

**PARKMAN**

**Chapter 1 - SRS Document**

* Merjem Sikiric

March 2021, Sarajevo

Table of Contents

[1. Abstract 3](#_Toc160470626)

[2. Document Revision History 3](#_Toc160470627)

[3. Introduction 4](#_Toc160470628)

[4. System Features and Use Cases 4](#_Toc160470629)

[4.1 System Features 5](#_Toc160470630)

[4.2 Use cases 8](#_Toc160470631)

[4.2.1 Detailed use cases 8](#_Toc160470632)

[4.3 Non-Functional Requirements 13](#_Toc160470633)

[4.4 Release plan 14](#_Toc160470634)

[4.5 References 16](#_Toc160470635)

[Document Revision History 18](#_Toc160470636)

[Objectives 18](#_Toc160470637)

[System decomposition 18](#_Toc160470638)

[Use case diagram 20](#_Toc160470639)

[Sequence diagram 21](#_Toc160470640)

[User interface 22](#_Toc160470641)

[Github Link 27](#_Toc160470642)

# Abstract

The idea of our project is that car can enter into parking and appropriate leave it without any ticket. Cameras will scan specific car plates and count how much hour they spend on parking. When customer decide to leave, on the toll plaza, that has also camera, and will once again scan the license plate and would display final bill.

# Document Revision History

Rev1.0 March 24, 2021 – initial version

Rev2.0 May 10, 2021 – We’ve changed Abstract part, Introduction part. We have worked more and in detail.

Rev3.0 May 13, 2021 – We have worked a lot on System Features

Rev3.1 May 13, 2021 – Surveillance cameras, added monitoring, must have

Rev3.2 May 13. 2021 - Added functional requirements in handle payment part

Rev3.3 May 13, 2021 – Added functional requirements in maps integration part

Rev3.4 May 13, 2021- Started working on Basic flows on system features

Rev4.0 May, 15 2021 - Changed basic flow at handle payment

Rev4.1 May 15, 2021- Changed basic flow at everything documented

Rev4.2 May 15, 2021-Added more references

Rev4.3 May 15, 2021- Added what can go wrong section at all use cases

Rev4.4 May 15, 2021- Added the solution for “what can go wrong”

Rev5.0 May 18. 2021 - All stated functionalities implemented

# Introduction

The purpose of this document is to introduce and give information about out software product, the parkMAN. Information about the user interface, software requirements and hardware of the system will be thoroughly discussed.

As the world progresses and advances, so does technology. The environment we inhabit is slowly becoming more and more digitized from the way we interact with one another to the way we function. However, as we know a car nowadays are not luxury it is something that we need on daily basis. On metropolis' parking are mostly crowded and customers are in rush to get on their destination. By setting surveillance cameras all around parking, customers can access to the fast bill for their parking place without ticket.

As I said at the beginning, the idea of our project is that car can enter into parking and leave it without any ticket. Cameras will scan car plates and count how many hours car has spent on parking. When customer decide to leave camera will scan again the license plate and would display final bill.

This level of customizability and management would allow user to not get into any kind of jam and we will have exact number of customers. Also, our goal is to save our environment with not using any kind of paper.

# System Features and Use Cases

## 4.1 System Features

When we talk about system features, I will tell you more about these in our project, such as surveillance cameras, managing entrance, counting section, reducing a jam, database management system, information and support and maps integration.

**SF1:** **Surveillance cameras:** They store the essential vehicle records, entering and exiting parking lot, like: car’s number plate, car’s entry time, car’s exit time, and provide secured monitoring.

Priority: Must have

**UR1.1:** Secured monitoring

**FR1.1.1:** **Monitoring (Must have):** Car driver will feel safe because it provides a sense of security.

**SF2: Manage entrance:** When the vehicle accesses the toll plaza, the surveillance camera is going to captures and stores the number plate images of car. Then, the toll gate will open and vehicle will access the parking lot.

**UR2.1:** Access the parking lot

**FR2.1.1:** Car accessing toll gate

**FR2.1.2:** Camera scanning car plates

**FR2.1.3:** Opening toll gate

**FR2.1.4:** **Access parking space (Must have):** The customer will park his car.

**SF3:** **Counting Section**: This would include calculation of number of hours that customer use the parking times the price per hour. It is acceptable that the payment should be done through cash.

