Scheduled!: An Integrated Scheduling System for Students

Ming Cheng

University of California, Davis

Davis, CA 95616, USA

mgcheng@ucdavis.edu

Varun Peesapati

University of California, Davis

Davis, CA 95616, USA

vpeesapati@ucdavis.edu

Ajay Miranda

University of California, Davis

Davis, CA 95616, USA

avmiranda@ucdavis.edu

Jiayi Xu

University of California, Davis

Davis, CA 95616, USA

jyixu@ucdavis.edu

Abstract

This paper describes the entire process of building the integrated scheduling system "Scheduled!" from an initial idea conception to a functional high-fidelity prototype. The app "Scheduled!" aims to help instill some semblance of order and clarity into the daily lives of our users. The scope of this app is currently restricted to that of college students in UC Davis. The initiative for building this system sprang up, upon conversing with some college student friends about their scheduling systems, and because of a necessity to help declutter the lives of these college students by reducing the already extreme coursework burden imposed on them when they must schedule their day. Systems like Google Calendar are not effective in this sense currently. The app is in the form of a website that can be easily accessed and viewed by students through their computers or their mobile phones.

Introduction

College is the first place where students get firsthand experience with skills that are essential for living in the outside world. Also, for most students, it is their first time living outside the comforts of their home. As such, they need to learn to organize themselves much better so that they get work done on time. However, for a student, the most important thing needed in an organizer would be their class schedule and a to-do list

containing the list of all the homework that needs to be completed on time and it is nearly impossible to find such an organizer on the Internet today. Even if we consider UC Davis students, they can only view their schedule on Schedule Builder and if they ever want to look at their to-do list, they would need to open Canvas on another tab. This can make the student to feel disoriented and at times, would even leave them frustrated. Therefore, to tackle this problem, we've developed a website called "Scheduled!" which integrates features like a student's schedule, their reminders, etc. into a single webpage.

The website is roughly divided into four different dashboards, each with its own functionality along with a navigation tool in the top-right corner of the webpage that allows the user to easily navigate through the page.

The first of the four dashboards, simply called as "Schedule," allows the user to enter and save their schedule for that day. The saved schedule will also be displayed on the same page. To enter a certain class into their schedule, the student only must enter the course, its location, the type of class (for example, is the class a lecture, discussion, or lab), its starting time, and its ending time. The user is also given the option to edit or remove a certain class from their schedule, giving them the liberty to make any necessary changes to their schedule.

The second dashboard, called "Weather," allows the user to check the weather outside, with the help of Google's weather API, and displays information related to it. The purpose of the "Weather" dashboard is to help the student to plan their day according to the weather outside.

The third dashboard, called "Reminders," allows the user to create a to-do list for formal tasks such as homework and related assignments. Like the "Schedule" dashboard, the users are also given the option to edit or remove a certain task. The "Reminders" dashboard also allows the user to set a priority level to each of the tasks, and depending on the priority level, that task would be displayed in a different color. For example, if a student had saved a certain task as a high priority task, then that task would be displayed in red. The reason for having a color-coded priority level system is because people perceive some colors in a similar fashion. For example, people perceive red to be dangerous or urgent.

Finally, the fourth dashboard, called "Sticky Notes," allows the user to save more informal tasks such as "taking out the trash" or "go for a haircut." The user is, again, given the option to freely edit or remove a certain task.

Background
The idea to create an integrated scheduling system came when we had inquired some of our friends, who are all college students, if they ever organize their day with the help of an organizer like Google Calendar. Surprisingly enough, their answer was that they don't. When asked why, they simply replied that organizers like Google Calendar don't have the feature to save their schedule for the quarter and as such, they would have had to access Schedule Builder whenever they want to plan their day.

Conceptual Model

A conceptual model of a product is the model of the product that the designers want the users to understand. The conceptual model of the system consists of the four main functionalities that are viewable on the webpage. Table 1 is a conceptual

Table 1: Conceptual Model

Objects	Attributes	Operations
Schedule	Class, Type, Starting time, Ending Time, Location	View, Add, Remove, Clear all
Weather	-	View
Reminders	Formal Tasks, Priority Level	View, Add, Edit, Remove, Clear all
Sticky Notes	Informal Tasks	View, Add, Edit, Remove, Clear all

model that we would want our users to understand.

