Task Management System

Introduction to Programming

CMPT 120L

TLMB

Marist College

School of Computer Science and Mathematics

Submitted to:

Dr. Reza Sadeghi

Fall 2023

Project Progress Report 1 of Task Management System

Team Name

Name of the team **TLMB**

Team Members

1. Lance Perdue Lance.Perdue1@marist.edu (Head)

2. Timothy Ford Timothy.Ford1@Marist.edu (Member)

3. Matthew Brana Matthew.Brana1@Marist.edu(Member)

4.Blaine Chesnut Blaine.Chesnut1@Marist.edu (Member)

Description of Team Members

1. Lance Perdue

My name is Lance Perdue, I have approximately a year of java experience through a dual enrollment program my senior year of high school. I have worked with these team members throughout the entire semester and we have been successful in all of our projects thus far. The more we work together, the more efficient and productive our communication has become. I feel that working with these members will continue to be successful in this final project and our experience working together will give our group an advantage. I was chosen to be our team's leader primarily for my experience using GitHub in the past, while my knowledge is limited, it still is useful in allowing our group to work efficiently using this new tool.

2. Timothy Ford

My name is Tim Ford and I am a freshman at Marist College pursuing a Bachelor of Science Degree in Computer Science. I have prior experience in both Python and more recently Java through AP Computer Science A during my high school education. I chose to stick with my team members that I have been with through the entire semester as I believe we have worked well together in the past and will only continue to grow as a group throughout this project. We chose our group head to be Lance Perdue because of his experience with Github in the past and also his willingness to step up and lead the group.

3. Matthew Brana

My name is Matthew Brana. I am a freshman at Marist and my major is Computer Science. I have an associate's degree from Brookdale Community College in New Jersey in Computer Science. I chose to work with my team members because we have worked together all year and our group assignments have gone smoothly. We decided on Lance Perdue to be our leader because he has the most knowledge of GitHub and set up the repository for our final group assignment.

4. Blaine Chesnut

My name is Blaine Chesnut and I am from Morristown, New Jersey. I am a freshman at Marist College and majoring in Data Science and Analytics although I do not truly have any background in working with computers outside of using them for school assignments. I have always had a love for analytics stemming from my love of sports. I wanted to work with this group since we have completed all of the prior group projects together. We work extremely well together and always succeed in our work. We chose Lance as the team head since he is very intelligent and a great leader.

Table of Contents

Table of Figures

Figure 3:

Figure 4:

Project Objective

A task management system (TMS) displays a calendar for the desired week, month, or year. Also, TMS organizes personal tasks of different users on a specific day. THe users should be able to see their individual calendar data and update them. Your TMS will store the data of different user types in distinct CSV files. This system should at least support the following items.

1. Admin user is capable of:

    a. Having admin user and password for login (a string of at least 8 characters

    b. Changing the admin user and admin password

    c. Adding a normal user to TMS by creating a new username, password, and corresponding recorded data

2. Each user should be able to:

    a. Add a task to TMS the task contains: title, time, duration, and description

    b. Remove a task

    c. Edit a task's details

    d. Search through TMS based on time, title, or duration and list the results on the screen. For instance, it should be able to list all scheduled works for one day.

3. TMS should be a user-friendly software, such that:

    a. it shows a welcome page and provides a menu of all functions to the user in all pages

    b. It illustrates the reports in a tabular form. For instance, it displays a well-organized calendar of every month, or year.

c. It shows a warning if a user tries to input contact information with a name that exists in the history.

d. TMS should provide an exit function and thank the user for using the software.

4. Optional: TMS should protect the user information, such that:

a. TMS passwords and the recorded information should be ciphered. In the simplest case, you can use the Caesar cipher methodology. The easiest way to understand the Caesar cipher is to think of cycling the position of the letters. in a caesar cipher with a shift of 3, a becomes d, b becomes e, c becomes f, etc. when reaching the end of the alphabet it cycles around, so x becomes a, y becomes b, and z becomes c.

GitHub Repository https://github.com/Taco956/CMPT-120-Final-Project

User Experience (UX) Design

When designing the user experience, we decided to start the user on a Login window. Upon entering their information, the system will determine if they have a valid login and direct them to a success popup, if they have a valid login as an admin and directed to an admin success popup, or are using an invalid login which will just route them back to the login screen. A successful admin login will bring the user to an admin window where users can be added or removed and the admin user can be updated, or the main menu can be accessed. A successful regular login will direct the user to the main menu. From the main menu, they are given the option to exit the system entirely or to access one of the four functions of the TMS. Each of these functions, Add, Remove, Edit, and Search, all allow the user to use their specific function, to return to the main menu, or to exit the system in its entirety.

