Homework 9: Stock Search Android App with Facebook Post – A Mobile Phone Exercise

1. Objectives

- ➢ Become familiar with Android Studio, Android App development and Facebook SDK for Android.
- ➢ Build a good-looking Android app using the Android SDK.
- ➢ Add social networking features using the Facebook SDK.

2. Background

2.1 Android Studio

Android Studio is the official IDE for Android application development, based on IntelliJ IDEA (https://www.jetbrains.com/idea/). On top of the capabilities you expect from IntelliJ, Android Studio offers:

- Flexible Gradle-based build system
- Build variants and multiple apk file generation
- Code templates to help you build common app features
- Rich layout editor with support for drag and drop theme editing
- Lint tools to catch performance, usability, version compatibility, and other problems
- ProGuard and app-signing capabilities
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

The home page of the Android Studio is located at:
http://developer.android.com/tools/studio/index.html

2.2. Android

Android is a mobile operating system initially developed by Android Inc., a firm purchased by Google in 2005. Android is based upon a modified version of the Linux kernel. As of December 2013, Android was the number 1 mobile OS, in unit sales, surpassing iOS, while iOS was still the most profitable platform.

The Android operating system software stack consists of Java applications running on a Java based object oriented application framework on top of Java core libraries running on the Dalvik virtual machine featuring JIT compilation.

The Official Android home page is located at:

http://www.android.com/

The Official Android Developer home page is located at:

http://developer.android.com/index.html

2.3 Facebook

Facebook is a social networking service launched in February 2004, owned and operated by Facebook, Inc. Users can add friends and send them messages, and update their personal profiles to notify friends about themselves and what they are doing.

Users can additionally post news feeds to their profiles, and these feeds may include images, besides text messages.

The Facebook homepage is available at:

http://www.facebook.com

Facebook provides developers with an application-programming interface, called the Facebook Platform.

2.4 Markit on Demand

Markit on Demand API provides detailed description about the stock information of company as well as historical stock values. You can refer to the API description on the following link:

http://dev.markitondemand.com/MODApis/

2.5 Amazon Web Services (AWS)

AWS is Amazon's implementation of cloud computing. Included in AWS is Amazon Elastic Compute Cloud (EC2), which delivers scalable, pay-as-you-go compute capacity in the cloud, and AWS Elastic Beanstalk, an even easier way to quickly deploy and manage applications in the AWS cloud. You simply upload your application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring. Elastic Beanstalk is built using familiar software stacks such as the Apache HTTP Server, PHP, and Python, Passenger for Ruby, IIS 7.5 for .NET, and Apache Tomcat for Java.

The Amazon Web Services homepage is available at: http://aws.amazon.com/

2.6 Google App Engine (GAE)

Google App Engine applications are easy to create, easy to maintain, and easy to scale as your traffic and data storage needs change. With App Engine, there are no servers to maintain. You simply upload your application and it's ready to go. App Engine applications automatically scale based on incoming traffic. Load balancing, micro services, authorization, SQL and NoSQL databases, memcache, traffic splitting, logging, search, versioning, roll out and roll backs, and security scanning are all supported natively and are highly customizable.

To learn more about GAE support for PHP visit the page:

https://cloud.google.com/appengine/docs/php/

3. Prerequisites

This homework requires the use of the following components:

1. This homework requires the use of the following components:

   A. Download and install Android Studio. You may use any other IDE other than Android Studio such as Eclipse, but you will be on your own if problems spring up.

   B. First you need to install Java on your local machine. You can download JDK 8 from - http://www.oracle.com/technetwork/java/javase/downloads/index.html. For windows users, after installing the JDK, you need to add environment variables for JDK.

      - Properties -> Advanced -> Environment Variables -> System variables -> New Variable

        Name: JAVA_HOME, Variable Value: <Full path to the JDK>

      - Typically, this full path looks something like C:\Program Files\Java\jdk1.8.0.

        Then modify the PATH variable as follows on Microsoft Windows:
        C:\WINDOWS\system32;C:\WINDOWS;C:\Program Files\Java\jdk1.8.0\bin
        This path may vary depending on your installation.

      - Note: The PATH environment variable is a series of directories separated by semicolons (;) and is not case-sensitive. Microsoft Windows looks for programs in the PATH directories in order, from left to right. You should only have one bin directory for a JDK in the path at a time. Those following the first instance are ignored. If you are not sure where to add the path, add it to the right of the value of the PATH variable. The new path takes effect in each new command window you open after setting the PATH variable.

