Software Requirements Specification

for

System and method for resolving the parking issues using smart vehicle parking.

Version 1.0 approved

Prepared by:

Gurpreet Kaur (2100290120082)

Ishita Srivastava (2100290120088)

Kritika (2100290120098)

KIET Group of Institutions

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this project is to address the parking space scarcity and congestion in urban areas. By aggregating privately owned vacant areas into a unified parking network, we aim to create a seamless and accessible parking experience. Our solution not only alleviates parking saturation but also contributes to Smart City initiatives and sustainable development. Through IoT-enabled modules, users can easily find available parking spots via a web app. Landowners benefit by generating additional revenue from their unused spaces. Safety and transparency are ensured through a rating system and pre-entry verification. This project aims to enhance urban mobility while optimizing existing resources.

1.2. Project Scope

1. Consolidate vacant parking spaces under a centralized platform: *Develop a web application that aggregates and displays available parking spaces from individual sources, such as empty plots, garages, and underutilized areas in private properties.*
2. Community Engagement and Collaboration: *Facilitate connections between individuals willing to rent out parking spaces and those seeking available parking spots, fostering a collaborative parking ecosystem.*
3. Promote Sustainable development: *Utilize existing empty slots, including residential spaces, for parking without additional infrastructure or costs, fostering sustainable development by optimizing available resources and minimizing environmental impact.*
4. Streamline parking management: *Offering personalized booking options based on user preferences like distance and cost, our platform revolutionizes the parking experience. Users can seamlessly reserve tailored parking spaces, boosting urban mobility convenience and efficiency.*
5. Secure Payment Integration: *Design a user-friendly platform that guides users through the verification process and ensures secure payment transactions using industry-standard protocols.*
6. Integration of Review and Rating Service: *Incorporate a review and rating system on the website to allow users to provide feedback on their parking experiences, improving transparency and enhancing user satisfaction.*

1.3. References

IEEE.IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2. Overall Description

2.1 System Environment

1. Entities and Relationships:

- ***Entities:***
    - ***Vehicle Owners:*** *These correspond to the users seeking parking spaces. In the ERD, we'd represent them as an entity called "VehicleOwner." Each VehicleOwner would have attributes like a unique identifier (perhaps a user ID), name, contact information, and any other relevant details.*
    - ***Vacant Space Owners:*** *These are the users who own parking spots and are willing to rent them out. We'll create another entity called "SlotOwner" to represent them. Attributes for SlotOwner might include an ID, name, address, and payment details.*
- ***Relationships:***
    - *Between VehicleOwner and SlotOwner, we have a relationship. Let's call it "Booking." A booking represents the agreement between a VehicleOwner and a SlotOwner for a specific parking spot. It connects the two entities.*
    - *The Booking relationship could have additional attributes, such as booking date, duration, and payment status.*

