

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: `abhi472`

The Founder's Cafe

Description

An app which is made for The Founder's Cafe, a coworking space, to provide booking and tracking functionality, for coworking members

Intended User

Startups, Freelancers which are associated with The Founder's Cafe , using their coworking space, booking meeting rooms and office space provided by TFC

Features

The Main features are

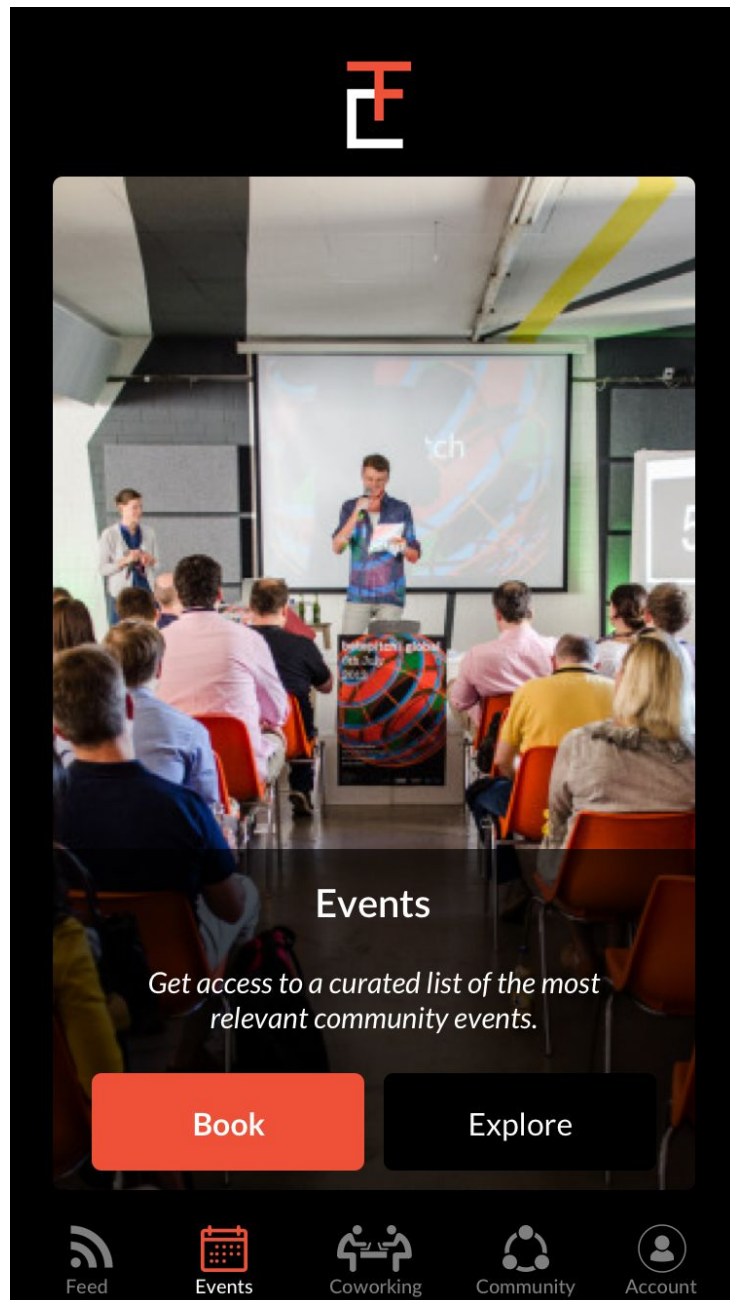
- Register yourself as an individual or a team
- Buy Team-Room, Coworking Space, Meeting Room , Virtual office etc.
- Register you events at TFC's facility.
- Join any listed events.

- Passwordless Login using OTP (although i could've used Account-kit sdk for this but my client refused).
- Create a team and refer members to join.
- Newsfeed section shows you blogs from different categories which you can select.
- The future functionalities will include use of IOTs like coffee machine fax- machine, VOIP using the app(currently not part of the app).

