

**Software++ (Group 15)**

# **Parking Lot++**

## **Requirements Document**

*with Use Cases, Domain Models and UI Models*

**Version 4.0**

**November 28th, 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
2.1	6/11/2021	Nilay Sondagar	Added Context Diagram, DFD0, ERD, and Data Dictionary
2.2	7/11/2021	Abdullahi Bashir	Added DFD1 components
2.3	9/11/2021	Logan Raffkind	Added DFD2 and DFD3 components
2.4	9/11/2021	Dan Melamed	Added DFD5 components
2.5	10/11/2021	Franklin Yang	Added DFD4 components
2.6	11/11/2021	Vyom Shah	Added DFD6 and DFD7 components
3.0	14/11/2021	Everyone	Final Review
3.1	25/11/2021	Nilay Sondagar	Incorporated marking feedback from <i>Requirements Document 1.0</i>
3.2	27/11/2021	Everyone	Added storyboards for main user flows
3.3	28/11/2021	Nilay Sondagar	Added <i>Conclusion &amp; Recommendations</i> and appendices
4.0	28/11/2021	Nilay Sondagar	Final Review

# Table of Contents

<b>1 Introduction</b>	<b>3</b>
<b>1.1 Project Overview</b>	<b>3</b>
1.1.1 Executive Summary	3
1.1.2 Context	3
1.1.3 Need	4
1.1.4 Scope	4
1.1.5 Stakeholders	4
1.1.6 Objectives	5
1.1.7 Glossary	7
<b>1.2 Project Approach</b>	<b>8</b>
1.2.1 Team Organization & Roles	8
1.2.2 Work Breakdown Structure	9
1.2.3 Milestones	9
1.2.4 Project Schedule	9
1.2.5 Deliverables	10
1.2.6 Risks	11
<b>1.3 Project Approval</b>	<b>12</b>
1.3.1 Stakeholders Sign-off	12
1.3.2 Team Members Sign-off	12
<b>2 Requirements</b>	<b>13</b>
<b>2.1 Functional Requirements</b>	<b>13</b>
2.1.1 Map View Requirements	13
2.1.2 Registration System Requirements	13
2.1.3 User Settings Requirements	14
2.1.4 Notification System Requirements	14
2.1.5 Administrator Portal Requirements	15
<b>2.2 Nonfunctional Requirements</b>	<b>16</b>
2.2.1 Performance & Scalability Requirements	16
2.2.2 Security Requirements	17
2.2.3 Usability Requirements	17
<b>2.3 Use Cases</b>	<b>18</b>
<b>2.4 Use Case Diagram</b>	<b>29</b>
<b>3 Domain Models</b>	<b>30</b>
<b>3.1 Context Diagram and DFD-0</b>	<b>30</b>
<b>3.2 DFD-1 to DFD-7</b>	<b>32</b>
<b>3.3 Entity-Relationship Diagram</b>	<b>41</b>
<b>3.4 Data Dictionary</b>	<b>41</b>
<b>3.5 Data Dictionary Notes</b>	<b>43</b>
<b>4 UI Models</b>	<b>44</b>
<b>4.1 End User UI</b>	<b>44</b>
<b>4.2 Admin &amp; Operator Desktop UI</b>	<b>48</b>
<b>5 Conclusion &amp; Recommendations</b>	<b>52</b>
<b>Appendix A: General Documents</b>	<b>53</b>

# 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

<b>Administrator</b>	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.
<b>Available / Availability</b>	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.
<b>End User</b>	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.
<b>Operator</b>	An employee of the University of Faketoria that has elevated permissions. They have the ability to manage parking lot statuses and logs.
<b>Parking Lot</b>	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.
<b>Parking Spot</b>	A spot in a parking lot that can hold a single vehicle. There are multiple parking spots in a parking lot.
<b>Password Policy</b>	An account password must be at least 8 characters in length, and contain an uppercase letter, a lowercase letter, and a number in order to be valid.
<b>Preferred / Favourite Parking Lot</b>	The Parking Lot that an End User would most want to park in if it was available.
<b>Real-time</b>	Refers to interactions and responses that must be completed under 1 second, measured from the start of processing.
<b>Registered User</b>	An End User who has an account with The System that can be used to store preferences.
<b>Status</b>	One of “AVAILABLE”, “FULL”, or “CLOSED”, that indicates the general availability of a Parking Lot.
<b>The Analysts</b>	Refers to the analyst organization (Software++).
<b>The Client</b>	Refers to the client organization (The University of Faketoria).
<b>The System</b>	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.
<b>Unregistered User</b>	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.

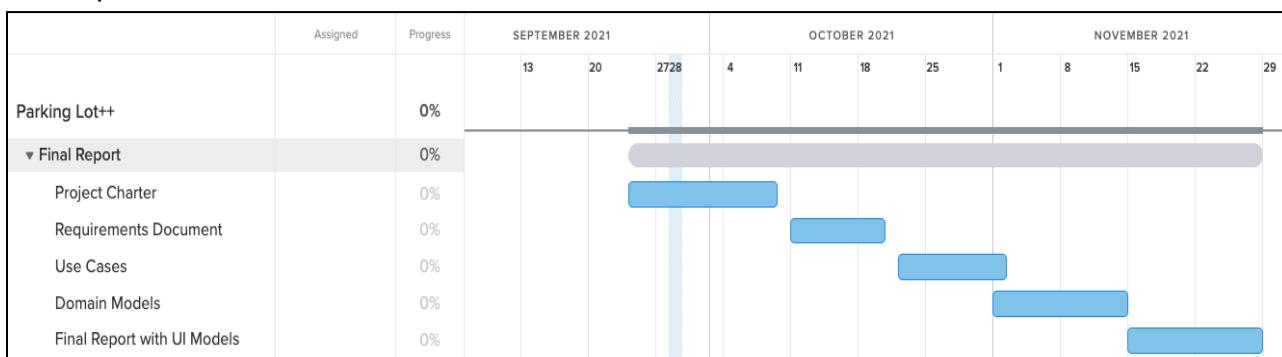
### 1.2.3 Milestones

The milestones for this project are as follows:

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

### 1. 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*



A handwritten signature in blue ink that reads "Nilay ++". The signature is fluid and cursive, with the last two letters of "Nilay" and the two plus signs being more stylized.



A handwritten date in blue ink that reads "Nov. 28/2021". The date is written in a clear, sans-serif font.

# 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:*** A Parking Lot shall display "CLOSED" when marked as "UNAVAILABLE" by an Administrator or Operator.

***MV-5:*** A Registered User shall be able to select a Parking Lot as their Preferred Parking Lot.

***MV-6:*** A Registered User shall be prompted to accept notification permissions when selecting a Preferred Parking Lot.

***MV-7:*** A Registered User shall be prompted to accept notification permissions when updating their Preferred Parking Lot.

***MV-8:*** A Registered User's Preferred Parking Lot shall be visibly labelled as their Preferred Parking Lot.

***MV-9:*** A Parking Lot's available capacity shall never be less than 0%.

***MV-10:*** 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 Unregistered User shall see an error when a password that does not meet the Password Policy is entered.

**RS-5:** An Unregistered User shall not be able to register with an email that is already associated with a Registered User.

### *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.

### *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:** A Registered User shall receive a notification if the Registered User's Preferred Parking Lot has availability.

**NS-3:** A Registered User shall be able to dismiss a notification sent to them

**NS-4:** A Registered User shall receive a notification when there is no longer availability in their Preferred Parking Lot.

**NS-5:** A Registered User shall not receive notifications about availability in their Preferred Parking Lot when they are currently located in their Preferred Parking Lot.

**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 Operators 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:** A Parking Lot’s availability shall be updated by the external hardware API in real-time.

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

**PS-3:** An operator account database backup shall be automatically performed by The System 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

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

<p><b>Alternative Flow(s):</b> 3b) OperatorSetNewPassword</p>	<p><b>Alternative Flow(s):</b> None</p>
<p><b>Use Case:</b> RegisteredUserNotificationOptOut</p> <p><b>ID:</b> 3</p> <p><b>Brief Description:</b> A Registered User chooses to opt out of receiving notifications</p> <p><b>Actor(s):</b> Registered User</p> <p><b>Preconditions:</b></p> <ol style="list-style-type: none"> <li>1. The Registered User must be authenticated.</li> <li>2. The Registered User must be registered to receive notifications.</li> </ol> <p><b>Main Flow:</b></p> <ol style="list-style-type: none"> <li>1. The Registered User wants to opt out of receiving notifications.</li> <li>2. The Registered User presses the "Disable Notifications" button.</li> <li>3. The Registered User receives a confirmation message stating that notifications have been disabled.</li> <li>4. The Registered User shall no longer receive notifications when their Preferred Parking Lot has availability.</li> </ol>	<p><b>Use Case:</b> RegisteredUserReceivesNotification</p> <p><b>ID:</b> 4</p> <p><b>Brief Description:</b> A Registered User receives a notification about their Preferred Parking Lot being available</p> <p><b>Actor(s):</b> Registered User</p> <p><b>Preconditions:</b></p> <ol style="list-style-type: none"> <li>1. The Registered User must be authenticated.</li> <li>2. The Registered User must have selected their Preferred Parking Lot.</li> <li>3. The Registered User must be registered to receive notifications.</li> <li>4. The Registered User must be viewing the map view.</li> </ol> <p><b>Main Flow:</b></p> <ol style="list-style-type: none"> <li>1. The Registered User is looking for an available Parking Lot.</li> <li>2. While The System checks for new notifications: <ol style="list-style-type: none"> <li>2.1. The System refreshes the page with javascript every 5 seconds.</li> <li>2.2. The System sends a notification to the Registered User.</li> </ol> </li> <li>3. The Registered User reads the notification.</li> <li>4. The Registered User dismisses the notification.</li> </ol>

<p><b>Postconditions:</b></p> <ol style="list-style-type: none"> <li>1. The Registered User is now not registered for receiving notifications.</li> </ol>	<p><b>Postconditions:</b></p> <ol style="list-style-type: none"> <li>1. The Registered User was alerted to availability in their Preferred Parking Lot.</li> </ol>
<p><b>Alternative Flow(s):</b></p> <p>None</p>	<p><b>Alternative Flow(s):</b></p> <p>None</p>

<p><b>Use Case:</b> RegisteredUserSelectsPreferredParkingLot</p> <p><b>ID:</b> 5</p> <p><b>Brief Description:</b> A Registered User selects their Preferred Parking Lot.</p> <p><b>Actor(s):</b> Registered User</p> <p><b>Preconditions:</b></p> <ol style="list-style-type: none"> <li>1. The Registered User must be authenticated.</li> </ol> <p><b>Main Flow:</b></p> <ol style="list-style-type: none"> <li>1. The Registered User wants to select their Preferred Parking Lot.</li> <li>2. The Registered User views the map view.</li> <li>3. The Registered User selects a Parking Lot.</li> <li>4. The Registered User clicks the "Favourite" button to select the Parking Lot as their Preferred Parking Lot.</li> <li>5. If the Registered User already has a Preferred Parking Lot then:           <ol style="list-style-type: none"> <li>5.1. A warning will be shown to the Registered User to indicate this will overwrite their existing Preferred Parking Lot.</li> </ol> </li> </ol>	<p><b>Use Case:</b> RegisteredUserDeletesPreferredParkingLot</p> <p><b>ID:</b> 6</p> <p><b>Brief Description:</b> A Registered User deletes their Preferred Parking Lot.</p> <p><b>Actor(s):</b> Registered User</p> <p><b>Preconditions:</b></p> <ol style="list-style-type: none"> <li>1. The Registered User must be authenticated.</li> <li>2. The Registered User must have selected their Preferred Parking Lot.</li> </ol> <p><b>Main Flow:</b></p> <ol style="list-style-type: none"> <li>1. The Registered User wants to delete their Preferred Parking Lot.</li> <li>2. The Registered User views the map view.</li> <li>3. The Registered User selects a Parking Lot.</li> <li>4. The Registered User clicks the "Unfavourite" button to deselect their Preferred Parking Lot.</li> <li>5. A warning will be shown to the Registered User to indicate this will delete their Preferred Parking Lot.</li> <li>6. If the Registered User is registered for notifications then:           <ol style="list-style-type: none"> <li>6.1. The warning will include a message that says they will</li> </ol> </li> </ol>
---	---

