**Project Documentation**

**Parking Management System**

**Developed by:**

[Enriko Hyka](mailto:ehyka21@epoka.edu.al)

[Enesi Musiçi](mailto:emusici21@epoka.edu.al)

[Enad Draçini](mailto:edracini21@epoka.edu.al)

[Erik Veshi](mailto:eveshi21@epoka.edu.al)

[Klajdi Gazidedja](mailto:kgazidedja21@epoka.edu.al)

[Serxhio Elezi](mailto:selezi21@epoka.edu.al)

[Isgen Ternova](mailto:iternova21@epoka.edu.al)

**Received By:** [Igli Draci](mailto:idraci@epoka.edu.al), [Shkumbin Fida](mailto:shfida@epoka.edu.al)

**1.Exclusive summary**

#### **1.1 Project Overview**

Moving on, this Vehicle Parking Management System project in PHP focuses mainly on keeping track of vehicle’s parking. Also, the system displays all the vehicle’s entry and outgoing records. In addition, the system allows adding parking slots too. This project only contains an admin panel. In an overview of this web application, an admin has all control over the system. He/she helps to maintain the flow of the system. An admin can simply add vehicle Categories by providing a category. After the management of vehicle Categories, the admin can simply insert the vehicle’s entry. For this, he/she has to enter the vehicle’s number, model name, category, and owner’s information.

#### **1.2 Project purpose**

After setting up the vehicle’s parking, now the user can manage outgoing vehicles. The system lists out all the entry and outgoing vehicle records. Here, the user can simply take an action for checking out the parking. In order to exit a vehicle’s parking, the user has to enter the total parking charge with remarks. Once a vehicle exits the parking, those records can be found under the outgoing vehicle records. Here, the system generates invoice receipts for each and every vehicle. The user can view details of each available vehicle. Also, the user can view and print out the parking receipt of each. Each record consists of parking number, total charge, vehicle registration number, owner details, and more.

**2. Product and service description**

#### **Reports, Earning Collections**

On the other hand, an admin can list out the reports between dates. These reports help out to state vehicle’s parking records between the selected dates. The steps are simple, the user only needs to select from and to date. And after that, the system displays it all between those mentioned dates where the user can also view their details. Additionally, the admin can view total earnings to date. And also, the system displays the current date and yesterdays’ total earning. He/she can search out the vehicle’s record using the vehicle’s registration number from the sidebar. Besides, an admin can have an overview records of the total vehicle’s entry with a number of in and out vehicles and total parking within a 24 hours span time. The system represents overall records using graphical charts like piechart to present a summary of records.

**2.1 Update Account Information, Password**

And finally, this whole section will be about additional features in this system. This system allows updating the user’s profile where he/she can update his/her name and contact details. Also, an admin can change the password by entering the current password with a new and confirmation password. The last feature is about company information. The system allows the user to update company settings. It includes of company’s name, the company’s email address, website URL, and company’s address.

a clean and simple dashboard is presented using *Admin Panel* with simple color combinations for greater user experience while using this Online Vehicle Parking Management System Project in PHP MySQL. For its UI elements, a free open-source CSS framework; Bootstrap is on board with some Vanilla CSS too. Presenting a new Online Parking Management System Project in PHP MySQL which includes an admin panel that contains all the essential features to follow up, and a knowledgeable resource for learning purposes.

**2.2 User's Characteristics**

There are 3 types of users that will interact with the system:

1. **Parking Guard**

* User can login
* User can recover forgotten password
* User can change password
* User can read his personal information
* User can change settings information
* User can change his personal information
* User can view total number of parked vehicles
* User can add/edit/delete records of Vehicle Categories
* User can add/edit/delete records of vehicles entry
* User can check out the vehicle
* User can print in a PDF the parking Ticket
* User can view parking reports
* User can logout

