Portable Notepad

Software Engineering
ITSC-3155
Final Project Report

12/17/2020

Conner Malloy, Evan Rohrwasser, Gerardo Quintana

Table of Contents

1. Introduction

Have you ever needed to take notes but either don't have an internet connection or don't want to depend on Google Docs or Office 365? Well our completely offline and portable notepad application is for you. For this project we created a portable application that allows a user to take notes on any computer they use without requiring an internet connection. Our solution allows users to be able to use the same, familiar and easy to use note application to access their notes on any computer without relying on cloud based services. Our program allows for creating, editing, storing, and transporting notes offline in a simple and easy to use way.

1.1 Purpose

The purpose of our application is to allow the user to take notes locally or portably, via USB, so that they can have access to their notes offline or on-the-go. Another purpose for this project is to allow users with multiple devices and no internet connection to transfer notes from device to device via USB. Many word processors such as Google Docs and Office 365 that people use for note taking, are increasingly complex for simple note taking tasks. We intended to develop a simpler application that is tailored specifically to simple note-taking tasks while allowing portability of the users notes and the application itself. We want our application to provide the simplest and easiest way to take notes locally or portably, without requiring access to the internet.

1.2 Scope

(1) The software product to be produced in this project is *Portable Notepad.*

(2) Portable Notepad is meant to allow users in an offline environment to generate notes from the portable app and allow for transport of the app and notes on a USB drive between all workstations. They are able to use the same app off the USB without worrying about compatibility issues.

(3) The benefits of this application lie in the context of the environment. Those using the application in a space lacking a network connection will greatly benefit from this application, as it will allow them to run the app and store the notes on the same drive on any workstation. The objective of this product is to support a niche user base who may be working in network limited environments and need a hassle-free solution to transferable note taking. The goal is to successfully deploy a localized note application that is portable. The scope of this project is the time and effort that we are going to have to put in not only building the application but planning testing and debugging and everything else that we are going to have to do. Our objectives are to create a fully functional note taking application that can be run off of a usb drive and work offline. Some other objectives will be to make sure we are meeting deadlines, distributing work evenly, and working as a team.

1.3 Definitions, Acronyms, and Abbreviations

Tkinter - a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python.

USB User - User of our program whose use case is portability and has our program on his/hers USB flash drive.

Local User - User of our program whose use case is simple note taking without using cloud services and has our program anywhere on his/hers computer.

1.4 Modeling and References

Tkinter Documentation - https://docs.python.org/3/library/tkinter.html

Python File Handling - https://www.guru99.com/reading-and-writing-files-in-python.html

Right Click Menu - https://www.geeksforgeeks.org/right-click-menu-using-tkinter/

User Stories - Appendix

Context Diagram - Appendix

Data Flow Diagram - Appendix

Use Case Diagram - Appendix

1.5 Overview

- Prompt user on first run to differentiate use cases
- Default filesystem access after first run for local user
- Creation of ‘Notes’ folder and forced access to ‘Notes’ folder after first run for USB user
- Both user stories have the following features:
  - Top Menu
  - Right-Click Menu
  - Cut/Copy/Paste
  - Select all
  - Undo/Redo
  - Open
  - Save
  - New
  - Exit

2. General Description

Our project idea is a usb portable application that incorporates an init script that sets up constants for the directory for saving and opening notes which will happen through our application that is installed on the usb drive. The problem we are trying to solve is being able to access your notes with an internet connection or offline. This will allow for storing, transporting, and taking notes in an offline fashion to be much easier. Motivation for this project derives from the need for localization of note-taking in environments that lack network connectivity. Environments that are closed off from outside network connectivity include research and development labs, government agencies, or any institution that only operates on an internal network. Essentially, the goal is to make the note app and the subsequent notes portable when the workstation isn't portable itself, but can't access an external network either.

