GNG2101
Deliverable J- User Manual

Trackbite

Submitted by

TEAM IDENTIFIER E21

Marco Vethanayagam, 300121798

Andy Ung, 300117710

Michael Bosciglio 300135179

Farid Nastaran Tehrani, 300119183

Fred Wei, 300131144

David Ajayi 300102740

December 10, 2020

University of Ottawa

Abstract

In this document the functionality and work which was done on Trackbite will be displayed. TrackBite is an android application that allows users to input, track and analyze their meal and exercise data. The application was produced based on a request made by a diabetic patient who was tired of writing down every meal's nutritional facts on paper. The application was started based on initial concepts or functionalities which were deemed necessary for meeting the needs of the client, moved on to design of pages, and finally the programming of the application based on the designs. The application was produced on Android studio using Java programming language. In between each prototype the team would have a meeting with the client to ensure that her needs were met. As a result of these meetings numerous functionalities were cut out of the application and also many were added. The application's main goal was to make sure it was easy to use and cut straight to the functionalities that were needed as this was one of the most important client needs, since the client found other applications out there complicated to use.

List of Acronyms

Acronym	Definition
UI	User Interface

List of Figures

---

1. Introduction

According to Diabetes Canada, one in three Canadians is living with diabetes or prediabetes [1]. With the continuous rise of sugary foods and inactive living it is now more important than ever to be able to track and improve habits. The application aims to fight this problem by helping diabetic patients track and analyze their food intake and exercise. The application has three main pathways, input of data, database for food input, and data analysis of food and exercise. The application allows the user to input a food item, enter the amount consumed, and analyze each or numerous nutrients based on a chosen time frame. This can also be done with exercises. By doing so the user can review a trend in their eating or exercise habits and improve upon it based on their needs. What sets this application apart from others is that it is simple to navigate so that even the most technologically novice users can easily use the application. Since the majority of deiebetic patients are above the age of 50, the simplicity of this application becomes more attractive to users than already existing competition [2].

2. Main Body

2.1 Application Functionalities

Before using any of the application's functionalities, the user needs to make an account. Each user has unique values relating to their height, weight, age and sex, and its important to store these per user account as these values are used to calculate the calories burned during an exercise. When opening the application, the user is on the signin page as seen in *Figure 13*. The user can click the *Register* to make a new account or type their information to login to a pre-existing account. Clicking *Register* shows inputs for email, username, and password, which are required to make a new account. This is seen in *Figure 14*. After making a new account, the user can then login with that information and proceed to the main functionality of the application.

The application has four main pathways that the user can progress through from the main page. The first one is the *User Profile*. As shown in *Figure 3*, *User Profile* allows the user to input their age, height and weight. One thing to take note is that when logging into a new account, the user is first directed to the profile page, and then redirected to the main page. The user profile will be saved and can be changed whenever decided by the user. The purpose of having a user profile is to ensure that the user can update their values whenever they change as they impact the caloric calculations made during analysis.

The second pathway from the main page is *My Meal*. If the user clicks the *Canadian Food Guide* button, they are taken to a page with a visual representation of the Canadian Food Guide, and are able to interact with it. The user can also click the *Current Intake* button which shows the current intake (in grams) of each major nutrient for that day. Finally, clicking *Enter a Meal* takes the user to a new page that gives them 3 options to enter a meal: *Simple Meal Entry*, *Detailed Meal Entry*, or *My Previous Meals*. The *Simple Meal Entry*, as shown in *Figure 4,* allows the user to input the main nutrients displayed on nutrition facts, such as carbohydrates, fats, sugar, and so on. This would usually be used when the food input has no explicit nutrition facts. The *Detailed Meal Entry*, as shown in *Figure 5*, allows the user to input nearly every nutrient found in food items including vitamins. This feature comes in handy when inputting an item which has a nutrition fact and is not based on estimates. After these food items are inputted, the food name along with its nutritional facts are saved to a database. In addition, the user is given the opportunity to add said inputted food into a local database known as *My Previous Meals*. The local database's purpose is for the user to add food that is commonly consumed, as they can later use this database as a form of food input. *Figure 7* shows the *My Previous Meals* database and *Figure 8* shows the UI that appears when clicking on a food in the database. As seen, they can modify the serving size and input a time frame for the food which allows the user to input previously saved meals quickly.

