



ENGG 6400: MOBILE DEVICE APP DEVELOPMENT

PROJECT REPORT

Group - 6

PROJECT TITLE: Healthifi Me

STUDENT NAME	STUDENT ID
PARVA GURAV	1256570
ASHWIN EMMATTY	1255648

DATE OF SUBMISSION: 27/11/2023

TOTAL WORDS: 2318

SUPERVISED BY:

Prof., Petros Spachos (SCHOOL OF ENGINEERING)

I. INTRODUCTION

In our contemporary era dominated by technology, cultivating a healthy and active lifestyle is paramount for individuals seeking holistic well-being. Recognizing this imperative, we proudly introduce "Healthifi Me" – a groundbreaking Android application poised to revolutionize conventional perceptions of fitness and well-being. Our motivation stems from a profound understanding of the significance of incorporating regular exercise into the daily lives of users.

The genesis of Healthifi Me is rooted in the aspiration to provide users with more than just a fitness app; it aspires to be a personalized, all-encompassing platform for managing overall health. The driving force behind this project lies in the burgeoning demand for a singular, user-friendly interface that seamlessly integrates various facets of fitness, including tailored exercise routines and dietary analysis, into a cohesive and comprehensive solution. Through its intelligent goal-setting mechanisms, real-time tracking capabilities, and the creation of a supportive social network, Healthifi Me endeavors to instill in users the confidence to actively and decisively take control of their well-being.

a) Why and What is Required for Healthifi Me?

The goal of Healthifi Me's conception was to offer customers a personalized, seamless platform for managing their overall health. This project is driven by the increasing need for a single, easily navigable interface that unifies several aspects of fitness, such as exercise regimens and dietary analysis, into a single, comprehensive solution. With its intelligent goal-setting, real-time tracking, and encouraging social network, the app aims to give users the confidence to take control of their well-being.

II. OVERALL DESIGN

a) Block Diagram

The overall design of Healthifi Me is visualized through a comprehensive block diagram that encapsulates the major components of the application.

