Goal Journal

Introduction to Programming

CMPT 120L

Marist College

School of Computer Science and Mathematics

Submitted To:

Dr. Reza Sadeghi

Date: April 4, 2024

Project Progress Report 1 of Project Name

Team Name

Team Sean

Team members

1. Sean Bruno sean.bruno1@marist.edu (Team Head)

Description of Team Members

1. My name is Sean Bruno, I like working by myself and making sure all the work is done. Especially when it comes to a project, so I have decided to peruse a group of just me. I have learned a lot about coding in this course as I have never coded before.

Table of Contents

Project Objective

Project Title: Goal Journal Managment System

Summary: My project will be a Goal Journal Managment System, which will be very similar to the Diary Managment System which was given in project sample 2. The Goal Journal will be available for the user to journal an achievement or when they complete something that day. For example, if their overall goal is to go to the gym more, then the user will be able to input when they went to the gym and can leave an entry with it. The data will be stored in a CSV file which stands for Comma Separated Values. The GJMS will be able to provide the following capabilities:

1. Requesting admin user and password for log in (a string of at least 8 characters)

2. Capability to change the admin user and password

3. Admin user should be able to add a user to GJMS by creating a new username and password for normal users, who are not able to define or remove a user

4. Admin user should be able to remove a user from GJMS by removing his/her username, password, and his/her other corresponding data

5. Each user should be able to:

A. Add a Journal record with the following details: time, date, duration, and description

B. Remove a Journal record

C. Edit a Journal record's details

D. Search through Journals based on time or duration and list the results on the screen. For Instance, it should be able to list all scheduled works for one day

6. GJMS should be a user-friendly software, such that:

A. It shows a warning if a user tries to put a new task with an overlapping time to the previous records.

B. GJMS should show a welcome page

C. GJMS should show a menu of all functions to the user

D. GJMS should provide the reports in a tabular form

E. GJMS should provide an exit function

GitHub Address

https://github.com/SeanBruno2/111_GoalJournalManagmentSystem_TeamSean.git

Graphical User Experience Design

In my Goal Journal Managment System I plan on having 6 pages. Including a login, main menu, adding a journal entry, removing a journal entry, editing a journal entry, searching for a journal entry, and a set password page.

Login:

Input: Username and Password

Output: Correct username and password will be granted access to the system. While an incorrect username and password will be sent a warning message saying incorrect.

Adding a Journal Entry:

Input: Time, date, duration, and description of something you completed to be closer to your goal. For example, Time:1230pm, Date:2/4/2024, Duration:60 minutes, Description: Went to gym for an hour to workout.

Output: Journal entry will be added to the csv value where all entry's will be stored.

Removing a Journal Entry:

Input: Time, date, duration, and description of journal entry you would like to remove.

Output: This journal entry will be removed from storage

Editing Journal Entry:

Input: Input: Time, date, duration, and description with the correction to the journal entry that you would like to edit.

Output: Entry will be edited in the csv file.

Search:

Input: Date and Time of entry you are searching for.

Output: The Journal entry that corresponds to the given date and time

Set Password:

Input: Username and Password in which you would like associated to account.

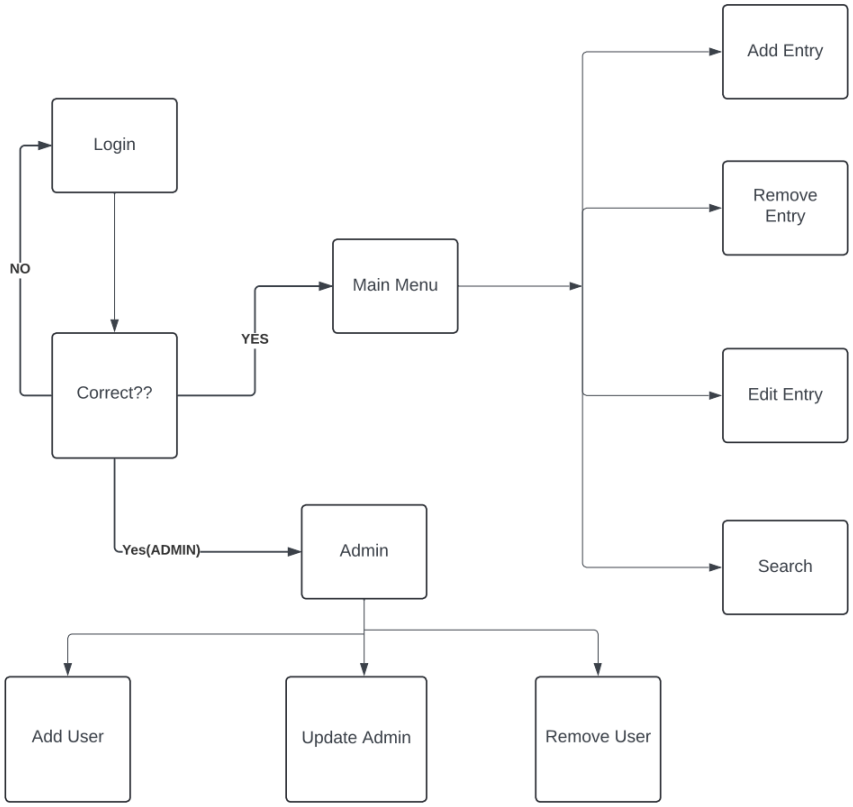
Output: The password corresponding to the username will be updated.

