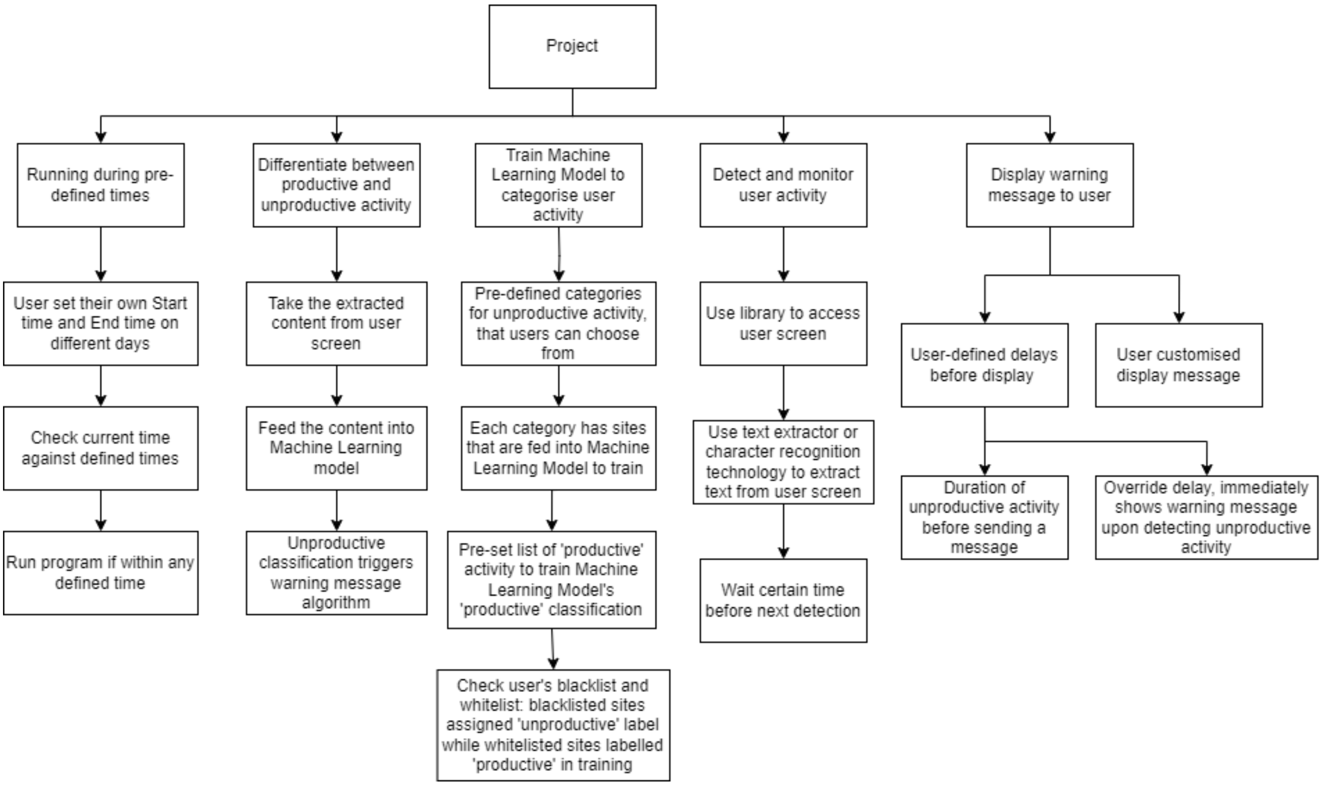
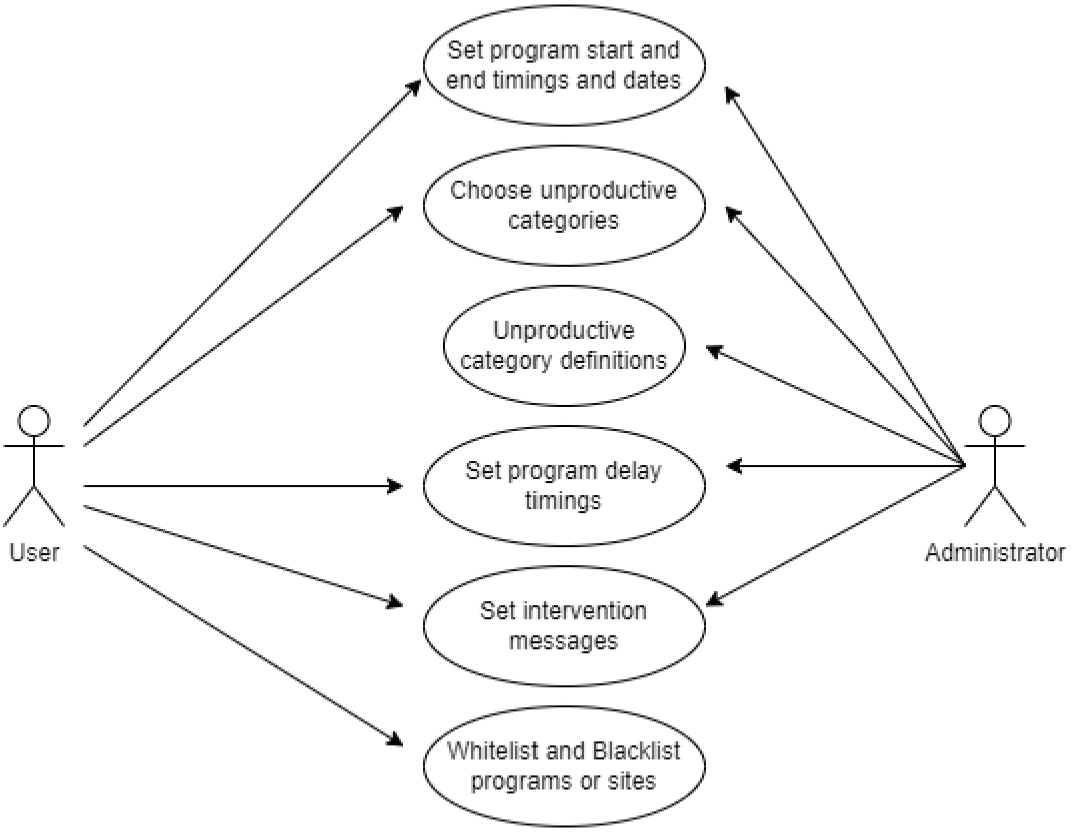
Criteria B

