Software Requirements

Specification for

CarSync

Version 1.0

Group No.: 8

Kashvin A/L Muthu Mani	**243UC2462Z**
George	
Muhammad Fahmi Aiman bin Mohd Fauzi	**243UC24635**
Lee Kah Wai	**243UC247K0**

Date: <14/12/2025>

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1	Kashvin, Kah Wai, Fahmi	Initial brainstorming and stakeholder identification.	10/11/2025
2	Kashvin, Kah Wai, Fahmi	System Overview, Gantt Chart creation, Problem Statement.	17/11/2025
3	Kashvin, Kah Wai, Fahmi	Use Case formation and description, Use Case diagram creation, and Use Case table completion.	24/11/2025
4	Kashvin, Kah Wai, Fahmi	Class Diagram Creation, Relationship Identification and Non-Functional Requirements completion.	1/12/2025
5	Kashvin, Kah Wai, Fahmi	Class Diagram correction, formatting correction.	8/12/2025
6	Kashvin, Kah Wai, Fahmi	Full document review: corrected inconsistencies, references, completed references section in APA7, and formatted all tables.	12/12/2025
7	Kashvin, Kah Wai, Fahmi	Final version for submission. All diagrams included (Class, Use Case, Gantt), revisions table completed, formatting polished, and document proofread.	13/12/2025

1 Project Introduction

The purpose of CarSync is to provide a centralised digital platform that streamlines the vehicle maintenance and care process. It allows for the connection between vehicle owners and service workshops to ease appointment bookings, service tracking, and efficient maintenance management.

Currently, workshops still use a manual logging and appointment system. This makes tracking services and viewing future appointments tedious and can lead to scheduling conflicts. Due to this, there exists an inefficient ecosystem that lacks transparency.

There are some key problems that need to be addressed. From a vehicle owner’s perspective, issues such as lost maintenance records, booking inconveniences, and missed vehicle services come to mind. On the contrary, a mechanic or a workshop suffers from operational inefficiency, lack of customer feedback, and limited communications. Carsync directly addresses these issues by digitalising and integrating the long, tedious processes into a single, user-friendly platform.

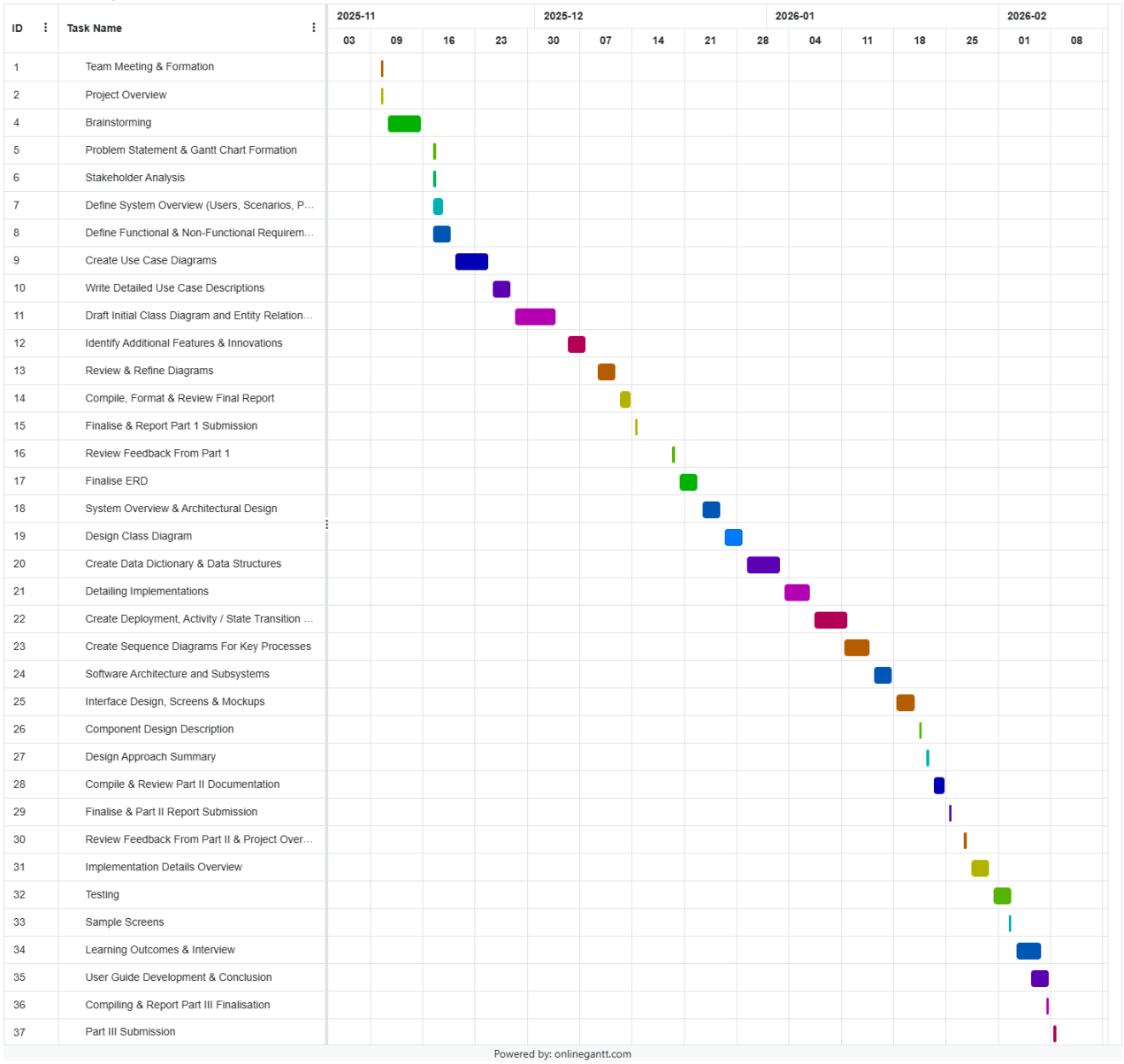
1.1 Team Members

Name	Actor/Processes
KASHVIN GEORGE	VEHICLE OWNER
Muhammad Fahmi Aiman	MECHANIC/WORKSHOP
LEE KAH WAI	ADMIN

1.2 Problem Statement

Information about vehicle maintenance is frequently documented via paper receipts, handwritten notes, or casual messages exchanged between owners and repair shops. As a result, service records become chaotic, hard to access, and easily misplaced. Owners will also find it difficult to recall previous services and future service dates, while workshops encounter challenges handling bookings and consistently documenting service information. The absence of a unified digital system results in ineffective communication, lost records, and limited visibility of overall vehicle maintenance operations.

1.3 Project Schedule



ID	Task Name
1	Team Meeting & Formation
2	Project Overview
4	Brainstorming
5	Problem Statement & Gantt Chart Formation
6	Stakeholder Analysis
7	Define System Overview (Users, Scenarios, P…
8	Define Functional & Non-Functional Requirem…
9	Create Use Case Diagrams
10	Write Detailed Use Case Descriptions
11	Draft Initial Class Diagram and Entity Relation…
12	Identify Additional Features & Innovations
13	Review & Refine Diagrams
14	Compile, Format & Review Final Report
15	Finalise & Report Part 1 Submission
16	Review Feedback From Part 1
17	Finalise ERD
18	System Overview & Architectural Design
19	Design Class Diagram
20	Create Data Dictionary & Data Structures
21	Detailing Implementations
22	Create Deployment, Activity / State Transition …
23	Create Sequence Diagrams For Key Processes
24	Software Architecture and Subsystems
25	Interface Design, Screens & Mockups
26	Component Design Description
27	Design Approach Summary
28	Compile & Review Part II Documentation
29	Finalise & Part II Report Submission
30	Review Feedback From Part II & Project Over…
31	Implementation Details Overview
32	Testing
33	Sample Screens
34	Learning Outcomes & Interview
35	User Guide Development & Conclusion
36	Compiling & Report Part III Finalisation
37	Part III Submission

2 System Overview

2.1 Description

Users:

- Vehicle Owner: Registers vehicles, books services, views service history, and manages appointments.
- Mechanic/Workshop: Manages bookings, records services, generates invoices, and views workshop reports.
- Admin: Oversees system users, generates reports, and maintains platform integrity.