User Interface Mocks

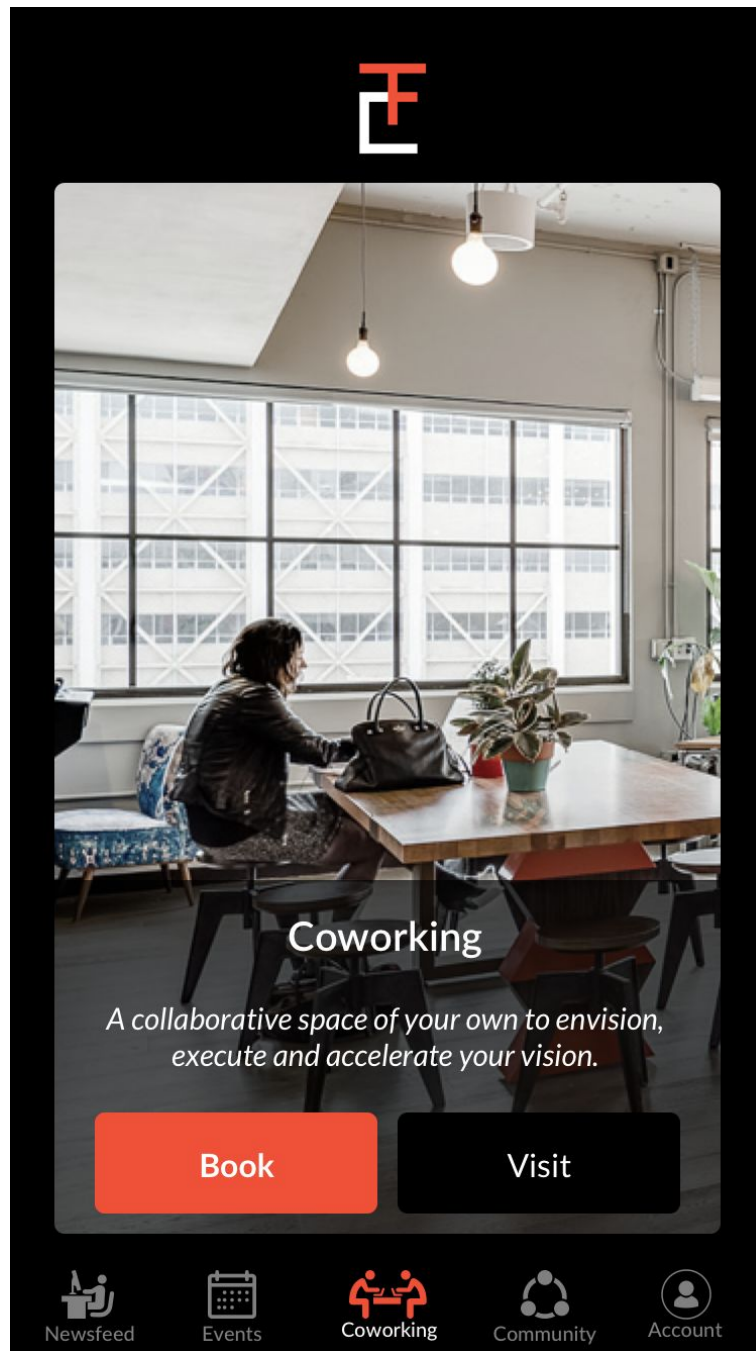
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



This is one of the elements of the home-screen .

Screen 2



Home-screen Coworking

Screen 3

The screen displays a 'Schedule Visit' form. At the top, there is a back arrow, the text 'Your City' above a location pin icon and 'Noida', and a logo. The form has two sections: 'Schedule Visit' and 'Contact Details'. The 'Schedule Visit' section contains two rows: 'Select Date' with the value 'Thu, 01/12/2016' and 'Select Time' with the value '3:30 PM'. The 'Contact Details' section contains three input fields for 'Name', 'Email', and 'Phone'. A large orange button labeled 'Confirm Visit' is at the bottom of the form. Below the form is a navigation bar with five icons: Newsfeed, Events, Coworking (highlighted), Community, and Account.