Administrative:

Input: Adding a User or Removing a User

Output: User will be added or removed

Flow Chart



Packages to be used

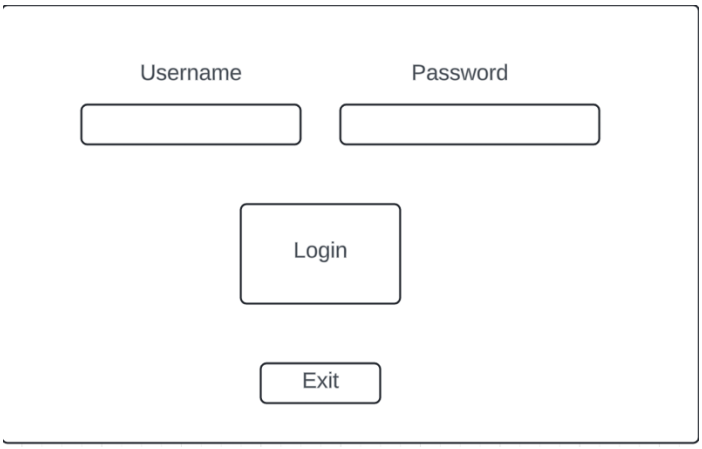
- OS
- Tkinter
- Csv

Virtual Environment

- These three packages are already installed. As the project goes on I hope to install more packages to better mine and the users experience.

GUI Design

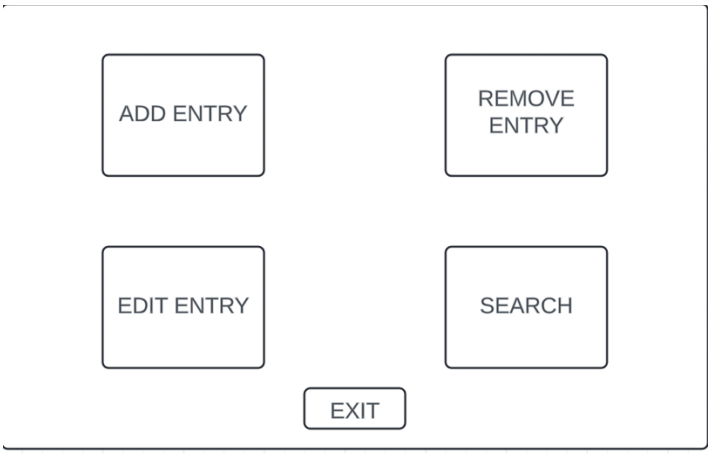
Login




```
import tkinter as tk

login=tk.Tk()
login.title('Login')
login.geometry("500x130")

lblUser=tk.Label(text="Username",font=("Arial",16))
lblUser.grid(row=0,column=0)
entUser=tk.Entry()
entUser.grid(row=1,column=0)

lblPass=tk.Label(text="Password",font=("Arial",16))
lblPass.grid(row=0,column=2)
entPass=tk.Entry()
entPass.grid(row=1,column=2)

btnLogin=tk.Button(text="Login")
btnLogin.grid(row=2,column=1)

btnExit=tk.Button(text="Exit")
```

```
btnExit.grid(row=3,column=1)
```