<p>5.2. If the Registered User selects “Confirm” then:</p> <p>5.2.1. Their existing Preferred Parking Lot will be replaced with 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>no longer receive notifications about Parking Lot availability.</p> <p>7. If the Registered User selects “Confirm” then:</p> <p>7.1. Their existing Preferred Parking Lot will be replaced 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><b>Postconditions:</b></p> <p>1. The Registered User now has a Preferred Parking Lot.</p> <p><b>Alternative Flow(s):</b></p> <p>None</p>	<p><b>Postconditions:</b></p> <p>1. The Registered User no longer has a Preferred Parking Lot.</p> <p><b>Alternative Flow(s):</b></p> <p>None</p>

<p><b>Use Case:</b> EndUserViewsParkingLots</p>	<p><b>Use Case:</b> UnregisteredUserCreatesAccount</p>
<p><b>ID:</b> 7</p>	<p><b>ID:</b> 8</p>
<p><b>Brief Description:</b> End Users want to see the locations and capacity information of Parking Lots</p>	<p><b>Brief Description:</b> An Unregistered User creates an account and becomes a Registered User</p>
<p><b>Actor(s):</b> End User</p>	<p><b>Actor(s):</b> Unregistered User</p>
<p><b>Preconditions:</b></p> <p>1. The End User is viewing the map view.</p>	<p><b>Preconditions:</b></p> <p>1. The Unregistered User must not already have an account.</p>

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

<p><b>Postconditions:</b></p> <p>None</p>	<p><b>Postconditions:</b></p> <ol style="list-style-type: none"> <li>1. A new account has been created for the Unregistered User.</li> <li>2. The Unregistered User is now a Registered User.</li> </ol>
<p><b>Alternative Flow(s):</b></p> <p>None</p>	<p><b>Alternative Flow(s):</b></p> <p>None</p>

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

<b>Actor(s):</b> Admin, Operator, Registered User	<b>Actor(s):</b> Administrator
<b>Preconditions:</b> 1. The Admin/Operator/Registered User must be logged in to their account	<b>Preconditions:</b> 1. The Administrator must be authenticated. 2. The Administrator is viewing the administrator portal.
<b>Main Flow:</b> 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.	<b>Main Flow:</b> 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”.
<b>Postconditions:</b> 1. The user is no longer authenticated.	<b>Postconditions(s):</b> 1. The selected Parking Lot is now labeled as UNAVAILABLE to all End Users.
<b>Alternative Flow(s):</b> None	<b>Alternative Flow(s):</b> None

<b>Use Case:</b> AdminMarkParkingLotAvailable	<b>Use Case:</b> AdminViewSystemLogs
<b>ID:</b> 11	<b>ID:</b> 12
<b>Brief Description:</b> An Administrator wants to set a Parking Lot as “AVAILABLE”	<b>Brief Description:</b> An Administrator wants to view the system logs
<b>Actor(s):</b> Administrator	<b>Actor(s):</b> Administrator

<p><b>Preconditions:</b></p> <ol style="list-style-type: none"> <li>1. Administrator must be authenticated.</li> <li>2. The Administrator is viewing the administrator portal.</li> </ol>	<p><b>Preconditions:</b></p> <ol style="list-style-type: none"> <li>1. The Administrator must be authenticated.</li> </ol>
<p><b>Main Flow:</b></p> <ol style="list-style-type: none"> <li>1. The Administrator wants to mark a Parking Lot as “AVAILABLE”.</li> <li>2. From the administrator portal, the Administrator selects a Parking Lot.</li> <li>3. In the selected Parking Lot, the Administrator changes its status to “AVAILABLE”.</li> <li>4. The change in status is then updated for all End Users who see the selected Parking Lot as “AVAILABLE”.</li> </ol>	<p><b>Main Flow:</b></p> <ol style="list-style-type: none"> <li>1. The Administrator navigates to the administrator portal.</li> <li>2. From the administrator portal, the Administrator selects a Parking Lot and “View System Logs”.</li> <li>3. The Administrator is now viewing the system logs of a specific parking lot.</li> </ol>
<p><b>Postconditions:</b></p> <ol style="list-style-type: none"> <li>1. The selected Parking Lot is now labeled as “AVAILABLE” to all End Users.</li> </ol>	<p><b>Postconditions:</b></p> <p>None</p>
<p><b>Alternative Flow(s):</b></p> <p>None</p>	<p><b>Alternative Flow(s):</b></p> <p>None</p>

<p><b>Use Case:</b> RegisteredUserDeletesAccount</p>	<p><b>Use Case:</b> RegisteredUserChangesPassword</p>
<p><b>ID:</b> 13</p>	<p><b>ID:</b> 14</p>
<p><b>Brief Description:</b> A Registered User deletes their account and becomes an Unregistered User</p>	<p><b>Brief Description:</b> A Registered User changes their account password</p>
<p><b>Actor(s):</b> Registered User</p>	<p><b>Actor(s):</b> Registered User</p>
<p><b>Preconditions:</b></p> <ol style="list-style-type: none"> <li>1. The Registered User must be authenticated.</li> <li>2. The Registered User must be on the account settings page.</li> </ol>	<p><b>Preconditions:</b></p> <ol style="list-style-type: none"> <li>1. The Registered User must be authenticated.</li> <li>2. The Registered User must be on the account settings page.</li> </ol>

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

<p><b>Use Case:</b> AdminDisableParkingLotAccess</p>	<p><b>Use Case:</b> AdminEnableParkingLotAccess</p>
<p><b>ID:</b> 15</p>	<p><b>ID:</b> 16</p>
<p><b>Brief Description:</b> An Administrator wants to disable End User’s access to Parking Lot information</p>	<p><b>Brief Description:</b> An Administrator wants to enable End User’s access to Parking Lot information</p>
<p><b>Actor(s):</b> Administrator</p>	<p><b>Actor(s):</b> Administrator</p>

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

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

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

<p><b>Alternative Flow:</b>  <b>UserLogin:OperatorSetNewPassword</b></p>
<p><b>ID:</b> 3.2</p>
<p><b>Brief Description:</b>  The Operator needs to create a new account password upon first login</p>
<p><b>Actor(s):</b>  Operator</p>
<p><b>Preconditions:</b></p> <ol style="list-style-type: none"> <li>1. The Operator must be authenticated.</li> </ol>
<p><b>Alternative Flow:</b></p> <ol style="list-style-type: none"> <li>1. The Operator must have logged in for the first time, using the default account password set by an Administrator.</li> <li>2. While the new account password is invalid: <ol style="list-style-type: none"> <li>2.1. The Operator is prompted to enter a new account password.</li> <li>2.2. The Operator is prompted to confirm the new account password.</li> <li>2.3. The Operator's new account password is validated.</li> </ol> </li> <li>3. The Operator is redirected to the administrator portal.</li> </ol>
<p><b>Postconditions:</b></p> <ol style="list-style-type: none"> <li>1. The Operator's new account password has been set.</li> </ol>

## 2.4 Use Case Diagram

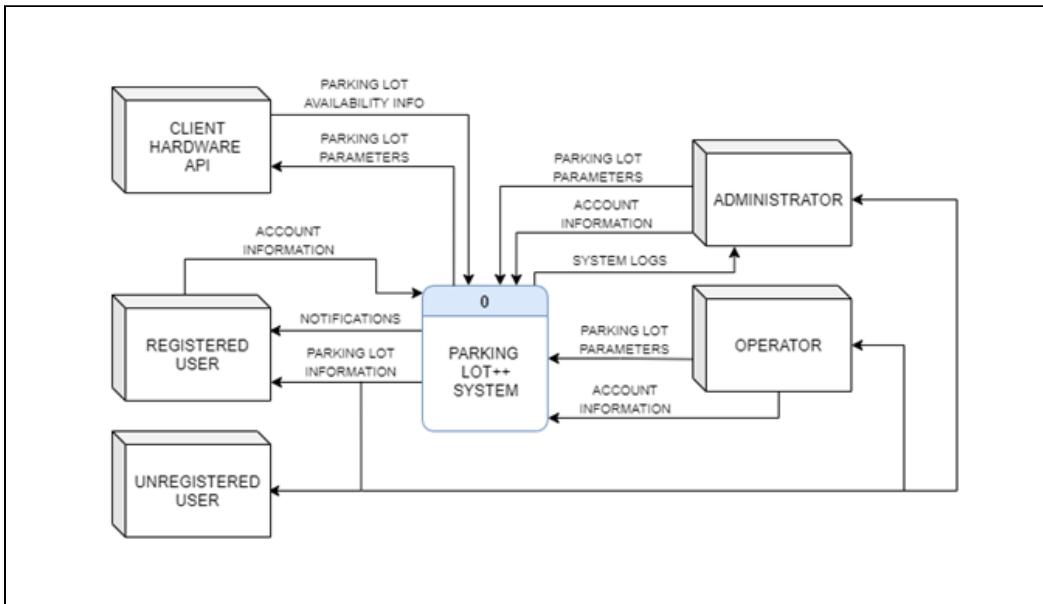
The below diagram shows a visual representation of how actors interact with The System, and how each use case interacts with another.



