Software Requirements Specification

for

Toronto Parking System

Version 1.0 approved

Prepared by Arnesh Dawar, Shivani Joshi, Omar Mohamud, Karmit Patel, Kamil Sarbinowski, and Priyam Shah.

York University

November 11th, 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
All	12/11/20	Initial Draft	1.0

1. Introduction

1.1 Purpose

The product, whose software requirements are specified in this document, is formally known as the Toronto Parking System. The system-to-be is intended to modernize the current parking system, which is not automated and relies on traditional methods such as pen and paper and using cash. It is believed that the current system is inefficient and outdated – by utilizing mobile apps, networking protocols and a database system, a unification of three actors and their respective systems into one allows for a more organized approach to a complicated logistic problem.

In addition to this, there are benefits for all parties involved: for customers, convenience and new features such as pre-booking spaces will be created; for the Toronto Parking Authority, spaces will be easier to monitor and payment can be collected instantaneously, without sending employees across the city; for the Toronto Police Service, payment can be more easily enforced and tickets issued easier and in an electronic manner, saving time and paperwork for all parties involved. The entire scope, goals, and aims of the system-to-be are further laid out in **Section 1.4**.

1.2 Document Conventions

In this document, the main actors identified are: the customer, henceforth referred to as the 'Patron'; the police officer on duty, henceforth referred to as the 'Enforcement Officer'; finally, the system-to-be itself, henceforth referred to as 'System'. Any requirement listed without a priority is assumed to be of critical or utmost importance; as a matter of fact, almost every requirement included is of critical importance to accomplish the goals set out in **Section 1.** Priorities are given on a scale of 1-10, where a requirement of priority 8 and above could be classified as critical or crucial, and a requirement of 4 or lower could be considered optional or nice to have, time permitting.

1.3 Intended Audience and Reading Suggestions

This document is intended to be read by the main stakeholders; namely, Professor Song Wang, the Toronto Police Service, and the Toronto Parking Authority. In addition, the development team responsible for implementation is also a part of the intended audience.

To save the development team time, it is suggested that they read the report in the following order:

1. Section 1
2. Section 2.1, 2.2
3. Section 4
4. Section 5
5. Section 3
6. Remainder of Section 2

The reason for this sequence is to provide the development team with background information of the problem before viewing our proposed system-to-be. This information (problem definition and overview of requirements) is mainly contained in sections 1 and 2.1-2.2. Following this, the functional and non-functional requirements contain the most vital information regarding what the system must include and what would be a benefit to include should time permit, and these are located in sections 4 and 5. As the developer team, these are of utmost importance but probably require the context of the problem outlined in sections 1 and 2.2 before they are read. Following this is the rest of section 2 and section 3, which cover the specifics of implementation such as languages, classes, libraries, software, and hardware used to implement the system-to-be. Given that more emphasis is being placed on the requirements being met, and that the skills and strengths of the developer team are currently unknown, these sections are more of a guideline rather than being firm requirements for implementation.

For example, the implementation of a database system can be made from a choice of options outlined in section 3.3 rather than having it as a constraint. Implementation is a last step and given the context of this project, it would make sense to allow the developer team a say in the final implementation software and hardware.

1.4 Product Scope

The software being specified in this SRS is a parking management and payment system for downtown Toronto. At present, there are over fifteen thousand spaces to be managed by the Toronto Parking Authority, and the payment for these spaces need to be enforced and collected manually.

The purpose of the software system being specified is to streamline and provide an efficient method for customers to park their vehicles and provide payment for their rented spaces. By utilizing this system, customers should be able to provide prepayment for parking spaces, book parking spaces well in advance, and have access to more payment methods than if they were to use a parking meter in person.

In addition, the system will allow Toronto Parking Services to update their fees per space in real-time, collect and monitor payments, and ensure that the amount of revenue collected from each space is maximized. In addition to this, payment would not need to be collected manually from the nine thousand different locations of parking spaces that they are currently being collected from. This can streamline the department and allow the redistribution of resources into another area such as purchasing more parking spaces and increasing revenue further.

Given the magnitude of the amount of spaces to be monitored across Toronto, there are cracks in the current, inefficient and manual system - especially on the enforcing side. By implementing this system the enforcement agency involved, the Toronto Police Service, would also benefit. The system should allow the enforcement agency to view, manage and update parking spaces in real-time. The system should also allow the enforcement agencies to fine or print tickets for users parked illegally. Doing so can allow a redistribution of Toronto Police Service resources and save money in the long run.

2. Overall Description

2.1 Product Perspective

The origin and context of such a parking system as defined in **Section 1.4**, the product scope, is essentially the unification of three existing, non-automated and non-computer aided subsystems that currently exist into a single system. This system should allow the actors outlined in **Section 2.3** to utilize the features laid out in **Section 4.**

It is the intent that this project work alongside the current system to provide users with the option of using the old system or the new one, at least in the interim. The reason for this is that not everyone who owns a car also has an internet connection at all times in order to pay the spot. However, it is the goal of the project to eventually replace the original, rudimentary system as pre-paying is allowed. It is inefficient for employees to travel around the city collecting payments from meters and having police officers checking for valid parking - ideally this should be automated or, at the very least, done with a centralized database storing information and allowing live updates.

2.2 Product Functions

The major functions required by the system are fully detailed and described in the attached Appendix A, use case diagrams. They are also detailed in sections 4 and 5. A brief overview of the functionality is as follows:

- Patrons should be able to register an account into the system, by providing a username, password, address and security question.
- While registering, they should be able to provide a security question and answer
- An individual patron should be able to search available parking spots at any location.
- Patrons should be able to search for parking spots near their current location.
- Patrons should be able to reserve a parking spot after browsing.
- Patrons should be able to pay for their reservations
- The system should be able to process a payment
    - This involves paying via debit, credit
- Patrons should be able to report an issue
- Enforcement officers should be able to register
- Enforcement officers should be able to search spots, and manage them
- Enforcement officers should be able to issue a ticket and/or fines
- Enforcement officers should be able to view, resolve, or update issues raised by patrons

