

# Progression of the DiaFriend Application for Self-Management Support in Portuguese Patients with Type 2 Diabetes Mellitus

Student: Kathryn Hockenbury

Primary Advisor: Peeranuch LeSeure, PhD

Secondary Advisor: Shelley Zhang, PhD

## Table of Contents:

1) Literature, Background, and Justification.....	2
2) Scholarly Question or Challenge.....	2
3) Methodology/approach: Development.....	3
4) Outcomes.....	4
5) Conclusions, Implications, and Consequences.....	6
6) References.....	8
7) Appendix.....	9

## **1. Literature, Background, and Justification**

As the University of Massachusetts Dartmouth is on the southern coast of Massachusetts, the South Coast community is a focus for many faculty and staff. For example, the UMass Dartmouth Public Policy Center wrote a Health Need Assessment for the South Coast community in 2019. Another example is a collaboration between the professors at the College of Nursing (CNHS) and the Computer and Information Science Department, where professors from two disciplines conducted research to develop a culturally sensitive diabetes self-management application to enhance diabetes self-management among Portuguese people with diabetes living in the South Coast community. Dr. Peeranuch LeSeure, an Assistant Nursing Professor, collaborated with colleagues from the nursing college and Dr. Shelley Zhang from the computer science department to develop a culturally sensitive diabetes self-management application. Dr. LeSeure and Dr. Zhang provided guidance and supervision to two undergraduate students, Sharan Kumar and Bhargavi Govardhanam, who diligently worked to develop a cell phone application that could potentially improve diabetes management tailored to the cultural needs of the target population for improving diabetes management.

This android application, named DiaFriend, can be downloaded off the Google Play store for free and used to track glucose levels. There are many different glucose-tracking apps for iOS and Android, but DiaFriend is specific to Portuguese culture by including foods that are typical of a Portuguese diet, which are not always included in other glucose-tracking apps. This application is critical because, according to UMass Dartmouth's Public Policy Center's 2019 Health Need Assessment, the percentage of adults diagnosed with diabetes in 2016 in Fall River was 11.2% and in New Bedford 11.8%. In contrast, the Massachusetts average is only 9.3%. Also, Fall River and New Bedford are the 3rd and 4th most populated cities with Portugal-born residents (Top). A fully functional DiaFriend Application could be life-changing for these patients who could manage their own diabetes. Having an easy, readily available, interactive application will help patients make health choices and see the impact of their carbohydrates and exercise on their glucose levels. The simpler and more user-friendly application will encourage patients to use the application daily and rely on it to estimate their general health.

## **2. Scholarly Question or Challenge**

On the South Coast, there is a large Portuguese population that has underprivileged access to health care. This has manifested specifically in patients with type 2 diabetes mellitus. They have a harder time keeping track of their intake and glucose levels in confusing and costly apps. Also, the information available is not centric to a typical Portuguese diet. The work Dr. LeSeure initiates is invaluable because opportunities to manage one's health should be equitable for all people, regardless of their background or circumstances. Diabetes is a significant health concern, and providing diverse populations with accessible and culturally appropriate tools to take control of their health is crucial.

The DiaFriend app was published on an Android application store but is no longer available to download due to its technical problems, including the lack of back-end development to make the application functional. This situation presents an opportunity to enhance the app's functionality, such as incorporating additional features, including exercise tracking and weight monitoring, both of which can significantly impact glucose levels for diabetes management. The project's challenges open doors for innovative solutions and a deeper understanding of creating comprehensive, user-friendly applications tailored to real-world healthcare needs.

The following functionalities were the original goal to be added to the DiaFriend app.

1. When the user enters their glucose level, that data will be saved and added to a graph. This graph will be viewable to the user for the day.
2. In the section for adding carbohydrates consumed, the page will display example graphics of carbohydrates, which can be clicked to add to the user's daily food log and edit their daily allowance of carbs. This daily allowance is set by the user and can be changed.
3. Add an exercise section with photo examples and track the length and intensity of the exercise.
4. Create and show carbohydrate and glucose level graphs using the data they enter in existing equations by incorporating the user's weight and age.
5. When users first download the app, they set their first name, age, weight, and goals. These goals can include staying within their carbohydrate allowance or meeting a certain amount of minutes/intensity of exercise.
6. When the user stays within their daily allowance or meets their preset goals, the screen will display encouraging messages and confetti.

This was a huge undertaking for an undergraduate student, and I had never worked on a project this large by myself. Even with the basis of the application having been previously developed. Making a fully functional application is a subject I have never studied previously and was certainly a challenge.

### **3. Methodology/approach: Development**

I decided on the timeline of this project and approved it by my advisors. We had laid out tasks every two weeks of the project. As the developer, I would work on these tasks and meet with my advisors once every two weeks. This is similar to an Agile methodology approach in which the scrum team would work in a two-week sprint before presenting a demo to the project manager and clients. Agile is common in the field of computer science today, but it is typically made up of small teams, not one person. A scrum team would feature a scrum master, a product manager, code developers, quality assurance testers, and documenters. The scrum master would

run the daily scrum meetings of the scrum team. They would keep track of the daily progress made by each member of the team and find solutions to the problems brought up by the team.

The product manager is the person responsible for approving each piece of the product and presenting it to the clients. Developers design and write code. Quality assurance testers will do unit, environmental, and possibly user interface testing. Documenters will keep records of all team meetings, user stories, tests, and client feedback. Small teams may indeed have people wearing multiple hats. Having developers also be testers and responsible for documentation is not abnormal. However, being by myself made this less of a typical Agile process. Rather than working from requirements, designing a solution, implementing, testing, and integrating, I started in the middle of a project. I downloaded the code from GitHub to Android Studios, an Integrated Development Environment (IDE). I then had to download and update all the libraries that the DiaFriend application is dependent on. This caused many errors in the project because the dependencies were out of date and, due to the number of errors, I could not run the program without crashing my laptop. After meeting with my advisors, I pivoted to Visual Studio Code, a different IDE. I then redownloaded packages, and libraries and changed settings to my computer just to run the application. With no prior experience in the programming language, I have spent the rest of the project trying to understand the code provided. I only had a few comments on some of the many files and no documentation, so I needed to reverse-engineer the files, breaking them down into pieces I could understand. This is when I could finally get to coding, and when I faced the biggest problem of this project. The back end of the application was non-functional. Users could type in numbers, but they wouldn't be saved anywhere. After working for the whole spring semester and the summer, I still did not have a solution, so my advisor and client had to pivot the functionality of the application. Rather than being a tracker, it would be a purely informational application. Users can check the calories of common foods, check their BMI, and see the calories burned on average from exercise. This change allowed me to finish implementing the new requirements of the application.

