

[SPOTLOCATOR APPLICATION]



SPOTLOCATOR

ABSTRACT

The SPOTLOCATOR APPLICATION is an innovative mobile application designed to streamline the parking experience at Jazan University. Built for the Android platform, this application provides real-time information regarding available parking spaces across the university campus, helping users—students, faculty, and visitors—locate, reserve, and navigate to suitable parking spots.

The primary objective of this project is to improve parking management at Jazan University, thereby reducing parking-related stress and enhancing convenience for all campus members. By providing a seamless experience, the SPOTLOCATOR APPLICATION aims to enhance user satisfaction and optimize the utilization of parking spaces, contributing to a more efficient campus environment.

KEYWORDS:

Parking management, Real-time , Android application, Parking reservation, User convenience, Parking stress reduction

Table of Content

1.	INTRODUCTION.....	7
1.1	Project overview statement	7
1.2	Project Goals and Objectives	7
1.3	Project Scope	8
1.4	Limitations and Constraints	8
1.5	Assumptions.....	9
2.	PREVIOUS WORKS	10
3.	FEASIBILITY STUDY.....	14
3.1	Purpose of the Feasibility Study.....	14
3.2	Justifications for the Proposed System	14
3.3	Economic Feasibility	15
3.4	Technical Feasibility	16
3.5	Desired System Functionality	16
4.	PROJECT PLAN	17
4.1	Work Breakdown Structure	17
4.2	Activity and Task List	18
4.3	Gantt Chart.....	19
5.	SRS – SYSTEM REQUIREMENTS SPECIFICATION	20
5.1	Development Methodology	20
5.1.1	Waterfall Methodology:.....	20
	Waterfall Methodology Phases:	21
5.2	System Requirements Analysis	22
5.2.1	Functional Requirements.....	22
5.2.2	Non-Functional Requirements:.....	25
5.3	Hardware Requirements Specification	26
5.4	Software Requirements Specification	26
6.	SYSTEM DESIGN	27

6.1	Applications Design	27
a.	Data Flow Diagram	27
b.	Use Case Diagram	31
c.	Activity Diagram	33
d.	Sequence Diagrams	35
e.	Class Diagram	38
f.	Entity Relationship Diagram.....	39
7.	CONCLUSION	40
8.	REFERENCES.....	41

List of figures

Figure 1: ParkMobile	10
Figure 2: ParkWhiz	11
Figure 3: SpotHero	12
Figure 4: Work Breakdown Structure.....	17
Figure 5: Activity and Task List	18
Figure 6: Gantt Charts	19
Figure 7: Waterfall Methodology	20
Figure 8: DFD Level 0.....	28
Figure 9: DFD Level 1 for admin	29
Figure 10: DFD Level 1 for user	29
Figure 11: DFD Level 1 for police	30
Figure 12: use case for admin	31
Figure 13: use case for user	32
Figure 14: use case for police.....	32
Figure 15:Activity Diagram For admin	33
Figure 16:Activity Diagram for police.....	34
Figure 17:Activity Diagram for user	34
Figure 18: Sequence Diagram for Admin	35
Figure 19: Sequence Diagrams for Police.....	36
Figure 20: Sequence Diagrams For User	37
Figure 21: Class Diagram	38
Figure 22: Entity Relationship Diagram (ERD).....	39

List of Tables

Table 1: Economic Feasibility	15
Table 2: Hardware Requirements Specification	26
Table 3:Software	26

1. INTRODUCTION

1.1 Project overview statement

The SPOTLOCATOR APPLICATION is an Android-based mobile application designed to optimize the parking experience at Jazan University by providing real-time information on available parking spaces across the campus. This application aims to assist students, faculty, and visitors in quickly locating, reserving, and navigating to available parking spots, reducing parking-related stress and improving campus space utilization.

1.2 Project Goals and Objectives

Goal

The project's main goal is to design and develop an Android application dedicated to parking management at Jazan University.

Objectives

- ✓ **Improve Parking Management:** Streamline the process of locating and managing parking spaces at Jazan University.
- ✓ **Enhance User Experience:** Provide a user-friendly interface that helps users quickly find available parking spots.
- ✓ **Optimize Space Utilization:** Maximize the usage of available parking spaces on campus to avoid congestion and underutilization.
- ✓ **Ensure seamless user experience** on Android devices with minimal latency.

1.3 Project Scope

The SPOTLOCATOR APPLICATION aims to address the parking challenges at Jazan University by offering a real-time parking management solution on Android devices. The project scope includes the following:

- ✓ **Parking Space Availability:** The app will provide real-time data on the availability of parking spots across the university campus.
- ✓ **Reservation System:** Users will have the ability to reserve parking spaces in advance, ensuring a smooth parking experience upon arrival.
- ✓ **User Profiles:** The application will support multiple user roles (students, faculty, and visitors), each with different access levels and preferences.
- ✓ **Mobile Platform:** The app will be developed exclusively for the Android platform, making it compatible with a wide range of Android smartphones.
- ✓ **Real-time Updates:** The app will be designed to provide live updates on parking space availability, ensuring users have up-to-date information.

1.4 Limitations and Constraints

The SPOTLOCATOR APPLICATION is subject to several limitations and constraints, including:

- ✓ **Platform Limitation:** The app will be developed exclusively for Android devices, excluding iOS users from accessing the application.
- ✓ **Campus Area Coverage:** The app will only cover parking spaces available within the boundaries of Jazan University, excluding off-campus parking facilities.
- ✓ **User Dependency:** The success of the app's parking reservation and navigation features is dependent on user compliance with the reservation system and correct use of the app.

- ✓ **Language Barrier:** The website interface is currently available only in English, which may pose a challenge for non-English speaking users or those less familiar with the language.
- ✓ **Internet Usage Requirement:** All users must have basic knowledge of using the internet to effectively interact with the application and access its features.
- ✓ **Internet Connection Dependency:** The application requires a stable internet connection for real-time updates, parking reservations, and navigation features. Without internet access, the app will not function properly, limiting its usability in areas with poor or no network coverage.

1.5 Assumptions

The following assumptions underlie the limitations and constraints of the SPOTLOCATOR APPLICATION:

- ✓ **Android Device Usage:** It is assumed that the majority of the target users, including students, faculty, and visitors at Jazan University, primarily use Android devices, minimizing the impact of platform exclusivity.
- ✓ **On-Campus Parking Focus:** The parking spaces within Jazan University are sufficient for campus members, reducing the need for off-campus parking solutions.
- ✓ **User Compliance:** Users will adhere to the app's reservation and navigation features, ensuring the effective operation of the system.
- ✓ **English Proficiency:** A significant portion of the app's target users are proficient in English or capable of navigating the app interface despite the language barrier.
- ✓ **Basic Internet Skills:** All potential users have a baseline familiarity with internet usage, which is considered a standard skill set in the context of a university campus.
- ✓ **Internet Availability:** Users will have access to reliable internet connectivity, either through personal mobile data plans or Jazan University's Wi-Fi network, ensuring uninterrupted access to the app's features.

2. PREVIOUS WORKS

2.1 ParkMobile:

ParkMobile is a mobile app that allows users to find and pay for parking in urban areas. Users can search for available parking spots, view prices, and pay directly through the app. It often includes features like extending parking sessions remotely and receiving notifications when the time is about to expire. ParkMobile partners with various municipalities and private parking operators to provide comprehensive coverage.

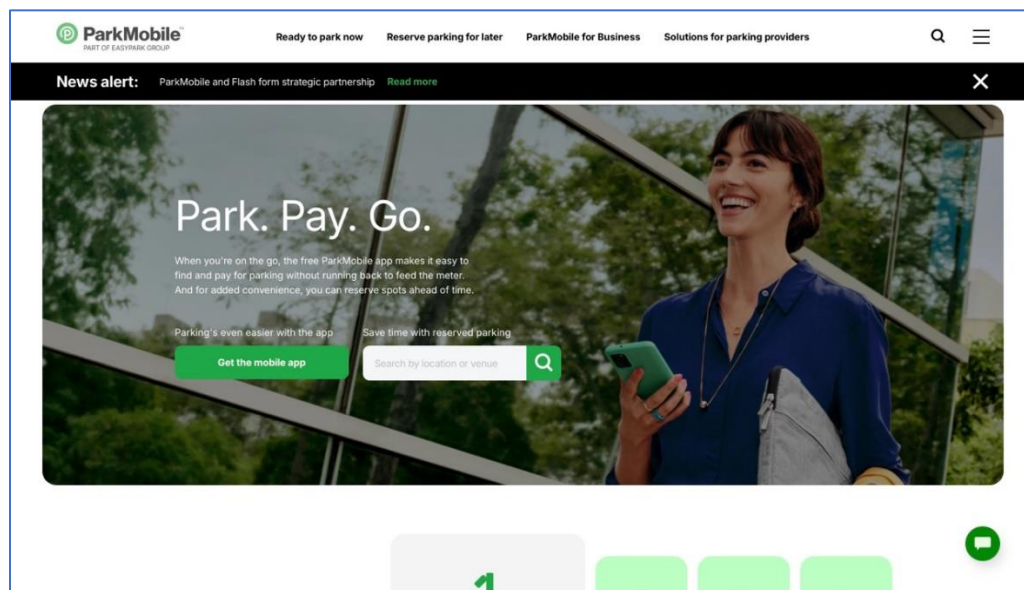


Figure 1: ParkMobile

Advantages: of this website are:

- Flexible options: ParkMobile allows users to reserve spots in advance or find available parking when they arrive.
- Wide coverage: It works across various cities, including specific locations like airports and event venues, offering convenience for frequent travelers.
- Integration: Integrates with parking meters for digital payments and curbside management.