Priority: Must have

**UR3.1: Handle payment**

**FR3.1.1.** Car leaving parking lot

**FR3.1.2.** Car coming to toll plaza

**FR3.1.3.** Camera scanning the car plates

**FR3.1.4.** **Counting section** (number of hours car spent on parking lot)

**FR3.1.5.** Display final bill

**FR3.1.6.** Car driver paying the bill

**SF4:** **Reduce a jam:** This kind of parking system allow people to save time and will be simpler to pay a bill without complication.

Priority: Must have

**UR4.1:** Reduce a jam

**FR4.1.1:** Customer will save time and pay simpler

**SF5:** **Database Management System:** At the end of the day, everything will be documented and will be into our system, from how much car we hosted during day to how much money did we earn. System database is considered as the most important functional requirement. It will help the company to store all the important data. In addition, this process will also help the company to retrieve all the stored data.

Priority: Must have

**UR5.1:** Everything documented

**FR5.1.1:** Car entering parking lot

**FR5.1.2:** Company storing all the important data

**FR5.1.3:** Car exiting parking lot

**FR5.1.4:** **Company storing all the important data (Must have)** Cameras will scan and store all important data.

**SF6: Information and Support:** If customer has questions on the parking or would like to make a reservation, he can always call our available numbers. This is a support feature that makes the experience of using the system quite good.

Priority: Should have

**UR6.1:** Get needed info

**FR6.1.1:** Have question

**FR6.1.2:** **Call support (Should have)** If customer has questions he should call our support.

**SF7: Maps Integration**

Our parking management system also helps with map management as it directs and guides users. You don’t have to worry about having to go through the stress of finding a parking space when you can see when you access the toll plaza the number of free spaces. The access to available parking space is also considered as an important functional requirement. Overall, this parking space can be accessed after the successful completion of the scanning process.

Priority: Could have

**UR7.1:** Directions and guidance

**FR7.1.1:** Successful scanning

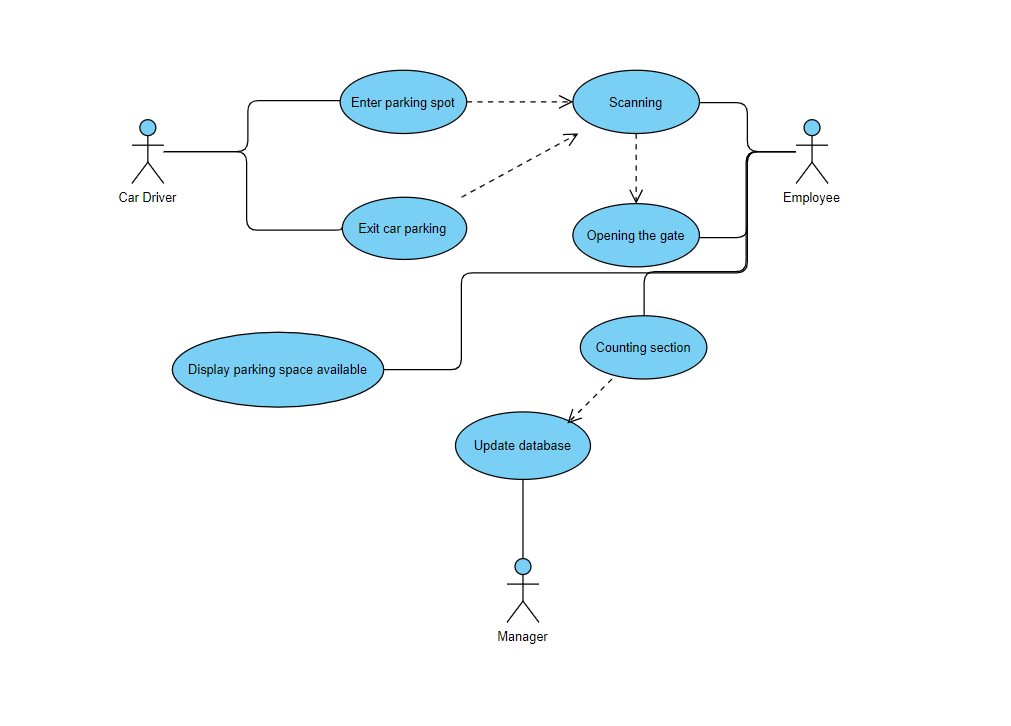
**FR7.1.2:** Entering parking lot

**FR7.1.3:** Accessing free space

|  |  |
| --- | --- |
| Primary Actor | Use Cases |
| CarDriver | Enter parking spot  Exit car parking |
| Emloyee | Scan number plate images of vehicle  Open the gate  Display parking space availability  Counting section |
| Manager | Database Management System  Info and support |

Table1.1 Primary Actors and Use Cases

## 4.2 Use cases



Picture 1. Use case diagram

### 4.2.1 Detailed use cases

**Use case ID and title: UR1.1: Secured monitoring**

**Description:** The purpose of secured monitoring is that it provides a sense of security, customer service, it deals with the security issues efficiently, it deters crimes at workplace etc. Customer will feel safe.

