AppleStocks

A Simple Application for Apple Stock Quotes Monitoring

Henrique Romão
up202108067@up.pt

Mariana Bessa
up202107946@up.pt

November 28, 2024

Abstract

To be written at the end.

Contents

1 Introduction

This application was developed to provide users with a simple and intuitive way to view Apple stock quotes for a selected recent period. Users can customize their experience by choosing the time interval, in days or hours, and the duration of the interval, ranging from 2 to 10.

2 Use Cases

The application supports the following use cases, which are represented in the use case diagram, in Figure 1.

- Choose View Options - where the user selects a time interval (Hourly or Daily) and a range (2–10).
- Get Stock Information - where the application connects with the Web API.
- See Stocks - when the user pushes the "Submit" button and sees the Plot Activity.

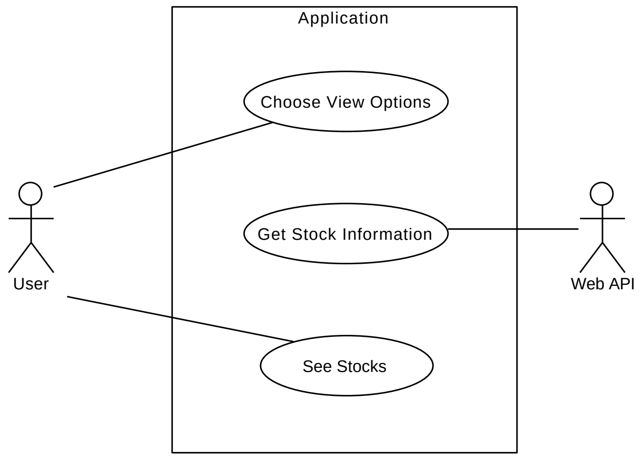


Figure 1: Use Cases Diagram.

3 Architecture

To make the application more user-friendly and less crowded, the application was divided into two activities: the Main Activity, where the user configures their preferences, and the Plot Activity, where the results are displayed.

3.1 Action Bar

Common to both activities, there is an action bar displaying the name of the app.

3.2 Main Activity

This activity also includes two **Spinner** objects: one for selecting the type of interval (hours or days) and another for specifying the interval range. These spinners are initialized in the Main Activity code using the `new ArrayAdapter<>()` method, with `setDropDownViewResource()` used to define their layout.

Both spinners have default values: "H" (hours) for the type of interval and "2" for the range. This means that, unless changed by the user, the stock quotes for the last two hours will be displayed by default.

A button is provided to allow the user to proceed, with its listener being implemented within the Main Activity. The `onClick()` method creates an **Intent** to open the Plot Activity, passing the user's spinner selections (as a string and an integer) to the new activity.

3.3 Plot Activity

In the plot activity is where, according to the choices made on the Main Activity, the HTTPS request is made, retrieving the stock information from the web API, and where the request is processed, displaying the information to the user.

3.3.1 HTTP Request

The first step is to read the intent passed from the previous activity, retrieving the time interval type and total number. With these two variables, it is possible to craft the URL, which will be used to do the HTTP request.

The URL for the HTTP request, is passed through a function that initiates a new thread where the rest of the processing will take place. Inside the thead, the processing is initiated by the `FetchData` private class, that takes a URL and establishes the HTTP connection. The connection status is assessed, and if it is succesfull, the processing will go on, otherwise, an error message is thrown.

3.3.2 Data Processing

Before the actual processing begins, the input stream received from the HTTP connection is converted into type `String`. With this `String`, function `processJsonData` will create a JSON object retrieve the closing quote prices and respective time. Redarding the time, it is necessary to convert the dates into timestamps, for the plot function, and so it is necessary to format the dates. A problem that arises is that, according to the choice of days or hours, the date format will be different, as it will include the hour and timezone in the last case. To deal with the different formats, according to the selected time interval type, passed through the intent, a different format is used for pasing the date strings into date objects, which are then converted to timestamps.

3.3.3 Data Plotting

This processed information is then passed to the `plotData` function. The data is plotted using the LineData widget, part of the MPandroid Chart library, [1]. This widget was choosen to allow for faster app development, taking use of already developed code, and considering its widespread use, it seemed a correct fit for the problem. For this widget the data is passed in an array of entries, and so the function joing the two arrays into a single array of entries with two values, one entrie for each dot in the plot. Additionally, the function also computes the maximum and minimum closing quote prices, and creates an entry array for each, to be plotted as horizontal lines delimiting the maximum and minimum value. Then, each entry array is properly formatted, and sent to the widget to be plotted. It is also necessary to format the x axis, otherwise the values seen would be the timestamps, which are difficult to interpret and not user-friendly. This way, the value is formatted to return a simple date string, and only the first and last values are shown, for a simpler graph analysis. The maximum and minimum values are also shown outside the plot.

