Palestine Technical University – Kadoorie

College of Engineering and Technology

Department of Computer Systems Engineering

Course name: Software Engineering

Project title:
SYARTI (CRMS)

By:

Ahmad Hamad	202210240 – Section 4
Amer Abuyaqob	202210101 – Section 4
Mohammad Khasati	202210286 – Section 4
Mustafa AbuAli	202210901 – Section 4

Supervisor: Dr. Osama Hamed

Tulkarm, Palestine

25 MAY 2025

TABLE OF CONTENTS

ABSTRACT

Traditional car rental processes require customers to visit multiple agencies or make numerous calls to check **vehicle availability**, making the process inefficient and **time-consuming**. Even after significant effort, customers may not find the desired vehicle, leading to frustration. Similarly, rental agencies relying on **paper-based** record-keeping face challenges in tracking customer details, **vehicle availability**, and **maintenance history**.

The proposed **Syarti (CRMS)** is a **web-based platform** that streamlines operations with **real-time fleet availability**, **online reservations**, and automated **customer management**. Customers can browse available vehicles, view promotions, **book rentals**, **manage accounts**, and **submit feedback**. On the other hand, rental agencies benefit from **centralized data storage**, **secure customer record management,** and **efficient vehicle tracking**.

By **digitizing and automating** these processes, **Syarti** enhances operational efficiency, improves customer satisfaction, and reduces administrative overhead, making car rentals more accessible and manageable for businesses in Palestine.

1 INTRODUCTION

1.1 PURPOSE

The purpose of this document is to provide a comprehensive description of **Syati** (**CRMS)**. It outlines the system's objectives, features, technical interfaces, operational constraints, and responses to user interactions. It also serves as a reference for stakeholders to ensure the system meets the specified requirements and objectives.

1.2 DOCUMENT CONVENTION

1.2.1 Text Styles

Style	Usage	Example
Bold	Mandatory requirements, key terms.	**real-time fleet availability**
Italics	examples, technical terms, or emphasis.	(*e.g., insurance, registration*), *API*

1.2.2 Abbreviations

Abbreviation	Meaning
CRMS	Car Rental Management System
KYC	Know Your Customer (ID verification)
API	Application Programming Interface

1.2.3 Terminology

- **Role-Based Access:** Privileged accounts (*e.g., Manager*) with restricted permissions.
- **Fleet:** Collection of vehicles owned by the rental agency

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

This document is intended for various stakeholders involved in the development and use of the **Syarti (CRMS)**. The key audiences are:

1.3.1 System Developers

- **Role:** Design, implement, and maintain the **CRMS**.
- **Focus:** Technical specifications, *API* integration, and security protocols.

1.3.2 Rental Agency Owners/Managers

- **Role:** Use the system to manage vehicles, customers, and financial workflows.
- **Focus:** Business features (*e.g., billing, reporting*) and user roles.

1.3.3 End Users (Customers)

- **Role:** Rent vehicles via the **Syarti** webapp.
- **Focus:** Booking workflows, account management, and feedback submission.

1.3.4 PTUK Faculty

- **Role:** Evaluate academic compliance and SRS structure.
- **Focus:** IEEE formatting, traceability, and project scope.

1.4 PROJECT SCOPE

This section defines the boundaries of the **Syarti (CRMS)**, including its core functionalities, technical limitations, and excluded features.

1.4.1 In-Scope

- **User Management:** Secure accounts (email/password), role-based access (Admin/Customer).
- **Vehicle Management:** Real-time availability, online booking.
- **Payment & Billing:** Automated cost calculation (daily/weekly rates).
- **Business Operations:** Maintenance/insurance tracking, basic reports.
- **Technical Scope:** Web-based (mobile-responsive).

1.4.2 Out-of-Scope

- Mobile apps.
- GPS tracking.
- Third-Party KYC tools.

1.4.3 Constraints

- Internet required.
- Budget limits.

1.4.4 Assumptions

- **User Literacy:** Customers and staff can navigate a web interface.
- **Legal Compliance:** Both Customers and agencies upload valid documentations
- **Manual Verification:** Agencies manually verify customers' IDs and driver Licenses.

REFERENCES

[1] Syarti Devs, "*Syarti (CRMS) Webapp*," 2025. [Online]. Available: https://p-t-u-k.github.io/Syarti/

[2] PTUK Software Engineering Course, "*Sw Eng Assignment - Part 1.pdf: Software Requirements Specification (SRS)*" 2025. [Online]. Available: https://lms.ptuk.edu.ps/mod/assign/view.php?id=633283

[3] PTUK Software Engineering Course, "*Sw Eng Assignment - Part 2.pdf: Software Requirements Specification (SRS)*" 2025. [Online]. Available: https://lms.ptuk.edu.ps/mod/assign/view.php?id=633305

[4] Syarti Devs, "*Syarti - GitHub Repository*," 2025. [Online]. Available: https://github.com/P-T-U-K/Syarti

[5] Syarti Devs, "*Syarti Project Diagrams – Miro Board*," 2025. [Online]. Available: https://miro.com/app/board/uXjVIyBPe9k=/?share_link_id=738229256317

[6] IEEE Standards Association, "*IEEE Recommended Practice for Software Requirements Specifications*" IEEE Std 830-1998, 1998. [Online]. Available: https://standards.ieee.org/ieee/830/1222/

2 Overall Description

2.1 Product Perspective

Syarti (**CRMS**) is currently developed as a standalone **front-end prototype** of a web application that acts as the **main entry point** for users before login for all user roles (**Customer**, **Admin**, and **Employee**). This **prototype** focuses solely on the presentation layer and *does not* include any **backend logic** or **database** connectivity.

The system is built using **HTML5**, **CSS3**, **Bootstrap 5 RTL**, and **Font Awesome icons**, and is deployed *publicly* via **GitHub Pages**. The user interface is *responsive* and optimized for both **desktop** and **mobile** browsers.

There are *no* external interfaces like **APIs** or **third-party services** connected yet. However, future development plans include implementing an **admin dashboard**, **backend logic**, and a **database** for *car inventory management*, *user accounts*, and *transactions*. While support for **payment gateways** is also considered, it *has not been* implemented yet.

2.2 Product Functions

The implemented **home page** *provides* users with a visually structured experience, laying the groundwork for more advanced functionality and extensive features planned for different **user roles** in future phases.

2.2.1 Implemented UI Features:

- **Navigation bar** for accessing different sections.
- **Booking form** for selecting location and dates.
- Static **car listing** with pricing and specifications.
- **View All Cars button**.
- **Testimonials** and **customer ratings**.

2.2.2 Planned Functional Features:

Customer Role:

- **Book**, **cancel**, and **manage** reservations.
- Secure **online payments**.
- View **rental history**.
- Manage **account details**, **legal documents**, and **payment options**.
- Submit and view **customer feedback**.

