POLITECNICO DI TORINO

Masters's Degree in Computer Engineering

Masters's Degree Thesis

Optimization and Expansion of a Wellbeing App: Improving User Experience, Data Management, and Health Knowledge Delivery

Supervisors

Prof. Maurizio MORISIO

Candidate

Pablo MUNOZ

SEPTEMBER 2024

Abstract

As life expectancy increases worldwide, a growing number of individuals are living into advanced age. However, this longevity is often accompanied by multiple chronic health conditions, negatively impacting the quality of life for many elderly individuals. The rapid aging of the population, coupled with an increasing prevalence of chronic diseases, underscores the urgent need for tools that not only extend life but also enhance well-being during later years. This thesis focuses on addressing this challenge through the enhancement of an existing wellbeing mobile application aimed at fostering healthy lifestyle habits. The project primarily focuses on two areas: improving the app's User Interface (UI) and User Experience (UX) to create a more intuitive, user-friendly platform, and optimizing the back-end structure to ensure efficient data management and scalability. A redesign of the app's theme and layout was carried out, adopting a modern dark mode interface with enhanced readability and interaction. Key user engagement features, such as knowledge pills and quizzes, were revamped to provide more concise, engaging, and personalized health information to users. Furthermore, the navigation flow was restructured to improve gesture detection and back-navigation functionality, making the app easier and more intuitive to use. Back-end improvements included consolidating how data is stored and retrieved, particularly for activity, sleep, and quiz information. The use of Fitbit as the primary data source improved data consistency, while the restructuring of database elements optimized performance and storage efficiency. These changes also enabled better personalization for users, storing essential data such as health goals and completed knowledge pills. Overall, these modifications aim to create a more robust, scalable, and maintainable platform that supports users in their pursuit of healthier and longer lives.

Keywords— Well being, Mobile app, Enhancement, Longevity

Table of Contents

Acronyms

AI

artificial intelligence

KP

Knowledge Pill

UI

User Interface

UX

User Experience

DB

Data Base

Chapter 1

Introduction

As society and technology advance, human life expectancy and the average age of the global population tend to increase. Over the past century, significant strides in medicine, public health, and living conditions have contributed to a remarkable rise in average life expectancy, coupled with a dramatic reduction in infant and early-age mortality rates [1]. In developed countries, where mortality crises such as diseases, civil unrest, and wars are less frequent, the average life expectancy has grown approximately to 80 years for men and 84.4 years for women [2].

Despite this increase in life expectancy and the growing number of individuals reaching 80 years or more [1], the human body and the modern lifestyle does't seem to contribute to the quality of life on this long-lasting individuals. While more people are living longer, many of them are not living healthier. Only 19 percent of individuals who reach the age of 100 do so without suffering from age-associated diseases, while the remaining 81 percent experience one or more chronic conditions [3]. This highlights a growing concern: as longevity increases, so too does the prevalence of chronic illnesses and the complexity of managing them.

The implications of these trends are significant, not only for the individuals affected but also for society as a whole. The medical and technological advances that have extended life come with enormous societal and financial costs[4]. For instance, in 1950, the chance of survival from the age of 80 to 90 was just 15–16 percent for women and 12 percent for men in most developed countries. By 2002, these figures had improved to 37 percent and 25 percent, respectively. Today, in Japan, the survival rate for women has surpassed 50 percent [1]. However, this increase in survival often comes at the cost of prolonged suffering and a poor quality of life. Many elderly individuals suffer with multiple chronic conditions such as dementia, strokes, and frailty, and depend on a long list of medications, all of which make their lives increasingly painful and complicated.

Looking at all this, we see an urgent need for tools and technologies that not only extend life but also improve the quality of those additional years. Modern medicine often focuses on diagnosing and treating chronic diseases mainly through drugs and surgery, intervening only after significant damage has already occurred. This reactive approach fails to prevent the deterioration that leads to costly and complex treatments. While various health monitoring applications exist, many fall short in providing ongoing, personalized support that empowers long-term behavior change, addressing the root causes of age-related illnesses. These apps often prioritize features that focuses on user engagement over those that contribute meaningfully to better health outcomes.

This thesis aims to address these gaps by enhancing and further developing an existing wellbeing app that provides users with comprehensive support for a healthier lifestyle. The primary focus will be on improving the user interface (UI) and user experience (UX) to create an intuitive, easy-to-navigate platform that delivers essential information on physical and mental health in a clear and engaging way. Additionally, the app will be enhanced with personalized health recommendations, daily information "pills", and interactive quizzes designed to reinforce positive behavior changes. These features will be supported by a robust backend and optimized database structure to ensure efficient data management and personalized user experiences.

Through the careful integration of user-friendly design, personalized guidance, and advanced backend optimization, this project seeks to create a wellbeing app that not only supports healthy aging but also stands out in an increasingly crowded market. By addressing the unique challenges faced by the aging population, this app has the potential to make a meaningful impact on both individual users and society as a whole, ultimately contributing to longer, healthier, and more fulfilling lives.

Chapter 2

Objectives

This thesis aims to significantly enhance and further develop a wellbeing app to provide users with an exceptional mobile experience, while also improving the backend structure for a more efficient data management, and a more comprehensible code for further modifications. The primary objective is to create a user-friendly and intuitive interface that delivers essential information on physical and mental health in a clear and engaging manner.

2.1 Specific Objectives

1. **Enhance the User Interface (UI) and User Experience (UX):** Develop a visually appealing, well-structured, and easy-to-navigate UI that distinguishes the app from existing wellbeing apps. Achieve an intuitive design that allows users to effortlessly find and access relevant information.

2. **Deliver Valuable Insights through Daily Information Pills and Quizzes:**
   - Develop a comprehensive knowledge base that offers users essential information on good practices for maintaining a long and healthy life, supported by relevant sources such as [5].
   - Create daily "information pills" that present bite-sized, valuable content.
   - Design a quiz system to reinforce user understanding and engagement, introducing new questions each week to avoid repetition.

3. **Optimize Backend and Database Structure:** Enhance the backend architecture and improve the database structure to ensure efficient data storage, retrieval, and management. Implement optimized data handling practices to improve performance, scalability, and data integrity within the app.

The proposed enhancements aim to create a wellbeing app that stands out from existing solutions. By prioritizing UI/UX design, increasing user engagement, optimizing backend and database structures, and delivering valuable insights through daily information pills and quizzes, this project will provide a unique value proposition that differentiates the app. Ultimately, this will lead to increased user satisfaction, retention, and overall success.

Chapter 3

Related Works

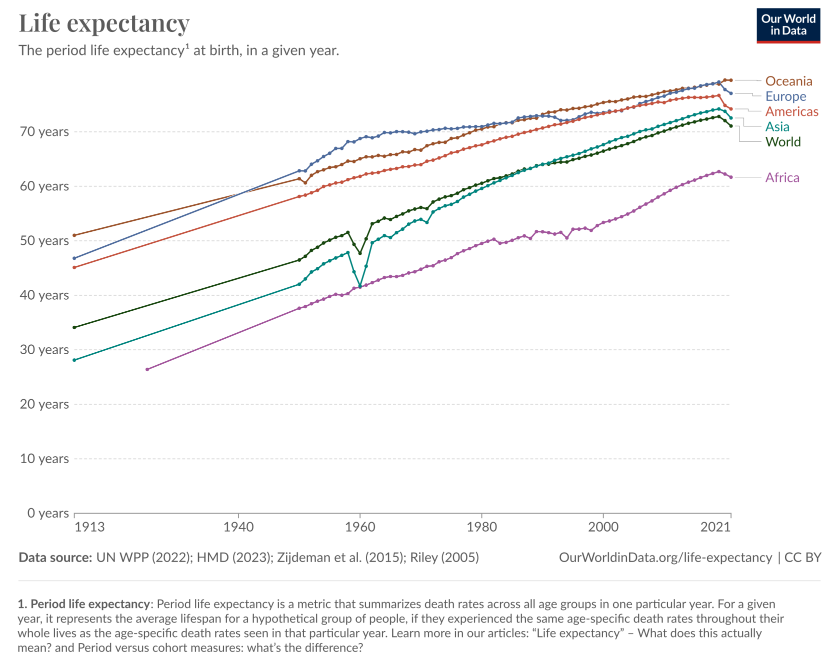

Figure 3.1: Average life expectancy per year, since 1900 [6]

The life expectancy of the global population has been steadily increasing over the past few decades. In 1974 (50 years ago), the average life expectancy was 58.3 years, but by 2021, it had risen to 71 years across both sexes worldwide [6]. As shown in *Figure 3.1*, this upward trend persists across all continents, with only minor deviations in specific years. This increase in life expectancy can largely be attributed to advancements in medical care, significant reductions in infant and early-age mortality, decreased tobacco use, and lower mortality rates from cardiovascular diseases [2, 7]. However, while people are living longer, this does not necessarily equate to a healthier life. Many individuals reaching advanced ages are often burdened with chronic conditions, reliant on medications, or have experienced major health events like strokes, which can diminish their quality of life [5].

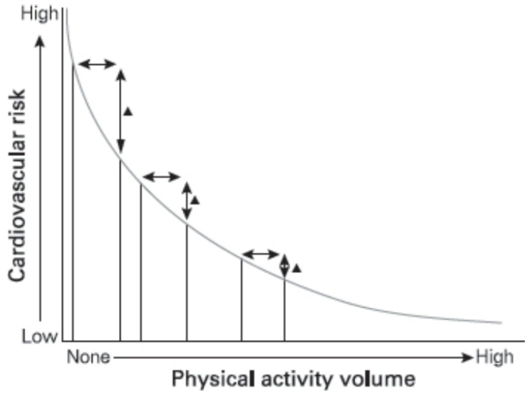

Figure 3.2: Relationship between physical activity and cardiovascular risk. Taken from [8]

Regular physical activity is a key factor in promoting longevity and a healthy lifestyle. A Harvard study demonstrated that moderate to intense physical activity significantly impacts both longevity and overall health [9]. Additionally, various studies have linked high levels of physical activity to reduced risks of numerous diseases, including coronary heart disease, cerebrovascular disease, hypertension, type 2 diabetes, colon cancer, and osteoporosis [8]. In contrast, inactivity has been associated with an increased risk of heart disease [5], as illustrated in *Figure 3.2*. Furthermore, regular physical activity is highly recommended for improving metabolic health, further contributing to overall wellbeing.

Another critical factor contributing to wellbeing and health is obtaining quality rest and sleep, which allows the mind to recover after a day of overstimulation. Mental rest can be just as vital as physical recovery following exercise. Studies indicate that prolonged sleep deprivation can have severe consequences, even in healthy individuals. For instance, research has shown that resting for five hours or less per night over the course of one week can significantly decrease insulin sensitivity [10]. Additionally, prolonged sleep deprivation can lead to death after a few months [5], underscoring the essential role of sleep in overall health. Sleep also provides preventive benefits by reducing cerebral inflammation and improving brain cell function.

A study by [11] compared different age groups, including older adults (80–105 years), younger adults (20–30 years), and middle-aged adults (60–70 years). The findings revealed that older adults (80+) tended to maintain a more consistent and strict sleep schedule, which was linked to better lipid metabolism, potentially contributing to their longevity. This consistency in sleep patterns may play a vital role in maintaining overall health and wellbeing as individuals age.

A person's diet plays a crucial role in their health and longevity. The connection between food, nutrition, and lifespan has been recognized for a long time [5]. Historically, nutritional concerns were mainly associated with nutrient deficiencies and an unbalanced diet. However, in developed countries, these issues have largely been replaced by new challenges. Today, the prevalence of under- and over-nutrition, caused by the consumption of foods high in empty calories but lacking essential nutrients, has led to widespread health problems such as obesity and type-2 diabetes.

A study [12] highlights the profound impact that diet can have on an individual's longevity. The research indicates that, even in cases of caloric deficit, the most significant factor related to health and lifespan is not the quantity of food consumed but the nutritional value of that food. In essence, the quality of the diet—what is eaten rather than how much—is key to promoting longevity and overall well-being.