      - Reboot your computer and type “java -version” in the terminal to see whether your JDK has been installed correctly.

   Set up the Android Studio environment so that you can run any sample android app on your phone/tablet/virtual device from it. Then you can start with this homework app. You will need to enable “Developer Options” and “USB debugging” if you are using an actual device. There are endless resources a simple search away on how to setup your Android Studio.

2. You also need to create a Facebook Developer application as you did for your homework 8. Follow the following steps to get started:

    a. **Download SDK**: Download the latest Facebook Android SDK Link: https://developers.facebook.com/docs/android

    b. Instructions to **import in Android Studio**: https://developers.facebook.com/docs/android/getting-started

    c. Create a **new app on Facebook** developer: https://developers.facebook.com/apps/

    d. Specify **App Info** related to the HW9 android application you are developing.

    e. **Key Hashes**: Specify Android key hash for the development environment using the commands mentioned.

    f. Track App Installs and App Opens: Not required.

    g. Next Steps: Utilize **Login** (optional) and **Share** tutorials to achieve the functionality required for the exercise. Note: In your Facebook application settings, you should go to the "Status & Review" section and choose "Yes" for the question "*Do you want to make this app and all its live features available to the general public*?" as you did for homework 8.

4. High Level Design

In this exercise, you will develop an Android Mobile application, which will have following functionality:

An Auto-complete edit box is provided to enter the stock name or symbol. The user can then select a stock from the suggestions. This feature would be developed using the Android's 'AutoCompleteTextView' component. Details on how to use the API would be covered in Section 5.

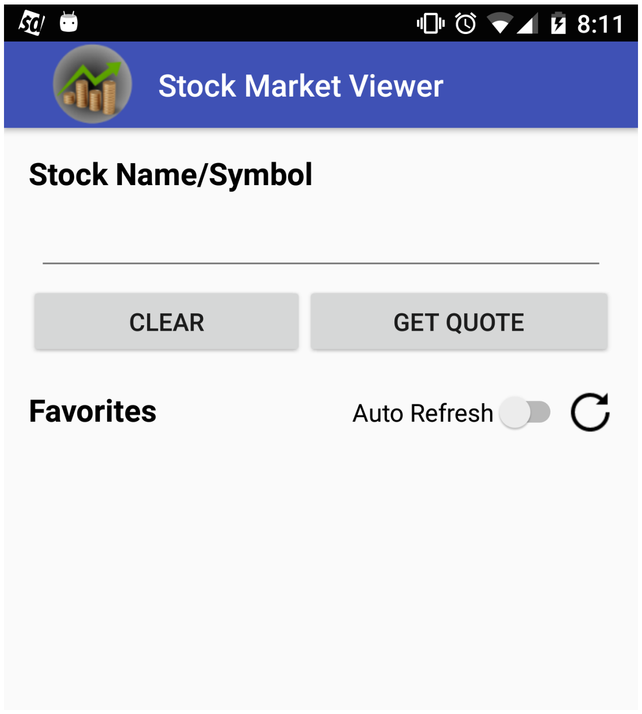


Figure 1. Search Form

Once the user has provided data and selected a result from the autocomplete list he would click on 'Get Quote', when validation must be done to check that the entered data is valid.

Once the validation is successful, we would get the stock details using our PHP script hosted on Amazon Web Services/Google App Engine, which would return the result in JSON format. We would display the stock details in a ListView component in the 'Current' tab. Furthermore, our PHP script would be responsible for rendering the HighCharts in the 'Historical' tab and also rending the news articles in the 'News' tab.

5. Implementation

5.1 Search Form

You must replicate the Search Form, as shown in Figure 1. The field names remain the same as in Homework 8.

The interface consists of the following:

- An 'AutoCompleteTextView' component allowing the user to enter the company name or symbol.
- Two buttons for interaction in the Search Form.
  - A button 'CLEAR' to clear the 'AutoCompleteTextView' component.
  - A button 'GET QUOITE' to get the quote, after validation.
- The Favorite ListView showing the list of favorite stocks.

- The Favorite List starts with an empty favorite list.

The form has two buttons:

a) GET QUOTE: Validations are first performed, when the button is clicked. If the validations are successful, then the stock details would be fetched from the server (either hosted on Google App Engine or Amazon Web Services). However, if the validations are unsuccessful, appropriate messages would be displayed and no further requests would be made to the server.
b) CLEAR: This button would clear the 'AutoCompleteTextView' and also clear any validations error, if present.

