

[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: [ansh94](#)

Daily Tech

Description

Daily Tech is an Android app that allows users to read their favorite technical news headlines everyday and share it with their friends easily. Users can also save the news they like for offline reading.

Intended User

This app is intended for anyone who is a technology lover and wants to keep himself abreast with the latest technical happenings in the world.

Features

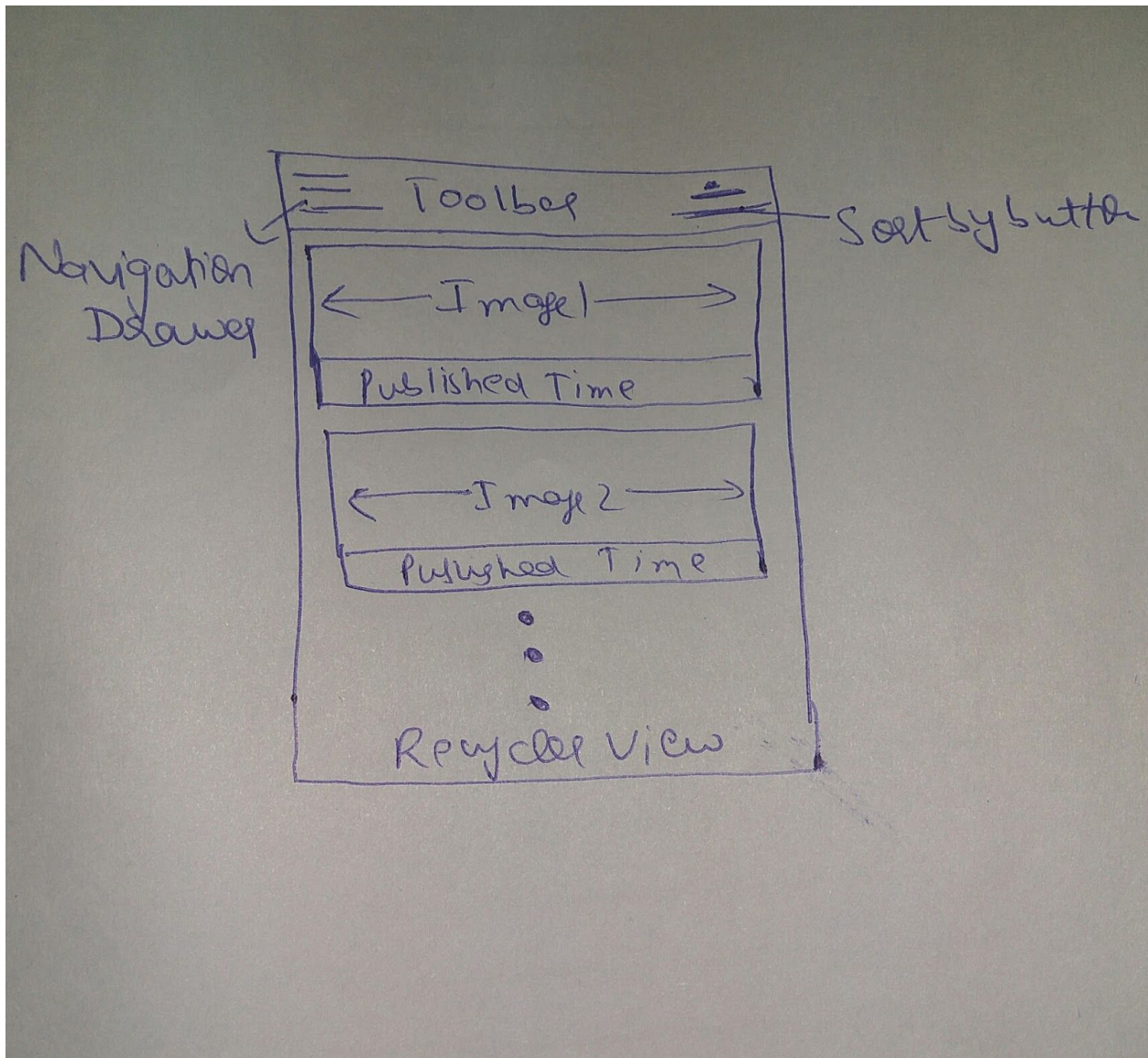
The main features of the app are:-

- Read latest news updated everyday.
- Ability to choose different sources for technical news.
- Favorite/Bookmark news headlines that you like so that you can read them later.
- Sort news by top, latest and popular.

