Department of Computer Science
University of Hong Kong
Final Year Project

Individual Report

Development of a smartphone app and website for prediction of coronary heart disease risk (Industry-based Project)

By:
Chan Hay Yin (3035686047)

Supervisor:
Dr. Loretta Choi
Abstract

With the rise in the number of deaths caused by coronary heart disease (CHD) in recent years, it is beneficial for people to assess and monitor their own risks. In collaboration with Dr. Youngwon Kim and his team from The University of Hong Kong’s School of Public Health, our project aims to provide an easy-to-use, cross-platform mobile app for users to monitor their risks. To do this, our app will be written in React Native, with the use of Firebase as the database. We will also use Fitbit smart watches to collect various data from the user to calculate the users’ risks using the risk prediction algorithm developed by Dr. Kim’s team. Included in our final deliverable is an app written based off our proposed framework, with dynamic data displays and graphs pulling data from the backend and displaying them to the user in the frontend clearly such that they can be informed about their own health conditions. We have also implemented themed discussion forums, where the users are able to read and join discussions on various health topics and issues that they are interested in.
Acknowledgements

I would like to thank our supervisor Dr. Loretta Choi, for providing relevant expertise and suggestions when we are stuck, as well as always checking and responding to our sudden questions and requests.

I would also like to thank Dr. Youngwon Kim and his team from the School of Public Health. Without them joining the industry-based project, we would not be able to join such a meaningful project.

Finally, I would like to thank my groupmates Justin, Melissa and Harney, for bringing their views and experience together towards the completion of the project, and bearing with my from-time-to-time lacklustre performance and my comparatively lack of development experience. It is a pleasure working with you all.
# Table of Contents

Abstract ................................................................................................................................. ii

Acknowledgements ................................................................................................................ iii

List of Figures ........................................................................................................................ vi

List of Tables ........................................................................................................................ vii

Abbreviations ......................................................................................................................... viii

1 Introduction .......................................................................................................................... 1

1.1 Coronary Heart Disease ................................................................................................. 1

1.2 Motivation ........................................................................................................................ 1

1.3 Objective and Deliverables ............................................................................................. 1

1.4 Project Contribution ........................................................................................................ 2

1.5 Report Outline .................................................................................................................. 2

2 Project Background and Literature Review ..................................................................... 3

2.1 Introduction ....................................................................................................................... 3

2.2 Currently Available Products .......................................................................................... 3

2.2.1 Coronary Heart Disease Risk Calculator by MCW .................................................... 3

2.2.2 ASCVD Risk Estimator Plus by ACC ........................................................................ 4

2.3 Comparing with our Product .......................................................................................... 5

2.3.1 Required Data Not Easily Accessible ......................................................................... 5

2.3.2 Inaccuracies from Algorithms .................................................................................. 5

2.3.3 Web-only Calculators ................................................................................................ 5

3 Methodology ........................................................................................................................ 6

3.1 Introduction ....................................................................................................................... 6

3.2 Frontend: React Native .................................................................................................... 6
3.3 Backend Technologies .......................................................................................... 7
  3.3.1 Firebase ........................................................................................................ 7
  3.3.2 Python Flask .................................................................................................. 7
3.4 Wearables: Fitbit .................................................................................................. 7
3.5 System Architecture ......................................................................................... 8
  3.5.1 Account Creation .......................................................................................... 8
  3.5.2 Fitbit Data Retrieval .................................................................................... 8
  3.5.3 Risk Calculation ............................................................................................. 8
3.6 Database Schema ...................................................................................................... 9
3.7 Summary ...................................................................................................................... 9

4 End Product .............................................................................................................. 10
  4.1 Introduction .......................................................................................................... 10
  4.2 Frontend .................................................................................................................. 10
    4.2.1 Login and Account Creation .......................................................................... 11
    4.2.2 Homepage ....................................................................................................... 12
    4.2.3 Risk and Community Tabs ............................................................................. 13
  4.3 Backend ..................................................................................................................... 14
    4.3.1 Swagger UI ..................................................................................................... 14
    4.3.2 API Endpoints ................................................................................................. 15
    4.3.3 Database ......................................................................................................... 17

