# TECHNICAL SOLUTION PROPOSAL
## Parking System

July 2023

Presented by Hậu Nguyễn Xuân

# AGENDA


1. Product Requirements
2. Technical Solution
3. DataBase Design
4. Q/A

# 1 —

# Product Requirement

# 1. Product Requirements

## OVERVIEW

### Current state

- There is no modern parking management system that integrates different types of parking lots.
- Drivers have to spend a lot of time and effort to find a suitable parking lot and pay for parking.
- Parking lot owners need to effectively manage fee collection and monitor parking.

### Business objectives

- Create a modern parking platform that integrates different types of parking lots.
- Improve the driver experience by providing easy parking options that meet their needs.
- Provide efficient management tools for parking lot owners.
- Increase efficiency and reduce costs for drivers and parking lot owners.

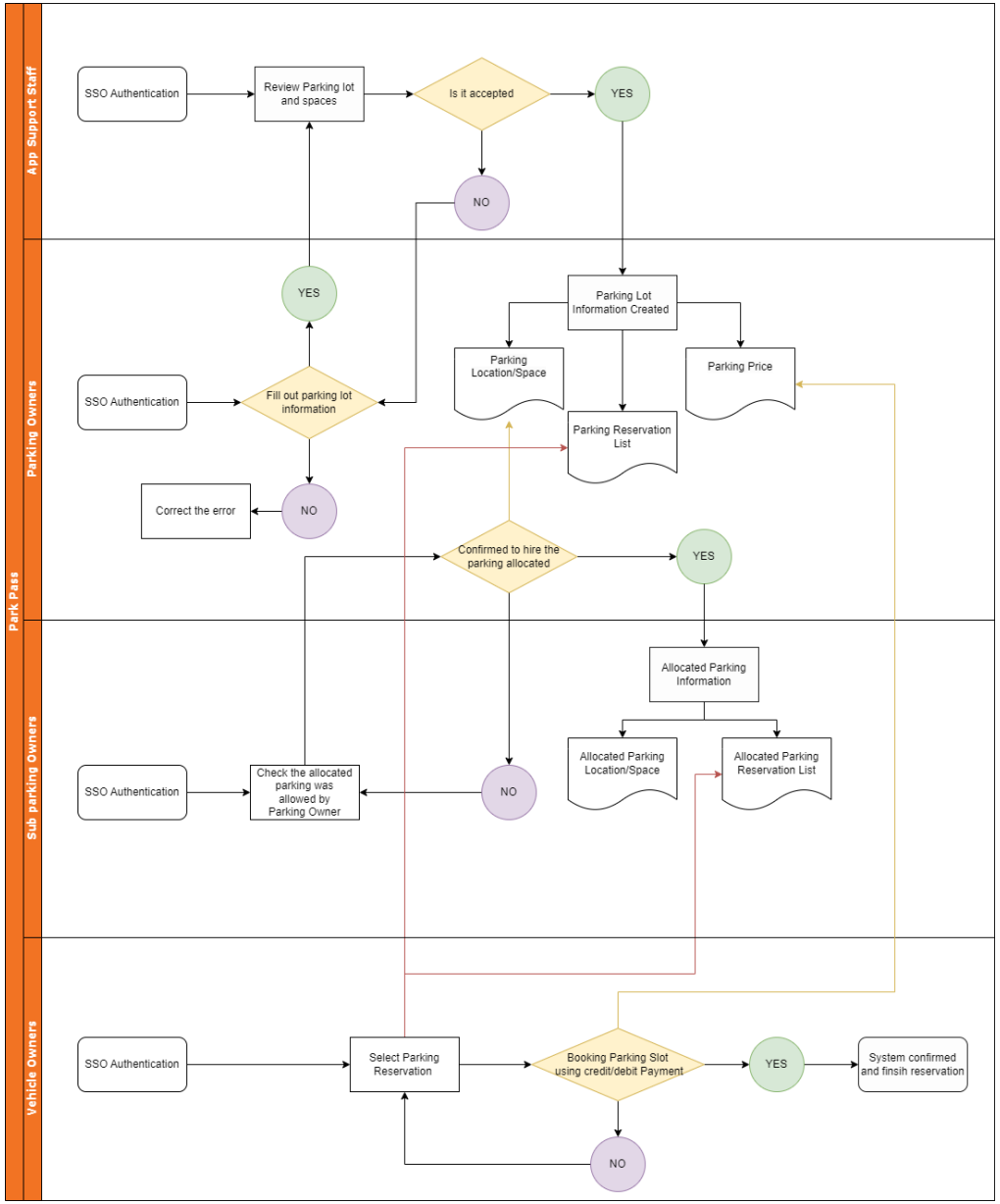
### Dependencies

- Cooperation from parking lot owners to integrate their parking lots into the platform.
- Agreement from drivers to use the application and find parking lots through the platform.
- Regulations and laws related to parking.

