**GROUP EVALUATION:**

This app’s main goal is to simplify the way students navigate the UC San  
Diego campus. A major issue that we found was the lack of error  
prevention. The app should force the user to turn on their GPS before  
engaging with the app.  This needs to be done to improve user error  
prevention, since a vague “you are here” location point on a map that is  
representing something as small as a campus would result in user errors.  
  Additionally, location pointers are marked with labels A,B,C etc.  This  
violates the consistency and standards heuristic.  The user will find it  
difficult to keep track of the pointers if many pointers are in use. The  
visual appearance of the application is confusing. There is an excessive  
amount of menus (side menu for sorting places, main navigation menu) that  
is confusing and overwhelming in the way information is displayed.  A  
final point that was troubling for us was the lack of a “you are here”  
button that automatically starts on the page.  This button should also  
automatically update.  This will be done to increase error prevention,  
because without it, a user could pass their destination, or even be  
traveling the opposite way and not be aware until they actively hit a

button to double check. We found this heuristic evaluation useful for  
addressing core issues in the navigability of the groups app and the  
ability to help the group prioritize their issues was very helpful.  
However, it was less useful for offering suggestions or insights as to  
how the group could approach solving the problems we identified. The  
evaluation helps get constructive criticism flowing, but doesn't help  
much by way of expressing useful desires for the groups project.

I found heuristic

**Visibility of system status**

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Prototype 1:

* It is difficult to tell where the user is on the map at first without knowing that you have to click on the GPS icon to turn it on and activation your current location.
* There is no loading status for the map.
* I like the fact that the search bar pulls up suggestions that contain the same letters as you type them in. It makes the user feel like the search is more dynamic.

Prototype 2:

* Again, like in the first prototype, there is no way to tell where the user is located without knowing to press on the GPS tab first.
* There is also no loading status for the map.
* Pressing on the GPS tab pulls up your current location, but the lack of real-time navigation makes this feature a little useless anyway.
* The search bar has the same functionality as the first prototype. It is nice that it pulls up locations based on what the user has typed in.

**Match between system and the real world**

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

Prototype 1:

* The map is static but should be dynamic, allowing users to get a closer look or zoom in if they wish.
* The map on this prototype is too general (it’s just a blob). Later, in creating and designing the map interface, there should be labels for buildings (by name, most importantly).
* The search should be able to recognize the same building if the user were to type in the full name of the building versus the abbreviated name.

Prototype 2:

* The icons are a little confusing and should be reconsidered. “Arrival Location” has no icon. “+” is a little ambiguous because tapping on it focuses on the search bar. The icon for the share tab is only recognizable if the user is familiar with Windows 8.
* Like in Prototype 1, the search should be able to recognize the same building if the user were to type in the full name versus the abbreviated name.

**User control and freedom**

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Severity:

Prototype 1:

* There is limited flexibility in the map. There is no control over zooming in or out of the map.
* It also appears that you can’t rotate or orient the map as well.
* The way the pins are labeled (A, B, C, D...) means that the user only has a limited number of places he can pin (26 pins). Additionally, the user has no simple way of changing pin labels. For example, what if I wanted Center Hall to be labeled “Z” instead of “E”? I would have to research for center and set a label for it again.
* I can see this app getting cluttered if the user has too many pins. There should be a way to organize pins or at least toggle them on/off.
* In a situation where the search results have multiple suggestions or too many past pins, the suggestion area (below the search bar) will cover up the entire screen. In this situation, there is no way to exit or leave search.

Prototype 2:

* There is no way to edit a pinned location.
* There is no way to toggle the visibility of pinned locations on the map, or to organize them.
* User can’t overlay his/her own location on top of pinned locations on the same map.
* The user is not able to select which letter label he wants for each location. The labels are assigned in the order of the locations searched. The user should be able to have the flexibility to pick and choose his own labels. For example, what if he wants to assign “C” to “CSE Building” instead of the default “A”?
* There is no way to cancel a share task.

