

# **Software Requirements and Design Document**

**For**

**Group 5**

Version 1.0

**Authors:**

Darbi Bauer  
Jacob Dienger  
Aidan Ryan  
Katie Skipper  
Sarah Rosenfeld

## 1. Overview

We are creating a mobile application for stock recommendations. Based on price movements and news events, our app will recommend stocks and options from the NYSE and NASDAQ. We will provide any relevant information about price drops/increases. Each user may control which stocks they monitor, and will be given alerts based on “out-of-the-norm behavior” as opposed to predefined alerts. The app’s GUI is implemented in Java, using Android Studio. We will use Twitter to monitor news events and Google’s NLP engine to perform sentiment analysis. All backend computations will be performed using CUDA C++ and Python.

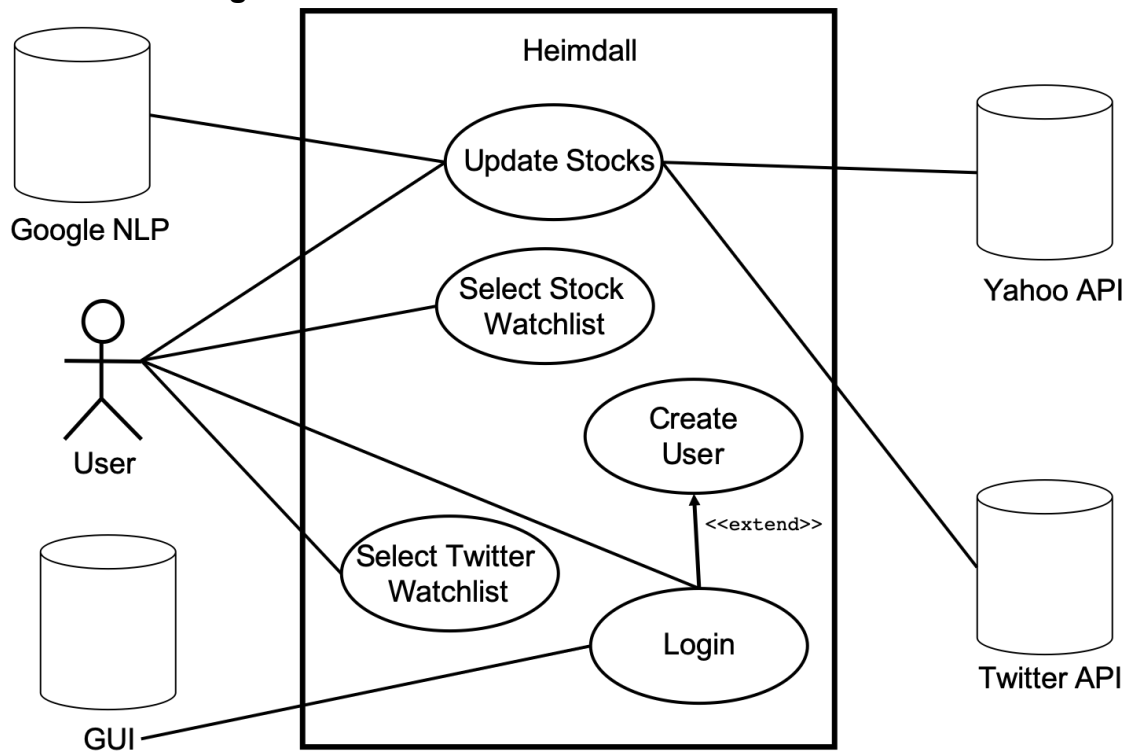
## 2. Functional Requirements

- Pull stock symbols and history from Yahoo API - HIGH
- Pull tweets from Twitter accounts specified by user - MEDIUM
- Send tweets to Google's NLP system - MEDIUM
- Calculate a score for each stock - HIGH
- Have an Android app interface - HIGH
- Store user data and stock analysis in MongoDB - HIGH

