Team Optima
AJ Miller
Thomas Curci
Chelky Lin
Shuler Cotton
CPSC 4140
April 2, 2018

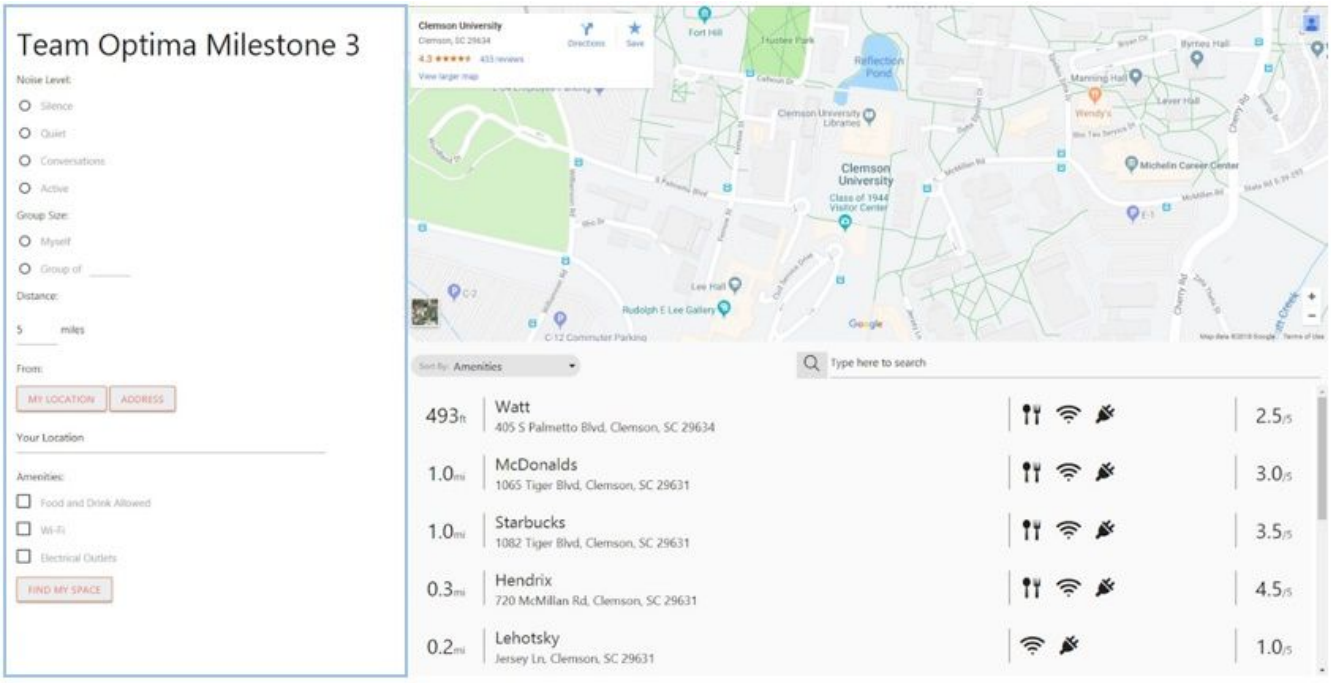
Table of Contents

Description of Prototype

Our project will be addressing the domain of “be a more efficient you” to identify and alleviate a common daily hassle that wastes time and reduces efficiency. We are interested in this domain because we believe there are many common challenges we face daily that can be avoided or eliminated through the application of positive human-computer interactions. Being a more efficient you allows for less stress from unnecessary challenges, opens more free time in your schedule, and allows for a greater amount of energy and thought to go towards what you find more important. In our

Milestone 1 we identified that tasks such as identifying the best parking lot on campus to check on the way to class or work, the optimal route from one location to another, and finding a location to study that meets one’s environmental needs are areas where people are losing efficiency. After the feedback from Milestone 1, the team decided to shift its focus to solve the following problem: people waste time and effort when finding a work or study location that is not always optimal. In Milestone 2, we designed three different systems that attempted to address this issue, and we decided to select our first design to build a prototype for. More details on this decision can be found in our Design Justifications section.

