

**American International University-Bangladesh (AIUB)**  
 Department of Computer Science

Faculty of Science & Technology (FST)

**CAR DEALERSHIP MANAGEMENT SYSTEM**

**Semester: Fall 25-26**

|  |  |  |
| --- | --- | --- |
| ***Group:*** | | ***Section:*** |
| **SL** | **Student Name** | **Student ID** |
| 1 | SAZID HASAN | 23-53956-3 |
| 2 | MS NURJAHAN AKTER ANIKA | 23-54416-3 |
| 3 | RABEYA BASHRI SADIA | 23-53625-3 |
| 4 |  |  |
| 5 |  |  |

Table of Contents

[1. Project Proposal xx](#_Toc214214201)

[1.1 Background to the Problem](#_Toc214214203) xx

[1.2 Selection of Process Model xx](#_Toc214214204)

[2. Software Requirements Specification (SRS) / PRD xx](#_Toc214214202)

[2.1 Scopes and Features xx](#_Toc214214203)

[2.2 User Story Table xx](#_Toc214214204)

[2.3 Requirements Traceability Matrix xx](#_Toc214214205)

[2.3.1 Functional Requirements xx](#_Toc214214212)

[2.3.2 Non-Functional Requirements xx](#_Toc214214216)

[3. Software Design xx](#_Toc214214202)

[3.1 System Deisgn xx](#_Toc214214203)

[3.2 UI Deisgn using Figma xx](#_Toc214214203)

[4. Git Workflow xx](#_Toc214214202)

[5. Software Testing xx](#_Toc214214202)

[6. Conclusion xx](#_Toc214214202)

**1. PROJECT PROPOSAL**



* 1. **Background to the Problem**

The Car Dealership Management System is a software application developed to manage and automate the daily operational activities of a car dealership. The primary goal of this system is to replace traditional manual processes with a centralized digital platform that ensures better organization, accuracy, and efficiency. It serves as an integrated solution for managing all dealership-related operations under a single system.

This system helps in maintaining detailed information about customers, employees, cars, purchases, payments, and feedback in a structured and systematic manner. By storing all data in a centralized database, the system reduces redundancy, prevents data loss, and ensures quick access to information. As a result, dealership management can easily monitor activities, track records, and make informed decisions based on real-time data.

The Car Dealership Management System significantly reduces manual workload and minimizes human errors that often occur in paper-based systems. Automated processes such as order placement, payment handling, and inventory updates improve overall operational efficiency and save valuable time for both employees and administrators. The system also enhances transparency by allowing users to track order status, payment history, and service records accurately.

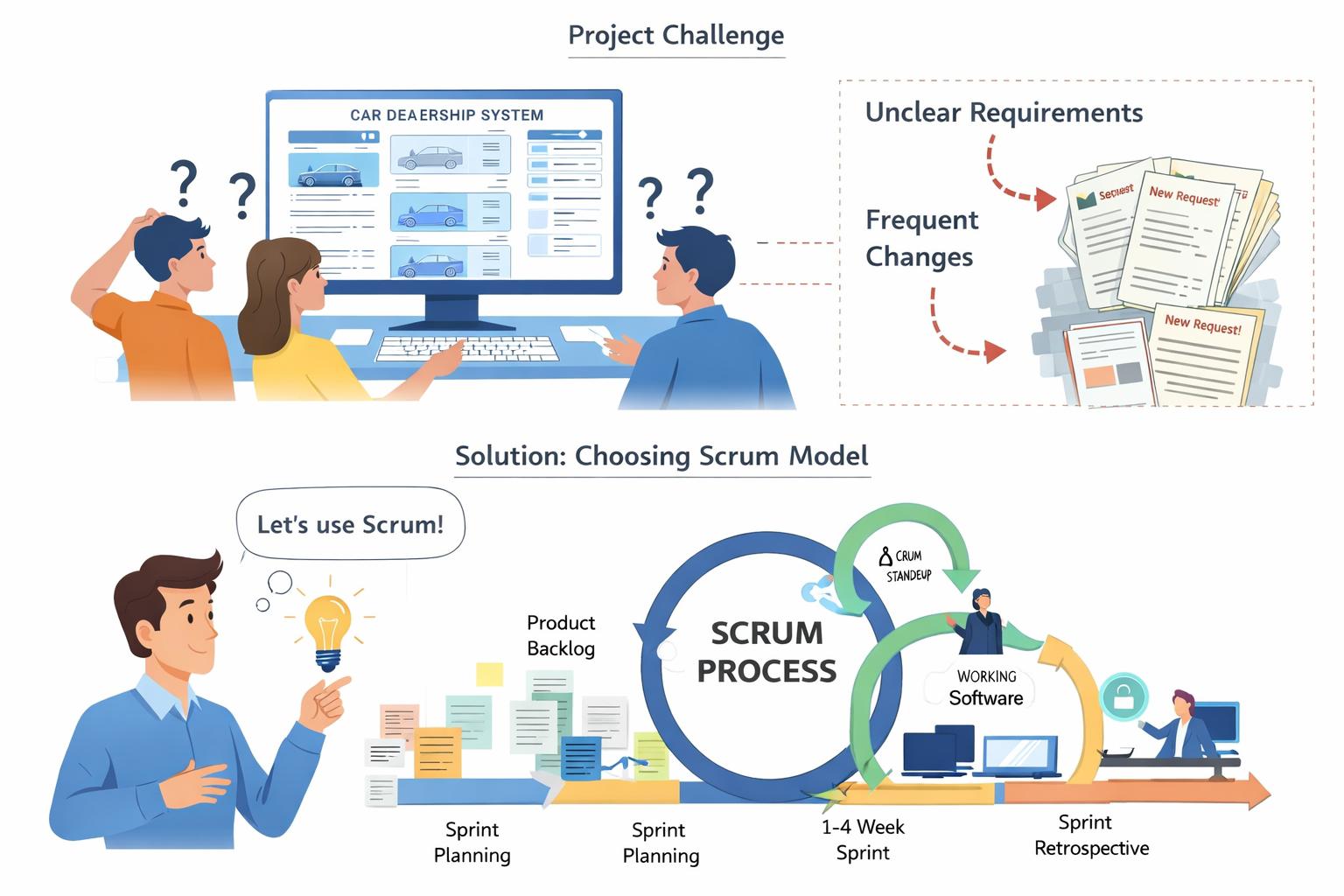
From the customer’s perspective, the system provides a convenient and user-friendly experience. Customers can easily browse available cars, view detailed specifications, book or purchase vehicles, make secure payments, and provide feedback after successful transactions. These features improve customer engagement and satisfaction by making the entire process faster and more reliable.

For employees and administrators, the system simplifies record management, sales tracking, and customer support activities. Role-based access ensures secure usage of the system while allowing employees to perform their assigned tasks efficiently. Overall, the Car Dealership Management System ensures smooth communication between the dealership and its customers, enhances productivity, and contributes to better service quality and customer satisfaction.