Furthermore, another study [13] explores the relationship between food composition, caloric intake, and the frequency and duration of fasting, and how these factors collectively influence longevity and health. This research seeks to develop dietary strategies aimed at boosting lifespan, reinforcing the idea that a balanced, nutrient-rich diet is essential for both maintaining good health and enhancing longevity.

Finally, advancements in wearable digital health technology (DHT) have introduced new methods for health monitoring and wellbeing improvement. Wearables, such as smartwatches and sensor patches, play a significant role in tracking physical activity, sleep, and physiological metrics like heart rate and glucose levels. These devices, increasingly integrated into clinical practice, allow real-time monitoring and personalized health interventions. As the adoption of wearable DHT becomes more widespread, these tools empower patients to take an active role in managing their health, reducing the need for frequent in-person visits. The global market for wearable medical devices is projected to reach $76 billion by 2028, highlighting the growing impact of such technologies on health outcomes and the overall healthcare and wellbeing [14].

Mobile health (mHealth) applications, particularly those integrated with wearable sensors and smartphones, have significantly expanded opportunities for health monitoring and promoting both mental and physical wellbeing [15]. These technologies allow for real-time, personalized health data collection, enabling users to track various metrics such as heart rate, physical activity, and stress levels. This data can then be used to provide insights into the user's health status and to promote healthier lifestyle choices. One good example is the BeWell app, which continuously monitors user activity and provides feedback to encourage healthier decisions regarding sleep, physical activity, and social interactions.

In addition to supporting physical health, mHealth apps also focus on mental health through mobile cybertherapy [15]. These apps offer a wide range of tools for mental health monitoring and treatment, including real-time assessments of emotional states using smartphone sensors and biosensors. For example, apps like T2 MoodTracker [16] allow users to self-monitor their emotional experiences over time, while other systems integrate physiological data, such as heart rate variability, to provide deeper insights into emotional and psychological health. These advancements underscore the potential for mHealth apps to empower users in managing their health, both physically and mentally, through continuous engagement and feedback.

Chapter 4

Current Application State

First, we will determine the current state of the app to establish a baseline for the proposed improvements. This evaluation will focus on two core elements of the application: the front-end, and the back-end. These aspects provide a comprehensive view of the app's design, functionality, and technical efficiency.

We will provide an analysis as objective and descriptive as possible, as the possible shortcomings and needed changes/enhancements will be discussed on a further chapter.

4.1 Front-end

The front-end of the app refers to all user-facing elements, including the user interface (UI) and user experience (UX). This section will assess the visual design, navigation, and overall responsiveness of the mobile app. A well-designed front-end is critical to engaging users and ensuring they can easily access and interact with the app's features, which is key to long-term user retention.

4.1.1 User Interface (UI)

The user interface (UI) is the primary point of interaction between humans and a system, allowing users to interact with computers or machines to complete tasks. As end users interact physically, perceptually, and conceptually with the system, it is through the UI that they experience and control these interactions. The primary goal of a well-designed UI is to provide users with an intuitive and simple means to use and navigate the system effectively.

We can see on *Figure* 4.1 the UI of the 3 main tabs of the application, the home page, the learning tab and the health tab. We see that all of them are constructed in a similar way, with a constant background and app bar theme (green with

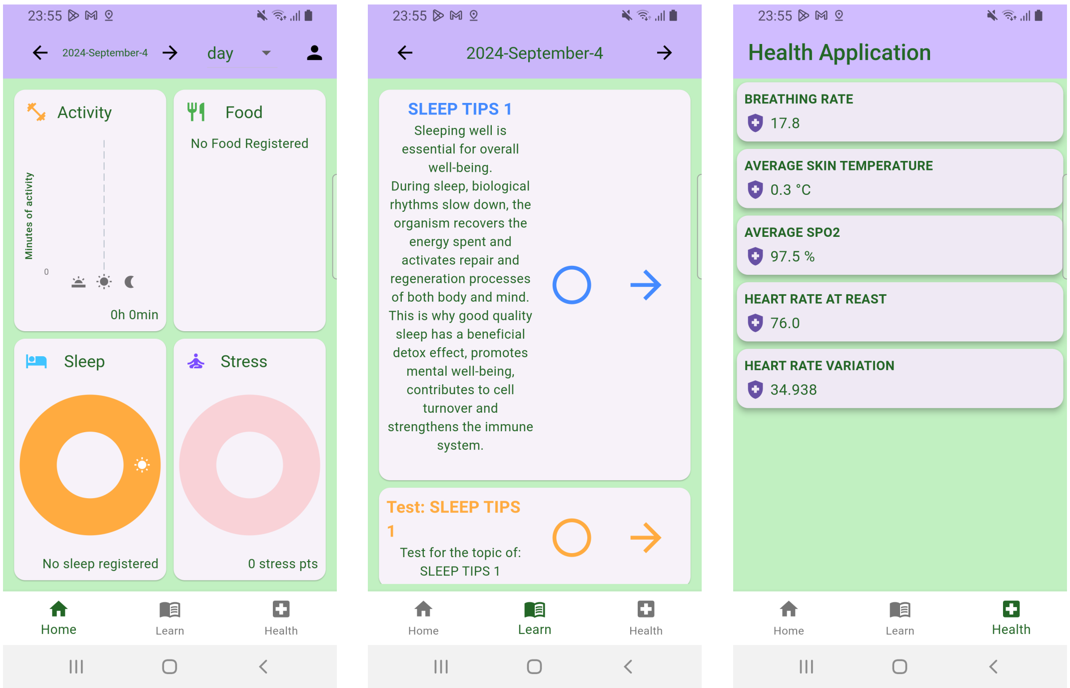

Figure 4.1: Main 3 tabs of the application

purple), and a foreground consisting of multiple gray containers. We see that all share the exact same bottom bar navigator, with 3 icons with its respective name, where the selected tab is highlighted on green.

For the home page, we see that on the app bar we have the date, with the possibility of going forward or backward, a drop down selector for the date interval, and a button to access the profile page. We see that the main content has a quadrant layout, containing 4 cards, each one of them corresponding to a different health retrieved value.

We have the activity section, which shows 3 different activity bars, one corresponding to each time of the day (morning, afternoon and night, each with its corresponding icon). We then have food, which contains a list of the ingested food and its weight. We have a sleep section that shows the slept portion of the day, and a stress section that shows the stress points. All of this sections have its numeric value on the down-right corner. You can see a filled example on *Figure* 4.2.

Then, we have the learning tab. In this case, we on the app bar the date, with two arrows at the side, that works as navigators to change the date from which the lesson is retrieved.

The main content is divided into two central cards, one for the KP, and the

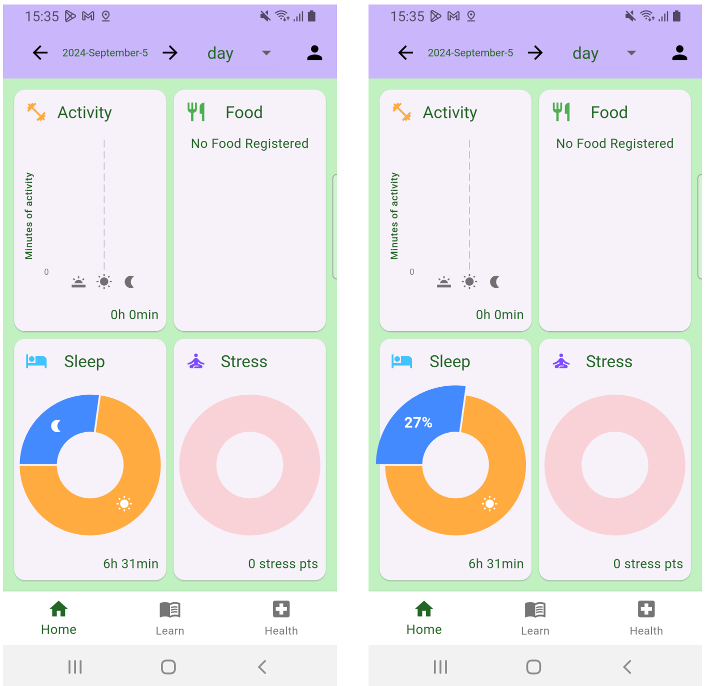

Figure 4.2: Home page with sleep information retrieved from Fitbit

other for the Test. The lesson card has the title in bold and blue, followed by a detailed yet concise description of the KP. It has a circle and arrow at the right to navigate into the lesson content. The test card is a little simpler, having the same bold title, with a shorter description, and the same circle and arrow that can be used to navigate into the quiz content. We can see an example of how the lesson content is seen on *Figure* 4.3, where we see that it has a constant theme, where the text of the content is on dark green, inside a white container.

Finally we have the health tab. This tab, on the app bar only have the title of the app on green letter over the purple background. For the main content, we have a list of 5 gray cards, with health information on each. Each have a green title with the health information, a purple icon, and the metric.

We have some additional pages, that are related to the user authentication and profile, that we can see on *Figure* 4.4. This ones have the same theme colors as the main pages that we have already seen. The only differences are that on the sign in page we have a white background, and a taller app bar, and that the warning banner on the profile is a different, much brighter yellow. Aside from that, they are fairly standard sign in and profile pages, which is to be expected as they come from the Firebase UI library.

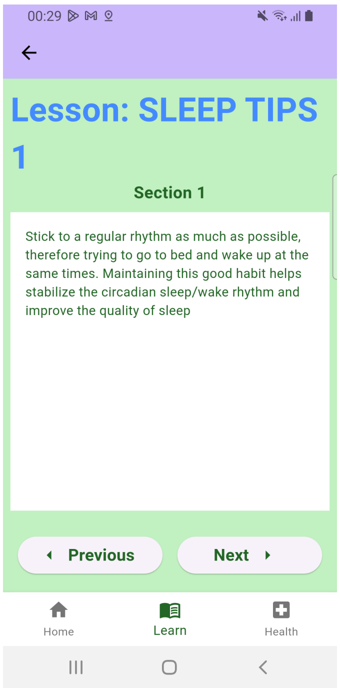

Figure 4.3: Lesson content

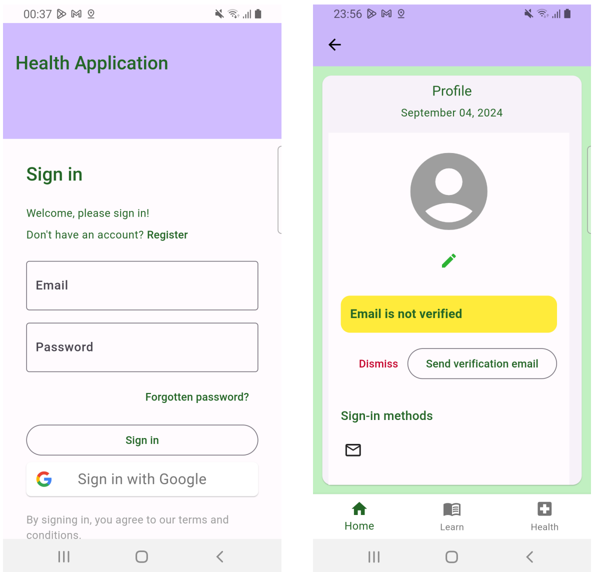

Figure 4.4: Auth related pages: Sign in and profile

4.1.2 User Experience (UX)

User Experience (UX) refers to how users interact with and perceive an application [17]. UX can be defined as a person's perception and reaction from the use of a product, system, or service. In this section, we will explore how these aspects shape the overall user experience in the current app. On the user experience side, we can see that this application has a mixed of both straightforward and good UX, and some that are not the best for performance and fluidity.

First of all, we have the navigation. The navigation between the main tabs of the app is straightforward, without any complications and has a really intuitive and fluid way, with the bottom navigation bar, as we can see on *Figure* 4.1. This is also true for the date change, as both the next and past date, and the date range selector are both well placed, they meet the eye with ease and can be used without any complications.

Regarding other navigation, such as entering the lesson content or the profile, the experience gets a little worse. Because of how the application was developed, the flutter navigator is not used, but a change of component. Due to this, expected normal actions such as using the back button to return to the previous page do not work, and they close the app completely.