2) Decomposition Diagram

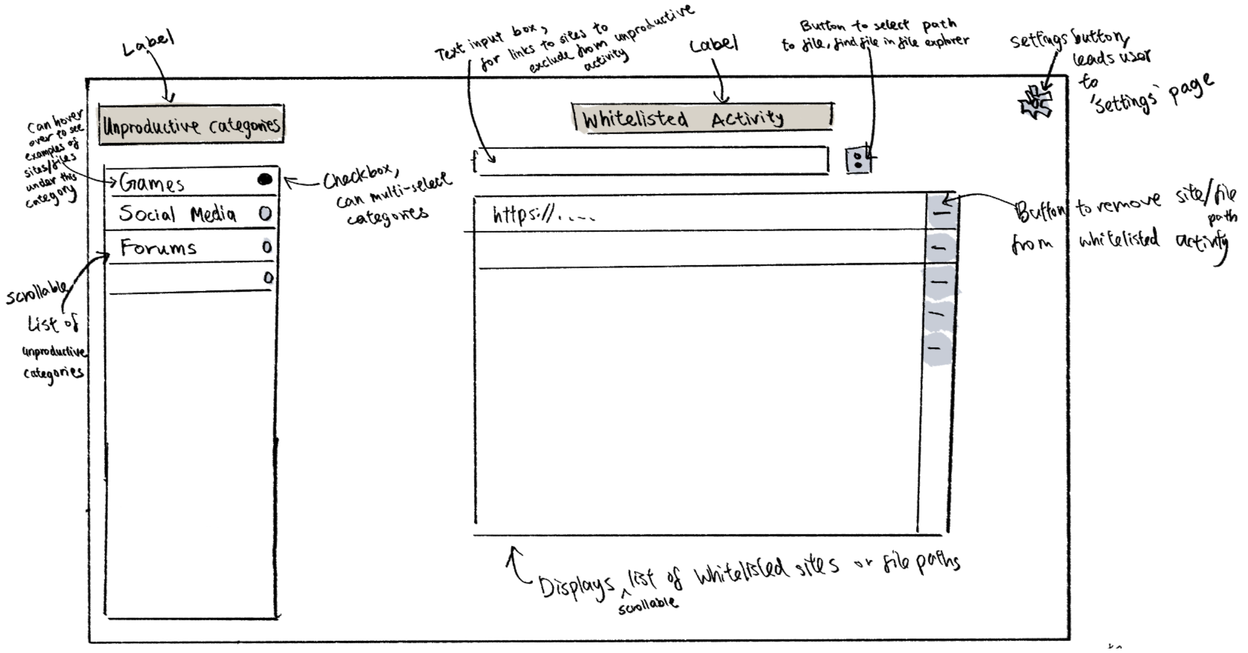

Project
Running during pre-defined times
User set their own Start time and End time on different days
Check current time against defined times
Run program if within any defined time
Differentiate between productive and unproductive activity
Take the extracted content from user screen
Feed the content into Machine Learning model
Unproductive classification triggers warning message algorithm
Train Machine Learning Model to categorise user activity
Pre-defined categories for unproductive activity, that users can choose from
Each category has sites that are fed into Machine Learning Model to train
Pre-set list of 'productive' activity to train Machine Learning Model's 'productive' classification
