

# GUI Leuphinder - Report



## Charlotte Albert

*Leuphana University Lüneburg*

Student ID: 3042257

E-mail: [Charlotte.Albert@stud.leuphana.de](mailto:Charlotte.Albert@stud.leuphana.de)

The app is the new GUI suitable for every student from Leuphana University. It showcases the prototype of an app helping students find classmates who are also looking to study for an exam, work on a project, or enjoy lunch together. The app matches students with similar time schedules, interests, or equivalent degrees to create a more connected, inclusive on-campus environment, nourishing teamwork and efficient workflows.

## Keywords

Matching, students, Digital Media, Leuphana, break or study activities, project & group work

## Table of Content

1. Introduction & GUI Idea	1
2. Methodology	
3. Concept & Design	2
4. Limitations	
4.1 Challenges	3
4.2 Solutions	
5. Future Features	
6. References	4
7. Appendix	7
8. Documentation	11

## 1. Introduction & GUI Idea

According to scientific research conducted by Wanyeki et al. in 2019, there is empirical evidence supporting the notion that teamwork can yield beneficial outcomes for our brain function and increase our overall productivity, especially in the workplace. Furthermore, an additional advantage of teamwork could be the potential for improved academic performance, as the exchange of knowledge and the act of explaining concepts to others facilitates better retention of information in our long-term memory.

This application is developed for university students and tested by the Leuphana University in Lüneburg. Every user, in this case, the student, will have their Leuphana and Email account linked to this application, making it easier to log into the interface. However, suppose you are newly registered with your article number, age, semester, major, minor, and interest. In that case, the system will create an account for you, leading to a main page where one can schedule a specific activity with other students from Leuphana, such as studying in the library together, working on a project, tutoring, or having lunch together, or going out for a chat after classes. If another student enters the same data for their break time, e.g., between classes, the GUI will create a match between the students, showing them where to meet around campus and at what time. If no student can be found at the given time, the app will suggest trying again later or choosing a different activity. Leuphinder aims to promote teamwork and connecting with fellow students with like-minded goals and interests, helping to create a more educational and comfortable environment for students. Our application is helpful for any student seeking a harmonious academic life, complete with opportunities for social engagement, forming new relationships, staying motivated, participating in practical group projects, and achieving well-planned exam preparations.

## 2. Methodology (Appendix A)

The idea behind this GUI came to my mind during the peak days of the COVID-19 pandemic when socializing on university campus seemed unmanageable and making meaningful connections with potential study buddies seemed more challenging than ever. Digital was the only way to go when seeking new connections. As a result, I asked myself why I could not match up students from the same university to work on tight project deadlines together with one click, all digitally.

Next, I considered which features should be included and which can't be missed when creating an easy-to-use GUI. I realized its complexity after creating the flowchart (Appendix B) and having all these remarkable ideas for a potentially great app. Therefore, I needed to create a bridge between the concept and a possible execution while creating a GUI with a maximum of 5 to 8 pages, which still emphasized its features and most crucial functionalities. The original design concept (Appendix C) was limited to the main critical five pages and two more as additional helpful windows. (Appendix D) The renewed concept offered a guideline to follow step by step the imagined vision behind. The GUI was created in a simplified form in the following trial and error process. At the same time, complex

features had to be iterated and adapted to the ongoing design process, especially when looking at my little programming experience at the given time. To make my process more accessible and structured, I created a working file for every window and used the main file to create the whole composition of the GUI. Lastly, I had to check for final potential errors and run it repeatedly until I was delighted and integrated all the changes. Lastly, the README file was written, and the project was uploaded to GitHub.

### **3. Concept & Design**

After logging into the account or newly registering on the app, the user reaches the main window, where one can retrieve all crucial information with only a few clicks. One can open the window to schedule appointments with other students by clicking “Find a study buddy.” This leads to the window where one can choose whether to schedule a break or to study, enter the available time, and the exact activity one wants to connect upon. If another student matches these choices, a window will pop up saying, “It is a match!” we found you a buddy; it displays their name, the email to connect on, and the place to meet for the scheduled “study, project or break” session. If no other student logged in the same data to meet a buddy, the following will show “No match found.” Please choose another activity or timeslot or try again later. When “try again” or “done” is clicked, the main page window opens again, and the user can either go to the map feature to find the meet-up point when unsure or can click “Find a buddy” to give it another try. If applicable (future feature), a profile site can be showcased or looked up after matching with a buddy. This profile includes name, email (in the future age, semester, major, minor,) and interests. Nevertheless, the GUI can only be seen as a prototype. At the same time, more explicit features are intended to be implemented if the original idea behind the whole concept is finished (Appendix D, remark: the project for now can be seen as a prototype of the final draft).

### **4. Limitations**

As mentioned in 3. The execution of the GUI has several limitations and several challenges were faced during the development process.

#### **4.1 Challenges**

1. The first difficulty when implementing the above mentioned concept was that I had to relearn old patterns and discuss structures from Tech Basics 1 without ever having programmed a GUI with the help of Tkinter.
2. Using padding correctly, it took several trials to get the title to the centre and the buttons on the login page aligned correctly.
3. When I click the start button after entering my login details, it shows a white window instead of the main one and doesn't destroy the old one, combining windows was a difficulty with many bugs.

4. The Find a Study Buddy - Button needs to be moved into the image and not above (mainwin)
5. The Schedule Window was not aligning in its arrangement, the window didn't delete after closing, how to only choose one activity without one in "study" and one in "have a break" lighting up blue.
6. Changing the font in all windows to "Aleo" only works when installing a certain plugin and importing it to the tkinker file.
7. Several bugs occurred during the merging process and the first draft being only a closed circle without Records/a Database to recall upon.

