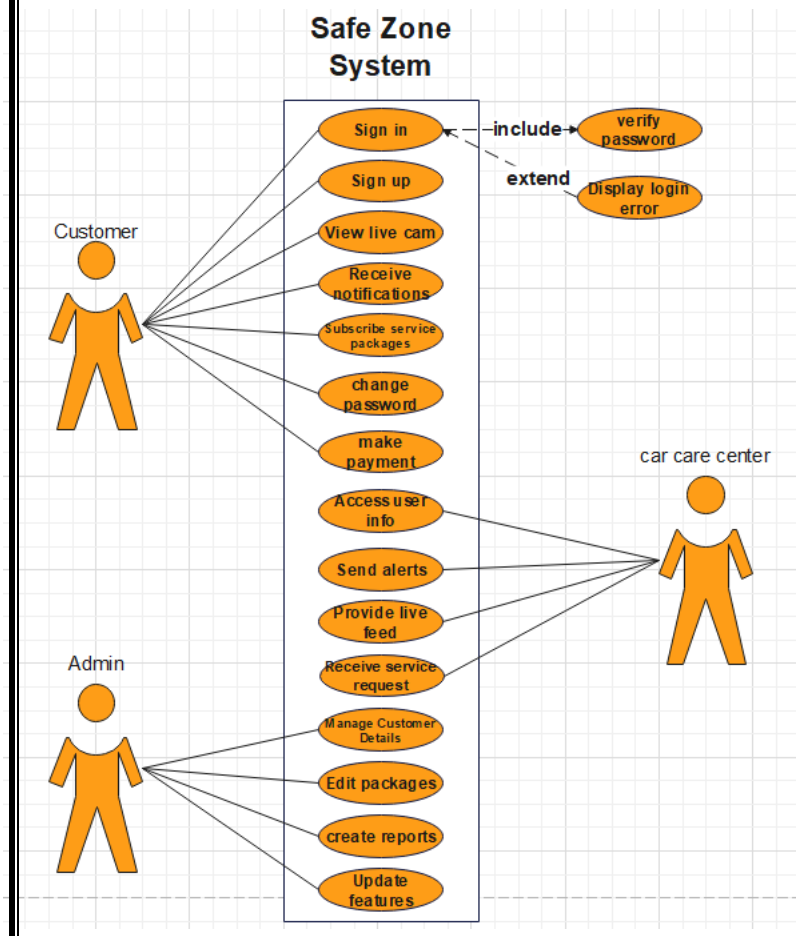


Use Case Diagram for Safe Zone App

Actors

- **Customer**: Users of the app who want to park and monitor their vehicles.

- **Car Care Center**: Provides maintenance services and live feed of the car care process.
- **Admin**: Manages the app's system, service packages, and customer details. ## Use Cases

- **Customer**: -

Signup/Login: To access the app and its features.

- **Reserve Parking**: To find and reserve a parking space.

- **View Live Camera**: To monitor their vehicle remotely. –

Receive Notifications: For updates on vehicle status and parking lot conditions. - **Car Care Center**: -

Receive Service Requests: From customers for vehicle maintenance.

- **Access User Info**: For servicing purposes.

- **Send Alerts**: About vehicle status during servicing.

- **Provide Live Feed**: Of the car care process upon request.

- **Admin**: -

Manage Customer Details: For administrative purposes.

- **Edit Service Packages**: To update service offerings.

- **Generate Reports**: For business analytics.

- **Update System Features**: To enhance app functionality.

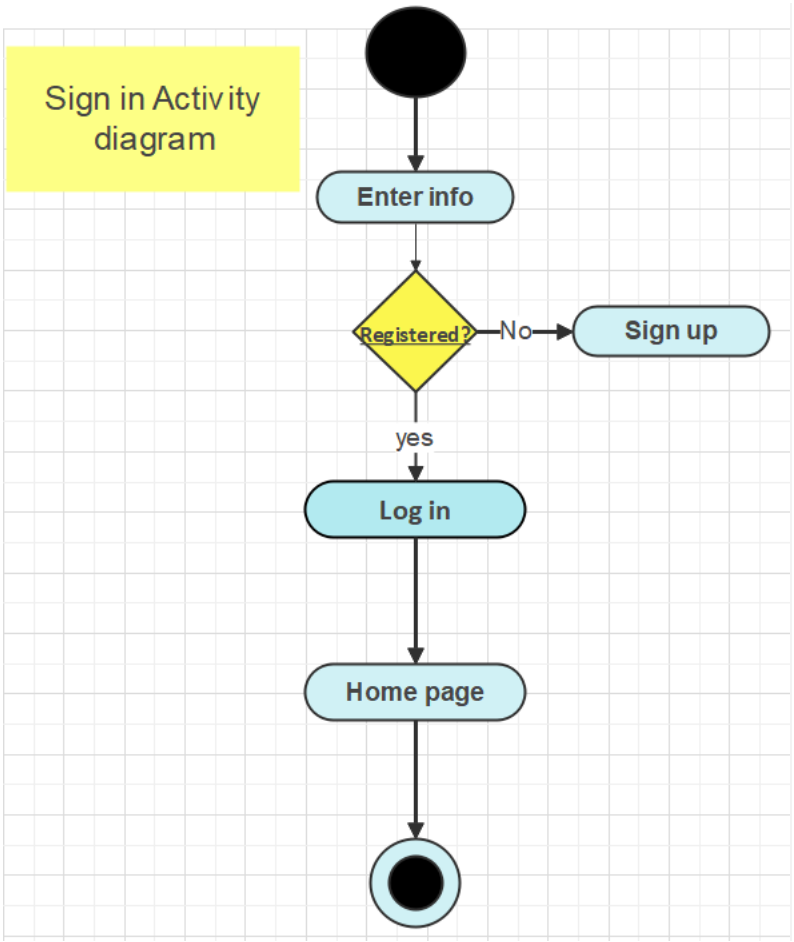
System - **Safe Zone App System**: Facilitates all interactions, processes payments, and provides real-time surveillance capabilities.

Use case description:

use case scenario for the Safe Zone System:

- **Use Case Name**: SAFE ZONE System
- **Use Case ID**: SZS-001
- **Description**: This use case allows customers to monitor their vehicles and request maintenance services through the Safe Zone App. The system facilitates real-time surveillance and service request management.
- **Actors**:
    - **Customer**: Can sign up, log in, subscribe service backages, view live camera feed, and receive notifications, make payment, change password.
    - **Car Care Center**: Receives service requests, accesses user info, sends alerts, and provides live feed.
    - **Admin**: Manages customer details, edits service packages, generates reports, and updates system features.

Sign in Activity diagram:-



- **Start Node**: Begin with a start node to indicate the initiation of the sign-in process.
- **Enter info**: Create an action state where the user enters their username and password.
- **Register**: Add a decision node to check if the entered info are correct.
    - If yes, proceed to the next action state(log in).
    - If no, direct to the action state "Sign up".
- **Sign-In**: If the credentials are verified, show an action state for a sign-in, leading to access to the system's features(home page).
- **Home page**: after signing in the user can view home page and other features included.
- **End Node**: Conclude with an end node indicating the completion of the process.