2.3 User Classes and Characteristics

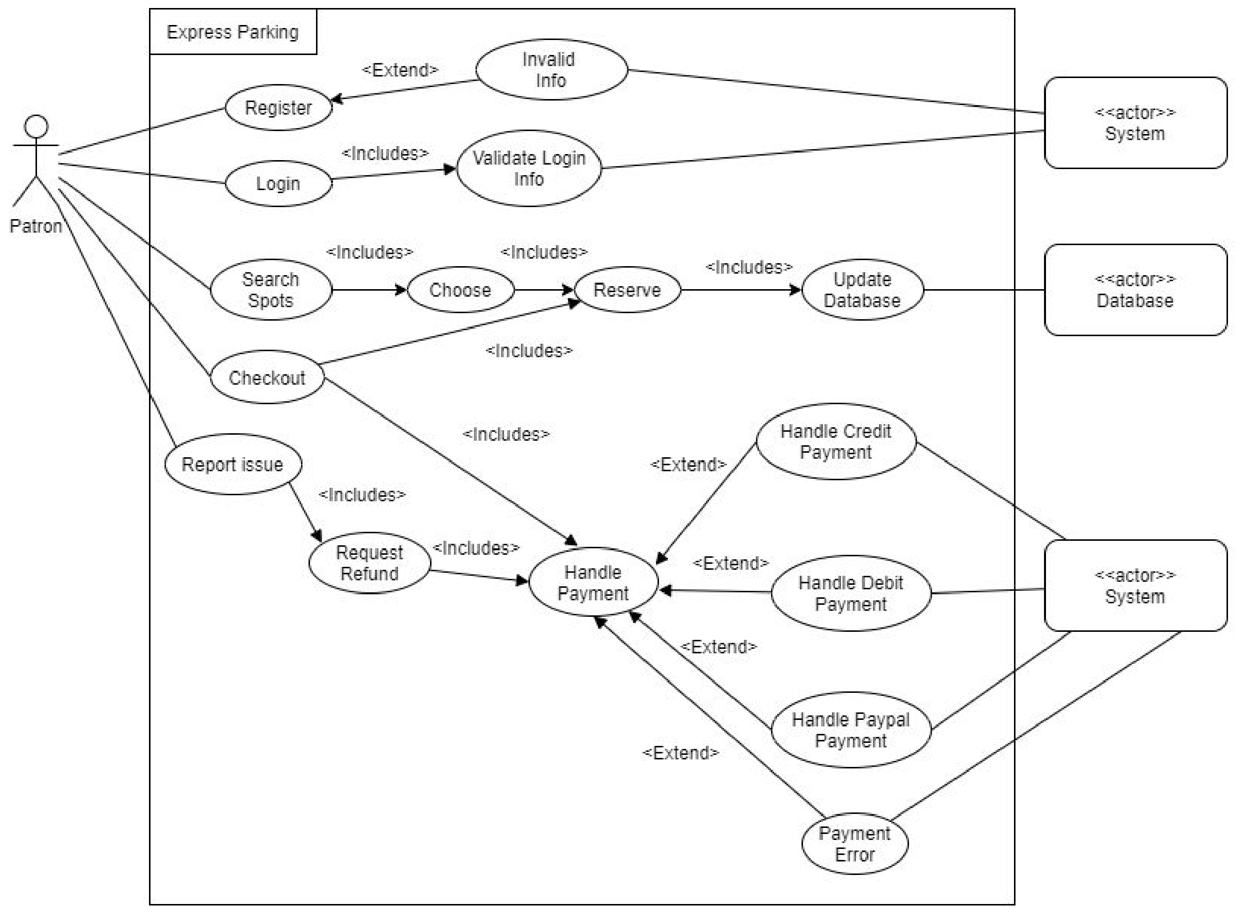

Figure 1 shows use cases for Patron

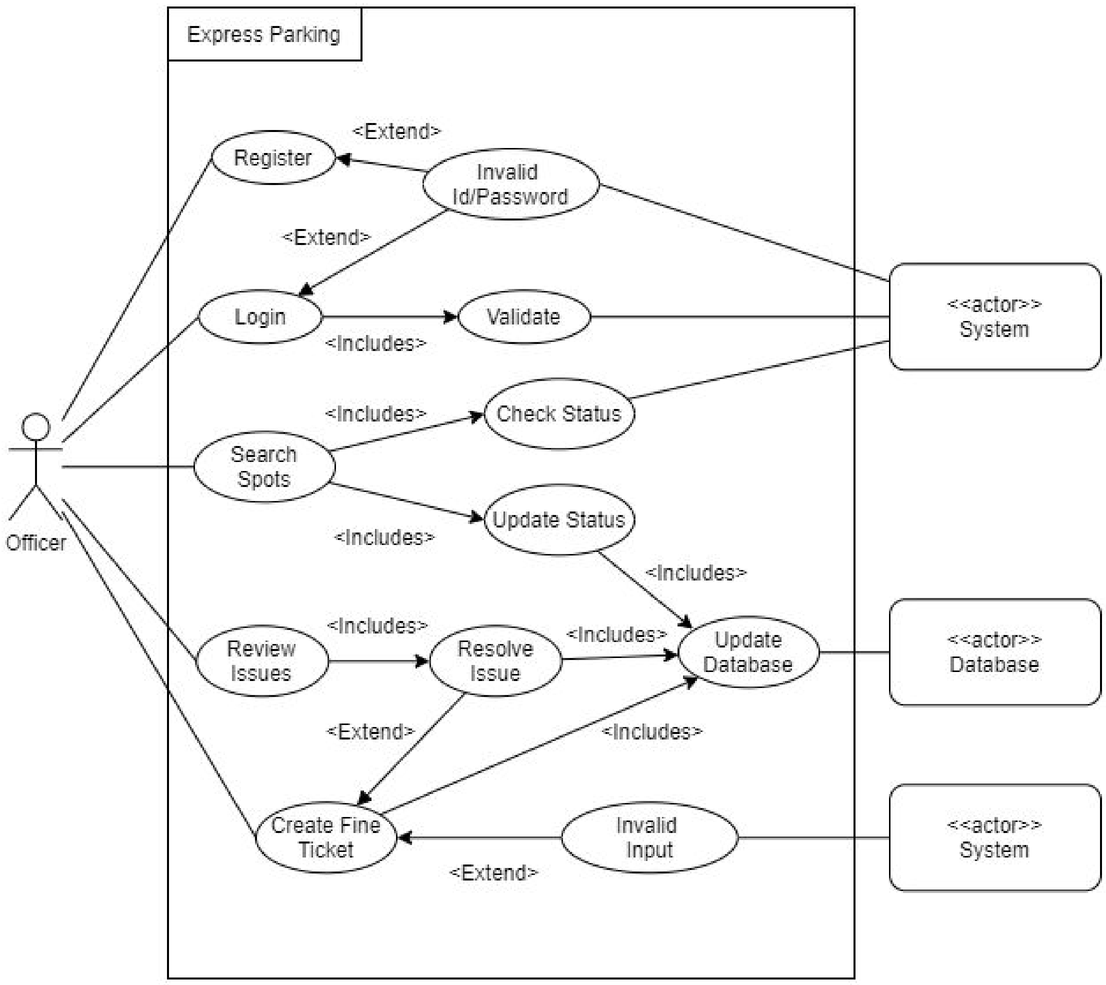

Figure 2 shows use cases for Enforcement Officer

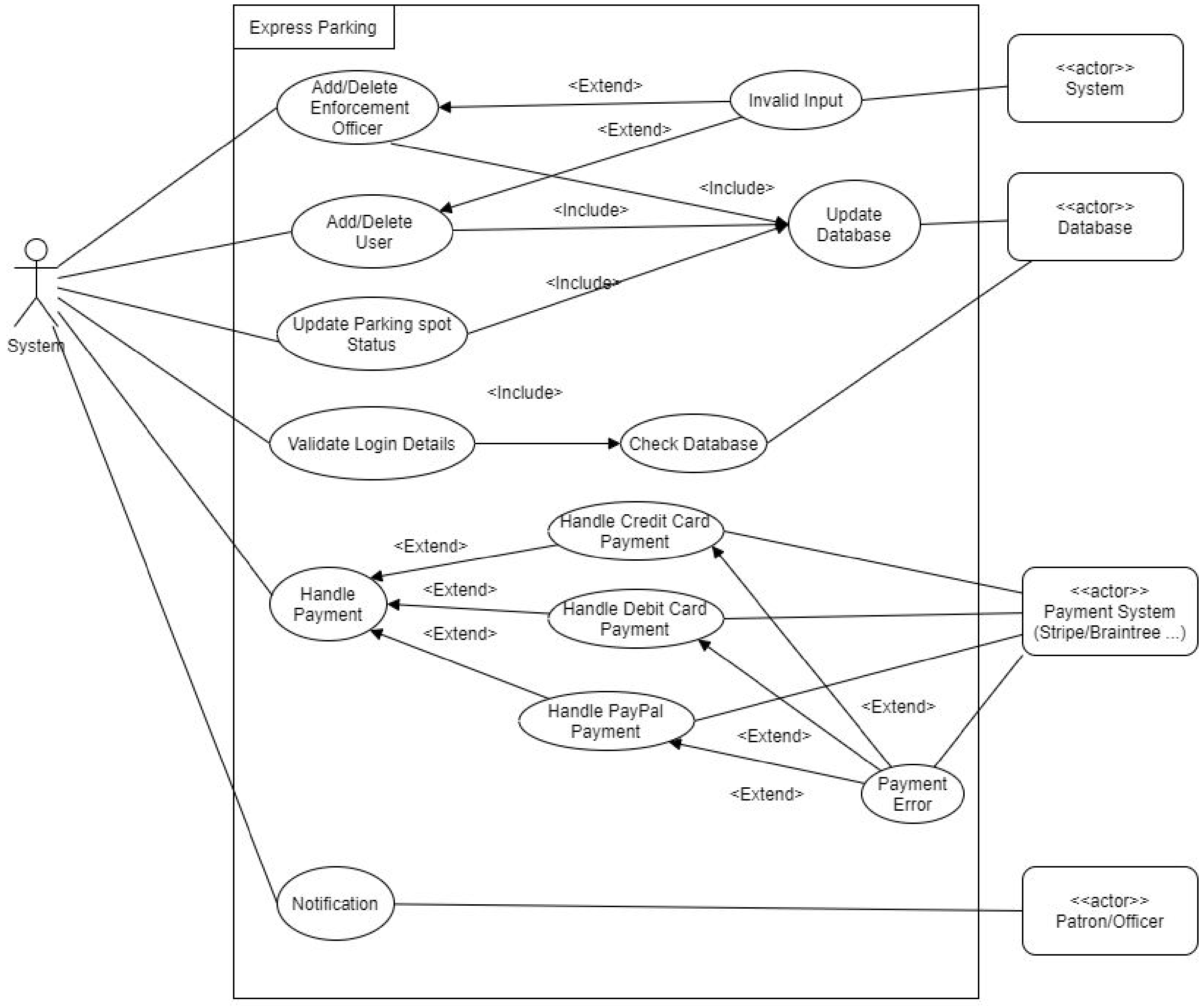

Figure 3 shows Use Cases for System

2.4 Operating Environment

The system is an online website that runs on web-browsers on multiple platforms. Hence there are certain requirements for its operating environment. Refer to **Section 3.3,** for detailed explanation.

Operating Systems:

1. Android (Android Jellybean or greater)
2. IOS (IOS 5.0 or greater)
3. Macintosh (Mac OS X or above)
4. Windows (Windows 2000, Xp or greater)
5. Linux (Red Hat 7.0 or above)

Hardware Platforms:

1. Mobile Devices (Android or IOS)
2. PCs (Windows)
3. Macintosh (Mac OS)

2.5 Design and Implementation Constraints

Given the nature of this project, being that the developer team is completely separate from the PM team, it is difficult to place implementation constraints on the developers. The reason for this is that the strengths and weaknesses of the team are still unknown. As a result, regarding **Section 3** and **Section 2**, implementation details can be taken as a guideline rather than as a strict requirement.
The PM team recommends using Java for Object Oriented Programming due to its portability and runnability on Android and iOS devices; a modern RDBMS and frontend combination compatible with Java (such as JSP & apache tomcat + derby) is recommended as well for the server side, and is taught at York University in some upper year courses.

2.6 User Documentation

The users will be provided with two documents for references:

1. The user will be provided with this SRS documentation to refer in case of any issues with the operating system or compatibility with hardware and software.
2. The user will also be provided with the use case diagrams and documentation to refer to the steps to use a particular feature.

The user will also be able to access a demo video that shows selected key functionality of the application.

2.7 Assumptions and Dependencies

This documentation and system assumes certain basic requirements are already fulfilled to use the application.
Assumptions:

1. The devices used by the users are all connected to the internet while interacting with the system.
2. All the devices and operating systems are compatible with the system as mentioned in **3.2 and 3.3.**

Dependencies (Services):

1. The system will be utilizing PayPal and other 3rd party payment processing frameworks to bill patrons.
2. The system will also be utilizing some 3rd party authentication service during login.
3. The system will be dependent on some online database service to provide features to store, update, delete and access data remotely.

