| **Name of Team: CALM** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **First name:** Arjon | **Last name:** Sadikaj | **York Email:** arjons@my.yorku.ca |  | **Lecture Section: M** | **Lab Section: 03** |
| **First name:**  Catherine | **Last name:**  Tighe | **York Email:**  cat03@my.yorku.ca |  | **Lecture Section: M** | **Lab Section: 01** |
| **First name:**  Landon | **Last name:**  Navarre | **York Email:**  Landonn@my.yorku.ca |  | **Lecture Section: O** | **Lab Section: 1** |
| **First name:**  Moh’d | **Last name:**  Halabi | **York Email:**  alhala10@my.yorku.ca |  | **Lecture Section: M** | **Lab Section: 01** |
| **Project Title:**  Stock-Information Database | | | | | |
| **Project Description:**  The project we’ve chosen to work on as a team is a mini database, as suggested in the project instructions (3). We’re looking to alter the database in the later phases of the project so as it can retrieve current stock information.  Our project will be developed primarily in Java, although we may have to extend into SQL for the database-proper. We’d be using Android studio, likely for development and deployment. As of phase 0, the scale of this database is unknown. Whether we select specific stocks or use an outside program to connect to the whole market isn’t decided, although we’ll likely end on the former. Either way, the stocks available to search would be displayed in a sidebar.  The functionality of this app could be, say, an app that when opened retrieves current stock information and updates it’s database, after which the user can search for specific stocks to retrieve their information. This can likely be achieved without needing SQL, retrieving information from an external database and feeding it into, let’s say; an ArrayList or other abstract list in Java.  We realise the scope of this project might be too big for a course like this but we have a good amount of experience between our members, and with work divided it shouldn’t be too much for any one of us. Everyone here has some knowledge of Comp-Sci, and given we’re not dealing with less-taught languages as is the case in EECS 1012, the technical side of this project shouldn’t create any problems we can’t solve.  \*Update in phase 1. We realize it would be very, very difficult to obtain information on every stock, currency, crypto, resource, etc. Major stock exchanges can’t even display that to you, there’s just too many. As such, we’d choose to implement what’s available in the database. E.g; starting with Canadian funds, or crypto. | | | | | |

**Phase 2:**

* Technical Requirements:

1. A database will be created. Tentatively, this will be a proper relational database, for which we’d learn a bit of SQL. \*If this is too much in the end, this would be easy to fake with information instead stored in a text or XML file, and contained while running in a List of sorts.
2. When opened, the app will retrieve stock information to update the database to recent figures. We can obtain this from sites like that of Nasdaq. It should be noted that given how many there are. There is no easy list online of every traded company’s stock, resource, currency, crypto, etc. We would pick which stocks to add to the database (See req. 4 and the scope update in the project description).
3. The user will be presented with a main screen containing a textbox with which to search the database.
4. Users will be told, in a label below the search box, the proper format with which to type searches in the database.
5. The user will have an option to see the stocks searchable in the database. This is necessary because we likely can’t implement data for everything traded.
6. Given the small size of a mobile screen, the user will be able to bring up a second screen, containing the content listed in requirement 5.
7. Users will have a search button under the editText box, which will return the information of the searched stock under the search button in the main screen.
8. Users will be presented with a message should their search return nothing. This would be beneath the button (we’d like to avoid popups, it makes a phone screen a bit cluttered). This may direct them to the panel showing what’s available to search.
9. Users, should their search not return a result. Will be presented with what they might have been looking for, should it be in our database.
10. Upon starting the app, a list of example stocks will be displayed in the blank space under the search button. Upon searching for a stock of your choice, this list will dissapear.
11. Somewhere in the app, likely in the panel mentioned in Req. 5, there will be a button or link to the source we get information from. This is to let users know where our figures are from and for the case that they want information we don’t have stored.
12. A footer will be included, and will list the author’s names as well as the app’s affiliation to Lassonde.
13. Given we aren’t going to be refreshing data in real time, the user will be informed on the screen of what time the app was opened, and thereby the time to which the information is accurate.

