Python Project

Written and Managed by Bahadir Kucuk
October 2022

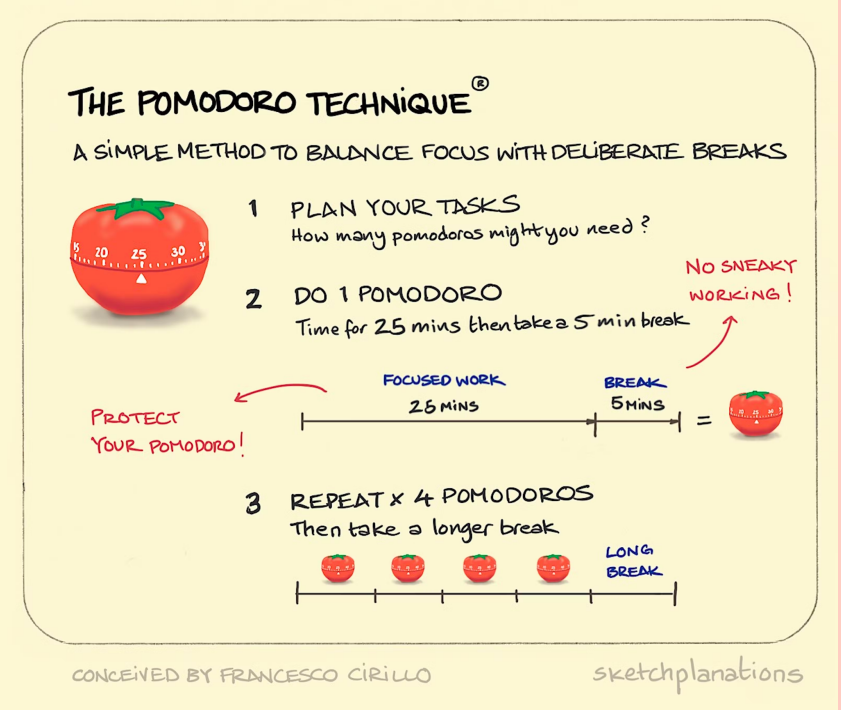
FENYX IT ACADEMY

Time Tracking App

Using Pomodoro Technique

Overview

The project is based on keeping time of any project with its subject by using the Pomodoro technique which was developed in the late 1980s by then-university student Francesco Cirillo. This technique helps the ones who are struggling to focus on their studies and complete assignments. With this project, the user will be able to track the time by using the Pomodoro technique and will be able to see the overview of the time spent on a specific subject under a project(topic).



A User Case for the Program

To illustrate, a student wants to study the “Equations” subject under the “Maths” project(topic). Firstly, the student will enter the project name as “Maths” and enter the subject in the program then he will be able to start the Pomodoro sessions. Before each session, the program will ask for a “To Do List” for that specific sub-session of Pomodoro, and also at the end of the session the program will ask again for which of these to-do lists are managed in that session.

After sessions, the student can start a new Pomodoro session after a long break. At the end of the Pomodoro sessions, the summary of the time tracked will be automatically emailed to recipients (users/parents/friends). Users can also manually send a summary email to recipients manually from the main menu.

GUI Details

1. **Login/Sign-Up Screen**
    a. Simple user login system
        i. User‘s email
    b. Simple user sign-up system
        i. User’s name
        ii. User’s email
2. **Main Menu Screen**
    a. User’s Workspace
        i. Add/Delete recipients’ emails
        ii. Delete a project
        iii. Delete a subject
    b. Start Pomodoro
        i. Add a new project or subject
        ii. Select the project and subject for the new Pomodoro to start
    c. Tracking History
        i. Select a project or all projects
        ii. Select a subject or all subjects
        iii. Select a period
            1. Today
            2. This week
            3. This Month
        iv. Total Time tracked
        v. Feature of sending an email of the tracked history (project, subject, and period) to the recipients.
        vi. An overview table of tracked history
3. **Pomodoro Screen**
    a. Main Menu Button
    b. Showing the current number of Pomodoro, for example, 1,2,3, or 4
    c. Add Task(s) for the current Pomodoro session (25 min.)
    d. Counting down timer
    e. Start/Pause time
    f. I am done with this session button (go to a short break or long break)
    g. Asking for the accomplishment of the tasks