* 1. **Selection of Process Model**

For the Car Dealership Management System, the most suitable software process model is the Scrum (Agile) Model. This project includes several features such as customer interaction, car inventory management, payment processing, and administrative control. These can be developed gradually in different stages. So, Scrum process model is suitable for this project as scrum developers the system in small iterations called sprint.

**Scrum Process Model:**

****

**Example Sprint Breakdown:**

* Sprint 1: Login, registration, car browsing
* Sprint 2: Car booking and payment
* Sprint 3: Admin panel and inventory management

**Why Scrum Was Chosen for This Project**

* The Car Dealership Management System may require changes in features such as car details, pricing, payment options, or admin functions. Scrum effectively handles these changes by allowing continuous updates and prioritization of the Product Backlog.
* Scrum develops the system incrementally through short development cycles called sprints, making it easier to implement complex features like customer management, booking, and inventory control in an organized manner.
* At the end of each sprint, Scrum delivers a working product increment, ensuring that usable parts of the system are available early and improved over time.
* Scrum encourages continuous feedback through Sprint Review meetings, where stakeholders can evaluate completed features and suggest improvements, leading to a more user-friendly system.
* Since testing is included in every sprint, issues are identified and fixed early, which improves the overall quality and reliability of the Car Dealership Management System.
* Scrum is highly suitable for group projects because it defines clear roles, promotes teamwork, and ensures effective communication among team members throughout the development process.

**Scrum is chosen because:**

* It develops the system in short sprints
* It supports changing requirements
* Daily scrum meeting
* Sprint review meeting (Last Day)
* It delivers working software incrementally
* It ensures continuous feedback and improvement

Scrum was chosen as the software development process model for the Car Dealership Management System because of its flexibility and adaptability to changing requirements. This project involves multiple modules such as user management, car inventory, order processing, payment handling, and feedback management. As these features may evolve during development, Scrum allows the team to easily accommodate changes without affecting the entire system.

Scrum follows an iterative and incremental development approach, where the system is developed in short cycles called sprints. This makes it easier to divide the project into smaller, manageable tasks and implement complex features gradually. Each sprint delivers a working increment of the system, allowing early identification of issues and continuous improvement of functionality.

Another important reason for choosing Scrum is its strong focus on continuous feedback. Through sprint reviews and regular communication, stakeholders and team members can review completed features and suggest improvements. This ensures that the system meets user expectations and remains aligned with project goals throughout the development process.

Scrum also emphasizes continuous testing and integration. Testing is performed during each sprint, which helps identify defects at an early stage and improves overall software quality and reliability. This reduces the risk of major issues appearing at the end of the project.

Finally, Scrum is highly suitable for group-based projects, especially in an academic environment. It promotes teamwork, clear role distribution, accountability, and effective communication among team members. Daily scrum meetings help track progress, resolve obstacles, and ensure that the project stays on schedule.

**Why Other Models Were Not Selected**

| **Model** | **Reason for Not Choosing** |
| --- | --- |
| **Waterfall Model** | It does not support changes, backtracking once development starts, which is unsuitable for a system like a Car Dealership Management System where requirements may change. |
| **V-Model** | Although it emphasizes testing, it is still like the Waterfall model and does not handle frequent requirement changes and backtracking well. |
| **XP (Extreme Programming)** | It requires highly experienced developers and continuous customer involvement, which is difficult to maintain in a student group project. |
|  |  |
| **Prototyping Model** | It mainly focuses on quick user interface development and may lead to incomplete system design if used alone. It is not ideal for managing a full system with many modules. |
| **Iterative Model** | Although it supports iteration, it lacks clearly defined roles, sprint planning, and structured feedback mechanisms compared to Scrum. |
| **DSDM** | It involves heavy documentation and strict control mechanisms, which can be difficult and time-consuming for students to manage. |

## 2. SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS) / PRODUCT REQUIREMENTS DOCUMENT (PRD)

**2.1 Scopes and Features**

The scope of the car dealership management system covers the complete automation of car dealership activities for Admin, Employee, and Customer users. The system allows users to manage cars, users, orders, payments, and feedback in a centralized and secure environment.

The system will:

* Replace manual record keeping
* Reduce human errors
* Improve efficiency and transparency
* Provide real time data access

**Key Features**

**Admin:**

* Create, update, delete employee accounts
* View, update, and delete customer accounts
* Add, update, remove car information
* View and manage all orders
* Approve or reject orders
* View all payment records
* View customer feedback and rating
* Track login history of users

**Employee:**

* Login using admin provided credentials
* View and edit personal profile
* View pending orders
* Check payment status of orders
* Approve or reject orders

**Customer:**

* Create an account and log in
* View and edit profile information
* Browse available cars with details
* View full specifications
* Place order for selected car
* Track order status
* Cancel pending orders
* Make payment for orders
* Submit feedback after successful order

**2.2 User Story Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **As A/An** | **I want to** | **So that** | **Acceptance Criteria** |
| Admin | Create a new employee profile | Employee can access the system | Show employee registration form and show notification of success after saving |
| Update customer type | If a customer completes any milestone as per company policy, they can get some more advantages according to their batch | Show customer type updated message |
| View all user information | See all my users including customers and employees | Show a user list table with update and delete button |
| Update user information | If any user is having trouble in updating their personal information | Show editable fields and success message after saving changes |
| Delete any user from database | The user cannot access the system anymore | show a confirmation popup message before deletion |
| View all log in history | I can track the users | Show a log in list table |
| View al existing cars | I can monitor the warehouse | Show a car list table showing all my existing cars in the system |
| Add a new car | Customer can see available cars | Show car entry form; required fields must be filled; show notification of success |
| Update car information | Car details stay accurate | Show updated car list table with edit option |
| Remove a car | The car is no longer available | Show confirmation popup |
| View all orders | I can monitor customer activity and their preferences | Show order list table |
| Approve an order | The customer can complete the order and get the product | Show order list with status updated to “Approved” |
| Reject an order | Customer is informed that the order cannot be proceed | Show order list table status with updated to “Rejected” |
| View all payments | I can verify transaction completion | Show payment list table |
| View all feedbacks | I can monitor customer satisfaction | Show feedback table including rating and comment |
|  |  |  |  |
| Employee | Log in with the given username and password | I can access the system | After successful log in, my associated data should be accessible |
| View my profile details | I can see my personal details | Show information list table and edit profile button |
| Edit my profile | My profile stays accurate | Show editable fields and success message after update |
| View pending orders with payment status | I can process customer requests | Show pending order list with details (car + customer + payment info) |
| Approve an order | The customer can proceed | Show order list with status updated to “Approved” |
| Reject an order | Customer is notified that order is rejected | Show order list with status updated to “Rejected” |
|  |  |  |  |
| Customer | Create account | I can access the system | Show registration form and success message after account creation |
| Log in with my account | The system can authenticate me | My associated data should be accessible |
| View my profile details | I can see my personal information | Show my information including edit profile button |
| Edit profile | My profile stays accurate | Show editable fields and success message after update |
| View available cars | I can choose a car to purchase | Show car list with image, specific specification, and view detail button |
| View car details | I can check full specifications | Show car details window with order button |
| Place an order | I can book a car | Show order list with new order added as “Pending” |
| Track my previous orders | I can see the progress of my orders | Show order list with real time status (Pending/ Approved/ Rejected) |
| Cancel my pending orders | I can change my decision | Show order list with cancel button only for pending orders and update status |
| Make payment of my pending order | My order can be confirmed | Show payment form |
| Give feedback | I can share my experience | Feedback option appears only when order status is “Successful”. And show success message after submitting |

**2.3 Requirements Traceability Matrix**

The requirement traceability matrix ensures that all requirements are properly implemented and tested.