### Solution

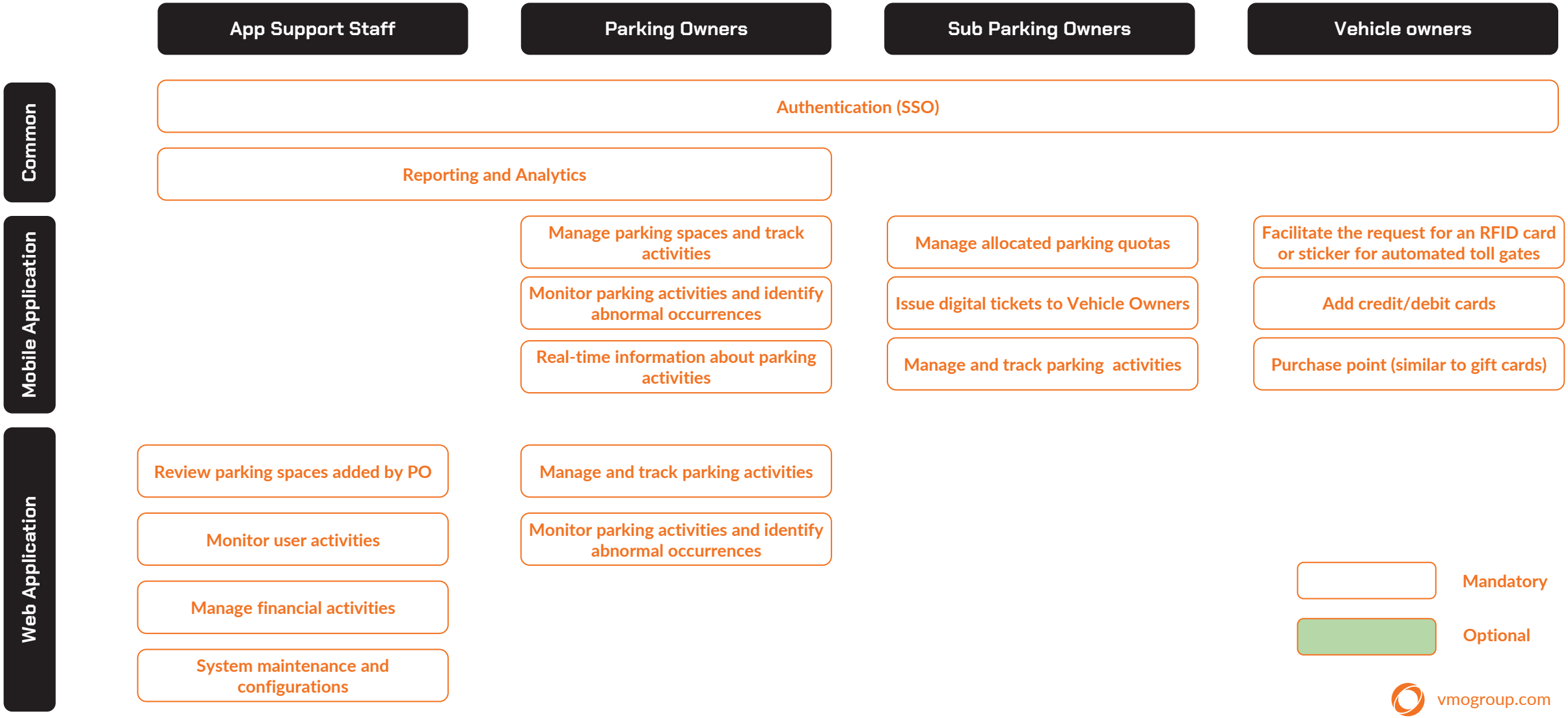
- Create a versatile parking platform that includes a web application and a mobile application.
- Integrate public parking lots, parking lots at buildings and urban areas, as well as private parking lots into the platform.
- Provide features such as parking lot search, pre-booking, online payment, and directions.
- Provide parking lot management features for parking lot owners, including customer information management, fee collection management, reporting, and statistics.
- Provide customer support through the application and other channels such as phone and email.
- Ensure the security and safety of user data and payment information.

# BUSINESS BASIC WORKFLOW - Create new parking space, payment and booking.



## 1. Product Requirements

# FEATURES LIST



# FEATURES LIST (details of App Support Staff)

| Feature | Description |
|---|---|
| Review parking spaces added by parking owners | ▪ Select parking lot, review the number of blocks and the operating company name,....<br>▪ Select block, review it is the back-ground block with no floors or have floors,...<br>▪ Select floor, review it's max height, slots and wings,...<br>▪ Can change the status of the parking lot if something not right,... |
| Monitor user activities, handle user queries, and resolve issues related to the app | ▪ Can check the user's reservation activities, what is actual entry/exit time compare to user's reservation, which vehicle do user usually use to pass, check if customer frequently delay their exit (can refer to penalty field in parking_slip) or not, ..... |
| Manage financial activities, including payment processing, refunds, and transaction logs. | ▪ Search, sort, filter payment processing base on payment, refunds and transaction logs to check if there is any unusual transaction. |
| Access to administrative features for system maintenance and configurations. | ▪ Can change the configuration of the system, like currency, company name, company website, company contact, .... (we will add some common tables like the language, setting,currency,.. later, for now we're focusing on the main table related to the main flow) |

# FEATURES LIST (details of Parking Owners)

| Feature | Description |
|---|---|
| Manage their parking spaces and track their usage, space utilization, and income. | ▪ CRUD parking lot.<br>▪ CRUD block.<br>▪ CRUD floor.<br>▪ CRUD parking slot.<br>▪ Show the all the space was used or empty, the number of block, floor, parking slot on selected parking lot.<br>▪ Show the information of customer who using the parking slot, like vehicle description, reservation schedule,....<br>▪ CRUD the pricing morning/midday/evening of parking lot in normal day or in special day.<br>▪ Show the income base on number of the parking slot was used in day. |
| Monitor parking activities and identify abnormal occurrences. | ▪ Check vehicles overstaying in the parking area base on actual exit time and start timestamp reservation,....<br>▪ Update parking lot, block, floor and packing slots information. |
| View real-time information about their parking spaces, including occupancy and revenue. | ▪ Calculate and show the real-time information. |

