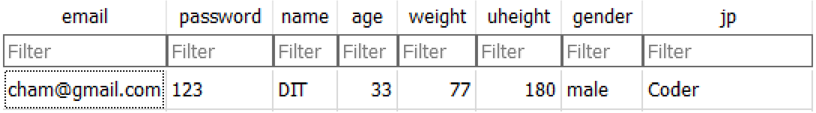
Test Report

Available test-cases:	14
Conducted test-cases:	14
Passed test-cases:	12
Failed test-cases:	02

User Credentials:



email	password	name	age	weight	uheight	gender	jp
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
cham@gmail.com	123	DIT	33	77	180	male	Coder

Test Specification

1. Check In and Check Out

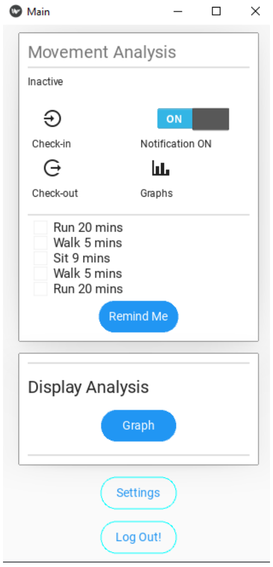
Linked to requirement: REQ_001, REQ_003, REQ_022

Test description:

1. Start the application. Enter the User Credentials provided and click ‘Login’ to enter the main screen.
2. Check Availability of the 2 Buttons- Check In/ Check Out and 1- Notification Switch, as well as the Graph Button
3. Label “Inactive” above the notification showing the status of the user
4. Set breakpoints at the following code sections
   a. At calling the method ‘check_in_deactivate’
   b. At calling the method ‘check_out_deactivate’
5. Click Check in Button
6. Click on the check in button again to check whether it can be clicked again
7. Click on the Checkout Button
8. Click on the check in button again to check whether it can be clicked again
9. Check whether the motion profile is visible.

Expected results:

At 2: We can should observe 2 buttons: check-in, check-out and a notification switch.
At 5: the Check-In button connects with the function in 3-a. The Button is disabled and the text label “Inactive" changes text.
At 6. The check in button is unclickable as it is disabled and movement profile starts and the function “check_in_deactivate”.
At 7: The Check-Out button connects with the function in 3-b while it gets disabled, the Check-in button is abled along with it.
At 8: The check in button is unclickable as it is disabled
At 9: label of motion profile is visible



□ Run 20 mins
□ Walk 5 mins
□ Sit 9 mins
□ Walk 5 mins
□ Run 20 mins

Partial results:

At 2: passed
At 5: passed
At 6: passed
At 7: passed
At 8: passed
At 9: passed

Testcase result: Passed

2. Notification

Linked to requirement: REQ_002

Test description:

1. Start the application. Enter credentials and enter the Main Screen
2. Click on the Check in Button to start the movement analysis.
3. Click Notification button to switch it off.
4. Click on the check box of motivation task
5. Click on the Remind me button
6. Click Notification button to switch it on.
7. Click on the Check box and remind me button.

Expected results:

At 3: The text of the switch will read: Notification off.

At 4: No check box is clickable(disabled)

At 5: Cannot click as it is disabled

At 6: Notification button now reads Notification On.

At 7. Remind me Button is enabled as well and the check boxes are enabled.

Partial results:

At 3: Passed

At 4: Passed

At 5: Passed

At 6: Passed

At 7. Passed

Testcase result: Passed

3. Notification

Linked to requirement: REQ_004

Test description:

1. Start the Application
2. Go to Main screen
3. Click Check-in
4. Notification should pop up after a while.

Expected results:

At 4. Since the Notification required a reading from the User's profile, we will not be able to get the notification when the user has been sitting for a longer period of time.

