GitHub Username: cybrum

My Stock Health

Description

This app can be used to monitor the stock market, the performance of stock portfolios and decide investment strategy. Users can add their stock information and track their portfolios.

Intended User

This app is for stock market users.

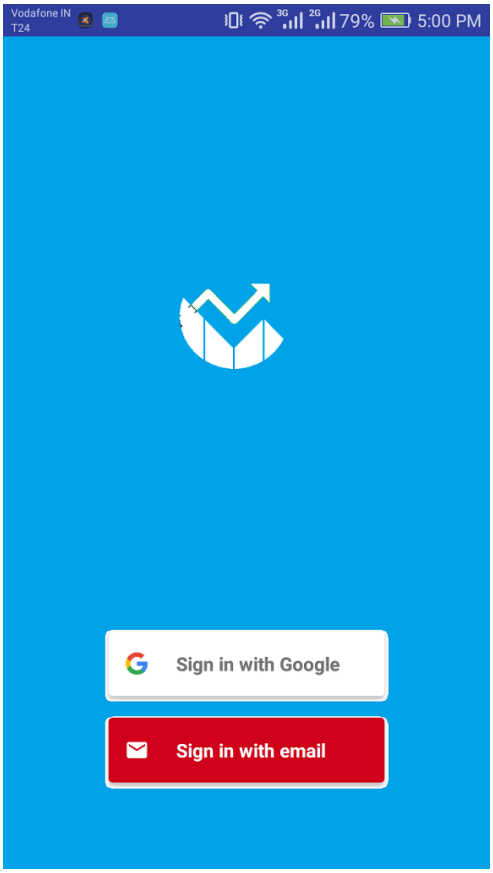
Features

This application monitors the stock market, the performance of your stock portfolios.
This includes:
✓ Login /Logout
✓ Secured Access to data.
✓ Save in a Real-time Cloud Database

✓ All portfolios and stock market quotes are stored locally on your Phone/Tablet.
✓ Visual representation of stock performance.
✓ Stocks widget that displays stock quotes and holdings

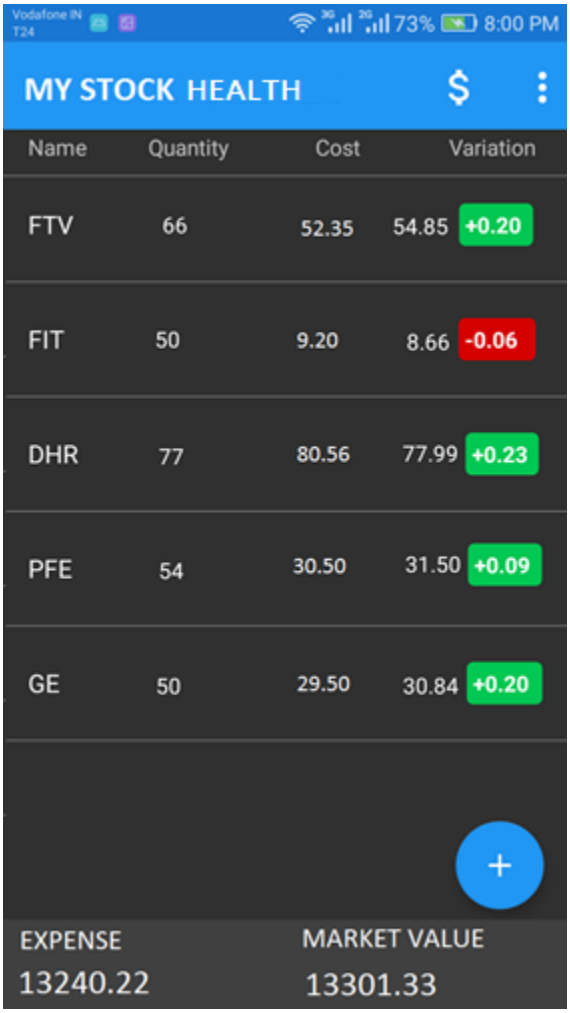
User Interface Mocks

Screen 1

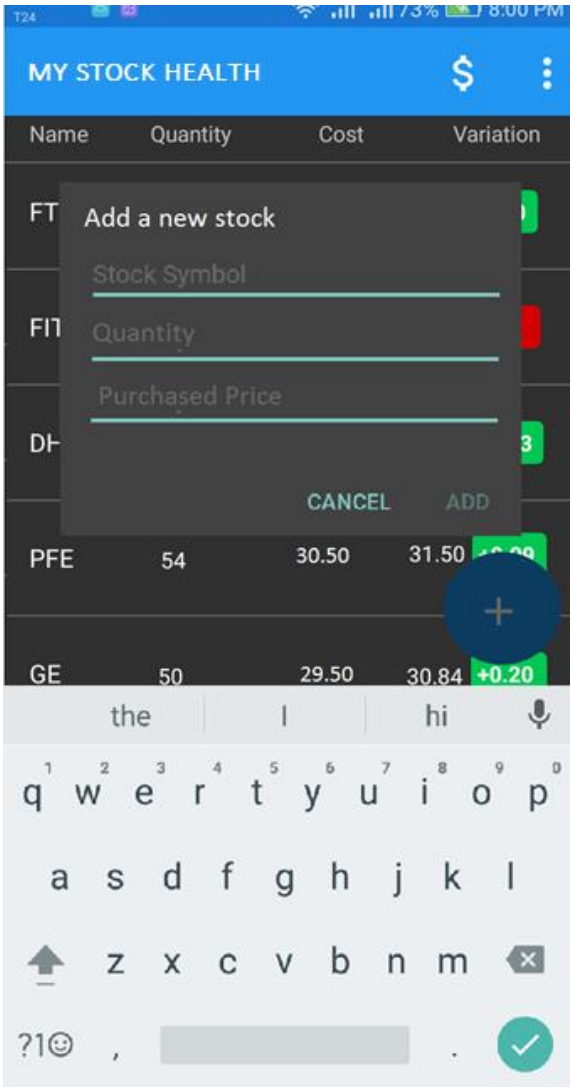


This screen will allow user to complete the login process through Google.

