

*Software Requirements Specification*  
*Group 1*

## Revision History

[illegible]



# Table of Contents

<b>1. PURPOSE.....</b>	<b>4</b>
1.1. SCOPE.....	4
1.2. DEFINITIONS, ACRONYMS, ABBREVIATIONS.....	4
1.3. REFERENCES.....	4
1.4. OVERVIEW.....	4
<b>2. OVERALL DESCRIPTION.....</b>	<b>5</b>
2.1. PRODUCT PERSPECTIVE.....	5
2.2. PRODUCT ARCHITECTURE.....	5
2.3. PRODUCT FUNCTIONALITY/FEATURES.....	5
2.4. CONSTRAINTS.....	5
2.5. ASSUMPTIONS AND DEPENDENCIES.....	5
<b>3. SPECIFIC REQUIREMENTS.....</b>	<b>6</b>
3.1. FUNCTIONAL REQUIREMENTS.....	6
3.2. EXTERNAL INTERFACE REQUIREMENTS.....	6
3.3. INTERNAL INTERFACE REQUIREMENTS.....	7
<b>4. NON-FUNCTIONAL REQUIREMENTS.....</b>	<b>8</b>
4.1. SECURITY AND PRIVACY REQUIREMENTS.....	8
4.2. ENVIRONMENTAL REQUIREMENTS.....	8
4.3. Performance Requirements.....	8

# Purpose

This document outlines the requirements for the Parking Garage Ticketing System (PGTS).

## 1.1. Scope

This document will catalog the customer, software system, and hardware requirements for the PGTS system. It will not, however, document how these requirements will be implemented.

## 1.2. Definitions, Acronyms, Abbreviations

- Parking Garage Ticketing System (PGTS)
- Graphical User Interface (GUI)
- Refresh Rate
  - The frequency with which the image on a computer monitor or similar electronic display screen is refreshed, usually expressed in hertz.
- Usage Reports
  - A report that displays critical information such as peak times, customer data and gross profit.
- Over Limit Fee
  - A fee that is implemented to encourage customers to adhere to the garage policy and pay the necessary amount for their stay