4 Interface

The application has a black background with purple accents, as these darker colors are more associated with the stock market. The theme used was `Theme.AppCompat.Light.DarkActionBar`. It is structured using `LinearLayout` to arrange the `TextView` elements, `Spinner` objects, and `Submit` button in vertical and horizontal sequences. When the `Submit` button is pressed, the application starts the `Plot Activity` using the `startActivity()` method, passing the selected user inputs as extras.

4.1 Main Activity

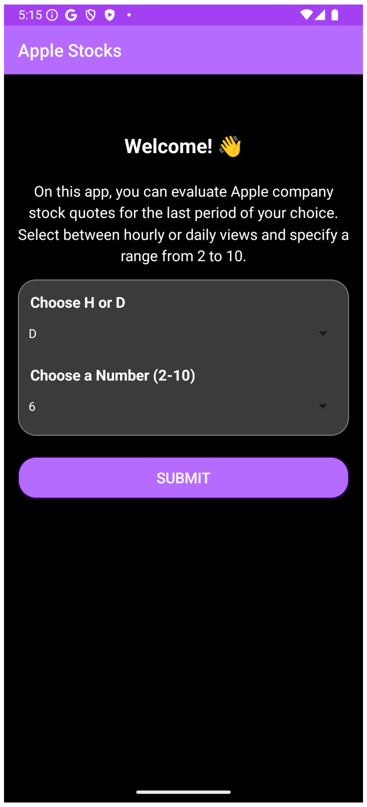
This activity is design to adapt to both horizontal (landscape) and vertical (portrait) screen orientations, maintaining a consistent and user-friendly layout.

4.1.1 Portrait

Upon opening the application, the user is welcomed by an action bar and two `TextView` elements, where one displays a greeting, and the other provides instructions on how to use the app, as it is shown in Figure 2a. Below these, additional `TextView` elements explain the purpose of each `Spinner` object, which are used for input selection.

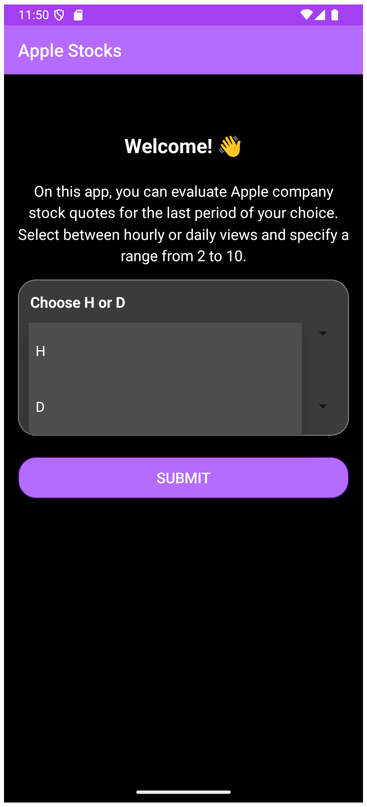
The layout for the spinner items and their drop-down lists was custom-designed in the XML files `/layout/spinner_dropdown_iten` which defines the appearance of the spinner drop-down items, and `/layout/spinner_item.xml`, which defines the appearance of the selected spinner items (Figures 2b and 2c). The spinner values are defined in the `strings.xml` file located in `/res/values/`.

Additionally, custom shapes were created to style the `Submit` button and the background of the spinner section. These shapes were implemented in the `/drawable/submit_button.xml` file, which defines the shape of the Submit button, and the `/drawable/round_button.xml` file, which shapes the background of the selection area.