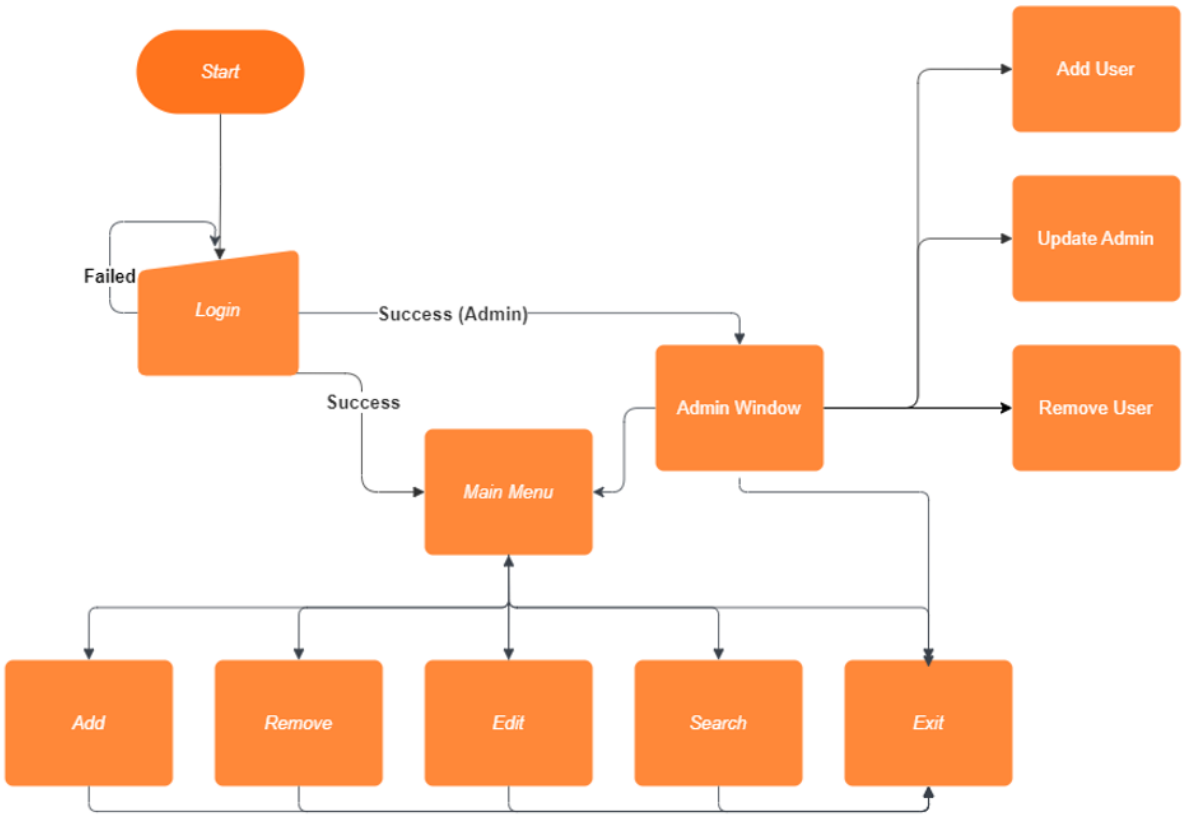

Figure 3.1: Flowchart showing the design of the User's intended experience

User Interface (UI) Design

Figure 4.1: Login Window

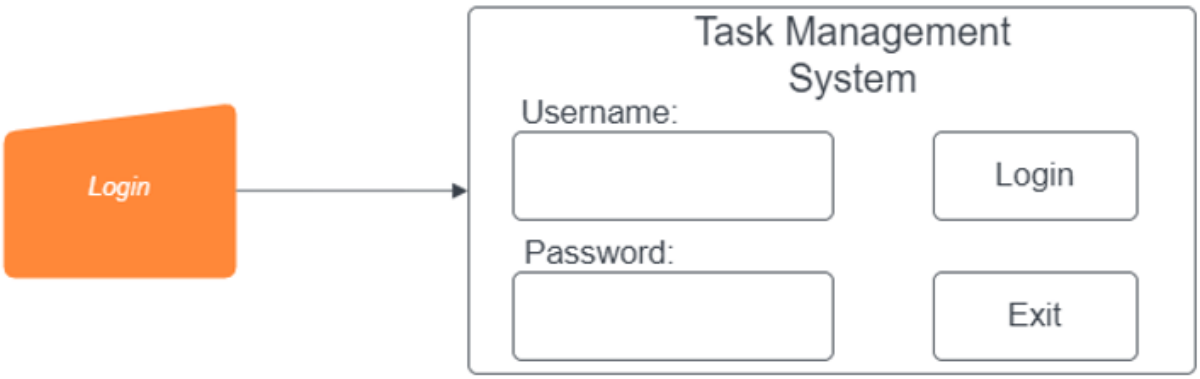

Figure 4.1 is the first window that will be shown to the user upon running the program. This login window has two entry boxes, one for the username and one for the password, and two buttons, one to attempt a login and one to exit the program.

Figure 4.2: Login failed popup

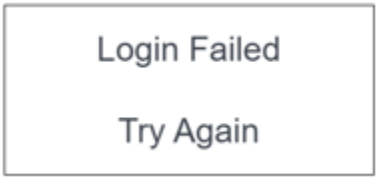

Figure 4.2 is a window that will quickly pop up whenever a user tries to login with unauthorized information, they will then just be directed back to the login window so they can try again.

Figure 4.3: Admin success popup