**2.3.1 Functional Requirements**

1. The system shall allow the admin to log in using a valid username and password
2. The system shall allow the admin to create new employee account
3. The system shall allow the admin to view all user information including customers and employees
4. The system shall allow the admin to update user information when necessary
5. The system shall allow the admin to delete any user from database
6. The system shall allow the admin to view log in history of all users
7. The system shall allow the admin to view all existing cars in the system
8. The system shall allow the admin to add a new car
9. The system shall allow the admin to update car details
10. The system shall allow the admin to remove a car from the system
11. The system shall allow the admin to view all order records
12. The system shall allow the admin to approve or reject customer order
13. The system shall allow the admin to view all payment records
14. The system shall allow the admin to view all customer feedbacks
15. The system shall allow employees to log in using admin provided credentials
16. The system shall allow employees to view their profile information
17. The system shall allow employees to edit their profile
18. The system shall allow employees to view pending customer orders with payment details
19. The system shall allow employees to approve or reject orders after payment verification
20. The system shall allow customers to create a new account
21. The system shall allow customers to log in using valid credentials
22. The system shall allow customers to view their profile information
23. The system shall allow customers to update their profile details
24. The system shall allow customers to view available cars
25. The system shall allow customers to view detailed specifications of selected cars
26. The system shall allow customers to place order for selected cars
27. The system shall allow customers to keep their choices in wish-list to order later
28. The system shall allow customers to track status of their orders
29. The system shall allow customers to cancel orders that are still pending
30. The system shall allow customers to make payment for pending unpaid order
31. The system shall allow customers to submit feedback after successful order completion

**2.3.2 Non-Functional Requirements**

Performance requirements:

* The system should load pages within 2-3 seconds
* Database queries should execute efficiency

Security requirements:

* User passwords must be stored securely
* Only authorized users can access restricted modules
* Customer payment details such as pay id, pin etc must be secured

Usability requirements:

* The interface should be user-friendly
* Buttons and forms should be clearly labelled
* Error message should be easy to understand

Reliability requirements:

* The system should not crash during normal operation
* Data should not be lost during system failure

Maintainability requirements:

* The system should not crash during normal operation
* Data should not be lost during system failure

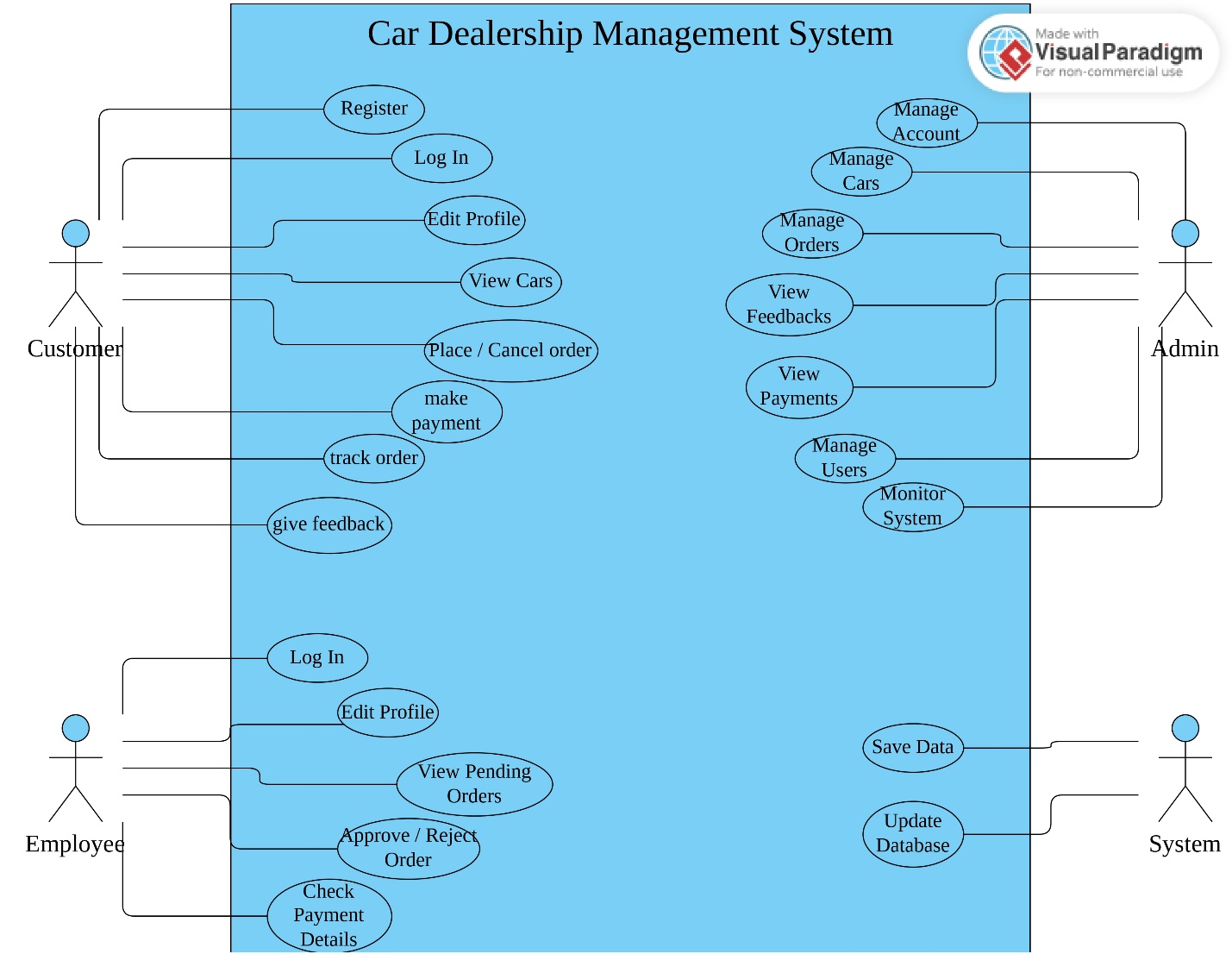
Scalability requirements:

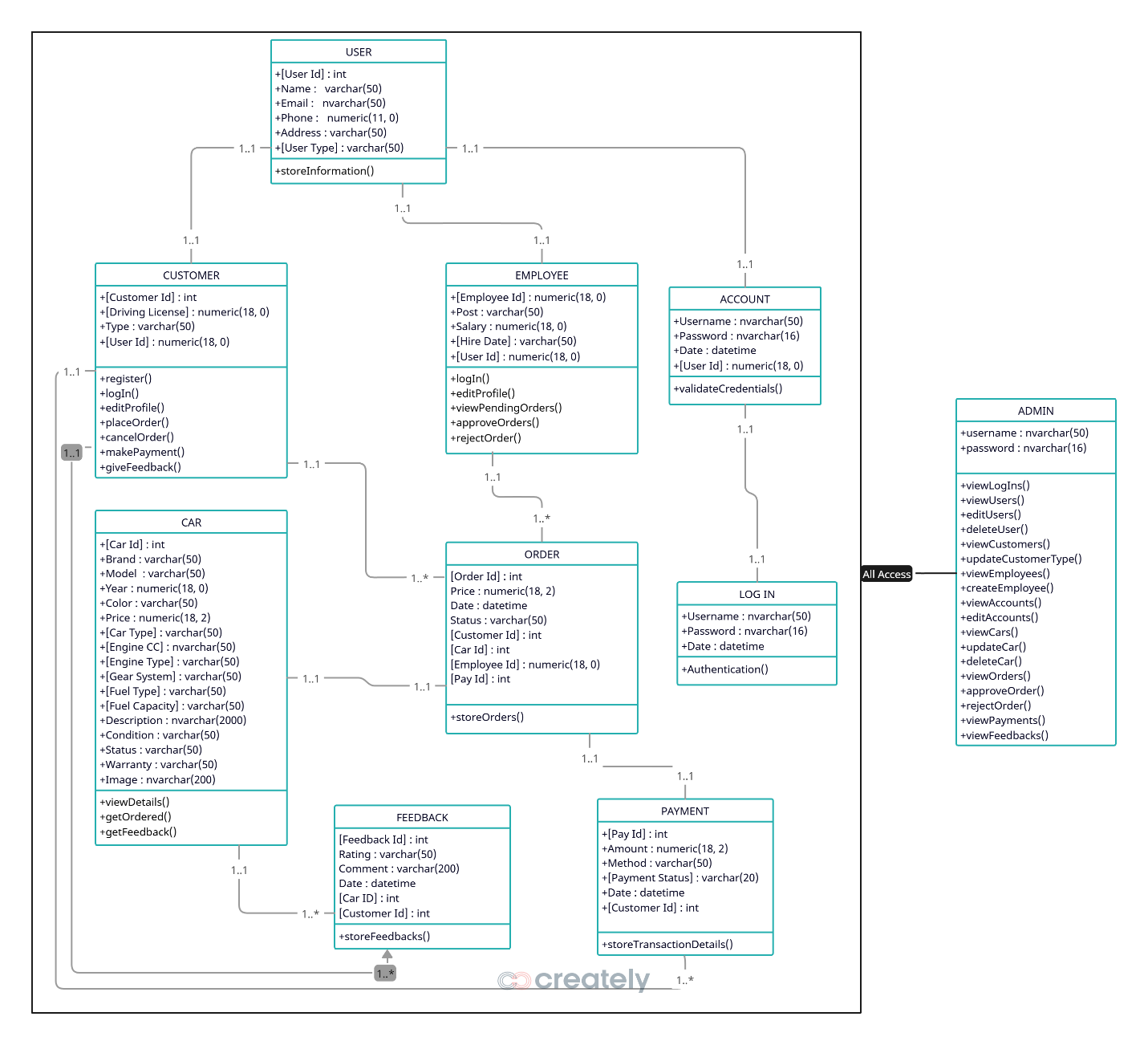
* The system should support future feature additions
* Database should handle increased data volume