User Interface Mocks

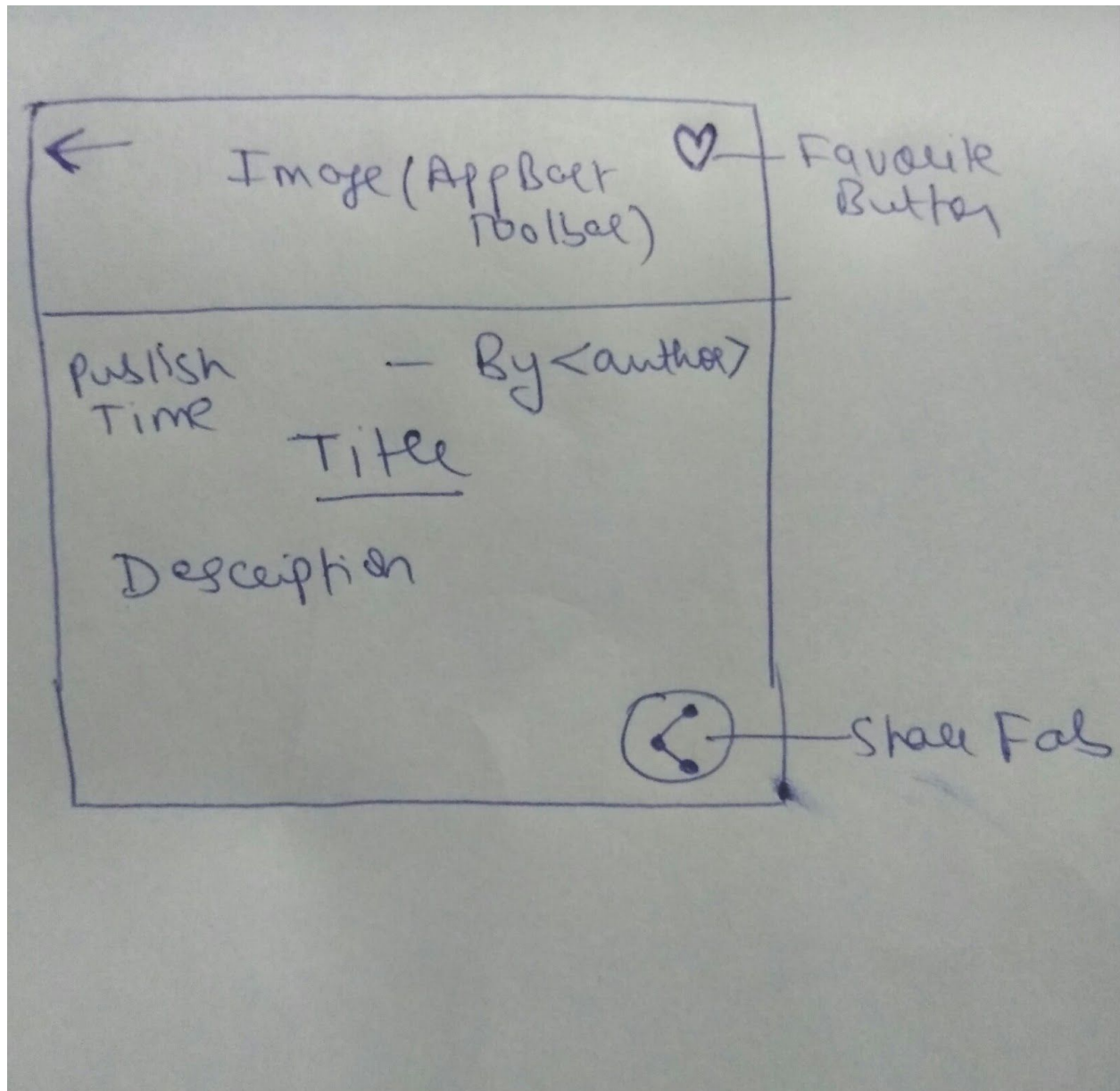
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1- Main Screen



The main screen which shows a list of news headlines in a recycler view with each item in a card view which shows the image of the news url and published time. It also has a navigation drawer which shows the list of sources from which the user can choose. The sort by button allows users to sort the news by top, latest or popular.

