# Software Requirements Specification

# PRJ566 – Winter 2023

# PRJ566 – Team No: 2

# Name of Project:  SmartPark

# Project Leader: Fevin Patel

**Last updated: May 27, 2023**

**Team Members:**

**1. Fevin Patel**

**2. Vraj Patel**

**3. Yash Patel**

**4. Dmytro Benko**

# TABLE OF CONTENTS

1. **Introduction/Overview - Document Information**
   1. **Document Authors**
   2. **Revision History**
   3. **Document Conventions**
   4. **Document Purpose**
   5. **Intended Audience**
   6. **Group Agreement**
2. **Project Overview**
   1. **Project Proposal**
   2. **Stakeholders and Users**
   3. **Functional Requirements**
   4. **Nonfunctional Requirements**
   5. **Project Scope**
   6. **System risks**
   7. **Operating Environment**
   8. **UI/UXD Interface Mockups**
3. **Process & Data Modeling**
   1. **UML Modeling: DFDs & Activity Diagrams**
   2. **Use Case Specification** 
      1. **Business Rules**
      2. **System Use Case Diagrams**
      3. **Use Case Description Tables**
   3. Prototype Demo
4. **Domain Class Diagram**
5. **Database (Select either 5.1 or 5.2)**
   1. **RDBMS Artifacts**
      1. Scripts to create, populate, delete tables
      2. Data Dictionary
   2. **NoSQL Artifacts**
6. **Work breakdown Structure (WBS)**
7. **Milestones**
8. **Acceptance Criteria**
9. **Implementation Schedule**
10. **Client / Faculty Sign-off**

# 1 - Introduction/Overview - Document Information

## 1.1 Document Authors

Fevin Patel, Vraj Patel, Yash Patel, Dmytro Benko

## 1.2 Revision History

|  |  |
| --- | --- |
| Week 03 | Sections of this document that were completed/updated this week, example: completed/updated  **1. Introduction/Overview - Document Information (completed)**  1.1Document Authors  1.2Revision History  1.3Document Conventions  1.4Document Purpose  1.5Intended Audience  1.6Group Agreement  **2. Project Overview**  2.1 Project Proposal |
| Week 04 | 2.2 Stakeholders and Users  2.3 Functional Requirements  2.4 Non-functional Requirements  2.6 System risks |
| Week 05 | 2.5 Project Scope  2.7 Operating Environment  2.8 UI/UXD Interface Mockups |
| Week 06 | **3. Process & Data Modelin**g 3.1 UML Modeling: DFDs & Activity Diagrams |
| Week 07 | 3.2 Use Case Specification  3.2.1 Business Rules  3.2.2 System Use Case Diagrams  3.2.3 Use Case Description Tables |
| Week 08 | 3.3 Prototype Demo |
| Week 09 | **4. Domain Class Diagram** |
| Week 10 | **5. Database (Select either 5.1 or 5.2)**  5.1 RDBMS Artifacts |
| Week 11 | **6. Work breakdown Structure (WBS)**  **7. Milestones**  **8. Acceptance Criteria** |
| Final | **9. Implementation Schedule** |

## 1.3 Document Conventions

Any text in red indicates an exception or error

Any text in blue is in-progress

Any text highlighted in yellow is an important point.

Any text *italicized* represents definitions.

Any text with ~~strike-through~~ is deleted.

Any text in bold shows sub-heading.

## 1.4 Document Purpose

The purpose of this document is to provide an overview of the project "Smart Park," a new website aimed at providing a platform for users to find and book parking spaces. The document outlines the key details, objectives, and requirements of the project. It serves as a guide for project stakeholders, team members, and anyone involved in the planning, development, and implementation of the project. The document aims to facilitate effective communication, alignment, and collaboration throughout the project's lifecycle.

## 1.5 Intended Audience

* Project Professor: The document is for our project professor to assess our project planning and progress.
* Project Team Members: The document is for our small team to understand and align the project requirements and goals.
* Project Stakeholders: The document is for individuals or organizations involved or interested in our project.
* Other Relevant Parties: The document may be shared with potential users, advisors, or partners for feedback and support.

## 1.6 Group Agreement

**TEAM AGREEMENT**

**Team #: 2**

**Project Title:** SmartPark - Revolutionizing Parking Management

**Project Time Frame:** Till PRJ666 next Semester

**Team Members:** Fevin Patel, Vraj Patel, Yash Patel, Dmytro Benko

**Team Leadership:** Fevin Patel

**Team Functions:**

* *We will share information and updates through WhatsApp and Teams channel.*
* *We will have conference calls to discuss progress, share ideas, and address any issues.*
* *We will collaborate and provide input on each other's tasks to ensure the best outcomes.*
* *We will communicate openly and respectfully, promoting a positive team environment.*

**Team Meetings:** We will generally meet to discuss project progress, assign tasks, brainstorm solutions and preferred meeting once a week via video conference.

**Team Problems:** If there are any problems or disagreements in the team, we promise to deal with them quickly and honestly. We want everyone to talk openly and listen carefully, and work together to find the best solutions for everyone involved.

**Team Commitment**  :

We agree to work together on the project until the end of PRJ666 next semester. We know that it's our job to make sure everything we produce is good quality.

**The undersigned members agree to work together on the project until the end of the PRJ666 next Semester. They recognize that as a team and individually they are responsible for the quality of all deliverables.**

**Name**  **Date**

|  |  |
| --- | --- |
| Fevin Patel | 28/05/2023 |
| Vraj Patel | 28/05/2023 |
| Yash Patel | 28/05/2023 |
| Dmytro Benko | 28/05/2023 |

# 2 - Project Overview

## 2.1 Project Proposal

**Project Background**

SmartPark is being developed as a solution to the increasing difficulties faced by car owners in finding and securing parking in urban environments. Increasing urbanization, a rise in vehicle ownership, and inefficient traditional parking systems have created a pressing need for a smart, technology-based solution.