Admin Role:

- Real-time **vehicle inventory** management.
- **Approve** or **reject** bookings.
- Oversee **customer accounts** and **documents**.
- Generate **car services reports**.
- Track **expenses**, **taxes**, and **financial reports**.
- Control **home page offers** and **pricing rules**.

Employee Roles:

- **Pickup/Dropoff Handlers:** View accepted car **bookings**, assign themselves to **car handoffs**, view customer **booking data** and his **contact info**.
- **Fleet Managers:** Manges **car details** (*add*, *remove*, *update*), **insurance** and **registrations**, **service logs**.
- **Legal Verifiers:** Review and **validate** customers' *IDs* or *Passports*, *licenses*, cars' *insurances* and *registrations*, and run **background checks**
- **Accountants:** Review all **tax** and **expense reports** (*fuel*, *insurances*, *registrations*, *maintenance fees*).

2.3 USER CLASSES AND CHARACTERISTICS

Syarti system is designed to support *multiple* **user types**, each with **distinct roles** and **responsibilities**. These user classes include **Customers**, **Admins**, and various categories of **Employees**. Each employee role has *tailored* system access *suited* to their specific tasks.

2.3.1 Customers

- General users who **rent cars** via the platform.
- Can **view**, **book**, and **cancel reservations**.
- **Manage** *profiles*, *documents*, and *payments*.
- Submit and view **feedback**.
- Use *smartphones* or *PCs*.
- Expected to have *low to moderate* **technical knowledge**.

2.3.2 Admins

- Business ***owners*** or system ***supervisors***.
- **Manage** *fleet inventory* and *bookings*.
- **Oversee** *customer profiles* and *legal docs*.
- **Review financial reports** (*expenses*, *taxes*, *service logs*).
- *Moderate to advanced* **technical familiarity** expected.

2.3.3 Employees

Pickup/Dropoff Handler

- Manages car **handovers** to/from customers.
- Views *approved* **bookings' data** and customers' **contact info**.

Fleet Manager

- Updates **vehicle data**, **maintenance records**, **insurance** and **registration**.

Legal Verifier

- Reviews *IDs* or *passports*, *licenses*, *insurance policies*, and *registrations documents*.
- Performs ***background checks*** for *new* customers.

Accountant

- Reviews *bills* and *taxes*, as well as *fuel*, *service*, *insurance* and *registration expenses*.

2.4 OPERATING ENVIRONMENT

- **Deployment Platform: GitHub Pages** (*public access* via direct **URL**)

- **System Type:** *Front-end-only prototype* (**HTML5**, **CSS3**)

- **Frameworks/Libraries:**
  - **Bootstrap 5 RTL**
  - **Font Awesome for icons**

- **Browser/Device Compatibility:**
  - Fully **tested** on modern browsers: *Chrome*, *Firefox*, *Safari*, *Edge*
  - **Responsive** on both *desktop* and *mobile*
  - *No minimum* **version requirements**

- **Language and Layout:**
  - **Arabic-only** interface
  - **Fully RTL** support (*including forms and components*)

- **Backend and Database:**
  - *No* **backend**, **APIs**, or **database** currently implemented
  - *No* **hosting plan** yet for future **backend**

2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

Syarti project is being developed solely for **educational** purposes and is currently subject to several *design* and *implementation* **constraints**. While *there are no* **restrictions** on the use of **backend technologies**, the team **lacks** the required experience in **backend development**, and **API integration**. As a result, the system is currently **limited** to a **front-end-only prototype** and hosted via **GitHub Pages** *by choice*, with *no external* **limitations**.

The application is designed for **Arabic-speaking users** only, and no **multilingual** or accessibility support is currently implemented. Due to the **academic nature** of the project, *there are no requirements* to follow **privacy laws**, **rental regulations**, or **data protection** practices, and the system *does not* **store** or **handle** any **user data**.

2.6 Assumptions and Dependencies

Syarti's prototype is developed under a set of **practical assumptions** and **dependencies**

2.6.1 Assumptions:

- Users *will have* access to **modern desktop** or **mobile** browsers.
- Users *will have* a stable **internet connection**.
- The development team *will acquire* **backend development skills** in future phases.

2.6.2 Dependencies:

- **Hosting** depends on the *continued availability* of **GitHub Pages**.
- **UI** depends on the **Bootstrap 5 RTL** framework and **Font Awesome icon** library.
- Future features *will depend* on integration with **third-party** services such as ***payment gateways*** and ***database*** systems.

3 System Features (Functional Requirements)

3.1 Vehicle Search and Filtering

3.1.1 Description and Priority

This feature provides a **user interface** element on the **home page** that allows **customers** to search for rental cars by specifying **pickup location**, **return location**, and **rental dates**. Currently, this functionality is implemented only as a **front-end prototype** *with no actual* **backend data processing** or **filtering logic**.

This is a **high-priority** feature for the **customer-facing** side of the system, as it *directly supports* the **booking workflow** and **user engagement** with the platform.

3.1.2 Stimulus/Response Sequences

- **Stimulus:** A customer visits the homepage and interacts with the "**احجز الآن**" (**Book Now**) search card.
- **Expected Future Response:**
  - The system will **fetch** available vehicles from the **backend** based on the entered **pickup location** and **dates**.
  - The matching results will be **displayed** in the *vehicle listing section* or redirected to a *search results page*.
  - If *no cars are available*, the system will **notify** the user accordingly.

3.1.3 Functional Requirements

These requirements describe intended future functionality:

- **FR3.1.1**: The system *shall allow* the **customer** to select a **pickup location**
- **FR3.1.2**: The system *shall allow* the **customer** to enter a **pickup date** and a **return date** using a **date picker**.
- **FR3.1.3**: The system *shall validate* that the **return date** *is not earlier* than the **pickup date**.
- **FR3.1.4**: *Upon clicking* the "**بحث**" (**Search**) button, the system *shall send* the **selected location** and **date data** to the **backend** for processing.
- **FR3.1.5**: The system *shall display* a list of **available vehicles** that *match* the **selected criteria**.
- **FR3.1.6**: The system *shall show* a **message** if *no matching* **vehicles** are found.

3.2 VEHICLE LISTINGS AND SPECIFICATIONS

3.2.1 Description and Priority

This feature presents **customers** with a visual list of **available vehicles**. Each listing displays key **vehicle specifications** such as *model name*, *fuel type*, *seating capacity*, *transmission type*, and *rental price*. Vehicles may also carry **badges** indicating their status (*e.g., new, special offer*).
Currently, this is a **static front-end UI component** designed to simulate dynamic listings. Once integrated with a backend, it will reflect real-time car data and availability.

