

PARKING LOT MANAGEMENT SYSTEM



Elvis Ilor and Mason Hawkins

IUPUI System Design.

Jafrina Jabin.

Contents

Customer Problem Statement	1
Customer Expectations:	1
Glossary of Terms	2
Nonfunctional Requirements	4
Plan of Work	4
Functional Requirements Specification	5
System Sequence Diagram and Activity Diagram	12

Customer Problem Statements & System Requirements

Customer Problem Statement

In urban and suburban areas, parking availability is a significant issue leading to frustration for drivers and inefficiency for parking lot managers. Traditional parking management systems often rely on manual processes that lack real-time updates, transparency, and scalability. This results in overcrowded parking lots, lost revenue for parking lot operators, and wasted time for drivers searching for spaces.

Customer Expectations:

- **Drivers:** Expect a system that provides real-time updates on parking availability, the ability to reserve parking spaces, and notifications when lots are full.
- **Parking Managers:** Require tools to monitor, manage, and optimize parking spaces in real time. They expect the system to streamline the process of adding, editing, and deleting parking spaces, and provide usage reports.

- **Parking Assistants:** Need to facilitate check-ins and manage billing efficiently.

The Parking Lot Management System (PLMS) addresses these challenges by providing a web based solution that allows drivers to access real-time availability and reservation options, while parking managers gain powerful tools to manage parking spaces more effectively.

Glossary of Terms

- **PLMS (Parking Lot Management System):** A web-based system for managing parking lots, offering real-time updates, reservations, and management features.
- **Real-Time Updates:** System-generated instantaneous updates on the current availability of parking spaces.
- **Reservation:** The capability for a driver to hold a parking space in advance for a designated period.
- **Parking Assistants:** Staff responsible for managing parking lot operations, assisting with driver check-ins and billing processes.

No. Priority Weight Description

No.	Priority Weight	Description
REQ-1	High	<p>The system should allow drivers to view real-time parking space availability.</p> <p>Scenario: Drivers can access real-time data on available parking spaces through an interactive map or list on their mobile devices. The system updates automatically, showing which spots are taken or available, reducing time spent searching for parking. This feature ensures that drivers receive accurate information on the go, minimizing frustration and unnecessary driving. This is particularly valuable in busy urban areas where parking is scarce and constantly changing.</p>
REQ-2	High	<p>The system should enable drivers to reserve parking spaces in advance.</p> <p>Scenario: Drivers can log into the system to reserve parking spaces for a specific time slot. The system blocks these reserved spaces from others, ensuring the driver has a guaranteed spot upon arrival. In case of a no-show or cancellation, the spot is released back into the available pool. This feature helps reduce parking congestion and ensures that</p>

No.	Priority Weight	Description
REQ-3	Medium	<p>customers who plan can avoid the hassle of finding a spot at the last minute. This is particularly useful for high-traffic areas or events.</p> <p>Managers should be able to add, edit, and delete parking spaces.</p> <p>Scenario: Managers have access to a dashboard where they can adjust the availability of parking spaces in real-time. They can add new spaces for events or remove spaces that are under maintenance. This feature gives parking managers complete control over space allocation and ensures that the system reflects the most up-to-date status. It allows managers to dynamically respond to changing parking demands during peak periods or special events.</p> <p>The system should provide automated notifications to drivers when a parking lot is full.</p> <p>Scenario: Drivers attempting to book parking in a full lot are automatically notified to look for an alternative location or are alerted when a space becomes available. For subscribed users, push notifications are sent the moment a spot opens. This proactive communication helps drivers avoid unnecessary trips to full lots and reduces congestion around busy parking facilities. The automated notifications enhance the user experience by offering real-time convenience.</p>
REQ-4	High	<p>Managers should be able to generate reports on parking space usage.</p> <p>Scenario: Managers can access detailed reports on parking lot usage, including peak hours, average stay durations, and revenue generation. These reports help optimize parking lot operations, offering insights into when to adjust staffing or change pricing structures. For example, during holiday shopping seasons, managers can use reports to predict demand and increase the price during peak times or offer promotions during off-peak hours.</p>
REQ-5	Medium	<p>Assistants should be able to handle driver check-ins and manage billing.</p> <p>Scenario: Parking assistants can scan driver reservations or manually assign parking spaces, updating the system in real-time. Assistants can also handle billing through multiple payment methods, such as cash, card, or mobile payment solutions like Apple Pay. The system automatically calculates charges based on the driver's stay and generates invoices for later payment if needed. This streamlines the check-in process, making it quicker and more convenient for drivers and parking attendants alike.</p>
REQ-6	Medium	<p>The system should prevent reservation conflicts by disallowing double booking of parking spots.</p> <p>Scenario: The system processes reservation requests in real-time, ensuring that once a parking spot is reserved, it is immediately marked as unavailable to others. This conflict-free reservation system guarantees that drivers do not arrive to find their reserved spot occupied. It also reduces frustration for both drivers and managers,</p>
REQ-7	High	

