Online Parking Information System

Requirements Determination and Use Case Analysis / System Proposal / Analysis Phase   
(Homework No.2)

Project team: Team 07

Instructor: Dr. Araz Yusubov

Submitted in partial fulfillment of the requirements of the INFT 2303: Systems Analysis and Design course project.

|  |  |
| --- | --- |
| GitHub repository | https://github.com/ADA-SITE-INFT2303-2023-Spring/sys-dev-project-team-07.git |
| Version date | Version information |
| 22.03.23 | Initial draft |
| 28.03.23 | Final draft |

|  |  |
| --- | --- |
| Other documents in the package | |
| Use Case Diagram | This document represents the relationship between use cases and the relationship between use case and actors. |
| Homework 1B (Updated) | Updated version of Homework 1B as Bonus 1 |

|  |  |  |
| --- | --- | --- |
| Team member | Contribution to this homework (NOT the project) | Estimated % |
| Lala Mahmudova | Definitions, External Actors, UC-03, UC-05, UC-06, References | 20% |
| Emil Hajiyev | Use Case Diagram, UC-01, UC-02, UC-04, Bonus 1, Functional Requirements | 25% |
| Faraz Bagher Nezhad | User Story Descriptions, UC-07, UC-08,UC-10, Bonus 1 | 20% |
| Orkhan Ismayilov | Functional and Nonfunctional Requirements, Design Constraints, UC-09 | 20% |
| Jamaladdin Talishinskiy | Introduction, UC-11 | 15% |

Table of Contents

[1. Introduction 3](#_Toc131271901)

[Definitions 4](#_Toc131271907)

[Requirements Definition 4](#_Toc131271928)

[2. Use Case Analysis 8](#_Toc131271999)

[External Actor Descriptions 8](#_Toc131272000)

[User Story Descriptions 9](#_Toc131272013)

[Use Case Descriptions 10](#_Toc131272052)

[Use Case Number: 10](#_Toc131272054)

[References 21](#_Toc131272480)

# Introduction

This is part of the System Proposal for a hypothetical project Online Parking Information System submitted for partial fulfillment of the requirements of the Systems Analysis and Design course in the School of Information Technologies and Engineering at ADA University, Baku, Azerbaijan.

The purpose of this System Proposal document is to provide a detailed overview of the Online Parking Information System. The system is designed to provide an efficient and convenient parking solution for drivers through a mobile application that enables them to find available parking spots in real time, reserve a spot beforehand, and pay for their parking time without the need for getting a ticket from QR Code machines.

To prepare the System Proposal document for the Online Parking Information System, the team has conducted extensive research on the requirements of the system and its users. This research involved studying similar parking systems and analyzing the parking industry trends and challenges. Based on the research findings, the team has defined the functional and non-functional requirements of the system, including features such as real-time parking availability tracking, reservation management, payment processing, and user authentication.

The team has created user stories to describe how the system will be used by drivers, vehicle owners, and system administrators. These stories provide a clear understanding of the needs and expectations of the users and will guide the development process. Finally, the team has conducted a use case analysis to define the various use cases of the system, including searching for available parking spots, reserving parking spots, and managing parking rates.

The System Proposal document brings together all this work to provide a comprehensive overview of the Online Parking Information System, its requirements, and how it will function.

## Definitions

|  |  |
| --- | --- |
| Term | Definition |
| Encryption | Process of converting data into a secret code to protect it from unauthorized access. |
| In-app navigation feature  Maintenance interval | A tool that allows users to navigate within an application using a map or other visual interface.  The time between scheduled maintenance tasks for a piece of equipment or a system. |
| Mobile wallet | A digital wallet that stores payment information and allows users to make payments using their mobile devices. |
| Parking bollard  Payment processing system | Physical devices installed on the ground that can be remotely controlled to allow or restrict access to a specific area.  The platform enables businesses to process electronic payments, such as credit card transactions. |
| Quick Response (QR) code | A type of two-dimensional barcode that can be scanned and read by a smartphone camera or a QR code reader. |
| Sensor | A device used for detecting the presence or absence of a vehicle in a parking slot and sending a signal to the system of vehicle occupation. |
| User authentication | Process of verifying a user's identity to ensure that they are whom they claim to be. |

# Requirements Definition

## Functional Requirements

**1. Driver registration:**

* The system shall validate the information provided by the driver during registration to ensure data accuracy and prevent fraud. (Information-oriented)
* The system shall send a confirmation email to the driver's email address to verify the email and confirm registration. (Process-oriented)

**2. View of available parking lots:**

* The system shall have an in-app navigation feature. (Process-oriented)
* The system shall display a real-time parking availability map for the user. (Information-oriented)
* The application must have a database of all available parking lots and their details (location, price, and number of available parking lots) (Information-oriented).