2. Attributes:

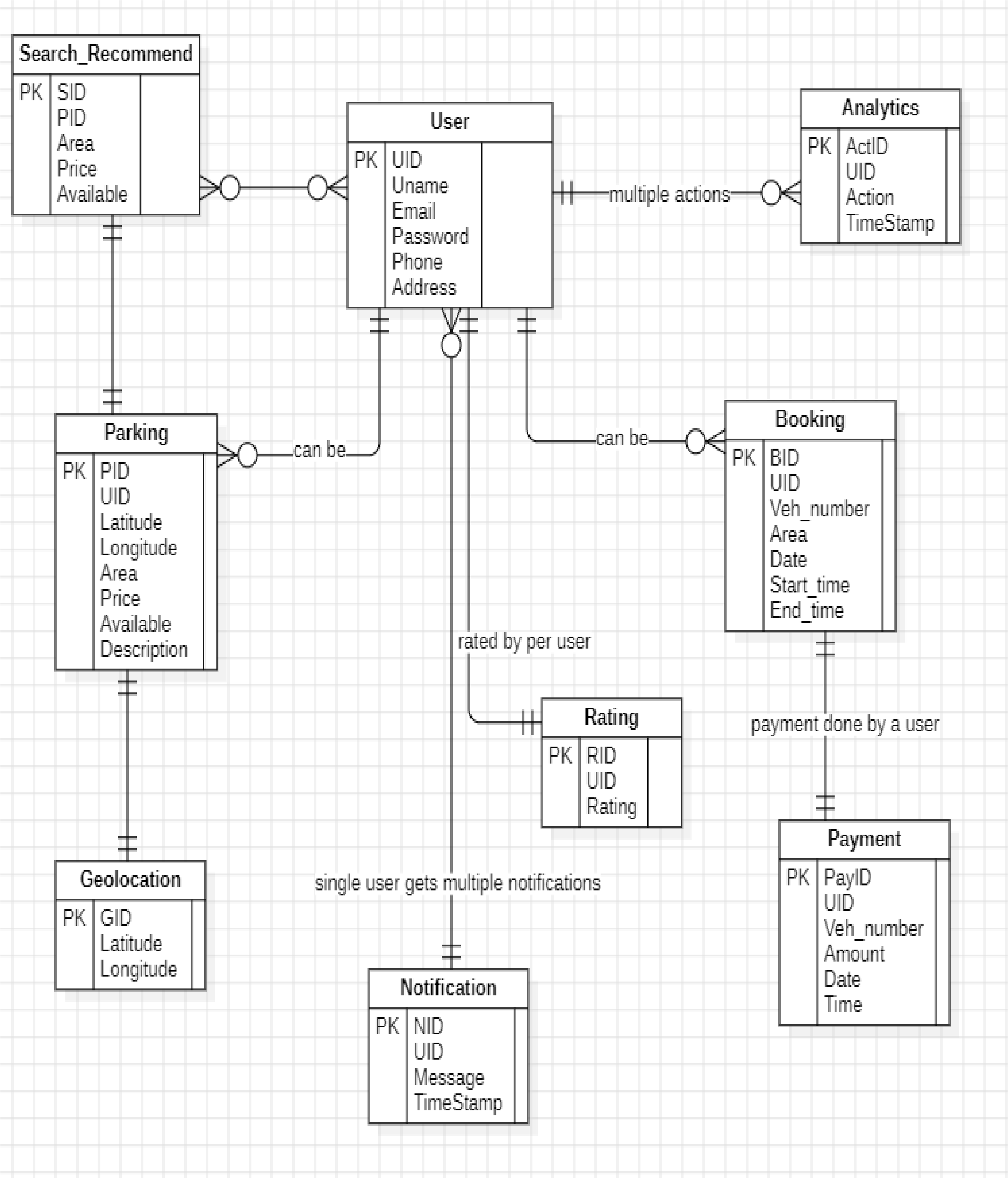
- ***VehicleOwner Attributes:***
    - *User ID*
    - *Name*
    - *Contact details (phone number or email)*
    - *Preferences (e.g., covered parking, charging stations)*
- ***SlotOwner Attributes:***
    - *Space ID*
    - *Location (address or geolocation)*
    - *Availability schedule (when the spot is free)*
    - *Rental price*

3. Geolocation and Navigation:

- *To incorporate geolocation, we can add an attribute to the SlotOwner entity:*
    - ***Location:*** *This attribute stores the coordinates (latitude and longitude) of the parking spot. You can use this information to display available spots on a map and provide navigation to users.*