5 Development Process ............................................................................................... 18
  5.1 Introduction .......................................................................................................... 18
  5.2 Development Schedule ....................................................................................... 18
  5.3 Work Distribution ................................................................................................. 19

6 Conclusion and Future Plans .................................................................................... 20
6.1 Introduction ........................................................................................................................................... 20
6.2 Difficulties Faced ................................................................................................................................... 20
  6.2.1 API limitations ................................................................................................................................. 20
  6.2.2 Uncertainties during development .................................................................................................... 21
6.3 Learning Points from the Project .......................................................................................................... 21
6.4 Future Plans ........................................................................................................................................... 21
6.5 Conclusion ............................................................................................................................................ 21
References ........................................................................................................................................................ 22

List of Figures

Figure 1.1. Initial mock-up of our app ........................................................................................................... 2
Figure 2.1. Screenshot of the Coronary Heart Disease Risk Calculator ......................................................... 3
Figure 2.2. Score system of the Framingham Point System ........................................................................... 4
Figure 2.3. Screenshot of the ASCVD Risk Estimator Plus ......................................................................... 4
Figure 2.4. Warning shown on the site when “Other” is chosen as the user’s race ....................................... 5
Figure 3.1. Initial app architecture .............................................................................................................. 8
Figure 3.2. Proposed Database Schema ..................................................................................................... 9
Figure 4.1 Structure of Frontend ................................................................................................................ 10
Figure 4.2 Login Screen .............................................................................................................................. 11
Figure 4.3 Signup Success Screen ............................................................................................................ 11
Figure 4.4 Personal Info Form .................................................................................................................... 12
Figure 4.5 Connecting with Fitbit .............................................................................................................. 12
Figure 4.6 Homepage ................................................................................................................................ 12
List of Tables

Table 2.1. Differences between using Flutter and React Native.................................................. 6
Table 4.1 Endpoints for /user.................................................................................................. 15
Table 4.2 Endpoints for /risk .................................................................................................. 16
Table 4.3 Endpoints for /community ....................................................................................... 16
Table 4.4 Endpoints for /fitbit................................................................................................. 17
Table 5.1. Project Development Schedule............................................................................... 18
Table 5.2 Work Distribution among Team Members............................................................. 19
# Abbreviations

<table>
<thead>
<tr>
<th>ACC</th>
<th>American College of Cardiology</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BMI</td>
<td>Body Mass Index</td>
</tr>
<tr>
<td>CHD</td>
<td>Coronary Heart Disease</td>
</tr>
<tr>
<td>ENMO</td>
<td>Euclidean Norm Minus One</td>
</tr>
<tr>
<td>iOS</td>
<td>iPhone OS</td>
</tr>
<tr>
<td>MCW</td>
<td>Medical College of Wisconsin</td>
</tr>
<tr>
<td>RA</td>
<td>Research Assistant</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Coronary Heart Disease

Chronic diseases have become more prevalent in recent years, causing more and more people’s health conditions being affected by these diseases across the globe. Among them are heart diseases, which are caused by cholesterol being deposited and accumulates inside the walls of arteries. This would narrow, or even in severe cases block the arteries, reducing its capabilities of carrying oxygen and nutrients to various parts of the body.

Our project focuses on coronary heart disease (CHD). Also known as ischaemic heart disease or coronary artery disease, the disease accounted for 58.5% of all heart diseases related deaths in Hong Kong during the year 2020, the most out of heart diseases [1]. There has also been an observable trend showing that the age of population being affected by CHD has been decreasing continuously in recent years. This in turn could lead to more of the population having their quality of life worsened due to increased medical expenses, and would also introduce more burden onto the public health system.

Heart diseases are often dubbed the “silent killer” among chronic diseases, as it is not easy for one to check their own heart’s conditions by themselves. Although online risks calculators from different academic societies are available for public use, they often require medical readings for calculation, and the results are ethnicity-based, making the resource infeasible for the general public.

With the series of algorithms developed by Dr. Youngwon Kim and his team from the School of Public Health, risk calculation no longer heavily depends on sophisticated data that are only obtainable with the help of medical professionals and equipment, leading to our collaboration.

