is it for the canteen management system to be integrated with the college website?
- What specific data or reports would be most useful for administrators in the new system?
- How should notifications and alerts be managed in the vehicle management system?
- What types of user roles and permissions should be defined for the new systems?
- Are there any additional features or improvements you would like to see in either the vehicle management or canteen management systems?

# FEASIBILITY STUDY

---

Feasibility study is an important phase in software development process. It enables developers to have a clear picture of the product being developed in terms of outcomes of the product, operational requirements for implementing it, etc. A feasibility study is conducted to determine whether the project will, upon completion, fulfil the objectives of the organization in relation to the work, effort, and time invested in it. As a result, a new application often undergoes a feasibility assessment before approved for development. A feasibility study enables the developer to predict the projects usefulness and potential future. An evaluation of a system proposals viability takes into account its impact on the organization, capacity to satisfy user needs, and efficient use of resources.

Various feasibility studies are:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

## **Technical Feasibility**

The Vehicle Management System (VMS) project is technically feasible due to the college's adequate IT infrastructure, which includes robust servers and network capabilities to support the application. The development team possesses the necessary expertise in Flutter, backend technologies, and database management. Utilizing Flutter for cross-platform development ensures broad compatibility and benefits from a large, active support community. This stable and well-supported technology stack will facilitate the successful implementation of the VMS.

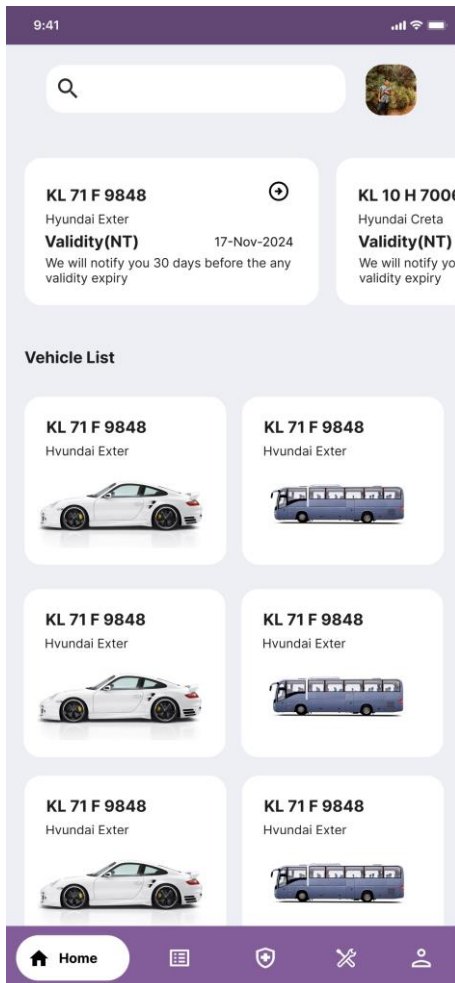
## **Operational Feasibility**

The VMS is feasible and poised for smooth adoption within the college. Initial feedback from stakeholders shows high acceptance and enthusiasm for the system. To ensure smooth integration and usage, training sessions will be conducted for users. The system will seamlessly integrate with existing campus databases and infrastructure, minimizing disruptions while leveraging current resources.