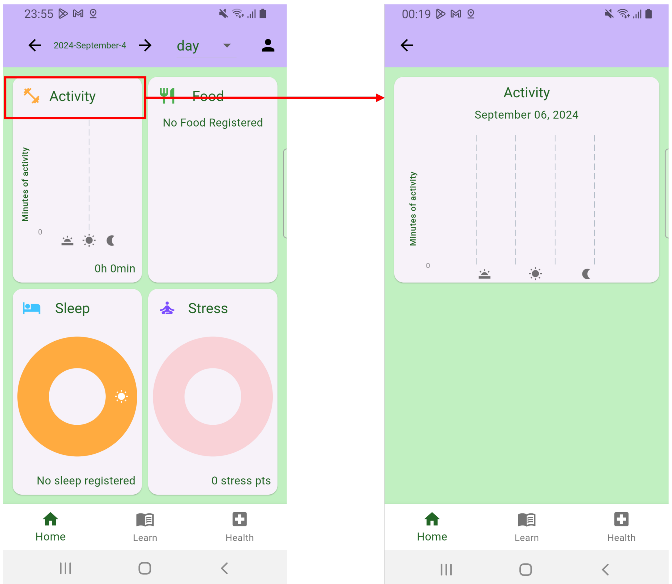

Figure 4.5: Navigation into the detailed chart and lesson content

Another relevant point with this, is the access to specific "sub pages", such as the lesson content or the detailed chart on the home page. On the home page, to access the detailed chart, you need to click on the title of the card that contains the information, as can be seen on *Figures* 4.5 and 4.6. This is not a really straightforward nor an easy way to navigate to and from the detailed page, as the user expects to be able to click over any portion of the card, which can obstruct the flow and the usability of the application. This happens also on the learning tab, where to enter the lesson content or the quiz tab, you need to click exactly on the arrow, as neither the circle nor the card do anything.

With the usability of the app, the information is presented in a clear way, with the values given to you written, and also graphically. The application even lets you press to see more information. This is a really good feature to learn more about your metrics, the problem is that it is not hinted anywhere, so arriving to it may not be easy nor direct. One problem with the flow of the application is that, as it do not the access tokens for the user, it asks the user to re-access and re-accept all the needed credentials every-time a fetch is going to be performed. This includes any time you change the date, the date range or you start the application, which can end up being really frustrating, and break the seamlessly ease of use that the app should have.

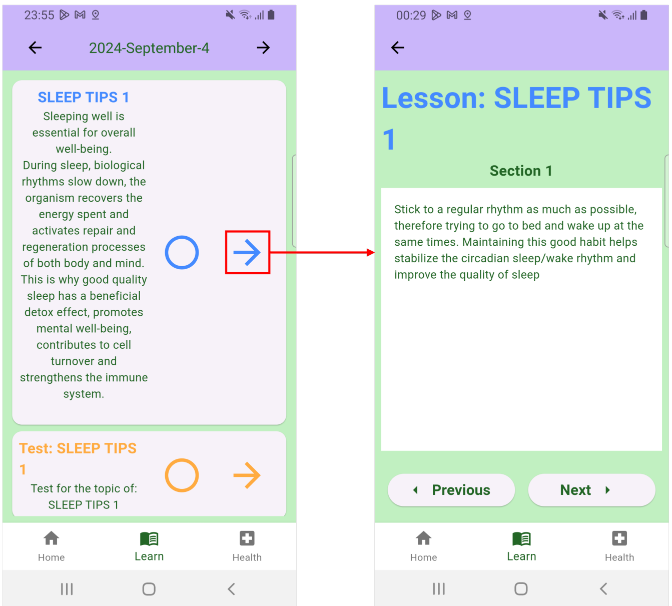

Figure 4.6: Navigation into the lesson or quiz content page

4.2 Back-end

The back-end manages the app's internal processing, including data retrieval, storage, and communication between the user interface and the server. This analysis will focus on the structure, scalability, and security of the backend infrastructure. Ensuring the backend is robust and efficient is essential for the app to handle growing user data and provide personalized insights effectively.

This application bases it's backend in 3 main API. It uses the firebase API for both user account management and authentication, and as a database (on Firestrore). For the information related to the user, it feeds itself from 2 APIs, the Fitbit and the Google Fit API.

4.2.1 Internal data management

First, we will mention how the authentication both of the app user, as the one related to the API information. The user authentication is managed with the Firebase authentication widget, that is directly imported form the library of Firebase Auth. This librabry manages the setting of the app state, it saves internally the log in state of the user, and updates the user on the backend. At the moment, the applciation supports both email-password or google log in/register. This application fetches data from Fitbit and Google Health API, the first one for the sleep and stress, and the later for the physical activity. For the google health authentication, the app requests that the user logs into its google account. For the Fitbit authentication, the user is redirected to the Fitbit page, where it needs to access its Fitbit account, and accept the scopes that the user needs.

First we will talk about how the application fetches and manages the retrieved information internally, and how this information is managed and distributed internally on the application. On the application, the information that is shared between all the pages, and that is global to the application, is shared through an app state manager internal to the flutter library, called *provider*. This is an app state manager that lets you access and modify data, independent of where on the widget tree you are. This allows for lifting an state up, or notifying a consumer widget that the app state has changed [18]. This is the file that is in charge of calling the methods that call the APIs and return the data.

For the fetching of the data, there are 2 main files, one for the Fitbit information and the other for the google fit information. Both of them are in charge of the fetching of their respective information. The Fitbit file fetches the information accessing the Fitbit Web API, where the application calls the respective API with the user's authentication token and the date. This applies also for the google health API fetching. All the data is then retrieved as a JSON value, and set into specific global variables on the provider, who notifies the consumer widgets of the change of values.

All the information that is retrieved is then set into global variables. This variables are of all data types, ranging from basic data types such as int or String, to chains of data such as Lists or Maps, on more complex data. The app data that is retrieved has no constant structure, nor data model for any of the data that is fetched.

4.2.2 Data base management

The data base is stored on Firestore, which is a non-relational storage model. The data is stored on the backend similar to MongoDB, where it works with collections and documents. Now, we will look into 2 main groups in which the data can be divided, and how it is stored, or if it is stored.

User related data

On user related data, the two main aspects are the user information, such as access tokens, user preferences, scopes, and the other would be the actual health data retrieved for the user. On this application, neither of that information is save on the backend. The only information that is saved on the backend, is the user ID, and the foods that the user registers. Aside from that, nor the access tokens, nor any information that is fetched from the API is saved.

Learning related data

For the learning tab, the information (both the KP and the quiz questions) are retrieved from the backend. The KP are saved on a collection called *lesson*, which contains the title, the description, the creation date, the sections of information, and the tags of the KP. For the quizes, they are saved on a collection called *questionary*, where there is an entry for each question, that contains the question, a list with the wrong answers, the correct answer and the KP id to which that question corresponds. There is also another collection called *user_lesson*, which contains an input for every time a user tries to complete a quiz. Each document contains the date, the lessonId, the userId and a boolean value that determines if the user completed and passed the quiz, or if it did not.

Chapter 5

Further Work Proposals

On this chapter, we will talk about the improvements that will be performed on the application. This improvements are based on use cases, trying the application out, speaking with the thesis supervisor and overall looking to complete/solve some possible errors or redundancies on the code. The future works planed will be divided on the same sections as in 4, which consists on a section for front-end and one for back-end.

5.1 Front-end

For the front-end, the proposed work will be divided into User Interface (UI) and User Experience (UX), as in the previous chapter. This approach ensures consistency throughout the document and allows for a more detailed explanation of the enhancements planned for the application.

5.1.1 User Interface (UI)

First, the entire theme of the application will be redesigned. This includes changes to the primary and secondary colors, accent colors, font styles, and the overall look and feel of the app. The proposed theme will adopt a darker palette (as opposed to the current light theme), with blue as the base color instead of the current white and green. This shift aligns with trends observed in leading well-being applications, such as *Liven* and *MyFitnessPal*, which feature dark backgrounds with lighter foreground elements. The decision to implement this change is also supported by the growing preference for dark mode among users. Research shows that up to 75% of Android users have dark mode enabled as their default setting, with similar trends observed on other platforms and applications (e.g., macOS, Twitter, iPhone), where dark mode usage consistently exceeds 50% [19].

Another major UI change will address the amount of information displayed to the user. Currently, pages like the Learning tab contain long descriptions under each knowledge pill. On the home page, the full display of the current date in the date picker results in small, hard-to-read text. To improve readability and clarity, the information will be trimmed, making it more concise, synthesized, and focused, removing unnecessary details. This change will also be applied to the *day* tables on the home page for consistency. Additionally, the activity graph in the *day* view will be replaced with a pie chart, similar to the representations used for sleep and stress, to present the information in a clearer, more direct manner.

The graphs on the home page will also undergo changes. All the *day* view graphs will be updated: the metric value will no longer be displayed at the bottom of the card but will be shown graphically in a more intuitive way. The overall design of the graphs will be enhanced, as more specific data will be fetched, with further details on this change discussed in the back-end section. The averages seen in the week and month views will also be displayed graphically instead of textually.

On the Lesson tab, the UI will be completely updated. The lesson cards will be smaller, more compact, and focused. These new designs will make them more visible and engaging, encouraging users to interact with them and follow through on their learning routines. The detailed descriptions will be removed, and the lesson content will be adjusted to be delivered on a more streamlined, distraction-free page, promoting focus on the material.

On the quiz content, the UI will also be expected to change. The quiz content will likely be changed as to be show the user if the question answers are correct or not, and to show the results of the quiz, looking to engage the user on performing the quiz activities.

5.1.2 User Experience (UX)

When addressing the User Experience (UX) of the application, several significant structural changes are required to enhance usability. Many current features are not intuitive for the users, leading to a troubled experience that needs to be optimized. By improving these elements, the goal is to create a more seamless, user-friendly interaction that maximizes efficiency and satisfaction.

The biggest issues in the existing UX is the gesture detection used for navigation. To access detailed graphs from the home page, users are forced to click on the title of the cards, rather than being able to interact with the entire card. This limited gesture recognition creates complications in navigation and can be confusing for users who expect the entire card to be clickable. A similar problem exists in the Learning tab, where users cannot access the learning content or quizzes by simply clicking anywhere on the card. Instead, they are restricted to clicking on a small arrow icon. This setup is counterintuitive and takes away from a smooth user interaction. Therefore, one of the primary UX improvements will be to enhance gesture detection across the app, allowing users to click on entire elements, such as cards, to access detailed information or features. This will generate a smoother navigation and reduce user frustration.

Another major aspect of the UX change involves refining the app's navigation system, particularly with how the Android back button functions. Currently, pressing the back button on any page causes the application to close abruptly, regardless of which screen the user is on. Whether it's the home page, a profile page, or the lesson content, the app lacks a proper back-navigation flow. This behavior breaks the expected user experience, as most apps allow users to return to the previous screen rather than shutting down entirely. To fix this, the navigation structure needs to be re-engineered so that pressing the back button takes the user back to the last page they visited. Implementing this vertical navigation flow will create a much smoother experience, preventing unintentional app closures and aligning the app's behavior with users' expectations.

Finally, on the UX, there will be a change that will avoid the need for the user to log in into Fitbit and to accept the google fit access every time the date or the day changes. The proposed way will be discussed on the back-end section.

In summary, these proposed UX improvements are aimed at addressing key pain points such as limited gesture recognition and a non-functional back button. By focusing on these core areas, the user experience will become more intuitive, reducing friction in navigation and improving overall engagement with the application.

5.2 Back-end

In this back-end section, the focus will be on how data is managed and what information is retrieved. The upcoming changes will be explained and categorized into internal data management and database management to maintain consistency throughout the document. Aside from this, many changes related to the code structure, comments, documentation ans such will be performed, aiming for a more maintainable, clear and modular code.

5.2.1 Internal data management

The underlying structure that manages the application will remain the same, with data across the app being handled through Flutter's *provider*. However, the primary changes will focus on what data is retrieved and how it is managed.

First, the activity data will now be fetched from Fitbit instead of Google Fit. This change aims to centralize data sources and ensure consistency across the application.

Regarding the fetched data, there will be adjustments to the type of data retrieved. For activity data, the current application simply fetches total activity minutes. Moving forward, the activity data will be retrieved by heart rate zone, breaking it down into the zones provided by Fitbit: moderate, vigorous, and peak activity. Similarly, changes will be made to how sleep data is fetched. Currently, the app retrieves only the total number of sleeping minutes. In the updated version, sleep data will be fetched in specific sleep stages, as defined by Fitbit: deep, light, R.E.M., and awake.

Another significant back-end update will involve changing how the data is managed internally. Currently, the fetched data is stored in various variables with inconsistent data types, ranging from strings to numeric values, and in some cases, lists or dictionaries (key-value pairs). To address this, data structures will be introduced to ensure consistency, apply data type checks, and improve the overall clarity, cleanliness, and organization of the code.

5.2.2 Data base management

In terms of the database, there will be some minor changes. First, a new collection will be created to store user information. This collection will include both back-end and front-end data. On the back-end, the primary data to be stored will include authentication and refresh tokens (used to retrieve data), the permissions the user has granted the application, as well as login dates and times. For front-end use, key data such as completed and selected Knowledge Pills (KP), as well as the user's activity and sleep goals, will be stored.

Another significant change involves how the KP and quiz information will be managed in the back-end. For the KP, the data will be stored as a streamlined data structure, which only includes essential information. Currently, extra and unnecessary details—such as the KP creation date or description—are stored, but these will be removed from the app as they are no longer needed.

Regarding the quiz information, the current database structure stores each question in a separate document within the questionnaires collection. Each document contains the question, incorrect answers, the correct answer, and the associated KP ID. The proposed change is to consolidate all questions from a single quiz into one document. This document will contain the full set of questions, with each question represented as a structured data element.

Chapter 6

Results and analysis

In this section, I will provide an overview of the changes made to the application throughout the course of the thesis. These modifications have been carried out with the primary goal of improving the overall functionality, user experience, and technical efficiency of the app. The results will be presented in two main categories: client-side changes, which focus on the front-end interface and user interactions, and server-side changes, which refers to back-end processes, database optimizations, and any additional server-related improvements.

Each category will show the specific updates implemented to improve the app's usability, performance, and long-term scalability. By breaking down the results in this manner, I aim to demonstrate how both the visible features of the app and the underlying infrastructure have been improved to deliver a better experience for the user, and have a more comprehensive app development for future developers.

6.1 Client

The first change implemented on the client side was the modification of the app's color theme. As mentioned in the previous section, the application's color scheme changed to a dark theme, utilizing a dark background with lighter foreground elements. Specifically, the primary and accent colors were updated from a light purple to a dark blue, while the background color shifted from lime green to a slightly lighter shade of blue. These changes can be seen on *Figure 6.1*. Additionally, the font colors and the onPrimary and secondary theme colors (elements that overlay the primary color or background) were adjusted from black to light gray and white, respectively, to ensure better contrast and readability within the new dark theme.

The next set of changes to the client side are related to improving the user experience related to accessing and exiting the application. A landing page was added, greeting the user each time they open the app. This page features a

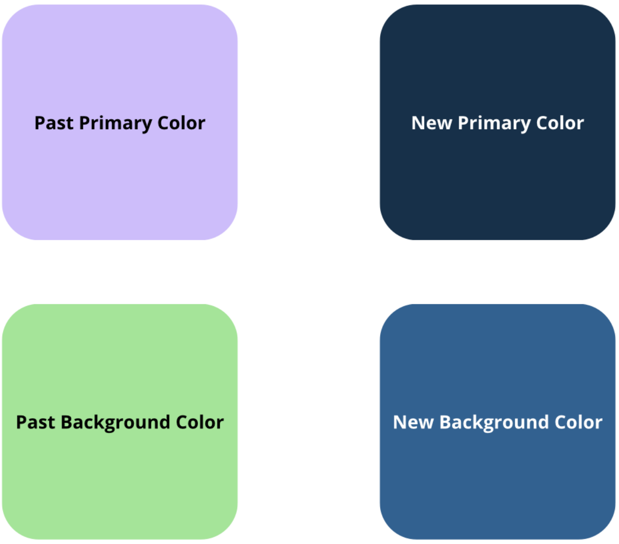

Figure 6.1: Theme color changes

heart icon accompanied by a welcoming message, which is displayed briefly before automatically redirecting the user to the home page. Additionally, a popover was added to prevent accidental exits when the user presses the back button. The popover asks the user to confirm whether they wish to exit the application, which reduces unintentional closures and encouraging users to remain engaged with the app for longer periods. This two can be seen on the *Figure 6.2*

A significant change implemented in the application involved the internal navigation between pages. Two additional pages were introduced: one is the "Recommendation" page, which is currently empty with a "coming soon" banner, as it is waiting for implementation in a future version of the application; the other is the "Personal Measures" page, which will be discussed in a following section.

Another major update was the introduction of a navigation system to facilitate page transitions. In the previous version, vertical navigation—such as moving from the home page to a detailed view or accessing the profile page—was achieved by simply replacing the widget on the same application page. With the introduction of a proper navigation system, users are now able to use the android back button as intended, navigating back to the previous page instead of exiting the app. This change significantly improves the user experience (UX), aligning the app's behavior with standard navigation patterns and enhancing its usability.

Now, I will divide the following changes in subsections, each dedicated to one of the pages on the app, to go more in detail of the changes performed on each of them.

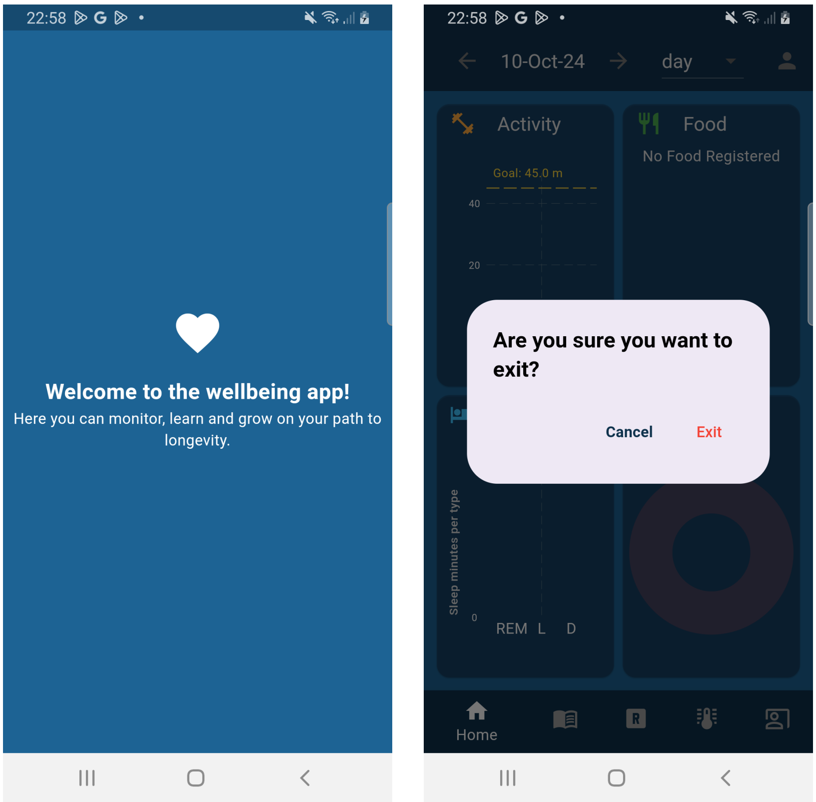

Figure 6.2: Landing page and quit popover

6.1.1 Onboarding page

Firstly, an onboarding page was added to the app, accessible when a user creates a health app account for the first time. The primary purpose of this page is to collect the initial personal metrics of new users, particularly those that are not expected to change frequently, except for weight, which may vary more often depending on the individual.

The onboarding process consists of three different types of pages, all of which are shown in *Figure 6.3*. The first page is a welcome page that provides a brief explanation of what the onboarding entails. The second type consists of the metric input pages, which include three separate pages for height, weight, and sex. For height and weight, users input their measurements in centimeters and kilograms, respectively, via text fields. For sex, users select from a dropdown menu (Male, Female, Other). The final page is the "Finish" page, which displays a thank-you message and a button to take the user to the home page.

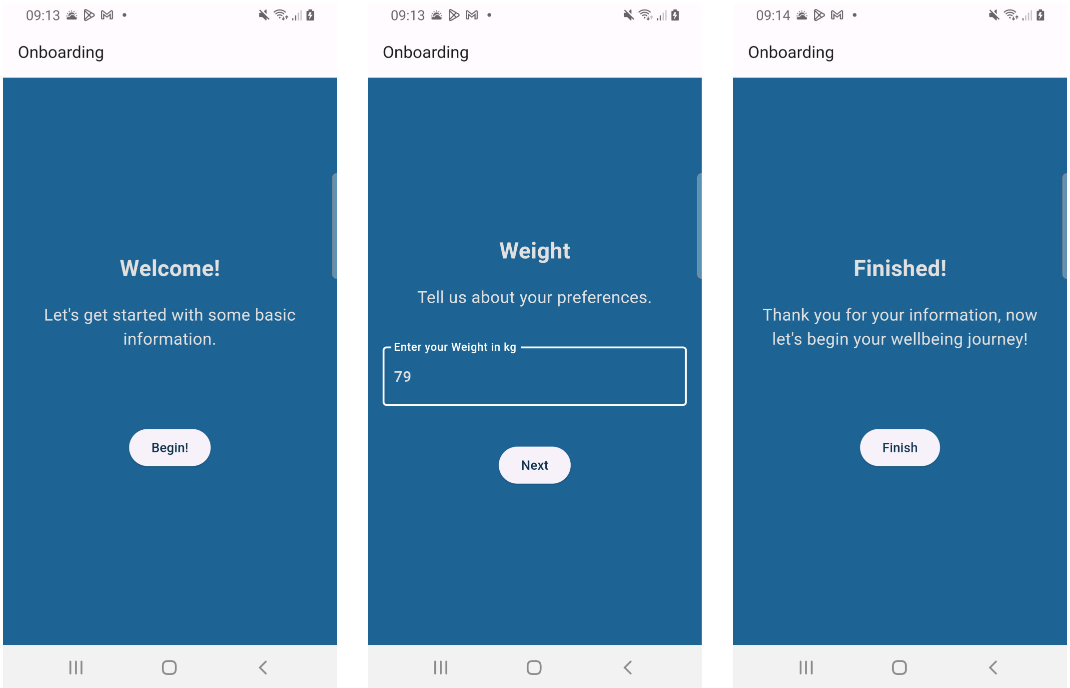

Figure 6.3: Onboarding page welcome message, edition of metric and finish message

6.1.2 Home page

The home page underwent significant changes, almost completely updating its UI and UX, that can be seen on *Figure 6.4*. To begin with, several smaller adjustments were made. First, the date selector was redesigned to be more concise, with a larger font size and a reduced, more specific date format, that aims to improve readability. Additionally, the icons on the bottom navigation bar were updated to accommodate the new app pages. Now, the navigation bar only displays the name of the selected page, while the others remain hidden, effectively highlighting the current page and reducing distractions for the user.

Another important UX improvement is that all metric cards are now fully clickable, rather than just the title. This allows users to click anywhere on the card to access the detailed graphs for each of the four metrics, aligning the app's functionality more closely with users' expectations.

The next big change performed on the home page was that now the sleep and the activity card charts changed from pie charts to bar charts. On this change, we aim to show the data clearer, on a more direct way and clear way. For the sleep and activity chart, we now divide the data into 3 bars.

Figure 6.4: Home page for the Day, Week and Month view, respectively

For the activity card, a dotted line was added to indicate the user's activity minutes goal, which will be discussed later, on the section related to the Personal Measures page. The card now displays three bars representing the activity minutes in the three heart rate zones calculated by Fitbit. Each bar corresponds to a percentage of the user's maximum heart rate, which Fitbit calculates using the Karvonen formula (220 minus the user's age). The activity zones are divided as follows:

- **Zone 1**: Less than 40% of the maximum heart rate (HR max). This zone is excluded as it is not considered active minutes by Fitbit.
- **Zone 2**: From 40% to 60% of HR max, also known as the fat burn zone.
- **Zone 3**: From 60% to 80% of HR max, referred to as the cardio zone.
- **Zone 4**: Above 80% of HR max, referred to as the peak zone.

For the sleep card, the design was also updated to a three-bar chart format. Each bar now represents one of the sleep stages tracked by Fitbit: *REM*, *Light*, and *Deep*. This values are calculated by the Fitbit by a combination of movement and heart rate variation, and the value of each stage is shown in minutes. Additionally, a bug was fixed that had previously included the time the user was in bed but awake in the sleep calculations. This time is now excluded in the new implementation, providing more accurate sleep data.

For the stress and food charts in the day view, no significant changes were made. However, a general adjustment applied to all cards was the removal of the value previously displayed in the lower right corner. Now, this value is either shown directly on the chart (e.g., on the stress pie chart label) or is available when the user interacts with the bar charts.

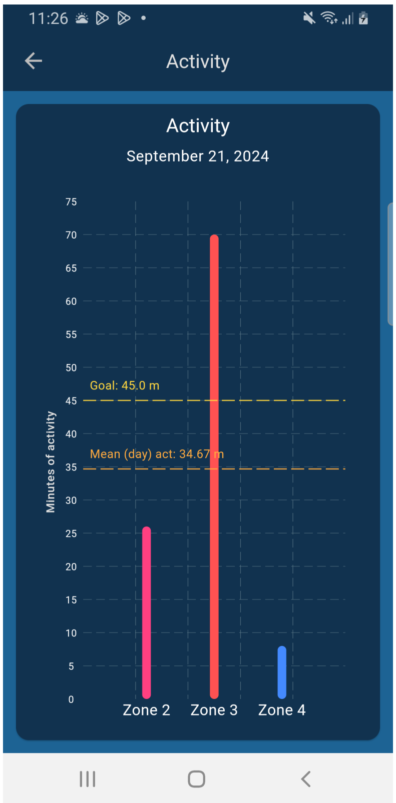

Figure 6.5: Activity detailed page per day

For the week and month views, the changes were similar. First, a bug affecting the calculation of sleep and stress on a weekly and monthly basis was resolved, and the application now displays accurate data. The format for displaying the data remains the same, with a bar chart showing the values per day (for activity and sleep, each day combines the three previously explained values). Another important update was the addition of a dotted line indicating the user's average per day for each metric over the week or month.

The activity detailed graph (Figure 6.5) also underwent some changes from its original version. First, the graph was made larger to provide a clearer view of the differences in activity minutes across zones. In addition to displaying the user's activity goal minutes, the detailed graph now includes another metric: the average activity minutes across the three zones.

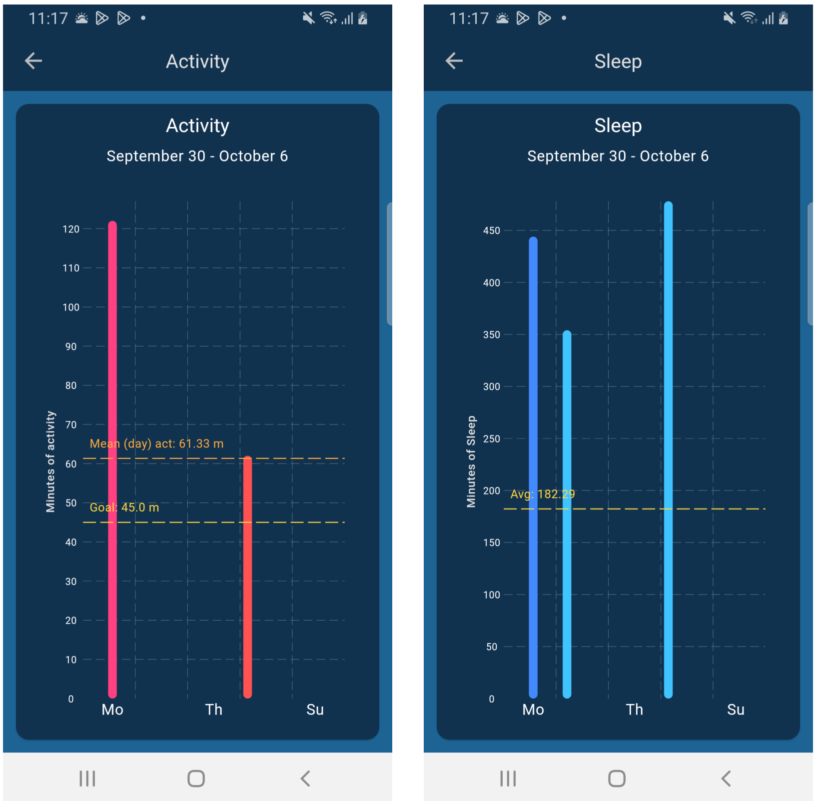

Figure 6.6: Detailed page for activity and sleep, per week

For the detailed graph in the week view (Figure 6.6), the changes related to the previous implementation were minimal. In this view, we retained the bar charts (one per day of the week), displaying the combined values for each day across activity and sleep zones. One addition, besides resizing the charts for better clarity, was the addition of the daily average for each metric. This allows users to compare their day-to-day performance with the current week's average. These updates were also applied to the month detailed page for activity, sleep, and stress metrics.

The final major change on the home page was to the food detailed graph, which includes the food list and the input feature that allows users to add new food entries. The modifications made to this page were bigger than those on any of the other detailed pages. First, the page layout was transformed from having two separate cards into a single unified card with a divider. This restructuring aimed to simplify

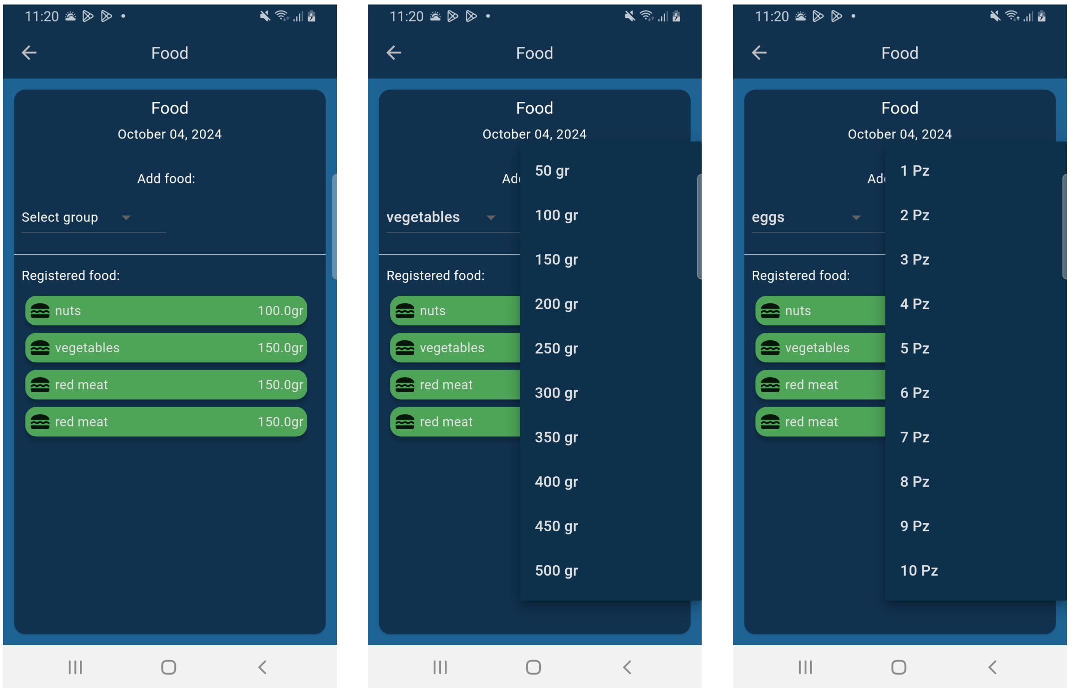

Figure 6.7: Detailed page for food addition, on day view

the user experience by replacing the large, complicated, and user-unfriendly food input form with a more compact, cleaner, and intuitive design.

The previous food input form consisted of one dropdown menu and three text fields. There were two primary issues with this implementation. First, users reported that it was too complex, and time-consuming, which discouraged them from logging any food entries. Second, the use of three text fields led to inconsistencies and inaccuracies in the input. For example, variations in how units like grams were entered (e.g., "Grams," "grams," "Gr," and "gr") were treated as distinct entries, even though they all referred to the same unit.

To address these issues, a simpler, faster, and more controlled food input form was designed. In the new input form (*Figure 6.7*), users interact with two dropdown menus, which must be completed in sequence. Initially, only the left dropdown menu is visible (as shown in the first image of 6.7), allowing users to select from a list of food groups based on what they consumed. After selecting a food group, the second dropdown menu appears. This menu lets users select the quantity of the food they consumed, with units predefined based on the selected food group and the most common units for that type of food. For example, grains or vegetables are measured in grams (gr), while other groups, such as eggs or sweets, are measured in pieces (pz).

6.1.3 Learn Page

The *Learn* page underwent a complete refactoring, resulting in a near-total transformation. The only aspect retained from the previous implementation was the general layout, which features a vertical view with two cards—one for the Knowledge Pill and the other for the quiz. However, the entire logic, as well as the UI/UX and internal structure, were significantly redesigned and improved.

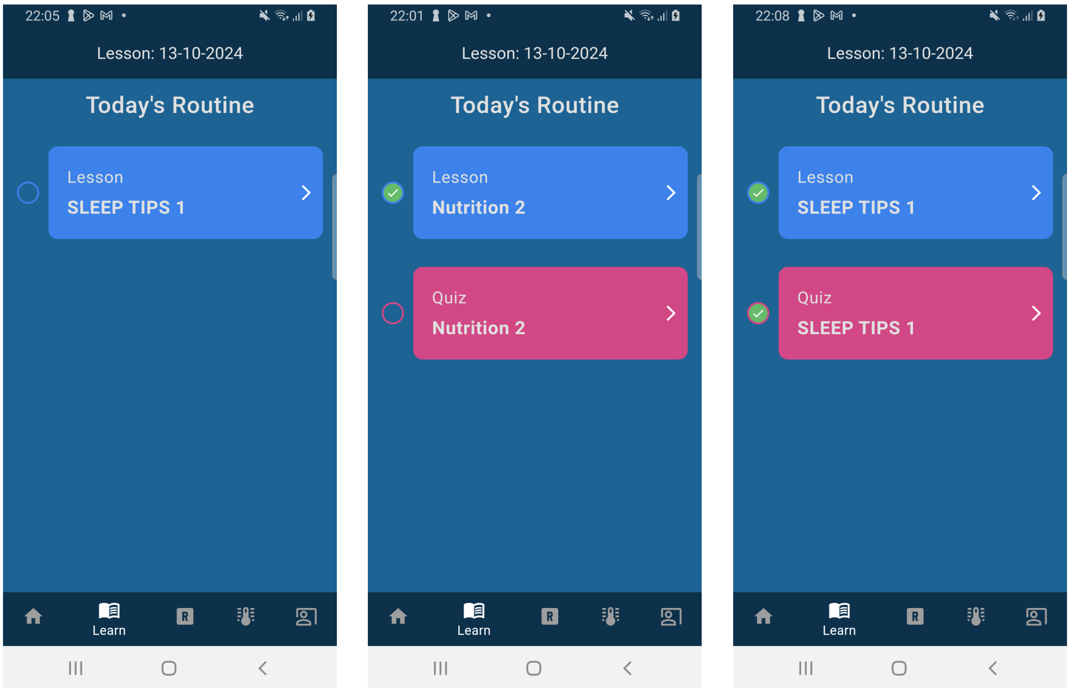

Figure 6.8: Initial learn page, with the KP completed, and with both the KP and quiz completed

The first significant UI change on the Learning page involved a complete redesign of the cards for both the Knowledge Pill (KP) and the quizzes (*Figure 6.8*). Previously, the cards were large, gray, and displayed the full description, and they could only be interacted with by tapping the arrow icon. The new design offers a more compact, minimalistic, and visually appealing card, displaying only the KP title, with full interactivity. Both cards feature a circle on the left, initially empty, but once the task is completed (whether reading the entire KP or passing the quiz) the circle fills with green and displays a white checkmark icon.