1.2 Motivation

Although many factors contribute to the risk of developing CHD, some of them are related to one’s lifestyle habits. Therefore, we hope to spread awareness to the general public through the project by getting more people to be concerned on their own health, and take various actions to reduce their own risks.

1.3 Objective and Deliverables

Upon completion of our project, our app aims to deliver an easy-to-use and user-friendly platform with the goal to allow users of all ages to monitor their own risks to CHD, and take preventive action from our app’s suggestions to reduce it. The CHD risk will be calculated based on user-input data and activity data obtained from the Fitbit device, with the results categorized and displayed in different tabs of the app.
1.4 Project Contribution

Through this project, we hope to raise public awareness on their own CHD risks. With the disease affecting more and more of the population, it is better for individuals to act before the situation gets out of hand unnoticed. Requiring fewer specific data from the user, this project would allow for users to assess their own risks anytime, anywhere. They will also be able to see the risk trend within the app, which can become additional motivation for the users to make changes to their current lifestyles for their own health.

Reducing the number of affected people would also be beneficial to the medical system, as it can reduce the amount of medical expenditure spend by both the individuals and the government. The saved budget can then be used on improving other aspects of one’s quality of life, all while your health is improving.

1.5 Report Outline

This report consists of two main parts. Firstly, the methodology and technology used for this project will be explained, along with justification for why we have made certain choices. Then, the current status of the project will be mentioned, before reaching a conclusion to this report.
2 Project Background and Literature Review

2.1 Introduction

In this chapter, we will take a look into two currently available products in the public. Section 2.2 would talk about how they work, and comparisons would be made with our proposed product in section 2.3.

2.2 Currently Available Products

2.2.1 Coronary Heart Disease Risk Calculator by MCW

The first product, hosted on the Medical College of Wisconsin (MCW) website, is a web-based calculator [2]. Based off the Framingham Point system derived from the Framingham Heart Study [3], users would have to input their gender, age, smoking status (yes or no), as well as their blood pressure, total and HDL cholesterol levels for calculation.

Figure 2.1 Screenshot of the Coronary Heart Disease Risk Calculator

Then, based on the inputs, the Framingham point system would assign point values to each criterion. The summed point values would correspond to the user’s 10-year risk value.
2.2.2 ASCVD Risk Estimator Plus by ACC

The second product, created by the American College of Cardiology (ACC), is also a web-based calculator. It requires the user to input their race, systolic and diastolic blood pressure, HDL and LDL cholesterol levels, as well as if they are currently suffering from any medical conditions or using any medical drugs to control their conditions.

Unlike the first calculator, the methodology behind this calculator was not disclosed. Therefore, we cannot take a look into how it works.
2.3 Comparing with our Product

Compared with our proposed product, there are 3 major disadvantages that exist in the currently publicly-available risk calculators. We will be discussing them in the below sections.

2.3.1 Required Data Not Easily Accessible

Both calculators mentioned inside section 2.2 requires the user to input various data that can only be obtained with the assistance of medical professionals. For example, a blood test called the lipid profile would have to be conducted by a doctor in order to obtain one’s cholesterol levels. Blood pressure readings might be easier to obtain due to monitors being easily obtainable, but they can only get their reading frequently if they have a monitor. These factors combined makes the currently available solutions to require a convoluted process, but the end result would not be the most precise result reflecting the user’s current risk level due to time constraints.

2.3.2 Inaccuracies from Algorithms

The ASCVD Risk Estimator Plus calculator can only provide accurate results for White and African American people, which was made clear inside of the calculator when the user proceeds to select “Other”. This means that it would limit the number of users that it can serve as a precise indicator.

