Task Management System

Introduction to Programming

CMPT 120L

Maddie&Gaby

Marist College
School of Computer Science and Mathematics

Submitted To:
Dr. Reza Sadeghi
Spring 2024

Project Progress Report 4 of Task Management System

Team Name

Maddie&Gaby

Team Members

1. Gabriela Ramon gabriela.ramon1@marist.edu (Team Head)
2. Madison Chan madison.chan1@marist.edu (Team Member)

Description of Team Members

1. Gabriela Ramon

   I am a freshman with a major in Applied Mathematics with a concentration in Finance and a minor in Data Science and Analytics. I am taking Intro to Programming this semester to build my computing skills while also fulfilling one of my major/minor requirements. Maddie and I chose to work together because we have worked well together on various past assignments in class. I chose to be the team head in order to manage our communication and become responsible for all our submissions.

2. Madison Chan

   I am a sophomore with a major in Business Administration with a concentration in Marketing as well as a double minor in Graphic Design and Data Science & Analytics. I am taking Intro to Programming because it is required for my minor. Gaby and I chose to work together because we worked on previous in-class assignments together. Gaby is the team head because she volunteered to be responsible for all future submissions.

Table of Contents

Table of Figures

Figure 1: Graphical User Experience Design

Figure 2: Graphical User Interface Design

Project Objective

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

1. The admin user is capable of:
    a. Having an 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 and password. A normal user is not able to define or remove other users.
    d. Remove users from TMS by removing their username, password, and corresponding recorded data.
2. Each user should be able to:
    a. Add a task to TMS. The task contains: a 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 work 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 on 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 the 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 this 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 letters. In a Caesar cipher with a shift of 4, 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.

Private GitHub Repository Address

https://github.com/gabyramon/CMPT-120L-112_Task-Management-System_Maddie-Gaby

Graphical User Experience Design

Login Page:

Input: Username and password in text

Output: Valid username and password enable user to access the main page. Invalid username and password will cause an invalid message box to show.

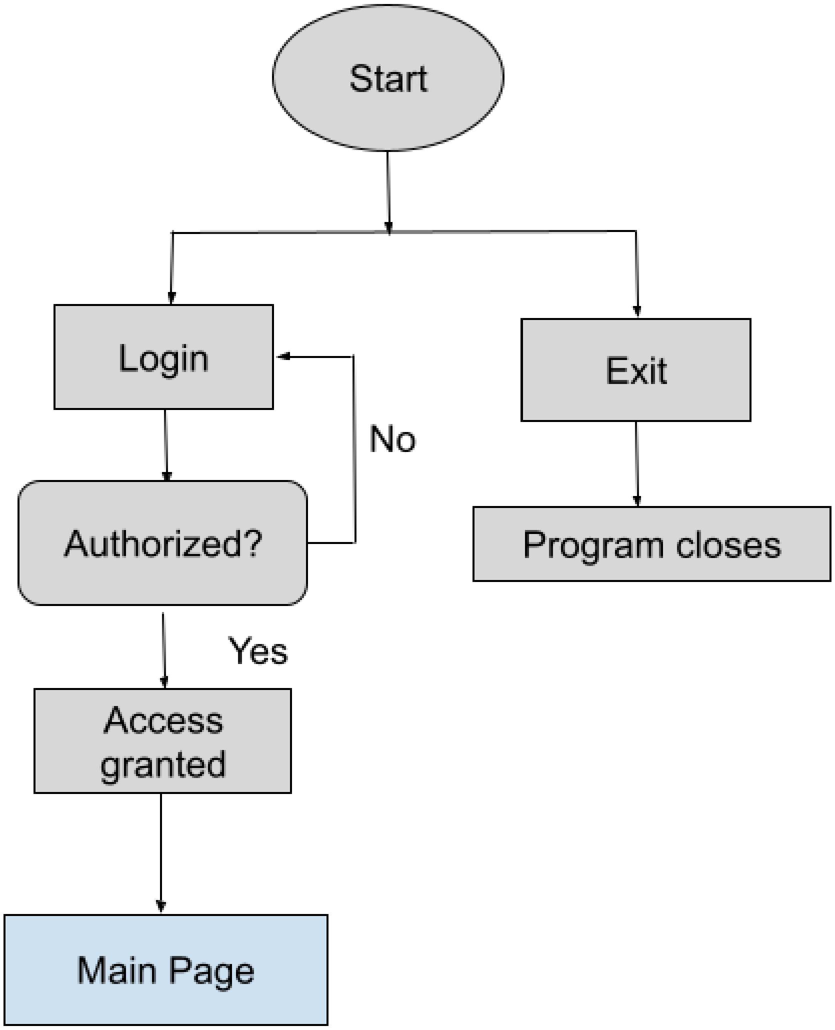


Figure 1.1: Login Page Flowchart

Main Menu:

Input: User will either select to add, remove, edit or search for a task

Output: Their selection will take the user to that page

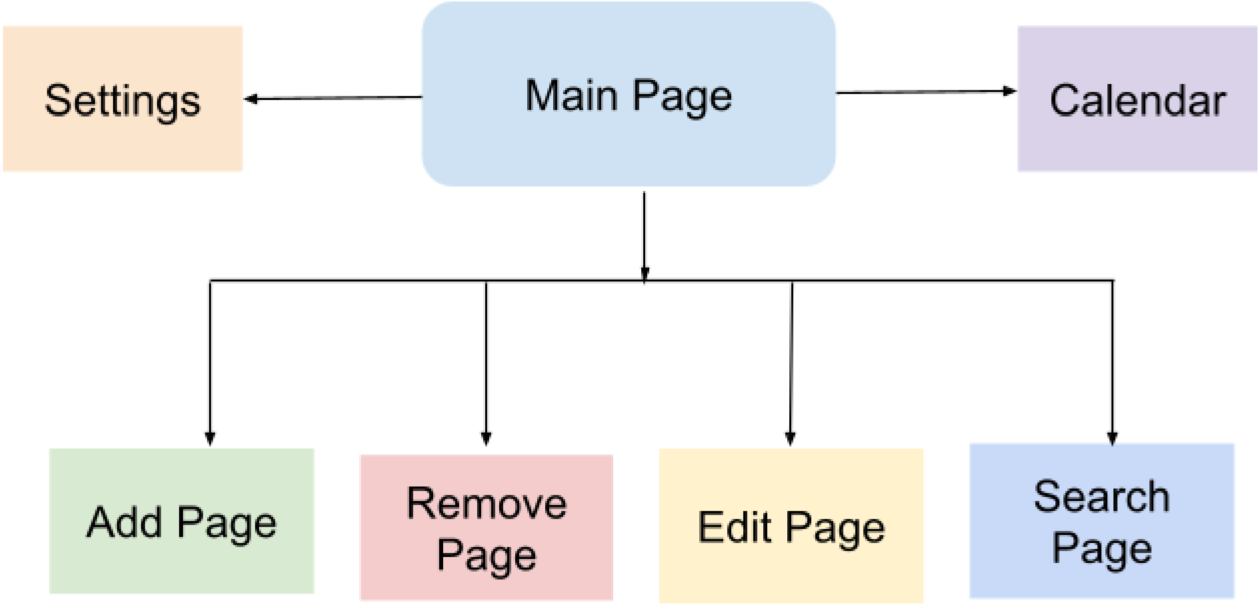


Figure 1.2: Main Page Flowchart

Add Page:

Input: User will enter a task that contains title, time, duration, and description

Output: Valid input will be saved and a success message will show. Invalid input will prompt the user to reenter the information.

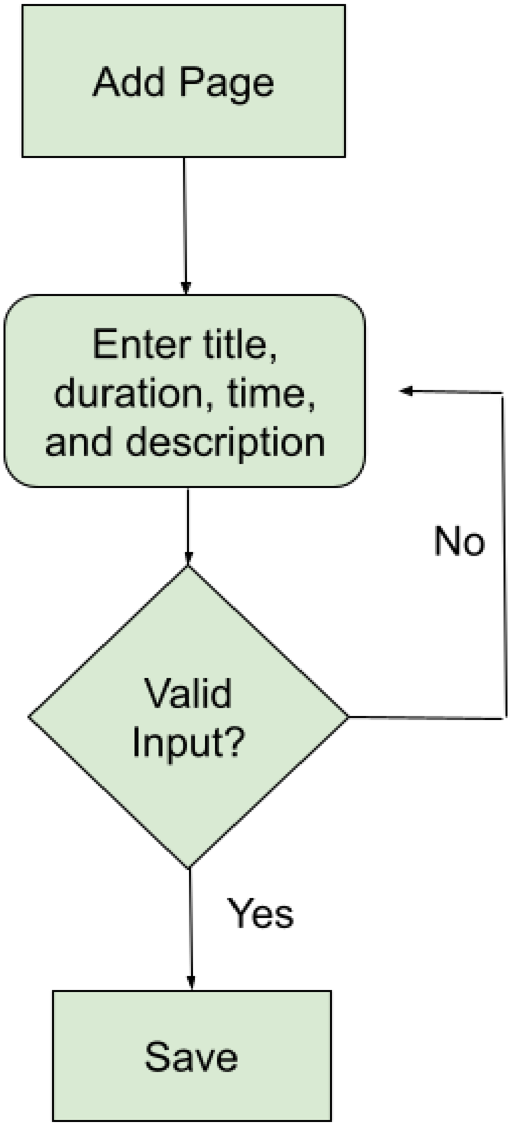


Figure 1.3: Add Page Flowchart

Remove Page:

Input: User will select the task that they choose to remove

Output: Valid input will be removed from the window' invalid input will show the user an error message and ask the user to choose an existing task.

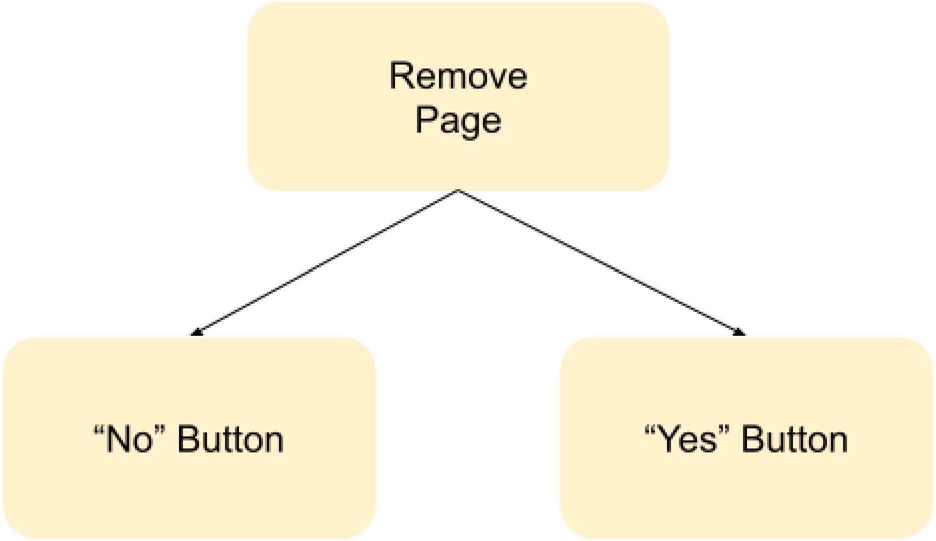


Figure 1.4: Remove Page Flowchart

Edit Page:

Input: User will select the task that they choose to edit, then select what part of the task they choose to edit (title, time, duration, or description), and then fix the task

Output: Input will be saved and a success message will appear.

