**Parking Portal**

By: Ezana Girmay

Girmay Ezana - System Proposal Part 1+2 - Parking Portal - v2.docx

Systems Design 3150

Professor: Andy Cameron

May 21, 2024



Table of Contents

[Executive Summary 4](#_Toc167218361)

[1.0 Introduction and Overview 5](#_Toc167218362)

[Problem Statement 5](#_Toc167218363)

[Project Vision and Scope 5](#_Toc167218364)

[Requirements Summary 5](#_Toc167218365)

[Stakeholders and Their Interests 5](#_Toc167218366)

[Expected Costs and Benefits 6](#_Toc167218367)

[Constraints 6](#_Toc167218368)

[Recommendation 6](#_Toc167218369)

[Document Overview 7](#_Toc167218370)

[2.0 System Initiation 8](#_Toc167218371)

[3.0 Feasibility Assessment 11](#_Toc167218372)

[Introduction 11](#_Toc167218373)

[Feasibility Analysis 11](#_Toc167218374)

[Additional Comments 12](#_Toc167218375)

[Conclusion 12](#_Toc167218376)

[4.0 Requirements Definition 13](#_Toc167218377)

[Introduction 13](#_Toc167218378)

[Functional Requirements 13](#_Toc167218379)

[Data Requirements 13](#_Toc167218380)

[Non-functional Requirements 14](#_Toc167218381)

[5.0 Requirements Model 15](#_Toc167218382)

[Introduction 15](#_Toc167218383)

[Use-Case Diagram 15](#_Toc167218384)

[Use-Case Descriptions 16](#_Toc167218385)

[6.0 System Evolution 25](#_Toc167218386)

[7.0 Conclusions and Recommendations 26](#_Toc167218387)

[Appendices 27](#_Toc167218388)

[Glossary 28](#_Toc167218389)

[Bibliography 29](#_Toc167218390)

Executive Summary

The Client Problem

Seattle Pacific University (SPU) is located in Queen Anne Hill and has around 3,000 students and staff members on campus most days (*SPU Facts | SPU*, n.d.). Because of this, it is often hard to find street parking, yet the parking lots owned by SPU are often nearly empty. Currently, SPU offers parking passes at $215 per quarter, but seeing how empty the parking lots are, it's fair to say that few students use them (*Parking | SPU*, n.d.). This is because of the high cost and that most students are unwilling to spend so much money all at once.

The Project Solution

To solve this problem, Parking Portal plans to offer short-term parking to students. Students are more likely to take advantage of the parking lot if they only have to pay in small increments throughout the quarter instead of all at once at the beginning of the quarter. Our application, made only for SPU students, will allow students to enter their license plates and pay for a parking pass by the hour or the whole day.

1.0 Introduction and Overview

## Problem Statement

Seattle Pacific University has a big campus that is divided by many streets. Because of this, parking near the buildings your classes are in is convenient and necessary to make it to class on time. While some people solve this by buying a parking pass from the school, many choose not to, mainly because of the price of the passes. This lack of students buying parking passes also results in making it harder for students to find parking on the street. This leaves SPU with parking lots all over campus that are not being utilized to their full ability. In order to solve all these problems, we plan to build a parking app where students can pay for hourly and daily parking passes right on their phones. This allows us to capitalize on the parking lots while also giving students the chance to make parking easier.

## Project Vision and Scope

To fully utilize SPU’s parking lots, we plan to build a mobile application that would allow SPU students to pay for parking on their phones. By offering parking passes for shorter periods of time and there for less money, we can remove any financial obstacles stopping students from using the parking lots. And by making payments as simple as possible we can allow students to pay for a spot in a matter of seconds, encouraging anyone in a rush to use the parking lots instead of spending time looking for parking.

## Requirements Summary

In order for the Parking Portal to attract the most amount of users, it must have certain requirements.

* The application must have an interface that allows parking passes to be bought quickly.
* To make the process as quick as possible, the application needs to save the user’s credit card information, so they do not have to re-type it every time they check out.
* We also plan to offer parking passes in both hourly increments and day passes.
* To ensure the parking lots are only being used by SPU students, the login should require an SPU email.
* To help users not get tickets, the app must send notifications when the time is almost up.

## Stakeholders and Their Interests

While Parking Portal only serves a specific group of people in a specified area, there are still a few groups of people who have an interest in it.