## 4.2 Solutions

1. I decided to receive guidance to overcome the fear of not seeing progress and not knowing where to start the project. I watched several "how-to" tutorials while asking other classmates for advice when unsure how to begin.
2. I initially used the pack option, yet I forgot that padding can be more accurate when using a grid or place as a placement option. (Solution: Understanding the difference between place, pack, and grid.)
3. The command needed to be changed to global fct. & connected to the right path of the library.
4. The placement needed to be changed again to make it work accordingly; see the final code.
5. Changed the layout from one radio box and one combobox to one radio box and two comboboxes.
6. A new plugin needs to be installed, however I adjusted the font in all windows uniformly.
7. Trial and Error fixing issues and creating a functional closed circle with minor inconveniences from the concept design.

## 5. Future Features

In the future, it may be possible to assess the success of Leuphana's study app. If other universities become convinced of the app's concept, it could lead to a university network. Additionally, the app prototype could be published and adapted in different cities and institutions, including schools, to facilitate group work among students or introduce the app during orientation week for incoming first-year students. Given the current digital culture, this app could be beneficial in enabling students to establish connections.

To enhance user experience, it could be feasible to incorporate features such as evaluating the overall app experience, including suggestions for improvements. The availability of study rooms could be indicated through pop-up messages or colour-coded indicators (e.g., a red or green light) next

to their names in the list, making it easier for study buddies to identify whether a particular building has vacant rooms for their study needs. Furthermore, the time slot selection and study scheduling feature could be further enhanced by sending users a pop-up message once a more suitable study buddy has been found. Alternatively, an option to "Switch buddies!" could be implemented to address a student's concerns or complaints after a study session with a particular buddy.

## 6. References

### *Report*

Wanyeki, M. N., Maina, C. W., Sanyanda, J. N., Kiiru, Dr. D. (2019). Impact of teamwork on employee performance: study of faculty members in Kenyatta University. In: Journal of Human Resource and Leadership ISSN 2519-9099, Vol. 4, Nr. 1, 1-8.

### *General Help and Basics*

Real Python. (n.d.). Python GUI Programming With Tkinter. Retrieved from <https://realpython.com/python-gui-tkinter/#building-your-first-python-gui-application-with-tkinter>

Python Software Foundation. (n.d.). Tkinter — Python interface to Tcl/Tk — Python 3.12.3 documentation. Retrieved from <https://docs.python.org/3/library/tkinter.html#module-tkinter>

Python Basics. (n.d.). Tkinter (GUI Programming) - Python Tutorial. Retrieved from <https://pythonbasics.org/tkinter/>

Stack Overflow. (n.d.). Tkinter understanding mainloop. Retrieved from <https://stackoverflow.com/questions/29158220/tkinter-understanding-mainloop>

### *Adding Images*

Python Guides. (n.d.). Python Tkinter Image + Examples. Retrieved from <https://pythonguides.com/python-tkinter-image/>

GeeksforGeeks. (n.d.). Loading Images in Tkinter using PIL. Retrieved from <https://www.geeksforgeeks.org/loading-images-in-tkinter-using-pil/>

### *Padding and Grid*

Python Examples. (n.d.). Tkinter Window Padding. Retrieved from <https://pythontexamples.org/python-tkinter-window-padding/>

Python Tutorial. (n.d.). Tkinter Grid Geometry Manager. Retrieved from <https://www.pythontutorial.net/tkinter/tkinter-grid/>

Python Guides. (n.d.). Python Tkinter Grid Padding. Retrieved from  
<https://pythonguides.com/wp-content/uploads/2021/06/python-tkinter-grid-padding.png>

Stack Overflow. (n.d.). Grid and Frames in Tkinter. Retrieved from  
<https://stackoverflow.com/questions/74874877/grid-and-frames-in-tkinter>

### **Combobox**

GeeksforGeeks. (n.d.). Combobox Widget in Tkinter (Python). Retrieved from  
<https://www.geeksforgeeks.org/combobox-widget-in-tkinter-python/?ref=lbp>

### **Radio Button**

GeeksforGeeks. (n.d.). RadioButton in Tkinter (Python). Retrieved from  
<https://www.geeksforgeeks.org/radiobutton-in-tkinter-python/>

Python Tutorial. (n.d.). Tkinter Radio Button. Retrieved from  
<https://www.pythontutorial.net/tkinter/tkinter-radio-button/>

### **Combining several Windows in one File**

Atlas. (2022, December 27). Working with multiple windows in tkinter [Video]. YouTube.<https://www.youtube.com/watch?v=unZlSLhzNOU>

How do you combine two windows in python tkinter? Hangman Project. (n.d.). Stack Overflow.<https://stackoverflow.com/questions/66844019/how-do-you-combine-two-windows-in-python-tkinter-hangman-project>

### **Finding Solutions to Issues**

OpenAI. (2023). ChatGPT (Mar 24 version).. <https://chat.openai.com/chat>

### **README File references**

Make a README. (n.d.). Retrieved from <https://www.makeareadme.com/>

matiassingers/awesome-readme. (n.d.). README.md. Retrieved from  
<https://github.com/matiassingers/awesome-readme/blob/master/readme.md>

Nyakundi, H. (2021, December 8). How to Write a Good README File for Your GitHub Project. FreeCodeCamp. Retrieved from <https://freecodecamp.org/news/how-to-write-a-good-readme-file/>