It is assumed that all the above mentioned dependencies lack vulnerabilities and have in-built security rules to ensure data integrity and security.

3. External Interface Requirements

3.1 User Interfaces

First Time Registered Users: First Time Registered Users must be given a small tutorial each page that the user can optionally skip.

Design Components: Individual design components like buttons, toolbars, menus, sliders and other user interactive components must be consistent throughout the system to ensure ease of learning for the users.

The GUI of a main screen or home screen is **required** to be different for Enforcement Officers and Patrons due to their differing access to features as is outlined throughout **Section 4**. Buttons should exist only for features that an actor has access or permissions to use. For example, Patrons should not be able to view the information of other Patrons, but the Enforcement Officer should be able to view the information of any Patron.

The GUI should be straightforward, accessible, functional, and concise. A focus should be on functionality rather than appearance; at a low priority of 6 is updating the appearance of the GUI to be appealing to the eye. Functionality is far more important and the system should be easy to use and bug free at a much higher priority of 10.

3.2 Hardware Interfaces

This section describes the hardware interface requirements for the system.

The system shall provide its patrons, feature to print parking confirmation. It also allows Enforcement Officers to print fine / parking tickets to be distributed manually. Hence, the system needs to interact with a printer using wireless(Wifi/Bluetooth) or wired(USB) connections. For better adaptability, the system shall be compatible with a wide variety of printers like Laser, Inkjet, and Dot Matrix that support the above connection methods.

REQ-3.2.1 The system must interact with a printer via Wifi, or wired connection. (eg. USB Connection)
REQ-3.2.2 The system shall be compatible with all types of printers. (Ed. Inkjet, Laser, or Dot Matrix)

The system is an online website that runs on a web-browser, hence it must be compatible with a large variety of devices. The software must be reasonably platform independent so it will run on a wide range of machines with minimal impact on its software capabilities. This will allow the developers to go through hardware changes without a significant re-writing of the software.

REQ-3.2.3 The system shall be compatible with commodity PCs.
REQ-3.2.4 The system shall be compatible with Macintosh Computers.
REQ-3.2.5 The system shall be compatible with handheld devices running Android or IOS.