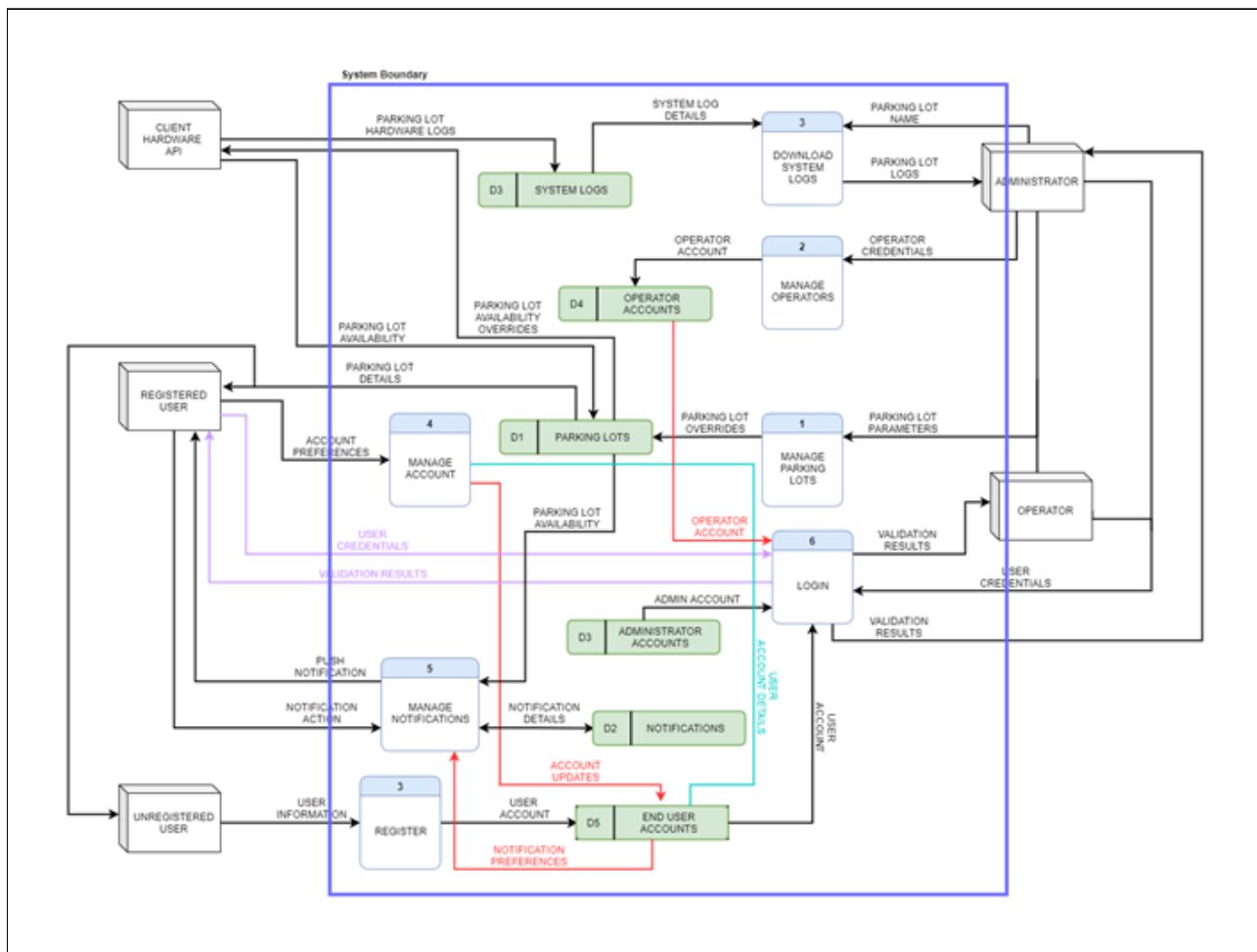
# 3 Domain Models

## 3.1 Context Diagram and DFD-0

**Context Diagram:**

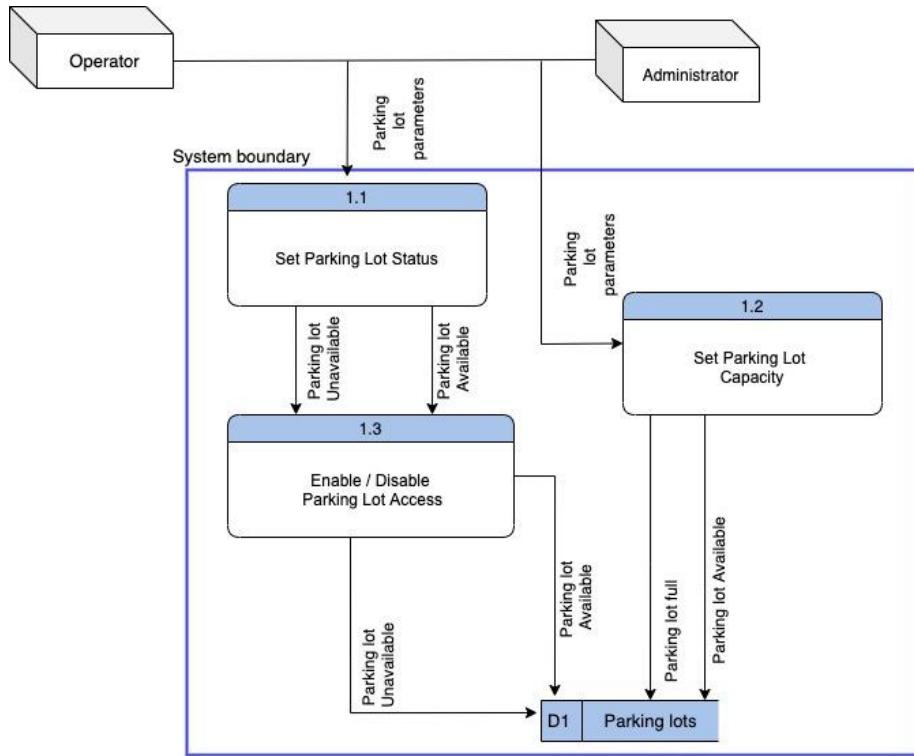


## DFD0 - Parking Lot++

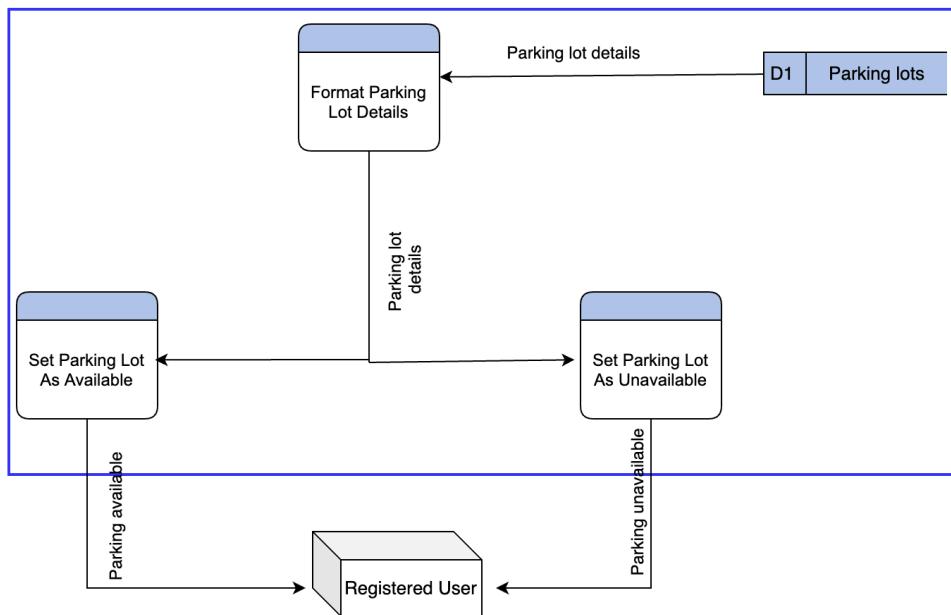


## 3.2 DFD-1 to DFD-7

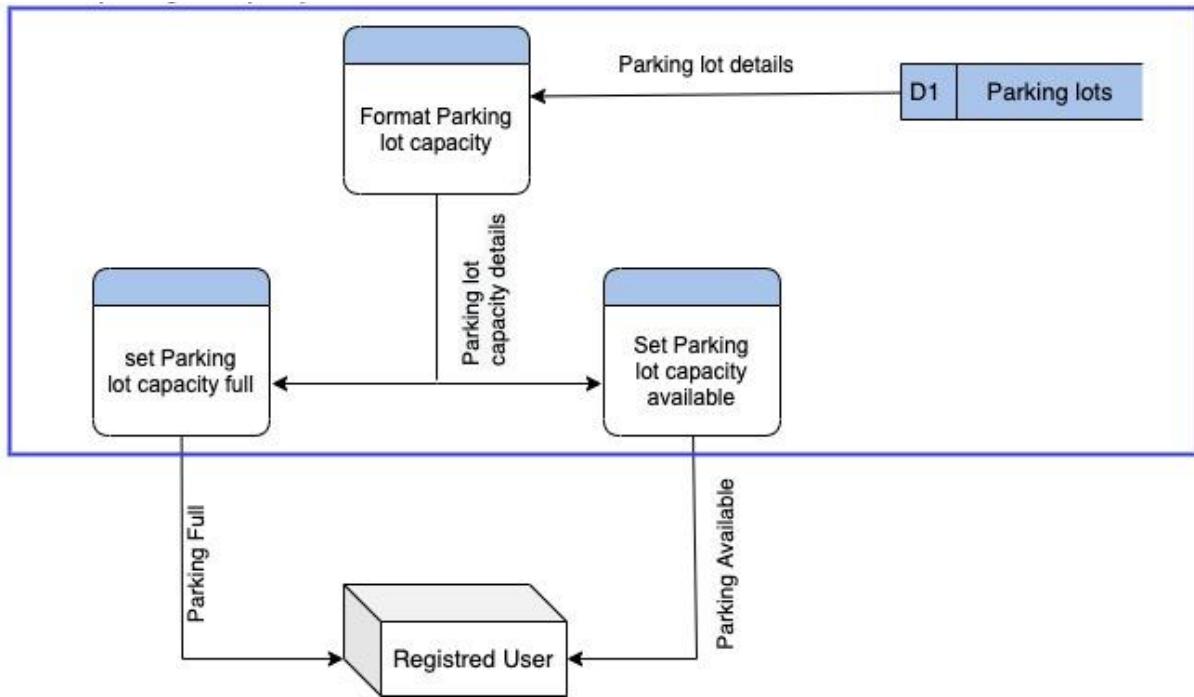