Scenario:
A vehicle owner needs to schedule maintenance but faces difficulties tracking service history and finding reliable workshops. Workshops nowadays struggle with manual booking systems and inefficient customer communication.

Problems Faced:

- Lost or disorganised maintenance records
- Inefficient manual booking and scheduling
- Poor communication between owners and workshops
- Lack of workshop performance insights

Key Processes:
1.User Registration and Authentication – Users can create accounts and log in to the system. The system verifies credentials and provides access based on roles.
(Vehicle Owner, Admin, Mechanic)

2.Service Booking and Scheduling – Vehicle owners can look for workshops and book service appointments. The mechanic can then view these bookings and either approve or reschedule the booking
(Vehicle Owner, Mechanic)

3.Maintenance Record Management – After a service is completed the mechanic can update the vehicle's record digitally through the app. This includes logging the service type, cost, time taken, parts used and date of service.
(Vehicle Owner, Mechanic)

4.Notification Management – System monitors mileage and time since previous service and provides a notice to alert the owner of upcoming services and maintenance tasks. (Vehicle Owner, Admin (automated))

5.Invoice and Payment History – System allows the workshop to issue digital invoices after completion of service. Vehicle owner can view, download and track payment history and status (Vehicle Owner, Mechanic)

6.Workshop Dashboard (with service statistics) – Provides mechanic with an overview of their business operations, such as number of appointments, services completed, customer feedback, revenue and popular services

(Mechanic)

7.Admin Report Generation and System Maintenance – Admin generates system reports such as number of users, active users, active workshops, overall platform usage statistics. Admin can also block or suspend users from the platform to maintain overall quality of the app. (Admin)

2.2 Actors

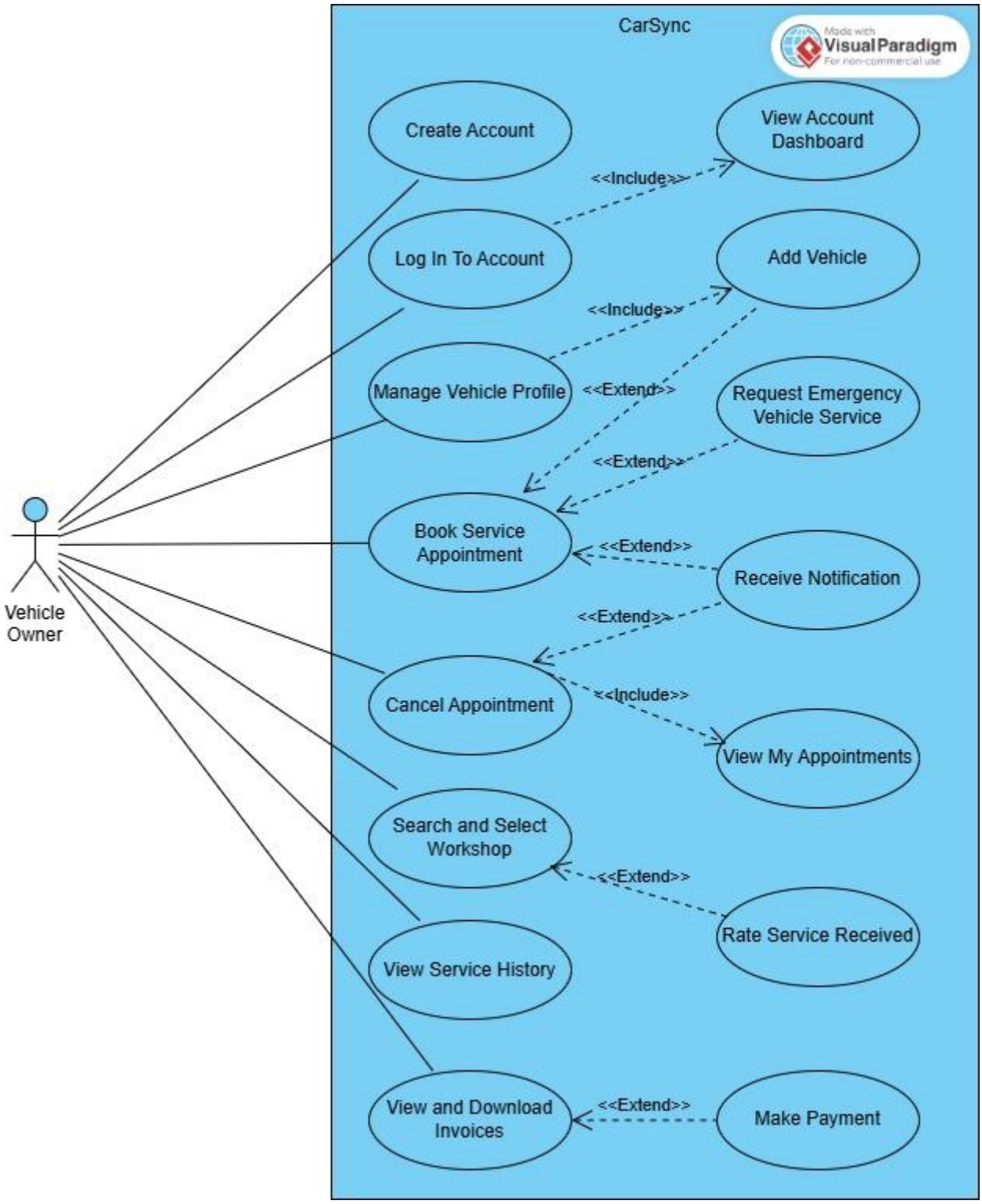
Actor	Use Cases
Vehicle Owner	1.Create Account 2.Log In to Account 3.Manage Vehicle Profile 4.Search and Select Workshop 5.Book Service Appointment 6.Cancel Appointment 7.View Service History 8.View and Download Invoices 9.View Account Dashboard (<<include>>) 10.Add Vehicle (<<include>> & (<<extend>>)) 11.Request Emergency Vehicle Service (<<extend>>) 12.Receive Notifications (<<extend>>) 13.View My Appointments (<<include>>) 14.Make Payment (<<extend>>) 15.Rate Service Received (<<extend>>)
Mechanic/Workshop	1. View and Manage Booking 2. Record Service Details 3. Generate Invoices 4. Record Service Details 5. View Workshop Report 6. Manage Vehicle Profile 7. Approve or Reject Booking (<<include>>) 8. Send Notification (<<include>>) 9. Check Vehicle History (<<extend>>)
Admin	1.Login / Authenticate 2. Manage User Accounts 3.Generate System Reports 4.View System Logs 5.Configure Notification Rules 6.Suspend/Block Users (<<extend>>) 7.Monitor Platform Usage (<<include>>) 8.Maintain System Data (<<include>>)