Disadvantages:

- Pricing confusion: Some users report occasional difficulties understanding pricing structures, particularly when fees aren't always clear upfront.
- Technical issues: There are reports of occasional glitches with booking or finding parking, causing frustration.

2.2 ParkWhiz

ParkWhiz is a parking reservation service that helps users find and book parking spots in advance, particularly in busy urban areas or near event venues. The app allows users to compare prices and options from various lots and garages, and often offers discounts for pre-booked spaces. It provides features like instant booking and navigation directions to the selected parking location.

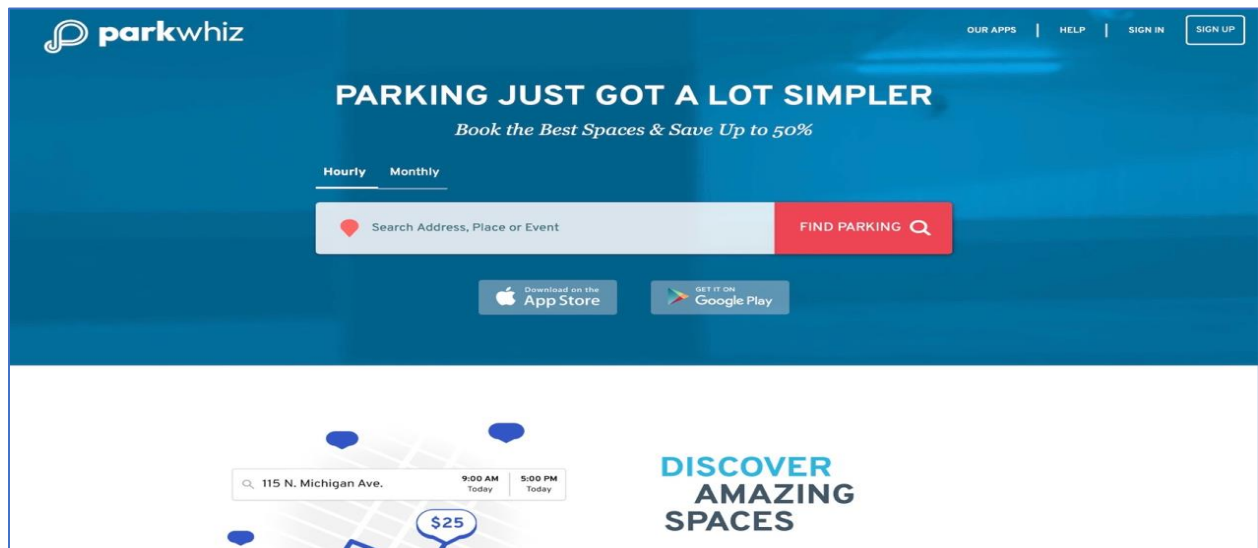


Figure 2: ParkWhiz

Advantages: of this website are:

- Discounted pricing: ParkWhiz helps users save up to 50% on parking rates by prepaying and booking in advance.
- No surprises: The app shows the total cost upfront, preventing unexpected charges.

- Simple interface: Users praise the ease of reserving a parking spot and avoiding the hassle of searching for parking on arrival.

Disadvantages: of this website are:

- Customer service issues: Some users have reported dissatisfaction with customer support, particularly when handling booking errors or garage closures.
- Coverage gaps: While it works in major cities, users in smaller areas may find limited availability.

2.3 SpotHero

SpotHero is another parking reservation platform that enables users to search for, compare, and book parking spots ahead of time. Users can find discounted rates at nearby garages, lots, and on-street parking. SpotHero also offers a user-friendly interface for managing reservations and provides directions to the parking spot. It often includes special deals for events, making it convenient for concert-goers and sports fans.

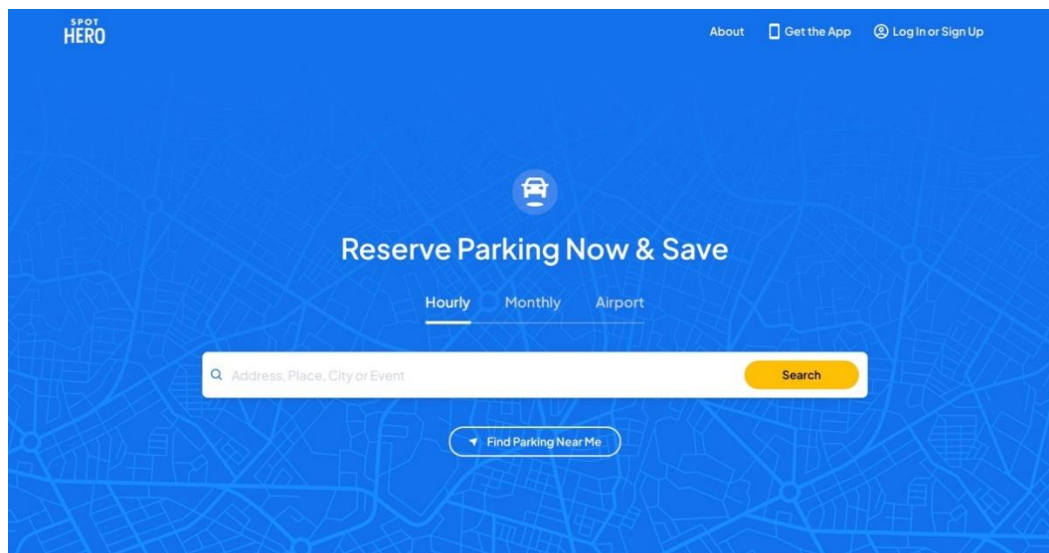


Figure 3: SpotHero

Advantages:of this website are:

- Large network: SpotHero offers parking options across the U.S., making it convenient for travelers and locals alike.
- Real-time availability: Users can see real-time availability, reducing the risk of arriving and not finding a spot.
- Easy extensions: It allows users to extend their parking session without returning to the lot.

Disadvantages:of this website are:

- User experience: Some users have experienced app glitches or difficulty finding booked spots, especially in busy locations.
- Cancellation policies: There are occasional issues with refunds or cancellations, especially for last-minute changes.

3. FEASIBILITY STUDY

3.1 Purpose of the Feasibility Study

The purpose of this feasibility study is to evaluate the practicality and potential success of the SPOTLOCATOR APPLICATION in addressing parking challenges at Jazan University. It aims to assess the technical, economic, and functional aspects of the project to determine its viability. By analyzing the feasibility, this study ensures that the proposed system meets the requirements of users effectively while being realistic and achievable within the given constraints.

3.2 Justifications for the Proposed System

The SPOTLOCATOR APPLICATION is justified for the following reasons:

- ✓ **Addressing Parking Challenges:**

Parking congestion is a persistent issue at Jazan University, causing delays and stress among students, faculty, and visitors. The system will provide a streamlined solution to this problem.

- ✓ **Efficiency in Space Utilization:**

By enabling real-time monitoring and reservations, the system optimizes the use of parking spaces, reducing instances of overutilization or underutilization.

- ✓ **Enhanced User Experience:**

The app provides a convenient and stress-free method to locate and reserve parking spots, saving users time and effort.

- ✓ **Alignment with Technological Advancements:**

With the growing reliance on mobile applications for daily tasks, introducing a digital parking management solution is both timely and relevant.

3.3 Economic Feasibility

The economic feasibility assesses whether the financial investment in developing the SPOTLOCATOR APPLICATION is justified by its potential benefits:

- **Cost-Benefit Analysis:**
 - **Costs:** The project involves development costs (developer salaries, testing, hosting services, and maintenance), training for users, and marketing expenses.
 - **Benefits:** Enhanced parking efficiency, reduced fuel consumption due to less searching for parking spots, and improved user satisfaction.
- **Return on Investment (ROI):**

By improving parking management, the application can save users time and reduce operational inefficiencies, leading to long-term cost savings for the university.
- **Revenue Streams:**

If desired, the app could integrate features like premium reservations or advertisements, creating opportunities for additional revenue.
- In terms of cost analysis, the newly proposed system necessitates :

Item	Cost
Server(computer with high specifications)	4000 SR
000webhost	200 SR at year
Internet Connection	100 SR Monthly

Table 1: Economic Feasibility

3.4 Technical Feasibility

Particularly in terms of the required hardware, software components and technologiesLet's break down the key points mentioned:

Technology Stack:

- Frontend:
 - Dart language
 - Flutter framework
 - pub.dev packages
- Backend (Server-Side):
 - PHP
 - MySQL Server as the backend database
- Development Environment:
 - Android Studio

We have already studied this technique and PHP is free and open source , MYSQL, android studio available for free , So The project is technically feasible.

3.5 Desired System Functionality

- ✓ **Real-Time Parking Availability:**Users can view the availability of parking spaces across Jazan University.
- ✓ **Reservation System:**Users can reserve parking spots in advance for a hassle-free experience.
- ✓ **Live Updates:**Continuous synchronization with the backend to ensure data accuracy.
- ✓ **User-Friendly Interface:**Intuitive design with easy navigation, ensuring accessibility for all user groups.

4. PROJECT PLAN

A project plan is a formal document that outlines the objectives, tasks, timeline, resources, and strategies required to achieve a specific project goal. It serves as a blueprint for the entire project, ensuring that everyone involved is aligned and working towards a common objective

4.1 Work Breakdown Structure

A Work Breakdown Structure (WBS) is a hierarchical decomposition of a project into smaller, more manageable components, tasks, and deliverables. It breaks down the project work into distinct, manageable sections, providing a visual representation of the project's scope and structure.