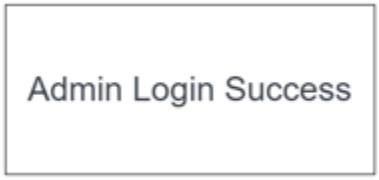

Figure 4.3 is the quick popup that will flash after the correct username and password are used for the admin login. This window will close on its own and then redirect the user to the admin window.

Figure 4.4: Admin Window that appears upon admin login

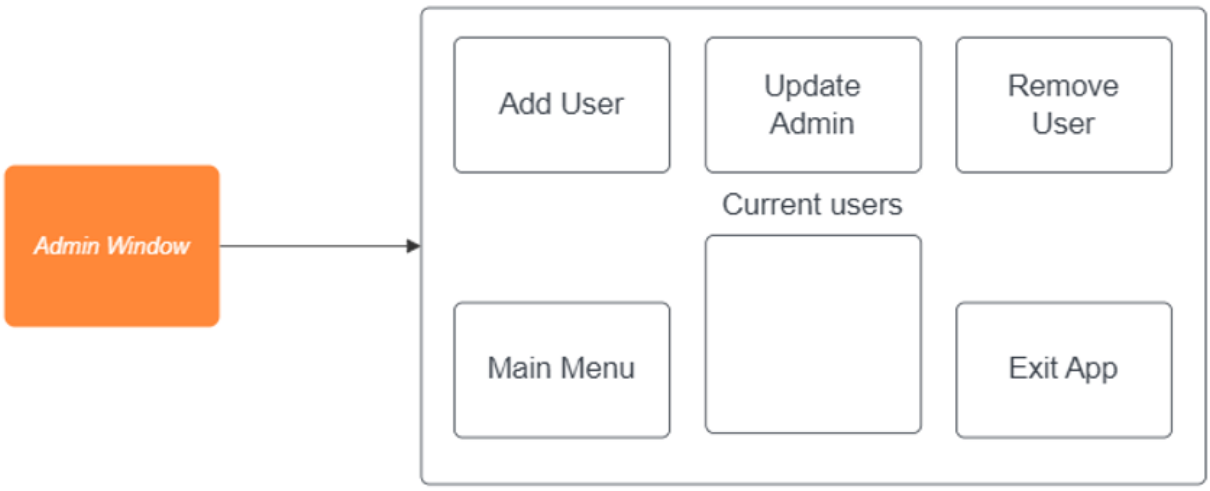

Figure 4.4 is the admin window that will appear after the popup redirects the user. It is made up of 5 buttons that allow the user to access the add user function, the update admin function, and the remove user function as well as to move to the main menu or to exit the program completely. There is also a text box that will show a current list of all the users that have access to the TMS.