**Priority:** Must have

**Pre-condition:** Surveillance cameras

**Post-condition:** Safe car, happy customer.

**Basic Flow:**

1. Scanning car plates

2. Access the parking lot

3. Secured monitoring all the time

**What can go wrong:**

Secured monitoring has some technical issues and stops working. Then, technical team that we have at the toll plaza will come and fix the problem.

**Use case ID and title: UR2.1: Access the parking lot**

**Description:** When the vehicle accesses the toll plaza, the surveillance camera is going to captures and stores the number plate images of vehicle. Then, the toll gate will open and vehicle will access the parking lot. That’s the point of this topic.

**Priority:** Must have

**Pre-condition:** Toll gate opening

**Post-condition:** Exit parking lot

**Basic Flow:**

1. Access the toll plaza

2.Toll gate opening

3.Access the parking lot

**What can go wrong:**

If there is no free space, the customer cannot access the parking lot. Then, the system will inform the user that the maximum number of spaces in the parking lot has been reached

**Use case ID and title: UR3.1: Handle payment**

**Description:** When customer come to get his vehicle, he has to pay the bill. Because of that we have our system feature named counting section. This would include calculation of number of hours that customer use the parking times the price per hour. Customer would be able to pay in cash.

**Priority:** Must have

**Pre-condition:** Access parking lot

**Post-condition:** Exit parking lot

**Basic Flow:**

1. Access the toll plaza

2. Camera scanning car plates

3. Display the bill

4. Paying the bill

**What can go wrong:**

If customer doesn’t have cash to pay, he cannot exit parking lot. Then, he has to go back at parking lot, park a car and find money.

**Use case ID and title: UR3.1: Reduce a jam**

**Description:** Nowadays we all have problem with finding free parking spot. In central areas of large cities, looking for a parking space account for more than 10% of the local circulation as drivers can spend 20 minutes looking for a parking spot. This feature will help in that.

**Priority:** Must have

**Pre-condition:** Access parking lot

**Post-condition:** Increased number of customers because of efficiency

**Basic Flow:**

1. Searching for parking

2. See free space at our parking spot

3. Access the parking lot

4. Finish and want to exit

5. Exit efficiently by reducing a jam

**What can go wrong:**

Things can go wrong only if we have some technical issues with our database management and payment system and everything just stop. In that case our technical team will fix everything as soon as possible.

**Use case ID and title: UR5.1: Everything documented**

**Description:** System database is considered as the most important functional requirement. It will help the company to store all the important data. In addition, this process will also help the company to retrieve all the stored data. By the end of the day everything will be documented and will be into our system, so if user need any information, he can find it there.

**Priority:** Must have

**Pre-condition:** Access parking lot

**Post-condition:** Check information if there’s a need

**Basic Flow:**

1. Access the parking lot

2. Camera scanning car plates

3. Storing all the important data

4. Car’s access parking lot for leaving

5. Camera scanning car plates

6. Everything is stored in our database management system

**What can go wrong:**

Things can go wrong only if we have some technical issues with our database management so it couldn’t store all information. In that case our technical team will fix everything as soon as possible.

**Use case ID and title: UR6.1: Get needed info**

**Description:** For any questions customer can call the support number and get needed information. If customer has questions on the parking or would like to make a reservation, he can always call our available numbers. This is a support feature that makes the experience of using the system quite good.

**Priority:** Should have

**Pre-condition:** Interest in our parking management system

**Post-condition:** Get needed information

**Basic Flow:**

1. Have question

2. Call our support number

3. Get needed info and answers

**What can go wrong:**

Things can go wrong only if we have a lot of daily calls and we’re unavailable. In that case car driver will ask our employee near toll gate all questions.