The third pathway from the mainpage is inputting exercise data. In this pathway, the user can select the kind of exercise they have done. This can be of type well-being or cardio exercises. In the cardio exercise, the user will be asked to pick the type of activity from walking, running and biking. After an exercise has been chosen, the user must input the duration of the activity, while also picking the intensity associated with the exercise as shown in *Figure 9*. The user does have the option to go back in case they did misclick or change their mind. If they continued, based on their choices with the addition of their personal profile, an approximation of calories would be calculated and then displayed to the user. This result would be saved to the database which could be used for the analysis function of the app. Similarly to cardio exercise, well-being functions in a similar fashion. Instead of providing the intensity, the yoga or dance type accounts for the intensity for the approximation. The user follows a similar approach and selects an exercise from the assortment and provides the duration. Similarly, the activity and calories burned will be shown and updated to the database. The user simply will be able to click “Done”, and will be redirected to the main page.

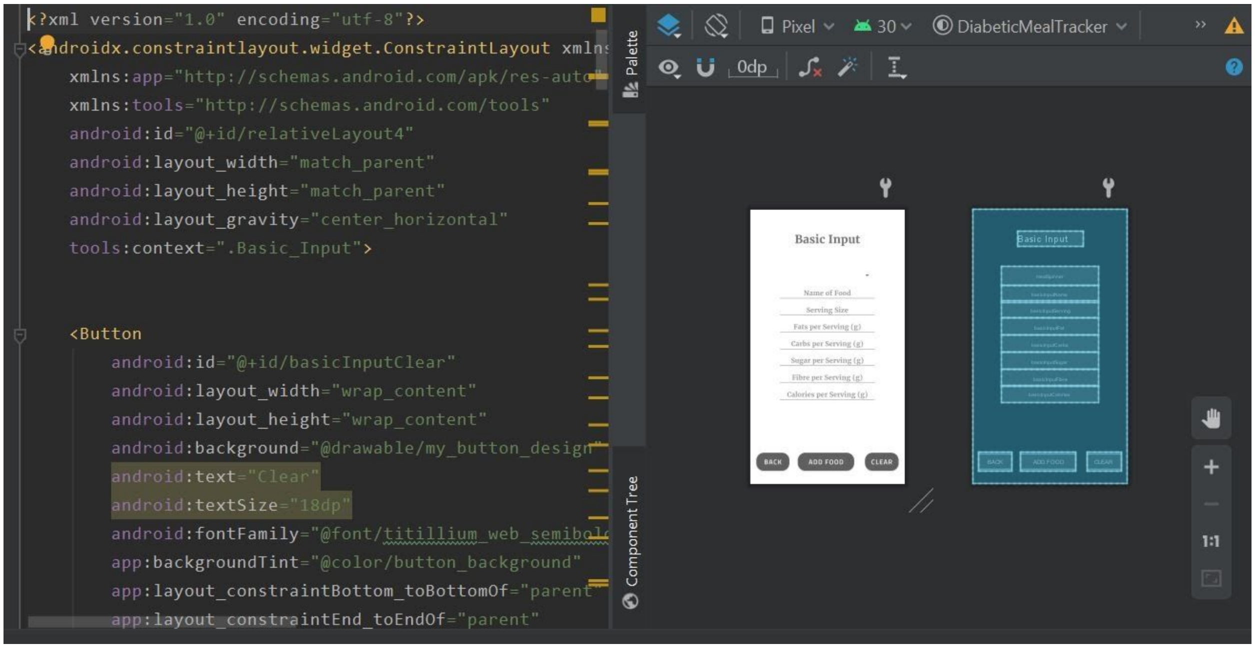
The application's fourth and final pathway is *Analyse My Data*. In this pathway, all the user’s meals consumed and exercises completed can be analyzed. The user has two options: *Exercise Data Analysis* or *Food Data Analysis*. If the user selects *Exercise Data Analysis*, they are asked to select the type of data to be analyzed as well as the time frame as shown in *Figure 10*. In addition, the user can also choose which type of graph they would like, between a Pie Chart and Line Graph. If the user chooses *Food Data Analysis* they will be asked which nutrient(s) they would like analysed as well as the time frame as shown in *Figure 15*. Both of these options also allow the user to type in a custom date in the form of *ddmmyyyy*. Once the user completes the fields shown in any of the data analysis pathways, the application will create the selected graph as shown in *Figure 11 and 12* respectively. The user can then save these by screenshotting their phone for future reference.

