

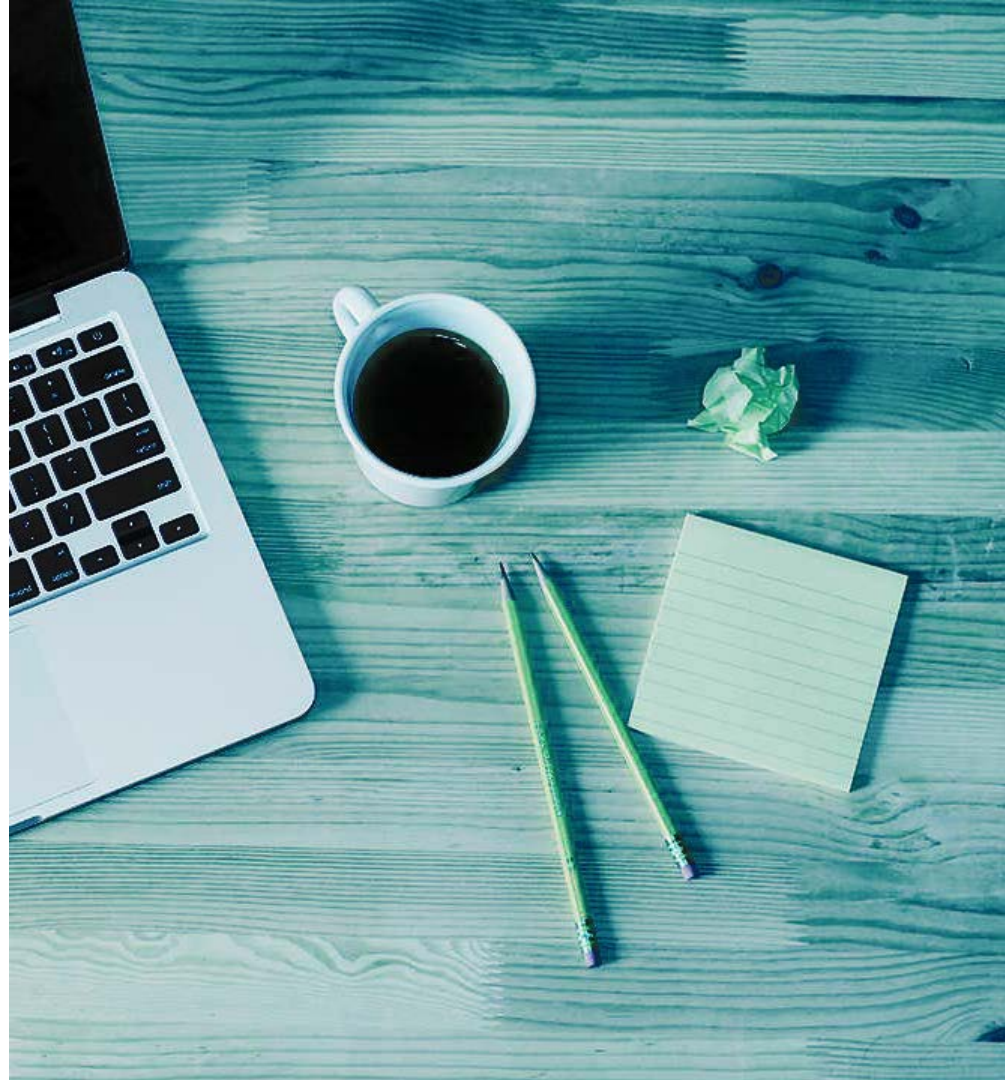


GOPARK

Automated Parking Allocation System



Meghnad Saha Institute of Technology



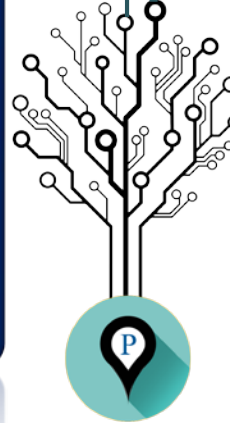
Parking Problem

Parking availability has always been a problem for all car drivers and owners throughout time.

- Parking is a menace in an urban environment without a systematic solution



- Vacant Parking Slots is hard to find in urban areas

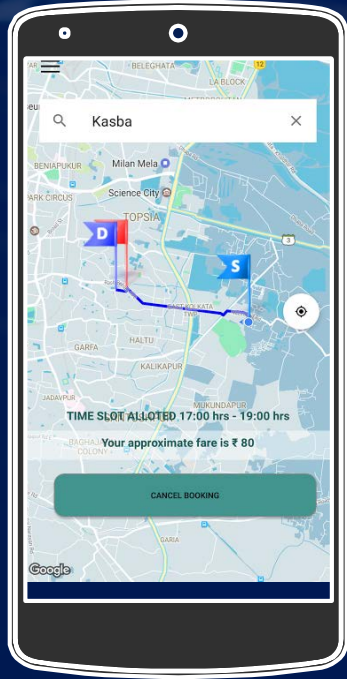


DEMAND

SUPPLY

The Problem >>

The R devance➤



“Why Is **Solving It** Important?”