**Use case ID and title: UR7.1: Directions and guidance**

**Description:** Our parking management system also helps with map management as it directs and guides users. You don’t have to worry about having to go through the stress of finding a parking space when you can see when you access the toll plaza the number of free spaces. The access to available parking space is also considered as an important functional requirement. Overall, this parking space can be accessed after the successful completion of the scanning process.

**Priority:** Could have

**Pre-condition:** Search for parking space

**Post-condition:** Access the parking lot

**Basic Flow:**

1. Searching for parking

2. See free space at our parking spot

3. Access the parking lot

**What can go wrong:**

Things can go wrong only if we have some technical issues and cannot display number of free spaces. In that case car driver will ask our employee are there any free spaces.

## 4.3 Non-Functional Requirements

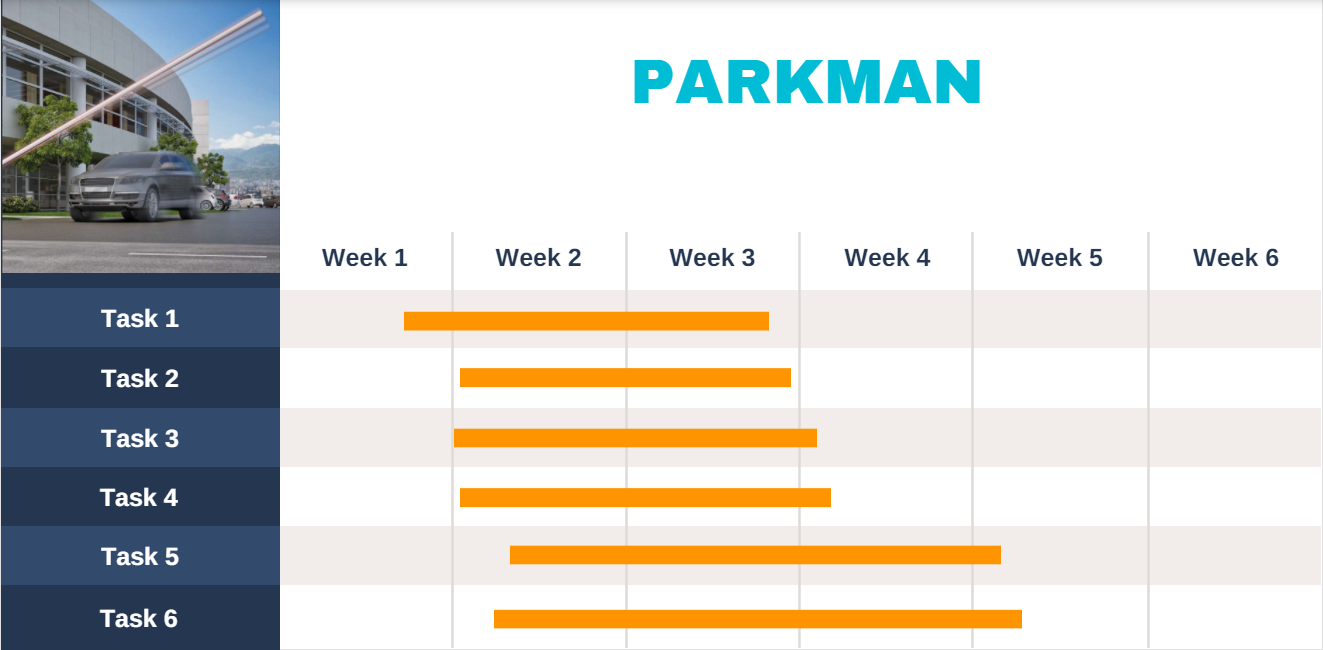
|  |  |  |
| --- | --- | --- |
| **Requirement** | **Definition** | **More details** |
| NFR1.: Availability | In terms of availability, the system will work for almost 24/7. The target customers can gain the required customer service during any time in a day. |  |
| NFR2.: Data Integrity | In terms of data integrity, this particular system will maintain consistence and accuracy of the collected and stored data. Different authentic validation and checking methods will ensure the degree of integrity and validity of the data and information. | Database management |
| NFR3.: Security | Any car will be under surveillance and will be safe. | Surveillance cameras |

