

Criteria A

The scenario

My client, Mr A, attends university. As a university student, he faces challenges with academic life, such as needing to complete numerous homework assignments and revising for examinations. Mr A has shared that one of his “biggest challenges is dealing with procrastination, in the forms of spending too much time on social media and other sites” ([Appendix 1](#)) instead of reading academic material and reviewing past notes or questions. Currently, he is not using any apps for assistance on reducing his procrastination. He mentioned, in an email to me ([Appendix 1](#)), that apps he has encountered such as PawBlock and LeechBlockNG, only work “on websites and not files or games” on his laptop. Additionally, the apps he has used “don’t intervene directly but rather rely on my motivation to use or check in on the app frequently, which reduces their effectiveness for me”. Hence, I will be creating a bespoke program, free for Mr A to use and targeting activity viewed on both browser and local desktop. To address Mr A’s problem, I conducted an interview with him to learn more about his needs, in which he proposed an app where the user can “tell the app what sort of sites to monitor, for instance social media and forums” ([Appendix 1](#)). He also suggested some key features for the app, such as the ability to “set guidelines” for the app’s intervention, “for instance during which periods of the day and after how long of staying on the sites”. To remind him to work productively, he suggested that the app could display “a message on my screen to alert me” ([Appendix 1](#)). The program will need to be able to access and analyse the content of what Mr A is looking at constantly in order to promptly remind him to get on task. Mr A mainly uses his “desktop for work and procrastination”, and wishes for the app to be “downloadable on my desktop” ([Appendix 1](#)), hence I will be programming a desktop application instead of a web or mobile app.

Rationale for proposed product

I am using Python to program this program due to its robust support for Object Oriented Programming. (Elan Maulani et al., 2024) Python’s OOP capabilities are essential and suitable for this project as using inheritance, I can create a base class for activities and derive subclasses for specific categories such as ‘Games’ and ‘Social Media’. This approach ensures modularity and scalability in my application design.

Python also offers the PyQt framework, a set of C++ libraries that can be used in Python. PyQt provides tools like Qt Designer, which provides a drag-and-drop interface, allowing me to design user interfaces (GUI) visually without needing to write code manually (Yanto, Bima, Bahron, & Ikasari, 2023).

This tool will assist me by accelerating the development process of my application by allowing me to focus on the functionality of the application rather than spending excess time on the appearance.

To accurately detect and classify the activities my client engages in, I will implement a combination of text and image analysis algorithms, enhanced by machine learning models. My application will track processes on Mr A’s computer using the python system and process utilities library. I will use a Logistic Regression model¹ to classify text content into productive or unproductive categories, and will use Optical Character Recognition engine² to convert

¹ A machine learning algorithm that predicts the probability of a binary outcome by applying a logistic function to a linear combination of input features.

²A technology that converts images into text

images to text and train the machine learning model³ to analyse and classify activities involving games or non-text applications. I hope that by integrating these techniques, the application will be able to accurately categorise activities based on the content Mr A accesses.

The application will also allow my client to customise the categories of activities to monitor, set specific times for the app to be active, and define the delay period before interventions.

Success Criteria

Success Criteria	Explanation+Justification
1.Program should classify user activity as productive or unproductive	<p>My client stated that “the main use of the app should be to warn me when I am doing or looking at something unproductive” (Appendix 1), hence the program needs to differentiate between productive and unproductive activities to meet this requirement.</p> <p>To classify activity as unproductive and productive, the program will use a combination of algorithms to capture the content being viewed by the user, then analyse and categorise the content as productive or unproductive using Machine Learning. I will use logging to ensure the algorithms are running and working well.</p> <p>Text Analysis: Using Natural Language Processing (NLP) techniques, the app will extract and analyse text from websites and documents. Libraries such as scikit-learn will be used to implement these techniques(Pedregosa, 2011) .</p> <p>Image Analysis: In order to allow my program to analyse content and activity both on the browser and on the local interface, the app will likely have to capture screenshots and use optical character recognition technology to extract the text content. I will likely use Tesseract OCR, due to its free and open-source nature, alongside the extensive amount of available tutorials and guides (PySquad, 2024) for its implementation in Python.</p>
2.User can choose the site types to classify as ‘unproductive’ from predefined list of activity categories	<p>My client said “I would like to be able to tell the app what sort of sites to monitor, for instance social media and forums”. The program should hence allow users to choose what sites the program should monitor and determine to be ‘unproductive’. Users have different definitions of productivity, and this feature makes the program adaptable to their unique needs.</p> <p>Example of Categories:</p> <ul style="list-style-type: none">• Social Media: facebook, instagram, snapchat, twitter, e.t.c• Forums: reddit, quora, e.t.c• Video Entertainment: youtube, netflix, disneyplus, e.t.c <p>The program should allow users to select the categories of activities that are deemed ‘unproductive’ to customise the program to the user's needs. This customization will be implemented through a user interface where users can tick boxes of predefined categories</p>

³ A computational model trained to make predictions or classifications, enabling the application to learn from data and improve its accuracy over time.

