

Software++ (Group 15)

Parking Lot++
Requirements Document
with Use Cases
Version 2.0

October 31st, 2021

Document History

Version	When	Who	What
0.1	14/10/2021	Nilay Sondagar	Copied over <i>Project Charter</i> & reformatted document
0.2	17/10/2021	Nilay Sondagar	Added assorted requirements
0.3	17/10/2021	Abdullahi Bashir	Added requirements
0.4	18/10/2021	Logan Raffkind	Added requirements
0.5	18/10/2021	Vyom Shah	Added requirements
0.6	18/10/2021	Franklin Yang	Added requirements
0.7	18/10/2021	Dan Melamed	Added requirements
1.0	19/10/2021	Everyone	Final Review
1.1	17/10/2021	Dan Melamed	Added use cases
1.2	17/10/2021	Abdullahi Bashir	Added use cases
1.3	18/10/2021	Logan Raffkind	Added use cases
1.4	18/10/2021	Nilay Sondagar	Added use cases and incorporated client feedback
1.5	18/10/2021	Franklin Yang	Added use cases
1.6	18/10/2021	Vyom Shah	Added use cases and <i>Use Case Diagram</i>
2.0	19/10/2021	Everyone	Final Review

Table of Contents

1 Introduction	3
1.1 Project Overview	3
1.1.1 Executive Summary	3
1.1.2 Context	3
1.1.3 Need	3
1.1.4 Scope	4
1.1.5 Stakeholders	4
1.1.6 Objectives	5
1.1.7 Glossary	6
1.2 Project Approach	7
1.2.1 Team Organization & Roles	7
1.2.2 Work Breakdown Structure	7
1.2.3 Milestones	8
1.2.4 Project Schedule	8
1.2.5 Deliverables	8
1.2.6 Risks	10
1.3 Project Approval	11
1.3.1 Stakeholders Sign-off	11
1.3.2 Team Members Sign-off	11
2 Requirements	12
2.1 Functional Requirements	12
2.1.1 Map View Requirements	12
2.1.2 Registration System Requirements	13
2.1.3 User Settings Requirements	13
2.1.4 Notification System Requirements	14
2.1.5 Administrator Portal Requirements	15
2.2 Nonfunctional Requirements	16
2.2.1 Performance & Scalability Requirements	16
2.2.2 Security Requirements	17
2.2.3 Usability Requirements	17
2.3 Use Cases	18
2.4 Use Case Diagram	29

1 Introduction

The entirety of this project's *Project Charter* can be found in this section.

1.1 Project Overview

1.1.1 Executive Summary

The goal for Software++ (The Analysts) is to design and develop a website (The System) for The University of Faketoria (The Client) that allows end users to find parking lots with available parking spots in real-time. The Client's current system only provides static maps of their parking lots and lacks any way to determine parking lot availability. The System will help end users find parking lots for their vehicles, display a map of all parking lot locations, show the number of available parking spots, and send notifications to end users when their preferred parking lot has availability. This System is expected to complete the requirements gathering phase by November 28, 2021, with development starting shortly after that. The System's full rollout is expected to commence in late 2022, and will be officially referred to as *Parking Lot++*.

1.1.2 Context

The Client provides multiple parking lots on their campus that allows students, staff, faculty members, and visitors to park their vehicles. This allows for easy access to The Client's campus, buildings, and amenities. The Client uses ticketing booths in each parking lot to allow drivers to pay for a parking spot. The Client has multiple options for parking such as *General Parking*, *Long Term Parking*, *Short Term Parking*, and *Reserved Parking*. The Client provides maps physically around campus and virtually on their website. These maps show all parking lot locations on campus, with no information about the availability of parking spots.

1.1.3 Need

The Client's current parking lot setup includes payment booths and maps that indicate the location of each parking lot. However, these maps simply display the location of each lot, and no additional information such as the availability of parking spots. This leads to difficulties in finding parking spots, heavy traffic on campus roads, loss of time and driver frustration. The Client would like to find a solution that can solve this problem, creating a better parking experience for all of their visitors. Each parking lot has been retrofitted with sensors to detect the entering and exiting of vehicles. This data is then aggregated and emitted via an internal API in preparation for a solution.

1.1.4 Scope

As there is an unlimited budget allocated to this project, the main project boundaries pertain to time. In order to create a minimum viable product within the given timelines, the following functionality has been deemed in and out of scope:

In scope

- The System will be designed as a web application.
- The System will store the minimum user information it needs to function correctly.
- The System will display the available capacity of each parking lot.
- The System will display the locations of each parking lot on a map.
- The System will contain a simple notification system for alerting end users of parking lot availability.

Out of scope

- The System will not show the specific location of each available parking spot in a parking lot.
- The System will not implement any functionality required for reserving parking spots.
- The System will not take monthly or yearly parking passes into account.
- The System is not required to implement a payment method for the purchase of parking, although this may be optionally added if time permits.

1.1.5 Stakeholders

The Senior Systems Analyst at the University of Faketoria and the university's students are the main focus of this project. The Senior Systems Analyst will oversee and maintain The System, and the students will be considered the end users.