**Problem Statement**

|  |  |
| --- | --- |
| The Problem of: | Locating available parking spaces in crowded downtown areas, dealing with inefficient ticketing systems, and contributing to traffic congestion. |
| Affects: | Car owners, businesses with parking spaces, municipal traffic management, and the overall downtown community. |
| The impact of which is: | Wasted time, increased frustration, decreased productivity, lost revenue opportunities, and increased traffic-related issues. |
| A successful solution would: | Streamline parking operations, provide real-time parking availability information, reduce traffic congestion, enhance revenue opportunities, and improve overall convenience for car owners. |

**Product Vision**

|  |  |
| --- | --- |
| For | Urban car owners and operators of parking spaces |
| Who | Struggle with locating available parking spaces and optimizing the use of these spaces |
| SmartPark | Is a web-based parking management solution |
| That | Leverages advanced technologies to provide real-time parking availability information, efficient reservation, and payment systems. |
| Unlike | Traditional parking management systems or basic parking apps |
| Our product | Offers a comprehensive parking solution by providing real-time parking availability, efficient online ticketing and payment system, and a user-friendly interface that significantly enhances the parking experience. |

## 2.2 Stakeholders and Users

|  |  |
| --- | --- |
| Project Team | The team members are responsible for the development and implementation of the SmartPark system. |
| Professor | Professor Yasser Elmankabady, overseeing the PRJ566 subject, providing guidance and evaluation. |
| Ministry of Education Ontario | Government entity regulating education in Ontario. They may have an interest in educational projects. |
| Parking Lot Owners | Owners of parking lots adopting and implementing the SmartPark system for their parking facilities. |
| Parking Lot Operators | Personnel operating the parking lots and utilizing the SmartPark system for day-to-day management. |
| Car Owners | End users of the SmartPark system benefit from the enhanced parking experience. |
| City Officials | Representatives from the local government or city administration involved in parking management. |
| Payment Gateway Providers | Companies or financial institutions provide payment processing services within the SmartPark system. |
| Legal Advisors | Legal experts ensuring compliance with relevant laws, regulations, and data privacy requirements. |
| Maintenance and Support Team | Individuals or department responsible for ongoing maintenance, support, and updates of the system. |
| Customers and Partners | Potential clients, business partners, or investors interested in the SmartPark system and its benefits. |

## 2.3 Functional Requirements

**1. User Management**

- User Registration

* The system will provide user registration functionality where users can create new accounts by providing their necessary details such as name, email, and password.
* The system validates the uniqueness of the email address to ensure each user has a unique identifier.
* The system stores the user information securely and confidentially.

- User Login

* The system will provide user login functionality for registered users to access their accounts.
* The system will authenticate the user's credentials (email and password) to grant access to the user's personalized features and information.

**2. Search and Booking**

- Parking Spot Search

* The system will provide a search functionality where users can enter their desired location, date, time, and duration of parking.
* The system will also use the information provided to retrieve a list of available parking spots that match the search criteria.
* The system also displays relevant details for each parking spot, including location, price, availability, and additional amenities.

- Parking Spot Booking

* The system will allow users to select a parking spot from the search results and proceed with the booking process.
* The system will display the booking details, including the selected spot, date, time, duration, and total cost.
* The system will provide a secure payment gateway for users to make online payments using various payment methods.

**3. Map Integration**

- Display Parking Spots on Map

* The system has integrated maps and geolocation services to visually display parking spots on a map interface.
* The system will utilize markers or pins to indicate the locations of available parking spaces.
* The system shall allow users to interact with the map to zoom in, zoom out, and explore different areas.

- Spot Details and Selection

* The system will provide detailed information about each parking spot when users click on a marker or pin.
* The system allows users to select a parking spot directly from the map interface and proceed with the booking process.

**4. Pricing and Payment**

- Pricing Calculation

* The system will calculate the cost of parking based on the selected duration and any additional fees or charges.
* The system also displays the calculated price to users during the booking process for transparency and confirmation.

- Secure Payment Gateway

* The system is integrated with a secure payment gateway that supports various payment methods, such as credit cards and digital wallets.
* The system will handle the payment process securely, encrypting sensitive payment information and adhering to industry standards and regulations.

**5. User Dashboard**

- Reservation Management

* The system provides a user dashboard where users can view and manage their reservations.
* The system shall display an overview of upcoming and past reservations, including details such as spot information, dates, and times.
* The system allows users to make changes to their reservations, such as modifying dates, times, or canceling bookings, if allowed.

**6. Confirmation and Notifications**

- Booking Confirmation

* The system allows sending confirmation email or notification to the user after a successful booking.
* The confirmation message includes details such as parking spot information, address, date, and time of the reservation.

- Reminder Notifications

* The system can send reminder notifications to users closer to the booking date to ensure a smooth parking experience.
* The reminder messages contain relevant details, including the reservation date, time, and parking spot information.

**7. Customer Support**

- Contact Form

* The system includes a contact form that allows users to submit inquiries, feedback, or support requests.
* The system will capture and store the user's contact information along with their message for further communication and support.

## 2.4 Nonfunctional Requirements

**1. Operational Requirements:**

**1.1 Availability:**  
The SmartPark system must be available to users 24 hours a day, 7 days a week, with minimal downtime for maintenance or upgrades.

**1.2 Reliability:**

The system must be reliable and ensure that it works properly without frequent failures or errors.

**1.3 Scalability:**

The system must be designed in such a way that it can cope with the growing number of users and parking spaces without any significant loss of performance.

**1.4 Maintainability:**  
The system should be easy to maintain and update, allowing for efficient bug fixes, improvements, and upgrades in the future.

**1.5 Usability:**  
The system should have a Userfriendly interface with clear instructions and intuitive navigation, ensuring ease of use by car owners and parking operators.