#### **4. Outcomes**

Completing this APEX project has provided me with invaluable experience that has prepared me for a career in computer science. Coding functionalities in the DiaFriend app, working with two advisors, acting as my real-world manager and client, and working on a large project outside of my normal class material prepared me for my career as a computer scientist. This assignment will give me an advantage because I have experience working on a real, fully functional product by mostly teaching myself.

First, I will start with everything I worked on and then everything I was able to complete. Also, see Figure 1 in the appendix for a diagram of all the files I worked on.

##### **The Back End, Saving Glucose Levels**

Working on the back end of the application was my main focus of the first semester. It was the biggest flaw with the existing app and ultimately proved to be too big of a challenge for me. Over many months, I tried everything I could find. I first tried saving the variables three or four different ways. I tried making text files for storage which the

user's input would be written into and read from it. I tried incorporating a database, MongoDB. I had never worked with a database before, so that was its own learning curve. Over the summer, I read a book about Dart, the programming language Flutter is based in. I also tried using a no SQL Microsoft Azure database to save the data, with no success. I asked my family and friends for help, trying to find anyone who knew Flutter or Dart. I reached out to other computer science professors, my manager at my IT engineering internship, and my co-workers. Ultimately, the back end was not capable of storing the information. With months of work and no success, my advisors and I decided changing the purpose of the application to be informational, and not based on user input for tracking, would be for the best. See the appendix Figures 2-3 showing how the glucose section of the application looks.

### Graphing Function

As a part of the tracking of glucose, the client wanted the glucose level to be graphed, so users could see if their level was high, moderate, or low over the day. They would track their glucose after every meal. This too will rely on the storing of the glucose level entered by the user. Once that variable is stored, however, the graph would be functional. See Figure 4 to see the graphing page.

### Authorization and Log-In Screen

While working on the back end, as described above, I worked on adding a login screen and user authorization. This involved adding a Google Firebase account for the app and attaching that as a plug-in. This, however, also relies on storage which is the problem of the back end. The login page is fully designed, but it would need back-end implementation with Firebase. This was a theme of many of the tasks I worked on. The front end was designed and developed, and, if someone else can add the back-end storage, then the app would be fully functional. See Figure 5 in the appendix to see the user interface of the log-in page.

### Medication Page

The medication page is a list function where users can list out the names of their medications, the dosage, and when to take the medication. It would stay in the app as a reminder to the user. This, too, is functional but cannot save the data. The user can enter the name, dosage, and time of day for multiple medications and delete them from the list but once the app is closed the data is lost. Appendix Figures 6 and 7 show the medication page.

### Body Mass Index Calculator

Body Mass Index (BMI) Calculator takes in the user's sex, height, and weight and, through a formula, calculates the BMI. The BMI appears on the screen but is not saved in the app. This piece is fully functional and useful to users, except that the app would not keep the past BMI calculations. Figure 8 in the appendix is the BMI calculator page.

### Adding Carbohydrates

As a part of the existing application, users could find the calorie amounts of common foods, some specific to the Portuguese diet, which makes DiaFriend unique to the

Portuguese community. I was able to add more foods to the list with pictures and calories given to me by my client. As the application's main focus became informational, this is a key aspect of making the app useful. I also made the pages for the other types of food functional by updating the existing code, so they appear in the application. See Figures 9-15 of the carbohydrate pages of food examples.

#### Adding Exercise

The application also allows users to find out approximately how many calories they could burn off by certain exercises. This information is important to the application as exercise can impact a person's diabetes. I made the existing file viewable to users and added additional exercises for reference. Figures 16-18 show the exercise pages in the application.

### **5. Conclusions, Implications, and Consequences**

This project has been immensely impactful on my academic career. I had never worked on a project of this magnitude before, and it came with hard lessons that I will carry with me. I have learned a lot over the past year working on this project. I am lucky that, despite all the troubles I had, this happened in an academic setting where I could be supported rather than learning these hard truths at my first job. I am proud of the work I accomplished, but I also would have approached this problem much differently. I am a team player, and I think having a team and more mentorship would have helped. I should have never taken on such a challenging task without prior research or a passion. I should have asked for more help earlier on, and that's a lesson I will continue to learn. I hope that this project has a future because I believe it could help patients and Dr. Lesure in her research.

Documentation was not provided to me with this project, but, at the time, I did not know what I was missing. I have now been taking the Computer Science Senior Design class and there is a large focus on documentation. Commenting on code, writing down steps you've taken, and creating design plans or models, are all helpful to a project, especially one like this which is passed down through students. It is impossible to know all that another developer was thinking or planning, but documentation can help tremendously. Even not knowing the programming language, and documentation would have reduced the learning curve because the design of the application, the libraries, and plugins could be explained.

As I mentioned, I had two advisors, one a nursing professor for whom the application was developed, and one computer science professor. While I am grateful for the professional and supportive experience, they are not well-versed in Flutter or the design of the application. This meant that the internet was my main support. When that failed, I reached out to everyone I could to get help, but, because this was a newer language, it was impossible to find help. If I were to do it all again, I would have chosen to start the project over in Java. Java is a more well-known and established programming language taught at UMass Dartmouth. This would have given me more mentors I could go to for help and would have likely made the experience less painful.

Lastly, the hardest lesson learned was due to a lack of research. Dart is the programming language Flutter is based in, and, over the summer, the Dart 3.0.0 version was released. I clicked to update my project to the Dart 3.0.0 version without looking into the changes that it would make to the application. When the first number in a version release is changed, that means it is a large update, and the large update from Dart version 2 to Dart version 3 was null safety. Null safety means that no references will have null values. For the DiaFriend application, that meant seventy thousand errors were caused. Some were solved with quick fixes, but some had to be located and solved line by line in files. This meant the application was unrunnable for weeks as I tirelessly fixed every error. This was an unforgettable mistake, and, even now, at the time of writing, the version is now up to Dart 3.84.0 meaning 84 times Dart as a whole had to be updated due to the null safety update.