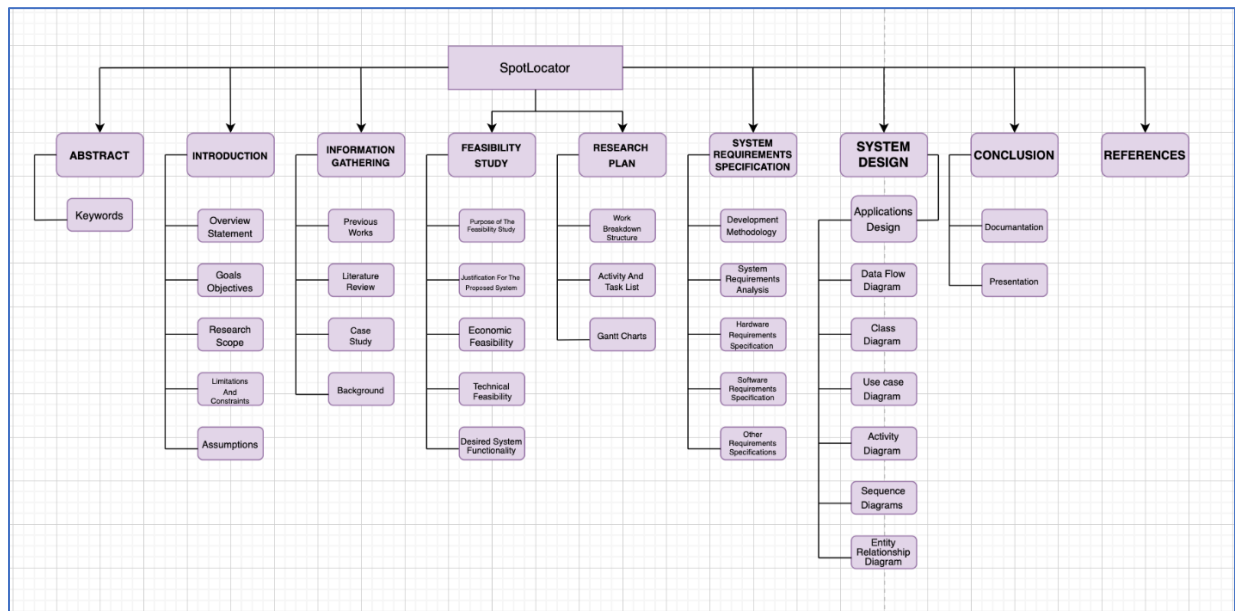


Figure 4: Work Breakdown Structure

4.2 Activity and Task List

The project involves several activities and tasks, each with its own timeline and dependencies. The Gantt chart is a useful tool for visualizing the project schedule and ensuring that the project is completed on time and within budget.



9	Task Name	Duration	Start	Finish	% complete
10	SpotLocator	50 days	9/16/2024	11/22/2024	100%
11	INTRODUCTION	32 days	9/16/2024	10/29/2024	100%
12	Project Overview Statement	2 days	9/16/2024	9/17/2024	100%
13	Project Goals and Objectives	1 day	9/18/2024	9/18/2024	100%
14	Project Scope	2 days	10/28/2024	10/29/2024	100%
15	Limitations and Constraints	2 days	9/23/2024	9/24/2024	100%
16	Assumptions	3 days	9/25/2024	9/27/2024	100%
17	INFORMATION GATHERIN	7 days	9/30/2024	10/8/2024	100%
18	PREVIOUS WORKS	2 days	9/30/2024	10/1/2024	100%
19	LITERATURE REVIEW	1 day	10/2/2024	10/2/2024	100%
20	CASE STUDY	2 days	10/3/2024	10/4/2024	100%
21	BACKGROUND	2 days	10/7/2024	10/8/2024	100%
22	FEASIBILITY STUDY	8 days	10/11/2024	10/22/2024	100%
23	Purpose of The Feasibility Study	1 day	10/11/2024	10/11/2024	100%
24	Justification For The Proposed System	3 days	10/14/2024	10/16/2024	100%
25	Economic Feasibility	2 days	10/16/2024	10/17/2024	100%
26	Technical Feasibility	2 days	10/21/2024	10/22/2024	100%
27	PROJECT PLAN	4 days	10/25/2024	10/30/2024	100%
28	Work Breakdown Structure	1 day	10/25/2024	10/25/2024	100%
29	Activity and Task List	3 days	10/28/2024	10/30/2024	100%
30	Gantt Charts	3 days	10/28/2024	10/30/2024	100%
31	SRS - SYSTEM REQUIREMENTS SPECIFICATION	5 days	11/4/2024	11/8/2024	100%
32	System Requirements Analysis	2 days	11/4/2024	11/5/2024	100%
33	Hardware Requirements Specification	1 day	11/7/2024	11/7/2024	100%
34	Software Requirements Specification	1 day	11/8/2024	11/8/2024	100%
35	SYSTEM DESIGN	9 days	11/12/2024	11/22/2024	100%
36	Applications Design	2 days	11/12/2024	11/13/2024	100%
37	Data Flow Diagram	2 days	11/14/2024	11/15/2024	100%
38	Class Diagram	1 day	11/18/2024	11/18/2024	100%
39	Use case Diagram	2 days	11/19/2024	11/20/2024	100%
40	Activity Diagram	1 day	11/20/2024	11/20/2024	100%
41	Sequence Diagrams	1 day	11/21/2024	11/21/2024	100%
42	Entity Relationship Diagram	1 day	11/22/2024	11/22/2024	100%
43	CONCLUSION	2 days	11/19/2024	11/20/2024	100%
44	REFERENCES	1 day	11/22/2024	11/22/2024	100%

Figure 5: Activity and Task List

4.3 Gantt Chart

Gantt Charts are visual representations of project schedules that illustrate tasks, activities, and their respective timelines. These charts display tasks as horizontal bars along a timeline, showcasing their start and end dates, durations, and dependencies. Gantt Charts provide a clear overview of project progress, enabling project managers to track activities, manage dependencies, allocate resources, and monitor timelines. They facilitate effective project planning, scheduling, and communication among team members by visualizing the project's workflow and progress.

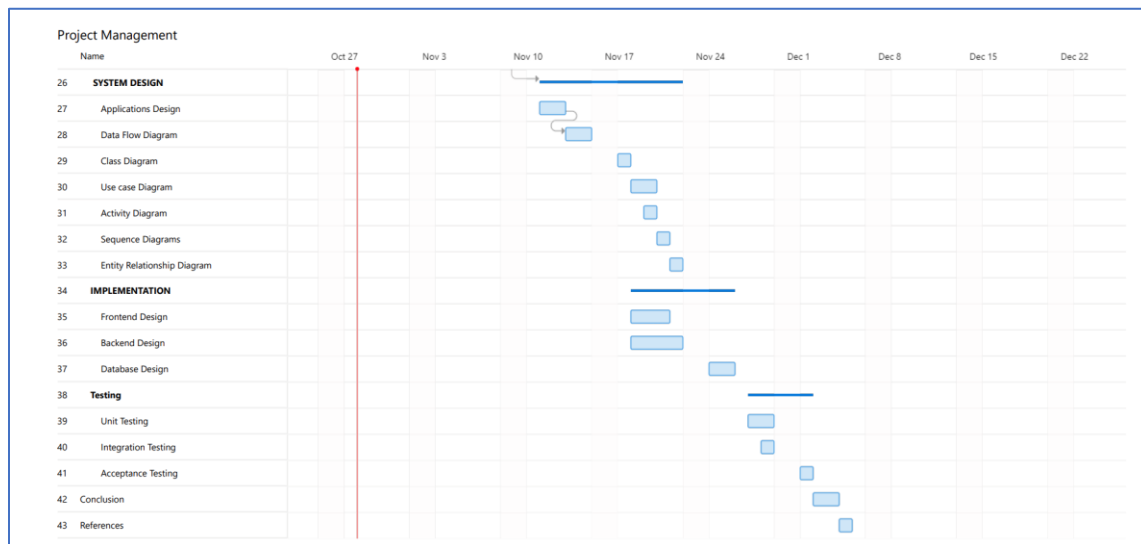


Figure 6: Gantt Charts

5. SRS – SYSTEM REQUIREMENTS SPECIFICATION

A System Requirements Specification (SRS) outlines the expected behavior, functions, and performance of a system. It serves as a blueprint for the development team, ensuring that everyone involved understands the project's requirements and goals.

5.1 Development Methodology

The development methodology refers to the approach or framework used to plan, structure, and control the process of developing an information system. It provides guidelines and practices for the efficient and effective creation of software or other systems. There are various methodologies, and the choice often depends on the project's specific requirements and characteristics .

5.1.1 Waterfall Methodology:

The Waterfall methodology is a linear project management approach that progresses sequentially through defined phases from initiation to completion, ensuring a systematic and orderly progression of project tasks.

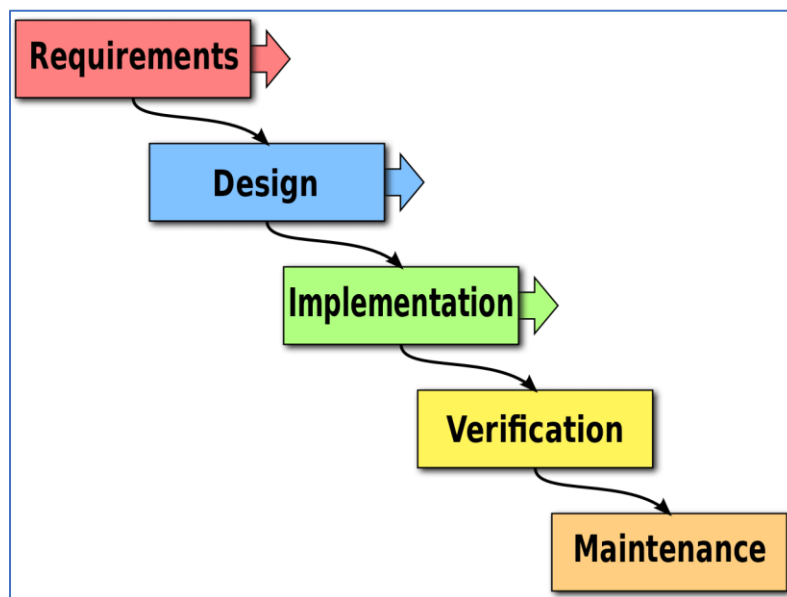


Figure 7: Waterfall Methodology

Waterfall Methodology Phases:

Requirements:

- Emphasizes thorough gathering and understanding of all project requirements upfront.
- Detailed documentation is utilized to describe project stages, costs, risks, success metrics, and timelines.

