Testing Plan

### Our project contains the following areas that require testing:

* Windows (GUI)
* Data file
* Settings File
* Drowsy Detection in Recording Window
* Alerting system
* Data Visualizations

### To test the success of our project, we will assess our project with the following set of goals:

1. Reliable Window navigation
2. Automatic creation of data file
   1. Clearing of existent data file if existent from previous run
      1. Occurs during switch from Main → Preview window
3. Automatic creation of settings file
   1. If not already present
4. Data from settings file can be presented to user in Settings Window
5. User input from Settings Window can be reliably saved to settings file
6. Thresholds derived from settings will update when navigating from Settings Window to Recording Window
7. Reliable eye and face detection
8. Reliable blink detection
9. Reliable alert creation and window switching
10. Wake-up button creates a datapoint
11. Snooze button pausing Alarm for user-specified duration
12. Accurate graph generations with collected data
13. Exports the data to a csv file

### To test our project on these targets, follow the next steps for each goal:

1. Reliable Window navigation
   1. Start program using executable (Main.exe)
   2. *The Main Window should open*
   3. Click ‘End Program’ button
   4. *Program should close*
   5. Reopen program using executable (Main.exe)
   6. *The Main Window should open*
   7. Click ‘Run’ button
   8. *Preview Window should open*
   9. Click ‘Settings’ button
   10. *Settings Window should open*
   11. Click ‘Save’ button
   12. *A popup message should appear “Settings saved to disk”*
   13. Click ‘OK’ button on popup message
   14. *Preview Window should reappear*
   15. Click ‘Settings’ button again
   16. *Settings Window should open*
   17. Click ‘Back’ button
   18. *Preview Window should reappear*
   19. Click ‘Continue’ button
   20. *Recording window should appear*
   21. Click ‘Settings’ button
   22. *Settings Window should open*
   23. Click ‘Save’ button
   24. *A popup message should appear “Settings saved to disk”*
   25. Click ‘OK’ button on popup message
   26. *Recording Window should reappear*
   27. Click ‘Settings’ button again
   28. *Settings Window should open*
   29. Click ‘Back’ button
   30. *Recording Window should reappear*
   31. Blink long enough for an alert to be triggered
   32. *Alert Window should appear with a continuous beeping noise*
   33. Click ‘Snooze’ button
   34. *No window change should occur*
   35. Click ‘Wake Up’ button
   36. *Recording Window should reappear*
   37. Click ‘End Record’ button
   38. *Results Window should appear*
   39. Click ‘Main Menu’ button
   40. *Main Window should appear*
   41. Navigate to Results Window
       1. Main → Preview → Recording → Results
   42. Click ‘End Program’ Button
   43. *Program should close*
2. Automatic creation of data file
   1. Delete data.csv file from directory (if existent)
   2. Start program using executable (Main.exe)
   3. *The file ‘data.csv’ should be created and the first line should contain:*
      1. *time,duration,type*
   4. End program
   5. Input random text into the data.csv file (in no particular format)
   6. Start program using executable (Main.exe)
   7. *Inspect data.csv and check that the random text is deleted and first line contains:*
      1. *time,duration,type*
3. Automatic creation of settings file
   1. Delete settings.json file from directory (if existent)
   2. Start program using executable (Main.exe)
   3. *The file settings.json’ should be created and contain:*
      1. *eye\_thresh:3, alarm\_thresh:10, snooze\_thresh:10 in JSON format*
   4. End program
   5. Input random text into the settings.json file (in no particular format)
   6. Start program using executable (Main.exe)
   7. *A popup message should appear “File Corruption detected! Reverting ‘settings.json’ to default values”*
   8. Click “OK” button
   9. *Main Window should appear*
   10. *settings.json should have the same default values as step c*
4. Settings info from settings file can be presented to user in Settings Window
   1. Open file ‘settings.json’, take note of values, close file
   2. Start program using executable (Main.exe)
   3. Click ‘Run’ button
   4. Click ‘Settings’ button
   5. *The settings values shown should equal the values stored in ‘settings. Json’*
5. User input from Settings Window can be reliably saved to settings file
   1. Start program using executable (Main.exe)
   2. Click ‘Run’ button
   3. Click ‘Settings’ button
   4. Edit all fields to a different integer value
   5. Click ‘Save’ button
   6. *Popup message should appear “Settings saved to disk”*
   7. Click ‘OK’ button
   8. Click ‘Settings’ button
   9. *Data in fields should be same as the values entered in step d*