The way users interact with the information and quizzes has also changed. In the previous version, both cards were always visible, regardless of user progress. In the current implementation, at the start of each day, only the KP card is available.

Once the KP has been completed and the user has gone through all the learning sections, the quiz card appears, allowing the user to take the corresponding quiz for the completed KP. This KP is renovated each new day, in a cycle that lasts for the amount of KP that are available on the DB, without repeating them until the end of the learning cycle. This is signaled by a popover, which lets the user know that there is a new KP available on the first access of the day (*Figure 6.9*).

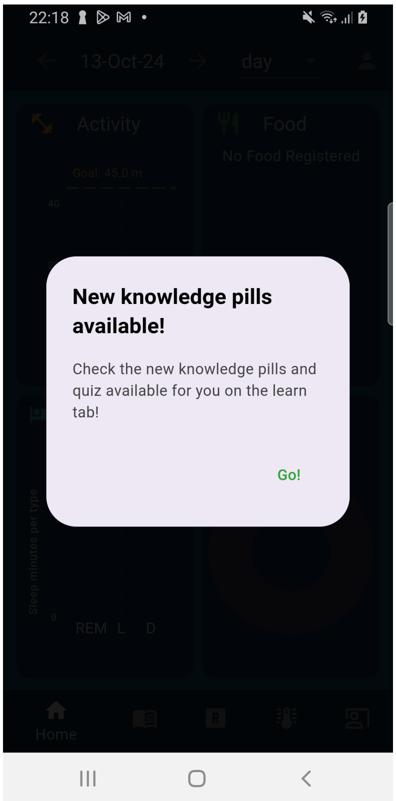

Figure 6.9: New lesson popover

In addition to the main lesson page, the sub-pages derived from it also underwent significant changes. First, the Knowledge Pill (KP) section page, which displays the KP content, received a complete UI overhaul (*Figure 6.10*). The design was updated to match the new theme. Furthermore, a series of introductory, general sentences were added to the page, with one randomly selected for each KP section as the header, aimed at creating a smoother and more engaging data delivery experience. Lastly, the navigation buttons at the bottom were also updated. It is now clearly indicated when the user is on the first page (the left button is disabled and displayed in a darker color) and when they have reached the last section (the right button turns green with a "Complete!" text).

The quiz page, where users complete the questionnaire for the daily Knowledge

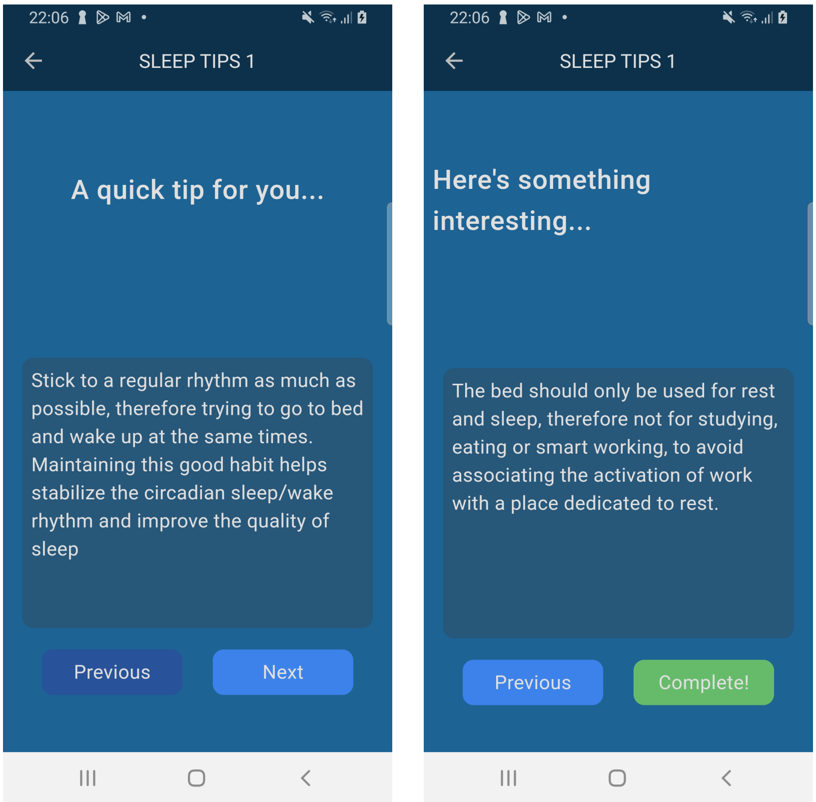

Figure 6.10: Lesson KP content display

Pill (KP), also underwent significant UI and UX improvements. In addition to the visual changes, a major flaw from the previous version was addressed: the randomization of the possible answers. In the past implementation, the correct answer was always listed first, making it easy for users to guess rather than truly engage with the content. In the new version, the client automatically randomizes the order of the possible answers each time the user accesses the quiz, encouraging users to focus on learning the material rather than relying on patterns to answer correctly.

The entire UI was refactored (*Figure 6.11*) to include larger, clearer fonts for the quiz questions, making it easier for users to understand what is being asked. Additionally, the user experience (UX) of interacting with the quiz was reworked. In the previous version, users had to select an answer and press a button to submit it, with no immediate feedback or confirmation of their response. In the new implementation, users simply click on their selected answer, and the app provides instant feedback. The selected answer changes color to indicate whether it was correct (green) or incorrect (red). This feedback remains visible for 0.5 seconds

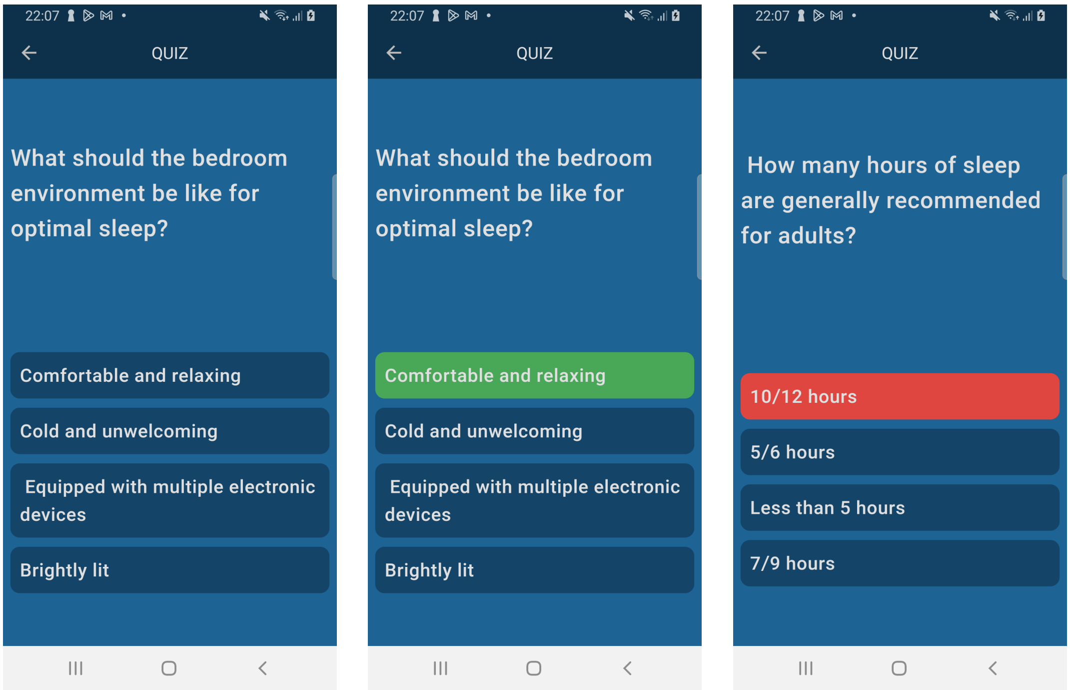

Figure 6.11: Quiz page, with right and wrong answer examples

before automatically moving the user to the next question. This creates a more engaging, straightforward, and interactive experience, providing the user with immediate feedback on their actions.

At the end of the quiz, a common issue in the previous version was that users couldn't always tell if they had completed the quiz, as they were abruptly returned to the lesson main page. To address this, a popover was added at the end of the questionnaire to provide users with clear metrics on their performance. There are two main popover scenarios: one for quiz completion and one for quiz failure (*Figure 6.12*). For quiz completion, users are congratulated, informed of the number of correct answers, and given a "Save" button to store their progress on the backend. In the failure case, the popover informs the user that they did not pass the quiz, shows how many questions they got right, and explains why the quiz was failed. In the current application, users need to answer at least 60% of the questions correctly to pass, which is emphasized in the failure popover message.

6.1.4 Health Measures Page

On the health measures page, the changes were much fewer compared to the other pages. The updates mainly focused on UI improvements on the client side (the

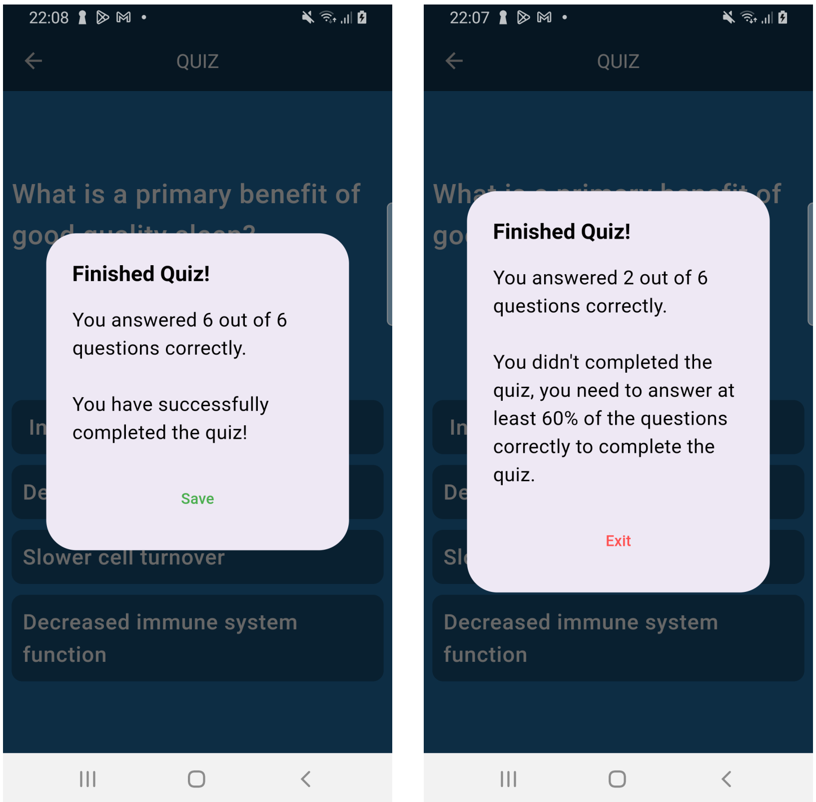

Figure 6.12: Quiz popover, indicating either completion or fail of the questioner

changes related to the data itself will be discussed in the Server section 6.2). The two main adjustments were as follows. First, the data presentation was made clearer by using a larger font size and more contrasting colors. Additionally, the formatting of health measures was improved by removing unnecessary decimal places that did not add value to the information being conveyed (on some of the health measures) to the user. Since the page now displays actual data, rather than placeholders as before, a "no data" message was also introduced for cases where data is unavailable. These changes are illustrated in (*Figure 6.13*).