**3. Reservation:**

* The system shall allow users to search for available parking spots based on their location, and time. (Process-oriented)
* The system shall display up-to-date details regarding the availability of location. (Information-oriented)
* The system shall allow users to reserve a parking spot within the desired time. (Process-oriented)

**4. Parking entrance:**

* The system shall provide a unique QR code to each registered driver. (Information-oriented)
* The system shall verify the scanned QR code and confirm the identity of the driver. (Process-oriented)
* The barrier shall be equipped with a mechanism to open automatically upon successful verification of the QR code. (Process-oriented)
* The system shall keep track of the number of available parking lots and update it in real-time. (Information-oriented)

**5. Vehicle detection:**

* The ultrasonic sensor must be installed and calibrated properly. (Process-oriented)
* The system must be able to receive signals from the sensor in real time. (Information-oriented)
* The system must be able to update the parking slot status in real time. (Information-oriented)
* The system must be able to log and display error messages in case of sensor malfunctions. (Information-oriented)

**6. Parking Time Tracking:**

* The mobile app must generate a unique QR code for each parking reservation. (Process-oriented)
* The special machine placed at entrances and exits must be able to read the QR code from the mobile app. (process-oriented)
* The system must be able to record the user's entry time accurately. (Process-oriented)

**7. Parking payment:**

* The system shall allow users to make payments for parking by using their chosen payment method, such as credit/debit card or mobile wallet options. (Process-oriented)
* The system shall furnish the user with a receipt or confirmation of payment once the transaction is completed successfully. (Information-oriented)

**8. Parking exit:**

* The system shall require a parking sensor to calculate the total parking fee and charge the user’s payment method. (Process-oriented)
* The system shall allow you to exit the parking lot after successful payment. (Process-oriented)
* The system shall update that parking space as vacant. (Information-oriented)

**9.Parking history:**

* The system shall allow users to view parking history, fees, and locations. (Information-oriented)
* The system shall provide users with the ability to download their parking history receipts. (Information-oriented)

**10. Contact customer support:**

* The system shall provide a customer support contact options such as email, phone number and online chat. (Information-oriented)
* The system shall allow user to give feedback or report any problem that they are faced. (Process-oriented)
* The system shall provide a response to all feedback and a solution to their problems within a certain time. (Process-oriented)

**11. Parking rates management:**

* The system must provide a secure login mechanism for the system admin. (information-oriented)
* The system must allow the system admin to edit parking rates for different time periods (hourly, daily, and monthly). (Process-oriented)

## Nonfunctional Requirements

**1.Operational requirements:**

* The system should be always accessible, except for planned maintenance intervals.
* The system should have the ability to process at least 100 concurrent user requests without causing a notable decline in performance.
* The system should be able to recover from a system failure in less than 5 minutes.

**2.Performance requirements:**

* The system should make transactions within 10 seconds.
* The system should be able to detect a car within less than 5 seconds.
* The system should update the information regarding available lots every minute.

**3. Usability requirements:**

* The system should possess a user-friendly interface that is simple to navigate.
* The system should provide precise guidance and feedback to the users throughout the entire process.

**4.Reliability requirements:**

* The system should be operational and available to users at least 99% of the time.
* The system should be able to recover from hardware and software failure within an hour.
* The system should be able to handle at least 10,000 user requests per day without any interruptions.

**5.Security requirements**:

* The system should use encryption to safeguard all communication between the user and the system.
* The system should mandate that users authenticate themselves with a username and password while accessing their accounts.
* The system should also have a recovery or backup plan in case of a security breach.

## Design Constraints

* The system should be designed to work with different types of mobile devices, including tablets, and should adapt an application to the limitations of those devices, such as screen size, operating systems, and processing power.
* The system should be compatible with the existing parking infrastructure for being suitable for the intended locations.
* The system should integrate with the payment processing systems used by parking lots.
* The system should be able to integrate with Quick Response (QR) code readers to identify the car to open a barrier for parking and exiting.
* The system should accept different types of payment methods and comply with relevant data privacy and security regulations.
* The system should be designed to provide high availability and low latency, facilitating real-time updates of the occupancy status of parking slots and payment transactions.