2.1 Project Perspective

The perspective of this project lies within the tasks and subsequent work performed for the tasks. We didn't see the app creation itself as particularly difficult, so we initially thought the project would be on the easier side of things. However, with the necessary steps of Agile development, our assumption was quite far from the truth. Following Agile methodology, we broke down the tasks into modeling, design, coding, and testing. From our view, there was quite a bit of work to be done for this project. It was important for us to structure our tasks by planning, which involved forming tasks descriptions, allocating the tasks to group members, creating task due dates, and updating task status'. Overall, these tasks were time consuming, but manageable; aligning with what we projected.

2.2 Project Components

System Models

- Context Diagram
- Data Flow Diagram
- Use Case 1 - USB User
- Use Case 2 - Local User
- Use Case Diagram

Version Control System

- Git (in conjunction with Github)

Code

- Python Scripting Language
- IDE (PyCharm, etc.)
- Python Standard Library

Deliverables

- Project Proposal
- First Project Demonstration
- Presentation Video
- Final Report
- Powerpoint

2.3 Specific Goals

Our goals for this project were to create an easy to use notepad application for users who needed access to their notes offline. Our main goal was to make the application easy for anyone to use. We did this by creating an init function that automatically opens Portable Notepad when the USB is connected. The application itself is familiar and simple. It is catered to note takers only so there is nothing added to the application to add confusing for the user.

2.4 Overview of programs related to specific goals

So the Portable Notepad application is the program that is related to all of the goals mentioned above. It is very similar to the Notepad application that already exists on every windows machine. This is what makes it so easy to use but offers the key feature that the windows Notepad app does not. Our application after being installed on a USB drive will automatically open when the USB is connected to a computer.

2.5 Assumptions and Dependencies

Assumptions that we made when designing and developing this application were that the user not only had a USB drive but had the need for an application like this. We realize that this application is a very niche product but we do believe that there is a need for it. Dependencies for this application are access to a computer and USB drive if the user wants to make use of the USB functionality.

