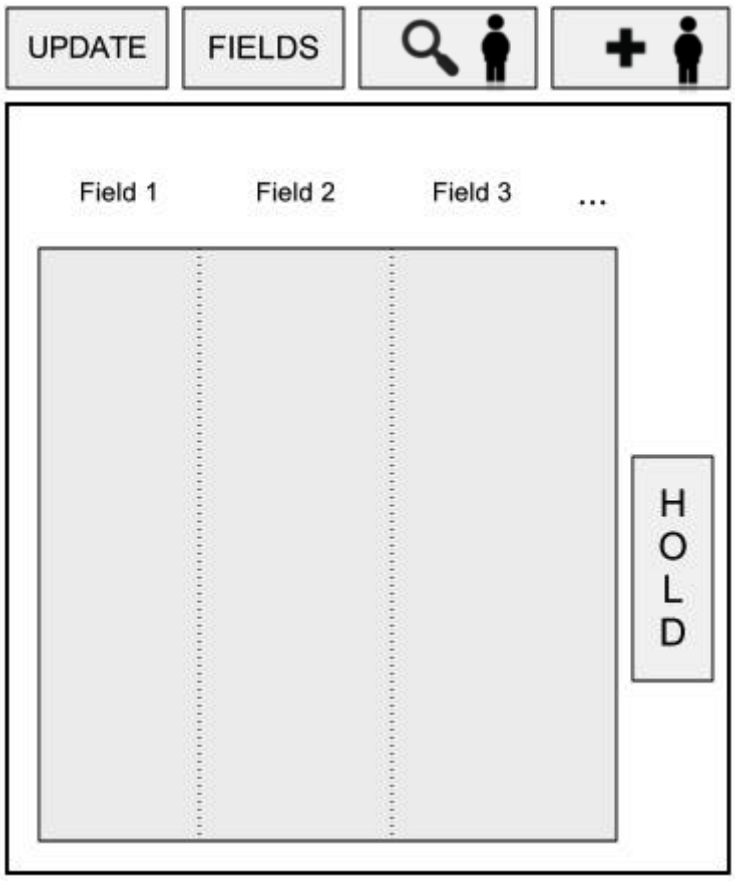
Headmaster Dream Design

The headmaster dream design is the design of an interface to add extra functionality to the Headmaster web service. This particular design is meant to add extra efficiency when updating or adding new information for new and existing students.

Top-Level Design / Layout



The button's description

This button is for adding new student tabs to the "Working field".

Search button for adding existing student tabs to the "Working Field"

Field button that triggers a drop-down. This dropdown would have check fields to select the fields that the user would like to give input for.

The Update Process

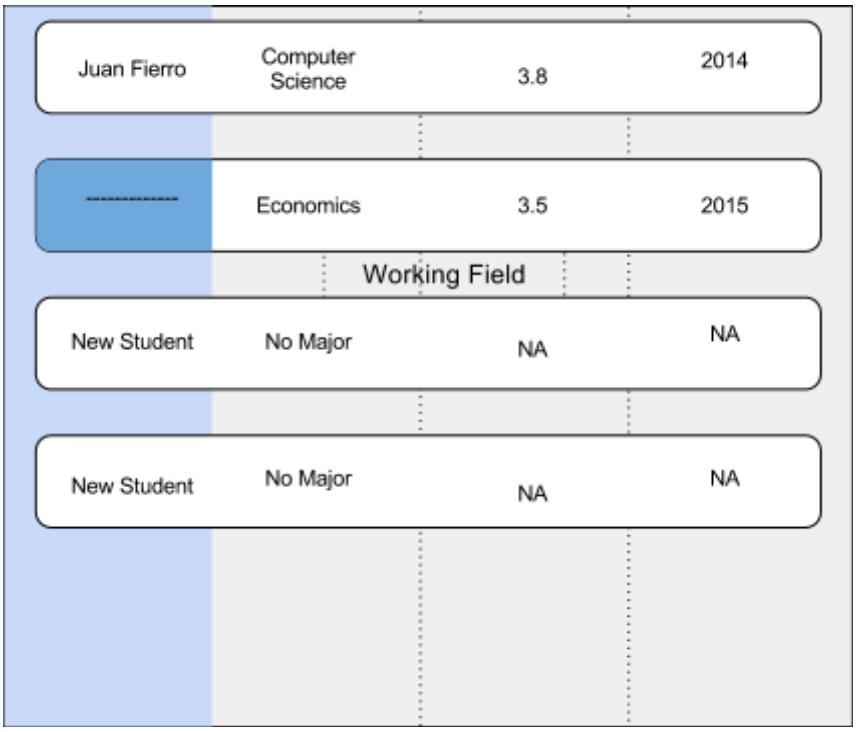
This button triggers the update process in the "Working field" as described below:

The white tabs are the tabs for created and new students. The blue column is the field that is currently being updated. The dark blue cell is the cell that is currently being filled with information from voice input. The update process follows the next steps:

1 The first row under the student tabs highlights in blue signaling that is being

updated.

2 The respective cells in the first column will highlight one after another waiting for the user to input the correct information by speech recognition.



3 The program will analyze the voice input and determine if it is good or not. For good input the program will go to the next cell. For bad input the cell will flash red twice and then go back to step 2.

4 The program will proceed through all the columns and rows till there are no more left to update.

The Hold button allows the user to input more than one expected voice command in the cell that is currently being updated. For instance, if the current student had two majors instead of one, the user could press the hold button while the major's cell is being updated to input two majors instead of only one.

Usage Scenarios

Here is an explanation of the scenarios in which this interface comes to play an important role of efficiency for the headmaster users.

Creating new students

This interface might be used for creating a list for the recently signed students that do not have a recorded profile yet. The fields will be selected by the user , and they will be updated to the created student profiles.

Updating existing students

This interface can also be used to update existing information from created user profiles. Users would be able to both, change existing information in already created fields and select new fields that have no saved information.

Creating and updating students profile

Choosing created students does not prevent the user from also adding new student tabs, which makes it possible to do both at the same time.

Design Rationale

Interaction Style

This design uses a combination of natural language entry and menu selection. This combination allows for a good level of consistency in the design due to the simplicity of it. The interface exposes the user to a minimal set of input actions which translates into little memory load for the user.

The menu part of the design breaks down the selection of students and fields . Natural language input take over when the user presses the update button, and it allows the user to input as much information as needed. Further the natural language implementation of the interface provides the user with the option of inputting more information than the default amount by pressing on “hold”.

Another important part of the design style is that it does not rely on a command language for the voice input. This prevents the user from running into errors due to command recall or voice initialization of the interface.

Stages-Of-Action

The interface mapping focuses on creating a mental image for the user that matches the user’s intention of update/create new students and input information for the students. The user knows he is creating a list of students because the list is physically appearing as he constructs it. Also, the allocation of tasks in the design allows for a very intuitive awareness of what action is taking place. The task are separated to create a complete sequence consisting of multiple and related steps as previously described.

More about the menu

Another relevant trait to point out about this interface is the use of skeuomorphic icons. This is used to create an easy interpretation of the task trigger and its promise behind it. The identification of this icons and the domain field they belong to, increase the predictability of the system making it easier for the user to interact with it. Further the color scheme is fairly simple so the user doesn't feel discouraged or distracted by the elements in the design when performing the task (avoiding color Blindness).

Usability metric analysis

Menu’s weakest metric

The weak metric of the menu part of the design is efficiency. The menus design of this interface is meant to have distinctive and clare elements that can be recognize easily, it is not focused on increasing rapid access to the task.

Menu' strongest metric

The strongest metric of the menu part of the design is learnability. This design is meant to excel the learnability and predictability of the interface. It relies on recognition instead of recall to make the manipulation of the menu more natural for the user. This allows the user to navigate easily through the menu to perform a given task because the interface objects fit the domain knowledge the user has.

Speech Recognition weakest metric

The weak metric of the speech part of the design is errors. This part of the interface uses speech recognition to accept input from users. At this point in time, speech recognition has not been perfectly implemented. This imperfection will lead to a number of higher error rates versus other input techniques.

Speech Recognition strongest metric

The strong metrics of the speech part of the design are learnability and efficiency. The design allows for users to easily recognize the current highlighted cell so they know where the current manipulation is taking place. The interface is meant for headmaster users. My assumption is that the headmaster users might no be proficient at typing on a keyboard which gives a relative advantage for them if they user speech recognition. This leads them to simulate a human to human interaction which is more intuitive than keyboard input.