# Use Case Analysis

## External Actor Descriptions

**Human actors:**

* Driver - end user of the system who will use a mobile application to locate and reserve parking spots, pay for parking fees.
* Admin/parking lot attendant - a person who will use the management console to monitor parking spaces, view occupancy data, and manage reservations.

**Hardware actors:**

* Sensor - hardware component that detects the presence of vehicles in parking spots and send data to system’s central servers.[[1]](#footnote-1)
* Automatic parking bollards – physical devices that are interacting with the system installed on the ground that can be remotely controlled to allow or restrict access to specific area.[[2]](#footnote-2)
* Mobile devices - hardware component used by drivers to access the mobile application.

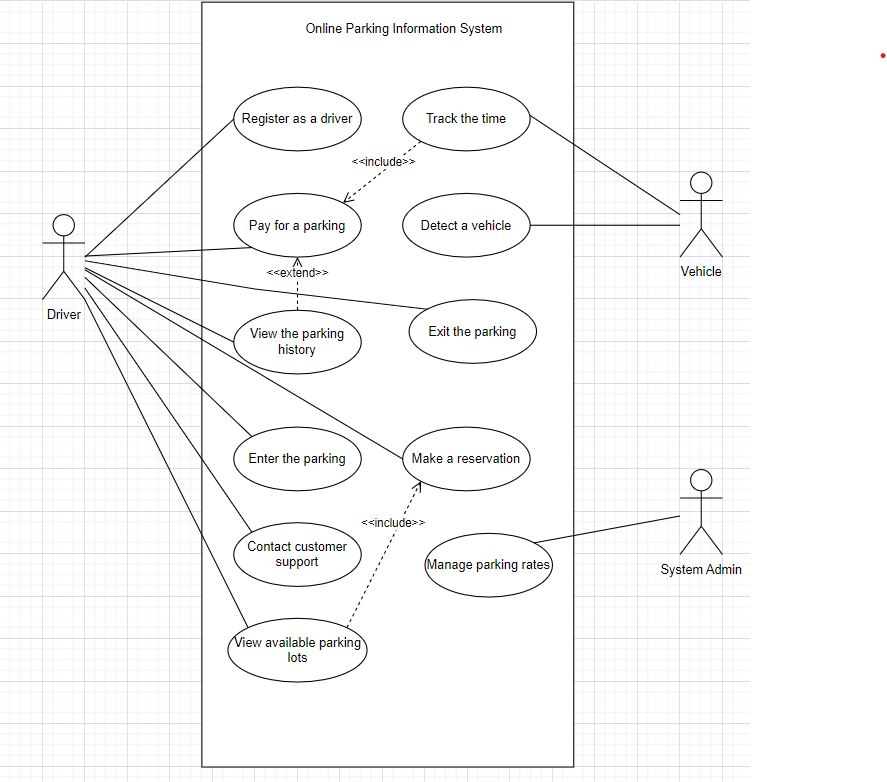
**Software system actors:**

* Central server - main component of the system that processes and manages the data collected from the sensors and cameras, manages reservations and payments made by customers, and generates reports for parking lot owners or other officials.
* Mobile application - component of the system that allows drivers to locate and reserve parking slots, pay for parking fees, and receive notifications.
* Payment Gateway - component of the system that securely processes payments made by customers through the mobile application.
* Management console - administrative component of the system that securely processes payments made by customers through the mobile application.[[3]](#footnote-3)

## User Story Descriptions

|  |  |  |
| --- | --- | --- |
| User story name | Description | Release |
| Driver Registration | The user creates an account in the Online Parking Information System as a driver to access all the features available. | R1 |
| View of available parking lots | The driver can view the availability of parking lots on the map and select an appropriate one. | R1 |
| Reservation | Drivers can reserve a parking spot beforehand when they come to the lot. | R1 |
| Parking Entrance | The barriers in parking open, when registered cars have reservations on parking spaces. | R1 |
| Vehicle Detection | Sensors and cameras will detect vehicle plate numbers to enter and leave the parking slots. | R1 |
| Parking Time Tracking | The system will calculate the duration of the driver’s parking time at the entrance to the parking slots with the help of QR code scanning. | R1 |
| Parking Payment | Drivers must pay the parking fee with a credit card, debit card, or online wallet. Their fee is calculated based on the time spent on a parking lot. | R1 |
| Parking Exit | Drivers are allowed to exit the parking lot after paying parking fees, and slots are made available in the system. | R1 |
| Parking History | Driver offers will be listed and saved in our database. | R2 |
| Customer Support | Drivers will get 24/7 support from our customer support team. | R2 |
| Parking Rates Management | System administrators can change the prices for specific locations, such as crowded places, or parking during rush hours. | R2 |