# FEATURES LIST (details of Sub Parking Owners)

| Feature | Description |
|---|---|
| Manage allocated parking quotas | ▪ Show all the space that he was allowed to manage, show which was used or empty, the number of block, floor, parking slot on selected parking lot.<br>▪ Show the information of customer who using the parking slot, like vehicle description, reservation schedule,....<br>▪ Show the income base on number of the parking slot was used in day. |
| Issue digital tickets to Vehicle Owners | ▪ Each parking slot has their own parking lot, block, floor and wings, we can use those information to create unique QR Code or pin code for customer. |
| Manage and track parking activities | ▪ Check vehicles overstaying in the parking area base on actual exit time and start timestamp reservation,....<br>▪ Manually update parking slot information if there is any problem. |

# FEATURES LIST (details of Vehicle Owners)

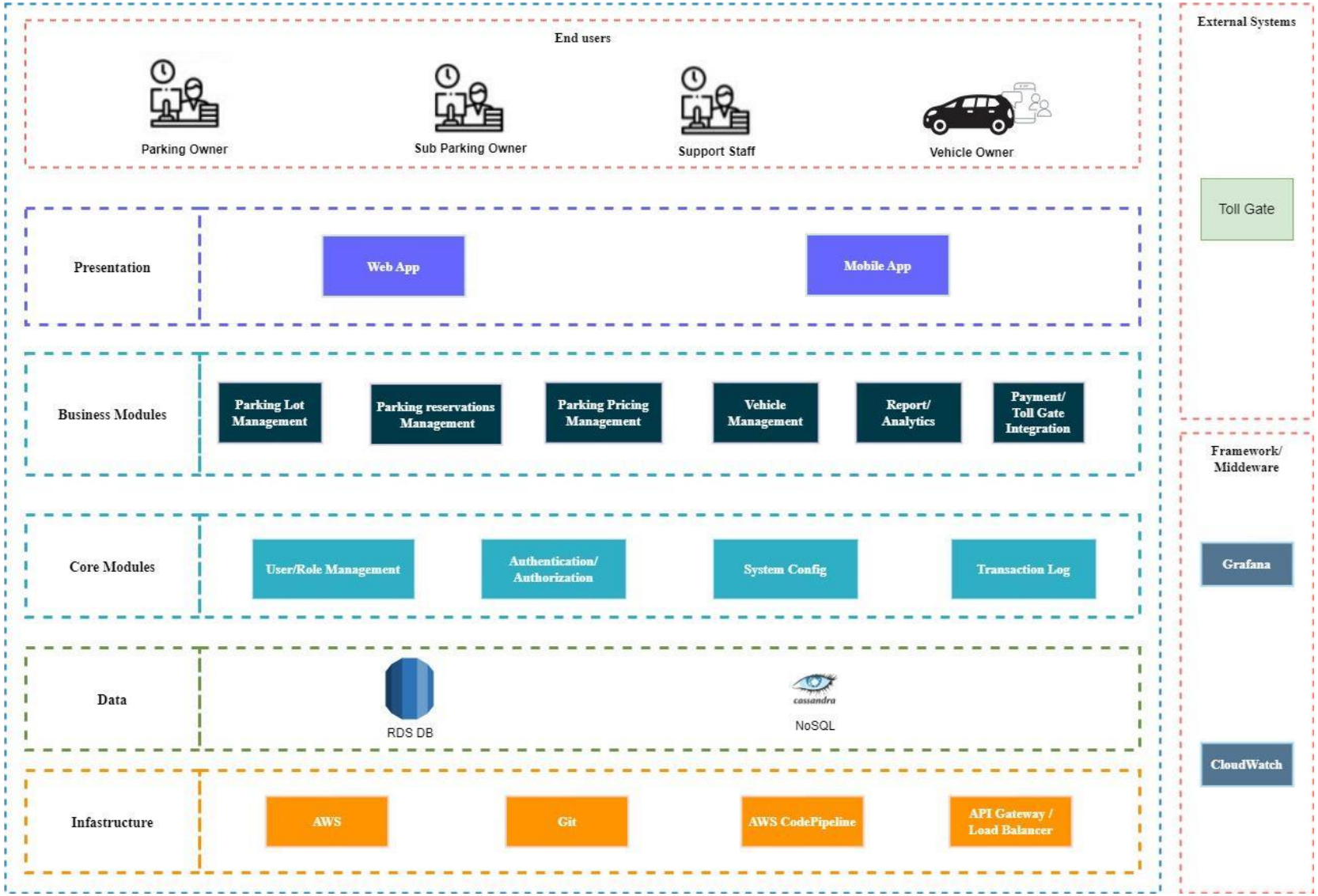
| Feature | Description |
|---|---|
| Facilitate the request for an RFID card or sticker for automated toll gates | ▪ App will integrate with automated toll gates.<br>▪ When a vehicle was using the automated toll gates by RFID card or sticker, the system will check the vehicle's reservation information or was filled by staff.<br>▪ After successfully pass the automated toll gates, the system will persist the actual entry time of the vehicle and waiting for calculate the actual exit time, base on customer's reservation. |
| Add credit/debit cards | ▪ CRUD information about credit/debit cards<br>▪ Can search, filter the history transactions base on day, month, year.. |
| Purchase point (similar to gift cards) | ▪ Each time customer completes parking and paid the parking fee successfully, the system will adding some point for customer.<br>▪ Customers can use point to receive additional benefits.<br>▪ Just like the configuration and setting of the system, we will add tables related to Purchase points later, for now we're focusing on the main table, which affected to the main flow |