**1.6 Compatibility:**  
The system must be compatible with a variety of devices and platforms, including desktop computers, mobile devices, and popular web browsers.

**2. Performance Requirements:**

**2.1 Response Time:**   
The system will provide Realtime information on parking availability and respond quickly to user request

**2.2 Concurrent Users:** The system should be able to handle a large number of concurrent users without significant performance loss

**2.3 System Speed:**   
The system must be designed to provide fast website load times, minimizing wait times for users.

**2.4 Data Processing:**

The system must efficiently process and manage large amounts of parking data, ensuring quick retrieval and accurate analysis.

**3. Security Requirements:**

**3.1 User Authentication:**  
Implement secure user authentication to ensure that only authorized users can access your system.

**3.2 Data Encryption:**  
Encrypt sensitive user data such as personal information and payment details to protect against unauthorized access and data breaches.

**3.3 Secure Communication:**   
Use secure communication protocols to secure communication between users and the system.

**3.4 Access Control:**   
Apply appropriate access controls, granting users the necessary privileges based on their roles (e.g.. memberships) and responsibilities.

**3.5 Backup and Recovery:**   
Back up data regularly and develop a recovery plan to minimize loss of data.

**3.6 Compliance:**   
Comply with relevant data protection regulations and industry security standards to protect user privacy and maintain data integrity.

## 2.5 Project Scope

* **Introduction**

This project aims to develop and implement the SmartPark web application, which revolutionizes parking management and enhances the parking experience for car owners. The stakeholders involved in this project include the development team, parking lot owners and operators, and car owners in downtown areas.

* **Requirements and Goals**

The scope of this project is to create a web application that provides real-time parking space availability, allows users to search for and reserve parking spots, facilitates secure online payments, and offers user support and feedback channels. The development team will utilize modern web app frameworks and technologies to deliver a user-friendly interface and seamless user experience.

* **Deliverables**

1. Developed SmartPark web application with a pleasing UI/UX design.
2. Real-time parking space availability feature that displays vacant spaces and relevant details.
3. Search functionality allowing users to find available parking spots based on location, date, time, and duration.
4. Selection and confirmation feature enabling users to choose and confirm their reservations, with a summary of reservation details.
5. User information collection for identification and smooth parking experience.
6. Integration with a secure payment gateway to enable online payments.
7. Contact and support system for users to seek assistance, provide feedback, or make inquiries.

* **Project Acceptance Criteria**

1. The SmartPark web application meets the specified requirements and functions as intended.
2. The real-time parking space availability feature accurately displays up-to-date information.
3. Users are able to search for parking spots, make reservations, and complete payments without encountering major issues.
4. The user interface and user experience of the web application are user-friendly and visually appealing.
5. The contact and support system effectively addresses user inquiries, aids, and collects feedback.

* **Exclusions**

1. Physical infrastructure for parking management (e.g., parking sensors, gate systems).
2. Backend server infrastructure (assuming it is already in place or handled separately).
3. External services or vendors for research or outsourced services.

* **Constraints**

1. Budget limitations for development, testing, and deployment.
2. Timeframe restrictions for completing the project within the allocated 2-semester timeline.

3. Potential challenges such as communication delays, changes in scope, or technical difficulties.

* **Change Control Process**

1. Proposed modifications should be submitted to the project manager/business owner (Professor here).
2. The project manager will assess the change request and involve necessary stakeholders for evaluation.
3. Based on the assessment, the project manager will approve or deny the change request.
4. If approved, the necessary adjustments to the project plan, timeline, and resources will be made.

By following this consistent outline for the project scope statement, stakeholders can have a clear understanding of the project objectives, deliverables, constraints, and change control process, ensuring a successful implementation of the SmartPark web application.

## 2.6 System Risks

|  |  |
| --- | --- |
| **Risk** | **Response** |
| Potential system unavailability or slowdown due to cloud computing issues | Use a reliable cloud service provider offering high uptime. Implement failover mechanisms and backup systems to ensure continuous service even in case of unforeseen issues. |
| User resistance to website adoption due to privacy concerns over data collection | Develop clear and transparent data usage policies and ensure robust data security measures, including encryption and anonymization. Educate users about these measures. |
| Technical glitches or software bugs impacting system functionality | Conduct thorough testing and quality assurance procedures before deploying the system. Implement regular software updates and patches to address any identified vulnerabilities or bugs. Establish a responsive technical support team to quickly address and resolve any user-reported issues. Monitor system performance and proactively identify and fix any potential technical issues. |
| Cybersecurity threats and data breaches | Employ industry-standard encryption algorithms to secure data transmission and storage. Implement access controls and authentication mechanisms to ensure only authorized personnel can access sensitive information. Regularly conduct security audits and vulnerability assessments to identify and address any potential weaknesses. |

## 2.7 Operating Environment

SmartPark will function within diverse parking scenarios, such as multi-level parking buildings, open surface parking lots, and residential complex parking. The system is designed to handle variable weather conditions and lighting scenarios.

The software components will be hosted on secure cloud-based servers, capable of processing high volumes of real-time data. The user interface will be accessible via Web browser.

The web applications are designed to function seamlessly across various networks. To ensure the highest level of data security, all data transactions will be conducted over encrypted connections, and the system infrastructure will include robust cybersecurity measures such as firewalls and intrusion detection/prevention systems.