### **Code from my self-study helping me to adapt learned patterns to my little GUI project**

<https://drive.google.com/drive/folders/1feXfBbeym5NNiAhmWHlqbc17VIjPWKQC?usp=sharing>

### Images

- [https://images.adsttc.com/media/images/55e6/0d8f/2347/5df0/b800/0004/newsletter/computer\\_grafik-universitat-luneburg-2.jpg?1441140107](https://images.adsttc.com/media/images/55e6/0d8f/2347/5df0/b800/0004/newsletter/computer_grafik-universitat-luneburg-2.jpg?1441140107)
- <https://www.leuphana.de/en/university/maps.html>
- [https://pbs.twimg.com/profile\\_images/1466013048179671040/KaGxaEfT\\_400x400.jpg](https://pbs.twimg.com/profile_images/1466013048179671040/KaGxaEfT_400x400.jpg)

## 7. Appendix

[https://drive.google.com/drive/folders/1kVShiV5P\\_lmtmOyDHuzqezhE-YhaYVBc?usp=sharing](https://drive.google.com/drive/folders/1kVShiV5P_lmtmOyDHuzqezhE-YhaYVBc?usp=sharing)

### Appendix A

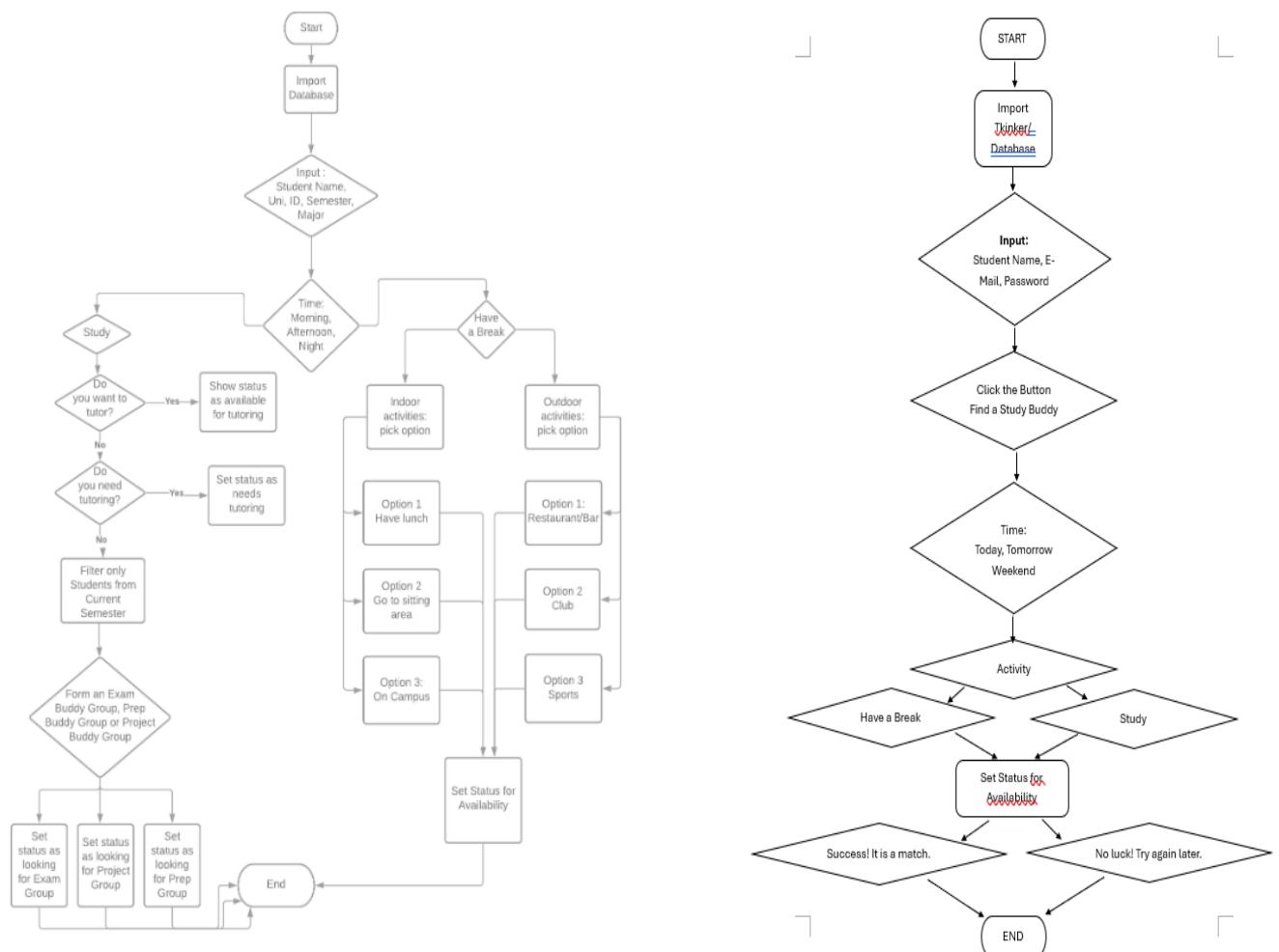
## METHODOLOGY SUMMARY

1. Idea generation
2. Design its structure via flowcharts
3. Build Design Interfaces (static)
4. Research basics of Tkinter
  - watch how-to videos on how to create potential future implemented widgets/features
  - e.g. Combo Box, Radio Boxes, picture implementation, logo as a button
5. Trial and error procedure – using the procedures from the basics introductions and my notes and adapting them to my own project and executing my designed idea
6. Simplify design choices due to limited knowledge resources
7. Reimplement and when issue is fixed – move on to the next window to create
8. After all individual windows are created, the design gets refined e.g. in design fonts or colours
9. Compose a final file where all windows are combined and integrated into one
10. Run until finalized (users tested, prototype testing UX)- save – upload to GitHub – write READ ME file and final testing

