Deliverable 3

The Tech Sense

Smart Indoor Parking System

Group 10

Raghav Sharma	**N01537255**
Kunal Dhiman	**N01540952**
Nisargkumar Pareshbhai Joshi	**N01545986**
Rushi Manojkumar Patel	**N01539144**

Table of Contents

1. Project Description

The app aims to provide a seamless parking booking experience for users, with additional functionalities to enhance user engagement and convenience. It begins with a splash screen, followed by login and sign-up pages offering multiple authentication options. The app features four main menu items: Promotions, Support, Favourites, and Notifications, each fully operational to cater to different user needs. The core functionality includes a "Park" section for booking parking spots, complemented by an "Activity" screen that displays all booking details. Additionally, the app offers a "Manage Account" section with essential options like Settings, Account Details, Help, and Logout etc, all functioning smoothly to provide users with an intuitive and comprehensive experience.

2. Effort and Participation

Team Members' Information and Participation:

Name	**Student ID**	**GitHub ID**	**Signature**	**Effort (%)**
Raghav Sharma	N01537255	RaghavSharma7255	RS	100%
Kunal Dhiman	N01540952	KunalDhiman0952	KD	100%
Nisargkumar Pareshbhai Joshi	N01545986	NisargJoshi5986	NJ	80%
Rushi Manojkumar Patel	N01539144	RushiPatel9144	RP	70%

3. GitHub Repo Link

GitHub Repository Link:
https://github.com/RushiPatel9144/SmartIndoorParkingSystem

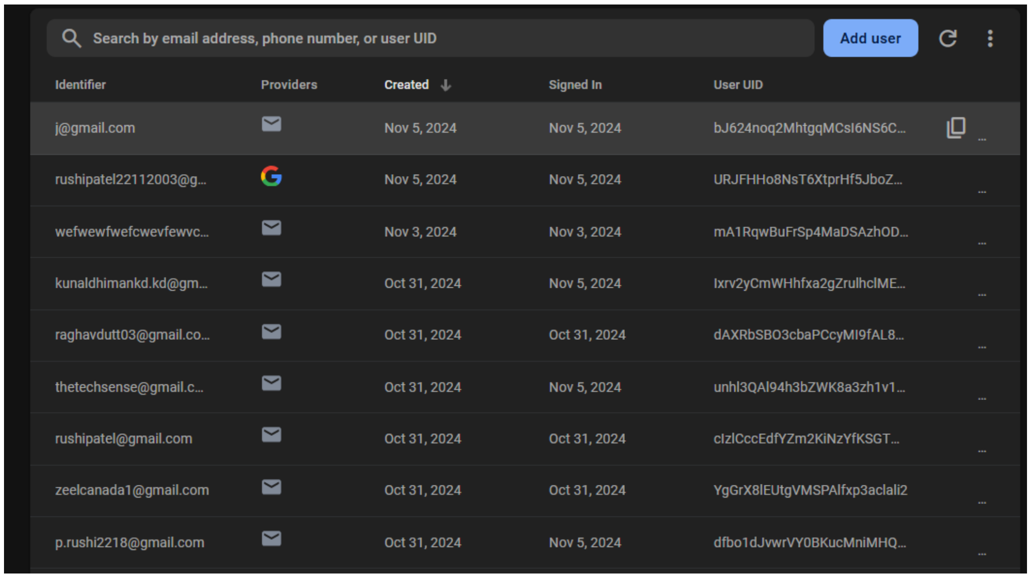
4. Login Functionality

Login Credentials for Testing:

- **Email:** thetechsense@gmail.com
- **Password:** admin123
- **Alternative:** p.rushi2218@gmail.com
- **Alternative Password:** 12345678

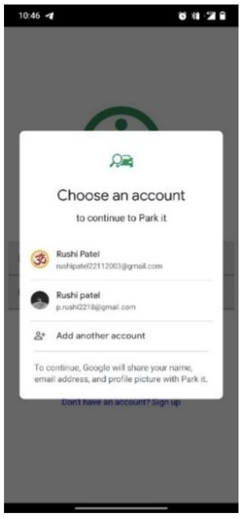
Screenshots:

- Screenshot showing the account in the DB



Identifier	Providers	Created	Signed In	User UID
j@gmail.com	✉	Nov 5, 2024	Nov 5, 2024	bJ624noq2MhtgqMCsI6NS6C...
rushipatel22112003@g...	G	Nov 5, 2024	Nov 5, 2024	URJFHHo8NsT6XtprHf5JboZ...
wefwewfwefcwevfewvc...	✉	Nov 3, 2024	Nov 3, 2024	mA1RqwBuFrSp4MaDSAzhOD...
kunaldhimankd.kd@gm...	✉	Oct 31, 2024	Nov 5, 2024	Ixrv2yCmWHhfxa2gZrulhclME...
raghavdutt03@gmail.co...	✉	Oct 31, 2024	Oct 31, 2024	dAXRbSBO3cbaPCcyMI9fAL8...
thetechsense@gmail.c...	✉	Oct 31, 2024	Nov 5, 2024	unhl3QAl94h3bZWK8a3zh1v1...
rushipatel@gmail.com	✉	Oct 31, 2024	Oct 31, 2024	clzlCccEdfYZm2KiNzYfKSGT...
zeelcanada1@gmail.com	✉	Oct 31, 2024	Oct 31, 2024	YgGrX8lEUtgVMSPAlfxp3aclali2
p.rushi2218@gmail.com	✉	Oct 31, 2024	Nov 5, 2024	dfbo1dJvwrVY0BKucMniMHQ...

- Screenshot showing successful Gmail authentication



Choose an account

to continue to Park it

Rushi Patel
rushipatel22112003@gmail.com

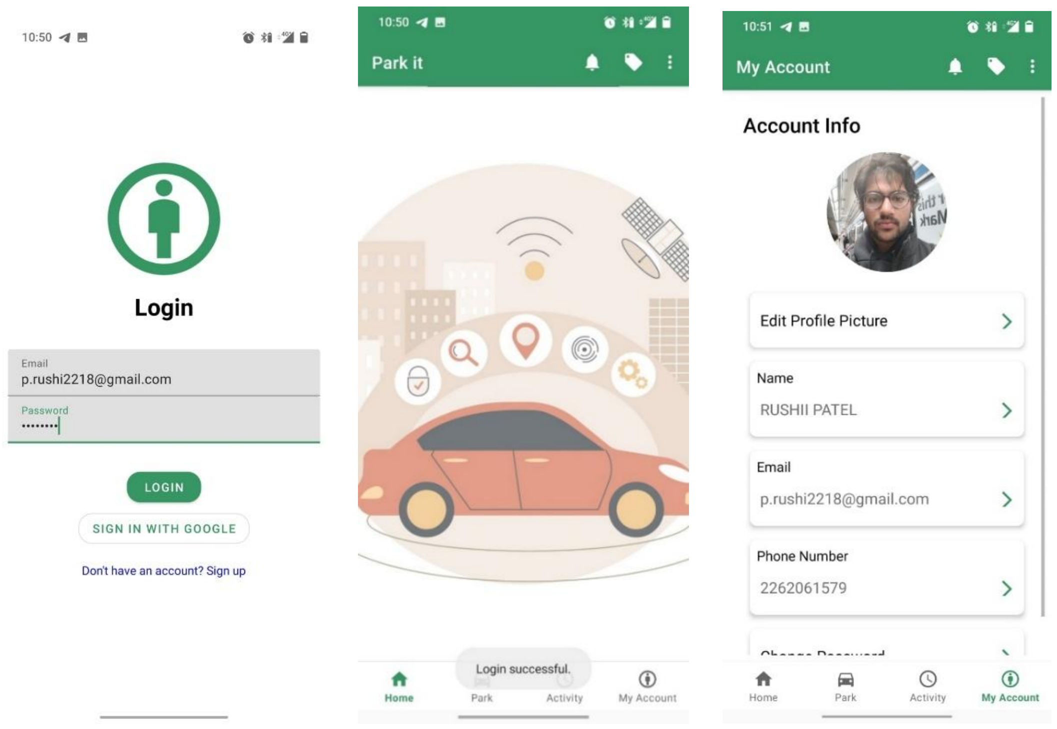
Rushi patel
p.rushi2218@gmail.com

Add another account

To continue, Google will share your name, email address, and profile picture with Park it.

Don't have an account? Sign up

- Screenshot showing login through app with these credentials



Email
p.rushi2218@gmail.com
Password
••••••••

Login successful.

Name
RUSHII PATEL
Email
p.rushi2218@gmail.com
Phone Number
2262061579

5. Sprint 3 Summary

Sprint 3 Summary (for each member):

Raghav (RS)

- **Implement Notifications System**: Completed, covering notifications setup and functionality for user alerts.
- **Implement Tab Layout in Activity Section**: Completed, possibly includes organizing activity views in a tab format for better navigation.
- **Store and Retrieve Sensor Data from Database**: Completed, handling database interactions for sensor data storage and retrieval.
- **Develop Forgot Password Feature**: Completed, implementing a feature for password recovery.
- **Implement Manage Account Functionality**: Completed, includes managing user account details.

- **Assisted in Park Section and Booking System**: Contributed to the implementation of these sections, supporting development tasks.
- **Supported Login Part and Database Work:** Helped with the login feature and assisted others in database-related tasks.
- **Assisted in UI**: Contributed to improving the app's user interface design.