1. **Receptionist**

* User can login
* User can recover forgotten password
* User can change password
* User can read his personal information
* User can change settings information
* User can change his personal information
* User can view total number of parked vehicles
* User can add/edit/delete records of Vehicle Categories
* User can add/edit/delete records of vehicles entry
* User can check out the vehicle
* User can print in a PDF the parking Ticket
* User can view parking reports
* User can logout

1. **Parking Analysts**

* User can login
* User can recover forgotten password
* User can change password
* User can read his personal information
* User can change settings information
* User can change his personal information
* User can view total number of parked vehicles
* User can add/edit/delete records of Vehicle Categories
* User can add/edit/delete records of vehicles entry
* User can check out the vehicle
* User can print in a PDF the parking Ticket
* User can view parking reports
* User can logout

**2.3 Assumption**

* It is assumed that some actions performed behind the scenes are performed regularly according to law.
* It is assumed that the profile of the receptionist is created initially by the administrator of the system and nobody else can add, delete or change information of the receptionist. If the receptionist changes, then the administrator of the system has to be contacted.
* It is assumed that while adding a new vehicle entries or changing his personal information, the receptionist should verify all the needed documents.

**2.4 Assumption**

Certainly! Here's a more detailed description:

"The efficient operation of the parking lot's management system relies on the availability and proficiency of its staff. Both the receptionist and every parking guard are required to have access to a personal computer throughout their working hours. This computer is essential for accessing a specialized web application designed to streamline parking operations.

Proficiency in using this web application is paramount for all staff members. They must be well-versed in its functionalities and understand their individual roles and responsibilities within the system. This knowledge ensures that staff can perform their duties effectively, avoid unnecessary overlaps, and respect the privacy of the parking lot's patrons.

Moreover, the proper utilization of the web application not only enhances operational efficiency but also contributes to the overall performance of the system. By utilizing the application correctly, staff can maximize its benefits and improve the overall management of the parking lot.

In addition to staff proficiency, the project's success is also contingent on a stable Internet connection. Given that the application relies on web-based technologies, a consistent and reliable Internet connection is crucial for accessing and retrieving data from the central database. Any interruption in the Internet connection could severely impact the functionality of the application and disrupt parking lot operations.

Therefore, ensuring that both staff members have access to personal computers and are proficient in using the web application, as well as maintaining a stable Internet connection, are critical factors in the successful implementation and operation of the parking lot management system."

**2.5 Dependencies**

**Parking Guard:**

Must have access to the system during working hours to perform their duties efficiently.

Responsibilities may include managing parking spaces, guiding vehicles, and ensuring smooth traffic flow.

**Parking Analyst:**

Should be able to access the web application to analyze parking data and make informed decisions.

Responsibilities may include monitoring parking lot occupancy, predicting demand, and optimizing parking space utilization.

**Admin:**

Responsible for managing the overall operation of the parking lot management system.

Must ensure that all users (parking guards and analysts) have the necessary access and training to use the web application effectively.

Dependencies:

No new parking space information can be added or existing information deleted without the admin's approval.

Parking analysts cannot access certain data or perform specific tasks without permission from the admin.

The system's functionality relies on a stable Internet connection to fetch data from the database.

Efficient use of the web application by all users (guards, analysts, and admin) is crucial for optimal performance of the system.