Partial results:

At 4: failed

Testcase result: failed

4. Notification & Check In Linked to Motivation Task Manager

Linked to requirement: REQ_005, REQ_006, REQ_007, REQ_022

Test description:

1. Start the application. Enter credentials and enter the Main Screen
2. Check Check-in Button
3. Check motivation tasks
4. Check checkboxes linking the tasks
5. Click on Check IN while the Notification Button is on.
6. Check the Motivation tasks provided by the app.
7. Click on any of the motivation tasks.
8. Click on Remind me button
9. Turn off Notification
10. Click on the motivation tasks checkbox.

Expected results:

At 2: Check IN button Is observed in the Main Screen

At 3: Five Motivation Tasks list can be observed in the Main Screen

At 4: Checkboxes besides the list of tasks are observed

At 5: The Motivation profile will start running and getting the users data

At 6: The Task will be provided to the user according to the user’s profile

At 7: Clickable. The User has the option to click on the tasks, that they want to do.

At 8: All task selected will be reminded in a notification after a specified time interval

At 9: Notification switch flips. Tasks selected prior are automatically unclicked

At 10: The Checkboxes are disabled

Partial results:

At 2: passed

At 3: passed

At 4: passed

At 5: passed

At 6: passed

At 7: passed

At 8: passed

At 9: passed

Testcase result: passed

5. Notification per Hour Settings

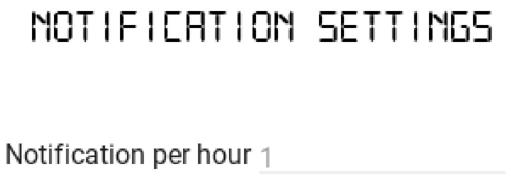
Linked to requirement: REQ_008

Test description:

1. Start the application. Enter credentials and enter the Main Screen
2. Go to Settings
3. Click on Notification Settings
4. Enter an Integer

Expected results:

At 4: Integer added will be used in a clock function. That will show the exact number of notification per hour.



NOTIFICATION SETTINGS

Notification per hour 1

Partial results:

At 4: passed

Testcase result: passed

6. Model Creation

Linked to requirement: REQ_009

Test description:

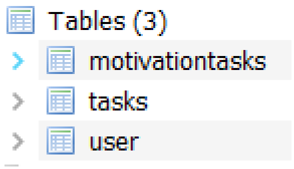
1. Start the application. Enter credentials and enter the Main Screen
2. Check movement analysis text.
3. Click on the Check-In Button.
4. Observe the changes in the movement analysis text.
5. Check motivationtask.db

Expected results:

At 2: clickable

At 3: start analyzing and movement profile is to be changed using accelerometer or random values generated.

At 5: data should be recorded in the database along with the movement profile of the user in the trainer.db in a table called movementtasks made using SQL Alchemy.



Tables (3)
- motivationtasks
- tasks
- user

Partial results:

At 2: passed

At 3: passed

At 4: passed

Testcase result: Passed

7. Settings

Linked to requirement: REQ_013

Test description:

1. Start the application. Enter credentials and enter the Main Screen
2. Click on the Settings button and select the User Profile Settings
3. Check User Information
4. Click “Back Button” and Go to Edit profile Settings
5. Check Text field and enter new Values
6. Go back to User Profile Setting and check data

Expected results:

At 3: Logged in User Data would be observable their personal data information such as email, username, age, height, weight, job position.

At 5: The user's profile would be shown in the text field and it would be editable. Entering values in the corresponding text field would make changes to the user profile in the database.

At 6: Going back to the user profile setting will show us the user's newly inserted data.

Partial results:

At 2: passed

At 5: passed

At 6: passed

Testcase result: passed

8. GUI - Information on sensor and memory access

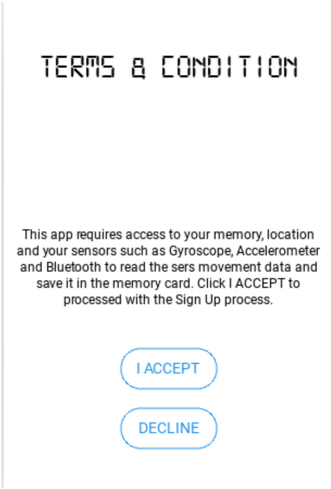
Linked to requirement: REQ_017

Test description:

1. Start the application.
2. Click on Sign Up Button
3. Check the terms and condition button
4. Click "I agree"
5. Go back to login and enter credentials.
6. Scroll down and Click Settings button.
7. Click System Settings
8. Click any Sensors.