Design:

- Involves designing technical solutions based on identified project requirements, scenarios, layouts, and data models.
- Transitions from a higher-level or logical design to a physical design using specific technologies

Implementation:

- Technical implementation phase where coding and development begin based on the detailed design.
- Due to comprehensive prior research and design efforts, this phase tends to be shorter.

Verification or Testing:

- Testing phase aimed at ensuring an error-free product and validation against the defined requirements.
- Uses design documents and user case scenarios as a basis for comprehensive testing.

Maintenance:

- Post-deployment phase involving ongoing updates, defect handling, and new software releases based on user feedback.
- Ensures the system's stability, addresses any post-deployment issues, and aligns with evolving user need

5.2 System Requirements Analysis

System Requirements Analysis is a crucial phase in both systems engineering and software development. It involves identifying, documenting, and managing the needs and conditions required for a new or altered system 14.

5.2.1 Functional Requirements

System Administrator

1. Show/Add/Edit User Account

- **Show User Account:**
 - The admin can view the details of registered users, including their name, email, and other relevant information.
- **Add User Account:**
 - The admin can create a new user account by entering the user's personal details such as name, email, and plate number.
- **Edit User Account:**
 - The admin can modify the details of an existing user account, such as updating the email, name, or plate number.

2. Add/Edit Parking Spaces

- **Add Parking Space:**
 - The admin can add new parking spaces to the system, specifying the location and availability status.
- **Edit Parking Space:**
 - The admin can modify the details of existing parking spaces, such as updating availability or renaming locations.

3. View Parking

- The admin can view a detailed overview of:
 - Available parking spaces.
 - Reserved spaces along with details of the user accounts (e.g., name and plate number) associated with the reservation.

4. Add/Delete Parking Police Account

- **Add Parking Police Account:**
 - The admin can create accounts for parking police by providing necessary details like name and email.
- **Delete Parking Police Account:**
 - The admin can remove existing parking police accounts when no longer needed.

System User

1. Registration

- Users can create an account by providing:
 - Name
 - Email address
 - Password
 - Vehicle plate number

2. Login

- Users can log into the system using their registered email and password.

3. View Parking

- Users can see:
 - Available parking spaces.
 - Reserved spaces (including their own reservations).

4. Reserve a Parking Space

- Users can:
 - Select an available parking space.
 - Confirm the reservation to ensure the space is allocated to them.

5. Cancel Reservation

- Users can cancel their existing reservation through the app, making the parking space available for others.

Parking Police

1. View Parking

- Parking police can access detailed information about parking spaces, including:
 - Available spaces.
 - Reserved spaces with the associated user details (e.g., name and plate number).

5.2.2 Non-Functional Requirements:

1. Performance :

- Response Time: less than 2 seconds.
- Scalability: Handle increased user loads seamlessly.

2. Security :

- Data Encryption: Encrypt sensitive user data.
- User Authentication: Strong authentication measures.
- Compliance: Adhere to healthcare regulations (HIPAA, GDPR).

3. Reliability :

- System Uptime: High availability.(%99.9)
- Fault Tolerance: Gracefully handle errors.

4. Usability :

- Accessibility: Follow WCAG guidelines.
- User Interface: Intuitive design.

5. Compatibility :

- Cross-Browser Compatibility.
- Device Compatibility.

6. Maintainability :

- Code Maintainability: Clean, well-documented code.
- System Updates: Seamless update processes.

5.3 Hardware Requirements Specification

No	Item	Description
1	laptop	Core i7 or higher
2	processor	3.0GHz or higher
3	RAM	2 GB or higher
4	Hard Disk	500 GB or higher
5	Server	With medium specification
	Smart phone	With android operating system

Table 2: Hardware Requirements Specification

5.4 Software Requirements Specification

Name	description
Android studio	The smart assistant application was built using the Android Studio program to design interactive interfaces and back-end programming, which was in the Dart language.
Dart	A programming language developed by Google, commonly used for building web, server, and mobile applications. It's particularly associated with Flutter for creating cross-platform mobile apps.
Flutter	Google's UI toolkit used for building natively compiled applications for mobile, web, and desktop from a single codebase.
VSCODE	The vscode program was used to write PHP codes that act as an intermediary to perform operations between the external database and the application.
PHP(MYSQL)	php(MYSQL) was used to create the database for the project
Microsoft Word	Word was used to write forms and reports for the project

Table 3:Software

6. SYSTEM DESIGN

System Design refers to the process of defining the architecture, components, modules, interfaces, and data of a system to satisfy specified requirements. It involves creating a blueprint for how the system will function, both at the high level (overall architecture) and at the detailed level (individual components and their interactions).

6.1 Applications Design

a. Data Flow Diagram

A **Data Flow Diagram (DFD)** is a graphical representation that illustrates the flow of data within an information system. It provides a clear visualization of how data moves between processes, data stores, and external entities, making it easier for both technical and non-technical users to understand the system's functionality.

1. DFD Level 0 (Context Level)

Level 0 is the highest-level Data Flow Diagram (DFD), which provides an overview of the entire system. It shows the major processes, data flows, and data stores in the system, without providing any details about the internal workings of these processes. It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows

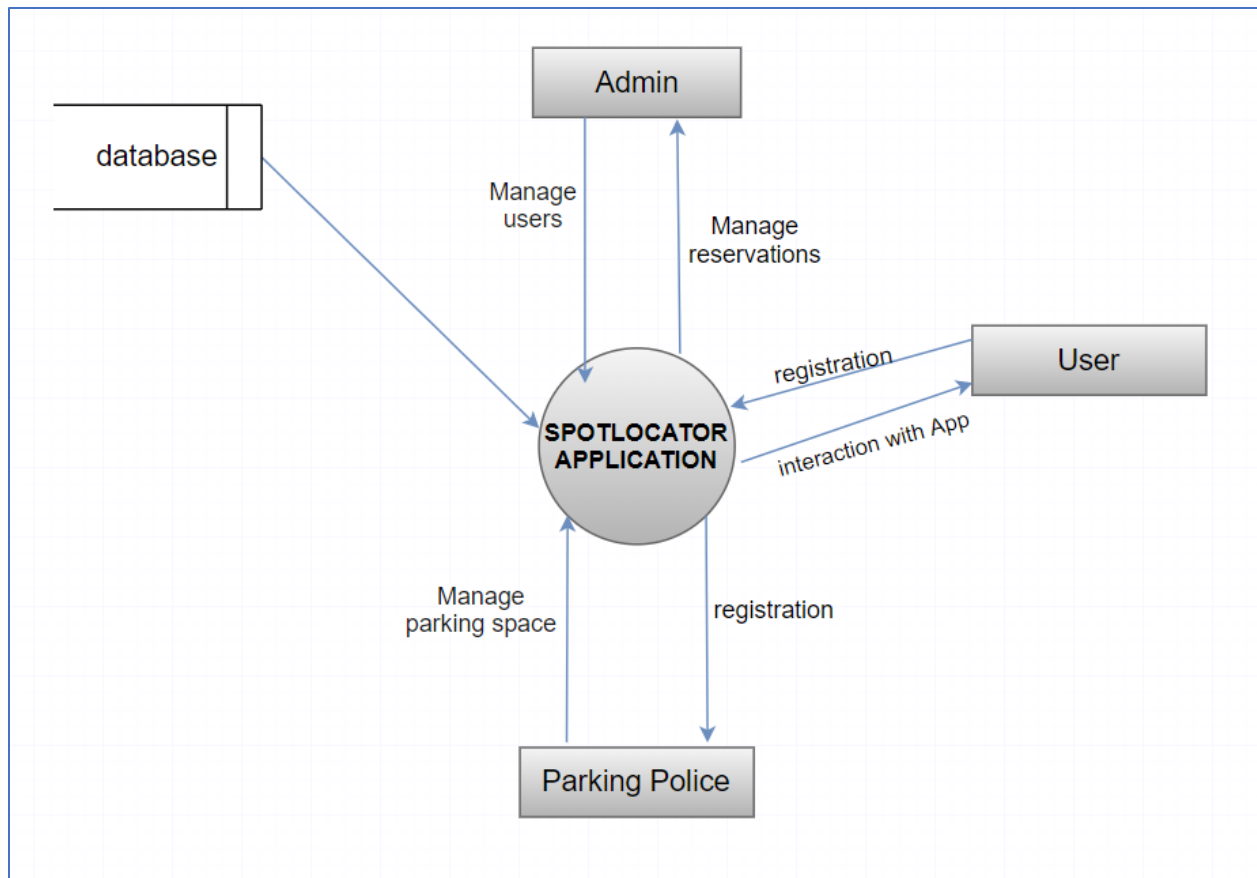


Figure 8: DFD Level 0

2. DFD Level 1

1-Level provides a more detailed view of the system by breaking down the major processes identified in the level 0 Data Flow Diagram (DFD) into sub-processes. Each sub-process is depicted as a separate process on the level 1 Data Flow Diagram (DFD). The data flows and data stores associated with each sub-process are also shown. In 1-level Data Flow Diagram (DFD), the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level Data Flow Diagram (DFD) into subprocesses.

