

# SensEdge - Sensor-based Multi Activity Annotator

**Course:** Mobile and Pervasive Computing (CSL7460)

**Instructor:** Dr. Suchetana Chakraborty

**Created By:**

Ghelani Shubham Bhaveshbhai (B20EE019)

Gohel Mruganshi Jatinbhai (B20CS014)

## Introduction:

SensEdge is a sensors-based multi-activity annotator app designed to record accelerometer and gyroscope data in real-time while performing various activities. The app allows users to add multiple activities dynamically, record the sensor data while performing the activity, and annotate it with the corresponding activity name. A CSV file that contains all the recorded data is created, which can be visualized in a graph at any time. The app provides a home display that shows GPS, accelerometer, and gyroscope data for all three axes. With features like Dark Mode and Light Mode, the app has a fantastic and immersive user interface.

## Motivation:

The motivation behind developing this app is the increasing demand for sensor-based data collection in various fields such as sports, health, and research. However, the process of collecting and annotating this data manually is time-consuming and prone to errors. This collected and annotated data has huge application impacts in various field. In research field, the app can be used to collect and analyze data based on human activities and movements, which can be useful in fields such as various technology gadgets, therapy and smart home technology. The ability to record and annotate data from multiple activities and sensors simultaneously can provide valuable insights into how different movements and activities affect the body. In real life, the app can be used by users who are interested fitness freak and aims to keep the track over their daily activites and movements. For example, athletes can use the app to monitor their performance during training sessions and practice matches, while fitness enthusiasts can use it to

track their workouts in gym and progress of their workout. The GPS data fetched from the app can also be used to track outdoor activities such as walking, running, sitting etc based upon the change in position and other sensors data. Moreover, the app has many applications in healthcare. Doctors and other health researchers can use this data to predict the disease of a patient. Thus it can help in diagnosis and can also be used for treatment by tracking the daily activities of the patient and thus providing the right treatment. The data can also serve as training dataset for various Machine Learning Models. With this motivation, we created SensEdge which is a modern solution for solving this data collection and annotation problem and automate it with higher accuracy in data.

## Methodology:

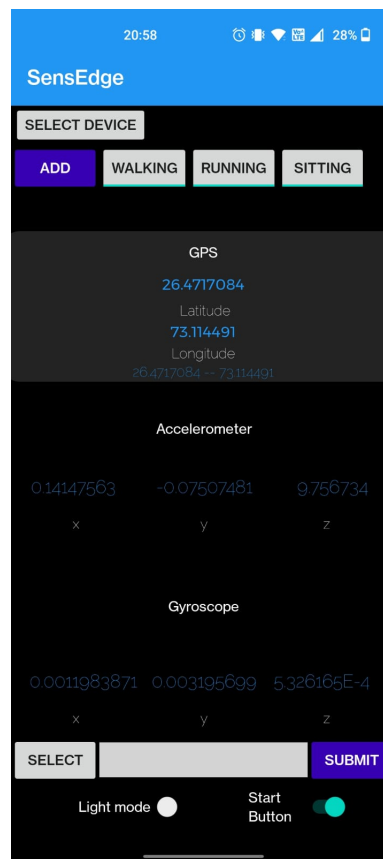
We have implemented various features in the app with systematic methodology. The methodology of the app involves the use of Android Studio, Java programming language, and Android sensors API to record the accelerometer and gyroscope data. The app uses CSVWriter library to write the data to a CSV file. The visualization of the data is achieved using the MPAndroidChart library. Various other Android Library and functions are also used. The features include Immersive and User-Friendly UI Design, Dynamically adding the activity buttons, Fetching the Sensors Data, Annotating it and Storing it as CSV, Selecting the CSV file for Visualization, Visualizing the Data, Adding a share button for plots etc.