Here is all that I have learned and what I will take away from this project. I have learned how critical documentation, mentorship, and research is. In the future, I will be conscientious in commenting on the code pieces I am working on and write down the steps that I take to solve a problem. This does take more time in the moment, but, over time, it is worth it for helping me and the developers who come after. Also, working with a team will mean that more documentation will have to be made to explain to others what I am working on and how that fits into the project. I have learned that I will need to seek out help from others. I will need to work collaboratively with coworkers and receive more frequent feedback. Lastly, I will always look into updates. Not that an update should never be accepted, but I will better prepare not to be thrown off by weeks of bug fixing. If an update is in the plan, it can be accounted for. All of these lessons and my experiences have better prepared me for a career in software engineering.



## 6. References

James, D. (2022, August 3). Bhargavi Govardhanam '23 designs app for diabetes research.

2022: Bhargavi Govardhanam '23 designs app for diabetes research. Retrieved September 18, 2022, from <https://www.umassd.edu/news/2022/bhargavi-govardhanam-releases-Diafriend.html>

JavaTpoint. (n.d.). *Flutter vs Java*. www.javatpoint.com. Retrieved November 27, 2022, from <https://www.javatpoint.com/flutter-vs-java>

*Top 101 cities with the most residents born in Portugal*. Top 101 cities with the residents born in Portugal (population 500+). (n.d.). Retrieved November 27, 2022, from <https://www.city-data/top2/h55.html>

UMass Dartmouth Public Policy Center. (2019). *Community Health Needs Assessment*. Southcoast Health. Retrieved November 27, 2022, from [https://www.southcoast.org/wp-content/uploads/2022/02/Southcoast-Health-CHNA-2019-Final\\_PPC-1.pdf](https://www.southcoast.org/wp-content/uploads/2022/02/Southcoast-Health-CHNA-2019-Final_PPC-1.pdf)

## 7. Appendix

### File Diagram

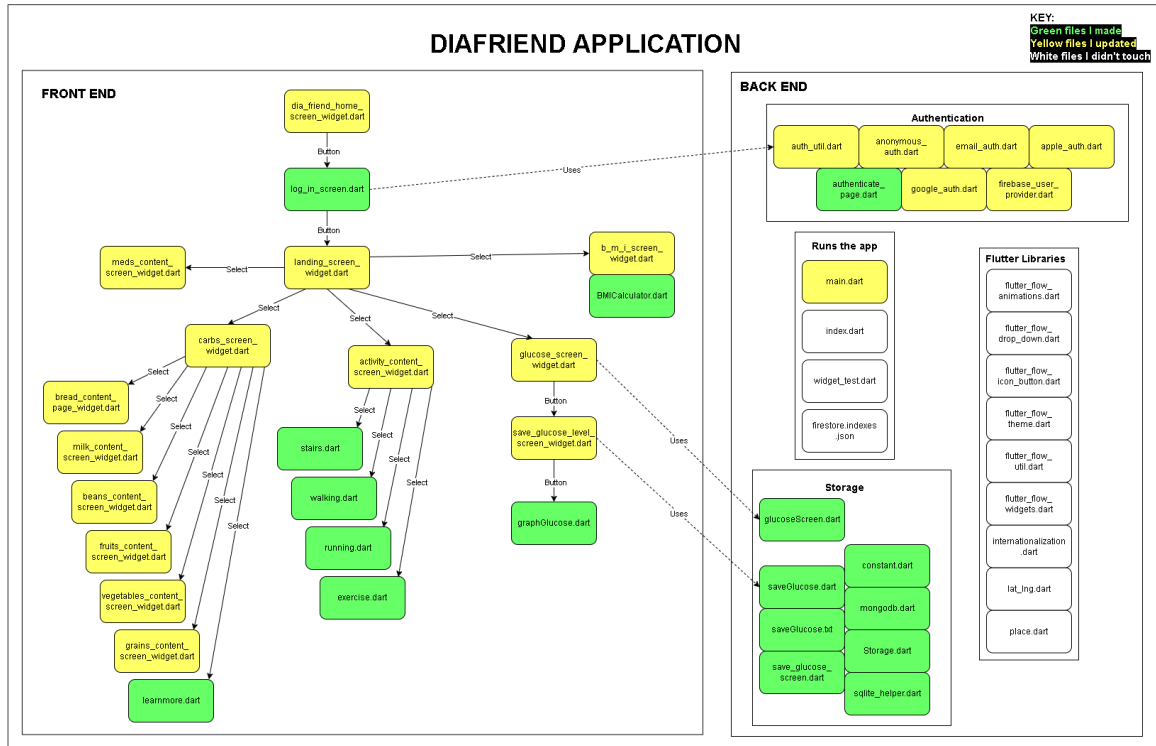


Figure 1: A diagram of all the files contained in the DiaFriend Application. Files on the left are a part of the front end, and on the right are a part of the back end. Green colored files are files I created and developed myself, while yellow files already existed in the app, but I edited, and updated them to be functional. White files I did not edit.