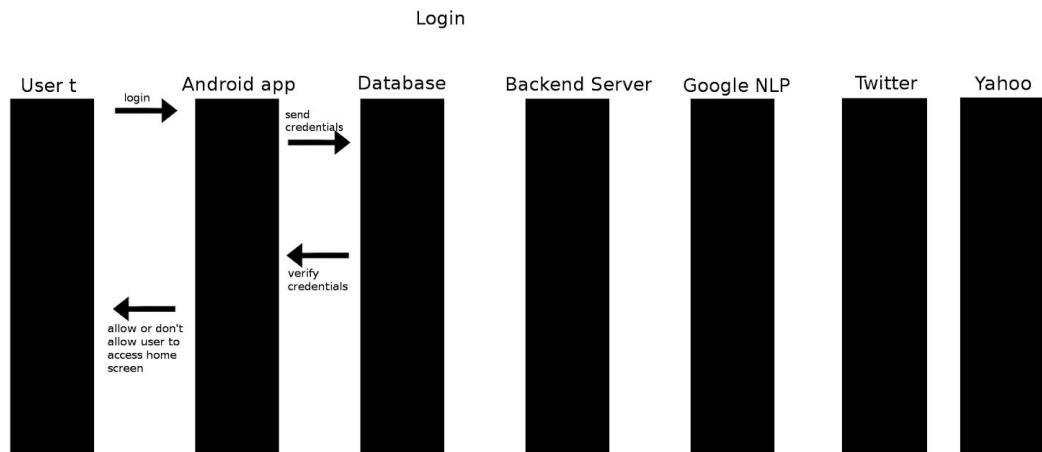
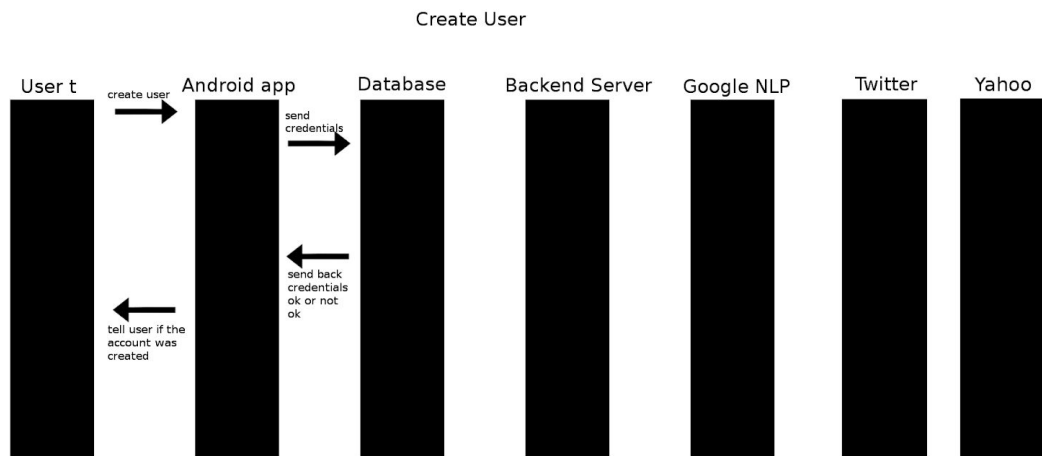
## 3. Non-functional Requirements

- Securely store user passwords in the database; required for user information safety
- Have a pretty app interface; a nice-looking app is more popular among consumers
- Up-to-date price data for both options and stocks; the application is useless if the data is not up to date