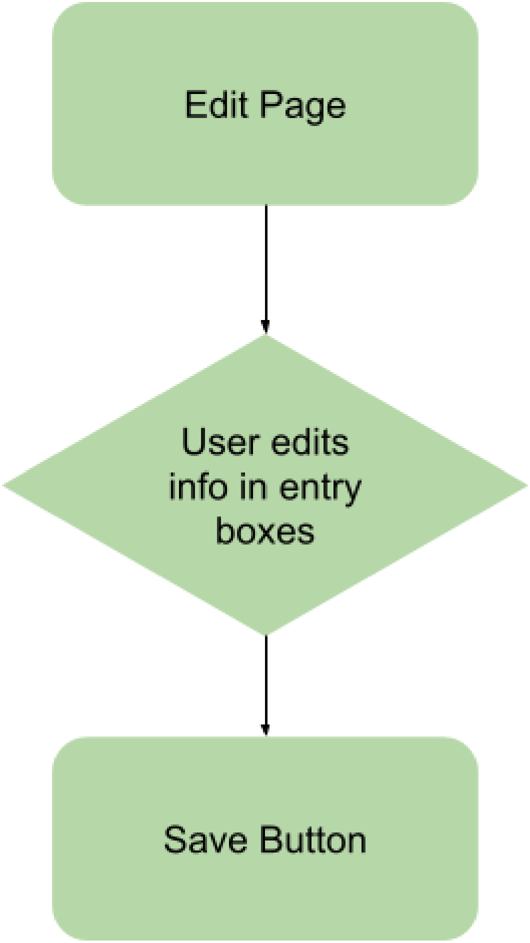


Figure 1.5: Edit Page Flowchart

Search Page:

Input: User will enter any key words related to the task they are searching for

Output: Program will display all tasks related to the key word(s) that were inputted.

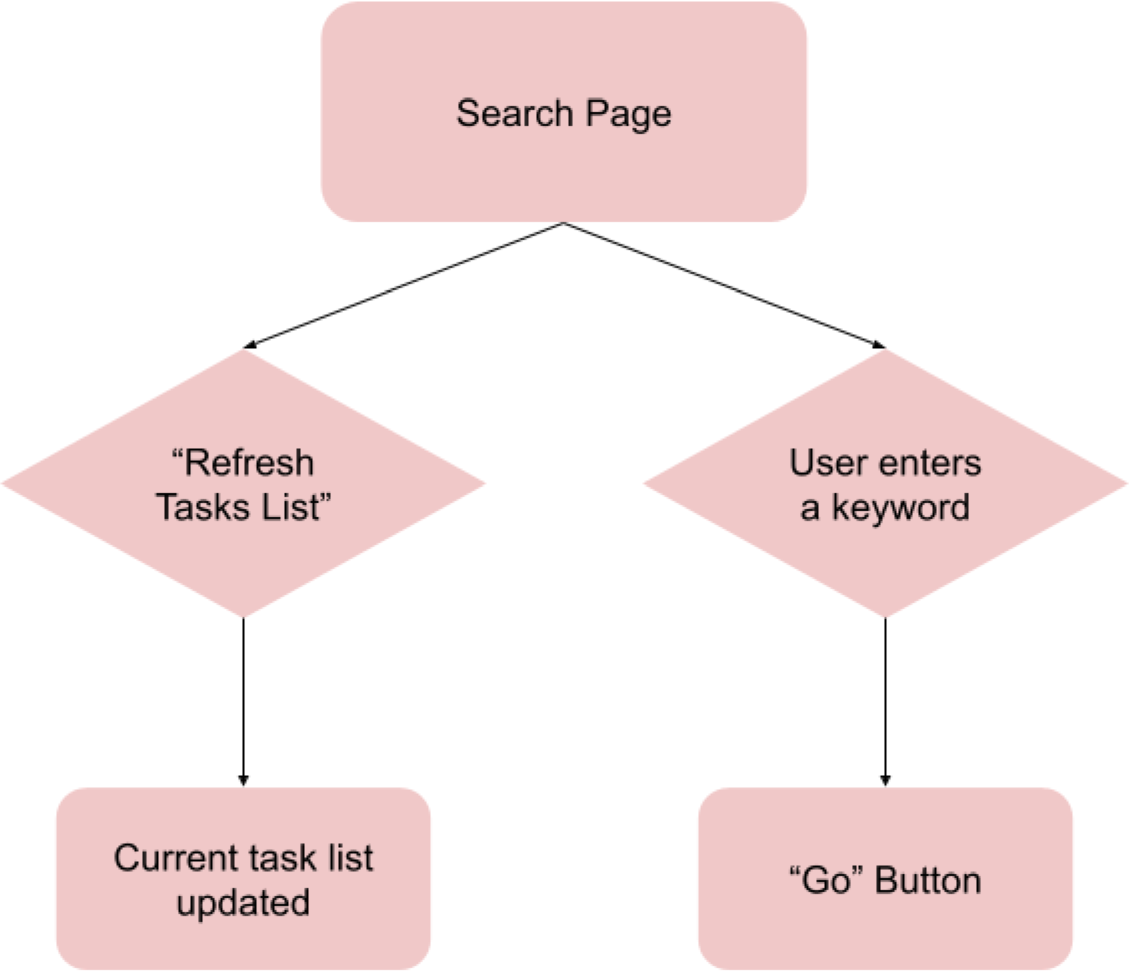


Figure 1.6: Search Page Flowchart

Settings Page:

Input: User will input the correct username and password. Invalid inputs will display an error message.

Output: Program will display an add, remove, and change button for the user to edit the username and password.

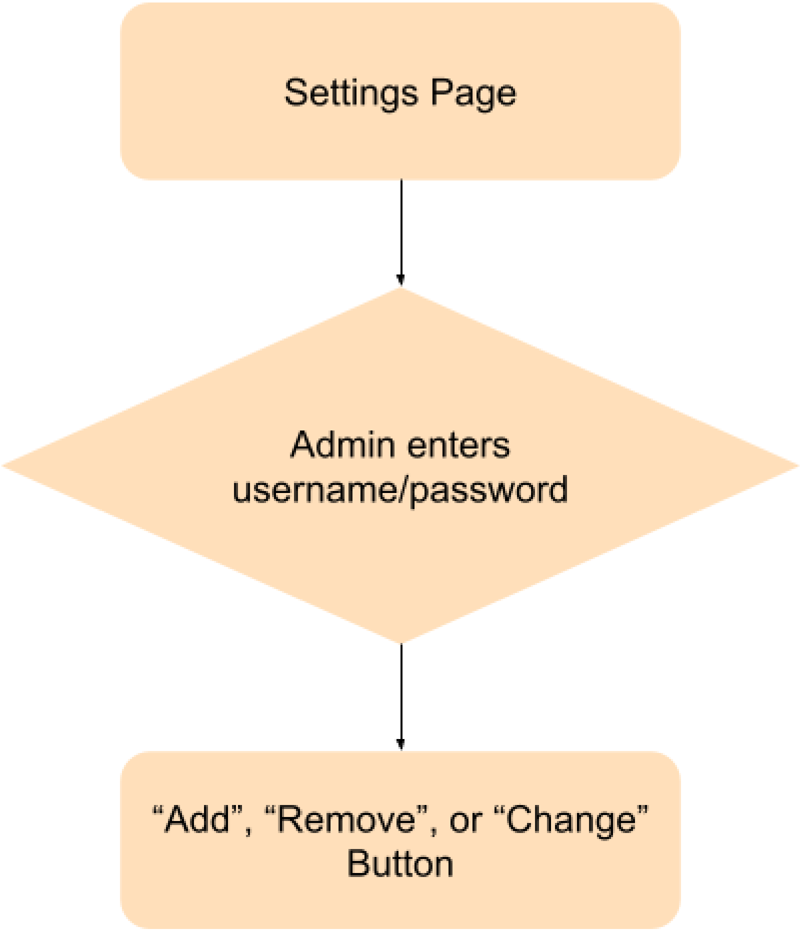

Settings Page
Admin enters username/password
"Add", "Remove", or "Change" Button

Figure 1.7: Settings Page Flowchart

Calendar Page:

Input: User will input the month and year that they choose to view.

Output: Main page will be updated with the calendar

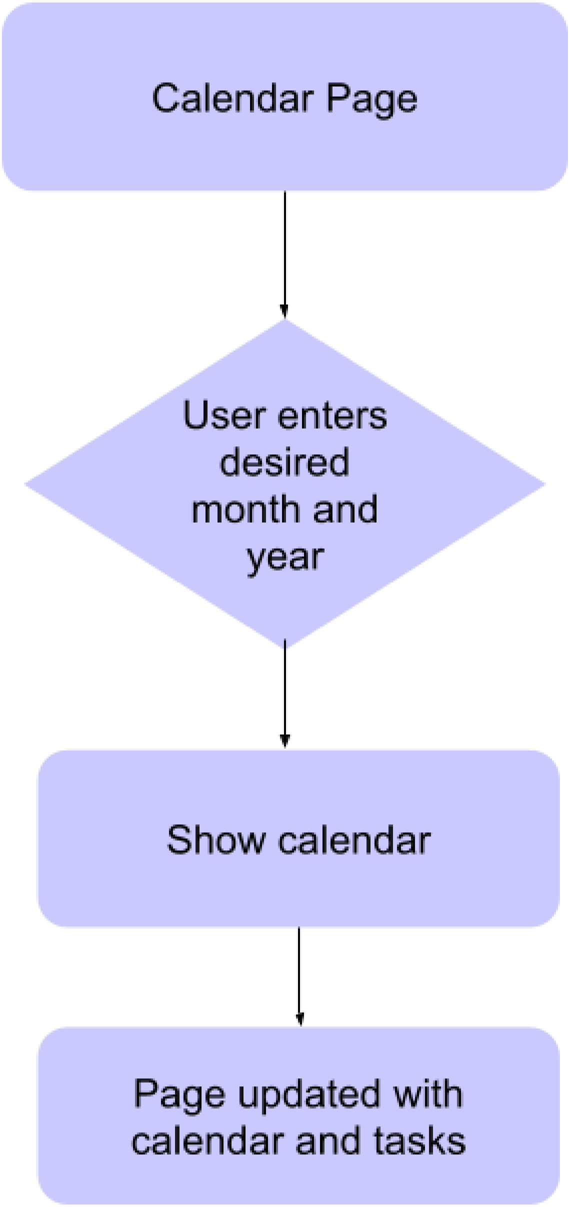
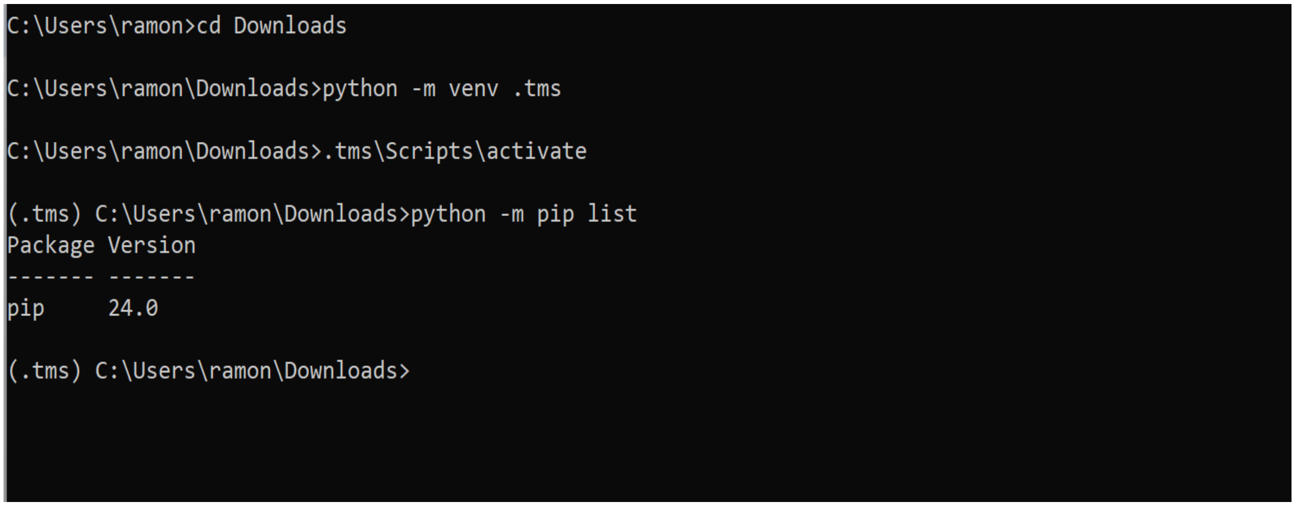

