Mini project report on

Parking Space Management System

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology in

Computer Science & Engineering

UE20CS352 –OOADJ Project

Submitted by:

Rakshita B R	**PES2UG21CS813**
Sayeda Huzefa	**PES2UG21CS816**
Shamya S	**PES2UG21CS817**
Sunil K	**PES2UG21CS821**

Under the guidance of

Prof. Nivedita Kasturi

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

TABLE OF CONTENTS

1. INTRODUCTION :

Parking Space Management System (PSMS) is a software application that helps to manage the allocation, utilization and monitoring of parking spaces in an organized manner. This system helps to ensure efficient use of parking resources and helps reduce traffic congestion and improve overall experience for drivers. The Parking Space Management System can be developed using Object Oriented Analysis and Design (OOADJ) approach, providing a robust, scalable, and flexible solution for managing parking spaces.

This has the potential to revolutionize the way parking spaces are managed and utilized. It can reduce the time taken to park a vehicle, reduce the risk of accidents, and improve overall efficiency in parking management. Additionally, it can provide valuable data on parking patterns and usage, which can be used to optimize parking lot layouts and improve future parking management strategies.

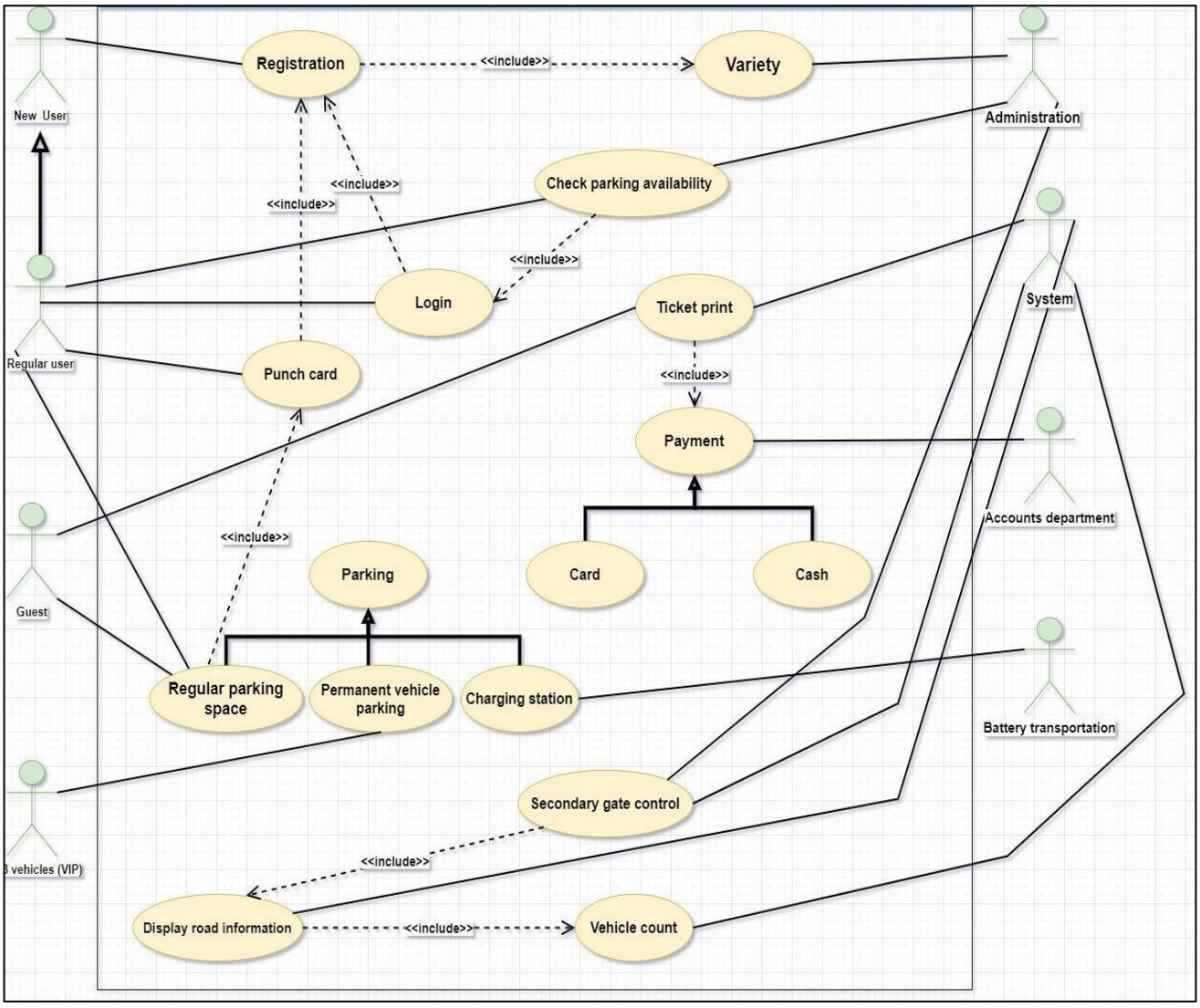
Overall, the Parking Space Management System has the potential to improve the parking experience for both drivers and parking lot operators, and it represents a significant step forward in parking management technology.

2. PROBLEM DEFINITION:

Parking system data flow is often used as a preliminary step to create an overview of vehicle parking without going into great detail. Which can later be elaborated. It normally consists of overall application dataflow and processes of vehicle parking process. It contains all of the user flow and their entities such all the flow of car parking, parking space, parking slots, parking fees, car owner and other many more information. All of the below diagrams has been used for the visualization of data processing and structured design of the vehicle parking process and working process.

This Parking Space Management System has the potential to revolutionize the way parking spaces are managed and utilized. It can reduce the time taken to park a vehicle, reduce the risk of accidents, and improve overall efficiency in parking management. Additionally, it can provide valuable data on parking patterns and usage, which can be used to optimize parking lot layouts and improve future parking management strategies.

3. USECASE MODELING:



Use Case Description:

Login

Purpose: Information for parking system.
Overview: The user login into the system to use enter the system.
Actor: **User**
Precondition: The user must be registered.
Flow the event: The **user** requests for log into the system. The system asks for **user's** ID and password. If **user** ID and password is valid then **user** can log in the system. If **user** ID and password is not valid then display gives output "Please enter correct password".

Check parking availability

Purpose: Checks for parking space for user's vehicle.
Overview: The user log into the system then checks parking availability for parking user's vehicle. Actor: **User**
Precondition: The user must be registered.
Flow of event: The **user** log into the system. **User** searches for parking space. If parking space available **user** books the parking space. If there is no parking space available **user** will not be allowed to park.

Ticket print

Purpose: For guest vehicle parking.
Overview: Guest requests for ticket to park vehicle. Actor:
Guest
Precondition: Guests don't need to be registered on the system.
Flow of event: If the **administration** approves then the **guest** can search for parking space. If parking space is available for guests the **system** asks for payment. After payment the **system** prints ticket for **guest**.

Registration

Purpose: Apply for registration to park vehicle.
Overview: The use case starts when the user log into the system then requests for registration. Then the system asks for employee or student ID. If the given ID is valid then the system gives ID and password to new user.
Actors: **New user, System**

Precondition: The system must be in a state ready for online registration.
Post condition: The system must have to store registration information.
Flow of event: Unregistered employee or student requests for registration. The system asks for employee or student ID details. If ID is valid then the system asks for parking time from student or employee. If the ID is invalid display gives output "Invalid ID". Then the system provides ID and password to the new user. Then the registration is complete.

Payment

Overview: Guests need to pay for vehicle parking.
Actors: **Guest, Accounts department**

Precondition: Need to get permission from administration.
Post condition: Print ticket,
Flow of event: **Guests** request for payment. If **administration** permits for park their vehicles they may pay with card or cash to **accounts department** in the **system** and get their parking ticket.

Punch card

Purpose: For parking vehicle
Overview: User punch card for parking vehicle in parking space.
Actors: **Users, Administration**

Precondition: Must be registered into system.
Post condition: Entering car to regular parking space.
Flow of event: If the punch card is valid the user will punch the card the gate will open for the user's vehicle to enter the parking space. If the punch card not valid then the system will ask for registration from **administration**.

