Chris Sterbenz

CSC415-01

10/18/16

**Project Name:** Get-A-Room

My mobile application falls under the second option of the project. It will be developed for Android devices, using Java as its main language. With the TCNJ Library’s consent, I will be using Springboard’s LibApp API to pull data on library statistics. To develop the application Android Studio will be used and BlueStacks will be used to perform preliminary quality tests.

Get-A-Room is a simplistic mobile application which provides students with the ability to quickly find open study rooms on any floor of the library—from anywhere they want. The goal of the project will be to significantly increase accessibility to study rooms for groups of students. This is only one module of the overall project, though. The app will strive to rebuild the way students interact with our fine campus. Initially this will be through finding openings in the library or elsewhere on campus, but there are many opportunities to expand once the core module is functional.

The biggest problem I will encounter in bringing this system to students is ensuring timely updates to room statuses. The longer it takes for the application to show students that a room just opened up, the less useful it is. Since the main focus of Get-A-Room is to unite students and group study locations with as little effort as possible, the algorithm I will implement will be aimed at predicting room status even without user-interaction. This is a machine learning algorithm which will take into account the various attributes a room can have (i.e. occupied, amt\_time\_occupied, last\_updated, floor, room\_number) as well as all of the data points it has collected on that specific room (or rooms in that area) to estimate the probability of that specific room being available. In essence, I will employ supervised learning to give students a list of rooms most probably available. As a reach goal, I’d like to implement a Traveling Salesman algorithm as well which will show students exactly the best path they should take through the library in order to find an open room.

Because LibApps already stores many of the fields required for this system, the data structures I use will be focused around queues, trees, and graphs. The queues will be used to ensure system updates can be verified before being sent to the application for user access. Graphs and trees will primarily be useful for the creation of the prediction and, eventually, routing algorithms.

Although I already have a grasp on many common machine learning algorithms, I expect to need to do more research into initially tuning the system. Since this is a mobile app there is also the problem of interfacing the algorithm with the system and, more generally, developing the simple user interface itself. Luckily Android development can be done using mostly Java which I am already comfortable using. XML is a language common to developing UI’s for Android and I will have to learn the tags associated with it. The biggest challenge will be becoming accustomed to either the Android SDK or the Google ADT, both of which include all the tools required to develop for Android using Java. To overcome this, I will use YouTube and other resources to gain an understanding as to what portions of the program I will need to use most.

**Open Source Licenses:**

1. Apache License 2.0: This license permits the licensee to do anything with the code provided he includes the original copyright and license in the new software. The licensee must also include notes of any changes he makes to the original software. It does, however, prevent the license owner from being held liable for software malfunctions or their trademark/name being used in a software extended from the original.
2. MIT License (Expat): This is a simple license which gives “rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software.” This means that it is incredibly open as long as the software is not provided with any form of warranty. This is very similar to the Apache License 2.0 but is much shorter and can be modified to meet the original software’s needs.
3. Mozilla Public License 2.0: Permits the licensee to create and distribute new software which extends the original as long as the copyright, license, and source code is included. This has the implication of being very conducive to use in any format—open source, closed source, commercial, etc. I like this license more than the MIT License because it requires that source code be released and, therefore, can be further expanded upon, applied elsewhere, improved, etc. In addition, it requires that the same license be extended to new software using the licensed code, with the grand exception of larger systems of which the original licensed portion makes up just a small part. This lends itself to be the most literal interpretation “open-source,” which is why it is the License am choosing to implement into Get-A-Room.

**Github:** https://github.com/csterbenz/Get-A-Room