2.2 Prototype Details

The final prototype of the android application “Trackbite” was made using Android Studio and Java. Before the final prototype, the first two prototypes were planned and designed using google slides and Balsamiq. The prototypes provided an idea of how many buttons and fields were in each screen. They also helped to create the structure of the prototype and to visualize the connections between the buttons and the navigation between the screens. After the first two prototypes were completed, the first stage of producing Trackbite was creating the basic UI (user interface) on Android Studio. After the pages obtained functionality, the UI was updated for all the pages to have a uniform font and color palette (black, white, and grey). The next step was to set up Firebase onto Android Studio. The database chosen was Cloud firestore because the queries were simpler and faster to make due to the structure of the database. Below is the breakdown of the main components of our application and it’s UI, functionality, and development process:

User Interface (UI):

The user interface was designed using Android Studio’s layout files. In Android Studio, there is a code component and design component for each layout. Below shows how the basic input layout UI was created:



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmln
    xmlns:app="http://schemas.android.com/apk/res-auto
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/relativeLayout4"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center_horizontal"
    tools:context=".Basic_Input">


    <Button
        android:id="@+id/basicInputClear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/my_button_design
        android:text="Clear"
        android:textSize="18dp"
        android:fontFamily="@font/titillium_web_semibol
        app:backgroundTint="@color/button_background"
        app:layout_constraintBottom_toBottomOf="parent
        app:layout_constraintEnd_toEndOf="parent"
```

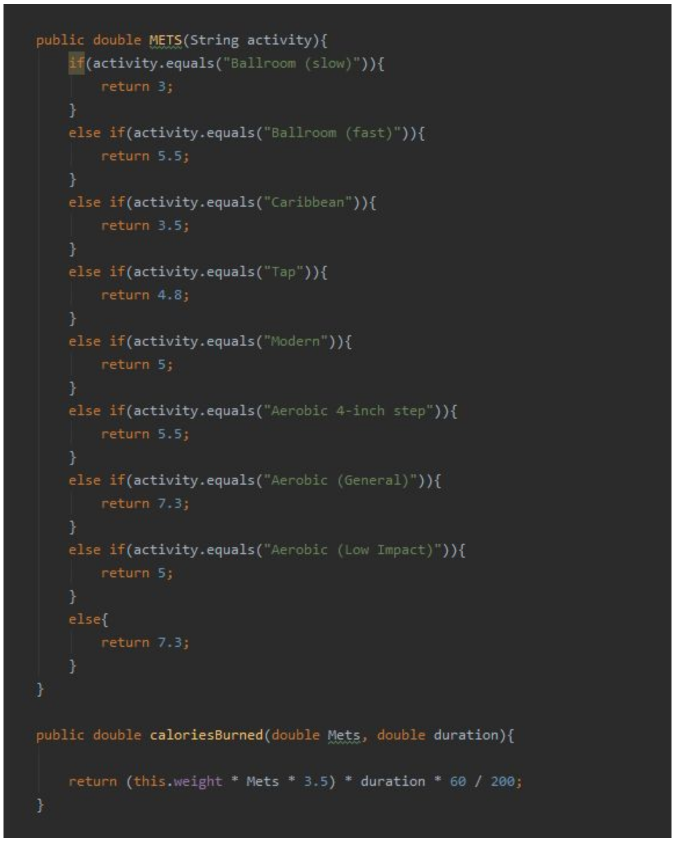
Figure 1. UI Design