# 2 —

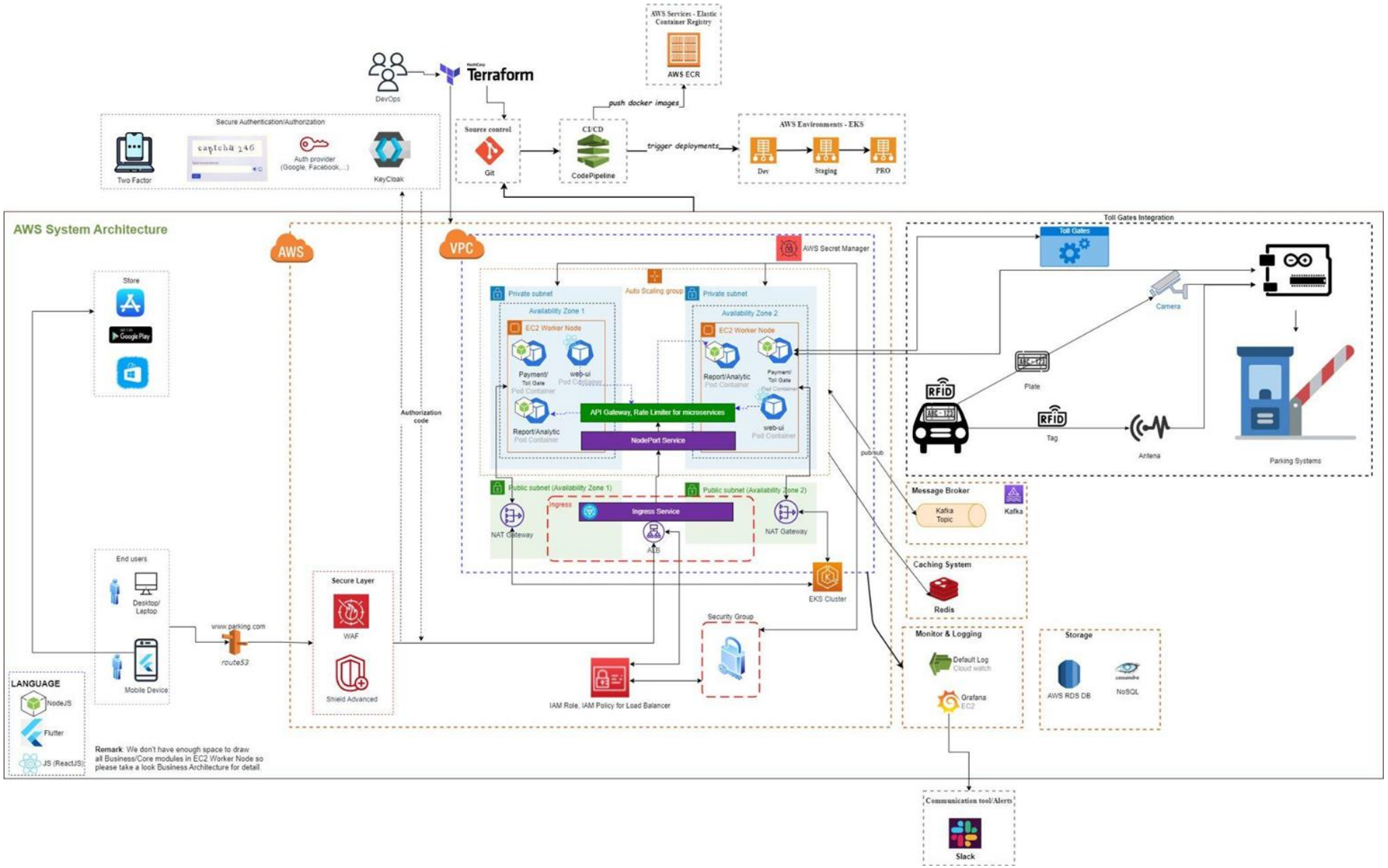
# Technical Solution

## 2. Technical Solution

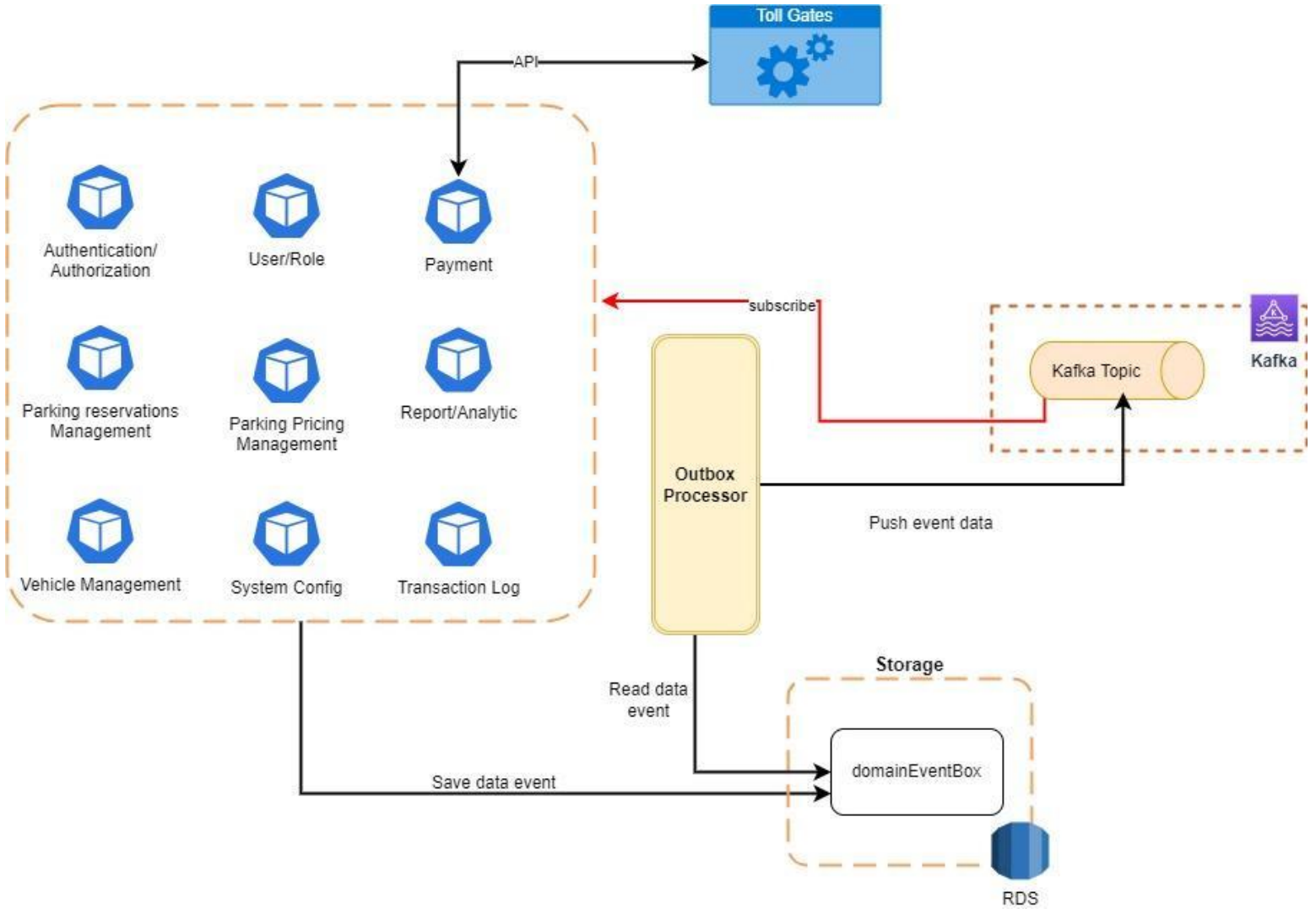
# BUSINESS ARCHITECTURE



# SYSTEM ARCHITECTURE - Cloud AWS



# SYSTEM COMPONENTS




## Outbox Pattern:

- is used to ensure consistency between microservices.

- provide a way to store the events that have occurred in a separate database table, and then publish those events to a message queue. The microservices can then consume the events from the message queue and update their own databases accordingly

# TECH STACK EVALUATION

| Tech stack | Evaluation |
|---|---|
| Message Broker (Kafka) | - Kafka is an open-source, distributed streaming platform, is able to provide a high availability and fault tolerance, low-latency message processing approach just like a traditional message queue. However, it brings additional possibilities that a typical message queuing system can fail to provides.<br>- Kafka provides a distributed, partitioned, replicated commit log service architecture. It provides the functionality of a messaging queue but with a broker pattern of many producers to many consumers at once<br>- Message queues (RabbitMQ,...) are limited because messages are removed from the queue after a single consumer processes them. This technique is incompatible when creating highly scalable applications. Kafka addresses the weaknesses of traditional message queues strategies providing fault-tolerant, high-throughput stream processing<br>- Enable asynchronous communication between components of the system via topics. |
| Redis | - Redis is more powerful, more popular, and better supported than Memcached. Memcached can only do a small fraction of the things Redis can do |
| Grafana | - Grafana is a popular open-source analytics and visualization tool.<br>- Grafana provides an alerting UI that users can use to set and manage alerts on metrics. Grafana sends alerts through several different notifiers, including email, PagerDuty, Slack, texts, and more<br>- Support multiple time-series databases such as InfluxDB, OpenTSDB, and Prometheus. |
| Flutter | - Flutter is a framework (SDK) in which we can create cross-platform applications for iOS, Android, etc.<br>- Reduce overall app development cost. |
| Cassandra (NoSQL) | - Open source, handle a big amount of data, high available by design.<br>- Cassandra's scalability benefits make it popular with companies working with large datasets, that have many concurrent users, and are expecting continued growth.<br>- Continuous availability, high performance and fast.<br>- Use Cassandra to log each transaction logs (payment) and to keep track user activities, report. |