### 1) UI Design:

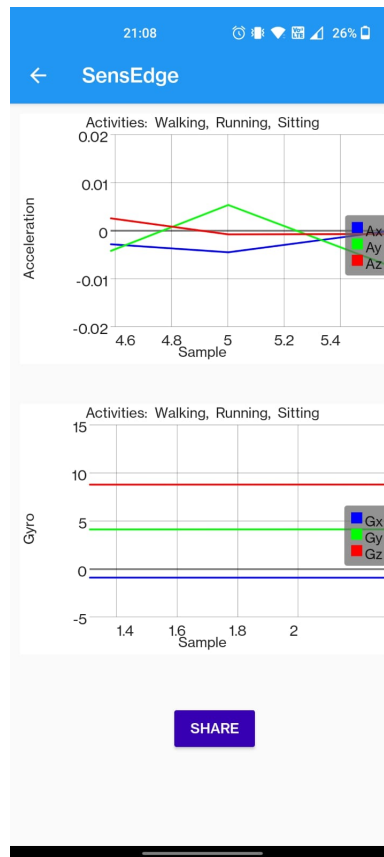
The app is created with a Immersive and User-Friendly UI which help the user to easily use the app. There are two main activity in the app. One is the Main Activity and second is Graph Activity. The Main Activity serves as the homepage and has all the major portion of the features such as Add Button, Sensors Data Display Area, Selecting the CSV Files, Light and Dark Mode etc. While the Graph Activity includes Visualization of the data in form of graphs. There are two graph areas in the activity and they are for two sensors data i.e. Accelerometer and Gyroscope. It also has a “Share” Button for sharing the graphs.

We first have “Add” button by which a user can dynamically add the activities he/she wants. Add buttons and all the dynamically added buttons will be viewed in a Horizontal Scroll View. Then comes the main part of the app which is the area for displaying the Sensors data. We have 3 card views each of them for GPS Data, Accelerometer Data and Gyroscope Data respectively. All three axes data is shown. User then can see a “Select” button for selecting the CSV file he wants to visualize.

There is a text box beside it which displays the name of the selected file. There is a submit button beside the text box which will take us to another activity i.e. Graph Activity. Upon selecting atleast one activity button, a “Start Button” will appear on the bottom right for recording the sensor data. There is also a feature of switching between the Dark Mode and Light Mode.



The another activity is Graph Activity. It consists of two graph areas which depicts the visualization of the selected CSV file. First graph area is for Accelerometer Data and another one is for Gyroscope Data. The activities are annotated on both of them. There is also a share button which is used to share the snapshots of the graphs.



## 2) Dynamically adding the activity buttons

With this feature, a user can dynamically add as many activities as he/she wants. The app has Add Button. When the “Add” button is clicked, a Builder Dialog Box appears and user can enter the name of the activity and click on “Ok”. A new toggle button will be created and it will be added at the last of the Horizontal Scroll View. This whole feature is implemented through Horizontal Scroll view and Linear Layout. Horizontal ScrollView is a FrameLayout, used to provide the child View element horizontal scrolling property.

## 3) Fetching the Sensors Data, Annotating it, and Storing it as CSV

The main purpose of the app is to record the sensor data of the phone, annotate it with the activity selected, and store it in the form of a CSV file which can later be used for various purposes. When a user selects atleast one activity button from the above scroll view, a “Start Button” will get activated at the bottom right corner. Once the activities are selected by the user, he/she can press the recording button and start the recording. The user can stop the recording whenever he wants and the CSV file with the annotated data is stored as a CSV file in the “Default Downloads” folder of phone storage.

To fetch sensor data in the app, we have used the Android Sensor API. Specifically, we used the “SensorManager” class to get the default sensors (accelerometer and gyroscope) and then register listeners to receive updates from these sensors. We first got an instance of the “SensorManager” by calling “getSystemService(Context.SENSOR\_SERVICE)”. Then, we got the default sensors for the accelerometer and gyroscope by calling “getDefaultSensor()” on the “SensorManager” object. Once we obtained the sensors, we use registered listeners to receive updates from them. When the sensor started to generate data, the onSensorChanged() method of the listener object was called. This method receives a SensorEvent object, which contained the sensor data. In conclusion, we had 3 main functions for starting the recording, fetching the data, and pausing the recording. The three functions are “onResume()”, “onSensorChanged()” and “onPause()” respectively. The snapshot of generated CSV is as below:

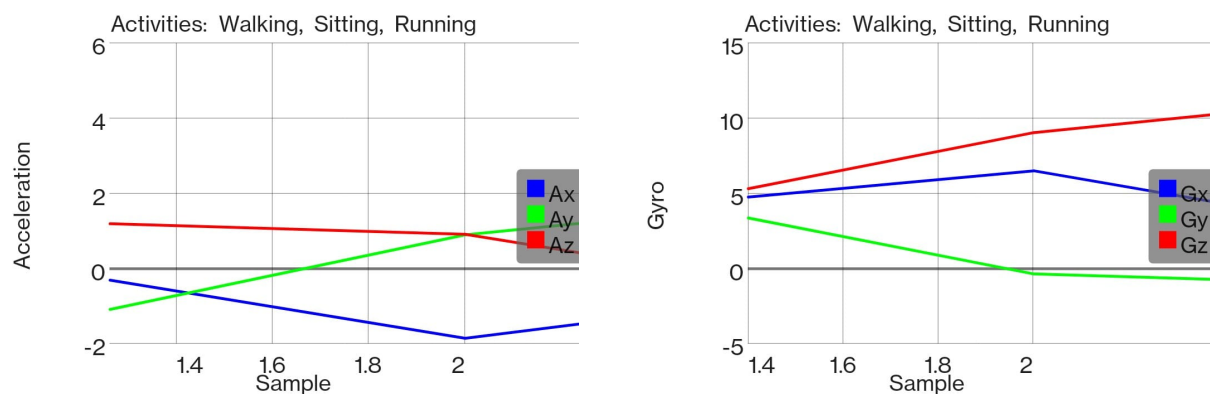
	A	B	C	D	E	F	G	H	I	J	K
1	AX	AY	AZ	GX	GY	GZ	Activity1	Activity2	Activity3	Timestamp	
2	-0.00426	-0.0024	-0.00439	0.168395	6.001797	7.762616	Running	Walking	Sitting	11/05/2023 02:32:38	
3	0.026764	0.014647	0.006525	0.113061	6.028717	7.650452	Running	Walking	Sitting	11/05/2023 02:32:39	
4	0.004527	-0.00652	-5.33E-04	0.109173	6.027819	7.637292	Running	Walking	Sitting	11/05/2023 02:32:39	
5	-0.00866	-0.00692	-0.0016	0.144168	6.009574	7.701001	Running	Walking	Sitting	11/05/2023 02:32:39	
6	0.004261	0.014647	-0.00133	0.116351	6.002097	7.682456	Running	Walking	Sitting	11/05/2023 02:32:40	
7	0.010652	0.010519	-3.99E-04	0.118744	6.005087	7.602297	Running	Walking	Sitting	11/05/2023 02:32:40	
8	-0.01079	-0.01012	-0.00679	0.100499	6.0009	7.660622	Running	Walking	Sitting	11/05/2023 02:32:40	
9	-0.00133	0.007856	2.66E-04	0.099601	6.026025	7.536195	Running	Walking	Sitting	11/05/2023 02:32:41	
10	-0.00413	0.019307	-0.00266	0.133998	5.994021	7.668398	Running	Walking	Sitting	11/05/2023 02:32:41	
11	-0.00453	-0.00386	1.33E-04	0.186939	6.200103	7.559824	Running	Walking	Sitting	11/05/2023 02:32:41	
12	-0.03941	0.397865	0.019574	-1.04985	5.414958	8.001898	Running	Walking	Sitting	11/05/2023 02:32:42	
13	0.018508	-0.00759	0.017044	0.943669	5.689534	6.355636	Running	Walking	Sitting	11/05/2023 02:32:42	
14	-1.43713	-1.14832	0.411979	0.959522	4.795516	8.247163	Running	Walking	Sitting	11/05/2023 02:32:42	
15	0.316774	0.493869	-0.12769	-0.11007	5.966503	7.016654	Running	Walking	Sitting	11/05/2023 02:32:43	
16	-0.2747	0.020373	-0.02064	0.145065	6.656633	8.116454	Running	Walking	Sitting	11/05/2023 02:32:43	
17	0.106523	0.011185	-0.03276	0.152542	5.631508	7.765607	Running	Walking	Sitting	11/05/2023 02:32:43	
18	-0.02583	-0.0213	0.01265	0.156132	5.851049	7.738987	Running	Walking	Sitting	11/05/2023 02:32:44	
19	-0.01851	0.012783	0.006658	0.104387	5.77747	7.829914	Running	Walking	Sitting	11/05/2023 02:32:44	
20	-0.02024	0.009454	-0.00866	0.397508	5.62433	7.712367	Running	Walking	Sitting	11/05/2023 02:32:44	
21	-0.00426	-0.05792	0.006125	0.326022	5.825327	7.880163	Running	Walking	Sitting	11/05/2023 02:32:45	
22	0.029427	-0.04674	-0.00905	0.39661	5.738886	7.974381	Running	Walking	Sitting	11/05/2023 02:32:45	
23	0.030226	-0.0036	-0.00533	0.363111	5.801997	7.864311	Running	Walking	Sitting	11/05/2023 02:32:45	
24	0.015978	-0.03542	-0.00293	0.4002	5.828318	7.795816	Running	Walking	Sitting	11/05/2023 02:32:46	
25	0.005592	0.001997	-0.00266	0.404088	5.801698	7.845168	Running	Walking	Sitting	11/05/2023 02:32:46	