Kunal (KD)

- **Develop Settings Functionality**: Completed, with two comments indicating potential discussions or clarifications.
- **Develop Activity Section Functionalities**: Completed, covering the main functions and layout for the Activity Section.
- **Embed Google Maps in Park Section**: Completed, involving the integration of Google Maps in the Park Section for location and navigation purposes.
- **Add Booking System**: Completed, which likely includes the implementation of the booking feature for users to reserve parking spaces or other resources.
- **Add Permanent Marker for Parking Spots in Map with Details**: In progress, focusing on adding a persistent marker feature in the map to display parking spot details.

Nisarg (NJ)

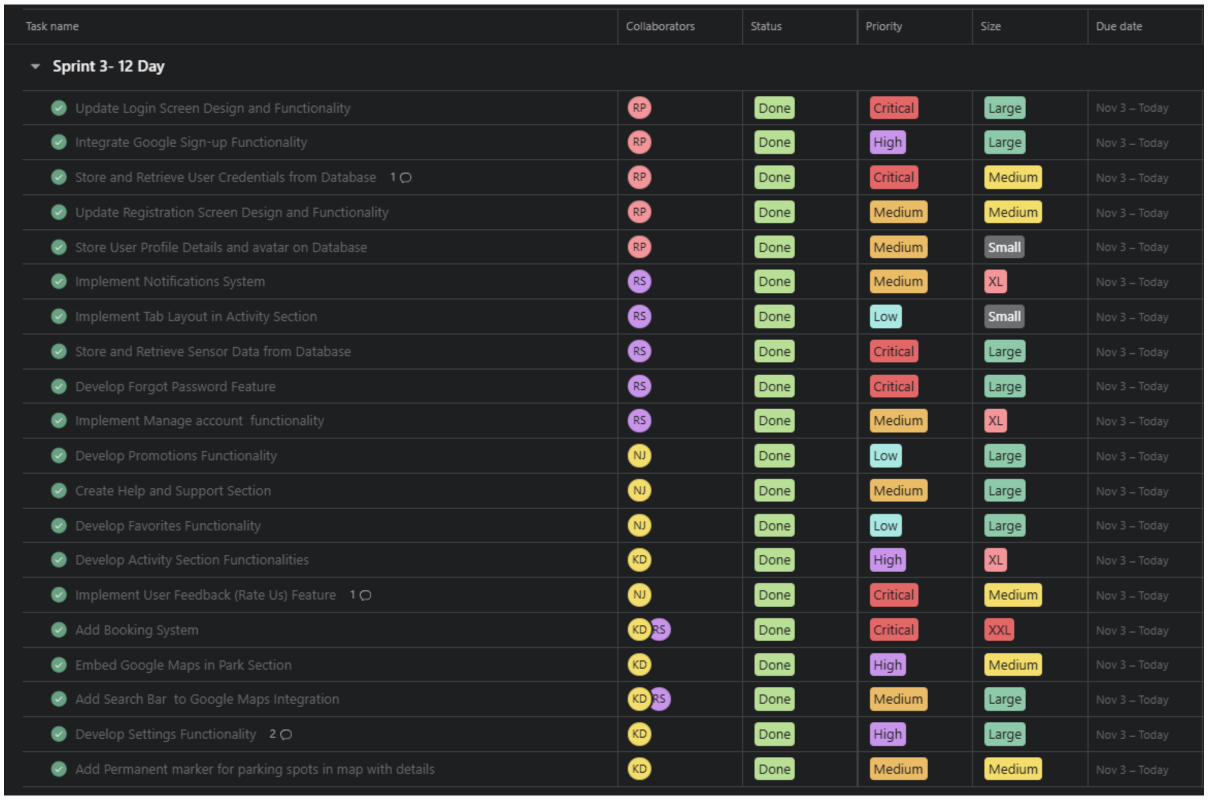
- **Develop Promotions Functionality**: Completed, potentially involving promotions display and logic for the app.
- **Create Help and Support Section**: Completed, providing users with assistance and support resources.
- **Develop Favorites Functionality**: Completed, enabling users to mark items or locations as favorites.
- **Implement User Feedback (Rate Us) Feature**: Completed, allowing users to rate the application.
- **Removed Hardcoded Values and Improved UI**: Worked on eliminating hardcoded values and enhancing the UI of the app.

Rushi (RP)

- **Update Login Screen Design and Functionality**: Completed, focusing on login interface and related features.
- **Integrate Google Sign-up Functionality**: Completed, adding Google sign-up integration.
- **Store and Retrieve User Credentials from Database**: Completed, handling credentials storage and access.
- **Update Registration Screen Design and Functionality**: Completed, updating the registration interface and process.
- **Store User Profile Details and Avatar on Database:** Completed, storing user profile data, including avatar, in the database.

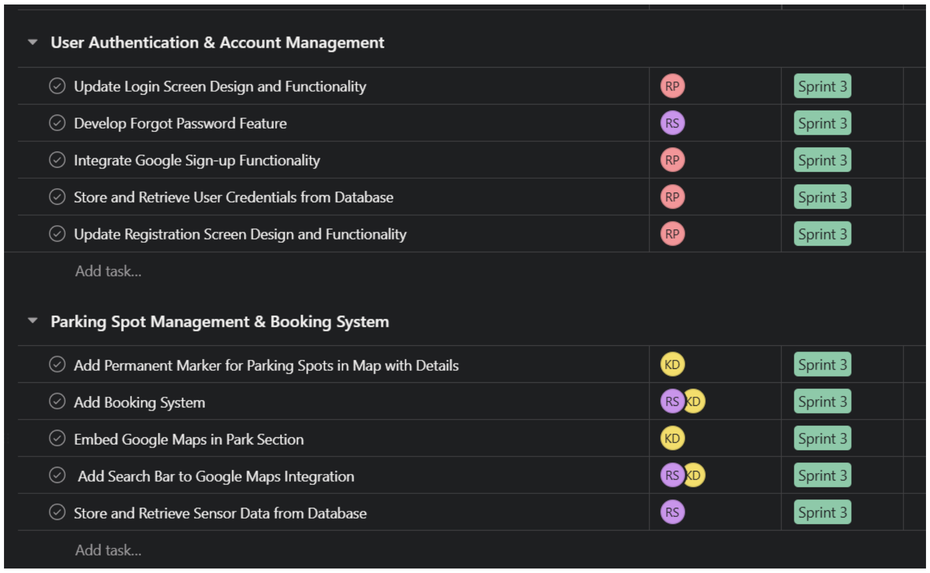
6. Sprint 3 Dashboard and Gantt Chart

6.1. Sprint Dashboard:

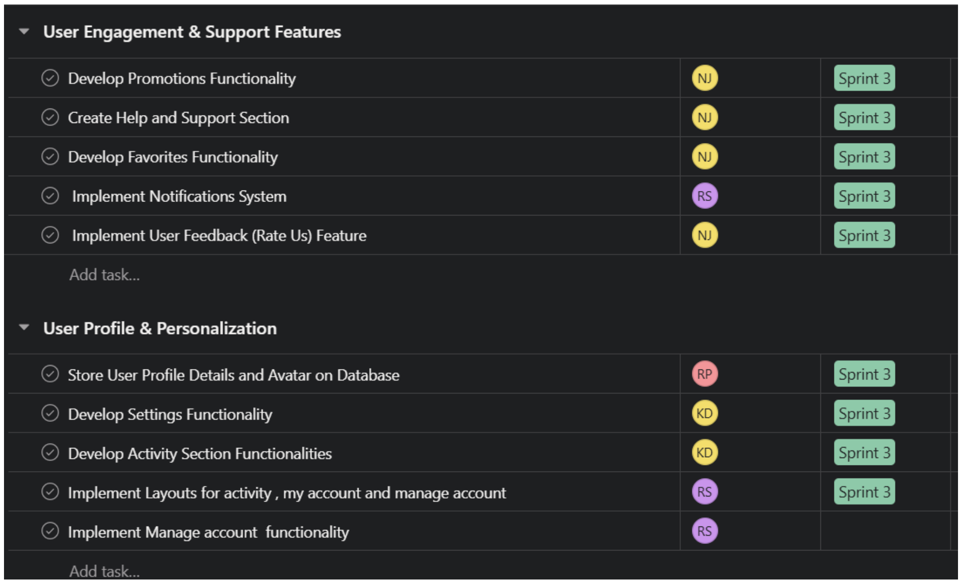


Task name	Collaborators	Status	Priority	Size	Due date
Sprint 3- 12 Day					
Update Login Screen Design and Functionality	RP	Done	Critical	Large	Nov 3 – Today
Integrate Google Sign-up Functionality	RP	Done	High	Large	Nov 3 – Today
Store and Retrieve User Credentials from Database 1	RP	Done	Critical	Medium	Nov 3 – Today
Update Registration Screen Design and Functionality	RP	Done	Medium	Medium	Nov 3 – Today
Store User Profile Details and avatar on Database	RP	Done	Medium	Small	Nov 3 – Today
Implement Notifications System	RS	Done	Medium	XL	Nov 3 – Today
Implement Tab Layout in Activity Section	RS	Done	Low	Small	Nov 3 – Today
Store and Retrieve Sensor Data from Database	RS	Done	Critical	Large	Nov 3 – Today
Develop Forgot Password Feature	RS	Done	Critical	Large	Nov 3 – Today
Implement Manage account functionality	RS	Done	Medium	XL	Nov 3 – Today
Develop Promotions Functionality	NJ	Done	Low	Large	Nov 3 – Today
Create Help and Support Section	NJ	Done	Medium	Large	Nov 3 – Today
Develop Favorites Functionality	NJ	Done	Low	Large	Nov 3 – Today
Develop Activity Section Functionalities	KD	Done	High	XL	Nov 3 – Today
Implement User Feedback (Rate Us) Feature 1	NJ	Done	Critical	Medium	Nov 3 – Today
Add Booking System	KD RS	Done	Critical	XXL	Nov 3 – Today
Embed Google Maps in Park Section	KD	Done	High	Medium	Nov 3 – Today
Add Search Bar to Google Maps Integration	KD RS	Done	Medium	Large	Nov 3 – Today
Develop Settings Functionality 2	KD	Done	High	Large	Nov 3 – Today
Add Permanent marker for parking spots in map with details	KD	Done	Medium	Medium	Nov 3 – Today