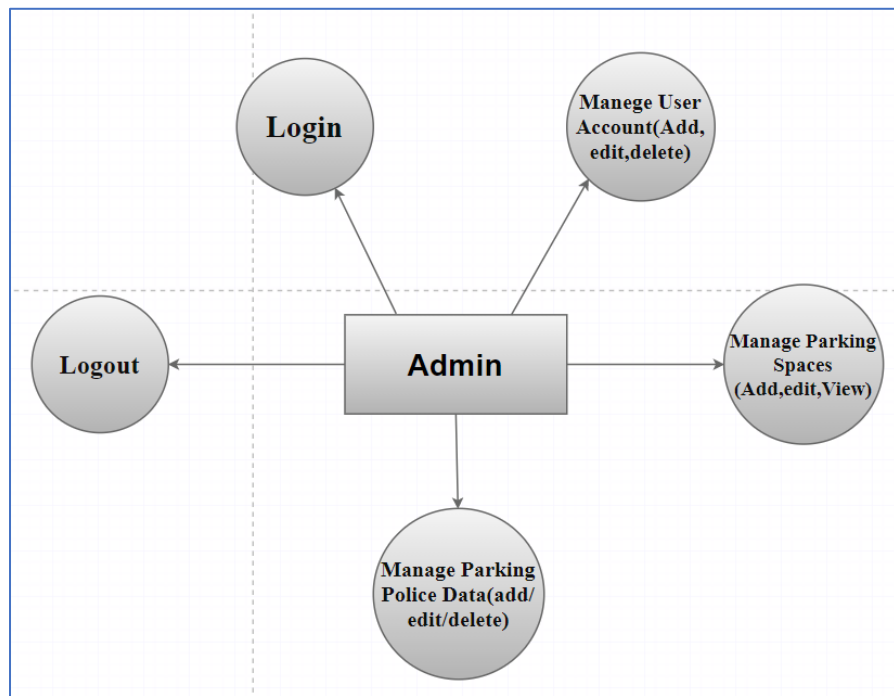


Figure 9: DFD Level 1 for admin

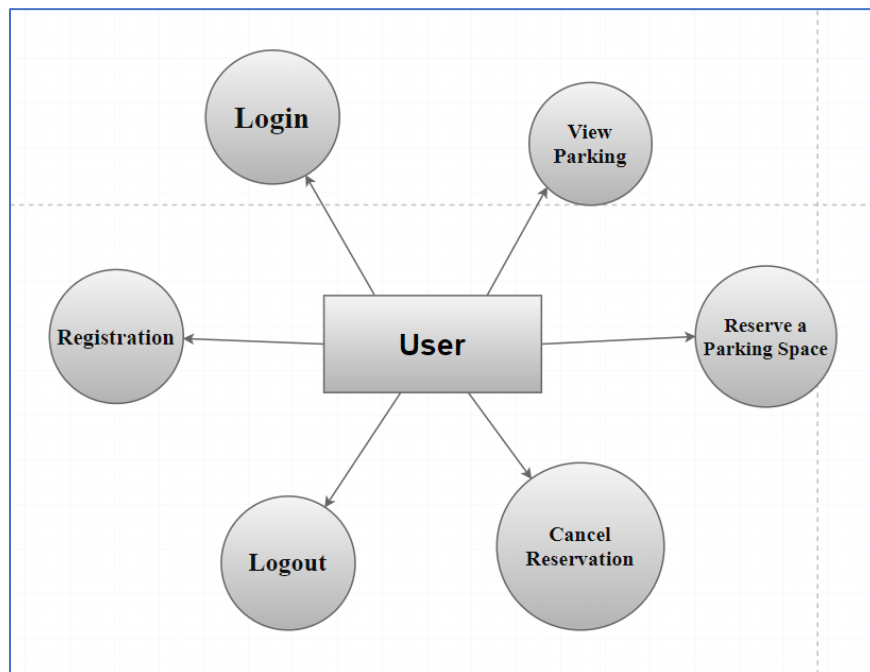


Figure 10: DFD Level 1 for user



Figure 11: DFD Level 1 for police

b. Use Case Diagram

A **Use Case Diagram** is a visual representation in UML (Unified Modeling Language) that illustrates the interactions between users (actors) and a system. It focuses on defining the functionality of the system by highlighting the goals or actions (use cases) the system provides to its users without delving into the implementation details. Key components include **actors**, which represent entities interacting with the system, and **use cases**, which are tasks or functions the system performs to meet user goals. The **system boundary** defines the scope of the system, and various **relationships** such as association, include, extend, and generalization connect actors to use cases or relate use cases to one another. Use Case Diagrams are particularly valuable for understanding system requirements, defining scope, and facilitating clear communication between developers, stakeholders, and users.