- Will Assure Users A Parking Slot from Beforehand- **Save Time**



- Avoid all chaos and unruly parking on the streets- **Orderly Parking**

- Let User Enjoy a hassle free drive – **Tension Free**



“Previous Solutions and Contributions toward the Problem”

Automated Parking Slot Allocation using RFID Technology

K.Ganesan, and K.Vignesh, School of Computing Sciences, Vellore Institute of Technology, Vellore – 632 014, India.



Reservation Based Parking System with Dynamic Slot Allocation, Hina Kousar, Kavitha Kumar, Shoney Sebastian.

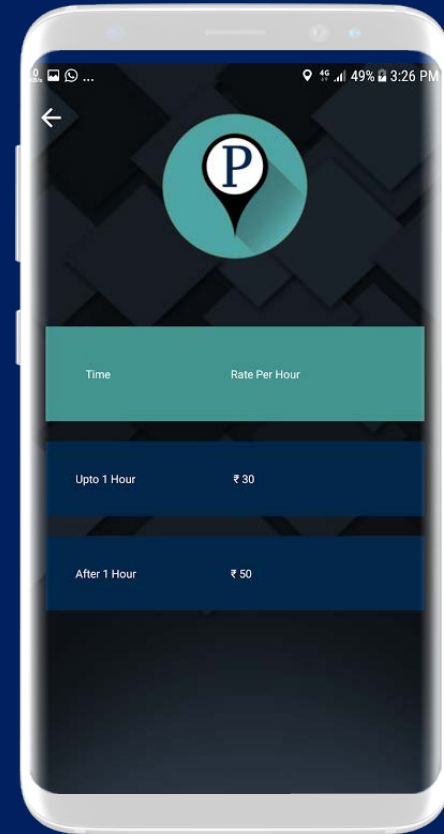
MS Computer Science Student, Dept. of Computer Science, Christ University, Bangalore, India.

How we plan to solve the issue?

Introducing!

g o Park

Automated Parking Allocation System



GoPark gives you
best of both worlds

Our Parking System can be broken down to Two Applications- User Application & Partner Application



USER APP

The Interface for users to book or schedule a slot and add cars and give feedback and use other features



PARTNER APP

The Application through which parking slot owners can add and update information regarding their slots