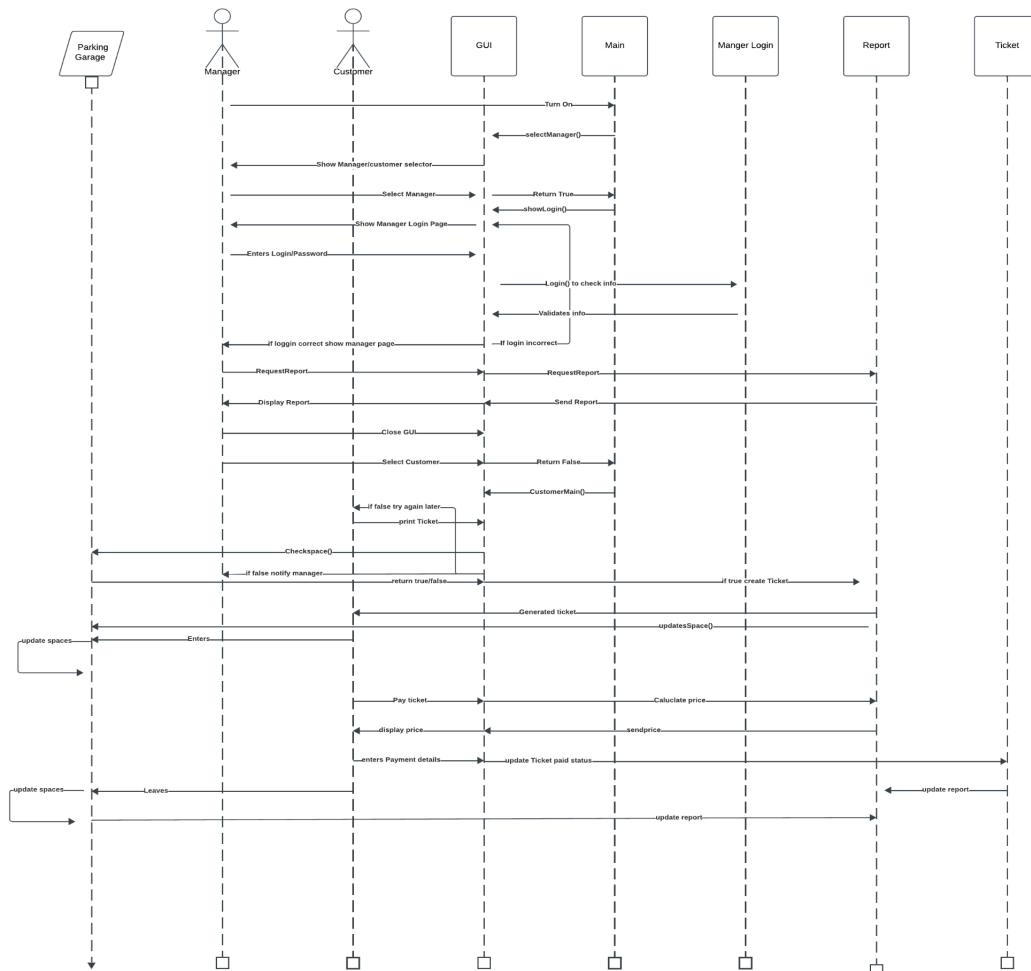
## 1.3. References

Use Case Specification Document – Step 2 in assignment description

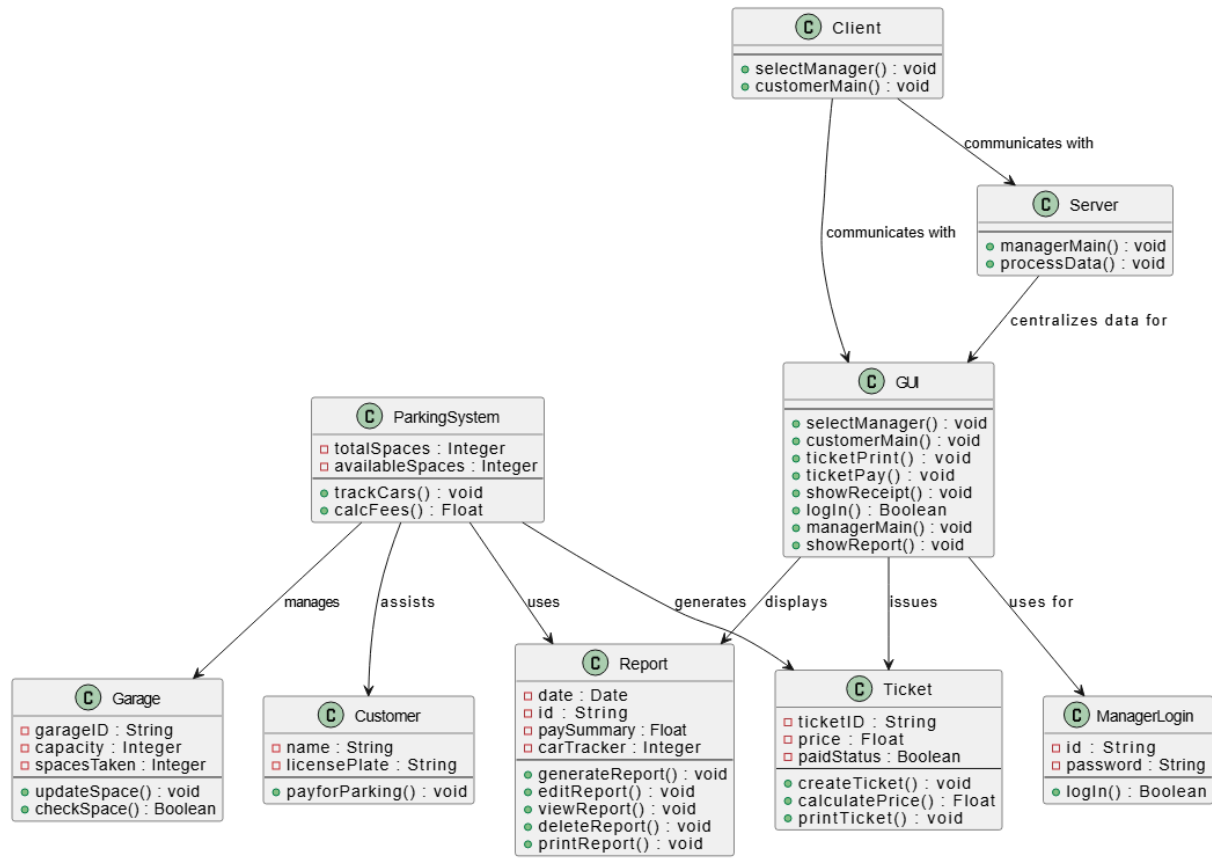
UML Use Case Diagrams Document – Step 3 in assignment description

