**How to Use this Template**

1. Create a new document, and copy and paste the text from this template into your new document [ Select All → Copy → Paste into new document ]
2. Name your document file: “**Capstone\_Stage1**”
3. Replace the text in green

[Description](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.sm4ra97uwo11)

[Intended User](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.aws88pzfmqca)

[Features](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.zheq5430xrpq)

[User Interface Mocks](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.giquerrw6g46)

[Screen 1](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.a4jdupabry3k)

[Screen 2](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.dpcbbkx5yry)

[Key Considerations](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.gvcvmae8jn8u)

[How will your app handle data persistence?](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.v8my7nhtvz0m)

[Describe any corner cases in the UX.](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.gw69vjn1ico0)

[Describe any libraries you’ll be using and share your reasoning for including them.](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.6yqqubmw5bs)

[Describe how you will implement Google Play Services.](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.qrxg682nywe6)

[Next Steps: Required Tasks](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.v518bncmggeg)

[Task 1: Project Setup](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.8oe8zpk3qsmp)

[Task 2: Implement UI for Each Activity and Fragment](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.rzllsk6uqztx)

[Task 3: Your Next Task](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.fdmohs7hes)

[Task 4: Your Next Task](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.umfwsvmx7tpn)

[Task 5: Your Next Task](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.kjidlkq4xm3u)

**GitHub Username**: FlixKo

FinanceOverview

Description

This application enables you to search for stocks you own. You may enter the amount of stocks you own and get an overview of your personal portfolio with its current value.

Intended User

Stock owners

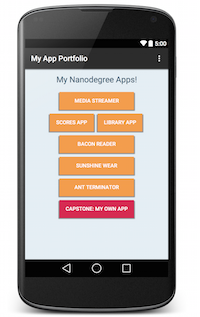
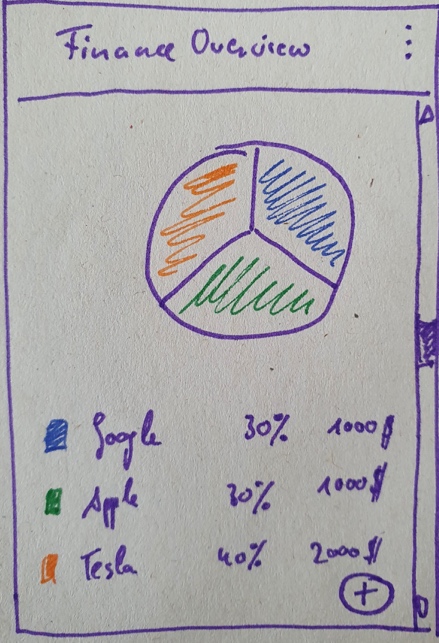
Features

* Search for a stock using its name
* Enter the amount you own of that stock
* Display the value of the stock you own
* Display the value of the whole portfolio
* Project is written solely in the Java Programming Language

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 Google Drawings, [www.ninjamock.com](http://www.ninjamock.com), Paper by 53, Photoshop or Balsamiq.

**Screen 1**

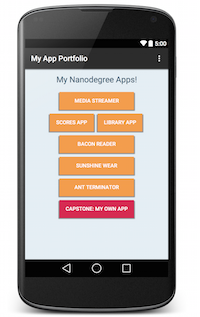
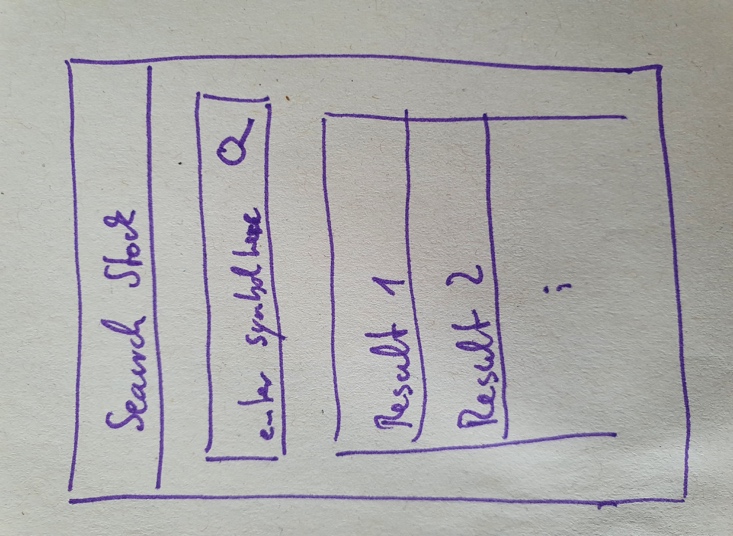


Overview of finance data. Ideally with a pie chart, not sure if it’ll be part of it though. It shows the stocks, their percentage within the portfolio and their current value (stock price \* number of stocks owned)

The preferences in the top are not needed for now (won’t be part of the final app)