Table2.1 A list of non-functional requirements

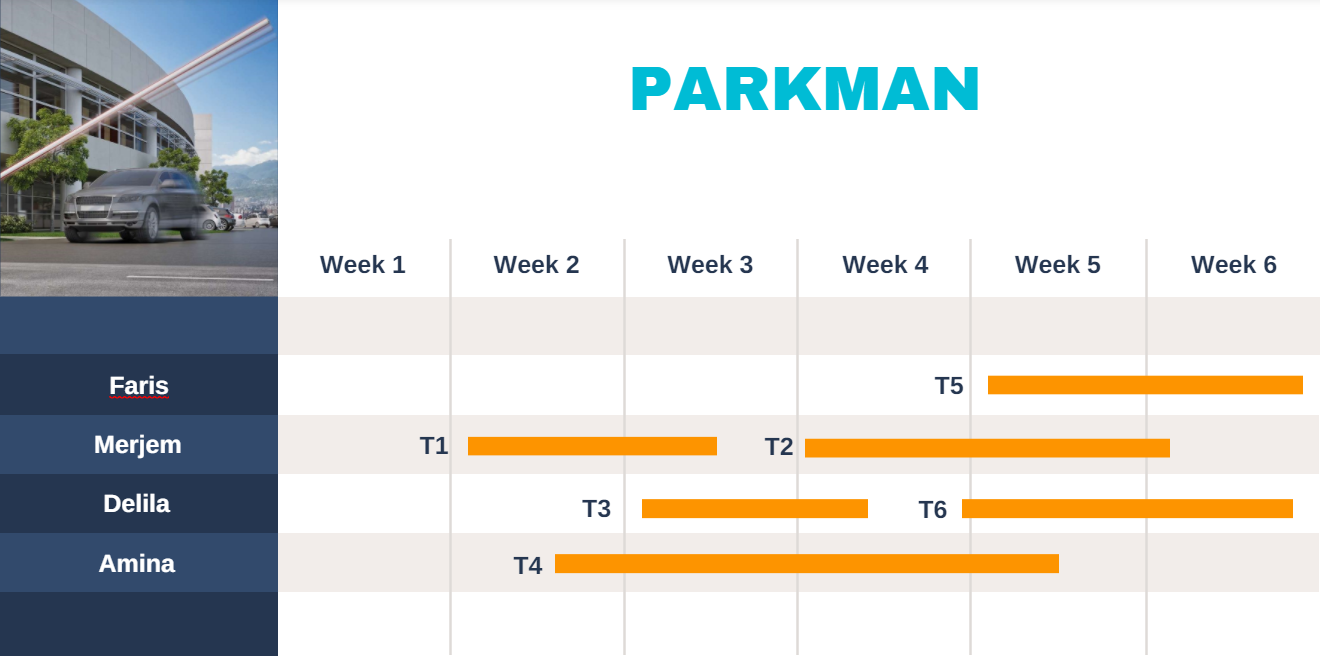
## 4.4 Release plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requirement | Duration | Increment | Priority | Dependencies | Release |
| FR1 | 1 day | 1 | High, must have | - | 1 |
| FR2 | 2 days | 2 | High, must have | - | 1 |
| FR3 | 2 days | 3 | High must have | - | 1 |
| FR4 | 1 day | 2 | High, must have | - | 1 |
| FR5 | 1 day | 2 | High, must have | - | 1 |
| FR6 | 3 days | 3 | Medium, should have | - | 1 |
| FR7 | 1 day | 1 | Low, could have | - | 1 |

Table3.1 A list of functional requirement-release plan



Picture 2. Timeline graph - Gantt chart



Picture 3. Staff allocation chart - Gantt chart

Some of the libraries that were used in the project are: "tkinter", "scikit-learn", "php", and "mysql".

"tkinter" is a package used for creating graphical interfaces, it has cross platform support and that makes it a great tool.

"scikit-learn" is a machine learning library, in this project it was used to extract licence plate information from car images.

"mysql" is a relational database management system.

"php" is a scripting language used for serverside scripts of web application.

## 4.5 References

Dr. Jean-Paul Rodrigue of Hofstra University’s Department of Global Studies and Geography - <https://www.parking-net.com/parking-news/skyline-parking-ag/traffic-congestion>

<https://parkoffice.io/blog/parking-management-system>

https://www.versionx.in/parking-management-system

**PARKMAN**

Chapter 2 - Design document

* Merjem Sikiric

April 2021, Sarajevo

# Document Revision History

Rev1.0 April 23, 2021 – initial version

Rev2.0 May 13, 2021 – Deleted second diagram, there is no need for it

Rev2.1 May 13, 2021 – Added return statement in sequence diagram

# Objectives

The purpose of this design document is that reader can see structure of our project (software structure), software components, interfaces and data necessary for the implementation phase.

