

GNG2101

Design Project User and Product Manual

Submitted by:

Pancreas Pals, Team C11

Khalil Aouadi, 300197227

James Couture, 300076065

Mounira Nihad Zitouni, 300190536

Regina Mayani, 300207233

Spencer Henry, 300073281

March 22, 2021

University of Ottawa

Table of Contents

Table of Contents	ii
List of Figures	iv
List of Tables	v
List of Acronyms and Glossary	vi
1 Introduction	1
2 Overview	2
2.1 Conventions	2
2.2 Cautions & Warnings	2
3 Getting started	3
3.1 Set-up Considerations	3
3.2 User Access Considerations	3
3.3 Accessing the System	3
3.4 System Organization & Navigation	3
3.5 Exiting the System	3
4 Using the System	4
4.1 <Given Function/Feature>	4
4.1.1 <Given Sub-Function/Sub-Feature>	4
5 Troubleshooting & Support	5
5.1 Error Messages or Behaviors	5
5.2 Special Considerations	5
5.3 Maintenance	5

5.4	Support	5
6	Product Documentation	6
6.1	<Subsystem 1 of prototype>	6
6.1.1	BOM (Bill of Materials)	6
6.1.2	Equipment list	6
6.1.3	Instructions	6
6.2	Testing & Validation	7
7	Conclusions and Recommendations for Future Work	8
8	Bibliography	9
	APPENDICES	10
	APPENDIX I: Design Files	10
	APPENDIX II: Other Appendices	11

List of Figures

Figure 2: Flowchart/Diagram of the Decomposition of the Application	4
Figure 3: Android phone	7
Figure 4: Home page	9
Figure 5 : Calculation page	10
Figure 6 : Add a meal to save meal category pop up	11
Figure 7 :Saved Meals page	12
Figure 8 : Notification page	13
Figure 9 : Main page of the application	16
Figure 10 : Calculation Page	17
Figure 11 : Pop-up when adding meal	18
Figure 12: Saved Meal page Categories	19
Figure 13 : Saved Meals in a Category	20
Figure 14 : Settings page	21
Figure 15 : notifications Page	22
Figure 16 : Screenshot of the files opening with Android Studio	26
Figure 17: Screenshot of Android Studio - Emulator section	27
Figure 18 : Picture of the Prototype Used for Analysis	28
Figure 19 : Calculation Page Questions	39

List of Tables

Table 1: Bill of materials	25
Table 2 : Comparison between the Target, the Prototype before Meal Bank, and Prototype after Meal Bank.	29
Table 3 : Representation of the Client Needs Met by the Prototype	31
Table 4 : Comparison between the Target, the Prototype before Meal Bank, and Prototype current Meal Bank.	33
Table 5 : The number of questions	38
Table 6: Referenced Documents	42
Table 7 : Error Messages or Behaviors and how to correct them	43

1 Introduction

This User and Product Manual (UPM) provides the information necessary for diabetics users to effectively use the Mealtime Insulin Calculator (MIC) and for prototype documentation.

This document has an objective to help users better understand and to document the past, current and future states of the Mealtime Insulin Calculator created by Pancreas Pals. This document will be going over the overview of the application, usage of the application, features, potential issues, build necessities, and future plans for the application.

In the overview of the application, the background idea and base of the project will be covered. This section focuses on the initial issue/problem statement given to the group. This section will also cover information about any health considerations before usage of the application.

The usage of the application section will cover the design and usage of the application. The different pages of the applications will be shown in detail and the general usage of each of these pages will be defined and described in detail.

The features section will go more in depth in the functionalities of the different pages. Within a page, different functions must be considered in order to make the application properly work. This section will give insight on where the calculations and considerations are taken in the application.

The potential issues section will go over any procedures or necessary actions if failure of the application. This section will cover any expected potential problems and how to resolve them.

The build necessities in this document covers information on what is necessary in order to operate the application.

