External interface requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

1. **User interface.**

   When the user opens the app, they will find/see the log in forms (fig 1) where they can log in if they're not first time users. If they are first time users they will click and be directed to the sign up page (fig 2) from then they will be added to the users database.

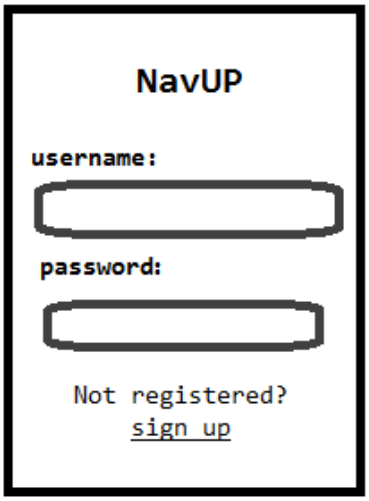

Fig 1.

Once users are logged in, the app will get their location (they can enter their location if the provided location is not correct or desired), then they will see one form on which they will input the desired location.

Fig 3.

Another extra functionality would include a module where logged in users can specify their favorite buildings or activities on campus, which would be stored in the database so that the user will be notified if there is an activity at the building or similar to the one the user spacified.

2. **Hardware Interface.**

   Operation systems or platforms supported: Android, IOS and browsers that support HTML, CSS, JavaScript (and other web-development and scripting languages). The NavUP app will depend on the UP Wi-Fi for buildings to be located and devices using the NavUP app must have Wi-Fi sensors

3. **Software Interface**

   The app is to be developed for Android, IOS and Web. Android studio will be used to develop for android (The API Android.location will be used amongst others), Xcode will be used to develop for ISO and web-development and scripting languages will be used for the Web app part.

   Software needed or to be implemented:

   Database of users (SQL+ or MongoDB)

   Data structures to store buildings, access points, routs and shortest paths.

   Real time data analysis and data streaming tools

4. **Communication Interface.**

   When the desired location is entered, the coordinates will be sent to the back-end software/data structure for the app to locate then give directions. The NavUP app may have a web based server, which will be created using PHP. The server will retrieve the needed information from the database/data structure. The HTTP server will use a push protocol to push notifications of updates onto the user's applications. Furthermore, whenever a user opens the NavUP app from their phone, a pull protocol will be used to retrieve and sync the latest updates from the server.