People directly affected by the project:

1. *Project Sponsor:* The project sponsor decides whether the project moves forward based on how it's implemented, and by evaluating the current system in place.
2. *The Security Department of Faketoria:* The university's security department will manage and update the system. If The System is not easy to use or has other problems, they may request that a new system be built.
3. *Students and Staff of Faketoria:* The students and staff are the major group influencing the success of the project. If the system helps them save time when searching for an available parking spot, they will support the continuation of the project.

People indirectly affected by the project:

1. *Environmentalists*: Reducing the amount of cars circling campus roads searching for available parking spots will have a net positive impact on the environment.
2. *Pedestrians*: The System aims to reduce congestion and improve efficiency, which creates a safer environment for pedestrians attempting to cross campus roads.

1.1.6 Objectives

A basic implementation of The System must, at the very least, meet these basic objectives:

1. The System must always show the number of spots available for each individual parking lot at The Client's campus.
2. The System must be able to update parking availability for each parking lot in real-time.
3. The System will show the end users a map of The Client's campus, detailing the location of each parking lot.
4. The System will use the hardware sensors placed at the entrances and exits of each parking lot in order to determine the number of vehicles in each parking lot.
5. When the parking is below 95% capacity, The System will mark that parking lot as "Available" to indicate to end users that there are available parking spots in that parking lot.
6. When a parking lot is at or above 95% capacity, The System will mark that parking lot as "Full" to warn end users that finding a spot at that lot may be difficult.
7. The System must allow end users to save a parking lot as their preferred parking lot.
8. The System will have a notification feature that alerts end users when their desired parking lot becomes available.
9. The System will communicate with The Client via an existing API to mark all parking lots as "Full" during holidays and closures.

1.1.7 Glossary

Administrator	An employee of the University of Faketoria that has elevated permissions. They have the ability to manage parking lot statuses, logs, and create Operators to help manage The System.
Available / Availability	Indicates whether a parking spot is not currently occupied by a vehicle, or indicates that a parking lot has space for more vehicles to park in it.
End User	Any entity who uses the finished Parking Lot++ web application to find available parking spots. Most end users will consist of students at the University of Faketoria.
Operator	An employee of the University of Faketoria that has elevated permissions. They have the ability to manage parking lot statuses and logs.
Parking Lot	A collection of parking spots that is located at the University of Faketoria. There are multiple parking lots found across their campus, each holding multiple parking spots.
Parking Spot	A spot in a parking lot that can hold a single vehicle. There are multiple parking spots in a parking lot.
Preferred Parking Lot	The Parking Lot that an End User would most want to park in if it was available.
Real-time	Refers to interactions and responses that must be completed under 1 second, measured from the start of processing.
Registered User	An End User who has an account with The System that can be used to store preferences.
The Analysts	Refers to the analyst organization (Software++).
The Client	Refers to the client organization (The University of Faketoria).
The System	The Parking Lot++ web application that will be utilized by end users to find available parking spots in parking lots found at the University of Faketoria.
Unregistered User	An End User who does not have an account with The System.

1.2 Project Approach

1.2.1 Team Organization & Roles

Each member of The Analysts, along with their role and description, are listed below for communication purposes.

Member	Role	Role Description
Abdullahi Bashir	Systems Analyst	Works with clients to provide feedback on their project and analytical deliverables. In addition, he assesses and describes requirements for proposed systems through an iterative prototype design and evaluation process in collaboration with the customer.
Dan Melamed	Technical Assistant & Systems Analyst	Assists the project team with planning, organizing, resolving disputes, requirements gathering, and other technical support needs. In addition, he determines the needs of clients for system development.
Logan Raffkind	Unpaid Intern	Helps any team members that need an extra hand.
Vyom Shah	Analyst	Gathers information to develop the website that the client requested, and also customizes, optimizes, and analyzes the project's requirements.
Nilay Sondagar	Director++ of Technical Compliance	Supervises the technological infrastructure that Software++ administers for their clients, as well as manages their projects on a high level.
Franklin Yang	Analyst & Communications Manager	Assists the client in resolving their problems. He also provides professional feedback on the proposed product, and manages the project's operations. In addition, he handles communication between the client and Software++.

1.2.2 Work Breakdown Structure

A list of high-level tasks that will be carried out during this project are as follows:

- An agreement between The Client and The Analysts regarding the scope of The System will be created and signed by both parties.
- Concrete functional and nonfunctional requirements will be determined.
- A set of valid and reproducible use cases will be created.

- The interdependencies of The System will be outlined using domain models.
- A final set of UI mockups will be created using The Client's branding guidelines.

