**Week 2 Use Case for Parking System**

For

ICT-4305 Object-Oriented Method & Program I

Luther Chikumba

University of Denver University College

September 23, 2022

Instructor: Dr. Sherri Maciosek

Table of Contents

[Use Case New Customer. 3](#_Toc114829603)

[Use Case Registering a new Customer. 3](#_Toc114829604)

[Use Case Billing. 4](#_Toc114829605)

[Alternative Flow 4A: Parking Office needs to replace the parking permit. 5](#_Toc114829606)

[Alternative flow 4B: Parking Office needs to update the customer information. 5](#_Toc114829607)

[Alternative flow 4C: A customer does not have compact car. 5](#_Toc114829608)

[Business Rules: 6](#_Toc114829609)

[Summary 7](#_Toc114829610)

[Reference: 9](#_Toc114829611)

# Use Case New Customer.

|  |  |
| --- | --- |
| Use Case 1 | New Customer |
| Actor | Customer |
| Basic Flow | Customer should purchase a parking permit and register with university parking office in order to use the parking lot. The customer can have one or more than one car that they can request for parking. Customer will be assigned a parking lot upon entry. Customer will scan their permit and get assigned a parking lot. They should keep the receipt that get print out to give back when they check out. |

# Use Case Registering a new Customer.

|  |  |
| --- | --- |
| Use Case 2 | Registering a customer |
| Actor | Parking Office |
| Basic Flow | Register a new customer and give permit. The parking office has many parking lots which a customer can use. Parking lot will scan the permits only on entry. |
| Alternative Flow 1 | If the customer has an account and lost the permit replace the parking permit |
| Alternative Flow 2 | If the customer has an account and would like to update the customer’s information. |
| Alternative Flow 3 | If the vehicle does have a compact car, then charge the regular price |

# Use Case Billing.

|  |  |
| --- | --- |
| Use Case 3 | Billing |
| Actor | Parking Office |
| Basic Flow | A bill is associated to a specific account that was created when a customer registered their vehicle. 20% off is given if the vehicle is a compact car. The bill is charged per hour from the time that the vehicle was entered until it checked ours. |

**Basic Flow: Registering a new Customer**.

|  |  |
| --- | --- |
| Description | This scenario describes the situation where the customer is at the Parking Office and registering a new account |
| 1 | Parking Office takes the customer information and register a new account |
| 2 | Customer pays the Parking Office for the permit. |
| 3 | Parking Office give the customer a new permit |
| Termination outcome | Customer has a permit |

# Alternative Flow 4A: Parking Office needs to replace the parking permit.

|  |  |
| --- | --- |
| Description | This scenario describes the situation where the customer needs to replace parking permit, |
| 4A1 | Parking Office verifies that the customer information is correct |
| 4A2 | Parking Office charges the replacement few |
| 4A3 | Parking Office gives the new parking permit |
| Termination outcome | Customer parking permit is replaced |

# Alternative flow 4B: Parking Office needs to update the customer information.

|  |  |
| --- | --- |
| Description | This scenario describes the situation where the Parking Office needs to update customer’s information |
| 4B1 | Parking Office verifies the old customer information. |
| 4B2 | Parking Office enter new information |
| Termination outcome | Customer information is updated. |

# Alternative flow 4C: A customer does not have compact car.

|  |  |
| --- | --- |
| Description | This scenario describes the situation where the customer does not have compact car |
| 4C1 | Parking Office verifies what vehicle does the customer have |
| 4C2 | Parking Office add 20% discount if the vehicle is compact |
| 4C3 | Else Parking Office charge regular price |
| Termination outcome | 20% off is added for compact vehicle |

**Post conditions:**

* Parking Office registers a new account for a customer
* Customer has parking permit
* Bill is paid for that month if not the parking permit is invalidated.

# Business Rules:

* 20% off only for compact vehicles
* All customers must have been registered at the Parking Office
* Any registered customer must have a valid parking permit to use the parking lot

# Summary

This assignment's most challenging part was figuring out the user's interactions with the customer or the Parking Officer. Initially, I started the assignment by trying to design a UML diagram, thinking that it would be easier if I thought through the Parking System pictorially. However, drawing a few drawings on paper; that process proved to be more difficult than I thought. I did spend quite a significant fragment of my time trying to use tools such as lucid to visualize what that would look like on paper if I were to draw the Parking System. After trying out different things and still hitting the wall with minimal progress, I decided to shift left, go back to the Week 1 assignment, and follow the steps and guidelines we used to do our group assignment.

What helped to get to my solutioning was reading the documentation we had from week one and the group assignment activity we had from that week. There was good documentation on creating these use cases and how to think through the process. Though I did not think it was a terrible thing, if anything, it did help me think through the whole system. My mind kept thinking about this process pragmatically, like in classes and methods needed to complete the system. However, that did not help me see the parking system's holistic view. While doing this assignment, I realized that writing the code is more manageable than actually doing the program design. It seemed to me that once you have the design, it becomes easy but more challenging to design the complete view of the program itself. No wonder I have always thought that planning is the most critical and vital for any given program or design than it is for implementation.

I think the assignment would have been a lot more straightforward if I knew when to start thinking of creating a UML diagram instead of the written one in written form. Though they both try to accomplish the same objective, I think the mind frame one uses to create UML diagrams differs from when creating written ones. Knowing how to pick up those interactions in a UML design and where they should go or how they tie to the use case is a very critical skill, and I believe it comes with time and practice. My current use cases can be better, but that perfection comes with practice and time. My old manager used to say do not let perfection get in the way of good enough if all the objectives are met. I plan on keep updating my use cases as we go along with the course.

The reasoning behind why I thought knowing these things would have been helpful are:

Then I would not have spent time trying to create UML diagrams if I had accepted. Given the time I had to think through the process and write a summary, I did not know how to.

Knowing the skills that are needed for creating a use case and their design would have helped me on how to think on this assignment.

# Reference: