Parking Garage Project Phase 1

**Parking Garage Project Phase 1**

**Software Requirements Specification**

# **Revision History**

| **Date** | **Revision** | **Description** | **Author** |
| --- | --- | --- | --- |
| 2/22/2025 | 1.0 | Started SRS document, finished Purpose and half of overall description. - 3hrs | Kurt, Vishal, and Raymond |
| 3/1/2025 | 2.0 | Finished SRS Document and UML Class Sequence diagrams - 6hrs | Kurt, Vishal, and Raymond |

**Github Link:**

<https://github.com/vishalVasanthakumarPoornima/Parking-Garage-Group-7>

# **Table of Contents**

[**Revision History 2**](#_ffcjvve8rkh1)

[**Table of Contents 3**](#_cl1tmj2disvg)

[**1. Purpose 5**](#_1fob9te)

[1.1. Scope 5](#_3znysh7)

[1.2. Definitions, Acronyms, Abbreviations 5](#_2et92p0)

[1.3. References 5](#_tyjcwt)

[1.4. Overview 5](#_3dy6vkm)

[**2. Overall Description 6**](#_1t3h5sf)

[2.1. Product Perspective 6](#_4d34og8)

[2.2. Product Architecture 6](#_2s8eyo1)

[2.3. Product Functionality/Features 6](#_17dp8vu)

[2.4. Constraints 7](#_3rdcrjn)

[2.5. Assumptions and Dependencies 7](#_lnxbz9)

[**3. Specific Requirements 9**](#_35nkun2)

[3.1. Functional Requirements 9](#_1ksv4uv)

[3.1.1. Common Requirements: 9](#)

[3.1.2. Ticketing System Requirements: 9](#)

[3.1.3. Parking Garage Requirements: 9](#)

[3.1.4. Parking Logs Requirements: 10](#)

[3.2. External Interface Requirements 10](#_4i7ojhp)

[3.3. Internal Interface Requirements 10](#_1ci93xb)

[**4. Non-Functional Requirements 11**](#_qsh70q)

[4.1. Security and Privacy Requirements 11](#_3as4poj)

[4.2. Environmental Requirements 11](#_49x2ik5)

[4.3. Performance Requirements 11](#_23ckvvd)

[1. Use Case: Login and Logout 12](#_yoxt4wq97c99)

[2. Use Case: Ticket Handling 12](#_z891yjif5shp)

[3. Use Case: Handling Parking Data 13](#_xx4u9rbq1r0m)

[4. Use Case: Payment Processing 13](#_lpslftu354v7)

[5. Use Case: Automatic vs. Manual Payment Handling 14](#_xppx7ykbgzqe)

[1. Use Case: Login and Logout 15](#_9ipg5hxlouoe)

[2. Use Case: Ticket Handling 15](#_3qzgzd8m47zm)

[3. Use Case: Handling Parking Data 16](#_1j1wt3fyo6e)

[4. Use Case: Payment Processing 16](#_gynrsha139tv)

[5. Use Case: Automatic vs. Manual Payment Handling 17](#_si2md9gimctk)

[4. Brainstorming: 18](#_oibo3e11tx0s)

[ParkingGarage 18](#_xfm3bpp5joon)

[ParkingSpace 18](#_uzgvotsu8gkl)

[Vehicle 18](#_mbdkcxqrls1q)

[Ticket 18](#_xutuc7ed8tjn)

[Payment 19](#_k4zy2ovy924n)

[Employee 19](#_k08pizpl8oau)

[Customer 19](#_7dc2bs4t29e0)

[SystemLog 20](#_arqatyf5rcnx)

[Class Diagram: 21](#_jh0667uqctlr)

[Sequence Diagram - Employee Side 23](#_f3p3yce0r79p)

# **Purpose**

This document outlines the requirements for the Parking Garage Group Project.

## **Scope**

This document will catalog the user, system, and hardware requirements for the Parking Garage System. It will not, however, document how these requirements will be implemented.

## **Definitions, Acronyms, Abbreviations**

Important terms:

* + 1. **Parking Garage Management System (PGMS):** The software and hardware system used to manage parking spaces, transactions, and customer entry/exit in a parking facility.
    2. **Entry/Exit Log:** A record of vehicles entering and leaving the parking garage, including timestamps and vehicle details.
    3. **Multithreading:** A technique used to allow multiple tasks to run concurrently within the same program, improving performance and efficiency.
    4. **Payment Gateway:** A system that processes financial transactions, allowing customers to pay for parking services.
    5. **Server:** The central computing unit that manages and coordinates the operations of the parking garage system, including handling multiple garages and transactions.
    6. **Thread:** A lightweight process that performs tasks concurrently with other threads in the system, used to handle multiple customer actions simultaneously.
    7. **Employee:** A worker who oversees the system and its responsibilities in managing the network of parking garages.
    8. **Unified Modeling Language (UML)**: Modeling used to visualize system design.