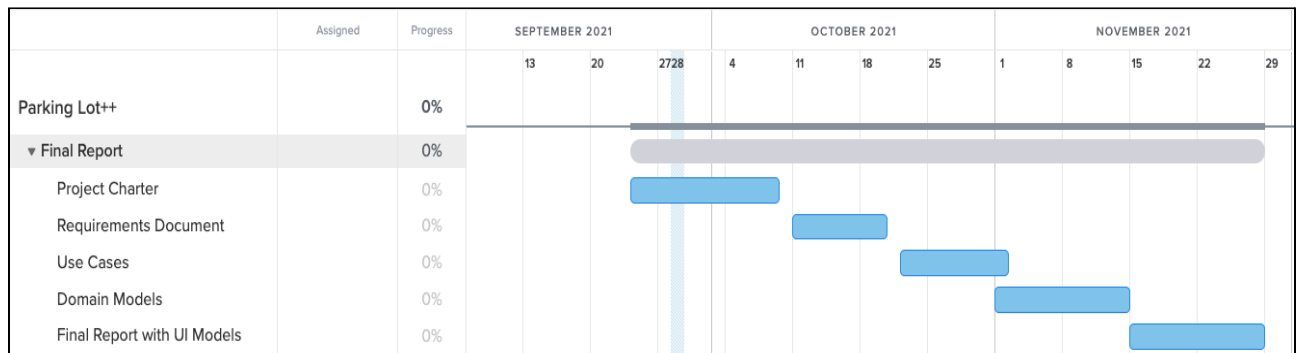
1.2.3 Milestones

The milestones for this project are as follows:

Milestone 1: Project Charter	October 7, 2021
Milestone 2: Requirements Report	October 19, 2021
Milestone 3: Requirements Report Review	October 21, 2021
Milestone 4: Use Cases	October 26, 2021
Milestone 5: Client Meeting 2	October 28, 2021
Milestone 6: Requirements and Use Cases Report	October 31, 2021
Milestone 7: Domain Models	November 2, 2021
Milestone 8: Requirements, Use Cases, Domain Models Report	November 2, 2021
Milestone 9: Client Meeting 3	November 15, 2021
Milestone 10: UI Models	November 23, 2021
Milestone 11: Final Report	November 28, 2021

1.2.4 Project Schedule

The deliverables for the requirements gathering phase of this project are listed in the gantt chart below. A more comprehensive chart will be appended to this document once development effort and timelines have been finalized.



1.2.5 Deliverables

There are several documents and models that are required by the client before the end of the requirements gathering phase. These include the following:

- Project Charter** October 7th, 2021
The document that outlines the scope, resources allocated, general timeline, and high-level objectives for the Parking Lot++ application. This will be a living

document that helps ensure that analyst and client organizations have a shared understanding of what function the Parking Lot++ application will serve.

2. Requirements Document

October 19th, 2021

This document will outline the functional and nonfunctional requirements that the Parking Lot++ application will need to meet in order to be considered an MVP. These requirements will also aid in the design of use cases, domain models, and UI models, as well as establish structure for the application testing phase.

3. Use Cases

October 31st, 2021

The *Requirements Document* will be extended to contain use cases that outline how end users of the Parking Lot++ application will interact with the system. These use cases will also aid in the application testing phase, and will describe basic flows for subsequent UI models.

4. Domain Models

November 14th, 2021

The *Requirements Document* will again be extended to include domain models, which define a structural and knowledge-based organization of application requirements. It will define key concepts and relationships among the entities of the application.

5. Final Report with UI Models

November 28th, 2021

This *Final Report* will be the final extension of the *Requirements Document*, and will include a finalized set of UI models that show all relevant application interfaces. These will be fully colorized and branded to match the client organization.

1.2.6 Risks

A list of risks categorized as *High / Medium / Low* probability and *High / Medium / Low* impact are listed in the table below. This list also includes ways to mitigate each risk.

No.	Risk Description	Probability (H/M/L)	Impact (H/M/L)	Planned Mitigation
1	Too many people use the website at the same time to check for available parking spots, causing a denial of service.	M	H	The System will run on a cloud server managed by a reputable company to reduce downtime, as well as implement resource scaling.
2	A hardware sensor fails to correctly count vehicles, resulting in invalid data input.	H	L	The API that aggregates data from the hardware sensors (maintained by The Client) can alert an administrator if the number of available spots exceeds the max capacity, or falls below 0.
3	End users do not understand how to use the website.	M	M	Create a beginner's guide or walkthrough tutorial.
4	A data breach or leak results in user data being exposed.	L	L	A minimal amount of user identifiable information (user email and password only) will be stored directly by The Software. All passwords will be hashed and encrypted when stored, mitigating the impact of a password leak.

1.3 Project Approval

1.3.1 Stakeholders Sign-off

By signing in the specified locations in *Section 3.1*, the client organization confirms that they have read through the entirety of the Project Charter and have agreed to the terms, scope, and timeline outlined in the aforementioned document.

Skylar Buck (Senior Systems Analyst)

Signature

Date

1.3.2 Team Members Sign-off

By signing in the specified locations in *Section 3.2*, the analyst organization confirms that they have read through the entirety of the Project Charter and will adhere to the terms, scope, and timeline outlined in the aforementioned document.

Nilay Sondagar (Director++ of Technical Compliance)

Signature

Date



October 31, 2021

2 Requirements

2.1 Functional Requirements

The requirements encompassing the core functionality of The System are described in the following sections, separated by the subsystems they apply to.

2.1.1 Map View Requirements

The following requirements describe the functionality required by the map view interface. This interface will be used by End Users to see availability of Parking Lots.

MV-1: An End User shall be able to view a map showing each Parking Lot's location.

MV-2: An End User shall be able to view the number of available Parking Spots in each Parking Lot unless the Parking Lot is marked as "CLOSED".

MV-3: An End User shall be able to view the total number of Parking Spots in each Parking Lot.

MV-4: If a Parking Lot's available capacity is strictly less than 5% of the total capacity, the Parking Lot shall be marked as "FULL".