4. Short Break Screen
    a. Main Menu Button
    b. Counting down timer (5 min.)
    c. Start/Pause time
    d. Skip button (go to next Pomodoro session)
5. Long Break Screen
    a. Main Menu Button
    b. Counting down timer
    c. Start/Pause time
    d. Skip button (go to the main menu)

General Info

- Each team will have 1 Mentor.
- Trello Board will be used in the project.
    - This is mandatory.
    - Mentor will check whether the team uses Trello actively.
    - Team mentors will also be added to the board.
- There will be at least 30-minute meetings with teammates every day.
- The content of the daily meeting is generally as follows:
    - What each teammate has done
    - The general direction of the project
    - Task sharing until tomorrow
- In Trello, there will be a list in the name of that specific date. Details, tasks, and things to be done at the meeting will be kept in Trello daily in that list.
- It is crucial to follow the steps in order to complete the project.
- A meeting will be held with 1 mentor of the group and group members(students) on the specified dates (once in 2-3 days). The dates are written below. The time of the meetings will be determined by each group mentor and group members.
- The part at each step will be completed until that meeting and the new step will be started.
- Each student in the team will present the part he/she did in that step to the mentor in the meeting.
- After each meeting, the mentors will make an assessment of whether each team member gets progress or not.
- After the project is completed, a project presentation will be made in a location or online where Pycoder's instructors will be present.

Definition of Done

- Full attendance at meetings with mentor
- UML Diagrams (Class, Usecase, Activity)
- A Program runs appropriately.
- Presentation of the project
- GitHub repository with a ReadMe.
- Medium article (optional)

Program Stack

- VSCode
- JSON for holding data
- UML Diagram(Only User-case and Class diagram, Activity diagrams)
- GitHub
- Object Oriented Programming
- PyQt5 for GUI (Default template will be provided for GUI)

Bonus

There are some “Bonus” parts under the subsections of the steps. If you have enough time, and eagerness to show your skills and make a more featured program, then try to implement them.

Instruction Steps

You have to stick to the schedule. You will have a progress meeting with your mentor once in 2-3 days (specified in the meeting schedule section). You have to complete related steps before the following meeting.

Details of Steps

Step 1 - Introduction

Duration: 0.5-day

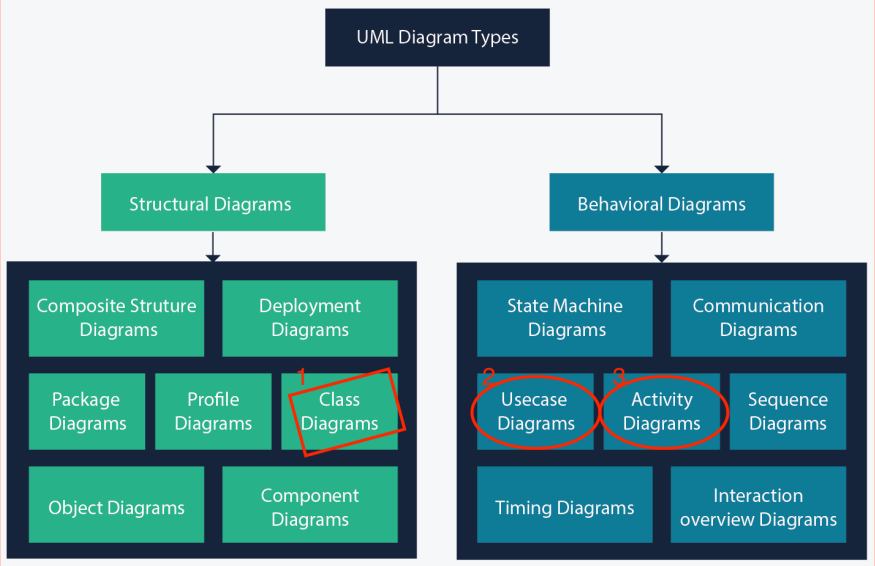
1. Reading instructions.
2. Understanding and discussing with your teammates the requirements of the project.