No.	Priority Weight	Description
REQ-8	High	<p>ensuring that no manual intervention is needed to resolve double-booking errors.</p> <p>The system should support multiple payment methods, including credit/debit cards and digital wallets.</p> <p>Scenario: The system provides drivers with flexible payment options, allowing them to choose between traditional methods such as credit or debit cards or digital wallet services like Google Pay and Apple Pay. In case of payment failures, the system prompts drivers to retry or use an alternative method. This flexibility ensures that drivers can complete their payments quickly and securely, enhancing overall user satisfaction and ensuring smooth parking lot operations.</p>

Nonfunctional Requirements

- **NREQ-1:** The system should be highly scalable to support multiple parking lots across different locations.
- **NREQ-2:** The user interface should be intuitive, easy to navigate, and require minimal user training.
- **NREQ-3:** The system should ensure 99.9% uptime, providing reliable services for both drivers and managers.
- **NREQ-4:** The system should provide real-time performance, ensuring updates are pushed within a 1-second delay to all users.

Plan of Work

Here is the progress made so far:

- **Week 1-2:** The development environment was successfully set up, and the initial database schema was designed and established. The schema aligns with the planned parking space management, including real-time availability and reservations.
- **Week 3-4:** We completed the proposal statement, problem statement, and system requirements documentation, ensuring that all stakeholders are aligned with the project objectives and requirements.

- **Week 5-6:** We Core system features, including user registration, login functionality, and the initial parking space display, have been integrated into the system. These features work in conjunction with the database to provide real-time updates on parking availability.

Functional Requirements Specification

- **Drivers:** Individuals using the system to find and reserve parking spots.
- **Admin:** Responsible for overseeing parking operations, adding/removing spots, and generating reports.
- **Parking Assistants:** Facilitate check-ins, manage billing, and help customers on-site.
- **System Administrators:** Oversee the system's maintenance and ensure it operates smoothly.
- **Investors/Sponsors:** Interested in the project's success from a business and revenue perspective:
- **Drivers:** Use the system to find and reserve parking, receive notifications, and pay for parking.
- **Parking Agents:** Assist drivers, manage check-ins, and process payments.
- **Admin:** Manage parking spaces, generate reports, and oversee the system's operations.
- **System:** Provides real-time availability, manages reservations, and processes payments.
- **Add parking spot:** To add a new parking spot, including selecting the type (compact, handicapped, large, motorcycle) and location. (2 points)

- **Edit parking spot:** To modify an existing parking spot detail (type, location, availability).
- **Remove parking spot:** To delete an existing parking spot from the system. (3 points)
- **Generate reports:** To generate usage reports on parking spaces (revenue, peak hours). (3 points)
- **Login/Logout:** To log in or out of the admin account. (2 points)
- **View account:** To view account details like admin privileges or the system status. (2 points)
- **Manage staff accounts:** To create, update, or remove accounts for parking

Driver

- **View parking:** To see the real-time availability of parking spaces in the selected area. (2 points)
- **Reserve parking spot:** To reserve a parking space in advance. (2 points)
- **Receive notification:** To receive alerts when a parking spot becomes available or if a reservation is about to expire. (2 points)
- **Pay for parking:** To pay for parking through the system using credit/debit cards or digital wallets. (2 points)
- **Check out/Exit lot:** To scan the ticket at the exit and complete the parking session. (2 points)

- **Cancel reservation:** To cancel a parking reservation. (2 points)