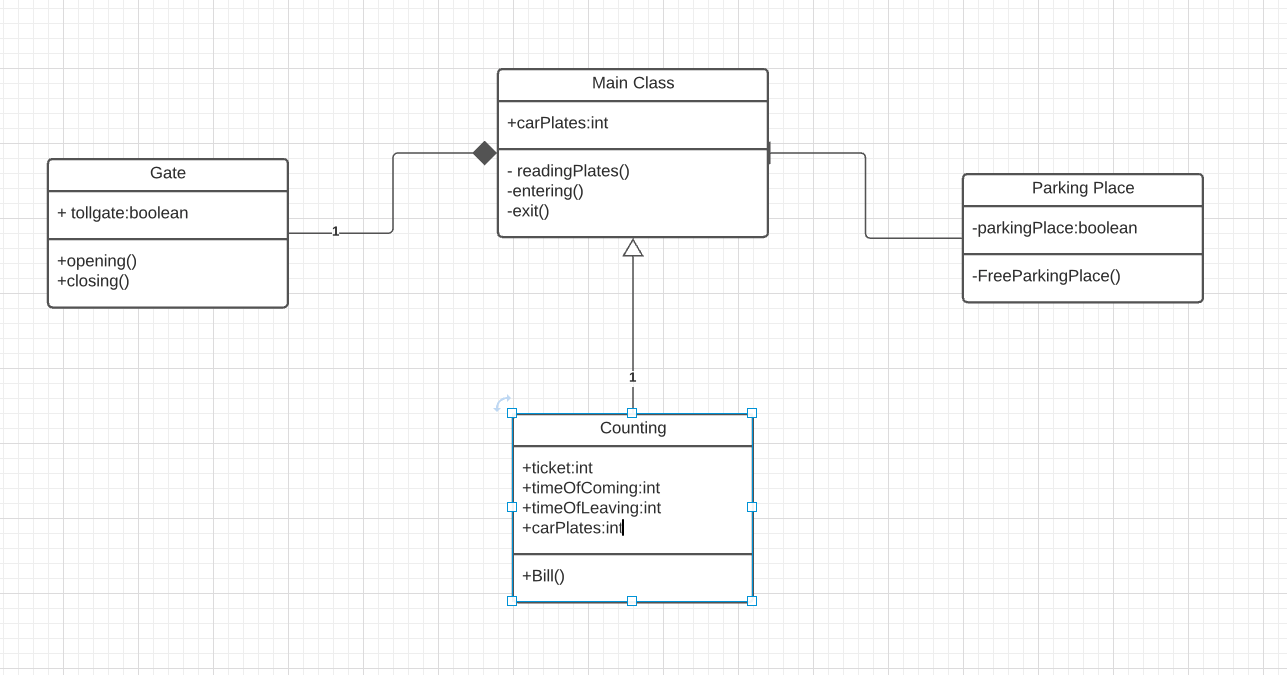
The idea of our project is that car can enter into parking and appropriate leave it without any ticket. Parkingain the city can be anaattractive subject because 30% of traffic\_congestion is caused by drivers circling to find an\_available parking for their vehicles. Thus, the vehicle needs to have a decision system that allows it to find the optimal parking in a short time. So, our project is going to work in this way: Cameras will scan specific car plates and count how much hour they spend on parking. When customer decide to leave, on the toll plaza, that has also camera, and will once again scan the license plate and would display final bill.

This level of customizability and management would allow user to not get into any kind of jam and we will have exact number of customers. Also, our goal is to save our environment with not using any kind of paper so we reduce trash.

# System decomposition

Our project has a few methods, method for scanning car plates, method for opening gate, method for counting how many hours the car was on the parking, etc. You can see better in following class diagram. We have Main Class ant three other class: Gate, Counting, Parking Place. Counting inheritance all attributes from the Main Class so that he can count the bill. Gate class is for the opening and closing the gate. Parking Pace class is here to make sure that has free parking places.