# TECH-STACK

## BACKEND



## WEB APP



## Mobile


## INFRASTRUCTURE







## Monitor and Logging


## SERVICES & TOOLS


## SECURITY

| Encryption | Access controls | Two-factor authentication | Regular security audits |
|---|---|---|---|
| Bcrypt (Password) | RBAC (based on the roles of users) | SMS-based | Vulnerability assessment |
| RSA (SSL/TLS) | ACLs (files, folders, or database tables) | Email-based | Penetration testing |

| Item | Evaluation |
|---|---|
| WAF | - Create security rules that control bot traffic and block common attack patterns such as SQL injection or cross-site scripting (XSS).<br>- Filter out unwanted traffic by IP address or by defining specific patterns. |
| Shield Advanced | - Provide protections against Distributed Denial of Service (DDoS) attacks for AWS resources at the network and transport layers (layer 3 and 4) and the application layer (layer 7). |

# SECURITY

| Item | Evaluation |
|---|---|
| Access control and authentication | - Perform mandatory authentication and authorization requests with sensitive data areas such as user data, transaction data, ...<br>- Implement authentication methods such as multi-factor along with captcha to ensure only the authenticated user has permission to proceed with the necessary steps. |
| Communication security | - Apply security protocol standards/encryption certificates like SSL/TLS on webapp to avoid attacks. It can be done proactively with plans/schedule security with attack tools or hire a 3rd party to do it on dev environment, and it is mandatory to do it when deploy on environments like staging, producion. |
| Code Security | - Run SonarQube, perform common coding security avoidance such as not validate input, catch program errors, pass in parameters that can cause SQL injection, XSS, ... throughout the coding process. You can install plugins into the IDE so that the plugin checks continuously. |
| Database Security | - A proxy server evaluates requests sent from a workstation before accessing the database server.<br>- Avoid using default network ports<br>- Make sure to configure your firewall to cover any security loopholes correctly. It's also essential to keep your firewalls updated, as this protects your site and database against new cyberattack methods.<br>- Use strong user authentication.<br>- Only allowing validated IP addresses (whitelist) to access the database |

## 2. Technical Solution

## SCALABILITY/AVAILABILITY AND INTEGRATION WITH AWS CLOUD SERVICES

| AWS Service | Detail | Scalability/Availability |
|---|---|---|
| EKS (Elastic Kubernetes Service) | - EKS is a fully managed Kubernetes service provided by Amazon Web Services (AWS). It allows users to easily deploy, manage, and scale containerized applications using Kubernetes, an open-source container orchestration system.<br>- EKS supports a variety of Kubernetes features, including automatic scaling, rolling updates, and self-healing, and it integrates with other AWS services such as Application Load Balancing, and AWS Identity and Access Management (IAM).<br>- EKS runs Kubernetes control and data plane instances across multiple Availability Zones to ensure high availability and minimal downtime.<br>- KS uses the architecture of AWS Regions to maintain high availability. |  |
| Auto scaling group | - The Kubernetes Cluster Autoscaler is a popular Cluster Autoscaling solution.<br>- Scalability limits are reached, the Cluster Autoscaler's performance and functionality degrades.<br>- Availability means that pods can be scheduled quickly and without disruption.<br>- EC2 Auto Scaling Groups are configured to launch instances that automatically join their Kubernetes Clusters. |  |
| AWS RDS DB | - When we provision a Multi-AZ DB instance, RDS automatically creates a primary DB instance and synchronously replicates the data to a standby instance in a different Availability Zone. In case of an infrastructure failure, Amazon RDS performs an automatic failover to the standby DB instance for ensuring High Availability of our relational database. |  |
| API Gateway, Rate Limiter | - API gateway is robust, and scalable.<br>- Rate Limiting is important to preventing malicious attacks on your APIs. It applies to the number of calls a user can make to an API within a set time frame, this is used to help control the load that's put on the system. Therefore, it improves the high availability. |  |