**3. Requirements**

| **Req#** | **Requirements** | **Comments** | **Priority** | **Date** | **SME**  **Reviewed /**  **Approved** |
| --- | --- | --- | --- | --- | --- |
| **FR\_1** | The system shall provide a login functionality for users to access the system. | It's good practice to include user authentication in your system. However, for security reasons, it's important to clarify whether the system will have different types of users (e.g., admins, parking guards, analysts) and how their access levels will be managed. | 3 | 25/03/2024 | Enriko Hyka  Erik Veshi |
| **FR\_2** | The system shall allow users to recover their forgotten password via email verification. | Implementing a password recovery feature with email verification is a standard practice to ensure the security of user accounts. It's important to include this functionality to provide a seamless and secure experience for users who may forget their passwords. | 1 | 25/03/2024 | Enriko Hyka  Erik Veshi |
| **FR\_3** | The system shall allow users to change their password. | Allowing users to change their password is another essential feature for ensuring account security and user convenience. This feature should be implemented securely, requiring users to verify their identity before allowing them to change their password. | 1 | 25/03/24 | Serxhio Elezi/ Klajdi Gazidedja |
| **FR\_4** | The system shall allow users to read their personal information. | Providing users with access to their personal information is crucial for transparency and user control. It's important to ensure that this feature complies with privacy regulations and that users can only access their own information. | 1 | 25/03/24 | Serxhio Elezi/ Klajdi Gazidedja |
| **FR\_5** | The system shall allow users to change their personal information. | Allowing users to change their personal information is a useful feature that can improve user experience. However, it's important to implement this feature securely, ensuring that users can only modify their own information and that any changes are validated and logged appropriately. | 1 | 25/03/24 | [Enesi Musiçi](mailto:emusici21@epoka.edu.al) / [Enad Draçini](mailto:edracini21@epoka.edu.al) |
| **FR\_6** | The system shall allow users to change their settings information. | Allowing users to change their settings information can enhance their experience with the system. This feature should include options such as notification preferences, language settings, and any other customizable aspects of the user experience. | 1 | 25/03/24 | [Enesi Musiçi](mailto:emusici21@epoka.edu.al)/[Enad Draçini](mailto:edracini21@epoka.edu.al) |
| **FR\_7** | The system shall display the total number of parked vehicles. | Displaying the total number of parked vehicles is a helpful feature for users and administrators to quickly see the current status of the parking lot. This information can be displayed prominently on the dashboard or a dedicated section of the system. | 1 | 25/03/24 | Enriko Hyka / Isgen Tetova |
| **FR\_8** | The system shall allow users to add, edit, and delete records of Vehicle Categories. | Allowing users to manage vehicle categories is important for maintaining an organized and efficient parking system. Users should be able to add new categories, edit existing ones, and delete categories that are no longer needed. | 1 | 25/03/24 | Enriko Hyka / Isgen Tetova |
| **FR\_9** | The system shall allow users to add, edit, and delete records of vehicles' entry. | Enabling users to manage vehicle entry records is crucial for keeping track of vehicles in the parking lot. Users should be able to add new entry records, edit existing ones (e.g., update entry time), and delete records when necessary. | 1 | 25/03/24 | Klajdi Gazidedja / Erik Veshi / [Serxhio Elezi](mailto:selezi21@epoka.edu.al) |
| **FR\_10** | The system shall allow users to check out the vehicle. | Allowing users to check out their vehicle is a key feature for a parking management system. This should include functionality to mark a vehicle as checked out, calculate the parking duration and fee (if applicable), and update the vehicle's status to indicate it has left the parking lot. | 1 | 25/03/24 | [Erik Veshi](mailto:eveshi21@epoka.edu.al)/ Isgen Tetova |
| **FR\_11** | The system shall allow users to print a parking ticket in PDF format. | Enabling users to print a parking ticket in PDF format is a useful feature for providing them with a physical record of their parking transaction. This can be implemented by generating a PDF document that includes relevant details such as the vehicle's information, entry time, parking duration, and any fees incurred. | 1 | 25/03/24 | [Enesi Musiçi](mailto:emusici21@epoka.edu.al) / [Enesi Musiçi](mailto:emusici21@epoka.edu.al) |
| **FR\_12** | The system shall allow users to view parking reports. | Enabling users to view parking reports can provide valuable insights into parking lot usage and revenue. Reports could include data such as the total number of vehicles parked, revenue generated, peak parking times, and other relevant metrics. | 1 | 25/03/24 | Serxhio Elezi |
| **FR\_13** | The system shall allow users to logout. | Implementing a logout feature is essential for user security and privacy. It allows users to safely end their session and ensures that unauthorized access to their account is prevented. | 1 | 25/03/24 | Enriko Hyka |

**3.2 Non-Functional requirements**

**3.2.1 Product requirements**

**3.2.1.1**

* **Login**

Login page will have 2 labels and 2 inputs :

* Username
* Password

In the input boxes we need to enter the correct credential to log in to the “Home” page.