Check user's blacklist and whitelist: blacklisted sites assigned 'unproductive' label while whitelisted sites labelled 'productive' in training
Detect and monitor user activity
Use library to access user screen
Use text extractor or character recognition technology to extract text from user screen
Wait certain time before next detection
Display warning message to user
User-defined delays before display
User customised display message
Duration of unproductive activity before sending a message
Override delay, immediately shows warning message upon detecting unproductive activity

3) Use-Case Diagram

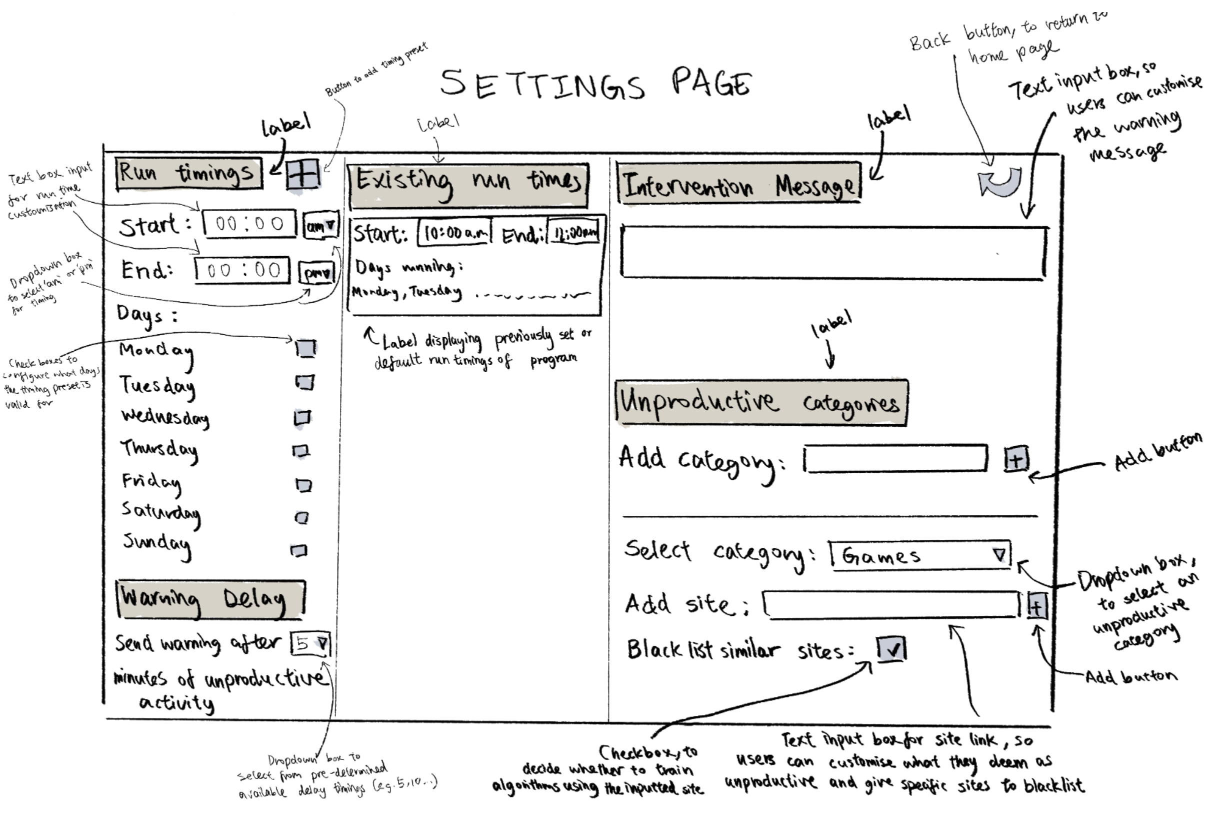

Set program start and end timings and dates
Choose unproductive categories
Unproductive category definitions
Set program delay timings
Set intervention messages
Whitelist and Blacklist programs or sites
User
Administrator