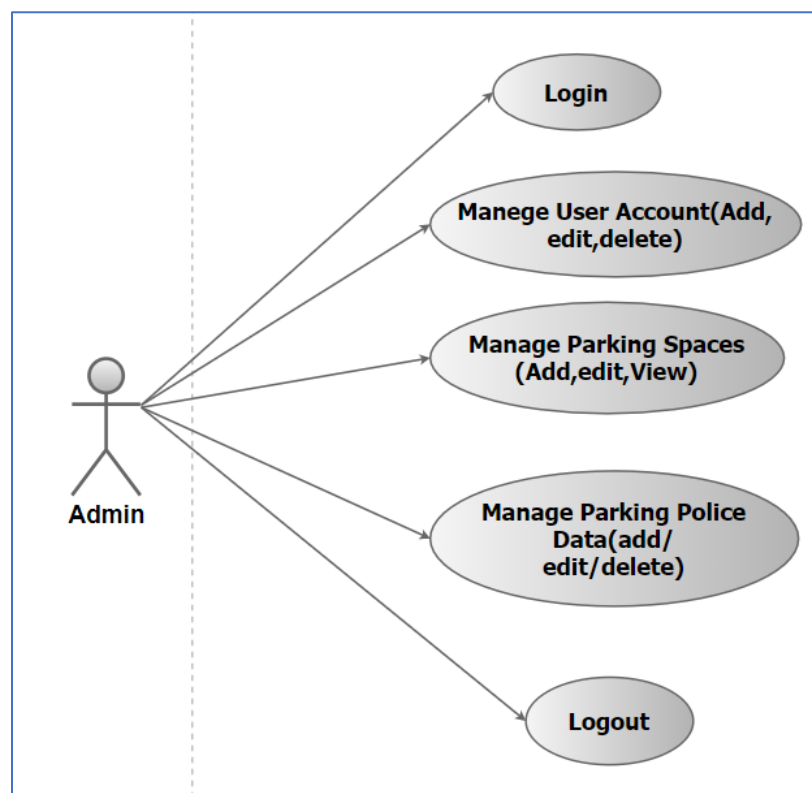


Figure 12: use case for admin

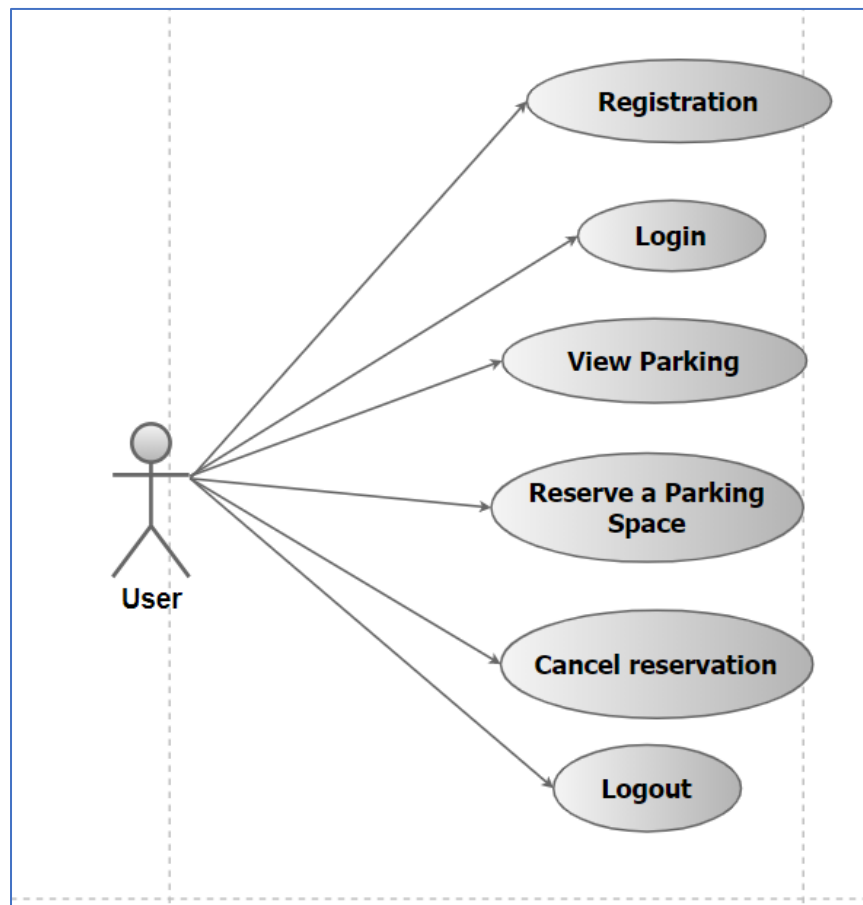


Figure 13: use case for user

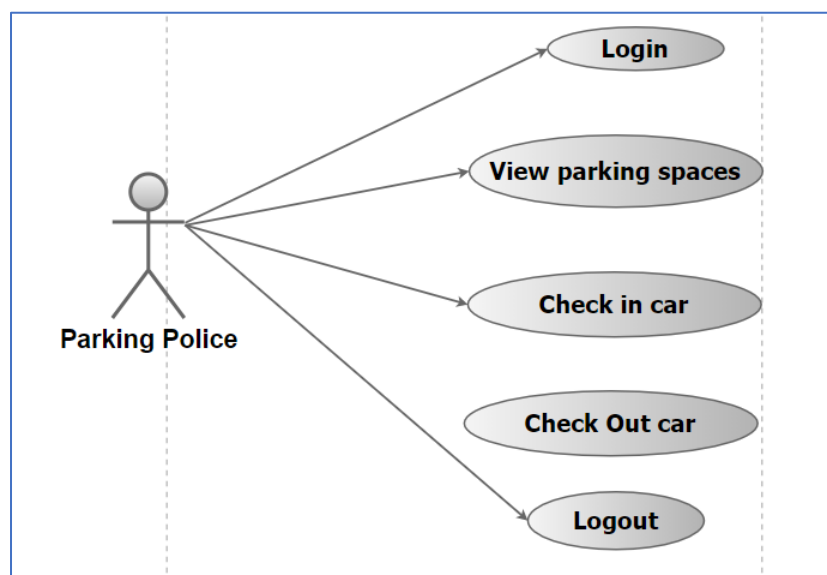


Figure 14: use case for police

c. Activity Diagram

An **Activity Diagram** is a UML (Unified Modeling Language) diagram that models the workflow or business processes of a system, focusing on the sequence of activities and decision points. It provides a detailed view of the control flow and behavior of the system by representing activities, decisions, parallel processes, and conditions. Key components include **activities** (represented as rounded rectangles), which indicate specific actions or tasks, and **control flows** (arrows) that show the progression between activities. Decision points, depicted as diamonds, are used to represent branching based on conditions, while start and end nodes mark the beginning and completion of the workflow. Activity Diagrams can also illustrate concurrent activities with synchronization bars. They are particularly useful for visualizing dynamic aspects of a system, optimizing workflows, and clarifying complex processes, making them a valuable tool for both developers and business analysts during system design.

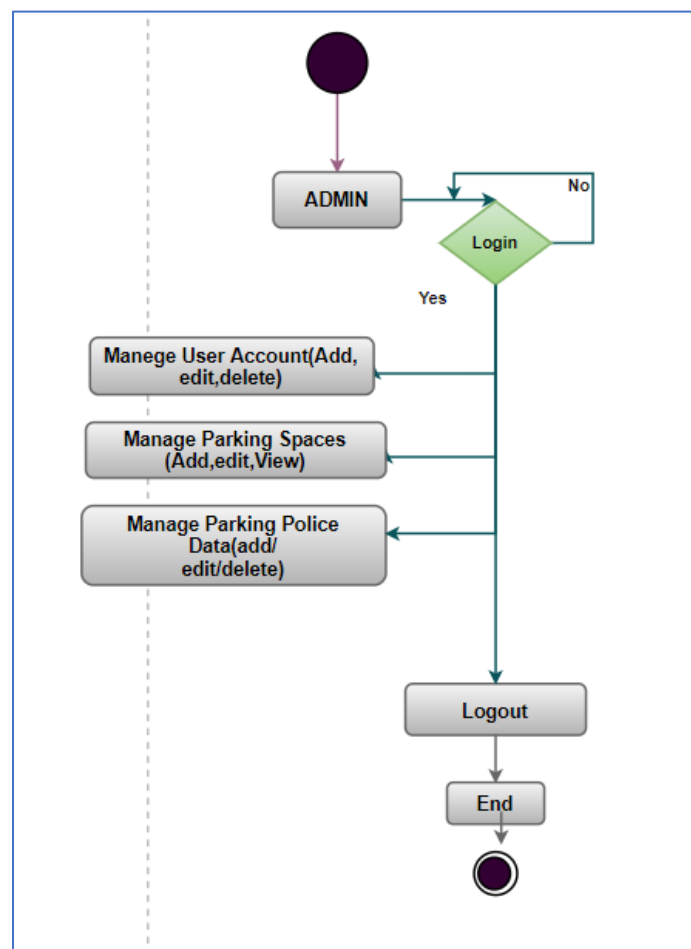


Figure 15: Activity Diagram For admin

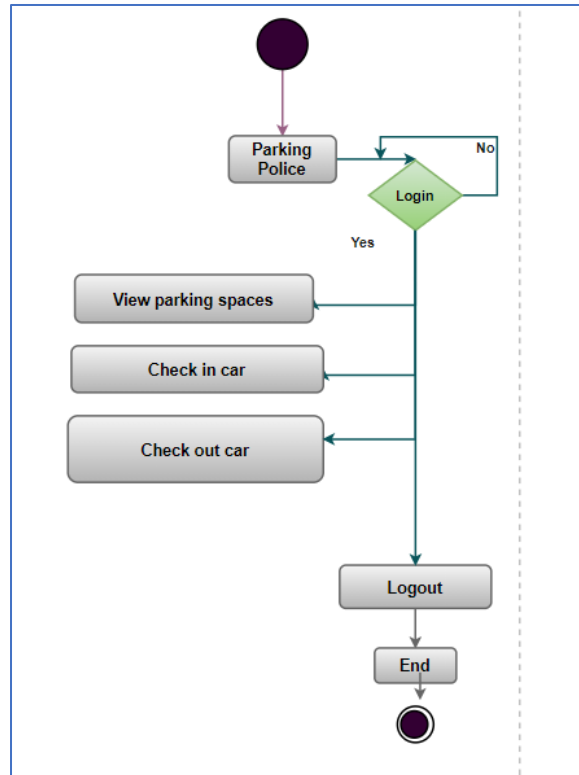


Figure 16:Activity Diagram for police

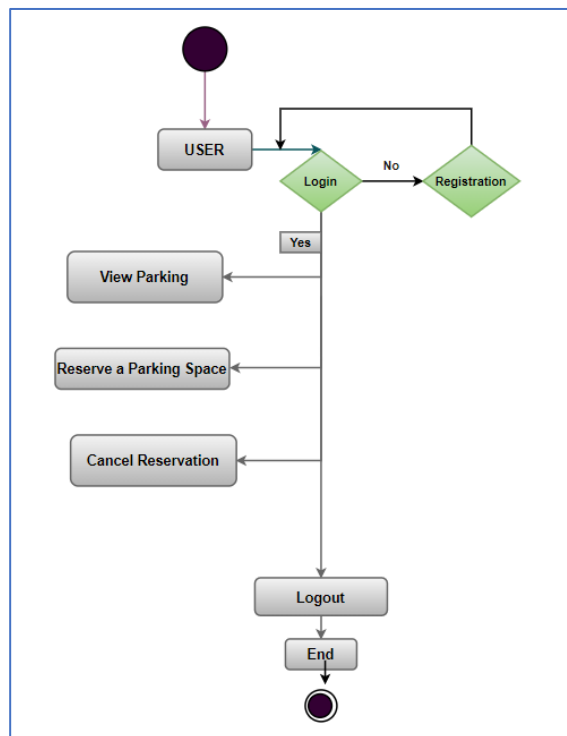


Figure 17:Activity Diagram for user

d. Sequence Diagrams

Sequence Diagrams visually map the flow of control, starting from a triggering event by an actor, and proceed step-by-step as objects interact via method calls, responses, or events. They can also include **loops**, **conditionals**, or **alternative flows** to represent complex scenarios. These diagrams are essential for detailing the logic of system interactions, identifying responsibilities within the system, and verifying that the system design meets user requirements.