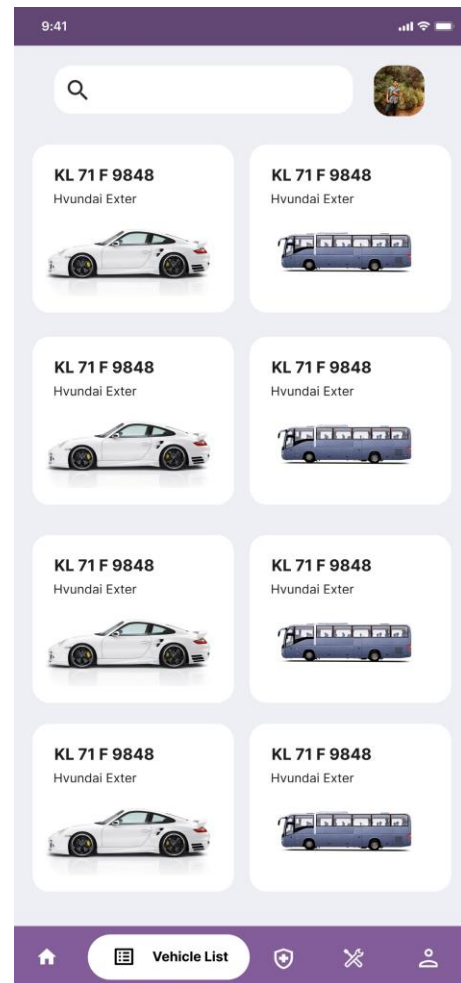
## **Economic Feasibility**

The Vehicle Management System (VMS) project presents a viable and cost-effective investment. The total estimated cost for development and deployment, including necessary hardware and software purchases, is modest, ensuring that it falls within a low budget. The anticipated benefits are substantial: increased operational efficiency, reduced administrative workload, and enhanced user satisfaction. These improvements are expected to lead to long-term savings and better resource utilization, effectively offsetting the initial costs.