The system shall allow a wide range of input methods for users to interact with the software.
REQ-3.2.6 The system shall allow touch based input functionality on handheld devices.
REQ-3.2.7 The system shall allow input via a physical or virtual keyboard and mouse input on PCs and Macintosh Computers.

3.3 Software Interfaces

This section describes the software interfaces and requirements for the system.

Since software will be running on several hardware platforms described in **3.2**, it shall also be running a large variety of Operating Systems supported by that hardware. At a minimum, the software must run on Android, IOS for handheld devices and Windows for PCs. Since, the goal is to make the software platform independent, the system should run on different Operating Systems too with minimal changes in code.

REQ-3.3.1 The system shall be compatible with Android Operating System (Android JellyBean or greater)
REQ-3.3.2 The system shall be compatible with Windows Operating System (2000, XP or greater)
REQ-3.3.3 The system shall be compatible with Macintosh Operating System (Mac OS X or greater)
REQ-3.3.4 The system shall be compatible with IOS (IOS 5.0 or above)
REQ-3.3.5 The system shall be compatible with Linux operating System (Red Hat 7.0 or greater)

The system needs to be connected to an online database at all times. The database and architecture design is independent of the software of the system, hence a wide range of database services shall be supported by the system. Although, one constraint of the database is it is required to have real-time update functionality. The communication between the software and database shall be handled by web browsers.

REQ-3.3.6 The system shall work with any online database service.
REQ-3.3.7 The online database service shall have a real-time update functionality.

3.4 Communications Interfaces

The software shall be an online system hence it is required to have constant internet connectivity. It shall provide features where it will be communicating with users using multiple forms of communication like emails and texts. The data storage and analysis in the system shall be handled by an online database that needs to interact with the software.

REQ-3.4.1 The system shall be accessible via the internet (Running with a browser, launched from a browser, accessible remotely)
REQ-3.4.2 The system shall be connected to an online database that handles data collection and analysis for the system.

REQ-3.4.3 The system shall communicate with users via email or text (Registered numbers only)

Database Type: SQL, noSQL, Object Oriented
Database Communication Protocols: SQL queries, HTTP Requests (get, put, delete, update)

4. System Features

The section describes the features provided by the system. Each feature consists of a description, priority, actor, main stimulus response for the feature to work as intended and functional requirements for the feature.

4.1 Patron: Registration

4.1.1 Overview, Description and Priority

This feature should allow any Patron who owns a vehicle and valid driver's license to register an account with the system. To do so, they must decide upon a username and password, as well as answer security questions. Usernames must be unique and passwords must meet the laid out requirements in 4.1.2. While registering, personal information including email address, home address, license information and payment method must be provided.

The priority of this feature is 10. In order for any other system feature to be implemented, Patrons would need to be registered into the system. As a result, it is of critical importance to the project.

4.1.2 Stimulus / Response sequence

1. The Patron accesses the system and decides to register an account.
2. The Patron sets a username and password for their account.
3. The Patron answers security question(s).
4. The system ensures that the username selected is unique and the chosen password is well constructed (at least 8 characters, at least 1 number, at least 1 special character).
5. The Patron starts filling in the required information for registration including email address, other personal information, license plate number(s) associated with their vehicle(s), it's type and desired payment method.
6. The Patron submits the information once done.
7. The system determines whether the payment method is valid, email address is well formed and the entered home address is in a valid geographical location (eg. in Canada).
8. The Patron gets a confirmation message containing their account details.

4.1.3 Functional Requirements

REQ-4.1.1 The system shall provide a registration form to the Patron which requires a unique username and a password.
REQ-4.1.2 The system shall provide a method of verifying that the username entered is unique and the password meets the requirements (length of 8, 1 number, 1 special character)
REQ-4.1.3 The system shall provide security questions and record the Patron's answer to the questions.
REQ-4.1.4 The system shall request the Patron's personal information including email, home address, license info, and payment method
REQ-4.1.5 The system shall verify that the email is well formed, that the address exists, that the license is not expired, and that the payment method is valid.

REQ-4.1.6 The system should add an entry to a database to keep track of this new user and to ensure that the username will not be re-used, as they are to be unique identifiers.

4.2 Patron: Login

4.2.1 Overview, Description and Priority

This feature should allow any Patron who owns is already registered via the process described in **4.1** to login to their account. An 'account' is defined in this context as being uniquely identified by username as a primary key in a database type system. After logging in, the Patron should be allowed to perform actions on their account such as updating and saving personal information. From this login, the Patron should also be able to search, reserve, and provide payment for parking spaces as will be defined in **4.3, 4.4,** & **4.5**, respectively. The priority of this feature is 10, as it is critical that this feature be implemented to allow access to all other meaningful features in the system.

4.2.2 Stimulus / Response sequence

1. The Patron visits the SaaP and decides to login.
2. The Patron identifies self.
3. The system ensures that the identity is valid.
4. The Patron can use all functionalities offered by the SaaP.
5. The Patron logs out.

4.2.3 Functional Requirements

The system shall provide a registration form to the Patron which requires a unique username and a password.

REQ-4.2.1 The system shall provide a method of logging into the system with a username and password.

REQ-4.2.2 The system shall provide a way to allow the user to edit the personal information related to their account.

REQ-4.2.3 The system shall provide options or an interface to allow the user to search for parking spaces, as defined in **4.3**.

REQ-4.2.4 The system shall provide options or an interface to allow the user to reserve a parking space, as defined in **4.4**.

REQ-4.2.5 The system shall provide options or an interface to allow the user to checkout their purchases, as defined in **4.5**

4.3 Patron: Search

4.3.1 Overview, Description and Priority

This feature should allow a patron to search for available parking spaces within a 25 km radius of their current location. The search results should be able to be filtered by vehicle type in order to ensure that the space will be sufficient for their currently selected and registered vehicle. After the Patron is logged on, as specified in **4.2**, they should be able to select an option which allows them to search. The Patron would then enter their current location by postal code and select a space from the returned results. Upon selecting a result, the information and status regarding the space should be displayed to the Patron. The priority of this system feature is 10, as it is critical to **4.4, 4.5**, and the main objective of the system.

4.3.2 Stimulus / Response sequence

1. The Patron logs in and decides to search.
2. The Patron enters the Postal Code for their desired search location and the type of vehicle(motorcycle or car) they would like to get a parking spot for.
3. The system ensures the entered Postal Code is in Toronto.
4. The system provides a list of all nearby single spaced parking meter locations.
5. The Patron can select and view any numbered parking spot from the list, apart from the ones marked 'X' (marked as not usable by the enforcement officers).
6. Upon selecting a numbered parking spot the patron can view it's complete availability status for that day (till the time it's usable).

4.3.3 Functional Requirements

The system shall provide a registration form to the Patron which requires a unique username and a password.

REQ-4.3.1 The system shall provide at least one method of collecting a user's current location (GPS, manual postal code entry).
REQ-4.3.2 The system shall provide a method of searching parking spaces near the given postal code.
REQ-4.3.3 The system shall provide filtering on the search results by a parking space's vehicle type, proximity, and availability.
REQ-4.3.4 The system shall provide options or an interface to allow the user to reserve a selected parking space from the search, as defined in **4.4**.