Figure 4.5: Add User function from admin window

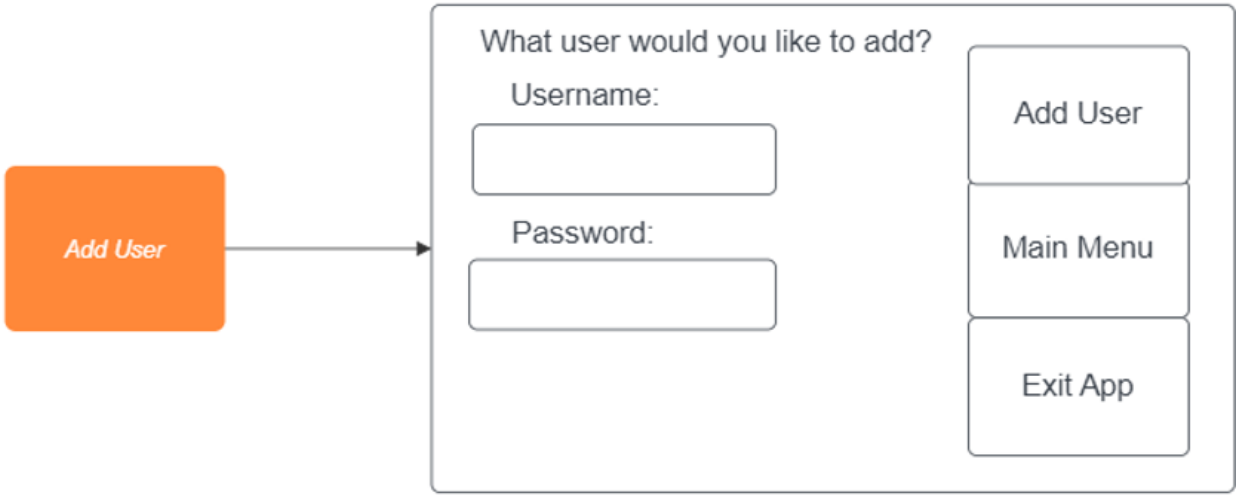

Figure 4.5 is an extension of the previous admin window, once clicked the user will be directed to this window with two entries and three buttons. The admin can add a user by entering the requested username and password and can be fully completed by then pressing the add user button. Once completed a non-admin user can use this newly created information to successfully login and store their own data through the task management system. This window also includes a main menu function which will close this window and redirect the user to the main menu. Lastly, this window contains an exit app button to allow the user to seamlessly close the application entirely.

Figure 4.6: Remove User function from admin window

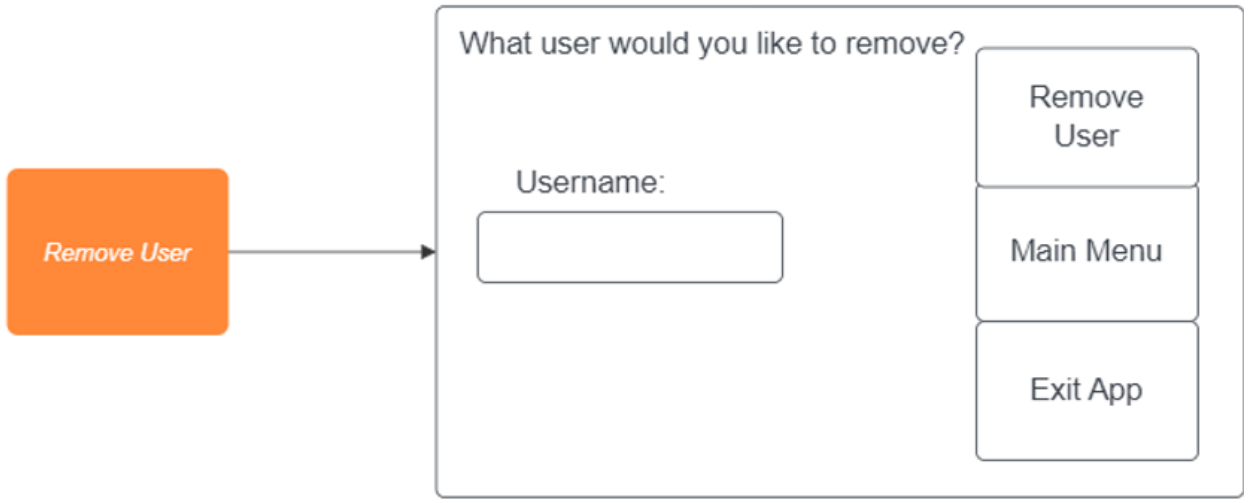

Figure 4.6 is another extension of the admin window which allows for the admin to remove any user other than the admin itself by entering their associated username in the provided entry. For this action to be completed the user will have to press the remove user button to finish the process, after which, that user will no longer be able to login and store information. This window also includes the same main menu and exit app button as the previous figure, allowing the user to navigate the application with little confusion or frustration.