Figure 2.4 Warning shown on the site when “Other” is chosen as the user’s race

2.3.3 Web-only Calculators

Both of the mentioned calculators are web-apps. Although they do have specific mobile UI for mobile devices, it is still inconvenient for users having to access the website each time they would have to calculate their risk, especially for elderly users who might not remember how to get back to the site. Having a specific app for this purpose would make the service much more accessible, as well as being a better platform to spread among the general public.
3 Methodology

3.1 Introduction

This chapter shows the proposed implementation of our project. The platforms that we are using for the frontend and backend, along with justification as to what factors affected our final decisions, would be brought up in section 2.2 and 2.3 respectively. Section 2.4 will show the architecture of the app, as well as outlining functions that keep the app running.

3.2 Frontend: React Native

When deciding the framework to be used for building the app, Flutter and React Native were brought into discussion, as both frameworks are capable of being deployed into both iOS and Android. After taking a look into the various strong and weak points of the two frameworks, we have decided to use React Native. Although Flutter-based apps will have better performance during runtime, the functionality of our current design of the app will not be affected dramatically by the difference. Besides, large companies like Bloomberg, Facebook, Walmart, as well as popular apps like Discord and Uber Eats, are already using React Native to power their apps, proving the framework’s versatility and stability. Finally, since some of our group members have previously worked with React, they will be comfortable in developing in React Native as they share the same syntax, allowing us to spend the time saved on learning the framework on other parts of development.

<table>
<thead>
<tr>
<th></th>
<th>Flutter</th>
<th>React Native</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported Libraries</td>
<td>Fewer</td>
<td>More</td>
</tr>
<tr>
<td>Documentation</td>
<td>Detailed</td>
<td>Detailed</td>
</tr>
<tr>
<td>App Performance</td>
<td>Better</td>
<td>Worse</td>
</tr>
<tr>
<td>App Size</td>
<td>Larger</td>
<td>Smaller</td>
</tr>
<tr>
<td>Prior Experience</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
3.3 Backend Technologies

3.3.1 Firebase

In order to store various forms of data, a database is for the app’s operation. For this, our app would be using Firebase, a free cloud hosting service created by Google that offers a multitude of features and rich documentation, beneficial to our project.

To begin with, Firebase offers NoSQL database that can be used to store user data. NoSQL is easier for our team to set up compared to traditional SQL, and enables us to make any necessary changes to the structure down the road conveniently. NoSQL also performs well with simple queries, which would be fit most of our needs for the app.

Firebase also has in-built authentication, an important feature for account logins and data entry and retrieval. This also makes it easy for our team to implement account creation and login without the need of further comparing different options, as well as ensuring that users’ personal data would be stored securely.

Finally, Firebase has its own dedicated analytics features all free of charge. This feature would be useful for later stages of the app, allowing us to check on the backend status all in one place. We can also use it reviewing changes and updates, as it can more or less be reflected from the change of user activities.

3.3.2 Python Flask

Besides on being able to host and process our business logic, we would have to implement our own API such that the frontend and backend can communicate with each other. To do this, we would be using Flask, a commonly-used Python-based framework used for building APIs. Along with the use of Flask-RESTx, an extension used for building RESTful API, it make sures that our endpoints would be consistent, while also allowing us to give custom, more informative error messages for debugging during development, with auto-generated Swagger UI documentation from the platform being a major plus for our development process.

3.4 Wearables: Fitbit

As mentioned in section 1.3, the user’s CHD risk based on data from both user inputs and the wearable device. After discussions within the team and with Dr. Kim’s team, we have decided to use Fitbit as our wearable of choice.

With the use of Fitbit’s API, our app would be able to access the user’s activity records. When a user creates an account on our app, they would be prompted to link their Fitbit account with the app. Since the API utilizes OAuth2 to allow for communication between third-party applications and the Fitbit side, it allows our app to get only the data that we need without compromising the security of one’s Fitbit account.
After that, the application would be able to retrieve the required data on a regular basis such that risk calculation can be done and kept up to date. Past records would also be stored in the database, such that we would be able to show the trend of the user’s risk throughout a period of time for the user to benchmark their efforts.

3.5 System Architecture

The app’s structure and main functions can be roughly divided into three parts, can be seen in Fig. 2.1 and sections below.

