SUP'COM
Higher School of Communication of Tunis

جامعة قرطاج
Université de Carthage

Scope statement

Smart Park

Realized by:

Fatma Krichen & Malek Elmechi

Supervised by:

Dr. Ing. Mohamed-Bécha Kaâniche

Table of Contents

List of Figures

1- Concept

1.1 Problem

Urban areas are facing significant challenges due to the rapid increase in vehicle ownership, resulting in heightened traffic congestion and a scarcity of available parking spaces. Drivers commonly experience difficulties in locating open parking spots, which often leads to extended search times averaging around 20 minutes, increased stress, and wasted fuel. This situation not only contributes to air pollution through higher carbon emissions but also diminishes the overall efficiency of urban transportation systems. Furthermore, many existing parking solutions fail to provide real-time information on space availability, creating uncertainty for drivers and exacerbating congestion. Consequently, municipalities often struggle to optimize parking resources, leading to either underutilized or overcrowded parking facilities.

1.2 Context

The increasing urbanization of cities worldwide is accompanied by a rapid rise in the number of vehicles, leading to intensified competition for limited parking resources. Traditional parking systems often rely on outdated technologies that cannot provide the necessary real-time data to users. As cities grow, the demand for innovative solutions becomes crucial.

The integration of Internet of Things (IoT) technology, location-based services, and mobile applications can create a transformative parking management system. By deploying smart sensors, cities can gather real-time occupancy data and provide drivers with accurate information about available parking spaces. This approach not only optimizes resource allocation but also enhances the overall driving experience.

1.3 Idea

Develop a smart parking management application that utilizes IoT sensors, machine learning algorithms, location-based services, and an intuitive user interface to help users find available parking spaces in real-time. The application will leverage TinyML technology to perform analyses directly on IoT devices, such as Raspberry Pi sensors, thus reducing bandwidth needs and enabling real-time decision-making.

2- Fonctionnalities

2.1 Sensors and IoT Network

- Real-Time Vehicle Detection: Sensors detect vehicle presence in real-time to identify occupied and available parking spots.

- Data Transmission: Transmit occupancy data via MQTT to a central server for real-time processing and storage.
- Local Processing with TinyML: TinyML models run directly on IoT devices (e.g., Raspberry Pi) to predict future parking availability based on historical data and current trends.

2.2 Mobile Application

- Parking Space Display: Display nearby available parking spaces on an integrated map and show estimated distances.
- Real-Time Navigation: Provide real-time navigation to guide users to selected parking spots.
- Live Availability Updates: Display up-to-the-minute information on parking availability as detected by IoT sensors.
- Alert Notifications: Send alerts when a nearby parking spot becomes available.

3- Components

- Raspberry Pi: Functions as the brain of the system, processing data from sensors and managing communication. It runs TinyML models for local analysis, enabling quick decision-making and efficient management of parking data.
- Infrared Sensors: Provide essential input for detecting vehicle occupancy at individual parking spots. These sensors ensure accurate, real-time monitoring of parking availability to keep the system updated.
- Zigbee: A low-power, mesh network technology that facilitates reliable data transmission between sensors and the Raspberry Pi. It ensures efficient communication in IoT networks with minimal energy consumption.

4- Technologies

4.1 Backend

- MongoDB: A NoSQL database that efficiently stores and manages flexible data structures like JSON. It handles large volumes of real-time parking data with high scalability and fast query performance, allowing the Smart Park system to expand seamlessly as new parking spaces are integrated.

- MQTT: A lightweight messaging protocol ideal for IoT applications due to its minimal bandwidth usage and reliable delivery. It ensures real-time data transmission between sensors and the backend, critical for up-to-date parking availability. Its low power usage makes it perfect for IoT sensor networks.

- RESTful API: Facilitates communication between the frontend and backend by providing a structured method for accessing parking data and updates. This enhances the application's modularity and scalability while ensuring seamless integration between different components.

- WebSocket: Supports real-time, two-way communication between the server and clients. It is essential for pushing immediate updates on parking availability to users, ensuring they always have the most current information, contributing to a responsive and dynamic user experience.

4.2 Middleware

- Jakarta EE: A robust framework for building scalable and secure backend applications. It provides a rich set of APIs that simplify enterprise-level development, ensuring efficient handling of complex operations and enhancing system reliability.

- Wildfly: An application server that hosts Jakarta EE applications, offering high performance and scalability. It supports rapid deployment, robust clustering, and advanced resource management, making it suitable for large-scale IoT applications.