This is a **high-priority** feature as it is the main method for users to browse **rental options**.

3.2.2 Stimulus/Response Sequences

- **Stimulus**: A customer visits the **home page** or performs a vehicle search.
- **Expected Future Response**:
  - The system *will retrieve* a list of **vehicles** that *match the filter* **criteria**.
  - Each vehicle *will be rendered* with its **photo**, **name**, **key specs**, **price**, and an "**احجز الآن**" (**Book Now**) button.
  - Badges *will be shown* for **special vehicles** (e.g., "*عرض خاص*" , *new arrivals*).
  - Clicking " **احجز الآن**" *will initiate* the **booking workflow** for the **selected vehicle**.

3.2.3 Functional Requirements

- **FR3.2.1**: The system *shall display* each **vehicle** with its **name**, **image**, **fuel type**, number of **passengers**, and **transmission** type.
- **FR3.2.2**: The system *shall show* the **daily rental price** next to each vehicle.
- **FR3.2.3**: The system *shall display* a **badge** (e.g., "عرض خاص" , "جديد") for **applicable vehicles**.
- **FR3.2.4**: The system *shall include* a " **احجز الآن**" button for each **vehicle card**.
- **FR3.2.5**: In the future version, this button *shall trigger* the **reservation process** for that **specific car**.
- **FR3.2.6**: The system *shall support* dynamic vehicle listings fetched from a **backend database** (**planned**).

3.3 Promotional Offers Display

3.3.1 Description and Priority

This feature highlights ***promotional rental deals*** and ***special offers*** to attract **customer** attention. In the current **front-end prototype**, these offers *are represented* using **visual badges** such as "**عرض خاص**" or "**جديد**" displayed on top of vehicle cards.

In future versions, these **badges** will be managed by the **admin panel** and *reflect real-time* **offers** configured through a **backend** system. This is a **medium-to-high priority** feature due to its **marketing value** and **influence** on **customer** *decision-making*.

3.3.2 Stimulus/Response Sequences

- **Stimulus**: A **customer** *visits* the **home page** or *scrolls* through the **car listings**.
- **Expected Future Response**:
  - The system *will detect* which **vehicles** are *linked* to **active promotions**.
  - For each qualifying vehicle, a **badge** (e.g., "عرض خاص", "جديد") *will be shown* in a *visible position* on the **car card**.
  - In the **admin interface**, **promotional tags** *will be added*, *edited*, or *removed* per **vehicle**.

3.3.3 Functional Requirements

- **FR3.3.1**: The system *shall display* a **promotional badge** on **vehicles** that *are part* of an **active offer**.
- **FR3.3.2**: The system *shall allow* the **admin** to *add* or *remove* **promotional tags** for each **vehicle**.
- **FR3.3.3**: The system *shall support* multiple types of **promotions** (*e.g., discount offers, new arrivals*).
- **FR3.3.4**: The **promotional badge** *shall appear* in the **top area** of the **car card** and *remain clearly visible* across screen sizes.
- **FR3.3.5**: The system *shall ensure* that **promotional badges** *do not* **overlap or obscure** key **vehicle information**.

3.4 Branch Location Selector

3.4.1 Description and Priority

This feature *allows* **customers** to select a **pickup** and **return** location from a **predefined list of branches**. In the *current prototype*, this functionality is implemented as a *static dropdown* in the **booking form** with *hardcoded* **location names**.

In *future versions*, the system *will support* **dynamic loading** of **available branches** from the **backend**. This feature is of **medium priority**, as it *directly supports* the **search** and **reservation workflow**.

3.4.2 Stimulus/Response Sequences

- **Stimulus**: A **customer** *opens* the **home page** and *interacts* with the *"مكان الاستلام"* dropdown in the **booking form**.
- **Expected Future Response**:
    - The system *will populate* the list of **branches dynamically**.
    - Once a **location is selected**, it *will be associated* with the **customer's booking intent**.
    - In a complete workflow, both **pickup** and **drop-off branches** may be supported.

3.4.3 Functional Requirements

- **FR3.4.1**: The system *shall display* a **dropdown list** of **available branch locations** in the **booking form**.

- **FR3.4.2**: The system *shall allow* the **customer** to select one **branch** as the **pickup location**.

- **FR3.4.3**: The system *shall store* the **selected location** for use during the **booking process**.

- **FR3.4.4**: The system *shall allow* for the **addition**, **modification**, or **removal** of **branches** by an **admin**.

- **FR3.4.5**: The system *shall validate* that a **pickup location** has *been selected* before processing a **booking request**.

3.5 Customer Testimonials Display

3.5.1 Description and Priority

This feature presents **customers' reviews** and **customers' ratings** as social proof to build trust with **potential renters**. In the current **front-end prototype**, **testimonials** *are displayed* as **static cards** with **star ratings**, **review quotes**, and **reviewer names** with **avatars**.

In future iterations, the system *will retrieve* **testimonials** *dynamically* from a **backend database**, possibly allowing **customers** to *submit* their own **feedback**. This is a **medium-priority** feature, valuable for **marketing** and **credibility** but *not essential* to core **booking workflows.**

3.5.2 Stimulus/Response Sequences

- **Stimulus**: A **customer** *scrolls* to the *"آراء عملائنا"* (**Customer Reviews**) section on the **home page**.

- **Expected Future Response**:
  - The system *will fetch* and *display* a curated list of **testimonials**, each containing a **review**, **star rating**, and **author details**.
  - In future versions, the list may be **paginated** or **randomized**.

3.5.3 Functional Requirements

- **FR3.5.1**: The system *shall display* **customer testimonials** including **star rating**, **review text**, and **reviewer name/avatar**.

- **FR3.5.2**: The system *shall display* multiple **testimonials** in a **card-based layout**.

- **FR3.5.3**: The system *shall support* the **storage** and **retrieval** of **testimonials** from a **backend database**.

- **FR3.5.4**: The **testimonials** section *shall be* **responsive** and **visually optimized** for both **desktop** and **mobile** views.

3.6 CONTACT AND SUPPORT FORM

3.6.1 Description and Priority

This feature allows **customers** to *initiate contact* with the **rental agency** for *inquiries*, *support*, or *feedback*. In the **current prototype**, the "**اتصل بنا**" (**Contact Us**) section is represented in the **navbar**, but the actual *form and logic* for **message submission** are *not yet implemented*.

In future versions, this section will include a *fully functional* **contact form** that **sends messages** to a designated **support channel**. This is a **medium priority** feature, essential for **post-booking** support and customer satisfaction.

3.6.2 Stimulus/Response Sequences

- **Stimulus**: A **user** *clicks* the "***اتصل بنا***" (**Contact Us**) link or *navigates* to the **support** section.
- **Expected Future Response**:
  - The system *will display* a **contact form** with fields like *name*, *email*, *subject*, and *message*.
  - Upon submission, the system *will store* or *forward* the **message** to the **support team**.
  - A **confirmation** or **success** notification *will be shown* to the **user**.