5.1.1 AutoComplete

The user can enter the stock name or symbol in the text view to get the stock information from our PHP script. Based on the user input, the AutoComplete would display the all the matching companies and symbols (see Figure 2) by making a HTTP request. The requests would be made only when the user enters a minimum of 3 characters to avoid unnecessary network calls. This needs to be implemented using AutoCompleteTextView.

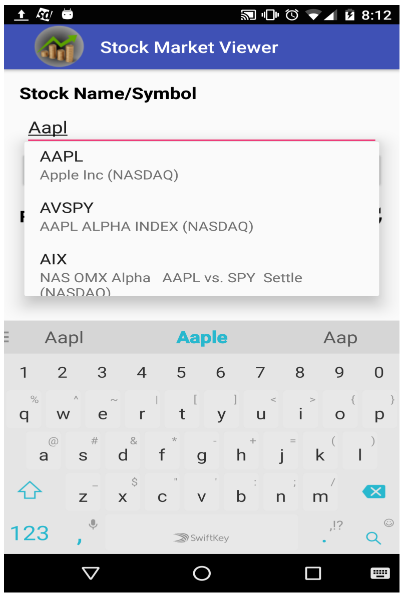
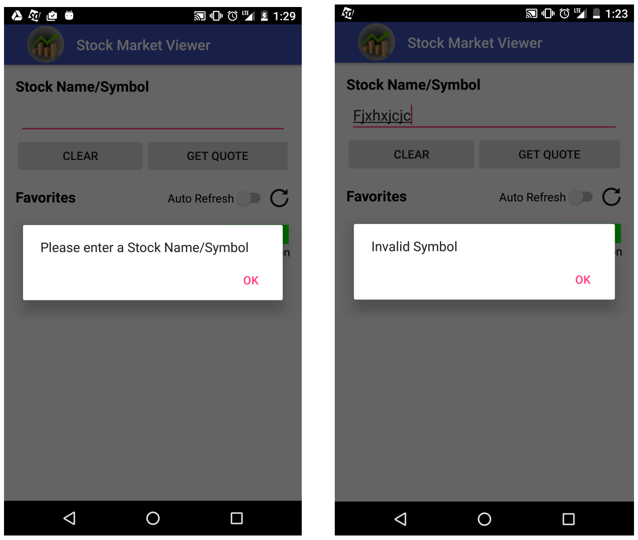


Figure 2: AutoComplete Suggestions

5.1.2 Validations

The following validations need to be implemented:

- Empty Entry: If the user does not enter anything in the AutoComplete component, you need to display an appropriate message to indicate the same.
- Invalid Selection: If the user enters an invalid entry which does not correspond to any valid stock symbol, you need to display an appropriate message to indicate the same.



Please enter a Stock Name/Symbol

Invalid Symbol

Figure 3: Validations

Once if the user input is successfully validated, the API calls should be made, otherwise appropriate messages should be displayed.

5.1.3 Get Quote Execution

Once the validation is successful, you should execute an HTTP request transaction to the PHP script which is located in the Google App Engine/Amazon Web Services.

The PHP script on Google App Engine/Amazon Web Services, modified after Homework 6, is used to retrieve data from Markit on Demand. You should pass the company symbol as a parameter of the transaction when calling the PHP script.

For example, if your Google App Engine/Amazon Web Services service is located at

http://example.appspot.com

and the user enters 'AAPL' as the company symbol, then a query of the following type needs to be generated:

http://example.appspot.com/?symbol=AAPL

The PHP script running on the Google App Engine/Amazon Web Services would extract the stock details of the company symbol, perform an API request to Markit on Demand, and returns the data in JSON data.

After obtaining the query results from the callback of the AJAX request, we would be displaying the results in a drop-down suggestion list.

Notice: In Homework 6 your PHP script produced HTML. In this exercise, the output must be changed to JSON and the PHP code must run on Google App Engine/Amazon Web Services.

5.2 Favorite List

The Favorite stocks would be displayed in a list as per Figure 4. The data for the favorite list should be loaded from 'SharedPreferences'. For more about shared preferences please refer to the appendix.