On the left is the code of the layout. The code dictates what is displayed on the design. The type of layout is constraint layout which means that all the elements are positioned based on its constraints. Elements are placed into the layout (Button is an example as in the code shown above) and its attributes are written to customize the element. Buttons are also assigned an onclick function that performs a task when it is clicked. In the basic input layout, there are 3 buttons (back, add food, clear), 7 edit text fields for input, a drop down spinner for the meal, and a textview for the title.

For the UI, we attempted to create a clear, consistent design where the positioning of elements were reasonable. For example, buttons that dictated progressing backwards in the app were placed on the bottom left, whereas buttons that dictated progressing forwards were placed on the bottom right. In addition, we kept the UI as clear as possible with the exceptions of the food input page to ensure that the user is able to navigate through the app with ease. Finally, we used a consistent color and font scheme for the app so that the application element's are not distracting.

Functionality of each Page:

For each page, there exists a Java class file which is how the functionality of the app is created. These class files contain functions that establish the logic for the apps functionality, such as calculations, navigation of pages, inputting values, saving values to a database, and retrieving data from a database.

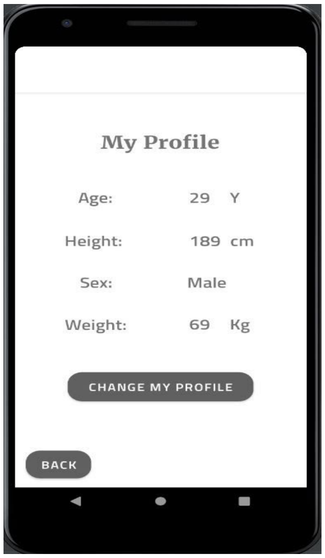
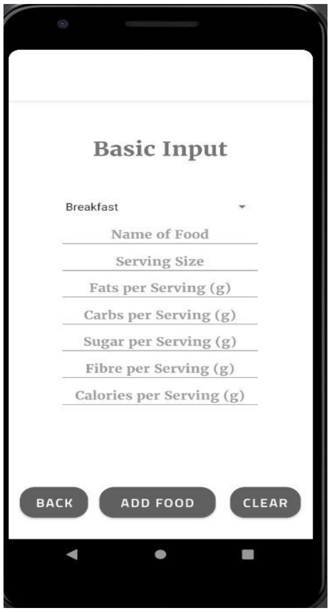


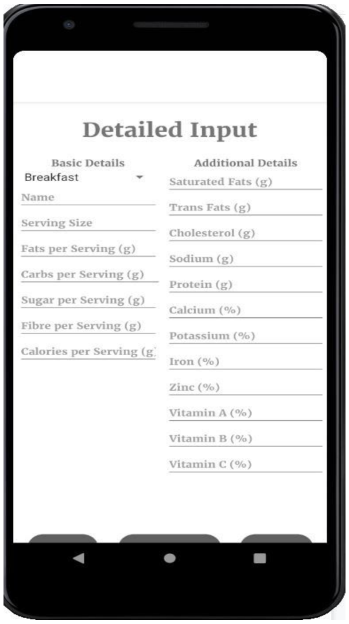
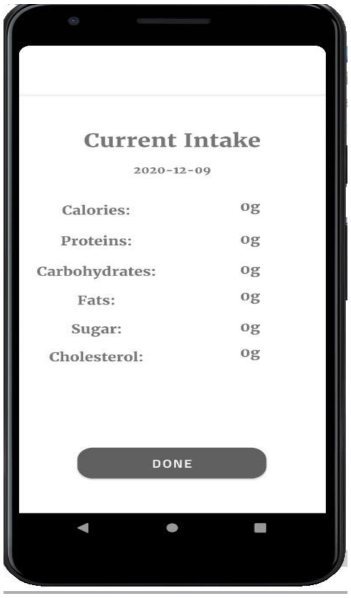
```
public double METS(String activity){
    if(activity.equals("Ballroom (slow)")){
        return 3;
    }
    else if(activity.equals("Ballroom (fast)")){
        return 5.5;
    }
    else if(activity.equals("Caribbean")){
        return 3.5;
    }
    else if(activity.equals("Tap")){
        return 4.8;
    }
    else if(activity.equals("Modern")){
        return 5;
    }
    else if(activity.equals("Aerobic 4-inch step")){
        return 5.5;
    }
    else if(activity.equals("Aerobic (General)")){
        return 7.3;
    }
    else if(activity.equals("Aerobic (Low Impact)")){
        return 5;
    }
    else{
        return 7.3;
    }
}