DEMAND

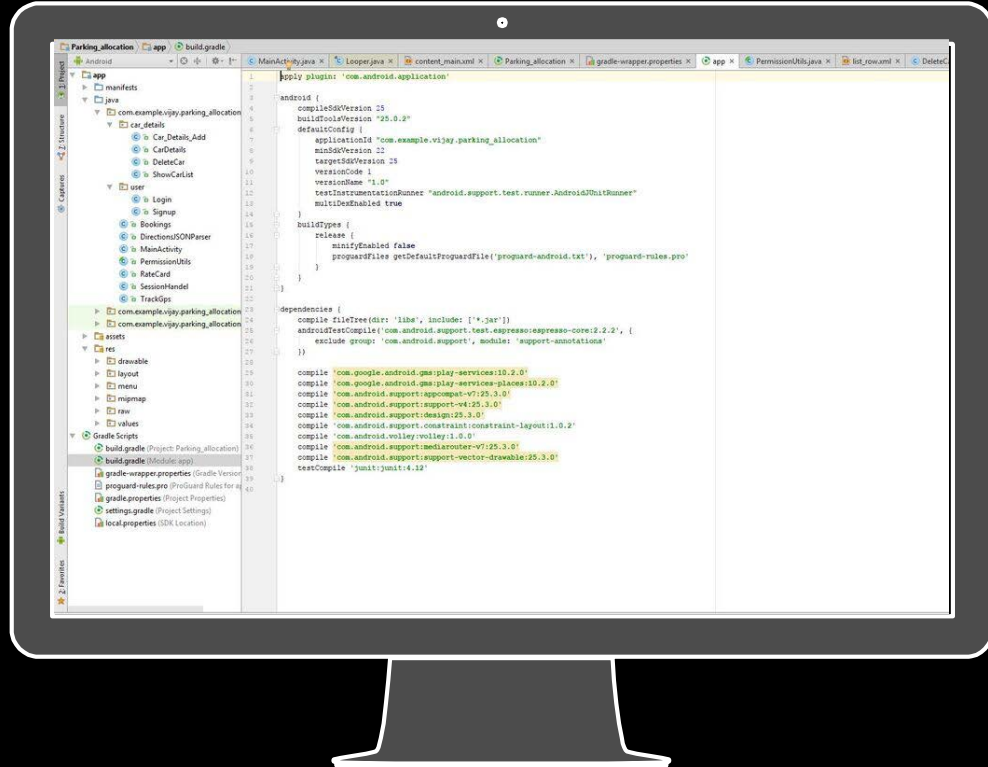
SUPPLY



Technical Requirements

For Android Application

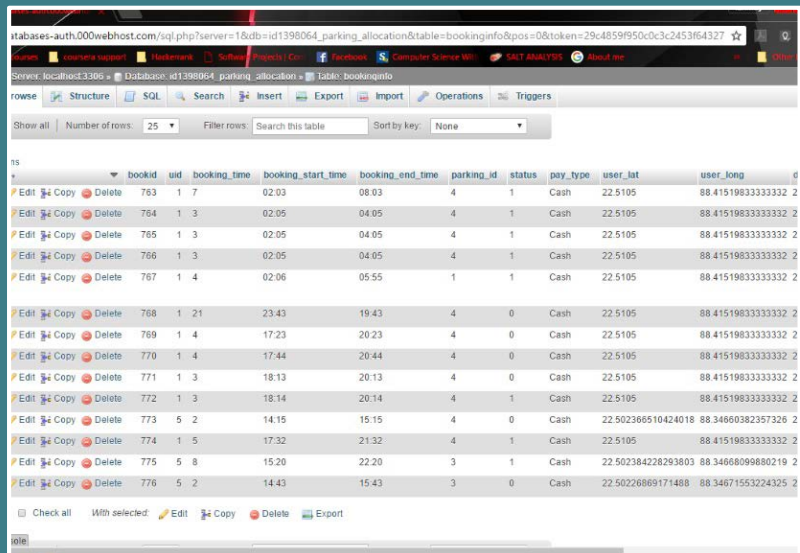
- We have implemented API level 22 and above meant for Android 5.1
- The android applications were created using Java and XML
- The network communication is done using Volley with the help of JSON.
- The APIs we have implemented are
 - Directions API for Android
 - Distance Matrix API for Android
 - Google Maps API for Android
 - Google Places API for Android



Technical Requirements

For Server

- We have implemented the logic using PHP
- The data has been stored using MySQL database
- The environment is an Apache Server
- We are using Apache/2.4.6 (CentOS)



The screenshot shows a web browser displaying a MySQL database interface. The URL is `localhost:3306/` and the database is `id1398064_parking_allocation`. The table `Table: bookinginfo` is selected. The table has 25 rows and 10 columns: `bookid`, `uid`, `booking_time`, `booking_start_time`, `booking_end_time`, `parking_id`, `status`, `pay_type`, `user_lat`, and `user_long`. The table contains 15 rows of data, each with a row number, a set of icons (Edit, Copy, Delete), and the column values. The data is as follows:

	bookid	uid	booking_time	booking_start_time	booking_end_time	parking_id	status	pay_type	user_lat	user_long
1	763	1	7	02:03	08:03	4	1	Cash	22.5105	88.41519833333332
2	764	1	3	02:05	04:05	4	1	Cash	22.5105	88.41519833333332
3	765	1	3	02:05	04:05	4	1	Cash	22.5105	88.41519833333332
4	766	1	3	02:05	04:05	4	1	Cash	22.5105	88.41519833333332
5	767	1	4	02:06	05:55	1	1	Cash	22.5105	88.41519833333332
6	768	1	21	23:43	19:43	4	0	Cash	22.5105	88.41519833333332
7	769	1	4	17:23	20:23	4	0	Cash	22.5105	88.41519833333332
8	770	1	4	17:44	20:44	4	0	Cash	22.5105	88.41519833333332
9	771	1	3	18:13	20:13	4	0	Cash	22.5105	88.41519833333332
10	772	1	3	18:14	20:14	4	1	Cash	22.5105	88.41519833333332
11	773	5	2	14:15	15:15	4	0	Cash	22.502366510424018	88.34660382357326
12	774	1	5	17:32	21:32	4	1	Cash	22.5105	88.41519833333332
13	775	5	8	15:20	22:20	3	1	Cash	22.502384228293803	88.3466099880219
14	776	5	2	14:43	15:43	3	0	Cash	22.50226869171488	88.34671553224325