Calendar Page
User enters desired month and year
Show calendar
Page updated with calendar and tasks

Figure 1.8: Calendar Page Flowchart

List of Used Packages

- OS
- Tkinter
- CSV
- Tkmacosx
- Pandas

Project’s Virtual Environment

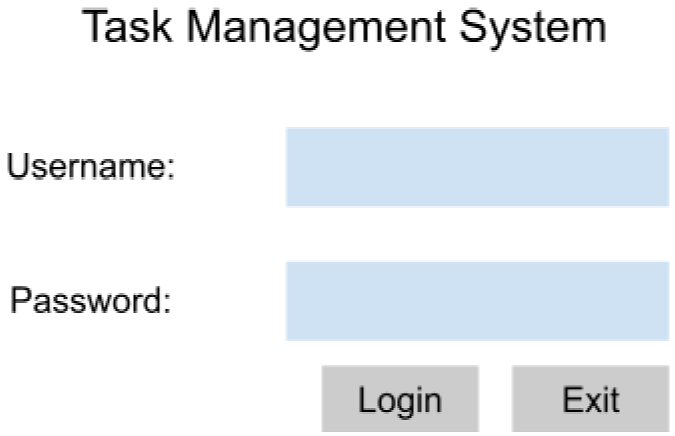


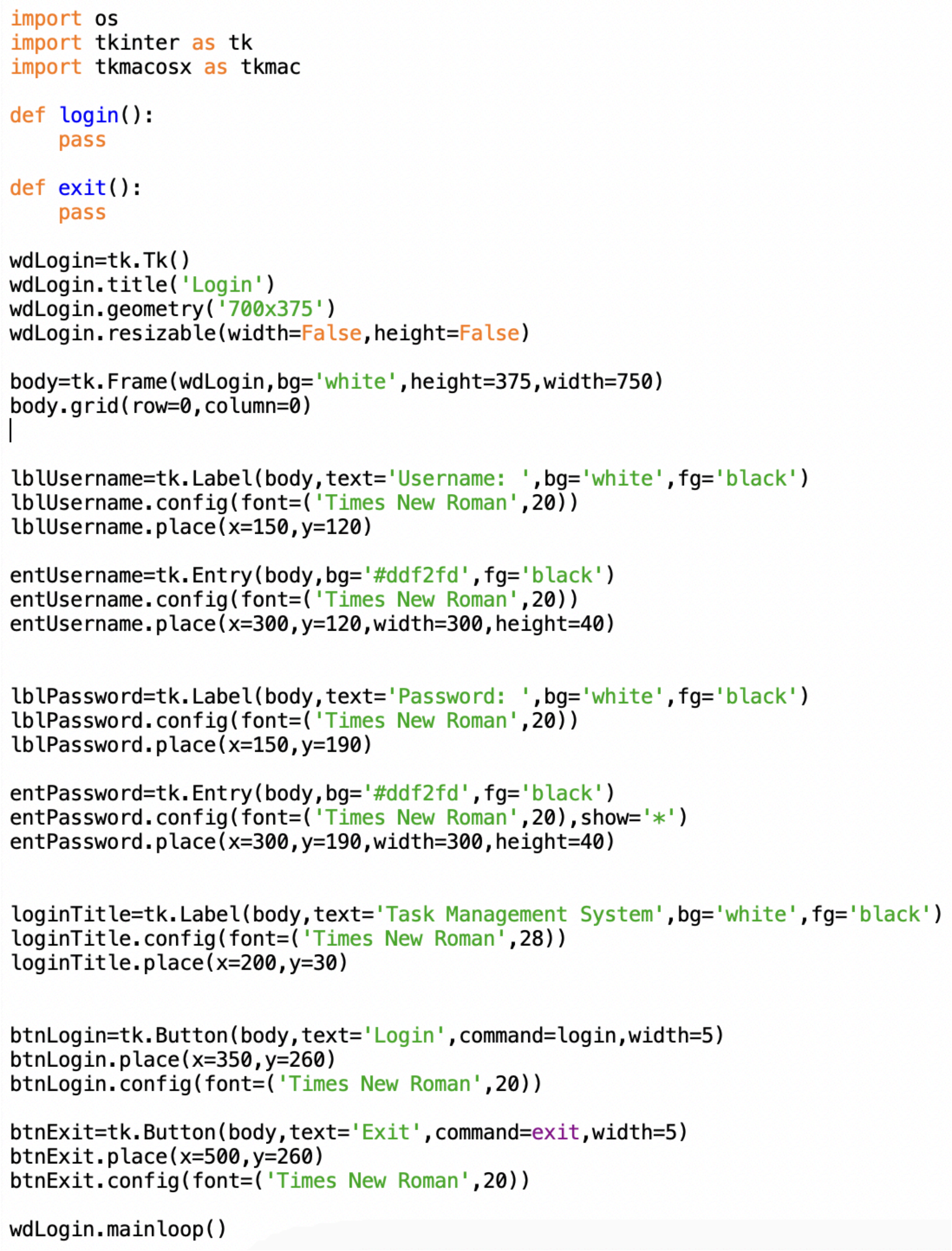
```
C:\Users\ramon>cd Downloads

C:\Users\ramon\Downloads>python -m venv .tms

C:\Users\ramon\Downloads>.tms\Scripts\activate

(.tms) C:\Users\ramon\Downloads>python -m pip list
Package Version
------- -------
pip     24.0

(.tms) C:\Users\ramon\Downloads>
```

Our group’s virtual environment does not currently contain any external packages.

Graphical User Interface Design

Login Page:
The Login Page is the first window that will be shown to the user upon running the program. It has two entry boxes where the user will enter their username and password, and two buttons to attempt a login or to exit the program.