There will be a link below the input and label if the user forgets the password.

The link will send the user to the “Password Recovery” page.

Below the link there will be a login button.

If the credentials are correct the user will log in to the “home” page.

If credentials are incorrect the user would have to try again.

* **Password Recovery**

In this page there will have 2 labels and 2 inputs :

* Password Recovery
* Email ID

In the password recovery input box you need to enter the new password.

In the Email Id input box you need to enter the email that is registered in the system.

Below there will be 2 buttons:

* Proceed
* Back

The proceed button will bring you to create a new password for recovery if the credentials are correct, if the credentials are invalid will pop up an error message.

The back button will bring you to the “login” page.

* **Dashboard**

**3.2.1.2 Learnability**

* Receptionists and parking guards should be able to master using the system within a few hours.
* The system should be intuitive for patients, requiring no specific training.
* This documentation serves as a user guide for parking guards, parking analysts, and the admin.
* Specific and detailed error messages will be shown to users in case of errors, helping them understand what went wrong.
* Users are responsible for their own actions.

**3.2.1.3 Accessibility**

* Each user is provided with a username and password upon registration.
* The system can be accessed by parking guards, parking analysts, and admin.
* Patients of the parking lot can also access the system.
* Registered users can access the system at any time and from any place.

**3.2.1.4 Efficiency**

* All operations will be fast and in real time.
* Once users have learned the system, they will be able to perform each operation within minutes.

**3.2.1.5 Memorability**

* The system is intuitive, so users can easily remember how to use it.
* Returning users will be able to re-establish proficiency within the first hour of use.
* User interfaces are designed to be easy to understand, using pictures, icons, buttons, and descriptions.

**3.2.1.6 Errors**

* The error rate is lower than the current error rate.
* Double-check procedures are applied for entering time-sensitive data, requiring user confirmation.
* Errors can be edited and corrected immediately.

**3.2.1.7 Satisfaction**

* The system is user-friendly and easy to use.

**3.2.1.8 Capacity**

* The application covers all the necessities of the parking lot.
* It works in real time, reflecting changes immediately to other users based on their clearance levels.
* To increase capacity and reduce delays, users store changes on their computers before sending them to the database.
* The application is stored on a web server and does not occupy a large space.

**3.2.2 Organization Requirements**

**3.2.2.1 Availability**

* The application must be available 24 hours a day, 7 days a week.
* Users should be able to access the system from any PC connected to the Internet.
* Downtime should be avoided to ensure continuous availability of the system.

**3.2.2.2 Latency**

* The system’s latency should be optimized for quick response times.
* Factors affecting latency include Internet connection strength, database size, and algorithm efficiency for fetching data.

**3.2.2.3 Monitoring**

* The system will be built to be secure and reliable, with user-friendly interfaces to prevent crashes.
* Periodic reports will be generated by the maintenance group to detect problems and improve the system.
* Field validation will be implemented to ensure data accuracy, with informative error messages for users.

**3.2.2.4 Maintenance**

* The system will use MySQL for the database and Apache server for hosting.
* It will be developed in modules to facilitate easy extension and maintenance.
* Restarting the system should resolve most malfunctions; if not, the maintenance department should be contacted.
* Users will be informed of any software updates or bug corrections.

**3.2.2.5 Operations**

* Users can log in and access their personal information at any time.
* Only authorized personnel can access the information entered into the system.
* Admins can add vehicle categories, insert vehicle entries, and manage the flow of the system.

**3.2.2.6 Standards Compliance**

* The system will comply with relevant coding standards and best practices for PHP development.
* Security standards, such as data encryption and user authentication, will be followed to protect user information.

**3.2.2.7 Portability**

* The system should be designed to be easily portable to different environments, allowing for deployment on various web hosting platforms.