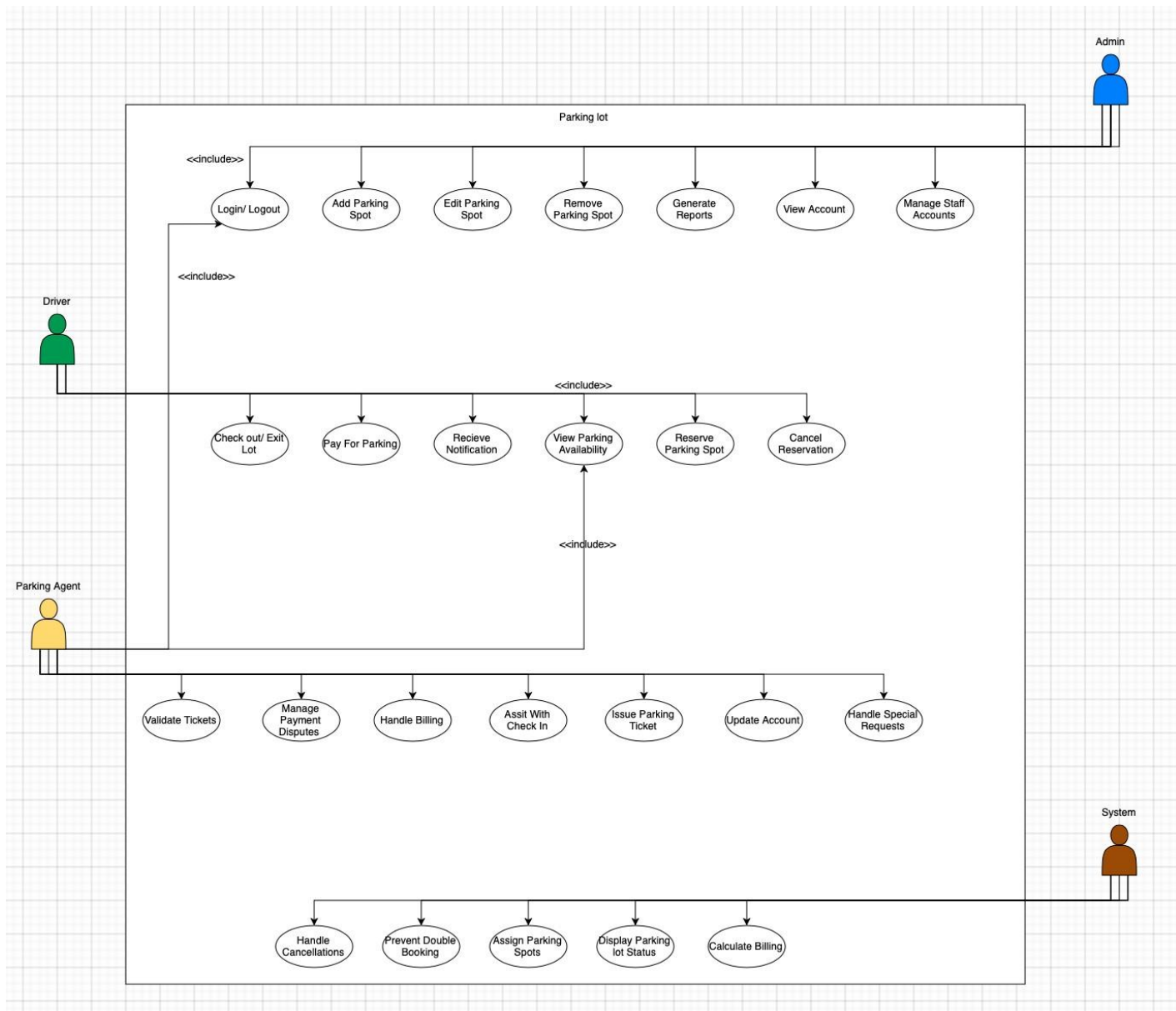
Parking Agent

- **Login/Logout:** To log in or out of the agent account. (2 points)
- **View parking availability:** To monitor parking availability across locations. (2 points)
- **Assist with check-ins:** To help drivers by scanning their reservation codes or manually assigning parking spots. (2 points)
- **Handle billing:** To assist drivers with payment processing, and accepting cash, card, or mobile payments. (2 points)
- **Issue parking ticket:** To issue parking tickets for drivers without reservations. (2 points)
- **Manage payment disputes:** To resolve payment disputes or generate invoices for later payment. (2 points)
- **Update account:** To update account details and payment information. (2 points)
- **Validate tickets:** To check the validity of parking tickets at the exit. (2 points)
- **Handle special requests:** To accommodate drivers with special requests (e.g., specific parking spots, assistance). (2 points)

System

- **Assign parking spots:** To automatically assign available parking spots based on vehicle type. (2 points)

- **Prevent double-booking:** To ensure no two drivers can reserve the same spot at the same time. (2 points)
- **Display parking lot status:** To show whether the parking lot is full or has available spots. (2 points)
- **Calculate billing:** To calculate the parking fee based on the time spent. (2 points)
- **Handle cancellations:** To process reservation cancellations and make the spot available again. (1 point)



1. View Parking Availability:

- Shared by **Drivers** and **Parking Agents**. Both need to see the real-time availability of parking spaces.