3.6.3 Functional Requirements

- **FR3.6.1**: The system *shall provide* a **contact form** with the following fields: *Name*, *Email*, *Subject*, and *Message*.
- **FR3.6.2**: The system *shall validate* that all **required** fields *are filled* and that the **email** is in a *valid format*.
- **FR3.6.3**: Upon form submission, the system *shall store* the **message** and optionally *notify* the **admin** or **support team**.
- **FR3.6.4**: The system *shall provide* a **confirmation message** to the **user** after **successful submission**.
- **FR3.6.5**: The contact form *shall be responsive* and *accessible* on both **desktop** and **mobile** devices.

3.7 USER LOGIN AND ACCOUNT MANAGEMENT

3.7.1 Description and Priority

This feature allows **admin, customers,** and **employees** to **log in** to the system using their **credentials** and **manage** their **account information**. While the current **prototype** includes a **"تسجيل الدخول" (Login)** button in the **navbar**, *no actual* **authentication mechanism** is *implemented yet*.

In future versions, **users** of all rules will be able to *register*, *log in*, *update* **profile information**, and *view* their **reservation history**. This is a **high-priority** feature, essential for providing **personalized** service and managing **bookings**.

3.7.2 Stimulus/Response Sequences

- **Stimulus**: A user clicks the ***"تسجيل الدخول"*** **(Login)** button.

- **Expected Future Response**:
  - The system *will redirect* the **user** to a **login page** or **modal**.
  - Upon entering **valid credentials**, the system *will authenticate* the **user** and *redirect* them to their **profile dashboard**.
  - **Invalid login** attempts *will trigger* appropriate **error messages**.
  - Logged-in **users** *will see* personalized **navigation** and *access* to **account-related features**.

3.7.3 Functional Requirements

- **FR3.7.1**: The system *shall provide* a **login interface** requiring an **email** and **password**.

- **FR3.7.2**: The system *shall validate* **user credentials** against **stored records**.

- **FR3.7.3**: The system *shall redirect* authenticated **users** to a **profile dashboard**.

- **FR3.7.4**: The system *shall allow* **new users** to **register** using an **email** and **password**.

- **FR3.7.5**: The system *shall allow* **users** to *update* their **account details**, such as *name* and *contact information*.

- **FR3.7.6**: The system *shall restrict access* to account features unless the **user** is **authenticated**.

- **FR3.7.7**: The system *shall provide* **error messages** for failed login attempts (e.g., *wrong password*, *unregistered email*).

3.8 BOOKING AND RESERVATION WORKFLOW

3.8.1 Description and Priority

This feature enables **customers** to submit a **car rental booking request** by selecting a **vehicle**, choosing **pickup/drop-off locations** and **dates**, and confirming their details. Currently, this is a **front-end-only placeholder**. In future versions, the **booking** will undergo **multi-stage processing**:

1. **Customer submits booking**
2. **Admin reviews and approves**
3. **Pickup/Drop-off handlers receive the request**
4. **One handler accepts and contacts the customer**

This is a **critical, high-priority feature** that connects the **customer**, **admin**, and **operational teams**.

3.8.2 Stimulus/Response Sequences

- **Stimulus**: A **customer** *selects* a **vehicle** and *submits* a completed **booking form**.
- **Expected Future Response**:
  - The system *creates* a **pending booking record** and *forwards it* to the **admin** for review.
  - Once the **admin** *approves* the **request**, the system *notifies* all **pickup/drop-off handlers** assigned to the selected **location**.
  - The **first handler** *to accept* the **request** is *assigned* to it and can *access* **customer contact details** to follow up.
  - The **customer** *receives* a **final confirmation message**.

3.8.3 Functional Requirements

- **FR3.8.1**: The system *shall allow* the **customer** to initiate a **booking request** by selecting a **vehicle** and filling out a **reservation form**.
- **FR3.8.2**: The system *shall validate* all **input fields** (*vehicle ID, dates, location*) before submission.
- **FR3.8.3**: The system *shall create* a **pending booking record** and *route* it to the **admin** for review.
- **FR3.8.4**: The **admin** *shall be able* to *approve* or *reject* any pending **booking request**.
- **FR3.8.5**: Upon approval, the system *shall broadcast* the **booking details** to all **pickup/drop-off handlers** at the selected **branch**.
- **FR3.8.6**: The first **handler** to *accept* the **booking** *shall be assigned* to it **exclusively**.
- **FR3.8.7**: The assigned **handler** *shall receive* the **customer's contact information** to *arrange the handover*.
- **FR3.8.8**: The **customer** *shall receive* a **final confirmation** when a **handler** is assigned.

- **FR3.8.9**: The system *shall update* the **vehicle's status** to “**reserved”** for the selected **date range**.
- **FR3.8.10**: All **booking statuses** (e.g., *pending, approved, accepted*) *shall be recorded* and *quarriable*.

3.9 ADMIN DASHBOARD AND ROLE-BASED ACCESS

3.9.1 Description and Priority

This feature provides a secure, **role-based administrative interface** for managing the **platform’s core data** and **workflows**. Depending on their **role**, system **users** will be *granted access* to specific **modules** within the **dashboard**.

The system supports the following **user roles**:

- **Admin** – Full system **control** and **configuration** authority
- **Fleet Manager** – Manages **vehicle records**, **availability**, and **maintenance**
- **Handler** – Coordinates **pickup/drop-off** operations and accepts **bookings**
- **Legal Verifier** – Reviews and validates **documentation** and **credentials**
- **Accountant** – Manages **billing**, **payments**, and **financial** reporting
- **Customer** – End **user** who can **book vehicles** and manage personal **bookings**

The **Admin Dashboard** is a **critical, high-priority feature** that serves as the **control panel** for *all roles except* the **Customer**.

3.9.2 Stimulus/Response Sequences

- **Stimulus**: A **registered** and **authenticated user** *logs* into the system.
- **Expected Future Response**:
  - The system *identifies* the **user's role** and *displays* only the relevant **dashboard modules**.
  - Each **module** *enables access* to the specific **data** and **functions** allowed for that **role**.
  - *Unauthorized access* to **higher-privilege modules** is blocked and logged.