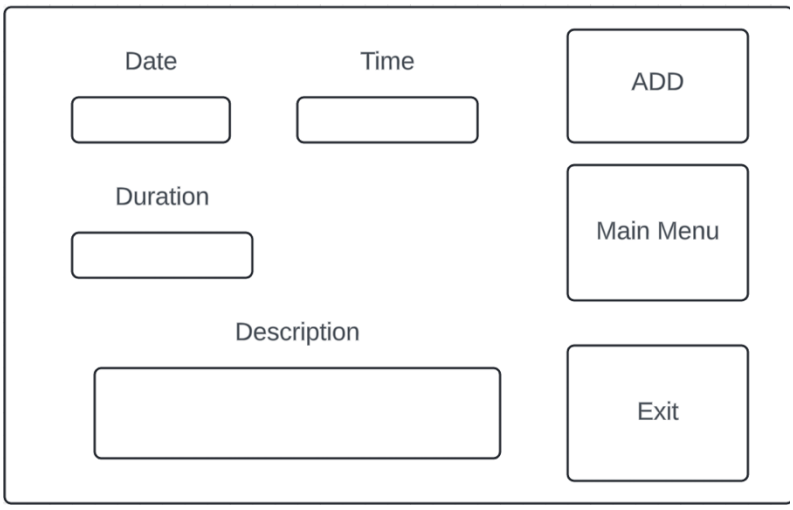
Main Menu




```
import tkinter as tk

main=tk.Tk()
main.title("Main Meu")
main.geometry("300x300")

btnAdd=tk.Button(text="Add\nEntry")
btnAdd.grid(row=0,column=0,sticky='news')

btnRemove=tk.Button(text="Remove\nEntry")
btnRemove.grid(row=0,column=2,sticky='news')

btnEdit=tk.Button(text="Edit\nEntry")
btnEdit.grid(row=1,column=0,sticky='news')

btnSearch=tk.Button(text="Search\nEntry")
btnSearch.grid(row=1,column=2,sticky='news')
```

```
btnExit=tk.Button(text="Exit")
btnExit.grid(row=3,column=1,sticky='N')
```

ADD

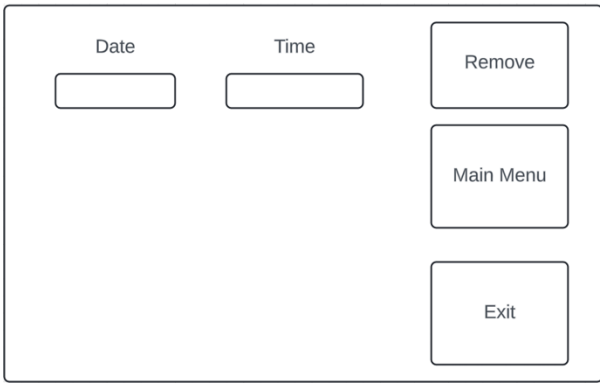



```
import tkinter as tk


Add=tk.Tk()
Add.geometry("700x180")
Add.title("Add")



lblDate=tk.Label(Add,text="Date(mm-dd-
yyyy)",font=("Arial",16))
lblDate.grid(row=0,column=0)
entDate=tk.Entry()
entDate.grid(row=1,column=0)
```

```
lblTime=tk.Label(Add,text="Time(00:00)",font=("Arial",16))
lblTime.grid(row=0,column=2)
entTime=tk.Entry()
entTime.grid(row=1,column=2)

lblDuration=tk.Label(Add,text="Duration(In minutes)",font=("Arial",16))
lblDuration.grid(row=2,column=0)
entDuration=tk.Entry()
entDuration.grid(row=3,column=0)

lblDescription=tk.Label(Add,text="Description",font=("Arial",16))
lblDescription.grid(row=4,column=1)
entDescription=tk.Entry()
entDescription.grid(row=5,column=1)

btnAdd=tk.Button(Add,text="Add")
btnAdd.grid(row=0,column=3)

btnMain=tk.Button(Add,text="Main Menu")
btnMain.grid(row=1,column=3)

btnExit=tk.Button(Add,text="Exit")
```

```
btnExit.grid(row=2,column=3)
```