# 3. SOFTWARE DESIGN

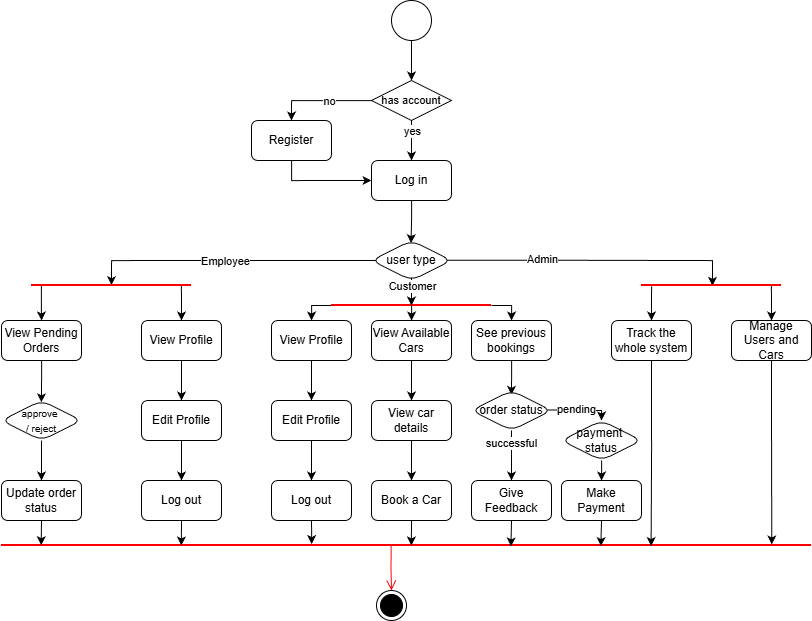
**3.1 System Design:**

Use Case Diagram





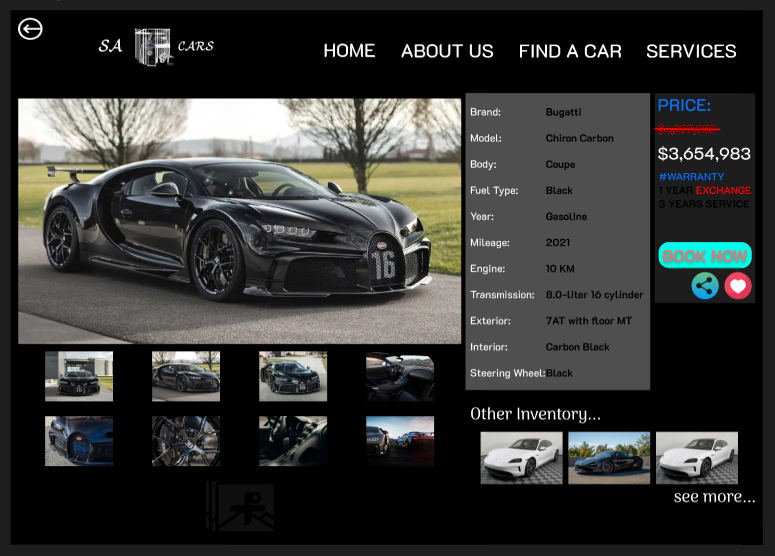
Class Diagram

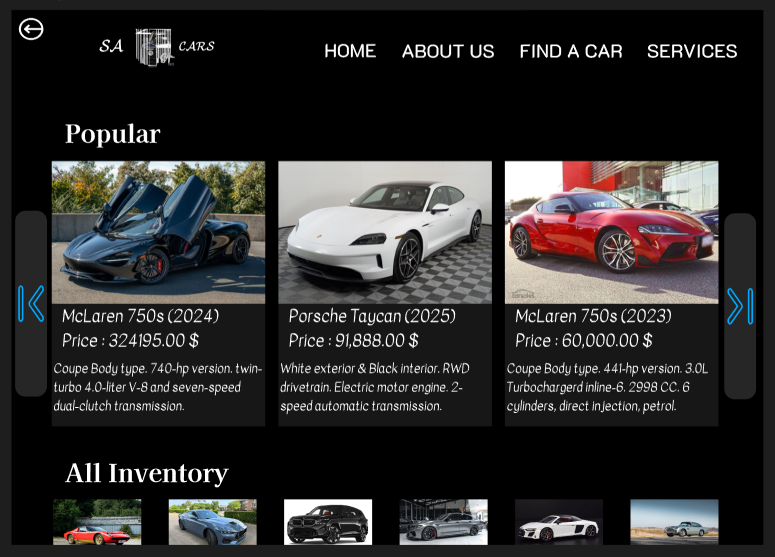


Activity Diagram

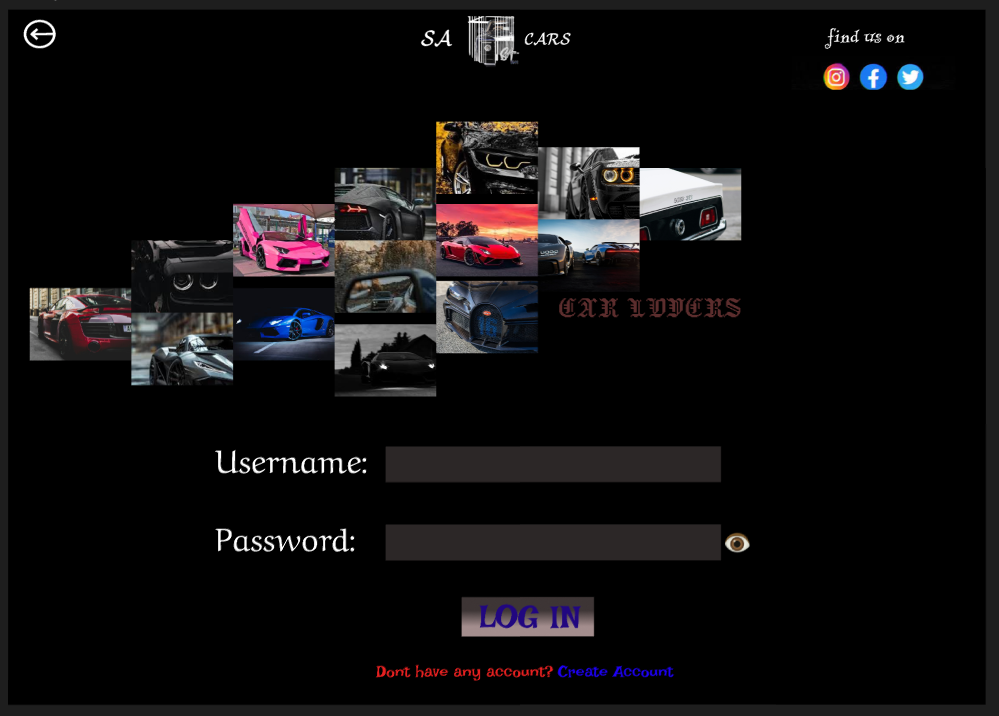
## UI / Wireframe Design using Figma





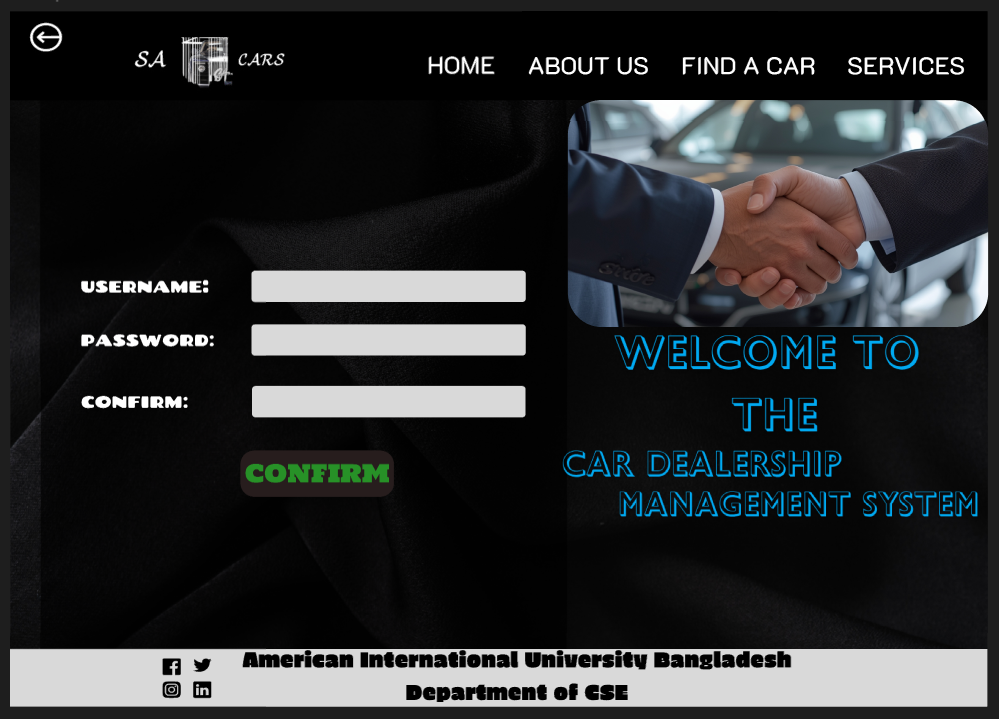


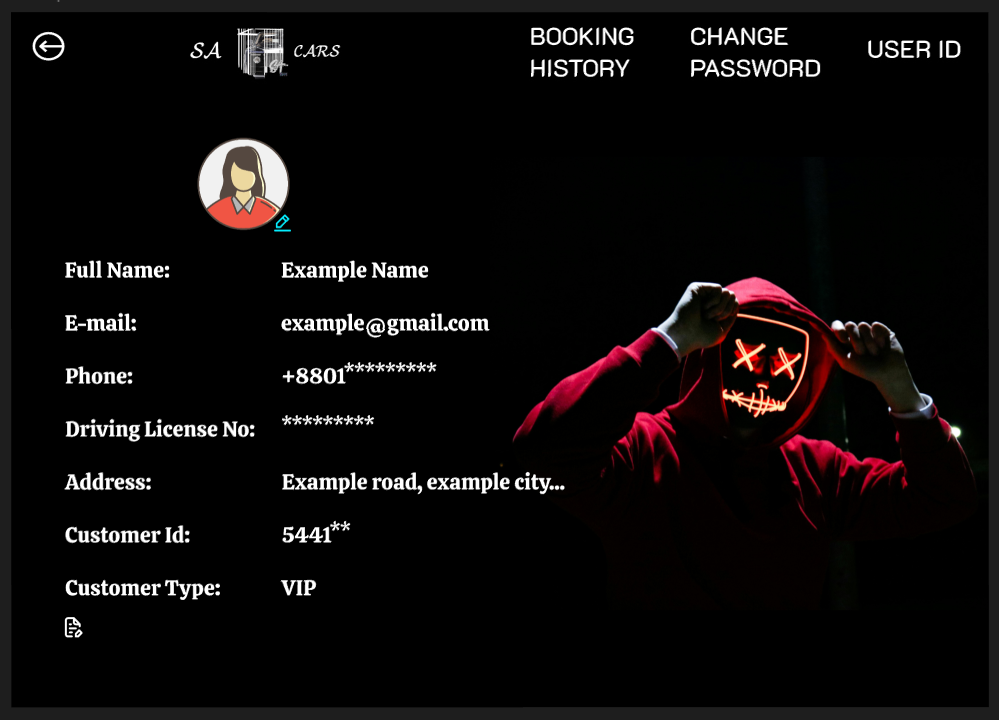


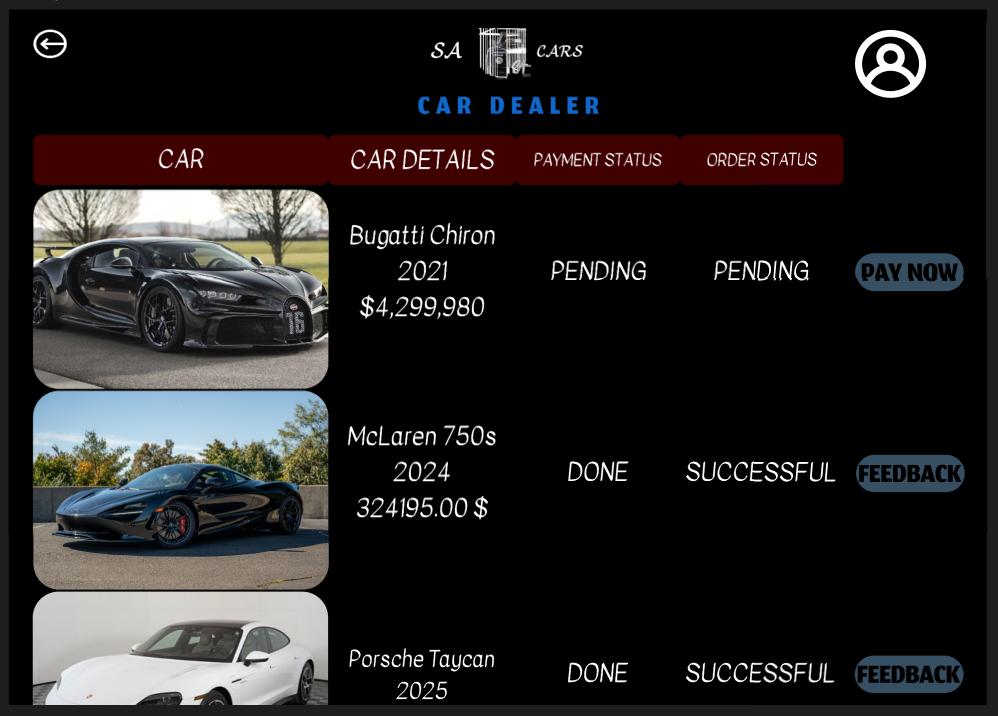




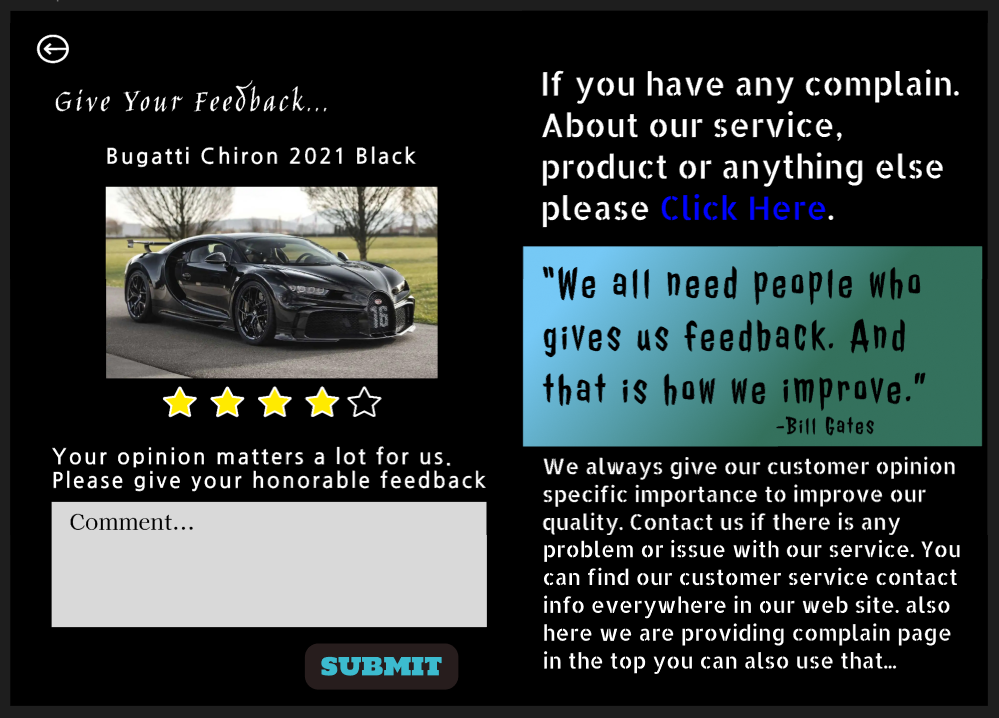


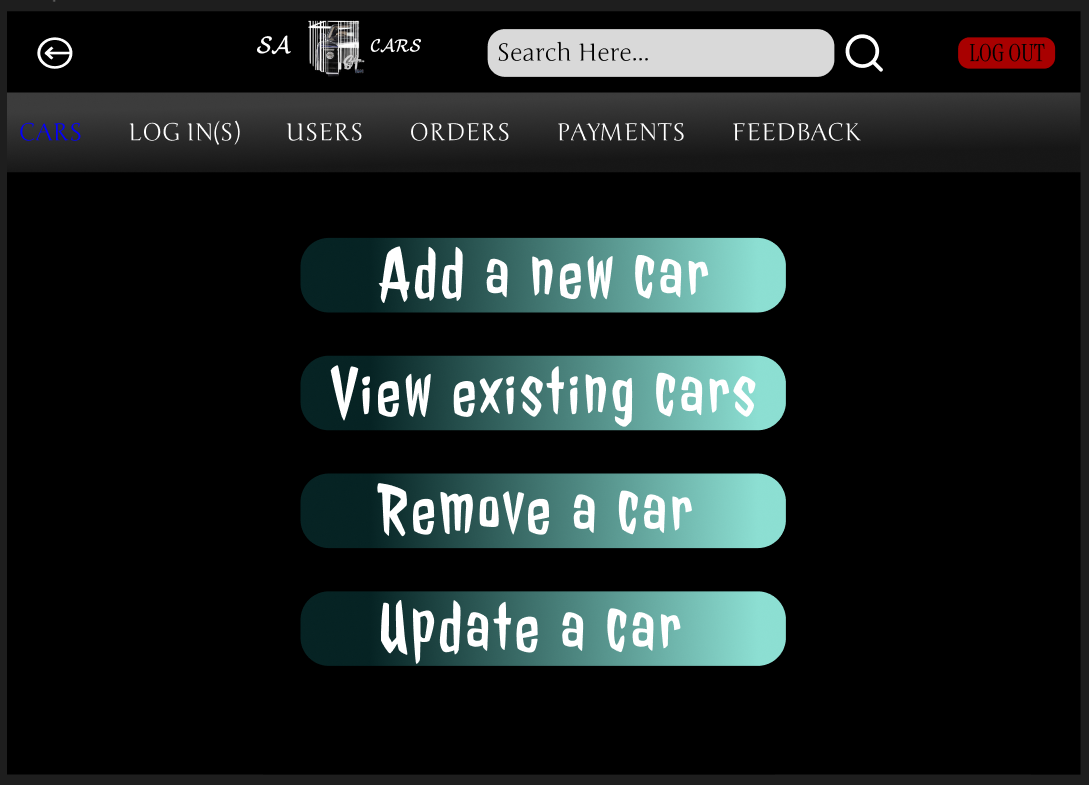












# 4. GIT WORKFLOW

* Create a central repository for the project on GitHub and set the **master (or main) branch** as the primary branch for integration.
* Each member should clone the repository and create their own **feature branches** for assigned tasks. Work on new features or fixes within these branches.