## The Back End, Saving Glucose Levels

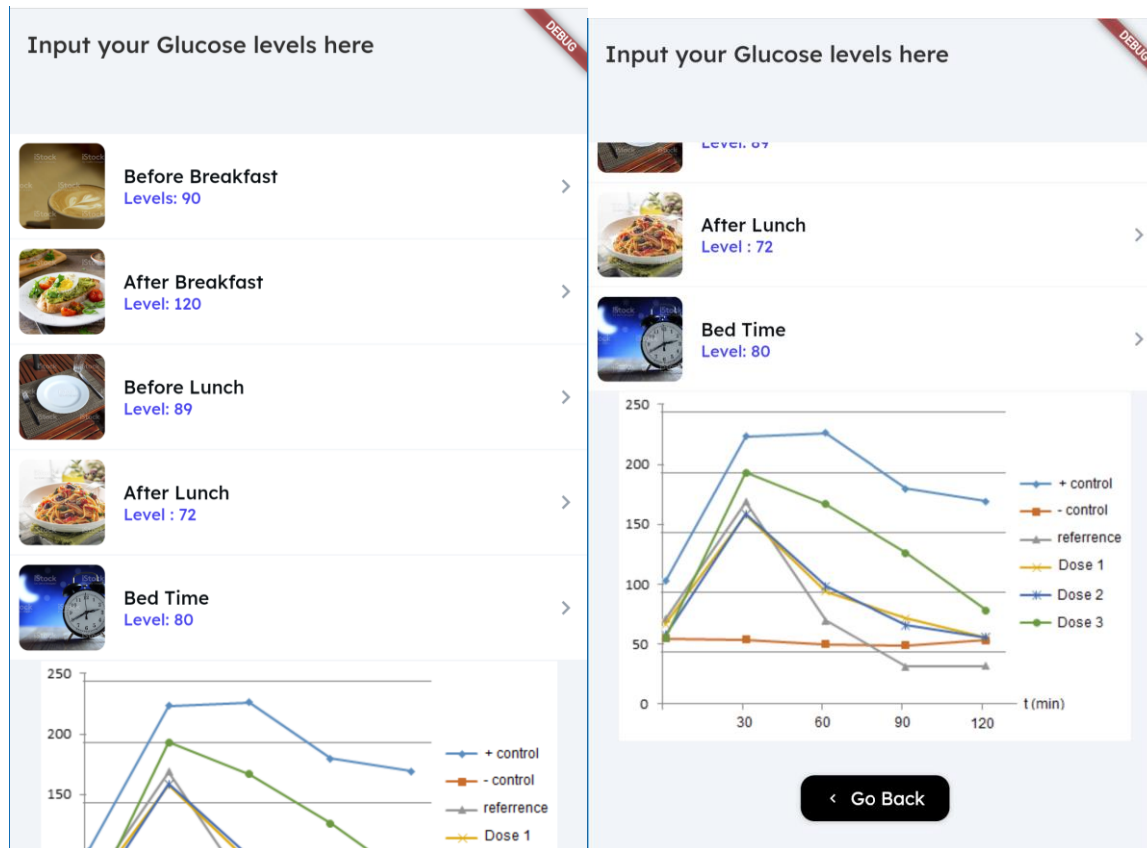
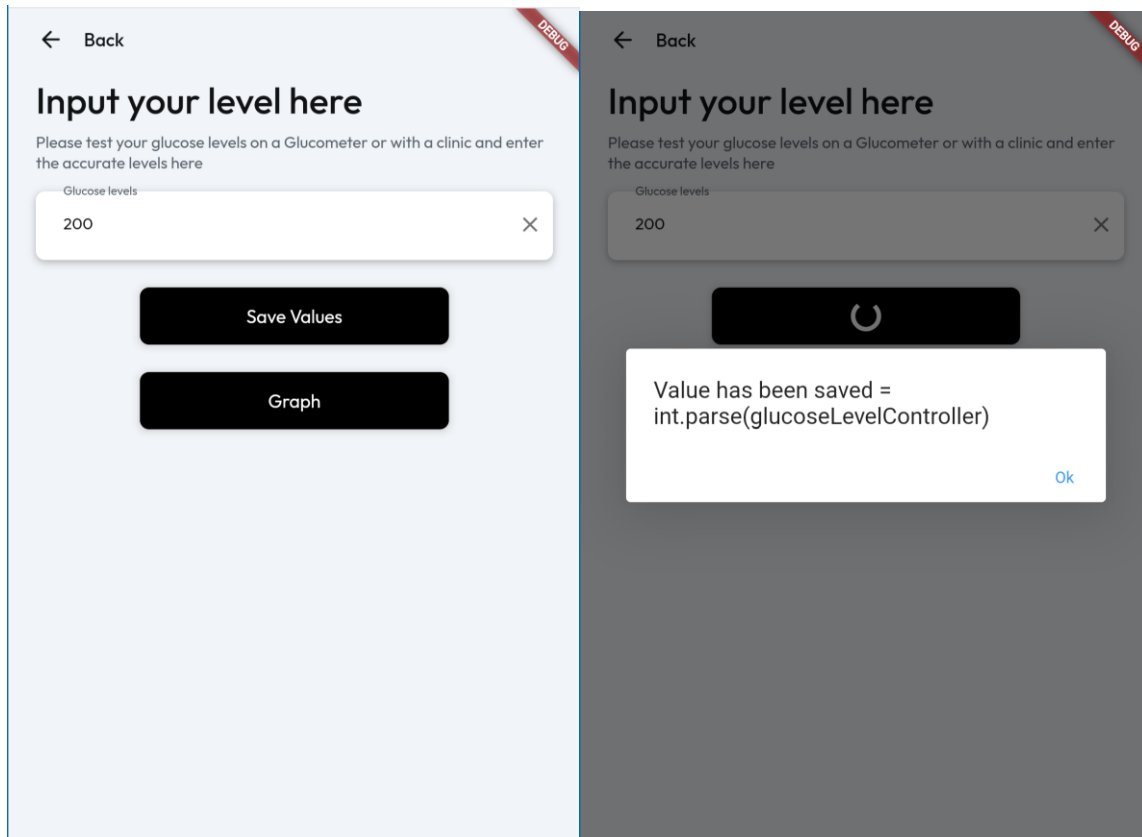
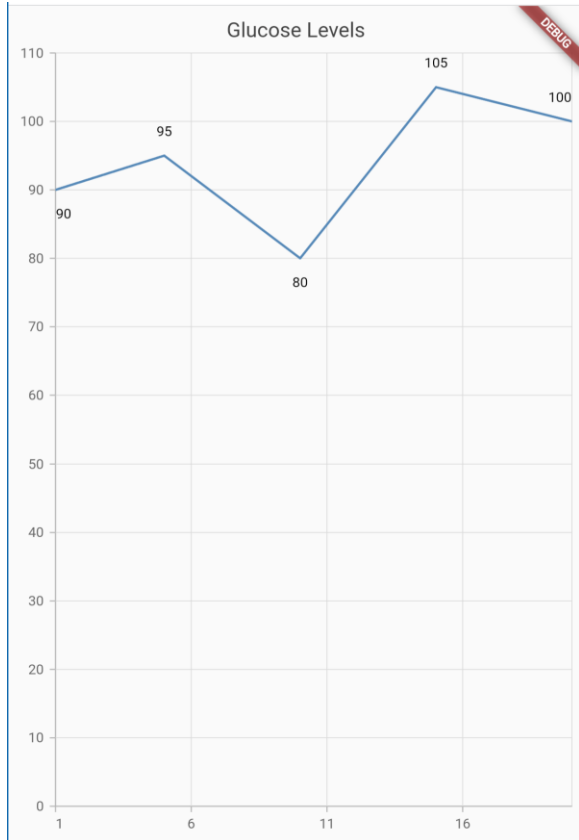


Figure 2: The opening glucose tracking screen. Here users select the time of day they are tracking their glucose at.