public double caloriesBurned(double Mets, double duration){

    return (this.weight * Mets * 3.5) * duration * 60 / 200;
}
```

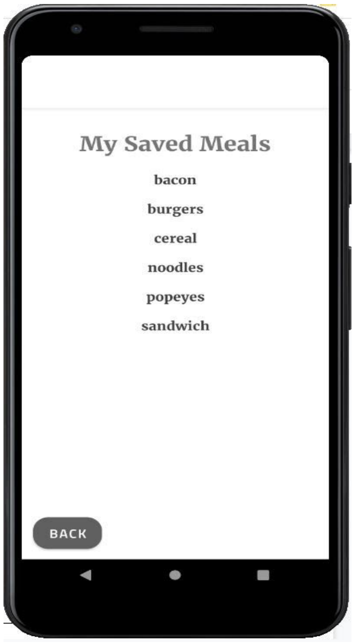
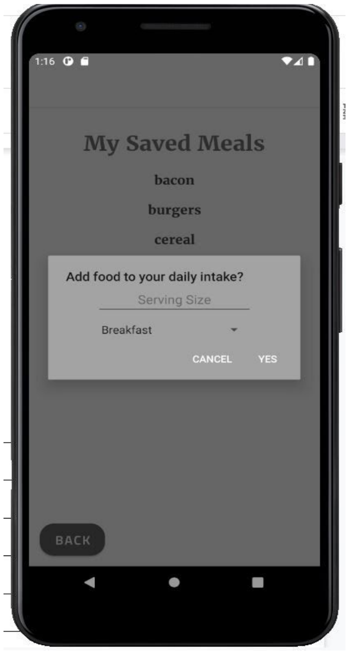
Figure 2. Java Activity File

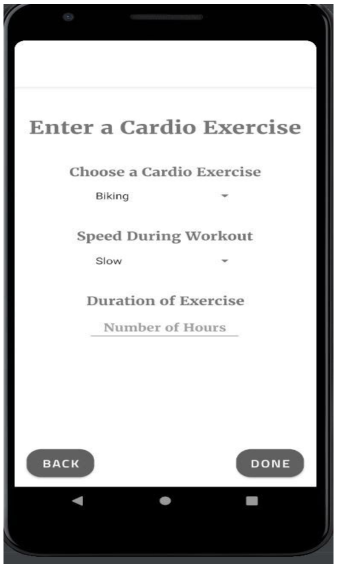
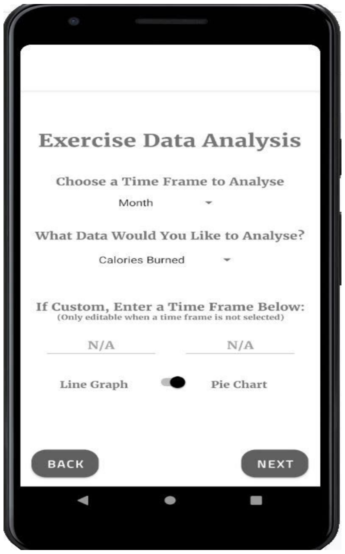
As seen in *Figure 2*, the functionality of calculating the MET value for a particular exercise and calculating the total calories burned is shown. This is one of many examples where the app's functionality is done in these Java class files. In the following table, the functionality of each page of the app can be seen.

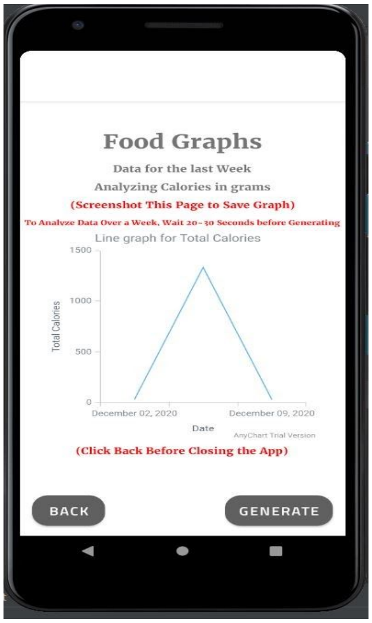
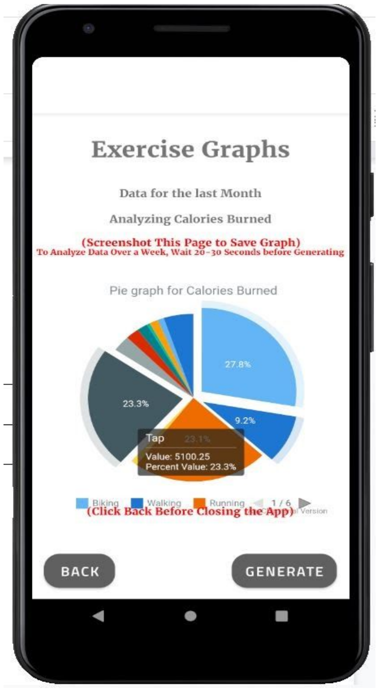
Table 1: Functionality for each page

 My Profile Age: 29 Y Height: 189 cm Sex: Male Weight: 69 Kg *Figure 3. Profile Page*	This page allows the user to input their age, height, sex, and weight. This profile is unique to each account. Accounts are created using firebase's authentication feature which allows users to register or login to an email/password account. The data in the profile page is saved to firebase and everytime the user updates their info, it overrides the data in the firebase and shows the updated data on the profile.
 Basic Input Breakfast Name of Food Serving Size Fats per Serving (g) Carbs per Serving (g) Sugar per Serving (g) Fibre per Serving (g) Calories per Serving (g) BACK ADD FOOD CLEAR *Figure 4. Basic Input*	The input fields in basic input are edit text fields on Android Studio. When the user clicks on "Add food", all the values inside the edit text fields are converted to String and sent to the daily collection section of the database. The data of each food is stored on a day by day basis and contributes to the total amount displayed in the current intake page. (Keep in mind that not all fields need to be filled in before you submit data). The "clear" button clears all of the input fields by setting the field value to "". This makes it easier for the user to input another food.

 Detailed Input Basic Details Breakfast Name Serving Size Fats per Serving (g) Carbs per Serving (g) Sugar per Serving (g) Fibre per Serving (g) Calories per Serving (g Additional Details Saturated Fats (g) Trans Fats (g) Cholesterol (g) Sodium (g) Protein (g) Calcium (%) Potassium (%) Iron (%) Zinc (%) Vitamin A (%) Vitamin B (%) Vitamin C (%) *Figure 5. Detailed Input*	The detailed input page is essentially the same as the basic input page. The difference is that the detailed input page has more input fields for other nutrients that the user may want to keep track of. (Keep in mind that not all fields need to be filled in before you submit data).
 Current Intake 2020-12-09 Calories: 0g Proteins: 0g Carbohydrates: 0g Fats: 0g Sugar: 0g Cholesterol: 0g *Figure 6. Current Intake*	When the user inputs any data, the values get added to the totals value for the current day in firebase. The values are then displayed in the current intake page.