### Use Case Diagram



## Use Case Descriptions

### Use Case 1

|  |  |
| --- | --- |
| **Use Case Number:** | UC-01 |
| **Use Case Name:** | Register as a Driver |
| **Actor(s):** | Driver |
| **Description:** | The user creates an account in the Online Parking Information System as a driver to access all the features available. |
| **Priority (Release)** | R1 |
| **Trigger:** | External - The driver decides to register for the Online Parking Information System. |
| **Pre-condition(s):** | * Driver has uploaded application of the system * Driver has an Internet access * Driver has a valid email |
| **Main (Success) Flow:** | 1. Driver opens an Online Parking Information System application. 2. System requests personal information i.e. name, surname, phone number, e-mail address, and a password. 3. The driver enters their personal information and submits the registration form. 4. System verifies that required information is provided. 5. If information is invalid System displays a message. Return to Step 2 6. The system sends a confirmation email to the driver's email address. 7. The driver clicks on the link in the confirmation email to confirm their registration. 8. The system confirms the registration and adds the driver's information to the database. |
| **Alternate Flows:** | Alternate Flow #1: The email confirmation link is not clicked within a specified time frame. The following steps occur:   1. The system detects that the email confirmation link has not been clicked within the specified time frame. 2. The system sends a reminder email to the user to click the confirmation link:  * If the user clicks the confirmation link within the time frame specified in the reminder email, the use case continues with Step 7 of the main flow. * If the user does not click the confirmation link within the time frame specified in the reminder email, the system marks the registration as expired.   Alternate Flow #2: The user enters an existing registered email. Following steps occur:   1. System detects that the email address provided is already registered in the system. 2. System displays an error message informing the user that the email address is already registered.   Alternate Flow #3: The user enters incorrect or incomplete information. The following steps occur:   1. System detects missing or incorrect information in the registration form. 2. System displays an error message informing the user of the issue and prompts them to correct the form. 3. User corrects the form and resubmits it. 4. System confirms that all required fields have been completed correctly. |
| **Post Condition:** | Alternate Flow #1: The driver’s information is deleted and is not stored.  Alternate Flow #2: The user is not registered as a driver but has the option to log in or use a different email address to register.  Alternate Flow #3: The driver is registered as a user of the Online Parking Information System |
| **Requirements:** | 1. The system shall validate the information provided by the driver during registration to ensure data accuracy and prevent fraud. 2. The system shall send a confirmation email to the driver's email address to verify the email and confirm registration. |

### Use Case 2

|  |  |
| --- | --- |
| **Use Case Number:** | UC–02 |
| **Use Case Name:** | View available parking lots |
| **Actor(s):** | Driver |
| **Description:** | The driver can view the availability of parking lots on the map and select an appropriate one. |
| **Priority (Release)** | R1 |
| **Trigger:** | External - The driver needs to find a parking lot to park their vehicle. |
| **Pre-condition(s):** | * The driver has a registered account in the system. * The driver has an active internet connection. * The system has permission to access the device's location. |
| **Main (Success) Flow:** | 1. The driver opens the application and selects the "Find Parking" option. 2. The system displays a map with all available parking lots marked on it. 3. The driver can zoom in/out or pan the map to see parking lots in different areas. 4. The driver can select a parking place to view its details (location, price, and number of available parking lots). |
| **Alternate Flows:** | Alternate Flow #1: No parking lots are available in the selected area. The following steps occur:   1. The system displays a message informing the driver that there are no available parking lots in the selected area. 2. The driver can zoom out or pan the map to see parking lots in other areas. 3. If no parking lots are available in any area, the driver can choose to receive a notification when a parking space becomes available.   Alternate Flow #2: The driver's location is not available. The following steps occur:   1. The system displays a message informing the driver that their location is not available. 2. The driver can manually search for parking lots in a specific area. |
| **Post Condition:** | The system displays the selected parking lot on the map with its current availability status.  The driver can navigate to the selected parking lot using the in-app navigation feature.  Alternate Flow #1: The driver is informed that there are no parking spaces in the vicinity.  Alternate Flow #2: The driver is informed that their location is not available. |
| **Requirements:** | 1. The system shall have the in-app navigation feature. 2. The system shall display a real-time parking availability map for the user. 3. The application must have a database of all available parking lots and their details (location, price, and number of available parking lots). |

