Medication Tracker

Intro

After deciding to take this course and realizing the semester long project, I began contemplating what I could do immediately. I had a lot of high ambitions and good ideas for what could be a very successful application, but I definitely needed to keep in mind what I could do with the given time in the semester. Going through many ideas, I landed on a Medication Tracker.

There are tons of people in this world who rely on multiple medications to take during a week. This gave me a broad user base that I could reach, along with helping people stay healthy. Personally, my grandfather has diabetes and has a ton of medications to take during the week. As he ages, remembering all of his daily tasks has become difficult, this application, that he tested, ended up getting his approval. With this type of personal motivation I was able to really work to get something adequate, and low-and-behold, I did!

Problem, Users and Tasks

Medication tracking can become very complicated and time consuming, especially for people who have medical conditions. This app strives to alleviate stress for people taking medication through the use of the scheduling features and task descriptions for each drug. This will help the users because throughout the day they can check to see which medications they need to take and when. One example of a task description would be the user needing to eat when taking a certain drug. These features will make tracking medication easy for all ages and ensure healthy living.

The primary users for this tool would be everyone that needs to take medicine. This means technically everyone could be a potential user at one point in time. The primary users of this tool would probably be older adults who may need extra assistance in remembering schedules because of either loss of memory or simply because an extremely busy schedule. Because this tool is rather general in its functionality and the user inputs the information needed, there wouldn’t need to be specific information sought out from medical professionals for development. Although specific information on drugs isn’t needed from professionals, advice on design and functionality could be very helpful information. Secondary users would include these medical professionals who could recommend this application to their clients. Essentially, anyone who is giving out medication to clients would be a secondary user. Having a clear knowledge of these users is integral to creating a sufficient application that will work for all the stakeholders.

The main task for this tool will be adding pills to the application. Within this process there could be many different types of functions, but I will focus on the schedule of when the medications are to be taken and a description for the pill. Because these functions are correlated I believe a single task hierarchy with two separate paths makes most sense. The home screen section is a little redundant but I wanted to keep it in for presentation purposes. I essentially used it as a title screen so people knew what application they were dealing with upon opening it up.
1) The user opens up the application
2) The user then can add a pill
   a. They can specify the day to take the pill
   b. They can specify the time to take the pill
   c. They can add a description to the pill
   d. They then submit the pill and it’s added to the schedule
3) Once they “add the pill” to the application they can see the metrics on the grid format and either close out, or keep adding other pills

**Design**

I went through two different types of main designs through my development process. The first turned out to be a little too ambitious to be completely effective, and I eventually, toned it down to be more compact but still meet the requirements. The first solution I worked extensively on, but realized after some time, that it would be best to rework the design so the application would be more efficient. If I had some more time maybe the first one could have the potential to be a slightly better application, but regardless, both met the requirements. I will go over each solution in detail and cover the differences between them.

The first design consisted of many different forms to complete all the different requirements of adding medications, managing them, and viewing them. The main screen had 3 options to add, manage, and view the calendar. The first view to take into consideration is the “add a med” form:
This was the initial design I had throughout the semester that I thought would work quite well. This still achieves everything I wanted for this form, but as you can see it is a little unorganized and doesn’t look quite as good as it probably should. If this were my final implementation I would have fixed it up a bit.

Take into consideration I planned on using military time, avoiding needing to include the am pm functionality. Something that this type of form does that my other doesn’t is being able to add the same medication for multiple days. As we will review later, with the second design, adding a medication can only be done once at a time per consummation.

The second important view from my first design iteration is the calendar I attempted to implement:

The main issue I had with this view was getting it to work efficiently. After doing some more logical thinking about the entire design of this application, and through testing, I realized the next and previous buttons were not relevant. When dealing with medication scheduling, one only needs a weekly view mainly because there is the ability to remove a medication schedule. If for some reason I thought having a view that needed multiple week functionality, I would rather have a full monthly view and still keep this weekly view without the next and previous buttons.