- Mosquitto Broker: An MQTT broker that manages message communication efficiently. It ensures lightweight and reliable transmission of data between IoT devices and the backend, which is crucial for maintaining real-time updates in smart parking systems.

4.3 Cloud

- Microsoft Azure: A comprehensive cloud platform for hosting applications and data storage. It provides high availability, security, and scalability, allowing seamless integration of backend processes and data management for Smart Park.

- Azure API Management: A tool for securing and managing APIs effectively. It offers features such as throttling, monitoring, and authentication, ensuring secure and reliable communication between different parts of the system.

4.4 Frontend

- Developing PWA mobile application: A Progressive Web Application (PWA) that offers a seamless, app-like experience on web browsers. PWAs enhance user engagement by providing offline access and faster load times, essential for a smooth user experience in smart parking

- Mapping APIs: Utilized for displaying parking locations and navigation. It provides accurate and dynamic mapping services, ensuring that users receive reliable directions and real-time updates on parking availability.

5- Architecture

The architecture of the Smart Park application is designed to facilitate seamless interaction between various components, enabling efficient parking management.

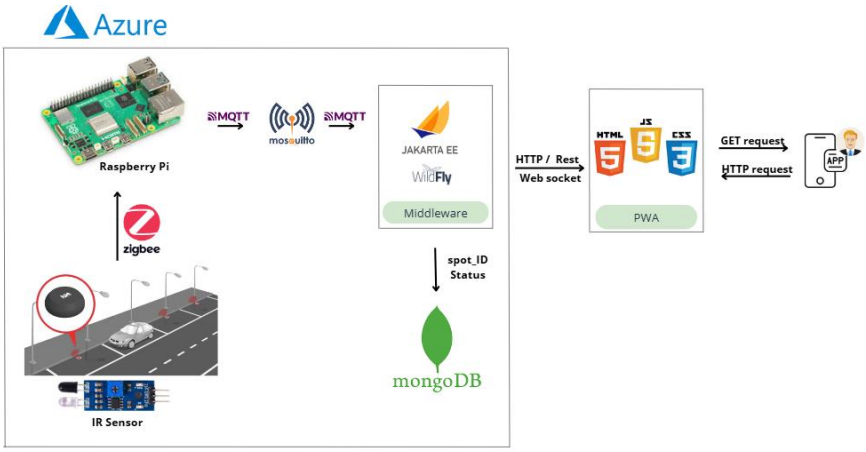

Figure 1 : Smart Park Architecture

6- Prototype

The prototype serves as a tangible representation of the Smart Park application, demonstrating its core functionalities and user interface. This initial version allows for user testing and feedback, guiding further development and enhancements.

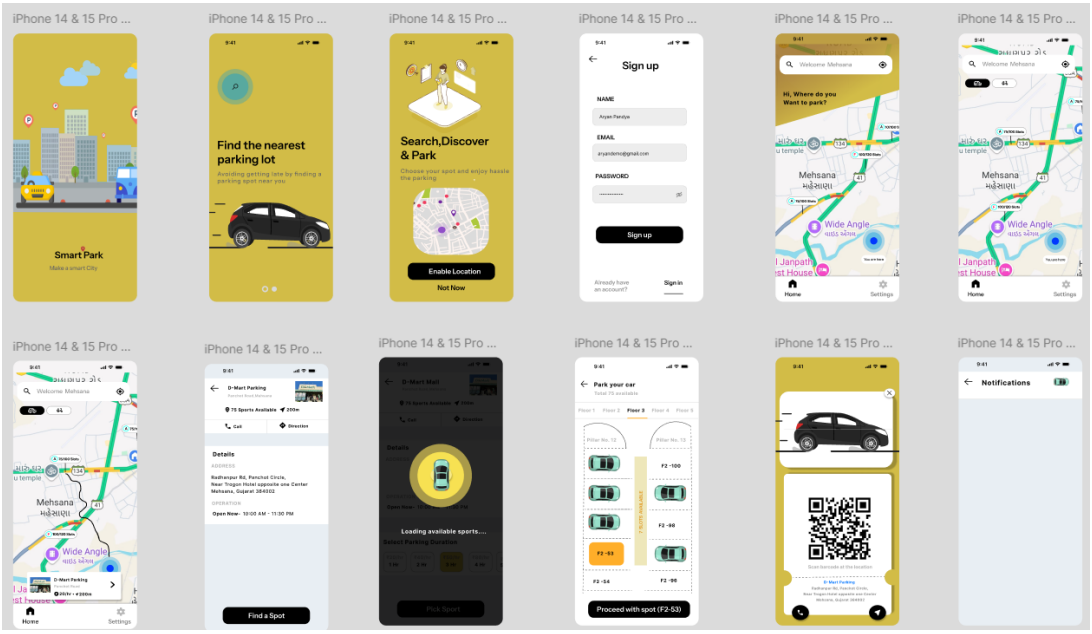

Figure 2 : Smart Park Prototype

7- The Business Model Canvas (BMC)