The future plans section covers information of future implementations for the application. The section will go in depth of different possibilities and ideas in order to improve the application to make it more user friendly and better adapted to help the user's health management.

A selection of projects were given to the GNG 2101 class. Students in the class gave the professor their preferred project choices. The students were then given the project for the semester. The Pancreas Pals had interest in making a software to help diabetic patients. The students were then put in contact with a client with diabetes type 2 in order to help simplify the client's everyday life.

This User and Product Manual has no security or privacy considerations. No information is necessary in order to use this document. This document is made simply to aid anyone using the Mealtime Insulin Calculator or giving insight to potential future clients.

2 Overview

The insulin Calculator is a native application available on devices using the Android system.

The problem the team faced was to create a native mobile application used to calculate the amount of carbs on a daily basis. The application is important because it's accessible for users everywhere and the application is simple and easier.

The fundamental needs of the user had was to have the app to be :

- Compatible with android
- User-friendly Interface (Simple to use)
- Simple carbs calculator
- Time-efficient app
- low priced
- Exportable meal history
- Notifications

Figure 1 : Picture of the application

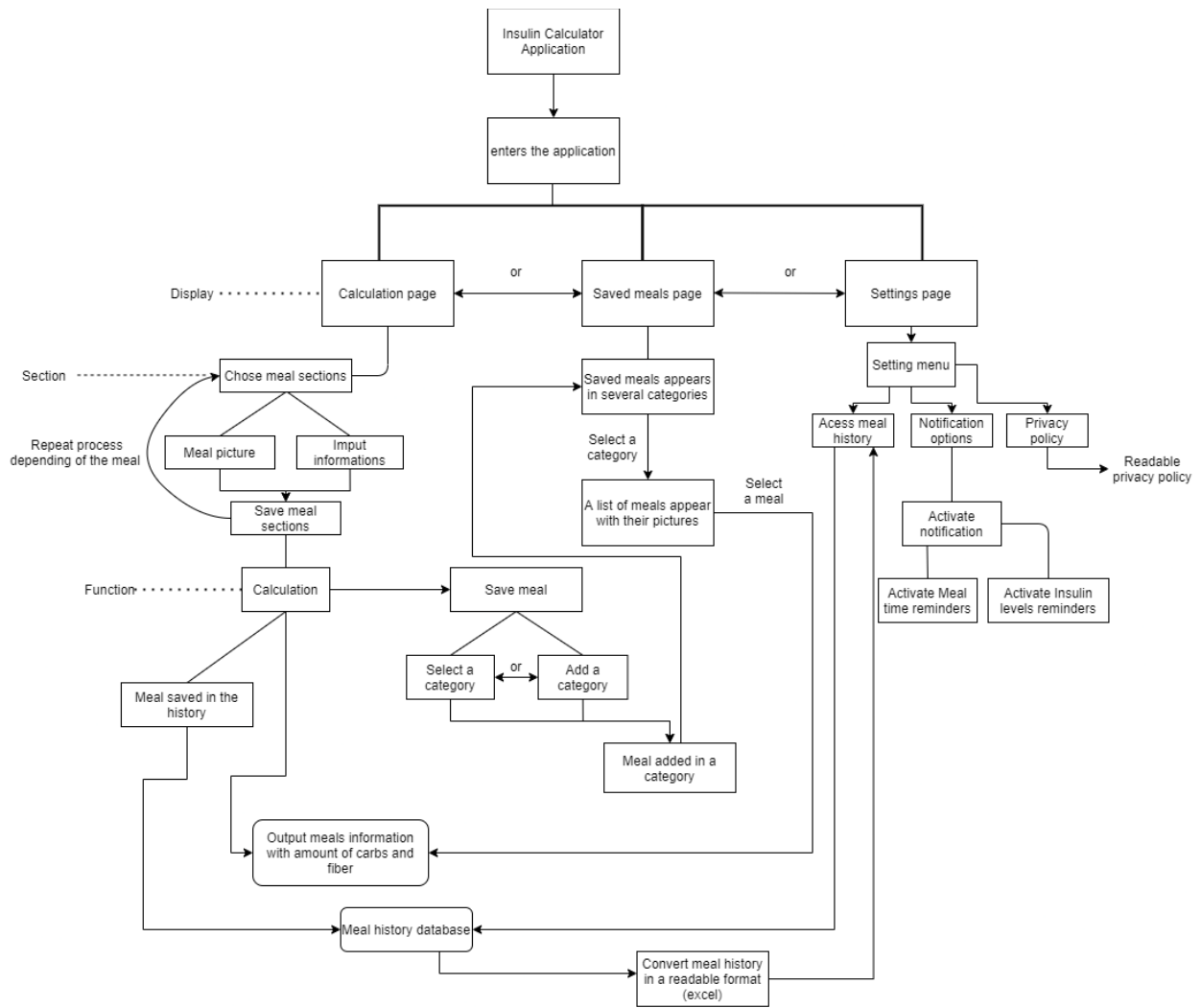


Caption: A screenshot of the calculation page englobing the division of the meal in different categories and their quantity of carbohydrate.

The main key functions of the application are the Home/Calculation page, the Saved meals, the Settings with, inside, a Notification settings .

Below is a diagram that helps better understand how each part of the application connects with each other and how the application holds together.

Figure 2: Flowchart/Diagram of the Decomposition of the Application



Caption: This image is a flowchart decomposition of the application showing all the sections and subsections of the application.

Detailed functional Decomposition of the Mealtime Insulin Calculator. Functional decomposition is a list of organized tasks required to satisfy customer needs.

2.1 Cautions & Warnings

I

The user should have the opinion of a health professional before using the application to know if this application applies to their health issues. The application is responsible to add carbohydrates together and fibers together. The application is NOT responsible for calculating insulin directly and recommending insulin dosages. Insulin dosages should be discussed with health care providers.