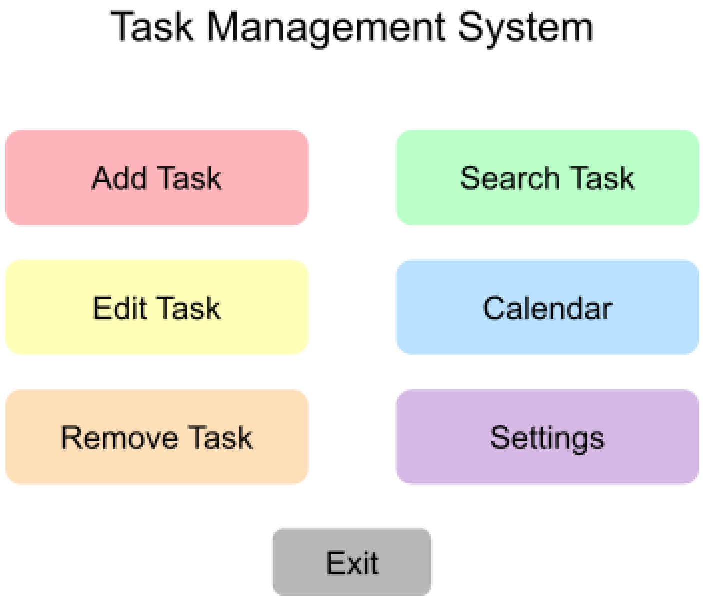


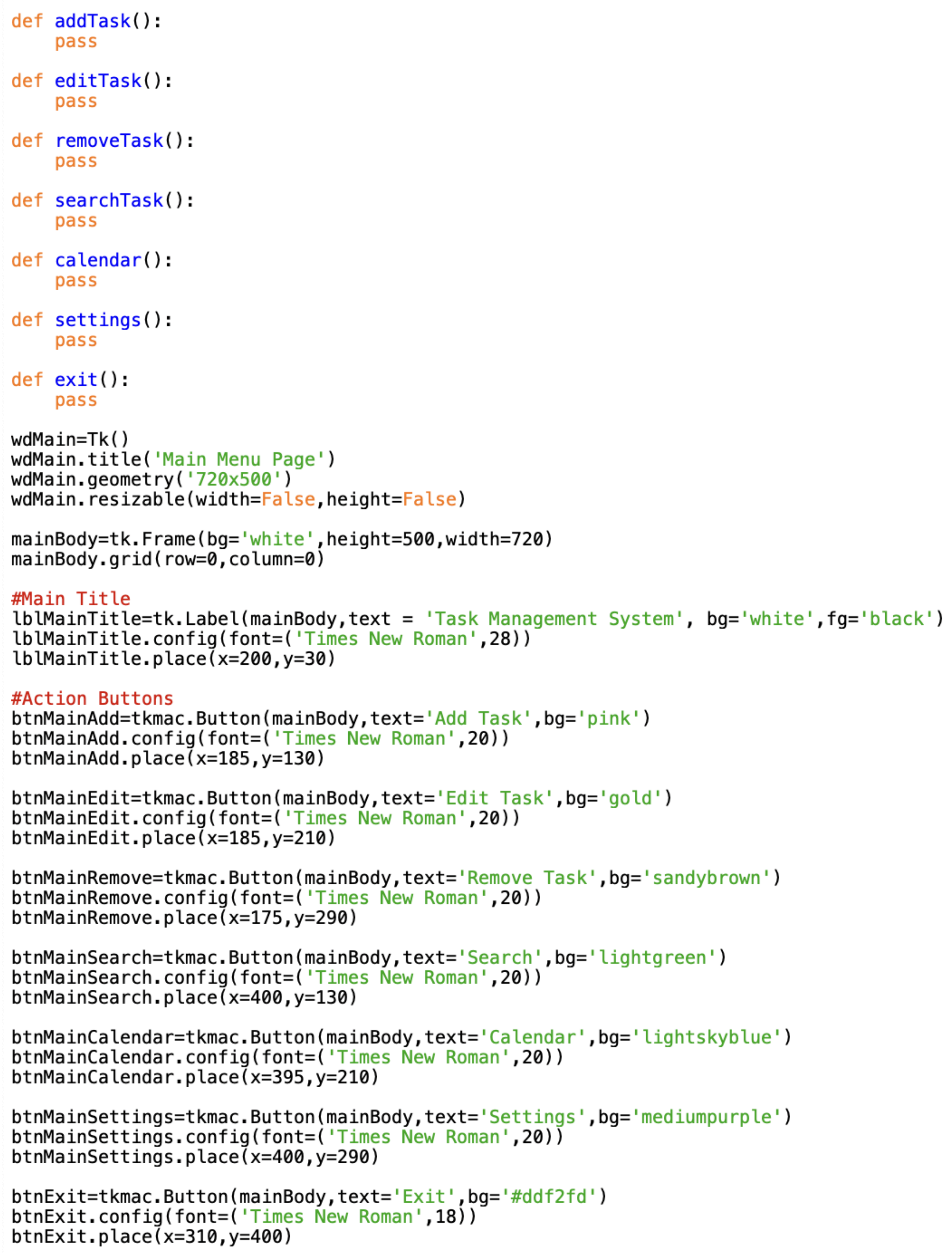
```
import os
import tkinter as tk
import tkmacosx as tkmac

def login():
    pass

def exit():
    pass

wdLogin=tk.Tk()
wdLogin.title('Login')
wdLogin.geometry('700x375')
wdLogin.resizable(width=False,height=False)

body=tk.Frame(wdLogin,bg='white',height=375,width=750)
body.grid(row=0,column=0)


lblUsername=tk.Label(body,text='Username: ',bg='white',fg='black')
lblUsername.config(font=('Times New Roman',20))
lblUsername.place(x=150,y=120)

entUsername=tk.Entry(body,bg='#ddf2fd',fg='black')
entUsername.config(font=('Times New Roman',20))
entUsername.place(x=300,y=120,width=300,height=40)


lblPassword=tk.Label(body,text='Password: ',bg='white',fg='black')
lblPassword.config(font=('Times New Roman',20))
lblPassword.place(x=150,y=190)

entPassword=tk.Entry(body,bg='#ddf2fd',fg='black')
entPassword.config(font=('Times New Roman',20),show='*')
entPassword.place(x=300,y=190,width=300,height=40)


loginTitle=tk.Label(body,text='Task Management System',bg='white',fg='black')
loginTitle.config(font=('Times New Roman',28))
loginTitle.place(x=200,y=30)


btnLogin=tk.Button(body,text='Login',command=login,width=5)
btnLogin.place(x=350,y=260)
btnLogin.config(font=('Times New Roman',20))

btnExit=tk.Button(body,text='Exit',command=exit,width=5)
btnExit.place(x=500,y=260)
btnExit.config(font=('Times New Roman',20))

wdLogin.mainloop()
```

Figure 2.1 Login Page Window

Main Page:

After a successful login, the user will be directed to the program's main page. This page is made up of six buttons where the user can add, edit, remove, or search for a task, or view the program’s settings and calendar. There is also an exit button that allows the user to quit the program entirely.



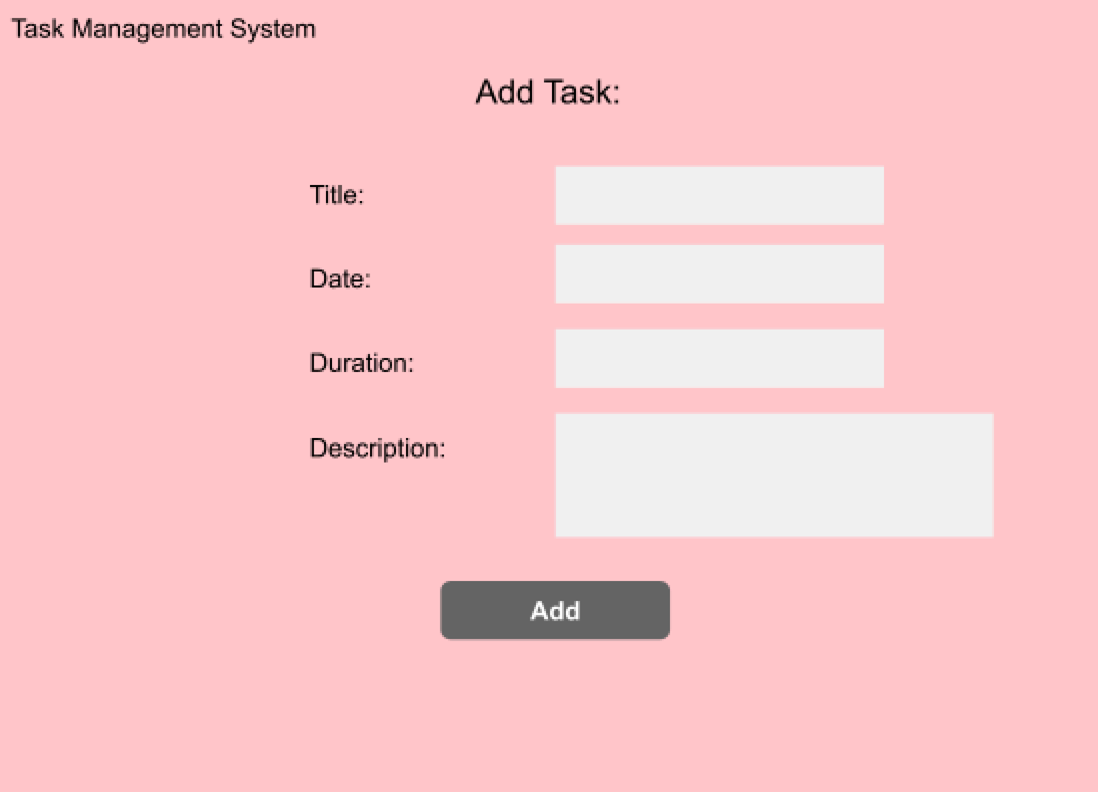


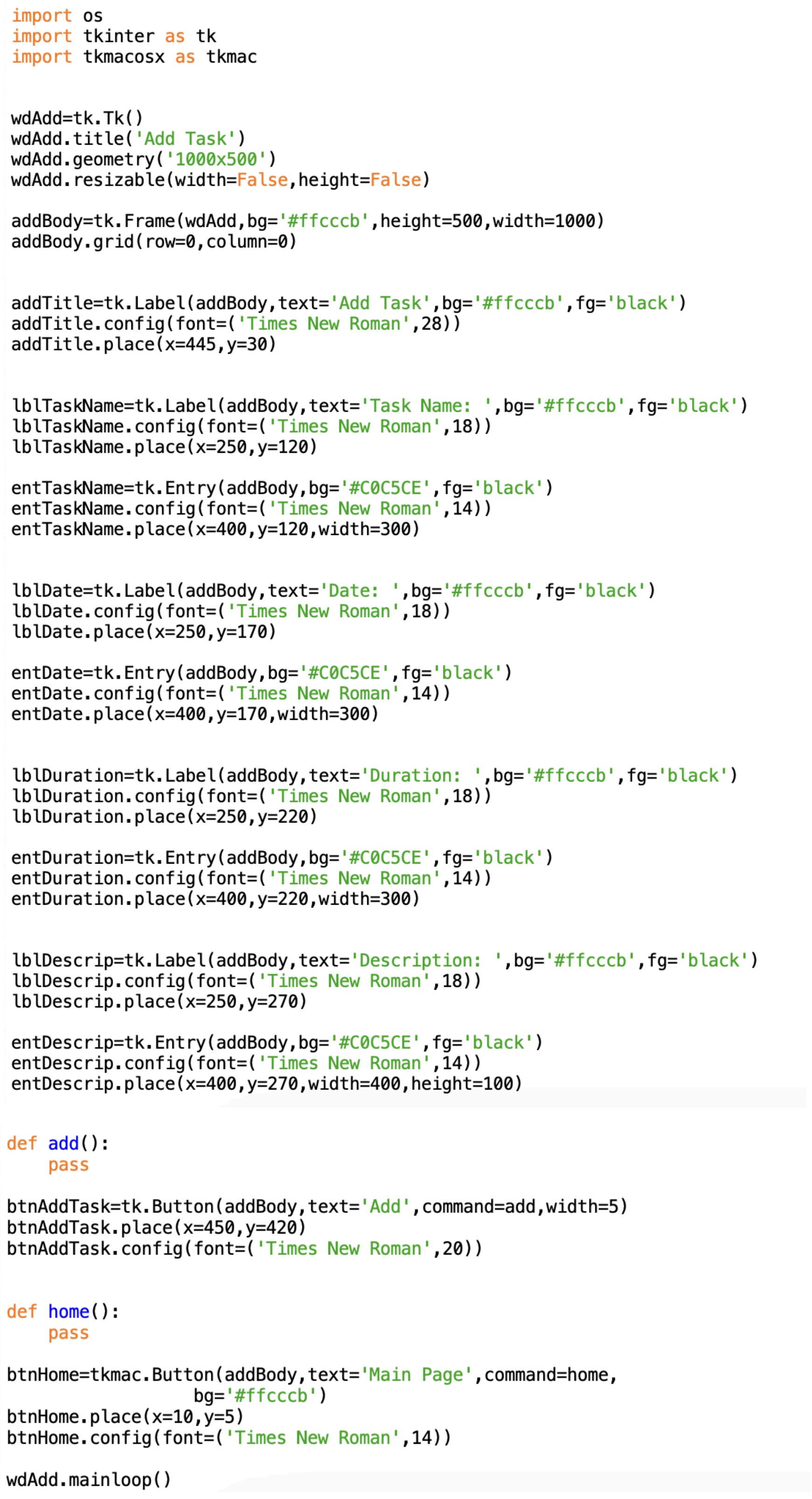
```
def addTask():
    pass

def editTask():
    pass

def removeTask():
    pass

def searchTask():
    pass

def calendar():
    pass

def settings():
    pass

def exit():
    pass

wdMain=Tk()
wdMain.title('Main Menu Page')
wdMain.geometry('720x500')
wdMain.resizable(width=False,height=False)

mainBody=tk.Frame(bg='white',height=500,width=720)
mainBody.grid(row=0,column=0)

#Main Title
lblMainTitle=tk.Label(mainBody,text = 'Task Management System', bg='white',fg='black')
lblMainTitle.config(font=('Times New Roman',28))
lblMainTitle.place(x=200,y=30)

#Action Buttons
btnMainAdd=tkmac.Button(mainBody,text='Add Task',bg='pink')
btnMainAdd.config(font=('Times New Roman',20))
btnMainAdd.place(x=185,y=130)

btnMainEdit=tkmac.Button(mainBody,text='Edit Task',bg='gold')
btnMainEdit.config(font=('Times New Roman',20))
btnMainEdit.place(x=185,y=210)

btnMainRemove=tkmac.Button(mainBody,text='Remove Task',bg='sandybrown')
btnMainRemove.config(font=('Times New Roman',20))
btnMainRemove.place(x=175,y=290)

btnMainSearch=tkmac.Button(mainBody,text='Search',bg='lightgreen')
btnMainSearch.config(font=('Times New Roman',20))
btnMainSearch.place(x=400,y=130)

btnMainCalendar=tkmac.Button(mainBody,text='Calendar',bg='lightskyblue')
btnMainCalendar.config(font=('Times New Roman',20))
btnMainCalendar.place(x=395,y=210)

btnMainSettings=tkmac.Button(mainBody,text='Settings',bg='mediumpurple')
btnMainSettings.config(font=('Times New Roman',20))
btnMainSettings.place(x=400,y=290)

btnExit=tkmac.Button(mainBody,text='Exit',bg='#ddf2fd')
btnExit.config(font=('Times New Roman',18))
btnExit.place(x=310,y=400)
```

Figure 2.2: Main Page Window

Add Page:

The Add Page has four entries where the user can add the task's title, date, duration and description. The add button will save the new task in a csv file and show a success message box. The upper left Task Management System button will bring the user back to the main page.



