Description

Intended User

Features

User Interface Mocks

Key Considerations

- How will your app handle data persistence?
- Describe any edge or 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 or other external services.

Next Steps: Required Tasks

- Task 1: Project Setup
- Task 2: Set up interaction between my system and Firebase.
- Task 3: Implement UI for Each Activity and Fragment
- Task 4: Test
- Task 5 (optional): Implement Chart Sharing

Open questions:

GitHub Username: **alex01001**

Zone Trading Alerts

Description

Project background:
Some time ago I've created a system which monitors the stock market and sends alerts when it identifies a moment when stock price is about to start growing (you can read more at http://stocksbuyalerts.com/). The system is based on the theory of so-called supply and demand zones. My software recognizes demand zones, assesses the probability of the price growth, and notifies the subscribers. The target audience for these alerts is day traders (people who actively buy and sell stocks within a day), so it is very important that subscribers receive alerts quickly. Currently, the system sends e-mail alerts only. Sometimes, e-mail delivery is delayed by e-mail servers and my subscribers receive alerts too late.

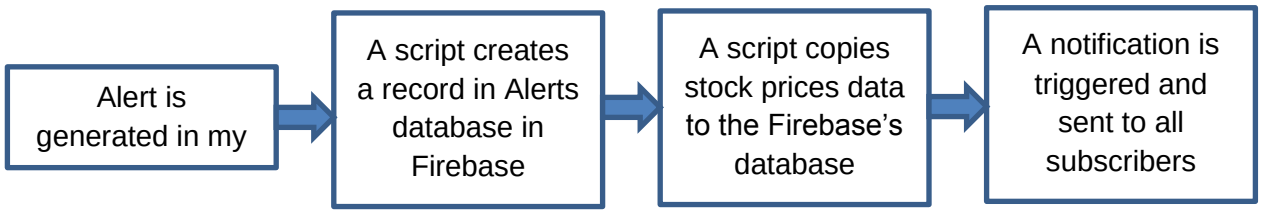
Intended User

The target audience for these alerts is day traders (people who actively buy and sell stocks within a day).

Features

- Receive the instant alerts (notification) to all subscribed users. Alerts will be generated in Firebase.
- Display the chart and key parameters for the stock for which alter is sent.
- Show historical charts for previous days (one month back).
- Enable a user to share successful trades in his/her social network (optional[1])

Process Flow:
The sequences of events when alert is generated:



When a user clicks on the alert, the app should make a query to the Firebase's database and copy all data related to the Alert (and store it in the app's local database). This information is static and won't be changed.

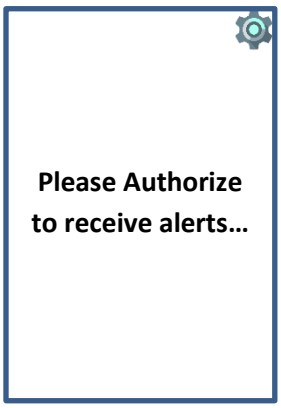
User Interface Mocks

1. Notification on home Screen

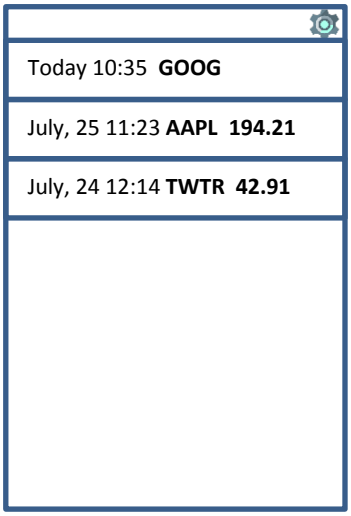
[1] I'll take care of it if I have time

2. Main Activity

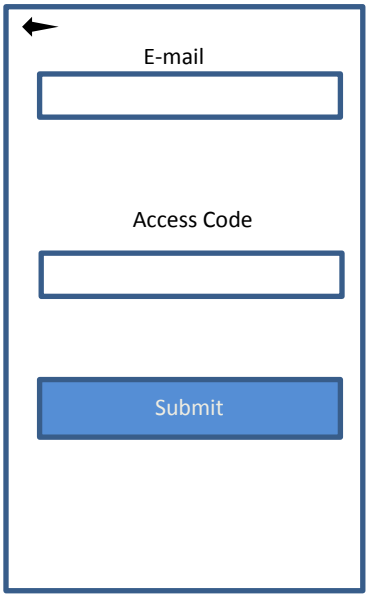
2a. User is not authorized.



2b. User is authorized.



3. User's authorization:



4. Stock chart view (always in landscape)

Key Considerations

How will your app handle data persistence?

I plan to use a firebase database.

Database structure:
The database will include two tables: TickerData and Alerts.
Tables structure:
TickerData (table which contains stock prices data with the resolution of one minute):

- Symbol – String
- Time – Timestamp
- Open – decimal (8,4)
- Close - decimal (8,4)
- High - decimal (8,4)
- Low - decimal (8,4)
- Volume – integer

Alerts:

- Symbol – String
- AlertTime – Timestamp
- ZoneTime – Timestamp
- ZoneTop - decimal (8,4)
- ZoneBottom - decimal (8,4)
- AlertPrice - decimal (8,4)

Describe any edge or corner cases in the UX.

The user will navigate back by clicking the “back button” on the toolbar.

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

I plan to use TradingView library to plot charts (https://www.tradingview.com/HTML5-stock-forex-bitcoin-charting-library/).

Describe how you will implement Google Play Services or other external services.

I’ll use Firebase for authentication, notifications, and database access.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Create the project in Android Studio.
- Configure the stock charting library and experiment with it.
- Configure firebase account.

Task 2: Set up interaction between my system and Firebase.

- Build a script to copy data from my database to the Firebase database.
- Build a notification-triggering script.

Task 3: Implement UI for Each Activity and Fragment

- Implement Main Activity
- Implement User Authorization Activity
- Implement and test the notification process.
- Implement the charting library (including reading data from the Firebase)

Task 4: Test

- Test the notifications delivery
- Test charting
- Check data accuracy

Task 5 (optional): Implement Chart Sharing

Implement functionality to share stock charts on users' social networks.

Open questions:

I currently don't know how to implement these features. Any guidance would be helpful.

- How to programmatically generate notifications in Firebase (from my own server)?
- How to programmatically populate the Firebase's database with stock prices data? I have data on my server and can build the software to connect to Firebase. Just need some guidance/education on how to do it best.
- How to authorize the subscribers on Firebase. All I know about the subscriber for now is just his name and e-mail. They don't need any sort of password right now. However, I'll need to restrict access to my Firebase database.
  The way I see it: I'll generate some sort of access code for each subscriber and it will become their password. I'll communicate this access code to each subscriber and set up the user accounts in the Firebase.
  Questions are:
  - Is it an appropriate way to organize users authorization?
  - Is there any API to automate the process of creation user account in Firebase? Imagine, I have 50 subscribers and for each of them I generate a pair <e-mail, access code>. Then I'll need to make some sort of a script which will copy those 50 records to database.