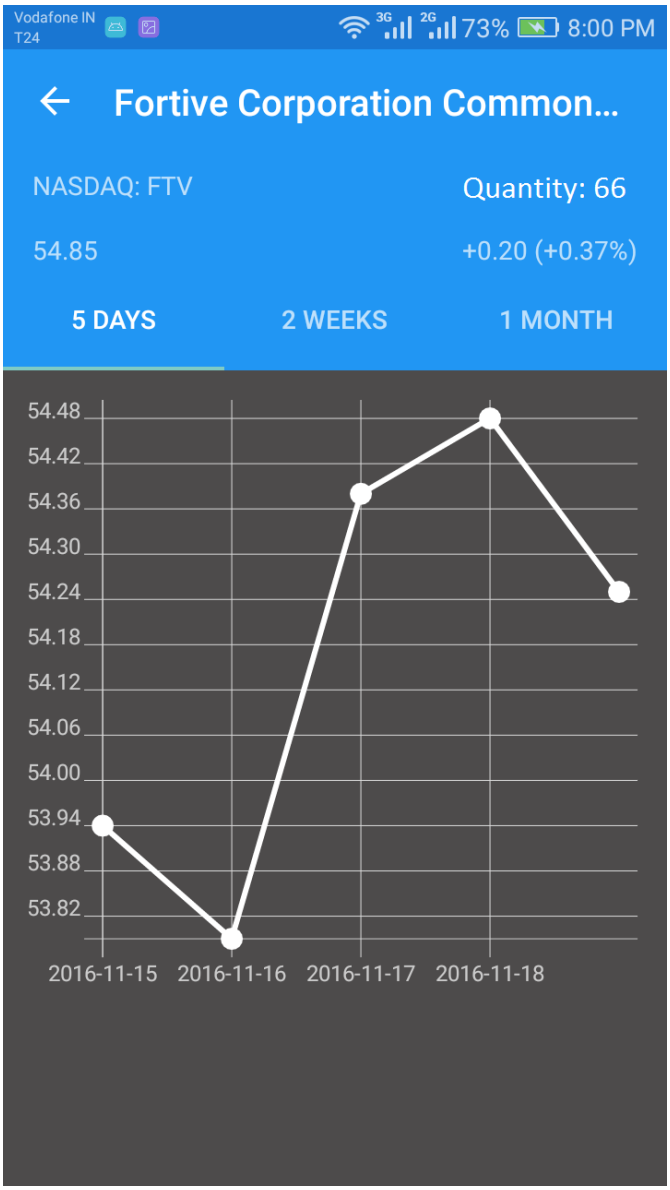
Screen 2


Vodafone IN
T24
73%
8:00 PM
MY STOCK HEALTH
$
Name
Quantity
Cost
Variation
FTV
66
52.35
54.85
+0.20
FIT
50
9.20
8.66
-0.06
DHR
77
80.56
77.99
+0.23
PFE
54
30.50
31.50
+0.09
GE
50
29.50
30.84
+0.20
+
EXPENSE
13240.22
MARKET VALUE
13301.33

Screen 3


MY STOCK HEALTH
Name
Quantity
Cost
Variation
Add a new stock
Stock Symbol
Quantity
Purchased Price
CANCEL
ADD
PFE 54 30.50 31.50
GE 50 29.50 30.84 +0.20

Screen 4



Key Considerations

How will your app handle data persistence?

The app will be having both online and offline storage.
The stock details will be saved in a locally and will be shared using content providers.

The stock name, quantity purchased and amount will be stored in firebase database and can be retrieved on any device with user login.

Describe any corner cases in the UX.

The login procedure of the user will be accomplished using Firebase UI.
Handle back button implementation to avoid flickering.
Add sign out to allow user to sign out temporarily.

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

firebase-database: To use firebase real time database.
firebase-auth : To use firebase authentication.
firebase-ui-auth: To use UI authentication solution
retrofit: Retrofit is a REST Client for Android and Java by Square. Retrofit can be configured with converter is used for its data serialization.
butterknife: Eliminate findViewById calls by using @BindView on fields.
recyclerview-v7: To show the stocklist in a more optimal view than ListView.
lecho:hellocharts-library: To show charts.

Describe how you will implement Google Play Services.

I will use google play service to perform Google Cloud Messaging and for Google+ login.

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

The first step after creating the base Android Studio project will be to set up a firebase libraries.
This will include:

- Adding project in the Firebase Console
- Enable email and google login in authentication.
- Generate google-services.json.
- I will also begin designing firebase data model and rules.

Task 2: Implement UI for Each Activity and Fragment

In this step, basic layout will be drawn using recyclerview and containers.

- Create home screen and set data using dummy values
- Add sample stocks.
- Fetch Data from Yahoo API
- Add a SyncAdapter to fetch stock related news periodically and show in UI.

Task 3: Add Firebase Dependencies

In this step, add firebase dependencies to the app

- Add firebase dependencies to build.gradle
- Implement Google Play Services
- Validate login
- Handle Error Cases

Task 4: Write to Firebase real-time database

In this step, I will implement read and write to firebase real-time database.
Users stocks, purchased quantity and spent amount will be stored in Firebase database.

Task 5: Test and generate a signed apk

At this point, I will test and optimize this app to find any missing/useful features.
Also I will generate a signed apk for final submission.