## **References**

Use Case Specification Document

UML Use Case Diagrams Document

Class Diagrams

Sequence Diagrams

## **Overview**

The Parking Garage Management System (PGMS) is designed to efficiently manage parking operations across multiple parking garages. The system monitors available parking spaces, processes and logs payments, and ensures proper management of the garage's capacity. It supports various payment methods and handles concurrent customer interactions through multithreading to ensure smooth operation even during peak times. Additionally, the system provides employees with real-time data on garage occupancy, payment status, and vehicle movement. The goal is to enhance the customer experience while maintaining efficient use of parking spaces and generating revenue for the facility.

# **Overall Description**

## **Product Perspective**

* + 1. The Parking Garage Management System (PGMS) operates as a standalone, networked software application designed to facilitate the management of a network of self-parking garages. It consists of a client-server architecture running over TCP/IP and provides an interactive graphical user interface (GUI) for customers and employees.

## **Product Architecture**

The system will be organized into 3 key major modules:

1. **The Ticketing System:** simultaneously reserves a spot for multiple customers and checks them out
2. **The Parking Garage:** an independent facility on the server side allowing independent tracking of parking availability , payments, and vehicle logs for each location.
3. **Parking Logs:** The history or logs of all tickets, including timestamps.

These modules will follow standard Object-Oriented design principles by sending and receiving data between each other and use that to control the parking garages. For example, the ticketing module will use the parking availability information for the corresponding garage within the system to see whether a new car should be allowed in.

## **Product Functionality/Features**

The **Parking Garage Management System** (PGMS) includes the following high-level features to ensure efficient operation, security, and a smooth customer experience:

1. **User Authentication and Access Control**
   * Secure login functionality for employees with unique usernames and passwords.
   * Only authorized employees can process payments, access usage reports, and manage parking garage operations.
   * Account locking after multiple failed login attempts to enhance security.
2. **Parking Space Monitoring and Availability**
   * Real-time tracking of available parking spaces in the garage.
   * Continuous updates to parking space availability after customer entry/exit.
   * System prevents overcapacity by ensuring that only available spaces are assigned to customers.
3. **Payment Processing**
   * Support for various payment methods, including credit/debit cards and cash.
   * Automated payment based on the duration of stay, with an option for manual payment processing by employees for customer assistance.
   * Real-time payment processing and immediate space updates after each transaction.
4. **Graphical User Interface (GUI)**
   * A user-friendly GUI for employees to manage parking garage operations.
   * Displays parking space availability, payment options, and fees.
   * Real-time updates to parking availability and payment status over TCP/IP for both server and client applications.
5. **Concurrent Client Handling**
   * The system must handle multiple customers simultaneously, processing transactions in parallel using multithreading.
   * Ensures that each customer’s payment and space update is processed independently without delays or data conflicts.
   * Proper synchronization mechanisms are in place to avoid data corruption in shared resources.
6. **Employee Login and Access Control**
   * Employees must log in with a unique ID and password for secure access to the system.
   * Only authorized employees can process payments and access usage reports.
7. **Usage Reporting for Management**
   * The system must generate daily, weekly, and monthly usage reports, including total revenue, peak times, and space utilization.
   * Tracks performance metrics, efficiency, and cost management to assist in parking garage administration and decision-making.
8. **Multi-Garage Management**
   * The server supports managing multiple parking garages at once, allowing independent tracking of each garage’s operations while synchronizing data across all locations.
   * Employees can manage multiple garages and view real-time data from all locations in a unified interface.
9. **Error Handling and System Diagnostics**
   * **Error Detection**: The system will automatically detect and log errors related to payment processing, server-client communication, or system operations.
   * **Error Logging**: Errors are logged with details (e.g., type, time, transaction ID) for troubleshooting and future reference.
   * **Alerts**: Alerts are sent to employees when critical errors (e.g., payment issues or communication failure) are detected, enabling quick action.
   * **System Monitoring**: Continuous system health checks to identify and address issues with server operations, TCP/IP connections, and more.
10. **~~Response Time and System Performance~~**
    * ~~The system must provide real-time processing including parking entry, exit, fee calculation, and payment processing.~~
    * ~~Updates to parking availability and transaction data must be presented to clients with minimal delay.~~
    * ~~The system must maintain responsive performance under normal and peak load conditions.~~

## **Constraints**

* + 1. Since multiple employees might be logged in at the same time, the system should ensure that transactions are processed accurately without conflicts.
    2. The system must keep track of all parking transactions, including entry time, exit time, and payments, storing them in a log file for quick reference.
    3. The system must handle TCP/IP errors and prevent unauthorized modifications to transaction data.
    4. The system should operate continuously during parking garage business hours, and any failures should be logged with an error message for debugging.

## **Assumptions and Dependencies**

* + 1. The system will support a maximum of 200 concurrent users (including customers and employees) per parking garage location, based on expected usage patterns.
    2. Parking entry and exit terminals will be properly configured and calibrated before deployment to ensure accurate ticket generation and tracking.
    3. The system will have a stable power supply to maintain continuous operation of the parking terminals, payment stations, and server.
    4. All sensors and automated barriers will be installed in strategic locations to ensure proper functionality without false detections or errors.
    5. The system relies on local TCP/IP networking for communication between client terminals and the central server.
    6. The accuracy of parking space availability tracking depends on proper usage by customers and enforcement by employees.
    7. The system requires regular maintenance to ensure all hardware (kiosks, ticket scanners, payment terminals) remains functional.
    8. Employees must be properly trained to operate the system and assist customers with payments and troubleshooting when necessary.