Welcome! 👋

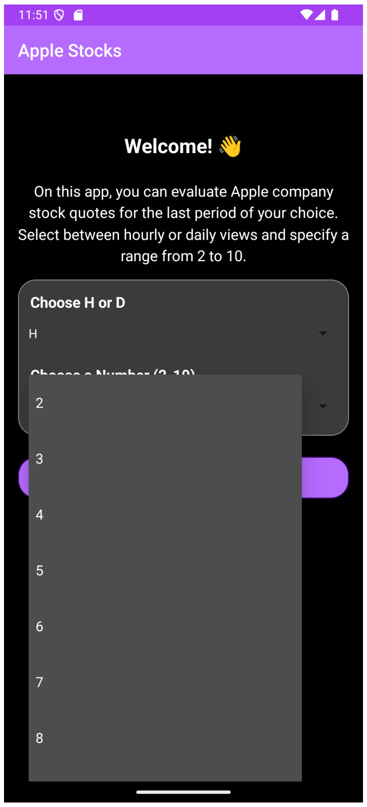
On this app, you can evaluate Apple company stock quotes for the last period of your choice. Select between hourly or daily views and specify a range from 2 to 10.

(a) Main Activity.


Welcome! 👋

On this app, you can evaluate Apple company stock quotes for the last period of your choice. Select between hourly or daily views and specify a range from 2 to 10.

(b) Interval Choice.


Welcome! 👋

On this app, you can evaluate Apple company stock quotes for the last period of your choice. Select between hourly or daily views and specify a range from 2 to 10.

(c) Range Choice.

Figure 2: Main Activity in Portrait.

4.1.2 Landscape

In the landscape orientation, a dedicated layout file (`/layout-land`) was created to optimize the user interface for horizontal screens.

This layout maintains the key aspects of the portrait view, with the main adjustment being the rearrangement of the `Spinner` objects: instead of being vertically aligned, they are displayed side by side (horizontally). This design enhances usability by taking full advantage of the horizontal screen space.

The layout is shown in Figure 3a, including the spinner options for interval and range choices (Figures 3b and 3c).

4.2 Plot Activity

The interface for the plot activity can be seen in Figure 4, with the activity still loading the data (Figure 4a) and with all the data loaded (Figure 4b). The activity takes a short time loading, but this wait is due to slow connectivity of the web API, having tested the speed for the same request in our personal computers and noticing a slight delay.

To have an indication of chosen time intervals, the number of time intervals and time interval type are written in the plot title.

4.2.1 Plot

In Figure 4b it is possible to see how the data is plotted, in a simple and clean manner, using a color with a good contrast. The plot is filled with a gradient to ease data visualization, when compared to having no fill, but it is not tiring to the eyes, as a solid colorful fill would be. As mentioned, to avoid an excess of information, only the first and last x axis values are written, with the title already providing enough information for the correct understanding of the graph.

Below the plot, the maximum and minimum close values can be seen, ensuring an exact reading, when compared to a reading done from the plot. It is also possible to note the mentioned maximum and minimum bars, in a grey dashed line, that help visualizing the results in a more graphical manner.

4.2.2 Loading Messages

Due to the noticed delay, and in hopes of giving some feedback to the user, the loading messages were added, acting as placeholders for the plot and *extrema* values. The color choosen was yellow, as it is costumary to exemplify warning and loading messages, and the three dots also help to signify the loading process is temporary, making the app more responsive.

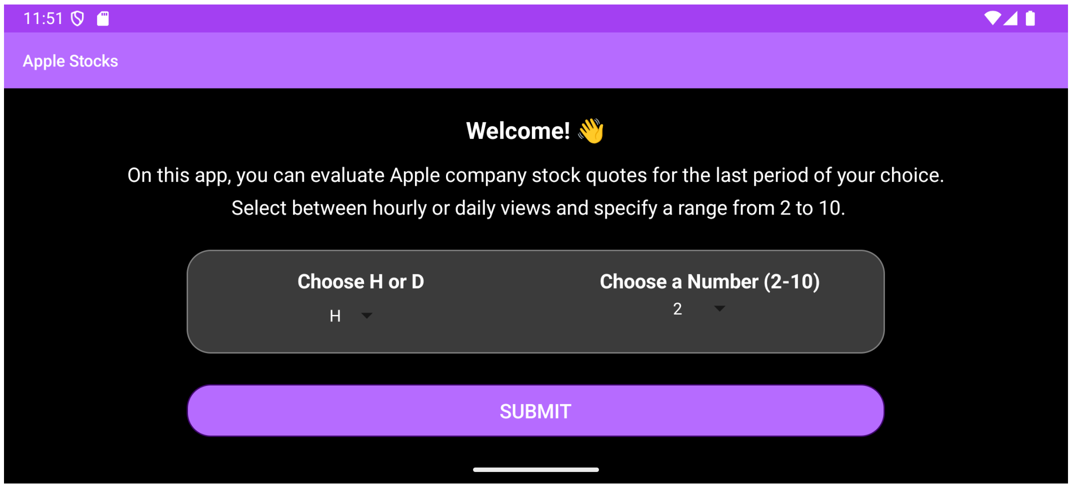
4.2.3 Landscape

5 User Experience

The application prioritizes simplicity and ease of use. The Main Activity's clean layout and clear instructions ensure users can quickly configure their preferences, while the Plot Activity delivers an engaging and informative visual representation of stock data.