6.2. Stories and Tasks



User Authentication & Account Management		
Update Login Screen Design and Functionality	RP	Sprint 3
Develop Forgot Password Feature	RS	Sprint 3
Integrate Google Sign-up Functionality	RP	Sprint 3
Store and Retrieve User Credentials from Database	RP	Sprint 3
Update Registration Screen Design and Functionality	RP	Sprint 3
Add task...		
Parking Spot Management & Booking System		
Add Permanent Marker for Parking Spots in Map with Details	KD	Sprint 3
Add Booking System	RS KD	Sprint 3
Embed Google Maps in Park Section	KD	Sprint 3
Add Search Bar to Google Maps Integration	RS KD	Sprint 3
Store and Retrieve Sensor Data from Database	RS	Sprint 3
Add task...		



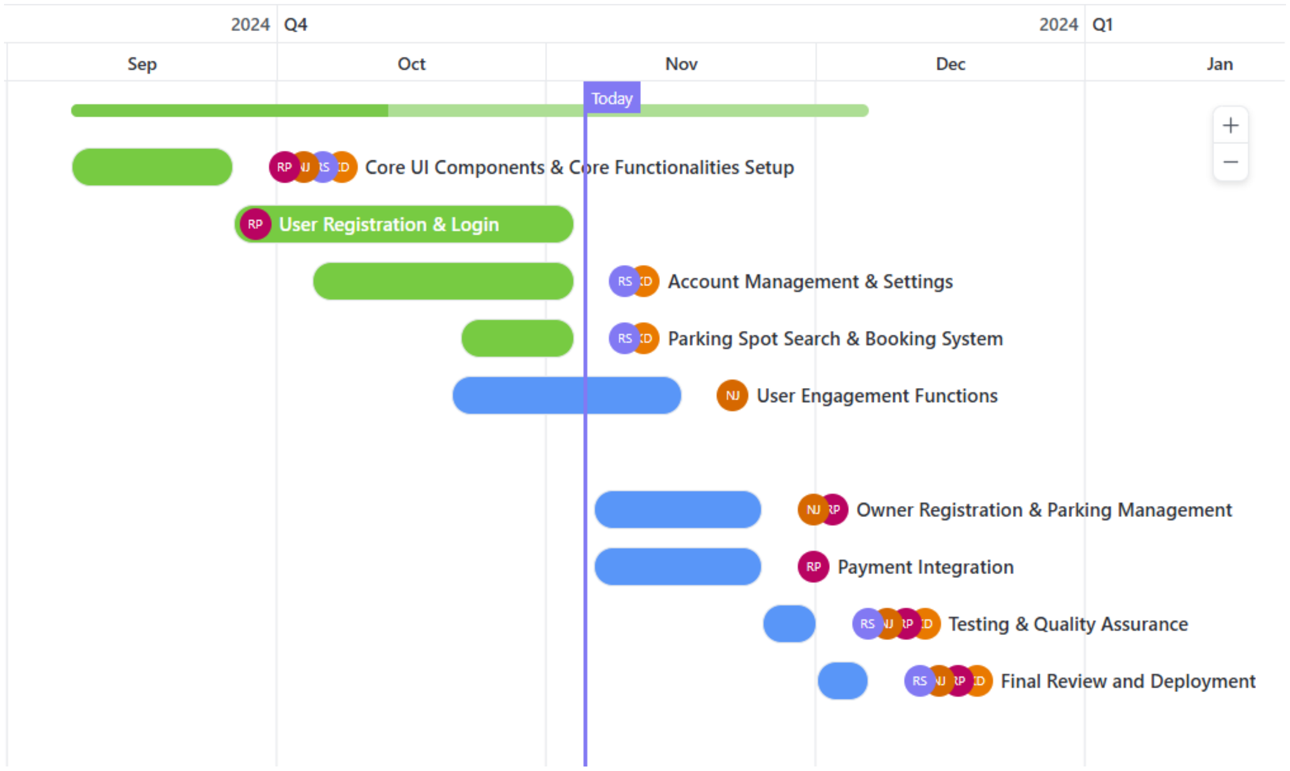
User Engagement & Support Features

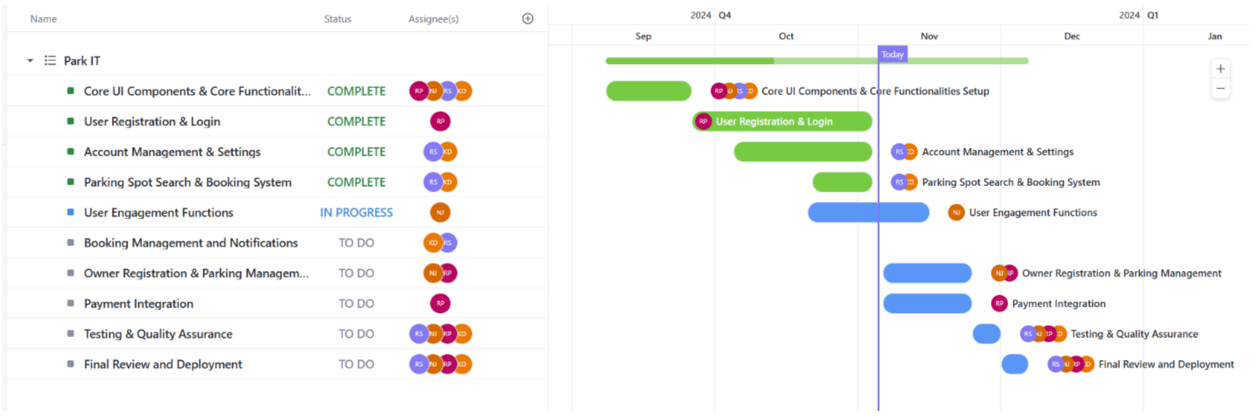
Develop Promotions Functionality	NJ	Sprint 3
Create Help and Support Section	NJ	Sprint 3
Develop Favorites Functionality	NJ	Sprint 3
Implement Notifications System	RS	Sprint 3
Implement User Feedback (Rate Us) Feature	NJ	Sprint 3
Add task...		

User Profile & Personalization

Store User Profile Details and Avatar on Database	RP	Sprint 3
Develop Settings Functionality	KD	Sprint 3
Develop Activity Section Functionalities	KD	Sprint 3
Implement Layouts for activity , my account and manage account	RS	Sprint 3
Implement Manage account functionality	RS	
Add task...		

6.3. Gantt Chart:





Name	Status	Assignee(s)
Park IT		
Core UI Components & Core Functionalit...	COMPLETE	RP NJ RS KD
User Registration & Login	COMPLETE	RP
Account Management & Settings	COMPLETE	RS KD
Parking Spot Search & Booking System	COMPLETE	RS KD
User Engagement Functions	IN PROGRESS	NJ
Booking Management and Notifications	TO DO	KD RS
Owner Registration & Parking Managem...	TO DO	NJ RP
Payment Integration	TO DO	RP
Testing & Quality Assurance	TO DO	RS NJ RP KD
Final Review and Deployment	TO DO	RS NJ RP KD

2024 Q4
2024 Q1
Sep
Oct
Nov
Dec
Jan
Today
Core UI Components & Core Functionalities Setup
User Registration & Login
Account Management & Settings
Parking Spot Search & Booking System
User Engagement Functions
Owner Registration & Parking Management
Payment Integration
Testing & Quality Assurance
Final Review and Deployment

7. Daily Standups and Meeting Records

Stand-up Meeting - Sprint 3 - October 28th 2024

Team Member	What did they do yesterday?	What will they do today?	Blockers
Raghav Sharma	Removed the Navigation Drawer, Updated Account Layout	Implement Notification System	None
Kunal Dhiman	Embedded Google Maps	Add SearchBar for parking spots in map	None
Nisarg Joshi	Created UI for RateUs Fragment	Setup the firestore Database for Feedback	None
Rushi Patel	User data storage in Firebase, user data retrieval	Sign in with Google, store profile photo to Firebase	None

Stand-up Meeting - Sprint 3- October 30th, 2024

Team Member	What did you do yesterday?	What will you do today?	Blockers
Raghav Sharma	Implemented Notification Sytem, Worked on ManageAccounts	Assist in Park Section and Booking System	None
Kunal Dhiman	Added Permanent Marker for Map, Added parking spots in the map	Implement Booking System	None

Nisarg Joshi	Setup the database, Implemented Promotions Feature	Implement Help & Support	None
Rushi Patel	Used user model instead of HashMap to store data	Store name and phone number in Firestore	None

