**Ari G. Eastman**

**Justin F. Harlow**

**Jeremy T. Griffen**

**Group Programming Project I CS150-01**

**2023/10/15**

**Python Daily-Journal Report**

# **The Abstract**

The goal of the Group Programming Project was to concisely show independent knowledge based on in-class teachings and findings to explore python in a semi-independent group project form. The task was to pick an exploration of python that was given by the project guidelines and apply the challenges of each choice given what the group chooses. The Daily Journal entries was the chosen project as it was concise and relatively straightforward given its use of simple user entry tasks. The project, through python, had the goal of applying user entries, allowing for users to enter responses, images, and tags, and to be able to search for passed entries based on these user specific inputs. The methodology of the project came down to using class and defining instances of said journal class that would allow for all of these aspects to be appended & fulfilled in the daily journal. The outcomes appeared successful through using this learned knowledge of python classes and instances as well as lists and appending to said lists and dictionaries. This created a user friendly and concise code that showed a daily-journal anyone could use.

# **Introduction To Journal**

## ***Context***

The project worked to develop & code through python to make a user based experience that was defined by the group this included a fitness tracker, a budget manger, and much more. Additionally it was defined that each group would need to use ​​Functions, Loops, Conditions, Formatted Strings, and List/Dictionary to have a baseline before starting. While the python code itself was the main objective it also included formatting a project and report that would coincide with the chosen programming made by the group. As a group, the chosen task from the project guideline was the Daily Journal project which tasked us to develop a user entry experience where they can write daily entries in a digital journal. Allow users to add images, tags, and dates to each entry. Additionally the project specified that the programmers must provide a search function to retrieve entries through tags, entries, etc. From all of this we began to brainstorm and work towards this programmed user experience and started with the skeleton using classes and instances and the challenges it would create.

## ***Challenges***

A few of the larger challenges we saw in the future of the project before beginning was communicating how we developed this and programmed it in python in a synthesized way and additionally how images would play a role in the daily journal experience. While these sound simple it was a larger challenge to adjust what we felt was right for both of these challenges with even the teacher specification they were curious on how the images would be implemented in the journal entries. Part of what was the problem would be if images in the entries would display or would it be a link? The idea would be that it would be cluttered or too layered for the user experience so throughout the project we interpreted what worked to be the best user experience in our programming project. Additionally, While developing the Daily Journal, one challenge we faced was implementing the "search entry" function. As more functionalities are added, the "search entry" function becomes longer and more complex, making it hard to read and figure out what is going on. To overcome this, we implemented a for loop that checks for each "date," "entry," and "tags."​

# **Group Methodology**

## ***technologies and tools***

The technology used was python through the use of jupyter notebook in anaconda's navigator. From this, it allowed the group to apply our programming knowledge like many other assignments in a .ipynb file. The tools used were based on class materials from CS150-01 which included notebooks and programming assignments to take inspiration and reference from to build the framework of a successful project.

## ***development process***

The development processes included segmenting the work into what was given to start being classes, functions, while loops, and if-else statements. These would act as the skeleton of design for implementing the daily journal. The journal itself would be defined as a class “class DailyJounral:” and would have the required user applications of the journal of search and input entries being defined in the class through the “def” function. The use of init, this would store journal entries from the user and make it able to search for entries given the instances created of the daily journal class. All of this combined with user inputs with the “def” functions would create the daily journal and the makings of the programming task.

# **Design and Implementation**

## ***architecture and design***

The design of the Daily Journal would come in the form of classes, functions, while loops, and if-else statements. Classes are defined as blueprints that define the attributes and behaviors of an object. It has data and functions that operate on that data.​ Functions work as blocks of code that perform a specific task.​ With While loops they repeatedly execute a code as long as a condition is true.​ Finally The daily journal would implement If-else statements which are used for decision-making in programming. They allow code to take different paths based on conditions.​ All of these aspects of python programming and knowledge from course material obtained would work to make the daily journal successful for programming and user accessibility.

## ***key features***

The features of the Daily Journal task of the project was to create a simple journal application that allows users to log daily entries and be able to search and find previous entries.​ The key features included the ability for the user can input new entries with date, text, tags, and optional images.​ Users can search for a previous entry using date, text, or tags used in a past entry.​ Important Code:​ Class, Functions, While loop, If-else statements.​

# **Challenges & Solutions**

## ***The Challenges***

As stated the challenges came from how to implement aspects like images and additionally simplifying this information in a concise presentation. One of the issues that was brought up by working was figuring out why errors happen and how to fix them. While developing the Daily Journal, another challenge we faced was implementing the "search entry" function. As more functionalities were added, the "search entry" function became longer and more complex, making it hard to read and figure out what is going on for the programmer and the user.