Class Diagrams – Step 5 in assignment description

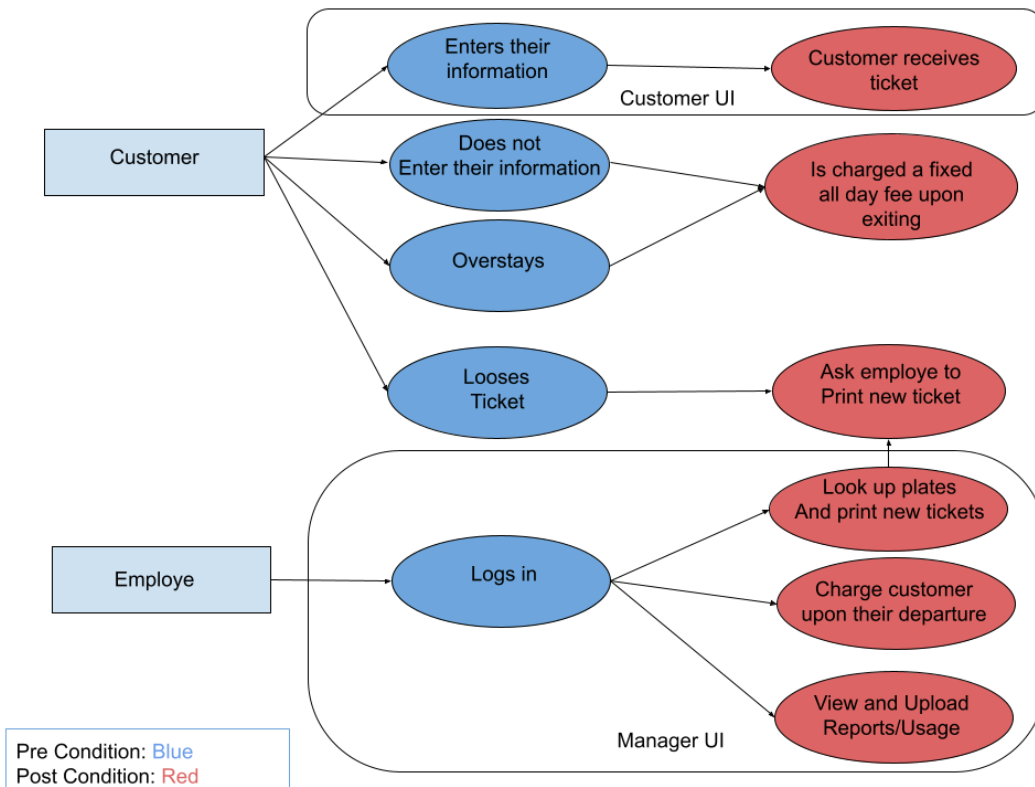
## Sequence Diagram:



## Class Diagram:



### Use Case Diagram:



### **Use Cases:**

**Use Case ID:** UC001

**Use Case Name:** Customer enters garage and pays (R0, R4, R5, R7, R8)

**Primary Actor:** Customer

**Pre-conditions:** none

**Post-conditions:** Customer successfully enters garage and pays

Basic Flow or Main Scenario:

1. Customer enters garage at the front gate and collects entry ticket
  - a. Ticket has unique ID and timestamp
2. Customer parks and reaches the kiosk to pay
  - a. Customer enters unique ID
  - b. Customer proceeds to choose their choice of stay from our time limit menu
3. Customer enters their credit card and successfully pays
4. A completed ticket is printed with the Unique ID, Timestamp of successful transaction, transaction ID, and their choice of stay

**Use Case ID:** UC002

**Use Case Name:** Customer pays at exit gate (R5, R8, R10)

**Primary Actor:** Customer

**Pre-conditions:** Customer does not pay at the kiosk before leaving

**Post-conditions:** Customer successfully pays at the exit gate and leaves

Basic Flow or Main Scenario:

1. Customer chooses not to pay at the kiosk before leaving for various of reasons
2. Customer arrives at exit gate without a completed ticket
3. Customer is forced to pay the “All day” price since they did not pay kiosk
  - a. This gives the system an easy method of collecting pay
4. Customer pays the fee and then successfully leaves the garage



**Use Case ID:** UC003

**Use Case Name:** Customer overstays (R4, R5, R9, R10)

**Primary Actor:** Customer

**Pre-conditions:** Customer overstayed their original choice

**Post-conditions:** Customer successfully pays fee and leaves the garage

Basic Flow or Main Scenario:

1. Customer overstayed their original choice of stay
  - a. If customers overstay less than 2 hours over their choice, then they will be charged double their original price.
    - i. If customer chose 2 hours to stay for \$6 and actually stayed 3 hours then they will be charged \$12
  - b. If customer overstay more than 2 hours over their choice, then they will be charge the “All day fee”
    - i. If customer chose 2 hours to stay for \$6 and actually stayed more than 4 hours, then they be charged \$25
2. Customer pays at kiosk to successfully resolve the over charge
  - a. If a customer does not pay at the kiosk and instead at the gate, they are charged the all day fee regardless how long they stayed at the garage.
3. Customer Successfully leaves the garage

**Use Case ID:** UC004

**Use Case Name:** Management Login (R11)

**Primary Actor:** Management

**Pre-conditions:** None

**Post-conditions:** Successful login

Basic Flow or Main Scenario:

1. Management provides valid UserID and Password
2. System checks validate the entry
  - a. If user enters valid entry then proceed to step 3
  - b. If user enter invalid entry thenretrystep2
3. System Grants login for Management

**Use Case ID:** UC005

**Use Case Name:** Management checks reports (R11, R12, R13, R14)

**Primary Actor:** Management

**Pre-conditions:** Successful login

**Post-conditions:** Able to view data

Basic Flow or Main Scenario:

1. Management logs in with correct credentials
2. Management clicks drop menu to choose what report they want to see

**Use Case ID:** UC006

**Use Case Name:** Management prints lost ticket (R4, R5, R11, R15)

**Primary Actor:** Management and customer

**Pre-conditions:** Successful login and lost ticket by customer

**Post-conditions:** Able to print new ticket

Basic Flow or Main Scenario

1. Customer loses ticket and talks to management company
2. Management logs in with credentials
3. Management prints new ticket for customer by using their last 4 digits of the card used to make the payment

**Use Case ID:** UC007

**Use Case Name:** GUI interaction (R?)

**Primary Actor:** Management and customer

**Pre-conditions:** none

**Post-conditions:** Able to interact with GUI

Basic Flow or Main Scenario

1. Customer and manager interacts with GUI (terminal)
  - a. Customer has the option to see drop down menu for pricing of the garage
  - b. Manager has the option to login and view / print reports
2. Customer and manager complete their desired requests
  - a. Customer is able to print their ticket and complete their purchase
  - b. Manager is able to view / print reports

## **1.4. Overview**

The Parking garage ticketing system (PGTS), is designed to print tickets and collect payment from customers. As operations increase throughout the day, so does the risk of traffic jams and loss of income. This system will assist in the garage's overall effectiveness ensuring a smooth operation for the parking garage management team and customer satisfaction

# Overall Description

## 2.1. Product Perspective

The **Parking Garage Ticketing System (PGTS)** is a comprehensive solution that streamlines the process of managing parking tickets and payments, enabling parking garage operators to optimize their daily operations and enhance customer experience. The PGTS stands out in the market by offering a fully integrated solution that not only automates ticketing and payments but also actively helps prevent revenue loss and congestion. Its adaptability and ease of use make it an ideal choice for parking facilities ranging from small garages to multi-level complexes.