 My Saved Meals bacon burgers cereal noodles popeyes sandwich BACK *Figure 7. Saved Meals Database*	When the user enters a food using the basic or detailed input page, a dialog will ask the user if they would like to save the food to the database. If they click yes, the food shows up as an option in the saved meals page. If the food is already in the saved meals page, a dialog will ask the user if they would like to update the nutritional data of the food in the database. This page is created using Android Studio RecyclerView and it retrieves all saved meal documents from firebase for an account and displays them in a list. Everytime a user inputs in a food, their account's saved meals collection in the firebase is checked to see if the food they inputted is already in it. A dialog is opened accordingly.
 1:16 My Saved Meals bacon burgers cereal Add food to your daily intake? Serving Size Breakfast CANCEL YES BACK *Figure 8. Dialog in the Database*	The saved meals page also allows the user to input food for the meals they saved. When the user clicks on an item in the recycler view, they receive a dialog with the options to input the serving size and meal time of the item. This feature should be used if they are having the food they saved again. Inputting data this value also contributes to the current intake for the day. The firebase calculates the additional value as (serving size * values of the saved meal) and the food is also record in the daily collection in firebase,

 Enter a Cardio Exercise Choose a Cardio Exercise Biking Speed During Workout Slow Duration of Exercise Number of Hours *Figure 9. Exercise Input*	The exercise input page works in the same way as the food input pages, but input fields are related to exercise. The data inputted are also put into a separate collection from the food inputs. After the user inputs in the hours of exercise they performed, the screen will display the number of calories burned. This total is calculated using the user profile data and the average calories burnt by the exercise.
 Exercise Data Analysis Choose a Time Frame to Analyse Month What Data Would You Like to Analyse? Calories Burned If Custom, Enter a Time Frame Below: (Only editable when a time frame is not selected) N/A N/A *Figure 10. Data Analysis Selection*	When the user goes to the data analysis selection pages, they are able to select the time frame and data they would like to display in a graph. All the date collections within the time frame are retrieved from the firebase and their information is displayed onto a graph using the AnyGraph library.

 Food Graphs Data for the last Week Analyzing Calories in grams (Screenshot This Page to Save Graph) To Analyze Data Over a Week, Wait 20-30 Seconds before Generating (Click Back Before Closing the App) Line graph for Total Calories 1500 1000 500 0 Total Calories December 02, 2020 December 09, 2020 Date AnyChart Trial Version BACK GENERATE *Figure 11. Graph of Food Input*	Example of graph for food. In the graph on the left (*Figure 11*), the line graph shown to the user has interactive elements. The graph lets the user click on data points to inspect the specific values on the graph.
 Exercise Graphs Data for the last Month Analyzing Calories Burned (Screenshot This Page to Save Graph) To Analyze Data Over a Week, Wait 20-30 Seconds before Generating (Click Back Before Closing the App) Pie graph for Calories Burned 27.8% 23.3% 9.2% Tap 23.1% Value: 5100.25 Percent Value: 23.3% Biking Walking Running 1 / 6 BACK GENERATE *Figure 12. Graph of Exercise Input*	Example of graph for exercise. In the graph on the left (*Figure 12*), the pie chart shown to the user has interactive elements. The graph lets the user click on data points to inspect the specific values on the graph, with the name and percentages from the total portion.