Expected results:

At 3: The user is directed to the Terms and Condition Screen which explains about the memory usage as well as using the sensors to be used when using this app.



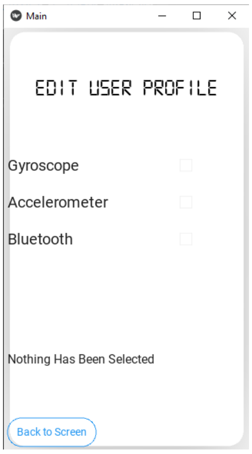
TERMS & CONDITION

This app requires access to your memory, location and your sensors such as Gyroscope, Accelerometer and Bluetooth to read the sers movement data and save it in the memory card. Click I ACCEPT to processed with the Sign Up process.

At 4: Clickable. And the User processed to the Signup Screen.

At 7: Clickable

At 8: The user can click any one of the Sensors displayed. Gyroscope/ Accelerometer/ Bluetooth.



Main

EDIT USER PROFILE

Gyroscope □

Accelerometer □

Bluetooth □

Nothing Has Been Selected

Partial results:

At 3: passed

At 4: passed

At 7: passed

At 8: passed

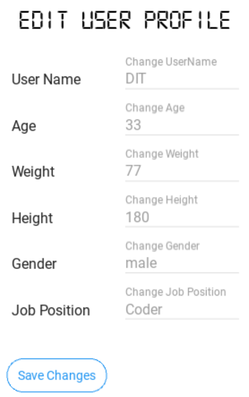
Testcase result: passed

9. GUI- User Identification

Linked to requirement: REQ_014

Test description:

1. Enter the app
2. Click on the Sign-Up button
3. Click on “I Agree” button
4. Check different Text Input labelled at the side.
5. Click on the Text Input field to enter the data.
6. Enter New User Data. ('cham@gmail.com', '123', 'DIT', 33, 77, '180', 'male', 'Coder')
7. Click Register Button which is connected to the function, registerbtn().
8. Enter the Credential used in the signup part in the text box in the login screen, Email: 'cham@gmail.com' and Password: 123
9. Click login
10. In the main screen, go to settings button and click on it.
11. In the setting screen, go to edit user setting button by clicking
12. Check the text fields and change values. Change age from 33 to 42



EDIT USER PROFILE

User Name	Change UserName DIT
Age	Change Age 33
Weight	Change Weight 77
Height	Change Height 180
Gender	Change Gender male
Job Position	Change Job Position Coder

Expected results:

At 2: Clickable and processed to terms and condition

At 4: Empty text field will as user new information.

At 6. Enter New user data. Where the passwords should match.

At 7. Clickable and Accessible

At 9. Clicking on login will lead you to the main screen means that point 8 was correct and it works.

At 12: The text field observe have a value of the old user data, clicking on it and changing the number will change the value in the database and the user profile value will have changes. Showing that the user can alter his values.

Partial results:

At 2: passed

At 4: passed

At 7: passed

At 9: passed

Testcase result: passed

10. Motivation Task Manager

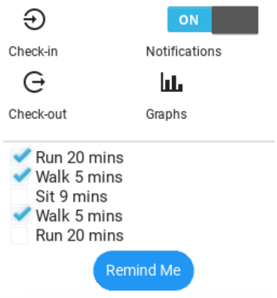
Linked to requirement: REQ_016

Test description:

1. Start the application.
2. Login in with the User Credential.
3. Check if notification is on in main screen
4. If not, click switch to turn on notification.
5. In the Main Screen, click on the motivation task checkboxes.
6. Click on remind me.