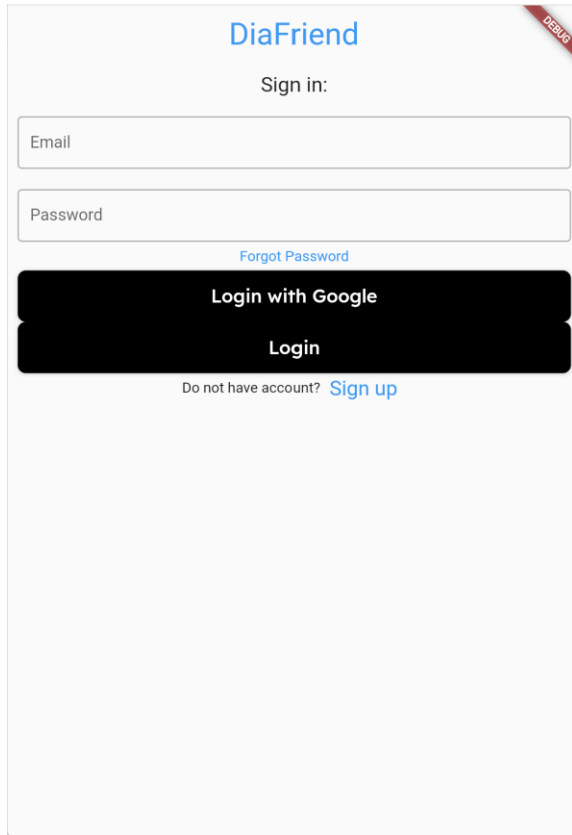
*Figure 3: Here users type in their current glucose level and click Save Values.*

## Graphing Function



*Figure 4: The glucose graphing screen. After users have saved their glucose they can choose to see the graph of their glucose over time. \* These are just hard coded values and not receptive to user input at this time.*

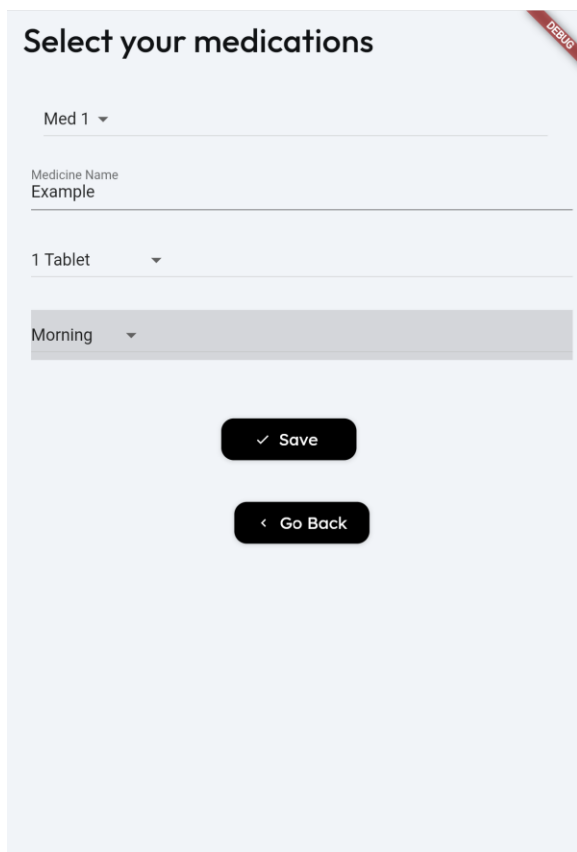
## Authorization and Log-In Screen



The image shows a mobile application login screen for 'DiaFriend'. At the top, the app name 'DiaFriend' is displayed in blue. Below it, the text 'Sign in:' is centered. There are two input fields: 'Email' and 'Password'. A blue link 'Forgot Password' is positioned below the password field. Two large black buttons are stacked: 'Login with Google' on top and 'Login' on the bottom. At the very bottom, the text 'Do not have account?' is followed by a blue 'Sign up' link. A red 'DEBUG' banner is visible in the top right corner of the app interface.

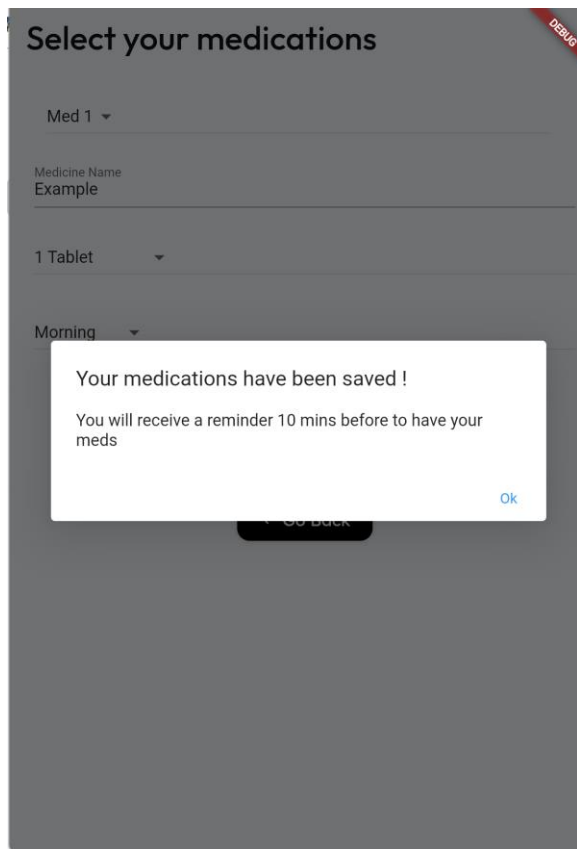
*Figure 5: After the opening screen this login screen appears. Users can type in an email and password. Then select either Login with Google or Login.*

## Medication Page



The image shows a mobile application screen titled "Select your medications". In the top right corner, there is a red diagonal banner with the word "DEBUG" in white. The form contains four input fields: a dropdown menu showing "Med 1", a text field with "Medicine Name" as a label and "Example" as the input, a dropdown menu showing "1 Tablet", and a dropdown menu showing "Morning". At the bottom of the form, there are two black buttons with white text: "✓ Save" and "< Go Back".