3.9.3 Functional Requirements

- **FR3.9.1**: The system *shall implement* **role-based access control** for all **authenticated users**.
- **FR3.9.2**: The system *shall display* a **customized dashboard interface** depending on the **user's role**.
- **FR3.9.3**: The **Admin** *shall have access* to all **modules**, including *user management*, *booking approvals*, *vehicle records*, *financial reports*, and *system settings*.
- **FR3.9.4**: The **Fleet Manager** *shall have access* to ***vehicle inventory***, **status updates**, **maintenance records**, and **availability**.
- **FR3.9.5**: The **Handler** *shall receive* **notifications** of **approved bookings** and *be able to accept/decline* assignments.
- **FR3.9.6**: The **Legal Verifier** *shall have access* to **booking documents**, **driver licenses**, and **verification history**.
- **FR3.9.7**: The **Accountant** *shall view* and *manage* **financial records**, **payment transactions**, and **billing history**.
- **FR3.9.8**: The system *shall hide and restrict* access to **modules** that fall outside a **user's** assigned **role**.
- **FR3.9.9**: The system *shall log* all **role-specific** actions in an **audit trail** for accountability.
- **FR3.9.10**: *Unauthorized access* attempts *shall be blocked* and *flagged* as **security alerts**.

3.10 VEHICLE MANAGEMENT, MAINTENANCE, AND INSURANCE TRACKING

3.10.1 Description and Priority

This feature enables **authorized users** to manage the **vehicle inventory**, ***update vehicle details***, and track both **maintenance records** and **insurance status**. While the **Admin role** has full access, **Fleet Managers** are *primarily responsible* for the *operational oversight* of **vehicles** (*adding*, *removing*, *updating*), as well as ensuring they are **properly maintained**, **insured**, and **available for booking**.

This is a **high-priority feature**, essential for maintaining a reliable and safe rental fleet.

3.10.2 Stimulus/Response Sequences

- **Stimulus**: A **Fleet Manager** *logs* in and *navigates* to the "**Vehicles**" module in the dashboard.

- **Expected Future Response**:
  - The system *displays a list* of all **vehicles**, along with their **status** (*available, reserved, under maintenance, etc.*).
  - The **user** can **add new vehicle entries**, **update existing records**, or **retire vehicles**.
  - **Maintenance events** (*e.g., oil change, tire replacement*), **insurance information** (*e.g., policy number, expiry date*), and **registration** papers all can be *logged* and *tracked*.
  - The system *may send* **notifications** for **upcoming maintenance** or **expiring insurance/registration**.

3.10.3 Functional Requirements

- **FR3.10.1**: The system *shall allow* **Admins** and **Fleet Managers** to **add, edit, or delete vehicle records**.

- **FR3.10.2**: The **vehicle record** *shall include* **attributes** such as: *make, model, year, fuel type, seating capacity, transmission, daily rate, and image*.

- **FR3.10.3**: The system *shall support setting* the **availability status** of each **vehicle** (*e.g., available, reserved, out-of-service*).

- **FR3.10.4**: The system *shall allow* **authorized users** to *log* **maintenance activities** for each **vehicle**, including *date, type of maintenance, and responsible technician*.

- **FR3.10.5**: The system *shall allow tracking* of **insurance policies**, including *provider, coverage details, and expiry date*.

- **FR3.10.6**: The system *shall notify* **Fleet Managers** and **Admins** of upcoming **insurance renewals** and **scheduled maintenance**.

- **FR3.10.7**: The system *shall allow* **deactivating vehicles** that are *permanently out of service* while *retaining* their **history**.

- **FR3.10.8**: The system *shall allow* **Legal Verifiers** to *view* **insurance documentation** for compliance purposes.

- **FR3.10.9**: The system *shall log* all **vehicle-related** *edits* and **maintenance** *actions* in the **audit log**.

3.11 PAYMENT AND BILLING MODULE

3.11.1 Description and Priority

This feature handles the **calculation**, **tracking**, and **management** of **rental payments** and **financial transactions**. While **customers** see **pricing information** and **billing summaries** during the **booking** process, **Accountants** are responsible for *reviewing transactions*, *generating reports*, and *managing financial records*.

Currently, *no* **payment gateway** is integrated in the **front-end prototype**, but this **module** is *planned as a future extension*. It is considered a **high-priority** feature due to its impact on **business operations** and **revenue flow**.

3.11.2 Stimulus/Response Sequences

- **Stimulus**: A **customer** *submits* a confirmed **booking**.
- **Expected Future Response**:
    - The system *calculates* the total **rental cost** based on **vehicle rate** and **rental duration**.
    - The **customer** is *shown* a detailed **billing summary** and offered **payment options**.
    - Once the booking is finalized, the transaction is *recorded and made accessible* to the **Accountant** for **review** and **reporting**.

3.11.3 Functional Requirements

- **FR3.11.1**: The system *shall calculate* the **total rental cost** using the **vehicle's rate** and selected **date range**.
- **FR3.11.2**: The system *shall display* a **billing summary** to the **customer** *before confirming* the **booking**.
- **FR3.11.3**: The system *shall record* each **confirmed transaction** along with **booking details** and **timestamps**.
- **FR3.11.4**: The system *shall allow* **Accountants** to *access* a **secure dashboard** for *viewing transactions*, *generating financial reports*, and *auditing history*.
- **FR3.11.5**: The system *shall support* **future integration** with online **payment gateways** (*e.g., PayPal, Visa, Reflect, Jawwal Pay ...etc.*).
- **FR3.11.6**: The system *shall mark* **bookings** as **"unpaid"**, **"paid online"**, or **"cash on pickup"**, *depending on the selected method*.
- **FR3.11.7**: The system *shall restrict access* to **financial data** to only **Admins** and **Accountants**.
- **FR3.11.8**: The system *shall store* all **billing** and **payment data** securely, in compliance with **financial data** protection standards.
- **FR3.11.9**: The system *shall allow* **printing** or **exporting** of *receipts* and *invoices* for each completed **transaction**.

3.12 BUSINESS AND FINANCIAL REPORTING MODULE

3.12.1 Description and Priority

This feature provides the system's **authorized users** with the ability to *generate, view, and export* a variety of **business intelligence** and **financial reports**. It is intended primarily for **Admins** and **Accountants**, but may also include **role-specific dashboards** for **Fleet Managers**, **Legal Verifiers**, and **Handlers**.

This feature is **medium-to-high priority**, supporting strategic decision-making and operational transparency across the organization.

3.12.2 Stimulus/Response Sequences

- **Stimulus**: An **Admin** or **Accountant** *accesses* the "**Reports**" **module** from the **dashboard**.

- **Expected Future Response**:
  - The system *displays a list* of **available reports** based on the **user's role**.
  - **Users** can *filter* by **date range**, **category** (*e.g.*, *bookings*, *payments*, *vehicle status*), or **branch**.
  - The **report** is *generated* and *displayed*, with the option to **download** or **print** it.

3.12.3 Functional Requirements

- **FR3.12.1**: The system *shall provide* a **reporting dashboard** accessible to **Admins** and **Accountants**.

