GitHub Username: Ravirajbhatc

StockTalk

Description

"StockTalk" gives information about price variation of your favourite stocks, thus helps you to understand price variation of stock over the periods and it lets you you to make better decision on trading.
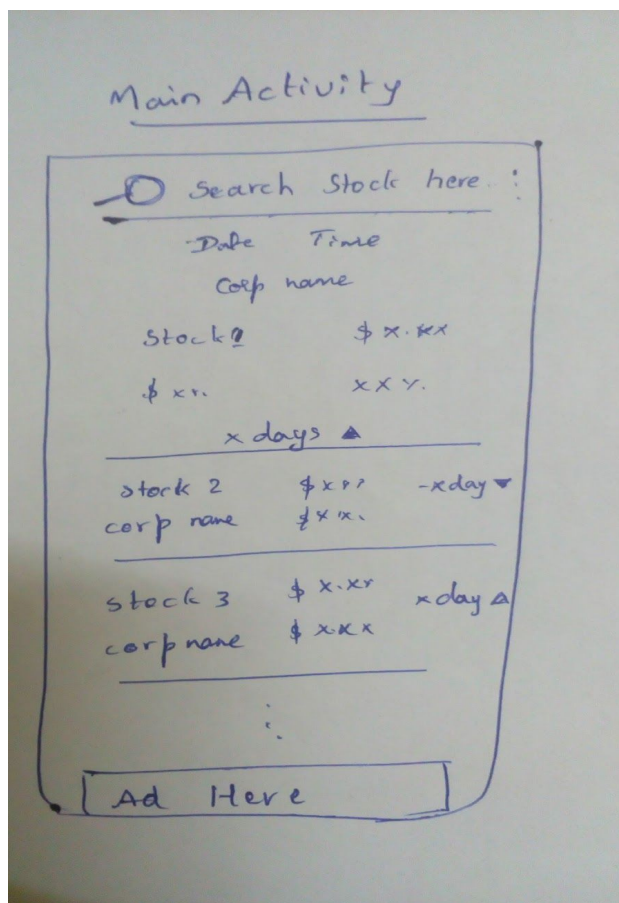
Intended User

Stock traders

Features

- Search and List stocks that you are interested
- Stock price changes updation
- Graphical chart representation of stock price variation
- StockTalk widget which gives stock market close information

User Interface Mocks

Screen 1 - Main Screen

This is the Main Activity UI. Search bar is at the top, menu will provide the option to sort the list information based on Alphabetical order, % change, recent closure etc. The first item will be presented in bigger than the rest item. After clicking on item user can get the details information.

Screen 2 - Detail Screen in Portrait Mode
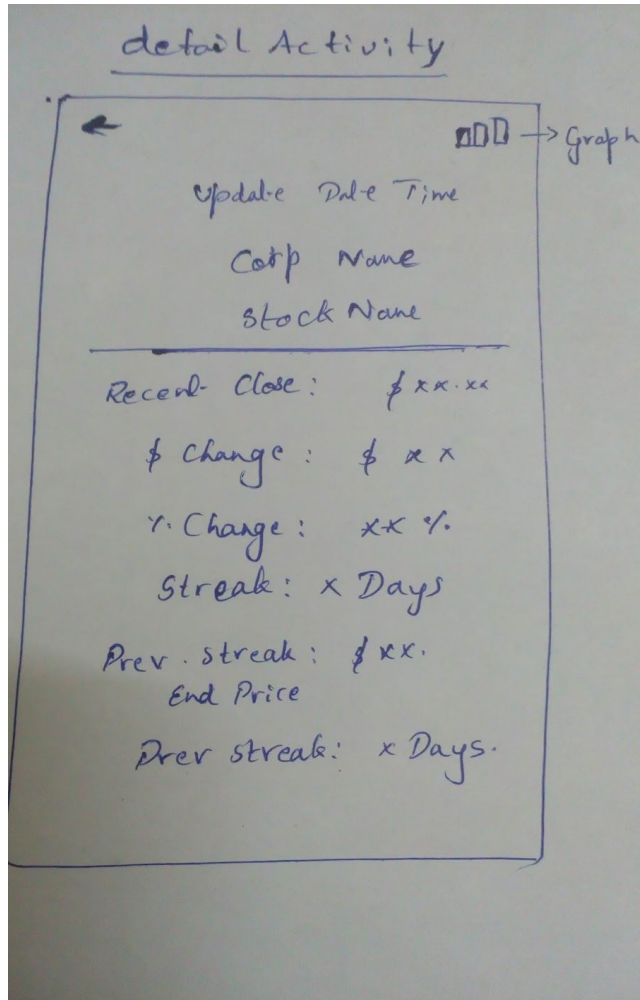
This is the UI for the Detail Activity in Portrait mode. The back arrow goes back to Main Activity and the Graph button in the top right will allow you to see the stock details in graphical format for last 1 year.