Secondary gate

Purpose: To count vehicle and display road information.
Overview: Secondary gate count vehicle and display information.
Actors: **System, Administration**

Flow of event: Secondary gate counts vehicle and displays information about road. If there is no space for parking then secondary gate display about that from **administration**.

Parking

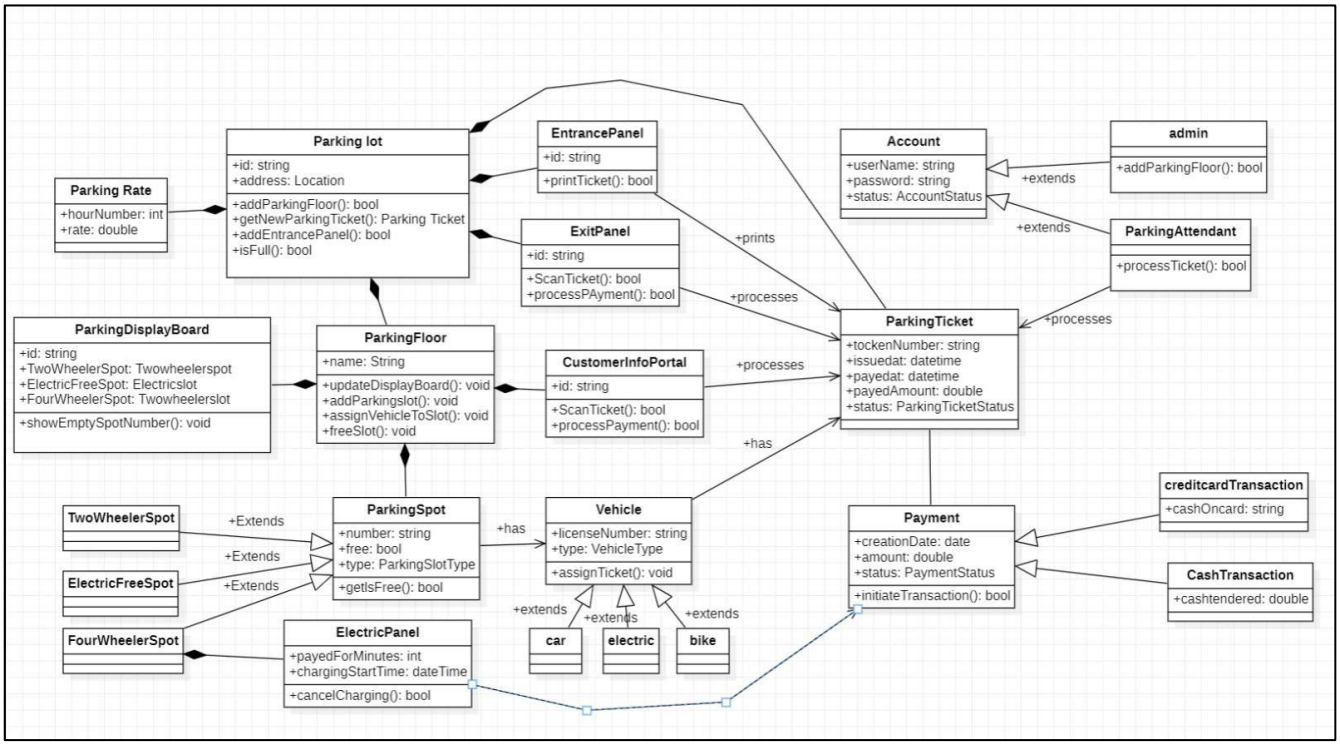
Purpose: To park vehicles of regular user, guest, VIP and battery transportation.
Overview: Regular user, guest, VIP and battery transportation requests to park their vehicle.

Actor: **Regular user, Guest,VIP vehicles and Battery transportation**

Precondition: Regular user or user must be registered into system excluding Guest, VIP and Battery transportation.

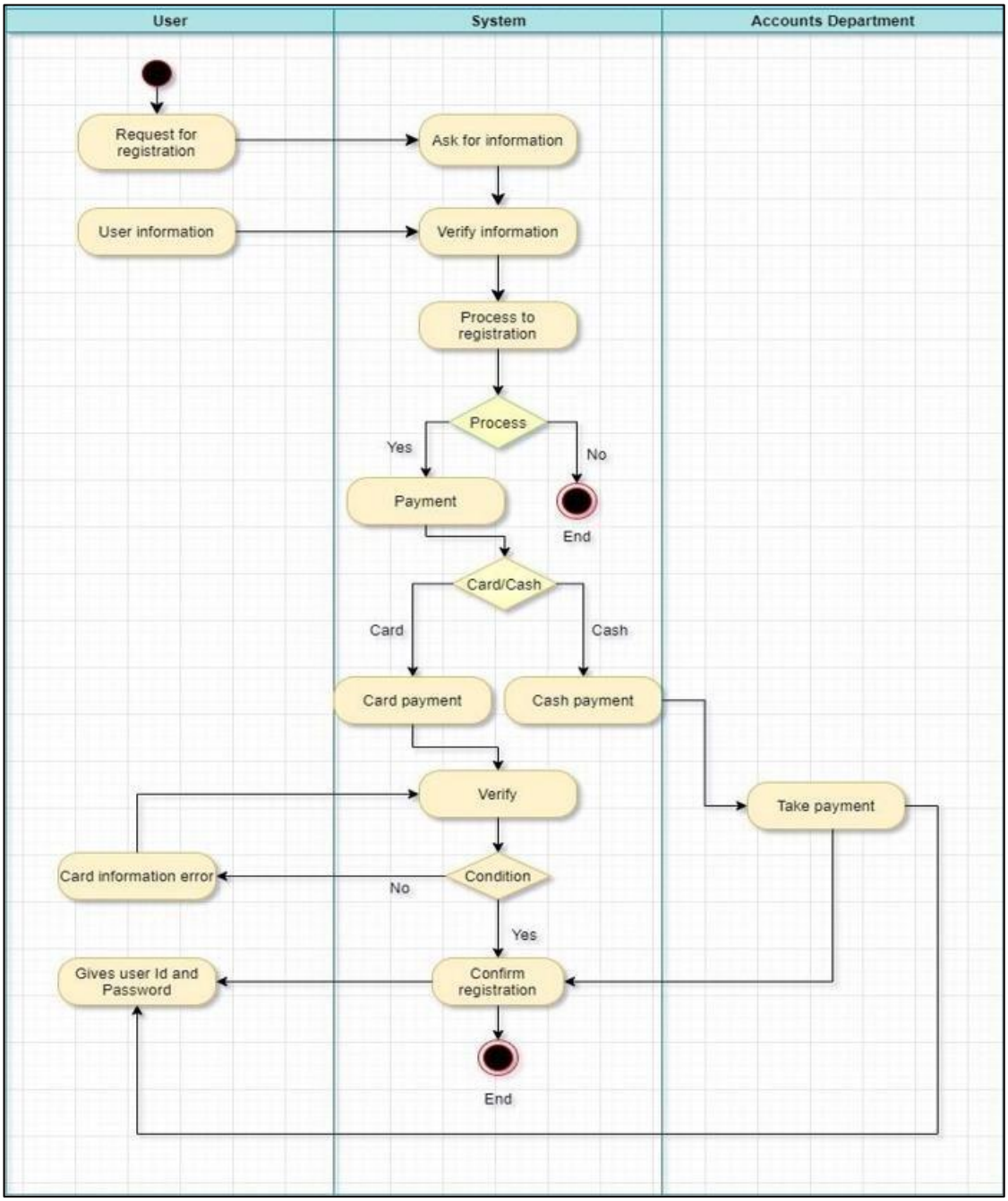
Flow of event: There are there type of parking space available. **Regular parking space** is for parking **regular user** to park their vehicles using registered punch card. **Permanent vehicle parking** is for parking **VIP vehicles** and they don't need any registration because they are the **system**. **Guests** parks their vehicle by paying into **system**. If any **regular user** vehicle arrives they need to leave the parking space because the space is registered for **regular user's** vehicle. **Charging station** is used for **battery transportation** service for students. If they run out of charge then **battery transportation** charges it's battery at **charging station**.