# USER INTERFACE DESIGN USING FIGMA

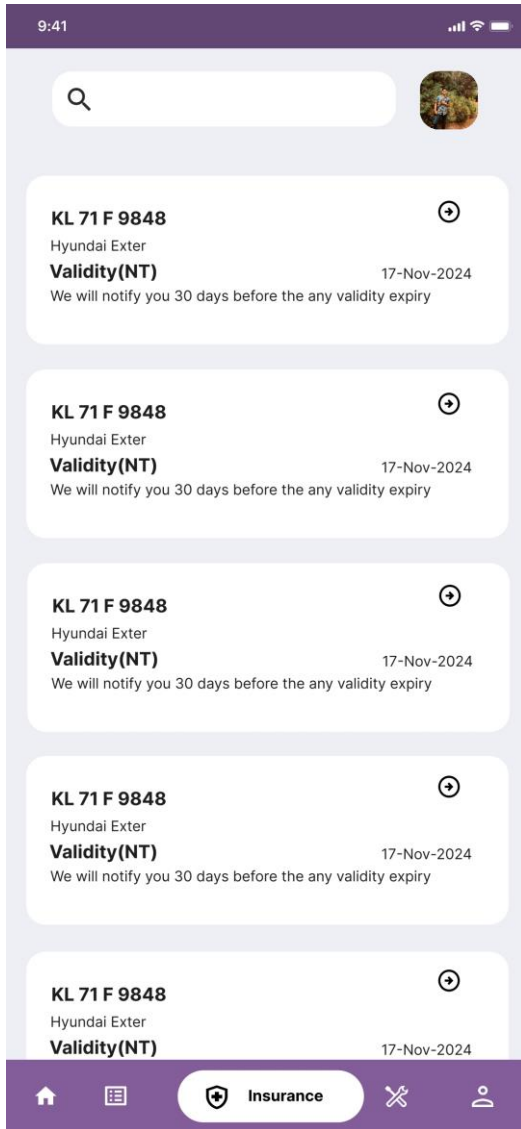


Home

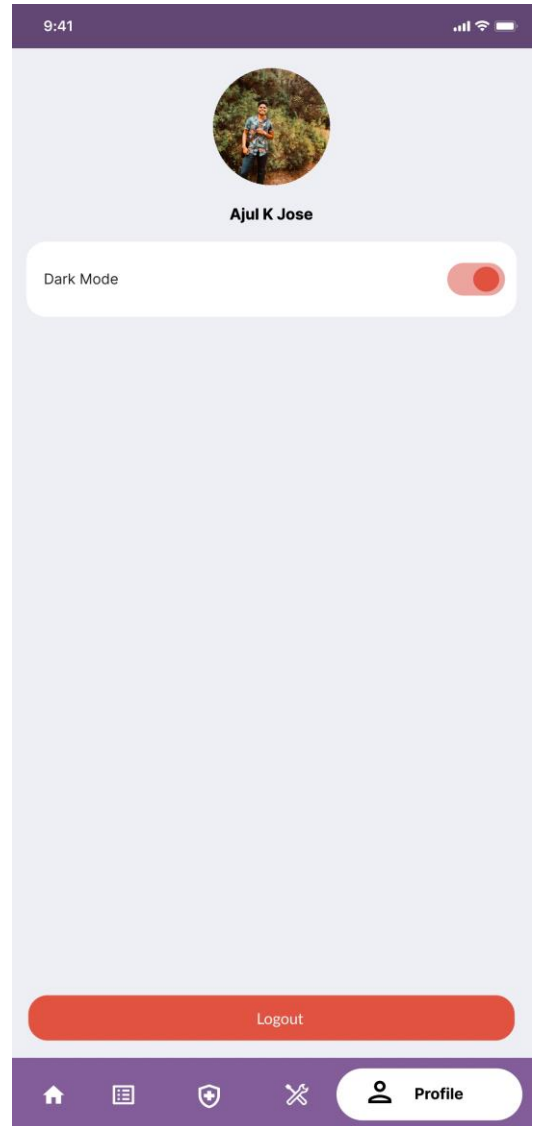


Vehicle List





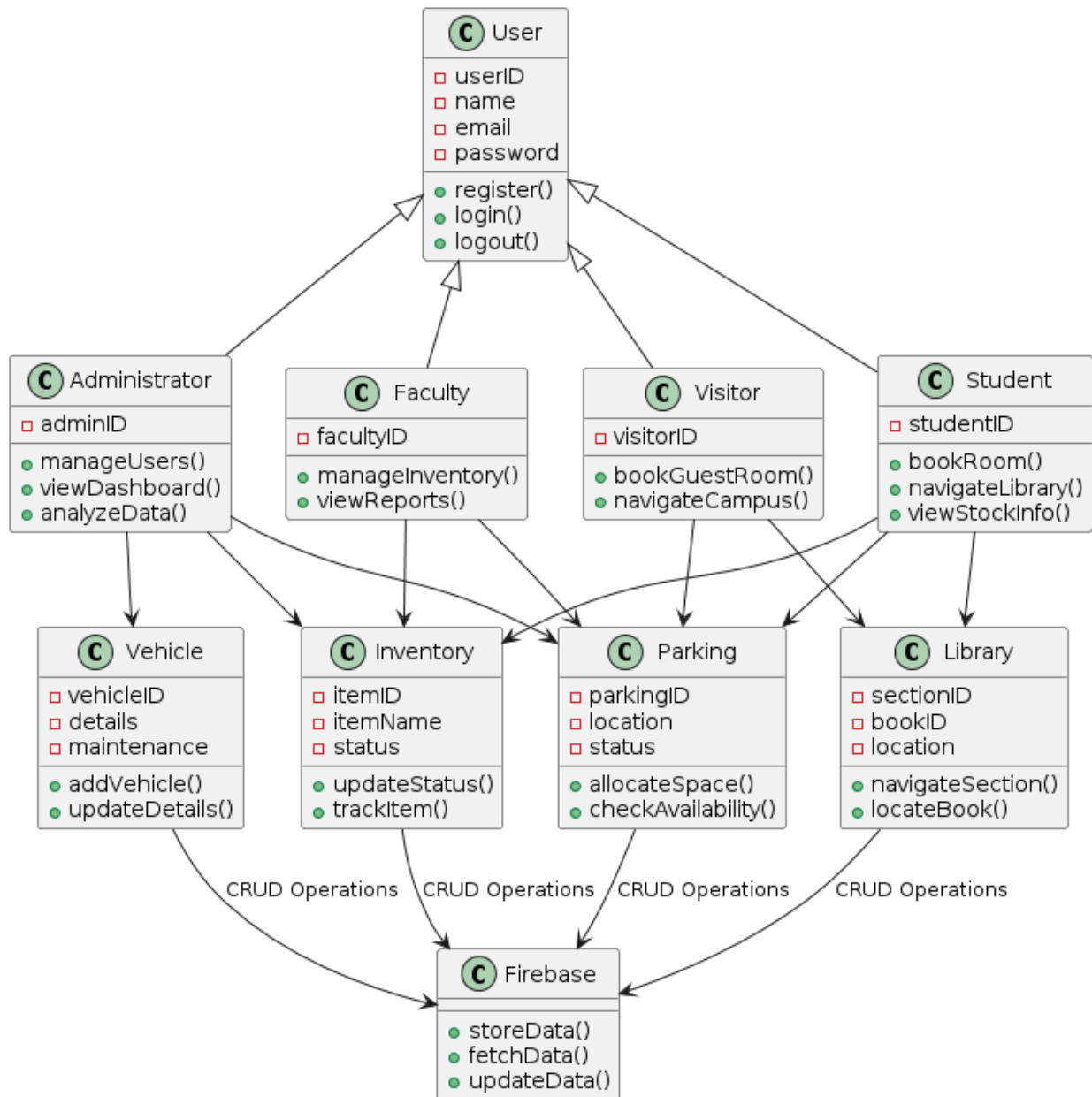
Insurance



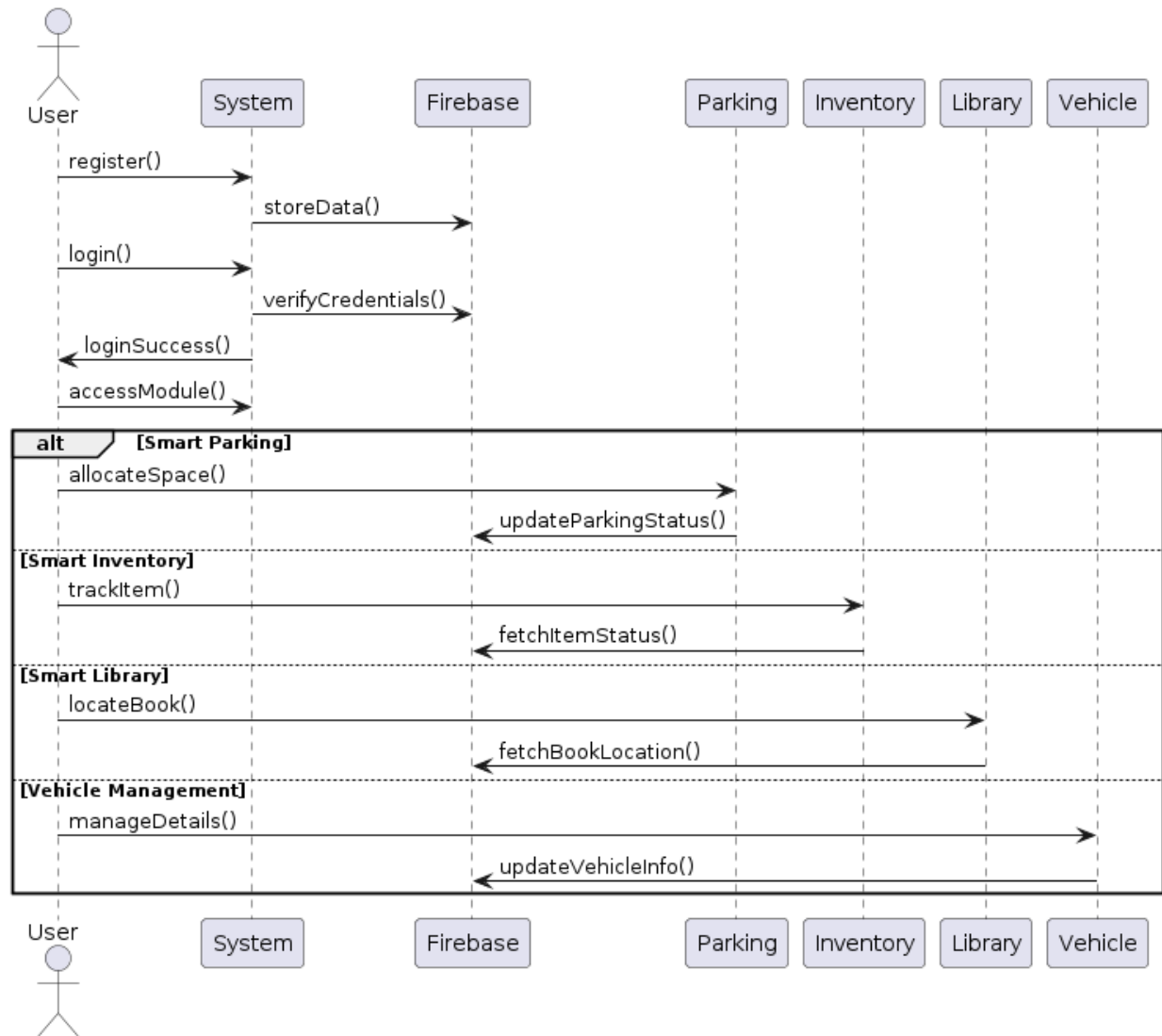
Profile

# UML DIAGRAM

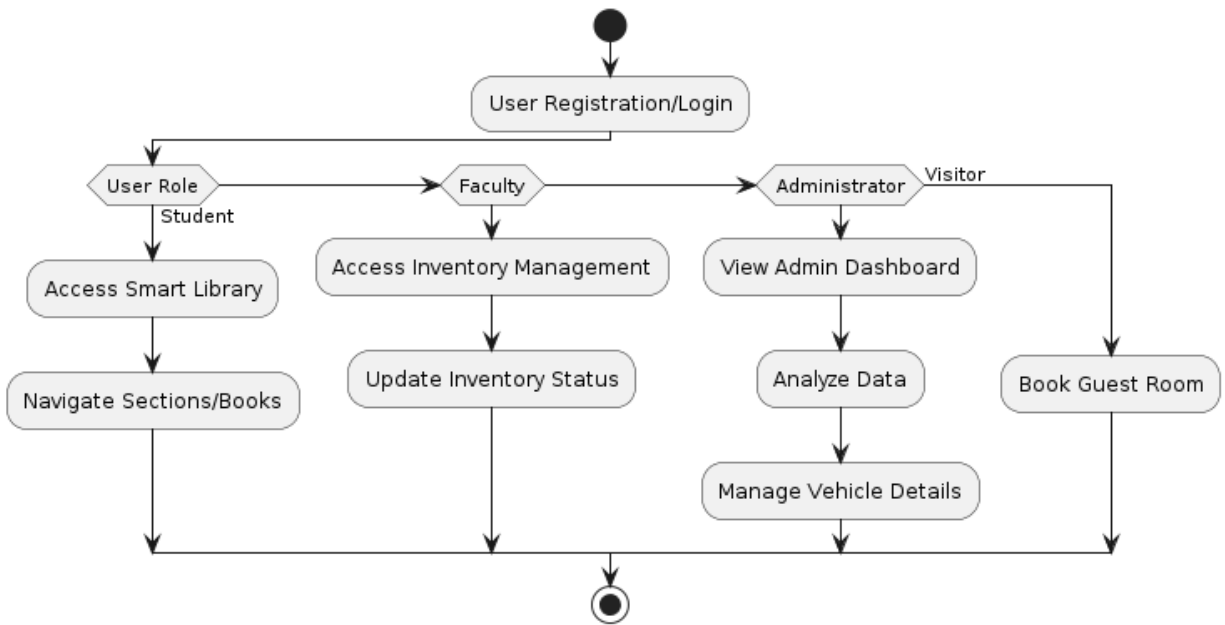
## Class Diagram



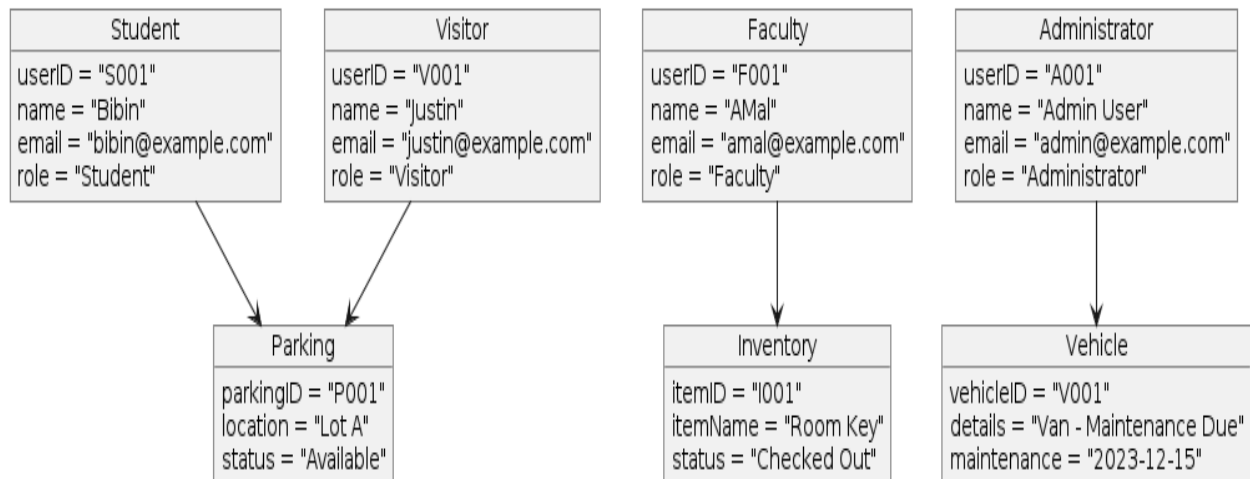
## Sequence Diagram

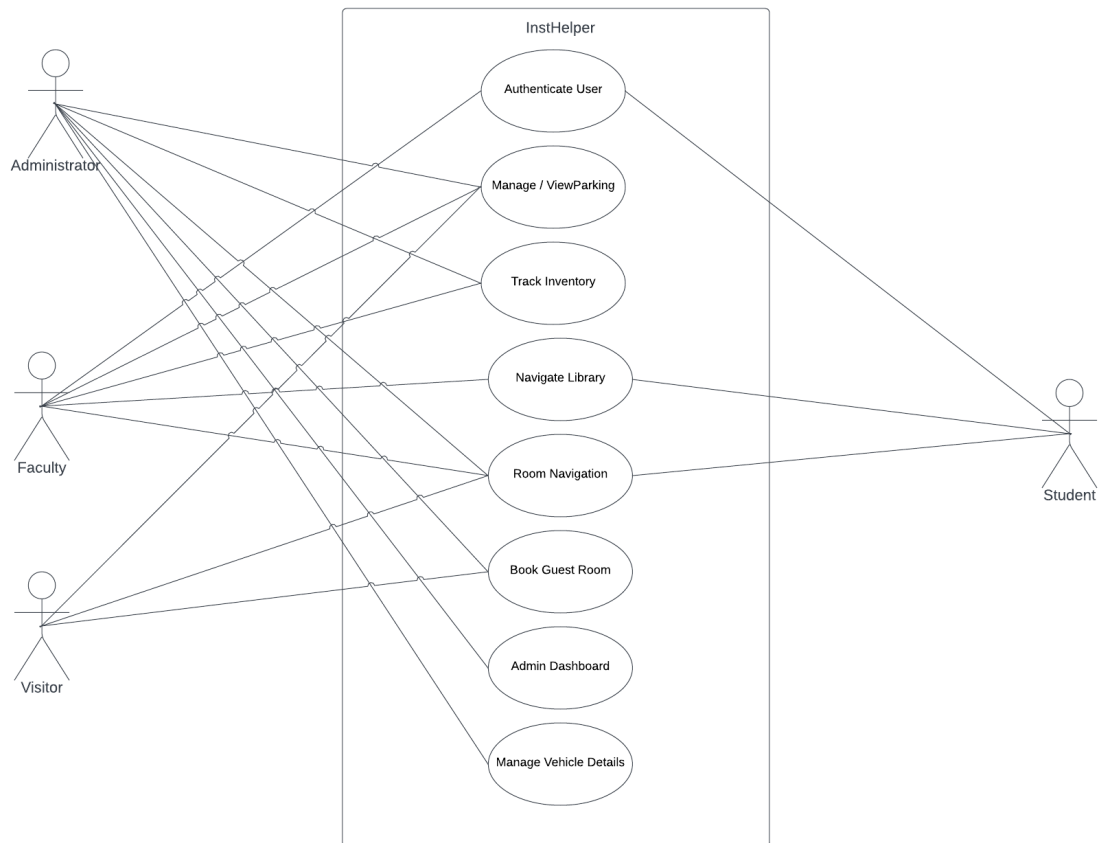