Screen 3 - Detail Screen in Landscape Mode
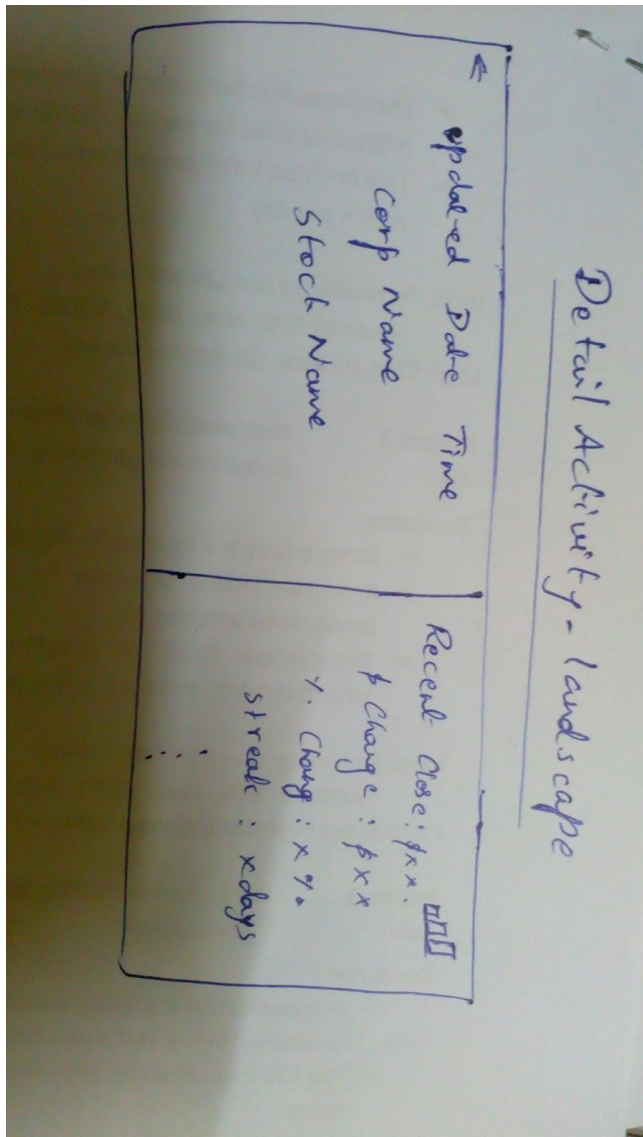
This is the UI for the Detail Activity in Landscape mode.

Screen 4 - App widget

This is the UI mock for app's widget.