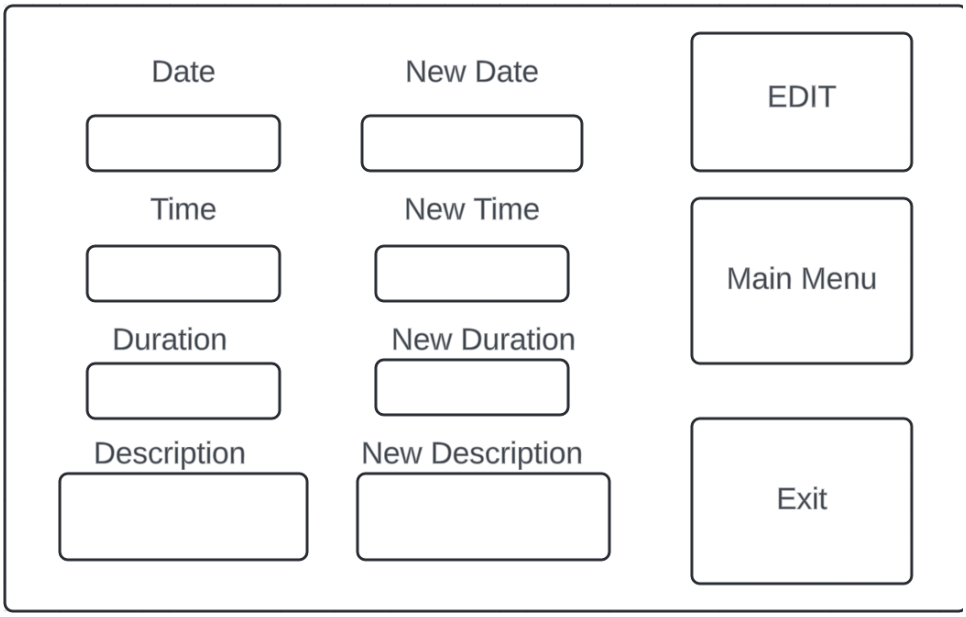
Remove




```
import tkinter as tk


Remove=tk.Tk()
Remove.geometry("550x140")
Remove.title("Remove")



lblDate=tk.Label(Remove,text="Date(mm-dd-
yyyy)",font=("Arial",16))
lblDate.grid(row=0,column=0)
entDate=tk.Entry()
entDate.grid(row=1,column=0)


lblTime=tk.Label(Remove,text="Time(00:00)",font=("Arial
",16))
lblTime.grid(row=0,column=1)
```

```
entTime=tk.Entry()
entTime.grid(row=1,column=1)


btnRemove=tk.Button(Remove,text="Remove")
btnRemove.grid(row=0,column=2)

btnMain=tk.Button(Remove,text="Main Menu")
btnMain.grid(row=1,column=2)

btnExit=tk.Button(Remove,text="Exit")
btnExit.grid(row=2,column=2)
```

Edit

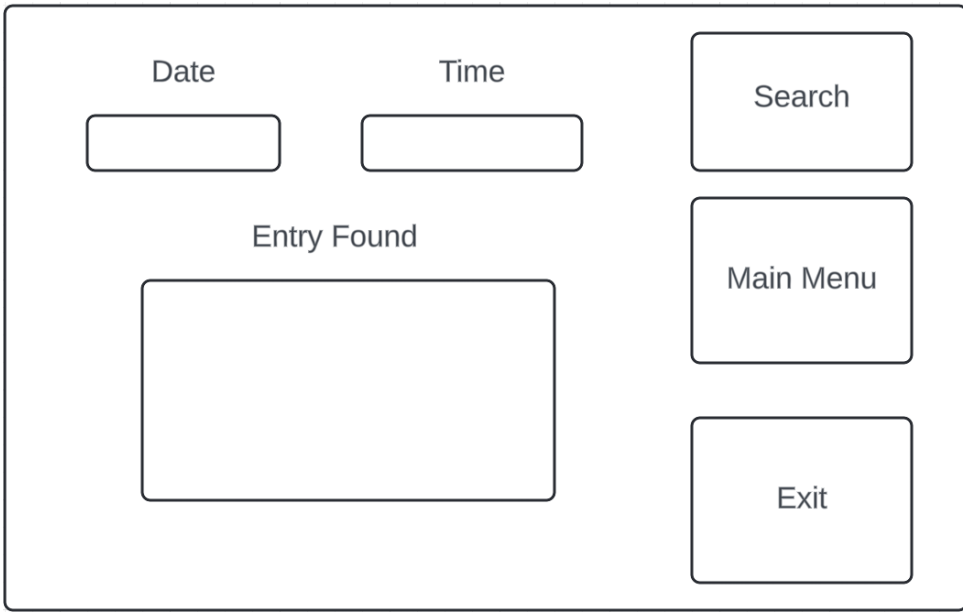



```
import tkinter as tk
```

```
Edit=tk.Tk()
Edit.geometry("550x230")
Edit.title("Remove")


lblDate=tk.Label(Edit,text="Date(mm-dd-
yyyy)",font=("Arial",16))
lblDate.grid(row=0,column=0)
entDate=tk.Entry()
entDate.grid(row=1,column=0)

lblNDate=tk.Label(Edit,text="New Date(mm-dd-
yyyy)",font=("Arial",16))
lblNDate.grid(row=0,column=1)
entNDate=tk.Entry()
entNDate.grid(row=1,column=1)

lblTime=tk.Label(Edit,text="Time(00:00)",font=("Arial",
16))
lblTime.grid(row=2,column=0)
entTime=tk.Entry()
entTime.grid(row=3,column=0)

lblNTime=tk.Label(Edit,text="New
Time(00:00)",font=("Arial",16))
lblNTime.grid(row=2,column=1)
```

```
entNTime=tk.Entry()
entNTime.grid(row=3,column=1)


lblNDur=tk.Label(Edit,text="New
Duration(Minutes)",font=("Arial",16))
lblNDur.grid(row=4,column=1)
entNDur=tk.Entry()
entNDur.grid(row=5,column=1)

lblDes=tk.Label(Edit,text="New
Desciption",font=("Arial",16))
lblDes.grid(row=6,column=1)
entDes=tk.Entry()
entDes.grid(row=7,column=1)


btnEdit=tk.Button(Edit,text="Edit")
btnEdit.grid(row=0,column=2)

btnMain=tk.Button(Edit,text="Main Menu")
btnMain.grid(row=1,column=2)

btnExit=tk.Button(Edit,text="Exit")
btnExit.grid(row=2,column=2)
```