## 2.8 UI/UXD Interface Mock-ups

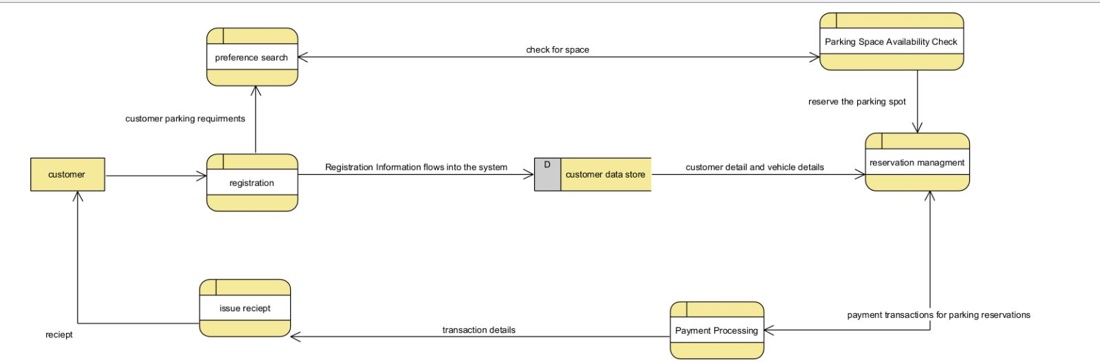
Please use the Figma template link to look Wireframe design of app or you can also look for same in Deliverables folder in Team 02 channel in MS Teams.

[Figma link to UI/UXD Interface Mock-ups](https://www.figma.com/file/CTZr8spUyJFm5b2c76Z5yN/SmartPark?type=design&node-id=0%3A1&t=cFiUG5RAnhyeFnv6-1)

# Process and Data Modeling

## **3.1 UML/DFD Modeling and Data Modeling**

### Data Flow diagram



Activity Diagrams

<https://lucid.app/lucidchart/ae21fd96-ec8f-431e-93d4-423a041a26a5/edit?viewport_loc=-1135%2C-1232%2C3821%2C1974%2C0_0&invitationId=inv_ce9721f8-8776-4eea-8bcf-0dfe27ef8e8e>

## **3.2 Use Case Specifications with corresponding interface mockups:**

**Each use case needs to have the following:**

## **3.2.1. Business Rules.**

|  |  |  |
| --- | --- | --- |
| Business Rule Number | Business Rule Description | Related UC |
| BR01 | User must provide a username, email and password to register for the website. | UC01 |
| BR02 | The system must verify availability of parking spaces in real-time. | UC02 |
| BR03 | User must reserve a parking space before entering the parking area. | UC03 |
| BR04 | Contactless payment options must be available and integrated with the system. | UC04 |
| BR05 | The system must generate a code with the parking space number after successful payment. | UC05 |
| BR06 | The system must ensure the secure handling and storage of user payment information, adhering to all relevant data security standards. | UC06 |
| BR07 | User must be informed of their reserved parking space via the email. | UC07 |
| BR08 | The system must provide suggestions for alternative parking lots in case of full occupancy. | UC08 |
| BR09 | Users should be able to cancel their parking reservation. | UC09 |
| BR10 | The system should regularly back up data to prevent data loss. | UC10 |
| BR11 | Users should receive an electronic receipt upon completion of payment. | UC11 |
| BR12 | The system should allow the user to provide feedback on their parking experience by email. | UC12 |
| BR13 | Users should be able to search and filter available parking spaces based on availability, and pricing. | UC13 |

**3.2.2. System Use Case Diagrams**

**1. Create a new user**

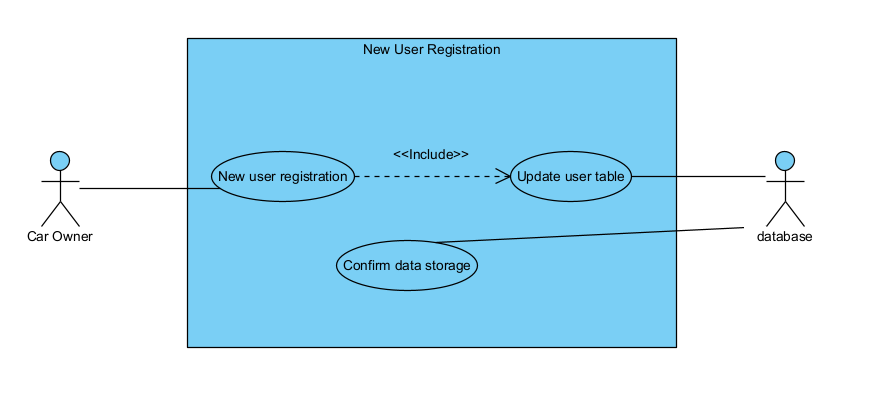
**2. Search for available parking   
  
3. Make a parking reservation  
  
4. Make an Online payment**

**5. Cancel parking reservation**

**3.2.3. Use Case Descriptions**

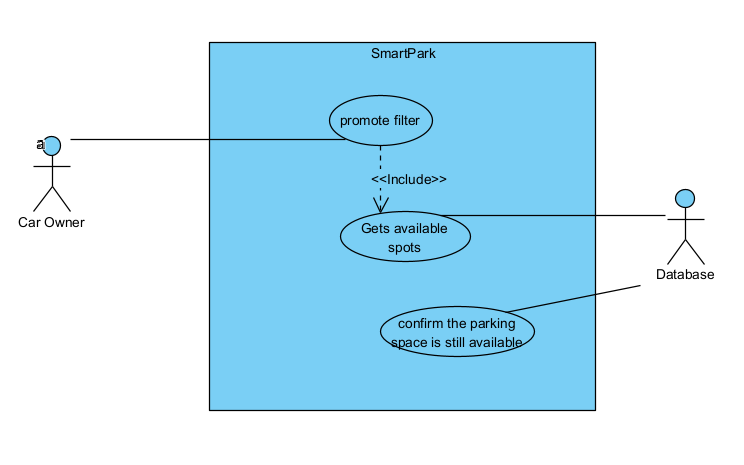
**1. Create a new user**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** New User Registration | | **ID: 1** | **Importance Level:** High |
| **Primary Actor:** Car Owner | **Use Case Type:** Detail, Essential | | |
| **Stakeholders and Interests:**  Car Owner – wants to register to use our website  SmartPark – provides registration to let car owners use the website | | | |
| **Brief Description:** This use case describes how users can register to the SmartPark’s webapp database of users | | | |
| **Trigger:** Car Owner wants to register to use the website **Type:** External | | | |
| **Relationships:**   * Association: Car Owner * Include: Update user table * Extend: * Generalization: | | | |
| **Normal Flow of Events:**   1. Car owner fills out fields for registration 2. System verifies that all required fields have been filled while registering 3. Car owner confirms all information is correct 4. System updates user table 5. System confirms data storage update 6. System sends verification link to car owner 7. System shows registration success page to the user | | | |
| **Sub Flows:** | | | |
| **Alternate/Exceptional Flows:**  2a. Failure to proceed with registration if all required fields are not entered  7a. If user found with same details, then registration is failed and asks to log-in instead | | | |