## 2.2. Product Architecture

The software will be created by utilizing the java coding language. By using object oriented programming with java, we are creating multiple classes that contain methods which will provide the software functionality based on our requirements. To keep a well organized system we will follow 3 modules: Parking Management Module, Fee Calculation Module and Garage Management Module.

## 2.3. Product Functionality/Features

The high-level features of the system are as follows (see section 3 of this document for more detailed requirements that address these features):

## 2.4. Constraints

### 2.4.1 Car and Parking Space Accuracy

- The software should be able to detect how many cars are entering and leaving the parking garage with precision. The margin of error should be within  $\pm 1\%$ .

### 2.4.2 Daily Reporting

- The system should be keeping track of how many cars are entering and leaving each day.
- The amount of money should be calculated at the end of the day.

### 2.4.3 Performance

- The system should be able to handle multiple users at a time (if we decide to do two lanes coming and out).
- The response time for the user interface (getting tickets, selecting stay times, inserting tickets) must not exceed 10 seconds under normal conditions.

### 2.4.4 Reliability

- The system must operate 24/7 without downtime and have a failure tolerance of 99.99%.

## 2.5. Assumptions and Dependencies

### 2.5.1 Uninterrupted power supply

- The system assumes an uninterrupted power supply for continuous operation. In case of a power outage, the system should display a message that the customer needs to visit the garage office to purchase and receive a ticket in person.

### 2.5.2 Timely Maintenance

- Regular maintenance and updates for the software. Any lack of maintenance could affect the accuracy and reliability of the system.

### 2.5.3 User Proficiency:

- The parking garage staff and management team have a basic level of technological proficiency, enabling them to easily navigate and utilize the system's functionalities with minimal training

## Specific Requirements

### 3.1. Functional Requirements

#### 3.1.1 Common Requirements:

##### 3.1.1.1 Functionality

- The software must perform the required tasks as expected and provide the desired functionalities to meet the needs of its customers.
- The system should offer features and capabilities that allow customers to complete their transactions with minimal effort and complexity.

##### 3.1.1.2 Performance

- The software must operate with acceptable performance levels, such as speed, responsiveness, and resource usage.
- The system should handle tasks efficiently without excessive delays, and it should perform well under typical and peak loads.

##### 3.1.1.3 Usability

- The software must be easy to use and intuitive for the end users.
- The interface should be simple, with clear navigation, well-labeled buttons, and instructions where needed.

##### 3.1.1.4 Training

- The parking garage attendant should be trained on how to handle situations where the system either malfunctions or is powered off.
- Needs to know to reset the system with very little complexity
- Needs to know how to use the system such as printing reports and tickets

#### 3.1.2 Parking Management Module Requirements:

##### 3.1.2.1 Tracking Sensor

- Tracks the number of cars parked and available spaces in the parking garage and each floor.
- Subtracts the max allowed parking spaces from the taken spaces to show the current live available spaces.

##### 3.1.2.2 Update Function

- Parking availability in real-time as cars enter or exit. (This only works in a parking garage setting)

##### 3.1.2.3 Capacity Alert

- An automated alert for the management team to know when capacity is rising.

##### 3.1.2.4 Check-Out

- Includes a timestamp recording and receipt of the transaction.

##### 3.1.2.5 Assigned IDs

- Assign customers when they purchase a ticket an unique ID and transaction number to ensure accurate record keeping.

##### 3.1.2.6 Data Recovery

- Every 10 minutes the software should record the transaction ID and unique Ticket ID onto a text file for the management team just in case of hardware / software malfunction.