## 4) Selecting the CSV file for Visualization

Once the CSV file is created and stored in phone storage, the user can select the desired CSV file using the “Select” Button. As soon as the “Select” button is pressed, System UI Picker appears for selecting the file from the phone storage. This was done using the concept of Intent() using which we can shift from one activity to another or from one app to another. Once the user selects the file, he will return to the app and the name of the selected file will be shown in the Text box present beside the “Select” button. It is important to note that the path of the CSV file is also stored in a variable in order to fetch the file for visualization. This path will be sent to other activity using the Intent() function. Upon pressing the submit button, a new activity i.e. Graph Activity will get created and the user will be directed there using the Intent() function.

## 5) Visualizing the Data

After clicking the submit button, the user will be directed to new activity that is created (i.e. Graph Activity). The graph activity has 2 graph areas for Accelerometer and Gyroscope Data each. The graph areas are implemented using “MPAndroidChart” Library of the Android Studio. The data of all three axes are visualized a single graph in order to get better comparison between the data. The file path is fetched using the Intent() function parameters from the previous activity. Using the path, the CSV file is opened using CSVReader(). Before plotting the graph, we first also need to remove some unwanted rows that may generated due to synchronization errors. After errors are removed, we can plot the graphs. The graphs, along with the annotations will look like



## 6) Share Button

A share button is also featured in the app. The user can use this share button to share the snapshots of the graphs that have been plotted via any social media app. When the share button is clicked, the snapshots of the graph will be taken. They will be converted in the form of BitMap and then stored in cache memory of the phone. Using the Intent() function, user will be directed to other social media app (Let's say Whatsapp). The path of the Bitmap stored will retrieve the Image and then the image can be send to others using Whatsapp.

## Future Works

An option of “Select Device” is added in homepage using which a user can choose the device from which he wants to fetch the sensor data. Wearables like Smartwatch, Earables etc can be used for fetching the sensor data and annotation and visualization part can be done for it. This has been left for future work.

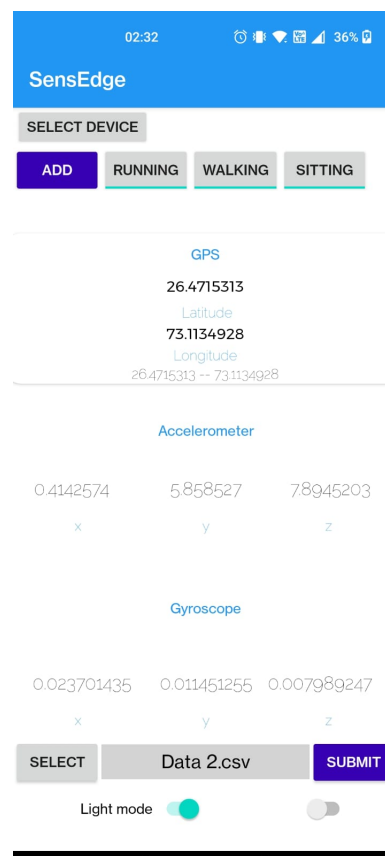
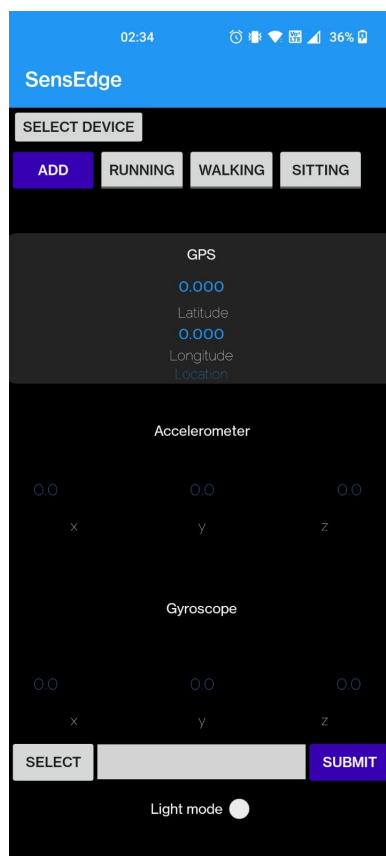
# Results