Figure 4.7: Update Admin function from admin window

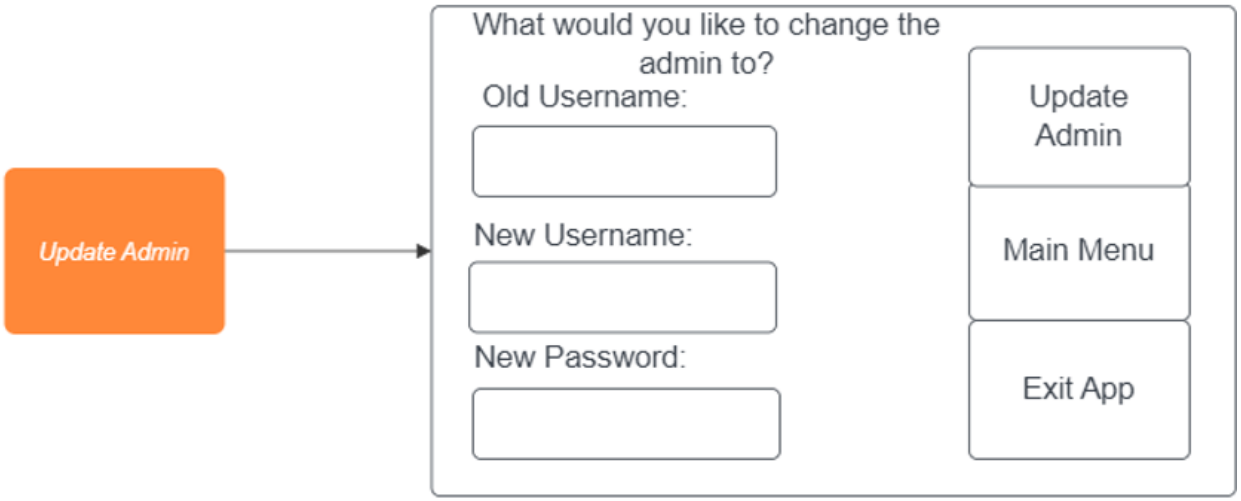

Figure 4.7 is the final window exclusive to the admin user and can be found through the main admin window. This window provides the admin the capability to change who has admin access, still maintaining one primary admin user. The admin must first enter their current username and then the new username and password they would like the admin user to be associated with. After which, the admin must click the update admin button to complete the change. Once completed the current user will no longer be able to login as admin with their current information and will instead be a standard user. The new username and password will be able to login as admin and have all admin capabilities. This window also contains the standard main menu and exit app functions that provide the user with options as to how they would like to proceed after completing their desired function.

Figure 4.8: User success popup

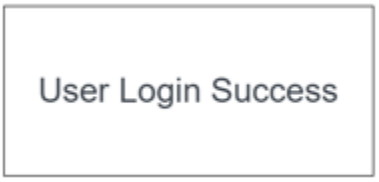

Figure 4.8 is another quick popup that will show the user that they have successfully logged in as a standard user and then redirects them to the main menu.