* Add files, stage them and commit changes with clear messages that describe the purpose of each update.
* Push commits from the feature branches to the remote repository so other members can see progress.
* Use **pull** to fetch and integrate changes from the remote repository into local copies, ensuring everyone stays updated.
* Merge feature branches into the **master/main branch** only after the work is tested and reviewed, resolving any conflicts that occur.
* Show evidence of collaboration by maintaining a clear commit history (using logs) with multiple commits, merges and contributions from all group members.
* Keep the repository organized with a clean history that tracks the project workflow from initialization to completion.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: Car Dealership Management | | Test Designed by: Sazid Hasan | |
| Test Case ID: TC\_1 | | Test Designed date: 05/01/2026 | |
| Test Priority (Low, Medium, High): High | | Test Executed by: Sazid Hasan | |
| Module Name: Valid User Login | | Test Execution date: 05/01/2026 | |
| Test Title: Verify login with valid username and password | | | |
| Description: Verify login page | | | |
| Precondition: The user has a valid username and password | | | |
| Dependencies: N/A | | | |
| Test Steps | Test Data | Expected Results | Actual Results / Status |
| 1. Go to the site 2. Enter username 3. Enter password 4. Click LOG IN | Username: sazid1 Password: sazid1 | Dashboard Opens | Actual Result: As expected Status: Pass |