---

Stand-up Meeting - Sprint 3 - November 3rd, 2024

Team Member	What did you do yesterday?	What will you do today?	Blockers
Raghav Sharma	Assist in Park Section and Booking System	Assist in Login Section and Gantt Chart	None
Kunal Dhiman	Implement Booking Section	Implement Activity and Design Principles	None
Nisarg Joshi	Implement Help & Support	Implement Favorites and Assist in Activity	None
Rushi Patel	Worked on storing profile photo in Firebase	ScreenFlow and Sprint Dashboard	None

8. Design Principles and Code Progress

Design Principles:

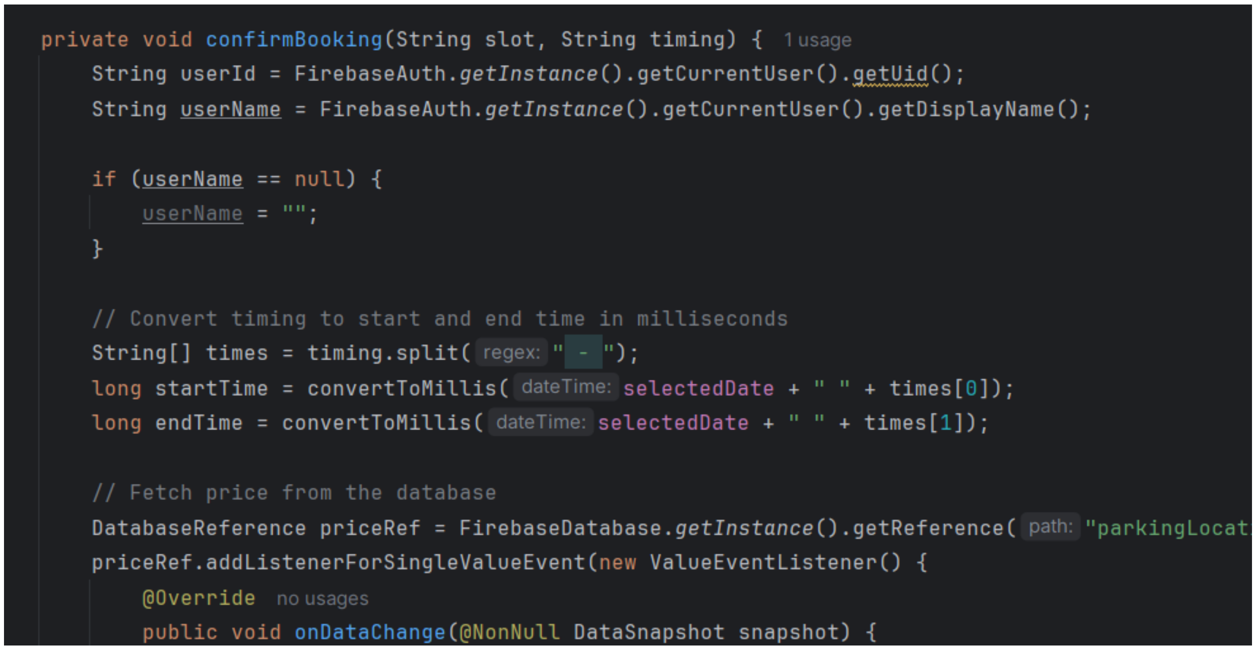
❖ **Principle 1: Separation of Concerns**

✓ Explanation: Separation of concerns is a design principle that states that a software system should be divided into distinct sections, each addressing a separate concern. This makes the code more modular, easier to maintain, and enhances readability.

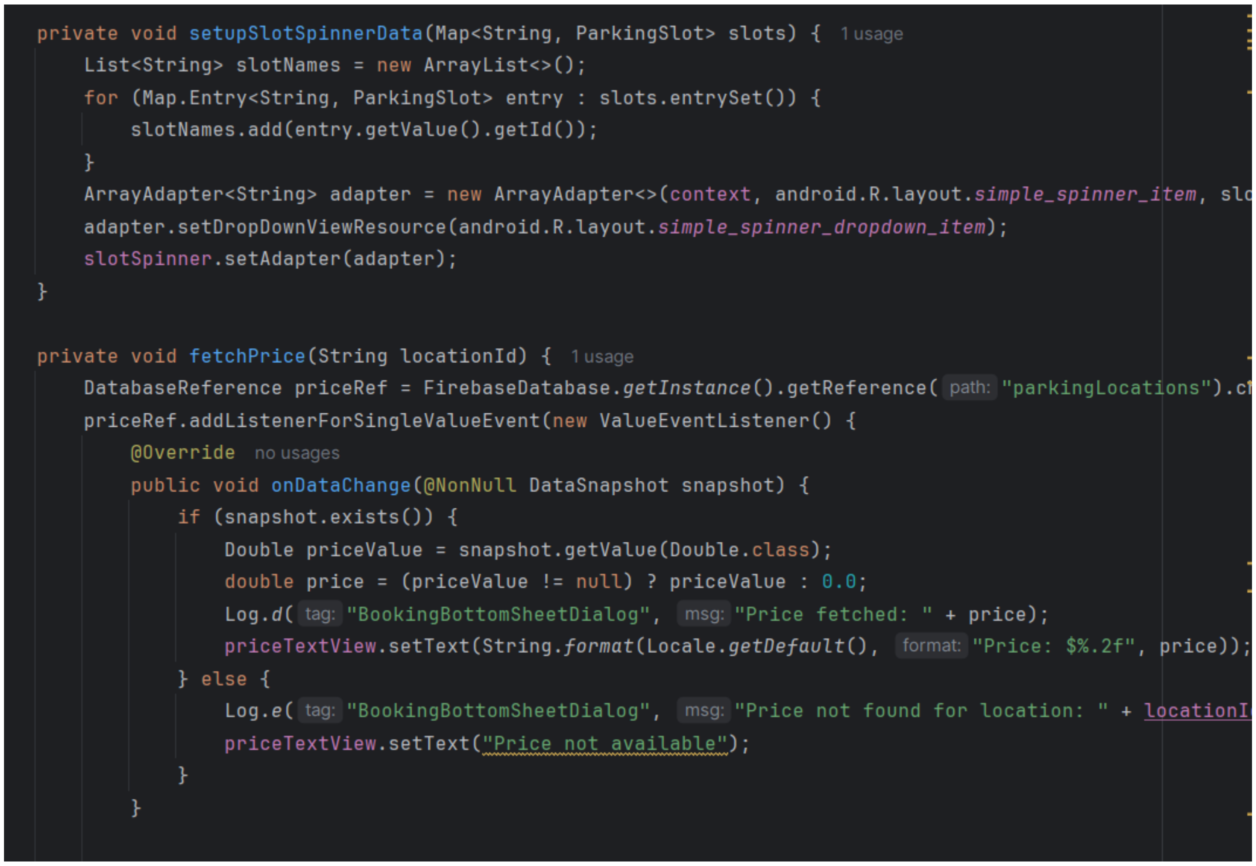
✓ Example: In ourBookingBottomSheetDialog class, we have separated the concerns of fetching data, setting up UI elements, and handling user interactions into different methods. For instance:

- fetchParkingLocationData(): Handles fetching parking location data from the database.
- setupSlotSpinnerData(): Sets up the slot spinner with data.
- fetchPrice(): Fetches the price for the selected slot.

- confirmBooking(): Handles the booking confirmation process.



```
private void confirmBooking(String slot, String timing) {  1 usage
    String userId = FirebaseAuth.getInstance().getCurrentUser().getUid();
    String userName = FirebaseAuth.getInstance().getCurrentUser().getDisplayName();

    if (userName == null) {
        userName = "";
    }

    // Convert timing to start and end time in milliseconds
    String[] times = timing.split( regex: " - ");
    long startTime = convertToMillis( dateTime: selectedDate + " " + times[0]);
    long endTime = convertToMillis( dateTime: selectedDate + " " + times[1]);

    // Fetch price from the database
    DatabaseReference priceRef = FirebaseDatabase.getInstance().getReference( path: "parkingLocat
    priceRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override  no usages
        public void onDataChange(@NonNull DataSnapshot snapshot) {
```



