

CSE 535 Final Report Group 55

Project 5 : Harmony

Devansh Bindlish Adarsh Shashidhara Manav Gakhar Gagan Nagaraj Chaitanya Sunil Kulkarni Suhas Suresh
dbindlis@asu.edu ashashi2@asu.edu mgakhar@asu.edu gnagara4@asu.edu ckulka11@asu.edu ssures61@asu.edu

Abstract—In today’s rapidly moving and frequently stressful world, prioritizing both physical and mental well-being is essential. Our application addresses these priorities by not only focusing on maintaining heart and mental health but also by providing recommendations such as restaurant suggestions, weather-related advice like monitoring AQI levels, and even guidance on when to wear appropriate clothing like a jacket or carry an umbrella. Additionally, Harmony conveniently summarizes the daily calories burnt based on data collected from various smartwatches.

Index Terms—Guardian Angel, Harmony, Health Application, Health Companion

I. INTRODUCTION

In an era characterized by the swift pace of modern living and the prevalence of stress, the imperative of nurturing physical and mental health has become paramount. Acknowledging this fundamental need, our project explores an innovative application designed to prioritize and bolster these essential facets of wellness. This paper delves into the multifaceted functionalities of our application, which not only focuses on preserving cardiovascular and mental well-being but also offers a comprehensive array of user-centric features.

Central to our application is its dedication to promoting a holistic approach to health, transcending the mere monitoring of vital signs. It extends its utility by furnishing users with invaluable recommendations, encompassing diverse aspects of daily life. From providing tailored restaurant suggestions to offering pertinent weather-related guidance, such as real-time monitoring of Air Quality Index (AQI) levels, or even offering advice on appropriate attire choices like wearing a jacket or carrying an umbrella, the application aims to seamlessly integrate health-conscious decisions into users’ routines.

Moreover, a distinguishing feature of this application, named Harmony, lies in its ability to synthesize and present crucial data about users’ physical activity. By harnessing the data acquired from various smartwatches, Harmony succinctly compiles and presents insights into the daily calories expended, empowering users with a comprehensive overview of their fitness endeavors.

Through this project, we elucidate the novel features and functionalities of Harmony, demonstrating its potential to significantly enhance users’ well-being by amalgamating health management with practical, everyday decision-making.

II. ARCHITECTURE

The application can be segmented into four primary components, each of which corresponds to a distinct facet of its functionality. These components are prominently featured with dedicated buttons on the mobile app’s home page. Data aggregation occurs from diverse origins, including smartwatches, the Google Location API [1][2], and user questionnaires. This amalgamated data is transmitted to our AWS server, wherein a Flask App [3], scripted in Python and leveraging an array of libraries and APIs, processes the information. Subsequently, the outcomes are relayed back to the Mobile App for display. The operational workflow of each component is outlined below:

Methodology

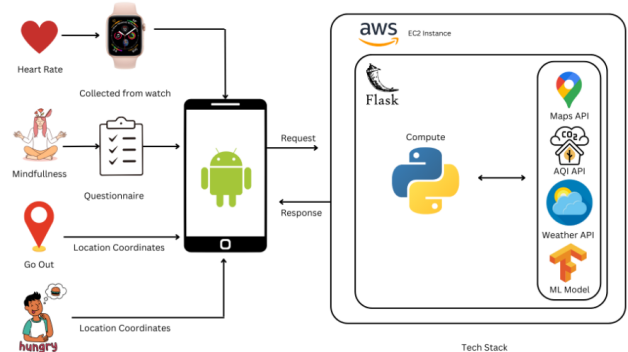


Fig. 1. System Architecture

A. Mindfulness by Chaitanya

This segment is dedicated to the Mental Health module of our application, providing users with a platform to engage with a series of thoughtful inquiries. Powered by an XgBoost Machine Learning Model, meticulously trained on a comprehensive mental health dataset procured from Kaggle, this feature takes a user-centric approach. Upon receiving the user’s responses, the Flask server takes charge, leveraging the model to process the input and generate a meaningful output. The resultant insights are seamlessly transmitted back to the mobile app.

Based on the received output, the application not only provides valuable suggestions for enhancing mental well-being

but also, in certain scenarios, encourages users to consider consulting a mental health specialist for further guidance and support. This holistic approach underscores our commitment to fostering mental health awareness and facilitating meaningful support within the digital realm.