Class diagram

 Figure 1: Class diagram

# Use case diagram

Diagram

Description automatically generated

Figure 2. Use case diagram

# Sequence diagram

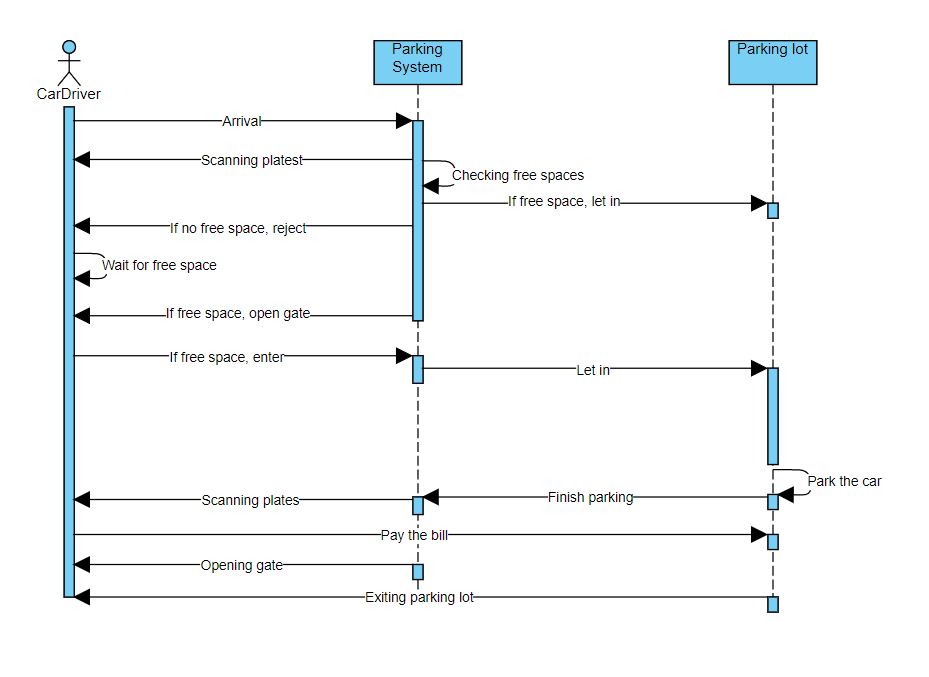


Figure 3. Sequence diagram

(For some reason we cannot add more stuff in this diagram but what we wanted to change is return statement, we wanted to add return statement on “pay the bill” - “bill payed”, “open gate” – “gate opened”, “scanning plates”- “plates scanned”)

# User interface



Figure 5. Our project’s logo

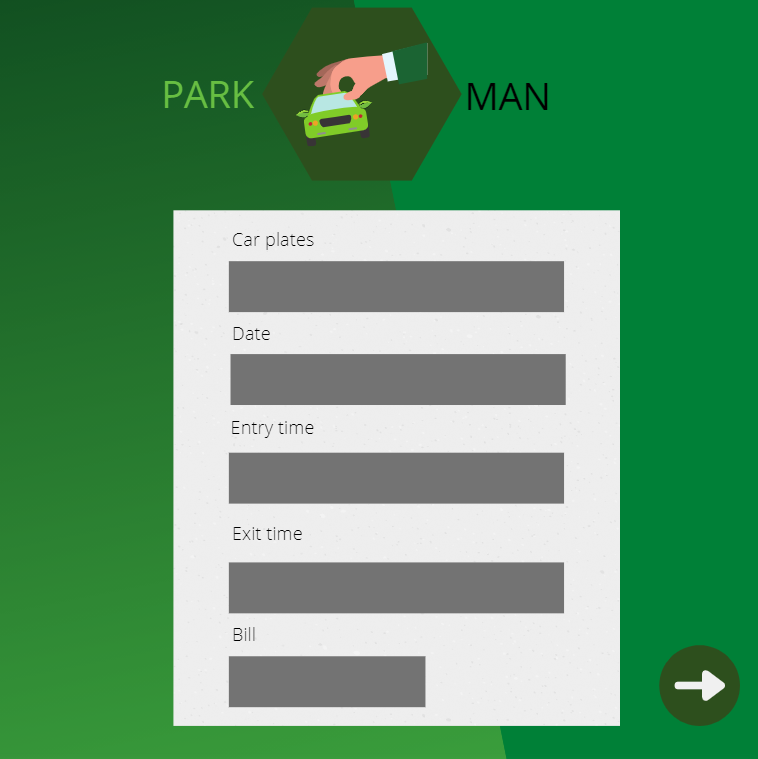


Figure 6. Sample for needed information about car’s plate



Figure 7. Scanned information about car’s plate



Figure 8. The end

# Github Link

https://github.com/SikiricM/parkman