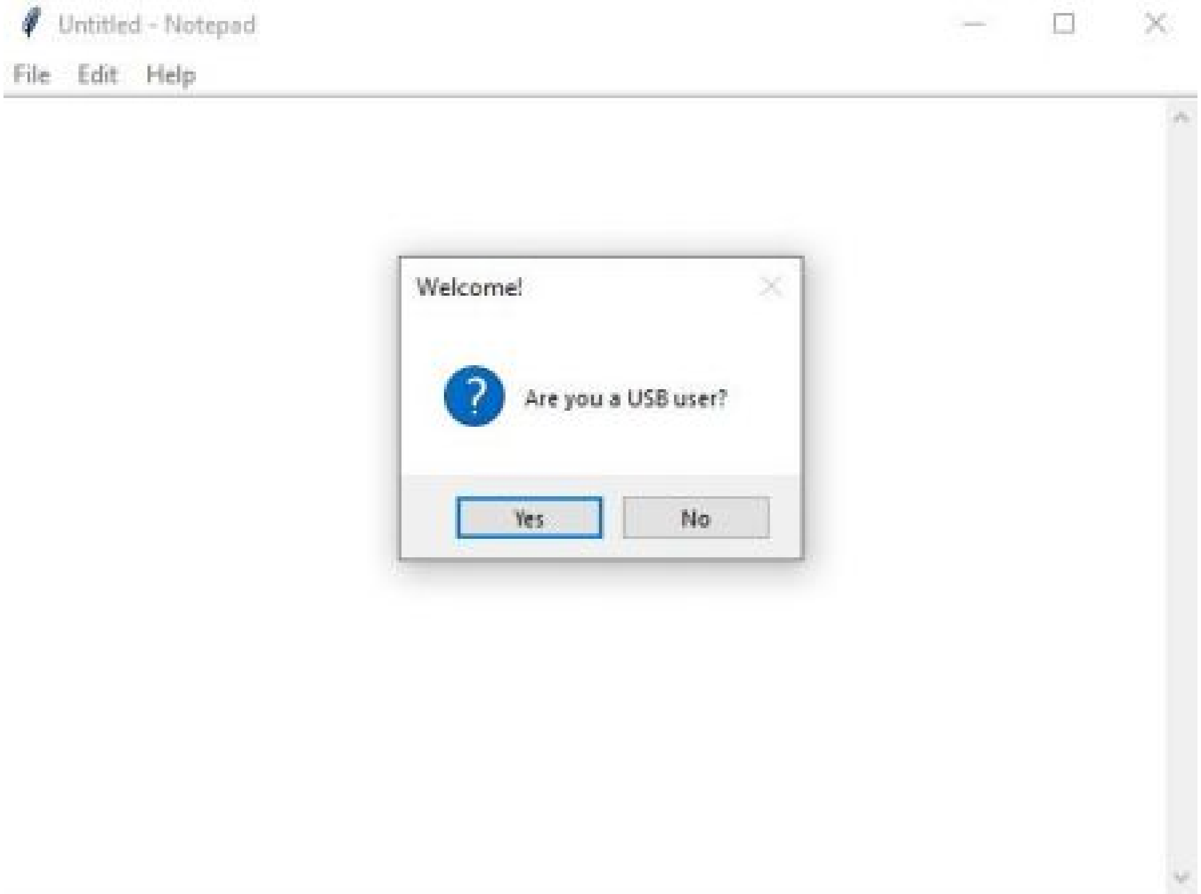
3. Effort

Task	**Estimated Time of Research**	**Actual Time of Research**	**Estimated Coding Effort**	**Actual Coding Effort**
Base Application	~30min	~1hr	Minimal	Moderate
Menu Bar	~30min	~30min	Minimal	Minimal
Right Click Menu	~1hr	~2hr	Moderate	Moderate
User Initialization	~1hr	~1hr	Moderate	Minimal
USB/Local Functionality	~1hr	~2hr	Maximum	Moderate
Save Changes Dialogue	~30min	~20min	Minimal	Minimal
Refactoring/Debugging	~2hrs	~3hrs	Maximum	Maximum

4. Programs Developed

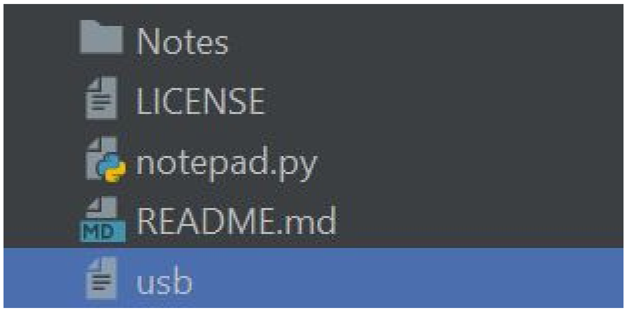
4.1 User Interfaces

When opening the application you will be given a prompt asking if you are using a USB.



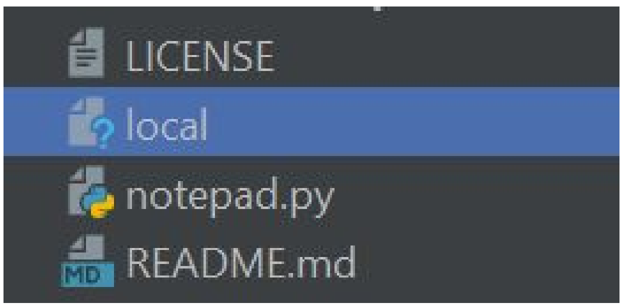
Are you a USB user?

When selecting ‘Yes’ a ‘Notes’ folder will be created in the root of the program directory as well as a usb file to indicate the user type.