### DFD1 - Manage Parking Lots



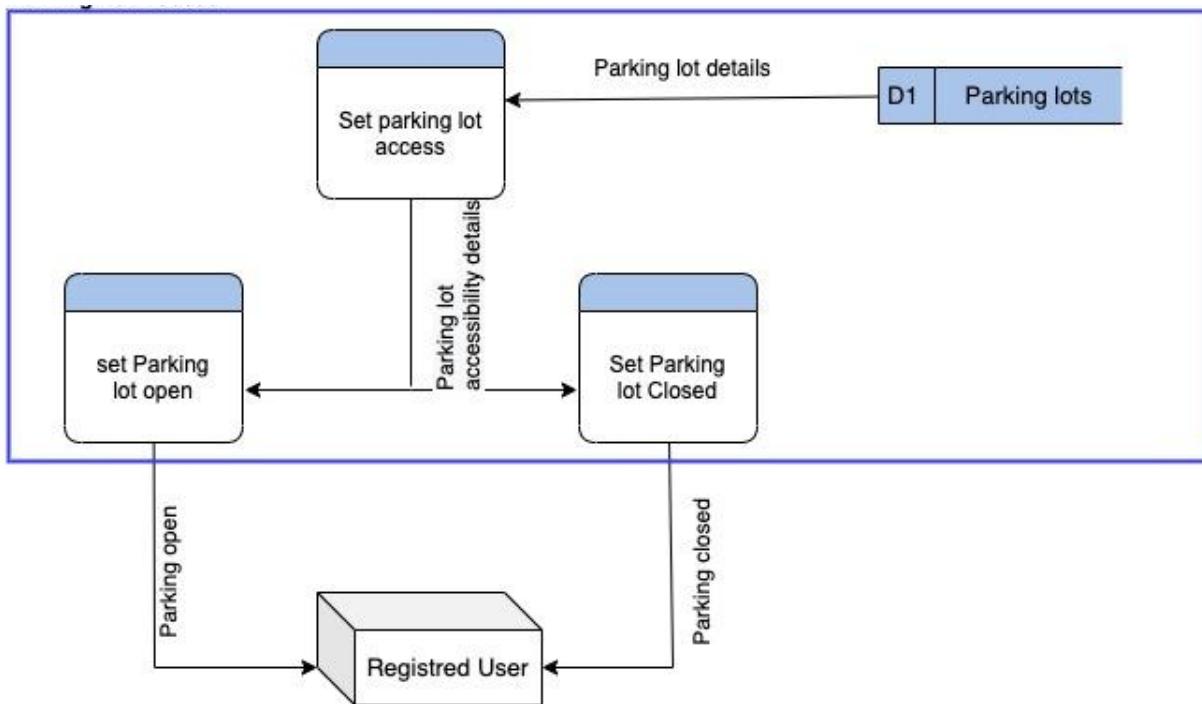
### DFD1.1 - Set Parking Lot Capacity



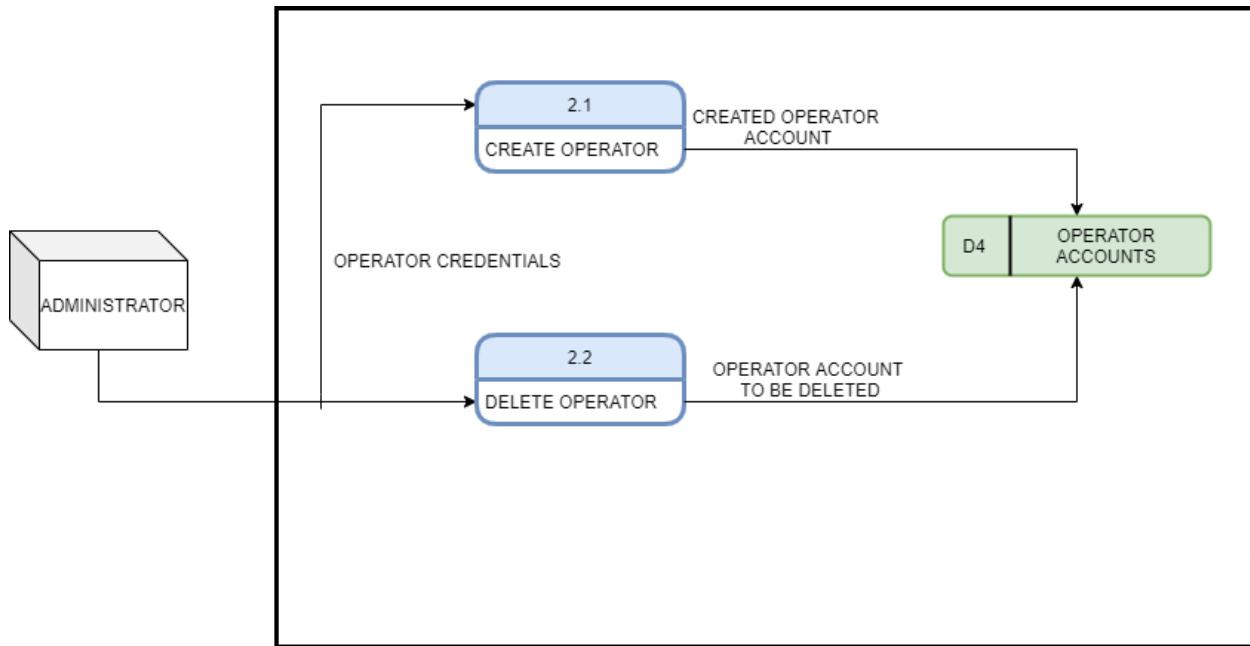
### DFD1.2 - Set Parking Lot Capacity



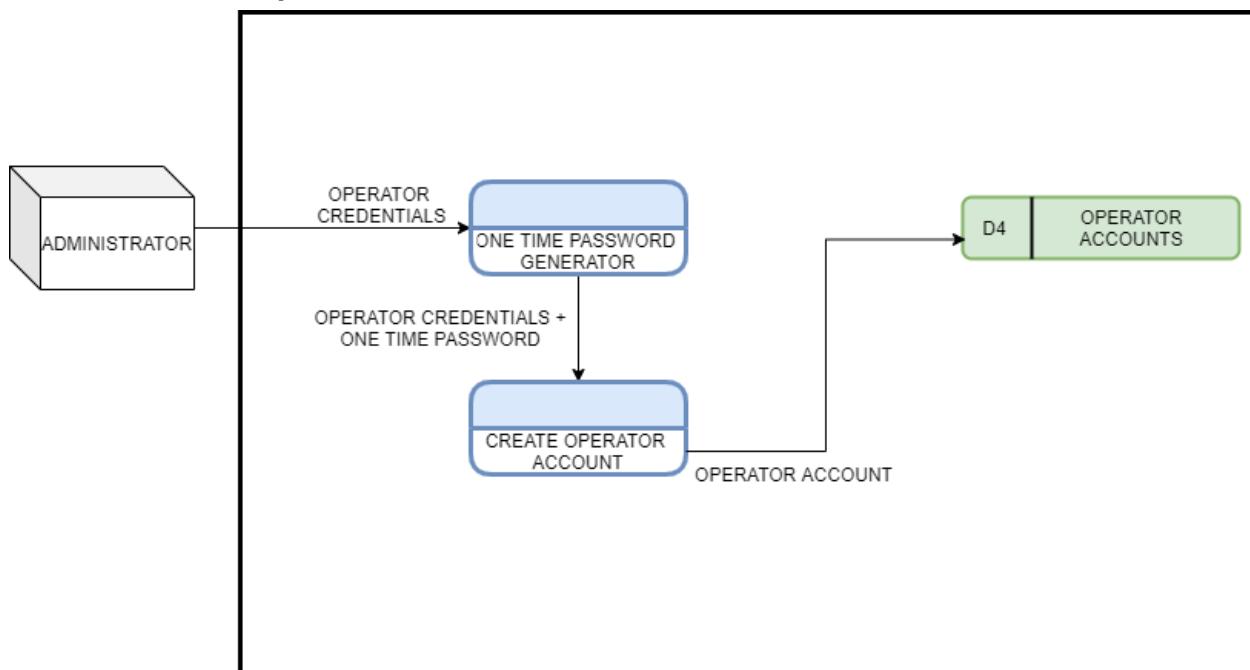
### DFD1.3 - Enable / Disable Parking Lot Access