2. Reserve Parking Spot:

- Only **Drivers** can reserve parking spots in advance.

3. **Receive Notifications:**

- **Drivers** receive alerts when a parking spot is available or when a lot is full.

4. **Pay for Parking:**

- **Drivers** can complete payments using various methods like credit card, debit card, or mobile payment.

5. **Check out/Exit Lot:**

- **Drivers** scan a ticket at the exit to calculate the parking fee.

6. **Cancel Reservation:**

- **Drivers** can cancel reservations, and the spot is made available again.

7. **Manage Check-ins:**

- **Parking Agents** assist **Drivers** with checking in by scanning reservation codes or manually assigning spots.

8. **Process Payments:**

- **Parking Agents** handle payments through cash, card, or mobile payment options.

9. **Issue Parking Ticket:**

- **Parking Agents** issue tickets to **Drivers** without reservations.

10. **Login/Logout:**

- Shared by **Parking Agents** and **Admin** to access their respective accounts.

11. Add/Edit/Remove Parking Spots:

- **Admin** can modify the availability and location of parking spots.

12. Generate Reports:

- **Admin** generates reports on parking usage, revenue, and peak hours.

13. Prevent Double-Booking:

- The **System** ensures that no two reservations are made for the same spot by blocking it once reserved.

14. Show Parking Lot Status:

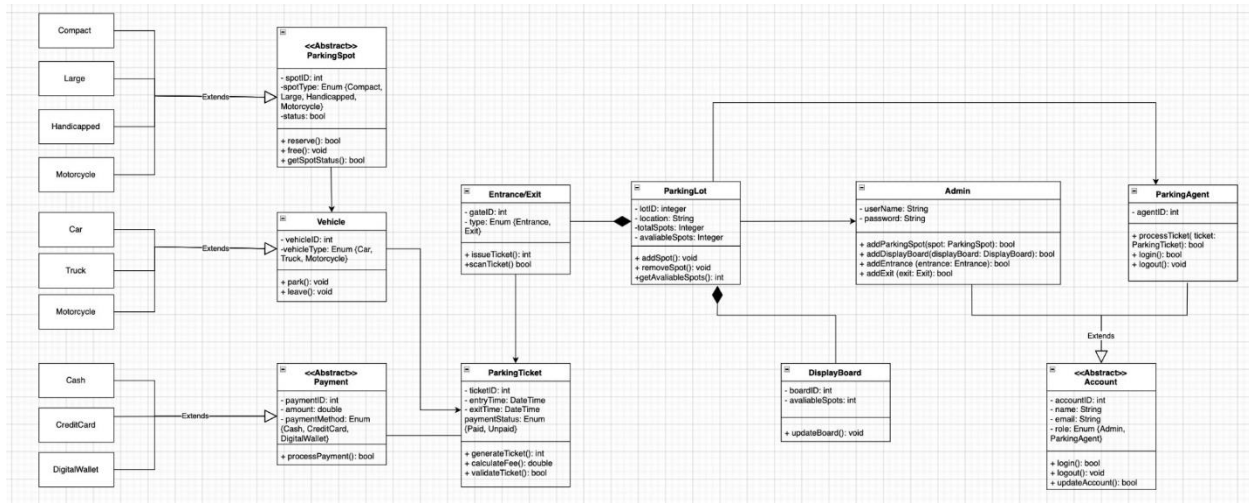
- The **System** shows whether the parking lot is full or has available spaces for **Drivers** and **Parking Agents**.

15. Calculate Parking Fees:

- The **System** calculates parking fees based on the duration of the driver's parking session.

16. Handle Cancellations:

- The **System** processes reservation cancellations and updates the availability of parking spaces.



System Sequence Diagram and Activity Diagram

Overview:

We structured our System Sequence Diagrams (SSDs) and Activity Diagrams based on the key use cases outlined in the Functional Requirements Specification. Below, we detail our approach to completing the diagrams for two complex use cases, ensuring a clear representation of the system's interactions and flows.