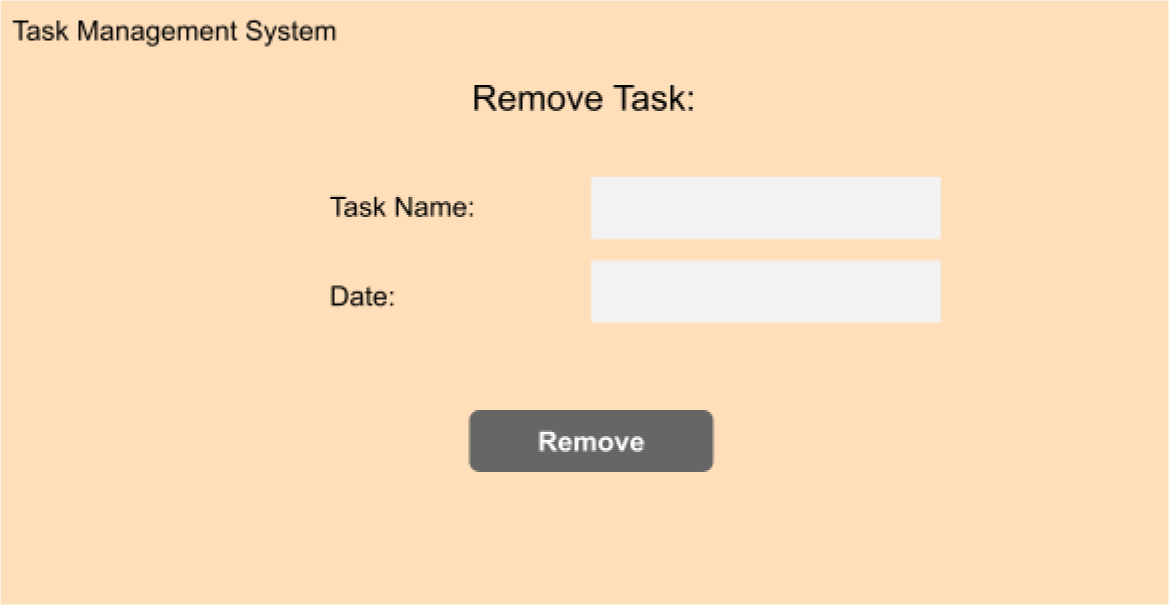


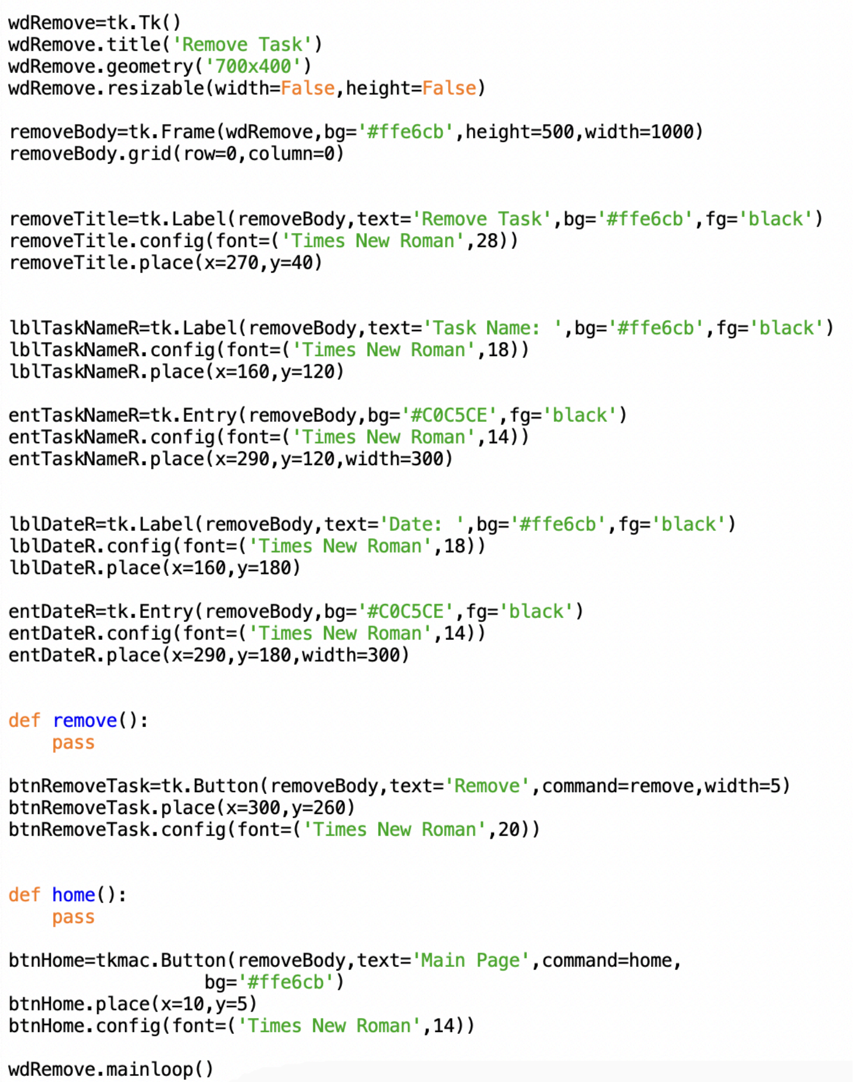
```
import os
import tkinter as tk
import tkmacosx as tkmac


wdAdd=tk.Tk()
wdAdd.title('Add Task')
wdAdd.geometry('1000x500')
wdAdd.resizable(width=False,height=False)

addBody=tk.Frame(wdAdd,bg='#ffcccb',height=500,width=1000)
addBody.grid(row=0,column=0)


addTitle=tk.Label(addBody,text='Add Task',bg='#ffcccb',fg='black')
addTitle.config(font=('Times New Roman',28))
addTitle.place(x=445,y=30)


lblTaskName=tk.Label(addBody,text='Task Name: ',bg='#ffcccb',fg='black')
lblTaskName.config(font=('Times New Roman',18))
lblTaskName.place(x=250,y=120)

entTaskName=tk.Entry(addBody,bg='#C0C5CE',fg='black')
entTaskName.config(font=('Times New Roman',14))
entTaskName.place(x=400,y=120,width=300)


lblDate=tk.Label(addBody,text='Date: ',bg='#ffcccb',fg='black')
lblDate.config(font=('Times New Roman',18))
lblDate.place(x=250,y=170)

entDate=tk.Entry(addBody,bg='#C0C5CE',fg='black')
entDate.config(font=('Times New Roman',14))
entDate.place(x=400,y=170,width=300)


lblDuration=tk.Label(addBody,text='Duration: ',bg='#ffcccb',fg='black')
lblDuration.config(font=('Times New Roman',18))
lblDuration.place(x=250,y=220)

entDuration=tk.Entry(addBody,bg='#C0C5CE',fg='black')
entDuration.config(font=('Times New Roman',14))
entDuration.place(x=400,y=220,width=300)


lblDescrip=tk.Label(addBody,text='Description: ',bg='#ffcccb',fg='black')
lblDescrip.config(font=('Times New Roman',18))
lblDescrip.place(x=250,y=270)

entDescrip=tk.Entry(addBody,bg='#C0C5CE',fg='black')
entDescrip.config(font=('Times New Roman',14))
entDescrip.place(x=400,y=270,width=400,height=100)

def add():
    pass

btnAddTask=tk.Button(addBody,text='Add',command=add,width=5)
btnAddTask.place(x=450,y=420)
btnAddTask.config(font=('Times New Roman',20))


def home():
    pass

btnHome=tkmac.Button(addBody,text='Main Page',command=home,
                     bg='#ffcccb')
btnHome.place(x=10,y=5)
btnHome.config(font=('Times New Roman',14))

wdAdd.mainloop()
```

Figure 2.3: Add Task Window

Remove Page:

This window allows the user to remove an existing task. There are two entry boxes where the user will enter the task name and date and a remove button that will remove the desired task. A success message box will appear if the task is successfully removed. The upper left Task Management System button will bring the user back to the main page.