4. CLASS MODEL:

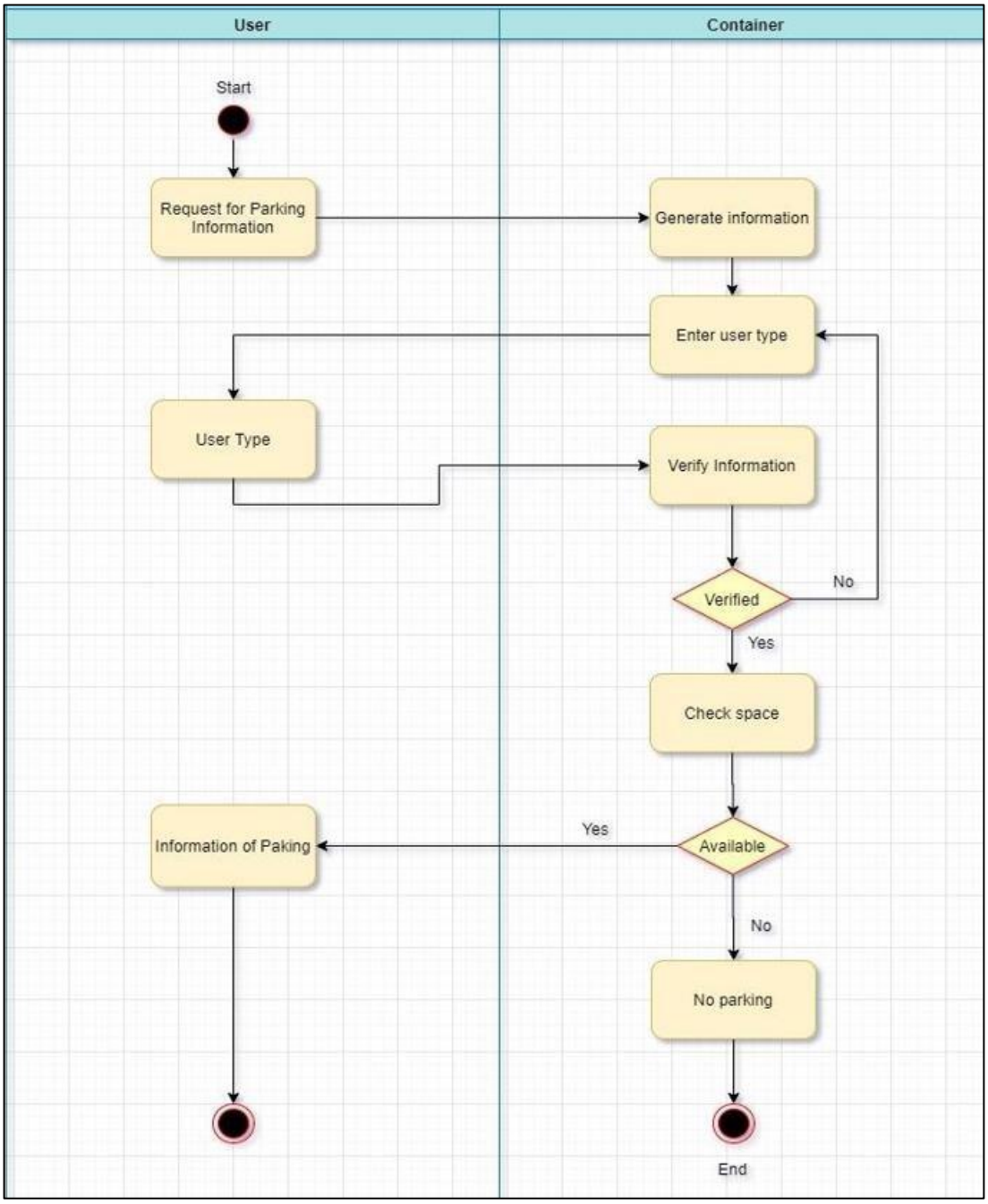


5. ACTIVITY MODELING:

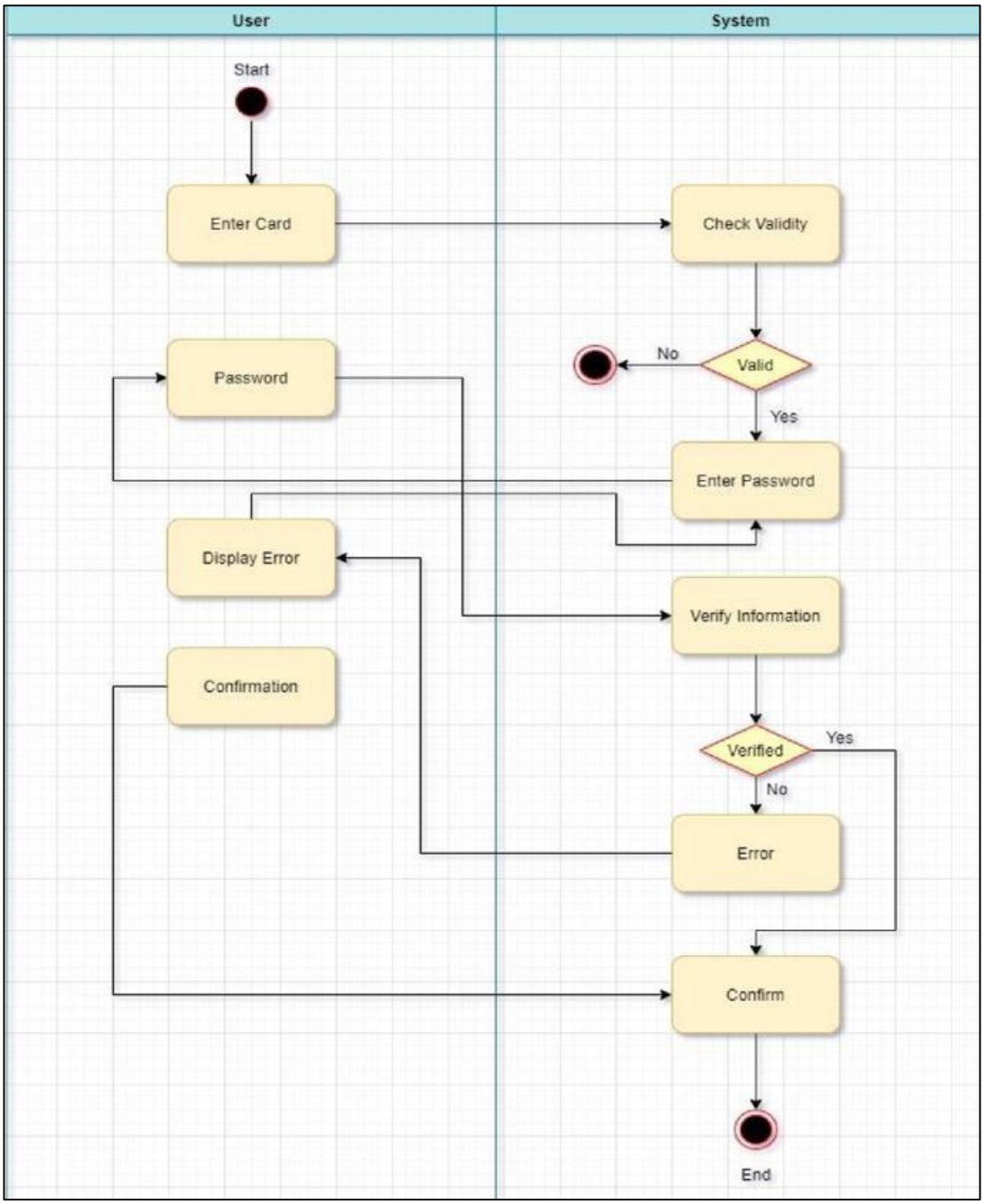
Registration:



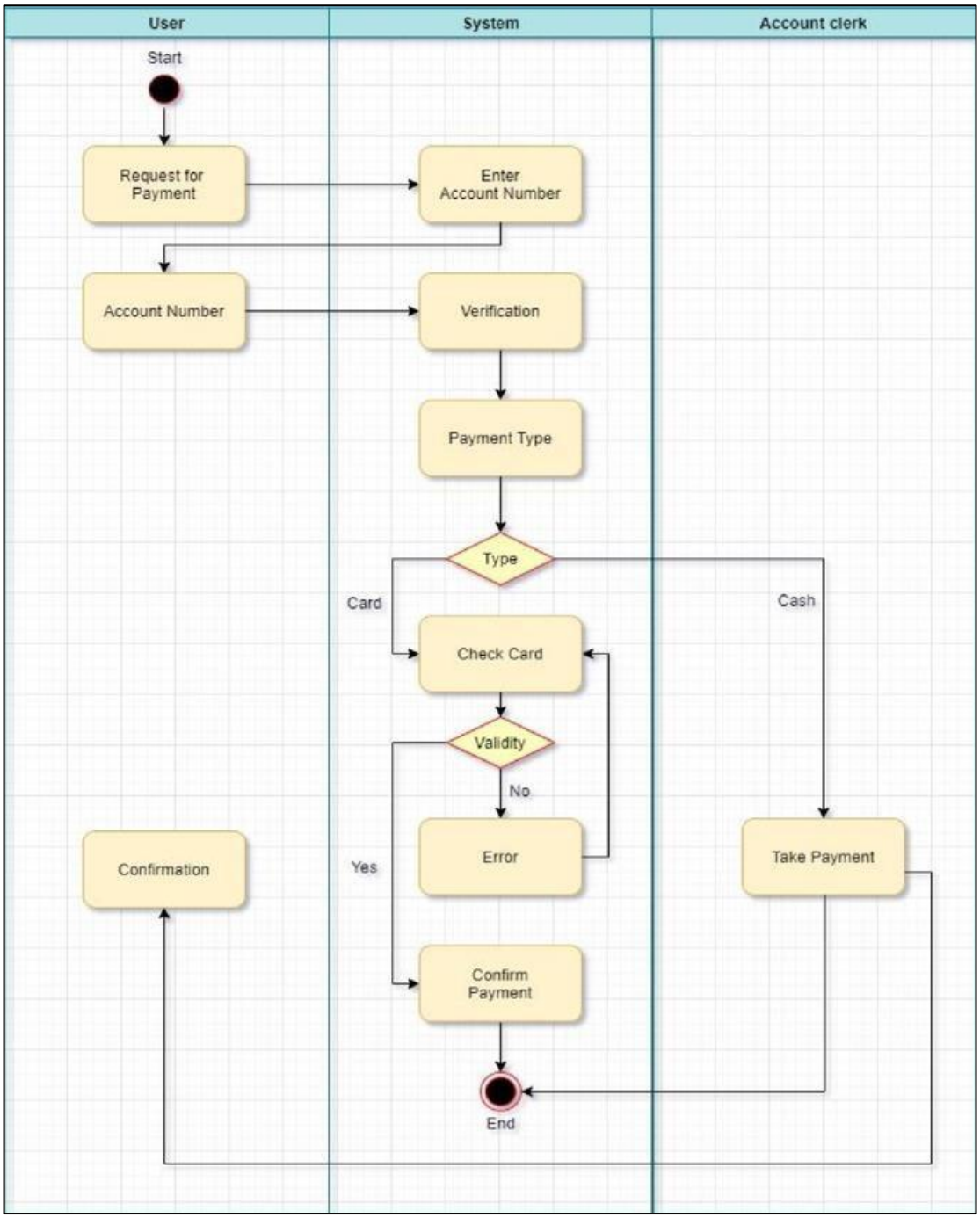
Parking Availability:



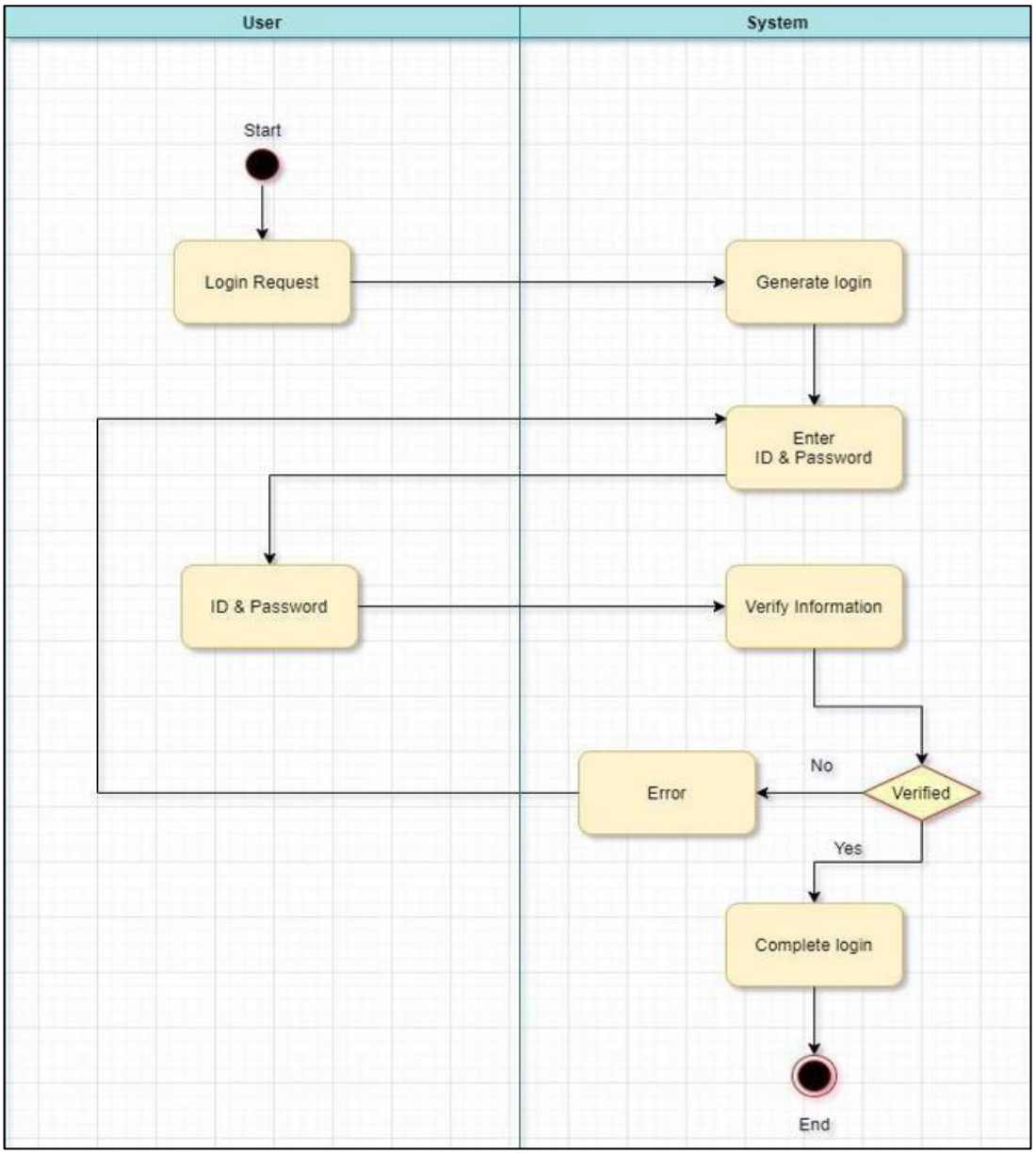
Punch card:


User
System
Start
Enter Card
Check Validity
No
Valid
Password
Yes
Enter Password
Display Error
Verify Information
Confirmation
Verified
Yes
No
Error
Confirm
End