4.4 Patron: Reservation

4.4.1 Overview, Description and Priority

This feature should allow a patron to access the results of their search as specified in **4.3**. From the results displayed, a patron should be able to select a parking space. The available time slots would then be displayed. From here, they should be able to reserve any available space which is not already in use and is available to be parked in. They should be able to select the time and duration of their reservation, and add the reservation to a cart-like system, in order to move to **4.5** and provide payment. The priority of this system feature is 10, as it is critical to **4.5** and the main objective of the system, similar to **4.2** and **4.3**.

4.4.2 Stimulus / Response sequence

1. The Patron selects a numbered parking spot they wish to reserve after browsing.
2. The Patron chooses between the reserve for now or later options.
    a. The Patron selects between 0.5 hour, 1 hour, 2 hours or 3 hours (max) if they are reserving starting now.
    b. The Patron enters a start time for later and chooses between 0.5 hour, 1 hour, 2 hour or 3 hour (max) time durations they want to reserve for.
3. The Patron submits once done.
4. The system ensures there are no conflicts between the Patron's preference and pre-existing reservations.
5. The system adds this reservation to the checkout cart.

4.4.3 Functional Requirements

The system shall provide a registration form to the Patron which requires a unique username and a password.

REQ-4.4.1 The system shall collect details about the location, duration, and timing of a reservation to be made on a given space.
REQ-4.4.2 The system shall use this information to update itself and ensure that no space is ever double booked.
REQ-4.4.3 The system shall provide a way to add the desired parking space(s) into a cart and forward the choices into a payment feature as defined in **4.5**.

4.5 Patron: Checkout

4.5.1 Overview, Description and Priority

This feature should allow a patron to checkout their cart and provide modifications in the form of either adding or removing items from it. From the checkout, they should be able to add or remove new payment options, and the total price should be calculated by the system. Once the Patron is satisfied with their purchase, they may initialize a system process which handles the payment, defined in **4.6**. The priority of this feature is 10 as it is one of the main goals of the stakeholders in this project to be able to collect or provide payments from this system.

4.5.2 Stimulus / Response sequence

1. The Patron goes to the checkout cart to either modify (add or remove items) thereafter they can pay for the items present in the cart or exit.
2. The system calculates the cost of the items present (including tax) after determining whether a discount for disability parking needs to be applied, depending on the parking lot chosen.
3. The system displays the total amount to be paid by the Patron.
4. The Patron can now select a payment option:
    a. Either from the ones saved under their account.
    b. Or add a new payment method.

5. The system handles payment.
6. The Patron received payment confirmation.
7. The Patron gets a printable version of the parking ticket.

4.5.3 Functional Requirements

The system shall provide a registration form to the Patron which requires a unique username and a password.

REQ-4.5.1 The system will retrieve a reservation request which consists of all the information and details which were specified and collected in **4.4**.
REQ-4.5.2 The system shall use this information to begin one of three sub-checkout processes specified in **4.6 - 4.9**

4.6 System: Checkout

4.6.1 Overview, Description and Priority
This feature should be implemented on the system side and involves the processing of payments and payment information into actual transactions which are authorized, logged in the local database, and further passed along to further handler features, namely **4.7, 4.8, 4.9**. The priority of this feature is 10 as it is crucial to the implementation of providing a payment system.

4.6.2 Stimulus / Response sequence
1. The patron preferred payment method is saved in the system.
2. The system determines the type of payment method saved.
3. The system sends payment authorization requests to the authentication and authorization approval system.
4. System receives payment approval and signals approval to Patron.

4.6.3 Functional Requirements

The system shall provide a registration form to the Patron which requires a unique username and a password.

REQ-4.6.1 The system will retrieve a payment request and process which of the three sub-checkout processes needs to be applied based upon payment type.
REQ-4.6.2 The system shall use this information to begin one of three sub-checkout processes specified in **4.7 - 4.9**

4.7 System: Handling Credit Card Payments

4.7.1 Overview, Description and Priority
This feature should be implemented on the system side and involves the processing of payments which the Patron has decided to use a credit card to pay for. In order to verify that a transaction is valid, it must be communicated to an External Payment Authorization Service. If

the payment should be verified and approved, an External Accounting System can charge the amount to the credit card and return an approval signal to the system-to-be. This approval can finally be forwarded to the Patron. The priority of this feature is 10.

4.7.2 Stimulus / Response sequence

1. The system sends the patron's credit card account details.
2. The system communicates with an External Payment Authorization Service System and requests account authentication.
3. External Payment Authorization Service System conveys authentication to the system.
4. The system requests payment approval, to check if the credit limit is not exceeded on the account, from an External Accounting System.
5. External Accounting System charges the amount to the credit card and signals approval to the system.
6. System receives payment approval and signals approval to Patron.

4.7.3 Functional Requirements

The system shall provide a registration form to the Patron which requires a unique username and a password.

REQ-4.7.1 The system should be able to communicate with an External Payment Authorization Service System to verify and approve the authenticity of transactions.
REQ-4.7.2 The system shall proceed with approval and communicate with an External Accounting System.
REQ-4.7.3 The system shall forward a message of approval or rejection to the Patron depending upon the reply received from the External Accounting System.

4.8 System: Handling Debit Card Payments

4.8.1 Overview, Description and Priority

This feature should be implemented on the system side and involves the processing of payments which the Patron has decided to use a debit card to pay for. In order to verify that a transaction is valid, it must be communicated to an External Payment Authorization Service. If the payment should be verified and approved, an External Accounting System can charge the amount to the debit card and return an approval signal to the system-to-be. This approval can finally be forwarded to the Patron. The priority of this feature is 10.

4.8.2 Stimulus / Response sequence

1. The system sends the patron's credit card account details.
2. The system communicates with an External Payment Authorization Service System and requests account authentication.

3. External Payment Authorization Service System conveys authentication to the system.
4. The system requests payment approval, to check if the credit limit is not exceeded on the account, from an External Accounting System.
5. External Accounting System charges the amount to the credit card and signals approval to the system.
6. System receives payment approval and signals approval to Patron.

4.8.3 Functional Requirements

The system shall provide a registration form to the Patron which requires a unique username and a password.

REQ-4.8.1 The system should be able to communicate with an External Payment Authorization Service System to verify and approve the authenticity of transactions.
REQ-4.8.2 The system shall proceed with approval and communicate with an External Accounting System.
REQ-4.8.3 The system shall forward a message of approval or rejection to the Patron depending upon the reply received from the External Accounting System.

4.9 System: Handling PayPal Payments

4.9.1 Overview, Description and Priority

This feature should be implemented on the system side and involves the processing of payments which the Patron has decided to use Paypal's system to pay for. An External Accounting System can charge the amount to Paypal, and return an approval or rejection signal from paypal to the system-to-be. This approval or rejection can finally be forwarded to the Patron. The priority of this feature is 10.