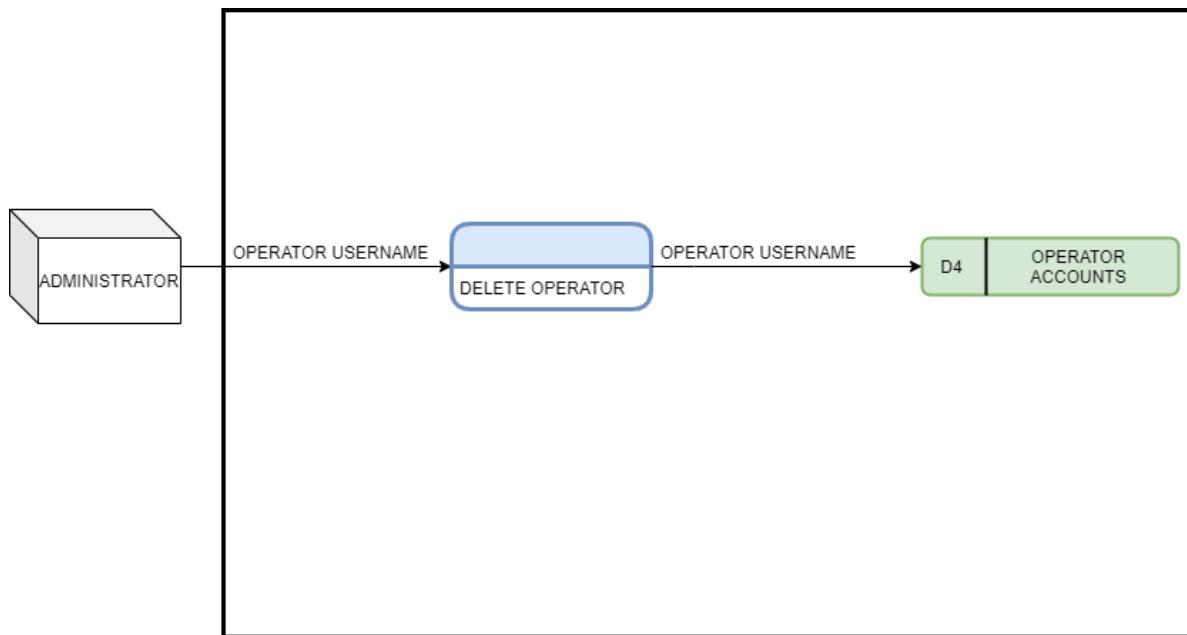
## DFD2 - Manage Operators



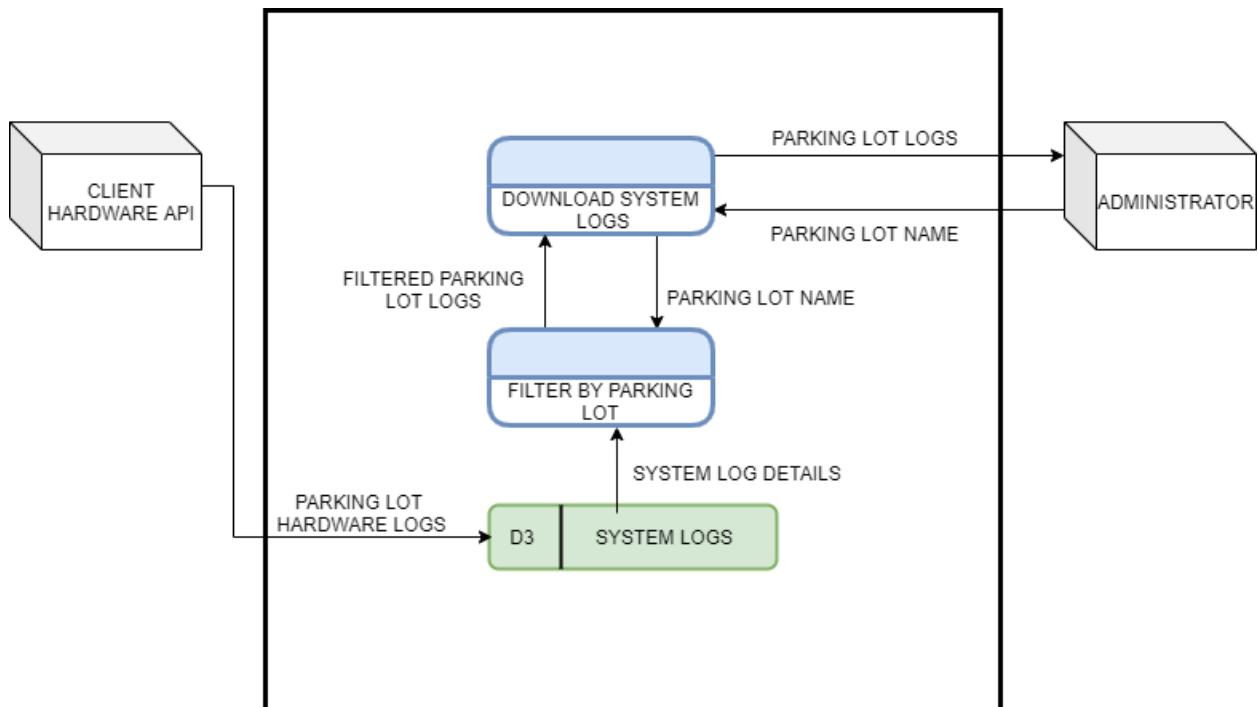
## DFD2.1 - Create Operator



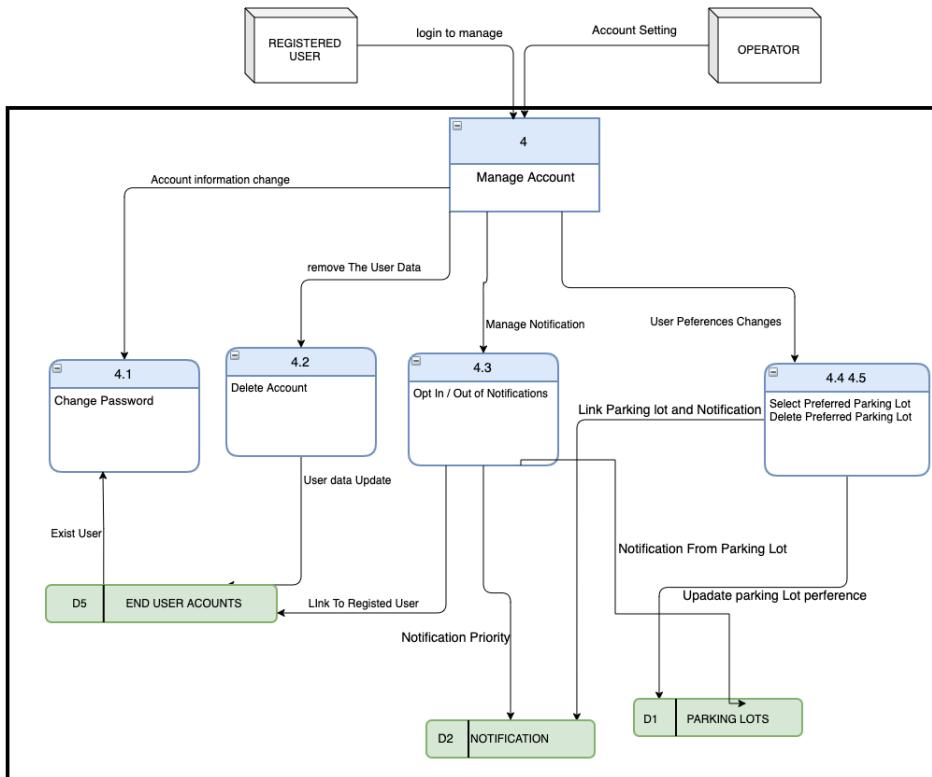
## DFD2.2 - Delete Operator



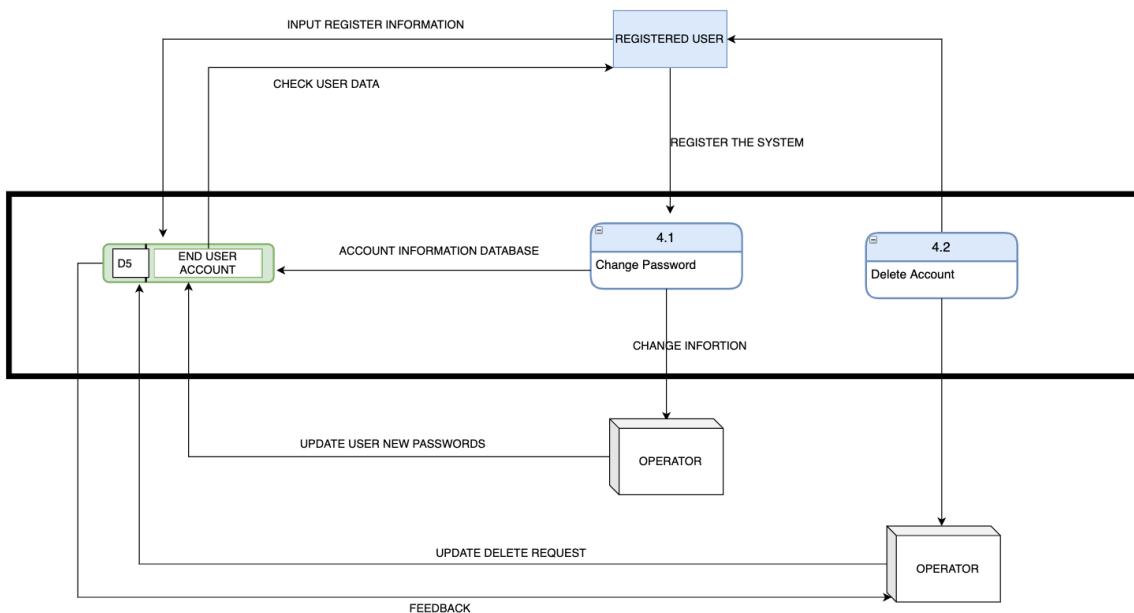
## DFD3 - Download System Logs



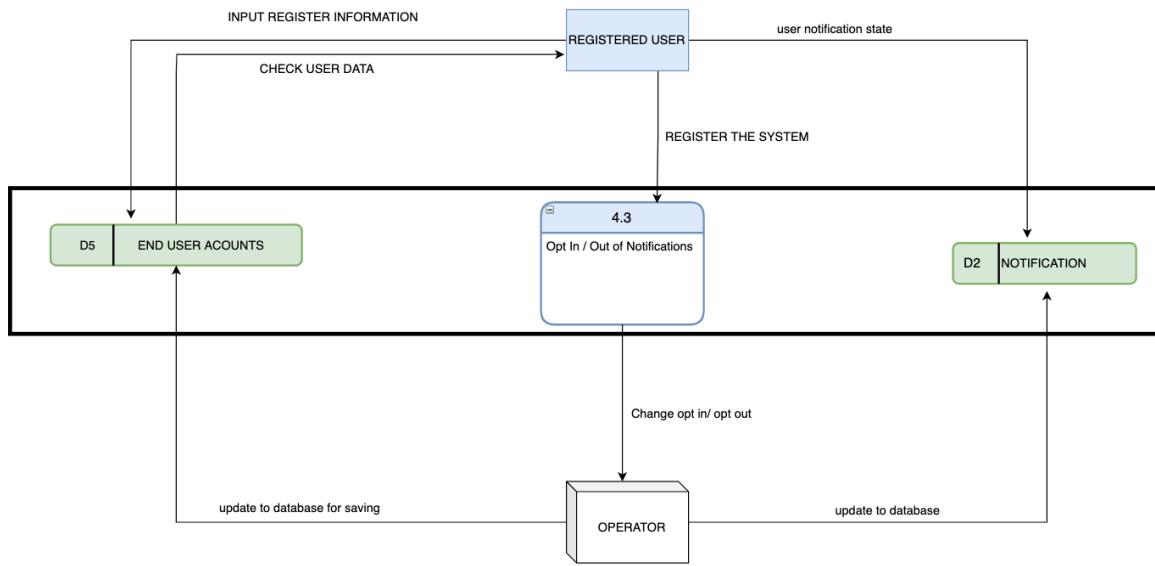
## DFD4 - Manage Account



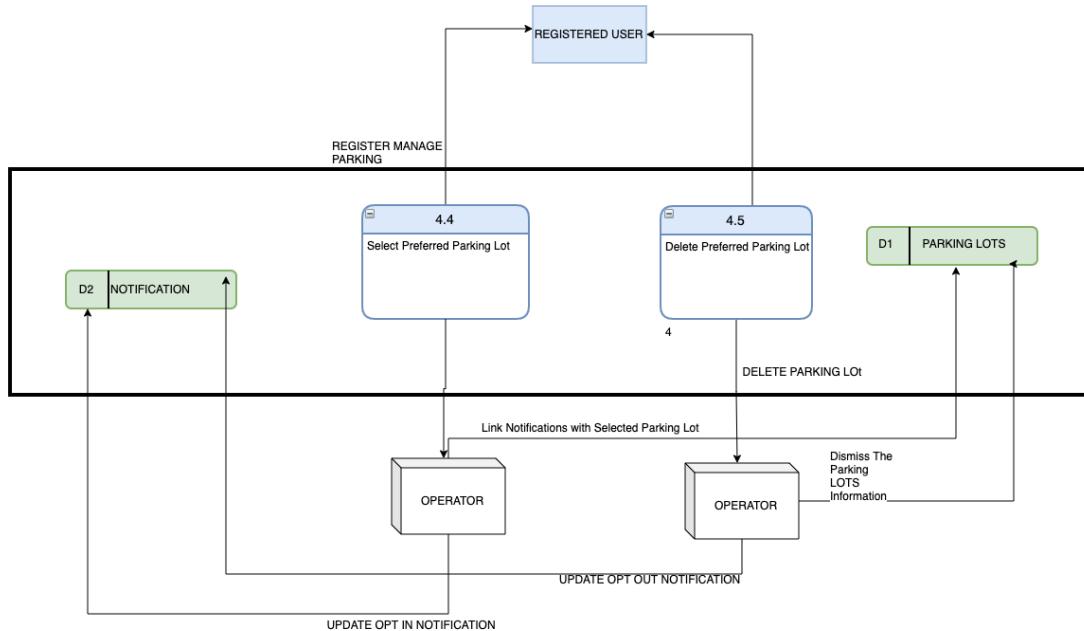
## DFD4.1 - Edit Account Details



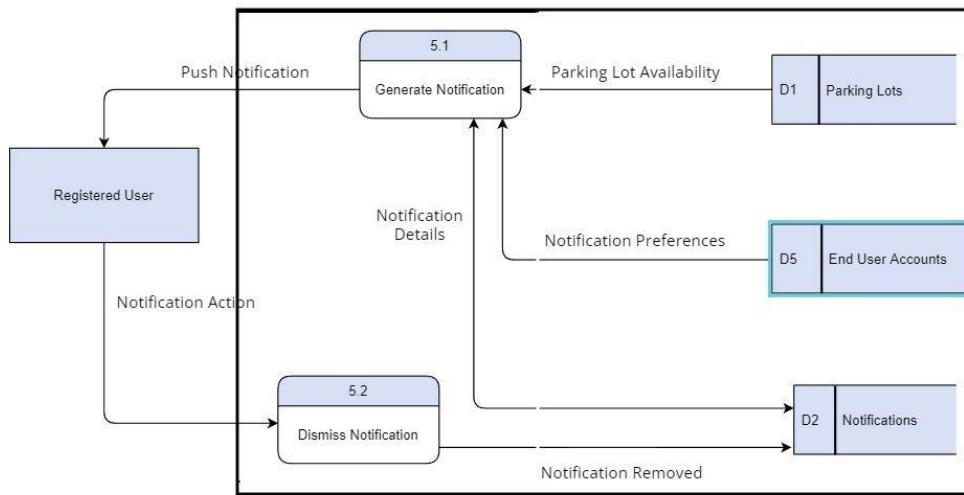
## DFD4.2 - Manage Notifications



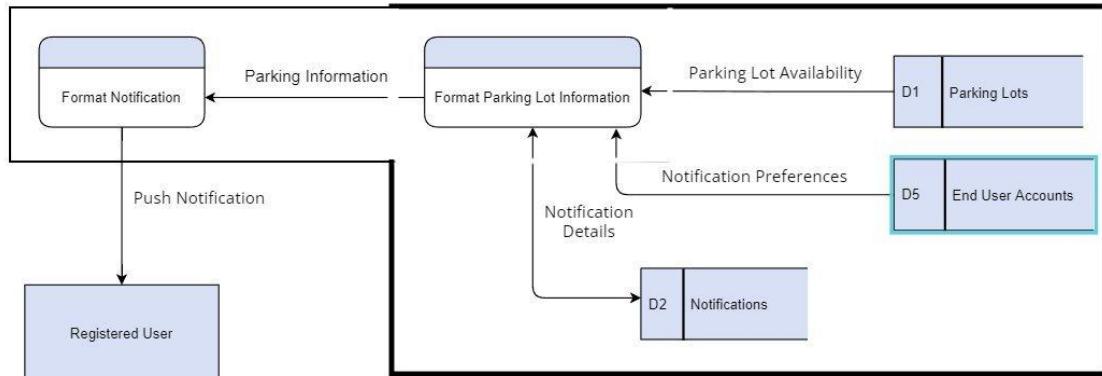
## DFD4.3 - Manage Preferred Parking Lot