Figure 3.1. Initial app architecture

3.5.1 Account Creation

When the user has just started using the app, they would be prompted to create a new account, or log into an existing account. In the case of account creation, users would be asked to fill in various data, including their age, gender, BMI, lifestyle habits and medical conditions. They would also be asked to create their account credentials, along with agreeing to link their Fitbit account to the app. After collecting all these information, the app would then send the data to the Firebase server, where it would be stored for later use.

3.5.2 Fitbit Data Retrieval

In order to calculate the user’s current CHD risk, we would have to call the Fitbit API and get the vital readings and activity data of users periodically. This will be done by the frontend from time to time automatically. Upon retrieval, the data would also be sent to the Firebase server and stored alongside the user input data.

3.5.3 Risk Calculation

The risk calculation would be done by the Firebase server upon request from the frontend, using the stored data collected from sections 2.4.1 and 2.4.2. The calculated results would then be sent back to the frontend, where it would be displayed to the user accordingly.
3.6 Database Schema

In order to store and retrieve data from different users, the data would have to be stored systematically within the database. To achieve this, the database would require all the necessary fields to store the user’s data. Since we will be showing the risk trend inside our app, it is important to store data from previous timeframes along with the user’s current data. All data from previous timeframes would have the column ‘id’ in each entry, which acts as the key for users to retrieve their past records. The proposed database schema can be seen in Figure 2.2. below.

Figure 3.2. Proposed Database Schema

```typescript
User: {
  id: String -> key
  name: {
    first_name: String
    last_name: String
  }
  birth_date: Date
  gender: String (should have an array of options later)
  health_data: {
    smoking_status: String (should have an array of options later)
    height: Float
    weight: Float
    alcohol_consumption: String (should have an array of options later)
    cholesterol_status: Boolean
    diabetes_status: Boolean
    blood_pressure_status: Boolean
    medication: ArrayStrings (should have an array of options later)
    dietary_intake: Int (based on reported food consumed)
  }
  risk_prediction: (subcollection)
}
*calculate BMI with height and weight
*calculate age from birth date

RiskPrediction: {
  id: String -> key
  results: Float
  date: Date
  height: Float
  weight: Float
  alcohol_consumption: String (should have an array of options later)
  cholesterol_status: Boolean
  diabetes_status: Boolean
  blood_pressure_status: Boolean
  medication: ArrayStrings (should have an array of options later)
  dietary_intake: Int (based on reported food consumed)
  ...other_factors
}
```

3.7 Summary

This chapter covered our choice of technologies that we will be using to deliver our project. Reasons for choosing React Native and Firebase were listed, and the design for our system architecture and database schema were shown. In the following chapter, we will be showing our current progress.
4 End Product

4.1 Introduction

This chapter outlines the final deliverables of the project. Section 3.2 and 3.3 would show screenshots of the end products of frontend and backend respectively.

4.2 Frontend

The frontend portion of the report would show screenshots of the application.

During development, our team utilized Expo to help streamline the development and testing process, as it allows for the app to be tested with the use of both emulators and the devices. As for the actual app, we have used external packages like Victory React Native and Axios alongside React Native to do various functions like graph generation and handling API calls.

We have also structured our directory to be clear such that locating files would be easy and intuitive. For example, UI components that are reused often are put in the same folder, while images are put in a separate one.

Figure 4.1 Structure of Frontend
4.2.1 Login and Account Creation

When opening the app for the first time, the user would be prompted to either login to their existing account or create a new one. Upon successful creation of a new account, the user would be shown the Success screen.

Newly created accounts would then be prompted to input their personal information like date of birth, dietary preferences and medical information. The app would also prompt the user to link their account with their Fitbit account such that information can be retrieved through the API.
4.2.2 Homepage

The homepage contains a dashboard that shows various activity data retrieved from Fitbit. The user can also access their own profile, where they can edit their own information and change their password of needed.
4.2.3 Risk and Community Tabs

With the use of the navigation tabs, users can navigate to the risk and community pages. The risk pages would show the user’s weekly and monthly risks inside a broken line graph, as well as a list containing all of their risk entries.

The community tab consists of forums themed after different health interest groups. In each forum are posts created by the admins, and users would be able to comment in each post.
4.3 Backend

This section will show screenshots of the backend API.