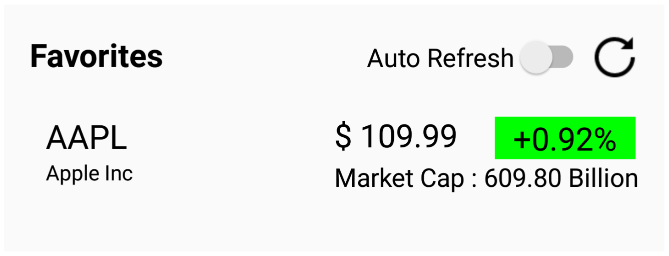


Figure 4: Favorite Stocks

Every entry in the favorite list should contains the following:

Field	Description
Symbol	Displays the symbol of the company
Company Name	Displays the name of the company
Stock Price	Displays the current stock price of the company
Change Percentage	Displays the percentage change of the price of the company. It should be rounded to 2 decimal places and the background should indicate an increase or decrease in the change. For example, the background should be green if the change is positive and red if it is negative. E.g. +1.50% in green background.
Market Cap	Displays the market cap of the current stock. Possible suffixes are {Billions, Millions, None}. Each value should be calculated and appended with appropriate suffix and rounded to 2 decimals. E.g. 5.71 Billion or 5.71 Million or 571000.

Additionally, there needs to be a few important features:

- Automatic Refresh – A switch (Refer to Section 7): when it is on it should refresh only the price and change percentage column every 10 seconds.
- Refresh button – Should refresh only the price and change column fields and not the rest of the table.
- Stock Details – The stock details activity should be loaded when the user clicks on any entry of the listView.

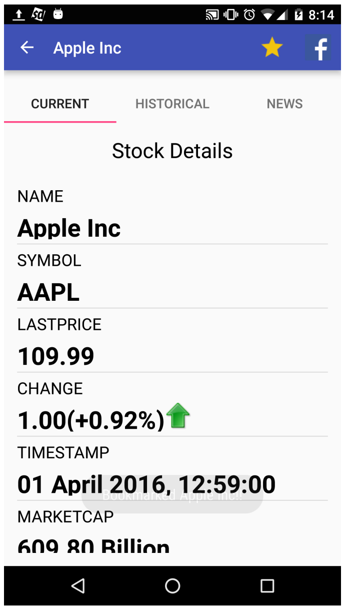
5.3 Stock Details

The Stock Details section should be designed as per Figure 5. The Stock details should be implemented using the ViewPager and TabLayout to render the tabs – 'Current', 'Historical' and 'News'.

The stock detail section should have 3 tabs:

- Current Stock
- Historical Charts
- News Feeds

The back button in the header should navigate back to the Search Form.



Stock Details

NAME
Apple Inc

SYMBOL
AAPL

LASTPRICE
109.99

CHANGE
1.00(+0.92%)

TIMESTAMP
01 April 2016, 12:59:00

MARKETCAP
609.80 Billion

Figure 5: Stock Details

The Stock Details would be starting with the 'Current' tab as loaded by default. Furthermore, the stock details would have a list showing all the stock values. The list of the stock details would be implemented using a ListView. The following stock would be displayed:

Field	Description
Company Name	Displays the name of the company
Symbol	Displays the symbol of the company
Stock Price	Displays the current stock price of the company
Change (Change Percentage)	Displays the change and percentage change of the price of the company. It should be rounded to 2 decimal places, followed by an indicator(image) which indicates whether the change is positive or negative. For example, the image should be a green upward arrow if the change is positive and red download arrow if it is negative.
TimeStamp	Display the timestamp of the last updated price.
Market Cap	Displays the market cap of the current stock. Possible suffixes are {Billions, Millions, None}. Each value should be calculated and appended with appropriate suffix and rounded to 2 decimals. E.g. 5.71 Billion or 5.71 Million or 571000.
Volume	Displays the volume of the current stock

Change YTD (Change YTD Percentage)	Displays the Change of the stock from the beginning of the current year till date. It should be rounded to 2 decimal places, followed by an indicator(image) which indicates whether the change is positive or negative. For example, the image should be a green upward arrow if the change is positive and red download arrow if it is negative.
High	Displays the highest value of the stock's price for the current date.
Low	Displays the lowest value of the stock's price for the current date.
Open	Displays the opening value of the stock's price for the current date.

Below the list of stock details, you need to show the today's stock activity in the form of an image. This would be implemented using an ImageView. The image of the current daily chart of the stock of the company should be retrieved via Yahoo charts API, as was done in HW8.

5.4 Historical Charts

This tab represents the historical charts showing the value of the stock in the past. It has to rendered as per Figure 6.