4) GUI Designs (Currently includes a lot of features that may not be implemented)

HOME PAGE

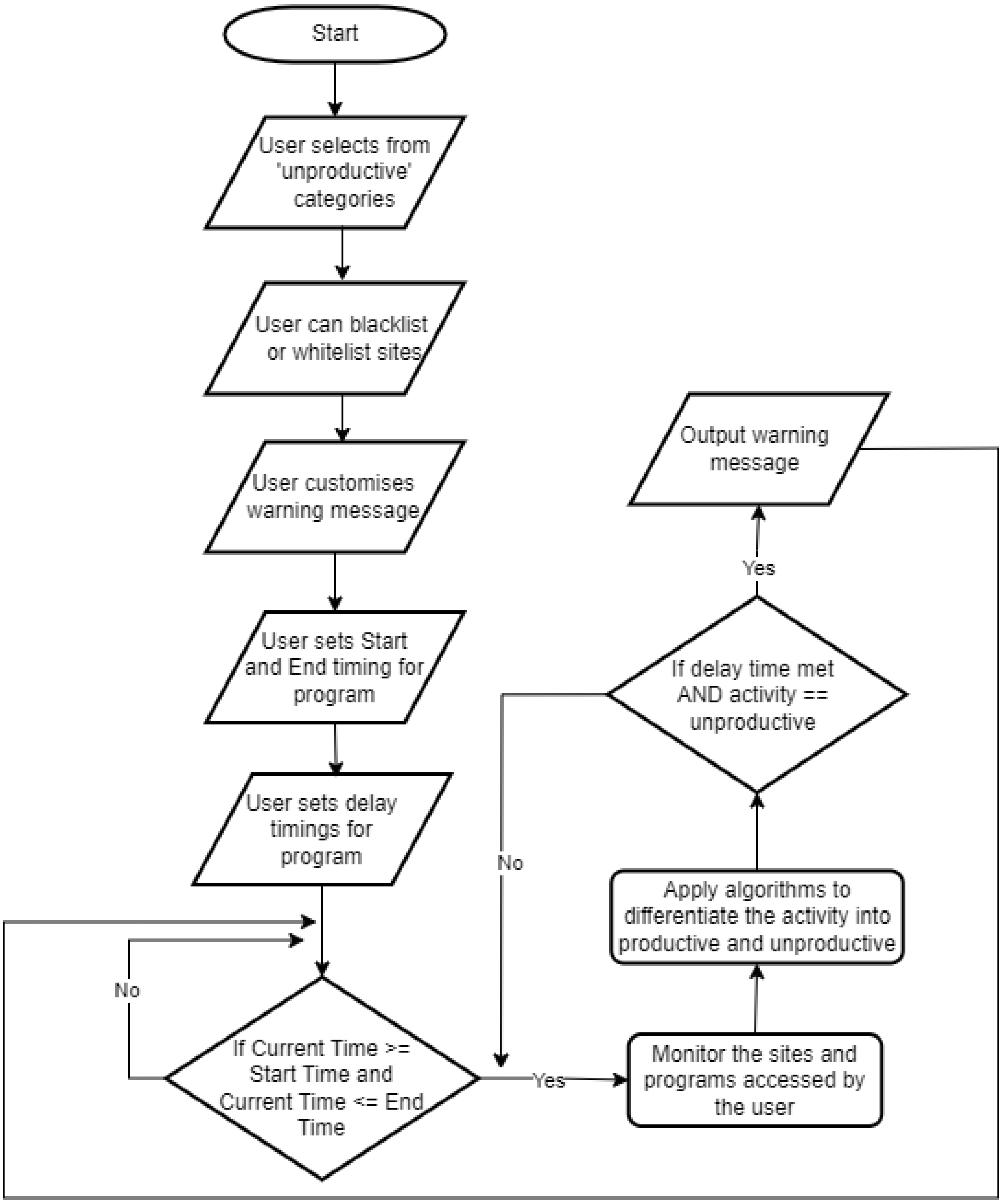

Label
Unproductive categories
Can hover over to see examples of sites/files under this category
Games ●
Social Media ○
Forums ○
○
Checkbox, can multi-select categories
Scrollable list of unproductive categories
Text input box, for links to sites to exclude from unproductive activity
Label
Whitelisted Activity
Button to select path to file, find file in file explorer
Settings button leads user to 'Settings' page
https://....
Button to remove site/file path from whitelisted activity
Displays scrollable list of whitelisted sites or file paths