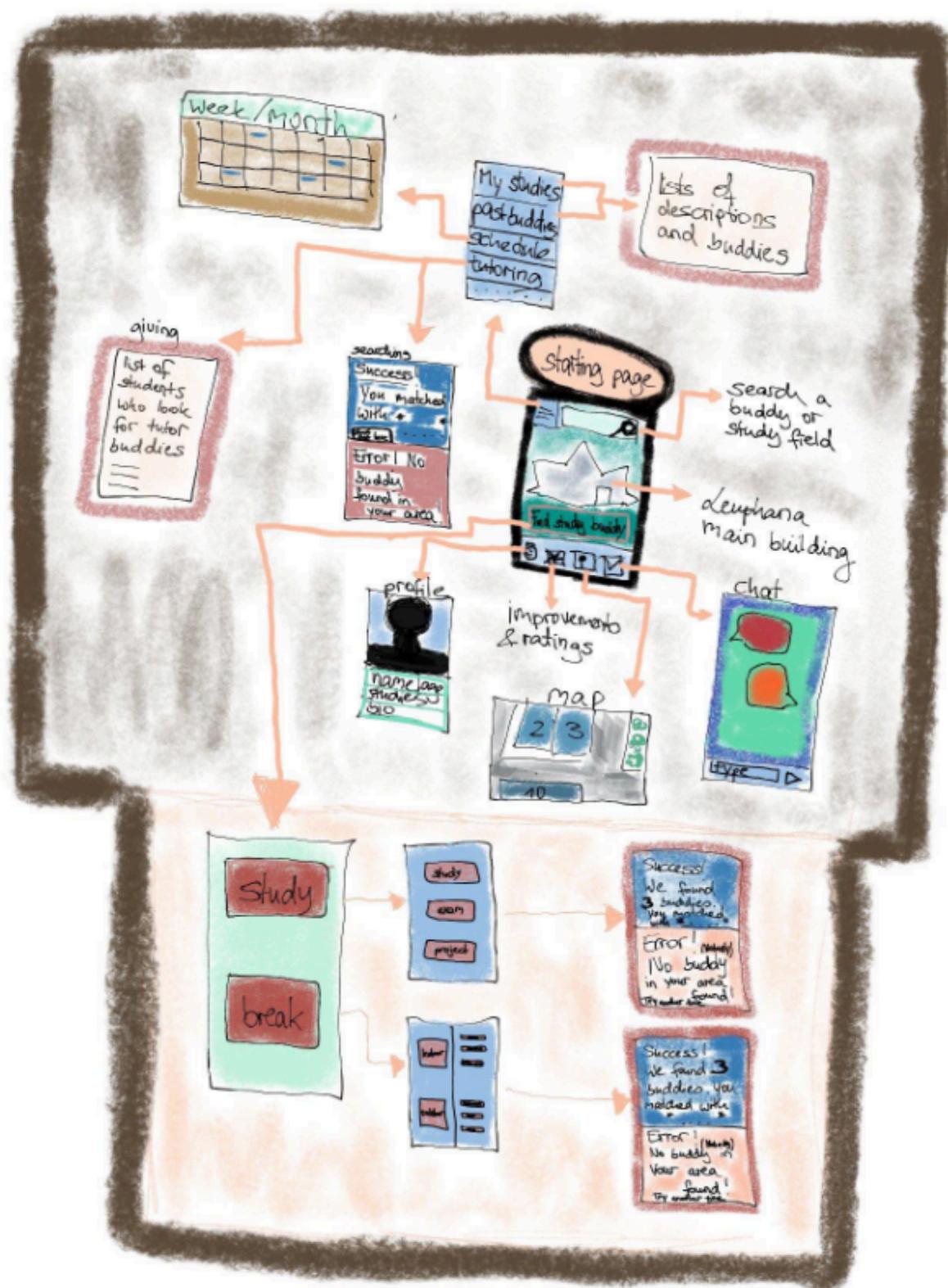
DONE

### Appendix B

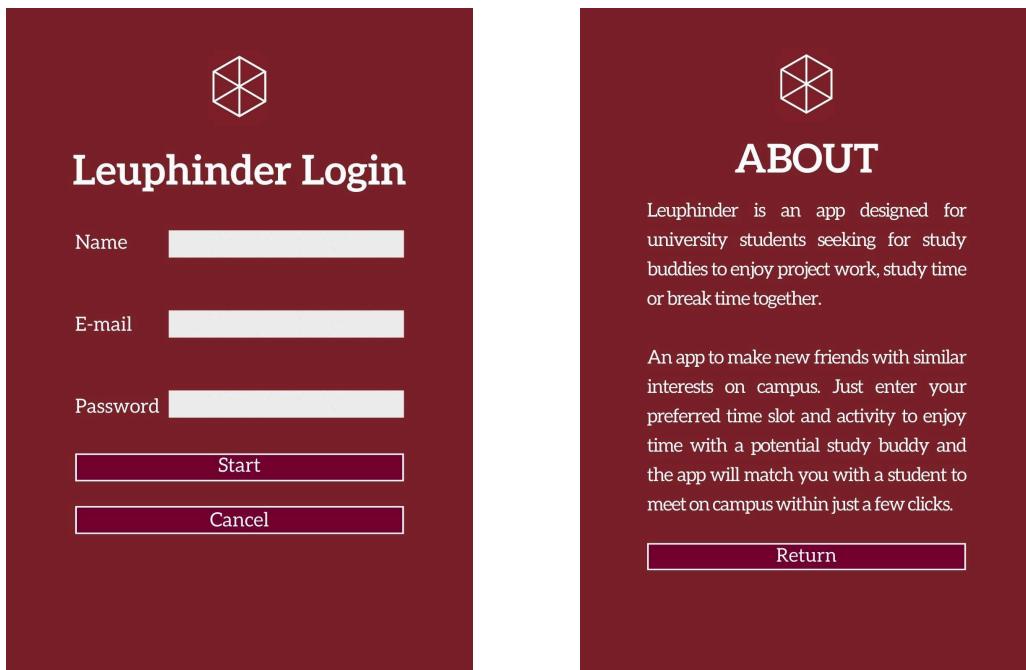
#### Original Idea - Simplified



## Appendix C



## Appendix D



used background motives in the final code:

<https://drive.google.com/drive/folders/1TgFrE2Q2FcSUocPdCuq3VnE7mMojCMsM?usp=sharing>

**Find my buddy - on the map****Return****Find a study buddy - Schedule for****TIME**

- today - 9 am to 12 pm  
 today - 12 to 3 pm  
 today - 3 pm to 6 pm  
 today - after 6 pm  
 tomorrow - 9 am to 12 pm  
 tomorrow - 12 to 3 pm  
 tomorrow - 3 pm to 6 pm  
 tomorrow - after 6 pm  
 next weekend

**ACTIVITY**

- for a project  
 for an exam  
 in the library

- go for lunch  
 go outside e.g. for a walk  
 go into the city

**Search****It's a match!**

SOMEONE WANTS TO MEET YOU



Name:

E-mail:

Location:

**Find my Study Buddy on the MAP****Cancel****No luck!**

TRY AGAIN LATER

**Try again****Cancel**

## 8. Documentation

(working file code, see challenges e.g. 3.; 4.)

```

File Edit Selection View Go Run Terminal Help ⏎ → ⏎ Search File Edit Selection View Go Run Terminal Help ⏎ → ⏎ Search
6-img & combobox(1).ipynb • experimenting file.py x Leuphinder_login.py Leuphinder_Mainpage.py
C:\Users\lotti>Documents>Leuphana assignments 2023>Tech Basics 2> experimenting file.py () tk
22