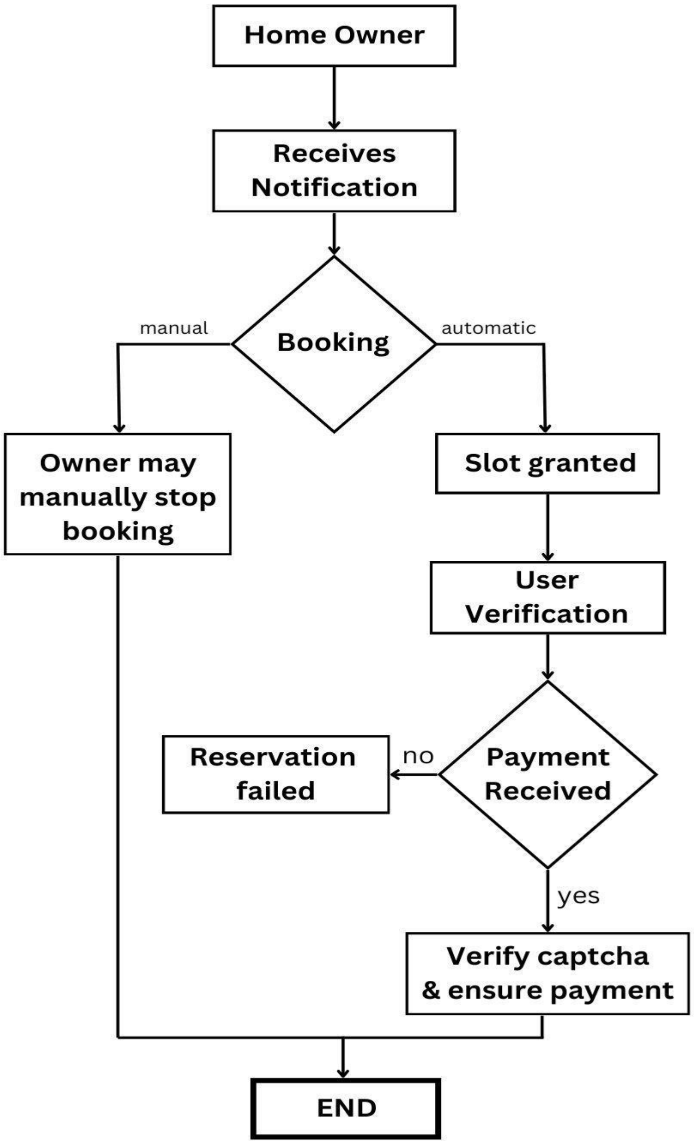
4. Booking and Payment:

- *The Booking relationship handles the process of reserving a parking spot. It connects a specific VehicleOwner with a SlotOwner.*
- *Additional attributes for Booking:*
    - *Booking date and time*
    - *Duration (start and end times)*
    - *Payment status (paid or pending)*


Search_Recommend
PK
SID
PID
Area
Price
Available
User
PK
UID
Uname
Email
Password
Phone
Address
Analytics
PK
ActID
UID
Action
TimeStamp
multiple actions
Parking
PK
PID
UID
Latitude
Longitude
Area
Price
Available
Description
can be
Booking
PK
BID
UID
Veh_number
Area
Date
Start_time
End_time
can be
rated by per user
Rating
PK
RID
UID
Rating
payment done by a user
Payment
PK
PayID
UID
Veh_number
Amount
Date
Time
Geolocation
PK
GID
Latitude
Longitude
single user gets multiple notifications
Notification
PK
NID
UID
Message
TimeStamp

2.2 User Classes and Characteristics

There would be two types of users the one who needs to park their vehicles and the vacant space owners who will act as service providers.



2.2.1 User with vacant space

1. ***Notification of Booking Requests:***
   - *When a potential user (a vehicle owner) wants to book their parking spot, the space owner receives a notification. This notification informs them that someone is interested in using their available space.*
   - *The notification includes details such as the user's name, requested booking date and time, and any specific preferences (if provided by the user).*