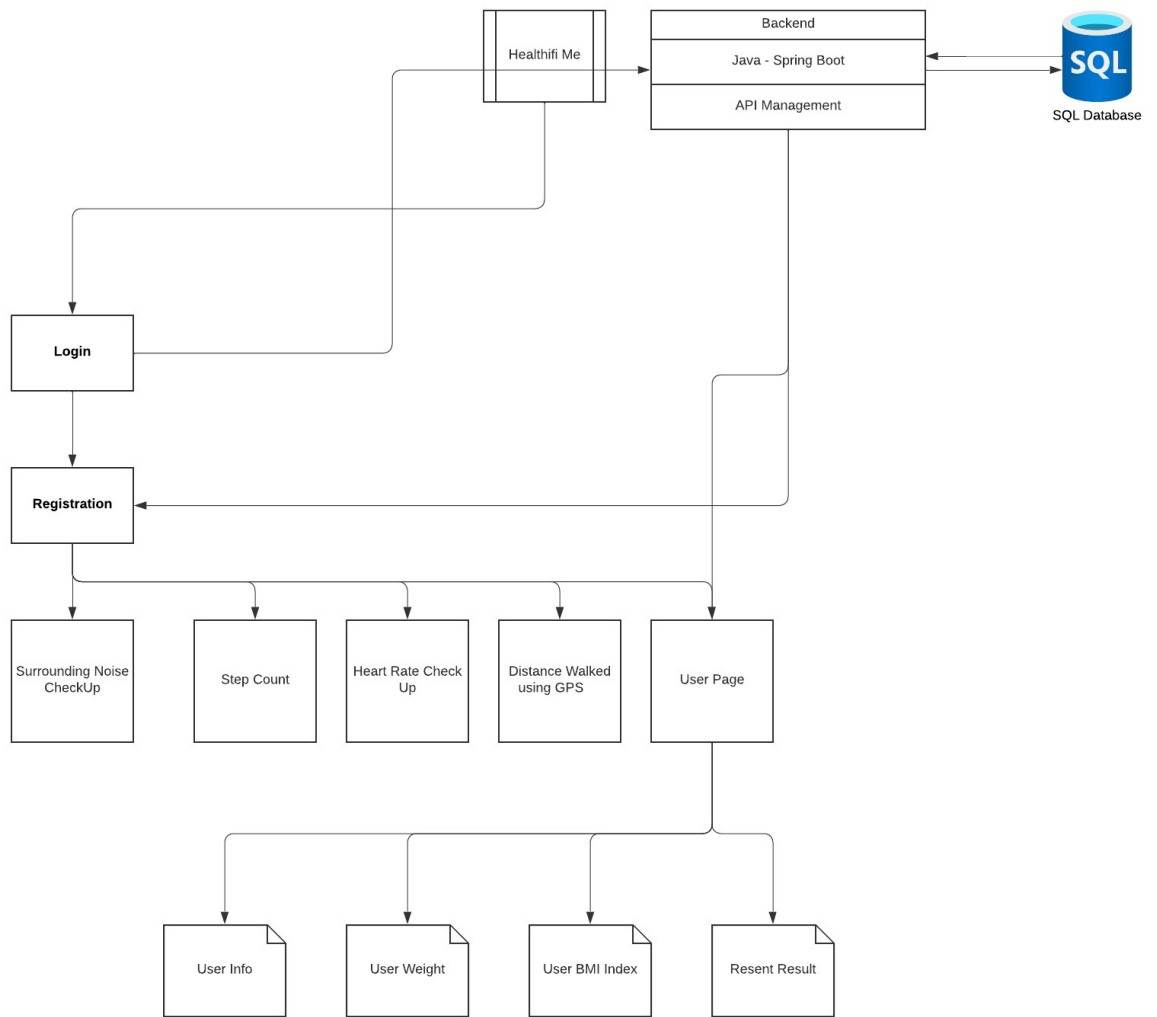


Fig – 2.1 Block Diagram of Healthifi Me

So, the core, Healthifi Me is the root of the app where the application runs, starting with Login/Registration then it lands you on the Surrounding Noise Detection page

1. Surrounding Noise Detection Page: This page will record the environment audio and will give the value in decibels.
2. Step Count: This page will have a timer and a step counter, which will start after clicking on a start tracking button and will calculate how many steps do you walk.
3. 3. Heart Rate Checkup: This contains one button that will redirect you to the Google Fit app where you can have a heart rate checkup using your phone's camera sensor.
4. Distance Walked using GPS: This page will give you your real-time location and once you click on a button, it will track you distance that you have covered.
5. User Pages contain a List view which has 4 major functionalities and 1 Logout Feature:

- a. User Info: This page will contain the details of user
- b. User Weight: This page will take the input from user and will give a bar chart of last 7 days of weight entered.
- c. User BMI Index: This page will take the entered weight and will give the BMI Index to user and gives a suggestion
- d. Recent Result: This page is under development; it will provide a report of all the features used in a pdf form.

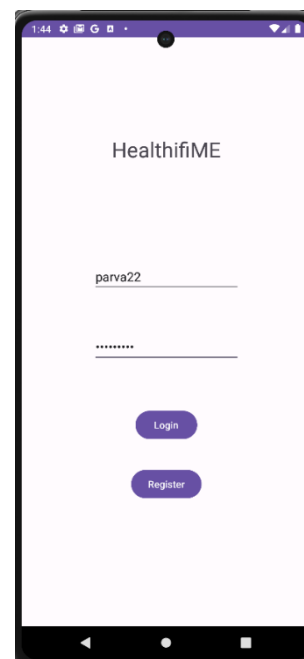
III. FUNCTIONALITY OF APPLICATION

FRONTEND (MOBILE APPLICATION)

1. Login Screen:

The login screen features two input fields, namely, username and password, accompanied by two functional buttons: the "Login" button and the "Registration" button.

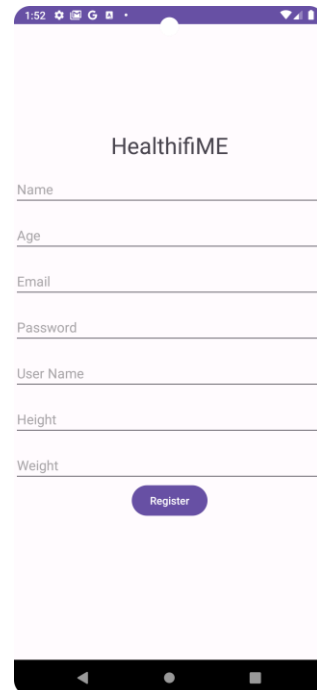
Upon clicking the "Login" button, the application triggers the login API, initiating the authentication process. The response from this API includes the user's details, which are then stored in shared preferences for seamless retrieval. Following successful authentication, users are directed to the Home page. In the event that the "Registration" button is selected, users are seamlessly redirected to the registration page, streamlining the user experience.