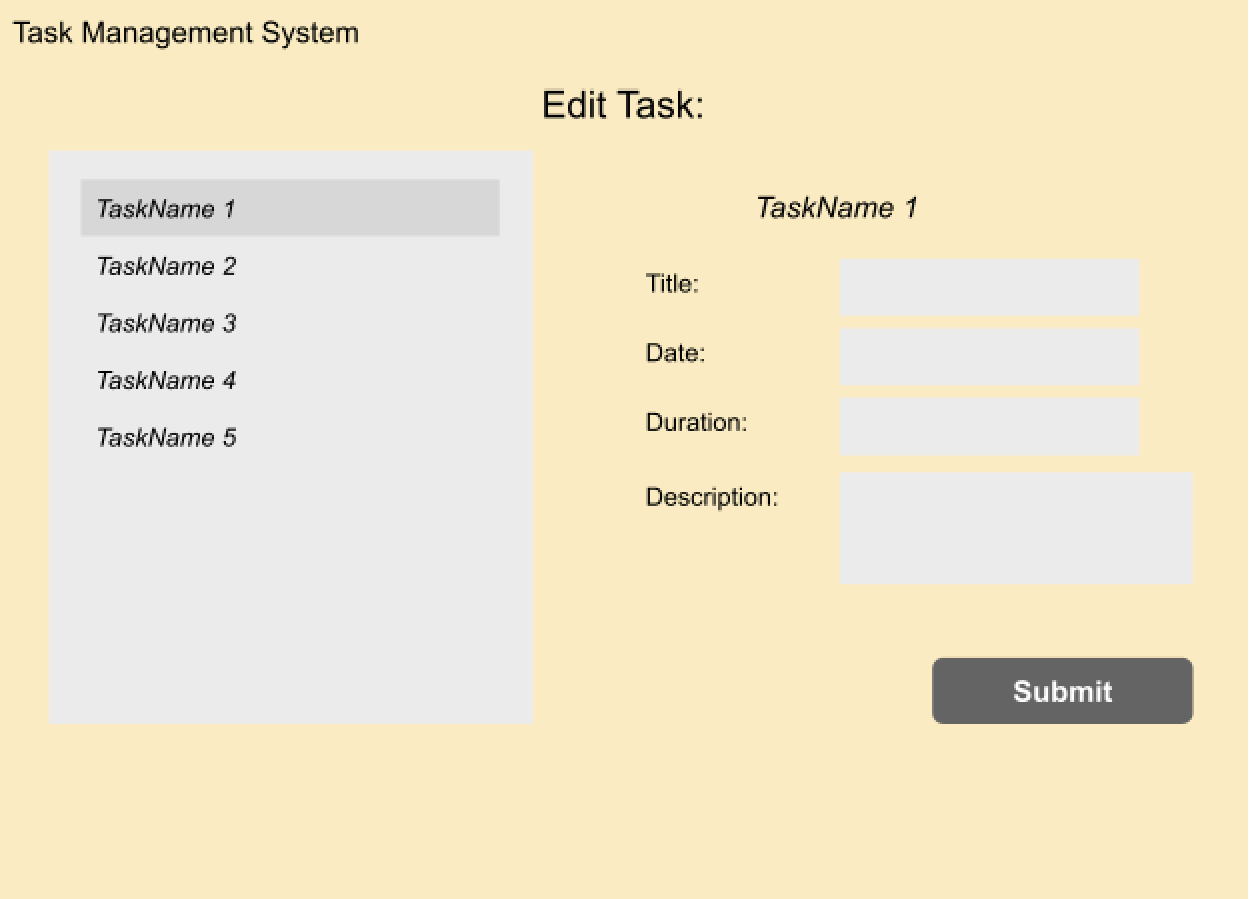


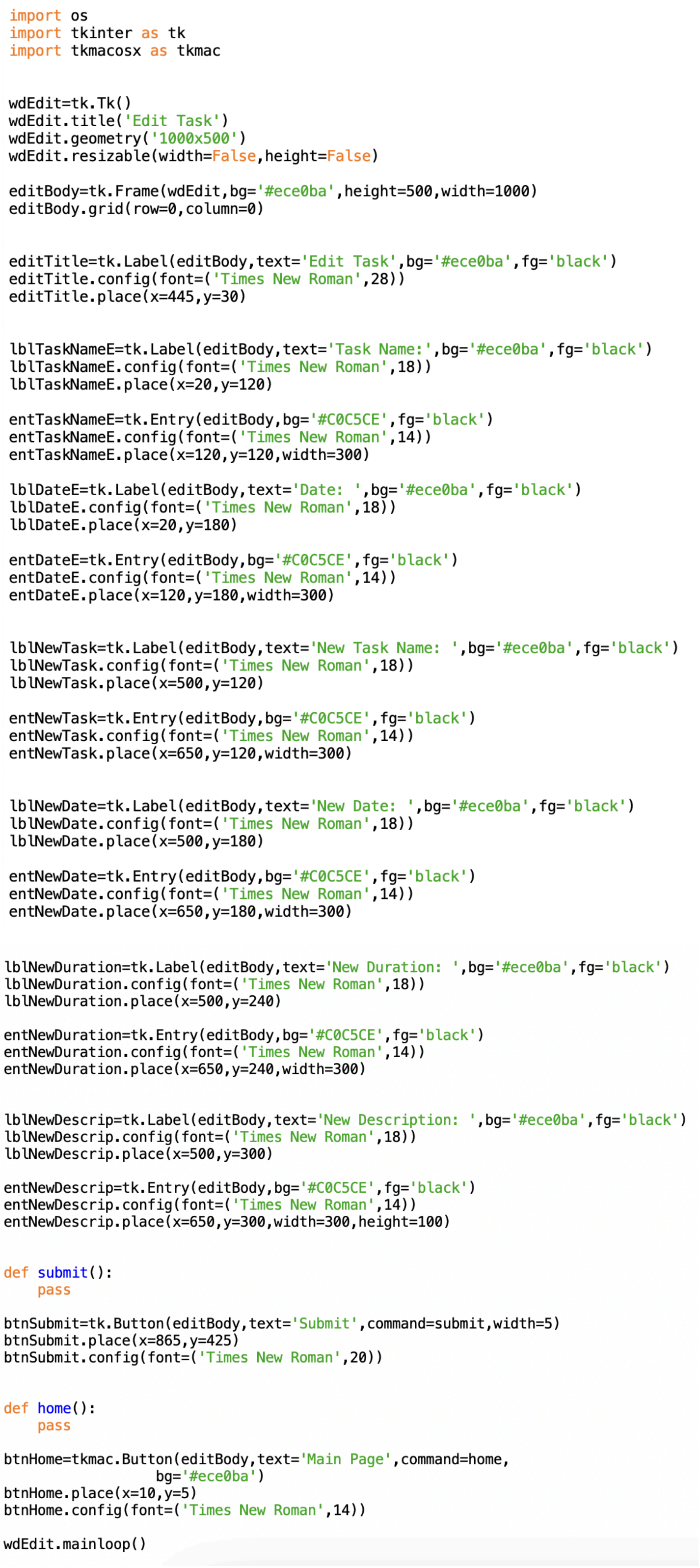
```
wdRemove=tk.Tk()
wdRemove.title('Remove Task')
wdRemove.geometry('700x400')
wdRemove.resizable(width=False,height=False)

removeBody=tk.Frame(wdRemove,bg='#ffe6cb',height=500,width=1000)
removeBody.grid(row=0,column=0)


removeTitle=tk.Label(removeBody,text='Remove Task',bg='#ffe6cb',fg='black')
removeTitle.config(font=('Times New Roman',28))
removeTitle.place(x=270,y=40)


lblTaskNameR=tk.Label(removeBody,text='Task Name: ',bg='#ffe6cb',fg='black')
lblTaskNameR.config(font=('Times New Roman',18))
lblTaskNameR.place(x=160,y=120)

entTaskNameR=tk.Entry(removeBody,bg='#C0C5CE',fg='black')
entTaskNameR.config(font=('Times New Roman',14))
entTaskNameR.place(x=290,y=120,width=300)


lblDateR=tk.Label(removeBody,text='Date: ',bg='#ffe6cb',fg='black')
lblDateR.config(font=('Times New Roman',18))
lblDateR.place(x=160,y=180)

entDateR=tk.Entry(removeBody,bg='#C0C5CE',fg='black')
entDateR.config(font=('Times New Roman',14))
entDateR.place(x=290,y=180,width=300)


def remove():
    pass

btnRemoveTask=tk.Button(removeBody,text='Remove',command=remove,width=5)
btnRemoveTask.place(x=300,y=260)
btnRemoveTask.config(font=('Times New Roman',20))


def home():
    pass

btnHome=tkmac.Button(removeBody,text='Main Page',command=home,
                  bg='#ffe6cb')
btnHome.place(x=10,y=5)
btnHome.config(font=('Times New Roman',14))

wdRemove.mainloop()
```

Figure 2.4: Remove Task Window

Edit Page:

The Edit Page allows the user to edit the task name, date, duration, or description of the chosen task. The user will enter what they want to edit and then select from the list box which task they want to edit, and then click the submit button. If any blanks are left empty, the original input will be kept. Lastly, there is a Main Page button on the top left that allows the user to return to the main menu.