**3.2.3 External Requirements**

**3.2.3.1 Data Protection**

* Personal information is secured with hashed passwords.
* Parking guards, parking analysts, and admins are required to handle personal information with care.

**3.2.3.2 Protection**

* Encryption of sensitive information such as passwords using hashing methods.
* Tracking user activity to hold users accountable in case of problems.
* Receptionists (admins) are responsible for the authenticity of the user's personal data they enter into the system.
* Validation of data for special characters and specific conditions before inserting or updating in the database.
* Confirmation of most actions by users through a pop-up window to prevent accidental actions.
* Restriction of user access to only their relevant information.

**3.2.3.3 Authorization and Authentication**

* User authentication using username, password, and reCaptcha.
* Authorization based on user type, ensuring each user accesses only relevant information.
* Use of sessions for the currently logged-in user.
* Display of a message if a user tries to log in with incorrect credentials.

**3.3 Domain Requirements**

This web application operates within the domain of parking lot management. The main purpose is to streamline the management of vehicle parking, focusing on recording entry and exit records, and facilitating the addition of parking slots. However, the utmost priority in this application is the security of data. As vehicle and user information are sensitive, access should be restricted to users with authorized accounts. This application is intended for use in a specific parking lot with a private network and is designed to operate independently without the need to communicate with any other system.

**4.Software Design / Diagrams**

**4.1 Requirements Analysis**

**4.1.1 User Scenarios List**

| **Nr** | **Name** | **Description** |
| --- | --- | --- |
| **US\_01** | User logs in | Users: Gate Opener, Receptionist and Analyst login using username and password |
| **US\_02** | User Recovers Password | Users can recover a forgotten password by providing their email and following the password recovery process. |
| **US\_03** | User Changes Password | Users can change their password by logging in and accessing the settings or profile section where they can update their password. |
| **US\_04** | User Reads Personal Information | Users can view their personal information such as name, email, and contact details. |
| **US\_05** | User Changes Settings Information | Users can modify their settings information, such as notification preferences or display settings. |
| **US\_06** | User Changes Personal Information | Users can update their personal information, such as name, email, and contact details. |
| **US\_07** | User Views Total Number of Parked Vehicles | Users can view the total number of vehicles currently parked in the parking lot. |
| **US\_08** | User Manages Vehicle Categories | Users can add, edit, or delete vehicle categories, such as car, motorcycle, or bicycle. |
| **US\_09** | User Manages Vehicle Entry Records | Users can add, edit, or delete records of vehicles entering the parking lot, including vehicle number, type, and owner information. |
| **US\_10** | User Checks Out Vehicle | Users can check out a vehicle from the parking lot, marking it as no longer parked. |
| **US\_11** | User Prints Parking Ticket | Users can print a parking ticket in PDF format for a vehicle parked in the parking lot. |
| **US\_12** | User Views Parking Reports | Users can view reports related to parking lot activity, such as total vehicles parked, revenue generated, or parking duration. |
| **US\_13** | User Logs Out | Users can log out of the system, ending their current session. |