Uniqueness

Our Algorithm

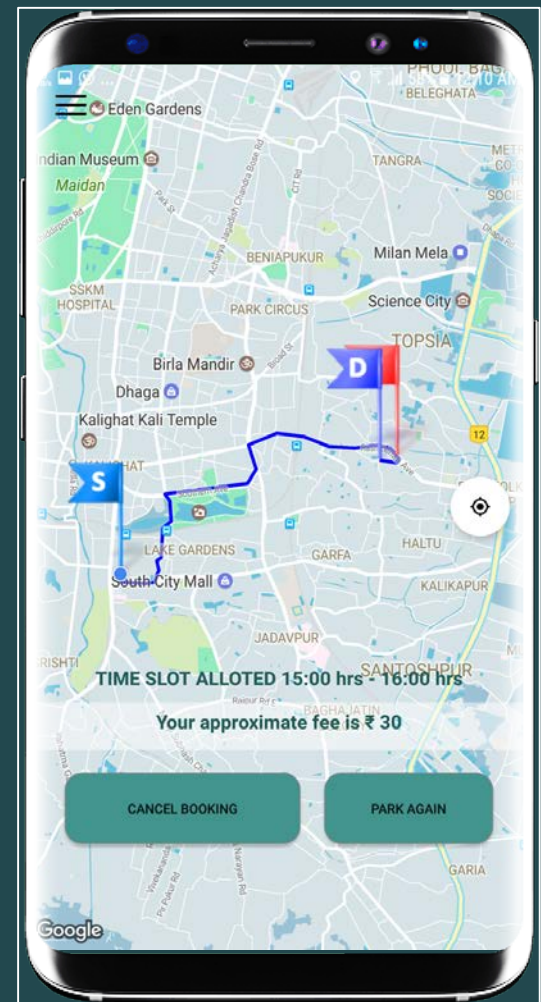
Proposed algorithm is based on Greedy Principle where the $((\text{Distance between LOC U and LOC Pi}) + (\text{Distance between LOC D and LOC Pi})) * C$ has to be minimal.

Scalability

Parking Lot
Owner

Location

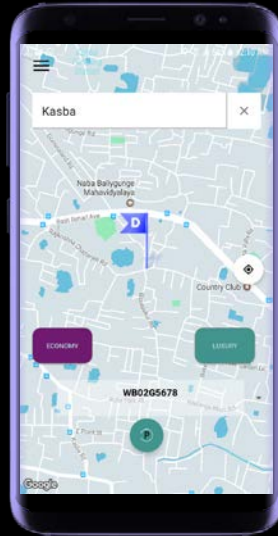
Number
of Slots



Solution Maturity



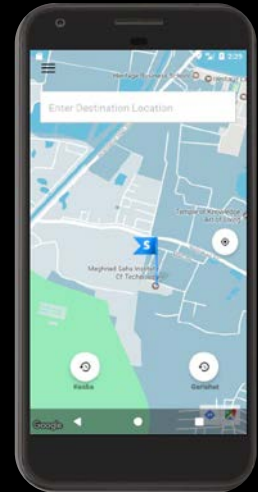
User Friendly
Application
With Refined
Interface



Select
Preferences To
Get A Tailored
Solution



Currently
Supports Up
to 400 Users
Per Day

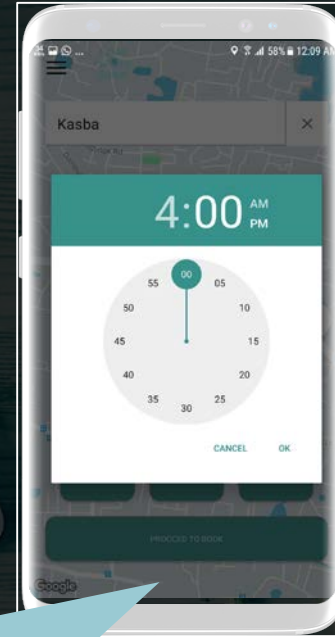


Easy Usable Interface



Challenges Faced

- ❑ Every day is divided into 24 slots each having full number of vacancies.
- ❑ By implementing this solution we have managed to bake in a system of scheduled booking



You can schedule your booking from beforehand

Limitation & Future Improvement



Bluetooth LE



Geo-fencing



Machine Learning



Internal Map



Pilot partner App
Modification

Special Regards to

Our Guide and Mentor

Asst. Professor Kamalesh Karmakar

Dept- Computer Science & Engineering

Meghnad Saha Institute of Technology

For being with us all throughout this journey



Thanks!

We are looking forward to your valuable feedback

Shayon Gupta
Vijay Pandey

Tanmoy Mukherjee
Poulomi Sen