MV-5: If a Parking Lot's available capacity is greater than or equal to 5% of the total capacity, the Parking Lot shall be marked as "AVAILABLE".

MV-6: Each Parking Lot shall display "CLOSED" when the University of Faketoria is closed.

MV-7: A Parking Lot shall display "CLOSED" when marked as "UNAVAILABLE" by an Administrator.

MV-8: A Registered User shall be able to select a Parking Lot as their Preferred Parking Lot.

MV-9: A Registered User shall be prompted to accept notification permissions when selecting a Preferred Parking Lot.

MV-10: A Registered User shall be prompted to accept notification permissions when updating their Preferred Parking Lot.

MV-11: A Registered User's Preferred Parking Lot shall be visibly labelled as their Preferred Parking Lot.

MV-12: If access to all Parking Lot information has been disabled, a "CURRENTLY UNAVAILABLE" message shall be shown to End Users.

MV-13: A Parking Lot's available capacity shall never be less than 0%.

MV-14: A Parking Lot's available capacity shall never exceed 100%.

2.1.2 Registration System Requirements

The following requirements describe the functionality required by the registration system. This system will allow Unregistered Users to create an account to select their Preferred Parking Lot, and become a Registered User.

RS-1: An Unregistered User shall be able to register for an account.

RS-2: An Unregistered User shall be prompted to enter an email when registering for an account.

RS-3: An Unregistered User shall be prompted to enter an account password when registering for an account.

RS-4: An account password shall be at least 8 characters otherwise an error will be shown.

RS-5: An account password shall contain at least one uppercase letter otherwise an error will be shown.

RS-6: An account password shall contain at least one lowercase letter otherwise an error will be shown.

RS-7: An account password shall contain at least one number otherwise an error will be shown.

RS-8: An email shall be used for only one account.

RS-9: If an Unregistered User enters an email already in use when registering for an account, an error shall be shown.

2.1.3 User Settings Requirements

The following requirements describe the functionality required by the user settings interface. This interface will allow Registered Users to manage their preferences and account settings.

US-1: A Registered User shall be able to delete their Preferred Parking Lot.

US-2: A Registered User shall be able to turn off notifications regarding the availability of their Preferred Parking Lot.

US-3: A Registered User shall be able to log in to their account.

US-4: A Registered User shall be able to log out of their account.

US-5: A Registered User shall be able to delete their account.

US-6: A Registered User shall be able to update their account password.

US-7: A Registered User shall be prompted to enter their current account password prior to updating their account password.

US-8: A Registered User shall confirm their new account password by re-entering the new account password when updating their account password.

US-9: The current account password entered by a Registered User when updating their account password shall be correct in order to update their account password.

US-10: The confirmed new account password entered by a Registered User when updating their account password shall match their new account password in order to update their account password.

US-11: If the confirmed new account password does not match the new account password when a Registered User is updating their account password, an error shall be shown.

US-12: If the current account password entered by a Registered User when updating their account password is incorrect, an error shall be shown.

2.1.4 Notification System Requirements

The following requirements describe the functionality required by the notification system. This system will be used to inform a Registered User about availability in their Preferred Parking Lot.

NS-1: Only Registered Users who have accepted notification permissions and have selected a Preferred Parking Lot shall receive notifications.

NS-2: If a Registered User's Preferred Parking Lot has availability and the Registered User is viewing the map, a notification shall be sent to the Registered User.

NS-3: Each notification shall be dismissable by the Registered User it was sent to.

NS-4: If a notification is dismissed by the Registered User and their Preferred Parking Lot is no longer available and the Registered User is not currently in their Preferred Parking Lot, another notification shall be sent to alert the Registered User that their Preferred Parking Lot is no longer available.

NS-5: If the Registered User enters their Preferred Parking Lot after receiving a notification, the notification shall be automatically dismissed.

NS-6: A Registered User shall not receive more than 5 notifications within a 60 minute window.

2.1.5 Administrator Portal Requirements

The following requirements describe the functionality required by the administrator portal. This portal will be used by an Administrator to create more Administrators and manage Parking Lots.

AP-1: Administrators shall be able to access the administrator portal.

AP-2: Operators shall be able to access the administrator portal.

AP-3: An Operator shall be prompted to change their account password the first time they login.

AP-4: An Administrator shall be able to create a new account with operator privileges.

AP-5: An Administrator shall be able to see all accounts with operator privileges.

AP-6: An Administrator shall be able to delete an account with operator privileges.

AP-7: An Administrator shall enter a username when creating a new account with operator privileges.

AP-8: An Administrator shall enter a default account password when creating a new account with administrator privileges.

AP-9: An Administrator shall be able to mark each Parking Lot as "UNAVAILABLE".

AP-10: An Operator shall be able to mark each Parking Lot as "UNAVAILABLE".

AP-11: An Administrator shall be able to mark each Parking Lot as "AVAILABLE".

AP-12: An Operator shall be able to mark each Parking Lot as "AVAILABLE".

AP-13: An Administrator shall be able to mark all Parking Lots as "UNAVAILABLE" at once.

AP-14: An Operator shall be able to mark all Parking Lots as "UNAVAILABLE" at once.

AP-15: An Administrator shall be able to mark all Parking Lots as "AVAILABLE" at once.