2.3 Assumptions and Dependencies

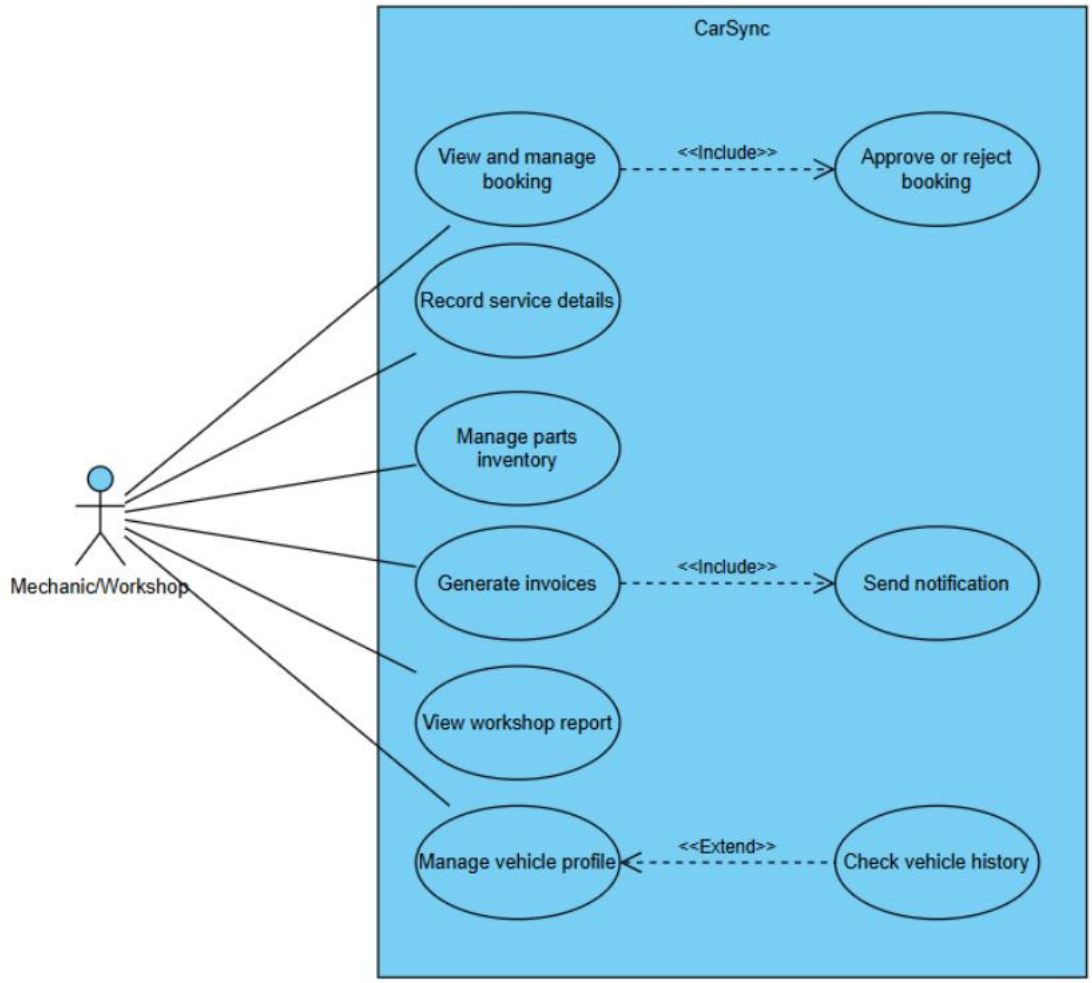
- It is assumed that all users will have internet access to use the system
- The system assumes that only authorized mechanics and workshops are registered and verified by the admin
- It is assumed that all vehicle owners will provide accurate/reasonable vehicle and mileage information
- The system depends on a functioning database server to store all maintenance records, bookings, and user information
- It is assumed that notification features will be simple (dashboard reminders) without requiring external SMS or email gateways (apart from first time user verification).
- For this project and prototype, it is assumed the system will handle a user base of approximately 50–100 active users and is not currently optimized for high-traffic enterprise loads.
- It is assumed that the Admin will manually verify the certification documents of workshops during the registration phase. The system does not automatically validate business licenses with external government databases.
- The "Locate Workshop" feature depends on the availability of an external API (such as Google Maps API or OpenStreetMap) to display workshop locations accurately.

2.4 Use Case Diagram

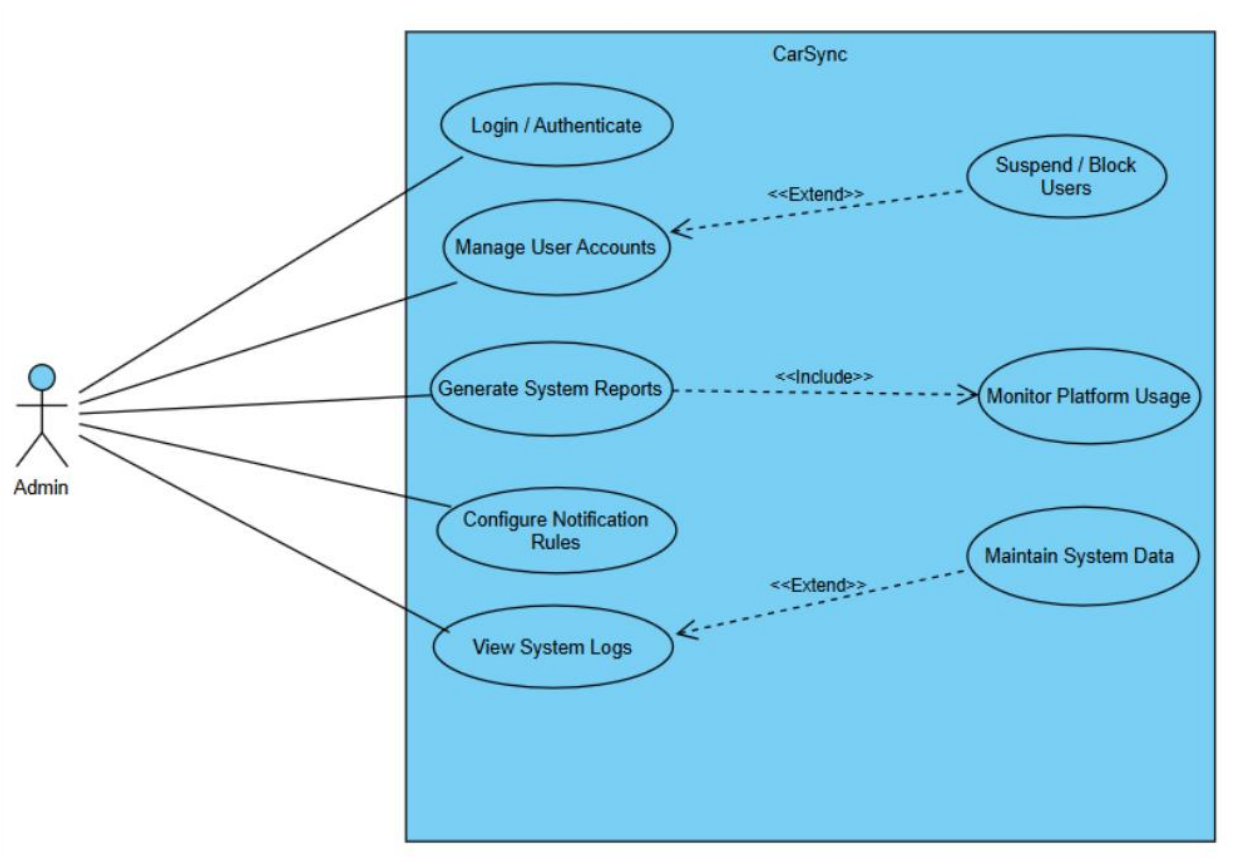
Vehicle Owner



Mechanic/Workshop



Admin


CarSync
Login / Authenticate
Suspend / Block Users
<<Extend>>
Manage User Accounts
Generate System Reports
<<Include>>
Monitor Platform Usage
Admin
Configure Notification Rules
Maintain System Data
<<Extend>>
View System Logs

3 Functional Requirements

3.1 Vehicle Owner

3.1.1 Create Account

Use Case Name		Create Account
Actor: Vehicle Owner		A new user creates an account in the system to access its features.
Preconditions		1. User is not logged in to the system. 2.User does not have an existing account.
Normal Flow	Description	1.User navigates to system's registration page. 2.User selects registration method (email or phone number). 3.User enters required personal data. (full name, address, identity card etc). 4.User provides credentials. (email, password, phone number for OTP). 5.System validates the credentials. 6.System creates new VehicleOwner account. 7.System sends confirmation email or OTP to the registered address or phone number. 8.The system displays a success message and redirects user to log in page.
	Postconditions	1.A new VehicleOwner account is created and stored into the database. 2.User authenticated and has an active session. 3.User can now proceed to log in.
Alternative Flow & Exceptions		**1.Duplicate Credential:** If provided email or phone number is already registered previously, system displays an error message and prompts user to enter an alternative credential. **2.Weak Password:** If chosen password does not meet security requirements, the system rejects it and prompts the user to create a stronger password. **3.OTP Verification Failure:** If OTP is invalid or expired system prompts user to request new OTP.
Non-Functional Requirements		**Performance**: Account creation must complete within 5 seconds of normal load. **Security**: Email/Phone verification must be implemented.

3.1.2 Log In to Account

Use Case Name		Log In to Account
Actor: Vehicle Owner		User enters their credentials to access their account and system features.
Preconditions		1. User has a registered account. 2.System authentication service is operational.
Normal Flow	Description	1.User navigates to login page. 2.User enters their registered email/phone number. 3.User clicks “Log In”. 4.System verifies credentials. 5.System creates a new session and redirects user to their account dashboard.
	Postconditions	1.User is authenticated and has an active session. 2.User has access to all authorized features.
Alternative Flow & Exceptions		**1.Forgot Password:** User clicks “forgot password”, enters email, receives reset link and follows it to create a new password. **2.Account Locked:** After 5 failed attempts, the account is locked temporarily; system displays a message that suggests password change. **3.Invalid Credentials:** System displays “Invalid email or password” and returns to login form.
Non-Functional Requirements		**Performance:** Authentication must be completed within 20 seconds. **Security:** Account lockout after 5 failed attempts.