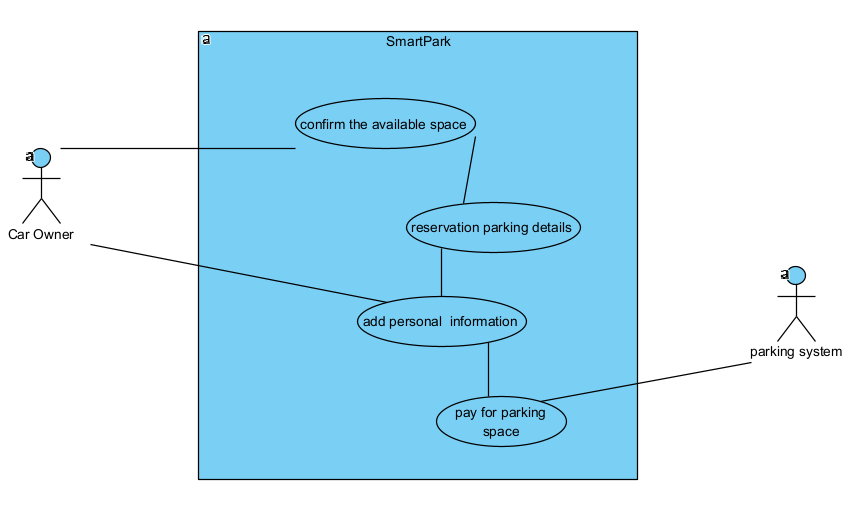
**2. Search for available parking**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** Search for available parking | | **ID: 2** | **Importance Level:** High |
| **Primary Actor:** Car Owner | **Use Case Type:** Detail, Essential | | |
| **Stakeholders and Interests:**  Car Owner – wants to search for vacant/available parking spots near them  Car owner – gets nearby spaces to park their vehicle  SmartPark – provides information on available spaces | | | |
| **Brief Description:** The car owner wants to find an available parking spot. | | | |
| **Trigger:** Car Owner wants to find parking spots on website **Type:** External | | | |
| **Relationships:**   * Association: Car Owner * Include: Gets available spots * Extend: * Generalization: | | | |
| **Normal Flow of Events:**  1. Car owner enters desired location, date, time, and duration of parking in search criteria  2. System retrieves real-time data on available parking spaces based on user search criteria  3. System displays a list of available parking spots, including their details and distance (approx.) from owners’ location | | | |
| **Sub Flows:** | | | |
| **Alternate/Exceptional Flows:**  2a. If no available parking spots are found, the system displays a message indicating unavailability. | | | |



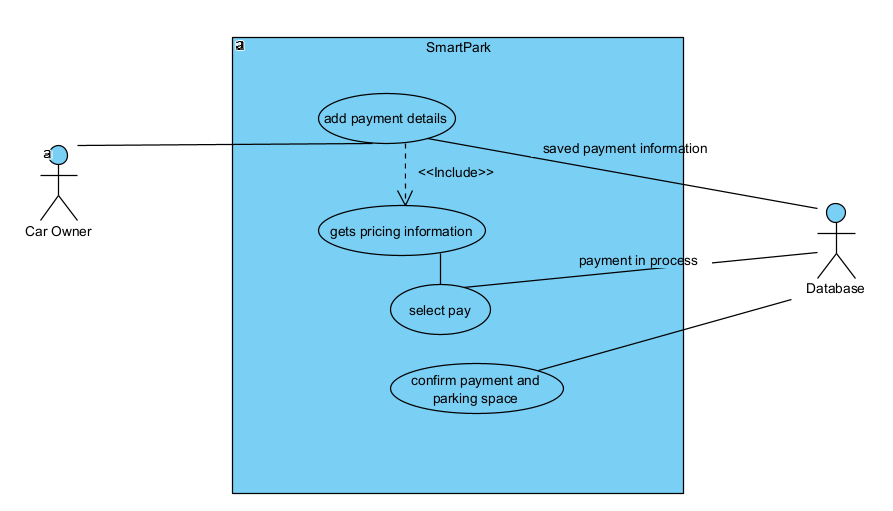
1. **Make a parking reservation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** Make a parking reservation | | **ID: 3** | **Importance Level:** High |
| **Primary Actor:** Car Owner | **Use Case Type:** Detail, Essential | | |
| **Stakeholders and Interests:**  Car Owner – wants to reserve/book an available parking spot  SmartPark – provides information on reserving a space | | | |
| **Brief Description:** The car owner wants to reserve a specific parking spot for a chosen date and time. | | | |
| **Trigger:** Car Owner wants to book a parking spot on website **Type:** External | | | |
| **Relationships:**   * Association: Car Owner * Include: * Extend: * Generalization: Adds a reservation | | | |
| **Normal Flow of Events:**  1. Car owner selects their desired parking spot from the list of available options  2. System displays a summary of the reservation details, including the selected parking spot, date, time, and pricing.  3. Car owner confirms the reservation by clicking the “Book Now” / “Confirm” button.  4. The system prompts the car owner to provide their personal information, such as name, contact details, and vehicle information.  5. The car owner submits the required information to complete the reservation. | | | |
| **Sub Flows:  S-1: Make an online payment  1. The user can make online payment with card or suitable payment service  2. Reservation gets successful** | | | |
| **Alternate/Exceptional Flows:** | | | |