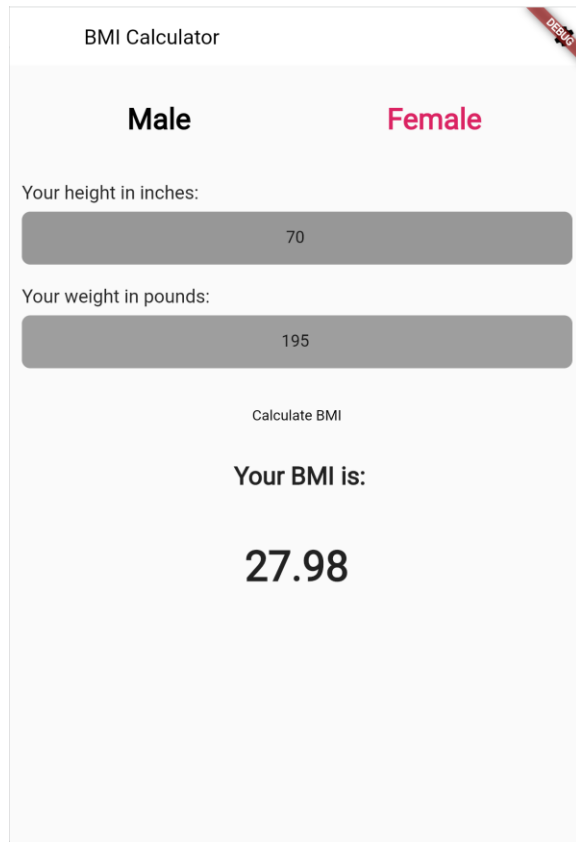
*Figure 6: The opening page of the medication page. Here users can select a medication to add. Type in its name, select a quantity and select a time of day. Then press Save.*



*Figure 7: After users press save, this pop up appears.*



## Body Mass Index Calculator

The image shows a mobile application interface for a BMI calculator. At the top, the title "BMI Calculator" is displayed. Below the title, there are two radio buttons for selecting sex: "Male" and "Female". The "Female" option is selected, indicated by a pink dot. Below the sex selection, there are two input fields. The first is labeled "Your height in inches:" and contains the value "70". The second is labeled "Your weight in pounds:" and contains the value "195". Below these inputs is a button labeled "Calculate BMI". At the bottom, the text "Your BMI is:" is followed by the calculated value "27.98" in a large, bold font. A small red ribbon graphic with the word "DESIGN" is in the top right corner.

BMI Calculator

**Male** **Female**

Your height in inches:

70

Your weight in pounds:

195

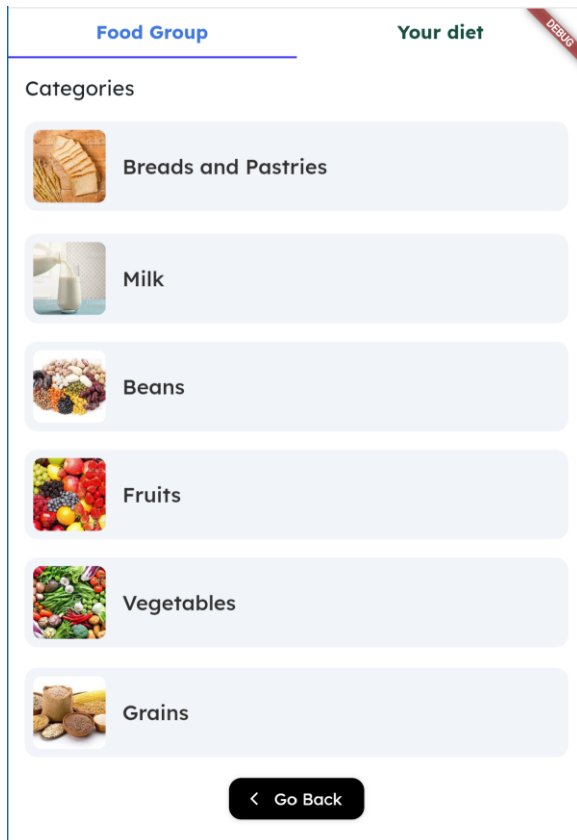
Calculate BMI

Your BMI is:

**27.98**

*Figure 8: The body mass index calculation screen. Users select their sex, enter their height and weight and select Calculate BMI.*

## Adding Carbohydrates



*Figure 9: This is the opening screen of the carbohydrates food section. Here users can select the type of food they have eaten.*












Breads and Pastries	Milk
 <b>Papo Secos / Portuguese Roll</b> 1 roll 100 Kcal	 <b>Chocolate Milk</b> 1 cup ~ 26 gms 208 cal
 <b>Pão Doce / Portuguese Sweet Bread</b> 1 slice 120 Kcal	 <b>Low fat (2%) milk</b> 1 cup ~ 12 gms 121 cal
 <b>White Bread</b> 1 slice 78 cal	 <b>Yogurt (low fat)</b> 1 cup ~ 30gms 161 cal
 <b>Brown Bread</b> 1 slice 76 cal	 <b>Yogurt (fruit, low fat)</b> 1 cup ~ 42gms 225 cal
 <b>Pastel de nata</b> 1 pastel 180 Kcal	 <b>Skim milk</b> 1 cup ~ 12 gms 86 cal
 <b>Arroz Doce / Rice pudding</b> 1/2 cup 142 Kcal	
<a href="#">&lt; Go Back</a>	<a href="#">&lt; Go Back</a>

Figure 10: The bread and pastries list.

Figure 11: The milk and dairy list.












Beans	Fruits
 <b>Black eye peas</b> 1/2 cup ~ 26 gms 134 cal	 <b>Apple</b> 1 medium 81 cal
 <b>Garbanzo beans</b> 1 cup ~ 45gms 121 cal	 <b>Banana</b> 1 medium 105cal
 <b>Pinto beans</b> 1 cup ~ 48gms 161 cal	 <b>Grapes</b> 1 cup ~ 30gms 114 cal
 <b>White beans</b> 1 cup ~ 45gms 225 cal	 <b>Orange</b> 1 medium 65 cal
 <b>Navy beans</b> 1 cup ~ 48 gms 259 cal	 <b>Apple</b> 1 medium 81 cal
	 <b>Dates</b> 10 dried 61cal
<a href="#">&lt; Go Back</a>	<a href="#">&lt; Go Back</a>

Figure 12: The beans list.













Vegetables	Grains
 <b>Carrot</b> 1 medium31 cal	 <b>Ready to eat Cereal</b> 1 cup110 cal
 <b>Corn</b> 1/2 cup89 cal	 <b>Cream of Rice</b> 3/4 cup95 cal
 <b>Beans, Lima</b> 1/2 cup cooked114 cal	 <b>Cream of Wheat</b> 3/4 cup95 cal
 <b>Green Peas</b> 1/2 cup63 cal	 <b>Pancake (4 inch diameter)</b> 10 count63 cal
 <b>Baked Potato</b> 1 large220 cal	 <b>Cheese Pizza</b> 1 slice290 cal
 <b>3 Bean Salad</b> 1/2 cup90 cal	 <b>Noodles (Spagetti)</b> 1 cup159 cal
<a href="#">&lt; Go Back</a>	<a href="#">&lt; Go Back</a>

Figure 14: The vegetables list.