### 

### Use Case 3

|  |  |
| --- | --- |
| **Use Case Number:** | UC-03 |
| **Use Case Name:** | Make a reservation |
| **Actor(s):** | Driver |
| **Description:** | This use case allows a driver to reserve a parking spot in advance using the smart parking system's mobile application. |
| **Priority (Release)** | R1 |
| **Trigger:** | External: The driver needs to reserve a parking spot in advance. |
| **Pre-condition(s):** | * The driver must have a registered account on the smart parking system's mobile application. * The driver must have a valid payment method linked to their account. * There must be available parking spots in the smart parking system that can be reserved. |
| **Main (Success) Flow:** | 1. The driver opens the smart parking system's mobile application. 2. The driver selects the "Reserve Parking" option. 3. The driver enters their desired location and parking duration. 4. The smart parking system's application checks for available parking spots in the specified location and duration. 5. If available parking spots are found, the driver selects a parking lot and confirms the reservation. 6. The smart parking system's application stores the reservation details and charges the driver's account. 7. The parking bollard opens when the driver approaches the reserved spot and clicks on “open the barrier” |
| **Alternate Flows:** | Alternate Flow #1: There are no available parking spots. The following steps occur:   1. The smart parking system's application notifies the driver that no spots are available. 2. The system suggests the driver alternative locations or times.   Alternate Flow #2: The driver cancels the reservation. The following steps occur:   1. If the driver changes their mind or encounters an issue, they can cancel the reservation before confirming it. 2. The smart parking system's application cancels the reservation and releases the spot for other drivers to reserve. |
| **Post Condition:** | 1. The driver's reservation details are stored in the smart parking system's database. 2. The driver's account is charged the appropriate fee for the reserved parking lot.   Alternate Flow #1: The driver is provided with alternative parking options or times.  Alternate Flow #2: The reservation is canceled in the system. |
| **Requirements:** | 1. The smart parking system's mobile application must have access to the smart parking system's database to check for available parking spots. 2. The mobile application must be able to process payment transactions securely and efficiently. 3. The smart parking system must have a way to notify the driver of the reservation confirmation and reservation details. 4. The parking slot should have an automatic parking bollard to prevent other people entering to reserved spot. |

### Use Case 4

|  |  |
| --- | --- |
| **Use Case Number:** | UC-04 |
| **Use Case Name:** | Enter the parking |
| **Actor(s):** | Driver |
| **Description:** | The driver scans the unique QR code available in the application before entering the parking lot. After successful scanning, the barrier opens, and the driver can enter the parking lot and take a free parking space. |
| **Priority (Release)** | R1 |
| **Trigger:** | External - The driver wants to enter the parking place. |
| **Pre-condition(s):** | * The driver is registered in the system and has a unique QR code available in the application. * The barrier is closed. * The parking place has available parking lots. * The QR code scanner is in good working condition and can properly scan the driver's QR code. |
| **Main (Success) Flow:** | 1. The driver approaches the entrance to the parking place. 2. The driver opens the application on their device and scans the unique QR code provided by the system. 3. The system verifies the QR code and confirms the identity of the driver. 4. The barrier opens automatically, and the driver enters the parking place. 5. The driver proceeds to take an available parking lot. |
| **Alternate Flows:** | Alternate flow #1: QR code scanning failed. The following steps will occur:   1. The driver attempts to scan the QR code. 2. The system detects that the QR code was not successfully scanned. 3. The system displays an error message indicating that the QR code scanning failed. 4. The driver is prompted to try scanning the QR code again or contact customer support if the problem persists.   Alternate flow #2: The parking lot is full. The following steps will occur:   1. The driver enters the parking place. 2. The system detects that all parking lots are currently occupied. 3. The system displays a message indicating that the parking lot is full. 4. The driver is prompted to try again later or find another parking place. |
| **Post Condition:** | The driver is inside the parking space.  Alternate Flow #1: The barrier is closed.  Alternate Flow #2: The driver fails to park the vehicle. |
| **Requirements:** | 1. The system shall provide a unique QR code to each registered driver. 2. The system shall verify the scanned QR code and confirm the identity of the driver. 3. The barrier shall be equipped with a mechanism to open automatically upon successful verification of the QR code. 4. The system shall keep track of the number of available parking lots and update it in real time. |