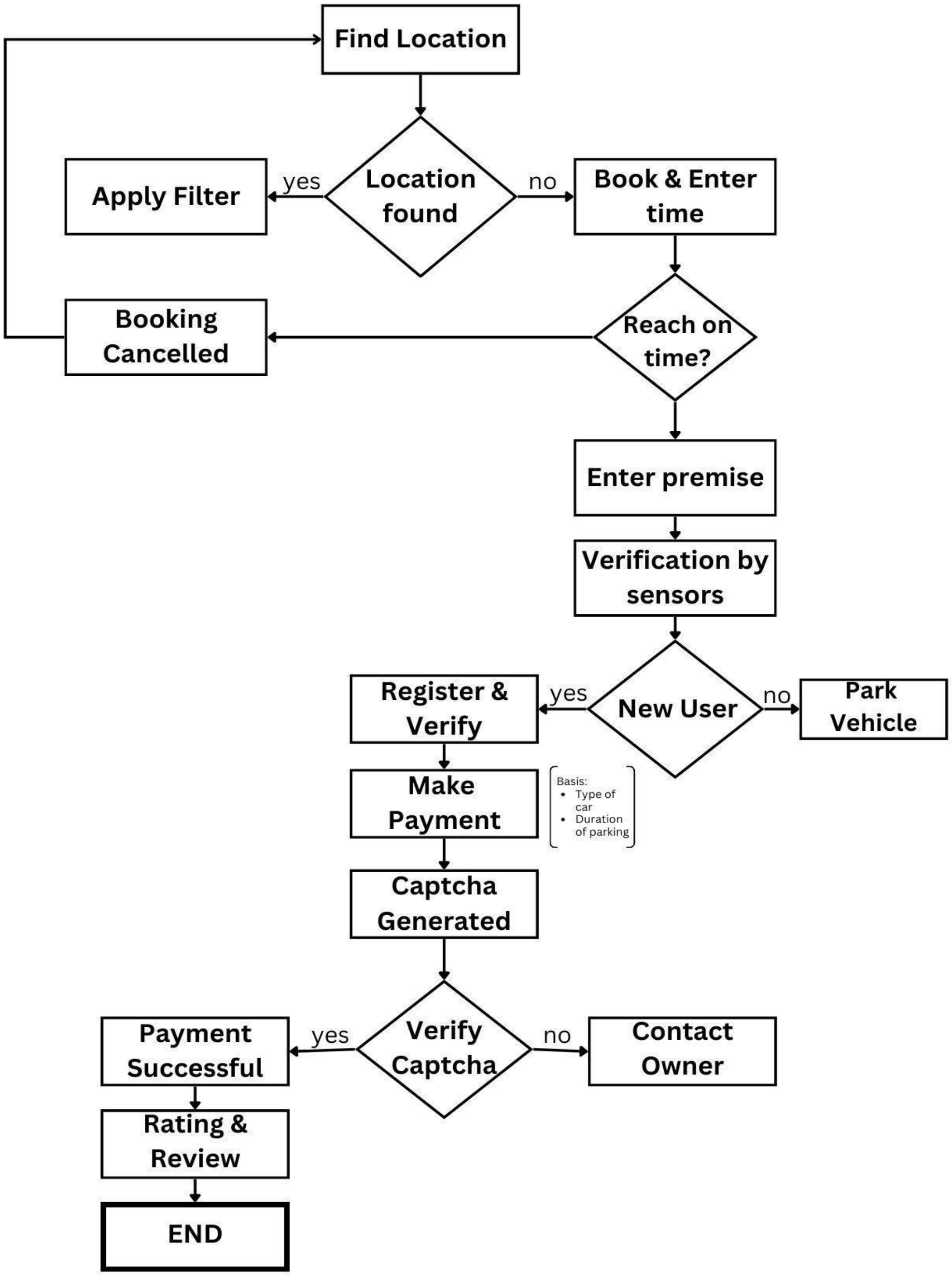
2. ***Manual Approval or Rejection:***
   - *The space owner has two options: manual approval or rejection.*
   - ***Manual Approval:** If the space owner approves the booking request, the slot is officially allotted to the user. The space owner can then proceed to the next steps.*
   - ***Rejection:** If the space owner rejects the request (perhaps because the spot is already booked or for any other reason), the user is promptly notified. In this case, the slot remains available for other potential users.*

3. ***Automatic Mode (Auto-Allotment):***
   - *Alternatively, the parking app can operate in automatic mode.*

4. ***User Verification and Payment:***
   - *Once the slot is allotted (either manually or automatically), the user proceeds to the parking premises.*
   - *The user's identity is verified at the entrance. Verification methods may include scanning a QR code, entering a PIN, or using biometric authentication.*
   - *After successful verification, the user can park their vehicle in the allotted spot.*
   - *Payment is processed automatically (if not already done during the booking process). The app deducts the appropriate amount from the user's preferred payment method (credit card, digital wallet, etc.).*

5. ***Optional: Contacting the Space Owner:***
   - *If the user encounters any issues (e.g., difficulty finding the spot, questions about rules), they can contact the space owner directly through the app.*
   - *The app provides contact details (phone number or messaging) for this purpose.*