23 win = tk.Tk()
24 win.title("Leuphinder")
25 win.geometry("720x240")
26 win.configure(bg="#7d212a")
27 win.resizable(0, 0)
28
29 path = r'C:\Users\lotti\Downloads\Beige Good Morning Instagram Story.jpg'
30
31 title_label = tk.Label(text="Leuphinder", fg="black", font=("Arial", 30), bg="#7d212a")
32 title_label.grid(row=0, column=0, columnspan=3, sticky="ew", padx=25)
33
34 email_label = tk.Label(text="E-mail", fg="black", bg="white")
35 email_label.grid(row=1, column=0, padx=20, pady=10, sticky="nw")
36
37 email_entry = tk.Entry(fg="black", bg="white")
38 email_entry.grid(row=1, column=1, columnspan=2, padx=(5, 20), pady=10, sticky="nw")
39
40 password_label = tk.Label(text="Password", fg="black", bg="#7d212a")
41 password_label.grid(row=2, column=0, padx=20, pady=10, sticky="nw")
42
43 password_entry = tk.Entry(fg="black", bg="white")
44 password_entry.grid(row=2, column=1, columnspan=2, padx=(5, 20), pady=10, sticky="nw")
45
46 forgot_password_button = tk.Button(text="Forgot?", bg="gray", command=destroy_win)
47 forgot_password_button.grid(row=4, column=0, columnspan=3, padx=10, pady=5, sticky="nsew")
48
49 start_button = tk.Button(text="Start", bg="gray", command=create_mainpage_window)
50 start_button.grid(row=3, column=0, columnspan=3, padx=10, pady=5, sticky="nsew")
51
52 win.mainloop()