2. Registration Page:

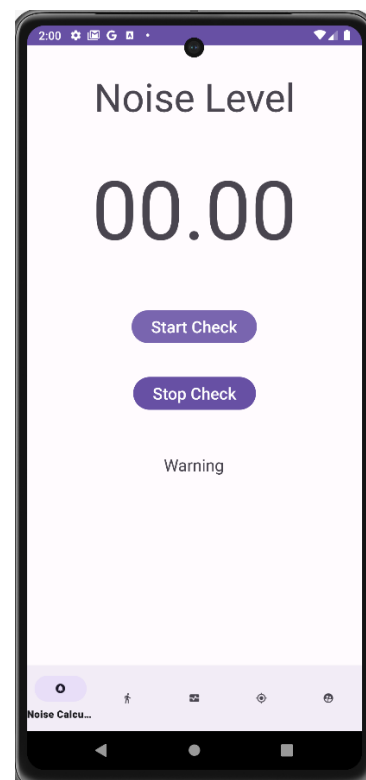
The registration page serves as a valuable resource for new users, prompting them to provide essential information such as Name, Age, Email, Password, User Name, Height, and Weight.

Upon completion of the requisite details, users can initiate the registration process by clicking on the "Registration" button. This action triggers the Registration API, facilitating the creation of a new user profile. Subsequently, the user's details are securely stored in the MySQL Database, ensuring the establishment of a comprehensive and accurate user database.

A screenshot of a mobile application's registration page. The page has a light purple background. At the top, the status bar shows the time 1:52 and various icons. Below the status bar, the app title "HealthifiME" is centered. Below the title, there are seven input fields with labels: "Name", "Age", "Email", "Password", "User Name", "Height", and "Weight". Each field has a thin purple border. At the bottom of the form, there is a purple button with the text "Register" in white. The bottom of the screen shows the Android navigation bar.

3. Surrounding Noise Detection Page:

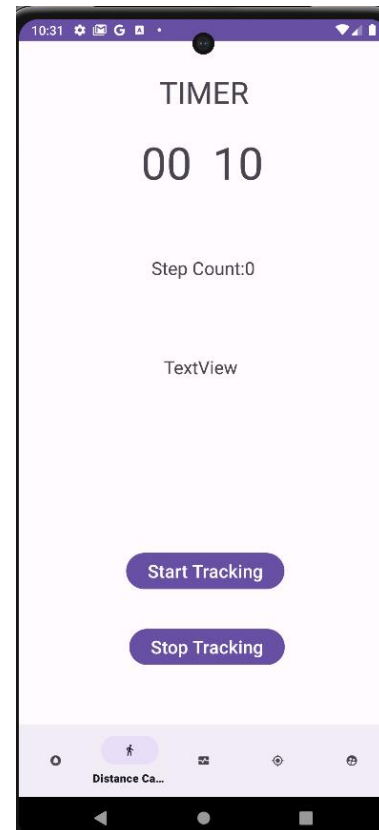
Using the device's microphone, our Android app records audio in real-time, analysing noise levels through the Audio Record class. After obtaining the necessary permissions, the app employs algorithms for amplitude analysis, averaging values or calculating decibels. Users receive real-time feedback on ambient noise, displayed as a meter or numerical representation. Optional threshold settings notify users when noise surpasses predefined levels. Proper handling of the audio recording lifecycle, including starting, stopping, error correction, and resource release, ensures optimal outcomes. Testing and potential calibration are essential due to variables like microphone quality and ambient conditions affecting accuracy ^[1].



4. Step Count:

The accelerometer, a sensor in mobile devices, measures acceleration in three dimensions (X, Y, Z axes) to detect device motion. Utilizing Android SDK's sensor APIs, developers access accelerometer data through the SensorManager and Sensor classes. By registering a SensorEventListener, apps can respond to device tilt, shake, or movement.

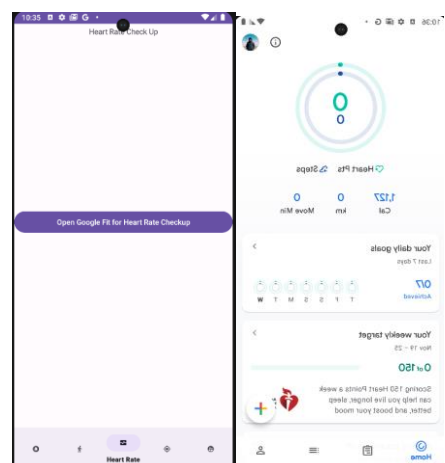
Calibrated and programmed accurately, the accelerometer interprets motion data to identify user steps. While smartphone accelerometers vary in accuracy, they can function as step-tracking devices. Steps produce a distinctive acceleration signal, recognized by oscillating peaks and valleys in accelerometer data [2].