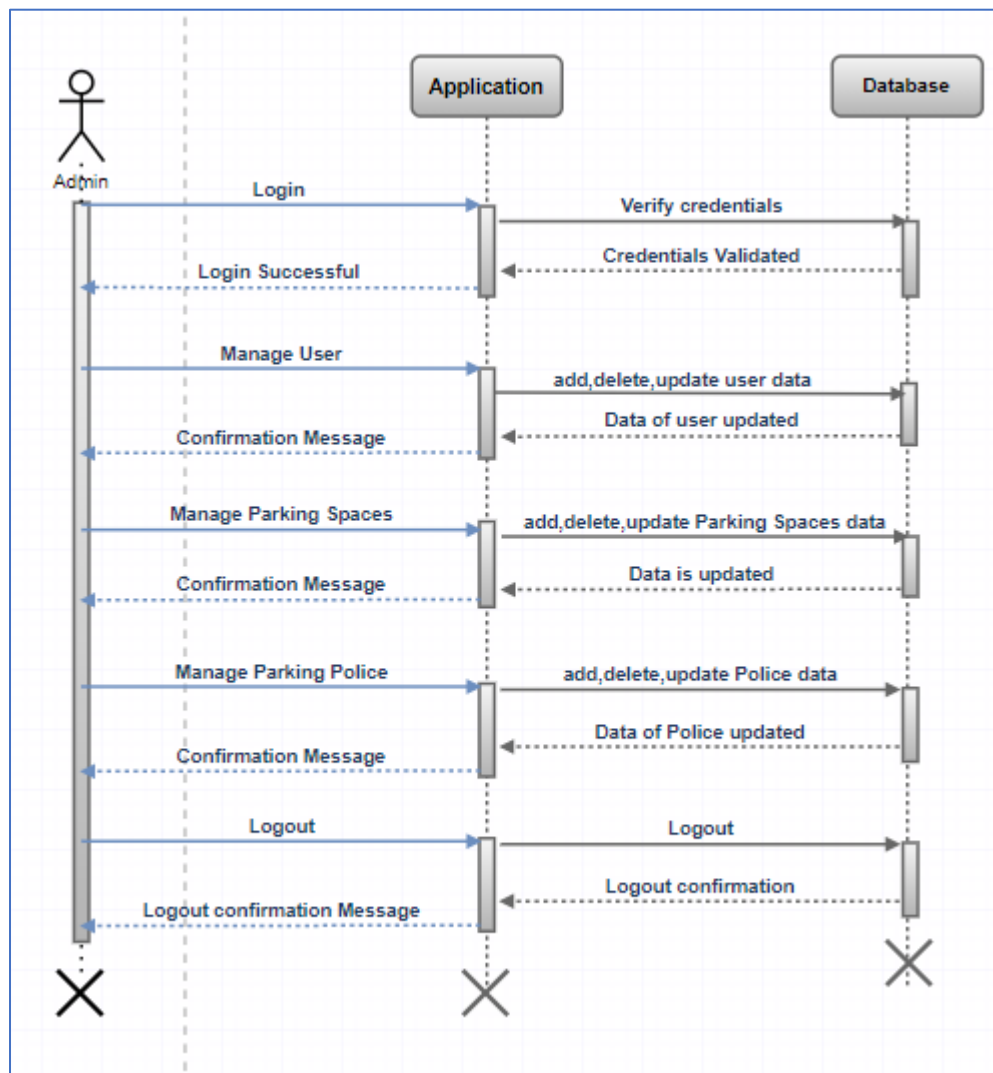


Figure 18: Sequence Diagram for Admin

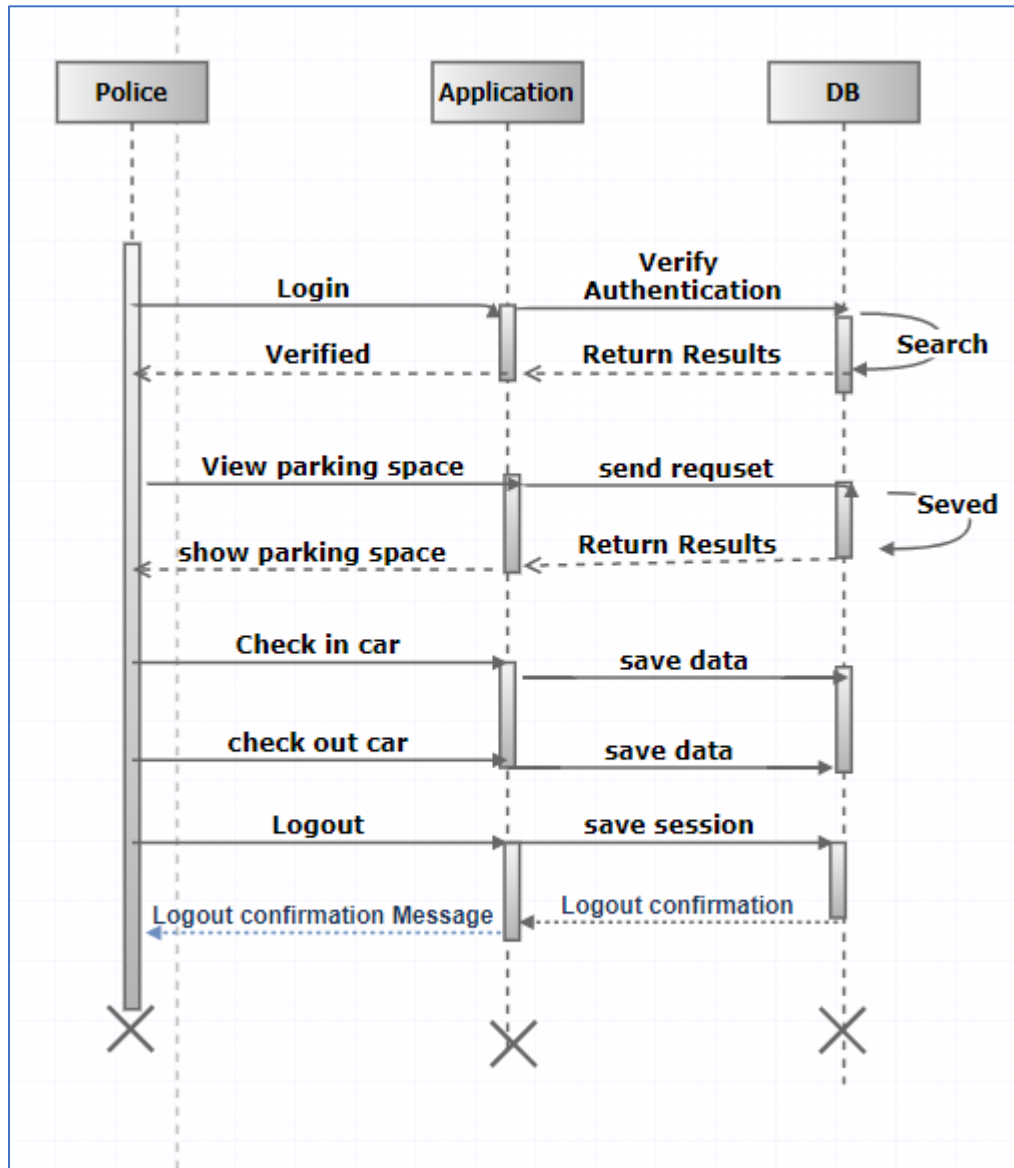


Figure 19: Sequence Diagrams for Police

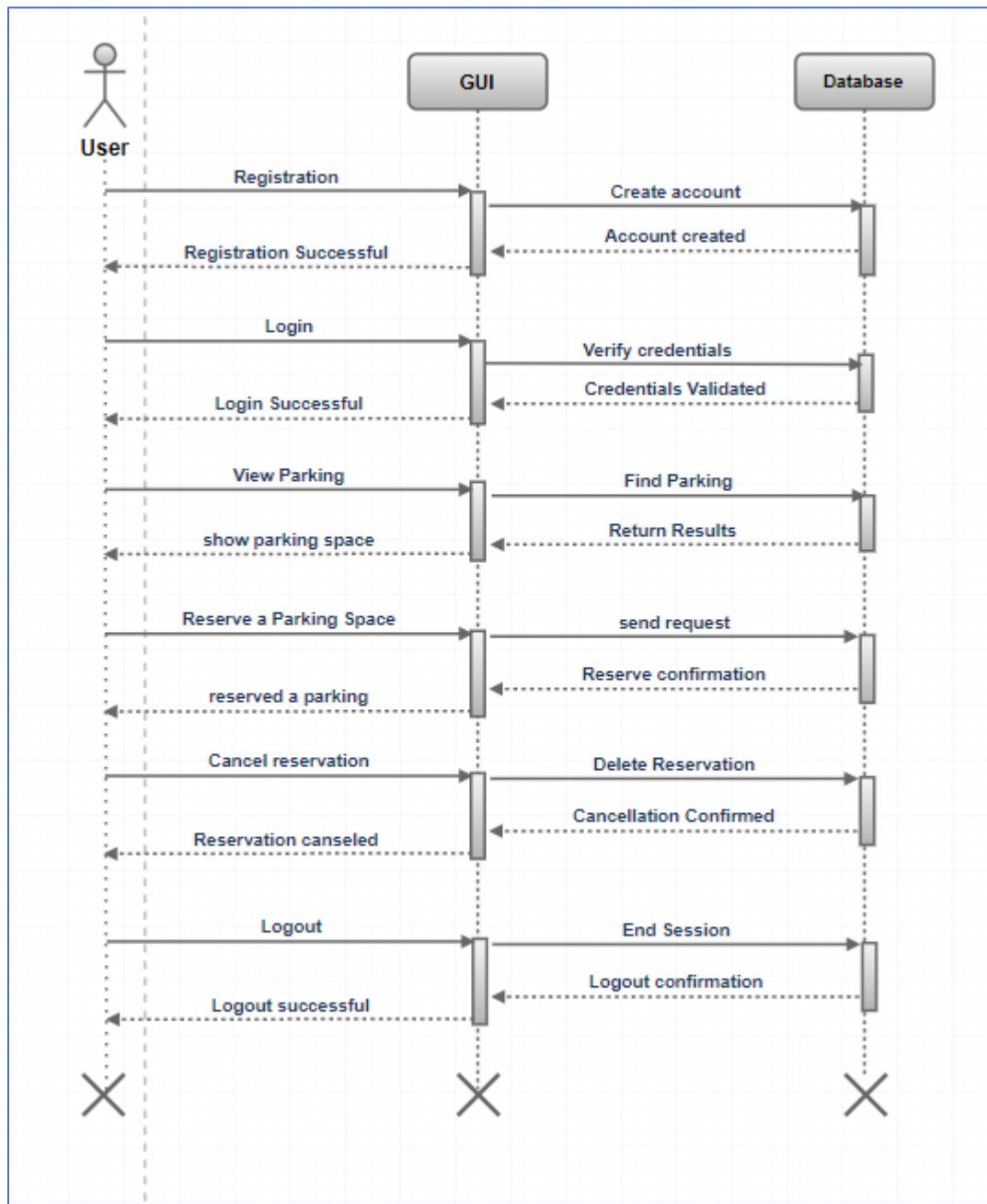


Figure 20: Sequence Diagrams For User

e. Class Diagram

A **Class Diagram** is a UML diagram that represents the static structure of a system by detailing its classes, attributes, methods, and relationships. Each class is shown as a rectangle divided into three sections for the class name, attributes, and methods. Relationships include associations, inheritance, aggregation, and composition, illustrating how classes interact. It provides a blueprint for designing object-oriented systems, identifying components and their responsibilities, and aligning the design with system requirements.