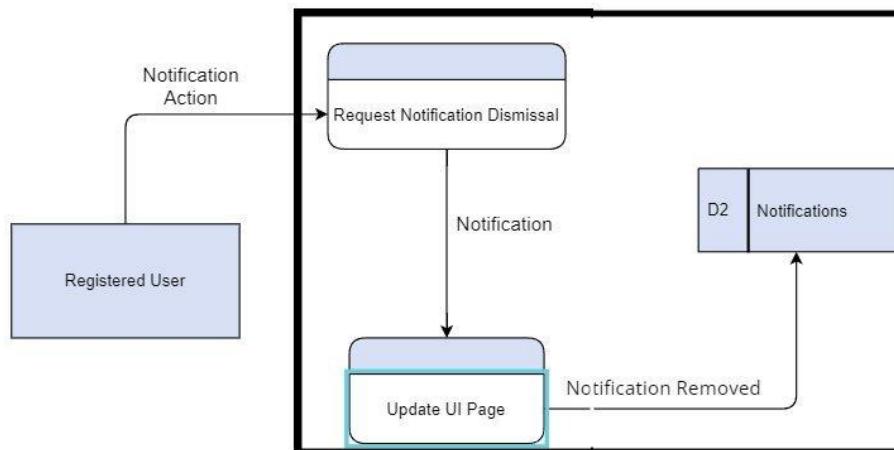
## DFD5 - Manage Notifications



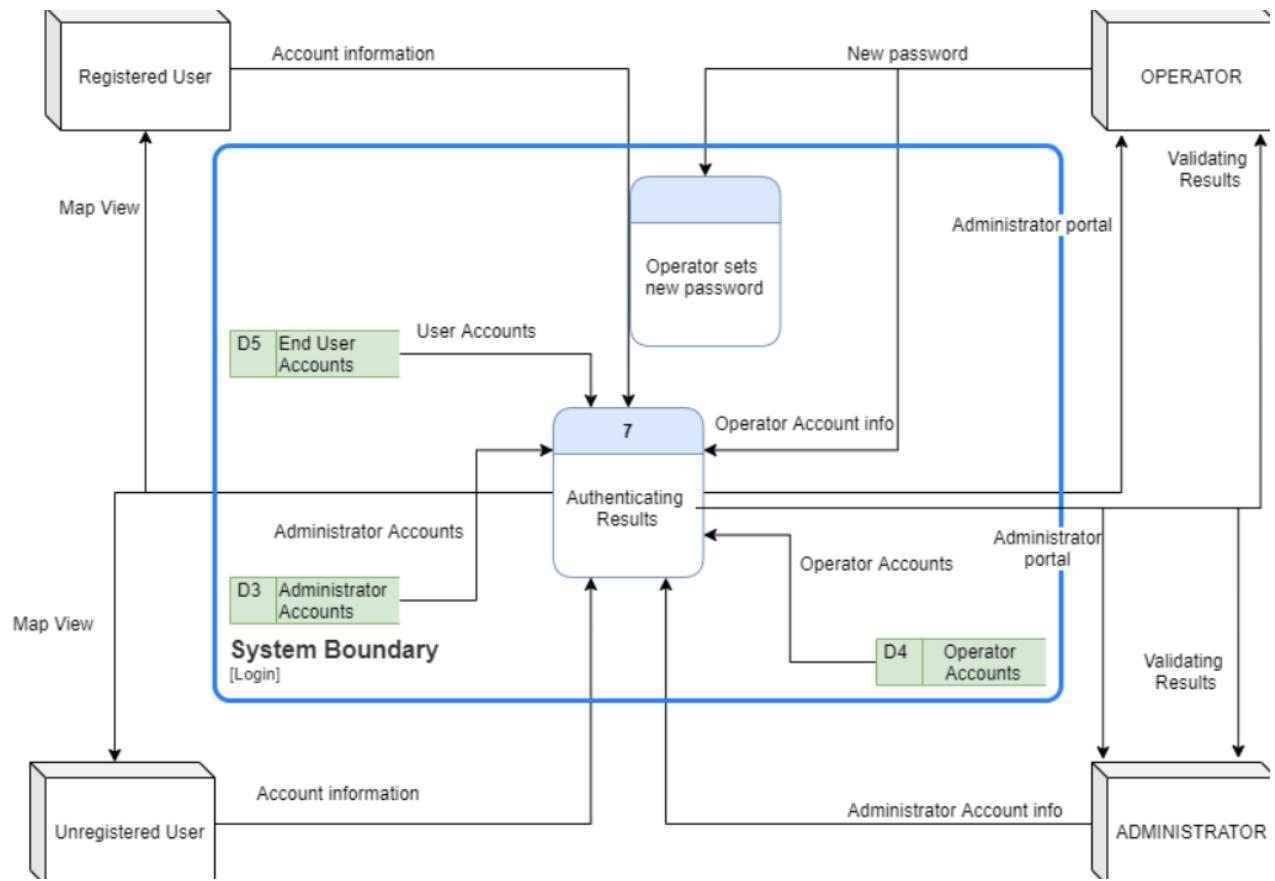
## DFD5.1 - Generate Notification



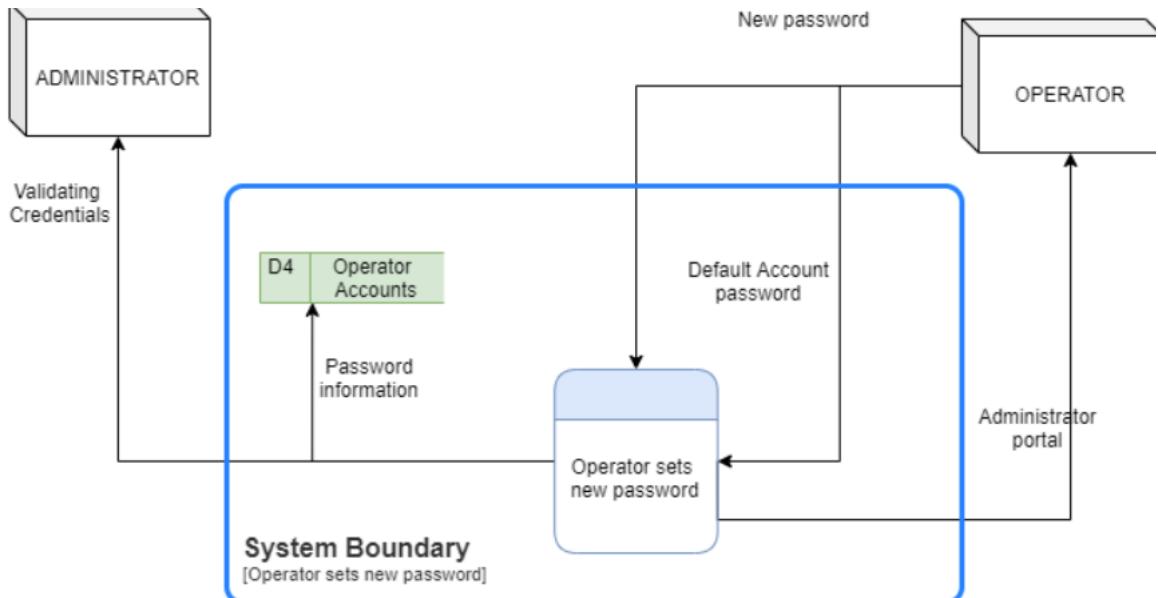
## DFD5.2 - Dismiss Notification



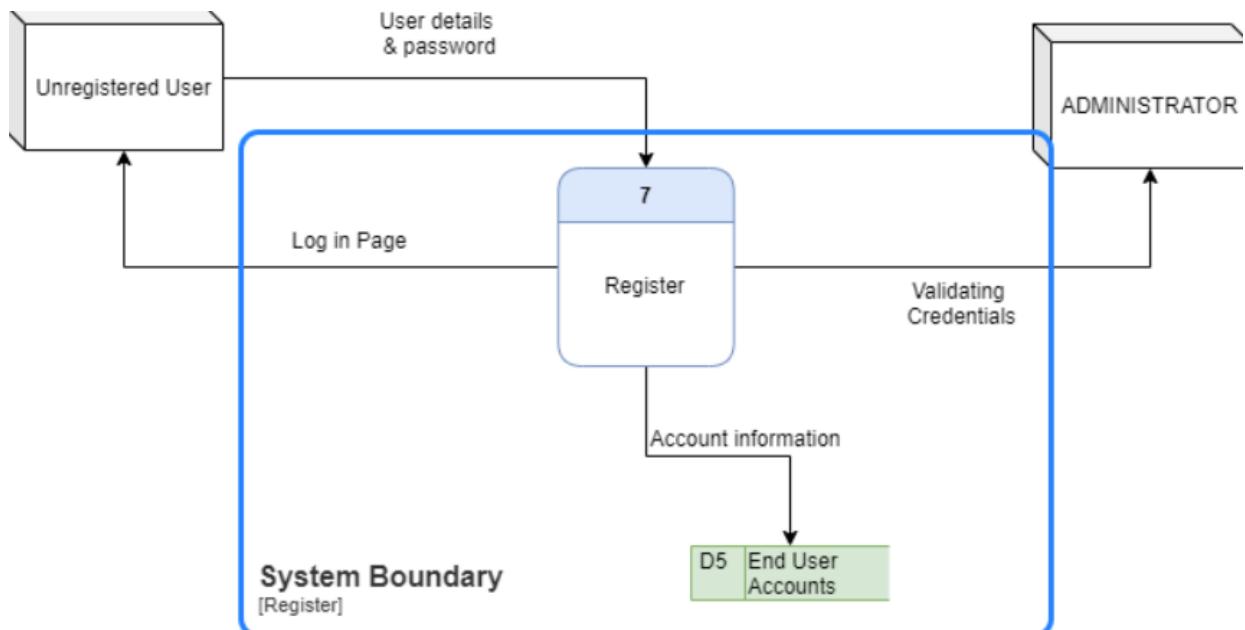
## DFD6 - Login System



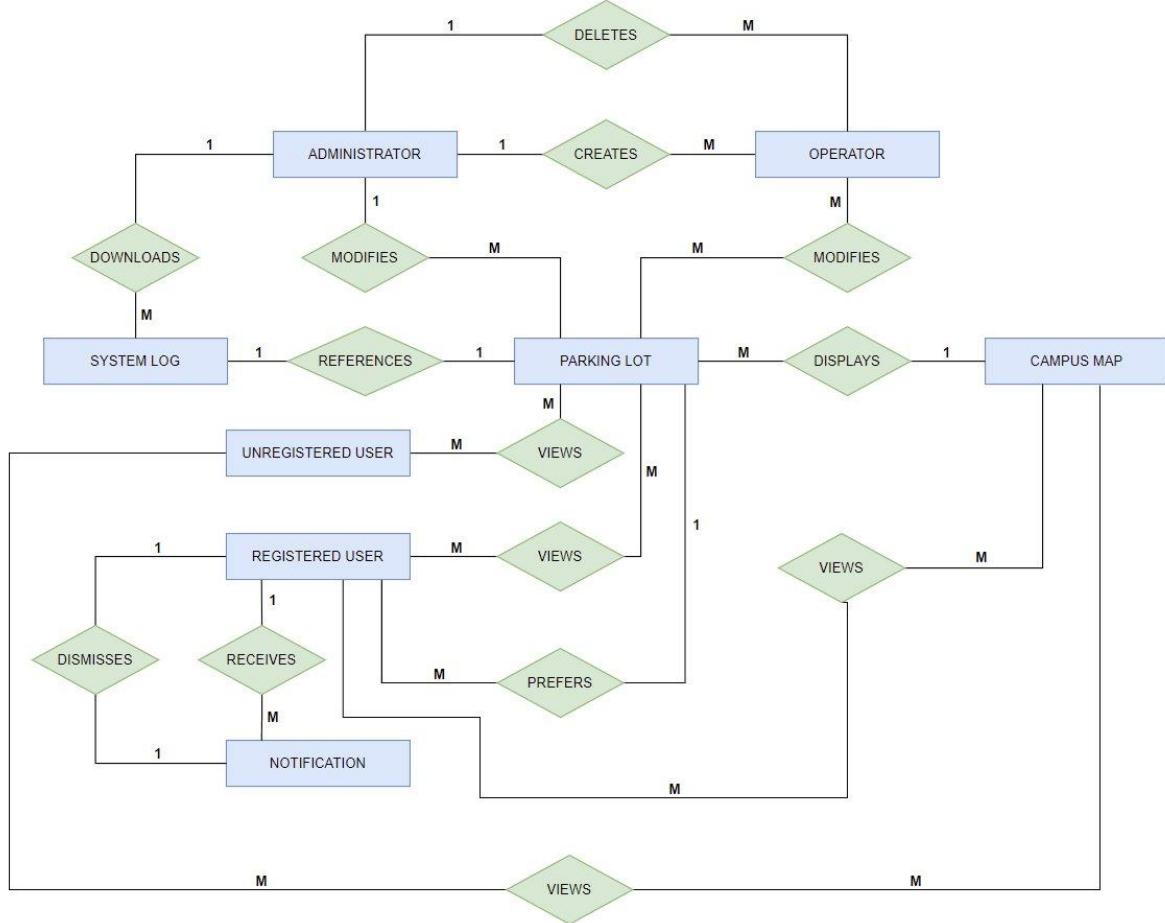
## DFD6.1 Operator Sets New Password



## DFD7 - Registration System



### 3.3 Entity-Relationship Diagram



### 3.4 Data Dictionary

Operator Table					
PK/FK	Field Name	Caption	Data Type	Field Size	Notes
PK	operator_id	Operator ID	INT		NOT NULL
	user_name	Operator Username	VARCHAR	100	NOT NULL
	role	User Type	CHAR	10	NOT NULL
	password_hash	Secured Password	CHAR	72	NOT NULL
	is_using_default_password	Default Password or Not	BOOLEAN		DEFAULT TRUE

Registered User Table					
PK/FK	Field Name	Caption	Data Type	Field Size	Notes
PK	user_id	User ID	INT		NOT NULL
	email	User's Email Address	VARCHAR	100	NOT NULL
	role	User Type	CHAR	10	NOT NULL
	password_hash	Secured Password	CHAR	72	NOT NULL
FK	preferred_parking_lot_id	Preferred Parking Lot ID	INT		
	show_notifications	Has Opted Into Notifications	BOOLEAN		DEFAULT FALSE

Parking Lot Table					
PK/FK	Field Name	Caption	Data Type	Field Size	Notes
PK	lot_id	Parking Lot ID	INT		NOT NULL
	lot_name	Parking Lot Name	VARCHAR	100	NOT NULL
	latitude	Parking Lot Latitude	DECIMAL	(8,6)	NOT NULL
	longitude	Parking Lot Longitude	DECIMAL	(9,6)	NOT NULL
	radius	Parking Lot Size (Radius)	INT		NOT NULL
	total_capacity	Total Parking Spots	INT		NOT NULL
	available_capacity	Available Parking Lots	INT		NOT NULL
	status	Parking Lot Availability Status	CHAR	20	NOT NULL

Notification Table					
PK/FK	Field Name	Caption	Data Type	Field Size	Notes
PK	notification_id	Notification ID	INT		NOT NULL
	title	Notification Title	VARCHAR	100	NOT NULL
	content	Notification Message	VARCHAR	100	NOT NULL
FK	for_user_id	User to Send Notification to	INT		NOT NULL
	is_active	Notification View Status	BOOLEAN		DEFAULT FALSE

Campus Map Table					
PK/FK	Field Name	Caption	Data Type	Field Size	Notes
PK	campus_id	Campus ID	INT		NOT NULL
	campus_name	Campus Name	VARCHAR	100	NOT NULL
	latitude	Latitude	DECIMAL	(8,6)	NOT NULL
	longitude	Longitude	DECIMAL	(9,6)	NOT NULL
	radius	Campus Size (Radius)	INT		NOT NULL
FK	parking_lots	Parking Lots in Campus	INT [0...n]	100	

Administrator Table					
PK/FK	Field Name	Caption	Data Type	Field Size	Notes
PK	admin_id	Administrator ID	INT		NOT NULL
	user_name	Administrator Username	VARCHAR	100	NOT NULL
	role	User Type	CHAR	10	NOT NULL
	password_hash	Secured Password	CHAR	72	NOT NULL

System Log Table					
PK/FK	Field Name	Caption	Data Type	Field Size	Notes
PK	log_id	Log ID	INT		NOT NULL
	log_type	Log Level Type	CHAR	10	NOT NULL
	message	Logged Message	VARCHAR	1000	NOT NULL
	lot_id	Associated Parking Lot ID	INT		

### 3.5 Data Dictionary Notes

The status field in the *Parking Lot Table* can take one of the following three values:

- **AVAILABLE:** This status indicates that the total available capacity is greater than or equal to 5%, determined by dividing the total available capacity by the total capacity, and multiplying by 100.
- **FULL:** This status indicates that the total available capacity is strictly less than 5%, determined by dividing the total available capacity by the total capacity, and multiplying by 100.
- **CLOSED:** This status can be manually set by an Administrator or an Operator, indicating that a Parking Lot cannot be accessed.

# 4 UI Models

The storyboards illustrating use cases as user flows are shown, complete with an appropriate color palette that adheres to The Client's branding guidelines.

## 4.1 End User UI

Storyboard - Create Account

**1. Unregistered Home**  
Unregistered User clicks Login button.