Utilizing the Flask API framework, the API is structured to be modular such that it can be scalable and readable even outside of the team. All API files are grouped inside of one folder such that they can be found and edited easily if needed. We have also included “requirements.txt” such that all dependencies can be dealt with conveniently during setup.

Figure 4.14 Structure of Backend

4.3.1 Swagger UI

By the user of Swagger UI, the generated API can be accessed with the use of localhost. We have added descriptions and example models such that users would know how to use each endpoint. Users can also try executing the API calls directly in the site.
4.3.2 API Endpoints

The implemented API endpoints are grouped under 4 namespaces: /user, /risk, /fitbit, and /community. When calling endpoints, our team made sure that each user should be identified with their UID instead of their username such that the calls can be consistent.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Route</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/user/</td>
<td>Returns the list of ID of users (not public visible)</td>
</tr>
<tr>
<td>POST</td>
<td>/user/</td>
<td>Creates a new user, in both Authentication and Database</td>
</tr>
<tr>
<td>GET</td>
<td>/user/&lt;uid&gt;/</td>
<td>Returns all data of the specified user with UID</td>
</tr>
<tr>
<td>DELETE</td>
<td>/user/&lt;uid&gt;/</td>
<td>Deletes the specified user with UID</td>
</tr>
<tr>
<td>GET</td>
<td>/user/info/&lt;uid&gt;/</td>
<td>Returns personal info of the specified user with UID</td>
</tr>
<tr>
<td>PUT</td>
<td>/user/info/&lt;uid&gt;/</td>
<td>Updates personal info of the specified user with UID</td>
</tr>
<tr>
<td>GET</td>
<td>/user/health/&lt;uid&gt;/</td>
<td>Returns health info of the specified user with UID</td>
</tr>
<tr>
<td>PUT</td>
<td>/user/health/&lt;uid&gt;/</td>
<td>Updates health info of the specified user with UID</td>
</tr>
<tr>
<td>GET</td>
<td>/user/name/&lt;uid&gt;/</td>
<td>Returns the username of the specified user with UID</td>
</tr>
<tr>
<td>GET</td>
<td>/user/verifyData/&lt;uid&gt;/</td>
<td>Returns whether the specified user completed sign-up</td>
</tr>
<tr>
<td>PUT</td>
<td>/user/verifyWearable/&lt;uid&gt;/</td>
<td>Returns whether the specified user connected Fitbit</td>
</tr>
</tbody>
</table>
The risk algorithm provided by Dr. Kim and his team has been implemented into the backend. In our current version, the algorithm takes the total amount of calories burnt by the user from the Fitbit, which is then converted into the necessary ENMO values needed to be plug into the algorithm.

Table 4.2 Endpoints for /risk

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Route</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/risk/&lt;uid&gt;/</td>
<td>Returns the overall risk of specified user</td>
</tr>
<tr>
<td>GET</td>
<td>/risk/&lt;uid&gt;/&lt;date&gt;/</td>
<td>Returns the overall risk of user on the specified date</td>
</tr>
<tr>
<td>POST</td>
<td>/risk/&lt;uid&gt;/&lt;date&gt;/</td>
<td>Creates new risk data of user on the specified date</td>
</tr>
<tr>
<td>GET</td>
<td>/risk/weekly/&lt;uid&gt;/</td>
<td>Returns the risk of specified user in the nearest week</td>
</tr>
<tr>
<td>GET</td>
<td>/risk/monthly/&lt;uid&gt;/</td>
<td>Returns the risk of specified user in the nearest month</td>
</tr>
<tr>
<td>GET</td>
<td>/risk/yearly/&lt;uid&gt;/&lt;year&gt;/</td>
<td>Returns the risk of specified user in the specified year</td>
</tr>
</tbody>
</table>