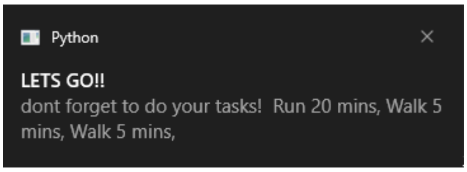
Expected results:

At 5. Motivation task of relevant user profile is automatically assigned to the person according the age and gender and job profile. They can click on the checkboxes, of what feels suitable.



- ✓ Run 20 mins
- ✓ Walk 5 mins
- □ Sit 9 mins
- ✓ Walk 5 mins
- □ Run 20 mins

At 6. Clicking on Remind me, would give the user a notification showing only the notification that the user has selected.



LETS GO!!
dont forget to do your tasks! Run 20 mins, Walk 5 mins, Walk 5 mins,

Partial results:

At 5: passed

At 6: passed

Testcase result: passed

11. Recording of the end user's movement profiles

Linked to requirement: REQ_019

Test description:

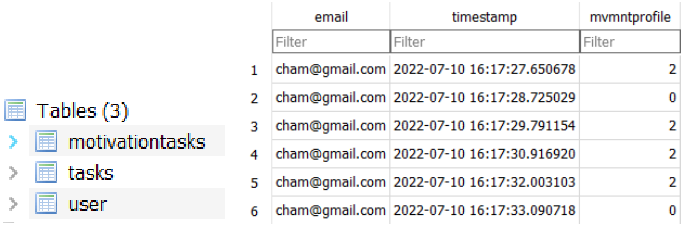
1. Start the application.
2. Login in with the User Credential.
3. Click on the Check in button.
4. Check Function called AI_callback_get_movement_data()
5. Check SQL Database, Trainer.
6. Check Table name, motivationtasks.

Expected results:

At 3. The app will start getting data from sensor/random().

At 4. This function is where the data from the sensor is predicted and saved in the database.

At 5. Opening trainer.db will show us the User Profile Table, Tasks table and the Movement Profile Table (which contain the columns containing the timestamp and the reading of that db).



	email	timestamp	mvmntprofile
1	cham@gmail.com	2022-07-10 16:17:27.650678	2
2	cham@gmail.com	2022-07-10 16:17:28.725029	0
3	cham@gmail.com	2022-07-10 16:17:29.791154	2
4	cham@gmail.com	2022-07-10 16:17:30.916920	2
5	cham@gmail.com	2022-07-10 16:17:32.003103	2
6	cham@gmail.com	2022-07-10 16:17:33.090718	0

Partial results:

At 5: passed

At 6: passed

Testcase result: passed

12. Graphs D/W/M

Linked to requirement: REQ_019, REQ_20, REQ_021, REQ_022

Test description:

1. Start the application.
2. Login in with the User Credential.
3. Click on the Check in button.
4. Check Function called AI_callback_get_movement_data()
5. Check SQL Database, Trainer.
6. Check Table name, motivation tasks.
7. Check Timestamp
8. Check Main Screen and click on the Graph Button.
9. Click on Daily
10. Click on Weekly
11. Click on Monthly
12. Set breakpoints at the following code sections
    a. At calling the method ‘DPiechart()’
    b. At calling the method ‘WPiechart()’
    c. At calling the method ‘MPiechart()’

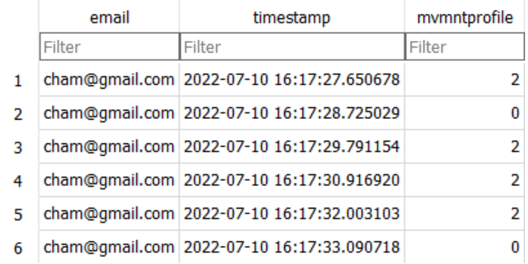
Expected results:

At 3. The app will trigger getting data from sensor/random ().

At 4. This function is where the data from the sensor is predicted and saved in the database.