Step 2 - UML Design

Duration: 1.5-2 days

UML Design (Making Plan)

Note: These diagrams will be in your presentation at the end. Please pay strict attention to this part of the project.



[1]

You can use this tool to draw UML Diagrams → https://app.diagrams.net/

1. Class Diagram
    a. Watch this tutorial → https://youtu.be/UI6lqHOVHic
    b. As a group, draw a Class Diagram for your project.
2. Usecase Diagram **(Bonus)**
    a. Watch this tutorial → https://youtu.be/zid-MVo7M-E
    b. As a group, draw a Use Case Diagram for your project.
3. Activity Diagram
    a. Watch this tutorial → https://youtu.be/Wf_xlagfHmg
    b. As a group, draw an Activity Diagram for your project.
    c. There should be an activity diagram for each flow of logic.
        i. Sign Up/Login Logic
        ii. Tasks (To-do-list) and Pomodoro sessions Logic
        iii. Program Start/Quit Logic

[1] source:https://creately.com/blog/diagrams/uml-diagram-types-examples/

Note: Present your diagrams to your mentor! You will also present them in the presentation at the end. After presenting your diagrams to your Mentor, You will be able to start programming part by cloning your default GUI.

Step 3 - Project Setup

Duration: 0.5-day

Cloning Project

Clone the project via this link:
https://github.com/kucukbahadir/TimeTrackingProject.git

Project setup

- Create virtual environment
  - https://docs.python.org/3/library/venv.html
- Install the libraries in requirements.txt
  - pip install -r /path/to/requirements.txt
- Run main.py
  - A new window will pop up (Login UI)

Understanding PyQt5

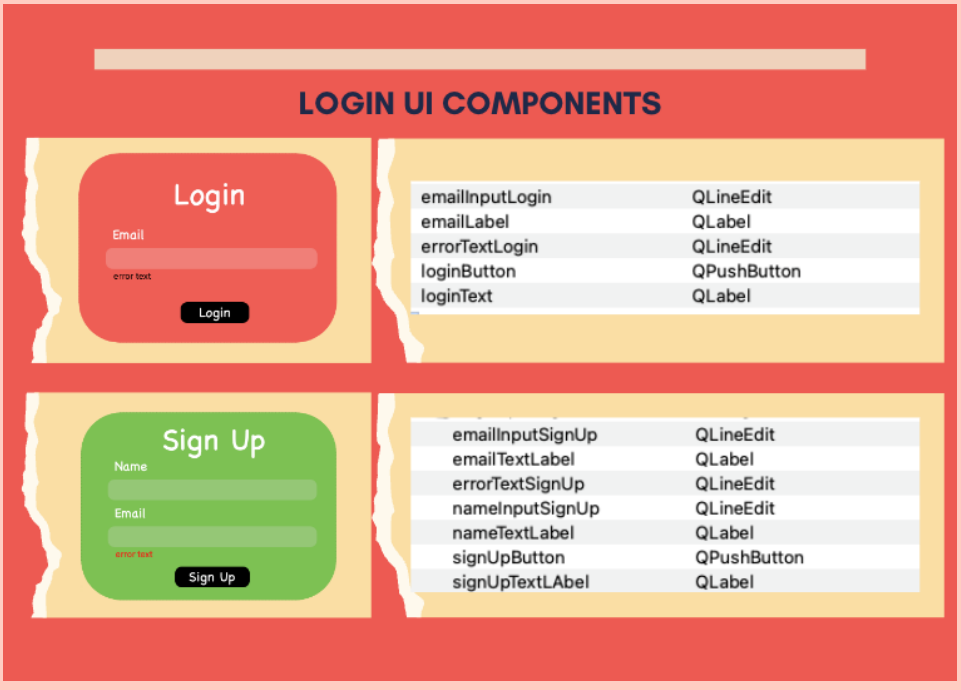
All UI files and their python code are provided. You just need to understand how to set a text in QLabel, how to get input with QLineEdit, and how to listen to the event of QPushButton.