6.1.5 Personal Measures

Another addition to the application is the Personal Measures page (*Figure 6.14*). This page was introduced as a place where users can store personal information related to them, which rarely changes, and is specific to the individual. It was created to record both the user's physical information and their goals, allowing users to set daily target goals for physical activity or sleep. These goals will be

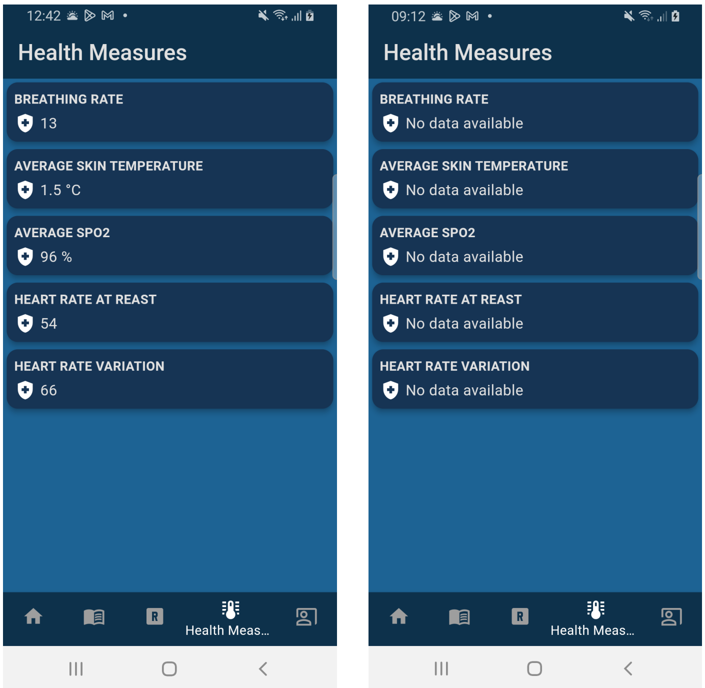

Figure 6.13: Health measures page, with and without values

valuable once the recommendation agent feature is implemented, and are used on the home page, to illustrate how the user is doing, on a daily basis.

The page is designed with two main vertically aligned sections, separated by a horizontal divider. In the first section, users can input their Personal Measures, which include three key metrics: *Sex*, which can be selected from a dropdown menu, and *Weight* and *Height*, which are input fields where users can enter exact values (height in centimeters and weight in kilograms).

The second section corresponds to the daily goals. In this section, there are two input fields for modifying these goals: one for the daily activity goal, measured in minutes, and another for the daily sleep goal, measured in hours. These goals are initially set based on the user's information stored in the backend, with default values of 30 minutes of activity per day and 8 hours of sleep. Users can update these goals as desired on this page.

6.2 Server

This section explains the changes, modifications, and improvements related to the database, data management, data fetching, and data structures. All of these

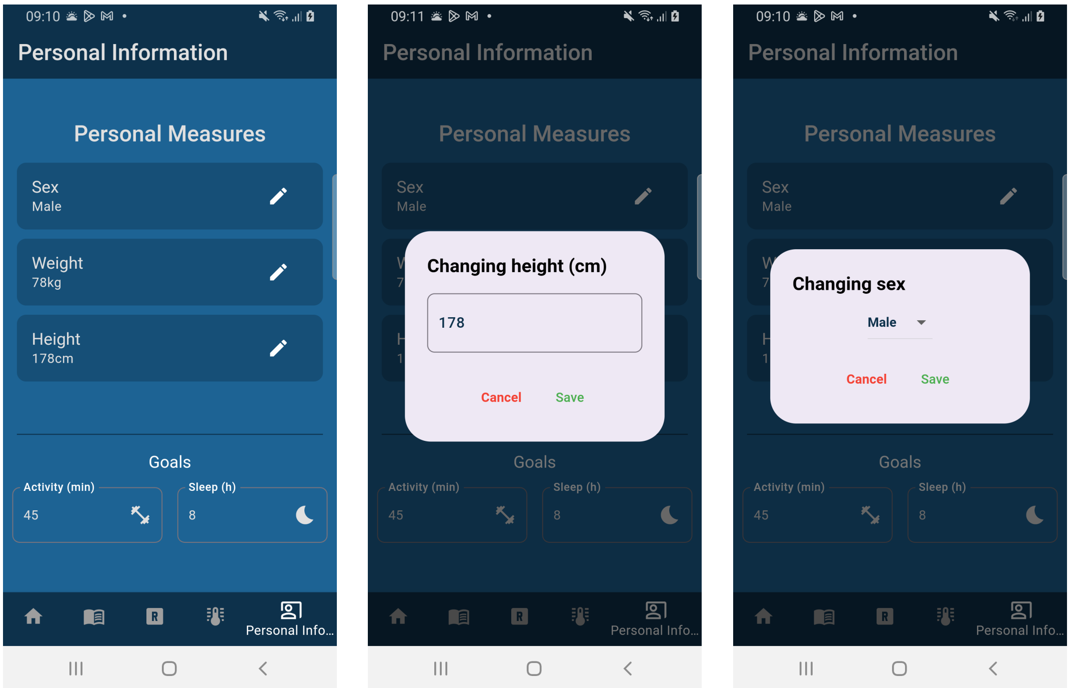

Figure 6.14: Personal Measures page, and editing of personal measures

adjustments were made with the goal of enhancing user experience, organizing the code and database more effectively, and providing a clearer development framework. This ensures that any developer looking to work further on the application in the future can do so without significant setbacks.

6.2.1 Data storing and Data fetching

The first change focused on addressing a major UX issue from the previous implementation. In this update, Fitbit authentication and refresh tokens are now stored on the Firebase backend for each user. These tokens enable the client to fetch Fitbit data whenever the user accesses the app, without requiring them to re-enter their Fitbit credentials each time. The backend stores both the authentication token, used to retrieve Fitbit information, and the refresh token, which allows the client to request a new authentication token when the current one expires. This eliminates the need for users to repeatedly enter their credentials, greatly improving the app experience.

Another issue discovered during app development was the difficulty in updating user consents. Previously, when a consent change was required, it was challenging to implement without accessing the source code, logging out, and deleting the user's account, forcing a complete re-login on both the health app and Fitbit connection. To address this, a constant list of required consents was added to the app's code. This list contains all the consents that users must agree to in order for the app to function properly.

When a user initially connects their Fitbit account and agrees to the required consents, this list is saved in the Firebase database. Each time the user accesses the app, the stored consent list is compared with the current required consent list (stored as a constant in the app). If the lists differ, the app automatically redirects the user to the Fitbit login page to update the necessary consents. If the lists match, the user can proceed with the app's normal functionalities. This solution accommodates both new and existing users, ensuring easy and seamless consent updates whenever new information is requested from Fitbit that requires additional permissions (a situation that occurred during development, which will be explained later).

The following list shows the different scopes used in the current implementation of the application:

- Activity
- Nutrition
- Sleep
- Heart rate
- Oxygen saturation
- Respiratory rate
- Temperature
- Weight
- Profile

Aside from the database changes, there were also updates to the data fetched and used by the application. This involved one major addition and a change in the source of the fetched data.

The main change was related to the source of activity data. In the previous implementation, activity minutes were obtained from Google Health, meaning users had to sign in to two separate services to collect all the necessary data, which also posed a risk of inconsistency. To address these issues, the source of activity data was switched from Google Health to Fitbit [20]. This change was made to simplify the user experience by requiring only one login for data access. Additionally, since the activity data now comes from Fitbit, which measures activity directly via the

Fitbit bracelet, it ensures that the data is consistent and aligned with all other information fetched from the Fitbit API, creating a more cohesive data flow within the app.

The other significant addition to the fetched data was the retrieval of the user's actual output variables, which are the ones displayed in *Figure 6.13*. In the previous app implementation, these were only mock values meant to showcase the UI and provide examples of possible outcomes. However, in the current version, the app fetches real output variables and displays accurate values. These output variables correspond to health metrics provided by Fitbit, which are derived from the health analysis Fitbit performs during the user's sleep. The fetched values include:

- **Breathing rate:** the user's average breaths per minute at night.
- **Average skin temperature:** the user's average skin temperature during sleep.
- **Average SPO2:** a measurement of the user's blood oxygen level.
- **Heart rate at rest:** the user's heart rate while at rest.
- **Heart rate variation:** a measure of the variability in the user's heart rate.

These metrics provide users with detailed insights into their overall health based on their sleep data.

6.2.2 Data structures and Data internal management

Another crucial aspect of the app modifications involved changes to the internal data structure and the management of data fetched from Fitbit. These changes were implemented with two primary goals in mind.

First, they aimed to enhance the performance and organization of the data within the app. Simplifying and organizing how the app handles Fitbit data ensures smoother and more efficient processing, which can reduce latency, be less prone to errors, and improve the overall user experience.

Second, these changes were designed to improve readability, reusability, and ease of understanding for future developers working with the backend and code of the application. By making the data structure more intuitive and modular, the code becomes more maintainable and easier to extend or modify, which is particularly important as the app evolves and new developers contribute to its development. This refactor supports long-term sustainability and helps avoid potential setbacks or confusion in future iterations.

First, there were several bugs in the way the fetched data was handled, particularly concerning the sleep data from Fitbit. These issues caused inaccuracies in how the app presented users with their sleep metrics. The two main problems were tied to how the app read and used the sleep data after fetching it from Fitbit.

The first issue occurred in the way sleep data was read, leading to the inclusion of the **awake** time in the total sleep calculation. This resulted in an incorrect and inflated sleep duration, giving users an inaccurate view of their sleep patterns. Fixing this involved excluding the awake time from the sleep calculation to provide a more accurate representation of the user's sleep.

The second issue related to the sleep categories displayed in the app. Initially, the app did not use the correct categories returned by the Fitbit API, causing the categories to default to a single value, preventing users from getting a detailed breakdown of their sleep stages. To resolve this, the app was updated to correctly fetch and display the three sleep categories Fitbit provides—**REM**, **light**, and **deep sleep**—while excluding the awake time from these categories. This fix ensured that users received a clear and accurate overview of their sleep patterns.

After resolving the previous issues, the next step focused on optimizing the use of data structures for better organization and management of the app's data. To achieve this, three data structures were introduced: two for managing knowledge pills and one for the food list. The new data structures aimed to enhance consistency and streamline data management by incorporating model verification, ensuring that incorrect information would be flagged, and speeding up the development process.

Previously, the app relied heavily on dictionaries (key-value pairs) to manage everything. This method was problematic as it depended solely on string keys and lacked clearly defined data types for each entry. Moreover, there were no default values, increasing the risk of errors. The new data structures provided a more stable framework and better control over the input and retrieval of data.

For knowledge pills, three interconnected data models were introduced: one for the knowledge pill itself, one for the quizzes, and one for the questions. These models work together, with the knowledge pill serving as the base model and referencing the quiz model externally.

The Knowledge Pill model includes the following attributes:

- **Id (String)**: A unique identifier for the knowledge pill. This identifier is also stored in a separate document on the backend, which contains a list of all available knowledge pills (KP). This structure allows the system to assign a different and random KP to each user daily, without requiring the client to fetch all the KP models, reducing client-side overhead and improving efficiency.
- **Title (String)**: The title displayed on the lesson page card.
- **Sections (List of Strings)**: A list of sections that make up the knowledge pill. Each section is shown on a separate page within the lesson.

- **Completed (Boolean)**: A flag to indicate whether the knowledge pill has been completed.
- **Quiz (Quiz data model)**: A reference to the associated quiz model.

The Quiz data model includes:

- **Id (String)**: A unique identifier for the quiz.
- **Lesson Id (String)**: The identifier for the lesson to which the quiz belongs.
- **Questions (List of question model)**: A list of questions included in the quiz.
- **Completed (Boolean)**: A flag to indicate if the quiz has been completed.
- **Correct Answers (Int)**: The number of correct answers given by the user.
- **Total Questions (Int)**: The total number of questions in the quiz.

Finally, the Question model includes:

- **Question Text (String)**: The text of the question.
- **Possible Answers (List of Strings)**: A list of all possible answers, including the correct one.
- **Correct Answer (String)**: The text of the correct answer.
- **Is Answered (Boolean)**: A flag to indicate if the question has been answered.
- **Is Correctly Answered (Boolean)**: A flag to indicate whether the question was answered correctly.

These structured models not only improved the integrity and reliability of the app's data but also made the development process more intuitive and easier to follow for future developers.

All these models share three common methods that enhance their functionality and interaction with the database. These methods are:

1. **fromMap**: This method is used to create an instance of the model from a dictionary, transforming the raw data from the database into a usable model object within the app.