SETTINGS PAGE

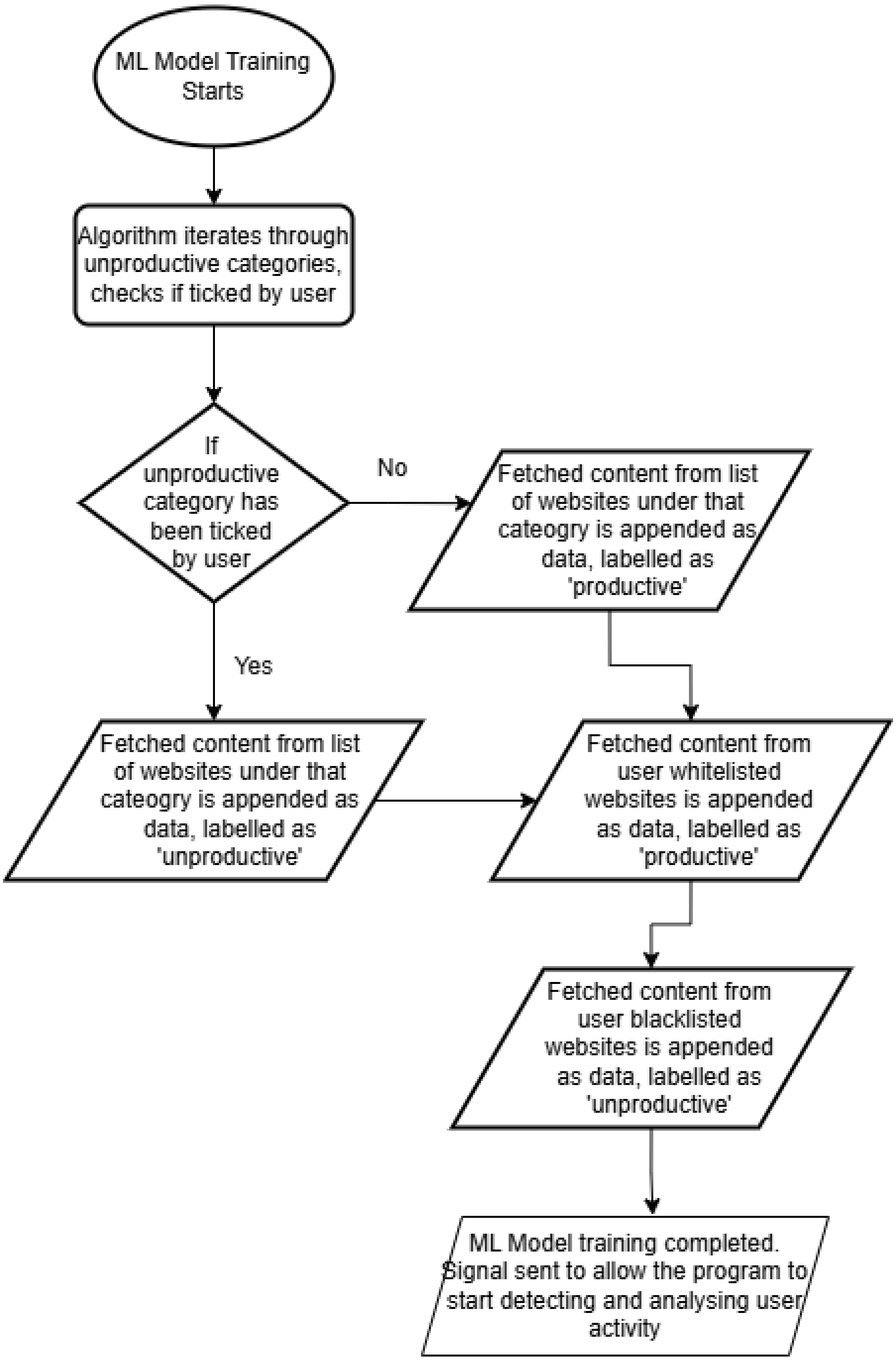

Button to add timing preset
Label
Run timings
+
Label
Existing run times
Label
Intervention Message
Back button, to return to home page
Text input box, so users can customise the warning message
Text box input for run time customisation
Start: 00 : 00 am ▾
End: 00 : 00 pm ▾
Dropdown box to select 'am' or 'pm' for timing
Start: 10:00 a.m End: 12:00 am
Days running:
Monday, Tuesday
Label displaying previously set or default run timings of program
Days:
Check boxes to configure what days the timing preset is valid for
Monday □
Tuesday □
Wednesday □
Thursday □
Friday □
Saturday □
Sunday □
Warning Delay
Send warning after 5 ▾ minutes of unproductive activity
Dropdown box to select from pre-determined available delay timings (e.g. 5, 10..)
label
Unproductive categories
Add category: +
Add button
Select category: Games ▽
Dropdown box, to select an unproductive category
Add site: +
Add button
Black list similar sites: ☑
Checkbox, to decide whether to train algorithms using the inputted site
Text input box for site link, so users can customise what they deem as unproductive and give specific sites to blacklist