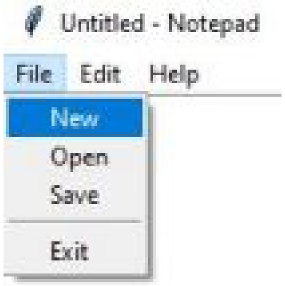
Notes
LICENSE
notepad.py
README.md
usb

When selecting 'No' a local file will be created indicating you are a local user

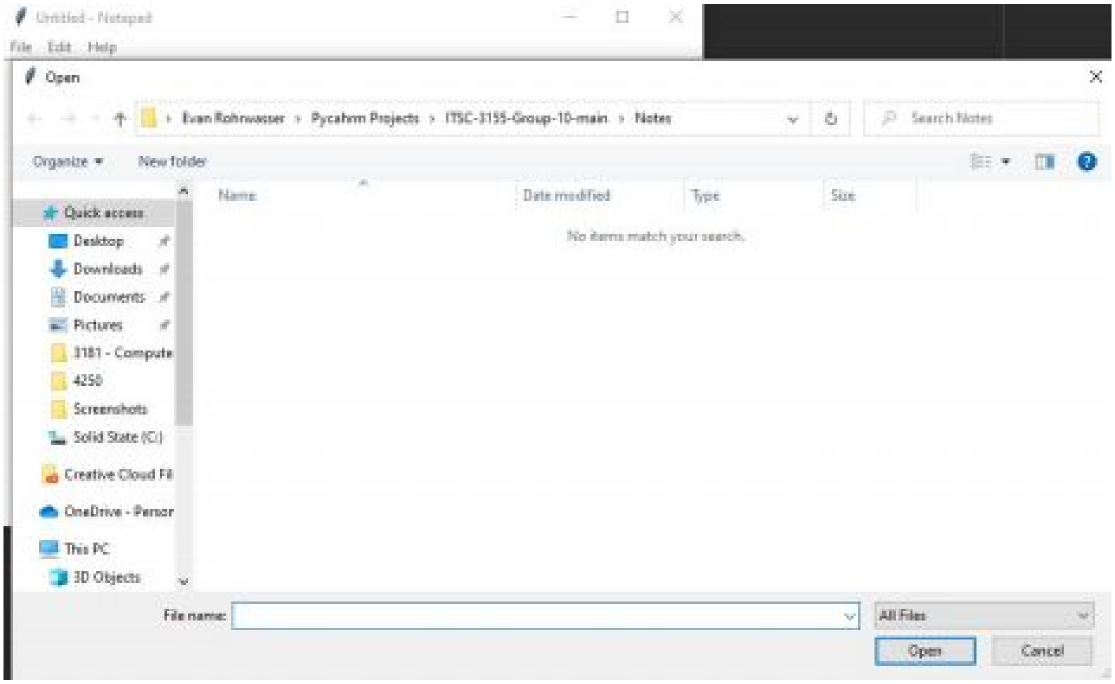


LICENSE
local
notepad.py
README.md

In the file tab we have the options for new, open, save and exit.