4.9.2 Stimulus / Response sequence

1. The system sends the PayPal account details.
2. The system communicates with an External Payment Authorization Service System and requests account authentication.
3. External Payment Authorization Service System conveys authentication to the system.
4. The system requests payment approval, to check if the funds in the account are greater than the remittance amount, from an External Accounting System.
5. External Accounting System charges the amount to the account and signals approval to the system.

4.9.3 Functional Requirements

The system shall provide a registration form to the Patron which requires a unique username and a password.

REQ-4.9.1 The system shall proceed with approval and communicate with an External Accounting System, forwarding a charge request to the provided PayPal account.
REQ-4.9.2 The system shall forward a message of approval or rejection to the Patron depending upon the reply received from the External Accounting System, which is in turn received from PayPal, Inc.

4.10 Patron: Reporting Issues

4.10.1 Overview, Description and Priority

This feature should be implemented in order to allow Patrons the ability to request refunds or get assistance regarding illegal parking or their space being occupied already. This can allow prompt assistance from enforcement agencies, as they should be able to react to issues as outlined below in **4.15**. The system or interface should have a clearly defined section which allows users to flag issues and request assistance. The priority of this feature is a 10, and it is crucially important in order to guarantee a Patron's satisfaction as a stakeholder.

4.10.2 Stimulus / Response sequence

1. The Patron is logged into the system and decides to report an issue.
2. The Patron can file either grievance:
    a. Violation of parking laws (eg. illegal unpaid parking)
    b. Inaccessible parking spot (eg. Snowed in parking spot due to bad weather)
3. The Patron fills out the required complaint fields which include violator's license plate number, type of vehicle, date, time and location of violation noticed along with a brief description of the incident.
4. The Patron thereafter can request a refund.
5. The system generates an incident report containing a unique ID.
6. The Patron receives a confirmation regarding incident creation, containing it's ID.

4.10.3 Functional Requirements

The system shall provide a registration form to the Patron which requires a unique username and a password.

REQ-4.10.1 The system shall provide a section, screen, or other designated interface to allow users to report issues.
REQ-4.10.2 The system shall receive and store input from the user which includes the required complaint fields outlined in 4.10.2.
REQ-4.10.3 The system shall allow Enforcement officers to view and respond to issues in real time, as is fully outlined in **4.15**.
REQ-4.10.4 The system shall allow Patrons to request a refund and generate an incident report, identified by a unique ID.

REQ-4.10.5 The system shall automatically send incident reports to the Patron's listed email address.

4.11 Enforcement: Registration

4.11.1 Overview, Description and Priority

This feature should be implemented in order to allow Enforcement Officers the ability to register themselves into the system as an enforcement agent. The Enforcement agent must provide their police badge number and the system should ensure that the badge number has been authorized to become listed as an enforcement officer in the system. The enforcement officer needs to set a username and password just like the Patrons do in **4.1**. The priority of this feature is 10.

4.11.2 Stimulus / Response sequence

1. The Enforcement Officer visits the SaaP to register.
2. The Enforcement Officer enters their police badge number.
3. The system determines whether the submitted police badge number has been authorized.
4. The Enforcement Officer sets a username and password.
5. The system ensures that the username selected is unique and the chosen password is well constructed (at least 8 characters, at least 1 number, at least 1 special character)
6. The Enforcement Officer fills in the required information for registration, including email address, personal information, unique badge number.
7. The Enforcement Officer submits the information once done.
8. The Enforcement Officer gets a confirmation message containing their username.

4.11.3 Functional Requirements

REQ-4.11.1 The system shall provide a registration form to the Enforcement Officer which requires a unique username and a password.
REQ-4.11.2 The system shall provide a method of verifying that the username entered is unique and the password meets the requirements (length of 8, 1 number, 1 special character).
REQ-4.11.3 The system shall require a valid police badge ID number, by cross checking the entered badge number with a Centralized Police System.

4.12 Enforcement: Login

4.12.1 Overview, Description and Priority

This feature should be implemented in order to allow Enforcement Officers the ability to log themselves into the system as an enforcement agent. The Enforcement agent should be given the following options which correspond to features in this section: update personal information, access all parking spaces, search and browse parking spaces, access to a given Patron's information, fine tickets, and review outstanding issues. These features will be outlined in **4.13 - 4.16.**

4.12.2 Stimulus / Response sequence

1. The Enforcement Officer visits the SaaP to login.
2. The Enforcement Officer identifies self.
3. The system ensures that the identity is valid.
4. The Enforcement Officer can use all functionalities offered by the SaaP.
5. The Enforcement Officer logs out.

4.12.3 Functional Requirements

REQ-4.12.1 The system shall allow a registered Officer to login, and provide a different interface than the one specified in **4.2** with the following options: update personal information, access all parking spaces, search and browse parking spaces, access to a given Patron's information, fine tickets, and review outstanding issues. These features are more fully explained in **4.13-4.15**.
REQ-4.12.2 The system shall provide a method of viewing and responding to outstanding requests filed via the feature described in **4.10**.
REQ-4.12.3 The system shall provide a method of applying fines to Patrons who are illegally parked or who have not paid for their space.

4.13 Enforcement: Search

4.13.1 Overview, Description and Priority

This feature should be implemented in order to allow Enforcement Officers the ability to search the current spaces and view, modify, or update any selected space. The officer can manage the space as they see fit.

4.13.2 Stimulus / Response sequence

1. The Enforcement Officer logs in and decides to search.
2. The Enforcement Officer enters the Postal Code of their desired search location.
3. The system ensures the entered Postal Code is in Toronto.
4. The system provides a list of all nearby single spaced parking meter locations.
5. The Enforcement Officer can select and view the complete availability status of any numbered parking spot from the list for the week.
6. The Enforcement Officer can then choose to edit the various attributes of the parking lot such as parking usability status, fee, hours of operation, free parking hours, parking spot tag to disability parking for spots.
7. The Enforcement Officer can change the usability status when they wish to block off parking spots in an area due to a special event in the city like construction, parades, criminal investigation, accidents, bad weather conditions, etc.

8. The Enforcement Officer can change the free parking hours (eg. no fee parking allowed between 6am-8am), parking fee, fine and disability discount amounts based on City of Toronto parking bylaws.
9. The Enforcement Officer can change the hours of operation (eg. parking only allowed between 8am - 10pm) as dictated by City of Toronto parking bylaws.
10. The Enforcement Officer can remove/ add disability parking spot tags in accordance with City of Toronto parking bylaws.
11. The Enforcement Officer can persist the new settings (modification(s)) for that day (only for 1 day), set it to recur for that day (eg. every Saturday) or for all days of that week (eg. unusable due to bad weather conditions for an entire week)

4.13.3 Functional Requirements

REQ-4.13.1 The system shall provide a collection of all spaces which can be browsed.
REQ-4.13.2 The system shall provide a method of searching across the collection of spaces.
REQ-4.13.3 The system shall provide a method of filtering and refining search results based upon location, whether the spot is in use or not, and pricing.
REQ-4.13.4 The system shall allow the search result space to be modified on availability, rate, and scheduling.
REQ-4.13.5 The system shall allow the changes to be either persisted or one time only depending upon the discretion of the officer.