5) Evidence of Algorithmic Thinking:

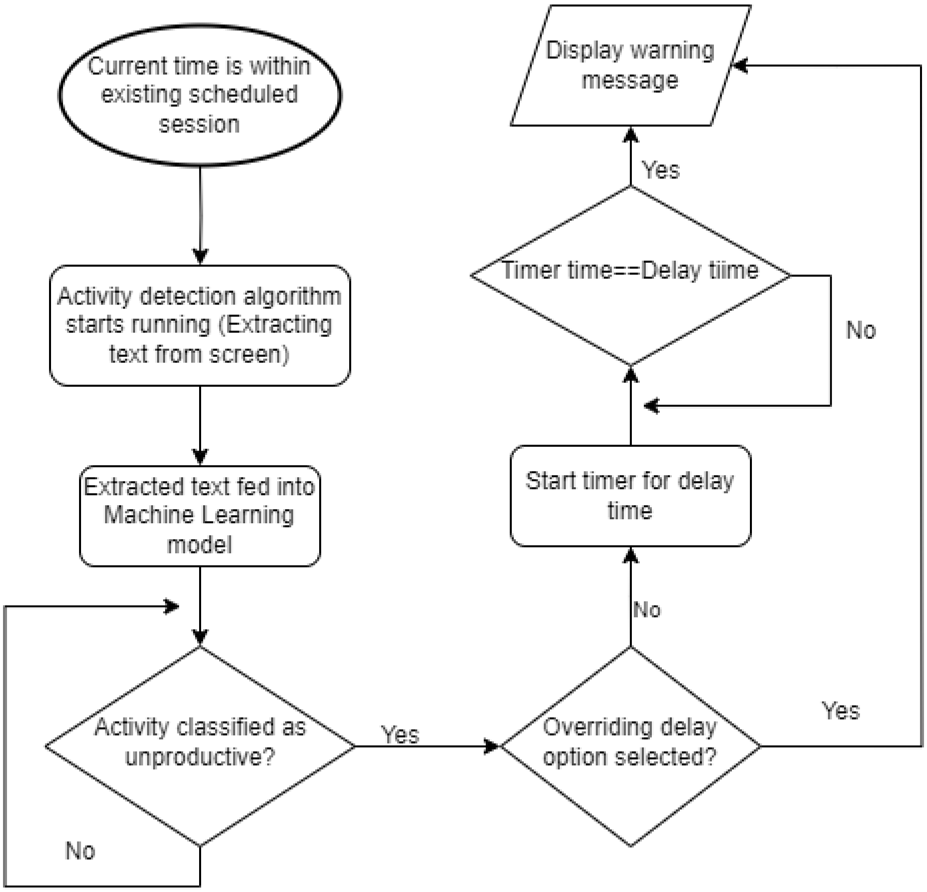
6) Flowchart


Start
User selects from 'unproductive' categories
User can blacklist or whitelist sites
User customises warning message
User sets Start and End timing for program
User sets delay timings for program
If Current Time >= Start Time and Current Time <= End Time
No
Yes
Monitor the sites and programs accessed by the user
Apply algorithms to differentiate the activity into productive and unproductive
If delay time met AND activity == unproductive
Yes
No
Output warning message

Training Machine Learning Model:


ML Model Training Starts
Algorithm iterates through unproductive categories, checks if ticked by user
If unproductive category has been ticked by user
No
Fetched content from list of websites under that cateogry is appended as data, labelled as 'productive'
Yes
Fetched content from list of websites under that cateogry is appended as data, labelled as 'unproductive'
Fetched content from user whitelisted websites is appended as data, labelled as 'productive'
Fetched content from user blacklisted websites is appended as data, labelled as 'unproductive'
ML Model training completed. Signal sent to allow the program to start detecting and analysing user activity

Flow from activity detection to displaying warning:



Current time is within existing scheduled session
Activity detection algorithm starts running (Extracting text from screen)
Extracted text fed into Machine Learning model
Activity classified as unproductive?
No
Yes
Overriding delay option selected?
Yes
No
Start timer for delay time
Timer time==Delay tiime
No
Yes
Display warning message