B. Go Out by Suhas

The Component is designed in a modular fashion. It mainly consists of 4 modules. Address Save module, Home, Work and Activities. The Address Save module is used for the user to save the home and work address using shared preferences. These addresses can be updated at any time. The addresses saved in this module can be accessed by any other module by calling the shared preferences class. The Home and Work modules can quickly display the time required for the user to reach home/work from the current location. It also gives useful traffic information.

The module connects to the Google Maps api for information. The Activities module takes the current location and suggests an activity present nearby. The module connects to Google Places API for activities. Some of the activities suggested are from the following categories: aquarium, casino, museum, park, stadium, zoo etc. All the processing is done on the phone itself. The tech stack used here is Android Studio and Kotlin for app development, Google Maps API and Google Places API for providing traffic and activity updates

C. Weather Wise by Gagan

The WeatherWise application embodies a symbiotic relationship with the Guardian Angel Health App, introducing a personalized weather companion that seamlessly integrates real-time forecasts, environmental insights, and tailored safety suggestions. Aligned with Guardian Angel's ethos, WeatherWise enhances environmental awareness and proactive health management by providing users with pertinent weather-specific health precautions. Its functionalities encompass environmental monitoring, offering real-time weather forecasts, air pollution levels, and UV indices. Moreover, it furnishes safety recommendations such as suggesting masks during high pollution or advising sunscreen for UV protection.

The WeatherWise app's structural design, catered for Android devices using Java within Android Studio, underscores user experience through meticulous UI/UX design. Its core functionalities encompass API integration, leveraging Java-based HTTP libraries to interface with the OpenWeather API, real-time data processing to generate personalized safety recommendations, and seamless user location management utilizing Android's location APIs. The architecture prioritizes scalability and cross-platform compatibility, laying the foundation for future enhancements and ensuring optimization for Android devices.

D. Stress Detection by Manav

This specialized module plays a pivotal role in gauging user stress levels through the analysis of ECG signals captured by the smartwatch [4]. Positioned strategically within our

comprehensive pipeline, we harness multiple modalities to gain insights into the user's current physical and mental well-being. The heart rate component intricately processes ECG signals, transforming them into Heart Rate Variability (HRV) metrics, mirroring those found in the SWELL dataset.

Utilizing a Decision Tree classifier, we categorize stress levels into three distinct classifications: "no stress," "time pressure," and "interruption." This classification system is purposefully designed to assist users in navigating the challenges of contemporary, stress-laden working conditions. By providing real-time insights into stress levels [5], we empower users with actionable recommendations tailored to their specific stressors, fostering a healthier and more resilient lifestyle. Our commitment extends beyond mere stress identification – we strive to offer personalized solutions, enhancing overall well-being in the dynamic landscape of daily life.

E. RestoFind by Adarsh

This segment of the Harmony Application is used to cater to users' restaurant discovery requirements. Based on the user's preferences the app determines the right restaurants for the user in locations nearby arranged in order of distance. The App uses the client-server model to its utmost by doing UI and light computation on the Android side i.e. the front end and hefty computation on the back end - server. On the client side, the Android application determines the user's current location based on location API and this along with the user credentials is sent encapsulated onto an HTTP request component in the form of JSON. This request is now sent to the back end flask server via Post requests invoking the required module based on the API endpoint invoked.

The back end server receives the creds and location which are input parameters to the restaurant recommender module in python flask. Using the Yelp Maps api to get restaurants based on location and based on user preference filtering we determine the list of restaurants in order of nearest distance to the user. This list is sent back to the Android app as a response finally displaying the same. All the major tasks like determining the location and the framework of HTTP requests and response is done via background threads, where the computation runs in the background and displays the outputs via callbacks when the result is available.

F. Workout Buddy by Devansh

This innovative feature seamlessly integrates with your workout routine, allowing you to capture and archive your heartbeat data throughout each session. The recorded values, stored with precision, undergo meticulous evaluation within our system. While we employ a pre-recorded CSV [6] file for our demonstrations, real-time implementations can readily incorporate live heart rate data. The data is subjected to rigorous pre-processing, including checks for NaN values, ensuring the integrity of the dataset.

Following this step, a suite of metrics is generated, providing a comprehensive understanding of your workout session. These metrics include the average heartbeat during the

workout, delineation of different heart rate zones experienced throughout the session, and, using a specialized formula, the calculation of total calories burned [7]. Beyond real-time insights, users can delve into historical trends, examining workout patterns over the past month. This visual representation serves as a motivating tool, empowering users to stay focused and maintain their fitness momentum. Once these calculations are complete, the data is seamlessly transmitted to the front end via a JSON response. This ensures a user-friendly display of all pertinent workout information, fostering a holistic approach to fitness tracking and motivation within our application.