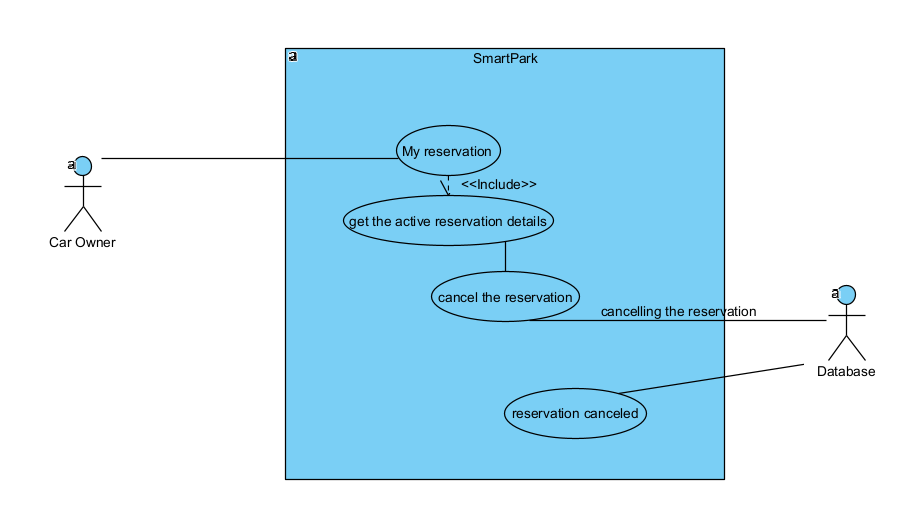
1. **Make an Online payment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** Make a parking reservation | | **ID: 4** | **Importance Level:** High |
| **Primary Actor:** Car Owner | **Use Case Type:** Detail, Essential | | |
| **Stakeholders and Interests:**  Car Owner – wants to pay for the reserved parking space  SmartPark – wants to charge amount to the user to confirm a booking | | | |
| **Brief Description:** The car owner wants to make a secure online payment for their parking reservation. | | | |
| **Trigger:** Car Owner wants to pay for the final booking of available spot **Type:** External | | | |
| **Relationships:**   * Association: Car Owner * Include: gets pricing information * Extend: * Generalization: Adds a reservation | | | |
| **Normal Flow of Events:**  1. Car owner reviews the total amount for the parking reservation  2. The system presents various payment methods, such as credit cards, digital wallets, or other accepted options.  3. The car owner selects their preferred payment method.  4. The system redirects the car owner to a secure payment gateway.  5. The car owner enters their payment details and completes the payment process. | | | |
| **Sub Flows:** | | | |
| **Alternate/Exceptional Flows:**  **5a.** Rejects payment if wrong information is given by user | | | |



1. **Cancel parking reservation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** Cancel parking reservation | | **ID: 5** | **Importance Level:** Medium |
| **Primary Actor:** Car Owner | **Use Case Type:** Detail, Essential | | |
| **Stakeholders and Interests:**  Car Owner – wants to cancel the booking he/she already have made  SmartPark – wants to release spot for other clients | | | |
| **Brief Description:** The car owner wants to cancel a previously made parking reservation. | | | |
| **Trigger:** Car Owner wants to cancel the parking reservation **Type:** External | | | |
| **Relationships:**   * Association: Car Owner * Include: get reservations details * Extend: * Generalization: Cancel a reservation | | | |
| **Normal Flow of Events:**  1. Car owner accesses their account and navigates to the "My Reservations" section.  2. The system displays a list of the car owner's active reservations.  3. The car owner selects the reservation they wish to cancel.  4. The system prompts the car owner to confirm the cancellation.  5. The car owner confirms the cancellation.  6. The system updates the reservation status to "Cancelled" and releases the parking spot. | | | |
| **Sub Flows:** | | | |
| **Alternate/Exceptional Flows:**  **5a.** If the reservation has already passed the cancellation deadline, the system may impose a cancellation fee or restrict the cancellation. | | | |