7) Pseudocode

Activity detection:

```
Function DetectUnproductiveActivity:
    If not currently_in_session or not model_trained:
        Print "Will not proceed with activity detection."
        Return

    Capture screen to screen_image
    Extract text from screen_image to screen_text

    If screen_text is not empty:
        Print "Extracted text from screen"
        Set unproductive_flag by predicting unproductive(screen_text)

        If unproductive_flag:
            If override_delay is enabled:
                Display warning message immediately
            Else:
                If unproductive_timer is not valid:
                    Start unproductive_timer
                Else If unproductive_timer elapsed >= wait_time:
                    Display warning message
                    Invalidate unproductive_timer
                Else:
                    Print remaining time
        Else:
            Reset unproductive_flag and unproductive_timer
    Else:
        Print "No text detected on screen"
```

Training Machine Learning Model:

```
Function TrainModel:
    Initialize data and labels lists

    For each category in categories:
        If category is selected:
            For each site in category:
                Fetch content from GivenSite
                If content exists:
                    Add content to data
                    Add "unproductive" to labels
        Else:
            For each GivenSite in category:
                Fetch content from GivenSite
                If content exists:
                    Add content to data
                    Add "productive" to labels

    For each site in whitelisted_sites:
        Fetch content from site:
        If content exists:
            Add content to data
            Add "productive" to labels

    For each site in blacklisted_sites:
        Fetch content from site:
        If content exists:
            Add content to data
            Add "unproductive" to labels

    For each site in productive_sites:
        Fetch content from site:
        If content exists:
            Add content to data
            Add "productive" to labels

    Fit vectorizer with data
    Fit model with transformed data and labels
```

Classifying activity content:

```
Function classifyContent(image):
    text = extractText(image)
    If text is not empty:
        features = transformTextToFeatures(text)
        prediction = predictProductivity(features)
        Return prediction
    Else:
        Return False

Function extractText(image):
    Use OCR to extract text from image
    Return extracted text

Function transformTextToFeatures(text):
    Convert text to numerical features using vectorizer
    Return features

Function predictProductivity(features):
    Use trained model to predict if features are unproductive
    Return prediction
```

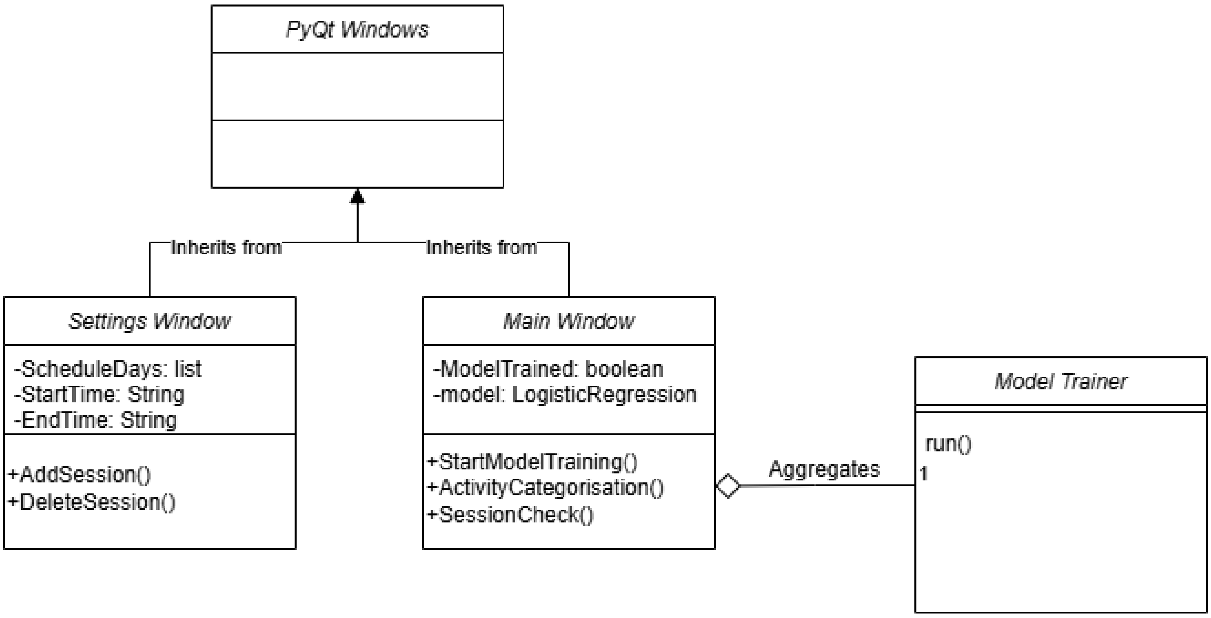
Session checking:

```
Function checkSessions():
    Get current time and day
    Set InSession to False

    For each session in sessions:
        Get start and end time
        If current day is in session's days:
            If session spans midnight:
                If current time is after start or before end:
                    Set InSession to True
                    Break
            Else:
                If current time is between start and end:
                    Set InSession to True
                    Break

    Update currently_in_session based on inSession
```