- **FR3.12.2**: The system *shall allow* **users** to *generate* **financial reports**, including:
  - Total **revenue** over time
  - **Payment method** breakdown
  - **Outstanding/unpaid** transactions

- **FR3.12.3**: The system shall support **operational reports** such as:
  - **Active** and **past bookings**
  - **Vehicle availability** and utilization
  - **Maintenance history** and costs
  - Pending **approvals** (*bookings, documents*)

- **FR3.12.4**: The system *shall allow* **Fleet Managers** to *generate* **reports** on **vehicle status** and **performance**.
- **FR3.12.5**: The system *shall allow* **Legal Verifiers** to view **documentation status** and **compliance reports**.
- **FR3.12.6**: **Reports** *shall be filterable* by **date range**, **branch**, **user type**, or **category**.
- **FR3.12.7**: The system *shall allow* **users** to **export reports** in *PDF* or *CSV* format.
- **FR3.12.8**: The system *shall ensure* that **users** only see **reports** permitted by their **assigned role**.
- **FR3.12.9**: The system *shall archive* all generated **reports** for **future reference and auditing**.

3.13 DOCUMENT UPLOADING AND VERIFICATION

3.13.1 Description and Priority

This feature allows **users** to upload required **documents** during the **booking** or **registration** process. Examples include **driver's licenses**, **ID cards**, and **insurance agreements**.

Uploaded **documents** are reviewed by **Legal Verifiers**, who *approve* or *reject* them based on **internal policies**. This is a **high-priority** feature in systems that require **legal compliance** before **confirming rentals**.

3.13.2 Stimulus/Response Sequences

- **Stimulus**: A **customer** is prompted to *upload* a **document** while **booking** a car or completing their **profile**.
- **Expected Future Response**:
  - The system *presents a secure* **upload interface** for *selecting* and submitting **files**.
  - Uploaded **documents** are *linked* to the **user** or **booking record** and set as "**pending review**."
  - **Legal Verifiers** are *notified* of new **submissions** and can *view*, *approve*, or *reject* them.
  - Once **approved**, the **document's status** changes to "**verified**" and is marked as *valid for future use*.

3.13.3 Functional Requirements

- **FR3.13.1**: The system *shall allow* **Customers** to upload one or more required **documents** (*e.g., ID, license*) in **supported formats** (*e.g., PDF, JPG, PNG*).
- **FR3.13.2**: The system *shall associate* each **uploaded document** with a **user profile** or **specific booking**.
- **FR3.13.3**: The system *shall notify* **Legal Verifiers** when a new **document** is *submitted*.
- **FR3.13.4**: **Legal Verifiers** *shall be able* to **view, approve, or reject** each **document submission**.
- **FR3.13.5**: The system *shall record* the **status** of each **document** (*e.g., pending, verified, rejected*).
- **FR3.13.6**: **Rejected documents** *shall include* a **reason field** visible to the **customer**.
- **FR3.13.7**: **Verified documents** *shall be stored securely* and *made available* for **future rentals**.
- **FR3.13.8**: The system *shall restrict access* to **sensitive document** data to only **Admins** and **Legal Verifiers**.
- **FR3.13.9**: The system *shall comply* with applicable **data privacy** and **retention policies**.

4 EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACES

This system presents a **web-based**, *Arabic-language* user interface that supports a **responsive layout** for both ***desktop*** and ***mobile*** devices. The current scope includes a *single-page* prototype (**home page**)

The user interface is implemented using **HTML5**, **CSS3**, **Bootstrap 5 RTL** framework, and **Font Awesome** for icons.

The interface includes the following **visual** components:

- **Top Navigation Bar:** Enables navigation to sections like **Home**, **Cars**, **Branches**, **Offers**, and **Contact**.
- **Hero Section:** Promotes the **service** with a large *headline* and a *booking* button.
- **Booking Form:** Allows users to select **pickup/drop-off** locations and **dates**.
- **Car Listings:** Displays **Cars** with labeled **badges** (e.g., "عرض خاص"), specifications (*fuel type*, *seating*, *transmission*), and *pricing*.
- **Testimonials:** Showcases **client feedback** and **ratings**.
- **Footer:** Provides *business information*, *social media links*, and *quick access links*.

All navigation links and **action buttons** are currently **static** and redirect to **placeholder anchors** (*#: the home page*).

4.2 HARDWARE INTERFACES

The system does **not require** interaction with any specialized **external hardware**.

The application is a **front-end**-only solution, accessible via standard **web browsers** on *desktop computers***,** *laptops, tablets***,** *and smartphones*. No **minimum** hardware specifications are required beyond support for modern **HTML5/CSS3** and **JavaScrip**t standards.

The prototype is hosted on **GitHub Pages**, and there is currently no server-side infrastructure or **back-end** processing.

4.3 4.3 SOFTWARE INTERFACES

The interface utilizes the following **front-end** libraries and **frameworks**:

- **Bootstrap 5 RTL:** Provides responsive UI components and Arabic right-to-left layout support.
- **Font Awesome:** Enables the use of scalable icons throughout the user interface.
- **HTML5 and CSS3:** Core web technologies for layout, structure, and styling.

Future versions may integrate **back-end** services and **databases**, which would introduce additional **software interfaces**.

5 NON-FUNCTIONAL REQUIREMENTS

This section outlines the **quality attributes**, **constraints**, and **expectations** that define how the **Syarti** system should perform. These are critical to ensure **reliability**, **accuracy**, **usability**, and **security** of the system, especially for **internal simulation**

5.1 PERFORMANCE REQUIREMENTS

- The system is a **web-based responsive application** accessible on both desktop and mobile platforms.
- As this is a **front-end prototype**, the system *will not be subjected* to real-world load; hence, *no live* **concurrent users**.
- **Real-time fleet** operations (like *adding, updating, booking, and returning cars*) must occur without **noticeable delay**, prioritizing **accuracy** over speed.
- **Page** and **component interactions** (e.g., *car booking, form submissions*) are expected to respond **instantly (within 1 second)** in development/test environments.

5.2 SAFETY REQUIREMENTS

- **Auto-save functionality** *is expected* when handling data-entry forms (e.g., *booking, car info updates*).
- A **daily backup mechanism** *is expected* to be **simulated** as a critical **safety routine** for **data preservation**.
- The system *must be able to recover* from **crashes** or **power failures**, ensuring minimal **data loss** during operations.
- The system *must maintain* **data consistency** after unexpected **interruptions** (e.g., *transactions must be atomic* when **booking** or **returning cars**).