   10. Enter all fields to a different integer value
   11. Click ‘Back’
   12. *Preview Window should appear*
   13. Click ‘Settings’ button
   14. *The values entered in step i should NOT be present, the values should be from Step d*
6. Thresholds derived from settings will update when navigating from Settings Window to Recording Window
   1. Delete settings.json from directory (if existent)
   2. Start program using executable (Main.exe)
   3. Navigate to Recording Window
      1. Main → Preview → Recording
   4. Look away from camera for 10 seconds
   5. *Alert Window should appear with a continuous beeping noise*
   6. Click ‘Wake Up’ button
   7. Click ‘Settings’ button
   8. *Settings Window should appear*
   9. Change eye\_thresh to 1 and alarm\_thresh to 3
   10. Click ‘Save’ button
   11. Click ‘OK’ on popup
   12. Look away from camera for 3 seconds
   13. *Alert Window should appear*
7. Reliable eye and face detection
   1. Start program using executable (Main.exe)
   2. Click ‘Run’ button
   3. The Preview Window should show the user’s webcam
   4. *The preview should show the following messages:*
      1. *‘Face Detected’ and ‘Eyes Detected’*
   5. *The face and eyes should be detected even when moving one’s head around*
      1. *Note: User needs to be fairly close to camera and in a well-lit room*
8. Reliable blink detection
   1. Start program using executable (Main.exe)
   2. Click ‘Run’ button
   3. The Preview Window should show the user’s webcam
   4. *When the user blinks, the following message should show: ‘Blink Detected’*
      1. *Note: User needs to be close to camera and in a well-lit room*
9. Reliable alert creation and window switching
   1. Delete settings.json from directory (if existent)
   2. Start program using executable (Main.exe)
   3. Click ‘Run’ button
   4. Click ‘Continue’ button
   5. Close your eyes for 10 seconds
   6. *Alert Window should appear with a continuous beeping noise*
   7. Press ‘Wake Up’ button
   8. *Recording Window should appear*
   9. Look away from camera (no face detected) for 10 seconds
   10. *Alert Window should appear with a continuous beeping noise*
10. Wake-up button creates a datapoint
    1. Delete settings.json from directory (if existent)
    2. Start program using executable (Main.exe)
    3. Click ‘Run’ button
    4. Click ‘Continue’ button
    5. Close your eyes for 10 seconds
    6. Once on the Alert Window, click ‘Wake Up’ button
    7. Click ‘End Record’ button
    8. Open the data file (data.csv)
    9. *The data file should have three lines in the following format:*
       1. *datetime of blink, duration of blink, 2*
       2. *datetime of button click, 0, 3*
11. Snooze button pausing Alarm for user-specified duration
    1. Start program using executable (Main.exe)
    2. Click ‘Run’ button
    3. Click ‘Continue’ button
    4. Close your eyes for 10 seconds
    5. Once on the Alert Window, click ‘Snooze’ button
    6. *The Alert Window should now have a countdown equal to the time specified in the settings*
12. Accurate graph generations with collected data
    1. Start program using executable (Main.exe)
    2. Click ‘Run’ button
    3. Click ‘Continue’ button
    4. Run a session that will generate data by blinking for longer than the duration specified in the settings
    5. Click ‘End Record’ button
    6. From the Results Window, click on any of the four data visualizations available
    7. *Each graph should accurately reflect the data collected in the data.csv file*
13. Exports the data to a csv file
    1. Start program using executable (Main.exe)
    2. Click ‘Run’ button
    3. Click ‘Continue’ button
    4. Run a session that will generate data by blinking for longer than the duration specified in the settings
    5. Click ‘End Record’ button
    6. Click ‘Export Data’ button
    7. *A new file will be created in the exports folder with the following naming convention:*
       1. *datetime of recording\_data.csv*
       2. *If no exports directory exists, a new one will be created*

### 

### Results from testing:

1. Reliable Window navigation
   * Pass
2. Automatic creation of data file
   * Pass
3. Automatic creation of settings file
   * Pass
4. Data from settings file can be presented to user in Settings Window
   * Pass
5. User input from Settings Window can be reliably saved to settings file
   * Pass
6. Thresholds derived from settings will update when navigating from Settings Window to Recording Window
   * Pass
7. Reliable eye and face detection
   * Pass
8. Reliable blink detection
   * Pass
9. Reliable alert creation and window switching
   * Pass
10. Wake-up button creates a datapoint
    * Pass
11. Snooze button pausing Alarm for user-specified duration
    * Pass
12. Accurate graph generations with collected data
    * Pass
13. Exports the data to a csv file
    * Pass