# INTEGRATION OF JAVASCRIPT FRAMEWORKS

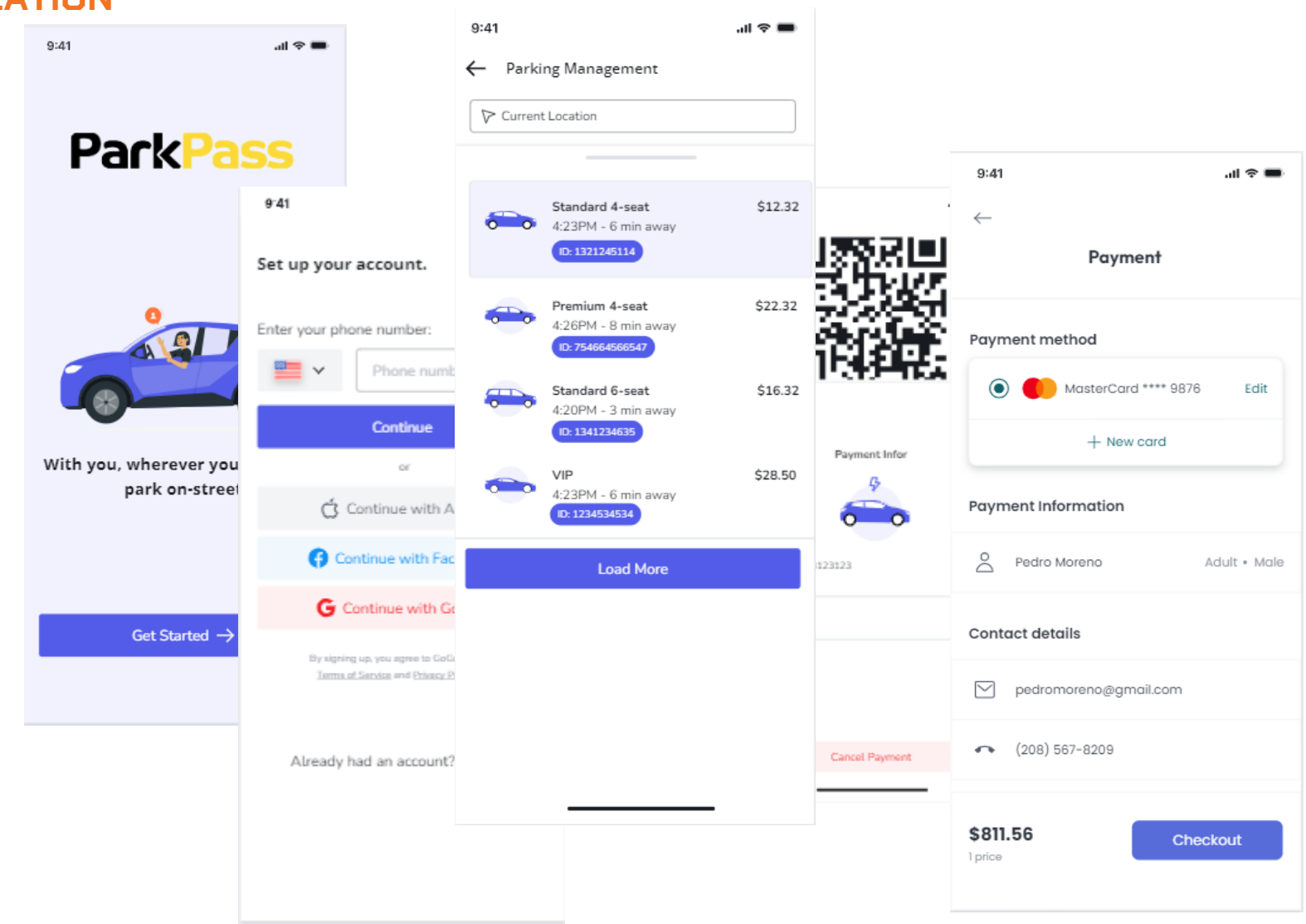
| Item | Evaluation |
|---|---|
| ReactJS | - Easy to learn, so It helps businesses quickly build their projects.<br>- Allow building such high-quality, rich user interfaces through its declarative components, which brings us to our next point.<br>- Components are highly customizable, due to this it speeds up the development process<br>- Has amplified React with the component reusability feature.<br>- Offer fast rendering , so greatly helps businesses in securing the first rank on Google Search Engine Result Page.<br>- React also has very strong community support. Currently, React JS has attained 136,079 stars on Github and 1,331 regular contributors.<br>- Thousands of companies have chosen React JS for their websites and mobile apps. React web framework is currently being utilized by famous companies including Netflix, Paypal, NASA, BBC, Lyft, and New York Times to name just a few. |
| PrimeReact | - Beautiful and diverse design: PrimeReact provides a variety of beautiful and diverse UI elements, including buttons, tables, charts, menus, sliders, input elements and more. These components are well designed and customizable to fit the design of your application.<br>- Flexibility: PrimeReact is highly flexible, allowing you to customize components and integrate them with other technologies such as Redux, GraphQL and Next.js.<br>- Compatible with different browsers: PrimeReact is thoroughly tested to ensure compatibility with different browsers and meets modern web standards.<br>- Multilingual support: PrimeReact supports multiple languages, allowing you to easily display UI elements in different languages.<br>- Massive Community: PrimeReact has a large community with plenty of documentation and support from other developers. This makes learning and developing apps easier.<br>- Support for mobile apps: PrimeReact provides UI components that are compatible with mobile apps, making it easier to develop mobile apps. |

## 2. Technical Solution

# MOBILE APPLICATION - UI VISUALIZATION

**MOBILE APP**

Flutter



# REPORTING AND ANALYTICS


| Tech stack | Evaluation |
|---|---|
| Jasper Report | - JasperReports is an open source reporting library.<br>- Widely used, also large community. |
| Apache Spark | - Spark is a unified **analytics** engine for large-scale data processing.<br>- It provides high-level APIs in Scala, Java, Python, and R, and an optimized engine that supports general computation graphs for data analysis. |

# REPORTING AND ANALYTICS

| Name | Description |
|---|---|
| Parking violations reports | Parking violations reports provide information on parking violations, such as the number of violations, the types of violations, and the areas where violations occur most frequently. This information can help parking owners to identify areas where enforcement may need to be increased or adjusted. |
| Occupancy reports | These reports provide information on the occupancy rates of different parking areas, such as the number of vehicles parked, the duration of parking, and the frequency of use. This information can help parking owners to optimize the use of their parking spaces and identify areas for improvement. |
| Maintenance reports | Maintenance reports provide information on the maintenance needs of the parking system, such as the frequency of repairs and maintenance, the types of repairs needed, and the costs associated with maintenance. This information can help parking owners to optimize their maintenance schedules and reduce maintenance costs. |
| Revenue reports | Revenue reports provide information on the revenue generated by the parking system, including total revenue, revenue per parking space, and revenue by time period. This information can help parking owners to track the financial performance of their parking operations and make informed decisions about pricing and revenue optimization |