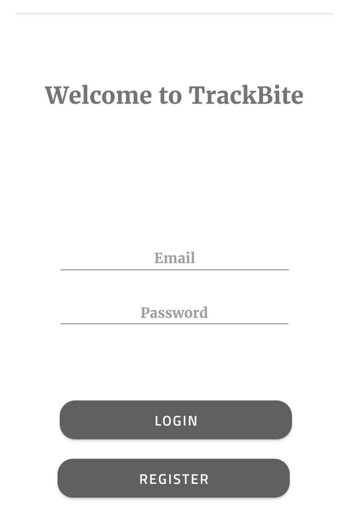
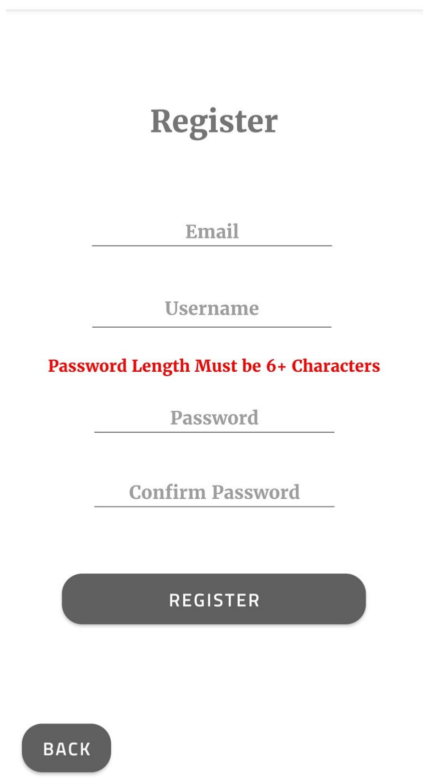


Figure 13. Login Page

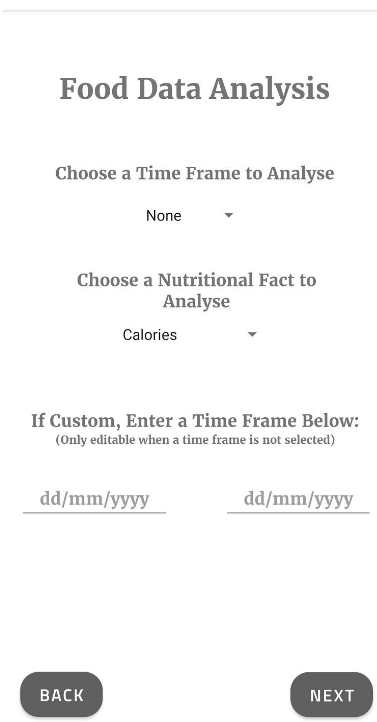
This displays the login page where the user can either log into their account or register for a new account.



Password Length Must be 6+ Characters

Figure 14. Register Page

This is the register page where the user can create an account with their email and a new username and password.

 Food Data Analysis Choose a Time Frame to Analyse None Choose a Nutritional Fact to Analyse Calories If Custom, Enter a Time Frame Below: (Only editable when a time frame is not selected) dd/mm/yyyy dd/mm/yyyy *Figure 15. Food Data Analysis Page*	The Food Data Analysis page allows users to analyze their previous meals and obtain a graphical representation of the data. The data can be analyzed based on a specific time frame or specific nutritional fact.

How to install Trackbite:

1. Connect your phone to a computer or follow the steps on your phone browser.
2. Visit the releases page on the app's github repository at

   https://github.com/Macomatic/TrackBite
3. On the latest release, select assets and download the APK onto your phone.
4. Go to the location of the APK file on your phone and install the application.
5. Open the application of your phone.

How to operate Trackbite:

1. When you first open up the app, you are greeted with a login screen.
2. Register an account using an email address and password.
3. When you complete the registration, fill out your personal details. The information will be used to calculate data for the analysis portions of the application.
4. Once everything is set up, you may visit all the pages of the application to see the features it provides.

How to maintain the application:

Developers that would like to help maintain the application should visit https://github.com/Macomatic/Diabetic-Meal-Tracker to pull the source code. New features are always welcomed. Libraries used that might need to be updated from time to time include: AnyGraph, Firebase UI, Canadian Food Guide, and RecyclerView.

2.3 Health and Safety Guidelines

The application aims to help diabetic patients maintain and track their food and exercise choices. It is important for patients to view the calculations and analysis with reservations as calculations are approximations with estimates on certain calculations.

2.4 Troubleshooting and Technical Instructions

Troubleshooting common errors:

1. Application glitches after input.

   This is most likely a null pointer error, the simple solution is to just go back to the input page and input your data again. There should be no issues afterwards. This issue was already fixed, but is something that the user should be aware of in case it occurs.

2. Graphs could not generate or crash when generated.