Figure 13: The fruits list.

Figure 15: The grains list.

## Adding Exercise

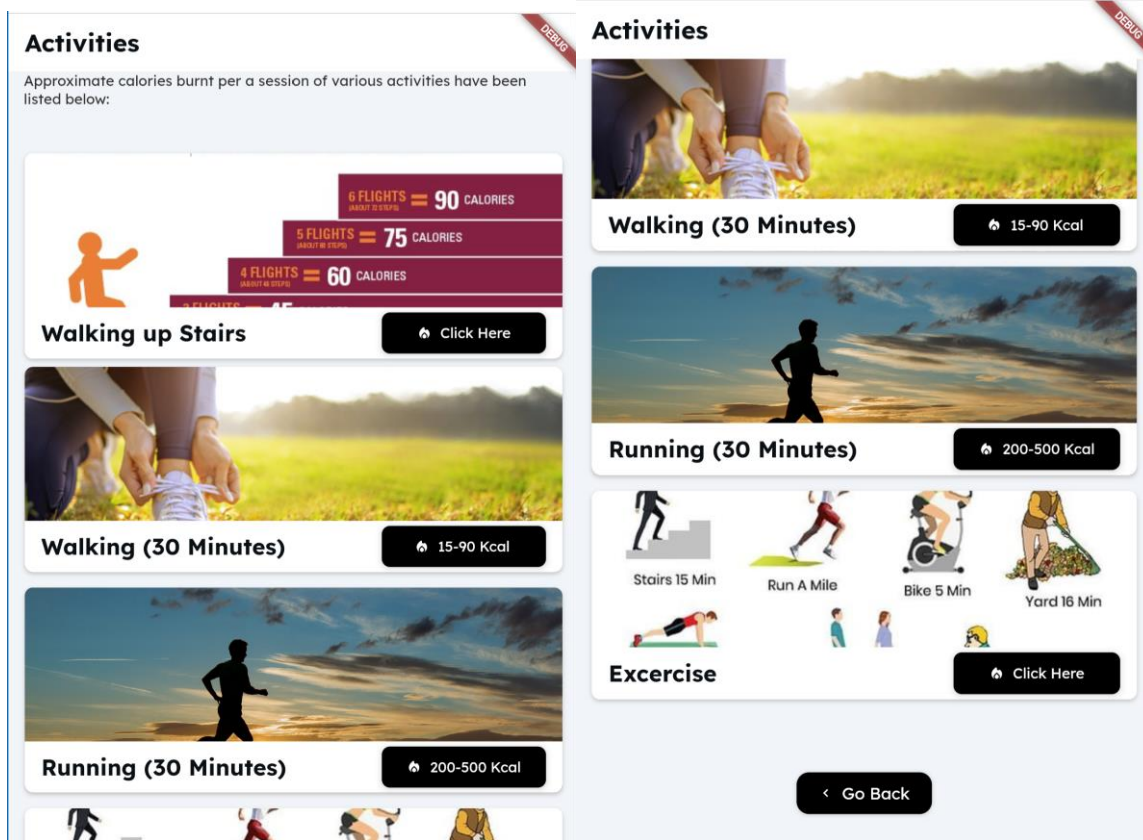


Figure16: The opening page of the activities section. Here users can see approximate calories burned for 30 minutes of walking and running.

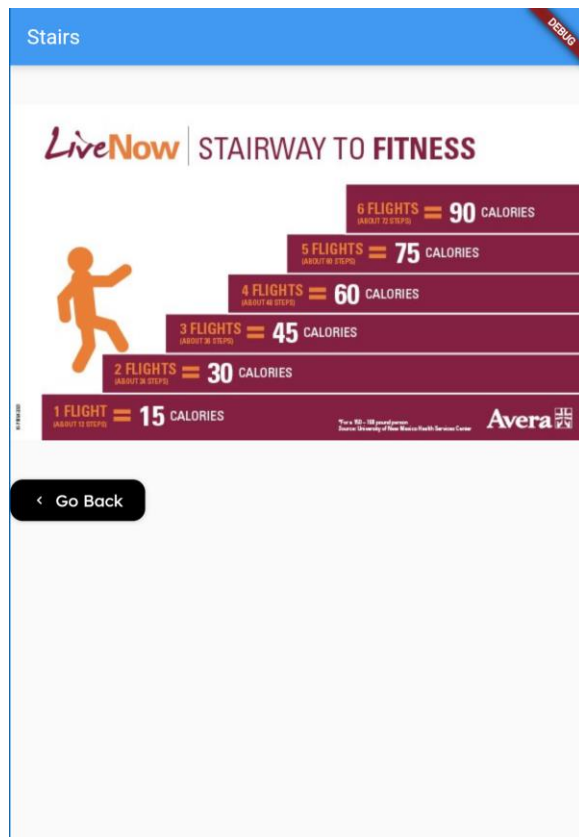


Figure 17: The full image of the walking up stairs activity option.