# REPORTING AND ANALYTICS

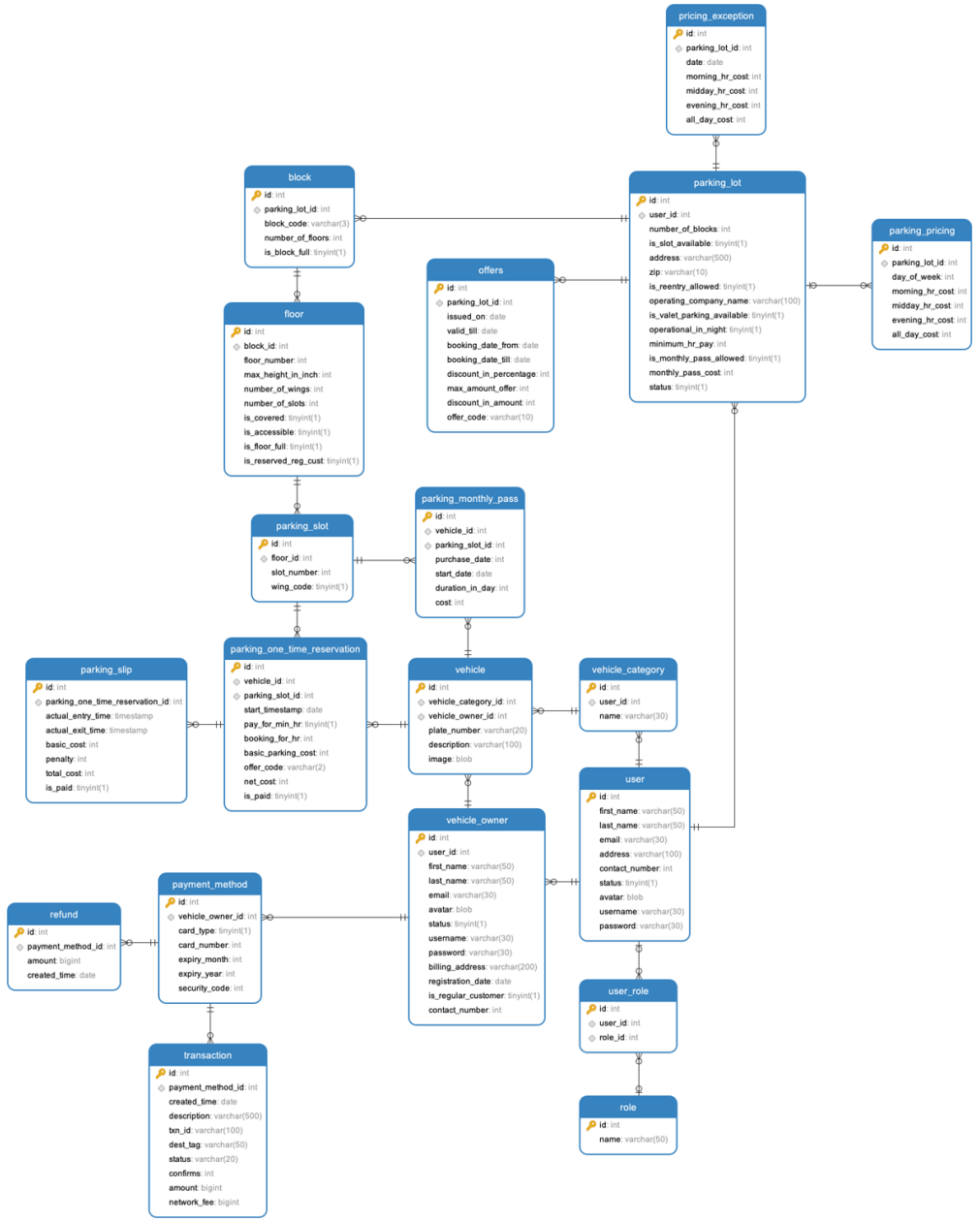
| Name | Description |
|---|---|
| User behavior analytics | User behavior analytics provide insights into the behavior of parking users, such as the times and days of peak usage, the average duration of parking, and the types of vehicles parked. This information can help parking owners to identify trends and patterns in user behavior and adjust their operations accordingly. |
| Parking duration analytics | Parking duration analytics would provide insights into how long vehicles are parked in specific areas, allowing parking owners to optimize pricing and availability based on demand. |
| Revenue forecasting | Revenue forecasting would use historical data to predict future revenue and occupancy rates, allowing parking owners to make informed decisions about pricing and capacity planning. |
| Heat maps | Heat maps would provide visual representations of parking usage and occupancy, allowing parking owners to quickly identify areas of high demand and adjust their operations accordingly. |

# SERVER CONFIGURATION

| Service Name | Hourly Cost (USD) | Configuration |
|---|---|---|
| Amazon EC2 | 0.0528 | ➢ 1 instance (Asia Pacific (HongKong))<br>➢ T3.medium, 2vCPUs, 4GB Mem, On demand.<br>➢ EBS gp2, 100GB<br>➢ Enabled Monitoring |
| Amazon RDS for PostgreSQL | 0.112 | ➢ 1 instance, (Asia Pacific (HongKong))<br>➢ Db.t3.med, 2vCPUs, 4GB Mem, On demand.<br>➢ Storage gp2, 30GB |
| EKS | 0.1 | ➢ 1 Kubernetes cluster. |
| WAF | 0.007 (Web ACL)<br>0.0014 (Rule) | ➢ Asia Pacific (HongKong)<br>➢ 0.60 per 1 million requests (resource Type Request), On demand. |

## 3. Database Design

# DATA MODEL - ENTITY RELATIONSHIP




Let’s build great
things together!