

CAR PARKING SOLUTIONS

Course - Object Oriented Programming CS-F213

Instructor in-charge - Dr. D V N Siva Kumar

By

Team No. 42



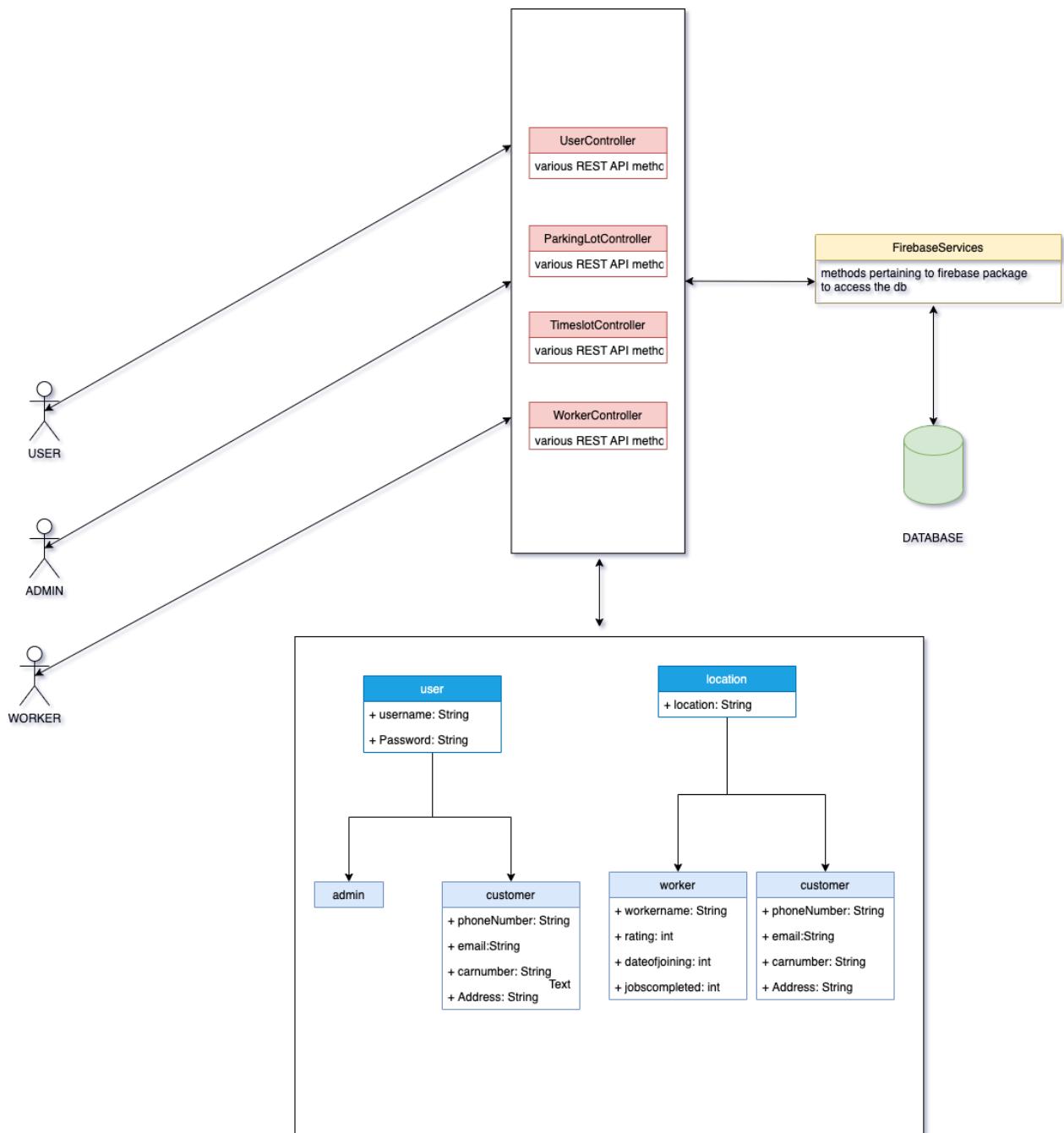
**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE,
PILANI (Hyderabad Campus)**

(December, 2021)

A. TEAM DETAILS

<u>Name</u>	<u>ID.No.</u>
1. ARCHIS SAHU	2020A7PS1692H
2. KARTIKAY DHALL	2020A7PS2087H
3. SUVIGYA SHARMA	2020A7PS0140H
4. RIYA SINGH	2020A7PS2048H

B. CLASS DIAGRAM



C. IMPLEMENTATION DETAILS

1. USER INTERFACE IMPLEMENTATION

Programming languages used: Dart
SDK/Framework: Flutter

We chose to build the front end on flutter not only because it promises absolute freedom in creating beautiful user interface regardless of the platform but also helps avoiding the hassle of making html pages and css files separately to beautify the page, and further separate JavaScript files to handle interactions and requests.

