Evaluation Assignment # 2

Heuristic Evaluation

Sree Ram Akula
(sreerama@mtu.edu)

Programming Staff - Staff Gauge

Staff Gauge

Staff Gauge application facilitates the citizen scientists to upload and visualize staff gauge hydrology data. Staff Gauge application forms a citizen scientist network which allows its users to upload the hydrology data about wetland and water stream from sites places on trails, in parks, and along rivers around the world. Basically our application has a HTML based form to collect the data from users which is compatible on mobile and web browsers. The data to be uploaded can be either text or picture which is stored in database(Tomcat) server in compliance with ODM schema typical to hydrology data storage. The users can not only have the access to data which they have uploaded but can have access to the data uploaded by other users. The primary users of the application generally clicks the pictures of staff gauges and upload it to the database which is used by the scientists for their research.

UI Domain

UI domain of the application is Web-Based User Interfaces(WUI).

Staff Gauge application is a web-based mobile application mostly used by the primary users such as volunteers and the scientists. Users start the application and select the location from the list or enter it. Then they can select any of the two forms to enter the data depending on the type of data they are interested to upload i.e either a text form or to upload a picture. If the text form is selected then it would ask the user to enter the measurements in meters then click next. If the image form is selected then the user can click the picture and click next or he can even cancel the picture if it is unclear or inappropriate. Then the you navigate to a new page which shows you the recent updates and the trends so that one can identify or make sure that what they have entered is a related data. If they feel its relevant data then they can click submit to upload their forms to database.

Heuristic Usability Principles (Jackob Nielson)

1. Visibility of system status

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

2. 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.

3. 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.

4. Consistency and standards

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

5. 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.

6. 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.

7. 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.

8. 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.

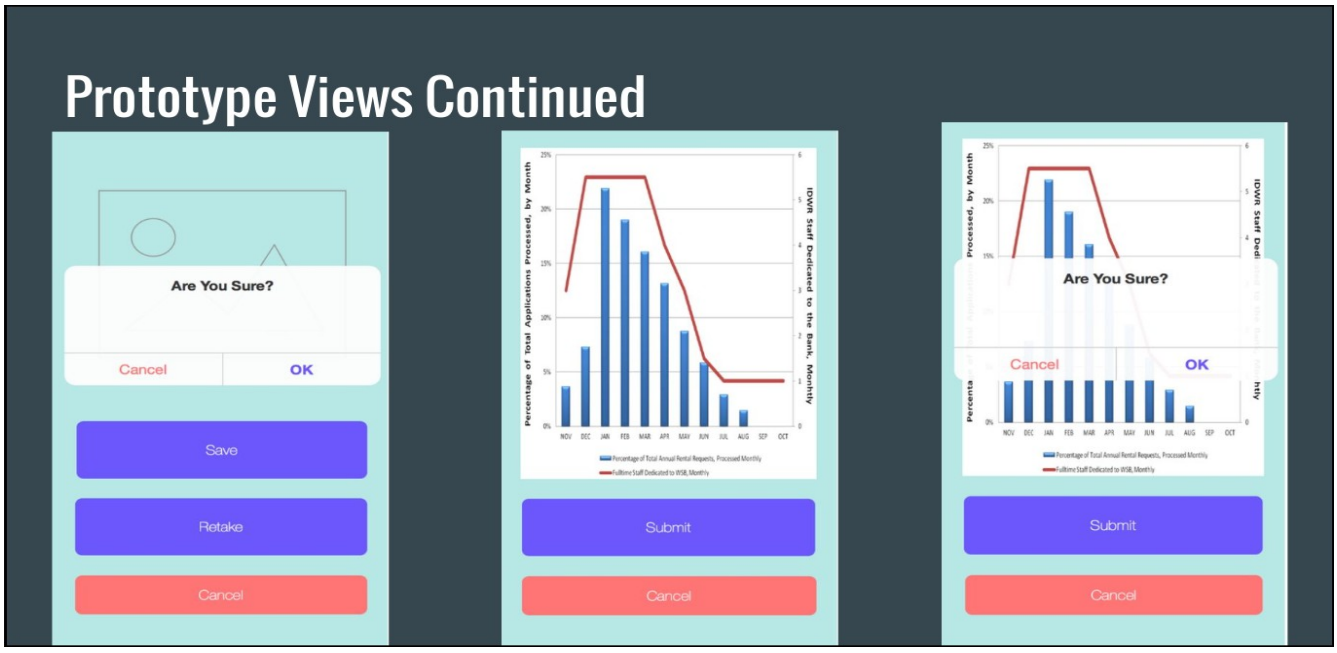
9. Help users recognize, diagnose, and recover from errors

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

10. 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.