Having selected our design, we developed a web-based application for our prototype. Our primary motivations for continuing with this design were the ease at which we could model, change, and deploy iterations of the prototype, the adaptability of a web application to reach both mobile and online users, and the availability of API tools that helped us to integrate a mapping system into our interface. Our prototype serves to allow a user to interact and gain experience with the primary functions we prioritized based upon our design. Since there were many components we mentioned in our design, we selected those that we believed to be most essential to the operation of the system. More details on the our decision to include and exclude features can be found in Implementation Challenges. The following is a detailed description of each primary function:

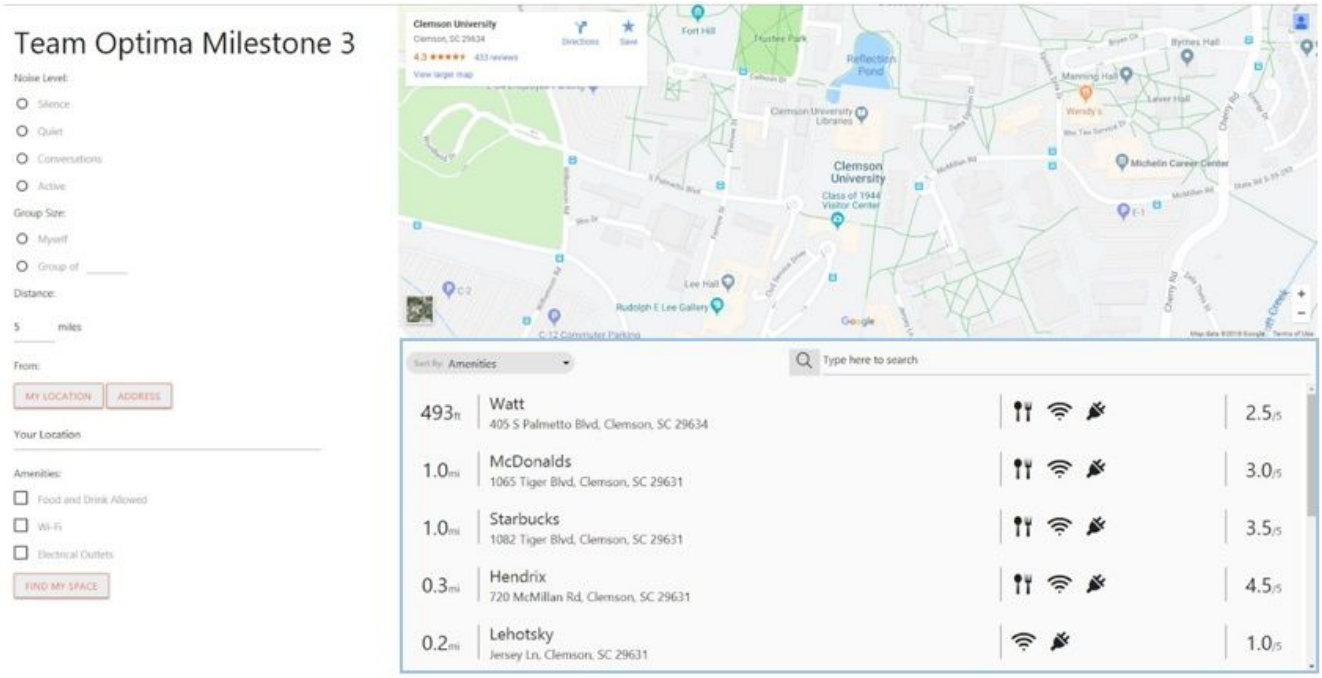


Selecting settings

The left side pane of the web application is designated for selecting the settings involved in picking a space. The current selection options are Noise Level, Group Size, Distance, Location, and Amenities. These selections are all optional, but used to narrow the scope of the locations selected. Each location within our backend database includes data points about its characteristics with respect to the selection panel (they are randomized and may not necessarily be accurate for the prototype since we randomized some to create our database size).

Finding a space

By inserting the settings that a user wants to see in their space within the selection menu, they can narrow the search scope that the system returns. When they are happy with their selections they can press “Find My Space” for the system to search for a location that meets their criteria and display it in the bottom panel. The selections can be changed and searches conducted again to refine the search.