3.1.3 Manage Vehicle Profile

Use Case Name		Manage Vehicle Profile
Actor: Vehicle Owner		User performs CRUD (Create, Read, Update, Delete) operations on their vehicle profile.
Preconditions		1. User is logged in to the system.
Normal Flow	Description	1.User navigates to “My Vehicles”. 2.System lists a display of current registered vehicles. 3.User clicks “Add Vehicle”. 4.User enters vehicle details (Model, Year, Mileage, License Plate etc). 5.User saves details. 6.System validates the details and stores the new vehicle in system database. 7.System confirms addition of new vehicle and updates the list.
	Postconditions	1.A new Vehicle object is created and linked to the owner’s account. 2.Vehicle list is updated.
Alternative Flow & Exceptions		**1.Edit Vehicle:** User selects vehicle, clicks “Edit” and modifies details, then saves modified details. **2.Delete Vehicle:** User selects vehicle, clicks “Delete” and confirms deletion, then system removes it. **3.Already Registered Vehicle:** If vehicle is already registered under another user, system rejects the addition and notifies user. **4.Vehicle With Active Appointments:** System prevents deletion and notifies user to cancel appointments beforehand.
Non-Functional Requirements		**Performance**: Adding/Editing/Deleting vehicle must be processed within 5 seconds. **Security**: Vehicle must be validated against a standard format.

3.1.4 Search & Select Workshop

Use Case Name		Search & Select Workshop
Actor: Vehicle Owner		User searches for workshops based on criteria, view details and selects one.
Preconditions		1. Vehicle Owner is logged in to the system. 2.The system has active workshop listings.
Normal Flow	Description	1.User navigates to “Find Workshop”. 2.User enters search criteria. 3.System returns list of workshops matching selected criteria. 4.User enters additional filters such as ratings or status of workshop. 5.User selects workshop to view its full profile, services, reviews and contact info. 6.User selects “Choose Workshop” to proceed.
	Postconditions	1.A specific workshop is selected for the next process. 2.User’s recent searches may be logged for personalisation purposes.
Alternative Flow & Exceptions		**1.No Results:** If no results match, system suggests user to broaden location or search criteria. **2.Service Unavailable:** If selected workshop does not offer needed service, system prompts user to choose another workshop or service.
Non-Functional Requirements		**Performance**: Search results must appear within 2 seconds. **Usability:** Workshop search results must display key information such as workshop name, rating, distance in a clean and consistent manner. **Reliability:** Workshop search results must reflect real-time workshop availability and service offerings with 98% accuracy.

3.1.5 Book Service Appointment

Use Case Name		Book Service Appointment
Actor: Vehicle Owner		Vehicle Owner selects a vehicle, finds a workshop, chooses a service and time slot then submits a booking request.
Preconditions		1. Vehicle Owner is logged in to the system. 2.Vehicle Owner has at least one vehicle registered to their profile. 3.At least one workshop is listed and active in the system.
Normal Flow	Description	1.User selects the “Book Service” option. 2.System displays a list of the user’s vehicle(s). 3.User selects the vehicle for service. 4.System presents a search interface for workshops. 5.User searches for or selects a workshop. 6.System displays available service types and time slots for the selected workshop. 7.User selects service type and preferred date/time. 8.User receives booking summary, reviews it and confirms. 9.System creates a new Service Appointment that has “Pending” status. 10.System notifies selected workshop of the new booking request.
	Postconditions	1.A new Service Appointment object is created and linked to the user, vehicle and workshop. 2.User has a comprehensive view of their appointment history and future schedule. 3.Appointment status is “Pending” while waiting for workshop approval. 4.Workshop is notified of request.
Alternative Flow & Exceptions		**1.No Vehicle in Profile:** If user has no vehicle in profile, flow extends to “Add Vehicle” before proceeding. **2.Time Slot Unavailable:** If selected time slot is unavailable, system suggests the nearest available alternative. **3.Workshop Rejects:** If workshop rejects the request, system notifies the owner and suggests booking with another workshop. **4.Session Timeout:**

	If user's session expires mid process, system saves progress and redirects to login page and restores progress after authentication.
Non-Functional Requirements	**Performance**: Workshop search results must appear within 3 seconds. **Reliability**: Booking submission should have a 99% success rate.

3.1.6 Cancel Appointment

Use Case Name		Cancel Appointment
Actor: Vehicle Owner		Vehicle Owner views their upcoming appointments and cancels a selected one, where they must provide a reason for cancellation.
Preconditions		1. Vehicle Owner is logged in to the system. 2.Vehicle Owner has at least one upcoming Service Appointment with status “Pending” or “Confirmed”.
Normal Flow	Description	1.User navigates to “My Appointments”. 2.System displays a list of upcoming appointments. 3.User selects an appointment to cancel. 4.User clicks “Cancel Appointment”. 5.System prompts for cancellation reason. 6.User confirms cancellation. 7.System updates appointment status to “Cancelled”. 8.System notifies the associated workshop regarding the appointment cancellation.
	Postconditions	1.The selected Service Appointment status is changed to “Cancelled”. 2.Workshop is notified on cancellation. 3.The time slot is freed for other bookings.
Alternative Flow & Exceptions		**1.Last Minute Cancellation:** If the appointment is within 24 hours, the system notifies the user of potential cancellation fee or temporary suspension of account. **2.Appointment in Progress:** If appointment status is already “In Progress”, the system prevents the cancellation request and instructs the user to contact the workshop immediately.
Non-Functional Requirements		**Performance**: Cancellation request must be processed within 3 seconds. **Reliability**: Cancellation must be processed and appear on all involved user and workshop views within 10 seconds.

3.1.7 View Service History

Use Case Name		View Service History
Actor: Vehicle Owner		Vehicle Owner can view chronological list of all past services, including details and costs.
Preconditions		1. Vehicle Owner is logged in to the system. 2.Vehicle Owner has at least one vehicle registered to their profile. 3.The selected vehicle has at least one associated Service Record.
Normal Flow	Description	1.User navigates to “My Vehicles” or dashboard. 2.User selects a specific vehicle from their list. 3.User selects the “Service History” tab. 4.System retrieves all ServiceRecord objects linked to that vehicle. 5.System displays the records in reverse chronological order, showing Date, Workshop, Service Type and Total Cost. 6.The user can select any record to view expanded details (parts used, mechanic notes, invoice link).
	Postconditions	1.The user has viewed the complete service history for selected vehicle. 2.System has logged this view activity. 3.User can access invoices from history.
Alternative Flow & Exceptions		**1.No Service History:** If vehicle has no service records, system displays a message: “No Service History Found”. **2.Data Unavailable:** If database cannot retrieve records, system displays a “Temporarily Unavailable” message.
Non-Functional Requirements		**Performance**: Service history list must load within 2 seconds even with many records (> 50). **Usability**: History must be sortable by date, cost, or service type.

3.1.8 View & Download Invoices

Use Case Name		View & Download Invoices
Actor: Vehicle Owner		Vehicle Owner can access, view and download service invoices.
Preconditions		1. Vehicle Owner is logged in to the system. 2.Vehicle Owner has at least one completed service. 3.At least one completed service with invoice exists.
Normal Flow	Description	1.User navigates to “Service History”. 2.System displays completed invoices in chronological order. 3.User selects a completed service. 4.User clicks “View Invoice”. 5.System displays PDF preview of selected invoice. 6.System serves the PDF file.
	Postconditions	1.The user has viewed/downloaded invoice PDF locally on device. 2.Download is logged in the system. 3.Invoice status may be marked as “Viewed”
Alternative Flow & Exceptions		1.**No Service History:** If vehicle has no service records, system displays a message: “No Service History Found”. 2.**Invoice Pending**: If database cannot retrieve records, system displays a “Invoice in Progress” message. 3.**Pay Invoice:** User can initiate payment and select method of payment (Funds Transfer, Online Banking, etc.)
Non-Functional Requirements		**Performance**: PDF generation within 5 seconds.