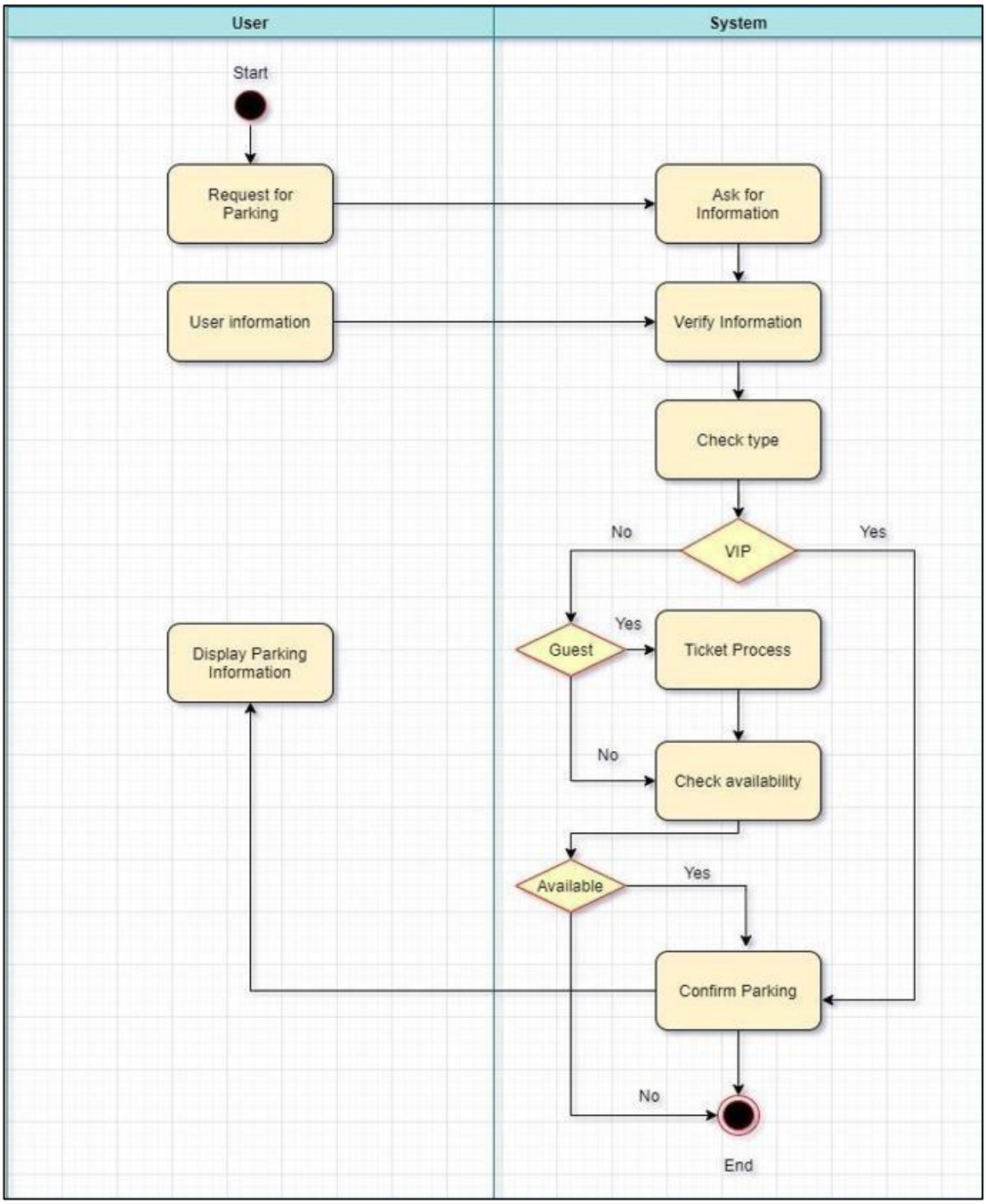
Payment:


User
System
Account clerk
Start
Request for Payment
Enter Account Number
Account Number
Verification
Payment Type
Type
Card
Cash
Check Card
Validity
No
Yes
Error
Take Payment
Confirmation
Confirm Payment
End

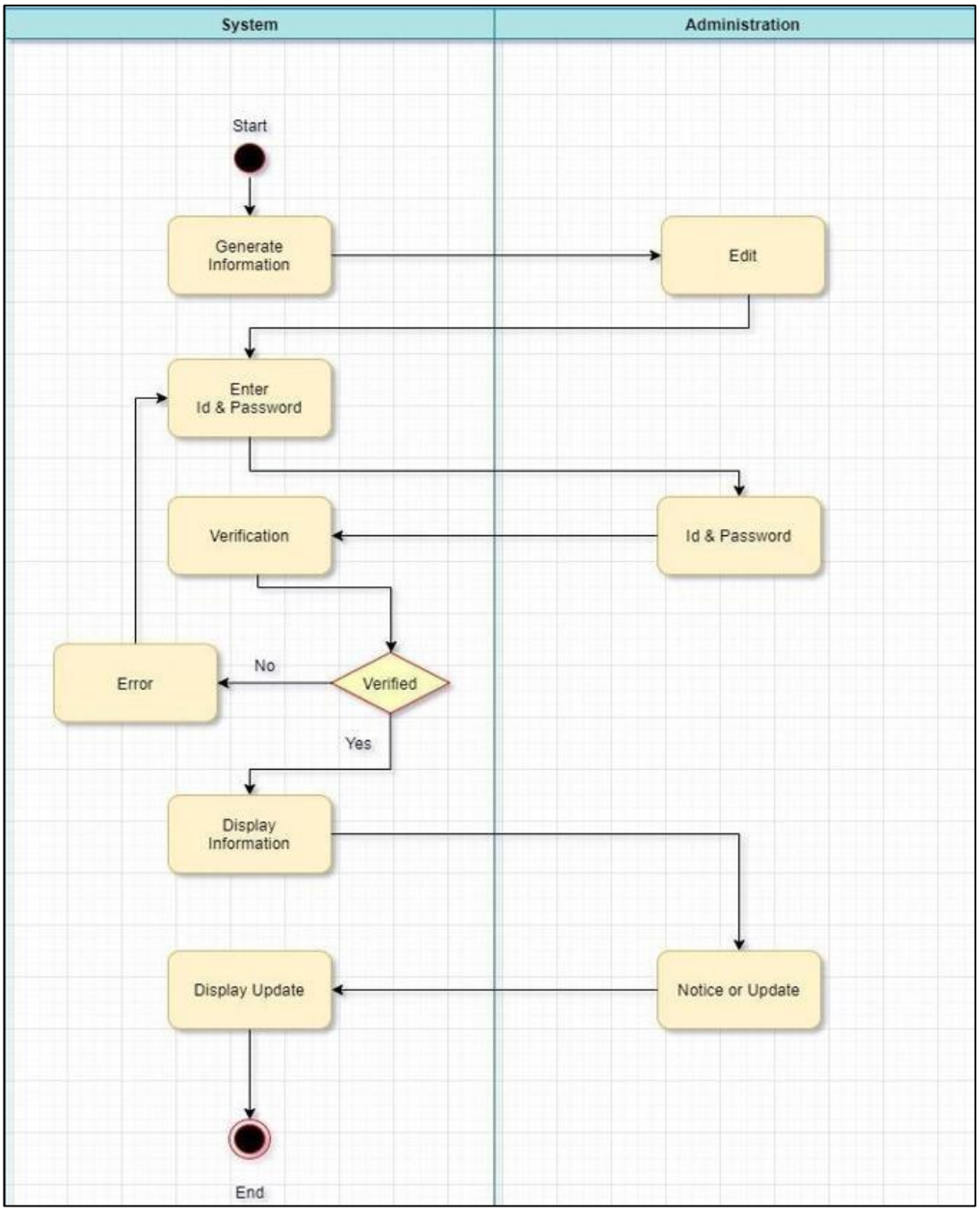
User Login:


User
System
Start
Login Request
Generate login
Enter
ID & Password
ID & Password
Verify Information
No
Error
Verified
Yes
Complete login
End