### 

### Use Case 5

|  |  |
| --- | --- |
| **Use Case Number:** | UC-05 |
| **Use Case Name:** | Detect the vehicle |
| **Actor(s):** | Sensor |
| **Description:** | This use case describes the process of detecting a vehicle in a parking slot. |
| **Priority (Release)** | R1 |
| **Trigger:** | External - A vehicle enters or exits a parking slot. |
| **Pre-condition(s):** | * The sensor is functional and calibrated properly. * The parking slot is not occupied by any vehicle. |
| **Main (Success) Flow:** | 1. The sensor continuously monitors the parking lot for the presence of a vehicle. 2. When a vehicle enters the parking lot, the sensor detects it and sends a signal to the system indicating that the lot is occupied. 3. The system updates the parking lot status to "occupied" in real-time. 4. When the vehicle exits the parking lot, the sensor detects it and sends a signal to the system indicating that the lot is now available. 5. The system updates the parking lot status to "available" in real-time. |
| **Alternate Flows:** | Alternate Flow #1: The sensor malfunctions or fails to detect the vehicle. The following steps occur:   1. The system does not receive a signal from the sensor. 2. The system logs an error message and alerts the maintenance team. |
| **Post Condition:** | The sensor detects a vehicle in the parking slot and sends a signal to the system.  Alternate Flow #1: The parking slot status remains unchanged. |
| **Requirements:** | 1. The sensor must be installed and calibrated properly. 2. The system must be able to receive signals from the sensor in real-time. 3. The system must be able to update the parking slot status in real-time. 4. The system must be able to log and display error messages in case of sensor malfunctions. |

### Use Case 6

|  |  |
| --- | --- |
| **Use Case Number:** | UC-06 |
| **Use Case Name:** | Track the time |
| **Actor(s):** | Driver |
| **Description:** | The process of tracking the time of a user's entry into a parking garage using a QR code in the mobile app. |
| **Priority (Release)** | R1 |
| **Trigger:** | External - user's arrival at the parking garage and their presentation of the QR code to the special machine |
| **Pre-condition(s):** | * The user has installed the mobile app and registered for an account. * The user has made a reservation for a parking slot in the garage. * The user has generated a QR code for their reservation in the mobile app. |
| **Main (Success) Flow:** | 1. The user arrives at the parking garage entrance. 2. The user shows the QR code from the mobile app to the special machine. 3. The special machine reads the QR code from the smartphone and confirms the user's entry. 4. The system records the user's entry. 5. The gateway opens and the user enters the parking garage. 6. The system completes tracking of a total time when the user clicks “I am ready to exit”. |
| **Alternate Flows:** | Alternate Flow #1: The user's payment method is invalid. The following steps occur:   1. System displays an error message indicating that the payment method is invalid. 2. User returns to step 1 of the main flow to update the payment method.   Alternate Flow #2: The user's account balance is insufficient. The following steps occur:   1. System displays an error message indicating that the account balance is insufficient. 2. User returns to step 1 of the main flow to add funds to the account. |
| **Post Condition:** | 1. The system has recorded the user's entry time. |
| **Requirements:** | 1. The mobile app must generate a unique QR code for each parking reservation. 2. The special machine placed at entrances and exits must be able to read the QR code from the mobile app. 3. The system must be able to record the user's entry time accurately. |

### Use Case 7

|  |  |
| --- | --- |
| **Use Case Number:** | UC-07 |
| **Use Case Name:** | Pay for a parking |
| **Actor(s):** | Driver |
| **Description:** | Drivers must pay the parking fee with a credit card, debit card, or online wallet. Their fee is calculated based on the time spent on a parking lot. |
| **Priority (Release)** | R1 |
| **Trigger:** | External - Drivers have to pay fees to park their vehicles. |
| **Pre-condition(s):** | * Driver registered into the system and the system has a payment field. * System shows two options: Pay via Credit/Debit card or Pay via Wallet Balance. * Pay via Credit/Debit cards are done by online transactions via bank. * Pay via Wallet is done by online transactions via terminals such as Milliön or E-manat by drivers beforehand. |
| **Main (Success) Flow:** | 1. Driver opens the system. 2. Registration done by Driver. 3. Driver presses the payment button; it shows two options: pay via Credit/Debit Card or Wallet. 4. Pay via Credit/Debit Card is done by online Bank transactions. 5. Pay via Wallet system takes money from Driver's Wallet balance, which is money transferred by Local terminals such as Milliön or E-manat by drivers beforehand. |
| **Alternate Flows:** | Alternate Flow #1: Transactions do not occur. The following steps occur:   1. The system displays a message informing the driver that your transaction has failed. 2. Driver contacts with 24/7 Contact Customer Support to explain that issue. 3. Contact customer support replies to the user and tracks the transactions.   Alternate Flow #2: There is no amount in the driver's wallet. The following steps occur:   1. Driver pays via Credit/Debit Card to transfer money online by bank transactions. 2. Driver transfers money to Wallet by Local terminals such as Milliön or E-manat. |
| **Post Condition:** | Drivers should pay fees by credit/debit card or wallet; if not, they cannot use the system.  Alternate Flow #1: The amount of money is returned to ’s Wallet Balance.  Alternate Flow #2: The driver’s money is transferred to the Wallet Balance. |
| **Requirements:** | * The system shall allow users to make payments for parking by using their chosen payment method, such as credit/debit card or mobile wallets options. * The system shall furnish the user with a receipt or confirmation of payment once the transaction is completed successfully. * The system shall make transactions within 10 seconds. |