5. Heart Rate Check Up:

The Heart Rate Checkup feature in HealthiFi Me seamlessly integrates with the Google Fit Application. This functionality involves sophisticated image processing and signal processing techniques.

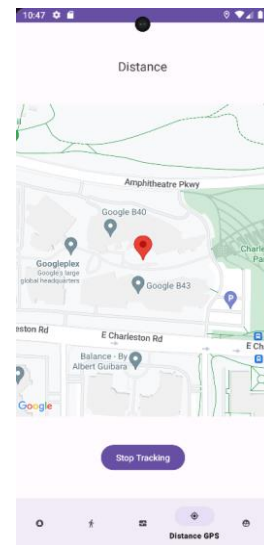
To conduct a heart rate checkup, users are prompted to place their index finger over the rear camera for a duration of one minute. During this time, the application's sensors employ advanced image and signal processing to detect pulses accurately. The detailed results and pertinent information regarding the user's heart rate are then provided, offering a comprehensive and precise assessment of their cardiovascular health.



6. Distance Calculation Using GPS:

Android Studio integrates GPS functionalities through Location APIs, enabling precise location tracking and enhancing user experiences.

- **Location Services:**
Utilizing Location Manager and FusedLocationProviderClient classes is crucial for accessing location services. The latter combines GPS, Wi-Fi, and cellular networks for improved performance, lower battery usage, and enhanced accuracy.
- **Permission Handling:**
Devices running Android 6.0 and above require specified permissions in the AndroidManifest.xml file, requested at runtime for GPS access.
- **Location Providers:**
Various providers, including GPS and network-based locations, cater to different accuracy needs. Developers choose based on device capabilities.
- **Location Updates:**
Apps can request periodic location updates to monitor user movements, specifying update interval, precision, and power conservation priorities [3].



GPS in Running Trackers:

GPS has revolutionized running tracking, offering runners:

- **Accurate Distance Tracking:**
Precise measurement of distances covered during a run, providing real-time data on pace and total distance.

- **Route Mapping:**
Mapping the runner's route on a map interface, aiding visualization and exploration of new routes.
- **Data Logging:**
Detailed historical data logging enables progress tracking, goal setting, and adjustment of training routines based on past performances.

7. User Page

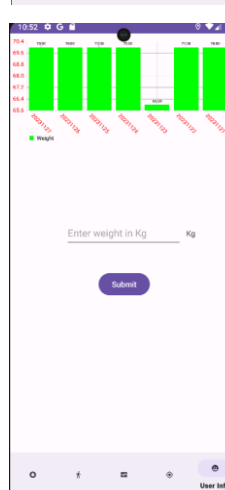
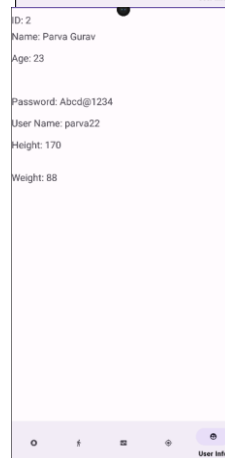
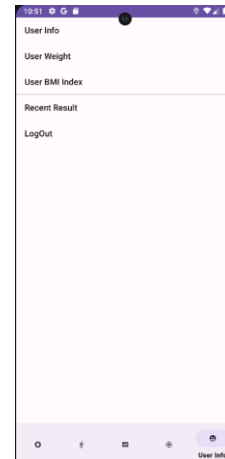
User Detail's home page provides a list of options to navigate. It provides:

- User Info Page
- User Weight page
- User BMI Index page
- Resent Result Report
- Logut Function

The User Info Page operates in conjunction with the unique user_id stored upon user login. Leveraging this identifier, the page calls an API designed to retrieve and present the user's comprehensive details. Through this seamless integration, users gain immediate access to a full spectrum of personalized information, enhancing their overall experience within the application.

The User Weight Page is designed to receive user input for their weight, initiating a subsequent call to a POST method API for data update.