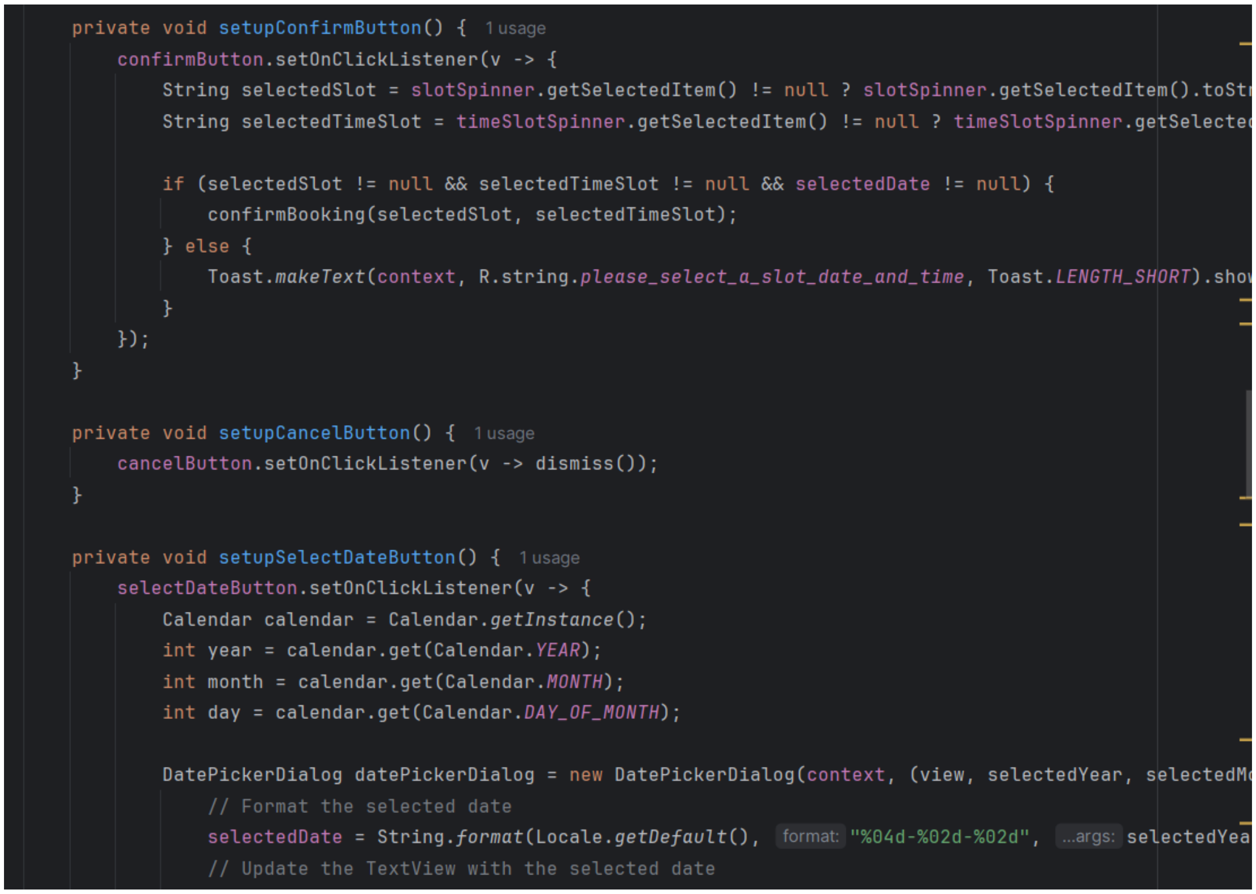
```
private void setupSlotSpinnerData(Map<String, ParkingSlot> slots) {  1 usage
    List<String> slotNames = new ArrayList<>();
    for (Map.Entry<String, ParkingSlot> entry : slots.entrySet()) {
        slotNames.add(entry.getValue().getId());
    }
    ArrayAdapter<String> adapter = new ArrayAdapter<>(context, android.R.layout.simple_spinner_item, slo
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    slotSpinner.setAdapter(adapter);
}

private void fetchPrice(String locationId) {  1 usage
    DatabaseReference priceRef = FirebaseDatabase.getInstance().getReference( path: "parkingLocations").c
    priceRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override  no usages
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if (snapshot.exists()) {
                Double priceValue = snapshot.getValue(Double.class);
                double price = (priceValue != null) ? priceValue : 0.0;
                Log.d( tag: "BookingBottomSheetDialog",  msg: "Price fetched: " + price);
                priceTextView.setText(String.format(Locale.getDefault(),  format: "Price: $%.2f", price));
            } else {
                Log.e( tag: "BookingBottomSheetDialog",  msg: "Price not found for location: " + locationI
                priceTextView.setText("Price not available");
            }
        }
```

❖ **Principle 2**: **Single Responsibility Principle (SRP)**

✓ Explanation: The Single Responsibility Principle (SRP) states that a class should have only one reason to change, meaning it should have only one job or responsibility. This makes the class more focused and easier to manage.

✓ Example: In ourBookingBottomSheetDialog class, each method has a single responsibility. For example:

- setupConfirmButton(): Sets up the confirm button and its click listener.
- setupCancelButton(): Sets up the cancel button and its click listener.
- setupSelectDateButton(): Sets up the select date button and its click listener.



```
private void setupConfirmButton() {  1 usage
    confirmButton.setOnClickListener(v -> {
        String selectedSlot = slotSpinner.getSelectedItem() != null ? slotSpinner.getSelectedItem().toStr
        String selectedTimeSlot = timeSlotSpinner.getSelectedItem() != null ? timeSlotSpinner.getSelecte

        if (selectedSlot != null && selectedTimeSlot != null && selectedDate != null) {
            confirmBooking(selectedSlot, selectedTimeSlot);
        } else {
            Toast.makeText(context, R.string.please_select_a_slot_date_and_time, Toast.LENGTH_SHORT).show
        }
    });
}

private void setupCancelButton() {  1 usage
    cancelButton.setOnClickListener(v -> dismiss());
}

private void setupSelectDateButton() {  1 usage
    selectDateButton.setOnClickListener(v -> {
        Calendar calendar = Calendar.getInstance();
        int year = calendar.get(Calendar.YEAR);
        int month = calendar.get(Calendar.MONTH);
        int day = calendar.get(Calendar.DAY_OF_MONTH);

        DatePickerDialog datePickerDialog = new DatePickerDialog(context, (view, selectedYear, selectedMo
            // Format the selected date
            selectedDate = String.format(Locale.getDefault(),  format: "%04d-%02d-%02d",  ...args: selectedYea
            // Update the TextView with the selected date
```

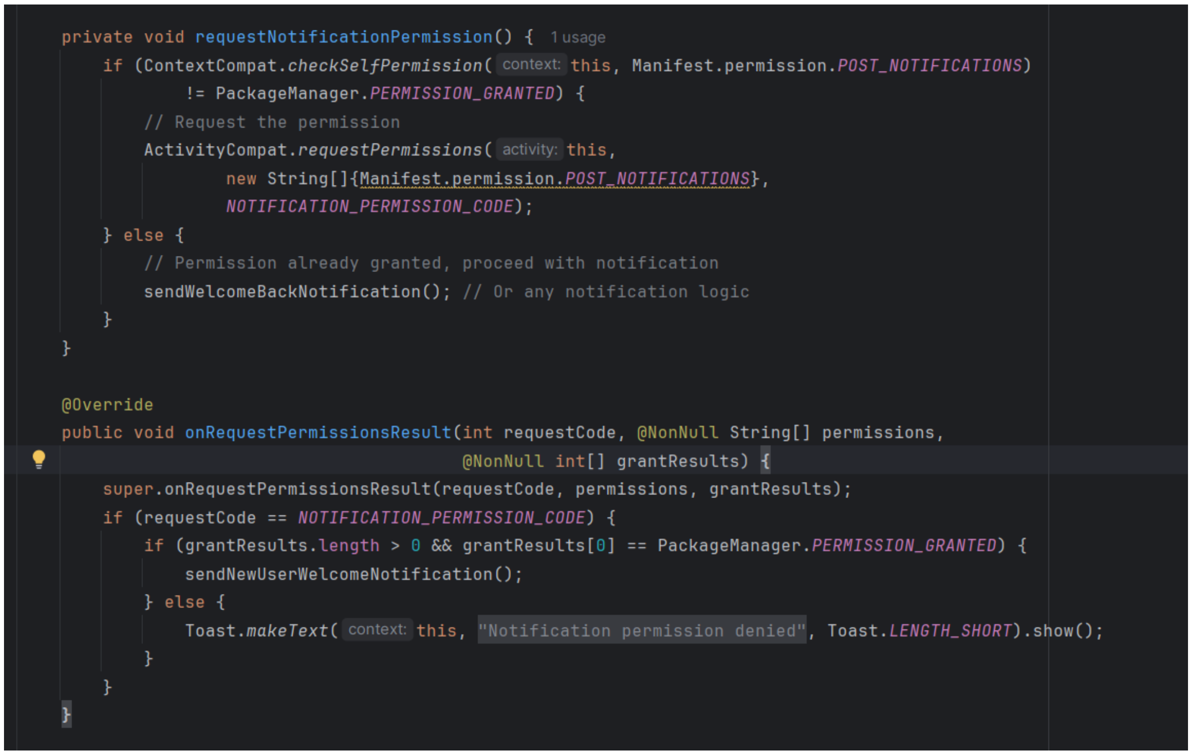
9. Runtime Permissions

Runtime Permissions Implemented:

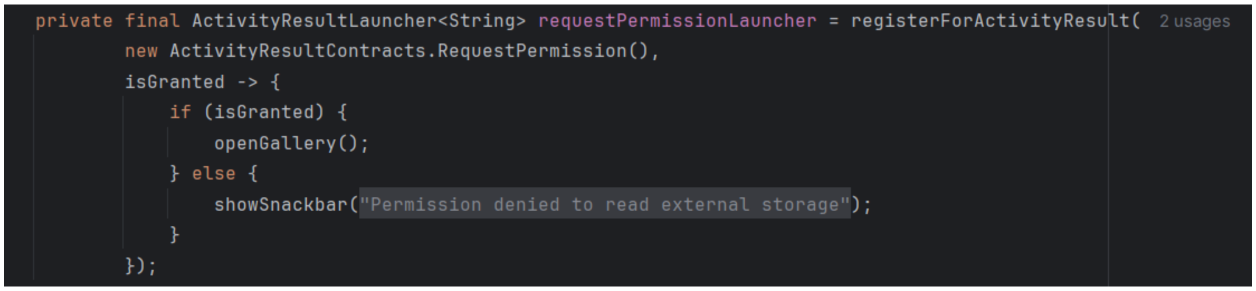
Description of Runtime Permissions Implemented and Their Purpose

1. **Location Permission**
    - The app request’s location permission to access the device's GPS and display the user's current location on the map.
    - This permission enhances the user experience by showing their real-time location within the app.
2. **Notification Permission**
    - The app checks if notifications are enabled for the app using Notification Manager. If notifications are not enabled, it prompts the user to open notification settings.
    - This permission ensures that the app can notify users of important updates, like parking availability, without the notifications being blocked.