The third relevant view was the “ManageMeds” feature:
As you can see I messed up on a few areas on this screen, mainly because it wasn’t a finished product yet. The functionality is pretty straight forward, but the recurring fact, of having all these different views making the application bloated and slow. I was able to achieve all of this functionality in one main view in my second iteration. In doing this, it reduces the amount of mouse clicks the user needs to achieve all of the relevant functionalities. This concludes the main forms of my first design iteration, although meeting all my requirements; it was a little bloated to be very effective.

My final design iteration that I used for my final project was held in one form. This one form still was able to meet all of the requirements with adding, managing, and view medications:

This view achieves all my requirements in one view, rather than having to click around and open up multiple windows. I would have liked to achieve another weekly calendar view, or figuring out a sorting function for this view, but at least I met all my requirements. At first, I thought having the *add* and *remove* buttons below the *schedule day* and *description* areas would be most logical, but after testing, my users concluded they would rather have them farther apart. They complained of accidently clicking the *add* or *remove* buttons which was messing up their process of completing the tasks. Although it looks a little odd having them spread out as such, at least this way, they will not make mistakes. One may be
concerned with the small grid area, especially for viewing the entire description. I ensured that one could resize the column dividers between the metrics, so that if needed, they can move things around to read everything. This implementation was lightweight, and very effective, overall I am satisfied with this application.

**Implementation**

The main implementation issue I had between the two UI designs was either creating a backend SQL database or saving it to a file. I didn’t think creating a database would be sufficient for this project because I needed to send the UI around for other people to use it. Because of this I would need every user to create the SQL database on their machine before using my application every time. This was the main issue I ran into with my first design; I couldn’t carry over the information to different views when only saving to a file. This led me to create the new design that saved to a file and still achieved all of my requirements. After realizing my limitations with the first iteration, I think I actually came up with a better solution for my requirements. Having everything in one view makes it easier for the user to achieve everything he/she would be looking to do. The final implementation was extremely quick and efficient, it definitely worked with my requirements well.

**Evaluation**

After using the various rounds of testing, one can really see how the iterative process really helps the development process. Heuristic, paper, and user testing each can arise multiple different types of issues that help make UI’s more efficient. I thought the paper and user testing were the most helpful testing scenarios. Both are done when the developer is there and can view specifically how the users go through the tasks and what needs to be fixed. During the paper testing I realized that the previous and next buttons on the weekly view were not relevant and could be removed. I also forgot to put a back button on the calendar view of my first design iteration; obviously, a careless mistake but thanks to the testing process it was found. With the new iteration, the main design change that the users gave me was moving the two buttons. This corrected mis-clicking mistakes, so that the user had to navigate to the button that ensured they were performing the task they wanted.

During these processes most of the design is implemented so the developer can really receive some great feedback to correct mistakes. I believe with all of these testing scenarios they were able to efficiently hash out all my potential issues with my UI design, and really helped to get sufficient feedback from users.

**Reflection**

The most I learned over the course of the semester was the different ways to effectively test applications. After each testing scenario I was able to go back and slightly edit the interface after feedback from the testers. This iterative process really gives developers the ability to make sure the final product is refined. Having this knowledge is key for future developers because this leads to projects being successful.

The main thing I would have done differently if I were to do this again, would be to start the strict development of my application sooner to get a better idea of what was possible and what wasn’t. When dealing with only design, there are a lot of ideas that I had that would be successful, but didn’t realize until after starting development that some of my ideas were not realistic. If I were to develop sooner, more along the lines of my prototypes, I could alter my design more as the process went along. In doing this, I could have gotten more feedback on my updated UI.
In conclusion, I created a sufficient application that met my requirements. I used the evaluation techniques to find what was the best way to go about creating this application – all the testing scenarios helped immensely. This course did a really good job to create a real world scenario of pushing out an application. From what I have experienced, the iterative process was replicated very well from the real world into the classroom. I can confidently say I learned a lot from this course, and am very pleased with the information I received during the semester.