Prototype/Implementation

In UX Design, the goal of a prototype is to test products and product ideas before spending time and money developing the final product. In other words, prototyping helps the designer to finalize on ideas before moving onto developing the final product. As such, we began designing our website by first sketching our ideas on a piece of paper and created a low-fidelity prototype.

Figure 1: A sketch of our initial ideas as a low-fidelity prototype

After finalizing on the ideas that we seemed were practical, we created a wireframe prototype using Adobe XD, that served as a foundation for our website's core visual design.
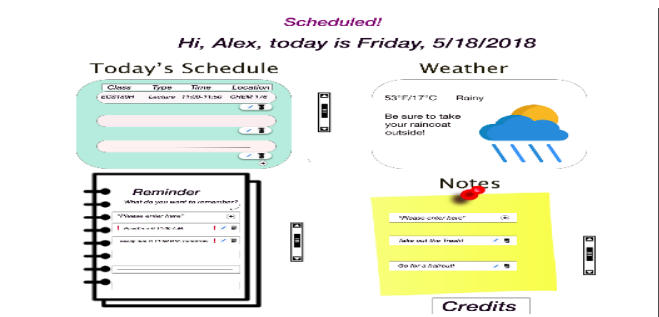

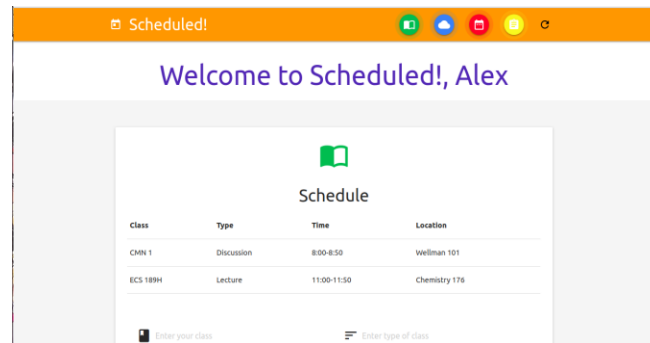**Figure 2: The wireframe prototype that was created based on our low-fidelity prototype**

Figure 3: The high-fidelity prototype that was developed based on our ideas from the low-fidelity prototype

We, then, proceeded towards developing a high-fidelity prototype. By definition, a high-fidelity prototype is a prototype that simulates the final product's functionality and design details. In order to develop a high-fidelity prototype from our low-fidelity prototype, we used web programming languages such as JavaScript, HTML, and CSS along with a Google Material Design wireframe that served as a template for our prototype. Figure 1 is a sketch of our initial ideas as a low-fidelity prototype. Figure 2 is the wireframe prototype that was created based on our low-fidelity prototype. Figure 3 is the high-fidelity prototype that was developed based on our ideas from the low-fidelity prototype.

Discussion

Upon presenting the high-fidelity prototype to our professor and TA, the project was met with much satisfactory response along with plenty of suggestions to improve our design in multiple areas. Some of these included incorporating a slow form animated feedback to direct the user to the intended section of the web page based on which button the user selects, and a help dialog box on the navigation bar that helps the user clarify the functionality of each button. Throughout the entire duration of this project, we learnt a lot about the design process, creating conceptual models, and creating both low-fidelity and high-fidelity prototypes. Also, the project provided us with a wonderful opportunity for all us to learn more about HTML and web programming in general, and which gave us more insight on web development.

Future Work

Though our high-fidelity prototype allows for user-testing, there are still multiple areas that need to be fixed. For starters, instead of having the user to manually enter and edit their schedule daily, we could have them enter their entire schedule at once and depending on the day, the website will only display the classes that are on that particular day. Also, in the "Weather" dashboard, rather than only displaying information about the weather, we could also print a message that differs with the weather outside. For

example, if the weather outside is sunny, we could have the program to print a message like "It's a great time to go out and play!" This would not only help the user to know about the weather, but will also help them to be more prepared in advance. As suggested by the professor and the TA during the demo, we can also work on making the navigation bar easier for the user to understand by including a help button and can also include a slow form animated feedback to direct the user to one of the dashboards whenever one of them is pressed. Also, instead of displaying the schedule as a table, we can display it as a calendar, where the user can see their schedule and plans for any given day.