4.14 Enforcement: Fine Ticket

4.14.1 Overview, Description and Priority

This feature should be implemented in order to allow Enforcement Officers the ability to issue a fine to a Patron who has not paid their parking fee or has parked illegally. While patrolling, the officer may encounter such a Patron's vehicle. By logging into the system and using feature 4.13 to look up the information on the spot the officer may then issue the ticket and receive a printed copy of it.

4.14.2 Stimulus / Response sequence

1. The Enforcement Officer is logged in and notices a parking violation.
2. The Enforcement Officer enters attributes such as the violator's license plate details, time and location of issuing the ticket, officer's badge number and fine amount in accordance with City of Toronto Parking bylaws.
3. The system ensures the entered badge number corresponds to the enforcement officers.
4. The Enforcement officer gets a printable version of the fine ticket.

4.14.3 Functional Requirements

REQ-4.14.1 The system shall receive a request to find the information about a given, particular space. It shall perform operations outlined in feature **4.13.**
REQ-4.14.2 The system should provide a way to charge a fee and inform the violator of their infringement and receipt of ticket.
REQ-4.14.3 Despite the fact that this feature should be limited to an officer's account only, the system should still implement a check to ensure that an officer issued this ticket.
REQ-4.14.4 The system should cross reference the badge number of the officer issuing the ticket and the one associated with their account.

4.15 Enforcement: Review an Issue

4.15.1 Overview, Description and Priority

This feature should be implemented in order to allow Enforcement Officers the ability to view, answer, and resolve issues that have been raised by Patrons as outlined in feature **4.10**. The officer should be able to view the request along with evidence provided by the Patron and make a decision on whether the report has merit or not. If it does, the officer can then take the appropriate action by generating a ticket for the violator and processing the refund desired as is also described in feature **4.10**. A report resolution summary is then generated and the case is marked as resolved.

4.15.2 Stimulus / Response sequence

1. The Enforcement Officer is logged into the system and reviews an incident report submitted by a Patron.
2. The Enforcement Officer determines whether the violation reported was valid or not.
3. The Enforcement Officer can then:
    a. Decide to grant a refund if the reported violation is legitimate and can proceed to generate a fine ticket for the violator.
        i. The system processes a refund to the saved payment method.
    b. Add a brief description as to why the refund was not granted.
4. The Enforcement Officer submits once done.
5. The system generates a report resolution summary containing the same ID as the incident report.
6. The Patron receives a notification regarding the decision made.

4.15.3 Functional Requirements

REQ-4.15.1 The system should be able to display the current list of unresolved issues to the officer when they are logged in.
REQ-4.15.2 The system should provide a way to view the reports of a given issue, whether it be resolved or not.
REQ-4.15.3 The system should generate reports and incidents uniquely, that is, they should have a unique identifier.
REQ-4.15.4 The system should all incidents to be updated and resolved.
REQ-4.15.5 The system should allow the officer to apply a fine to the violator; relies on **4.14**
REQ-4.15.6 The system should notify the Patron of the updated status of their case.
REQ-4.15.7 The system should be able to issue refunds; relies on **4.5, 4.6**

4.16 System: Add/Delete Enforcement Officer

4.16.1 Overview, Description and Priority

This feature should be implemented in order to allow Enforcement Officers to actually be added to the system upon registration during feature **4.11**. It should also include the ability to remove an officer should they be reassigned to a different department or quit for any reason.

4.16.2 Stimulus / Response sequence

1. The system receives a registration request from the Enforcement Officer.
2. The system validates that the badge number entered by the Enforcement Officer is authorized.
3. The system approves registration and adds the Enforcement Officer it’s recorded.
4. The system sends out a confirmation.
5. The system removes the Enforcement Officer once their authorization is removed.
6. The system updates its record.

4.16.3 Functional Requirements

REQ-4.16.1 The system shall receive a registration request as outlined in **4.11**, and verify that the badge number is authorized.
REQ-4.16.2 The system should have a collection of officers which can be updated; namely, an officer can be added or removed.

4.17 System: Add/Delete Patron

4.17.1 Overview, Description and Priority

This feature should be implemented in order to allow Patrons to actually be added to the system upon registration during feature **4.1**. It should also include the ability to remove a Patron should they decide to delete their account.

4.17.2 Stimulus / Response sequence

1. The system receives a registration request from the Patron.
2. The system ensures that the Patron has a valid driver's license.
3. The system approves registration and adds the Patron to its record.
4. The system sends out a confirmation.
5. The system removes the Patron once their authorization is removed.
6. The system updates its record.

4.17.3 Functional Requirements

REQ-4.17.1 The system shall receive a registration request as outlined in **4.1**, and verify that the badge number is authorized.
REQ-4.17.2 The system should have a collection of Patrons which can be updated; namely, an officer can be added or removed.

4.18 System: Update

4.18.1 Overview, Description and Priority

This feature should be implemented in order to allow the system to ensure that all records are up to date and all entries in the database are consistent with what the current system has recorded locally. In addition to this, any local changes made to the system need to be pushed to all users of the system in real time. Priority: 10

4.18.2 Stimulus / Response sequence

1. The system ensures that the addition/deletion of the Enforcement Officer account is up-to-date in its record.
2. The system ensures that the addition/deletion of the Patron account is up-to-date in its record.
3. The system ensures that the account and activity details of the Patron and the Enforcement Officer are up-to-date.(eg. personal information).
4. The system must ensure that the status of the numbered parking spots(eg. reserved, available, not usable) are up-to-date to prevent conflict.
5. The system ensures that changes made by the Enforcement Officer with regards to free parking hours, parking fees and disability discounts are registered up-to-date.
6. The system ensures that the resolved issues are deleted from its record after a reasonable period of time.

4.18.3 Functional Requirements

REQ-4.18.1 The system shall be able to request data from, send data to, and update a central database to which all actors in the system report to.
REQ-4.18.2 The system should have a collection of Patrons and a collection of Officers which can be updated at any time.
REQ-4.18.3 The system should always perform an update after any action is taken, be it by the system or by an actor; this is to ensure consistency across all actors and also with the central database.

4.19 System: Validate

4.19.1 Overview, Description and Priority

This feature should be implemented in order to allow the system to ensure that all payment transactions are valid and approved. The system must ensure that all changes made during an update are also valid. Priority: 10

4.19.2 Stimulus / Response sequence

1. The system must check whether the payment method is either Credit, Debit or Paypal
2. The system must check if the email address is well-formed and the password is well-formed(i.e at least 8 characters, at least 1 number, at least 1 special character)
3. The system must check if the entered home address is in a valid geographical location (eg. in Canada).
4. the system must check if the badge number of the Enforcement Officer is authorized and unique.
5. The system must ensure that the Enforcement Officer and the Patron's entered login credentials match with the corresponding registered account.
6. The system must give the Enforcement Officer and Patron 3 tries to answer their security questions if the login credentials are not matching with the registered account.
7. The system must ensure if the entered postal code is in Toronto.
8. The system must ensure the Patron's selected reservation preferences do not conflict with the pre-existing reservations by providing the overlap period and suggesting different available spots.
9. The system must ensure that the type of vehicle chosen is either a car or a motorcycle.
10. The system must ensure that the payment method selected during the checkout is either debit, credit or PayPal.