Usability Problems

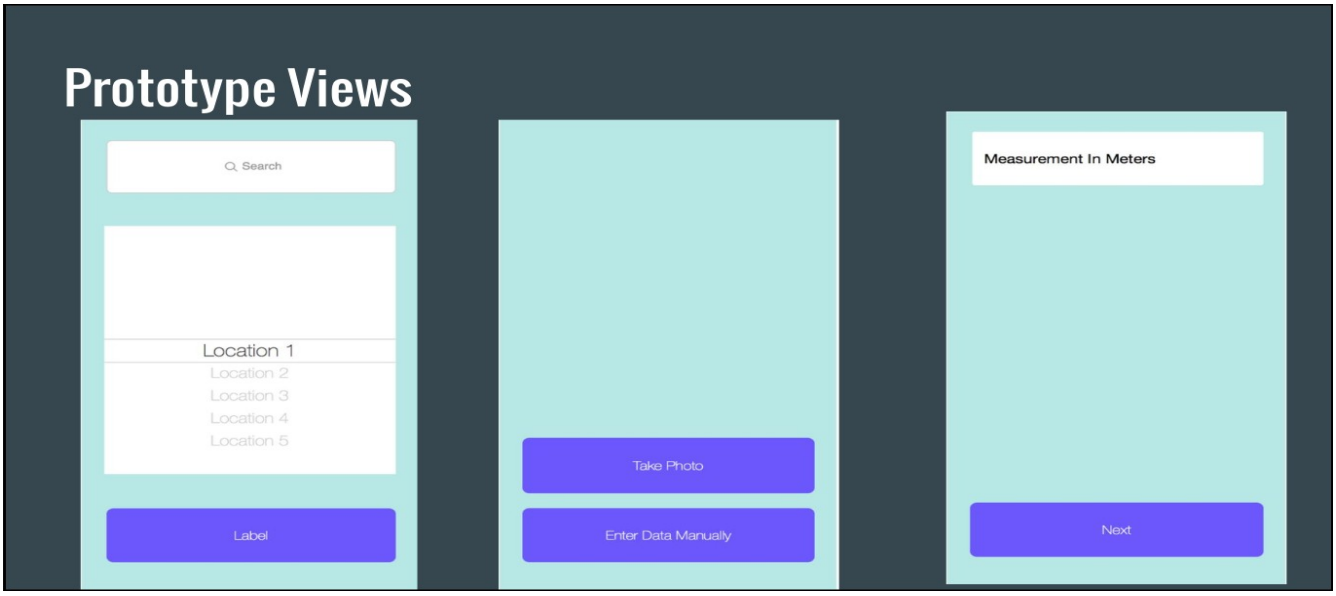


- In the prototype we can see that the graph which gives us the recent trends of the heights of staff gauge, here a user can't differentiate his submission with the other trends so its

kind of confusing for them to figure out which one is their submission.

- There is no scroll down menu for the application such that users can navigate to other pages directly. If there would be one it would be user friendly.
- How to confirm that my information is uploaded or not? There is no end page which says 'Thank You your form has been uploaded', if there would be a page specifying the message it would give confirmation to the users that their work has been uploaded.
- There is no specific filter to prevent the uploading of in appropriate pictures.
- There is no information page which gives the users an idea of how the pictures needs to be clicked and to correctly guess the height in meters when entered manually.

Critical Usability Concerns



- How can a user new to some place get to know the exact location where he is in? If there is GPS implemented in the application it would be for such users to directly give the current location.
- What if a user wants to enter information both by clicking the picture and manually? It would be better if there would be options for both picture and information in one form.
- What if a user wants to upload a picture from his mobile gallery? It would be better if there is an option to select pictures from his gallery to upload.

- The users may get irritated to confirm every page to navigate to next page. Rather it would be better to use a back button to navigate to previous page to make any changes if they have done something wrong.

Story

David Fernandes is on a trip to Maldives in summer vacation. He comes across some water body and finds a staff gauge in there. He starts the staff gauge application and tries to enter the location but as he is new to place he finds his location and checks the name of the place and then searches for the location in the list on the application. He then navigates to the forms page where he can select one of them but he's not sure which one to select then selects the picture form then clicks on the camera button and clicks the picture then clicks on next but he finds a confirmation pop up. He then selects 'ok' and navigates to next page there he find the recent trends of data but couldn't easily figure out which one represents his data as everything looks similar. Then he clicks submit button but he again sees the pop up for confirmation, which makes him feel that filling the form is a bit lengthy. After he confirms the submission he couldn't confirm that his information is uploaded or not. Without any confirmation he quits the application.