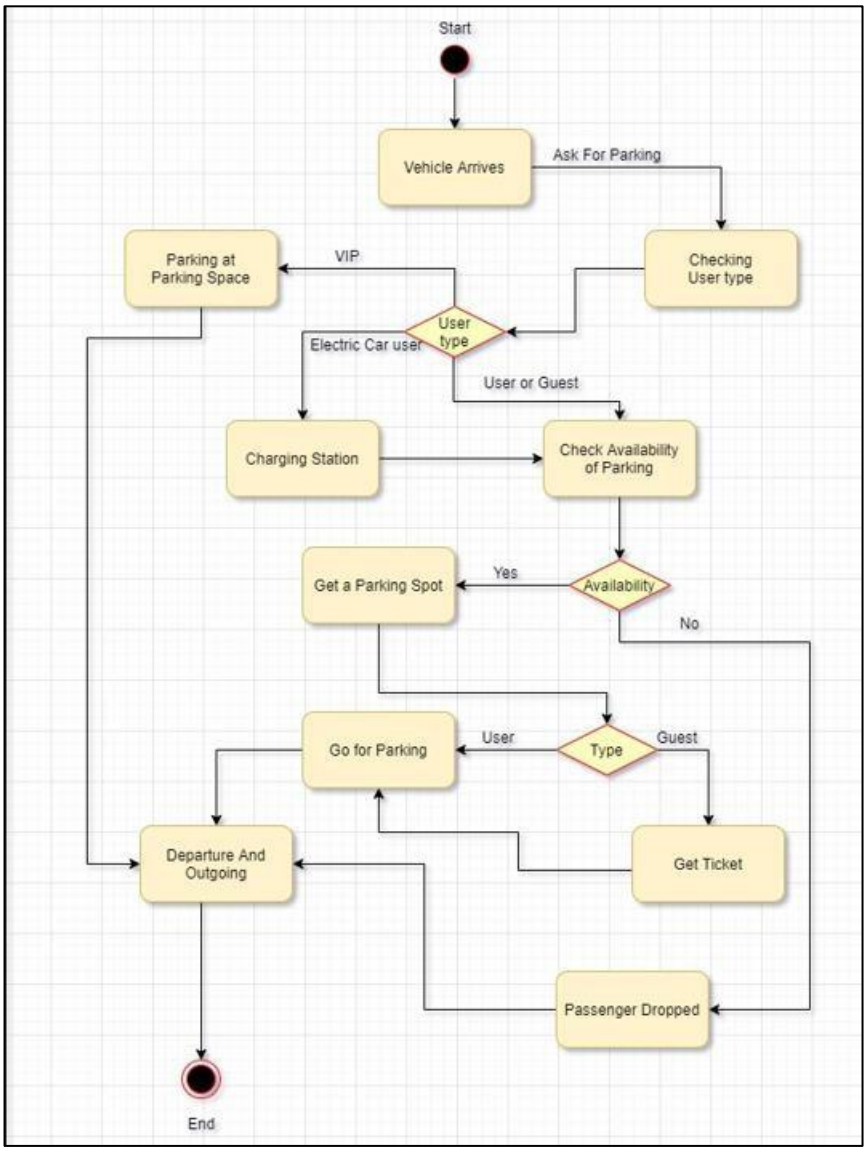
Parking:


User
System
Start
Request for Parking
Ask for Information
User information
Verify Information
Check type
No
VIP
Yes
Yes
Guest
Ticket Process
Display Parking Information
No
Check availability
Available
Yes
Confirm Parking
No
End

Secondary Gate:



6. STATE MODELING:



7. IMPLEMENTATION :

Tools and Frameworks Used:

- **Maven:** Maven is a build automation tool used primarily for Java projects. Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

- **Python Geolocater**: It is possible using geopy to extract the coordinates meaning its latitude and longitude. Therefore, it can be used to express the location in terms of coordinates.

- **Java Swing**: Java Swing tutorial is a part of Java Foundation Classes (JFC) that is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. Unlike AWT, Java Swing provides platform-independent and lightweight components. The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

- **MongoDB:** MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public Licence.

6. RESULTS SCREENSHOTS :

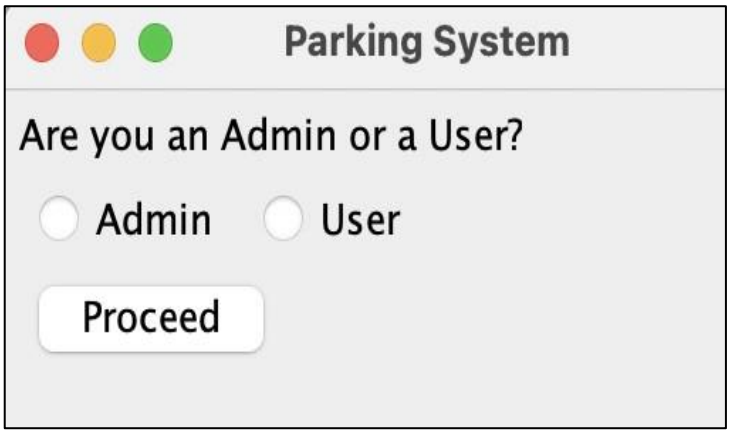


Figure 1: Default Display shown to the users

A very first onboarding page which asks if the user is a admin of parking lot or a user who wants his/her car to be parked