* The students at SPU can now park near their buildings in a way that is both convenient and affordable.
* SPU, as they could benefit finance from more students using their parking lots.
* The Office of Safety and Security at SPU, which currently oversees the parking lot, and with the introduction of our application, will probably be given the job of ensuring no one is overstaying their passes.
* The neighbors around SPU have an interest, as this could lessen the number of cars parked near their houses.

## Expected Costs and Benefits

Cost

The first and most expensive cost for Parking Portal is developing the application. Since the app is not very complex, maintenance should not cost much unless new features are introduced. We will also need to store transaction information, so we will also need to pay for storage every month. One cost that will be needed is for signs in the parking lots informing students they can use Parking Portal to pay for parking.

Benefits

Besides the financial benefits, the use of Parking Portal also has many intangible benefits for SPU. By having more students park in the lots, residents in the neighborhood will not have to deal with so many students parking on the streets. This can go a long way in improving the relationship between SPU and its neighbors. Students will also be happier being able to park closer to class, which might help with their attendance as they don’t have to walk as far and do not have to guess how long it will take them to find parking.

## Constraints

While building out this application, there are a few constraints we plan to build into the software and limitations we expect to run into.

* Students Only: This software limitation we plan to implement will limit the app to students only. This is important because we do not want random people parking in the school’s parking lots.
* Application maintenance: While the app is only running at SPU, there might not be enough users to justify having a big team after the project is done, which might hinder any updates and maintenance. To solve this, we could hire part-time employees in cycles to help implement fixes and updates.

## Recommendation

To achieve our vision for Parking Portal we require SPU and its variance departments to work with us from the time we roll out the application and as we continue to serve the students for years to come. As we roll out the application, we hope SPU’s variance departments will include us in their email updates and allow us to hang posters throughout the school. We also hope that incoming students will be informed of our services during orientation and that we are mentioned on their website. We also hope that SPU will provide us with student emails so we can send out our own advertisements and have a list of which emails we should allow to sign up for our application.

The biggest department we hope SPU will permit us to work with in the future is the Office of Safety and Security, seeing how they are in charge of the parking lots (*Parking | SPU*, n.d.). Being able to have the Office of Safety and Security be in charge of ticketing would significantly simplify the personnel we need. This transition for them will be as simple as making a portal to send tickets to the user and sending updates to their approved license plates list from our servers. To determine the exact workings of the system, we would need to see the current system that the department is using. We also hope to work with the students themselves. By working directly with the computer science students and faculty at the school, we can staff our team for any maintenance and upgrades we need. The last thing we hope SPU can help with is funding to build and deploy the application.

Document Overview

The remainder of this document is made up of 5 more major sections: System Initiation, Feasibility Assessment, Requirements Definition, Requirement Model, and System Evolution. In System Initiation, I introduce Parking Portal and the vision for it. This includes a brief introduction to the system requirements, benefits, and the competitive landscape. In the Feasibility Assessment section, I cover five different types of feasibility. After explaining the feasibility level of each, I conclude how feasible the entire project is. In the Requirements Section, I outline everything we need to do in order to have a minimum viable product (MVP); this includes both functional and non-functional requirements. Requirement Model section contains a use case diagram where I go over the users’ interactions with our system. Last, in the System Evolution I discuss future updates and plans we have in mind for Parking Portal.

2.0 System Initiation

**0. General Project Information**

|  |  |
| --- | --- |
| **Project Name:** | *Parking Portal* |
| **Two Sentence Request Description:** | *To fill and properly utilize SPU's parking through a mobile application. By allowing quick and easy options to buy day and hourly passes, students are more likely to use the parking lots.* |
| **Requested Launch Date(s):** | *Start of the Fall quarter of 2024.* |
| **Department(s) Affected By Project:** | *The Office of Safety and Security who oversees the parking lots.* |
| **Project's Customers:** | *The students at SPU.* |
| **Date Request Submitted:** | *4/16/2023* |

1. **Project Sponsor and Manager**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Sponsor** | |  | **Business Project Manager & Requestor** | |
| **Name:** | Andy Cameron |  | **Name:** | *Girmay, Ezana* |
| **Title:** | Professor |  | **Title:** | *Head Project Designer* |
| **Department:** | Computer Science - SPU |  | **Department:** | *Product development - SPU* |
| **eMail:** | acameron@spu.edu |  | **eMail:** | *girmaye@spu.edu* |