3.1.9 View Account Dashboard (<<include>> of Log In)

Use Case Name		View Account Dashboard
Actor: Vehicle Owner		Vehicle Owner can access, view and edit account dashboard.
Preconditions		1. Vehicle Owner is logged in to the system. 2.Vehicle Owner is authenticated.
Normal Flow	Description	1.System automatically redirects to dashboard after successful log in. 2.System displays vehicle(s), next appointment, recent activity. 3.System displays dashboard widgets. 4.User can navigate to and access features from dashboard.
	Postconditions	1.Dashboard is displayed with current user data. 2.Last dashboard access time is recorded.
Alternative Flow & Exceptions		**1.No Vehicles/Appointments:** Dashboard shows empty state messages with "Add Vehicle" and "Book Service" prompts. 2.**Session Expired:** System redirects to log in page.
Non-Functional Requirements		**Performance**: Dashboard loaded within 5 seconds. **Usability:** All dashboard widgets must be visible. **Reliability:** Dashboard data must be accurate.

3.1.10 Add Vehicle (<<include>> of Manage Vehicle Profile and (<<extend>>) of Book Service Appointment)

Use Case Name		Add Vehicle
Actor: Vehicle Owner		Vehicle Owner can add vehicle(s).
Preconditions		1. Vehicle Owner is logged in to the system. 2.Vehicle Owner is on the “Manage Vehicle Profile” page.
Normal Flow	Description	1.User selects “Add New Vehicle”. 2.User enters details of vehicle (make, model, year, mileage). 3.System validates information. 4.System checks for duplicate information. 5.System saves vehicle details.
	Postconditions	1.New Vehicle object is created. 2.Vehicle is linked to owner’s profile. 3.Vehicle information is included in dashboard updates.
Alternative Flow & Exceptions		**1.Duplicate Information:** System displays “This Vehicle is Already Registered”. **2.Invalid Information:** System shows specific format error and provides formatting examples. **3.Database Connection Lost:** System saves data locally and retries when connection is re-established.
Non-Functional Requirements		**Performance**: Vehicle Addition must be processed within 5 seconds. **Security:** Vehicle verification must follow ISO standards. **Data Integrity:** System must prevent duplicate vehicles across all users.

3.1.11 View My Appointments (<<include>> of Cancel Appointment)

Use Case Name		View My Appointments
Actor: Vehicle Owner		Vehicle Owner can access and view their appointments.
Preconditions		1. Vehicle Owner is logged in to the system.
x	Description	1.User navigates to “My Appointments”. 2.System fetches all appointments for the user. 3.System displays all appointments in sections such as “upcoming, past, pending. 4.User can select any appointment for details.
	Postconditions	1.Appointment list is displayed with current status. 2.User can take any action (cancel, modify) from this view.
Alternative Flow & Exceptions		1.F**ilter Appointments:** User can filter by date range, workshop, or service type. 2.No Appointments: System displays “No Appointments Found” with “Book A Service: prompt. 3.Data Inconsistency: If appointment status doesn’t match workshop records, system shows “Verifying Status”.
Non-Functional Requirements		**Performance**: Appointment List must load within 2 seconds for users. **Usability:** Must support sorting data by date, workshop, and service type. **Availability:** Appointment data must be available 99.9% of the time.

3.1.12 Request Emergency Vehicle Service (<<extend>> of Book Service Appointment)

Use Case Name		Request Emergency Vehicle Service
Actor: Vehicle Owner		Vehicle Owner can request for help in case of an emergency.
Preconditions		1.User is logged in to the system. 2.User is in booking flow. 3.User has selected a vehicle.
Normal Flow	Description	1.User checks “Emergency Service” box. 2.User provides emergency details. 3.System prioritises the booking. 4.System notifies workshop immediately.
	Postconditions	1.Booking is marked as “Emergency”. 2.Workshop receives urgent notification.
Alternative Flow & Exceptions		1.**Emergency Service Unavailable:** No workshops available for emergency booking, system suggests normal booking. 2.**High Emergency Fee**: System warns about emergency service charge. 3.**False Emergency Report:** Workshop can report false emergency; system logs and may suspend emergency privileges. 4.**No Workshop Response:** If no workshop response within 10 minutes, system escalates to admin.
Non-Functional Requirements		**Performance**: Emergency notification must reach workshops within 5 seconds. **Reliability:** Emergency requests must have a 99.9% delivery success rate. **Safety:** System must include emergency contact information verification

3.1.13 Receive Notifications (<<extend>> of Book and Cancel Appointment)

Use Case Name		Receive Notifications
Actor: Vehicle Owner		Vehicle Owner can choose to receive notifications.
Preconditions		1. A triggering event occurs. 2. User has notification preferences configured.
Normal Flow	Description	1.System detects completed event. 2.System generates notification message. 3.System sends via preferred channel. 4.User receives notification.
	Postconditions	1.Notification is delivered. 2.Notification is logged with delivery status. 3.User notification count increments.
Alternative Flow & Exceptions		1.**Notification Preferences:** User can customise which event triggers notification. 2**.Do Not Disturb:** System obliges “quiet hour” settings. 3.**Delivery Failure:** After 3 failed attempts, system logs failure and tries alternate channel. 4.**Spam Prevention:** System limits to 10 notifications per hour per user.
Non-Functional Requirements		**Performance**: Notification must be delivered within 30 seconds of event. **Privacy:** Notification content must not include sensitive personal content. **Scalability:** System must support 10,000 concurrent notifications.

3.1.14 Rate Service Received (<<extend>> of View Service History)

Use Case Name		Rate Service Received
Actor: Vehicle Owner		Vehicle Owner can rate service received.
Preconditions		1.User is logged in to the system. 2.User is viewing a completed service record. 3.Service was completed within the last 30 days. 4.Service has not been rated yet.
Normal Flow	Description	1.User clicks “Rate This Service”. 2.User provides star-based rating (1-5). 3.User adds optional comments. 4.User submits rating.
	Postconditions	1.Rating is saved to service record. 2.Workshops average rating updates. 3.Service is marked as “Rated”.
Alternative Flow & Exceptions		1.**Anonymous Rating:** User can choose to rate anonymously. 2.**Edit Rating:** User can edit rating within 7 days of submission. 3.**Inappropriate Content:** System filters and flags offensive comments for review. 4.**Rating Abuse Detection:** System detects and investigates suspicious rating patterns.
Non-Functional Requirements		**Performance**: Rating submission must process within 5 seconds. **Fairness:** Rating system must prevent manipulation and fake reviews. **Usability:** Rating interface must be completable in under 30 seconds.

3.1.15 Make A Payment (<<extend>> of View and Download Invoices)

Use Case Name		Make A Payment
Actor: Vehicle Owner		Vehicle Owner can make a payment through the platform.
Preconditions		1.User is logged in to the system. 2.User is viewing an invoice. 3.Invoice status is “unpaid”. 4.User has payment status configured.
Normal Flow	Description	1.User clicks “Pay Now” on invoice. 2.User selects payment method. 3.User confirms payment amount. 4.System processes payment. 5.System updates invoice status.
	Postconditions	1.Invoice status changes to “Paid”. 2.Payment transaction is recorded. 3.Receipt is generated and sent.
Alternative Flow & Exceptions		1.**Partial Payment:** User can choose to pay partial amount if allowed by workshop. 2.**Payment Gateway Failure:** System saves payment attempt and retries. 3.**Insufficient Funds:** System notifies user and suggests alternate method. 4.**Fraud Detection:** System flags and hold suspicious transactions for review.
Non-Functional Requirements		**Performance**: Payment processing must be completed within 5 seconds. **Reliability:** Payment success rate must be 99.95% or higher. **Security:** Transaction must be safe and secure.