Upon successful update, the page dynamically showcases the user's weight data over the past seven days, visually presented through an informative bar chart. This feature offers users a comprehensive and visually intuitive overview of their weight history, aiding in



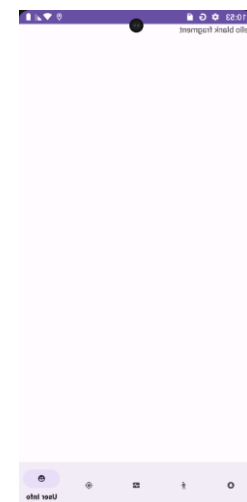
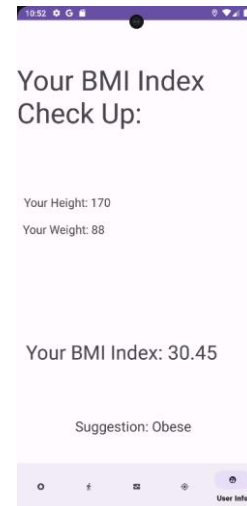
tracking and monitoring their progress effectively.

The User BMI Index Page seamlessly integrates with an API, retrieving height and weight data to perform real-time mathematical calculations within the Android platform. The result is a precise BMI Index, presented to the user.

In addition to displaying the BMI Index, the page offers personalized suggestions based on the calculated BMI. This feature not only provides users with valuable health insights but also encourages informed decisions regarding lifestyle and wellness, contributing to a holistic approach to health management.

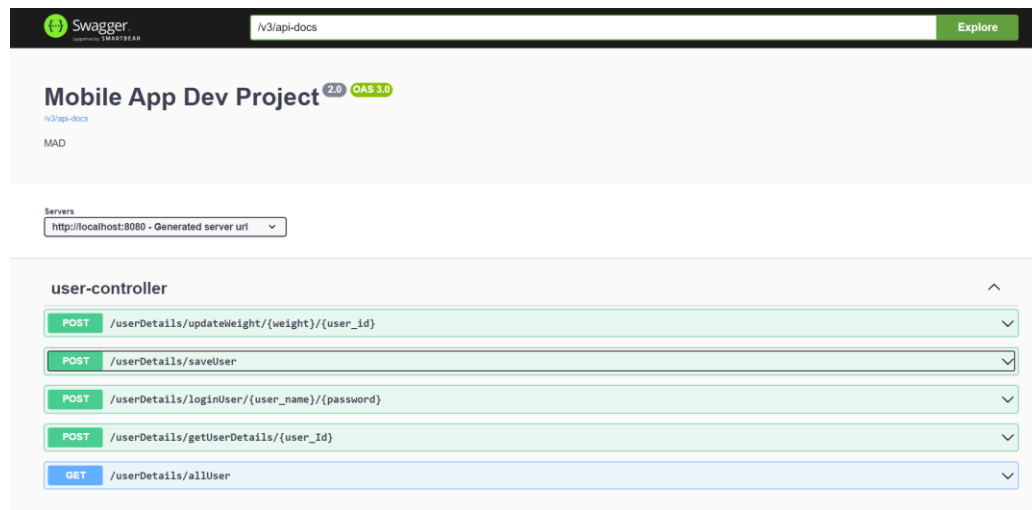
The Recent Result Page, currently in development, aims to consolidate comprehensive user data, including details such as surrounding noise, distance walked, steps counted, and heart rate measurements. Additionally, it will encompass information on weight and BMI Index.

As a user-centric feature, the page is designed to generate a PDF report that encapsulates all relevant health and fitness metrics. This report serves as a user-friendly tool, offering a clear and accessible overview of the individual's overall health progress and data trends.



BACKEND IMPLEMENTATION

Backend of Android is implemented in Java – Spring Framework, where we have created multiple API for storing and retrieving data from database.



The presented image showcases the UI version of our backend, aptly named "Swagger," serving as a robust documentation and API management tool for our structured backend system [4].

Outlined below are the key APIs encapsulated within our backend:

- **updateWeight:** This API facilitates the updating of a user's weight. Upon receiving the user's ID for authentication, the system verifies the request. Once authenticated, the API accepts the provided weight and updates the user's weight in our database.
- **saveUser:** Dedicated to the registration process, the "saveUser" API creates a new user record and assigns a unique user ID. This user ID, generated using a UUID method with 16 digits and characters, serves as a verification mechanism, ensuring the user's authenticity.
- **loginUser:** The "loginUser" API is designed for user authentication. By taking the user's username and password as inputs, the system verifies the user's credentials. Upon successful verification, the data stored during the "saveUser" API process is retrieved and provided as a response.
- **getDetails:** Utilized in the UserInfo page, the "getDetails" API retrieves and displays all pertinent user details. By taking the user's ID as input, the API responds with a comprehensive user model, offering a snapshot of the user's stored information.