To understand how to use these components and start to implement the login screen, you can follow this tutorial: https://youtu.be/RL9nGmv3uSU

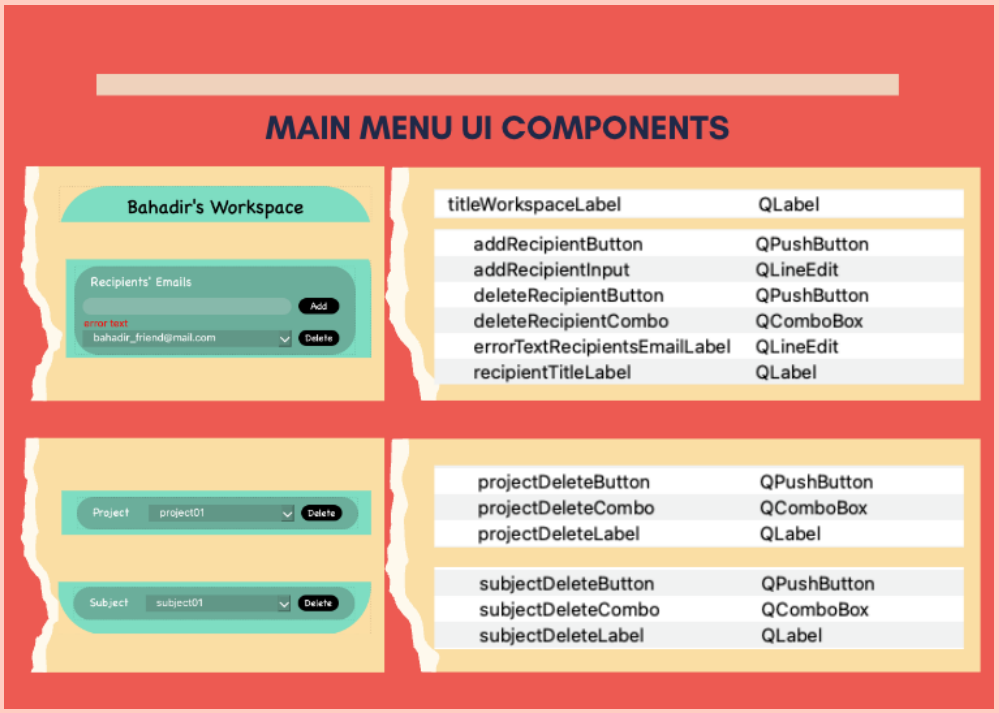
UI Files Components

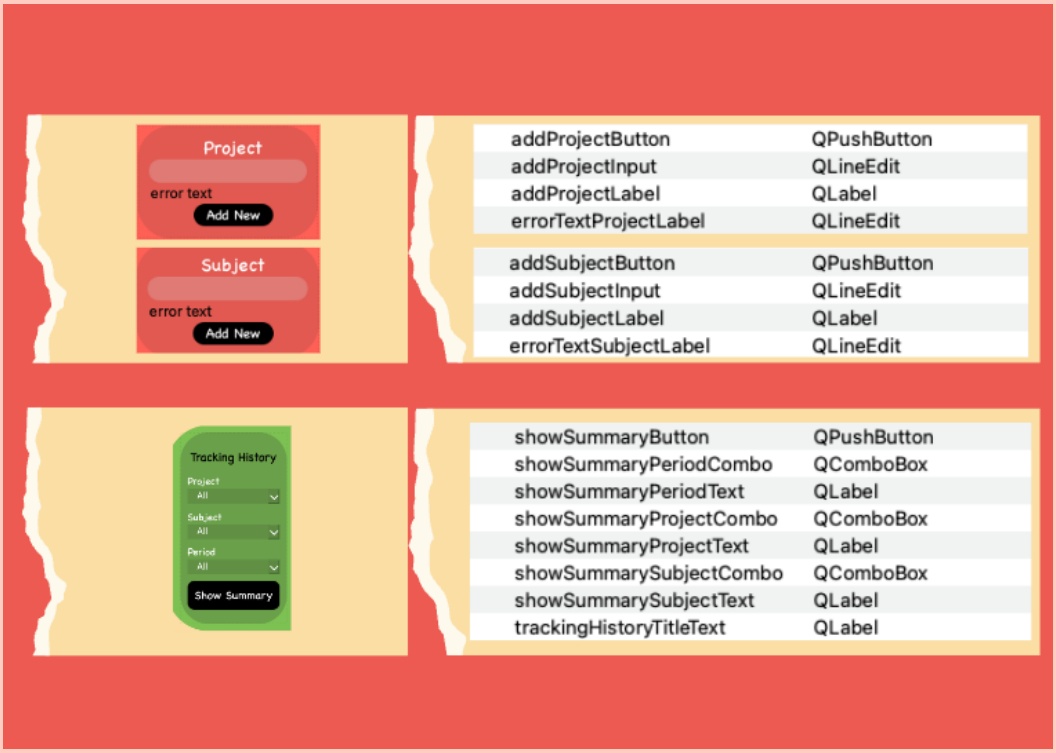
In this section, all UI components are matched line by line with its related screenshot. When you need to change or get any input on UI, you can have a look here.

Login UI Components

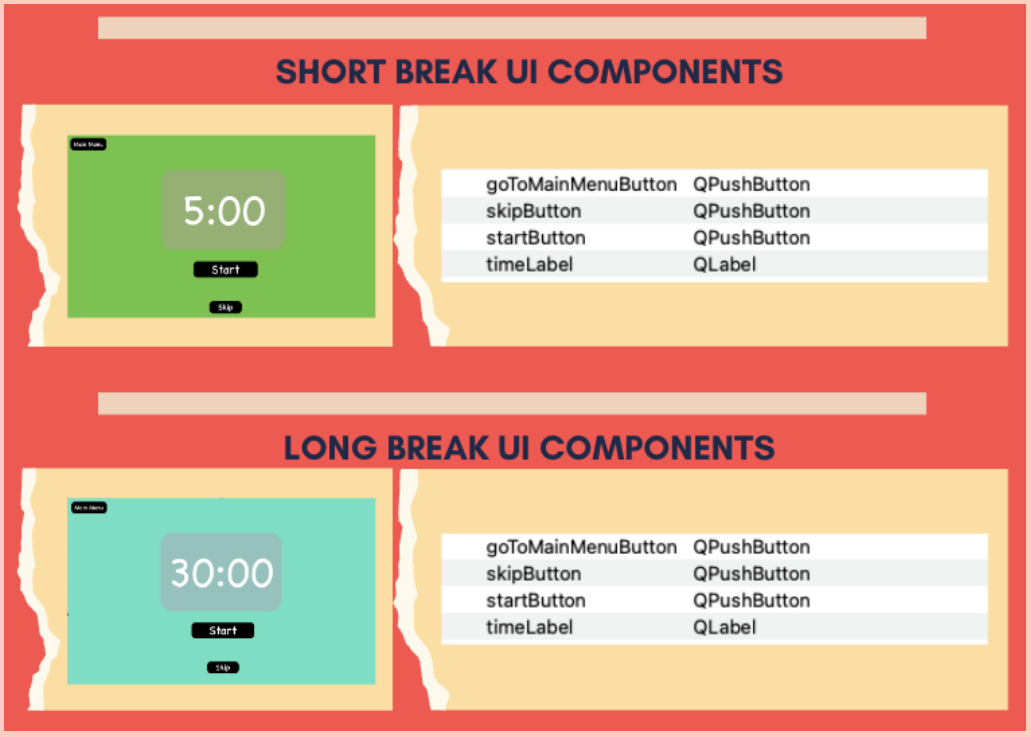


Main Menu UI Components





Breaks UI Components



Step 4

Duration: 0.5-day

Design a JSON File like a Database

User and their related info:

- Login credentials
    - Email (mandatory)
- Projects and their related total tracked time
    - Subjects and their specific tracked time
        - For each Pomodoro session

        - Start/end time
        - Date
        - Tasks (To-do-list)
            - Status (success/fail)
- Email recipients (default should be user's email at least)

Note: You have to have a default JSON File in your repository and Readme in order to show as an example.