Figure 4.9: Main menu that will appear after successful login

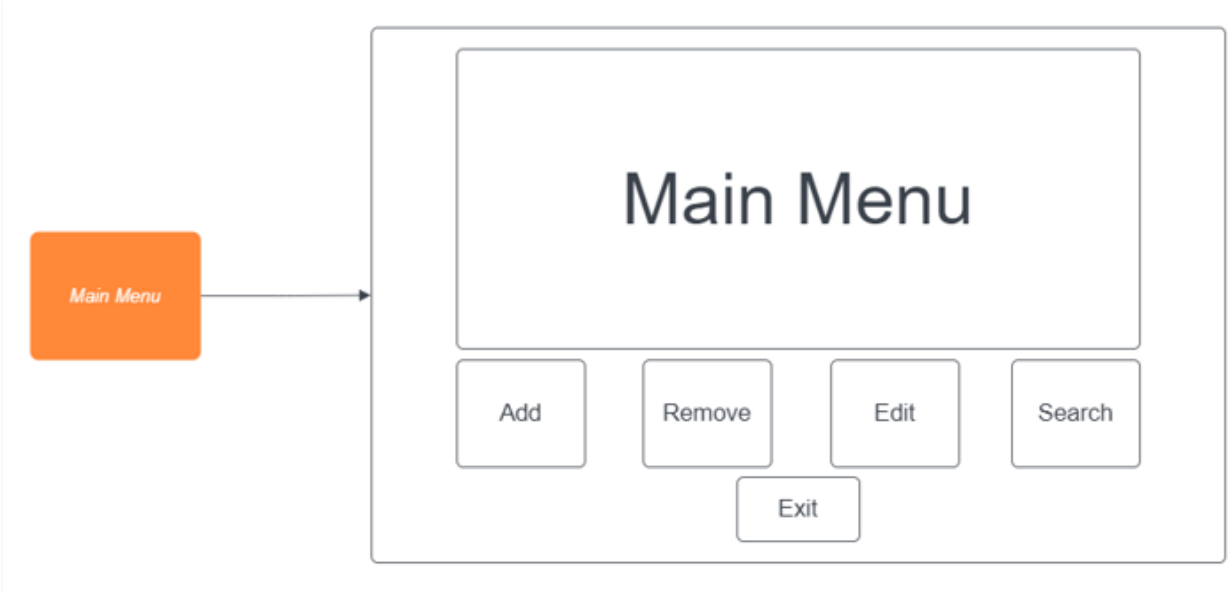

Figure 4.9 is the main menu that the user will interact with either immediately if they are not an admin or through the main menu button on any admin window. This menu allows the user to manipulate and view the task management system along with its contents via the built in buttons which will be further explained in the following Figures. The main menu also will contain a calendar showing the current tasks and their associated traits located in the large central box. Lastly, the main menu has an exit button allowing the user to quit the program entirely when desired.

Figure 4.10: Add function from the main menu

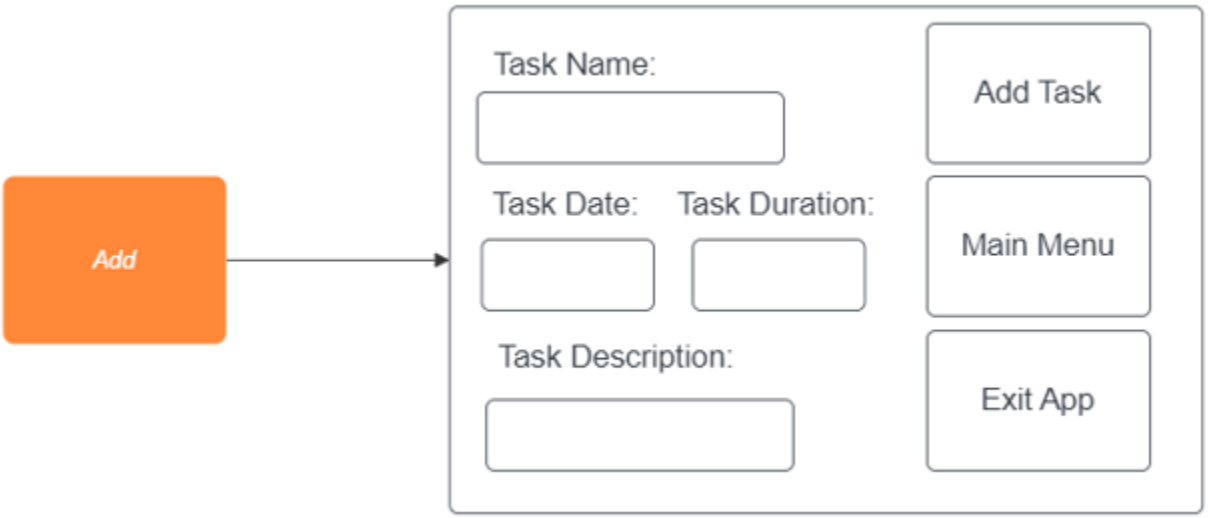

Figure 4.10 is the window that will appear when the user wishes to add a task. The text boxes will ask the user to input the task name, date, duration, and description, all of which will be stored when the user presses the Add Task button. The window will close upon the press of any of the three side buttons. Add Task will store the data in the text boxes and then close, Main Menu will return the user to the main menu (Figure 4.9), and Exit App will close everything including the main menu.