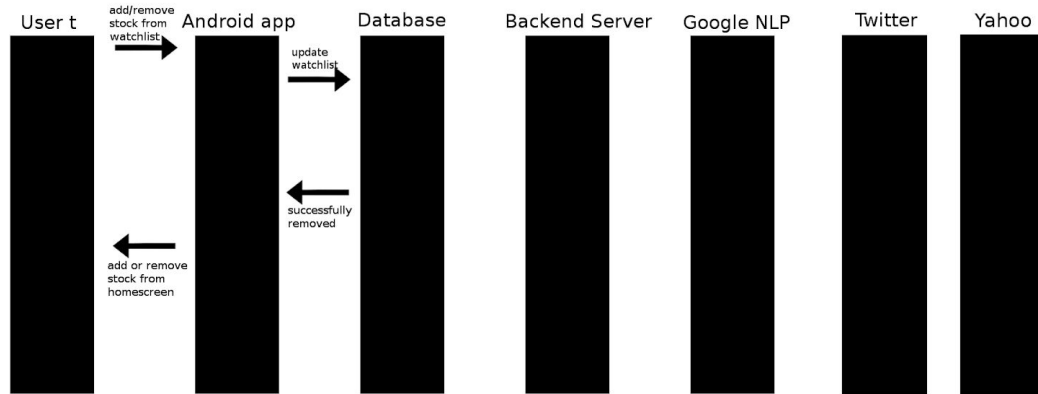
## 4. Use Case Diagram



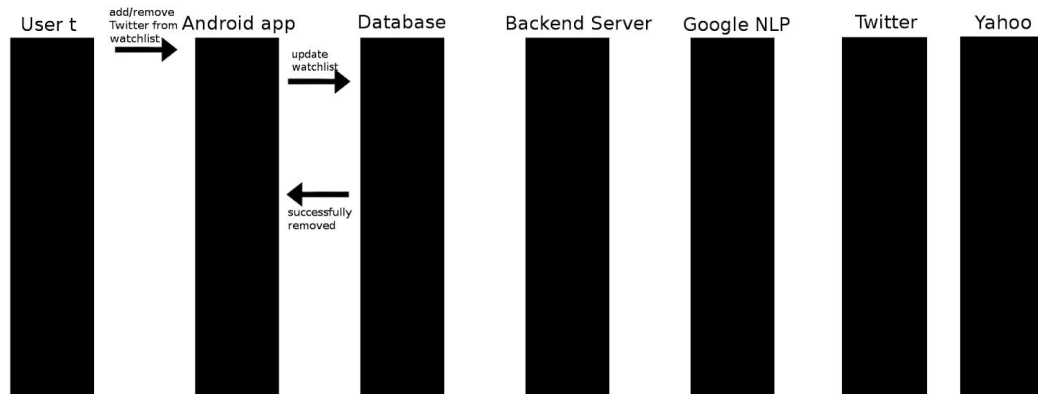
## 5. Class Diagram and/or Sequence Diagrams



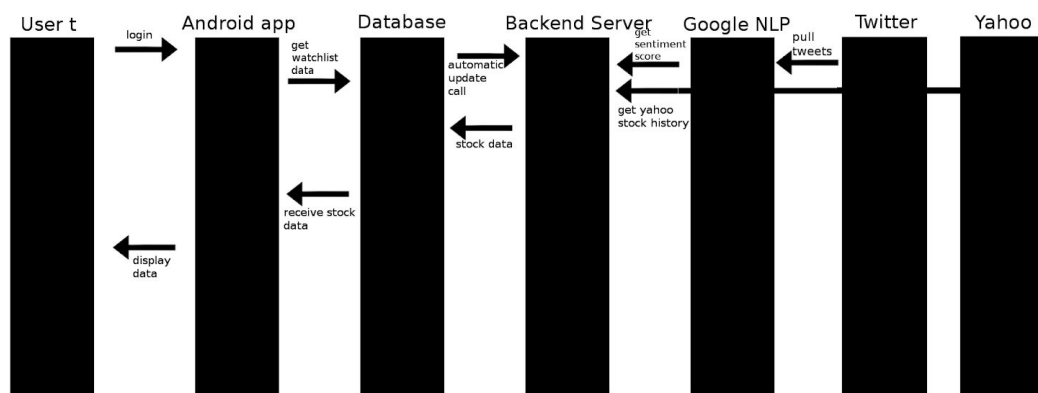
### Selecting Stocks



### Selecting Twitter



### Updating Stocks



## 6. Operating Environment

Our users will interact with the Android app interface, which uses data from our database. The database must pull the stock analyses from our backend server, which contains the actual data that we want to show users. Our backend server is using mathematical formulas to rate stock performance based on stock history. Stock performance ratings are also based on Twitter feed sentiment analysis, which is pulled from a program that will also be running on the backend server.

Twitter -> Google NLP -> Backend Server

Yahoo -> Backend Server : Mathematical Formulas[takes in data from Yahoo and NLP] -> Database

Database <-> Android App

## 7. Assumptions and Dependencies

- Yahoo's Stock API properly updating and allowing data to be pulled
- Twitter for Python library pulling tweets
- Google's NLP processing our data
- Our backend server processing the stock data that we push, and returning the analyzed data back
- The UI may look different on other phones than what we test on, which could be an issue if an element doesn't fit on the screen