We code everything on Dart which is pretty similar to JAVA in ways of Object Oriented programming and flutter does the rest for us by building its respective index.html file and js files!

There are “widgets”, a class of objects in flutter, which basically represent the components visible on the screen when rendered. There are classes that we can define to control the various variables and to pass them to functions and call the REST API methods that we define in the back-end.

There is a wide range of packages available on Flutter which we can use to perform various functions like authorizing the user, sending

OTPs, converting json to a defined class of objects, performing http requests to the server seamlessly and much more.
Here's a list of some of the packages that we used

```
shared_preferences: ^2.0.9
json_annotation: ^4.3.0
json_serializable: ^6.0.1
dio: ^4.0.4
http: ^0.13.4
firebase_core: ^1.10.0
firebase_auth: ^3.2.0
email_auth: ^1.0.0
google_sign_in: ^5.2.1
firebase_admin: ^0.2.0-dev.7
cloud_firestore: ^3.1.0
flutter_datetime_picker: ^1.5.1
```

2. SERVER-SIDE BUSINESS LOGIC

Programming languages used: JAVA
SDK/Framework: Spring Framework

We have made various classes in the “models” package as shown in the above class diagram.

We have separate “controller” classes in the “controller” package where we have defined the REST API methods with the help of “@RestController” annotation which is a part of the spring framework.

“services” package has the files related to initializing, linking and interacting with the firebase database and we use preestablished firebase methods to make new functions to perform the required function.

The server receives requests from the client, it processes it and then does the needful function, for example fetching or posting data from the database, and then returning data back to the client.

3. DATABASE: Firebase

We chose Firebase as the database because other than storing data in a realtime database, it also provides functionalities for authorization of accounts like google sign-in and email sign-in and maintaining the authorization state.

4. TESTING: Postman

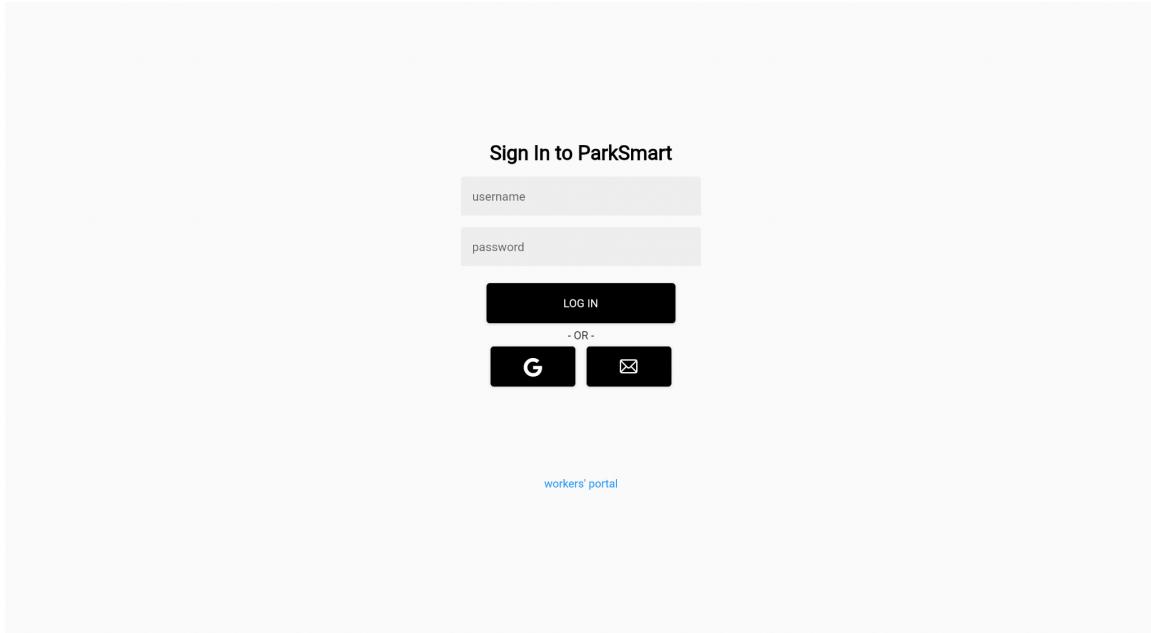
For testing the APIs we used postman to perform GET, POST, DELETE, PUT requests before implementing it in the front end.

