

Headmaster Dream Design

Britain Southwick

Abstract

My design for headmaster combines menus, forms, and dialogues with a small amount of direct manipulation to create an ideal interface for administrative work. Some of the current functionality will be left in place, while many others will be changed or improved.

Contents

1 Introduction

Headmaster is a web-app designed for use in administrative work in schools, such as tracking a students grades, classes, gpa, etc. The user interface I have designed for it primarily utilizes menus, forms, and dialogues with a small amount of direct manipulation. The goal is to provide an easily learnable and efficient tool for school administrators to use without much training. I prioritized learnability and efficiency in planning my design, as the app is designed for use by administrators and office staff who may or may not have much in the way of computer training. Some aspects of the original Headmaster client were kept intact, while others were completely overhauled. Here, I will document the general features of the web app, crucial aspects of specific pages, and the motivations behind these designs.

2 Background, Preliminary, and Related Work

To aid in my design, I researched guidelines on user interface design for web app, as well as successful apps themselves. Some aspects are drawn from personal experience with features that I thought worked well or failed miserably.

3 Login Screen

The Headmaster login screen will largely be kept the same. I see no reason to tamper with such a simple design that properly conveys all of the needed features. The only change will be adding links under the username and password bars to reset or retrieve the username and password through a confirmation email. The username retrieval link will simply be labeled "I forgot my username," while the password retrieval link will be labeled "I forgot my password."

4 General Features

The primary mode of interaction will be menus, forms and dialogues. The large majority of buttons and menus will be labeled with text, rather than icons as it improves learnability. The use of menus, forms, and dialogues in of itself is a sort of affordance, as it conveys to the user that the software is intended for work, not fun or creativity (at least more so that direct manipulation or speech control would). The presence of buttons, forms, and a lack of skeuomorphic design and features with no purpose conveys would remind users of other business tools, such as Microsoft Word, which primarily uses menus, forms, and dialogues, or even other school administration systems, which seem to be largely based on menus, forms, and dialogues. It is a common interface style that is familiar and free from the distractions that excessive direct manipulation can cause (people like to play with them).

Keeping in line with Dmitry Fadeyev's article on user interface design, that suggest that all interface features be available, but simplified, I will feature a search bar at the top of every page in the app that will allow you to search for students or for features of the web app itself [?]. A drop down menu on the right side of the search field will show what domain is currently being searched in, namely students, features, or everything. For instance, if a user could not remember where to go to create an event, they could type in "create event" into the search bar and select features and the search bar would display a drop down list of suggestions that could be clicked on like links.

5 Student Pages

Another major change is that the drop down menus for the Students, Events, etc. will be replaced with two buttons that toggle the state of the app. One button will be labeled with something along the lines of "Work by Student." Toggling this button will display a box empty the middle section of the screen and display a box with a search bar in it to search for student. Under the search bar will be 5 side by side check boxes that are labeled "Senior," "Junior," etc. Checking one will display all students of that year below the control box labeled with the appropriate year at the top. Checking an additional box, or checking the box marked "All," will create a border around each block of students and label it appropriately. From here a specific student can be selected and modified much like the current version of headmaster. Finally there will be a button labeled create, which will open a new tab on the students pane to create a new student.

This brings us to another change in the student management pages, which the ability to have multiple student pages open at once in the form of tabs, much like a web browser. The tabs will be displayed at the top of the page, under the toggles and search bar. Much like Google Chrome, the tabs can be dragged and rearranged to give the user a degree of customization. Each tab can be closed by clicking on an x button at the top of the tab, which will prompt the user to confirm that they want to close the tab.

What these two design elements do is allow the user to perform various tasks in sequence without having to repeatedly navigate through countless pages and endure the load times on each page. This way, the home page is always available by switching tabs and student pages can be quickly pulled up. This should boost efficiency and possibly learnability, as it is similar to web browsers many users are familiar with. It will keep all the same hallmarks of browser tabs, like the tab itself looking like a folder tab (an affordance that points to its function as a device for thumbing quickly through multiple pages).

The student edit and creation pages themselves will be more or less the same, as the current layout provides all the necessary functions, gives clear indication as to what each section is, and hides irrelevant information to keep the screen from getting cluttered. Only some small changes, like a calendar widget for absences that marks on the calendar each day the student was absent. The current design is very much inline with the various user interface guidelines and provides an efficient way to manage students. The only complaint I have is that it is not obvious at first that the different sections are drop down menus, but a small skeuomorphic affordance, like a slight crinkled or accordion look at the bottom of each button may better show their purpose. Or at the very least an arrow pointing down to signify expansion. Additionally, the Grants pane will be linked to the Grants page so that a student can be added to a grant from their own page as well as from the grant's page. An Events pane will be added, which lists events the student has attended on a calendar widget that can also show absences and other important dates for that student. Just like with grants, students can be added to events from their page in addition to the Event page.

Last but not least, I would add a safeguard to saving changes that first asks for confirmation before saving and then disables the button until further changes are made to prevent repeated submissions of the same stuff. This follows the guidelines proposed by Dmitry Fadeyev [?] and Smashing Magazine in a different article on user interface design on web app [?]. Both articles stress having a forgiving interface, which helps prevent errors. By both forcing a user to confirm submissions and disabling further submissions of the same thing, errors caused by misclicks will be drastically reduced.

6 Features Page

The other toggle button on the top of the screen will activate feature mode, which will allow the user to modify one aspect of a student's information rapidly for many students. Clicking this button will function similarly to the student button in that it will present a search bar to find features. Instead of checkboxes, it will display a bunch of buttons that expand like the sections in the student editing page when clicked on. Each button will be labeled with the description of the features it contains, such as "Events," "Grants," and "Student Management." The search bar at the top will have a drop down menu on the right that can be used to select Events, Grants, and everything else that is currently searchable in Headmaster.

Just like the student toggle, the feature toggle will allow users to open multiple tabs in order to better manage efficiency. Again the pages for creating and editing a grant or event will be largely unchanged since they do a perfectly reasonable job already. They are concise and clear and get the job done. The big change here is that when you click on a student management feature, such as GPA or Major/Minor, it will open a tab that will have the editing fields from the student editing page appropriate for that feature on the right hand side and a list of students like the one from the student main page on the left. Clicking on a student name will display their relevant information in the editing fields, which can be changed just like you would on the student's page. This way, workers can quickly change one aspect of many student in quick succession. For instance, when grades come in at the end of the semester (provided teachers don't upload them directly to the server), the workers can just select the Grades button and go through and update each student's grade without having to go through each of the student's pages. This will greatly improve efficiency and allow for workers to get through a much higher volume of students in a short amount of time.