**4.1.1.2 User Scenarios Extended**

| **Nr** | **Name** | **Description** |
| --- | --- | --- |
| **US\_01** | User Login | - User selects their user type (Parking Guard, Receptionist, Analyst).  - User is redirected to the login page.  - User enters their username and password.  - User presses the login button.  - If the data is correct, the user is redirected to their profile page.  - If the data is incorrect, an error message is shown, and the user repeats the process. |
| **US\_02** | User Recovers Forgotten Password | - User clicks on the "Forgot Password" link.  - User enters their email address.  - User receives an email with a password reset link.  - User clicks on the reset link and enters a new password.  - User's password is successfully reset. |
| **US\_03** | User Changes Password | - User navigates to the settings or profile section.  - User selects the option to change their password.  - User enters their current password and new password.  - User confirms the new password.  - User saves the changes.  - Password is successfully changed. |
| **US\_04** | User Reads Personal Information | - User navigates to the profile section.  - User views their personal information such as name, email, and contact details. |
| **US\_05** | User Changes Settings Information | - User navigates to the settings section.  - User modifies their settings, such as notification preferences or display settings.  - User saves the changes. |
| **US\_06** | User Changes Personal Information | - User navigates to the profile section.  - User selects the option to edit their personal information.  - User updates their information, such as name, email, or contact details.  - User saves the changes. |
| **US\_07** | User Views Total Number of Parked Vehicles | - User navigates to the dashboard or parking overview section.  - User views the total number of vehicles currently parked in the lot. |
| **US\_08** | User Manages Vehicle Categories | - User navigates to the vehicle categories management section.  - User adds a new vehicle category or selects an existing category to edit or delete.  - User saves the changes. |
| **US\_09** | User Manages Vehicle Entry Records | - User navigates to the vehicle entry records management section.  - User adds a new entry record for a vehicle or selects an existing record to edit or delete.  - User saves the changes. |
| **US\_10** | User Checks Out Vehicle | - User selects the option to check out a vehicle.  - User enters the vehicle's information, such as license plate number or entry record ID.  - User confirms the checkout.  - Vehicle is marked as checked out and removed from the parking lot. |
| **US\_11** | User Prints Parking Ticket | - User selects the option to print a parking ticket.  - User chooses the vehicle for which to generate the ticket.  - User selects the print option.  - Parking ticket is generated in PDF format and ready for printing. |
| **US\_12** | User Views Parking Reports | - User navigates to the reports section.  - User selects the type of report to view, such as total vehicles parked, revenue generated, or parking duration.  - User generates and views the selected report. |
| **US\_13** | User Logs Out | - User selects the logout option.  - User is logged out of the system and redirected to the login page. |

**4.1.2 Users Cases**

| **Name** | User Logs in |
| --- | --- |
| **Summary** | User enters personal information to access his account. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User gains access to his account after typing his correct username and password. |
| **Precondition** | User must have an active account and should choose one of the alternatives (Parking Guard/Receptionist/Analyst) before logging in. |
| **Alternatives** | The user can access only one account at a time and can have only one role (Parking Guard/Receptionist/Analyst), but a Parking Guard, Receptionist, or Analyst can have two accounts because they can also be Admins. |
| **Post Condition** | User is logged on to his account. |

***UC\_01-US\_1- User logs in***

| **Name** | User Recovers Forgotten Password |
| --- | --- |
| **Summary** | User can recover a forgotten password by providing their email address and following the password recovery process. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User clicks on the "Forgot Password" link and enters their email address. They receive an email with a password reset link. After clicking on the reset link, they are prompted to enter a new password. Once the new password is confirmed, the user's password is successfully reset. |
| **Precondition** | User must have an active account and must have forgotten their password. |
| **Alternatives** | If the user does not receive the password reset email, they can request another reset email. |
| **Post Condition** | User's password is successfully reset. |

***UC\_02 -US\_2- User Recovers Forgotten Password***

| **Name** | User Changes Password |
| --- | --- |
| **Summary** | User can change their password by navigating to the settings or profile section and updating their password.. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User navigates to the settings or profile section, selects the option to change their password, and enters their current password along with the new password. After confirming the new password, the user saves the changes and their password is successfully updated. |
| **Precondition** | User must have an active account and be logged in. |
| **Alternatives** | If the user forgets their current password, they can use the password recovery process to reset it. |
| **Post Condition** | User's password is successfully changed. |

***UC\_03 -US\_3- User Changes Password***

| **Name** | User Reads Personal Information |
| --- | --- |
| **Summary** | User can view their personal information such as name, email, and contact details. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User navigates to the profile section where they can view their personal information. This information may include their name, email address, contact number, and any other relevant details. |
| **Precondition** | User must have an active account and be logged in. |
| **Alternatives** | If the user needs to update their personal information, they can do so in the profile or settings section. |
| **Post Condition** | User has viewed their personal information. |