Back arrow

Your City
📍 Noida

Logo

Schedule Visit

📅 Select Date Thu, 01/12/2016

🕒 Select Time 3:30 PM

Contact Details

Name

Email

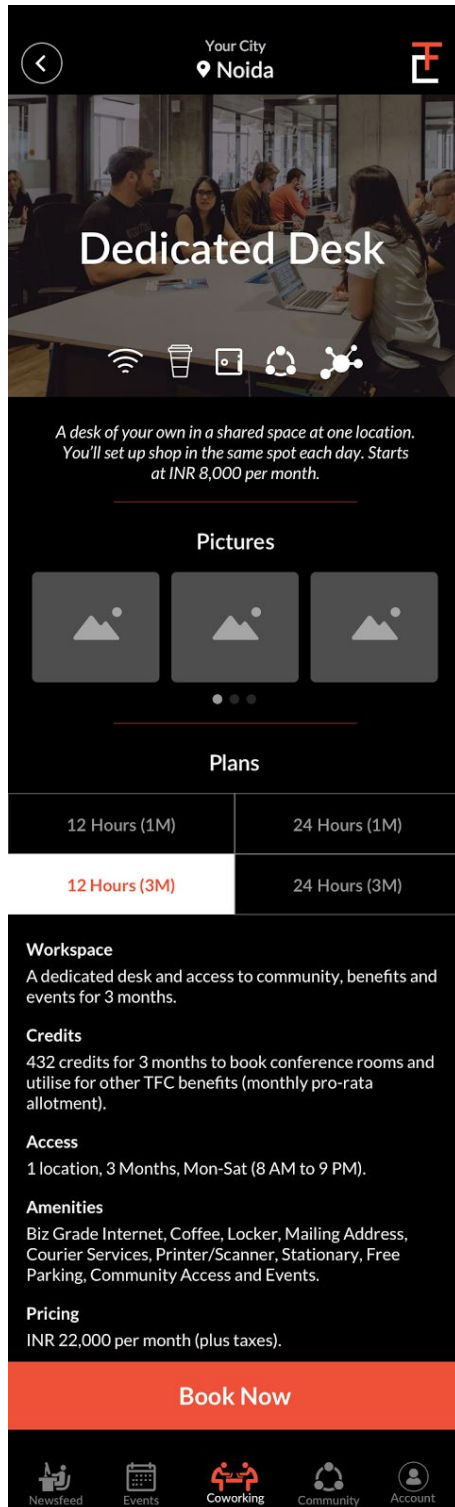
Phone

Confirm Visit

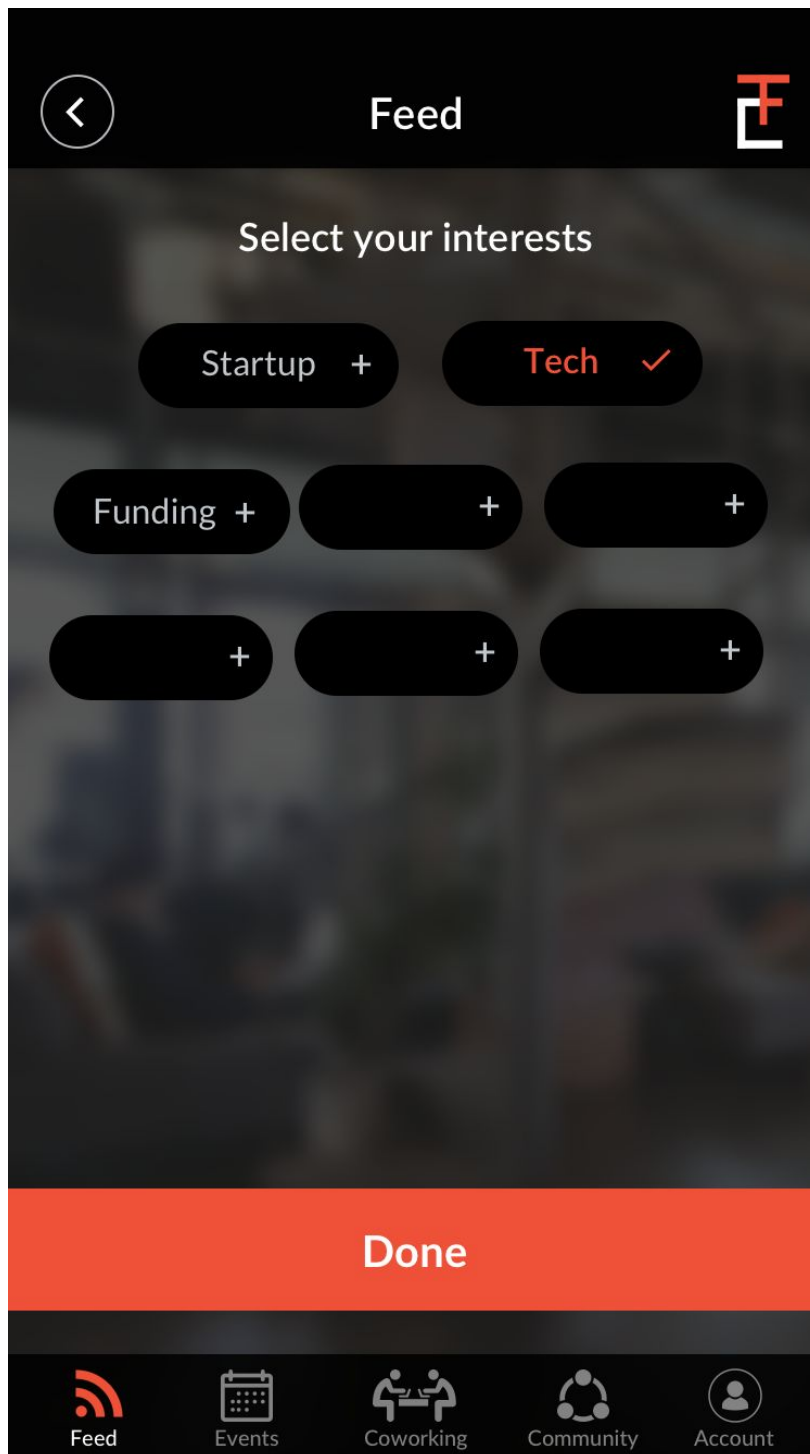
Newsfeed Events **Coworking** Community Account