```

```

File Edit Selection View Go Run Terminal Help ⏎ → ⏎ Search File Edit Selection View Go Run Terminal Help ⏎ → ⏎ Search
6-img & combobox(1).ipynb • experimenting file.py x Leuphinder_login.py Leuphinder_Mainpage.py
C:\Users\lotti>Documents>Leuphana assignments 2023>Tech Basics 2> experimenting file.py () tk
1 import tkinter as tk
2 from PIL import Image, ImageTk
3
4 def destroy_win():
5     win.destroy()
6
7 def create_mainpage_window():
8     mainpage_win = tk.Toplevel()
9     mainpage_win.title("Main Page")
10    mainpage_win.geometry("400x300")
11    mainpage_win.resizable(0, 1)
12
13    python_image = Image.open(path)
14    resized_image = python_image.resize((400, 700), Image.LANCZOS)
15    tkinter_image = ImageTk.PhotoImage(resized_image)
16
17    label = tk.Label(mainpage_win, image=tkinter_image)
18    label.pack()
19
20    def open_windowmainpage():
21        win.destroy()
22
23    win = tk.Tk()
24    win.title("Leuphinder")
25    win.geometry("720x240")
26    win.configure(bg="#7d212a")
27    win.resizable(0, 0)
28
29    path = r'C:\Users\lotti\Downloads\Beige Good Morning Instagram Story.jpg'
30
31    title_label = tk.Label(text="Leuphinder", fg="black", font=("Arial", 30), bg="#7d212a")
32    title_label.grid(row=0, column=0, columnspan=3, sticky="ew", padx=25)
33
34    email_label = tk.Label(text="E-mail", fg="black", bg="white")
35    email_label.grid(row=1, column=0, padx=20, pady=10, sticky="nw")
36
37    email_entry = tk.Entry(fg="black", bg="white")
38    email_entry.grid(row=1, column=1, columnspan=2, padx=(5, 20), pady=10, sticky="nw")
39
40    password_label = tk.Label(text="Password", fg="black", bg="#7d212a")
41    password_label.grid(row=2, column=0, padx=20, pady=10, sticky="nw")
42
43    password_entry = tk.Entry(fg="black", bg="white")
44    password_entry.grid(row=2, column=1, columnspan=2, padx=(5, 20), pady=10, sticky="nw")
45
46    forgot_password_button = tk.Button(text="Forgot?", bg="gray", command=destroy_win)
47    forgot_password_button.grid(row=4, column=0, columnspan=3, padx=10, pady=5, sticky="nsew")
48
49    start_button = tk.Button(text="Start", bg="gray", command=create_mainpage_window)
50    start_button.grid(row=3, column=0, columnspan=3, padx=10, pady=5, sticky="nsew")
51
52    win.mainloop()

```

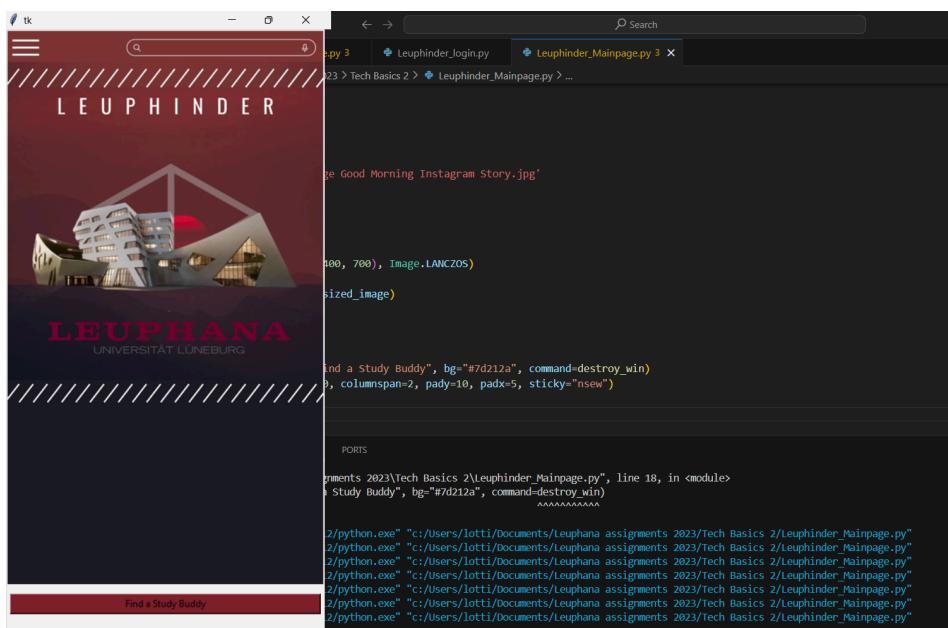
```

File Edit Selection View Go Run Terminal Help ⏎ → ⏎ Search File Edit Selection View Go Run Terminal Help ⏎ → ⏎ Search
6-img & combobox(1).ipynb • experimenting file.py x Leuphinder_login.py Leuphinder_Mainpage.py code(1).py
C:\Users\lotti>Documents>Leuphana assignments 2023>Tech Basics 2> experimenting file.py () tk
1 import tkinter as tk
2 from PIL import Image, ImageTk
3
4 def destroy_win():
5     win.destroy()
6
7 def create_mainpage_window():
8     mainpage_win = tk.Toplevel()
9     mainpage_win.title("Main Page")
10    mainpage_win.geometry("400x300")
11    mainpage_win.resizable(0, 1)
12
13    python_image = Image.open(path)
14    resized_image = python_image.resize((400, 700), Image.LANCZOS)
15    tkinter_image = ImageTk.PhotoImage(resized_image)
16
17    label = tk.Label(mainpage_win, image=tkinter_image)
18    label.pack()
19
20    def open_windowmainpage():
21        win.destroy()
22
23    win = tk.Tk()
24    win.title("Leuphinder")
25    win.geometry("720x240")
26    win.configure(bg="#7d212a")
27    win.resizable(0, 0)
28
29    path = r'C:\Users\lotti\Downloads\Beige Good Morning Instagram Story.jpg'
30
31    title_label = tk.Label(text="Leuphinder", fg="black", font=("Arial", 30), bg="#7d212a")
32    title_label.grid(row=0, column=0, columnspan=3, sticky="ew", padx=25)
33
34    email_label = tk.Label(text="E-mail", fg="black", bg="white")
35    email_label.grid(row=1, column=0, padx=20, pady=10, sticky="nw")
36
37    email_entry = tk.Entry(fg="black", bg="white")
38    email_entry.grid(row=1, column=1, columnspan=2, padx=(5, 20), pady=10, sticky="nw")
39
40    password_label = tk.Label(text="Password", fg="black", bg="#7d212a")
41    password_label.grid(row=2, column=0, padx=20, pady=10, sticky="nw")
42
43    password_entry = tk.Entry(fg="black", bg="white")
44    password_entry.grid(row=2, column=1, columnspan=2, padx=(5, 20), pady=10, sticky="nw")
45
46    forgot_password_button = tk.Button(text="Forgot?", bg="gray", command=destroy_win)
47    forgot_password_button.grid(row=4, column=0, columnspan=3, padx=10, pady=5, sticky="nsew")
48
49    start_button = tk.Button(text="Start", bg="gray", command=create_mainpage_window)
50    start_button.grid(row=3, column=0, columnspan=3, padx=10, pady=5, sticky="nsew")
51
52    win.mainloop()