1. **Business Problem or Opportunity: The motivation for this request**

| *SPU has a lot of empty parking spots, and the quarterly price for parking is higher than most students want to pay. While parking on the street is a viable option, it gets busier during certain hours of the day and when you are running late, you might not want to park too far away. By offering day passes and hourly passes conveniently and quickly, students are more likely to use the parking spots. Which can help SPU earn more money from their parking lot rather than letting it sit empty.* |
| --- |

1. **Justification, Impact, and Importance**

**Assumptions**

|  |
| --- |
| * *Include at least two. Add more rows to each table as needed.* |
| * Students will pay for day passes when parking is hard to find on the streets. |
| * Enough students will use the parking portal to make a profitable impact. |

**Competitive Landscape / Context**

|  |
| --- |
| * *Include at least two.* |
| * Street parking is free. |
| * That most people who use their car already have quarterly parking. |

**Tangible Return, Opportunity, or Value One Time On-Going**

|  |  |  |
| --- | --- | --- |
| * *Include at least two. Estimate the best you can.* | $ 0 | $ 0 |
| * Day passes for people | $ 0 | $10 |
| * More tickets are given to people who overstay their pass time. | $100 | $ 0 |

**Intangible Benefits Impact or Value**

|  |  |
| --- | --- |
| * *Include at least two.* | $ 0 |
| * Utilize the parking spots SPU has instead of leaving them empty. | $ 0 |
| * Reduce the number of cars taking up all the spots near people's homes. | $ 0 |

1. **Product Requirements** 
   1. **Must Haves**

|  |
| --- |
| * + 1. *Include at least two. Add more rows to each table as needed.* |
| * + 1. Quick payment portal where students can even pay ahead before they get there. |
| * + 1. Multiple parking passes from the whole day to per hour so students can choose what's best for them. |

* 1. **Could Haves** (Nice to Haves)

|  |
| --- |
| * + 1. *Include at least two.* |
| * + 1. Make a parking spot reservation so you can book the spot you want beforehand. |
| * + 1. Able to buy metro cards with student discount prices. |
| * + 1. Allow students to offer and ask for carpools. |
| * + 1. Students can pay for any ticket they receive from SPU on the application. |

* 1. **Won't Haves** (Don't Do's, aka Out of Scope)

|  |
| --- |
| * + 1. *Include at least two.* |
| * + 1. Visitors can't make an account since it is only for student parking. |
| * + 1. If there are any spots available and which parking lot they are in. |

3.0 Feasibility Assessment

**Introduction**

In this section, we will look at 5 different types of feasibility and make a conclusion on the project as a whole feasibility. By analyzing how beneficial and practical the project is in each of these categories, we hope to score them on a scale of high, medium, and low, with high being the best and low being the worst.

**Feasibility Analysis**

Technical Feasibility

Technical feasibility relating to the successful design and development of our project is **high**.

* Between our application’s simplistic design and the main customers being of an age where they are proficient with technology, our user’s familiarity with similar applications should be high.
* Our developer’s familiarity should be high in all aspects required from app development and databases. One aspect where we might not be as high is processing online transactions.
* The Project size is composed of a handful of different parts, including developing the UI, storing the information in a database, and handling online transactions. A team of 5 people with experience in this should be enough for the project.

Resource Feasibility

Resource Feasibility relating to being equipped in both personnel and systems is **medium**.

* While the project has a few different computer concepts, we should be able to fill a team with people with experience in all.
* To store all the information of our customers, including their license plates, we will need to either set up a database in the school's systems or a cloud service.
* There are no new hardware and software requirements needed.

Scheduling Feasibility

Scheduling Feasibility relating to meeting deadlines and having the resources we need is **high**.

* The plan deadline is the fall quarter of next year, which is plenty of time to implement this project.
* While it is important to have the project ready by next fall quarter, even if we do not, it simply results in a slower roll-out.
* Developers should be free during the development phase, with no other projects to work on.

Organizational Feasibility

Organizational feasibility relates to the urgency, and the acceptableness of the solution is **median**.

* Similar companies offer online parking systems, but since this project is local and not too advanced, we can save money by developing it ourselves and not have to share the profits.
* While the consequence of not developing the project isn’t high, the potential profits and proper utilization of SPU resources to help students are strong motivators.

Legal and Contractual Feasibility

Legal Feasibility related to financial reporting, privacy, local and state laws, and ethics is **high**.