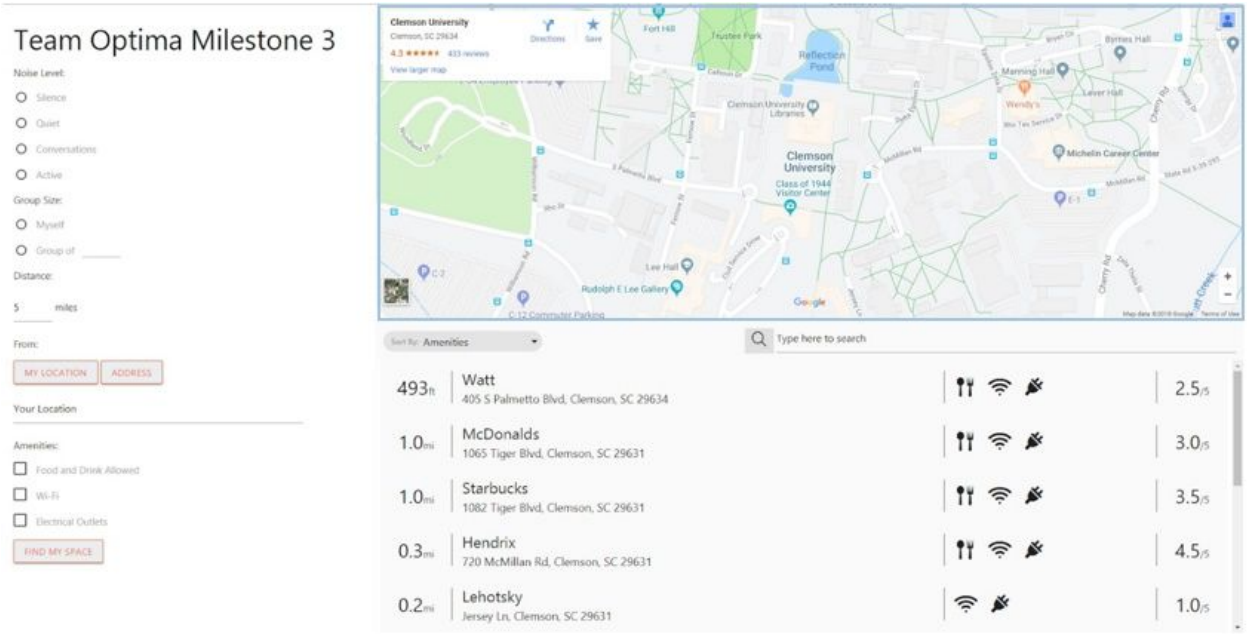


Sorting Locations

Once the user has selected their preferred settings and has conducted a search for locations nearby, they are able to sort the locations based upon three criteria within the prototype. The sorting criteria are based upon the information displayed within the bottom panel of (from left to right) distance from the selected location, the name of the Location with address beneath, amenities available, and overall rating from other users. The list will reorder based upon the sort criteria selected. The user can also search the list for the names of particular locations they may be interested in or looking.