AP-16: An Operator shall be able to mark all Parking Lots as “AVAILABLE” at once.

AP-17: An Administrator shall be able to disable access to all Parking Lot information from End Users.

AP-18: An Administrator shall be able to enable access to all Parking Lot information from End Users.

AP-19: An Administrator shall be able to manually change the available capacity of each Parking Lot.

AP-20: An Operator shall be able to manually change the available capacity of each Parking Lot.

AP-21: An Administrator shall be able to manually change the total capacity of each Parking Lot.

AP-22: An Operator shall be able to manually change the total capacity of each Parking Lot.

AP-23: An Administrator shall be able to download system logs for each Parking Lot.

AP-24: An Administrator shall be able to log in to their account.

AP-25: An Administrator shall be able to log out of their account.

AP-26: An Operator shall be able to log in to their account.

AP-27: An Operator shall be able to log out of their account.

2.2 Nonfunctional Requirements

The requirements encompassing the nonfunctional aspects of The System are described in the following sections.

2.2.1 Performance & Scalability Requirements

These requirements pertain to the performance, reliability, and scalability of The System.

PS-1: Parking Lot availability shall be updated in real-time.

PS-2: A user account database backup shall be automatically performed once a week.

PS-3: An operator account database backup shall be automatically performed once a month.

PS-4: Each table in the database shall be able to store at least 10,000 entries.

PS-5: The System shall be able to handle at least 1000 concurrent End User connections.

PS-6: The admin portal shall be able to handle at least 100 concurrent Administrator and Operator connections.

PS-7: The System shall be fully operational during The Client's working hours, excluding scheduled maintenance.

PS-8: The System shall be able to respond to each request in under 3 seconds.

2.2.2 Security Requirements

These requirements pertain to the security and legal compliance of The System.

SL-1: No user identifiable information other than an End User's email shall be stored.

SL-2: Each account password stored in the database shall be hashed with a unique salt.

2.2.3 Usability Requirements

These requirements pertain to the overall usability and format of The System.

UR-1: The System shall be available as a website.

UR-2: The System shall be accessible from both desktop and mobile devices.

UR-3: The System shall support the 3 most recent versions of Chrome, Safari, Firefox, Edge, Chrome (mobile), and Safari (mobile).

UR-4: An End User shall spend less than 5 minutes finding an available parking spot, if one exists.

2.3 Use Cases

Use Case: UserLogin	Use Case: RegisteredUserNotificationOptIn
ID: 1	ID: 2
Brief Description: Allows a Registered User, Administrator, or Operator to log into their account	Brief Description: A Registered User chooses to opt in to receiving notifications
Actor(s): Registered User, Administrator, Operator	Actor(s): Registered User
Preconditions: <ol style="list-style-type: none"> 1. The user must already have an account. 	Preconditions: <ol style="list-style-type: none"> 1. The Registered User must be authenticated. 2. The Registered User has selected a Preferred Parking Lot.
Main Flow: <ol style="list-style-type: none"> 1. The user wants to log into their account. 2. While the user's credentials are invalid: <ol style="list-style-type: none"> 2.1. The user enters their user ID. 2.2. The user enters their account password. 2.3. The user's credentials are validated. 3. The user has successfully logged into their account. 4. If the user is an Administrator or Operator, then: <ol style="list-style-type: none"> 4.1. The user is redirected to the administrator portal. 5. Else: <ol style="list-style-type: none"> 5.1. The user is redirected to the map view. 	Main Flow: <ol style="list-style-type: none"> 1. The Registered User wants to opt into receiving notifications. 2. The Registered User sees a prompt for enabling notifications. 3. If the Registered User presses the "Enable Notifications" button then: <ol style="list-style-type: none"> 3.1. The Registered User receives a test notification. 3.2. The Registered User shall now receive notifications when their Preferred Parking Lot has availability. 4. The prompt is closed.
Postconditions: <ol style="list-style-type: none"> 1. The user is authenticated. 	Postconditions: <ol style="list-style-type: none"> 1. The Registered User is now registered for receiving notifications.

<div>Alternative Flow(s):</div> <div>3b) OperatorSetNewPassword</div>	<div>Alternative Flow(s):</div> <div>None</div>
---	---

<div>Alternative Flow:</div> <div>UserLogin:OperatorSetNewPassword</div>
<div>ID: 3.2</div>
<div>Brief Description:</div> <div>The Operator needs to create a new account password upon first login</div>
<div>Actor(s):</div> <div>Operator</div>
<div>Preconditions:</div> <div>1. The Operator must be authenticated.</div>
<div>Alternative Flow:</div> <div>1. The Operator must have logged in for the first time, using the default account password set by an Administrator.</div> <div>2. While the new account password is invalid:</div> <div>2.1. The Operator is prompted to enter a new account password.</div> <div>2.2. The Operator is prompted to confirm the new account password.</div> <div>2.3. The Operator's new account password is validated.</div> <div>3. The Operator is redirected to the administrator portal.</div>
<div>Postconditions:</div> <div>1. The Operator's new account password has been set.</div>