At 5. Opening trainer.db will show the User Profile Table, Tasks table and the Movement Tasks Table

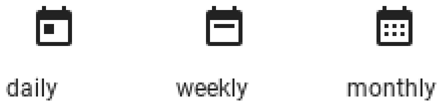
At 6: Motivation Tasks name contains the columns containing the timestamp and the movement profile of user.



	email	timestamp	mvmntprofile
	Filter	Filter	Filter
1	cham@gmail.com	2022-07-10 16:17:27.650678	2
2	cham@gmail.com	2022-07-10 16:17:28.725029	0
3	cham@gmail.com	2022-07-10 16:17:29.791154	2
4	cham@gmail.com	2022-07-10 16:17:30.916920	2
5	cham@gmail.com	2022-07-10 16:17:32.003103	2
6	cham@gmail.com	2022-07-10 16:17:33.090718	0

At 7: The Timestamp column contains the date and time that is to be used in the Graphical plotting.

At 8. Graph Button will lead to the Graph Screen which contains 3 buttons.



At 9. Clicking on Daily Button will trigger a Pie chart that will display today’s data

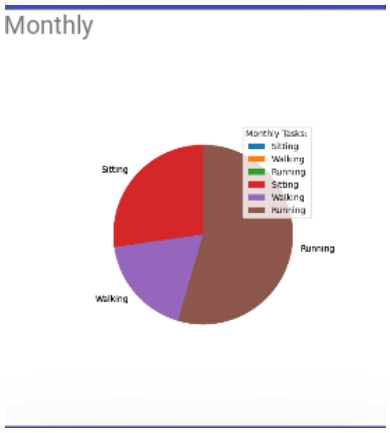
At 10. Clicking on Weekly Button will trigger a Pie chart that will display weekly’s data

At 11. Clicking on Monthly Button will trigger a Pie chart that will display monthly’s data

At 12. All data collected from movement profile is summed and using time stamp, we get the daily values by counting it and plotting it.

At 13. All data collected from movement profile is summed and using time stamp, we get the weekly values by counting it and plotting it.

At 14 All data collected from movement profile is summed and using time stamp, we get the monthly values by counting it and plotting it.



Partial results:

At 5: passed

At 6: passed

At 9: passed

At 10: passed

At 11: passed

At 12: passed

At 13: passed

At 14: passed

Testcase result: passed

13. Testcase

Linked to requirement: REQ_024

Test description:

1. Start the Application.
2. Go to Sign up
3. Check text input field
4. Go Back
5. Check Main Screen
6. Check language of the Motivation Task
7. Go to settings screen -> Edit User Data
8. Check Language of Input required

Expected results:

At 3. We observe that the user is asked to type male or female which is in English.

At 6. Motivation Tasks given are given in English.

At 7. Edit User Data, input required is also similar to number 3, enter male or female (not Herr or Frau)

Partial results:

At 3: Passed

At 6: Passed

A t 7: Passed

Testcase result: passed

14. Granting sensor and memory permissions

Linked to requirement: REQ_018

Test description:

1. Start The Application and login with credentials.
2. Go to Main Screen

Expected results:

At 2. We expect a pop up/notification to come up asking that the User to allow the user to access the specific sensor of the target platform.

Partial results:

At 2: failed

Testcase result: failed

15. AI Model Recording

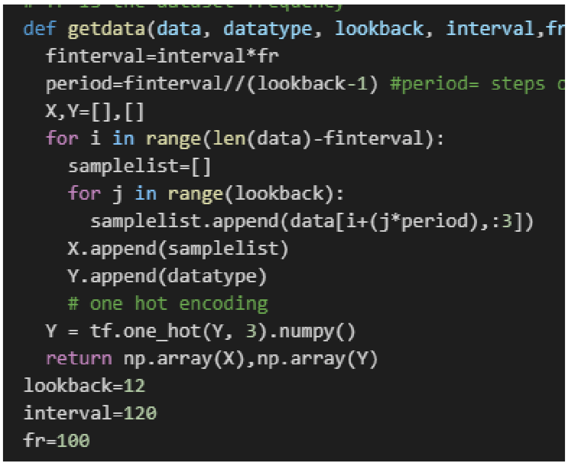
Linked to requirement: REQ_010, REQ_011, REQ_012