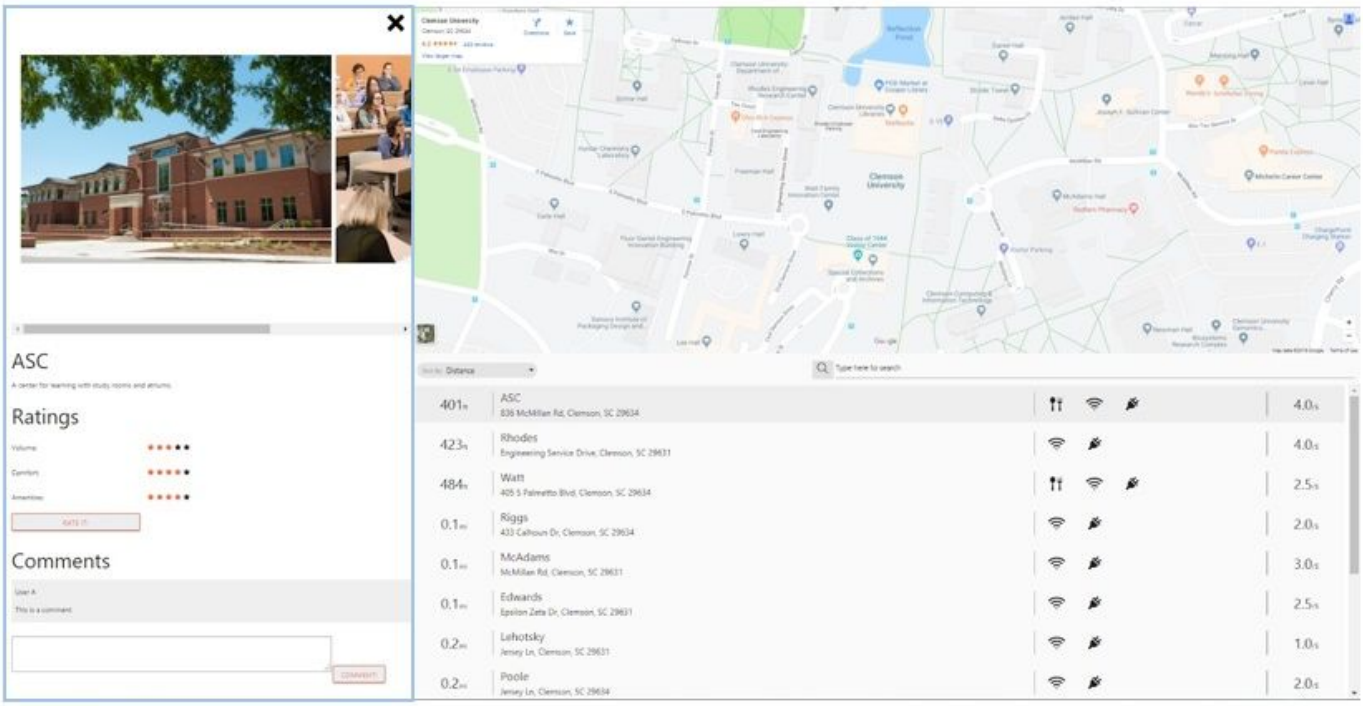
Selecting Locations

Once the user has selected, found, and sorted a satisfactory set of locations, they can then select the location to find out more information about the location. Clicking on any of locations that are displayed in the bottom panel will activate a new panel on the left that contains a description and pictures about the location and rating and commenting functionality.



Interacting with the Map

The user is also able to interact with the Google Map component in the top panel. Unfortunately due to technological limitations from our decision about how to incorporate the map we are not in this prototype able to set widget flags for the location and send API calls without reloading the entire panel. This was something we figured out how to resolve, but did not consider it a critical function over other components that needed more work. We would have liked for the user to be able to see all of the locations in the bottom panel on the map and be able to select the widgets as well to load the more information panel on the left side and be able to recenter the map based upon entered location.



Rating and Commenting on Locations

Once a location is clicked by the user from the bottom panel, then a new panel will appear on the left side of the screen with more information about the location. Within this panel the user is able to see pictures of the location, a more detailed description of the location, ratings and comments from other users about the space, and rate and comment on the location themselves to enable data collection from users. Although this prototype primarily demonstrates distribution of information to users (the most common process), we also included the features of data collection here that would help to enable the system to work on the basis of crowd-sourcing.

Design Justifications

We felt that creating a web application for our prototype was the best course of action. There are quite a few reasons why developing a web application would be the best way to implement a solution to this problem. One of the major reasons we chose the web application over the dynamic display and the compass mobile application was that the web application allowed our users to customize their search results based on the things that they desired when we gathered data for the previous milestones. With the dynamic display, we would've had to decided as developers what information would be put on the display so that users could make a well-informed choice on what study space would be best for them. With the web application, we are not limited to how many different types of preferences a user could filter their study space results with. For example, if we found out that many people think that there should be an option to filter by the size of the building, we could easily implement that within a day. With the

dynamic display we are limited by space, and therefore if we wanted to add that feature we would have to remove other important criteria such as how noisy a building typically is. With the compass mobile application, we could also implement it without having to remove other features.