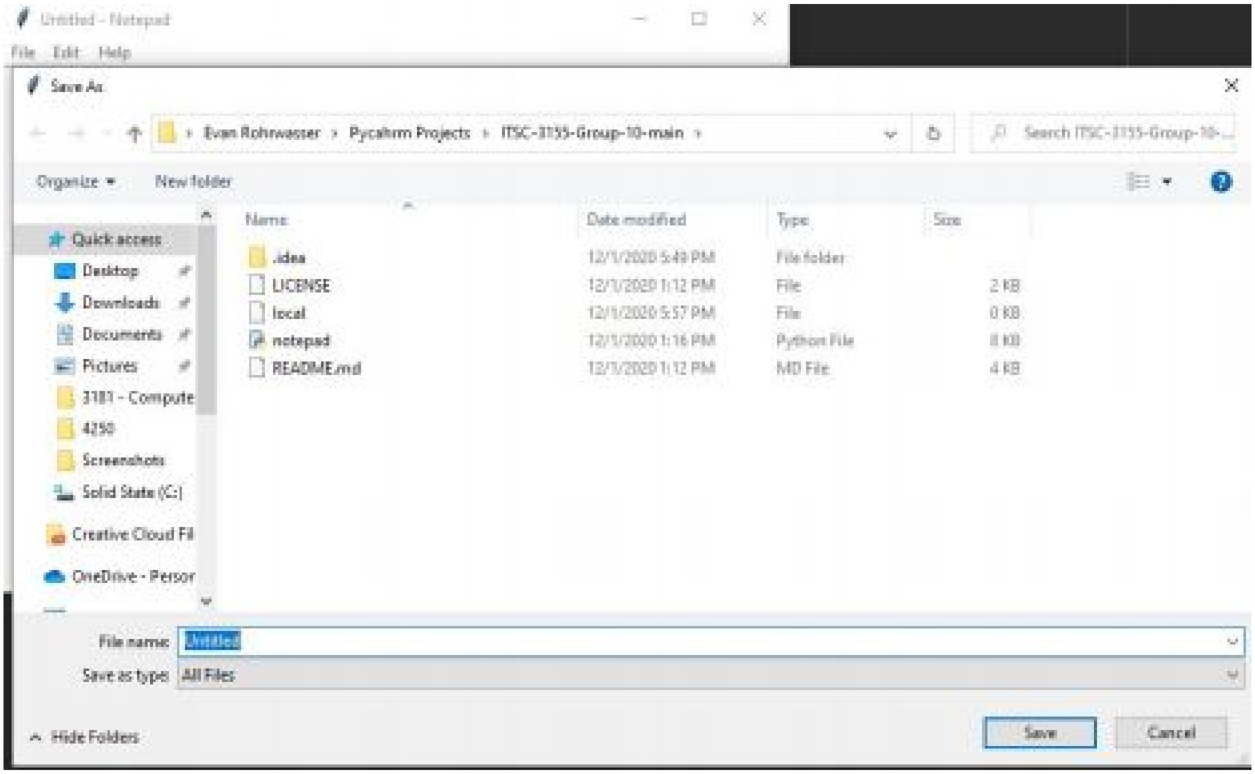


When selecting open or save a file explorer window will pop up for you to either open or save respectively. As a USB user, the file explorer will default to opening and saving notes to the 'Notes' folder that was created.



No items match your search.

If you are not a USB User you will just see the directory from which the program was run



Evan Rohrwasser › Pycahrm Projects › ITSC-3155-Group-10-main ›

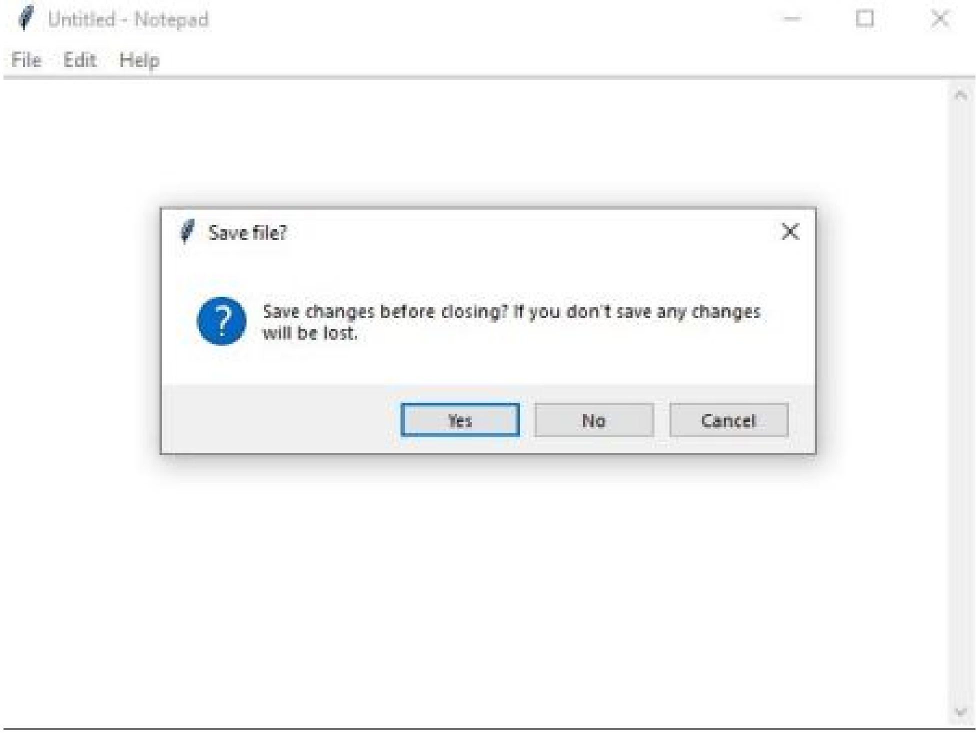
Search ITSC-3155-Group-10-...