```

The screenshot shows a Jupyter Notebook interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** Includes icons for file operations, search, and help.
- Code Cell:** Contains Python code for creating a Tkinter application. The code imports tkinter, PIL, and Image, defines a destroy\_win function, creates a window, resizes an image, creates a PhotoImage, and sets up a label and button.
- Output Cell:** Shows a Traceback for a NameError and the full stack trace.
- Bottom Navigation:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS tabs.
- Bottom Status Bar:** Includes icons for file operations, search, and help, along with the text "In [23], Col 15".



## challenge no. 5

The screenshot shows a Windows desktop environment with a Python application running in the foreground. The application window title is "Combobox Example" and the main title bar says "Find a study buddy - Schedule for". The window contains sections for "ACTIVITY" and "TIME", with a dropdown menu open under "TIME". Below the window, a code editor displays Python code for creating a GUI with a combobox. The code uses Tkinter and ttk modules. A "WATCH" panel shows variables like `label1` and `study\_box`. A "CALL STACK" panel lists several Python debugger sessions. The bottom of the screen features a taskbar with various icons and a system tray showing the date and time.

```
File Edit Selection View Go Run Terminal Help < > ⌘ search
Combobox Example - x x schedule page.py 8:49 Python Debugger: Current ...
Find a study buddy - Schedule for

ACTIVITY
TIME
for a project
go into the code

WATCH
label1 = tk.Label(text="Find a study buddy - Schedule for", bg="#d2d2a", fg="white", font=("Times New Roman", 15)).grid(row=0, column=0)
# Label for Combobox:
labelstudy = tk.Label(text="TIME", font=("Times New Roman", 15)).grid(row=5, column=0, padx=10, pady=25)
n = tk.StringVar() # I have created 'n', a special variable to interact with strings in tkinter app.
study_box = ttk.combobox(textvariable=n) # creating the combobox
# Using ['values'] in combobox to set options for user to choose from:
study_box['values'] = ('today - 9am to 12pm', 'today - 12pm to 3pm', 'today - 3pm to 6pm', 'today - after 6pm', 'tomorrow - 9am to 12pm', 'tomorrow - 12pm to 3pm', 'tomorrow - 3pm to 6pm', 'tomorrow - after 6pm')

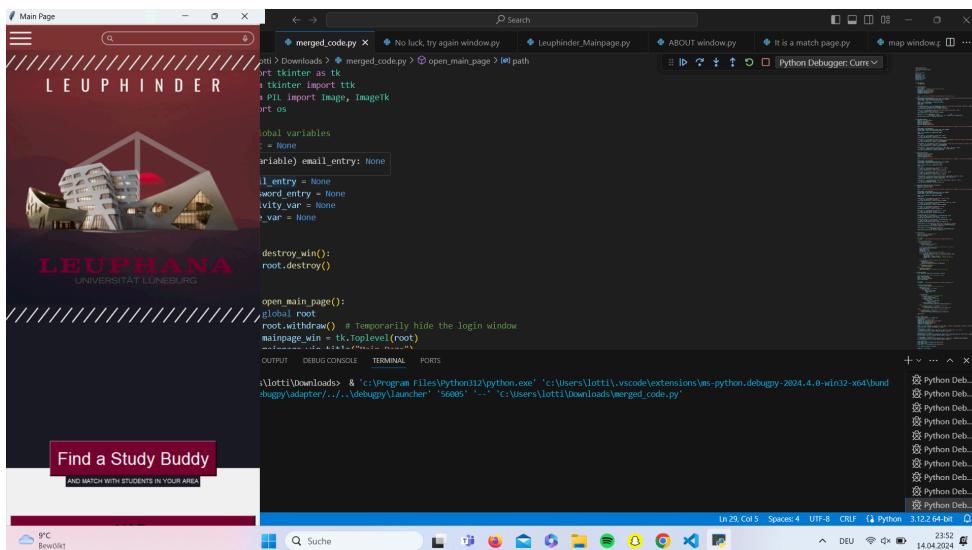
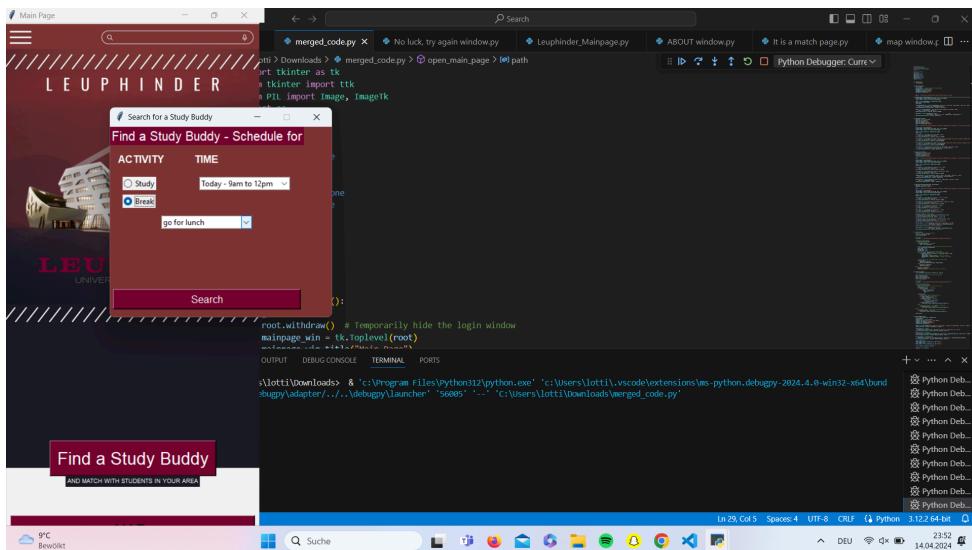
CALL STACK
Python Debugger: Cu... RUNNING
Python Debugger: Cu... RUNNING
Python Debugger: Cu... RUNNING
Launch RUNNING

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER
buggy.launcher' '5396' '' 'c:\users\lotti\documents\leuphana assignments 2023\Tech Basics 2\final app version 1.py'
PS C:\Users\lotti\Documents\Leuphana assignments 2023\Tech Basics 2> cd 'c:\users\lotti\documents\leuphana assignments 2023\Tech Basics 2' & 'c:\program files\python312\python.exe' 'c:\users\lotti\vscodeextensions\lsp-python.debugger-2024.4.0-win32-x64\bundled\libs\debug\adapter\..\..\de
bugger\launcher' '5395' '' 'c:\users\lotti\documents\leuphana assignments 2023\Tech Basics 2\schedule window final.py'
PS C:\Users\lotti\Documents\Leuphana assignments 2023\Tech Basics 2> cd 'c:\users\lotti\documents\leuphana assignments 2023\Tech Basics 2' & 'c:\program files\python312\python.exe' 'c:\users\lotti\vscodeextensions\lsp-python.debugger-2024.4.0-win32-x64\bundled\libs\debug\adapter\..\..\de
bugger\launcher' '5396' '' 'c:\users\lotti\documents\leuphana assignments 2023\Tech Basics 2\schedule page.py'

Breakpoints
Raised Exceptions
Uncaught Exceptions
User Uncaught Exceptions
Live Share
20°C Meist sonnig
In 52 Col 15 Spaces: 4 UIE-8 CRF Python 3.12.2 64-bit DEU 10.04.2024 1945
```