The Business Model Canvas serves as a strategic tool that visualizes the different components of the Smart Park business model. It outlines the key partners, activities, resources, value propositions, customer segments, channels, relationships, cost structure, and revenue streams that drive the project's success.

Figure 3 : The Business Model Canvas

8- Marketing Study

8.1 4P Marketing Mix

The 4P Marketing Mix outlines the key elements necessary for the successful marketing of the Smart Park application. It focuses on product, price, place, and promotion strategies tailored to meet the needs of the target audience.

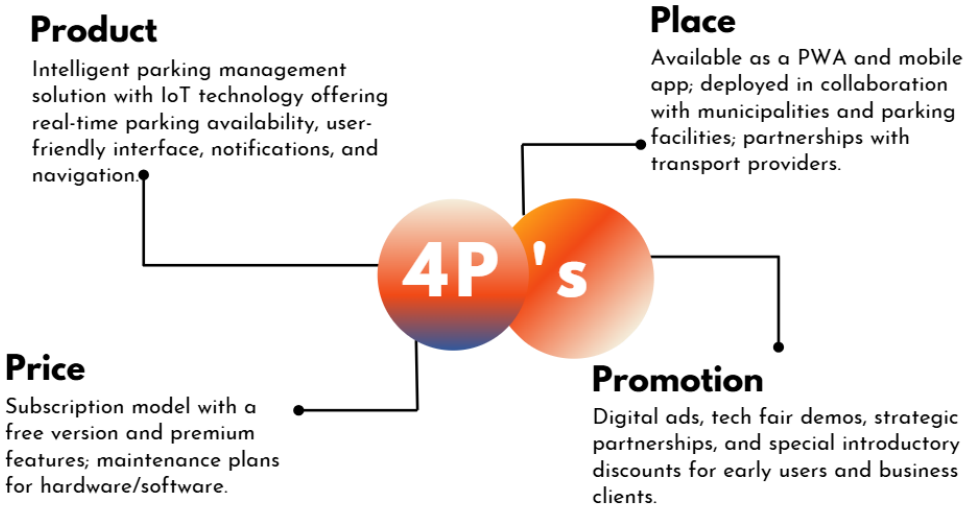

Figure 4 : 4P Marketing Mix

8.2 SWOT

The SWOT analysis provides a comprehensive overview of the internal and external factors that can impact the success of the Smart Park project. It highlights the strengths, weaknesses, opportunities, and threats related to the smart parking management application.

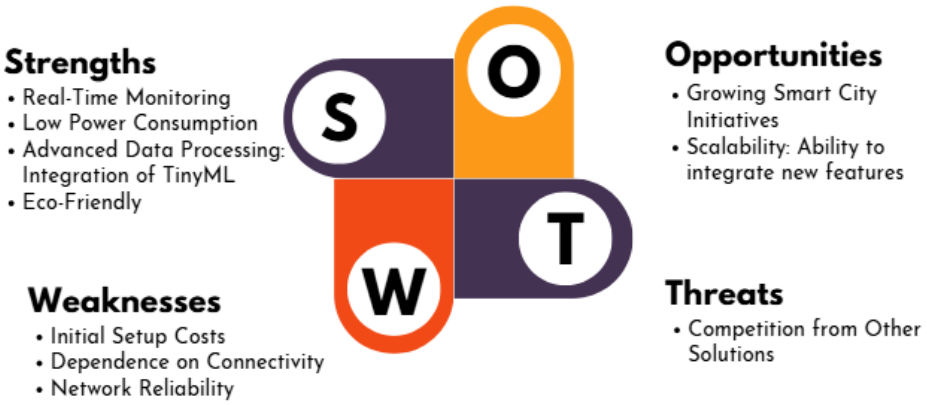

Figure 5 : Swot

9- Delivrables

At the end of the project, the following items will be delivered:

• Smart parking mobile application.

• Source code for different project components on GitHub.

• Smart parking prototype/simulation.

• Detailed report about the solution.