Name	Date modified	Type	Size
.idea	12/1/2020 5:49 PM	File folder	
LICENSE	12/1/2020 1:12 PM	File	2 KB
local	12/1/2020 5:57 PM	File	0 KB
notepad	12/1/2020 1:16 PM	Python File	8 KB
README.md	12/1/2020 1:12 PM	MD File	4 KB

File name: Untitled

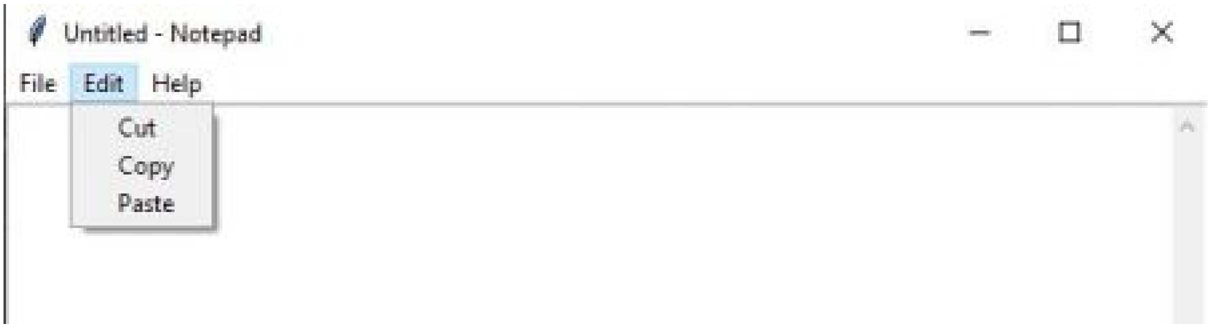
Save as type: All Files

When selecting new you will be first asked to save your current file if you haven't already. After saving, a new blank notepad will open.

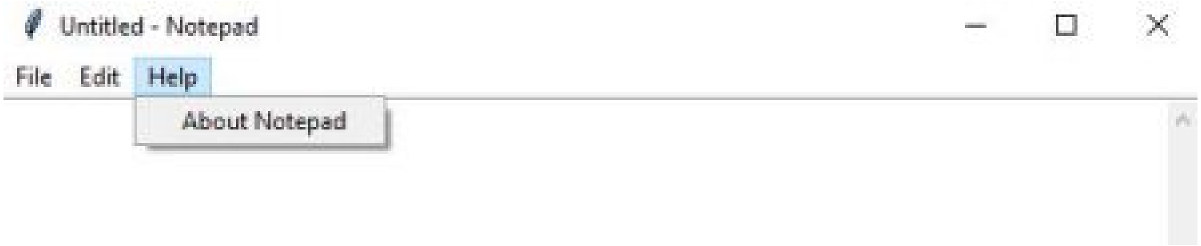


Save changes before closing? If you don't save any changes will be lost.