Figure 4.11: Remove function from the main menu

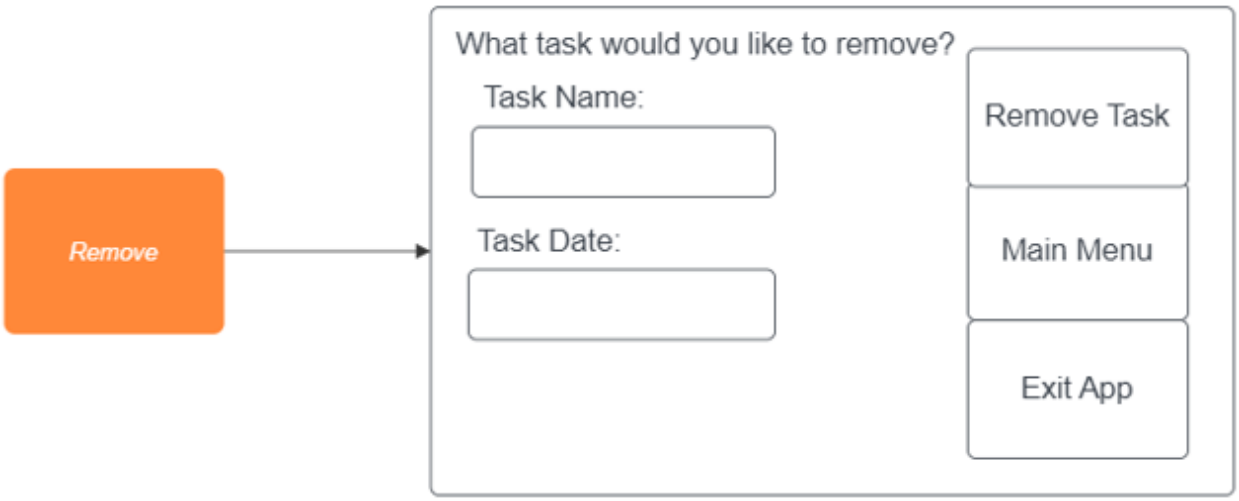

Figure 4.11 is an extension of the main menu window that can be reached by clicking the remove button. This window gives the user the capability to remove any task within the TMS by providing the name and date of the desired task. After which, the user will have to click remove task to finish the operation and then the task will be fully removed from the TMS profile of that user. This window also has a main menu and exit app button allowing the user to either return to the main menu or quit the app after completing their desired function.