III. IMPLEMENTATION

In the implementation phase, our project transitioned from conceptualization to a fully functional application, seamlessly blending health management features with everyday decision-making. The Android application, meticulously crafted for an intuitive user experience, harnessed the platform’s capabilities to incorporate real-time health monitoring, stress assessment, restaurant recommendations, and weather guidance. Background threads optimized responsiveness, ensuring efficiency in computation and smooth user interactions. Simultaneously, the Flask server, acting as the backend powerhouse, orchestrated intricate computations with machine learning models for mental health assessment and restaurant recommendations. The symbiotic interplay between the Android frontend and Flask backend, encapsulated in a client-server model, facilitated seamless data exchange, emphasizing the collaborative essence of our implementation.

A cornerstone of our implementation was the integration of diverse data sources, including smartwatches, for comprehensive physical activity insights. The Harmony application adeptly processed and presented valuable information on daily caloric expenditure, offering users a holistic overview of their fitness endeavors. Throughout this phase, we prioritized the robustness, efficiency, and scalability of the system, ensuring Harmony not only meets current user needs but is poised for future expansion and feature enhancements. Embracing an iterative and collaborative development approach, this phase culminated in the creation of an application seamlessly marrying health-conscious features with the pragmatism of daily life.

IV. DEMONSTRATION

A significant portion of the components within our application is designed to be effortlessly accessible to end-users. Upon successful deployment of both the Android application and Flask server, a myriad of functionalities becomes demonstrable, contingent on the application having internet access and access to the user’s location. In our illustrative demonstration, we’ve employed placeholder data as a representation, but it’s crucial to note that real-time ECG signals from popular smartwatches can seamlessly replace this placeholder data, offering a dynamic and personalized experience.

This application serves as a testament to its multifaceted capabilities, showcasing the user’s ability to comprehensively assess and manage both their physical and mental well-being. Users can gain insights into their mental health status, ascertain stress levels, receive tailored recommendations for recreational activities, discover nearby dining options, and precisely measure their workout metrics. The demonstration underscores the application’s versatility, illustrating its adeptness in providing a holistic understanding of a user’s health and wellness—all encapsulated within the user-friendly interface of an Android application.

As we continue to refine and enhance our application, the potential for real-time ECG signals and seamless integration with popular smartwatches stands as a testament to our commitment to delivering a robust and user-centric health management tool.

REFERENCES

- [1] Google, "Google Maps Platform Places API," Google Developers. [Online]. Available: <https://developers.google.com/maps/documentation/places/web-service>.
- [2] Google, "Google Maps Distance Matrix API," Google Developers. [Online]. Available: <https://developers.google.com/maps/documentation/distance-matrix/overview>.
- [3] Grinberg, Miguel. Flask web development: developing web applications with python. " O'Reilly Media, Inc.", 2018.
- [4] Koldijk, S., Sappelli, M., Verberne, S., Neerincx, M. A., Kraaij, W. (2014, November). The swell knowledge work dataset for stress and user modeling research. In Proceedings of the 16th international conference on multimodal interaction (pp. 291-298).
- [5] Dahal, K., Bogue-Jimenez, B., Doblas, A. (2023). Global Stress Detection Framework Combining a Reduced Set of HRV Features and Random Forest Model. *Sensors*, 23(11), 5220.
- [6] Mampoumbius. (2020, December 16). Fitbit Fitness Tracker Data. Kaggle. <https://www.kaggle.com/datasets/arashnic/fitbit/>.
- [7] Heart rate based calorie burn calculator. Fitness Volt. (2023, August 3). <https://fitnessvolt.com/heart-rate-calorie-burn/>.
- [8] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Matrix factorization and neighbor based algorithms for the Netflix Prize Problem: Proceedings of the 2008 ACM Conference on Recommender Systems," ACM Conferences, 01-Oct-2008. [Online]. Available: <https://dl.acm.org/doi/abs/10.1145/1454008.1454049>. [Accessed: 30-Apr-2023].
- [9] "Explicit matrix factorization: ALS, SGD, and all that jazz," Medium, 11-Aug-2016. [Online]. Available: <https://blog.insightdatascience.com/explicit-matrix-factorization-als-sgd-and-all-that-jazz-b00e4d9b21ea>. [Accessed: 30-Apr-2023].