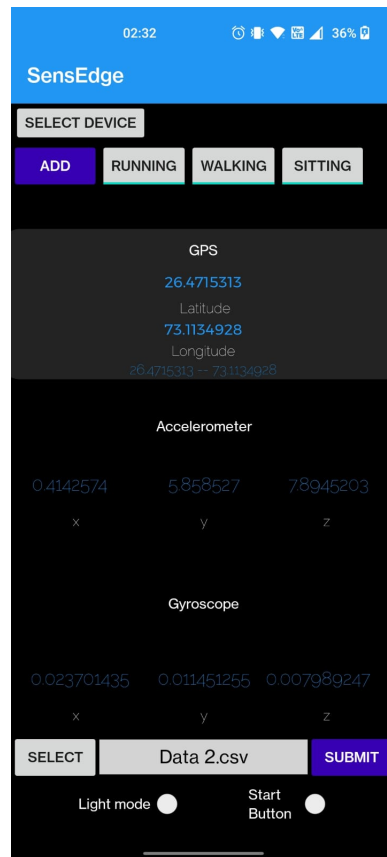
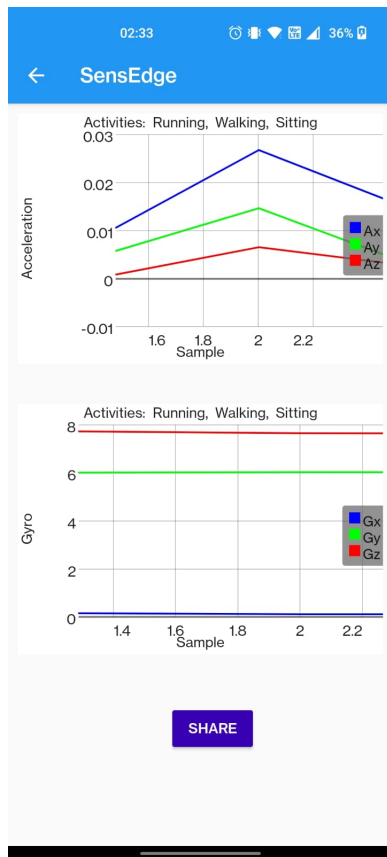
The app successfully achieved the goals of dynamically adding activities, fetching sensor data, annotating and storing data as CSV, and visualizing the data. The UI of the app is user-friendly and easy to navigate.

The app was tested on various devices, and it worked smoothly without any crashes or major issues. The sensor data was accurately recorded and annotated. The CSV files were successfully stored and retrieved. The data visualization feature also worked efficiently and provided clear graphs for accelerometer and gyroscope data.

Furthermore, the implementation of the share button for plots has been successful. This feature allows the user to share the graph with others through different platforms. The addition of the GPS data on the home screen has also been successful.

## Screenshots





## Conclusion

The SensEdge app provides an easy-to-use platform for users to dynamically add activities and record sensor data using the accelerometer and gyroscope sensors. The app allows the user to store the annotated sensor data in CSV format and visualize it in the form of two graphs, one for accelerometer and the other for gyroscope, making it easier for the user to analyze the data.

Overall, SensEdge provides a user-friendly and intuitive interface that enables users to record, annotate, store, and analyze sensor data in a streamlined and efficient way.

In conclusion, SensEdge is a promising tool for researchers and developers who want to collect, annotate, and analyze sensor data in a simple and efficient way. The app can be extended to incorporate other wearable devices, making it more versatile and adaptable to different research and development needs.