Evaluation Report 2

Heuristic Evaluation

Consultant: Kuber Dutt Sharma

(kdsharma@mtu.edu)

Team Dead Birds

Scientist: David Flasphor
djflaspo@mtu.edu

Team Details:

S. No.	Name	Role	Department
1.	Eric Vasey	Team Leader	Computer Science
2.	Rebecca Driver	Team Leader	Humanities
3.	Caleb Chapman	Team Member	Computer Science
4.	Jonathan Faron	Team Member	Computer Science
5.	Ryan Fenton-Garcia	Team Member	Computer Science
6.	Bradley Ross	Team Member	Computer Science
7.	Kuber Dutt Sharma	Consultant	Computer Science
8.	Haitang Duan	Consultant	Computer Science

System Description:

The undergraduate team aims at developing an app which would serve as a simple, convenient solution for documenting and identifying dead birds found, as well as analyzing causes for bird mortality, for David Flasphor. The app is a citizen science project, and the primary users of the app are people in any region. The primary users are tasked with collecting data about the dead carcass of the bird and nearby scenario by taking/uploading pictures as well as entering data manually.

UI Domain:

The application under development is a website that belongs to a data cataloging website UI domain.

Jakob Nielsen's Heuristic Principles

Following is a list of heuristic principles that apply to this application:

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

   The UI is form based. If the data about the surroundings cannot be fetched by the GPS, I suggest that the UI must have linked forms. The process of capturing data through these forms can be numbered or shown using '%' completed. By incorporating this, the user will always be aware of his progress.

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.

The forms must be interactive and designed in such a manner that the user does not lose interest in the application while entering information.

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.

   Some user of this app would be travelling on highways and stopping to enter complete information in one go would be difficult for them. The app must have a feature of capturing some data and saving it as a draft. So that at a convenient time the user could return to it and complete the data entering. Also features to over-write incorrect data should be provided.

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

   Designing structured yet interactive forms will be empirical for the success of this application. The forms would need to be designed in such a way that maximum amount of data is captured through minimum user effort.

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.

   The forms should be structured in such a way that the scope of user error is minimal. The developers have suggested providing pictorial assistance for the users. Even after this an error occurs while the user is entering the data, there must an option to rollback to some nearby checkpoint. This can be achieved by adding assistance features like auto correct and auto fill.

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.

   If the information to be entered is large, and it cannot be captured in one form, there will be need of connected yet separate forms. The developers need to keep in mind that similar type of information should be captured in one form. For example, data about bird death and nearby environment can be captured in one form. While, data about the road and nearby structures can be captured using a different form.

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.

Most user of this app will be novices. Only a limited number of fields should be marked mandatory, because we cannot expect a novice to know all.

8. Aesthetic and minimalist design and documentation
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.

I suggest that the application should have a small tutorial. The tutorial should auto-run the first the user uses the application. There should be a link to re-run the tutorial anytime the user wants. Since this is a citizen science application and the user base is so wide, a detailed yet crisp tutorial is essential for this application.

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.

Since the developers are using a pre-existing database for recognition of the birds, the system should prompt an alert message when a user selects a bird species which is barely or never seen in a specific area. The solution for such problems can be integrated in the system.

Usability Concerns and Suggestions

1. Even though the app can have both, registered and guest users, the option to login or to register on the website must always be available.
2. In most cases the device in use would be handheld devices, so the forms must be designed in a way in which they can capture maximum data. In the current prototype, the space occupied by the image frame is too big. The data captured through the form is vague and limited.
3. The current prototype does not explicitly define how the user adds the photos. I would suggest that the user should be provided an option to use the camera on the first form itself. So that they can start the data filling with the image in mind.
4. I would suggest that the app should prompt a message when the image is successfully uploaded. Since the bird carcasses decay fast. If possible the developers can program the system not accept images below a certain level of resolution.

Critical Usability Concerns and Suggestions

1. A major portion of the users of this application can be commuters who would not always find time to complete the form in one go.
The application must have an option to save progress or return later to complete the form.
2. Since the application is citizen science based. The user would be required to upload images for the scientist's observation.
If possible the application must have an option from letting people upload inappropriate images.

3. In most cases the user would be speculating the information about the scene, because of lack of expert subject knowledge.
I would suggest that the scientist be provided with two views of the database. One which is the user entered data. While the other can be entries that he or his team have verified and catalogued. This would make it easy for the scientist to perform any analytical operations on the data collected over time.
4. If need be the developers can drop the idea of guest users. As it will be easier to keep records for registered users. Along with the heat-maps the users should be able to look at all entries provided by them. While doing so if possible a glimpse of the scientists review must be made able for entries made by the users.
5. Since not many forms have been developed in the prototyping phase, I would suggest that the developers pay attention at the structuring of the forms and also make sure that the flow of information and user attention in between forms is smooth.
6. Currenlty the application lacks description of its purpose and its methodology. The application must have a contact us or about us column, so that the users understand the entire process and need of the data collection.

A short Story..

Kyle is a software professional who likes photography. He often travels to nearby places on weekends. During his travel he often observes bird deaths on the highway. He searches the internet for information regarding the same and comes across the dead birds app.

This weekend when Kyle was out clicking pictures, he again saw a carcass of a dead bird. Since Kyle was carrying his camera he thought of clicking a high resolution picture with it. He used his phone to go online and check his location for the website. He started a form as a guest user and entered data about the nearby environment and thought of adding the pictures later. Kyle continued with his trip and came back home later. Now the semi-completed form is available on Kyle's phone and he has transferred the pictures to his laptop from his camera. Since he started the form as a guest user there is no way he could use the same form on his laptop.

So Kyle had to register on the website and start a new form. On this form he used the information available on his phone and the pictures that were now on his laptop. After completing the form Kyle submits his observations.

In the past 3 months Kyle has now submitted 4 reports. He always wonders what steps are taken once he submits all this information. He also wonders if the observations he made in his report were precise or not. He wants to indulge into bird photography and wants to know more about the species found in the area. The ways to prevent their deaths. But after all the effort he is disillusioned.