**3.3 - Corresponding Mockups**

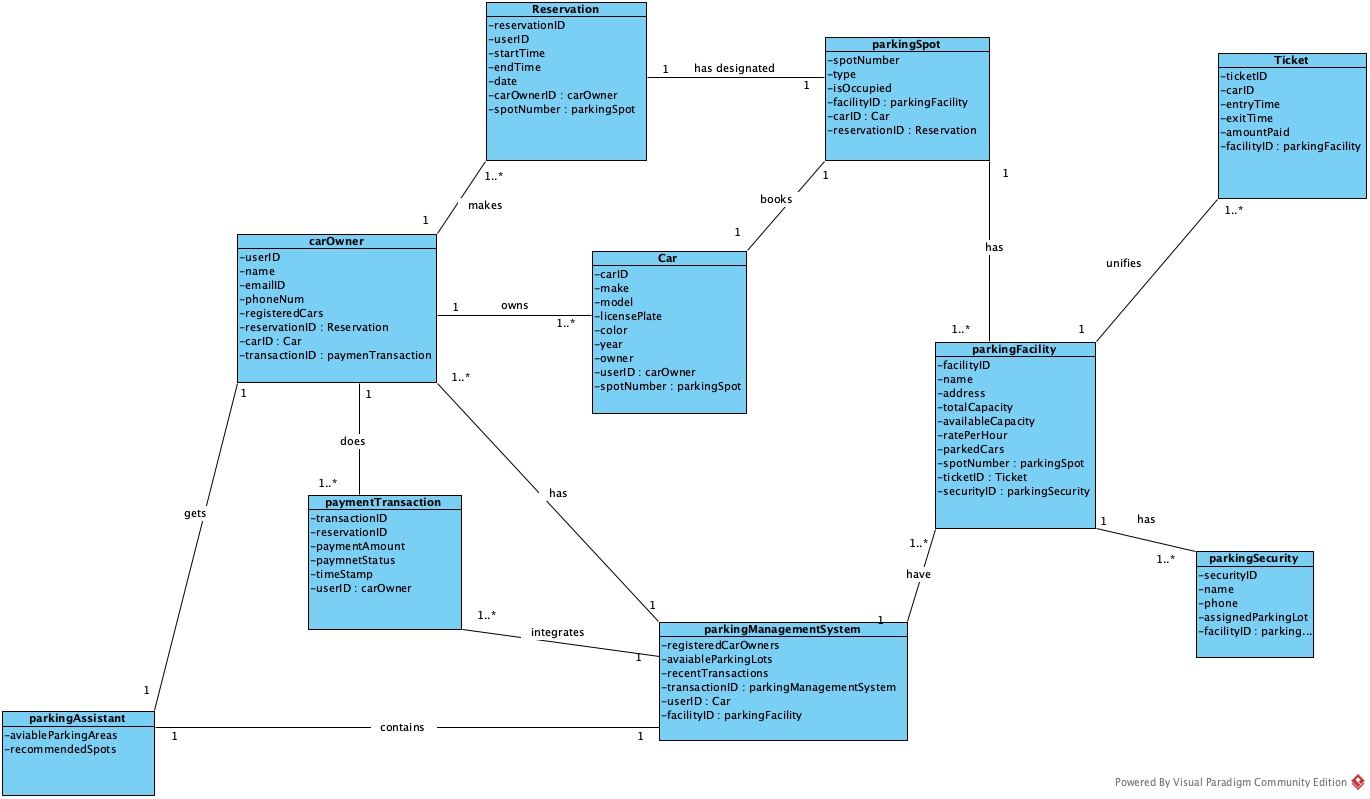
## - Presentation (.pptx) link below:

[Prototype Demo.pptx](https://seneca.sharepoint.com/:p:/s/2023-05-08PRJ566ZAA-Team02/EdTjpTVemJ9BkhlNBQoiNBEB7utlCxOGYgwfRJ_3Qv-xLQ?e=E1MYbp)

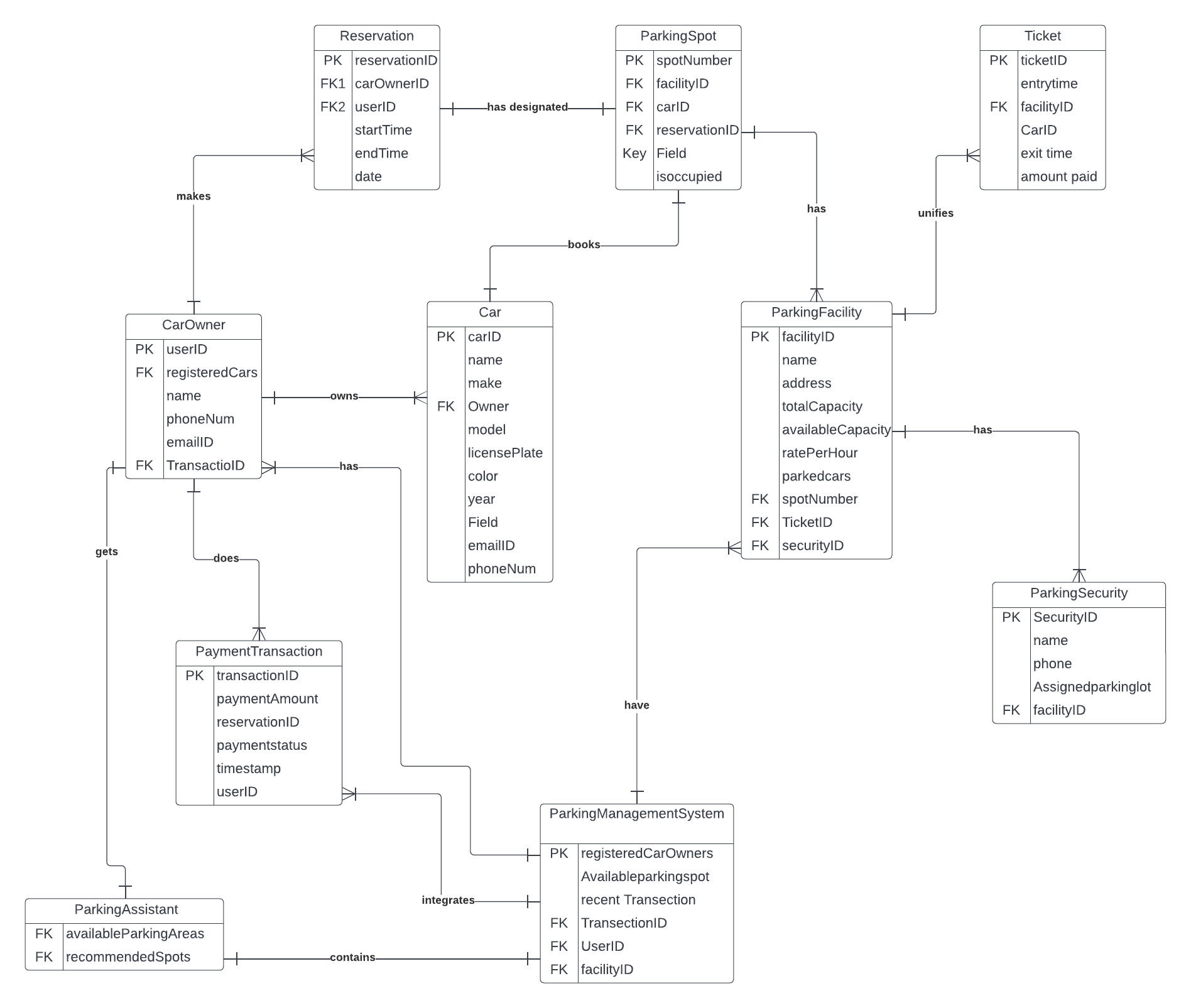
-Video Demo to Presentation link below:

<https://web.microsoftstream.com/video/664b8aef-ab11-47dd-9ceb-272b47d06a25>

# Domain Class Diagram



# Database

**ERD Diagram:**  
  


# Work Breakdown Structure (WBS)

## 

## Work Breakdown Structure

[WBS Smart Park.docx](https://seneca.sharepoint.com/:w:/s/2023-05-08PRJ566ZAA-Team02/EWmx7QGV5aNKtk-TCBUu-pwBwW9j5djuj9_6BRzBtZODRw?e=fIk7rH)

# Milestones

1. Requirements Gathering

2. Pre-development Planning

3. Implementation – Back-End

4. Implementation – Front-End

5. Quality Assurance Testing

6. User Acceptance Testing

7. Deployment

8. Support

9. Project Completion

# Acceptance Criteria

Milestone 1: Gathering Requirements

Deliverable: A document listing both the functional and non-functional needs (Software Requirements Specifications or SRS).

Acceptance: The document must be clear and thorough, reflecting input from everyone involved, and approved by both the development team and client.

Milestone 2: Planning Before Development

Deliverable: The project's Work Breakdown Structure (WBS) and Implementation Schedule in the SRS Document.

Acceptance: These plans should detail all major tasks and who will complete them, with approval required for the assigned teams.

Milestone 3: Building the Back-End

Deliverable: The database and application logic for the software.

Acceptance: This part of the software must be fully functional and integrated without major problems.

Milestone 4: Building the Front-End

Deliverable: The user interface of the software.

Acceptance: The user-facing part of the software should be complete, user-friendly, and integrated with the back-end.

Milestone 5: Testing for Quality

Deliverable: Quality checks and fixes.

Acceptance: The QA team must find and report any issues, which the developers then fix, ensuring the software complies with requirements.

Milestone 6: Testing by Users

Deliverable: A report on User Acceptance Testing (UAT).

Acceptance: End-users or the client must test the software, and the report must show that it meets user needs and requirements.

Milestone 7: Launching the System

Deliverable: The SmartPark system, fully deployed.

Acceptance: The system must be live, accessible, and functioning for its end-users.

Milestone 8: Ongoing Assistance

Deliverable: A continued support plan.

Acceptance: A plan for future maintenance and bug fixes must be drawn up and agreed upon by all parties.

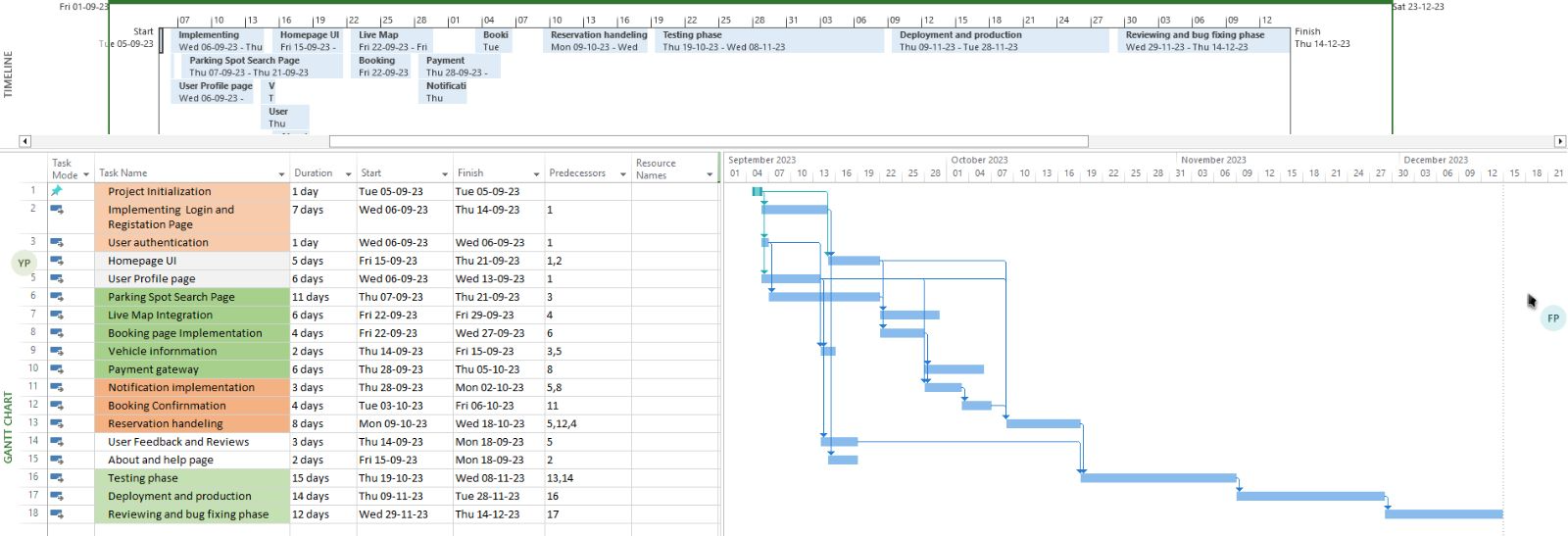
Milestone 9: Finishing the Project

Deliverable: Handing over the completed project.

Acceptance: The client must formally accept all deliverables and express satisfaction with the final product.

# Implementation Schedule

We are going to use Waterfall model for developing the project, here is the photocopy of it:



# Client / Faculty Sign-off

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

X .

Name of Client/Rep/Professor

Company Name