The screenshot shows a Python development environment with the following details:

- File Explorer:** Shows a project structure with files like `final app version 1.py`, `schedule window final.py`, and `schedule page.py`.
- Run and Debug:** A dropdown menu is open.
- VARIABLES:** A sidebar showing variables for a study buddy.
- WATCH:** A sidebar showing a break point for a scheduled task.
- CALL STACK:** A sidebar showing multiple Python debugger entries.
- BREAKPOINTS:** A sidebar showing breakpoints for raised exceptions.
- Code Editor:** The file `schedule window final.py` is open, containing code to create a window for finding a study buddy.
- Scheduled Task Window:** A modal titled "Find a Study Buddy - Schedule for" is displayed, listing activities and times.
- Search Bar:** A search bar at the bottom left with the placeholder "Search for a Study Buddy".



*other documentations of challenges mentioned in 4.1 can be found here:*

<https://drive.google.com/drive/folders/17hCDV69veW9LqsDvD4YegXkOvpXVMb0T?usp=sharing>

- first trial and error to create an image within a window and have the

“Find a Study Buddy” - Button in it (Layout-Error):

[https://drive.google.com/file/d/1PbcRnUos6mudfpzWWjBvc2\\_LhIPFpSe2/view?usp=sharing](https://drive.google.com/file/d/1PbcRnUos6mudfpzWWjBvc2_LhIPFpSe2/view?usp=sharing)

<https://drive.google.com/file/d/1WYENGMeRPvyqtzaEVU9Wi-4wyIpjEzxx/view?usp=sharing>

- first trial attempting to build the schedule page:

<https://drive.google.com/file/d/1qcpnEreIpveNumM8VwBrE4RObgSvGavR/view?usp=sharing>

- trial and error in merging the windows together:

<https://drive.google.com/file/d/19IgWPGwv7kb8TTU1jlfwAQkJQVBC8L/view?usp=sharing>

<https://drive.google.com/file/d/1uSxyvnxO4z7oOFGDivUZHIuCBEZodDva/view?usp=sharing>

**Important files:*****README:***

<https://docs.google.com/document/d/1S5oKr3O9re66fD9rAHAbQg63qgqHPU8N/edit?usp=sharing&oid=101482898738851679471&rtpof=true&sd=true>

***merged FINAL code:***

[https://drive.google.com/drive/folders/1xJM32k0LjNNeu4\\_BTYq32nnJFqf3q5B?usp=sharing](https://drive.google.com/drive/folders/1xJM32k0LjNNeu4_BTYq32nnJFqf3q5B?usp=sharing)

***Code for every window separately:***

[https://drive.google.com/drive/folders/1K\\_FyRh5dcahrfwg9y7C65mV6gzLM96ul?usp=sharing](https://drive.google.com/drive/folders/1K_FyRh5dcahrfwg9y7C65mV6gzLM96ul?usp=sharing)