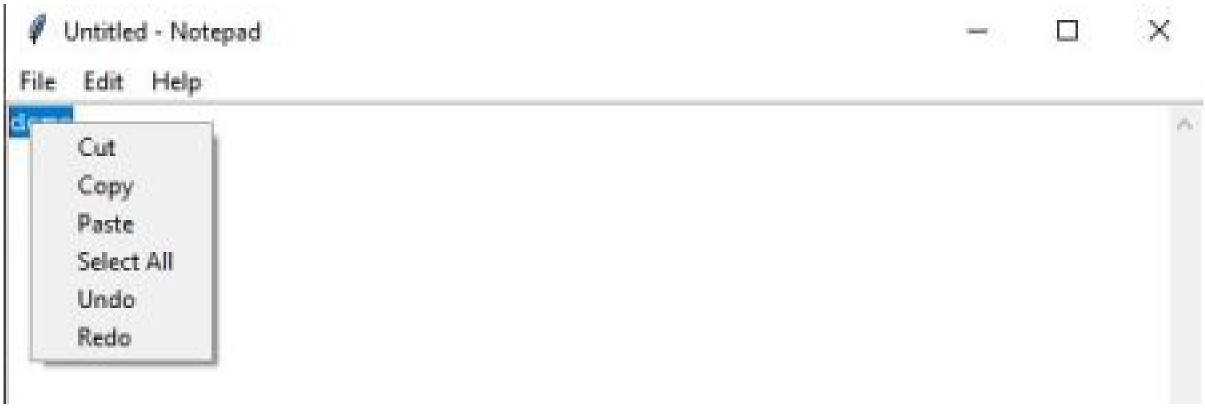
The edit tab has the options cut, copy and paste.



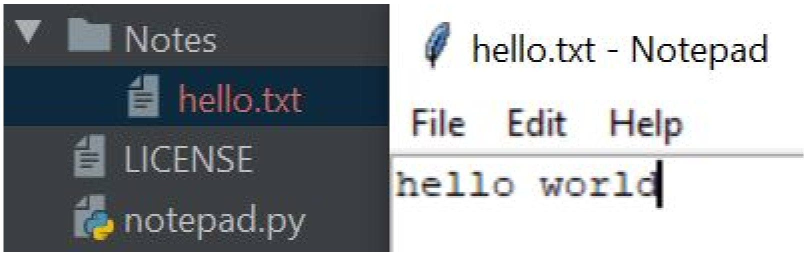
And the help tab has a link to the info about our application



Our right click menu.



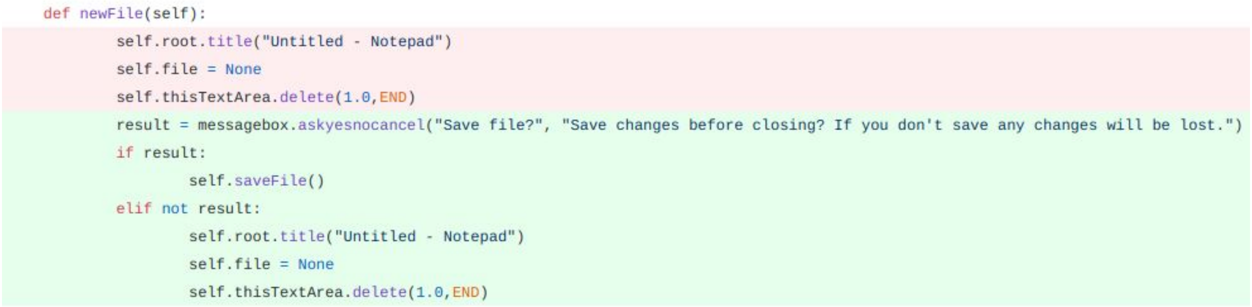
After saving as a usb user a ".txt" file it will be placed in the Notes folder