These APIs collectively form the backbone of our backend infrastructure, enabling seamless interaction with user data and ensuring a secure and efficient user experience.

IV. What Did You Learn - What Would You Do Differently?

Lessons Learned:

Creating Healthifi Me has been a very educational experience. The relevance of real-time feedback in fitness apps, careful treatment of user data, and accurate sensor calibration are among the important lessons learned. The necessity of smooth cross-app interactions for thorough health tracking was brought to light by the integration of many APIs, most notably Google Fit.

Areas for Improvement:

In retrospect, it could have been advantageous to use a more iterative testing strategy to spot and resolve such issues early in the development cycle. Furthermore, it would be beneficial to increase analytics integration and user feedback systems in order to obtain insights for ongoing improvement. Integrating Heart Rate Checkup feature's intricate picture and signal processing highlighted the necessity for sophisticated algorithms and possible cooperation with medical professionals in order to achieve accuracy.

V. FUTURE WORK

- **Enhanced Noise Detection:**

Introduce advanced features by integrating the light sensor to assess whether the user's phone is covered, enhancing the accuracy and context-awareness of noise level detection.

- **Recent Results Section:**

Develop the "Recent Results" section to consolidate and present comprehensive reports of all user activities, including noise detection, distance walked, steps counted, and heart rate measurements.

- **Refinement of Heart Rate Checkup:**

Further refine the Heart Rate Checkup feature by exploring advanced image and signal processing techniques. Consider collaboration with health experts to ensure accuracy and reliability in providing users with valuable health insights.

- **User Engagement Features:**

Explore gamification elements and challenges to enhance user engagement. Consider features like achievement badges, personalized health challenges, and interactive notifications to keep users motivated and involved.

- Integration with Wearable Devices:

Extend the app's compatibility by integrating with popular wearable devices, allowing users to sync data seamlessly and providing a more holistic health tracking experience.

- AI-Driven Recommendations:

Implement artificial intelligence algorithms to analyze user data and provide personalized health and fitness recommendations. This could include tailored workout plans, dietary suggestions, and lifestyle adjustments.

- Accessibility Features:

Enhance the app's accessibility by incorporating features for users with diverse needs, such as voice commands, larger text options, and compatibility with screen readers, ensuring inclusivity.

VI. CONCLUSION

In conclusion, the ground-breaking Android software "Healthifi Me" aims to completely transform users' perspectives on health and fitness. With its user-centric design, real-time tracking, and flawless integration of many functions, the app gives people the confidence to take charge of their health. We are committed to improving user experience and encouraging a holistic approach to health, which is why we are working to refine the Heart Rate Checkup, add additional capabilities, and improve noise detection in the future.

"Healthifi Me" combines exercise, nutrition analysis, and customized goal-setting to provide a comprehensive vision for a healthy lifestyle. Our dedication to promoting a healthy future is demonstrated by the app's user-friendly design, a wide range of capabilities, and strong backend infrastructure. Our continuous endeavours center on enhancing current features, including sophisticated functionality, and welcoming creativity to furnish consumers with an all-encompassing and captivating health-tracking encounter.

REFERENCES

- [1]. Stack Overflow. (n.d.). "Get the Microphone sound level (Decibel level) in Android." [Online]. Available: [https://stackoverflow.com/questions/7197798/get-the-microphone-sound-level-decibel-level-in-android#:~:text=You%20can%20use%20MediaRecorder.getMaxAmplitude\(\)%20.&text=Get%20the%20noise%20level%20using,mRecorder%20%3D%20new%20MediaRecorder\(\)%3B%20mRecorder](https://stackoverflow.com/questions/7197798/get-the-microphone-sound-level-decibel-level-in-android#:~:text=You%20can%20use%20MediaRecorder.getMaxAmplitude()%20.&text=Get%20the%20noise%20level%20using,mRecorder%20%3D%20new%20MediaRecorder()%3B%20mRecorder).
- [2]. Android Developers. (n.d.). "Motion sensors." [Online]. Available: https://developer.android.com/develop/sensors-and-location/sensors/sensors_motion

[3]. Google for Developers. (n.d.). "Maps SDK for Android Quickstart." [Online]. Available: <https://developers.google.com/maps/documentation/android-sdk/start>

[4]. Baeldung. (n.d.). "Swagger 2 Documentation for Spring REST API." [Online]. Available: <https://www.baeldung.com/swagger-2-documentation-for-spring-rest-api>