## Activity Diagram



## Object Diagram



**Use Case Diagram**

# TABLE DESIGN

---

## 1. tbl\_drivers

No	Fieldname	Datatype	Key	Constraints	Description
1	id	int(11)	PRIMARY	NOT NULL, AUTO_INCREMENT	Unique identifier for each driver
2	name	varchar(30)			Driver's name
3	contact	varchar(30)			Driver's contact number
4	license	varchar(100)			Driver's license number
5	onCreated	timestamp		NOT NULL, DEFAULT current_timestamp()	Timestamp of record creation

## 2. tbl\_fitness

No	Fieldname	Datatype	Key	Constraints	Description
1	id	int(11)	PRIMARY	NOT NULL, AUTO_INCREMENT	Unique identifier for each fitness record
2	vehicle_id	varchar(30)	Foreign	NOT NULL, FOREIGN KEY (vehicle_id) REFERENCES tbl_vehicles(vehicle_id)	Vehicle registration number
3	exp_date	date			Expiration date of fitness certificate
4	documents	varchar(300)			URL to fitness documents
5	onCreated	timestamp		NOT NULL, DEFAULT current_timestamp()	Timestamp of record creation

**3. tbl\_fuel**

No	Fieldname	Datatype	Key	Constraints	Description
1	fuel_id	int(11)	PRIMARY	NOT NULL, AUTO_INCREMENT	Unique identifier for each fuel type
2	type	varchar(20)			Name of the fuel type

**4. tbl\_insurance**

No	Fieldname	Datatype	Key	Constraints	Description
1	id	int(11)	PRIMARY	NOT NULL, AUTO_INCREMENT	Unique identifier for each insurance record
2	vehicle_id	varchar(30)	Foreign	NOT NULL, FOREIGN KEY (vehicle_id) REFERENCES tbl_vehicles(vehicle_id)	Vehicle registration number
3	exp_date	date			Expiration date of insurance
4	documents	varchar(300)			URL to insurance documents
5	onCreated	timestamp		NOT NULL, DEFAULT current_timestamp()	Timestamp of record creation

**5. tbl\_pollution**

No	Fieldname	Datatype	Key	Constraints	Description
1	id	int(11)	PRIMARY	NOT NULL, AUTO_INCREMENT	Unique identifier for each pollution record
2	vehicle_id	varchar(30)	Foreign	NOT NULL, FOREIGN KEY (vehicle_id) REFERENCES tbl_vehicles(vehicle_id)	Vehicle registration number