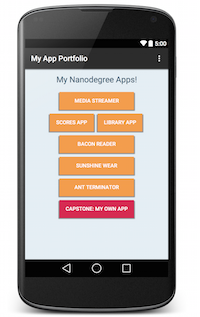
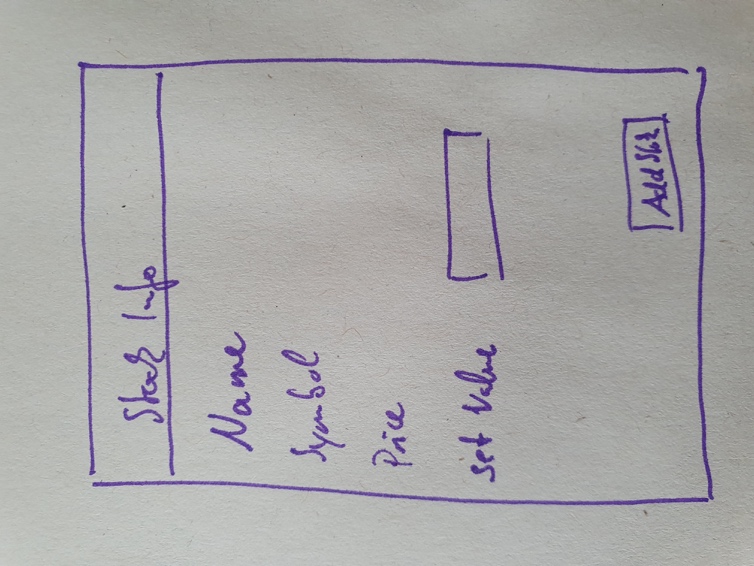
The + button lets the user add new stocks (via a search sreen)

Each Stock is clickable to display a details view. Details view will be reused from the search results screen (see later). The button in that screen will not be called “add stock” but “save” (to allow the user to change the number of shares he owns of that particular stock)

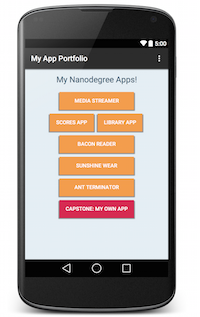
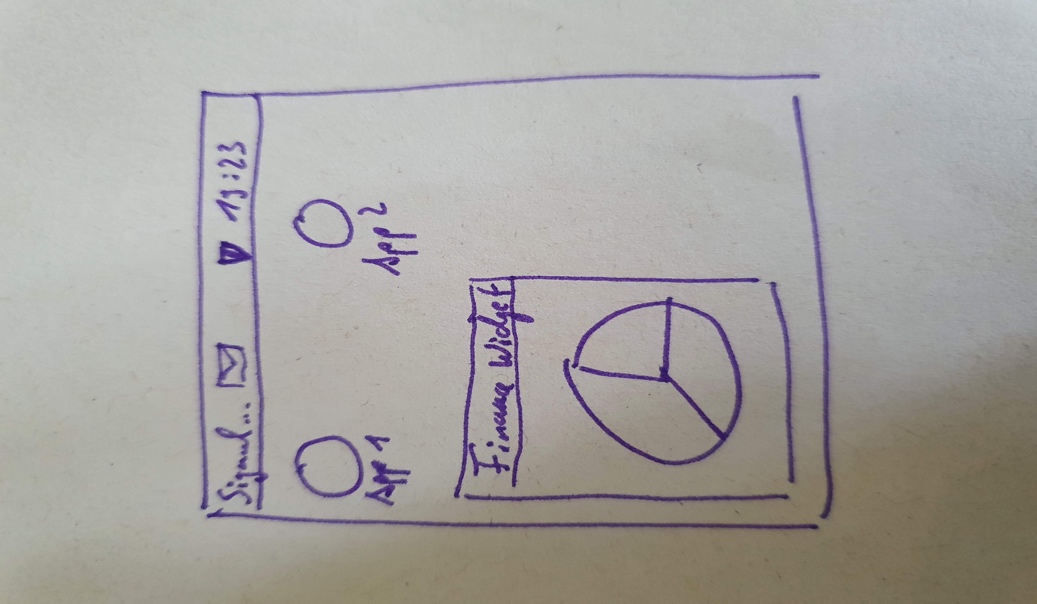


Search for a stock using its name or symbol.

Display the results as a clickable list.



Details view of stocks. User has to enter a value for number of stocks owned to add it to the list. When stock gets added, the list on the main screen gets updated, as well as the pio chart.



Widget either displays the pie chart or the list of stocks

Key Considerations

App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.

App includes support for accessibility. That includes content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues.

**How will your app handle data persistence?**

Store the amount of stocks the user has in a rooms database as key value

[“stock\_symbol”, “stock\_amount”]

**Describe any edge or corner cases in the UX.**

Update the portfolio after the user has entered the data in the preferences section

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

|  |  |  |
| --- | --- | --- |
| **Library / Tool** | **Version** | **Reason** |
| Butterknife | 7.0.1 | make UI-Binding easier |
| achartengine | 1.0.0 | draw the pie chart |
| Android Studio | 4.0 | IDE |
| Gradle | 4.0.0 | Building |
| Firebase Crashlytics | 17.0.0 | Crash reports |
| Google Analytics | 17.4.4 | App usage |

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

Fetch data from the Alphavantage API: <https://www.alphavantage.co/documentation/>

Using AsyncTasks

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**

* Setup Github Repository
* Create new Android Studio Project
* Connect to Alphavantage API
* Create JSONUtils to extract the relevant Stock Data
* Create Data Objects for the Stocks
* Disply the data in simple TextView, to ensure the API works as expected

**Task 2: Implement UI for Each Activity and Fragment**

* Build MainActivity Screen with portfolio overview
* Build Search Screen
* Build Stock Details View Screen

**Task 3: Handle Error Cases**

* Handle Error Cases
  + No internet connection
  + No initial value in portfolio
  + No search results found
  + No shares number inserted on details sreen

**Task 4: Improve UI**

* Include Material Design Effects