# Computer Engineering Department

# ParkNow

## Project Advisor: Vinodh Gopinath

Malik, Kunjan (MS Software Engineering) ,
Nayak,Praveen (MS Software Engineering)
Shrivastav, Shivam (MS Software Engineering)
Savant Yadnyshree (MS Software Engineering)

## Introduction

ParkNow provides users a convenient way to find a parking space using their mobile devices. In this project, we propose a crowdsourced data capture approach in combination with mobile phone GPS signals to provide accurate, reliable real-time parking availability information.

Feature in the project :

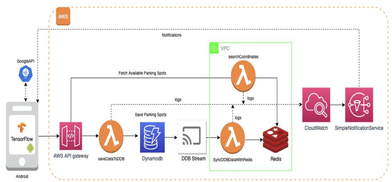
1. Find the free parking spots at the destination.
2. Save the current parking spot.
3. Release the current parking spot.
4. Navigate to the parking spot.


The search for a parking space adds up to an estimated $1,374 per driver in wasted time, fuel, and emissions according to the analysis made by INRIX 2019 references [5]. Moreover, finding parking space during a pandemic has become more difficult because focus has shifted to personal usage of vehicles over public commute. This review aims to identify the major pain points for a hassle free parking system

## Methodology

**Project Architecture:**

The project is divided into 3 main layers
1.UI Layer
2.Backend service built in AWS Lambda.
3. Database

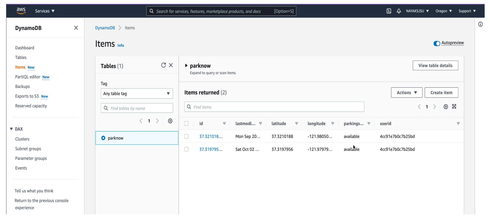



The UI Layer receives input from the user or fetches the user's current location. Based on the destination address the call goes to the backend services which fetches the parking spots nearby the specified radius.The results are sorted in the ascending order.

## Methodology

Once the user selects the desired parking spot, the google maps API is called which helps the user to navigate to the parking spot.

Database :
DynamoDB: as our primary database being the source of truth for storing the parking coordinates. We also attached a database update trigger to DynamoDB using Lambda function to maintain the latest copy of data to an in-memory database, Redis, to serve all the read requests for our APIs. The data flow from DynamoDB to Redis will happen in real time. Table Structure of DB

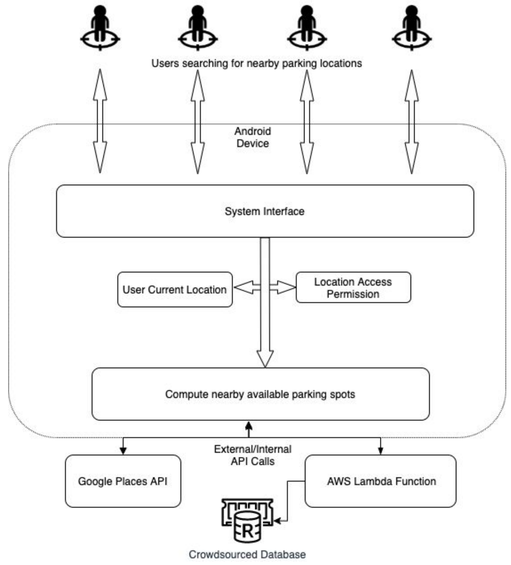



Redis: Redis is an in-memory key-value based data structure store, used to manage cache mechanisms with optional durability. We use Redis as our secondary database, to serve operations in our application such as updating the status of the parking spot and searching the parking spots to reduce the read and write loads on the primary database. Redis serves as the best choice for our use case as it supports GeoSpatial queries which helps us search the parking spots nearby the provided location coordinates.

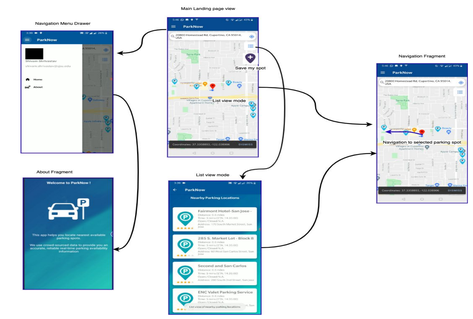

## Analysis and Results

The fundamental part of the ParkNow application is to provide users with a parking spot using the crowdsourced data technique.

- Fetching the user's current location or allowing the user to input the destination address. For this the application requires users GPS permissions.
- The Nearby parking spots are fetched calling the API that fetched the results from the Database and also from the Google Places API. The final result has the integration of the parking spots from the database and the Places API.




When the user selects the parking spot from the list of parking spots. A Geofence is added around the location coordinates. This Geofence API is used to detect when the user enters or exits the parking spots. When the user arrives at the location, a notification is triggered to the user's device and when the user accepts it the Database is updated with the occupied status. Similarly when the user is exiting the parking spot a notification is triggered to confirm the user is leaving. Based on the user action the database is updated with available status against the parking spot.




## Summary/Conclusions

We have developed an Android Mobile Application that has the capability to use custom crowd sourced data to identify the nearby parking spots. We have demonstrated how cloud providers take on all of the hassles associated with infrastructure, maintenance and utility management for the servers which helped us easily manage the entire architecture at one place without using any additional 3rd party sensors. Eventually the goal is to reduce the carbon footprint and save money on fuel.

## Key References

[1] Nie, Y., Xu, K., Chen, H. et al (2019) Crowd-parking: A Annual Conference of the IEEE Industrial Electronics Society, Vol. 1. IEEE, 2019.

[2] Chen, Mingsong; Fang, Runze; Peng, Lei (2020) Exploration of Parking Guidance based on Vehicle Crowdsourcing, IEEE, 2020.

[3] Bashar M. Nema, Ali Nafaa Jaafar (2019) Geolocation Android Mobile Phones Using GSM/UMTS, Baghdad Science Journal, Vol 16(1), 2019.

[4] Muharum A.M., Joyejob V.T., Hurbungs V. ,Beeharry Y. Enersave API:Android-Based Power- Saving Framework, Future Computing and informatics Journal, Vol 2(1), 2017.

[5] Roadway traffic and parking information- INRIX https://inrix.com/scorecard/

## Acknowledgements

We would like to extend our sincere thanks to our Project Advisor, Prof. Vinodh Gopinath, who helped us throughout our project by giving us needed direction and guidance. We would also like to thank Dr. Dan Harkey for the encouragement.