# **Specific Requirements**

## **Functional Requirements**

### **Common Requirements:**

* + - 1. Payment Method
      2. Parking Duration
      3. Parking Space Tracking
      4. Concurrent Client Handling
      5. Graphical User Interface
      6. Employee Login
      7. Usage Report
      8. Multi Garage Management
      9. Error Handling
      10. ~~Response Time~~

### **Ticketing System Requirements:**

* + - 1. The system must generate a unique **ticket ID** for each vehicle entering the parking garage.
      2. The ticket must store the **entry time, ticket ID, and associated parking garage ID**.
      3. The system must automatically **reserve a parking spot** upon issuing a ticket, ensuring available space before allowing entry.
      4. Customers must provide their ticket at exit for **fee calculation based on duration**.
      5. The system must allow employees to manually override a ticket in case of lost or damaged tickets.
      6. The system must prevent **duplicate ticket issuance** for a single vehicle within the same session.
      7. The ticketing module must communicate with the **parking garage module** to verify space availability before generating a new ticket.
      8. The system must remove the reservation upon checkout to free up the parking space.

### **Parking Garage Requirements:**

* + - 1. Each parking garage must operate **independently** while being managed under a unified system.
      2. The system must track the **total number of available and occupied parking spaces** for each garage.
      3. The system must **update parking space status** in real time when vehicles enter or exit.
      4. Each parking garage must maintain a **separate vehicle log** to track parked cars, including their ticket ID and entry time.
      5. The system must allow for **multiple parking garages to be managed simultaneously**, keeping data separate for each.
      6. The system must **verify ticket validity** before allowing a car to exit.
      7. Payments must be handled per garage, ensuring that transactions are properly recorded under the correct facility.

### **Parking Logs Requirements:**

* + - 1. The system must maintain a **persistent log of all parking transactions**, including timestamps for entry, exit, and payments.
      2. The system must store **ticket history**, including ticket ID, entry time, exit time, and amount paid.
      3. The log system must support **daily, weekly, and monthly reports** for garage usage statistics.
      4. Employees must be able to retrieve **past transaction logs** based on ticket ID, date, or vehicle information.
      5. The system must prevent unauthorized **modifications or deletions** of existing logs.
      6. Logs must be stored in a structured format to enable **efficient searching and filtering**.
      7. The system must provide alerts for **unresolved tickets** where a vehicle has entered but has not yet checked out.

## **External Interface Requirements**

* + 1. The system must be able to communicate with parking sensors to detect vehicle entry, exit, and individual parking space occupancy in real time.
    2. The system should provide a graphical user interface (GUI) accessible on a standard computer for employees to monitor parking availability and process payments.
    3. The system must generate parking logs that can be exported in a structured text format for external review and reporting.
    4. The ticketing system must interface with automated ticket dispensers to issue tickets upon vehicle entry.
    5. The payment system must support integration with external payment processing hardware (e.g., card readers, cash terminals) to accept customer payments.

## **Internal Interface Requirements**

* + 1. The ticketing module must communicate with the parking garage module to check space availability before issuing a ticket.
    2. The parking garage module must update the parking logs module whenever a vehicle enters or exits the facility.
    3. The payment processing system must retrieve ticket data from the ticketing module to calculate the correct fee.
    4. The employee interface must allow authorized personnel to monitor parking availability, process transactions, and review logs with role-based access.
    5. The system must allow multiple parking garages to function independently while sharing a common management interface.

# **Non-Functional Requirements**

## **Security and Privacy Requirements**

* + 1. **Access Control:** Only authorized employees can access administrative functions, while customers are limited to self-service features.
    2. **Data Protection:** All transaction records, parking logs, and employee credentials must be securely stored.
    3. **Secure Communication:** Client-server communication must be protected against interception or tampering using secure protocols.
    4. **Audit and Logging:** All system activities must be logged, with access restricted based on user roles for employees and auditing.

## **Environmental Requirements**

* + 1. **Operational Reliability:** The system must function reliably withstanding varying weather conditions such as heat, cold, and humidity.
    2. **Hardware Compatibility:** The user interface should be accessible on standard computers and kiosks with minimal hardware requirements.
    3. **Network Dependency:** The system must operate efficiently over local TCP/IP networks, ensuring that critical functions like ticket validation and payment processing remain uninterrupted during temporary connectivity issues.

## **Performance Requirements**

* + 1. **Transaction Processing:** The system must process parking entry, exit, and payment transactions with minimal time to ensure a smooth user experience.
    2. **Real-Time Updates:** Parking space availability and fee calculations must be updated immediately when a vehicle enters or exits.
    3. **Logging Efficiency:** All transactions, login attempts, and system events must be recorded and stored without delay for accurate tracking and auditing.
    4. **User Interface Responsiveness:** The system's GUI must load and display critical parking and payment information with minimal delay of user interaction.