**Consistency and standards**

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Prototype 1:

* The labeling of pins using “A, B, C, D…” is confusing and misleading because of two reasons. 1) The label letter doesn’t necessarily mean that the building or location starts with that letter, although it can easily be confused to fit into this standard. 2) This type of labeling generally implies that there is some type of order or priority, which there isn’t in this case.
* There should be a way to remove pins without having to first open the search bar. Users should be able to remove it directly from the side menu where they appear, or maybe on the pin itself on the map.

Prototype 2:

* Again, as in Prototype 1, the use of letters as labels is confusing and doesn’t fit with usual standards.
* The search bar should be consistent and remain at the top of the screen where users would expect it to be. The current implementation of this prototype moves the search bar downwards on the screen if there are currently locations pinned.

**Error prevention**

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Prototype 1:

* I like the fact that the search bar pulls up suggestions that contain the same letters as you type them in. It helps to reduce spelling errors if someone didn’t know how to spell the name of the location they are looking for. It would be a good idea to incorporate both the name and the abbreviation form of locations (example: Computer Science & Engineering Building AND CSE Building).
* The GPS should be turned on before the user engages with the app to help reduce confusion or error in where the user thinks he is currently located.
* There is currently no way of handling a situation where the user types in a location that does not exist on the map.

Prototype 2:

* The GPS should be prompted to be turned on before the user interacts with the app in order to prevent users from confusing their location. The user should not have to click on the GPS tab to locate themselves. This will prevent the user from traveling in the wrong direction or pass their destination without realizing it.
* The share page should have a cancel button or a dialog that asks the user to confirm before they want to send information in order to prevent the user from sending wrong locations.
* Similar to Prototype 1, there is no way to handle invalid user input in the search bar.

**Recognition rather than recall**

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Prototype 1:

* Users have to click and hold on a label (on the side menu) to know which building it represents and/or to drop its pin on the map. This can be inefficient for users because they would have to click through each label to find a certain one they are looking for (which would be time consuming), or memorize which labels represent which location.

Prototype 2:

* Since the GPS tab is on a different map than the campus map, the user would have to remember his or her location before navigating back to the campus map. I would put the user’s location on the same map as the pinned places.
* In the share tab, the user would have to look up the name of the location that they want to share. This would be inconvenient in a situation where the user is not familiar with the campus or building names and would have to navigate back to the campus map or the GPS tab to locate his surroundings and then remember the name and type it into the share tab.

**Flexibility and efficiency of use**

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Prototype 1:

* I like how the search bar saves my previous searches. This makes it faster for me to search a location if I have looked it up previously.
* Again, the user’s location should be turned on by default for flexibility. For example, what if I know that I’m near a building, but I’m just unsure of which building it is because I’m surrounded by so many. I should be able to open the app, see my own location on the map, and just take a glance at the buildings around me without having to do a search and pin it.
* The user should be able to remove a pin from the map if they don’t want to see it because they are already familiar with it, but keep it stored as a saved location on the side menu in case they want to refer back to it or share it with a friend.

Prototype 2:

* Previous locations are saved at the top of the screen, which helps familiar users quickly access locations that they have already looked up.

**Aesthetic and minimalist design**

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Prototype 1:

* I can see this design getting too cluttered if the user has accumulated too many pins.

Prototype 2:

* There are way to many menus. There is a main navigation on the bottom, a menu on the right side of places that have been pinned, and a search bar/menu with pinned locations at the top.
* The pinned locations on top of the search bar are unnecessary takes up space. If there were more locations added, it would move the search bar downwards, which would enclose upon the space of the map.
* The “search” and the “+” tab have similar functionalities – they both add a location to the map. Having both options is redundant and unecessary.
* This implementation can also be cluttered if the user were to pin several locations.

**Help users recognize, diagnose, and recover from errors**

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Prototype 1:

* There is no form of help or error messages implemented on this prototype. For example, if the user were to look up a location that didn’t exist on the map or misspell a location, there would be nothing to notify the user or correct him.

Prototype 2:

* There is also no error message for Prototype 2.

**Help and documentation**

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Prototype 1:

* There is no existing help documentation in this prototype.

Prototype 2:

* There is no existing help documentation in this prototype.