D. SCREENSHOTS OF PROJECT MODULES

1. SIGN-IN & SIGN-UP OPTIONS

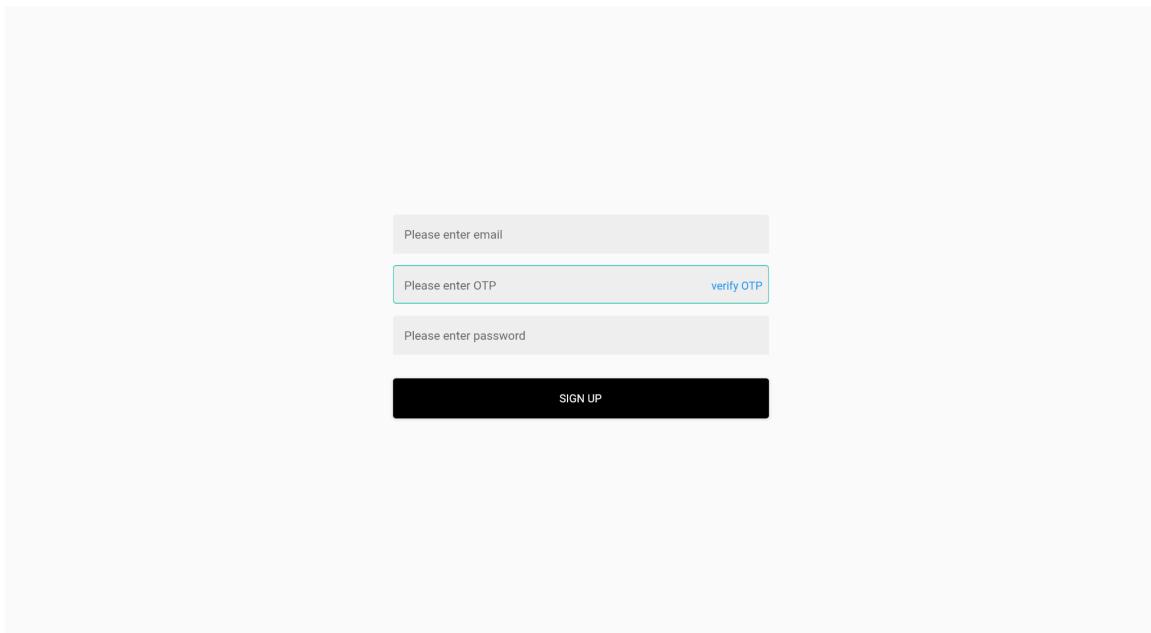
• SIGN-IN PAGE

- The user is provided with 3 modes of signing-in namely
 1. Username & password
 2. Google Account
 3. Email sign-in



- **SIGN-UP PAGE**

- If a new user wants to sign-up for the services website, he can enter his mail ID which will be verified by an OTP. Upon successful verification, the user will be able to set a password for this new account.



- The user will then be redirected to the registration page which will store the information about him for future bookings.

Please enter username

Please enter firstname

Please enter lastname

Phone number

address

car registration number

SAVE DETAILS

- **USER INFORMATION**

- The user will be able to check his information entered at the time of registration by clicking on the My Info button at the bottom left of the dashboard.

 log out

username:	archissahu
name:	ARCHIS SAHU
email:	archisxsaahu@gmail.com
Phone Number:	7901335361
Car Registration Number:	AP34 4657
Address	VM319, BPHC
Wallet: Rs	300



- SEARCH PARKING LOT

- After successful registration or in the case of pre-registered users, the search option will be available with filters of Location, Date, Checkin and Checkout time.