3	exp_date	date			Expiration date of pollution certificate
4	documents	varchar(300)			URL to pollution documents
5	onCreated	timestamp		NOT NULL, DEFAULT current_timestamp()	Timestamp of record creation

## 6. tbl\_trips

No	Fieldname	Datatype	Key	Constraints	Description
1	id	int(11)	PRIMARY	NOT NULL, AUTO_INCREMENT	Unique identifier for each trip
2	vehicle_id	varchar(300)	Foreign	NOT NULL, FOREIGN KEY (vehicle_id) REFERENCES tbl_vehicles(vehicle_id)	JSON array of vehicle registration numbers
3	driver	varchar(300)			JSON array of driver names
4	purpose	varchar(300)			Purpose of the trip
5	starting_time	varchar(10)			Starting time of the trip
6	route	varchar(300)			JSON array of route coordinates or place names
7	starting_km	varchar(300)			JSON array of starting kilometers for each vehicle
8	ending_km	varchar(300)			JSON array of ending kilometers for each vehicle
9	onCreated	timestamp		NOT NULL, DEFAULT current_timestamp()	Timestamp of record creation



10	onUpdate	timestamp		NOT NULL, DEFAULT current_timestamp() ON UPDATE current_timestamp()	Timestamp of last update
----	----------	-----------	--	--	-----------------------------

## 7. tbl\_vehicle

No	Fieldname	Datatype	Key	Constraints	Description
1	id	int(11)	PRIMARY	NOT NULL, AUTO_INCREMENT	Unique identifier for each vehicle
2	assigned_driver	varchar(50)	Foreign	NOT NULL, FOREIGN KEY (driver_id) REFERENCES tbl_drivers(driver_id)	Name of the assigned driver
3	chassis_no	varchar(100)			Chassis number of the vehicle
4	current_mileage	double			Current mileage of the vehicle
5	total_km	double			Total kilometers traveled
6	emergency_contact	varchar(10)			Emergency contact number
7	engine_no	varchar(100)			Engine number of the vehicle
8	fuel_type	varchar(10)	Foreign	NOT NULL, FOREIGN KEY (fuel_type) REFERENCES tbl_fuels(fuel_id)	Type of fuel used
9	model	varchar(100)			Model of the vehicle
10	ownership	varchar(30)			Owner of the vehicle

11	purpose_of_use	varchar(100)			Purpose for which the vehicle is used
12	registration_date	varchar(20)			Date of vehicle registration
13	registration_number	varchar(30)			Vehicle registration number
14	uploaded_files	text			JSON array of URLs to uploaded files
15	vehicle_type	varchar(20)	Foreign	NOT NULL, FOREIGN KEY (vehicle_type) REFERENCES tbl_vehicle_type(id)	Type of vehicle
16	onCreated	timestamp		NOT NULL, DEFAULT current_timestamp()	Timestamp of record creation
17	onUpdated	timestamp		NOT NULL, DEFAULT current_timestamp() ON UPDATE current_timestamp()	Timestamp of last update

## 8. tbl\_vehicle\_gallery

No	Fieldname	Datatype	Key	Constraints	Description
1	id	int(11)	PRIMARY	NOT NULL, AUTO_INCREMENT	Unique identifier for each gallery entry
2	vehicle_id	varchar(30)	Foreign	NOT NULL, FOREIGN KEY (vehicle_id) REFERENCES tbl_vehicles(vehicle_id)	Vehicle registration number
3	image	varchar(500)		NOT NULL	JSON array of image URLs

**9. tbl\_vehicle\_type**

No	Fieldname	Datatype	Key	Constraints	Description
1	id	int(11)	PRIMARY	NOT NULL, AUTO_INCREMENT	Unique identifier for each vehicle type
2	type	varchar(30)			Name of the vehicle type
3	image	varchar(300)			URL to the image representing the vehicle type