Another reason we chose the web application was because users have the ability to find a location even when they are off campus. The dynamic display, being a physical display, is only able to be used by those who are near the physical display. This greatly limits the amount of planning that people can perform when finding a location. This is where the web application blows the competition out of the water. The compass mobile application is geared more towards those who are already on campus walking around. Technically, one could use the compass mobile application from an off-campus location, but it would probably be an unpleasant experience to follow the directional arrow through streets that aren't aligned with the arrow. If we were to fully develop our web application we could add the functionality to route the trip to your selected location through a service such as Google Maps and display it right there on the web page.

One more reason the web application was chosen was because it's a simple design to use. In this day and age, almost everyone would know how to work a simple webpage with a few radio buttons and a few regular buttons. There would be a much lower barrier to entry for the amount of technological knowledge somebody would need to use the web application. On the other hand, many teenagers or young adults may have a learning curve when trying to comprehend the compass mobile application. This group of people are not used to having to read a physical map or having to understand how a compass works. The interface is relatively complex in that regard and may turn a lot of young people off from using the application. Since our main target would be college students it's important that we develop something that they all can interpret easily.

In studio 2, we were presented feedback by our peers that helped us get rid of the idea of a kiosk which was basically the web application, but only accessible if you were physically near the kiosk. Instead of the kiosk, we came up with the idea of the compass mobile application. The compass application gave us many more great ideas to use in our implementation of the web application prototype than the kiosk would've given us. We also added the ability to choose a location that is outdoors as opposed to an indoor setting. This was suggested to use by one of our peers and we thought it was a really great idea as some people prefer to sit in an outdoor setting to do their studying. If we were to fully develop this prototype, we would likely add it as one of our preferences.

Technology Justification

We used HTML, CSS, and JavaScript to develop this web application. While they are the most basic web development languages, they are easy to pick up and start developing with. These basic languages utilize browser technologies, which allows the application to work on multiple platforms and reach the largest target user base. We also used JQuery to assist with JavaScript-related code and Materialize CSS to assist with styling. Using these two frameworks, we did not have to write all of the CSS styling and

components from scratch. Materialize is based on Google's Material Design which is a guideline for user-friendly and aesthetically pleasing interfaces. We also used SVG and icon fonts to simplify and display iconography with high resolution. Overall, these technologies allowed us to develop an application prototype that considers the user experience while providing an aesthetically pleasing interface.

Implementation Challenges

Implementation challenges we experienced were mostly challenges in styling elements to appear as desired, as well as creating new functionality and components. Lack of a thorough understanding of CSS led to many cases of frustratingly trying several combinations of CSS rules to find a combination that did what we wanted. The main cause behind this problem is that some combinations of CSS rules and HTML elements have hidden conflicts that prevent them from interacting as one would expect. The difficulty behind creating new functionality and components was mostly due to the bundled issues that arise with imperative programming and web development. An alternative would be to use functional programming and incorporate a state manager that would keep track of the state of all elements and reduce side-effects. These challenges led to the following issues.

Implement floors and rooms

Due to time restrictions, we were not able to implement the 3rd mockup from our first design in P2, which was the ability to search floors and rooms of different buildings if applicable. We had planned to be able to do all 3 mockups, but decided to focus on making the first and second mockups work as intended. Most of our final days were spent debugging a variety of issues with these first two mockups and adding new features that were suggested in studio 3. We felt that it was more important to have something useful than having all 3 mockups that were more error prone.

Add API support for Google Maps

In our mockups we planned to add a functional Google Map that showed the location of the building you selected on the map. However, we decided that this one bit of extra functionality would not be worth the time it would've taken to learn the API and implement it just for a prototype. Our time would be better spent focusing on the main features of our application instead. If we were not limited by time, we would've included this in the coming weeks. The address button and field also are non-functional for this reason. If we use Google Maps API we should be able to easily get this functional in the backend.

Add more preferences

There were some suggested preferences that we should add to our application such as whether or not the location was indoors or outdoors. We also got a recommendation to determine at what times of the day a location is noisy, and present that information to

the user. This would've either involved having access to a large data set (which we don't have) or crowdsourcing this information at the time the user submitted their review. We also chose not to focus on this because of the lack of time between when it was suggested (Studio 3) and the due date. It is probably a very niche feature that users would like to have, as users can usually get a grasp of how the sound will be at that time of the day without needing it displayed to them.