	(e.g., social media, games, entertainment) to mark them as 'unproductive' in the Machine Learning model.
3. User can set schedule for when to run the program	<p>The client “would like to be able to set guidelines on how to intervene, for instance during which periods of the day and after how long of staying on the sites rather than working”. Accordingly, the program should allow users to set the run times for the program. Flexibility in scheduling allows the program to adapt to varying work schedules and personal preferences, enhancing its usefulness. Users can tailor the app's activity to fit their daily routines, ensuring it provides support when most needed.</p> <p>Users will be able to define specific times during which the program should be active. This will be implemented through a scheduling interface where users can set start and end times for each day of the week.</p> <p>Examples of Guidelines: Active during weekdays from 9 AM to 5 PM or only active during study hours from 6 PM to 9 PM.</p>
4. Program should monitor user activity during scheduled run time	<p>The program should respect the user's chosen run time schedules and run the unproductive and productive activity monitoring during the run times.</p> <p>For example, given the user's chosen schedule of the program being active during weekdays from 9 AM to 5 PM, the program should only run during this period of time and none other. During the period, the unproductive and productive activity monitoring should occur.</p>
5. User can set guidelines for how long the program should wait before intervening	<p>One of my client's requested features of the program is to allow users to “set guidelines on how to intervene”. Specifically, how the program should intervene and display warning messages to users upon detecting unproductive activity, for instance “after how long of staying on the sites”. To fulfil this requirement, my program should allow users to choose the duration of time needed to be reached where unproductive activity is continuously detected before displaying a warning message.</p> <p>This customization helps users avoid unnecessary interruptions and ensures interventions are timely. By setting an appropriate delay, users can balance between occasional breaks and maintaining focus on productive tasks.</p> <p>The app will allow users to specify a delay period before it intervenes with any alerts or actions. This will be implemented through a settings interface where users can input the desired delay in minutes.</p> <p>Examples of Guidelines:</p> <ul style="list-style-type: none"> • Intervene after 10 minutes of unproductive activity. • Delay intervention by 15 minutes during break times. • No delay, the program immediately displays a warning message upon sensing unproductive activity

<p>6. App should display an intervention warning after detecting unproductive activity, respecting user-set guidelines</p>	<p>My client declared that “the main use of the app should be to warn me when I am doing or looking at something unproductive”. To achieve the goal of warning the users, my client mentioned the warning “could be displaying a message on my screen to alert me” of the use. These messages should respect the user’s guideline for how long the program should wait before intervening. This means if the user chose the no delay option, the warning message will immediately appear after unproductive activity is selected, while if the user selected 10 minutes delay, then the message would appear after 10 minutes of unproductive activity.</p> <p>Reminders help users stay on track and maintain focus, contributing to increased productivity and reducing procrastination. The timely alerts provide gentle nudges to refocus on productive activities.</p> <p>After the defined delay period, the app will notify the user with a message to remind them of their unproductive activity. The message will be displayed as a pop-up alert on the screen.</p> <p>Examples of Messages:</p> <ul style="list-style-type: none"> • "Time to get back to your studies!" • "You have been on social media for 15 minutes. Please return to your work."
<p>7.App should allow for customization of intervention messages</p>	<p>Personalised messages can be more effective in motivating users to return to productive activities. Customization ensures the messages resonate with the user, increasing the likelihood of positive behaviour changes.</p> <p>Users will be able to customise the messages that appear when the app intervenes, allowing them to personalise the reminders to be more motivational or encouraging.</p>
<p>8.Program should run in the background</p>	<p>My client wishes for the program to run “automatically”, as he often forgets “to use apps” (Appendix 1).</p> <p>To do so, I will have to build the program to operate in the background, frequently checking if the current time is within scheduled run times, and run the algorithm and programs for unproductive activity warning if so.</p> <p>The app will provide real-time feedback on the user’s activity, instantly categorising it and displaying the current status (productive or unproductive).</p>
<p>9.Save user changes and settings</p>	<p>To ensure the user can reset the delay times, schedule times, and rewrite intervention messages, with their changes being saved.</p> <p>To do so, I will use a file to save user preferences and load the preferences from the file every time the program is run.</p>

Bibliography

Elan Maulani, I., Azis, I., Cahya, M. N., Komarudin, K., & Sagita, A. B. (2024). Implementation of object-oriented programming with Pyqt: Development of calculation application. *Devotion : Journal of Research and Community Service*, 5(1), 156-163. <https://doi.org/10.59188/devotion.v5i1.679>

Jain, A. (2023). Comparative analysis of Java and Python in machine learning: Investigate and compare the suitability and performance of Java and Python for machine learning tasks. *International Journal of Research Publication and Reviews*, 4(12), 1151-1167. <https://doi.org/10.55248/gengpi.4.1223.123305>

Oktaviyanus, Yanto, M., Bima, P. E., Bahron, M., & Ines Heidiani Ikasari. (2023, September 30). Pemrograman Menggunakan Java NetBeans. *jurnalmahasiswa.com*. <https://jurnalmahasiswa.com/index.php/biikma/article/view/555>

PySquad. (2024, August 1). Tesseract OCR for RAG with Python. Medium. <https://medium.com/@pysquad/tesseract-ocr-for-rag-with-python-04ab798e8886>

Pedregosa, F. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12. <https://doi.org/10.48550/arXiv.1201.0490>