Table 4.3 Endpoints for /community

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Route</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/community/</td>
<td>Returns the list of existing forums (communities)</td>
</tr>
<tr>
<td>POST</td>
<td>/community/</td>
<td>Creates a new forum</td>
</tr>
<tr>
<td>GET</td>
<td>/community/&lt;forumID&gt;/</td>
<td>Returns data of the specified forum with forum ID</td>
</tr>
<tr>
<td>POST</td>
<td>/community/&lt;forumID&gt;/</td>
<td>Creates a new post under the specified forum</td>
</tr>
<tr>
<td>DELETE</td>
<td>/community/&lt;forumID&gt;/</td>
<td>Deletes the specified forum</td>
</tr>
<tr>
<td>GET</td>
<td>/community/post/&lt;postID&gt;/</td>
<td>Returns data of the specified post with post ID</td>
</tr>
<tr>
<td>POST</td>
<td>/community/post/&lt;postID&gt;/</td>
<td>Creates a new comment under the specified post</td>
</tr>
<tr>
<td>DELETE</td>
<td>/community/post/&lt;postID&gt;/</td>
<td>Deletes the specified post</td>
</tr>
</tbody>
</table>

The Fitbit API has a rate limit of 150 calls per hour for each user. Since a number of our endpoints would be called on regular time intervals, we will have to make sure that the total number of calls needed would not exceed the limit when combined while maintaining the data to be as up-to-date as possible.
Table 4.4 Endpoints for /fitbit

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Route</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/fitbit/auth/url/&lt;uid&gt;/</td>
<td>Returns a redirect URL for Fitbit user authentication</td>
</tr>
<tr>
<td>GET</td>
<td>/fitbit/auth/token/</td>
<td>Returns the tokens required to fetch Fitbit data</td>
</tr>
<tr>
<td>POST</td>
<td>/fitbit/user/&lt;uid&gt;/</td>
<td>Returns user’s profile on Fitbit</td>
</tr>
<tr>
<td>GET</td>
<td>/fitbit/activities/&lt;uid&gt;/</td>
<td>Returns user’s daily activities summary on Fitbit</td>
</tr>
<tr>
<td>GET</td>
<td>/fitbit/weeklyAvgCal/&lt;uid&gt;/&lt;date&gt;/</td>
<td>Returns user’s weekly average burnt Calories</td>
</tr>
<tr>
<td>GET</td>
<td>/fitbit/weeklySteps/&lt;uid&gt;/</td>
<td>Returns number of user’s weekly steps on Fitbit</td>
</tr>
<tr>
<td>PUT</td>
<td>/fitbit/storeToken/&lt;uid&gt;/</td>
<td>Updates user’s authorization token on database</td>
</tr>
</tbody>
</table>

4.3.3 Database

As mentioned in section 3.3.1, we chose to use Firebase for its NoSQL database structure allowing for easier changes during development if needed. However, a finalized structure is still necessary to ensure that data stored is consistent after our product’s deployment. Shown below is an example of user entry currently being stored in our final data structure. The user’s login credentials are not stored as the login process is handled by Firebase’s inbuilt authentication. This allows us to ensure security and privacy of users’ data are being upheld.

Figure 4.17 Example stored in finalized structure
5 Development Process

5.1 Introduction

This chapter would outline our development timeline, as well as showing the contributions for each member.

5.2 Development Schedule

<table>
<thead>
<tr>
<th>Table 5.1. Project Development Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct 2022</td>
</tr>
<tr>
<td>- Detailed project plan</td>
</tr>
<tr>
<td>- Project web page</td>
</tr>
<tr>
<td>- Finalized app architecture</td>
</tr>
<tr>
<td>- Finalized UI design</td>
</tr>
<tr>
<td>- Set up database</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Nov 2022</td>
</tr>
<tr>
<td>- Complete basic UI (Front-end)</td>
</tr>
<tr>
<td>- Implement user authentication</td>
</tr>
<tr>
<td>- Finalize database schema</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Dec 2022</td>
</tr>
<tr>
<td>- Disease risk prediction algorithm implementation</td>
</tr>
<tr>
<td>- Fitbit API integration</td>
</tr>
<tr>
<td>- Complete basic app features (e.g., account creation, user data input)</td>
</tr>
<tr>
<td>- Create application hosting container in Firebase</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Jan 2023</td>
</tr>
<tr>
<td>- First presentation</td>
</tr>
<tr>
<td>- Preliminary implementation</td>
</tr>
<tr>
<td>- Detailed Interim report</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Feb 2023</td>
</tr>
<tr>
<td>- Additional features implementation (e.g., health article page, personalized health tips)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Mar 2023</td>
</tr>
<tr>
<td>- Finalized web platform</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
19