Use Case: RegisteredUserNotificationOptOut	Use Case: RegisteredUserReceivesNotification
ID: 3	ID: 4
Brief Description: A Registered User chooses to opt out of receiving notifications	Brief Description: A Registered User receives a notification about their Preferred Parking Lot being available
Actor(s): Registered User	Actor(s): Registered User
Preconditions: <ol style="list-style-type: none"> 1. The Registered User must be authenticated. 2. The Registered User must be registered to receive notifications. 	Preconditions: <ol style="list-style-type: none"> 1. The Registered User must be authenticated. 2. The Registered User must have selected their Preferred Parking Lot. 3. The Registered User must be registered to receive notifications. 4. The Registered User must be viewing the map view.
Main Flow: <ol style="list-style-type: none"> 1. The Registered User wants to opt out of receiving notifications. 2. The Registered User presses the "Disable Notifications" button. 3. The Registered User receives a confirmation message stating that notifications have been disabled. 4. The Registered User shall no longer receive notifications when their Preferred Parking Lot has availability. 	Main Flow: <ol style="list-style-type: none"> 1. The Registered User is looking for an available Parking Lot. 2. While The System checks for new notifications: <ol style="list-style-type: none"> 2.1. The System refreshes the page with javascript every 5 seconds. 2.2. The System sends a notification to the Registered User. 3. The Registered User reads the notification. 4. The Registered User dismisses the notification.
Postconditions: <ol style="list-style-type: none"> 1. The Registered User is now not registered for receiving notifications. 	Postconditions: <ol style="list-style-type: none"> 1. The Registered User was alerted to availability in their Preferred Parking Lot.

<p>Alternative Flow(s):</p> <p>None</p>	<p>Alternative Flow(s):</p> <p>None</p>
<p>Use Case: RegisteredUserSelectsPreferredParkingLot</p>	<p>Use Case: RegisteredUserDeletesPreferredParkingLot</p>
<p>ID: 5</p>	<p>ID: 6</p>
<p>Brief Description: A Registered User selects their Preferred Parking Lot.</p>	<p>Brief Description: A Registered User deletes their Preferred Parking Lot.</p>
<p>Actor(s): Registered User</p>	<p>Actor(s): Registered User</p>
<p>Preconditions:</p> <ol style="list-style-type: none"> 1. The Registered User must be authenticated. 	<p>Preconditions:</p> <ol style="list-style-type: none"> 1. The Registered User must be authenticated. 2. The Registered User must have selected their Preferred Parking Lot.
<p>Main Flow:</p> <ol style="list-style-type: none"> 1. The Registered User wants to select their Preferred Parking Lot. 2. The Registered User views the map view. 3. The Registered User selects a Parking Lot. 4. The Registered User clicks the "Favourite" button to select the Parking Lot as their Preferred Parking Lot. 5. If the Registered User already has a Preferred Parking Lot then: <ol style="list-style-type: none"> 5.1. A warning will be shown to the Registered User to indicate this will overwrite their existing Preferred Parking Lot. 5.2. If the Registered User selects "Confirm" then: <ol style="list-style-type: none"> 5.2.1. Their existing Preferred Parking Lot will be replaced with 	<p>Main Flow:</p> <ol style="list-style-type: none"> 1. The Registered User wants to delete their Preferred Parking Lot. 2. The Registered User views the map view. 3. The Registered User selects a Parking Lot. 4. The Registered User clicks the "Unfavourite" button to deselect their Preferred Parking Lot. 5. A warning will be shown to the Registered User to indicate this will delete their Preferred Parking Lot. 6. If the Registered User is registered for notifications then: <ol style="list-style-type: none"> 6.1. The warning will include a message that says they will no longer receive notifications about Parking Lot availability. 7. If the Registered User selects "Confirm" then: <ol style="list-style-type: none"> 7.1. Their existing Preferred Parking Lot will be replaced

<p>the new Preferred Parking Lot.</p> <p>5.3. The warning will be closed.</p> <p>6. Else:</p> <p>6.1. The Registered User will have selected a Preferred Parking Lot.</p> <p>7. include(RegisteredUserNotificationOptIn)</p>
<p>Postconditions:</p> <p>1. The Registered User now has a Preferred Parking Lot.</p>
<p>Alternative Flow(s):</p> <p>None</p>

<p>with the new Preferred Parking Lot.</p> <p>8. The warning will be closed.</p> <p>9. The Registered User will have deleted their Preferred Parking Lot.</p>
<p>Postconditions:</p> <p>1. The Registered User no longer has a Preferred Parking Lot.</p>
<p>Alternative Flow(s):</p> <p>None</p>

Use Case: EndUserViewsParkingLots
ID: 7
<p>Brief Description:</p> <p>End Users want to see the locations and capacity information of Parking Lots</p>
<p>Actor(s):</p> <p>End User</p>
<p>Preconditions:</p> <p>1. The End User is viewing the map view.</p>

Use Case: UnregisteredUserCreatesAccount
ID: 8
<p>Brief Description:</p> <p>An Unregistered User creates an account and becomes a Registered User</p>
<p>Actor(s):</p> <p>Unregistered User</p>
<p>Preconditions:</p> <p>1. The Unregistered User must not already have an account.</p>

