Group Task – 01 Assessment Report

Only for Course Teacher						
		Needs Improvement	Developing	Sufficient	Above Average	Total Mark
Allocate mark & Percentage		25%	50%	75%	100%	10
Functionality	04					
Code Quality	03					
Collaboration & Presentation	03					
					Total obtained mark	
Comments						

Semester:Spring 2025 Group No:

Name: MD. SOHAG MILON SHANTO **ID:242-35-405**

Name: MD.SHARIARE HOSSAIN SEEJAN **ID:242-35-513**

Name: NAFIS MAHMUD **ID: 242-35-857**

Batch: 43 **Section: C2** **Course Code:** SE 133

Course Name: Software Development Capstone Project

Course Teacher Name: MD. ABDUL HYE ZEBON **Designation:** Lecturer

Submission Date: 17 /07/25

Group Task – 01

SOFTWARE REQUIREMENT SPECIFICATIONS & SYSTEM

Software Requirements Specification (SRS) For Parking Lot Management System

1. Introduction

1.1 Purpose
The Parking Lot Management System automates the management of vehicle check-in and check-out, slot allocation, billing, and history tracking. The system aims to maximize slot utilization, reduce congestion, and streamline parking operations.

1.2 Scope
Developed in C programming language, this system handles vehicle entry, slot assignment, fee calculation based on parking duration, and persistent storage of all parking data through file handling.

1.3 Definitions, Acronyms, and Abbreviations

- **Slot**: A parking space assigned to a vehicle.
- **Admin**: Authorized personnel managing the system.
- **Billing**: Fee calculation based on parking duration.

1.4 References

- C Programming Language documentation
- Parking lot operational standards

---

2. Overall Description

2.1 Product Perspective
The system serves as a standalone C-based application managing parking logistics. It replaces manual processes with automated vehicle tracking, billing, and slot allocation.

2.2 Product Functions

- Vehicle Check-in: Assign slots based on availability and vehicle type.
- Vehicle Check-out: Calculate parking charges, generate bills, and release slots.
- Slot Tracking: Monitor slot availability in real-time.
- Vehicle History: Maintain logs of previous parking records.
- Admin Dashboard: View slot status, vehicle history, and generate reports.

2.3 User Characteristics

- **Vehicle Owners**: Park vehicles and pay fees.
- **Parking Staff**: Manage vehicle entries and exits.
- **System Admins**: Oversee system operations, reports, and data integrity.

2.4 Constraints

- Implementation limited to C programming with file handling for data persistence.
- Real-time operation dependent on sequential file access.
- Fixed number of parking slots predefined.

3. Specific Requirements

3.1 Functional Requirements

ID	Description	Priority
FR1	The system shall allow vehicle check-in by recording vehicle type, entry time, and assigning an available slot.	High

ID	Description	Priority
FR2	The system shall track parking slot availability and update in real-time.	High
FR3	The system shall calculate parking fees based on vehicle type and parking duration.	High
FR4	The system shall generate and display a bill at vehicle exit.	High
FR5	The system shall maintain a persistent vehicle history in files.	High
FR6	The system shall provide an admin interface to view slot availability and vehicle history.	Medium
FR7	The system shall release parking slots upon vehicle exit.	High

3.2 Non-Functional Requirements

ID	Description	Priority
NFR1	The system shall operate efficiently with minimal delay in slot assignment and billing.	High
NFR2	The system shall ensure data integrity through reliable file handling mechanisms.	High
NFR3	The system shall be designed for ease of maintenance and scalability.	Medium

3.3 Performance Requirements

- Slot assignment and billing operations must complete within a few seconds to avoid congestion.
- File access and updates should be optimized to support real-time usage.

3.4 Design Constraints

- Must be implemented using the C programming language.
- Persistent data storage must use file handling (no databases).

---

4. System Features

4.1 Vehicle Check-in

- Input: Vehicle type (car, bike, etc.), entry time.
- Process: Assign the nearest available slot based on vehicle type and availability.
- Output: Confirmation of slot assignment.

4.2 Vehicle Check-out

- Input: Vehicle identification (e.g., slot number or ticket number).
- Process: Calculate parking duration, compute fees, generate bill, release slot.
- Output: Billing receipt and slot status update.

4.3 Admin Interface

- View real-time slot availability.
- Search vehicle history logs.
- Generate usage and billing reports.

---

5. External Interface Requirements

5.1 User Interfaces

- Console-based menus for check-in, check-out, and admin operations.

5.2 Hardware Interfaces

- Parking gate control (optional, if integrated).
- Billing counters (for payment processing).

5.3 Software Interfaces

- File system for data storage.

---

6. Other Requirements

6.1 Data Management

- Use structured files to store vehicle logs, slot statuses, and billing records.
- Ensure backup and recovery procedures for files.