* Our database will store all transactions so reports for tax and other regulations should be easy.
* All information besides credit card and license plate is already stored by the school, so as long as we continue following SPU’s current security measures, we have a low risk of data breaches.
* By only storing credit card numbers on users' phones, we have a low risk of losing users' private information.
* Following the Family Educational Rights Act (FERPA) results in zero problems and can be easily done.

**Additional Comments**

While this assessment covers 5 different areas of feasibility, it's important to remember that there are also smaller areas of feasibility that shouldn’t be completely ignored. In order to predict and mitigate any future risks while working on the project, weekly assessments of all these areas will take place.

**Conclusion**

The overall feasibility assessment of the project is high. When looking at the development team, we can fill all the spots needed with individuals with the right knowledge and skills. We also have all the hardware and software needed to complete this project before set deadlines. All our legal and ethical responsibilities are easily manageable and mitigated. this project’s high feasibility and minimal risk offer all stakeholders a good opportunity to fully utilize the parking lots in SPU.

4.0 Requirements Definition

**Introduction**

This section covers the requirements needed by Parking Portal to deliver an MVP that can compete in the market. This includes functional requirements that describe what processes and services our system will have. These processes and services are what make up the core of what Parking Portal and are the fundamental jobs. Then, we lay out the Non-functional requirements that describe attributes and performance that our system must be built to.

**Functional Requirements**

1. Register Account
   1. To prevent people who are not students from accessing Parking Portal, account registration should require an SPU email.
   2. Once users have logged in, they will stay logged in until either they log out or after a designated period when we log them out and ask them to verify their email again.
2. Buying Passes
   1. When buying the pass, the application will offer two different types of passes, hourly and daily.
   2. To allow the users to quickly buy the passes they want, we will allow them to store their credit card information on the application.
   3. Users will be able to buy a pass before arriving at the parking lot by scheduling the start time of their pass.
3. Parking Pass Time
   1. After buying a pass, users can see how much time they have left on their pass on the application.
   2. When the pass is about to expire, the user will get a notification letting them know and to add more time to their pass if they want to extend it.
4. User History
   1. Users will be able to see all past transactions, including start time, end time, parking lot, and price.
5. Tickets
   1. If a ticket is issued in the parking lot and the car’s license plate matches a user’s, the ticket will be issued on the application.
   2. If the user is issued a ticket, they will be able to pay it using the application.

**Data Requirements**

1. To check emails for new accounts, the system will maintain a list of all emails that are allowed in the database.
2. The system only needs to store the user’s name to make an account.
   1. The name should be stored as a string requiring 2 characters and no symbols.
3. Lot name should be built into the app as a list of strings.
   1. The system should move the last used lot name to the top of the list so it is the first lot name displayed when the user reopens the application.
4. The system allows up to 2 different license plates and credit cards to be stored for a user.
   1. Users can only select one license plate and card when buying a pass.
   2. The user must enter the license plate before buying a pass.
5. The system should store the start time, duration, price, license plate, last 4 digits of the card, and lot name for each pass bought.
   1. User history should be updated with this information as soon as card payment is accepted.
6. All license plates with passes should be stored in a linked list starting with the closes end time.
   1. End time should be an int following epoch data format and license plates should be stored as a string (Binyamin, 2022).
   2. The linked list should be checked for expired passes once every minute.
   3. Once the user buys a pass, the license plate and end time should be sent to the database.
7. For a ticket, the system should store the license plate, time of the violation, lot name, price, if paid, and a photo.
8. The system should store a separate list of all tickets not paid for by month and generate a report every 3 months.

**Non-functional Requirements**

1. Project Oriented Requirements
   1. The application should be released before the start of the school year to gain as much traction as possible.
2. Operational Requirements
   1. Users should be able to complete purchases of a pass in less than 1 minute from opening the app to closing it.
   2. The design for the application should be as minimal as possible to help maintain the quickness of the transaction.
   3. The storage required to download the app should be as little as possible.
   4. The application should be able to run on both Android and iPhone and older operating systems for both phones.
3. Performance Requirements
   1. The credit card processes should be less than 5 seconds to keep transactions quick
   2. The application home screen should lead straight to the select hours or day pass for purchase.
4. Security Requirements
   1. To minimize Parking Portals risk, all user’s credit card information will be stored only on their phones.
   2. The application will require users to log in every 3 months to maintain security.
5. Cultural and Political Requirements

5.0 Requirements Model

**Introduction**

In this section, we provide a use case diagram for our system. This diagram shows how our users, located outside the box, interact with the system and the events, ovals in the box, that accrue with these interactions. Each event in the diagram also has a longer description that goes into more detail located after the diagram. The use cases, or events, located inside the box can also be connected with each other by either an extended line or an include line (Larman, 2002). Lines labeled extend show optional paths that could be taken if chosen to. The lines labeled include show cases that must be followed and are part of the natural order of events. In short, we hope to provide a general idea of the functionality of our application and the scope and limits we hope to provide to our users.