<p style="text-align: center;">Main Flow:</p> <ol style="list-style-type: none"> 1. The End User wants to view the availability of Parking Lots. 2. The End User can pan the map with their fingers or mouse to see available Parking Lots. 3. Each Parking Lot should be shown geographically on the map. 4. Each Parking Lot marked as "AVAILABLE" should be highlighted in green. 5. Each Parking Lot marked as "UNAVAILABLE" or "CLOSED" should be highlighted in red. 6. Each Parking Lot should show its status, total capacity, and available capacity when clicked. 7. If the End User is a Registered User and has selected a Preferred Parking Lot then: <ol style="list-style-type: none"> 7.1. The Preferred Parking Lot should show a star icon next to it. 	<p style="text-align: center;">Main Flow:</p> <ol style="list-style-type: none"> 1. The Unregistered User wants to create an account. 2. While the Unregistered User's credentials are invalid: <ol style="list-style-type: none"> 2.1. The Unregistered User is prompted to enter a user ID. 2.2. The Unregistered User is prompted to enter a new account password. 2.3. The Unregistered User's user ID and account password are validated. 2.4. If the Unregistered User's user ID or account password are invalid then: <ol style="list-style-type: none"> 2.4.1. An error is shown to the Unregistered User. 3. The Unregistered User creates a new account.
<p style="text-align: center;">Postconditions:</p> <p>None</p>	<p style="text-align: center;">Postconditions:</p> <ol style="list-style-type: none"> 1. A new account has been created for the Unregistered User. 2. The Unregistered User is now a Registered User.
<p style="text-align: center;">Alternative Flow(s):</p> <p>None</p>	<p style="text-align: center;">Alternative Flow(s):</p> <p>None</p>

<p style="text-align: center;">Use Case: UserLogout</p>	<p style="text-align: center;">Use Case: AdminMarkParkingLotUnavailable</p>
<p style="text-align: center;">ID: 9</p>	<p style="text-align: center;">ID: 10</p>
<p style="text-align: center;">Brief Description:</p> <p>Allows a Registered User, Administrator, or Operator to logout of their account</p>	<p style="text-align: center;">Brief Description:</p> <p>An Administrator wants to set a Parking Lot as "UNAVAILABLE"</p>

Actor(s): Admin, Operator, Registered User	Actor(s): Administrator
Preconditions: 1. The Admin/Operator/Registered User must be logged in to their account	Preconditions: 1. The Administrator must be authenticated. 2. The Administrator is viewing the administrator portal.
Main Flow: 1. The user clicks the "Logout" button. 2. The user is logged out of their account. 3. The user is redirected to the login page.	Main Flow: 1. The Administrator wants to mark a Parking Lot as "UNAVAILABLE". 2. From the administrator portal, the Administrator selects a Parking Lot. 3. In the selected Parking Lot, the Administrator changes its status to "UNAVAILABLE". 4. The change in status is then updated for all End Users who see the selected Parking Lot as "UNAVAILABLE".
Postconditions: 1. The user is no longer authenticated.	Postconditions(s): 1. The selected Parking Lot is now labeled as UNAVAILABLE to all End Users.
Alternative Flow(s): None	Alternative Flow(s): None

Use Case: AdminMarkParkingLotAvailable	Use Case: AdminViewSystemLogs
ID: 11	ID: 12
Brief Description: An Administrator wants to set a Parking Lot as "AVAILABLE"	Brief Description: An Administrator wants to view the system logs
Actor(s): Administrator	Actor(s): Administrator
Preconditions: 1. Administrator must be authenticated.	Preconditions:

2. The Administrator is viewing the administrator portal.
Main Flow: <ol style="list-style-type: none"> 1. The Administrator wants to mark a Parking Lot as "AVAILABLE". 2. From the administrator portal, the Administrator selects a Parking Lot. 3. In the selected Parking Lot, the Administrator changes its status to "AVAILABLE". 4. The change in status is then updated for all End Users who see the selected Parking Lot as "AVAILABLE".
Postconditions: <ol style="list-style-type: none"> 1. The selected Parking Lot is now labeled as "AVAILABLE" to all End Users.
Alternative Flow(s): None

1. The Administrator must be authenticated.
Main Flow: <ol style="list-style-type: none"> 1. The Administrator navigates to the administrator portal. 2. From the administrator portal, the Administrator clicks "View System Logs". 3. The Administrator is now viewing the system logs.
Postconditions: None
Alternative Flow(s): None

Use Case: RegisteredUserDeletesAccount
ID: 13
Brief Description: A Registered User deletes their account and becomes an Unregistered User
Actor(s): Registered User
Preconditions: <ol style="list-style-type: none"> 1. The Registered User must be authenticated. 2. The Registered User must be on the account settings page.

Use Case: RegisteredUserChangesPassword
ID: 14
Brief Description: A Registered User changes their account password
Actor(s): Registered User
Preconditions: <ol style="list-style-type: none"> 1. The Registered User must be authenticated. 2. The Registered User must be on the account settings page.