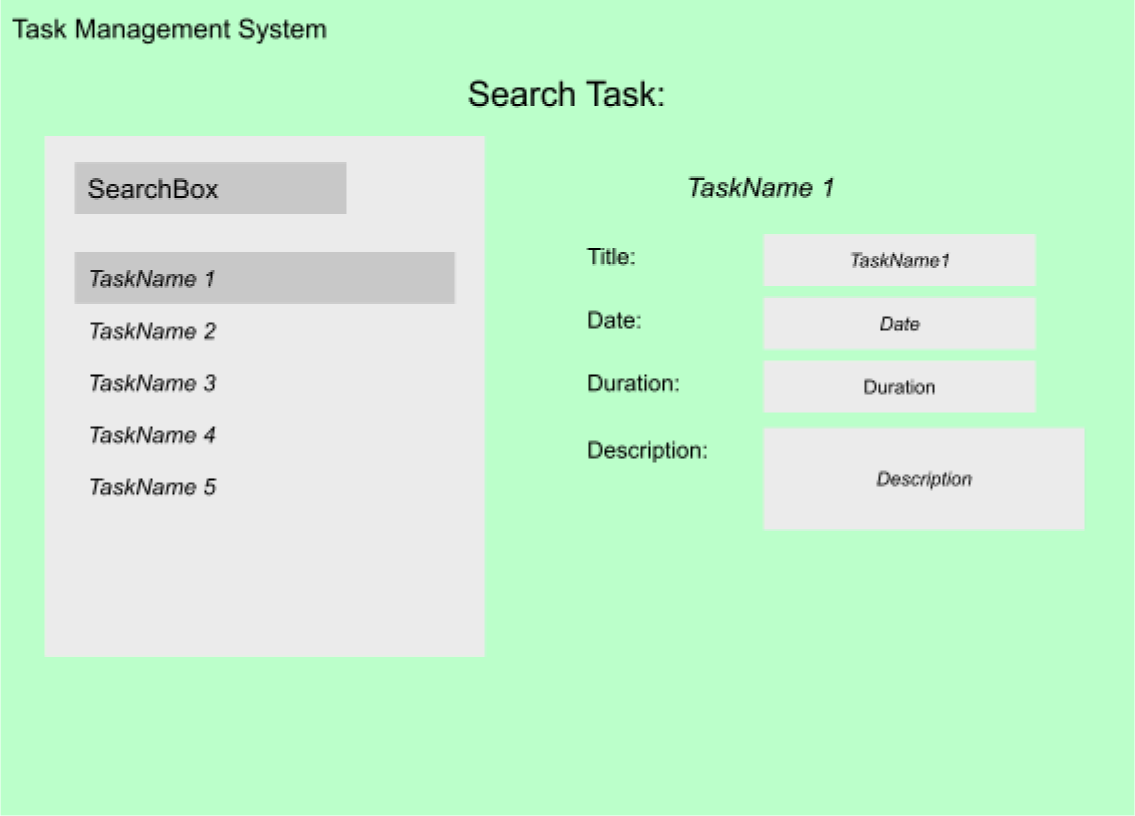


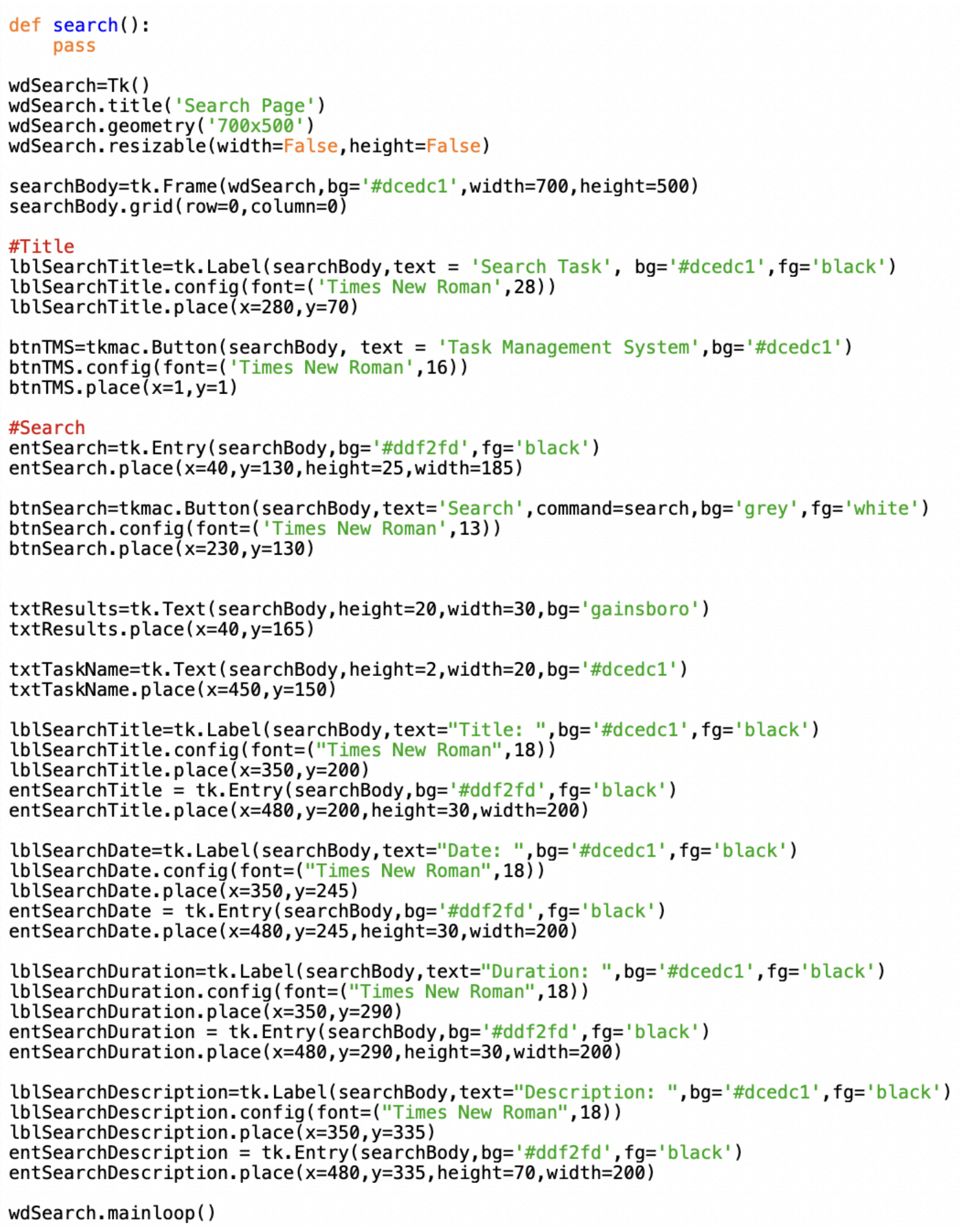
```
import os
import tkinter as tk
import tkmacosx as tkmac


wdEdit=tk.Tk()
wdEdit.title('Edit Task')
wdEdit.geometry('1000x500')
wdEdit.resizable(width=False,height=False)

editBody=tk.Frame(wdEdit,bg='#ece0ba',height=500,width=1000)
editBody.grid(row=0,column=0)


editTitle=tk.Label(editBody,text='Edit Task',bg='#ece0ba',fg='black')
editTitle.config(font=('Times New Roman',28))
editTitle.place(x=445,y=30)


lblTaskNameE=tk.Label(editBody,text='Task Name:',bg='#ece0ba',fg='black')
lblTaskNameE.config(font=('Times New Roman',18))
lblTaskNameE.place(x=20,y=120)

entTaskNameE=tk.Entry(editBody,bg='#C0C5CE',fg='black')
entTaskNameE.config(font=('Times New Roman',14))
entTaskNameE.place(x=120,y=120,width=300)

lblDateE=tk.Label(editBody,text='Date: ',bg='#ece0ba',fg='black')
lblDateE.config(font=('Times New Roman',18))
lblDateE.place(x=20,y=180)

entDateE=tk.Entry(editBody,bg='#C0C5CE',fg='black')
entDateE.config(font=('Times New Roman',14))
entDateE.place(x=120,y=180,width=300)


lblNewTask=tk.Label(editBody,text='New Task Name: ',bg='#ece0ba',fg='black')
lblNewTask.config(font=('Times New Roman',18))
lblNewTask.place(x=500,y=120)

entNewTask=tk.Entry(editBody,bg='#C0C5CE',fg='black')
entNewTask.config(font=('Times New Roman',14))
entNewTask.place(x=650,y=120,width=300)


lblNewDate=tk.Label(editBody,text='New Date: ',bg='#ece0ba',fg='black')
lblNewDate.config(font=('Times New Roman',18))
lblNewDate.place(x=500,y=180)

entNewDate=tk.Entry(editBody,bg='#C0C5CE',fg='black')
entNewDate.config(font=('Times New Roman',14))
entNewDate.place(x=650,y=180,width=300)


lblNewDuration=tk.Label(editBody,text='New Duration: ',bg='#ece0ba',fg='black')
lblNewDuration.config(font=('Times New Roman',18))
lblNewDuration.place(x=500,y=240)

entNewDuration=tk.Entry(editBody,bg='#C0C5CE',fg='black')
entNewDuration.config(font=('Times New Roman',14))
entNewDuration.place(x=650,y=240,width=300)


lblNewDescrip=tk.Label(editBody,text='New Description: ',bg='#ece0ba',fg='black')
lblNewDescrip.config(font=('Times New Roman',18))
lblNewDescrip.place(x=500,y=300)

entNewDescrip=tk.Entry(editBody,bg='#C0C5CE',fg='black')
entNewDescrip.config(font=('Times New Roman',14))
entNewDescrip.place(x=650,y=300,width=300,height=100)


def submit():
    pass

btnSubmit=tk.Button(editBody,text='Submit',command=submit,width=5)
btnSubmit.place(x=865,y=425)
btnSubmit.config(font=('Times New Roman',20))


def home():
    pass

btnHome=tkmac.Button(editBody,text='Main Page',command=home,
                  bg='#ece0ba')
btnHome.place(x=10,y=5)
btnHome.config(font=('Times New Roman',14))

wdEdit.mainloop()
```

Figure 2.5: Edit Task Window

Search Page:

The Search Page allows the user to type in the task name, date, or duration in the search box and any tasks containing the searched entry will appear in the list box. If the user selects a task from the list box and then clicks the More Info button, all of the info associated with that task will appear on the right. Lastly, there is a Main Page button on the top left that allows the user to return to the main menu.

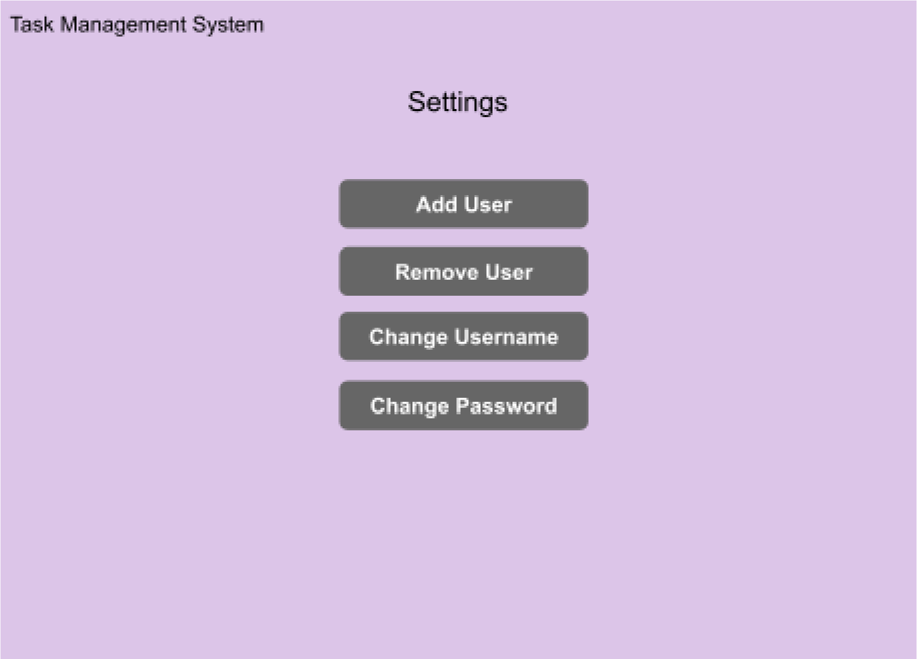


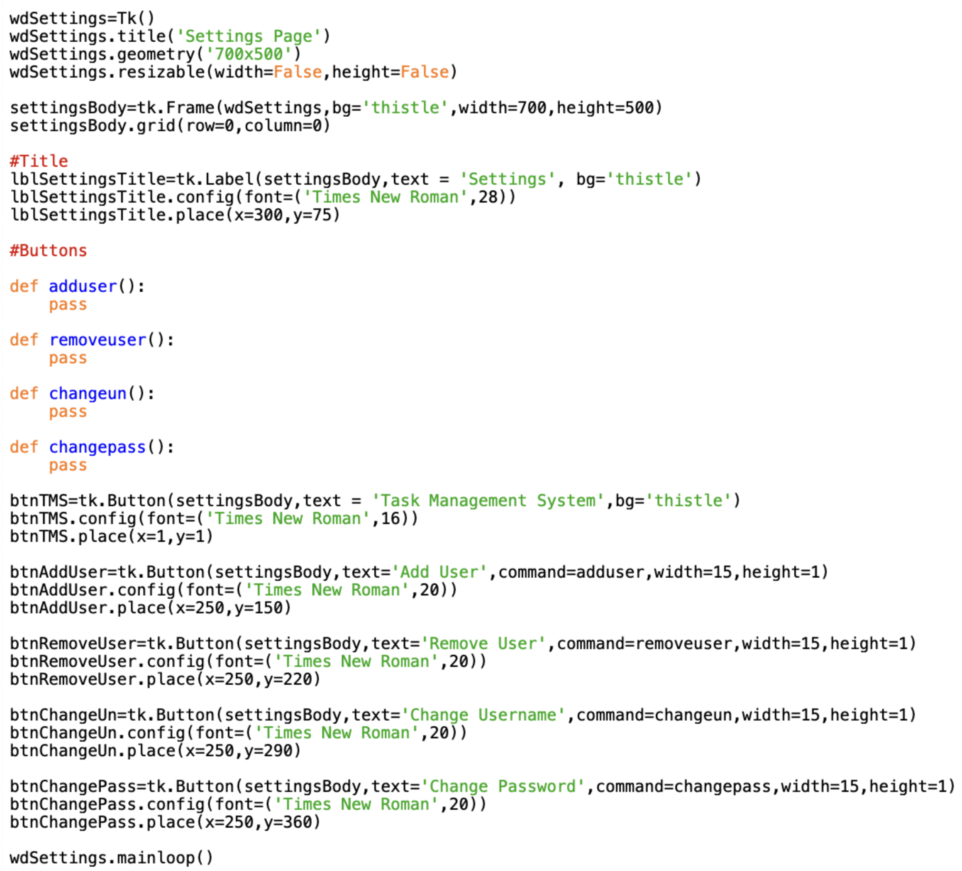


```
def search():
    pass

wdSearch=Tk()
wdSearch.title('Search Page')
wdSearch.geometry('700x500')
wdSearch.resizable(width=False,height=False)

searchBody=tk.Frame(wdSearch,bg='#dcedc1',width=700,height=500)
searchBody.grid(row=0,column=0)

#Title
lblSearchTitle=tk.Label(searchBody,text = 'Search Task', bg='#dcedc1',fg='black')
lblSearchTitle.config(font=('Times New Roman',28))
lblSearchTitle.place(x=280,y=70)

btnTMS=tkmac.Button(searchBody, text = 'Task Management System',bg='#dcedc1')
btnTMS.config(font=('Times New Roman',16))
btnTMS.place(x=1,y=1)

#Search
entSearch=tk.Entry(searchBody,bg='#ddf2fd',fg='black')
entSearch.place(x=40,y=130,height=25,width=185)

btnSearch=tkmac.Button(searchBody,text='Search',command=search,bg='grey',fg='white')
btnSearch.config(font=('Times New Roman',13))
btnSearch.place(x=230,y=130)


txtResults=tk.Text(searchBody,height=20,width=30,bg='gainsboro')
txtResults.place(x=40,y=165)

txtTaskName=tk.Text(searchBody,height=2,width=20,bg='#dcedc1')
txtTaskName.place(x=450,y=150)

lblSearchTitle=tk.Label(searchBody,text="Title: ",bg='#dcedc1',fg='black')
lblSearchTitle.config(font=("Times New Roman",18))
lblSearchTitle.place(x=350,y=200)
entSearchTitle = tk.Entry(searchBody,bg='#ddf2fd',fg='black')
entSearchTitle.place(x=480,y=200,height=30,width=200)

lblSearchDate=tk.Label(searchBody,text="Date: ",bg='#dcedc1',fg='black')
lblSearchDate.config(font=("Times New Roman",18))
lblSearchDate.place(x=350,y=245)
entSearchDate = tk.Entry(searchBody,bg='#ddf2fd',fg='black')
entSearchDate.place(x=480,y=245,height=30,width=200)

lblSearchDuration=tk.Label(searchBody,text="Duration: ",bg='#dcedc1',fg='black')
lblSearchDuration.config(font=("Times New Roman",18))
lblSearchDuration.place(x=350,y=290)
entSearchDuration = tk.Entry(searchBody,bg='#ddf2fd',fg='black')
entSearchDuration.place(x=480,y=290,height=30,width=200)

lblSearchDescription=tk.Label(searchBody,text="Description: ",bg='#dcedc1',fg='black')
lblSearchDescription.config(font=("Times New Roman",18))
lblSearchDescription.place(x=350,y=335)
entSearchDescription = tk.Entry(searchBody,bg='#ddf2fd',fg='black')
entSearchDescription.place(x=480,y=335,height=70,width=200)

wdSearch.mainloop()
```