**Use-Case Diagram**

Caption: Use case diagram screenshot made with Lucidcharts.

A diagram of a parking portal

Description automatically generated

## Use-Case Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: View purchase history | | **ID**: 1 | **Importance**: Should have |
| **Primary Actor**: Customer | **Use Case Type**: Overview, Essential | | |
| **Supporting Actors:** one | | | |
| **Stakeholders and Interests**: | | | |
| **Brief Description**:  The customer is able to view their previous purchased passes. | | | |
| **Trigger**: Customer selects button to go to purchase History.  **Type** (mark one): \_X\_ External \_\_\_ Temporal | | | |
| **Relationships**:  **Association**: Customer  **Include**:  **Extend**:  **Generalization**: | | | |
| **The Normal Flow of Events**:   1. Click button to go to customer history page from the main page. 2. Select from a list on how you want history to be sorted by. 3. Click tab of individual bought passes to get more information. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**: | | | |
| **Special Requirements:** | | | |
| **To do/Issues:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Buy pass | | **ID**: 2 | **Importance**: Must Have |
| **Primary Actor**: Customer | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:  CEO- Interested in overall sale of tickets which leads to the success of the company. | | | |
| **Brief Description**:  The user is able to purchase a pass for their car allowing them to park at SPU’s parking lots. | | | |
| **Trigger**: Customer buys pass on application.  **Type** (mark one): \_X\_ External \_\_\_ Temporal | | | |
| **Relationships**:  **Association**: Customer  **Include**: 3.Send pass expired notification  **Extend**:  **Generalization**: | | | |
| **The Normal Flow of Events**:   1. Select pass from options. 2. Select car’s license plate and credit card you want to use for purchase. 3. Click confirm purchase to charge credit card and purchase pass. 4. Charge card for amount and notify customer that process went through, and they can start using the lot. 5. Display how much time is left for the pass on main page. | | | |
| **Sub-flows**:  S1:   1. Choose between day pass or select how many hours you want to use the lot for. 2. Change start time if user doesn’t want the pass to start immediately. | | | |
| **Alternate/Exceptional Flows**:   * 1. If no information is saved for the customer or the card is expired, prompt the user for information.   2. If credit card fails to process prompt the user for a new one. | | | |
| **Special Requirements:** | | | |
| **To do/Issues:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Send pass expired notification | | **ID**: 3 | **Importance**: Could have |
| **Primary Actor**: | **Use Case Type**: Overview, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:  Customer- Reminds them that their pass is expiring and to either move their car or add more time to their pass before they get ticketed. | | | |
| **Brief Description**:  Sends a notification to the user to inform them their pass has expired and to add more time if they need it. | | | |
| **Trigger**: Buying a pass through the application.  **Type** (mark one): \_\_\_ External \_X\_ Temporal | | | |
| **Relationships**:  **Association**:  **Include**:  **Extend**: 9.add time to current pass  **Generalization**: | | | |
| **The Normal Flow of Events**:   1. User buys pass.  * 2.Buy pass  1. When time runs out for pass notification to add more time to pass will be send out. 2. When user clicks notification, it will take them to the application where they can select how much more time they would like to add.  * 9. Add time to current pass | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:  2.1. If user does not want to add time, they can simply ignore the notification. | | | |
| **Special Requirements:** | | | |
| **To do/Issues:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Look at issued tickets | | **ID**: 4 | **Importance**: Must have |
| **Primary Actor**: Customer | **Use Case Type**: Detail, Essential | | |
| **Supporting Actors:**  Lot Manager- Allows them to quickly give tickets without having to print out the ticket and put it on the car. | | | |
| **Stakeholders and Interests**: | | | |
| **Brief Description**:  User is able to see all tickets issued to them by the parking lot managers, whether they are already paid for or still need to be paid. | | | |
| **Trigger**: Customer selects button to go to ticket history.  **Type** (mark one): \_X\_ External \_\_\_ Temporal | | | |
| **Relationships**:  **Association**: Customer  **Include**:  **Extend**: 5.Pay ticket  **Generalization**: | | | |
| **The Normal Flow of Events**:   1. Click the button to go to customer ticket history page from the main page. 2. Select from a list of how the customer wants their tickets sorted. 3. Click tab of individual ticket to get more information. | | | |
| **Sub-flows**:  S3: Click on video to get a clearer view of customer’s car in lot during the stated ticketing time. | | | |
| **Alternate/Exceptional Flows**: | | | |
| **Special Requirements:** | | | |
| **To do/Issues:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Pay ticket | | **ID**: 5 | **Importance**: Should have |
| **Primary Actor**: Customer | **Use Case Type**: Overview, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:  CEO-In charge of making sure customers pay tickets and implementing ticket collecting practices for company. | | | |
| **Brief Description**:  Customers can pay for any ticket issued to them by the lot managers. | | | |