5.3 SECURITY REQUIREMENTS

- **Basic authentication** required for all roles (**Admin**, **Customer**, **Accountant**, **Legal Verifier**, **Fleet Manager**, **Pickup/Drop-off Handler**).
- **Password hashing** should be considered for *future implementation*.
- Sensitive data such as **payment information** and **driver license numbers** should be identified and marked for **future encryption**

- Each user role should have **clearly defined permissions**, such as:
  - ***Customers*:** browse, book, return cars.
  - ***Admin*:** full access.
  - ***Fleet Manager*:** manage cars.
  - ***Legal Verifier*:** verify documents.
  - ***Accountant*:** view transactions.
  - ***Pickup/Drop-off Handler*:** manage car handovers.

5.4 SOFTWARE QUALITY ATTRIBUTES

5.4.1 Reliability

- The system *should run without failures* during a complete **rental transaction cycle** (*book, return, record update*).
- All system features *should remain functional* after standard **user actions**.

5.4.2 Maintainability

- The system **code** *should be* **modular** and **documented** to allow easy **future improvements**.
- Source **code** *should follow consistent* **naming conventions** and **comments** for readability.

5.4.3 Scalability

- The system *is not intended to scale* for high **user traffic** but must support **logical extension** (e.g., *adding new user roles or views*).

5.4.4 Portability

- The system *must be accessible* across **different devices** (*desktops, tablets, smartphones*) using modern browsers.

5.4.5 Usability

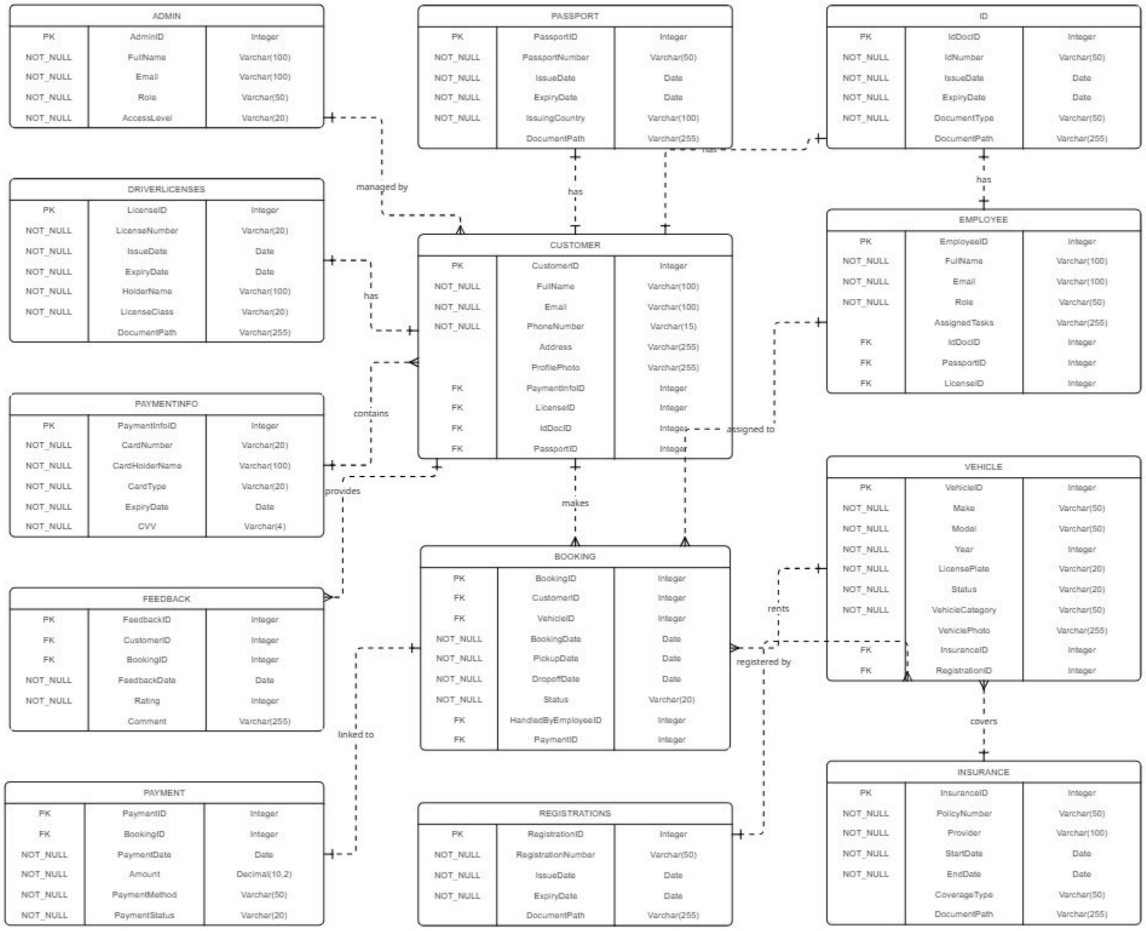
- The system *should feature* a **clean**, **intuitive interface** for all **users**.
- **Users** *should be able to learn* **basic operations** (*booking, managing cars, reviewing status*) without external guidance.

5.4.6 Business Rules

- **Users** *may rent* **multiple vehicles** simultaneously.
- **Late returns** *incur* additional **charges** based on *extended time*.
- **Users** with a **history** of **late returns** are *deprioritized* during **booking conflicts** (added to a *low-priority list*).
- There are *no rules on* **minimum** or **maximum rental durations**.

6 DIAGRAMS

6.1 ENTITY RELATIONSHIP DIAGRAM (ERD)



ADMIN		
PK	AdminID	Integer
NOT_NULL	FullName	Varchar(100)
NOT_NULL	Email	Varchar(100)
NOT_NULL	Role	Varchar(50)
NOT_NULL	AccessLevel	Varchar(20)

PASSPORT		
PK	PassportID	Integer
NOT_NULL	PassportNumber	Varchar(50)
NOT_NULL	IssueDate	Date
NOT_NULL	ExpiryDate	Date
NOT_NULL	IssuingCountry	Varchar(100)
	DocumentPath	Varchar(255)

ID		
PK	IdDocID	Integer
NOT_NULL	IdNumber	Varchar(50)
NOT_NULL	IssueDate	Date
NOT_NULL	ExpiryDate	Date
NOT_NULL	DocumentType	Varchar(50)
	DocumentPath	Varchar(255)

DRIVERLICENSES		
PK	LicenseID	Integer
NOT_NULL	LicenseNumber	Varchar(20)
NOT_NULL	IssueDate	Date
NOT_NULL	ExpiryDate	Date
NOT_NULL	HolderName	Varchar(100)
NOT_NULL	LicenseClass	Varchar(20)
	DocumentPath	Varchar(255)

CUSTOMER		
PK	CustomerID	Integer
NOT_NULL	FullName	Varchar(100)
NOT_NULL	Email	Varchar(100)
NOT_NULL	PhoneNumber	Varchar(15)
	Address	Varchar(255)
	ProfilePhoto	Varchar(255)
FK	PaymentInfoID	Integer
FK	LicenseID	Integer
FK	IdDocID	Integer
FK	PassportID	Integer