References

[1] Phil Jay. Mpandroidchart. `https://github.com/PhilJay/MPAndroidChart`, 2020. Accessed: 2024-11-27.



11:51

Apple Stocks

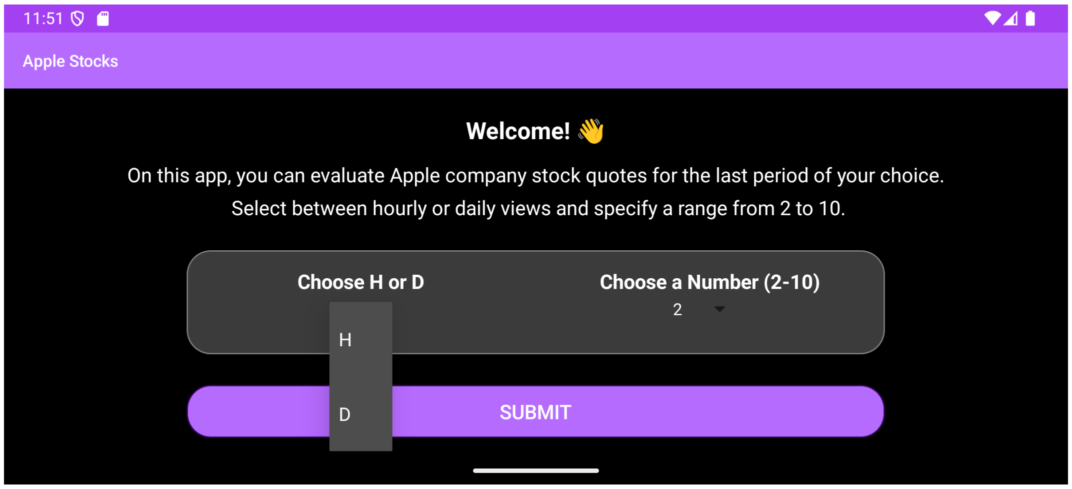
Welcome! 👋

On this app, you can evaluate Apple company stock quotes for the last period of your choice. Select between hourly or daily views and specify a range from 2 to 10.

Choose H or D: H

Choose a Number (2-10): 2

(a) Main Activity.



11:51

Apple Stocks

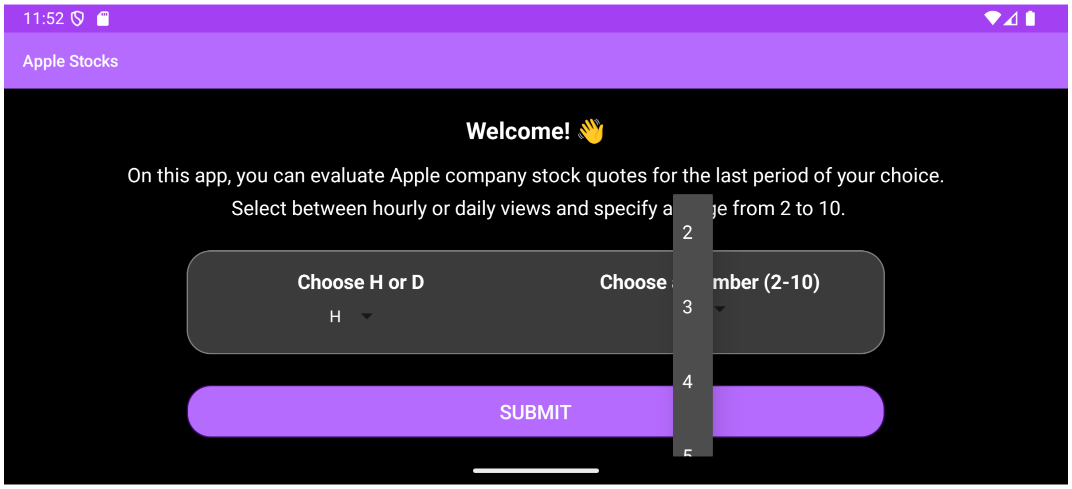
Welcome! 👋

On this app, you can evaluate Apple company stock quotes for the last period of your choice. Select between hourly or daily views and specify a range from 2 to 10.