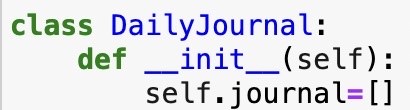
## ***The Solutions***

To overcome many of the aspects of the “Search Entry” function for the user the daily journal implemented a for loop that checks for each date, entry, and tags. For the image implementation, for the user and daily entries it was decided to program the input where users could enter the url through the user entry to hold an images address which would allow a user to click on the image url address and then bring the image up in a tab of their digital journal.

# 

# 

# **Results and Findings:**



## **\_\_init\_\_(self):**

*Initializes an instance of the class. It creates an empty list called journal to store journal entries.*



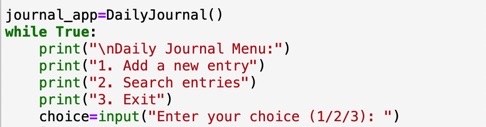
## **add\_entry(self, date, entry, tags, image):**

*This adds a new entry to the journal. It takes in date, the journal text, tags, and an optional image. It creates a dictionary with these aspects and appends it to the journal list.*



## **search\_entries(self, key):**

This is for searching entries based on a key. It goes through the journal list and looks for matches in the date, entry, or tags of each entry. If it finds matches, it prints the date, entry, tags, and image, if there is one, for each matching entry. Else, it prints "No matching entries found".



## **Journal\_app**

*An instance of the DailyJournal class is created: journal\_app = DailyJournal(). A while loop will present a menu to the user until the user chooses to exit.*



## **New Entry (choice == '1')**

*Add a New Entry (choice == '1') will ask the user for the date, journal entry text, tags (comma-separated), and an image. It then calls the add\_entry function of the journal\_app object to add this entry to the journal.*



## **Search Entries (choice == '2')**

*Search Entries (choice == '2') will ask the user for a key to search entries. It then calls the search\_entries function of the journal\_app object to find and display entries that match the provided key.*



## **Invalid Choice**

*Exit (choice == '3') will print "Goodbye" and exit the loop. Else, print “invalid choice”*

## 

## **.IPYNB**

*Additionally, our results and findings can be found through the final draft of the Journal on Github:* [*DailyJournal.ipynb*](https://github.com/JGiffen/GP1_JTG_AGE_JFH/blob/main/DailyJournal.ipynb)

# **Discussion:**

## ***Significance:***

To move out of the perspective of the daily journal as an assignment, how is it practical? Throughout creating the journal in python it shows the implication of digital user experiences and how critical that is in the modern world. Python is a vital tool and language that everyone interprets now in the digital age and when programming the daily journal acts as a primary example of allowing individuals to make interpretations based on interacting with just code. Translating a mass digital language into intractable applications provides a more accessible world and that was one of the major takeaways from the task as it was drafted and worked on to its final product.

## ***lessons learned:***

## Throughout working on the daily journal, the insights gained were the optimizing of the functionality of Python such as better utilizing a for loop given our search entry.​ As well the programming informed the group on how to better design code with functions and classes, as well as implementing user interaction.​ Most informative was the challenges of how to better handle errors and how to fix them when trying to complete a programming task.​ In the future, if given more time, the daily journal would have implemented more capabilities of editing entries and additionally searching concisely for the users entries as well.

# **Conclusion:**

The project was informative in experience but also informed us of our current ability in python & programming with the course ending soon. ​ The daily journal uniquely taught expanded the knowledge of how python utilizes user input and its functionality pertaining to problem solving. The daily journal​ taught how to design and program more user friendly​ experiences as well as the restraints & capabilities of using Multiple computers for python. Ultimately, The daily journal task that was implemented succeeded and worked towards an informative group programming project.

# **References**

Matthes, Eric. *Python Crash Course: A Hands-on, Project-Based Introduction to Programming*. No Starch Press, 2023.

Sarkar, Sayani. “CS-150 Notebooks and Lectures.” CS-150. CS-150, Bellarmine University, Bellarmine University.