Keweenaw Time Travelers

Group 0b100

Evaluation Assignment 2 - Heuristic Evaluation

2/18/2020

– Author –

Marissa Walther

– Team Members –

Ryan Ader, Kyle Alberth, Daniel Lambert,
Mason Sayles, Ruoxi Tian, Rebecca Barkdoll,
Christian LaCourt, Evan Monko, Manu Chemudupa

Undergraduate Design

The undergrad group's main task is to work on creating the search functionality for the pre-existing KeTT application. The search function will be added as a pullout menu when a user is looking at the pre-existing map. The user can expand that menu, allowing the user to input search criteria and refine the search with filters. The filters appear in separate expandable boxes below the search bar. Examples of the filters include year, record type, and location. Once the user inputs the search criteria and selects any filters, the results will appear below the filter area. The results will be organized into different categories such as people, structures, stories, and records. If no result is found, then an error message will pop up telling the user to double check their spelling. When the user interacts with a search result, the map may change to represent the location for that record and allow the user to interact with the map.

UI Domain

For this application, the domain is a GIS. The information is mainly presented in a map where different records and events have taken place over history. Once presented, the user is given the ability to search through these records and events and place filters on the results in order to find certain events.

Heuristic Usability Principles

Visibility: The search functionality should be able to be minimized to allow the map to not be covered. Likewise, the search functionality should be easy to find in the minimized state for the user to put in queries.

Feedback: The information that is presented when the user hits search should be able to be clearly presented below the search bar. Likewise, when the user hits different filters, the search results should be filtered accordingly to provide the correct results.

Affordance: The user should be presented with an option that when clicked, provides a help dialog to help the user know how to use the search and filter options. The search feature is designed to imitate other popular search features in how they work, allowing the user to also take that knowledge and use it with the application.

Consistency: The application will be consistent in its behavior. The search functionality only searches records in its database according to the user's criteria.

Control: The user has complete control over what search criteria they give the system. The system only executes the query after the user hits search.

Robustness: The user is presented with an error message when no results are found for their search criteria. The user will also get an error message if the search fails due to the API failing or other system errors.

Usability Problems

- The user might not know the correct spelling of a record when they are presented with an error message saying that the spelling is incorrect. This violates the feedback principle since the user doesn't know how to get to the correct record.

- Due to the large number of records, the application might not be able to handle the results in a timely manner. This would violate the responsiveness principle.

- There is not a clear all filters button on the form. This violates the user control principle.

Critical Usability Concerns

Lack of the ability to clear all the filters: This presents a problem since if there are a lot of filters selected, then the user has to go through the list and unselect every selected filter in order to make a new query. Say that Bob wanted to find all the records from Houghton in the year 1890 where all of the results should be employment records. If Bob wants to go and make a new query, he now has to unselect the filters for Houghton, 18909, and employment records. If Bob forgets one, then his future query would not provide the information he is looking for, along with the time that is needed to unselect everything.

Lack of suggestions when the user misspells something: This presents a problem since the user might not know the correct spelling for a record, meaning that the user will not be able to find that record unless they are looking at a full list. Say that Bob now wants to find a record for the street name of Montezuma, but can't spell it correctly. Bob spells it as Montzuma, and gets the error message that no matches were found. Bob now has

to either search for all the letters that he knows is correct, or try different search criteria to find his record.