***UC\_04 -US\_4- User Reads Personal Information***

| **Name** | User Changes Settings Information |
| --- | --- |
| **Summary** | User can modify their settings information, such as notification preferences or display settings. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User navigates to the settings section where they can modify various settings. This may include changing notification preferences, updating display settings, or customizing other user-specific configurations. |
| **Precondition** | User must have an active account and be logged in. |
| **Alternatives** | If the user wants to revert to default settings, they can do so by selecting the appropriate option in the settings section. |
| **Post Condition** | User has successfully changed their settings information. |

***UC\_05 -US\_5- User Changes Settings Information***

| **Name** | User Changes Personal Information |
| --- | --- |
| **Summary** | User can update their personal information, such as name, email, and contact details. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User navigates to the profile section where they can edit their personal information. They can update their name, email address, contact number, or any other relevant details. After making the changes, the user saves the updates to their profile. |
| **Precondition** | User must have an active account and be logged in. |
| **Alternatives** | If the user wants to cancel the changes, they can do so before saving the updates to their profile. |
| **Post Condition** | User has successfully changed their personal information. |

***UC\_06 -US\_6- User Changes Personal Information***

| **Name** | User Views Total Number of Parked Vehicles |
| --- | --- |
| **Summary** | User can view the total number of vehicles currently parked in the parking lot. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User navigates to the dashboard or parking overview section where the total number of parked vehicles is displayed. This information provides an overview of the current parking situation. |
| **Precondition** | User must have an active account and be logged in. |
| **Alternatives** | If the user wants more detailed information about the parked vehicles, they can access the detailed parking records section. |
| **Post Condition** | User has viewed the total number of parked vehicles. |

***UC\_07 -US\_7- User Views Total Number of Parked Vehicles***

| **Name** | User Manages Vehicle Categories |
| --- | --- |
| **Summary** | User can add, edit, or delete vehicle categories, such as car, motorcycle, or bicycle. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User navigates to the vehicle categories management section where they can perform various actions related to vehicle categories. They can add a new vehicle category, edit an existing category, or delete a category that is no longer needed. After making the changes, the user saves the updates to the vehicle categories. |
| **Precondition** | User must have an active account and be logged in. |
| **Alternatives** | If the user wants to cancel the changes, they can do so before saving the updates to the vehicle categories. |
| **Post Condition** | User has successfully managed the vehicle categories. |

***UC\_08 -US\_8- User Manages Vehicle Categories***

| **Name** | User Manages Vehicle Entry Records |
| --- | --- |
| **Summary** | User can add, edit, or delete vehicle entry records. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User navigates to the vehicle entry records management section where they can perform various actions related to vehicle entry records. They can add a new entry record for a vehicle, edit an existing entry record, or delete a record that is no longer needed. After making the changes, the user saves the updates to the vehicle entry records. |
| **Precondition** | User must have an active account and be logged in. |
| **Alternatives** | If the user wants to cancel the changes, they can do so before saving the updates to the vehicle entry records. |
| **Post Condition** | User has successfully managed the vehicle entry records. |

***UC\_09 -US\_9- User Manages Vehicle Entry Records***

| **Name** | User Checks Out Vehicle |
| --- | --- |
| **Summary** | User can check out a vehicle that is leaving the parking lot. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User navigates to the checkout section where they can check out a vehicle that is leaving the parking lot. They enter the vehicle's information, such as license plate number or entry record ID, and confirm the checkout. The system marks the vehicle as checked out and removes it from the parking lot. |
| **Precondition** | User must have an active account and be logged in. |
| **Alternatives** | If the user enters incorrect information for the vehicle checkout, they can correct the information before confirming the checkout. |
| **Post Condition** | Vehicle is successfully checked out and removed from the parking lot. |