3.2 Mechanic/Workshop

3.2.1 Manage Service Booking

Use case name		Manage Service Booking
Actor		Mechanic/workshop
Preconditions		1. Mechanic is logged into the system. 2. At least one pending booking request exists in the system.
Normal Flow	Description	1. System displays the Booking Management Dashboard with a list of pending and confirmed bookings. 2. Mechanic selects a pending booking request to review. 3. System shows detailed booking information 4. Mechanic checks real-time workshop availability for the requested time slot. 5. If the time slot is available, mechanic approves the booking. 6. System updates the booking status from "Pending" to "Confirmed". 7. System automatically sends a confirmation notification to the customer. 8. System updates the workshop calendar and removes the time slot. 9. Booking appears in the mechanic's scheduled appointments.
	Postconditions	1. Booking status is changed to "Confirmed" in the database. 2. Customer receives booking confirmation with details. 3. Workshop calendar is updated with the new appointment. 4. Time slot is marked as unavailable for other bookings.
Alternative Flow & Exceptions		**Reschedule booking:** 1. If the requested time slot is unavailable, mechanic selects "Reschedule" option. 2. System 2-3 suggests alternative available time slots based on workshop schedule.

	3. System sends a rescheduling time and date to the customer with the alternative options. 4. Customer selects a preferred time. 5. System updates the booking. **Customer Rejects All Alternatives:** 1. Customer rejects all suggested alternative time slots. 2. System cancels the booking request. 3. Booking status changes to "Cancelled by Customer". 4. Time slot remains available for other bookings.
Non-Functional Requirements	**Reliability**: System must maintain 99% uptime. Prevent double-booking with 100% accuracy. **Security**: Only authenticated users can manage bookings. All data is encrypted.

3.2.2 Record Service Details

Use case name		Record Service Details
Actor		Mechanic/workshop
Preconditions		1. Mechanic is logged into the system.
Normal Flow	Description	1. Mechanic selects the ongoing service job from their dashboard. 2. System displays the vehicle details and service type. 3. Mechanic recording service details (update service tasks performed, logs parts used) 4. System validates parts availability and deducts used parts from inventory. 5. Mechanic marks the service as "Completed". 6. System updates the vehicle's service.
	Postconditions	1. Service details are saved and linked to the vehicle's history. 2. Parts inventory is updated. 3. Vehicle's service record updated. 4. Service job status is updated to "Completed".
Alternative Flow & Exceptions		**Additional Repairs Needed:** 1. Mechanic pauses the service recording. 2. System helps create an estimate for additional work. 3. If customer approved, mechanic continues recording the additional service. 4. If customer not approved, mechanic completes only the original service.
Non-Functional Requirements		**Accuracy:** Parts usage must be accurately deducted from inventory. Service history must be consistent with recorded data.

3.2.3 Manage Parts Inventory

Use case name		Manage Parts Inventory
Actor		Mechanic/workshop
Preconditions		1. Mechanic is logged into the CarSync system with inventory management permissions. 2. The workshop has an existing inventory database with current stock levels.
Normal Flow	Description	1. Mechanic navigates to the "Parts Inventory" section. 2. System displays the current inventory list. 3. Mechanic performs action (view inventory, update, add new part, reset reorder alert). 4. System saves changes and updates the inventory database.
	Postconditions	1. Inventory records are updated with the latest stock levels. 2. Any new parts are added to the inventory database. 3. Alerts are generated for low stock parts.
Alternative Flow & Exceptions		**Generate Inventory Report:** 1. Mechanic requests an inventory report. 2. System compiles data and generates a report for review or sharing.
Non-Functional Requirements		**Performance:** Inventory list must load within 3 seconds. Search and filter operations must respond less than 2 second. **Reliability:** Inventory data must be consistent and accurate. Concurrent updates by multiple mechanics must be handled without data corruption.

3.2.4 Generate Invoices

Use case name		Generate Invoices
Actor		Mechanic/workshop
Preconditions		1. Mechanic is logged into the CarSync system. 2. A service job is marked as "Completed" in the system.
Normal Flow	Description	1. Mechanic navigates to the "Invoices" section or selects "Generate Invoice" from a completed service job. 2. System automatically populates the invoice 3. System validates the invoice totals and calculations. 4. Mechanic selects "Issue Invoice" to finalize. 5. System marks the invoice as "Issued" and saves it to the database. 6. System sends the invoice to the customer. 7. System updates the workshop's financial records.
	Postconditions	1. Invoice is created and stored in the system. 2. Invoice is marked as "Issued" and linked to the service record. 3. Customer receives the invoice notification. 4. Workshop's accounts receivable is updated.
Alternative Flow & Exceptions		**Apply Discounts or Promotions:** 1. System recalculates the total and shows the discounted amount. **Calculation Error**: 1. System highlights the discrepancy and prevents issuing until resolved. 2. Mechanic manually adjust or recalculate.
Non-Functional Requirements		**Performance:** Invoice generation must complete less than 3 seconds. PDF creation and email sending within 5 seconds. **Accuracy:** All calculations must be 100% accurate and comply with tax

	regulations. Automatic tax rate application based on location.

3.2.5 View Workshop Reports

Use case name		View Workshop Reports
Actor		Mechanic/workshop
Preconditions		User is logged into the CarSync system.
Normal Flow	Description	1. User navigates to the "Reports" or "Analytics" section. 2. System displays a dashboard 3. Mechanic select from predefined report types (service summary report, financial report, inventory report, customer report, performance report). 4. System generates the report.
	Postconditions	1. User has accessed the requested report with accurate date. 2. Report can be shared, printed, or exported.
Alternative Flow & Exceptions		**Custom Report Creation:** 1. User selects "Create Custom Report" and chooses data fields, filters, and visualizations. 2. System guides the user through building a custom report. **Scheduled Reports:** 1. User can schedule reports to be generated. 2. System sends the report at the scheduled time.
Non-Functional Requirements		**Performance:** Report dashboard must load within 3 seconds. Standard reports must generate within 5 seconds. **Usability:** Intuitive interface with clear filters and visualizations. Reports should be easy to interpret even for non-technical users.

3.2.6 Manage Vehicle Profile

Use case name		Manage Vehicle Profile
Actor		Mechanic/workshop
Preconditions		1. Mechanic logged into the CarSync. 2. The mechanic has access to the vehicle's records.
Normal Flow	Description	1. Mechanic accesses the "Vehicles" section or searches for a specific vehicle. 2. System displays the vehicle's profile. 3. Mechanic performs one or more of the actions (update mileage, edit vehicle details, view or update service history) 4. System saves updates.
	Postconditions	1. Vehicle profile is updated with the latest information. 2. New service reminders are scheduled in the system.
Alternative Flow & Exceptions		**Add New Vehicle:** 1. If the vehicle doesn’t exist in the system, mechanic can choose "Add New Vehicle". 2. System prompts for basic details. 3. New vehicle profile is created and linked to the current service appointment.
Non-Functional Requirements		**Performance:** Vehicle search must return results within 2 seconds. Profile updates must save within 1 second.

3.2.7 Approve or Reject Bookings (<<include>> of View and Manage Booking)

Use case name		Approve or Reject Bookings
Actor		Mechanic/workshop
Preconditions		1. Mechanic logged into the CarSync system. 2. There is at least one pending booking request in the "View and Manage Bookings" queue.
Normal Flow	Description	1. Mechanic accesses the "View and Manage Bookings" section. 2. System displays each booking request. 3. Mechanic selects a specific pending booking to review in detail. 4. System shows full customer profile and contact details, complete vehicle service history, requested service requirements and workshop availability for the requested time slot 5. Mechanic evaluates the booking 6. Mechanic decides and selects either approve or reject booking. 7. System logs the decision.
	Postconditions	1. Booking status is updated to either "Confirmed" or "Rejected". 2. Customer receives appropriate notification of the decision. 3. Workshop calendar is updated accordingly. 4. Booking is removed from the pending requests queue. 5. Decision is recorded in system audit logs.
Alternative Flow & Exceptions		**Request Modification Instead of Rejection:** 1. Mechanic can select "Suggest Alternative" instead of rejecting.

	2. System prompts for alternative date/time suggestions. 3. Alternative proposal is sent to customer for consideration. **Conditional Approval:** 1. Mechanic can select "Approve Pending Customer Confirmation". 2. System sends detailed estimate/requirements to customer. 3. Booking is confirmed only after customer accepts conditions.
Non-Functional Requirements	**Performance:** Booking review interface must load within 2 seconds. Approval/rejection actions must process within 1 second. **Reliability:** System must prevent conflicting approvals for the same time slot. All decisions must be persistently logged.