Visit Facility Screen

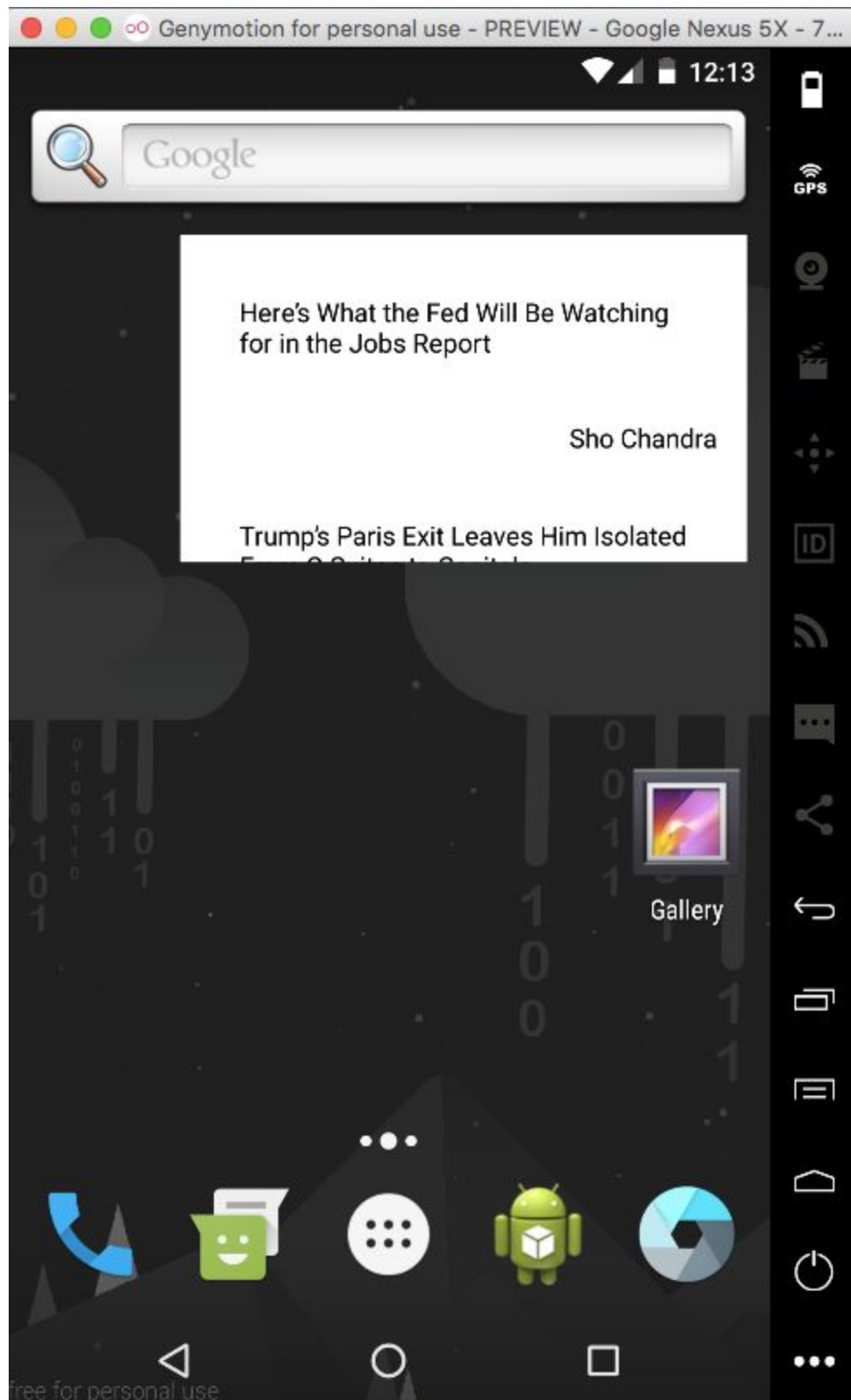
Screen 4



Screen 5

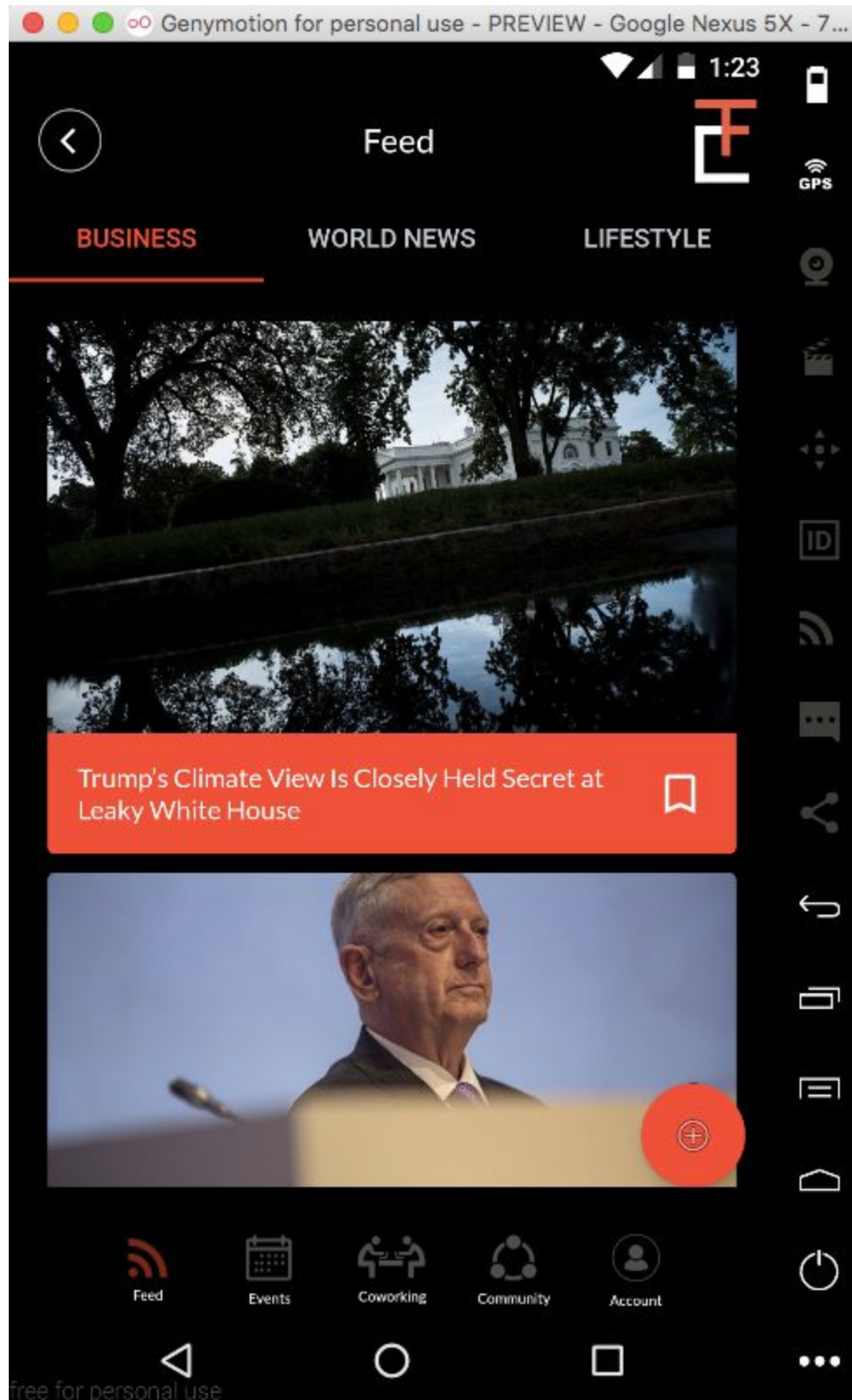


Screen 6



UI for app widget showing list of saved news feed

Screen 6



Key Considerations

How will your app handle data persistence?

There is no much need of data persistence at the current moment in the app, but for the sake of requirements i am persisting the cities available in our app, making my own content provider.

Describe any corner cases in the UX.

UX in this case had a peculiar demand from client, the app as you can see has a tablayout with a viewpager in the home screen, so we can swipe to get to the next fragment. But apart from this i was required to make those tabs as navigation across all activities.

To explain this please check screens 2 , 3 , 4. On screen 2 i am on homescreen coworking if i click on book i go to screen 4 and if i click on visit i go to screen 3 but as you can see both of these are part of coworking section so the tab is showing coworking as marked.

To implement this i am using multiple fragment and few activities in my apps, hiding the different fragment parts according to different cases.

Describe any libraries you'll be using and share your reasoning for including them.

1. Volley to handle network request.
2. Picasso for caching and loading images.
3. Calligraphy for setting fonts to different layouts' texts.
4. Butterknife for dependency injection.
5. Jackson for JSON parsing
6. Razorpay for online payments.

Describe how you will implement Google Play Services.

I am using admob and Firebase analytics in my app.