6. ***Rating and Feedback (Post-Experience):***
   - *After using the parking spot, the user has the option to rate their experience.*
   - *High ratings contribute to the overall reputation of the space owner and can attract more users in the future.*


Find Location
Location found
yes
Apply Filter
no
Book & Enter time
Reach on time?
Booking Cancelled
Enter premise
Verification by sensors
New User
yes
Register & Verify
no
Park Vehicle
Make Payment
Basis:
Type of car
Duration of parking
Captcha Generated
Verify Captcha
yes
Payment Successful
no
Contact Owner
Rating & Review
END

2.2.2 User with vehicle to park

1. ***Find Location:***
    - *The user begins by searching for an available parking location. This step involves using the app to identify nearby parking spots based on their current geolocation.*
    - *The app may display a map with available parking areas, indicating their proximity to the user.*

2. ***Book & Pay:***
    - *Once the user selects a parking spot, they proceed to book it. This involves confirming the reservation and making a payment.*
    - *The booking process likely includes specifying the desired date, time, and duration for parking.*
    - *Payment can be made through the app using various methods (credit card, digital wallets, etc.).*

3. ***Booking Time Reached?***
    - *The flowchart branches based on whether the user arrives at the booked parking spot within the specified time.*
    - *If the user arrives on time, they proceed to the next steps. If not, the booking may be cancelled automatically, and the spot becomes available for other users.*

4. ***User Verification:***
    - *Upon reaching the parking premises, the user needs to verify their identity. This step ensures that only authorized users can access the parking area.*
    - *Verification methods could include scanning a QR code, entering a PIN, or using biometric authentication.*

5. ***New User vs. Returning User:***
    - *The flowchart distinguishes between new users and returning users.*
    - *New users likely need to register within the app, providing necessary details (name, contact information, vehicle details).*
    - *Returning users skip this step and proceed directly to the next stage.*

6. ***Payment Verification:***
    - *The app verifies the user's payment status. If the payment was successful, the user can proceed.*
    - *If payment verification fails, the user may need to resolve the issue (e.g., update payment details) or choose an alternative payment method.*

7. ***Enter Premises:***
    - *Once verified and payment is confirmed, the user gains access to the parking area.*
    - *They can park their vehicle in the designated spot.*

8. ***Contact Owner (Optional):***
    - *If any issues arise (e.g., spot not available, payment discrepancy), the user can contact the parking space owner.*
    - *Contact details may be provided within the app.*

9. ***Rating & Review (Optional):***
    - *After using the parking spot, users have the option to rate their experience and leave a review.*

10. ***End:***
    - *The process concludes once the user has parked their vehicle and completed any necessary interactions (payment, verification, etc.).*

3. Requirements Specifications

3.1 Hardware Requirements

IOT modules and sensors would be implemented to ensure security and user verification.

- *ESP 8266*
- *Ultrasonic*
- *IR/PIR*
- *Blynk*

3.2 Functional Requirements

1. User Authentication and Registration
Users should be able to create accounts, log in, and manage their profiles.
Implement secure authentication mechanisms (e.g., OAuth, JWT).

2. Parking Spot Search and Navigation
Users can search for available parking spots based on location, time, and other filters.
Integration with maps (Google Maps, OpenStreetMap) for navigation to the selected spot.

3. Real-Time Availability Updates
Display real-time availability status for each parking spot.
Update availability dynamically as spots are booked or become free.

4. Booking and Payment
Users can book parking spots in advance.
Implement secure payment gateways for booking and payment processing.

5. Notifications
Send notifications to users for booking confirmation, reminders, and updates.
Notify users of any changes in availability or booking status.

6. Rating and Reviews
Allow users to rate and review parking spots.
Aggregate ratings to help users make informed decisions.

7. Admin Dashboard

Admins can manage parking spaces, view bookings, and handle disputes.
Monitor system health and performance.

8. IoT Integration

If using IoT sensors, integrate them to detect spot availability.
Update availability status automatically based on sensor data.

9. Security and Privacy

Encrypt user data and ensure secure communication.
Implement role-based access control (user, admin).