Reading/Writing on a JSON File

- When the program is opened, you need to read the JSON file in order to have the tracking history and all information of users.
- Before a specific user quits the program, all related information has to be already written to the JSON file.

Note: Build a "Utils" module in order to write your helper functions like this. You can inherit this utils module in some classes.

Step 5

Duration: 0.5-day

Implement User Module

While the program is being used, all information related to the user has to be an attribute of this user object. All information that is held in JSON for a user should be in the user object.

Step 6

Duration: 0.5-day

Implement User SignUp/Login Activity Diagram

- Sign Up
    - A new user should be able to sign up with just a username and an email address.
    - No need for a password basically.
    - You have to check whether the username is already taken.
- Login
    - A user should be able to enter his/her username and log in.
- Controls
    - Check previous users
        - The user has to sign up with a unique username.
    - Check whether there is a username that is entered while logging in.

Note: No need for email verification. Think of it as a simple signup/login system

Bonus: There can be a sign-up/login system with a password. You should be able to reset the password, thus you will need a related user interface, too.

Note: The order and duration of the steps after this point is not important. The rest of the steps are possible ones.

Step 7

Duration: 0.5-day

Implementation "Topic" and "Subject" classes

Step 8

Duration: 0.5-day
Implementation of Break classes
Tip: You create an abstract class and inherit it for both short and long break.

Step 9

Duration: 0.5-day
Implementation of “Pomodoro Session”
Tip: It should include tasks and short breaks.

Step 10

Duration: 0.5-day
Implementation of “Pomodoro”
Tip: A Pomodoro class should include all Pomodoro sessions and a long break.

Step 11

Duration: 1 day
Build and control the logic among classes and related User Interface(s)

Step 12

Duration: 1-2 day
Implementation of tracking history

Step 13

Duration: 0.5 day
Implementation of sending email

You can send a text in that email that gives the information of pomodoro sessions (project, subject, pomodoro sessions and related tasks(success/failure). It would be nice to see the total tracked time of a summary which is sent via the button in the main menu.
You can follow this tutorial for sending emails: https://youtu.be/g_j6ILT-X0k

Tip: It is important to send emails with a basic html table rather than a text.
You can follow this tutorial for sending emails with HTML → https://youtu.be/0N5BxEHnvuc

Note: After the long break, the pomodoro summary has to be emailed to recipients. Also, in the main menu, when a user wants to send an email to recipients, the user should be able to send it by clicking the button.

Note: Create a new gmail account to use this feature. Do not use yours due to security issueses.

Step 14

Duration: 1 day
Implementation of Main Menu UI

Step 15

Duration: 1 day
Complete Your Program
Finish your program to test.

Test Your Program

- Test your program
    - Try to find bugs in the program
- At least get feedback from **3 people** who are not from Pycoders
    - In the presentation, show this feedback
        - What they said for your game, etc.
- Try to correct your program in the light of your test results and feedback

Bonus: Change your UI since it is good to have a different UI than others. You can easily play with colours, design of components etc.

- Install “PyQT5 Designer”
- Load a UI file under the UI directory in the project. For example “login.ui”
- Save it and run your program.

Bonus: Get a “.EXE” or “.DMG” file and share it with your friends
Please follow the following videos in order to make a program working on your computer without python. In this way, you can set up your program on anyone’s computer to play with your program like a normal program.

For Windows (.exe);
https://youtu.be/UZX5kH72Yx4

For MacOS (.dmg);
https://youtu.be/IIAlkQEw8Gc

Step 16

Present Your Program

To complete the project, all members have to show up in the presentation and present the program.