Although admob has been added for the purpose of the project and has no need in the app itself.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

There is nothing major with this project build , you only need to enter razorpay test id in manifest along with fabric's api key.

Task 2: Implement UI for Each Activity and Fragment

- Build ui for splash screen
- Build HomePage ui
- Build HomePage Fragments' ui
- Build sub-pages of each of these fragments

Task 3: Your Next Task

- Integration of functionalities to each fragment.
- Adding margins and sizes according to the ui mock up.
- Making custom layouts for multiple selection spinner with disabled slots.
- Making custom layout for toolbar.

Task 4: Your Next Task

-

Task 5: Your Next Task

- Implementing Payment gateway using Razorpay sdk.
- Implementing Firebase Crash reporting.
- Implementing Firebase Analytics.

Other Considerations :

As mentioned that this app is made for a client, my app doesn't support tablet ui of screen rotation because it didn't need these features at first place.

Also this app uses volley for networking hence use of AsyncTask becomes redundant.

But i do understand that these are the key metrics that need to be fulfilled in order to get this app passed

So i have come with a workaround,

There is this section called news feed as you can see in screen

Now i can add screen rotation , tablet ui , loaders and asyncTask to this section of the app (no other part of the app will have screen rotation except for this news feed).

Now my question is if i do that does my app qualifies for clearance?

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"