Settings loading/writing:

```
Function loadSettings():
    Try to open settings file:
        Load settings into a dictionary
        Convert session times from strings to QTime objects
        Return settings
    If file not found:
        Return default settings

Function saveSettings(settings):
    Convert QTime objects in settings to strings
    Write settings to the settings file
```

8) UML Class Diagram



9) Data Structures Table

Variables	Data Type	Explanation + Validation
Categories	List[Category]	A list of Category objects defining the types of activities to differentiate as unproductive. Each element must be an instance of a subclass of Category.
Schedule	Dict[str, Tuple[time, time]]	A dictionary mapping days of the week to tuples of start and end times. Keys must be strings representing days of the week (e.g., "Monday"). Values must be tuples containing two time objects. Example: {"Monday": (time(9, 0), time(17, 0))}. time(9, 0) is an instance of the time class from Python's datetime module. time(9, 0) represents 9:00 AM.
delayPeriod	Integer	The delay period in minutes before the app intervenes. Must be an integer within the range [1, 120]. Example: delayPeriod = 10.
message	String	The customizable intervention message e.g. "Time to get back to your studies!"
log	List[Dict[str, Union[str, datetime]]]	A list of dictionaries, each representing a logged activity with its status and timestamp. Each dictionary must contain keys activity, status, and timestamp. activity: String representing the activity. status: String representing the classification (e.g., "productive" or "unproductive"). timestamp: datetime object representing the time of the activity. Example: log = [{"activity": "facebook.com", "status": "unproductive", "timestamp": datetime.now()}].

10) Test Plan

Success Criteria no.	Test purpose	Data entered	Expected output	Explanation
1	Validate activity classification	- Text from a social media post, social media category selected -Text from a social media post, social media category not selected -A local video game, with the game category selected -A local video game, with the game category not selected	-Classified as unproductive -Classified as productive -Classified as unproductive -Classified as productive	Ensure the program correctly classifies different types of activities, no matter on a browser or a local file.
2	Test category selection functionality	- Select "Social Media" and "Games" categories -Deselect all categories	-Categories added to monitored list -Monitored list should be empty	Ensure users can select and save predefined categories to monitor.
3	Validate setting schedule functionality	- Set schedule to run from 9 AM to 5 PM on weekdays -Delete an existing schedule	- Schedule saved and updated to config file -Schedule deleted on UI and from config file	Ensure the user can customise the date and time that the app runs
4	Verify monitoring of user activity during scheduled run time	-Running program during time within a scheduled run time -Running program outside of schedules run times	-Logging should display 'monitoring start' and classification feedback -Logging should display 'not in schedule', 'no prediction activity occurring'	Ensure the app respects the user's specified schedule for monitoring.
5	Verify intervention delay setting	- Set delay period to 10 minutes -Select override delay button	- Delay period saved and interventions occur after 10 minutes of continuous unproductive activity -Override delay set to true in config file, interventions appear immediately upon detection	Ensure the app waits the specified delay period before intervening.

6	Validate intervention message display	-Don’t edit custom message, leave as default - Set custom message "Please return to your studies"	- Default message displayed during intervention - Custom message displayed during intervention	Check if intervention message is being displayed correctly
7	Validate customizable intervention message	-Don’t edit custom message, leave as default - Set custom message "Please return to your studies"	-Default message is saved in config file -Intervention message changed to new custom message in config file	Ensure users can customise intervention messages and they are saved correctly.
8	Program should run in the background, unless user chooses to close it	-Open laptop without opening program -Open program and closing the window	-During scheduled run time, unproductive activity classification is occurring and warning messages appear, respecting user delay options -Program stops. Next time program runs should be if the user opens the program or if the laptop is closed then opened.	Ensure the app runs independently, not relying on user motivation.
9	Program should save user changes and run with updated settings	-Open program without changing any settings -Change selection of categories in unproductive category selection -Add websites to blacklist and whitelists	-The user preference storage file should remain the same -The preference file should update to have the user selected categories under unproductive categories -The preference file should update to have the added websites in the blacklist or whitelist	To ensure the users can customise the program and their changes are saved.