

Breakdown:

1. Short description
2. Top Parts of the application - Frontend, backend, database, rest API, apis involved, cloud deployment
3. Components and tech stacks
4. Modules - deeper level for each component and all involved functions.

Give estimated time of completion (2 min) and what you're going to cover like the app overview, architecture and the system flow. I'll take 2 mins and explain the basic app overview, architecture, and system flow.

Mobile Parking App for Singapore Residential Streets w/ AI Co-Pilot

Short Description

A crowd-sourced parking **mobile application** for residential / non-commercial spots in Singapore. Users can access and update information on available street parking slots and also get directions to the parking locations. The app features an AI chat bot where the users can interact with the app using natural language. I used the expo platform which can run the app both on android and iphone.

Short description, Problem it is solving, components and tech stacks, system flow

The backend uses Dijkstra's algorithm to optimize routes and provides a Node.js-based REST API server to process requests. Machine learning models predict parking availability.

Problem it is Solving

My friend used to visit me often and he couldn't find a parking spot nearby and had to keep crossing streets to park. The app addresses the challenge of finding parking slots on Singapore residential streets by providing real-time updates on parking availability.

Components

- **Frontend:**

- **Technology:** React Native, Expo
- **Features:** User interface for browsing and updating parking slots, AI co-pilot for NLP interactions.

- **Backend:** Node.js-based REST API server to process requests.

- **Technology:** Node.js, Express.js
- **Features:** REST API server, Dijkstra's algorithm for route optimization, machine learning for parking predictions.

- **Server:**

- **Hosting:** AWS cloud
- **Features:** Real-time data processing, caching to reduce external mapping service reliance.

- **APIs:**

- **Internal:** Custom APIs for parking data and route optimization.
- **External:** OpenStreetMap API for map data, OpenAI and Anthropic Claude APIs for NLP.

- I have deployed the frontend using expo and the backend using AWS cloud.

System Flow:

Users interact with the app, which sends data requests to the backend using REST protocol. The backend processes these requests, performs necessary operations such as crud operations for updating and storing data into the sqlite database. The database stores relevant data, and external APIs provide additional functionalities such as mapping and NLP. The AI co-pilot helps users via chat, enhancing the overall user experience.

the nodejs server to talks to the open ai api to perform natural language processing and allows users to interact with the chatbot. It also uses over pass api to get all the relevant road data.