April 2023
- Finalized project web page
- Finalized tested implementation
- Final report
- Final presentation

In progress

May 2023
- Project exhibition

Incomplete

5.3 Work Distribution

Table 5.2 Work Distribution among Team Members

<table>
<thead>
<tr>
<th>Frontend</th>
<th>Backend</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Implementation, Fitbit Connection</td>
<td>Vieri, Mellisa</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Backend</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/user and /community</td>
<td>Justin</td>
<td></td>
</tr>
<tr>
<td>/risk</td>
<td>Hayward</td>
<td></td>
</tr>
<tr>
<td>/fitbit</td>
<td>Vieri, Mellisa</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Database</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Firestore and authentication management</td>
<td>Justin</td>
<td></td>
</tr>
<tr>
<td>Database Design</td>
<td>Justin, Hayward</td>
<td></td>
</tr>
</tbody>
</table>
6 Conclusion and Future Plans

6.1 Introduction

This chapter will round up the whole Final Year Project. Section 6.2 will mention on some of the challenges we have faced during development. Some good learning points would also be mentioned in Section 6.3. Future possible development directions would be talked about in section 6.4, and the final conclusion would be reached in Section 6.5.

6.2 Difficulties Faced

6.2.1 API limitations

While exploring the possibilities of the Fitbit API, we noticed that there was a rate limit for the amount of calls each user can do per hour. Therefore, we had to consider how often should these calls be made such that the functionality of the app can be maintained while still providing the most up-to-date data for the user.

During the implementation of calculating ENMO, our original plan was to use the Fitbit API to fetch the raw accelerometer data for the Fitbit. However, we soon realized that the data was only available through the device API instead of the web API that we were using, and would require us to make an app for the Fitbit device just to fetch one data, ruining the app’s user experience. Therefore, we have to opt for a different solution that can work seamlessly enough while getting the necessary data that we need.
6.2.2 Uncertainties during development

A smaller difficulty is actually from our collaborators Dr. Kim and his team. We understand that their team are busy with academic research, but there were definitely times where more efficient and effective communication between the two parties would have been beneficial to the whole project. For example, our team has only been able to get our hands on the risk calculation algorithm on February, only to find out that the algorithm requires for us to calculate for the user’s ENMO value. This took us additional effort and time spent on researching how the value can be obtained. If Dr. Kim could have mentioned the necessity of the ENMO value in previous exchanges, it would have saved us quite some time.

6.3 Learning Points from the Project

At the beginning of semester 2, our team has decided to have weekly meetings on campus. This allowed us to show what we have completed during the week, discuss about upcoming tasks and ways of implementation, as well as setting goals to be achieved before next meeting. This promoted effective communication among the team, and we were able to voice concerns and give suggestions to our development direction, which was very beneficial for the whole project.

6.4 Future Plans

Our product actually has one planned feature missing in our final deliverables. Originally, Dr. Kim’s team has expressed on hoping to implement direct-to-customer (DTC) genetic tests into the app, such that users can also connect their DTC accounts to the app. However, we realized that the application deadline for DTC academic programs, which grants access to their API, has already passed, leaving us with no choice but to skip its implementation. It would be great if we can try implementing this feature in the future.

Also, our app only provides compatibility with Fitbit wearables. We hope to expand the line-up of compatible wearables in the future to include brands like Garmin and Apple such that more users can be benefitted by the app.

6.5 Conclusion

As the saying goes, “prevention is better than cure”. With CHD becoming a threat to an increasing number within the population, reducing one’s risk of developing the disease is crucial for both the individual and society. With Dr. Kim’s newly developed risk assessment algorithms, our project aims to raise public awareness towards the issue, as well as making it easier to monitor and reduce one’s risk, such that everyone can be more concerned about their own health, and be able to live their life more fruitfully.
References