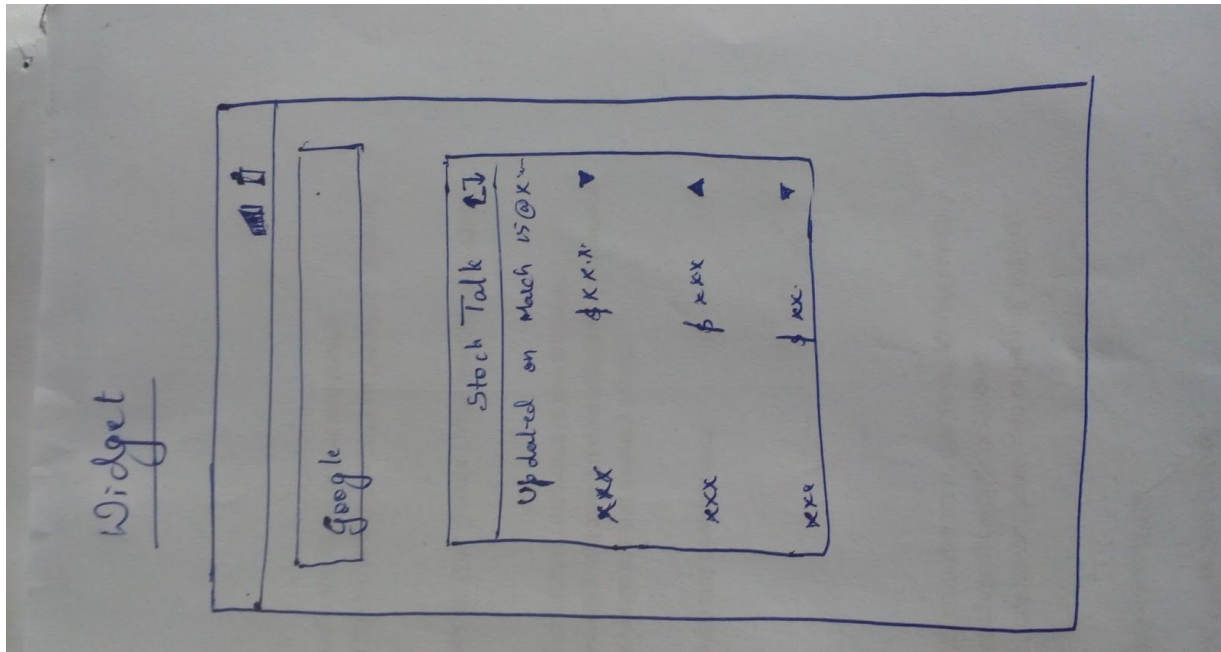# Key Considerations

How will your app handle data persistence?

The app will use a content provider that will save data to the SQLite database.

Describe any corner cases in the UX.

User return to a Stock listing screen if they hit the back button from the individual stock detail page.

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

1. Event Bus API to Send events across fragments and activities :
    https://github.com/greenrobot/EventBus
2. Yahoo Finance API to fetch stocks information : http://financequotes-api.com/

3. Android Chart API very flexible and easy API to implement different kinds of chart in android apps: https://github.com/PhilJay/MPAndroidChart
4. Google Play Service – AdMob will be used to show Interstitial Ad
5. Google Play Service – Analytics to monitor user actions and crash reports.
6. Recycler View extension library which provides advanced features : https://github.com/h6ah4i/android-advancedrecyclerview
7. A beautiful, easy-to-use, and customizable dialogs API : https://github.com/afollestad/material-dialogs
8. Schematic: Automatically creates a Content Provider backed by a SQLite database. It's very simple over building the database manually (optional) : https://github.com/SimonVT/schematic

Describe how you will implement Google Play Services.

- Google Play Service – AdMob will be used to show Interstitial Ad
- Google Play Service – Analytics to monitor user actions and crash reports.

Next Steps: Required Tasks

Task 1: Project Setup

- Research on similar apps and learning from their experiences.
- Creating the design architecture for our application.
- Creating project using latest version of android studio and specifying Target, minimum SDK
- Configure libraries

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for MainFragment list items.
- Build UI for Detailed Fragment.
- Build UI for Menu items.

Task 3: Implement Yahoo Finance API

- Query Information required using Yahoo finance API based on input in search text box.
- Display queried stocks information in UI using cursorloader.

Task 4: Data Persistence

- Implement Content Provider.
- Store required information in database as per design requirements.
- Build the logic to create/modify/delete the database information.

Task 5: Implements Widgets & SyncAdapter

- Build UI for Collection Widget to show information from main fragment.
- Implement widgets logic.
- Implement SyncAdapter to sync stocks information at market close time.

Task 6: Google Play Services implementation

- Create an Interstitial Ad to be shown upon App exit fragment.
- Use Analytics to monitor user actions and crashes that occur.

Task 7: Responsive Design Implementation

- Design for mobile landscape mode.
- Design for tablet in both portrait and landscape modes.
- Fix the issue data persistence because of UI/UX change

Task 8: Build & Sign

- Create a signed apk version