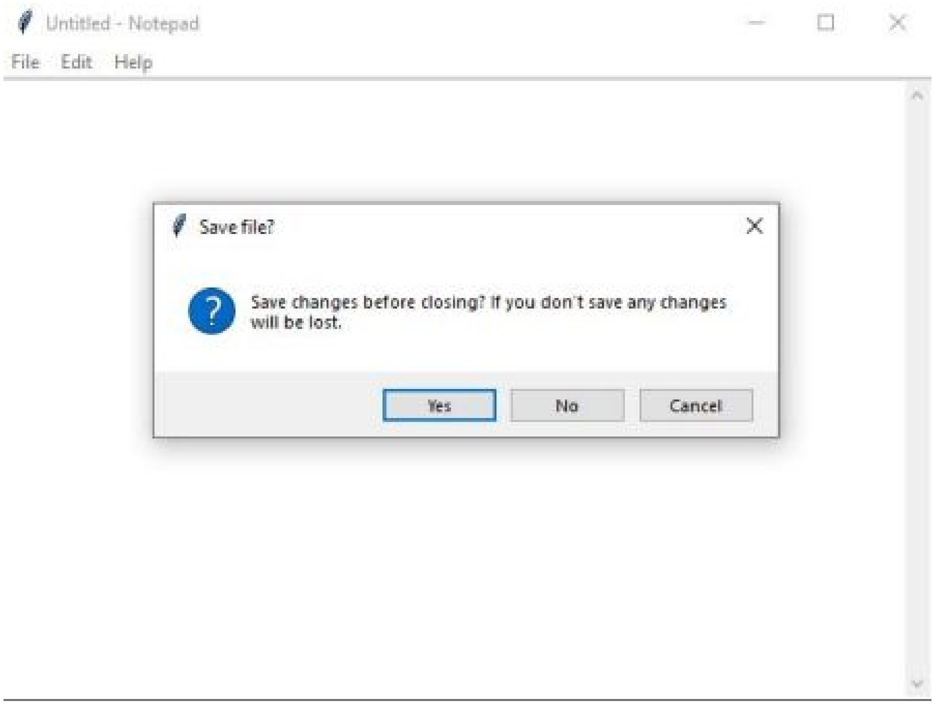
hello world

4.2 Comparison

Initially when creating a new note in our application a blank note would be created without first asking the user if they would like to save their current work. Below is the code that was added and a screenshot of the outcome in the running application.



```
def newFile(self):
        self.root.title("Untitled - Notepad")
        self.file = None
        self.thisTextArea.delete(1.0,END)
        result = messagebox.askyesnocancel("Save file?", "Save changes before closing? If you don't save any changes will be lost.")
        if result:
                self.saveFile()
        elif not result:
                self.root.title("Untitled - Notepad")
                self.file = None
                self.thisTextArea.delete(1.0,END)
```



Save changes before closing? If you don't save any changes will be lost.

5. Discussions and Conclusions

We faced various challenges throughout working on the project, everything from technical issues to problems brought about due to the current COVID-19 environment. One challenge that we faced initially was getting a grasp of where everyone in our group was at in terms of coding and in what languages. This was a crucial step in determining not only what we wanted to do but what language we were going to do it in. Another challenge we faced was having one of our teammates ghost us on the project without warning. So we had to adapt the project to be done by three people rather than the four we initially started with. Another challenge we faced was working well as a team in the at home environment we are all in, due to the covid-19 virus. It was difficult at times to communicate well and to set times we could all work on the project together. We learned many things from this project, one that is especially significant to note is teamwork and how crucial good communication is when working on a software development project. Another contribution to our learning experience was how to work with a version control system, in this case git, to manage the code development process. Making changes between the three of us and reviewing the code base was challenging and fun at the same time.

As stated in the Manifesto for Agile Software Development, “individuals and interactions over processes and tools” reflects our group’s experience during this project. Despite the challenges COVID-19 has thrown at us, we found that communication and collaboration within the group was the spearhead of the process. Whether it is our utilization of google chat, video calls, or notes within the work materials, staying connected and communicating kept us on track to finish the project in a timely manner.

A. Appendix

A.1 Appendix 1:User Stories
https://github.com/ctmalloy/ITSC-3155-Group-10/blob/main/Demo%201.pdf

A.2 Appendix 2: Use Case Diagram
https://github.com/ctmalloy/ITSC-3155-Group-10/blob/main/Use%20Case%20Diagram.pdf

A.3 Appendix 3: Context Diagram
https://github.com/ctmalloy/ITSC-3155-Group-10/blob/main/Context%20Diagram.pdf

A.4 Appendix 4: DFD
https://github.com/ctmalloy/ITSC-3155-Group-10/blob/main/DFD.pdf