| **Trigger**: Customer selects to pay for ticket on the application.  **Type** (mark one): \_X\_ External \_\_\_ Temporal | | | |
| **Relationships**:  **Association**:  **Include**:  **Extend**:  **Generalization**: | | | |
| **The Normal Flow of Events**:   1. Go to individual ticket page.  * 4.Look at issued tickets  1. Select pay for ticket button. 2. Select card user wishes to use. 3. Click confirm to charge credit card. | | | |
| **Sub-flows**:  S3. If customer doesn’t have a card on record, ask user to input new card information. | | | |
| **Alternate/Exceptional Flows**:   * 1. If credit card fails to process prompt the user for a new one. | | | |
| **Special Requirements:** | | | |
| **To do/Issues:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Issue ticket | | **ID**: 6 | **Importance**: Must have |
| **Primary Actor**: Lot Manager | **Use Case Type**: Overview, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**: | | | |
| **Brief Description**:  hen individuals park on SPU’s lot without having bought a ticket or a quarterly pass from the school, the Lot Manager will issue tickets to them. If the individual’s car’s license plate is in the Parking Portal app, the ticket will be issued directly to them. | | | |
| **Trigger**: Find a car with license plate that’s not on the approved license list.  **Type** (mark one): \_X\_ External \_\_\_ Temporal | | | |
| **Relationships**:  **Association**:  **Include**: 4.Look at issued tickets  **Extend**:  **Generalization**: | | | |
| **The Normal Flow of Events**:   1. The scanner on the car locates a car that does not have permission to park in SPU’s lot. 2. Lot manager double checks that the system has the right license plate number. 3. When confirmed, the system will checks if the license plate is in Parking Portal’s system. 4. If in system a ticket will automatically be filled out and sent to customer. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**: | | | |
| **Special Requirements:** | | | |
| **To do/Issues:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Scan license plates in lot | | **ID**: 7 | **Importance**: Should have |
| **Primary Actor**: Lot Manager | **Use Case Type**: Overview, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**: | | | |
| **Brief Description**:  Lot manager will drive a car with a license plate scanner through the parking lots looking for cars parked there without permission. | | | |
| **Trigger**: Drive car through parking lots with scanner attached to it.  **Type** (mark one): \_X\_ External \_\_\_ Temporal | | | |
| **Relationships**:  **Association**: Lot Manager  **Include**:  **Extend**: 6.Issue ticket  **Generalization**: | | | |
| **The Normal Flow of Events**:   1. Enter work car that contains a license plate scanner. 2. Drive through SPU’s parking lots scanning license plates. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**: | | | |
| **Special Requirements:** | | | |
| **To do/Issues:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Generate Report | | **ID**: 8 | **Importance**: Should have |
| **Primary Actor**: CEO | **Use Case Type**: Overview, Essential | | |
| **Supporting Actors:** Lot Manager | | | |
| **Stakeholders and Interests**:  Accountant- Sales reports are sent over to him to file taxes. | | | |
| **Brief Description**:  Use case enables CEO and Lot Managers to generate reports to help them summarize and understand how the business is doing. These reports can include tickets issued and the number of customers with passes. | | | |
| **Trigger**: Select generate report button.  **Type** (mark one): \_X\_ External \_\_\_ Temporal | | | |
| **Relationships**:  **Association**: CEO, Lot Manager  **Include**:  **Extend**:  **Generalization**: | | | |
| **The Normal Flow of Events**:   1. Click on generate report bottom on the employee portal’s main page. 2. Choose from the list of what you want to be included in the report. 3. Select time frame you want to generate the report from. 4. Select Download to download report. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**: | | | |
| **Special Requirements:** | | | |
| **To do/Issues:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name**: Add time to current pass | | **ID**: 9 | **Importance**: Should have |
| **Primary Actor**: Customer | **Use Case Type**: Overview, Essential | | |
| **Supporting Actors:** | | | |
| **Stakeholders and Interests**:  Lot Manger-When Customers passes don’t expire there are less tickets to issue. | | | |
| **Brief Description**:  When customer's passes have expired, they are sending a notification reminding them. If they select the notification, it will take them to the app where they can add more time to their pass. | | | |
| **Trigger**: Select add time button.  **Type** (mark one): \_X\_ External \_\_\_ Temporal | | | |
| **Relationships**:  **Association**: Customer  **Include**: 2.Buy pass  **Extend**:  **Generalization**: | | | |
| **The Normal Flow of Events**:   1. Customer selects add time to pass from main page. 2. They select how much time they want to use the lot for. 3. They click confirm and credit card is processes. 4. The end time for the pass is updated. | | | |
| **Sub-flows**: | | | |
| **Alternate/Exceptional Flows**:   * 1. If customer clicks the notification send to them, they are taken strait to step 2. * 3.Send pass expired notification   1. If credit card fails to process, prompt the user for a new one. | | | |
| **Special Requirements:** | | | |
| **To do/Issues:** | | | |