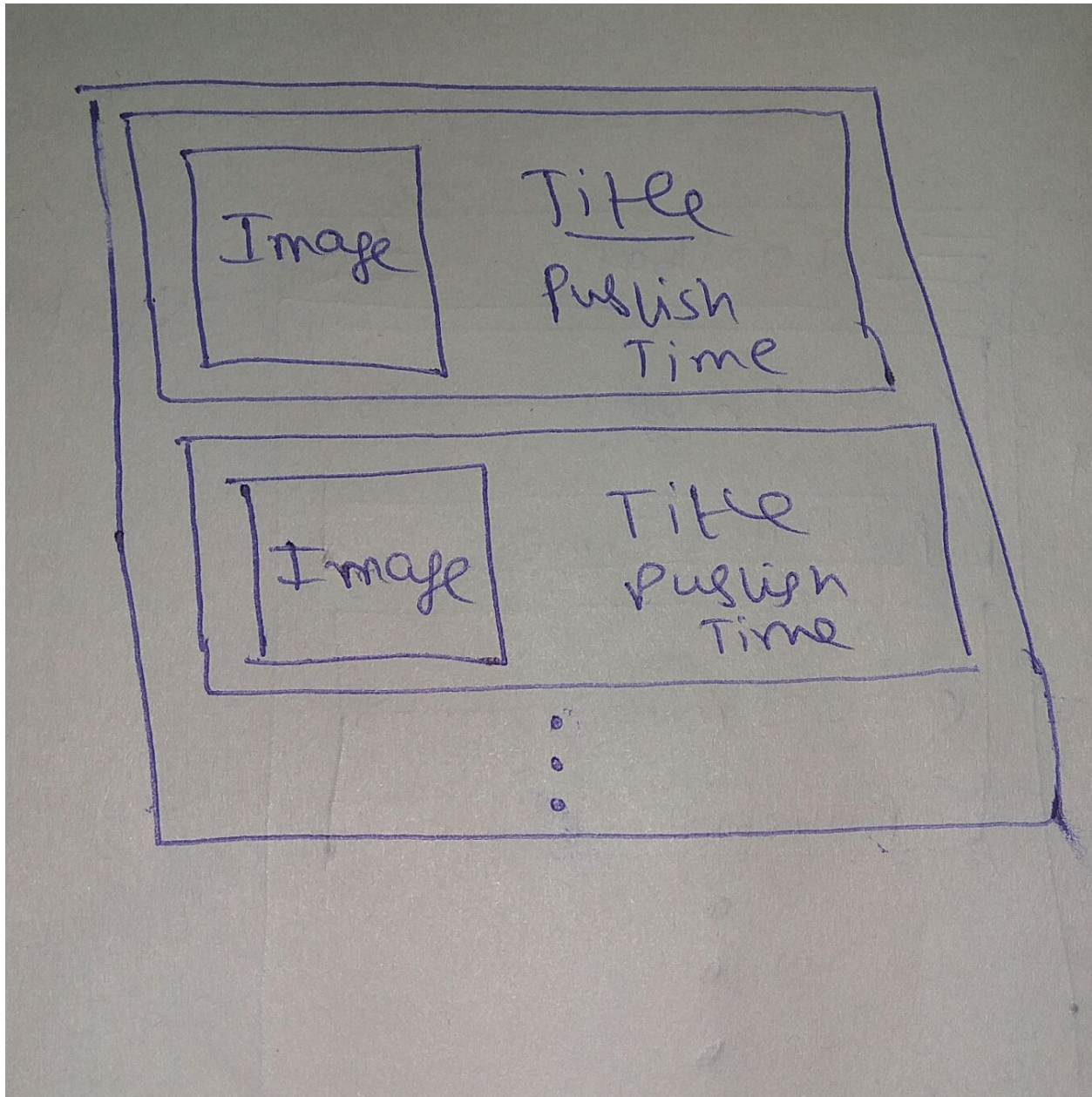
Screen 2 - Details Screen



This is the detail screen which is shown when a user clicks on a news article from the main screen. It shows all the metadata about the news such as image, title, author, description. It also

has favorite button to allow users to favorite/bookmark the news. The share fab allows users to share the news with their friends.

Screen 3 - Widget UI



This is the ui for the collection widget that will be shown on the home screen. It shows all the news articles in a recycler view. Clicking on a news article will take you to the detail screen of that article.

Key Considerations

How will your app handle data persistence?

Data persistence will be handled using SQLite database(Content Providers) and Shared Preferences.

Describe any corner cases in the UX.

A corner case would be rotation of the device when the news is being loaded.

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

- Picasso to handle the loading and caching of images.
- Retrofit to make network calls to News API
- Butterknife for view binding

Describe how you will implement Google Play Services.

I will be using Google Admobs for showing ads in the detail screen and Firebase for Analytics, Crash Reporting, Storage.

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

First I will initialize a new project in Android studio and do the following steps

- Add all the required dependencies
- Make proper packages for easy code maintenance.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for DetailActivity
- Build UI for SettingsActivity/ Preference Fragment
- Build UI for Widget

Task 3: Implement the Main Activity

Write the code for showing the news articles on main screen

- Use retrofit to make network calls to the news api
- Make the NewsAdapter for showing and populating the news data in a recycler view.
- Implement the navigation drawer for showing and loading different news articles based on source selected from navigation drawer.
- Implement sorting news feature based on top, latest and popular.

Task 4: Using loaders, handling Data persistence and Background Tasks

- Write code for Detail Activity.
- Implement data persistence when user presses the favorite button.
- Implement Loader(CursorLoader) to asynchronously load favorite news data from database.
- Use IntentService and/or FirebaseJobDispatcher to regularly update data from news api in the background
- Implement sharing feature when the user clicks on share fab.

Task 5: Clean up and finalizing

- Implement the widget functionality
- Clean up the code and move all strings to strings.xml
- Check for error cases, app crashes and do the required error handling

Add as many tasks as you need to complete your app.

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**"