3.2.8 Send Notification (<<include>> of Generate Invoices)

Use case name		Send Notification
Actor		Mechanic/workshop
Preconditions		1. An invoice has been successfully generated in the system. 2. Customer's valid contact information is available in the system.
Normal Flow	Description	1. System automatically triggers the notification process. 2. System retrieves customer's contact number, invoice details and workshop contact information. 3. System generates invoice summary, total due amount, due date, payment methods and full invoice file. 4. System send notification. 5. Customer receives the notification and can access their invoice. 6. System updates the invoice status to "Notification Sent".
	Postconditions	1. Customer received invoice notification. 2. Invoice status is updated to "Notification Sent" in the system. 3. Notification is logged with complete delivery details. 4. Payment timer begins based on invoice due date. 5. System is ready for payment processing.
Alternative Flow & Exceptions		**Custom Notification Message:** 1. Mechanic can choose to add a personal message to the standard notification. 2. System appends the custom message to the standard template.

	3. Personal message is logged separately for record-keeping.
Non-Functional Requirements	**Performance:** Notifications must be sent within 30 seconds of invoice issuance. Bulk notifications must process within 2 minutes.

3.2.9 Check Vehicle History (<<extend>> of Manage Vehicle Profile)

Use case name		Check Vehicle History
Actor		Mechanic/workshop
Preconditions		1. Mechanic is logged into the CarSync system. 2. Vehicle exists in the system
Normal Flow	Description	1. Mechanic selects the "View History" or "Service History" option for the current vehicle. 2. System retrieves and displays the vehicle's complete service history. 3. System shows comprehensive vehicle history 4. System provides visual indicators for regular maintenance intervals, repeated repairs, warranty coverage status on previous work and recommended next service.
	Postconditions	1. Mechanic has comprehensive understanding of vehicle's service background. 2. Service history is accessible for current service decision making. 3. Any historical patterns or recurring issues are identified. 4. Vehicle profile remains unchanged.
Alternative Flow & Exceptions		**No Service History Available** 1. System displays: "No service history found for this vehicle." 2. Provides option to mark this as the vehicle's first recorded service. 3. Suggests checking external databases or asking customer for paper records.
Non-Functional Requirements		**Performance:** Complete service history must load within 3

	seconds. Filtering and searching within history must respond within 3 second. **Accuracy:** Historical data must be consistent with original service records. No data corruption or loss in historical information.

3.3 Admin

3.3.1 Login / Authenticate

Use Case Name		Login / Authenticate
Actor		Admin
Preconditions		The system must be online and connected to the database. The Admin must possess valid registered credentials.
Normal Flow	Description	1. The Admin accesses the CarSync web portal login page. 2. The Admin enters their username and password. 3. The system validates the entered credentials against the secure database. 4. The system grants access and redirects the Admin to the main Dashboard.
	Postconditions	The Admin is successfully logged in and granted access to authorized features.
Alternative Flow & Exceptions		Invalid Credentials: If the username or password does not match the database records in Step 3: a. The system displays an "Invalid Login" error message. b. The system clears the password field and prompts the Admin to try again. c. The use case restarts at Step 2.
Non-Functional Requirements		**Security**: Passwords must be encrypted (hashed) before being stored or validated. **Performance**: The login validation process should take less than 2 seconds.

3.3.2 Manage User Accounts

Use Case Name		Manage User Accounts
Actor		Admin
Preconditions		The Admin is logged in. User accounts (Vehicle Owners and Workshops) exist in the database.
Normal Flow	Description	1. The Admin selects "User Management" from the dashboard menu. 2. The system retrieves and displays a list of all registered users. 3. The Admin searches for and selects a specific user profile. 4. The system displays the user's details (Name, Contact Info, Verification Status). 5. The Admin verifies documents or updates user details. 6. The system saves the changes to the database.
	Postconditions	The user profile information is updated and accurate in the system.
Alternative Flow & Exceptions		Suspend / Block User (Extension): At Step 4, if the Admin detects a policy violation (e.g., a workshop posting fake services): a. The Admin clicks the "Suspend User" button. b. The system prompts for confirmation. c. The Admin confirms the action. d. The system updates the user's status to "Blocked". e. The use case ends.
Non-Functional Requirements		**Audit Trail**: Any change to a user's status (especially suspension) must be logged with a timestamp and the Admin ID for accountability.

3.3.3 Generate System Reports

Use Case Name		Generate System Reports
Actor		Admin
Preconditions		The Admin is logged in. Sufficient historical data (bookings, invoices) exists in the system.
Normal Flow	Description	1. The Admin clicks the "Generate Reports" button on the dashboard. 2. The Admin selects the desired report type (e.g., Total Revenue, User Growth). 3. Include (Monitor Platform Usage): The system automatically runs a background process to aggregate raw usage statistics and data points. 4. The system calculates totals and generates a visual representation (graph or chart). 5. The system displays the report to the Admin.
	Postconditions	A generated report is displayed on the screen and available for download.
Alternative Flow & Exceptions		No Data Available: If no data exists for the selected period: a. The system displays a "No Data Found" message. b. The system returns the Admin to the Report Selection screen.
Non-Functional Requirements		**Accuracy**: Statistical calculations must be 100% accurate based on the database records. **Performance**: Report generation should not take longer than 5 seconds to load.

3.3.4 Maintain System Data (<<extend>> of View System Logs)

Use Case Name		Maintain System Data
Actor		Admin
Preconditions		The Admin must be logged into the system. The system database must be accessible and online.
Normal Flow	Description	1. The Admin selects the "System Maintenance" option from the main dashboard. 2. The system retrieves the current system status. 3. The system displays a list of maintenance options (e.g., Clear Cache, Backup Database). 4. The Admin selects a specific maintenance task. 5. The system executes the requested task. 6. The system displays a success message confirming the maintenance is complete.
	Postconditions	The selected maintenance task is successfully executed, and the system data is optimized or backed up.
Alternative Flow & Exceptions		Configure Notification Rules (Extension): At Step 3, the Admin may choose to "Configure Notification Rules" instead of a standard maintenance task. The Admin updates the mileage threshold for alerts, and the system saves the new configuration. View System Logs (Inclusion): During Step 2, the system automatically triggers the "View System Logs" function to display recent errors or warnings alongside the maintenance options.
Non-Functional Requirements		**Reliability**: Maintenance operations must ensure data integrity and should not corrupt existing records. **Performance**: Routine maintenance tasks should complete within 60 seconds.

3.3.5 View System Logs

Use Case Name		View System Logs
Actor		Admin
Preconditions		The Admin is logged in.
Normal Flow	Description	1. The Admin clicks the "View Logs" option on the dashboard. 2. The system retrieves the latest error logs, security alerts, and activity history from the database. 3. The system displays the logs in a chronological list (newest first). 4. The Admin filters the list (e.g., by error type or date) to find specific events.
	Postconditions	The Admin has successfully reviewed the system activity history.
Alternative Flow & Exceptions		Database Connection Error: If the system cannot retrieve logs from the database: a. The system displays a "Connection Timed Out" error. b. The system advises the Admin to contact technical support.
Non-Functional Requirements		**Data Integrity**: The log view must be read-only. The Admin cannot delete or modify log entries from this view to ensure the audit history is preserved.

3.3.6 Configure Notification Rules

Use Case Name		Configure Notification Rules
Actor		Admin
Preconditions		The Admin is now logged in.
Normal Flow	Description	1. The Admin navigates to the "Notification Settings" page. 2. The system displays the current automation rules (e.g., "Remind Vehicle Owner at 5000km"). 3. The Admin edits a specific parameter (e.g., changes 5000km to 8000km). 4. The Admin clicks "Save Configuration". 5. The system validates the input and updates the global settings in the database.
	Postconditions	The new notification rules are saved and become active immediately for all users.
Alternative Flow & Exceptions		Invalid Input: If the Admin enters a negative number or invalid character for mileage: a. The system shows an error: "Mileage must be a positive number". b. The system prevents the save action. c. The use case allows the Admin to re-enter the value.
Non-Functional Requirements		**Usability**: Critical changes to system rules should trigger a confirmation popup to prevent accidental edits.