### 

### Use Case 8

|  |  |
| --- | --- |
| **Use Case Number:** | UC-08 |
| **Use Case Name:** | Exit the parking |
| **Actor(s):** | Driver |
| **Description:** | Drivers are allowed to exit the parking lot after paying parking fees, and slots are made available in the system. |
| **Priority (Release)** | R1 |
| **Trigger:** | External - Driver have to pay fees to leave the parking area. |
| **Pre-condition(s):** | * The driver has a registered account in the system. * The driver has an active internet connection. * The driver has to pay a parking fee. |
| **Main (Success) Flow:** | 1. Driver opens the system, clicks on finish the offer. 2. Sensor calculates the duration of time that the driver was parking the vehicle. 3. Driver pays the parking fees by Credit/Debit card or wallet. 4. Driver shows the QR code to the barrier after paying fees. 5. The barrier goes up to let the driver leave the parking slot. 6. The driver parking slot will be available in our system, for next customers. |
| **Alternate Flows:** | Alternate Flow #1: The driver does not pay a parking fee. The following steps occur:   1. The driver cannot leave the parking slot, because the barrier will not open. 2. After 1 month, the driver will be banned from our system, and the driver cannot register the system with that plate number, till pay fees.   Alternate Flow #2: The barrier does not open for the driver. The following steps occur:   1. The driver contacts Contact customer support. 2. Contact customer support team gets information from the database. 3. The customer support team fixes the issue to help drivers leave the parking slot.   Alternate Flow #3: The slot does not seem available for the next reservations. The following steps occur:   1. Our Technics move to that parking slot. 2. They look up our system sensors. 3. They fix the sensor to work properly. |
| **Post Condition:** | 1. The driver has successfully exited the parking lot. 2. The parking space is marked as available in the system. |
| **Requirements:** | 1. The system shall track the real-time availability of parking spots using cameras. 2. The system shall require parking sensor to calculate the total parking fee and charge the user’s payment method. 3. The system shall allow to exit the parking lot after successful payment. 4. The system shall update that parking space as vacant. |

### 

### Use Case 9

|  |  |
| --- | --- |
| **Use Case Number:** | UC-09 |
| **Use Case Name:** | View the parking history |
| **Actor(s):** | Driver |
| **Description:** | The driver will be able to view their past parking history, including the date, fee, and location of parking |
| **Priority (Release)** | R2 |
| **Trigger:** | External - the driver will select the ‘View Parking History’ option in the app or on the website |
| **Pre-condition(s):** | * The driver should have an actively registered account for the smart parking system. * The driver should have parked their car at least once before using this option |
| **Main (Success) Flow:** | 1. The driver chooses the ‘View Parking History. 2. The system requires the driver to identify him/herself. 3. The system checks the driver’s credentials and displays parking history, time, and location. 4. The driver can choose ‘additional detail’ to see the start time, end time, and amount paid. |
| **Alternate Flows:** | 1. When the driver inputs the wrong login information, the system will give an error and request to attempt again. 2. If the driver has not used an application previously, the system will show an absence of any parking records. 3. While a technical works on a system or a system failure, the system will display an error that says the driver to try a bit later |
| **Post Condition:** | The driver will be able to view him/herself parking history, including time, and location of each parking session |
| **Requirements:** | 1. The system shall allow users to view parking history, fees, and locations. 2. The system shall provide users with the ability to download their parking history receipts. |