 log out

Please enter location

date 

check-in time 

check-out time 

select car type 

SEARCH



- A list of the available Parking Lots with the specified filters will be displayed, where the user can select the one that suits his budget and car size as well. The “STARRED” parking lots are the suggested sizes for the user’s car type

 log out

bitsh

2021 - 12 - 5 

21 : 54 : 52 

23 : 54 : 57 

sedan 

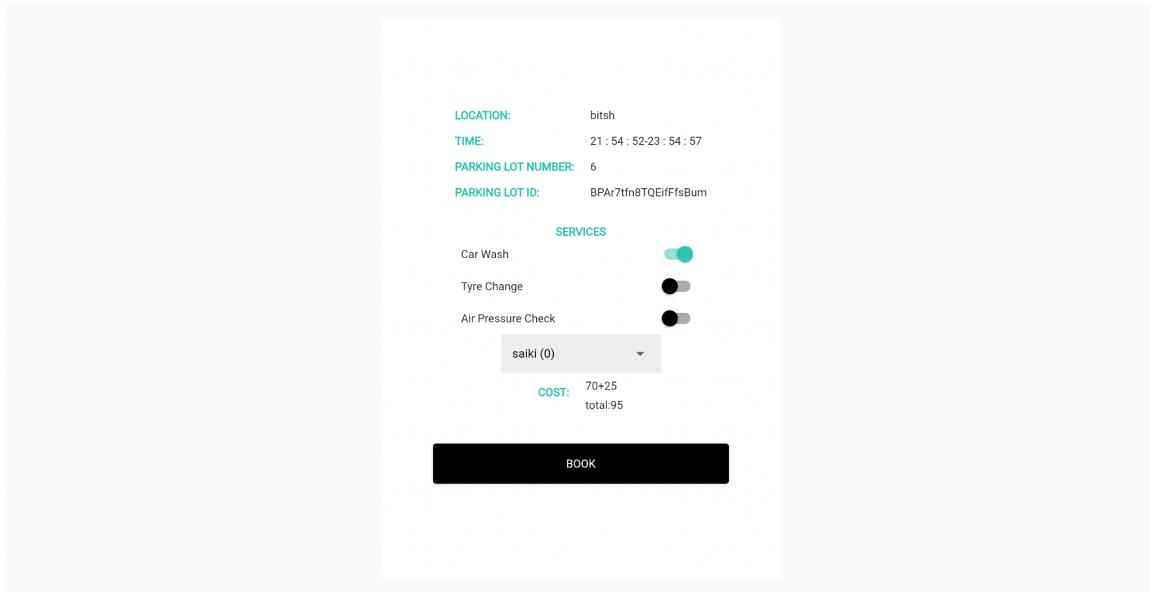
SEARCH

6 bitsh  cost: 70
ID: BPar7tn8TQEifFfsBum
size: sedan

cost: 80



- The user can now
 1. Cross check the details of his preferred parking lot
 2. Avail one or more of the services provided at that parking lot
 3. Select a worker of the user's choice (if available) by seeing his rating alongside his name.
 4. Check the total amount payable for availing this service.



- **BOOKING CONFIRMATION**

- On successful booking of the parking lot with the specified services and worker, a confirmation message will be displayed.



[go to home page](#)

- The user can see his booking history by clicking on the bottom right button available at the dashboard page.
- The bookings that have been completed will be shown in black, having an option of rating the services, whereas the future bookings will be shown in green, with an option of cancelling any of the upcoming bookings.