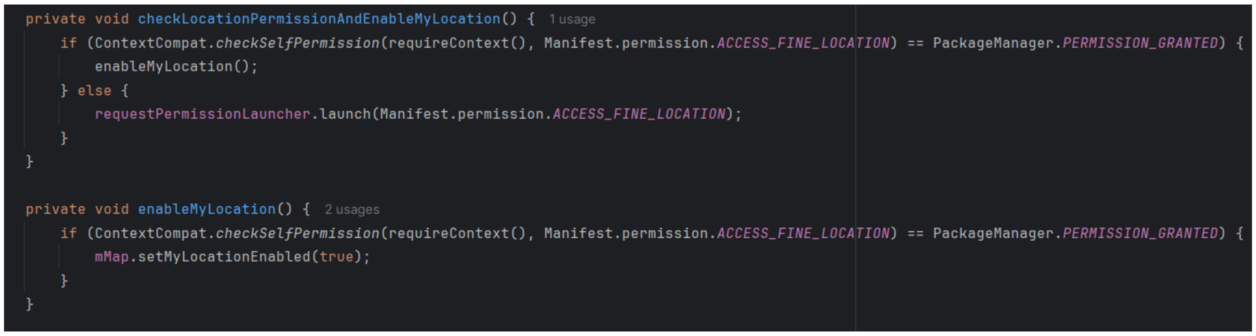
3. **External Storage Permission**
    - The app requests permission to access external storage so users can select a profile picture from their gallery.
    - This permission allows the app to retrieve images from the user's device, enabling them to personalize their profile by setting a profile picture.



```
private void requestNotificationPermission() {  1 usage
    if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.POST_NOTIFICATIONS)
            != PackageManager.PERMISSION_GRANTED) {
        // Request the permission
        ActivityCompat.requestPermissions( activity: this,
                new String[]{Manifest.permission.POST_NOTIFICATIONS},
                NOTIFICATION_PERMISSION_CODE);
    } else {
        // Permission already granted, proceed with notification
        sendWelcomeBackNotification(); // Or any notification logic
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
                                       @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == NOTIFICATION_PERMISSION_CODE) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            sendNewUserWelcomeNotification();
        } else {
            Toast.makeText( context: this, "Notification permission denied", Toast.LENGTH_SHORT).show();
        }
    }
}
```

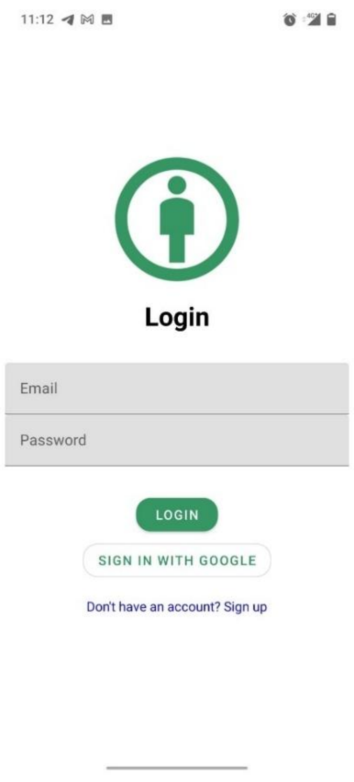


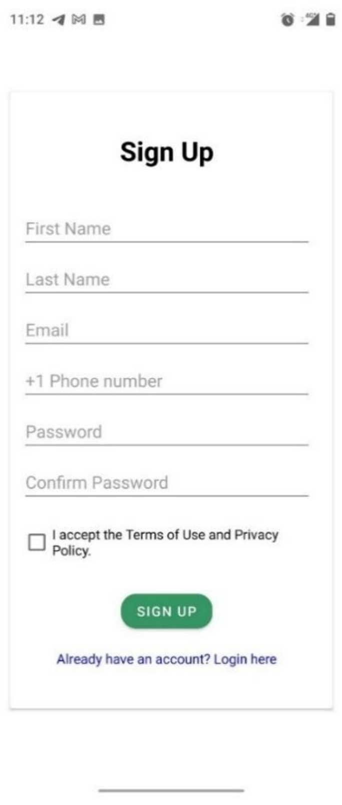
```
private final ActivityResultLauncher<String> requestPermissionLauncher = registerForActivityResult(  2 usages
        new ActivityResultContracts.RequestPermission(),
        isGranted -> {
            if (isGranted) {
                openGallery();
            } else {
                showSnackbar("Permission denied to read external storage");
            }
        });
```

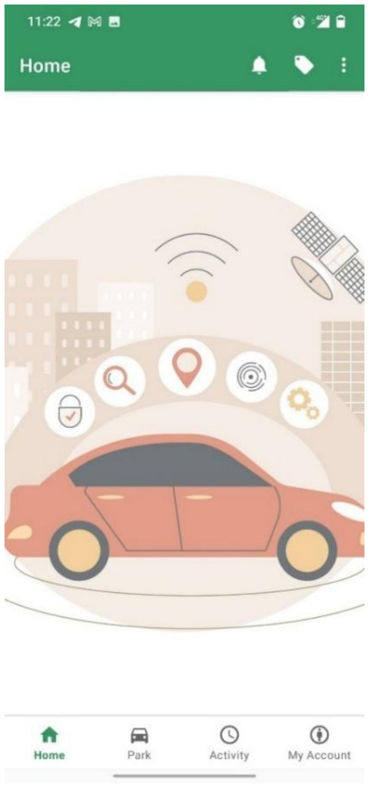


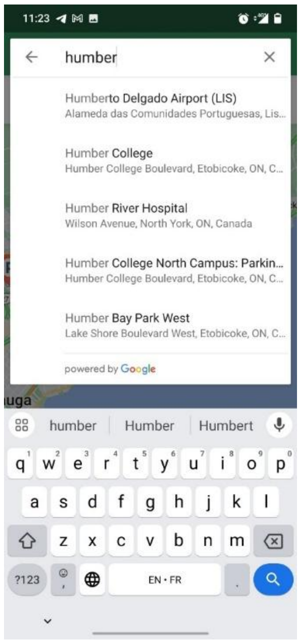
```
private void checkLocationPermissionAndEnableMyLocation() {  1 usage
    if (ContextCompat.checkSelfPermission(requireContext(), Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
        enableMyLocation();
    } else {
        requestPermissionLauncher.launch(Manifest.permission.ACCESS_FINE_LOCATION);
    }
}

private void enableMyLocation() {  2 usages
    if (ContextCompat.checkSelfPermission(requireContext(), Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
        mMap.setMyLocationEnabled(true);
    }
}
```

10. UI Design Completion








humber

Humberto Delgado Airport (LIS)
Alameda das Comunidades Portuguesas, Lis...

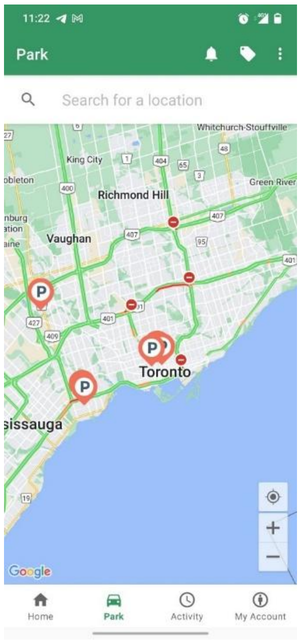
Humber College
Humber College Boulevard, Etobicoke, ON, C...

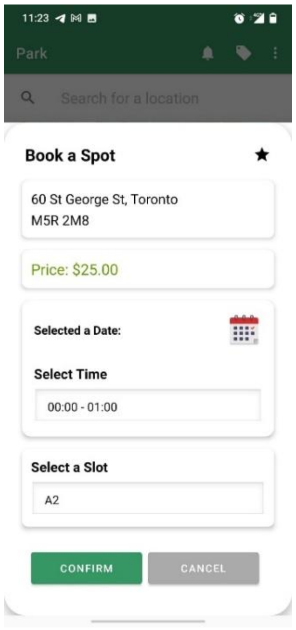
Humber River Hospital
Wilson Avenue, North York, ON, Canada

Humber College North Campus: Parkin...
Humber College Boulevard, Etobicoke, ON, C...