**2. Account Login**  
Unregistered User clicks the Register Now button.

**3. Create Account**  
Unregistered User fills in the registration fields.

**4. Create Account Error**  
Password error is shown.

**5. Create Account Filled In**  
Unregistered User clicks the Create Account button.

**6. Account Created**  
Unregistered user is now a Registered User.

Storyboard - Login

1. Logged Out Home

2. Account Login

3. Account Login Filled In

4. Account Login Error

1. Homepage  
Registered User clicks Login button.

2. Login  
Registered User fills out the login form.

3. Login Attempt  
Registered User click the Login button.

4. Login Error  
If credentials are invalid, an error is shown.

Storyboard - Manage Notifications

1. Logged In Home

2. Account Settings

3. Manage Notifications

4. Disable Notifications

1. Homepage  
Logged In User clicks the Account Settings button.

2. Account Settings  
Logged In User clicks the Manage Notifications button.

3. Manage Notifications  
Notification preferences are shown.

4. Disable Notifications  
Logged In User clicks the toggle to disable notifications.

Storyboard - Delete Account

1. Logged In Home

2. Account Settings

3. Delete Account

4. Unregistered Home

1. Homepage  
Logged In User clicks the Account Settings button.

2. Account Settings  
Logged In User clicks the Delete Account button.

3. Homepage  
Warning is shown and Logged In User clicks the Delete Account button.

4. Unregistered  
The Logged Out Home view is shown.

Storyboard - Favourite / Unfavourite Preferred Parking Lot + Notification Opt-in / Opt-out

1. Logged In Home

2. Favourite Parking Lot

3. Parking Lot Favoured

4. Disable Notifications

5. Unfavourite Parking Lot

6. Parking Lot Unfavoured

1. Homepage  
Logged In User clicks an available parking lot.

2. Favourite Parking Lot  
Logged In User clicks the Favourite button.

3. Parking Lot Favoured  
Success is shown, and notification preferences are shown.

4. Disable Notifications  
Logged In User clicks the Notification toggle to disable notifications.

5. Unfavourite Parking Lot  
Logged In User clicks the Unfavourite button.

6. Parking Lot Unfavoured  
Success is shown.

Storyboard - Log Out of Account

1. Logged In Home

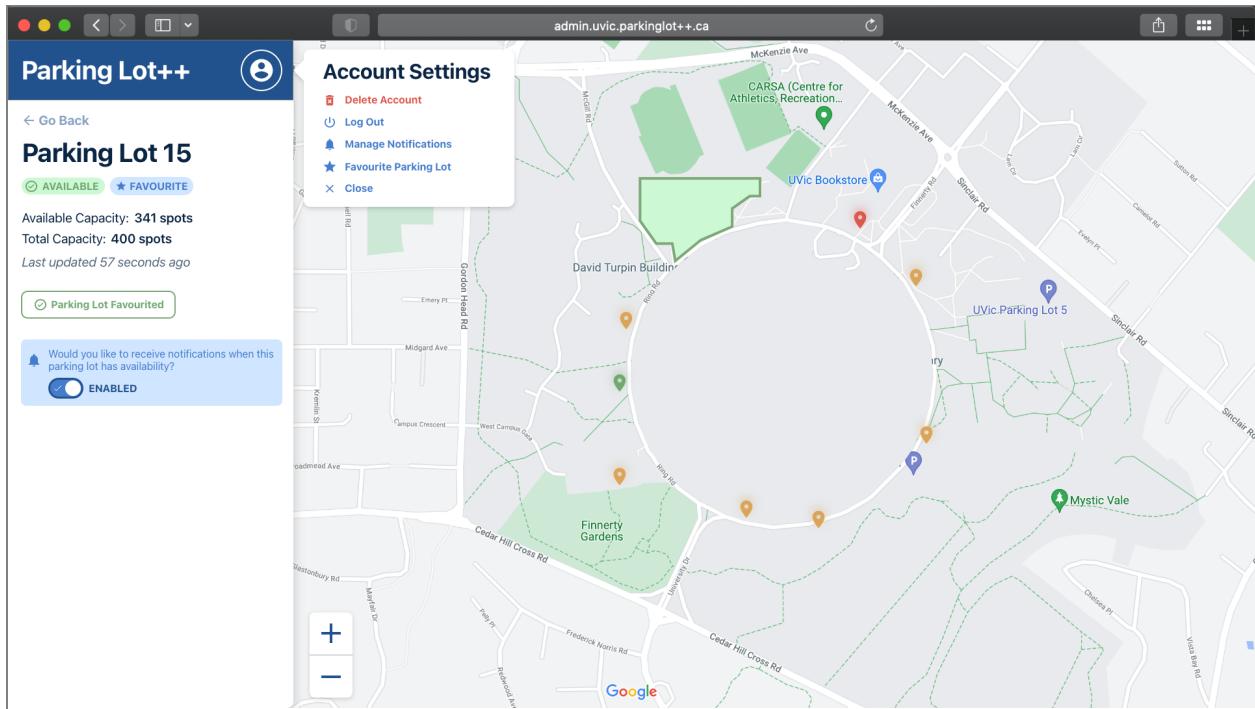
2. Account Settings

3. Logged Out

1. Homepage  
Logged In User clicks the Account Settings button.

2. Account Settings  
Logged In User clicks the Log Out button.

3. Logged Out  
The Logged Out Home view is shown.



*Example of a Desktop view for End Users*

## 4.2 Admin & Operator Desktop UI

Storyboard - Administrator / Operator Log In

**1. Login Page**  
Administrator / Operator visits the login page.

**2. Login Page Filled Out**  
Administrator / Operator enters their credentials and clicks the Login button.

**3a. Admin Home**  
Administrator sees administrator portal home page.

**3b. Operator Home**  
Operator sees operator portal home page.

**3a/b. Successfully Logged In (Admin)**  
Administrator sees administrator portal home page.

**3a/b. Successfully Logged In (Operator)**  
Operator sees operator portal home page.

Storyboard - Administrator / Operator Log Out

**1. Logout**  
Administrator / Operator clicks the Logout button.

**2. Login Page**  
Administrator / Operator sees the Login page.

Storyboard - Administrator Downloads System Logs

The storyboard consists of four panels illustrating the process of downloading system logs:

- 1. Login Page**  
Administrator clicks the download System Logs button.
- 2. System Logs Page**  
Administrator sees the systems log download page.
- 3. Enters Log Criteria**  
Administrator enters log criteria and clicks the Download System Logs button.
- 4. System Logs Downloaded**  
System logs are zipped and downloaded onto the Administrator's computer.