4.19.3 Functional Requirements

REQ-4.19.1 The system shall be able to request the type of transaction that occurred, along with all pertinent information such as time and location.
REQ-4.19.2 The system should be able to verify information regarding the actor and also cross reference it with stored info & database entries, such as badge number of an officer or address of a Patron.
REQ-4.19.3 The system should ensure that every username is unique.
REQ-4.19.4 The system should ensure that logins are denied for an hour if the credentials are wrong and the security question is answered incorrectly three times.
REQ-4.19.5 The system should ensure that the vehicle chosen exists, and is licensed & registered.

4.20 System: Notify

4.20.1 Overview, Description and Priority

This feature should be implemented in order to allow the system to ensure that all changes pertaining to an actor or stakeholder are properly notified to them. The system must ensure that errors, parking tickets, fines, successful completion of features and updates to an account are notified. Priority: 10

4.20.2 Stimulus / Response sequence

1. The system sends an error message to the Patron or the Enforcement Officer if the entered username is not unique and suggests one, password or email address is not well-formed.
2. The system sends account creation confirmation to the Enforcement Officer containing their unique username, once added to its record.
3. The system sends account creation confirmation to the Patron containing their unique username, once added to its record.
4. The system notifies the Patron and the Enforcement Officer with the security questions if the entered login credentials are invalid.
5. The system notifies the Patron and the Enforcement officer if the Postal Code entered is outside the City of Toronto limits.
6. The system notifies the Patron if the entered payment method or information is invalid.
7. The System notifies the Patron with a payment approval on receiving a valid authorization from the external accounting system.
8. The system notifies the Patron of a payment error if it receives an invalid authentication or authorization signal from the external authorization or accounting system.
9. The system must notify the Patron with an error message if it cannot communicate with the external systems.
10. The system notifies the Patron regarding incident report generation containing a unique ID, once the report an issue request is added to its records.
11. The system must notify the Enforcement Officer of the creation of a summary report containing the unique ID corresponding to that complaint, for resolved issues.
12. The system must notify the Patron of the decision made in regards to the complaint(s) filed which are associated with a unique ID.

4.20.3 Functional Requirements

REQ-4.20.1 The system shall log all updates and changes at time of occurrence.
REQ-4.20.2 The system shall notify a Patron or Officer if their personal account information changes, or is invalid at time of entry.
REQ-4.20.3 The system shall notify a Patron of the state of their transaction, depending upon if their payment succeeds or fails.

REQ-4.20.4 The system should notify a Patron that they are parked illegally or have been fined.
REQ-4.20.5 The system should notify a Patron if they report an issue and their report has been successfully generated.
REQ-4.20.6 The system should notify a Patron if they report an issue and their report has been looked at and marked as resolved.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

REQ-5.1.1 The system should provide its users with real time information about parking spots, fines, and fees, with live updates on location timings for a parking spot.
REQ-5.1.2 The system should be able to handle at least 10,000 users every hour.
REQ-5.1.3 The system shall handle at least 1000 concurrent user requests.

5.2 Safety Requirements

REQ-5.2.1 The system should ensure that online data storage service has in-built security rules to ensure data integrity of users.
REQ-5.2.2 The system will be accessed via a browser on mobile devices or PCs. Hence, using the system should not cause any harm to the device it is on like downloading files without permission, infecting the device with virus, or any data loss on the host device.

5.3 Security Requirements

The software system needs to have a robust security and authentication system in place to prevent unauthorized users from getting access to the parking system.

All users of the system must have unique credentials. This could be done by having a unique username associated with a password entered by the user. This method will allow the system to authenticate and verify unique users, and grant access to the system. If a user cannot be identified by the credentials, they shall not be granted access to the system and will have to enter correct credentials.

REQ-5.3.1 All users of the system shall login using unique credentials (eq. username and password)
REQ-5.3.2 All login attempts shall be done in a secure manner.
REQ-5.3.3 The system shall handle all login errors and prevent any un-authenticated users to get access to the system. (Refer login uses case for details on error handling)
REQ-5.3.4 Each user shall either be trusted or not trusted.

5.4 Software Quality Attributes

REQ-5.4.1 **Portability:** The system shall be used with a wide range of devices mentioned in **Section 3,** hence it needs to be compatible with a wide range of hardware platforms running different operating systems.

REQ-5.4.2 **Usability:** The system's user interface should be intuitive, easy to use and quick to respond to any user actions. It must be what any user expects in a general sense.
5.4.2.1 The interface should contain consistent components, and when something goes wrong, it shall explain the user in a meaningful way.

5.4.2.2 It should adhere to a set of user interface design principles for maximal consistency.
5.4.2.3 It should also clearly reflect the level of access. (i.e. patron and enforcement officer)

REQ-5.4.3 Availability: The system will be used by multiple users 24 x 7. Hence, it is required to remain operational as much as possible. This can be achieved by hosting it on an online web server or a local server that runs all the time.
5.4.3.1 The system shall be available 99.5% of the time. This means that the system can have a down time of upto 48 hours for each operational year.

REQ-5.4.4 Accessibility: Since the system shall be used by a wide range of audience each with different levels of tech literacy, the system shall provide graphical aids like tooltips, dialogue boxes to users.
5.4.4.1 New users should be given a brief tutorial of the webpage once they login.
5.4.4.2 The system shall provide meaningful graphical aids like tooltips and dialogue boxes.

REQ-5.4.5 Correctness: Error correcting is very important to maintain full functionality of any system. Hence,
5.4.5.1 The system shall ensure all the necessary precautions are taken to error correct any data transactions between the host device and database.
5.4.5.2 The system shall avoid starting infinite loops that can harm the host device as well as the database.

REQ-5.4.6 Reliability: The system should be reliable for the users to utilize its functionality.
5.4.6.1 The system shall not corrupt any user data and always store data on the database effectively.
5.4.6.2 Any data stored on the host devices (if any) shall be encrypted to an acceptable level to maintain data integrity.

REQ-5.4.7 Modifiability:
5.4.7.1 The system can add/remove patrons and add/remove enforcement officers without affecting the functionalities for other users.
5.4.7.2 The system shall allow parking enforcement officers to add/remove/update parking spot information without affecting other parking spot functionalities.
5.4.7.3 The system shall allow updating user information without affecting other user data in the database.

6. Other Requirements

Internationalization
The system is intended to be used by the Toronto Parking Authority, and no other 3rd Party. Hence, Internationalization of the system is not necessary. Although, the application shall be accessible in multiple languages local and certain international to accommodate the diverse user base of the system.
Legal Requirements
Please refer to the Parking Authority of Toronto website for more information on legal requirements and restrictions for the system.
Website: https://parking.greenp.com/

Appendix A: Glossary

A1. Definitions
Patron: Patron is a registered customer that uses the system to reserve parking spots.

Enforcement Officer: Enforcement Officer is a registered user that uses the system to update parking spot information, check spot information, fine parking tickets and review issues reported by patrons.

A.3 Acronyms
GUI: Graphical User Interface
UI: User Interface
SaaP: Software as a Product
OS: Operating System
IOS: iPhone Operating System
PC: Personal Computer
SRS: System Requirements Specification
CC: Credit card