Humber Bay Park West
Lake Shore Boulevard West, Etobicoke, ON, C...

powered by Google




Book a Spot

60 St George St, Toronto
M5R 2M8

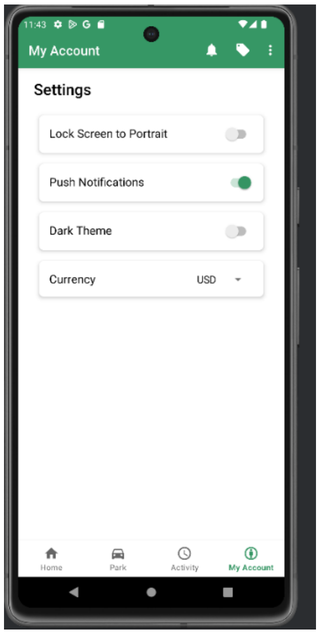
Price: $25.00

Selected a Date:

Select Time
00:00 - 01:00

Select a Slot
A2

CONFIRM CANCEL


My Account

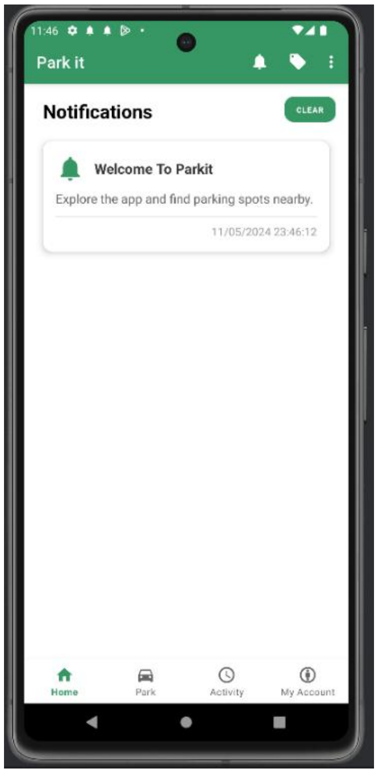
Settings

Lock Screen to Portrait

Push Notifications

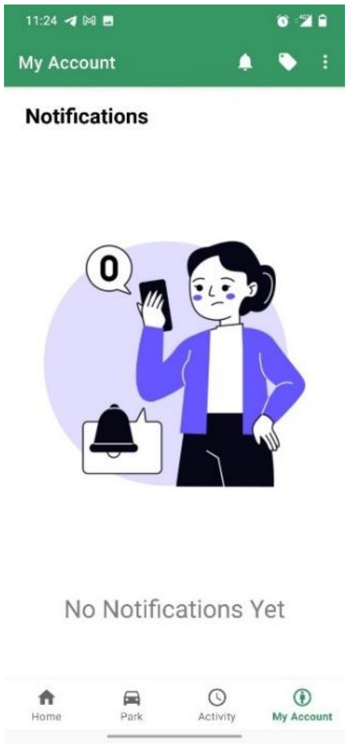
Dark Theme

Currency USD


Park it

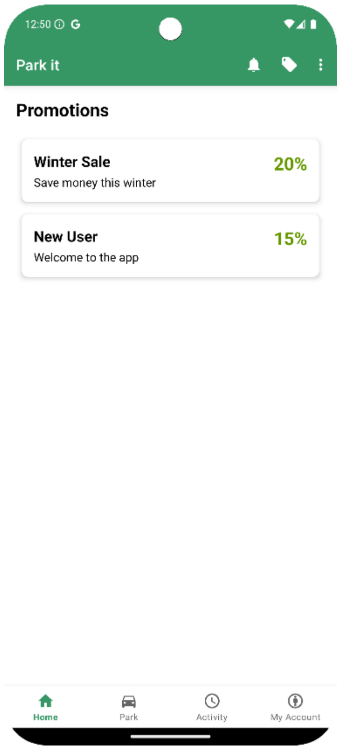
Notifications CLEAR

Welcome To Parkit
Explore the app and find parking spots nearby.
11/05/2024 23:46:12


My Account

Notifications

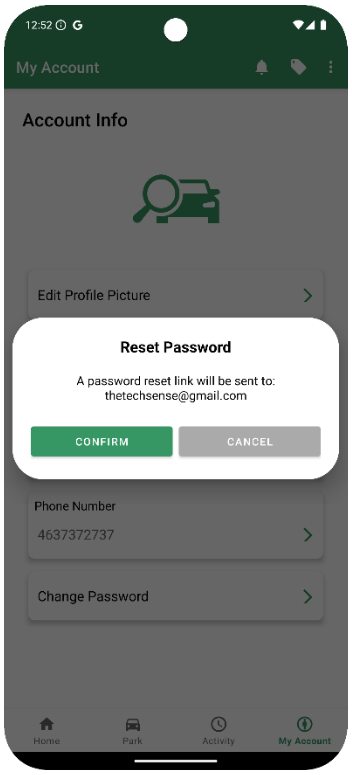
No Notifications Yet


Park it

Promotions

Winter Sale 20%
Save money this winter

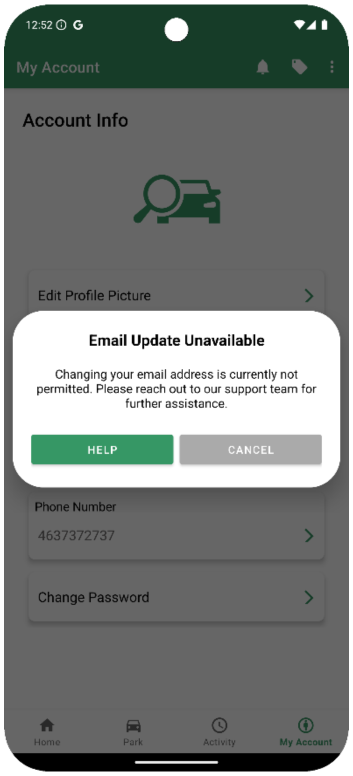
New User 15%
Welcome to the app



Reset Password

A password reset link will be sent to: thetechsense@gmail.com

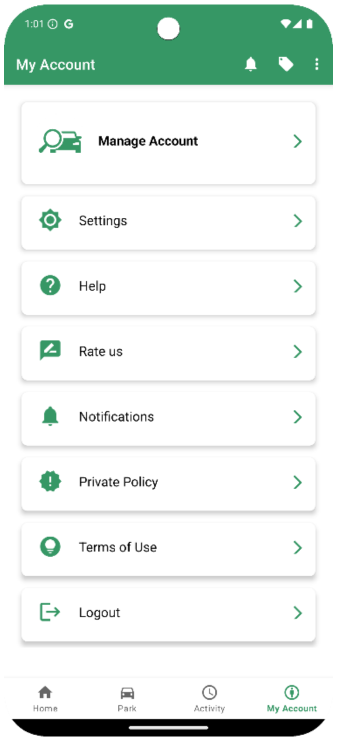
CONFIRM CANCEL

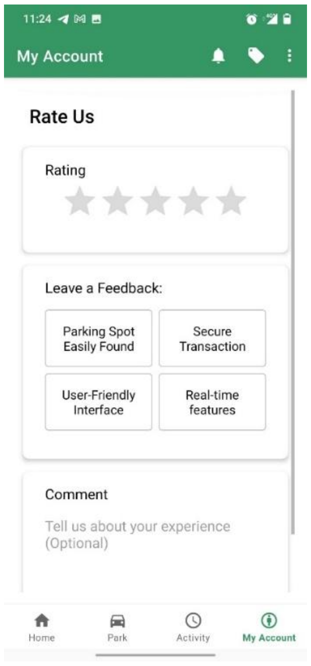


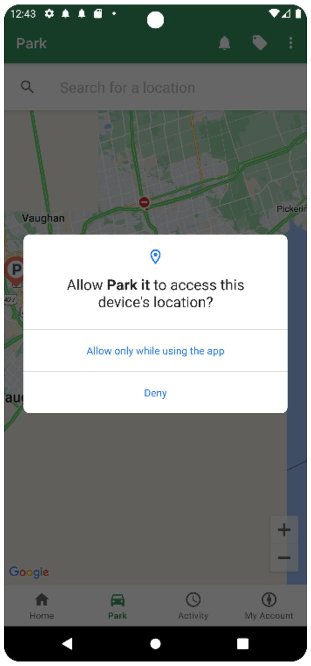
Email Update Unavailable

Changing your email address is currently not permitted. Please reach out to our support team for further assistance.

HELP CANCEL



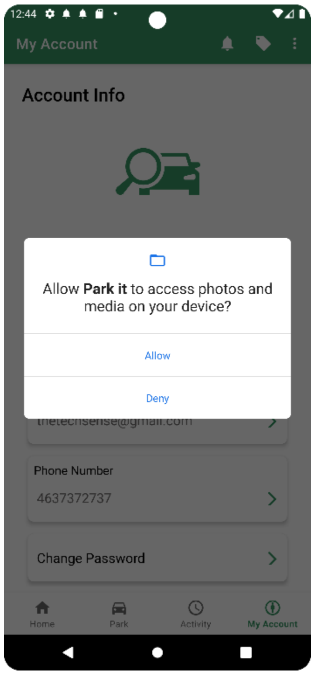




Allow **Park it** to access this device's location?

Allow only while using the app

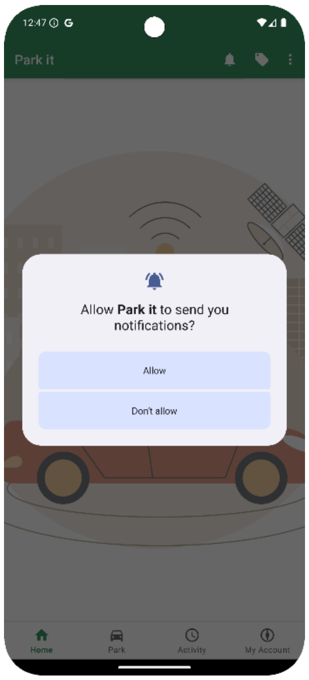
Deny



Allow **Park it** to access photos and media on your device?

Allow

Deny



Allow **Park it** to send you notifications?