log out

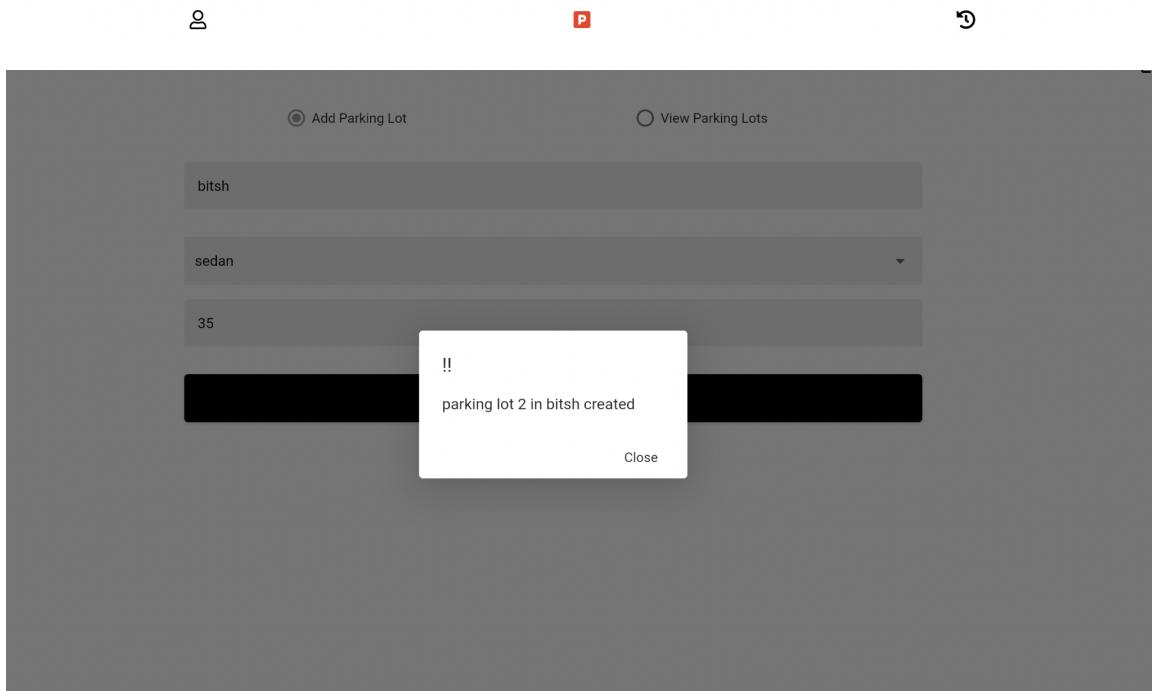


2. ADMIN SIDE INTERFACE

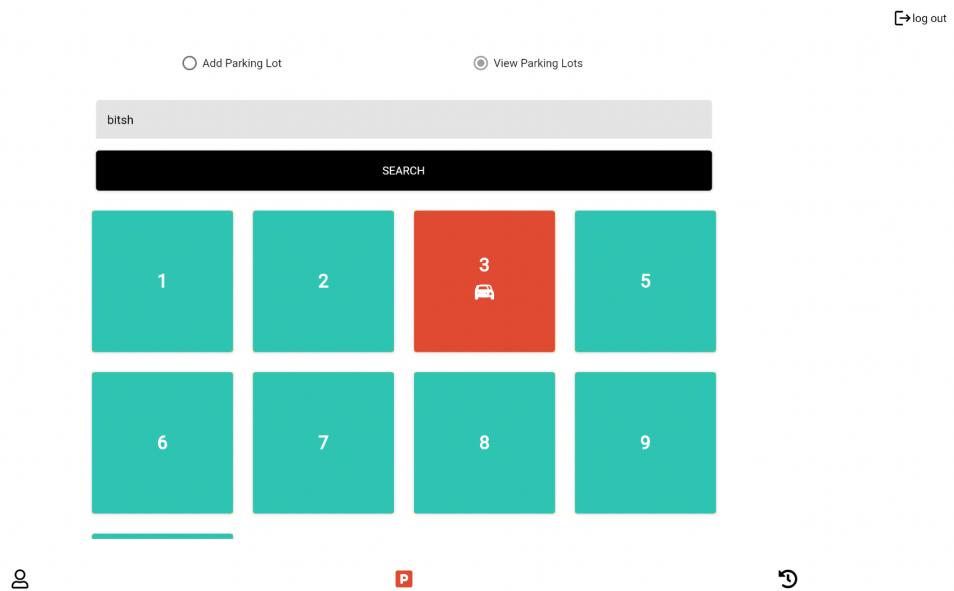
● MANAGE PARKING LOTS

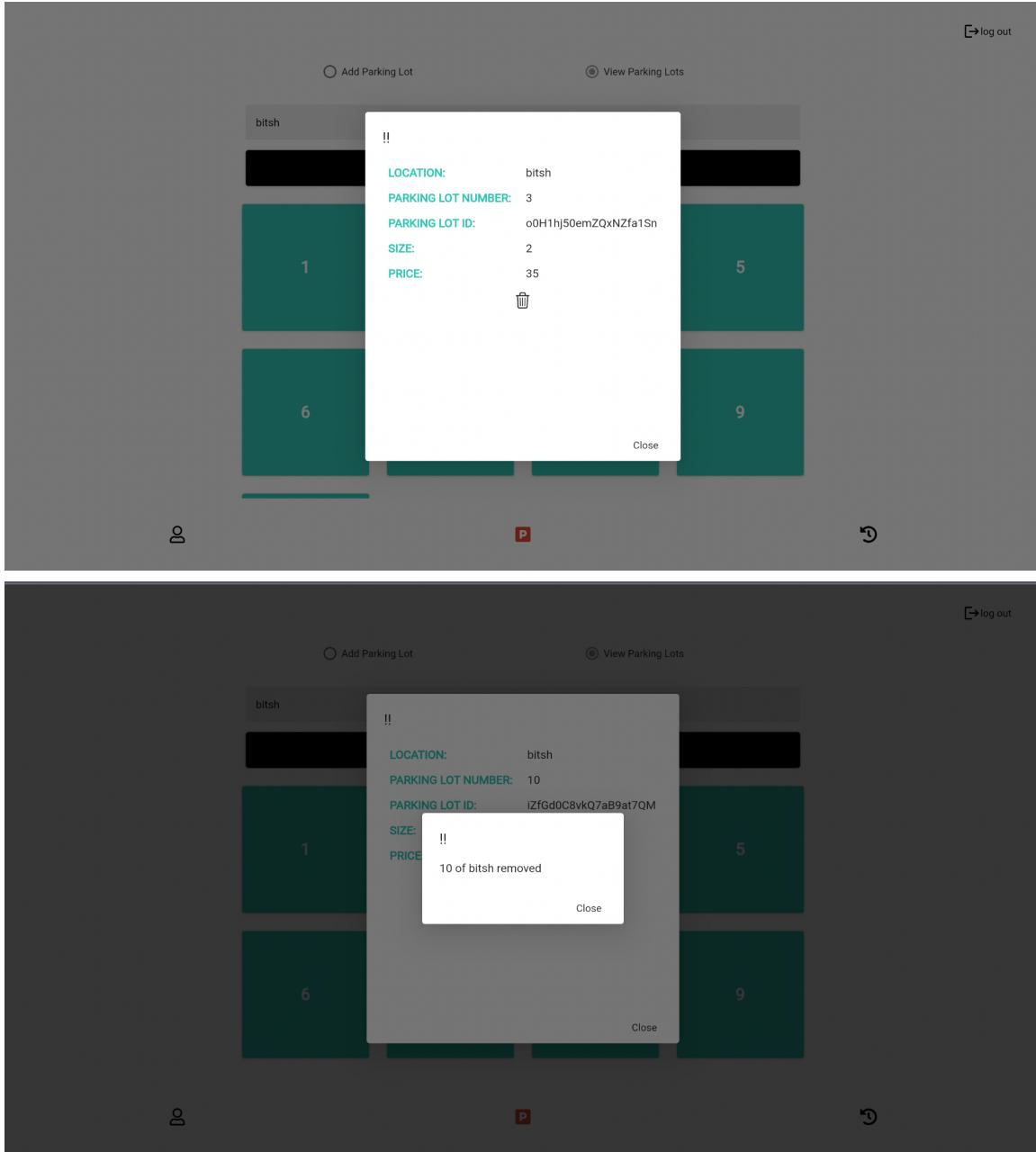
- The Admin will have options of adding a new parking lot or delete an existing one by specifying the details as shown below.

The screenshot shows a user interface for managing parking lots. At the top right is a 'log out' button. Below it are two links: 'Add Parking Lot' (radio button selected) and 'View Parking Lots'. The main area contains three input fields: 'location' (containing 'bitsh'), 'select car type' (containing 'sedan'), and 'price' (containing '35'). A large black 'ADD PARKING LOT' button is at the bottom.



- The Admin can also view the status of all the Parking lots in real-time, where the occupied time-slots of the particular parking lot will be displayed in red and the free ones will be displayed in green.





● MANAGE WORKERS

- The Admin can add a new worker as well as view the list of existing workers from his dashboard. The list shows Date of Joining and No. of jobs completed by the workers. There is an option for the Admin to remove workers if needed.

 log out

Add Worker

View Workers

location

workename

username

password

ADD WORKER



Add Worker

View Workers

bitsh

ramu

ramu09

12345678

!!

worker ramu assigned at bitsh
allotted ID No.: ramu09

Close

 log out

 Add Worker

 View Workers

bitsh				
SEARCH				
worker name	rating	# jobs completed	date of joining	
shamu	0	0	2021-12-05 18:49:04.142	
ramu	-1	3	2021-12-05 22:15:09.457	
salki	2	5	2021-12-05 20:13:48.140	



We have learnt a lot from this insightful project. We sincerely thank you and look forward to utilizing the concepts we learnt in this course in our future endeavours.

THANK YOU