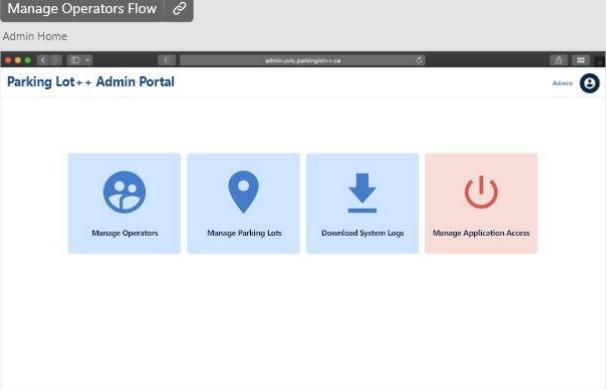
Storyboard - Administrator Disables Application Access

The storyboard consists of four panels illustrating the process of disabling application access:

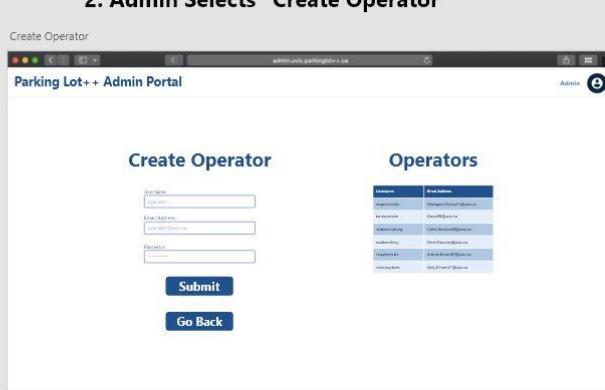
- 1. Home Page**  
Administrator clicks the Manage Application Access button.
- 2. Manage Application Access**  
Administrator sees the application access page.
- 3. Disable Application Access**  
Administrator flips the access toggle and the application is now inaccessible to End Users.
- 4. End User View**  
End Users are no longer able to access the Parking Lot++ application.

**Creator Operator Flow**

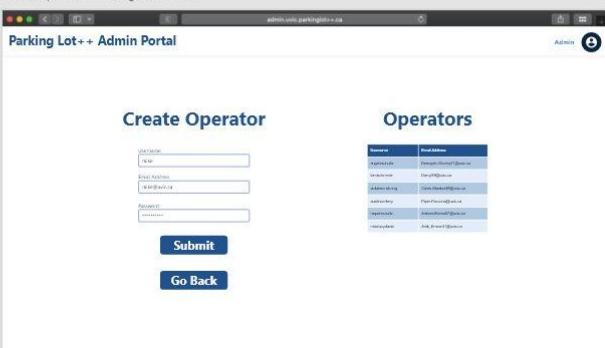
**1. Admin Selects "Manage Operators"**



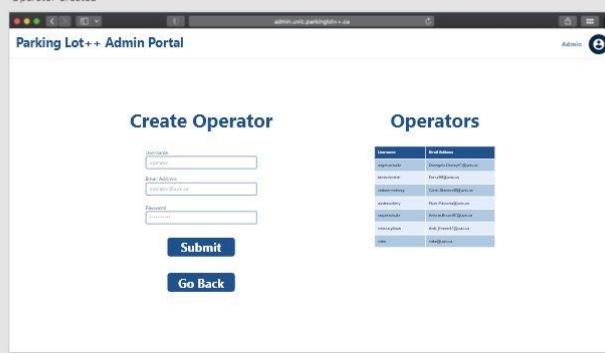
**2. Admin Selects "Create Operator"**



**3. Admin enters username, email, and password**

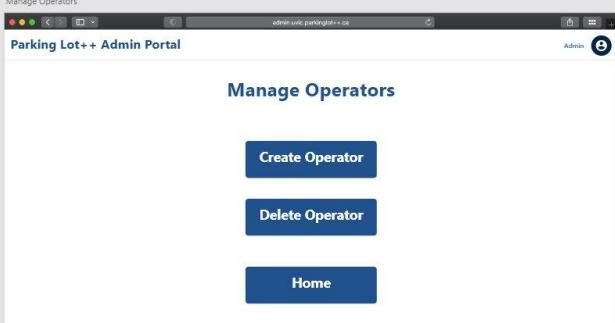


**4. Operator is added to the table on the right**

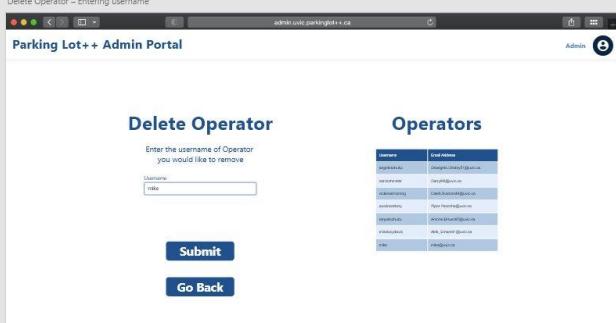


## Delete Operator Flow

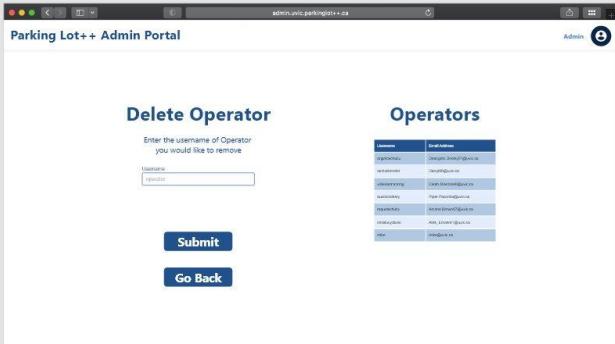
**1. Admin Selects "Manage Operators"**



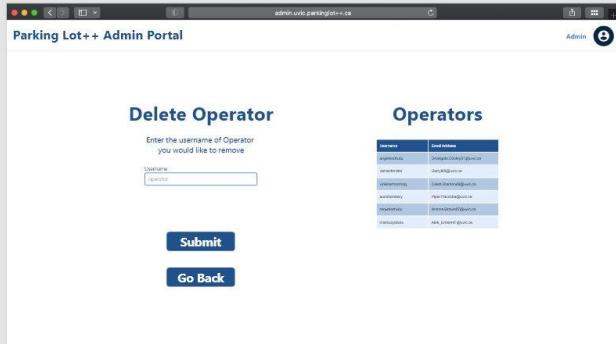
**3. Admin enters username of operator to delete**



**2. Admin Selects "Delete Operator"**

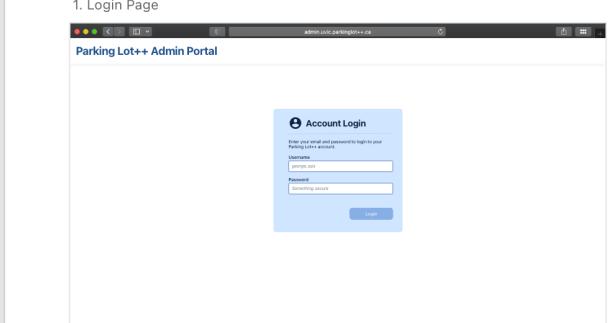


**4. Operator is removed from the table on the right**

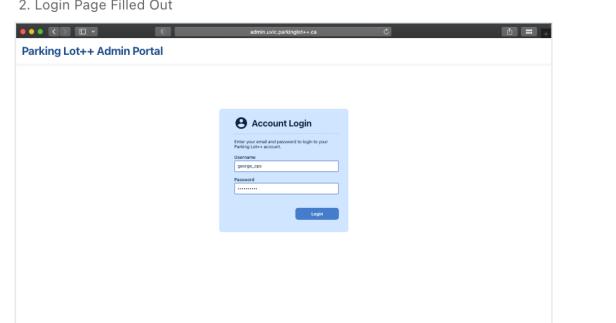


### Storyboard - Operator Change Password on First Login

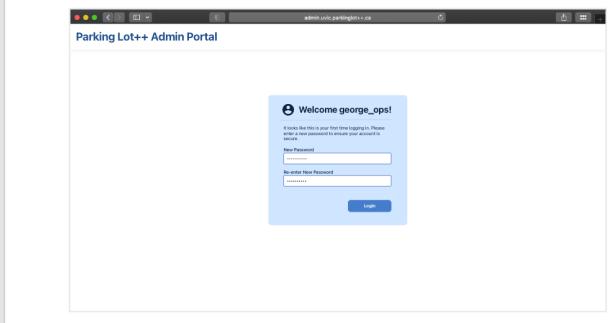
**1. Login Page**  
Operator visits the login page.



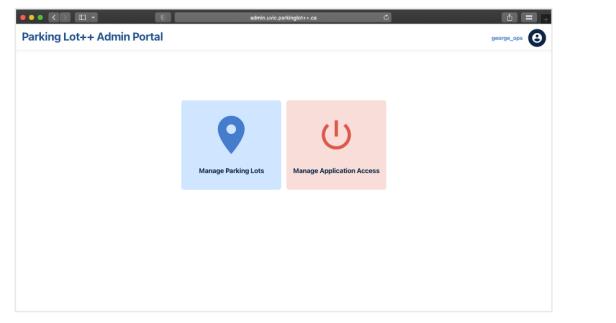
**2. Login Page Filled Out**  
Operator enters their credentials and clicks the Login button.



**3. First Login Change Password**  
Operator is prompted to change their password since this is their first login. Operator enters the new password and clicks the Login button.



**4. Operator Home**  
Operator's password has been successfully updated and they now see the Operator portal home page.



## 5 Conclusion & Recommendations

The Parking Lot++ application will greatly improve the parking experience for the students at the University of Faketoria, by giving students central access to parking lot availability. With the recent uptake in students, faculty, and visitors, this application is essential to decreasing wasted time on campus. The Software++ team looks forward to releasing the Parking Lot++ application as an MVP in late 2022, and will continue to enhance and support the application as end users begin to report feedback.

For any future analysts who are working on Parking Lot++ or a similar application, we strongly suggest gathering as much information as possible at the beginning of the project. This, in conjunction with visually describing user flows with the client, will help prevent miscommunication, and ambiguous requirements later in the process. We also suggest that all models (use cases, DFDs, storyboards, etc.) all have a singular template that can be shared between the team. This will help decrease the “normalization” time needed to ensure all document elements are consistent before submission. In terms of modelling, we suggest looking closely at the use cases, DFDs and ERD when creating storyboards. This will help find missing information or data in the DFDs and ERD that are required for a good user experience.

Finally, we would like to thank the University of Faketoria and the CSC375 teaching team for providing a wonderful learning experience in the form of this final project.

# Appendix A: General Documents

All project related documents can be found at the project's website (<https://nilaysondagar.github.io/csc375-analyst-website/>), including:

- General project information
- Our team and their roles at Software++
- The Client's RFP
- The Project Charter
- Client meeting notes and requirements

All domain / project specific terms are defined in the glossary, found in *Section 1.1.7*.