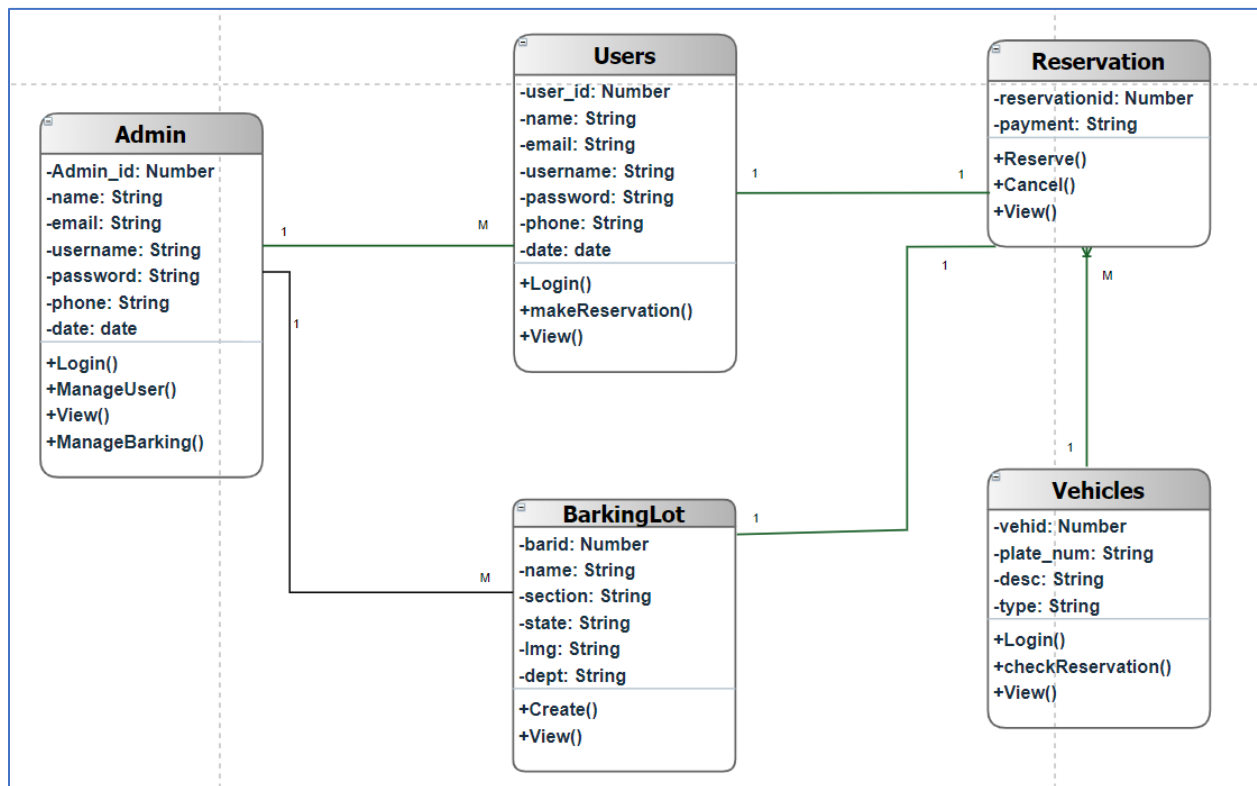


Figure 21: Class Diagram

f. Entity Relationship Diagram

An Entity Relationship Diagram (ERD) is a visual tool used in database design to represent the structure of data and the relationships between entities within a system. It outlines how various entities, such as objects or concepts, relate to each other in a database. Key components of an ERD include entities, which are typically represented as rectangles and correspond to tables in a relational database; attributes, which describe the properties of entities; and relationships, which define how entities are associated, often depicted as diamonds. ERDs also highlight primary keys (unique identifiers for entities) and cardinality, which specifies the number of occurrences one entity can have in relation to another (e.g., one-to-many). ERDs play a critical role in structuring databases, ensuring data consistency, and simplifying the management of interrelated data.

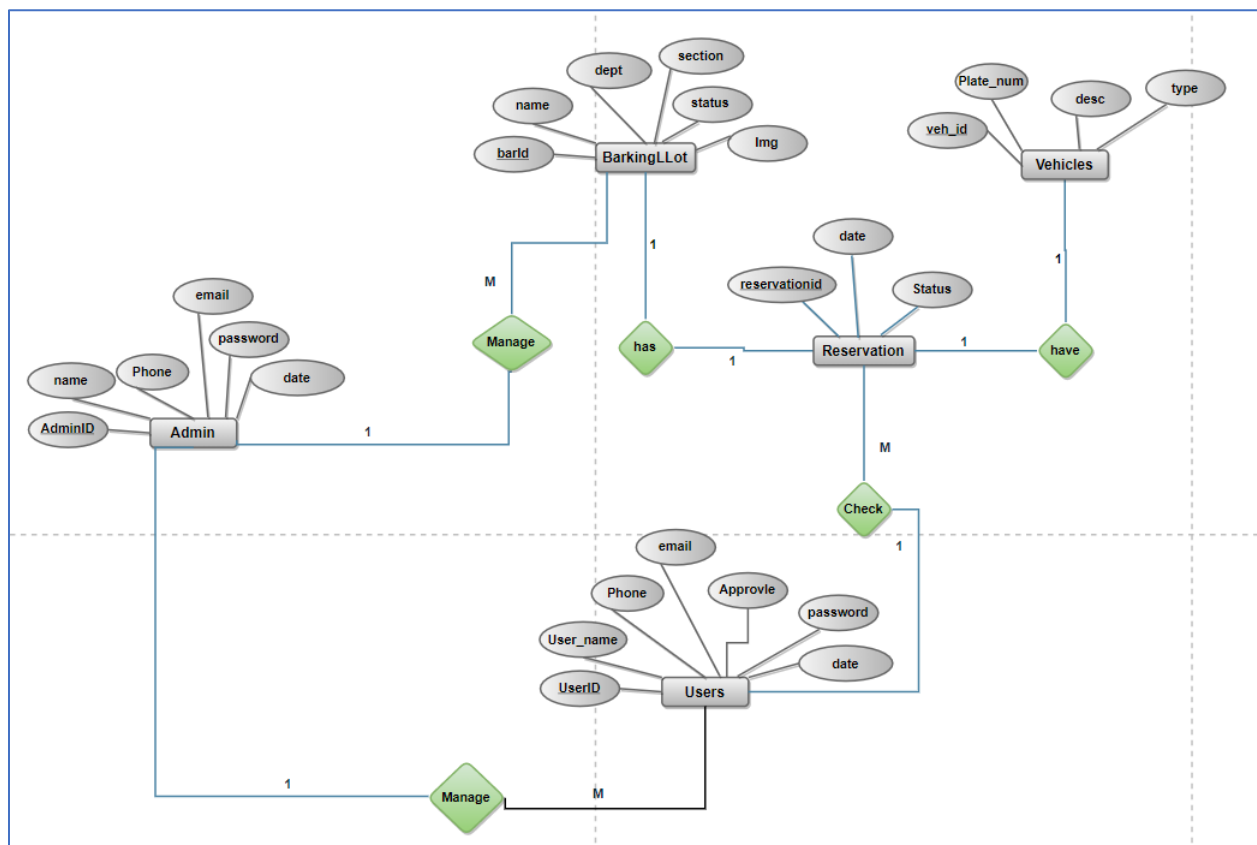


Figure 22: Entity Relationship Diagram (ERD)

7. CONCLUSION

The **SPOTLOCATOR APPLICATION** represents a forward-thinking solution to the persistent parking challenges at **Jazan University**. By leveraging real-time data, user-friendly design, and a reservation system, the app aims to streamline parking management, reduce congestion, and enhance the overall experience for students, faculty, and visitors. The project successfully addresses critical pain points, such as the difficulty of finding available parking spots and the stress associated with campus parking. The app's focus on real-time updates and seamless navigation ensures that users can efficiently locate and reserve parking, contributing to better space utilization and a more organized campus environment. Despite certain limitations, including platform exclusivity (Android) and dependency on internet connectivity, the **SPOTLOCATOR APPLICATION** is poised to significantly improve the parking experience at Jazan University. By focusing on user convenience, it not only meets the immediate needs of the campus community but also sets a foundation for future enhancements and expansion, such as incorporating additional campuses or adding multilingual support to further increase accessibility.

In conclusion, the **SPOTLOCATOR APPLICATION** is a valuable tool that aligns with Jazan University's commitment to innovation and user satisfaction, offering a practical and effective solution to campus parking challenges.

8. REFERENCES

1. [Feasibility study - Wikipedia](#)
2. [Economic feasibility - Vocab, Definition, and Must Know Facts | Fiveable](#)
3. [What is technical feasibility? \(With benefits and tips\) | Indeed.com UK](#)
4. [Project Management Plan: Samples, Examples & Free Template | Teamwork.com](#)
5. [What Is a Work Breakdown Structure \(WBS\) In Project Management? \(projectmanager.com\)](#)
6. [The Ultimate Guide to Activity Lists for Project Managers - HogoNext](#)
7. [Gantt Charts Explained: A Practical Guide for Project Managers \(teamgantt.com\)](#)
8. [How to write system requirement specification \(SRS\) documents \(jamasoftware.com\)](#)
9. [The Traditional Waterfall Approach \(umsl.edu\)](#)
10. https://en.wikipedia.org/wiki/Requirements_analysis
11. [What Is a Database? | Oracle India](#)
12. [What is Application Architecture? | Institute of Data](#)
13. [What Is Three-Tier Architecture? | IBM](#)
14. [Use case diagram - Wikipedia](#)
15. [What is DFD\(Data Flow Diagram\)? - GeeksforGeeks](#)
16. [Levels in Data Flow Diagrams \(DFD\) - GeeksforGeeks](#)
17. <https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/>
18. [UML Sequence Diagram Tutorial | Lucidchart](#)
19. [UML Class Diagram - Javatpoint](#)
20. [What is Entity Relationship Diagram \(ERD\)?](#)