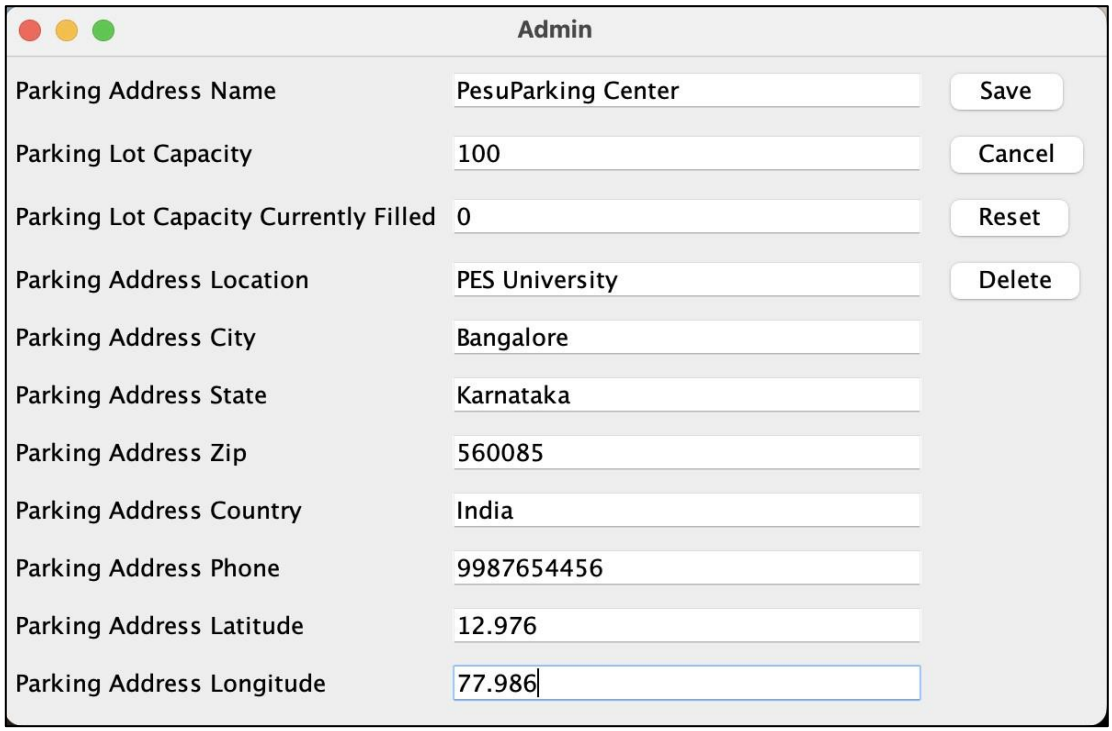


Figure 2: Admin Interface

Parking Lot Registration form for the admin of the parking lot

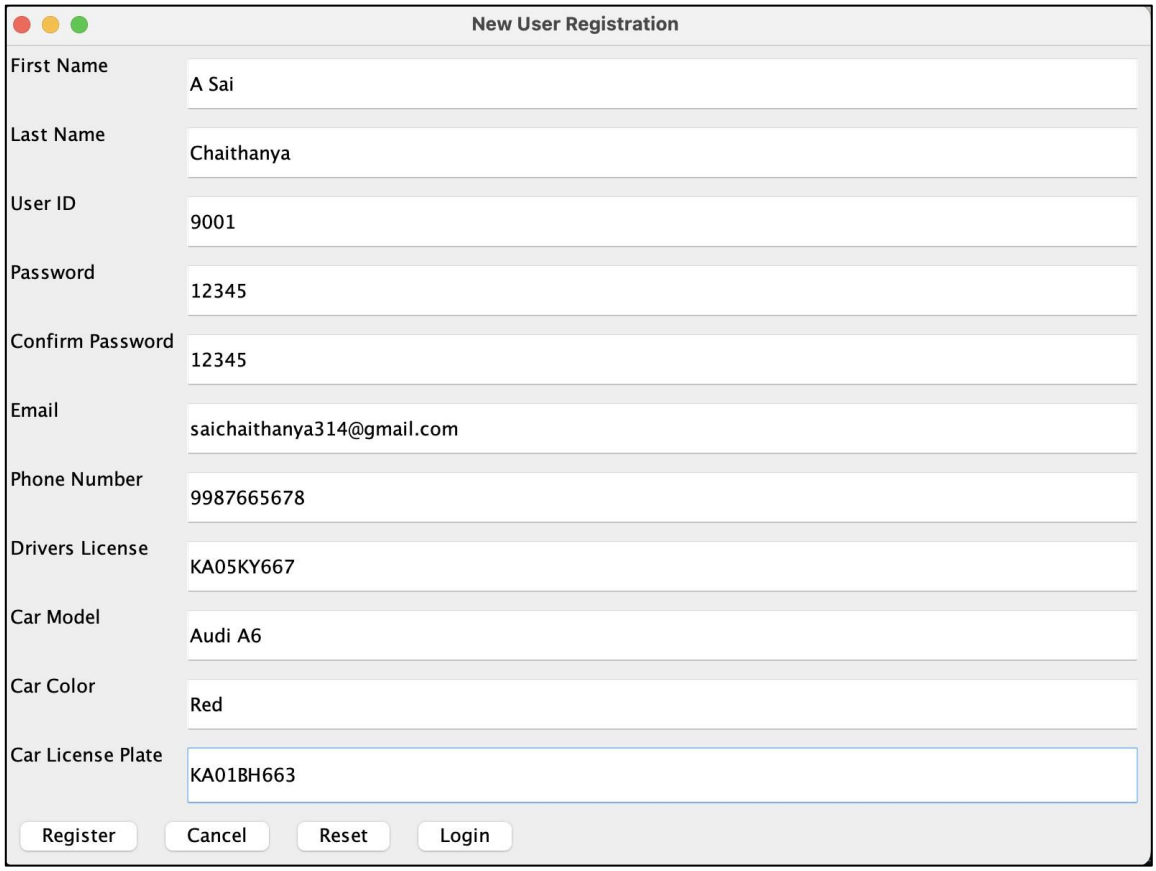


Figure 3: New User Registration

Registration form for the user which takes in all the information related to user and his/her car

