Cognitive Walkthrough

Briefly, the software is for finding helpers around my location on the subjects I need help on. This is a user system, so the sign-up process is needed. On signing up, the user put information about him such as name, location, things he needs help on, and things he can help others. The whole purpose of this system is to find the helper who can help me and possibly I can help, also. It is an iPhone application, and we hope to extend it to the web.

It is not something that is commonly used before, so we assume no general information about the system. When a user first runs the application, it prompts the user with three buttons which are [Register], [Login] and [Try me]. There are only these three buttons displayed on the first screen so it is straight-forward to users what options they have. The user happens to be a registered user so he pushes [Login] button to login.

After the user logs in he sees a list of helpers he found last time on the screen and five control buttons on the top. The five buttons are [Search], [Map], [Help List], [Message], and [Logout]. The user wants to get help on CS 465 class. He pushes [Search] button to search for the helpers but he finds that the screen it shows after pushing [Search] button is the result from the last search. He pushes [Help List] button to see if that is the correct function.

[Help List] shows the list of the subjects he input last time. He was not so sure if he can modify the list. He finds the [+ ] button on the right top corner and recognizes it to be the [add] button. He successfully adds CS 465 to the list. He wants to get rid of some of the subjects he already got helped on. There is only [edit] button and [+ ] so he thinks [edit] might do the job. He pushes the [edit] button and the list changes its look so that it is “editable.” However, there is no explicit button to remove an item from the list. There is red buttons next to every item on the list. He touches that button, and [delete] button appears on the right. And he successfully deletes the items he does not want from the list.

Finally he presses [Search] button to search for the helpers around him who can help him. The GPS feature of the iPhone is shown and many pins around the current location are shown. The names of helpers are shown on the pins. He pushes the closest pin and the screen changes to the list of the helpers available at that location. Then he selects the person who is on the first on the list. He then presses [Messages] button to send a message to that person to arrange the meeting to get helped.

Heuristic Evaluation

1. Validity of system status
a. Users are informed about the system status on every situation. When there is an error during searching for the location and the helper lists, the pop-up window opens up to inform the user to know that there was an error trying to get the information.

b. There will always be feedback on the button usages and for most of the buttons, the screen will change according to the buttons the users pushed.

2. Match between system and the real world.

a. All there are to be used for the language are class names and locations, thus it provides simple languages.

b. Words and phrases used are all those that the users use frequently (classnames and locations etc), it is very familiar to users. Also the way to find other users is conceptually similar to what most of the people do in newspapers.

3. User control and freedom

At any step within the application, there is always a button which let the user goes back to home menu as well as previous menu

4. Consistency and standards

Due to iPhone's touch-screen, almost all actions can be executed by either scrolling the screen or touch the button. Scrolling is used as it is used in Operating system and touching is used as if the click is used in Operating system.

5. Help users recognize, diagnose, and recover from errors

Messages are helpful in a way such that all the error messages or confirming messages consist of everyday languages as well as it explains the cause of errors and way to solve them.

6. Error prevention

When pinpointing on the Google map, it is sometimes hard to get the exact location unless the place is zoomed in. Also Tabs named "search", "help list" are ambiguous since unlike the usual [help], "help" in the program refers to help that the user want to get. Also "search" has two different functions within the application. One is for usual [search], and other for "searching the match".

7. Recognition rather than recall
Objects are accessible if the objects are directly related to the users' current task.

8. Flexibility and efficiency of use

There are tabs on top of the applications to easily navigate, but it might be more useful if more shortcuts are provided.

9. Aesthetic and minimalistic design?

There are too many screen changes to that sometimes it's hard to keep track of current pages.

10. Help and Documentation

Necessary help and documentations are provided but it is sometimes insufficient.