***UC\_10 -US\_10- User Checks Out Vehicle***

| **Name** | User Prints Parking Ticket in PDF |
| --- | --- |
| **Summary** | User can generate and print a parking ticket in PDF format. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User navigates to the parking ticket generation section where they can generate a parking ticket for a vehicle. They enter the vehicle's information, such as license plate number or entry record ID, and select the option to generate a PDF ticket. The system generates the ticket in PDF format, which can then be printed. |
| **Precondition** | User must have an active account and be logged in. |
| **Alternatives** | If the user needs to reprint a ticket, they can do so by accessing the ticket history section and selecting the option to reprint. |
| **Post Condition** | User has successfully generated and printed a parking ticket in PDF format. |

***UC\_11 -US\_11 User Prints Parking Ticket in PDF***

| **Name** | User Views Parking Reports |
| --- | --- |
| **Summary** | User can view various parking reports, such as occupancy rate or revenue. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User navigates to the reports section where they can view different types of parking reports. This may include reports on occupancy rate, revenue, or other relevant statistics. The user can select the desired report and view it on the screen. |
| **Precondition** | User must have an active account and be logged in. |
| **Alternatives** | If the user needs to export a report for further analysis, they can do so by selecting the export option and choosing the desired format (e.g., PDF, CSV). |
| **Post Condition** | User has successfully viewed a parking report. |

***UC\_12-US\_12 User Views Parking Reports***

| **Name** | User Logs Out |
| --- | --- |
| **Summary** | User can log out of the system to end their session. |
| **Actor** | Parking Guard/ Receptionist/ Analyst |
| **Description** | User navigates to the logout section where they can log out of the system. Upon clicking the logout button, the user's session is terminated, and they are redirected to the login page. |
| **Precondition** | User must be logged in. |
| **Alternatives** | If the user accidentally clicks the logout button, they can cancel the logout action and continue using the system. |
| **Post Condition** | User is successfully logged out of the system. |

***UC\_13-US\_13 User Logs Out***

4.2 Behavioral Diagrams

4.2.1 Use Case Diagrams

4.2.2 Activity Diagrams

4.2.3 State Diagrams

4.2.4 Sequence Diagrams

4.2.5 Collaboration Diagrams

4.3 Data Flow Diagrams

4.4 Entity Relationship

4.4.1 Database Schema Design

4.4.2 Entity Relation Diagram

4.5 Structural Diagrams

4.5.1 Class Diagram

4.5.2 Object Diagrams

4.5.3 Component Diagrams

4.5.4 Deployment Diagram

**5. Implementation Technology**

PMS is a dynamic Web Application. For the creation of this software, we have combined Client Side

Scripting and Server Side Scripting. The communication between the client and the server will take

place via HTTP protocol.

For the Client Side Scripting we have used the following technologies:

• HTML 5 (HyperText Markup Language)

• CSS 3 (Cascading Style Sheets)

• JavaScript

• Ajax (Asynchronous JavaScript and XML)

• jQuery (JavaScript Framework Library)

For the user interface (UI) we have used an open source template, designed with Bootstrap, to make

The software is intuitive and more easy and attractive for the user. We have made the necessary changes

to this template in order to make it similar with what we had planned before (Sketches and Detailed

Design).

For Server Side Scripting, we have used PHP. This allows the users to interact with the software and

with each other. We have used the OOP approach (Object Oriented Programming). We have used

classes for each system user, actor: guard class, receptionist Class, analyst Class.

We have used a CRUD class, which contains all the functions needed to modify: patients, receptionists,

doctors, medical records, medical visits and feedback.

We have also used a validation class, which contains all the necessary functions for validation text,

passwords, email etc.

To administrate MySQL over the web, phpMyAdmin has been used. The database that we have created

In the same server, you can also find the entire project (Web Application) under a personal account:

[Link]

This project is also published in github, where you can find the step by step procedure of the creation

of this project (diagrams, designs, requirements, meeting reports, personal logs, etc.):

[https://github.com/EnrikoHyka/CEN-302-Software-Engineering-Project]

In the following print screens, you can see the implemented product.

**6. Application Design Structure**

**7. Application Page Result**