Test description:

1. Check The model.py file
2. Check the getdata() (Image-uploaded)
3. Check concatenate function
4. Check Dataset used.

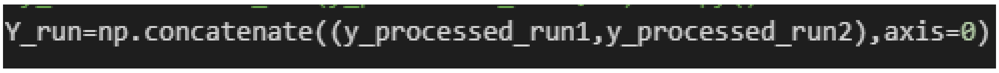
Expected results:

At 2. Frequency is set to 100 meaning 1 sec, so we set Lookback as 12 while 120 is the interval of the whole preprocessing. This is according to the data provided by the professor. Check image attached below.



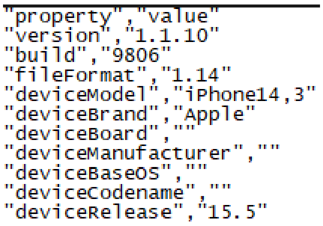
```
def getdata(data, datatype, lookback, interval,fr
  finterval=interval*fr
  period=finterval//(lookback-1) #period= steps c
  X,Y=[],[]
  for i in range(len(data)-finterval):
    samplelist=[]
    for j in range(lookback):
      samplelist.append(data[i+(j*period),:3])
    X.append(samplelist)
    Y.append(datatype)
    # one hot encoding
  Y = tf.one_hot(Y, 3).numpy()
  return np.array(X),np.array(Y)
lookback=12
interval=120
fr=100
```

At 3. The Image below shows 2 different dataset concatenate. Each of them attained through different mobile phones



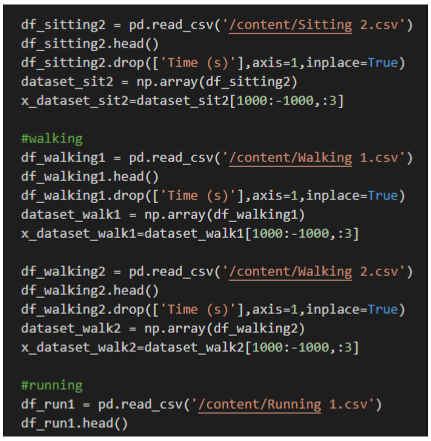
```
Y_run=np.concatenate((y_processed_run1,y_processed_run2),axis=0)
```

One is attained using Redmi Note 8, XiaoMI device while the other was attained using Iphone.



```
"property","value"
"version","1.1.10"
"build","9806"
"fileFormat","1.14"
"deviceModel","iPhone14,3"
"deviceBrand","Apple"
"deviceBoard",""
"deviceManufacturer",""
"deviceBaseOS",""
"deviceCodename",""
"deviceRelease","15.5"
```

At 4. Data Set Contains dataset of Users: Sitting/Walking/ Running



```
df_sitting2 = pd.read_csv('/content/Sitting 2.csv')
df_sitting2.head()
df_sitting2.drop(['Time (s)'],axis=1,inplace=True)
dataset_sit2 = np.array(df_sitting2)
x_dataset_sit2=dataset_sit2[1000:-1000,:3]

#walking
df_walking1 = pd.read_csv('/content/Walking 1.csv')
df_walking1.head()
df_walking1.drop(['Time (s)'],axis=1,inplace=True)
dataset_walk1 = np.array(df_walking1)
x_dataset_walk1=dataset_walk1[1000:-1000,:3]

df_walking2 = pd.read_csv('/content/Walking 2.csv')
df_walking2.head()
df_walking2.drop(['Time (s)'],axis=1,inplace=True)
dataset_walk2 = np.array(df_walking2)
x_dataset_walk2=dataset_walk2[1000:-1000,:3]

#running
df_run1 = pd.read_csv('/content/Running 1.csv')
df_run1.head()
```

Partial results:

At 2: passed

At : 3 Passed

At 4: Passed

Testcase result: passed