Figure 2.6: Search Window

Settings Page:
The Settings Page provides the user the option to add another user, remove a user, change a username, or change a password. If any of these buttons are clicked, a new window appears to allow them to perform that task.



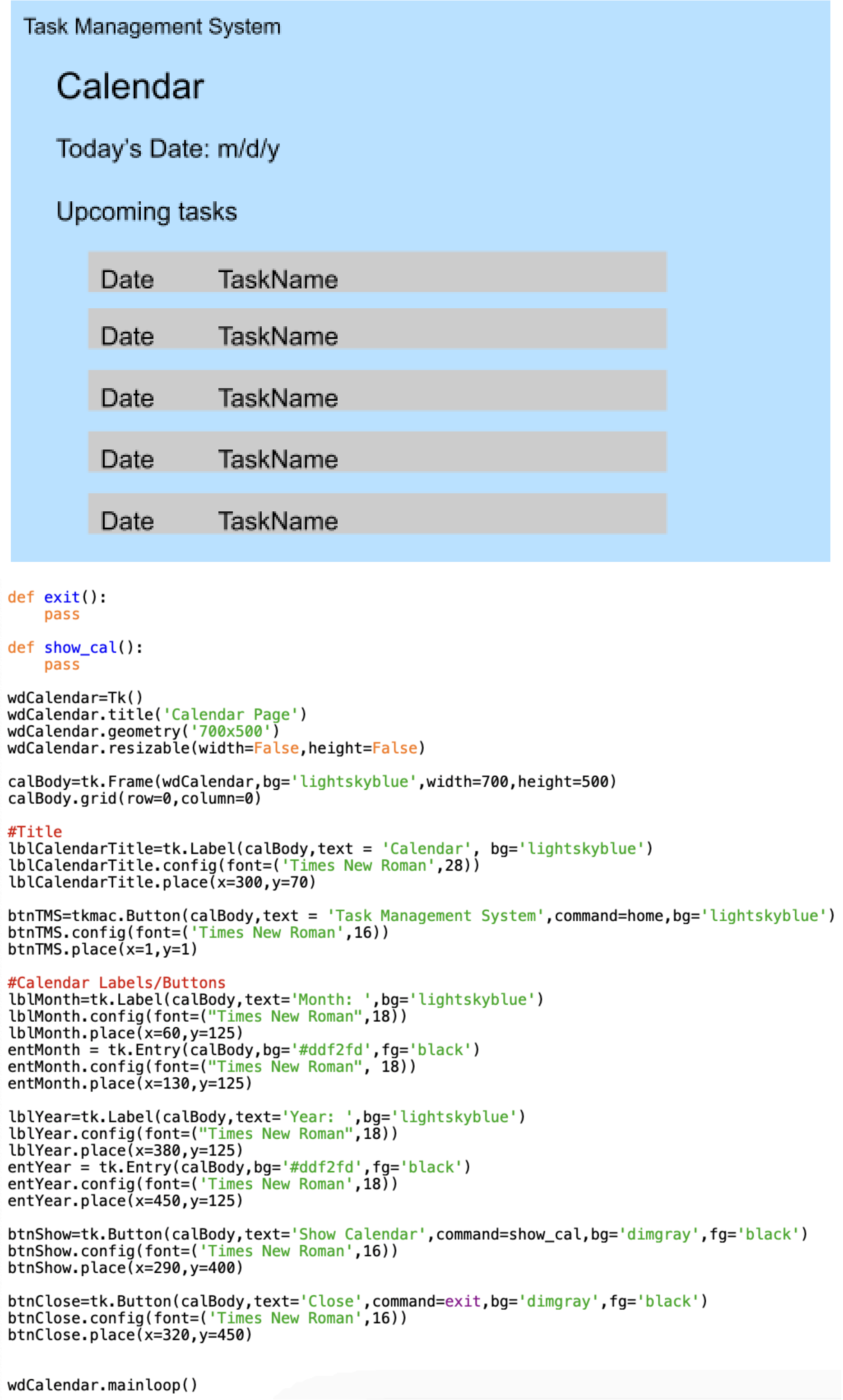


```
wdSettings=Tk()
wdSettings.title('Settings Page')
wdSettings.geometry('700x500')
wdSettings.resizable(width=False,height=False)

settingsBody=tk.Frame(wdSettings,bg='thistle',width=700,height=500)
settingsBody.grid(row=0,column=0)

#Title
lblSettingsTitle=tk.Label(settingsBody,text = 'Settings', bg='thistle')
lblSettingsTitle.config(font=('Times New Roman',28))
lblSettingsTitle.place(x=300,y=75)

#Buttons

def adduser():
    pass

def removeuser():
    pass

def changeun():
    pass

def changepass():
    pass

btnTMS=tk.Button(settingsBody,text = 'Task Management System',bg='thistle')
btnTMS.config(font=('Times New Roman',16))
btnTMS.place(x=1,y=1)

btnAddUser=tk.Button(settingsBody,text='Add User',command=adduser,width=15,height=1)
btnAddUser.config(font=('Times New Roman',20))
btnAddUser.place(x=250,y=150)

btnRemoveUser=tk.Button(settingsBody,text='Remove User',command=removeuser,width=15,height=1)
btnRemoveUser.config(font=('Times New Roman',20))
btnRemoveUser.place(x=250,y=220)

btnChangeUn=tk.Button(settingsBody,text='Change Username',command=changeun,width=15,height=1)
btnChangeUn.config(font=('Times New Roman',20))
btnChangeUn.place(x=250,y=290)

btnChangePass=tk.Button(settingsBody,text='Change Password',command=changepass,width=15,height=1)
btnChangePass.config(font=('Times New Roman',20))
btnChangePass.place(x=250,y=360)

wdSettings.mainloop()
```

Figure 2.7: Settings Window

Calendar Page:

The Calendar Page acts as a "To Do" list and shows all upcoming tasks sorted by date.



```
def exit():
    pass

def show_cal():
    pass

wdCalendar=Tk()
wdCalendar.title('Calendar Page')
wdCalendar.geometry('700x500')
wdCalendar.resizable(width=False,height=False)

calBody=tk.Frame(wdCalendar,bg='lightskyblue',width=700,height=500)
calBody.grid(row=0,column=0)

#Title
lblCalendarTitle=tk.Label(calBody,text = 'Calendar', bg='lightskyblue')
lblCalendarTitle.config(font=('Times New Roman',28))
lblCalendarTitle.place(x=300,y=70)

btnTMS=tkmac.Button(calBody,text = 'Task Management System',command=home,bg='lightskyblue')
btnTMS.config(font=('Times New Roman',16))
btnTMS.place(x=1,y=1)

#Calendar Labels/Buttons
lblMonth=tk.Label(calBody,text='Month:  ',bg='lightskyblue')
lblMonth.config(font=("Times New Roman",18))
lblMonth.place(x=60,y=125)
entMonth = tk.Entry(calBody,bg='#ddf2fd',fg='black')
entMonth.config(font=("Times New Roman", 18))
entMonth.place(x=130,y=125)

lblYear=tk.Label(calBody,text='Year:  ',bg='lightskyblue')
lblYear.config(font=("Times New Roman",18))
lblYear.place(x=380,y=125)
entYear = tk.Entry(calBody,bg='#ddf2fd',fg='black')
entYear.config(font=('Times New Roman',18))
entYear.place(x=450,y=125)

btnShow=tk.Button(calBody,text='Show Calendar',command=show_cal,bg='dimgray',fg='black')
btnShow.config(font=('Times New Roman',16))
btnShow.place(x=290,y=400)

btnClose=tk.Button(calBody,text='Close',command=exit,bg='dimgray',fg='black')
btnClose.config(font=('Times New Roman',16))
btnClose.place(x=320,y=450)


wdCalendar.mainloop()
```

Figure 2.8: Calendar Page Window

Data Storage

The first CSV file that the program uses is the "*users.csv*" file which stores all of the usernames and passwords for the program. This CSV file is used and/or referenced in the **Login** window, the **Add User** window, the **Remove User** window, the **Change Username** window, and the **Change Password** window. From the **Login** window, the user has the option of entering an existing username and password to continue to the main page, or the user can click the *New User* button to create a username and password for themselves. Once logged in, the user has the option to add, remove, and change their username or password in the **Settings** window.

The other CSV files associated with our Task Management System are the CSV files created for every user when they add a task. When the user adds a task, the code checks to see if the user already has a CSV file under their username. If no file exists, then the code creates one and adds 4 columns titled *Task Name, Date, Duration,* and *Description*. When the user goes to the **Edit Task** window, the program opens the existing CSV file under the user's username, and changes whatever the user wishes to change. In the **Remove Task** window, the code asks the user to choose a task they would like to remove from the listbox and then click remove. Following that, the code opens the CSV file and deletes the task that the user chose to remove. The **Calendar** window displays all the user's upcoming tasks sorted by date by opening the CSV file associated with that current user.

The last CSV file that is associated with our program is a temporary CSV file that is used when the user chooses to change their username or password. When the user changes their username or password, the code opens a temporary CSV file that duplicates what is in the original *"users.csv"* file except for the new username or password that they are changing. Then the code replaces the *"users.csv"* file for the temporary one and changes the name back to *"users.csv"*.