EMPLOYEE		
PK	EmployeeID	Integer
NOT_NULL	FullName	Varchar(100)
NOT_NULL	Email	Varchar(100)
NOT_NULL	Role	Varchar(50)
	AssignedTasks	Varchar(255)
FK	IdDocID	Integer
FK	PassportID	Integer
FK	LicenseID	Integer

PAYMENTINFO		
PK	PaymentInfoID	Integer
NOT_NULL	CardNumber	Varchar(20)
NOT_NULL	CardHolderName	Varchar(100)
NOT_NULL	CardType	Varchar(20)
NOT_NULL	ExpiryDate	Date
NOT_NULL	CVV	Varchar(4)

VEHICLE		
PK	VehicleID	Integer
NOT_NULL	Make	Varchar(50)
NOT_NULL	Model	Varchar(50)
NOT_NULL	Year	Integer
NOT_NULL	LicensePlate	Varchar(20)
NOT_NULL	Status	Varchar(20)
NOT_NULL	VehicleCategory	Varchar(50)
	VehiclePhoto	Varchar(255)
FK	InsuranceID	Integer
FK	RegistrationID	Integer

FEEDBACK		
PK	FeedbackID	Integer
FK	CustomerID	Integer
FK	BookingID	Integer
NOT_NULL	FeedbackDate	Date
NOT_NULL	Rating	Integer
	Comment	Varchar(255)

BOOKING		
PK	BookingID	Integer
FK	CustomerID	Integer
FK	VehicleID	Integer
NOT_NULL	BookingDate	Date
NOT_NULL	PickupDate	Date
NOT_NULL	DropoffDate	Date
NOT_NULL	Status	Varchar(20)
FK	HandledByEmployeeID	Integer
FK	PaymentID	Integer

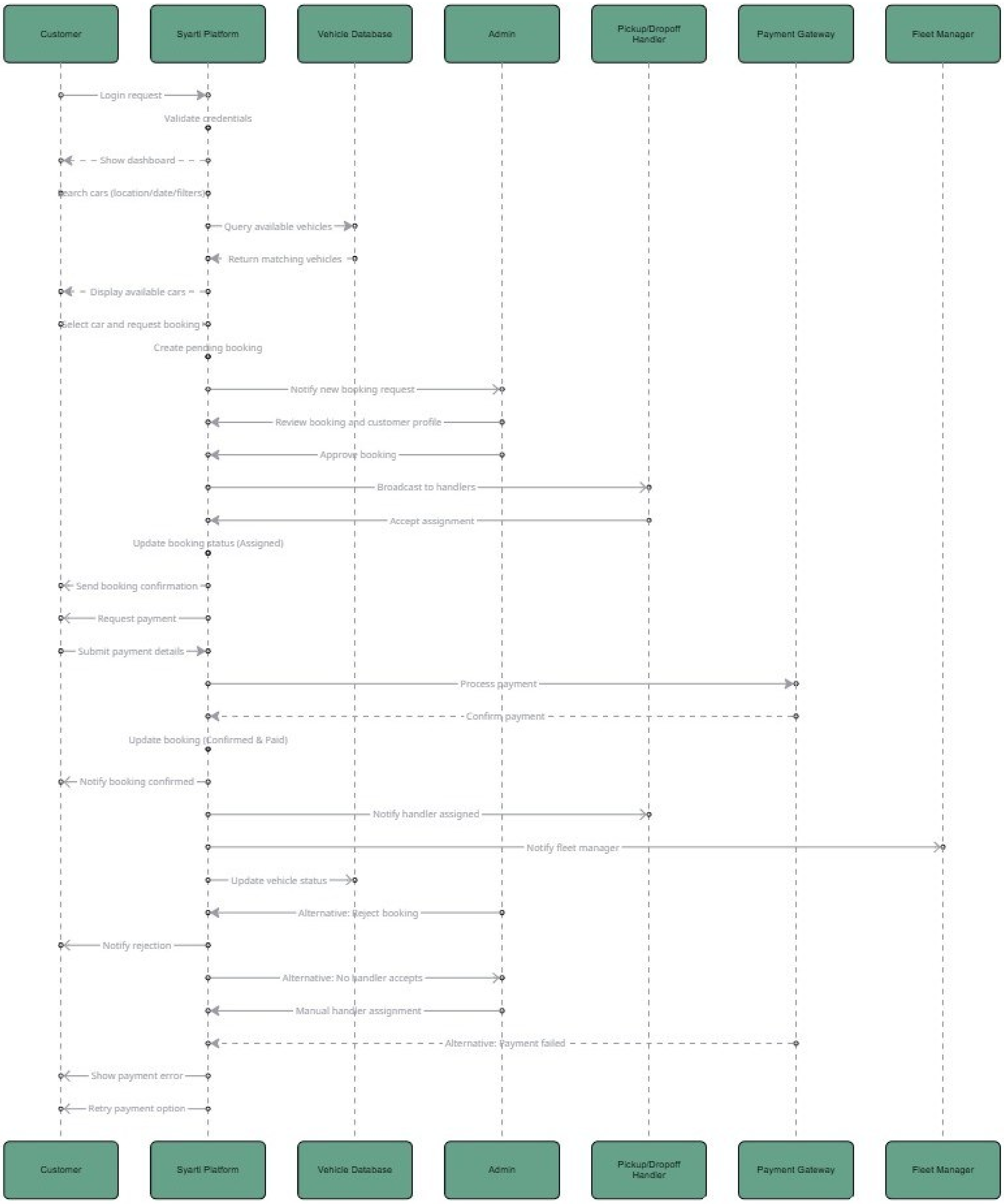
PAYMENT		
PK	PaymentID	Integer
FK	BookingID	Integer
NOT_NULL	PaymentDate	Date
NOT_NULL	Amount	Decimal(10,2)
NOT_NULL	PaymentMethod	Varchar(50)
NOT_NULL	PaymentStatus	Varchar(20)

REGISTRATIONS		
PK	RegistrationID	Integer
NOT_NULL	RegistrationNumber	Varchar(50)
NOT_NULL	IssueDate	Date
NOT_NULL	ExpiryDate	Date
	DocumentPath	Varchar(255)

INSURANCE		
PK	InsuranceID	Integer
NOT_NULL	PolicyNumber	Varchar(50)
NOT_NULL	Provider	Varchar(100)
NOT_NULL	StartDate	Date
NOT_NULL	EndDate	Date
	CoverageType	Varchar(50)
	DocumentPath	Varchar(255)

Syarti ER-Diagram

6.2 SEQUENCE DIAGRAM



Syarti Sequence Diagram