Figure 4.12: Edit function from the main menu

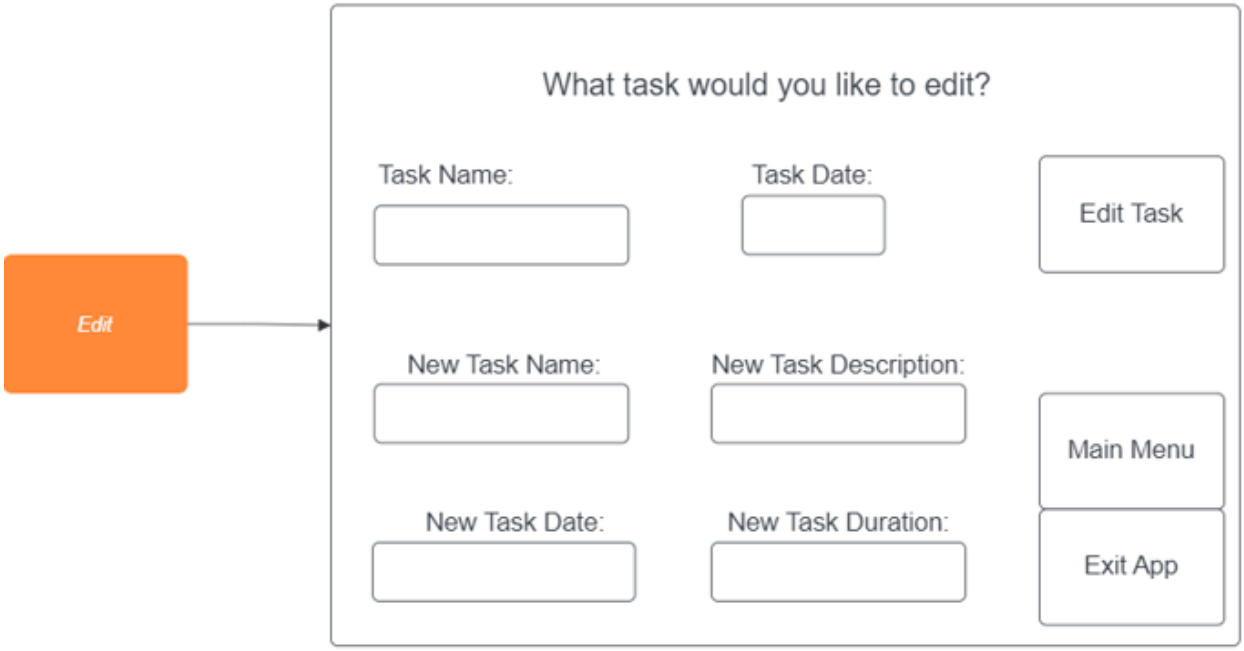

Figure 4.12 is the window that will appear when the user wishes to edit a task that already exists in the system. The text boxes will call for the user to enter the task name and date of the task they want to edit. This will allow us to identify the correct task if it is one that appears multiple times. Next the user will input the new data for the task. The window will close upon the press of any of the three side buttons. Edit Task will update the data in our system and then close, Main Menu will return the user to the main menu (Figure 4.9), and Exit App will close everything including the main menu.

Figure 4.13: Search Function from the main menu

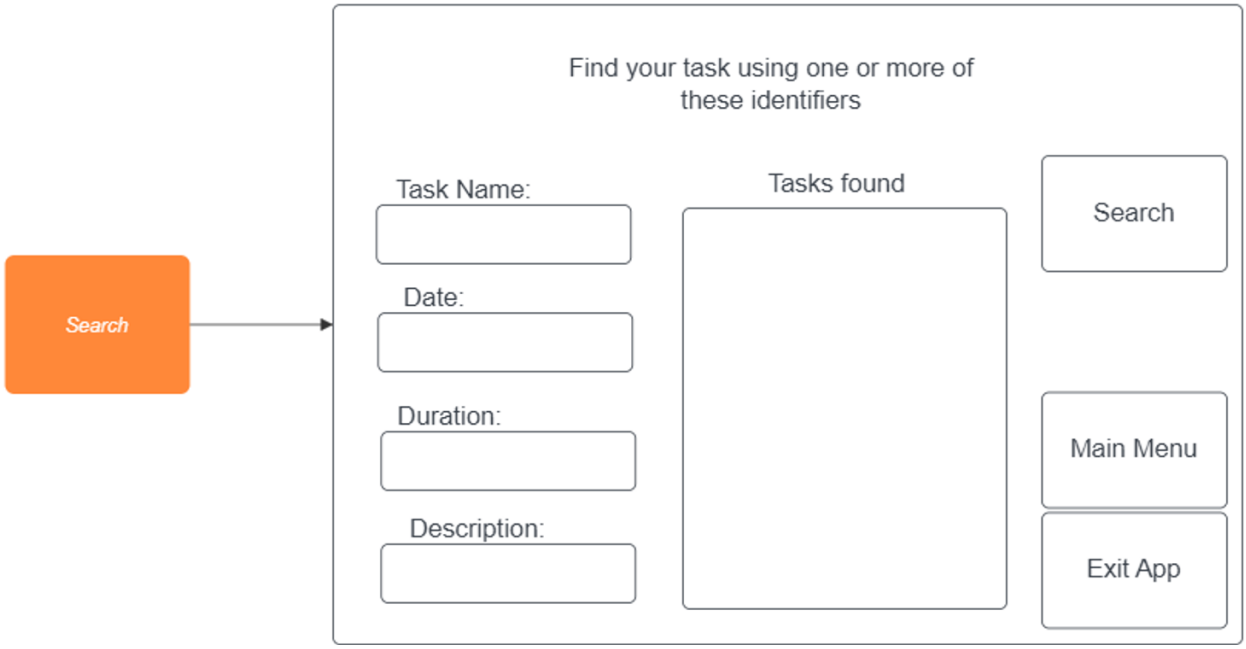

Figure 4.13 is the final function a user can perform within the TMS and can be found from the main menu by clicking the search button. This function will allow a user to find tasks with matching attributes to the provided information allowing for both general searches of a single day or a complex search to find an exact task. Once the user provides the desired information in the entries, they can press search to complete the action and find the tasks with matching attributes. This window also includes the standard main menu and exit functions allowing the user to complete more tasks if they desire or quit the app entirely.

Full UX and UI Diagram
https://drive.google.com/file/d/1bygt078_N-g_0CjdQ1B8VBVUbyjlgudS/view?usp=sharing