A heuristic evaluation for ‘maple syrup app’ designed by team GoggleFox

For

Professor Robert Pastel
Michigan Technological University
Computer Science Department

By
Isidore Kafui Dorpenyo
Michigan Technological University
Department of Humanities
I am Isidore kafui Dorpenyo, a PhD student in the rhetoric and technical communication department at MTU. I am working with the gogglefox group. The document I present will seek to identify usability concerns of the “maple syrup app” developed by GoggleFox. GoggleFox is the name of the undergraduate group I am working with. The group has been tasked to design an app that can help calculate the economic value generated by maple syrup enthusiasts. GoggleFox is required to design a “mobile app where people could self-report their production levels, days with good sap-flow, total number of trees tapped, etc. would facilitate this data collection.”

**Description of the design**

In response to the task provided by Mr. David Kossak, the scientist they are working with, GoggleFox has designed a mobile app: maple syrup app. The app which consists of tools that can help users to check temperature data, number of taps, records of taps, the tree that is tapped, and yield calculator will help gather data on key production numbers and using those numbers to calculate production statistics. The app provides a simple how to guide and helps users to tap maple trees with ease.

**Identification and description of the UI domain**

The app is designed for Android devices of at least 4.0 and higher. The app as the group describes it “will optionally connect to the local GPS device to tag the location of the maple syrup data upload, if the user so desires…It will involve the users’ device connecting to the internet, either through a data plan or through a local wi-fi network. Most or all interaction between the user and application will be through the touch screen interface with buttons, text boxes, and menus.”

**List of heuristic usability principles**

This section of the paper presents some usability principles against which the app will be examined. Even though there are several of such principles, my examination will be based on the ten evaluation principles that Jakob Nielsen mentions in the book *Usability Engineering*. According Nielson, designs should:

- **Be simple and dialogue should be Natural**: User interface should be as simple as possible. The interface should match the users’ task in a natural way. In this regard, mappings between computer concepts and user concepts should be simple in a way that the users navigate through the system with ease. Information that will be used together should be displayed close to each other.

---

- **Speak the users’ language**: This principle maintains that terminology in user interfaces should be based on the users’ language and not on system-oriented terms. Dialogue should be in users’ native language and not in a foreign language. Interface should view interactions from users’ perspective.

- **Minimize user memory**: Interfaces should display dialogue elements to the users and allow them to choose from items generated by the interface and edit them. Displaying too many objects and attributes will result in a relative loss of salience for the ones of interest to the user. Therefore, designer should make conscious effort to match object visibility with user’s needs.

- **Be consistent**: If users know that the same command or the same action will always have the same effect, they will feel more confident in using system, and they will be motivated to explore more. Same information should be located at the same place on all screens.

- **Provide feedback**: The users must constantly be made aware of how system is interpreting their inputs. Feedback should not wait until system failure. Feedback should not be expressed in abstract terms.

- **Provide clearly marked exits**: Users should not be trapped in the systems. They should always find their way out. System response time should also be as fast as possible

- **Provide shortcuts**: Experienced users should be able to perform frequently used operations.

- **Provide good error messages**: Errors messages are important for two reasons: 1) They represent where the user is in trouble and particularly will be unable to use the system. 2) Messages present opportunities for helping the user understand the system better.

- **Prevent errors**: The system should avoid the error situation in the first place.

- **Provide enough help and documentation**: regular users may want documentation to enable them to acquire high level of expertise.

**List of usability problems**

You have a good design idea but some of your choices violate usability principles. Below, I provide a couple of such infelicities. On page 7 of your walk-through document, I couldn’t help but ask, where does the user go? What does he or she do when he/she is done with the tools on the page? This violates the principle of navigation. Users’ should not in any way be trapped in the system. Do you give users enough information to understand what the check mark is? Do all users understand what this arrow signs are? Think about how you help users to navigate the system.
Back arrows, though consistent, do not provide a natural dialogue for the user. It does not direct the user appropriately. I will say that consistency is one of the principles you violate a lot. If you are using the same color tab for items on the left, do so. In some cases you have squares and in others you have circles. Compare pages 7 and 10. You have a lot of colors. Alignment is also not consistent. On some pages you have items scattered all over. Page 12 is a tacit example.

Sometimes contrast is not strong enough. What is the purpose of the guide tab on page 11? Is it about how to use the app or about how to tap trees?

Since the sign-up page comes very late in your development, can I assume that one uses the app before signing up? Why does it come so late? When does the user sign up or log in?

“Synching” page is blank. What does the user do when he/she gets there?

**Identification of critical usability concerns**

I am afraid some of your usability goals might not be met. For instance, consistency is not achieved that much. Be consistent in the way you use colors, icons, and shapes.

What is that icon on page 8? I mean that little icon on the right corner of the tree type tab?

Do you anticipate the user encountering any errors? How can the user find his/her way out when he/she encounters an error? How do you give users error messages? It appears you assume that the user would be able to use system smoothly without encountering any difficulty.

Do you have any manuals or documentations for the app?

You don’t have any editing tools. How can user edit app when error occurs?

**Short story with regard to usability concern**

I go to the field on a very cold morning. I tap trees and I record. I save it and later I realized that I typed in the wrong information. I am looking for the edit sign. It is nowhere to be found. Well, I guess I have to keep the error because I am struggling to find the editing tool. The government statistician comes around to collect the data from local users and while he was compiling records, he finds out that some of the information does not help to arrive at the desired results. There is no way the goal of calculating how much maple is tapped by enthusiasts will be met.