3.3.7 Suspend/Block Users (<<extend>> of Manage User Accounts)

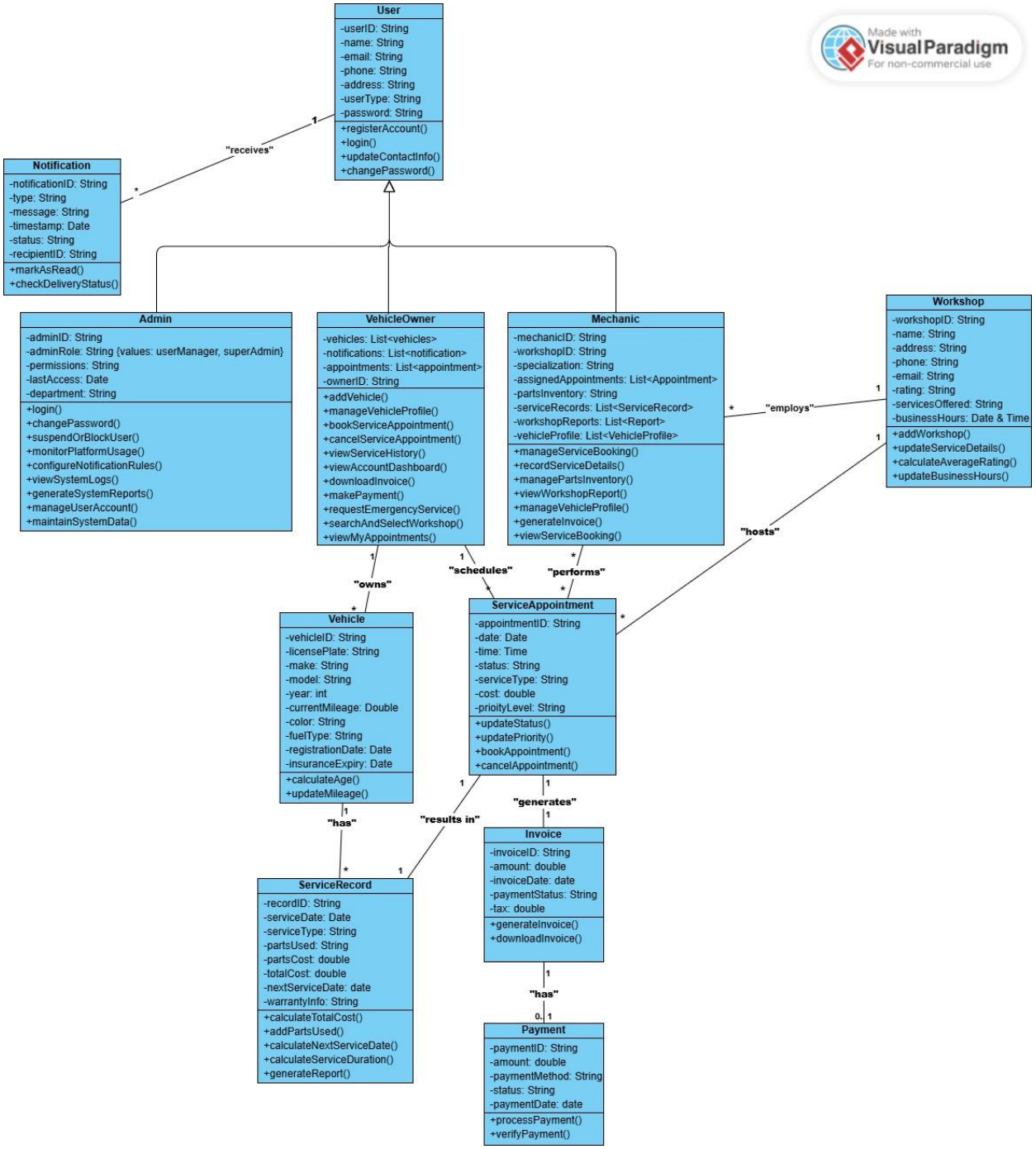
Use Case Name		Suspend / Block Users
Actor		Admin
Preconditions		The Admin is logged in. A user account exists that has violated platform policies.
Normal Flow	Description	1. The Admin identifies a policy violation. 2. The Admin selects the "Suspend" option on the user's profile. 3. The system displays a confirmation popup warning of the consequences. 4. The Admin confirms the suspension. 5. The system updates the user's status to "Blocked" in the database. 6. The system sends an automated email notification to the user explaining the suspension.
	Postconditions	The user account is deactivated and they can no longer log in.
Alternative Flow & Exceptions		Accidental Suspension: If the Admin clicks "Cancel" at Step 4: a. The system aborts the process. b. The user status remains "Active".
Non-Functional Requirements		Security: Only Admins with "Super User" privileges can perform this action. Audit Trail: This action must be permanently logged with the Admin's ID and timestamp.

3.3.8 Monitor Platform Usage (<<include>> of Generate System Report)

Use Case Name		Monitor Platform Usage
Actor		Admin
Preconditions		The database contains active user and transaction data.
Normal Flow	Description	1. The function is triggered automatically when the Admin requests a report. 2. The system queries the database for key metrics (Total Bookings, Active Users, Revenue). 3. The system aggregates the data based on the selected time range (e.g., Last 30 Days). 4. The system formats the raw data into visual statistics (charts/graphs). 5. The system returns the processed data to the "Generate System Reports" function.
	Postconditions	Raw data is successfully converted into readable statistics.
Alternative Flow & Exceptions		Database Timeout: If the query takes too long due to high traffic: a. The system retries the query once. b. If it fails again, it returns a "Data Unavailable" error to the main report screen.
Non-Functional Requirements		Performance: Data aggregation for up to 10,000 records should complete within 3 seconds.

4 System Models

4.1 Class Diagrams / ERD



4.2 Classes / Entities

Class / Entity	Description
User	Superclass of Admin, VehicleOwner, and Mechanic. One User (1) can receive many Notifications (*).
Workshop	One Workshop (1) employs many Mechanics (*) and hosts many ServiceAppointments (*).
Mechanic	Inherits from User. Many Mechanics (*) are employed by one Workshop (1). Many Mechanics (*) perform many ServiceAppointments (*).
Admin	Inherits from User. Does not directly participate in vehicle ownership or service appointments.
VehicleOwner	Inherits from User. One VehicleOwner (1) owns many Vehicles (*) and schedules many ServiceAppointments (*).
ServiceRecord	Each ServiceRecord (1) results from one ServiceAppointment (1) and is associated with one Vehicle (1). One vehicle (1) can have many ServiceAppointments (*).
ServiceAppointment	One VehicleOwner (1) can schedule many ServiceAppointments (*), many ServiceAppointments (*) are performed by many Mechanics (*), and hosted by one Workshop (1).
Invoice	Each Invoice (1) is generated from one ServiceAppointment (1) and has zero or one Payment (0..1).
Payment	Each Payment (1) is linked to one Invoice (1).
Vehicle	Each Vehicle (1) is owned by one VehicleOwner (1) and has many ServiceRecords (*).
Notification	A User (1) can receive many Notifications (*).

5 Non-Functional Requirements

1.Performance - *CarSync can support at least 50 concurrent users without performance issues.*

2.Performance - *Capable of storing up to 10000 service records and vehicle profiles while maintaining query response time under 3 seconds.*

3.Reliability - *Carsync should be available 99% of the time during business hours.*

4.Security - *Role-Based Access Control. (Vehicle owners cannot access Admin dashboards and Mechanics cannot view the financial data of other workshops.)*

5.Usability - CarSync's interface must be basic and intuitive enough for non-tech-savvy users to book a service appointment within 3 minutes without external assistance.

6.Scalability - The system must support a 30% increase in users and service records annually without requiring significant system changes or performance degradation.

7.Portability - CarSync must be accessible and fully functional on the latest versions of Chrome, Firefox, and Safari browsers, as well as iOS and Android mobile browsers.

6 References

1. *How to Draw Class Diagram? (n.d.). Www.visual-Paradigm.com.* https://www.visual-paradigm.com/support/documents/vpuserguide/94/2576/7190_drawingclass.html
2. *How to Draw Use Case Diagram?* (n.d.). Www.visual-Paradigm.com. https://www.visual-paradigm.com/support/documents/vpuserguide/94/2575/6362_drawinguseca.html
3. Rome, P. (2020, August 11). *What are Non Functional Requirements — With Examples*. Perforce Software. https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples
4. *Diagram maker for developers*. (2024). Gleek.io. https://www.gleek.io/blog/multiplicity-class-diagram