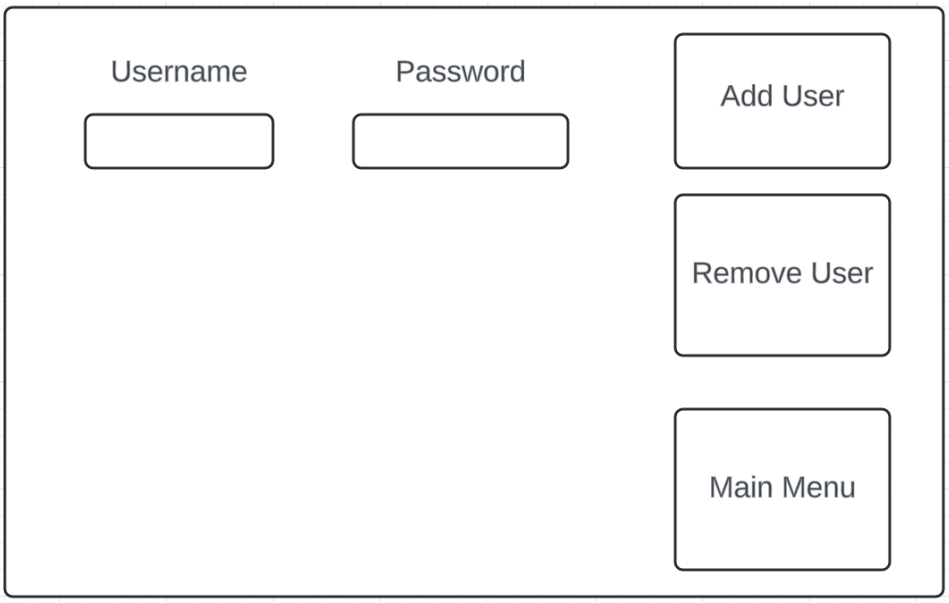
Search




```
import tkinter as tk

Search=tk.Tk()
Search.geometry("677x180")
Search.title("Search")


lblDate=tk.Label(Search,text="Date(mm-dd-
yyyy)",font=("Arial",16))
lblDate.grid(row=0,column=0)
entDate=tk.Entry()
entDate.grid(row=1,column=0)

lblFound=tk.Label(Search,text="Entry
Found",font=("Arial",16))
```

```
lblFound.grid(row=3,column=1)
entFound=tk.Entry()
entFound.grid(row=4,column=1)

lblTime=tk.Label(Search,text="Time(00:00)",font=("Arial
",16))
lblTime.grid(row=0,column=2)
entTime=tk.Entry()
entTime.grid(row=1,column=2)



btnSearch=tk.Button(Search,text="Search")
btnSearch.grid(row=0,column=3)

btnMain=tk.Button(Search,text="Main Menu")
btnMain.grid(row=1,column=3)

btnExit=tk.Button(Search,text="Exit")
btnExit.grid(row=2,column=3)
```

Admin




```
import tkinter as tk

Admin=tk.Tk()
Admin.geometry("550x150")
Admin.title("Admin")


lblUser=tk.Label(Admin,text="Username",font=("Arial",16))
lblUser.grid(row=0,column=0)
entUser=tk.Entry()
entUser.grid(row=1,column=0)

lblPass=tk.Label(Admin,text="Password",font=("Arial",16))
lblPass.grid(row=0,column=1)
entPass=tk.Entry()
entPass.grid(row=1,column=1)
```

```
btnAdd=tk.Button(Admin,text="Add\nUser")
btnAdd.grid(row=0,column=2)

btnRemove=tk.Button(Admin,text="Remove\nUser")
btnRemove.grid(row=1,column=2)

btnMain=tk.Button(Admin,text="Main\nMenu")
btnMain.grid(row=2,column=2)
```