### 

### Use Case 10

|  |  |
| --- | --- |
| **Use Case Number:** | UC-10 |
| **Use Case Name:** | Contact customer support |
| **Actor(s):** | Driver |
| **Description:** | Drivers will get 24/7 support from our customer support team to solve their difficulties. |
| **Priority (Release)** | R2 |
| **Trigger:** | External - The Driver should get registered in our system. |
| **Pre-condition(s):** | * The driver has a registered account in the system. * The driver has an active internet connection. |
| **Main (Success) Flow:** | 1. The driver opens the parking system application. 2. The driver navigates to the "Support" section of the application. 3. The driver selects the type of issue they are experiencing (e.g., payment, reservation, technical issue). 4. The driver provides a brief description of the issue. 5. The driver submits the support request. 6. The parking system sends a confirmation message to the driver, indicating that the request has been received. 7. A customer support representative reviews the request and contacts the driver to provide assistance. 8. The customer support representative works with the driver to resolve the issue. |
| **Alternate Flows:** | Alternate Flow #1: The driver cannot write in the chat box. The following steps occur:   1. The driver calls our Contact customer support, to tell his/her issue in words. 2. Our Contact customer support team helps them to get rid of this issue. |
| **Post Condition:** | The driver's issue has been resolved to their satisfaction.  The parking system has recorded the support request and the actions taken to resolve the issue. |
| **Requirements:** | 1. The system shall provide customer support contact options such as email, phone number, and online chat. 2. The system shall allow the user to give feedback or report any problem that they are faced. 3. The system shall provide a response to all feedback and a solution to their problems within a certain time. |

### 

### Use Case 11

|  |  |
| --- | --- |
| **Use Case Number:** | UC-11 |
| **Use Case Name:** | Manage parking rates |
| **Actor(s):** | System admin |
| **Description:** | System administrators should manage the parking fees charged to drivers for parking in the parking lot. |
| **Priority (Release)** | R2 |
| **Trigger:** | External - Changes in operating costs, competition, and demand require a change in the parking fees charged to drivers. |
| **Pre-condition(s):** | * The system admin must have access to the system with the necessary permissions to manage parking rates. * The parking lot sensors must be functional and accurately track the time parked by drivers. |
| **Main (Success) Flow:** | 1. The system admin navigates to the "Manage Parking Rates" section of the system. 2. The system displays the current parking rates and options for editing them. 3. The system admin selects the option to edit the parking rates. 4. The system displays a form for editing the parking rates, including options for setting hourly, daily, and monthly rates. 5. The system admin enters the new rates and submits the form. 6. The system updates the parking rates for all drivers and stores the changes in the database. 7. The system displays a confirmation message to the system admin indicating that the changes have been successfully saved. |
| **Alternate Flows:** | Alternate Flow #1: System admin enters invalid inputs. The following steps occur:   1. If the system admin enters invalid inputs into the form, such as non-numeric values, the system displays an error message indicating the invalid input. 2. The system admin is prompted to correct the invalid input and resubmit the form. |
| **Post Condition:** | The parking rates have been updated in the system and are immediately in effect for all drivers.  Alternate Flow #1: The parking rates remain unchanged. |
| **Requirements:** | 1. The system must provide a secure login mechanism for the system admin. 2. The system must allow the system admin to edit parking rates for different time periods (hourly, daily, and monthly). |

# References

1. Abhishek, K. (2021). *Smart parking management system* . ResearchGate. Retrieved from: https://www.researchgate.net/publication/353924045\_Smart\_Parking\_Management\_System

2.Borklu, H. R., & Kalyon, S. A. (2017). A Design Study of an Innovative Barrier System for Personal Parking Lots. *Journal of Science*, 113–123.

3.Stiawan, R., Kusumadjati, A., Aminah, N. S., Djamal, M., & Viridi, S. (2019). An ultrasonic sensor system for vehicle detection application. *Journal of Physics: Conference Series*, *1204*. https://doi.org/10.1088/1742-6596/1204/1/012017

1. Stiawan, R., Kusumadjati, A., Aminah, N. S., Djamal, M., & Viridi, S. (2019). An ultrasonic sensor system for vehicle detection application. *Journal of Physics: Conference Series*, *1204*. https://doi.org/10.1088/1742-6596/1204/1/012017 [↑](#footnote-ref-1)
2. Borklu & Kalyon, *A Design Study of an Innovative Barrier System for Personal Parking Lots* 2017 [↑](#footnote-ref-2)
3. Abhishek, K. (2021). *Smart parking management system* . [↑](#footnote-ref-3)