Step 1: Identify Key Use Cases

As a team, we selected two key use cases from the specification document that are both complex and involve multiple interactions:

1. Reserve Parking Spot

- Actor: Driver
- Goal: Enable the driver to find and reserve a parking spot through the system.

2. Pay for Parking

- Actor: Driver or Parking Agent

- Goal: Process parking payment either at the exit or through a payment kiosk.

Step 2: Create System Sequence Diagrams (SSDs)

We broke down the sequence of interactions between the actor and the involved system components for each use case.

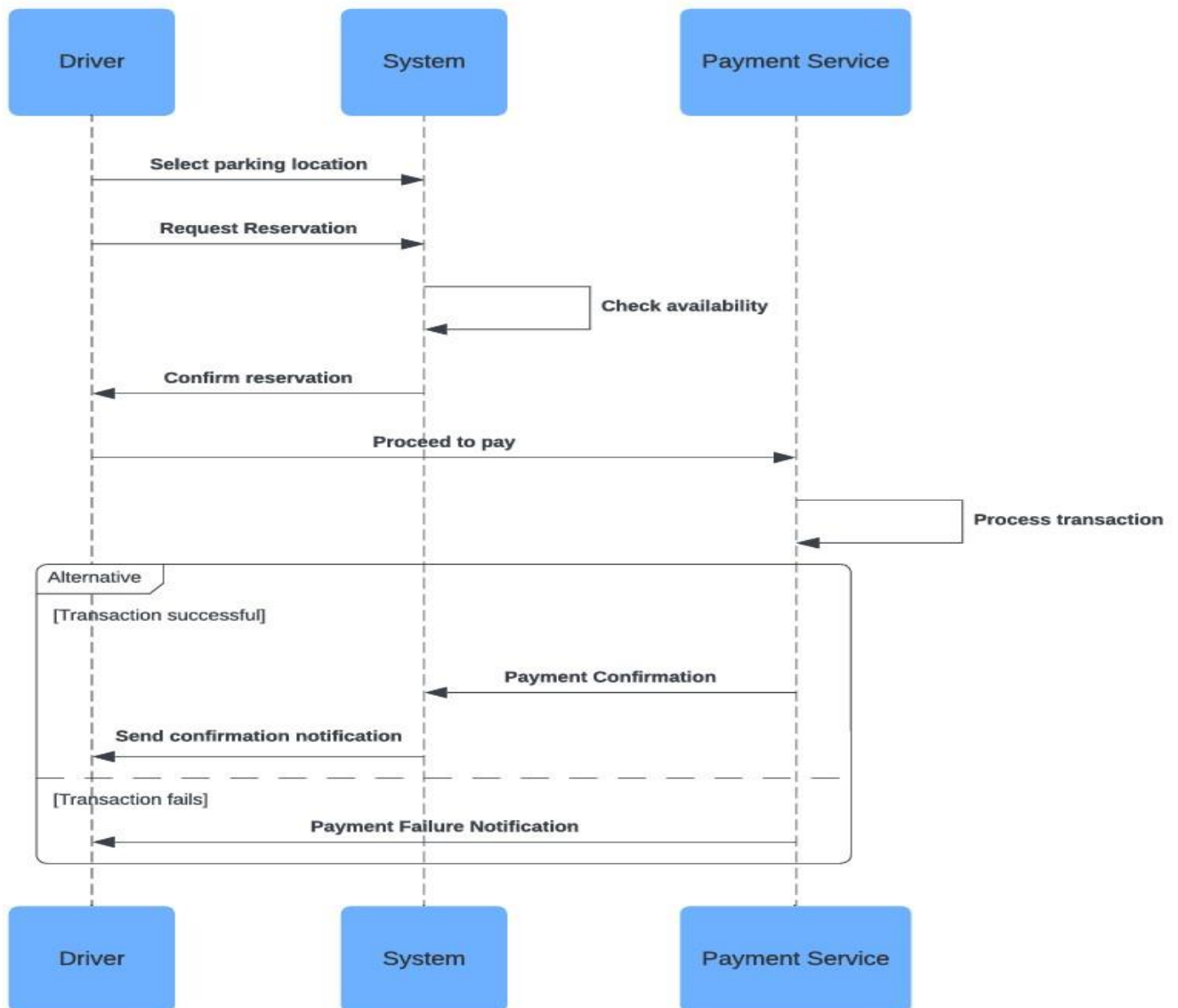
1. SSD for Reserve Parking Spot

- Actor: Driver
- Objects Involved: System, Payment Service, Notification System

Sequence Flow:

3. The driver selects a parking location and requests a reservation.
4. The system checks the availability of parking spaces and confirms the reservation.
5. The driver proceeds to payment.
6. The payment service processes the transaction.
7. If the transaction is successful, the system sends a confirmation notification to the driver.
8. If the transaction fails, a payment failure notification is sent to the driver.