Allow

Don't allow

11. Sprint goals, list sprint goals

- **Add Booking System:** Implement the functionalities for the booking system, including confirmation emails, and calendar integration.
- **Implement Notifications System:** Ensure notifications are delivered reliably and effectively to users across various channels.
- **Develop Promotions Functionality:** Create a system for managing and delivering promotions to users, including discounts, and special offers.
- **Create Help and Support Section:** Develop a comprehensive help centre with FAQs and contact information.
- **Develop Favourites Functionality:** Allow users to save their favourite locations for easy access.
- **Embed Google Maps in Park Section:** Integrate Google Maps to display parking availability and directions.
- **Add Search Bar to Google Maps Integration:** Implement a auto complete search bar to allow users to find specific locations or activities within the map.
- **Add Permanent Marker for Parking Spots in Map with Details:** Display permanent markers on the map for parking spots, along with additional details such as availability and pricing.
- **Implement Tab Layout in Activity Section:** Organize activity information into clear and easy-to-navigate tabs.
- **Implement User Feedback (Rate Us) Feature:** Allow users to provide feedback on the app through a rating system.
- **Update Login Screen Design and Functionality:** Improve the login screen's design and functionality for a better user experience.
- **Develop Settings Functionality:** Implement settings options for users to customize their preferences and manage their screen settings like portrait lock and theme change.
- **Integrate Google Sign-up Functionality:** Allow users to sign up using their Google accounts.
- **Store and Retrieve Sensor Data from Database:** Ensure that sensor data is collected and stored securely in the database and can be retrieved for analysis and display.
- **Store and Retrieve User Credentials from Database:** Securely store and retrieve user credentials to enable login and account management.
- **Develop Forgot Password Feature:** Implement a feature that allows users to recover their passwords if they forget them.

- **Update Registration Screen Design and Functionality:** Improve the registration screen's design and functionality for a better user experience.
- **Implement Manage Account Functionality:** Allow users to manage their account settings, preferences, and personal information.
- **Store User Profile Details and Avatar on Database:** Store user profile information, including their avatar, in the database to personalize the user experience.

12. Functionalities of app that were implemented in this release.

User Authentication (Login and Registration):

- **Description:** The authentication system allows users to securely sign up, log in, and manage their accounts. We integrated both **email/password login** and **Google Sign-Up/Sign-In** options for flexibility. This ensures users can quickly access their accounts and start using the app.
- **Features:**
  - **Email/Password Authentication:** Secure login and registration with email and password.
  - **Google Sign-In:** Users can sign up and log in using their Google accounts for faster access.
  - **Forgot Password:** Allows users to recover their password if forgotten.
  - **Profile Management:** Stores user details and avatars securely in the database.

Parking Spot Booking System:

- **Description:** This feature enables users to browse available parking spots and book them in advance.
- **Features:**
  - **Parking Availability Map:** Real-time updates of available parking spots displayed on an interactive Google Map.
  - **Booking System:** Users can select and reserve a parking spot for a specific time.
  - **Booking Confirmation:** Once a spot is reserved, users can view it categorized under "Upcoming," "Active," or "History" in the Activity section.

Notifications System:

- **Description:** The app includes a notifications system that alerts users about important events like booking confirmations, reminders, and promotions. The notifications are delivered in real-time.
- **Features:**
  - **Booking Notifications:** Alerts users when a parking spot is successfully booked.
  - **Reminders:** Sends reminders about upcoming bookings or available parking spots.
  - **Promotions and Updates:** Notifies users about special promotions and app updates.

Google Maps Integration:

- **Description:** Integrated Google Maps within the app to display parking spots and help users navigate to their reserved spots.
- **Features:**
  - **Search bar on Map:** Users can search for specific parking spots on the map and view details.

Help and Support Section:

- **Description:** A dedicated section for users to access help. It provides contact information for assistance.

User Feedback (Rate Us) Feature:

- **Description:** The "Rate Us" feature allows users to provide feedback on the app's performance and functionality. This helps improve the app by gathering user insights.
- **Features:**
  - **User Feedback Collection:** Allows users to rate the app on a scale and leave comments or suggestions.

Activity Fragment Feature:

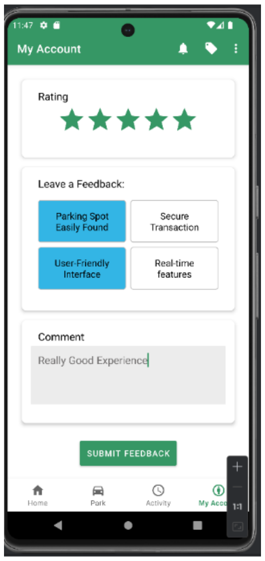
- **Description:** In our parking app, we have different fragments for displaying active bookings, booking history, and upcoming bookings.
- **Features:** Each fragment would handle its own data fetching and UI updates, while the hosting activity manages the overall navigation and lifecycle.

13. Additional Features Added Since Deliverable 2:

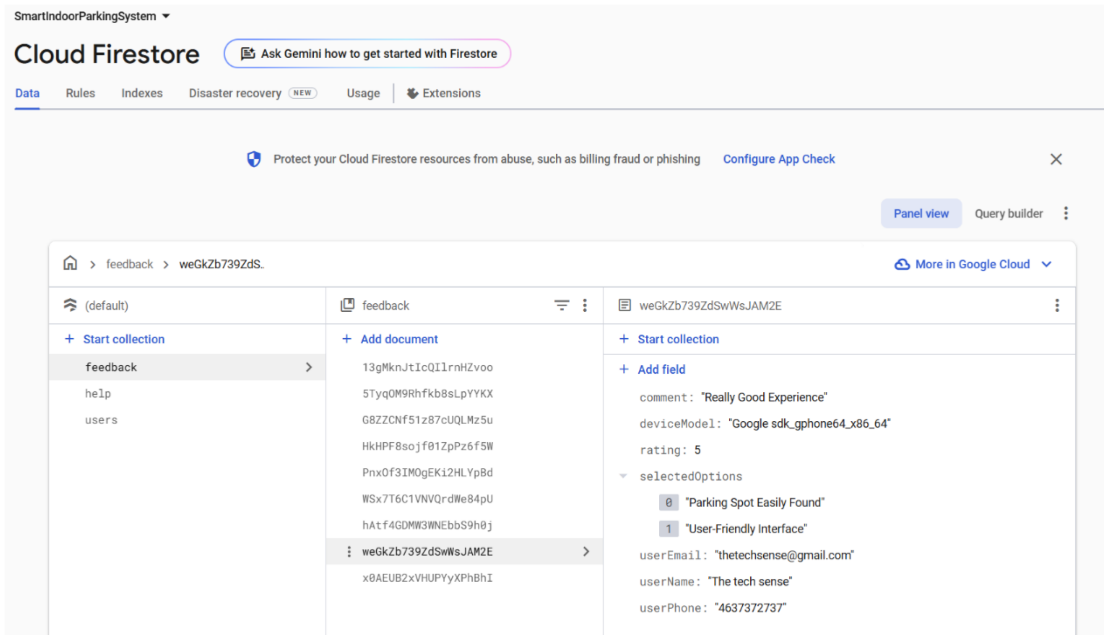
In this sprint, we enhanced the app by adding functionality to all the menu bar items. This includes a notification system, a promotions screen, a help and support section, and a favourites feature where users can save specific parking spots for future reference. In the main layout, we implemented key functionalities across three main sections.

First, in the Park fragment, we embedded Google Maps along with an autocomplete search bar. On this map, we added permanent parking spot markers for users. When a user clicks on a marker, a bottom sheet dialog appears, prompting them to book a spot and select the duration. Next, we have the Activity fragment, which displays all bookings organized into three tabs: Active, Upcoming, and History, based on the booking status. Lastly, we removed Navigation Drawer and the Manage Account tab includes multiple options with unique functionalities, such as Settings, Help, Account Details, and Logout, each working smoothly to enhance user experience.

14. Provide screenshot how the data from the Customer Feedback Screen stored in the DB.



Really Good Experience



Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing

feedback > weGkZb739ZdS.

(default): feedback, help, users

feedback:

```
13gMknJtIcQIlrnHZvoo
5TyqOM9Rhfkb8sLpYYKX
G8ZZCNf51z87cUQLMz5u
HkHPF8sojf01ZpPz6f5W
PnxOf3IMOgEKi2HLYpBd
WSx7T6C1VNVQrdWe84pU
hAtf4GDMW3WNEbbS9h0j
weGkZb739ZdSwWsJAM2E
x0AEUB2xVHUPYyXPhBhI
```

weGkZb739ZdSwWsJAM2E:

```
comment: "Really Good Experience"
deviceModel: "Google sdk_gphone64_x86_64"
rating: 5
selectedOptions
    0 "Parking Spot Easily Found"
    1 "User-Friendly Interface"
userEmail: "thetechsense@gmail.com"
userName: "The tech sense"
userPhone: "4637372737"
```

15. Conclusion

The Smart Indoor Parking System project has made significant progress, with most key features successfully implemented. These include user authentication, parking spot booking, Google Maps integration, notifications, and promotions.

During Sprint 3, the team completed important tasks such as integrating Google Sign-Up, developing the Forgot Password feature, and setting up personalized user profiles. We also focused on enhancing the user interface for a better overall experience.

While there are still a few tasks to complete, such as finalizing the payment functionality for the booking system and refining the UI, the project is on track. The team has overcome challenges and collaborated effectively to meet the project goals. The final product will offer a seamless and efficient solution for indoor parking, making it easier for users to book and manage their parking spots.