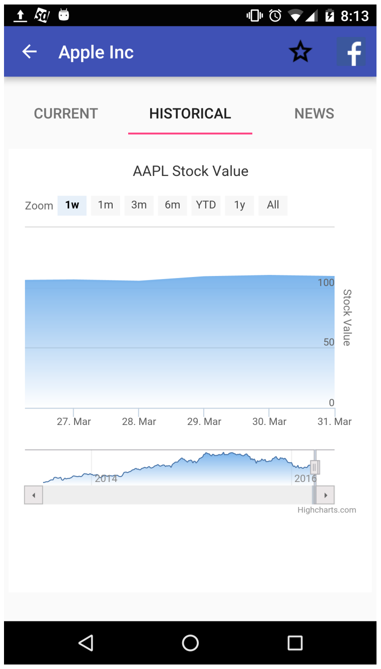


Figure 6: Historical Charts

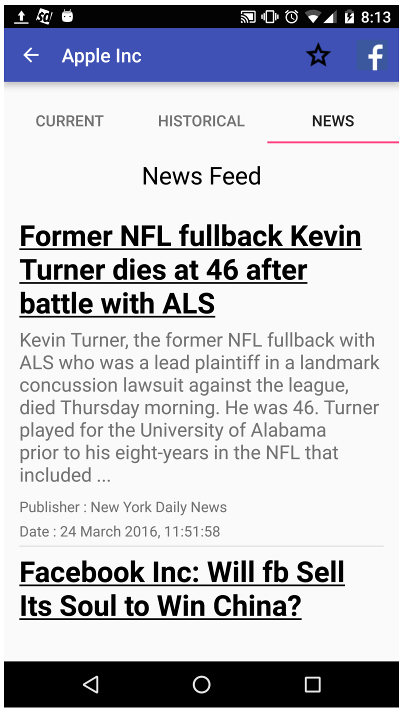
This has to implemented using a WebView using HighCharts API. This is similar to implementation in HW8.

It should have the following details:

- It should have the following zoom levels: 1 week, 1 month, 3 months, 6 months, 1 year, YTD and All.
- It should default to showing 1 week worth of stock data.
- The title of the chart should be company symbol stock value.
- The X Axis should be date time.
- The Y Axis should be Stock Value

5.5 News Feed

This tab represents 4 news articles related to the current stock. This would be rendered as per Figure 7.



News Feed

Former NFL fullback Kevin Turner dies at 46 after battle with ALS

Kevin Turner, the former NFL fullback with ALS who was a lead plaintiff in a landmark concussion lawsuit against the league, died Thursday morning. He was 46. Turner played for the University of Alabama prior to his eight-years in the NFL that included ...

Publisher : New York Daily News

Date : 24 March 2016, 11:51:58

Facebook Inc: Will fb Sell Its Soul to Win China?

Figure 7: News Feed

The news articles need to implemented in a ListView. Each of the entry in the list should display the following details:

Field	Value
Article Title	Represents the title of the news article
Article Content	Represents the content of the news article
Publisher	Represents the publisher of the news article
Date	Represents the date of the news article

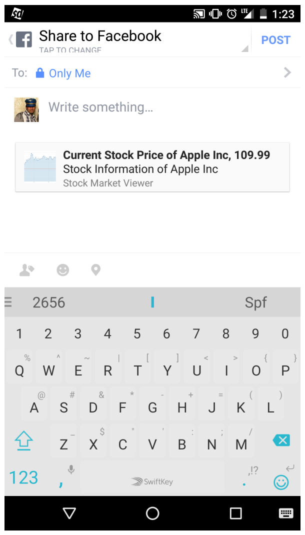
You will be using the Google News Feed API or Bing Search API.

5.6 Facebook Share

The “Facebook” icon should be provided to post stock information on Facebook (similar to HomeWork #8). The Facebook post from the Android app will contain exactly the same contents as from HomeWork #8.

When the user posts information on Facebook, a success message should be displayed on the Android screen. When the user cancels the Facebook dialog, a corresponding message should be displayed on the screen.

While posting information on Facebook, the app will authorize (log in) the user to Facebook; requests the user permission to access the user’s data; and then post the message to Facebook just like what was done in Homework #8.



Current Stock Price of Apple Inc, 109.99
Stock Information of Apple Inc
Stock Market Viewer

Figure 8. Facebook

6. Implementation Hints

See the HW9 IOS Clues file.

7. Material You Need to Submit

Unlike other exercises, you will have to “demo” your submission “in person” during a special grading session. Details and logistics for the demo will be provided in class, in the Announcement page and in Piazza.

You should also ZIP your Android source directory and SUBMIT the resulting ZIP file. Make sure that the source path does not include the .apk binary file.