# 5. SOFTWARE TESTING

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: Car Dealership Management | | Test Designed by: Sazid Hasan | |
| Test Case ID: TC\_2 | | Test Designed date: 06/01/2026 | |
| Test Priority (Low, Medium, High): Medium | | Test Executed by: Sazid Hasan | |
| Module Name: Login records | | Test Execution date: 07/01/2026 | |
| Test Title: Verify login records are getting stored successfully in database | | | |
| Description: Test the website login record page from admin | | | |
| Precondition: Proper database connection | | | |
| Dependencies: A user must log in using valid username and password first | | | |
| Test Steps | Test Data | Expected Results | Actual Results / Status |
| 1. Log in using a valid username and password 2. Log out from the user account 3. Log in using admin account 4. Go to log in record page | Username: sazid1 Password: sazid1  Username: admin  Password: admin | The username and log in timing should be stored in database accurately | Actual Result: As expected Status: Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: Car Dealership Management | | Test Designed by: Sazid Hasan | |
| Test Case ID: TC\_3 | | Test Designed date: 06/01/2026 | |
| Test Priority (Low, Medium, High): Medium | | Test Executed by: Sazid Hasan | |
| Module Name: Order Module | | Test Execution date: 07/01/2026 | |
| Test Title: Verify booking can be cancelled easily | | | |
| Description: I want to cancel my booking that I placed earlier | | | |
| Precondition: The order must be placed earlier | | | |
| Dependencies: N/A | | | |
| Test Steps | Test Data | Expected Results | Actual Results / Status |
| 1. Log in to the system 2. Go to booking history 3. Select an order to cancel 4. Click Cancel | Order Id: 01 | Order shall be cancelled and deleted from database | Actual Result: As expected Status: Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: Car Dealership Management | | Test Designed by: Sazid Hasan | |
| Test Case ID: TC\_4 | | Test Designed date: 16/01/2026 | |
| Test Priority (Low, Medium, High): Medium | | Test Executed by: Sazid Hasan | |
| Module Name: Car Module | | Test Execution date: 17/01/2026 | |
| Test Title: View Car | | | |
| Description: View all available cars | | | |
| Precondition: Customer logged in | | | |
| Dependencies: N/A | | | |
| Test Steps | Test Data | Expected Results | Actual Results / Status |
| 1. Log in 2. Go to cars | N/A | View all available cars with details | Actual Result: Cars loading failed Status: Failed |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: Car Dealership Management | | Test Designed by: Mst Nurjahan Akter Anika | |
| Test Case ID: TC\_1 | | Test Designed date: 08/01/2026 | |
| Test Priority (Low, Medium, High): High | | Test Executed by: Anika | |
| Module Name: Order Module | | Test Execution date: 09/01/2026 | |
| Test Title: Verify placing order | | | |
| Description: Verify that a logged in customer can place an order for a selected car | | | |
| Precondition: Customer must be valid and car must be available in stock | | | |
| Dependencies: Customer must log in first with valid account | | | |
| Test Steps | Test Data | Expected Results | Actual Results / Status |
| 1. Log in to the system 2. Go to cars module 3. Select a car 4. Click place order | Car Id: 1001 | Order is placed successfully | Actual Result: As expected Status: Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: Car Dealership Management | | Test Designed by: Mst Nurjahan Akter Anika | |
| Test Case ID: TC\_2 | | Test Designed date: 14/01/2026 | |
| Test Priority (Low, Medium, High): High | | Test Executed by: Anika | |
| Module Name: Payment Module | | Test Execution date: 14/01/2026 | |
| Test Title: Verify payment for placed order | | | |
| Description: Test verifies that payment can be made only after an order is placed | | | |
| Precondition: Order must be placed first | | | |
| Dependencies: Order placement | | | |
| Test Steps | Test Data | Expected Results | Actual Results / Status |
| 1. Go to booked cars 2. Select a pending order 3. Click make payment 4. Enter payment details 5. Click pay now | Payment method: Card  Pay Id: 11111 | Payment is completed successfully and order status updated to “Paid” | Actual Result: As expected Status: Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: Car Dealership Management | | Test Designed by: Mst Nurjahan Akter Anika | |
| Test Case ID: TC\_3 | | Test Designed date: 02/01/2026 | |
| Test Priority (Low, Medium, High): High | | Test Executed by: Name | |
| Module Name: Employee Module | | Test Execution date: 18/01/2026 | |
| Test Title: Verify order approval | | | |
| Description: Test that employee can approve an order | | | |
| Precondition: Pending order must be paid already | | | |
| Dependencies: There must be an existing pending order | | | |
| Test Steps | Test Data | Expected Results | Actual Results / Status |
| 1. Log in with an employee account 2. Go to pending orders tab 3. Select an order with payment status done 4. Click approve | Order Id: 01 | Order status updated to “Successful” and customer must be informed | Actual Result: Order status was not updated Status: Failed |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: Car Dealership Management | | Test Designed by: Mst Nurjahan Akter Anika | |
| Test Case ID: TC\_4 | | Test Designed date: 10/01/2026 | |
| Test Priority (Low, Medium, High): Low | | Test Executed by: Anika | |
| Module Name: Customer Module | | Test Execution date: 15/01/2026 | |
| Test Title: Verify updating profile | | | |
| Description: Verify that a customer can update his/her profile | | | |
| Precondition: Customer must be valid | | | |
| Dependencies: Customer must log in first with valid account | | | |
| Test Steps | Test Data | Expected Results | Actual Results / Status |
| 1. Log in to the system 2. Go to profile 3. Click Edit 4. Change any information  5. Click Done | Customer Id: C001 | Customer profile updated successfully | Actual Result: As expected Status: Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: Car Dealership Management | | Test Designed by: Rabeya Bashri Sadia | |
| Test Case ID: TC\_1 | | Test Designed date: 14/01/2026 | |
| Test Priority (Low, Medium, High): Low | | Test Executed by: Sadia | |
| Module Name: Feedback Module | | Test Execution date: 15/01/2026 | |
| Test Title: Give feedback | | | |
| Description: Submit feedback for a successful order | | | |
| Precondition: Order approved | | | |
| Dependencies: Order Approval | | | |
| Test Steps | Test Data | Expected Results | Actual Results / Status |
| 1. Log in with customer account 2. Go to booking history 3. Select a successful order 4. Click give feedback  5. Give feedback  6. Click submit | Rating: 5 stars | Feedback saved | Actual Result: As expected Status: Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: Car Dealership Management | | Test Designed by: Rabeya Bashri Sadia | |
| Test Case ID: TC\_2 | | Test Designed date: 15/01/2026 | |
| Test Priority (Low, Medium, High): Medium | | Test Executed by: Sadia | |
| Module Name: User Records Module | | Test Execution date: 16/01/2026 | |
| Test Title: Delete user from database | | | |
| Description: Test deleting data of a specific user from database | | | |
| Precondition: User must exist in database | | | |
| Dependencies: Admin log in | | | |
| Test Steps | Test Data | Expected Results | Actual Results / Status |
| 1. Log in with admin account 2. Go to user records 3. Select a user 4. Click delete | User Id: C001 | User must be deleted from database | Actual Result: As expected Status: Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: Car Dealership Management | | Test Designed by: Rabeya Bashri Sadia | |
| Test Case ID: TC\_3 | | Test Designed date: 16/01/2026 | |
| Test Priority (Low, Medium, High): High | | Test Executed by: Sadia | |
| Module Name: Car Records Module | | Test Execution date: 17/01/2026 | |
| Test Title: Verify adding new car | | | |
| Description: Verify admin can add a new car to database | | | |
| Precondition: The specific car is not already existed in the database | | | |
| Dependencies: if any | | | |
| Test Steps | Test Data | Expected Results | Actual Results / Status |
| 1. Log in with admin account 2. Go to Car records 3. Click add a new car 4. Fill up the must require data  5. Click submit | New Car Id: 002 | New car should be added to database and be available for customers | Actual Result: Car is not added in database Status: Failed |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: Car Dealership Management | | Test Designed by: Rabeya Bashri Sadia | |
| Test Case ID: TC\_4 | | Test Designed date: 16/01/2026 | |
| Test Priority (Low, Medium, High): Medium | | Test Executed by: Sadia | |
| Module Name: Create Account Module | | Test Execution date: 17/01/2026 | |
| Test Title: Verify successful account creation | | | |
| Description: Verify a customer can create a customer profile successfully | | | |
| Precondition: N/A | | | |
| Dependencies: N/A | | | |
| Test Steps | Test Data | Expected Results | Actual Results / Status |
| 1. Go to create account page 2. Fill the require data accurately 3. Click next 4. Insert unique username and password  5. Click confirm | New User Id: C002 | Successful message should be shown and user shall log in using the profile | Actual Result: As expected Status: Pass |

# 6. CONCLUSION

The Car Dealership Management System project was developed to demonstrate the effective application of software engineering principles in a real-life business environment. The main objective of this project was to design and develop a system that can automate and simplify the daily operations of a car dealership. The system successfully manages essential activities such as user registration and management, car inventory handling, order placement and approval, payment processing, and customer feedback collection. By automating these processes, the system significantly reduces manual effort, minimizes human errors, improves data accuracy, and enhances overall operational efficiency.

During the development of this project, the Scrum (Agile) process model was followed to ensure flexibility and continuous improvement. The project was divided into multiple sprints, allowing the team to develop, test, and refine features incrementally. This approach helped in handling changing requirements efficiently and ensured that each module was reviewed and improved through regular feedback. The Software Requirements Specification (SRS) played a crucial role in clearly defining the system scope, functional requirements, non-functional requirements, and user roles, which ensured that the development process remained structured and aligned with user needs.

The system was designed with a user-friendly interface and role-based access control to ensure ease of use and security for all types of users, including administrators, employees, and customers. Security considerations such as secure authentication, protected user data, and controlled access to system functionalities were carefully addressed. Extensive testing was performed on different modules of the system to verify correct functionality, identify defects, and evaluate system performance. Although some test cases failed, these issues highlighted areas that can be improved in future versions of the system.

Overall, this project provided valuable hands-on experience in requirement analysis, system design, Agile-based development, Git-based version control, and software testing. It enhanced the team’s understanding of collaborative software development and problem-solving in a real-world context. The Car Dealership Management System has strong potential for future enhancement, such as integrating advanced payment gateways, real-time sales analytics, notification systems, and mobile application support. With these improvements, the system can be scaled further to meet industry-level requirements and provide greater value to both dealership management and customers.