
- Description
- Intended User
- Features
- User Interface Mocks
  - MainActivity
  - SettingActivity
- 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 Main Activity and Fragment
  - Task 3: Implement UI for Setting Activity
  - Task 4: Add SharedPreferences Data
  - Task 5: Update the MainActivity

GitHub Username: PratikVishwakarma

inTouch

Description

inTouch is a news app. It provides all the summarised stories contain only headlines and facts, to help you stay informed of the current affairs. Whether it's the latest government policies or shake ups in bollywood. This App is easy to operate and user can also choose the types of information which they want to read.

Not sure how to write a good description? Search 5-star apps on the Play Store for inspiration.

Intended User

This app is for all the user who can understand english or hindi and this app is only for android platform so user must have a android device. This app is for all types of user like Students, Businessman and woman, Families etc.

Features

List the main features of your app. For example:

- Show news or facts in both english and hindi
- Show the images related to the content
- User can choose category
- If internet connection is slow then user can also turn off the image so the app can work perfectly even in slow internet connection

User Interface Mocks

MainActivity

This is the first screen which contains actual news, it is a fragment on mainactivity screen. From this screen we can redirect to the settingActivity and to the web Browser which will go to the resource url provided at the bottom of the screen.

MainActivity screen content

- News Image
- News Headline
- News Content
- News Source
- Setting icon
- Refresh icon

SettingActivity

This is the basic concept of the second screen which is SettingActivity, it provides different options for user to configure the news like by selecting the category or language.Which will I provide on the second screen that is SettingActivity.

SettingActivity screen content

- Language Selector (Radio Button)
- Categories Selector (Icons)
- Show or Don’t show news image option

Key Considerations

How will your app handle data persistence?

For backend this app is fetch data from the Firebase. This app is connect to the firebase whenever the app is start and fetch the new data from the cloud. The app structure is according to the Firebase structure, it means each parent or child node contains a POJO in the app so they can fetch the content and use them very easily.

Describe any corner cases in the UX.

This app contains two activities MainActivity and SettingActivity. MainActivity is the launcher activity so on back press in paused the app. Setting Activity is to change the default setting, on back press or on select any category it goes to the MainActivity and show the related content.

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

This app use Picasso to handle the loading and caching of images.

Describe how you will implement Google Play Services.

This app use Firebase and it's dependent Google Play Services.

Next Steps: Required Tasks

First I connect this app to Firebase.Then create the UI part according to the content I want to deliver to the user and then create POJOs of every parent, child and sub-child present in the Realtime database.
The main feature of this app is to show the latest news or facts to the user so on start this app will fetch all the latest news from the Firebase cloud. And then store them into ArrayLists and set them to the adapters. This app also image related to the content. But Realtime database only contain the image name. The image is fetch from the Firebase Storage if image option is on otherwise it can not fetch the image.
This app requires internet connection to run because it can not store any data for offline working.

Task 1: Project Setup

This app is developed in different steps

First configure the app and add the required library and dependencies.

- Configure Firebase to the App
- Paste the “google-services.json” file in the app folder
- Add the Picasso library
- This app requires Internet to run so add Internet-Permission in AndroidManifeast.xml

Add the required resources and layouts

- Add all the strings to the strings.xml file
- Add all the styles or color to the styles.xml and color.xml

Creating support packages

- Create model package for storing the POJOs of the news.
- Create utils package for any support file.

Task 2: Implement UI for Main Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build UI for Fragment_Main
- Build the Transition for activity calling
- Add animation for icons

Task 3: Implement UI for Setting Activity

Create SettingActivity.java. This activity provide the different category of the news to the user so, user can select the category of news in which they are interested.

- Create layout
- Add icons and images
- Create elements

Task 4: Add SharedPreferences Data

Create sharedPrefrence to store the current settings

- Initialize the shared preference and set it on “All News” by default.
- Fetch the news according to the stored category.
- On changing modify the value stored in shared preference.
- Then refetch the data.

Task 5: Update the MainActivity

In this phase I update the code of mainActivity for better working or offline working :

- Check the internet connection first.
- If internet not available and value on sharedPrefrence is exist, then show the stored news.
- If internet available fetch new feeds and update the value of sharedPrefrence.
- If (!internet && storeData == null) then show a Toast “internet is required for very first time”

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**”