2. **toMap**: This method converts the model back into a dictionary, allowing it to be saved in the database. These two methods are crucial for handling the data flow between the app and the database, which uses dictionaries to store and retrieve information.

3. **copyWith**: This method enables developers to create a copy of an existing model with specific fields updated or modified. This functionality is especially useful when a model needs to be partially updated, such as marking a knowledge pill or quiz as completed, without modifying the other attributes.

In addition to these shared methods, the Quiz and Question models have an extra method:

- **answerQuestion**: This method is invoked when the user selects an answer to a quiz question. It marks the question as answered and checks whether the selected answer is correct. If the answer is correct, the method updates the Correct Answers count in the quiz model, ensuring the app tracks the user's performance in real-time.

These models are stored as dictionaries in the backend, with the attribute names as keys and the corresponding values as values. However, there is one key difference for the Knowledge Pill model. Instead of storing the quiz as an embedded object, it uses a quizId parameter, which references the associated quiz stored separately in the database. This design simplifies the data structure, avoiding duplication and ensuring that each knowledge pill only references the necessary quiz without including all the quiz details within itself.

The Food Entry model was created to structure and manage the user's food intake entries within the app, ensuring that the data is well-organized and easily stored in the backend. The attributes of this model are:

- **Date (DateTime)**: This field records the date on which the food entry was made, helping to track daily consumption.

- **Food Group (String)**: This specifies the category or group to which the ingested food belongs (e.g., fruits, vegetables, grains), providing context for the type of food consumed.

- **Amount (String)**: A string that combines both the quantity of food ingested and the unit of measurement (e.g., "200 gr" or "2 pz"), with the two components separated by a space.

- **User Id (String)**: The unique identifier of the user making the food entry, linking the entry to a specific individual.

The Food Entry model includes the same methods as the previously discussed models, which are the **fromMap** and **toMap** methods, used mainly to interact with the DB. Additionally, it has a **display method** designed to easily debug and display the food entry data. This method aids developers by simplifying the process of reviewing and verifying the information being sent to or received from the backend, ensuring smoother development and testing.

By implementing this structured approach to food entries, the app ensures that users' dietary habits are recorded consistently, allowing for better tracking and analysis of their health metrics.

6.3 Extra services

Finally, an additional service was developed on top of the changes made to the application. This service is a Python-based tool that enables administrator users to view, read, and add knowledge pills (KPs) and quizzes directly to the database.

This extra service was done because, in the previous implementation of the app, the only way to add a new KP or quiz was by accessing the database directly and manually inputting the required fields. This method was not user-friendly and demanded technical knowledge, making it impossible for non-technical users to add, modify, or remove KPs and quizzes.

Since the responsibility of adding lessons and quizzes may not always fall on technical personnel, but instead on users with administrative credentials, there needed to be a way for them to perform these tasks in an intuitive and non-technical manner.

To solve this issue, a Python script was developed. This script allows users to run the application from the terminal following simple instructions. When the application starts, a main menu appears with four options:

- **Create a new lesson:**
  - The user is prompted to enter details such as the lesson title, its content sections, and relevant tags.
  - Once all information is provided, the lesson is saved to the Firestore database.
  - The lesson Id is automatically added to the data structure that has all the available KP IDs
- **Show all lessons:**
  - This option retrieves and displays all the lessons currently stored in Firestore.

  - The user can view the list of existing lessons, each of which may contain information such as the lesson title, sections, and associated tags.

- **Create a new quiz:**
  - The application fetches the list of available lessons, and the user is prompted to select one to associate with the new quiz.
  - The user is guided through a series of prompts to create the quiz questions.
  - After completing the quiz creation, it is saved to Firestore and linked to the selected lesson for future use within that context.

- **Show all quizzes:**
  - This option fetches and displays all quizzes that have been created and stored in Firestore.
  - The user can view details such as the lesson each quiz is associated with and its list of questions.

- **Quit:**
  - This option allows the user to exit the application, terminating the current session.

This tool significantly simplifies the process of managing KPs and quizzes, making it more accessible to non-technical users while maintaining the integrity and organization of the database.

Chapter 7

Conclusions

The redesigned wellbeing app achieved significant improvements in both user experience and functionality. Test users reported notable enhancements in usability and experience, validating the design choices aimed at simplifying interactions and making health tracking more intuitive. The app's new features, including a visually appealing interface and personalized user insights, successfully encouraged consistent engagement with the platform. By focusing on these aspects, the app better supported users in pursuing healthier lifestyle habits, thus fulfilling the project's primary objective.

Several areas for future development could further enhance the app's capabilities. Incorporating new data input technologies, such as integrating information from Samsung Health or adopting smart devices like the Apple Watch, would broaden the app's range of potential users. Additionally, the implementation of a recommendation page—powered by large language models (LLMs)—could offer personalized health advice based on the user's metrics. This feature would elevate the app from a simple monitoring tool to a comprehensive health advisor, providing users with actionable, tailored insights.

Moreover, expanding the app's scope by integrating more advanced sensors and wearable devices could improve the accuracy and depth of health data collection. This would create a richer foundation for analysis and recommendation generation, ensuring that users receive timely and precise guidance.

In conclusion, the project's outcomes demonstrated the potential of mobile technologies to promote wellbeing, with considerable gains observed in both user and developer satisfaction. The app's architecture, built on scalable and flexible technologies, positions it well for future growth and innovation. As the digital health landscape continues to evolve, the app is well-equipped to adapt to emerging trends and technologies, offering users an increasingly valuable tool for managing their health and wellness.

Bibliography

[1] Fernando Colchero et al. «The emergence of longevous populations». In: *Proceedings of the National Academy of Sciences* 113.48 (2016), E7681–E7690. DOI: 10.1073/pnas.1612191113. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1612191113. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1612191113 (cit. on p. 1).

[2] Matthew McGough Shameek Rakshit and Krutika Amin. *How does U.S. life expectancy compare to other countries?* 2024. URL: https://www.healthsystemtracker.org/chart-collection/u-s-life-expectancy-compare-countries/#Life%20expectancy%20at%20birth,%20in%20years,%201980-2022 (cit. on pp. 1, 6).

[3] Jessica Evert, Elizabeth Lawler, Hazel Bogan, and Thomas Perls. «Morbidity profiles of centenarians: survivors, delayers, and escapers». en. In: *J. Gerontol. A Biol. Sci. Med. Sci.* 58.3 (Mar. 2003), pp. 232–237 (cit. on p. 1).

[4] Paul B Baltes and Jacqui Smith. «New frontiers in the future of aging: from successful aging of the young old to the dilemmas of the fourth age». en. In: *Gerontology* 49.2 (Mar. 2003), pp. 123–135 (cit. on p. 1).

[5] L. Fontana. *The Path to Longevity: How to reach 100 with the health and stamina of a 40-year-old*. Hardie Grant Books, 2020. ISBN: 9781743586693. URL: https://books.google.com.co/books?id=tRrkDwAAQBAJ (cit. on pp. 3, 6, 7).

[6] Saloni Dattani, Lucas Rodés-Guirao, Hannah Ritchie, Esteban Ortiz-Ospina, and Max Roser. «Life Expectancy». In: *Our World in Data* (2023). https://ourworldindata.or expectancy (cit. on p. 5).

[7] Colin D. Mathers, Gretchen A. Stevens, Ties Boerma, Richard A. White, and Martin I. Tobias. «Causes of international increases in older age life expectancy». In: *The Lancet* 385.9967 (Feb. 2015), pp. 540–548. ISSN: 0140-6736. DOI: 10.1016/S0140-6736(14)60569-9. URL: https://doi.org/10.1016/S0140-6736(14)60569-9 (cit. on p. 6).

[8] I. M. Lee, R. S. Paffenbarger, and C. H. Hennekens. «Physical activity, physical fitness and longevity». In: *Aging Clinical and Experimental Research* 9.1 (Feb. 1997), pp. 2–11. ISSN: 1720-8319. DOI: 10.1007/BF03340123. URL: https://doi.org/10.1007/BF03340123 (cit. on p. 6).

[9] I-Min Lee and Jr. Paffenbarger Ralph S. «Associations of Light, Moderate, and Vigorous Intensity Physical Activity with Longevity: The Harvard Alumni Health Study». In: *American Journal of Epidemiology* 151.3 (Feb. 2000), pp. 293–299. ISSN: 0002-9262. DOI: 10.1093/oxfordjournals.aje.a010205. eprint: https://academic.oup.com/aje/article-pdf/151/3/293/264648/151-3-293.pdf. URL: https://doi.org/10.1093/oxfordjournals.aje.a010205 (cit. on p. 6).

[10] Orfeu M Buxton, Milena Pavlova, Emily W Reid, Wei Wang, Donald C Simonson, and Gail K Adler. «Sleep restriction for 1 week reduces insulin sensitivity in healthy men». en. In: *Diabetes* 59.9 (June 2010), pp. 2126–2133 (cit. on p. 6).

[11] Diego Robles Mazzotti, Camila Guindalini, Walter André dos Santos Moraes, Monica Levy Andersen, Maysa Seabra Cendoroglo, Luiz Roberto Ramos, and Sergio Tufik. «Human longevity is associated with regular sleep patterns, maintenance of slow wave sleep, and favorable lipid profile». In: *Frontiers in Aging Neuroscience* 6 (2014). ISSN: 1663-4365. DOI: 10.3389/fnagi.2014.00134. URL: https://www.frontiersin.org/journals/aging-neuroscience/articles/10.3389/fnagi.2014.00134 (cit. on p. 6).

[12] Cem Ekmekcioglu. «Nutrition and longevity – From mechanisms to uncertainties». In: *Critical Reviews in Food Science and Nutrition* 60.18 (2020). PMID: 31631676, pp. 3063–3082. DOI: 10.1080/10408398.2019.1676698. eprint: https://doi.org/10.1080/10408398.2019.1676698. URL: https://doi.org/10.1080/10408398.2019.1676698 (cit. on p. 7).

[13] Valter D. Longo and Rozalyn M. Anderson. «Nutrition, longevity and disease: From molecular mechanisms to interventions». In: *Cell* 185.9 (Apr. 2022), pp. 1455–1470. ISSN: 0092-8674. DOI: 10.1016/j.cell.2022.04.002. URL: https://doi.org/10.1016/j.cell.2022.04.002 (cit. on p. 7).

[14] Stephen H. Friend, Geoffrey S. Ginsburg, and Rosalind W. Picard. «Wearable Digital Health Technology». In: *New England Journal of Medicine* 389.22 (2023), pp. 2100–2101. DOI: 10.1056/NEJMe2303219. eprint: https://www.nejm.org/doi/pdf/10.1056/NEJMe2303219. URL: https://www.nejm.org/doi/full/10.1056/NEJMe2303219 (cit. on p. 7).

[15] Andrea Gaggioli and Giuseppe Riva. «From mobile mental health to mobile wellbeing: opportunities and challenges». en. In: *Stud Health Technol Inform* 184 (2013), pp. 141–147 (cit. on pp. 7, 8).

[16] American Psychiatric Association. *T2 Mood Tracker Sample App Evaluation*. 2019. URL: https://www.psychiatry.org/psychiatrists/practice/mental-health-apps/evaluations/t2-mood-tracker (cit. on p. 8).

[17] M A T Pratama and A T Cahyadi. «Effect of User Interface and User Experience on Application Sales». In: *IOP Conference Series: Materials Science and Engineering* 879.1 (July 2020), p. 012133. DOI: 10.1088/1757-899X/879/1/012133. URL: https://dx.doi.org/10.1088/1757-899X/879/1/012133 (cit. on p. 13).

[18] Flutter Development Team. *Simple app state management*. 2024. URL: https://docs.flutter.dev/data-and-backend/state-mgmt/simple (cit. on p. 15).

[19] Alexander Eser. *Dark Mode Usage Statistics: Users Optimize Battery, Reduce Eye Strain*. 2024. URL: https://worldmetrics.org/dark-mode-usage-statistics/ (cit. on p. 17).

[20] Google. *FitBit*. URL: https://store.google.com/category/watches_trackers?hl=it&utm_source=fitbit_redirect&utm_medium=google_ooo&utm_campaign=category (cit. on p. 36).