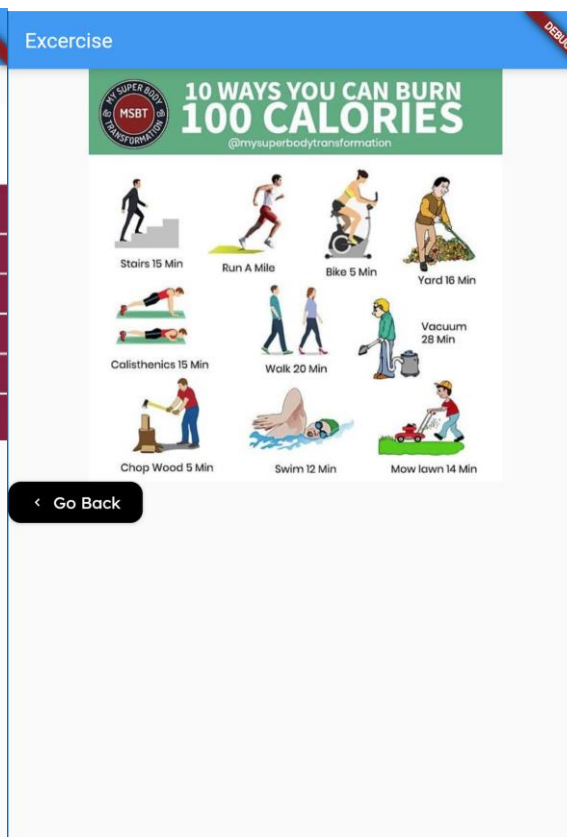
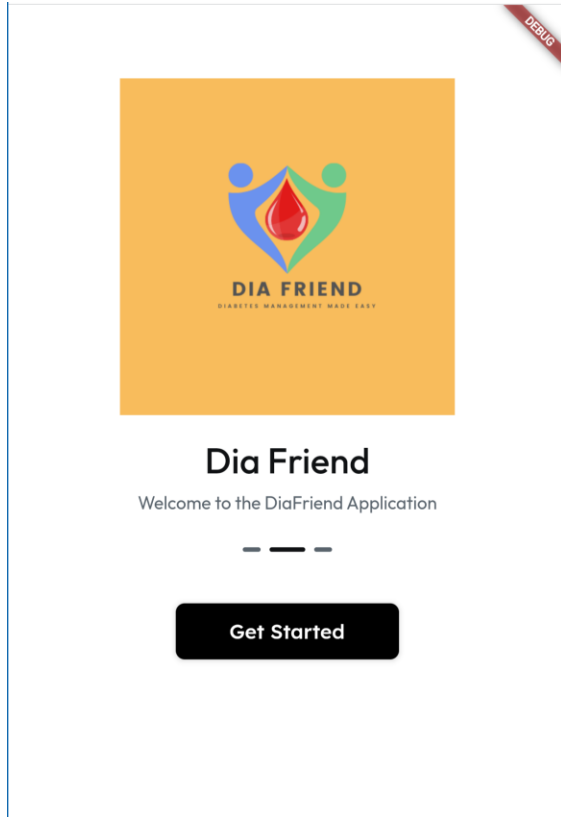


Figure 18: The full image of the exercise option of the activities section.

## Additional Screens to the Application



*Figure 19: The opening screen to the application.*

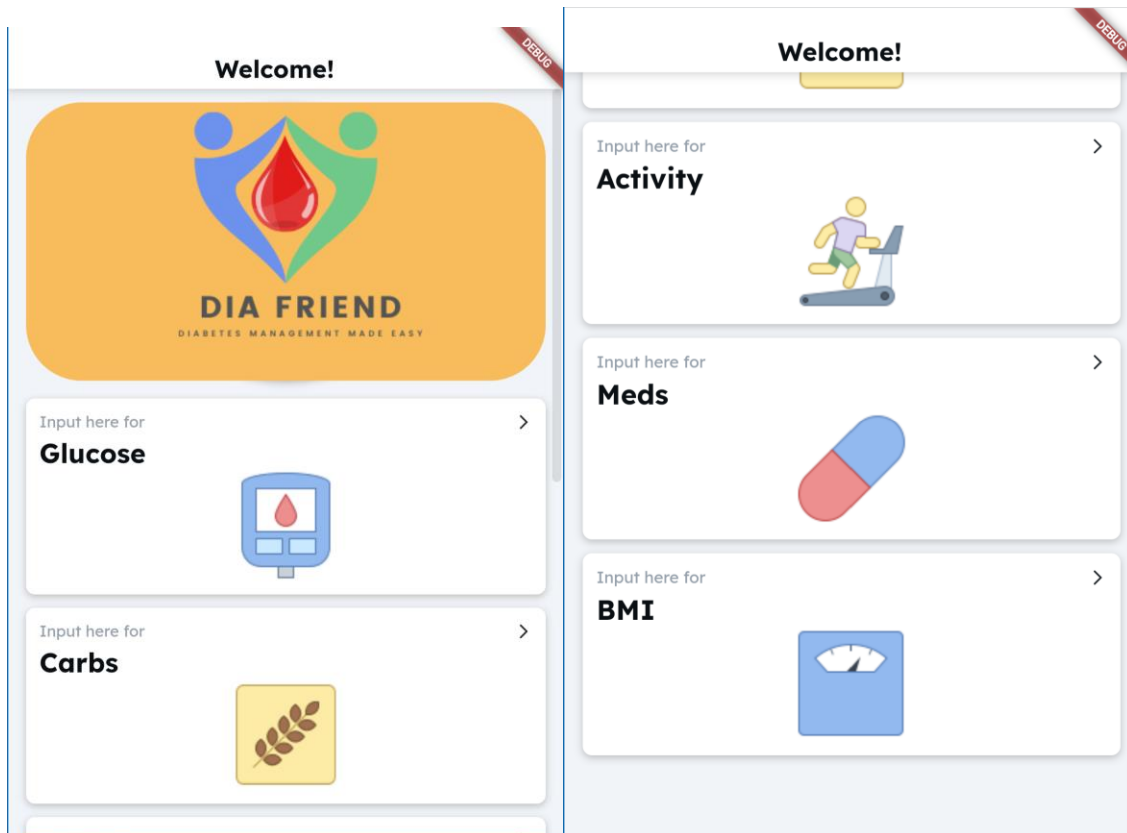


Figure 20: The landing page of the application, which lists the 5 function options.

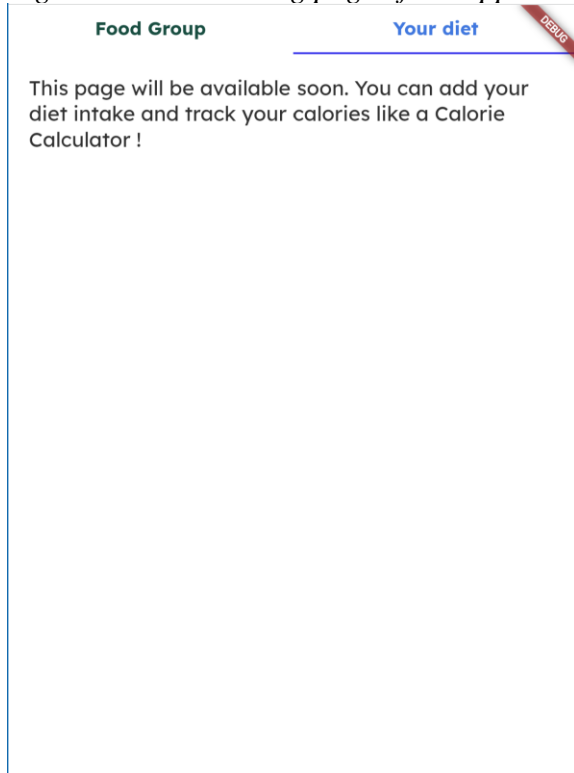


Figure 21: The Your Diet Intake page, outside of the scope of my assignment.