Diagram:



1. SSD for Pay for Parking

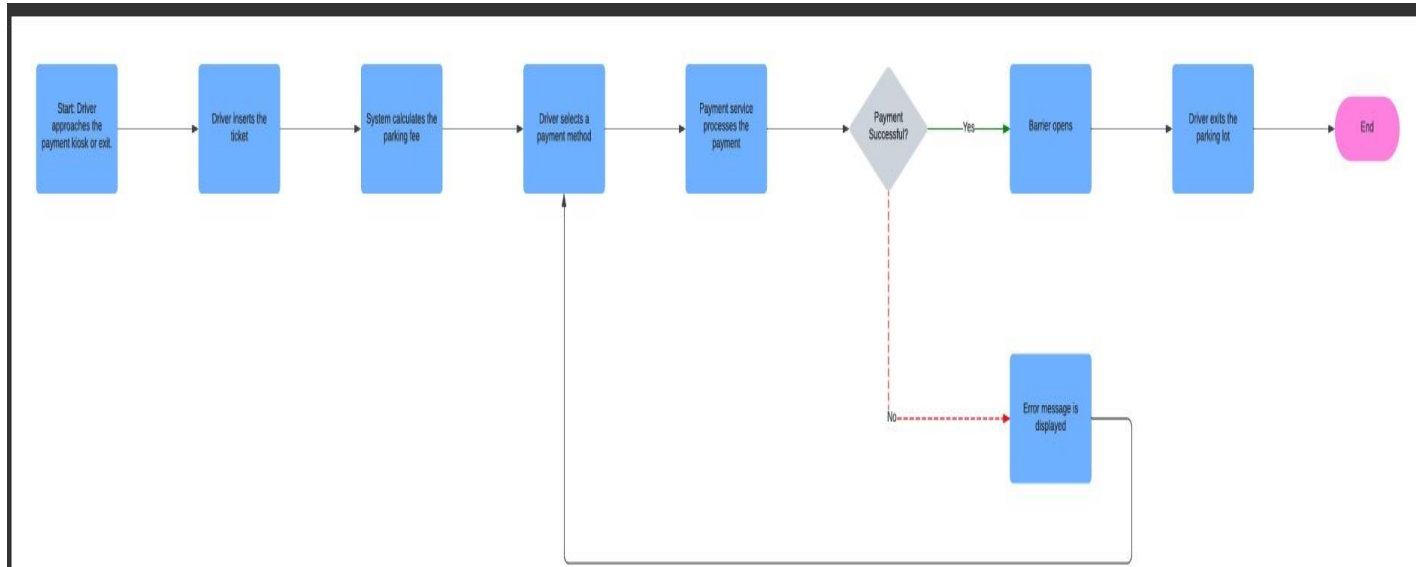
- Actor: Driver
- Objects involved: System, Payment Service

Sequence Flow:

1. The driver inserts the parking ticket at the kiosk or exit.

2. The system calculates the parking fee.
3. The driver selects a payment method (e.g., card, mobile payment).
4. The payment service processes the transaction.
5. If the payment is successful, the system opens the barrier, and the driver exits.
6. If the payment fails, an error message is displayed.

Diagram:



Step 3: Create Activity Diagrams

We used activity diagrams to illustrate the flow of events from start to finish for each use case.

Example: Activity Diagram for Pay for Parking

- Initial State: The driver approaches the payment kiosk or exit.

Actions:

1. The driver inserts their parking ticket.

2. The system calculates the parking fee.
 3. The driver selects a payment method (e.g., card or mobile payment).
 4. The payment service processes the transaction.
 5. If the payment succeeds, the barrier opens. If not, an error message is displayed.
- Final State: The driver exits the parking lot.

Diagram: This activity diagram includes:

- Decision nodes to handle success or failure in payment (e.g., "Payment Successful? Yes/No").
- Swim lanes to separate the activities performed by the Driver, System, and Payment Service, making it easier to visualize their roles.