Submit new locations

This was a piece of feedback from studio 3. We prioritized data distribution over data collection from users since that is the more frequent task and doesn't help as much to solve the user's individual problem of not being able to find the optimal study space.

Crowdsourcing (capacity) / gamification

This was something we had been considering since P1 but it ended up not being a main focus of the prototype because of the complexities involved with this kind of data collection. This feature more than likely would have been combined with mockup 3. An incentive system such as gamification would have to be implemented after our final product was finished. It is an idea that we would probably implement given the time and resources to do so.

Feedback from Studio 3

We got a lot of good feedback from our studio 3. On Piazza, our peers mentioned the fact that our GUI looked incredibly neat. There were a few critiques that we took into account when adding final touches to the prototype. It was suggested to us to add a rating system to each building. We successfully did that with a rating for amenities, comfortability, and noisiness. We were also told that users should be able to add written reviews for a building in case they weren't able to give an accurate review with the options we gave them. We successfully added a comment feature to the rating system. We increased the amount of color on our application by adding pictures of each of the buildings and giving it a Clemson orange color scheme. One person also mentioned that some of the wording looked crammed together, so we spread everything out a bit more so that it is more appealing to look at and easier to navigate.

Evaluation Plan and Usability Specifications

The four criteria the team will evaluate are:

- Usability: the main goal of the class and any software system
- Learnability: important so that users will remember how to use the system
- Helpfulness: important in solving the problem of users finding the best study space

- Efficiency: important to user's experience but also to Team Optima's goal of making the experience as efficient as possible

The qualitative data collected from the survey will be directly correlated to the user's experience with the prototype. Users will be asked to either agree, disagree, or remain undecided with statements regarding the prototype's usability, learnability, helpfulness and efficiency. This method will be similar to the SUMI method, which has been justified through 25 years of testing and usage [1]. Examples of statements include:

- It is easy to make the prototype do exactly what you want. (usability)
- Learning new functions is difficult. (learnability)
- I can understand and respond to the prototype's information correctly. (helpfulness)
- This prototype does not work fast enough. (efficiency)

There will likely be 16 statements that users will be asked to complete. This will allow 4 data points from each of the 4 evaluation criteria to be collected. At the end of the survey, there will be a required open-ended question in which users will be prompted to leave their overall feedback on the prototype. Not only will this allow the team to capture missing information not captured in the tests or statements, but this text data can also be visualized through a word cloud to highlight key phrases that might be of interest. The survey, including the statements and the open-ended feedback, will be administered to the users immediately following a usability testing session.

The usability testing session will aim to direct users to complete key system tasks. These tasks are defined below, as well as a description of why they are included/what previous milestone and/or studio requires them to be addressed during testing:

Task	Why	Time limit	# of slips	# of mistakes	# of Q's
Find the study space that is closest to you.	P1	5	0	0	1
Find a study space that is silent and has wi-fi.	P1	6	1	0	0
Find a study space that is just for yourself and is quiet within 1 mile of your location.	P1	7	1	0	0
Search for "Watt" in the results search bar.	Studio 3	4	1	0	1
Sort a list of study spaces by their rating.	Studio 3	2	0	0	0
Select the study space with the greatest rating.	P1	4	0	0	1
Rate the selected location.	Studio 3	8	2	0	1
Comment on the selected location.	Studio 3	6	3	0	1

Time limit: The maximum amount of time (in seconds) that the user is expected to complete the specific task.
# of slips: This is the maximum number of mistyped keystrokes, redundant button clicks, or missed button clicks.
# of mistakes: The user does not perform the goal of the task.
# of Q's: This is the number of questions the participant asks while trying to complete the specific task. These can be rhetorical questions or direct questions to the observer.

This usability testing will enable the team to document the user's experience with the prototype. This data, in addition to the aforementioned survey data, will give quantitative and qualitative insight into the main evaluation criteria of the prototype: usability, learnability, helpfulness, and efficiency.

Bibliography

[1] “What is SUMI?”, *SUMI*. [Online]. Available: http://sumi.uxp.ie/about/whatis.html