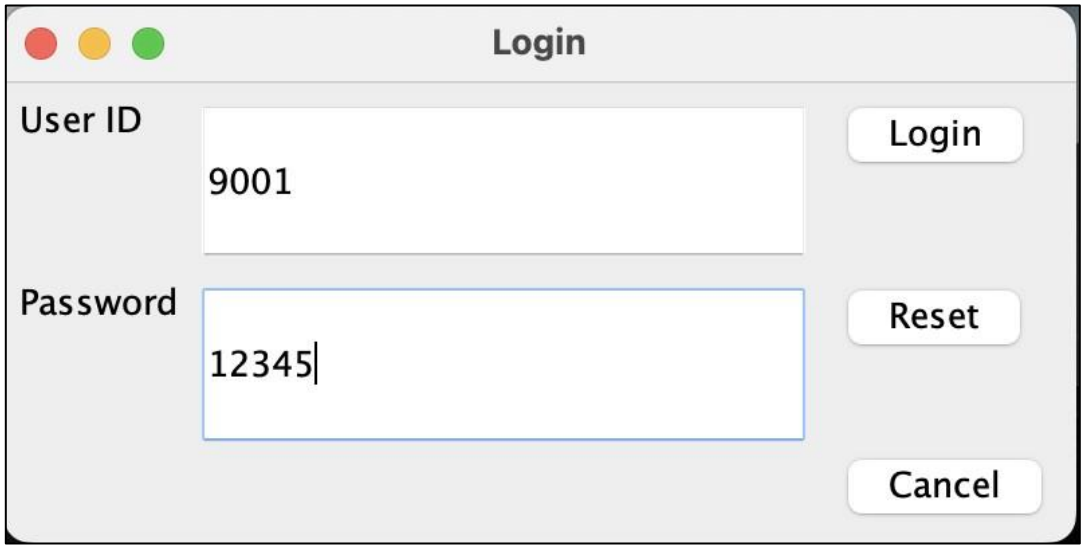


Figure 4: User Login

Login Page to sign in to the application with correct credentials

Figure 5: Geolocation Permission

Permission asked for the user to fetch his geolocation

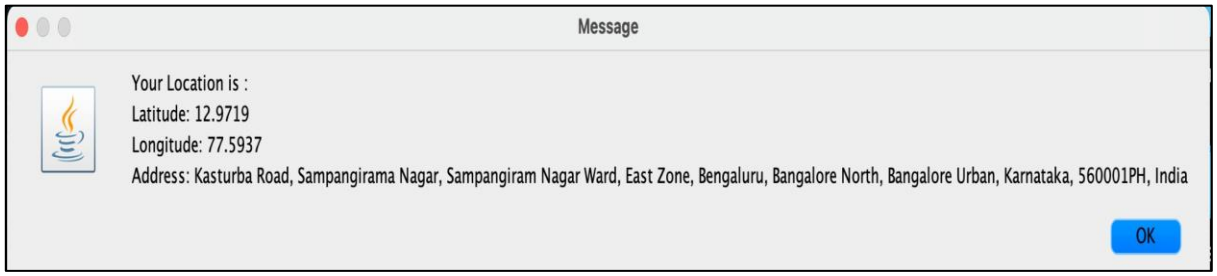


Figure 6: GeoLocation Details

Geolocation details of the user fetched by the PythonAPI

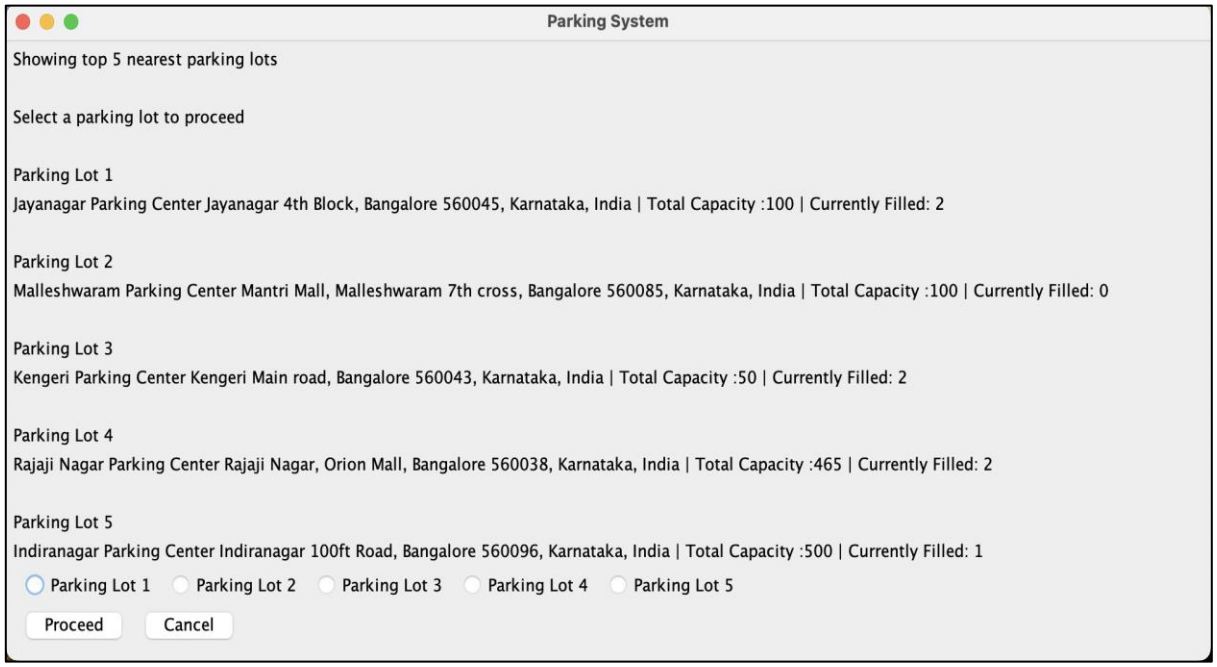


Figure 7: Available Parking lots

Shows top 5 nearest parking locations to the user

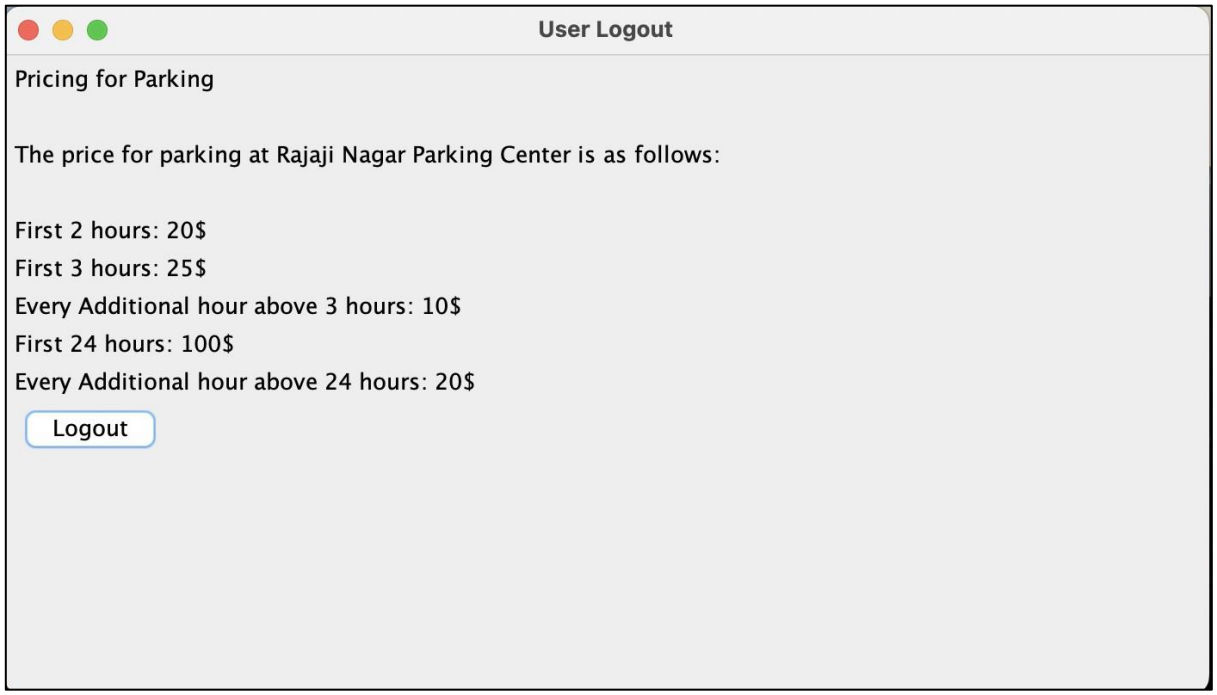


Figure 8: Parking Lot Price Details

Shows the pricing details of parking lot

Figure 9: Welcome back screen

Shows user's details when the user logs back in

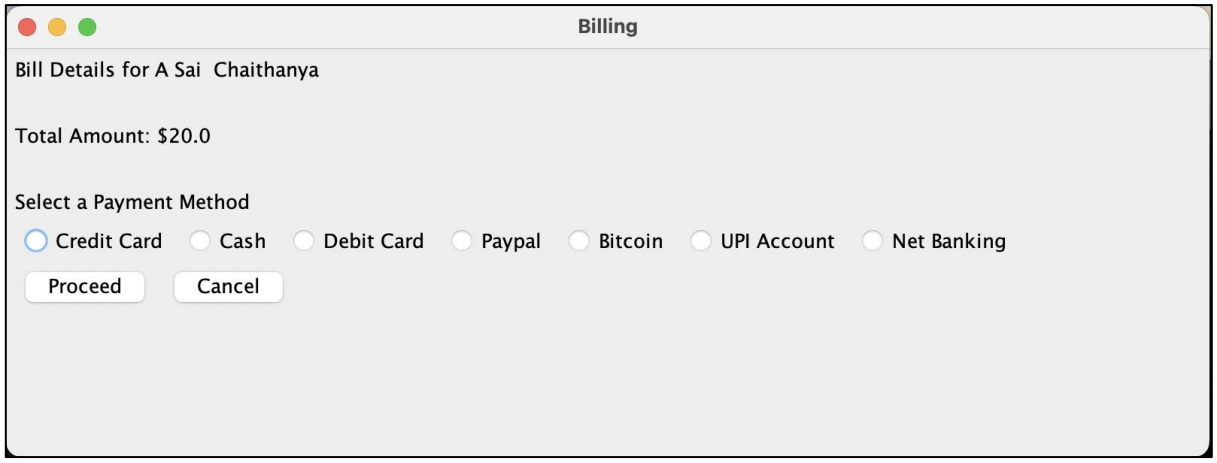


Figure 10: Billing screenshot

Shows the parking charges and asks user for payment method

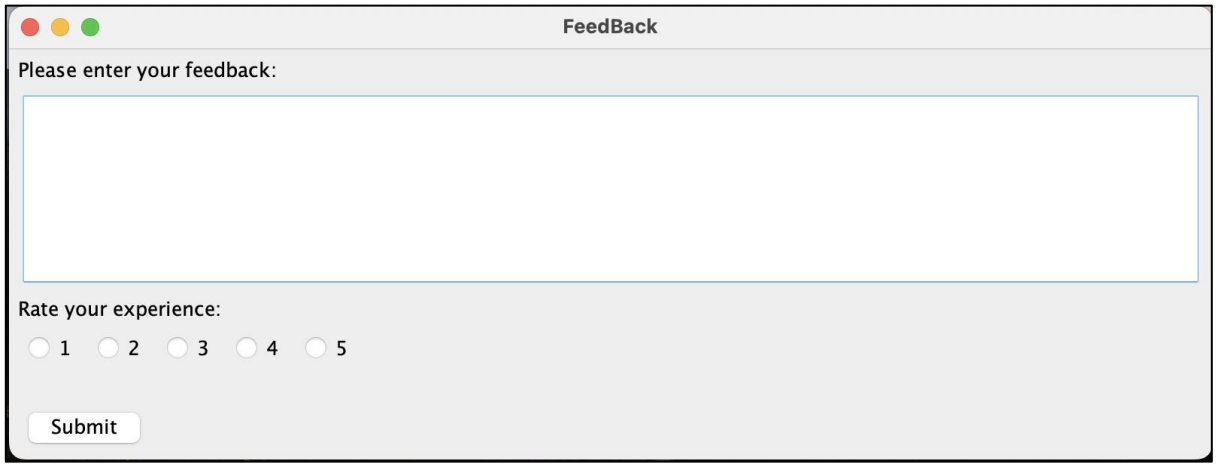


Figure 11: User Feedback screen

User can provide feedback and rate the Parking lot parking experience