### **3.1.3 Fee Calculation Module Requirements:**

- 3.1.3.1 Choice of Stay
  - Gives the customer choices based on their preference on how long they want to park - 1hr, 2hr, 4hr, all day, etc.
- 3.1.3.2 Collect Payment
  - Calculates the price based on the users choice of parking selection and collects the payment.
- 3.1.3.3 Over Limit Fee
  - If a customer decides to overstay their limit, a fee will be generated and the customer needs to pay before leaving.

### **3.1.4 Garage Management Module Requirements:**

- 3.1.4.1 Management Login
  - This role requires a login with a username and password.
  - This user has the power to access daily summaries and end of day reports.
  - This user has the power to manually print a new ticket and collect payment from customers.
- 3.1.4.2 On Demand Summary
  - Generates various usage and financial reports. Reports the garage's activity at any time when demanded by the parking garage owner.
- 3.1.4.3 View IDs
  - Prints the unique ticket and transaction IDs in case for a review.
- 3.1.4.4 End of Day Report
  - Prints at the end of the day of: how many cars arrived at the garage, average time spent, busiest and slowest hours, and the total amount made for the day.

## **3.2. External Interface Requirements**

- 3.2.1 User Interface - customer check out:
  - The system should display a menu screen where customers can select their choice of stay and purchase their ticket
- 3.2.2 Management Interface:
  - The system should display a section on the screen that provides a login screen for the management team
- 3.2.3 Report Interface:
  - The management can view daily, weekly, monthly, and on the demand reports of the garage statistics.
- 3.2.4 Print ticket interface
  - This interface provides a screen where management can print a ticket for the customer on demand.
- 3.2.5 Graphical User Interface (GUI)
  - This interface provides a screen to the customer and manager to provide them access to the necessary function to complete their desired actions

### **3.3. Internal Interface Requirements**

#### **3.3.1 Capacity Alert Interface :**

- The interface that tracks the cars coming in and out of the garage. When the counter detects specific car capacity levels, it will send out an alert to the management team to take precautions

#### **3.3.2 Tracking Sensor interface:**

- The interface will track the cars parked and available spaces on each floor of the garage and display them on a screen so the customers can make a decision based on this information

#### **3.3.3 Update garage interface:**

- The interface will display on a screen to the customers before they enter the garage how many parking spots are left in whole. This will give the customer a good sense if the garage is slow or busy.



# Non-Functional Requirements

## 4.1. Security and Data Integrity Requirements

- 4.1.1 Data Backup
  - In case of a situation where the software crashes and erases all existing data, there should be an external file that handles data storage to ensure the flow business is not interrupted.
- 4.1.2 Security
  - The software needs to recognize that the customer made a purchase and then it can proceed to the next step of printing a ticket.
  - If the customer hasn't paid, the software will not raise the gates for the customer to leave (print message saying so).

## 4.2. Environmental Requirements

- 4.2.1 Dust and particles
  - All hardware components, including sensors and vending machine, must be protected against dust and small particles that are common in an outdoor setting.
- 4.2.2 Power Stability
  - The hardware of the system should operate on a stable and reliable power source to ensure that machines are running at full capacity at all times.

## 4.3. Performance Requirements

- 4.3.1 GUI Responsiveness
  - The system's GUI must respond to user inputs (e.g., button clicks, form submissions) within 500 milliseconds to provide a smooth user experience.
- 4.3.2 Fee Calculation
  - Parking fee calculations must be completed in less than 200 milliseconds for each customer when requested.
- 4.3.3 Usage Reports
  - Generating usage reports (e.g., daily, weekly, or monthly reports) should not take more than 5 seconds, even for large datasets.
- 4.3.4 Parking Availability and Car Count
  - must be updated across all clients in real time (within 1 second) when cars enter or leave the garage.
- 4.3.5 Refresh Rate
  - Our software should be refreshing at minute intervals to display the accurate amount of available spaces to customers to avoid traffic congestion and prevent overparking.
- 4.3.6 Peak Handling
  - The software should be able to handle multiple customers at a time and rapid transaction rates.