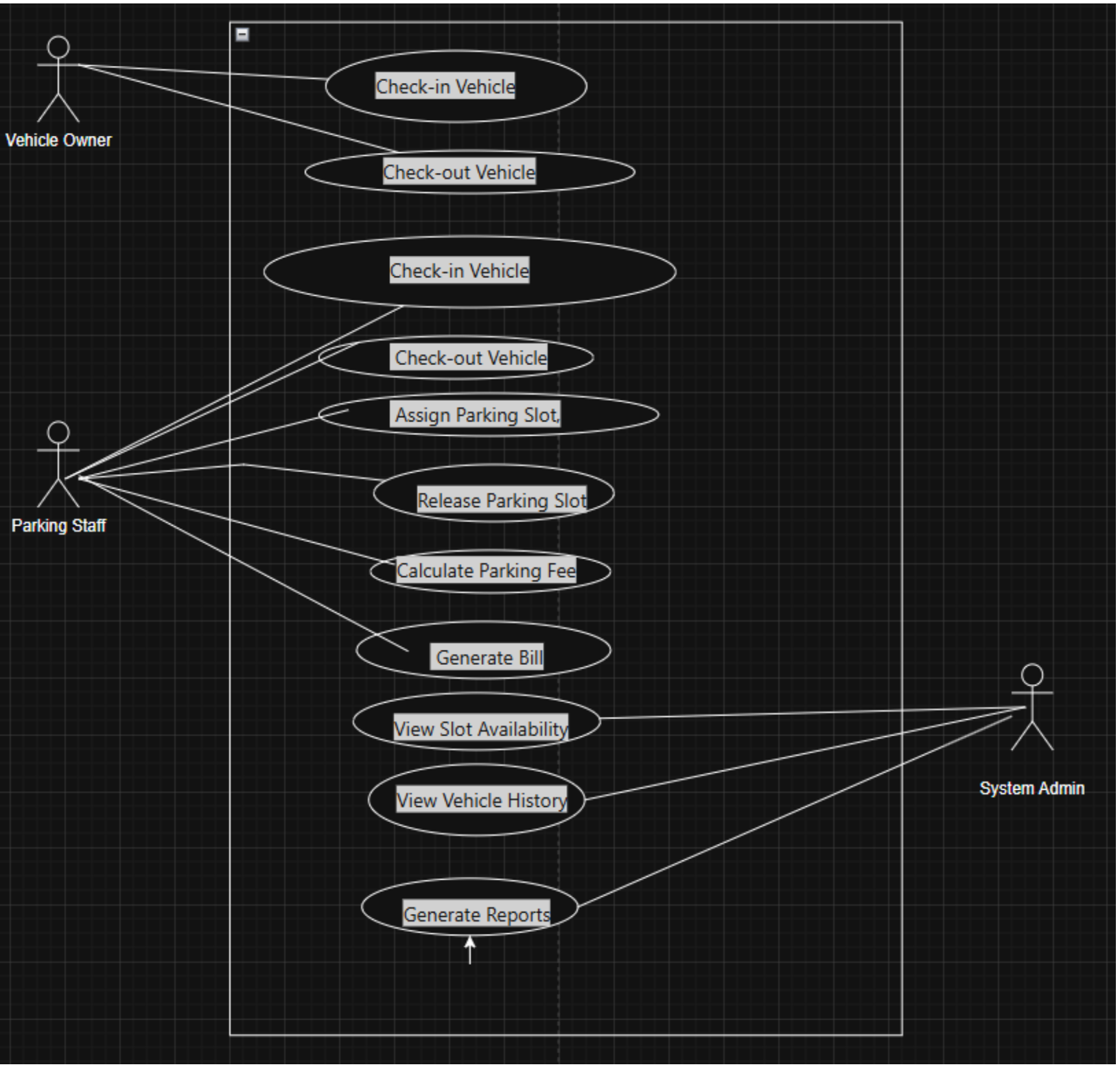
6.2 Security

- Admin access control for sensitive operations.
- Basic input validation to prevent data corruption.

7. Conclusion

The Parking Lot Management System is designed to efficiently automate vehicle access, slot allocation, and billing using C programming and file handling for persistent data storage. By addressing key functional requirements such as real-time slot tracking, fee calculation, and vehicle history management, the system aims to enhance operational efficiency, reduce congestion, and optimize parking slot utilization. With consideration for stakeholder needs and practical constraints, this solution provides a scalable and maintainable approach to managing parking logistics in a controlled environment.

USE CASE DIAGRAM


Vehicle Owner
Check-in Vehicle
Check-out Vehicle
Check-in Vehicle
Check-out Vehicle
Assign Parking Slot,
Parking Staff
Release Parking Slot
Calculate Parking Fee
Generate Bill
View Slot Availability
View Vehicle History
Generate Reports
System Admin