<p style="text-align: center;">Main Flow:</p> <ol style="list-style-type: none"> 1. The Registered User wants to delete their account. 2. The Registered User clicks “Delete Account”. 3. A warning prompt is shown asking the Registered User to confirm that they want to permanently delete their account. 4. If the Registered User clicks “Confirm Deletion” then: <ol style="list-style-type: none"> 4.1. The Registered User’s account is deleted. 4.2. The Registered User is redirected to the map view. 5. Else: <ol style="list-style-type: none"> 5.1. The warning prompt is closed. 	<p style="text-align: center;">Main Flow:</p> <ol style="list-style-type: none"> 1. The Registered User wants to change their password. 2. The Registered User clicks the “Change Password” button. 3. The registered user enters their previous password. 4. While the Registered User’s new account password is invalid: <ol style="list-style-type: none"> 4.1. The Registered User is prompted to enter their new account password. 4.2. The Registered User is prompted to confirm their new account password. 4.3. The Registered User’s new account password is validated. 5. The Registered User has successfully changed their password.
<p style="text-align: center;">Postconditions:</p> <ol style="list-style-type: none"> 1. The Registered User is no longer authenticated. 2. The Registered User no longer has an account. 3. The Registered User is now an Unregistered User. 	<p style="text-align: center;">Postconditions:</p> <ol style="list-style-type: none"> 1. The Registered User’s new account password has been set.
<p style="text-align: center;">Alternative Flow(s):</p> <p>None</p>	<p style="text-align: center;">Alternative Flow(s):</p> <p>None</p>

<p style="text-align: center;">Use Case: AdminDisableParkingLotAccess</p>	<p style="text-align: center;">Use Case: AdminEnableParkingLotAccess</p>
<p style="text-align: center;">ID: 15</p>	<p style="text-align: center;">ID: 16</p>
<p style="text-align: center;">Brief Description:</p> <p>An Administrator wants to disable End User’s access to Parking Lot information</p>	<p style="text-align: center;">Brief Description:</p> <p>An Administrator wants to enable End User’s access to Parking Lot information</p>
<p style="text-align: center;">Actor(s): Administrator</p>	<p style="text-align: center;">Actor(s): Administrator</p>

<p>Preconditions:</p> <ol style="list-style-type: none"> 1. The Administrator must be authenticated. 2. The Administrator is viewing the administrator portal. 	<p>Preconditions:</p> <ol style="list-style-type: none"> 1. The Administrator must be authenticated. 2. The Administrator is viewing the administrator portal.
<p>Main Flow:</p> <ol style="list-style-type: none"> 1. The Administrator wants to disable End Users from viewing Parking Lot information. 2. From the administrator portal, the Administrator clicks "Disable Parking Lot Reporting". 3. The change in status is then updated for all End Users who see an info message on the map view that tells them that the application is currently unavailable. 	<p>Main Flow:</p> <ol style="list-style-type: none"> 1. The Administrator wants to enable End Users to view Parking Lot information. 2. From the administrator portal, the Administrator clicks "Enable Parking Lot Reporting". 3. The change in status is then updated for all End Users who are now able to see the map view.
<p>Postconditions:</p> <ol style="list-style-type: none"> 1. All Parking Lot information is hidden from End Users. 	<p>Postconditions(s):</p> <ol style="list-style-type: none"> 1. All Parking Lot information is available to End Users.
<p>Alternative Flow(s):</p> <p>None</p>	<p>Alternative Flow(s):</p> <p>None</p>

<p>Use Case: AdminCreateOperator</p>	<p>Use Case: AdminDeleteOperator</p>
<p>ID: 17</p>	<p>ID: 18</p>
<p>Brief Description: An Administrator creates a new Operator account</p>	<p>Brief Description: An Administrator deletes an Operator account</p>
<p>Actor(s): Administrator, Operator</p>	<p>Actor(s): Administrator, Operator</p>
<p>Preconditions:</p> <ol style="list-style-type: none"> 1. The Administrator must be authenticated. 	<p>Preconditions:</p> <ol style="list-style-type: none"> 1. The Administrator must be authenticated. 2. There must be at least one Operator account.

<p style="text-align: center;">Main Flow:</p> <ol style="list-style-type: none"> 1. The Administrator navigates to the administrator portal. 2. From the administrator portal, the Administrator selects the option to create a new Operator account. 3. The Administrator is prompted to enter an account user ID. 4. The Administrator is prompted to enter a default account password. 	<p style="text-align: center;">Main Flow:</p> <ol style="list-style-type: none"> 1. The Administrator navigates to the administrator portal. 2. From the administrator portal, the Administrator selects the option to manage Operator accounts. 3. The Administrator selects the Operator account to be deleted. 4. The Administrator clicks the “Delete” button next to the Operator account they would like to delete. <ol style="list-style-type: none"> 4.1. A warning will be shown to the Administrator to indicate this will permanently delete that Operator account. 4.2. If the Administrator selects “Confirm” then: <ol style="list-style-type: none"> 4.2.1. The Operator account will be deleted. 4.3. The warning will be closed.
<p style="text-align: center;">Postconditions:</p> <ol style="list-style-type: none"> 1. A new Operator account with the chosen user ID and default account password is created. 	<p style="text-align: center;">Postconditions:</p> <ol style="list-style-type: none"> 1. The Operator account is deleted. 2. The deleted Operator no longer has access to the administrator portal.
<p style="text-align: center;">Alternative Flow:</p> <p>None</p>	<p style="text-align: center;">Alternative Flow:</p> <p>None</p>

2.4 Use Case Diagram

The below diagram shows a visual representation of how actors interact with The System, and how each use case interacts with another.