Choose H or D

Choose a Number (2-10): 2

(b) Interval Choice.



11:52

Apple Stocks

Welcome! 👋

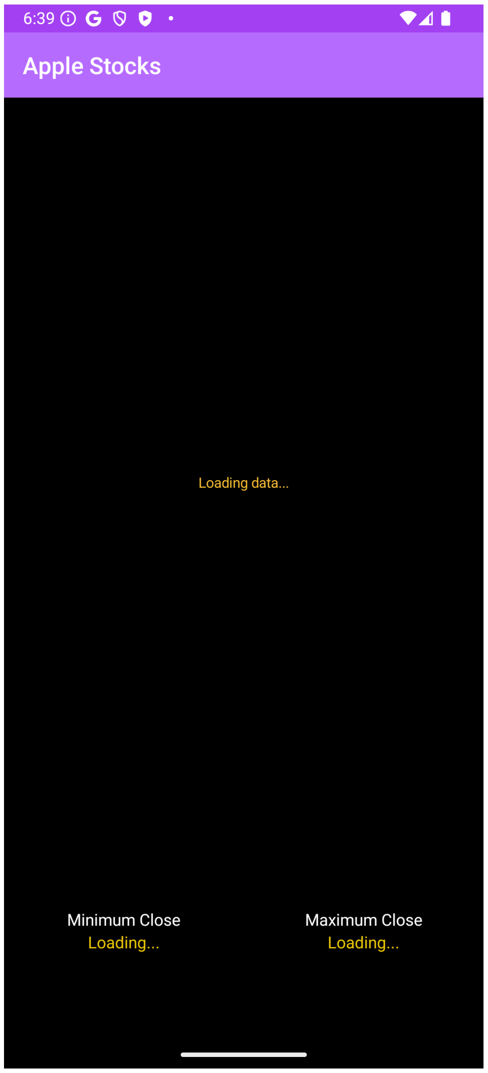
On this app, you can evaluate Apple company stock quotes for the last period of your choice. Select between hourly or daily views and specify a ge from 2 to 10.

Choose H or D: H

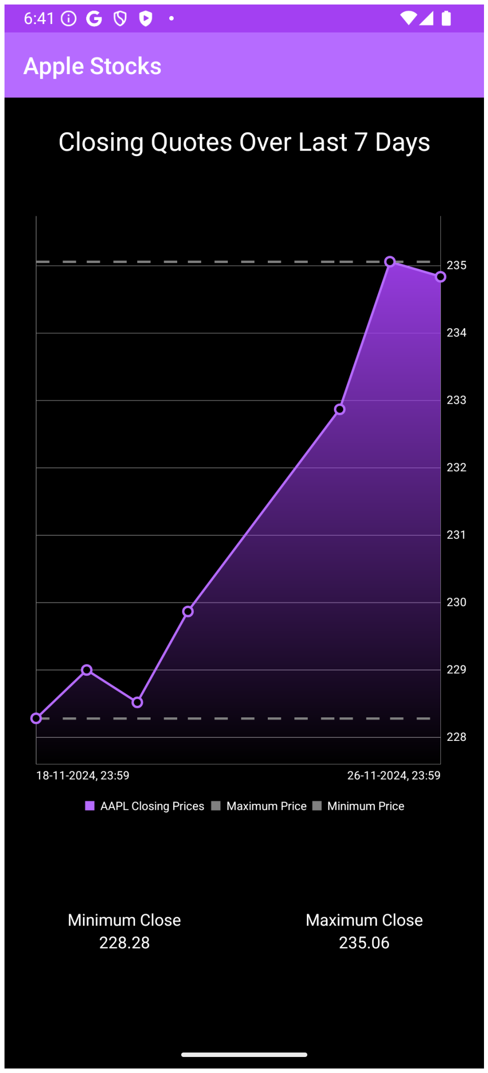
Choose mber (2-10)

(c) Range Choice.

Figure 3: Main Activity in Landscape.



(a) While Loading.



(b) Loaded.

Figure 4: Plot Activity.