

CS411

Team Member: Jesse Conde, Kenise Neal, Aliya Tazhibayeva

Deliverables:

1. **It must utilize a database.** A simple way to meet this requirement is to require a user to store profile information in the database. You'll also be using it as a cache.

For the database within our project, we plan to create a database of all the specific study spots on campus, down to as narrow as possible, whether it be per room or per building. Each building will contain data in regards to how many people there are in that building currently, how many people are usually in that building, and the maximum number of people allowed in that building. Using the average number of people in comparison to the current amount of people we would be able to create a heat map, where if the current number of people is above the average, then it will get warmer, the hottest being the maximum number of people, and the coldest being no people. Using this database will allow a student to more easily visualize where on campus may be the best spots to study at depending on what level of people you prefer to be around. Those who like crowded areas may opt for warmer heat maps, while those who like being secluded will search for the cold areas on the map. We will implement with noSQL databases

2. **It must correlate at least two publicly available data sets via API from the Internet.**

For our application, the essential information that needs to be displayed is the extent of how busy a specific study spot is. We plan to split study spots according to the buildings and indicate the busyness using open-source Heatmap APIs available online. The heatmap of the buildings would give a good sense to the users of the areas on campus, which are less crowded during the midterms or final exams season. The actual headcount of people studying at a study spot at a given time would be calculated through the number of devices connected to the given nearest wifi portal. We need to investigate if we could use BU IT's API to fetch this information. Besides, we are planning to indicate the trend of hours during which any given study spot building is the busiest and the most available, utilizing Google Map API.

3. **It must use third-party authentication**, for example logging in with Twitter or Facebook using OAuth.

We plan to use third-party authentication for the study space application. This will be potentially used to save the locations that each user "favorites", like how Twitter saves the tweets that the user likes. We feel this is the one of the best benefits of using third-party authentication with our software. This will be used to make sure the "favorites" marked by the user are protected even though multiple users may favor the same location.

4. **It must have a decoupled architecture**, similar to what we looked at in class during the 'dogfooding' lecture. The implication is that you'll need a front end and a back end, and the two will communicate via a RESTful interface. It's too early to discuss technologies, but this does mean that there will be JavaScript in the front end. Since the back end is responding to requests and just returning data, it doesn't necessarily need to be in JavaScript...Python, Java, PHP, and so on would work.

The application we plan to write our code in is JavaScript on the frontend and Python on the backend. We also plan to use Visual Code Studio to produce the code and make sure any issues are debugged.