   This is most likely due to the lack of input data for the specified time frame. Make sure that the inputted time frame is correctly formatted and that you have inputted data during that time frame. This shouldn't happen in the newest build of TrackBite, however this is something to be aware of.

3. Conclusions and Recommendations for Future Work

3.1 Conclusion

Prior to the creation of TrackBite, our client used a sheet of paper to track her food and vitamin intake, daily. The issue was that the client wanted to analyse and refer back to previous data sets of her food intake with ease. Implications arose when considering existing apps; they had the majority of the functionality desired by the client, however lacked simplicity. TrackBite aimed to provide a simple solution to inputting and tracking one's food intake and exercise activity. By using an online database to store food and exercise inputs, integrating a simplistic UI and application design, and implementing an analysis feature, TrackBite became the ideal app for our client. Through the previously-defined manual, any user should be able to understand the functionality and flow of the application, and use it with ease.

3.2 Recommendations

After the launch of TrackBite, there are still many features that can be implemented. An additional feature that can be implemented is a reminders function. Due to time constraints, a reminders function could not be made. Currently, there is no way for the user to set goals or reminders within the app to remind them to perform an action. In the future, we plan to add a reminders function that will allow users to set reminders and goals. The reminders functionality would allow users to remind themselves to complete tasks with due dates and set importance. Goals would be implemented such that they would be present constantly until the user confirms the goal has been completed. Another feature that needs to be implemented is a meaningful integration of the Canadian Food Guide. Currently, there is a singular page that just displays the meal portion recommendations of the Canadian Food Guide. However, there is no implementation of a comparison between the user’s food trends and that recommended by the Canadian Food Guide. This was due to the newer version of the guide being less metric based and more recommendation based. Finding a way to meaningfully integrate the Canadian Food Guide into the app is definitely something that can be worked on in the future. Finally, a global food database is a feature that can be implemented in the future. The app has a user database that is filled by food items that the user chooses. One thing we wanted to implement, but couldn’t due to time limitations was having a global database that is pre-populated full of food items so that a user can quickly search for and input items. In the future, implementing this as a food input method can be extremely beneficial.

References:

[1] K. Powderly, “Page Not Found,” Diabetes Canada, Apr. 08, 2019. https://www.diabetes.ca/media-room/press-releases/one-in-three-canadians-is-living-with-diabetes-or-prediabetes (accessed Dec. 07, 2020).

[2] Public Health Agency of Canada, “Diabetes in Canada - Canada.ca,” 2017. https://www.canada.ca/en/public-health/services/publications/diseases-conditions/diabetes-canada-highlights-chronic-disease-surveillance-system.html.