6.0 System Evolution

While we continue to make our MVP, we also have a few features and upgrades we hope to add to our system. The first is being able to allow users to buy Orca bus passes from our app. Currently, SPU offers a discount of 30% for all students but they are required to fill out an online form every 3 months and wait for the school to get a new card for them to pick up (*Student Commuting | SPU*, n.d.). By offering it right on our app, we can simplify the process and encourage more students to use it. Another feature we also hope to add is a carpool feature where students can find other students. This will allow people to post as someone looking for a carpool or someone offering to carpool, and once they find each other, they can communicate between their student emails. While this part of the application won’t be profitable, it will encourage more people to download the app, leading to more sales.

7.0 Conclusions and Recommendations

SPU has a lot of empty spaces in its parking lot, and while it offers quarterly passes to its students, we do not believe that is the best way to get people to use the parking lots. We believe that with Parking Portal, we can properly utilize this empty space for SPU. By offering our passes for shorter time periods and only when students want to, more students would be willing to spend money on parking. Our application also has a quick and easy design to encourage students who are in a rush to choose us over trying to find a parking space. Our new system also hopes to integrate our app with ticketing, which means the lot managers will not even need to get out of their cars to issue a ticket. This also gives students an easy way to pay off these tickets. In order for us to make Parking Portal a reality, we need two things. First is approval and help from SPU and its departments. The biggest of these departments is the Office of Safety and Security, which we hope will be responsible for issuing tickets. The second thing we need is funding for our application. This will help us assemble a team of software engineers, pay for signs in the parking lot, and pay for cloud storage.

# **Appendices**

# **Glossary**

FERPA – Family Educational Rights Act

MVP – Minimal viable product

SPU – Seattle Pacific University

# **Bibliography**

Binyamin, A. B. (2022, February 22). *Unix Epoch Times*. The Information Lab Nederland. <https://theinformationlab.nl/2022/02/21/unix-epoch-times/>

Larman, C. (2002). *Applying UML and patterns: An Introduction to Object-oriented Analysis and Design and the Unified Process*. Prentice Hall Professional.

*Parking | SPU*. (n.d.). <https://spu.edu/administration/office-of-safety-security/parking>

*SPU Facts | SPU*. (n.d.). <https://spu.edu/about-spu/spu-facts>

*Student commuting| SPU*. (n.d.). <https://spu.edu/administration/office-of-safety-security/student-commuters>

*User case | Lucidchart*. (n.d.). <https://lucid.app/lucidchart/ef9de283-7287-4abf-864a-01ab46d9ea93/edit?view_items=eBpC7I5ERwRD&invitationId=inv_dd3bc520-ac27-4191-a4e5-fb483d3417a2>