3 Getting started

3.1 Set-up Considerations

Going on the following github link:
(<https://github.com/Mealtime-Insulin-Calculator/Mobile-application>), downloading the apk file on the user phone will download the file. Once done, press on open download and click yes. This will prompt the file to be installed on the phone. After that, the application will be accessible in applications. For more information about getting apk files working, the following link has additional information:

<https://www.nextpit.com/android-for-beginners-what-is-an-apk-file>

Figure 3: Android phone



Caption: Screenshot of an android phone emulator.

To begin, the equipment needed for this application is an android phone.

3.2 User Access Considerations

One of the user groups that will utilize the application are diabetics, people who want to keep track of their sugar intake and health care professionals. It is important to keep in mind that the database is local and therefore is device based. The user can not access their information from a different device.

One of the restrictions placed on the application is the ability to use the app on a non android phone. The application solely works on an android platform. Also, the application is one user one application. In this case, the user needs to upload locally the file of the application. Finally, the application only has english settings and the language can not be changed.

3.3 Accessing the System (GO THROUGH BEFORE HANDING IN)

In order to access the System, users must first download the Mealtime Insulin Calculator through an APK file provided through Github (Please check section 3.1 for the link). Once downloaded, users will have full access to the application by simply opening the application. There is currently no user ID or login system developed for the application as of yet.

3.4 System Organization & Navigation

In this section, each screen of the Mealtime Insulin Calculator will be shown. The functions and features on all screens will be described in detail.

3.4.1 Home page

Figure 4: Home page



Caption : The main menu for the Mealtime Insulin Calculator, where food is put into specific categories and then added to the meal in a Pi Chart.

At the bottom of the home page, there is a navigation bar. In the navigation bar, the user can navigate through his saved meals, the settings and the calculation page. On the middle of the home page, a pie chart will present the carbohydrates distribution of each meal category. The colors don't mean anything, it is for visual representation purposes only.

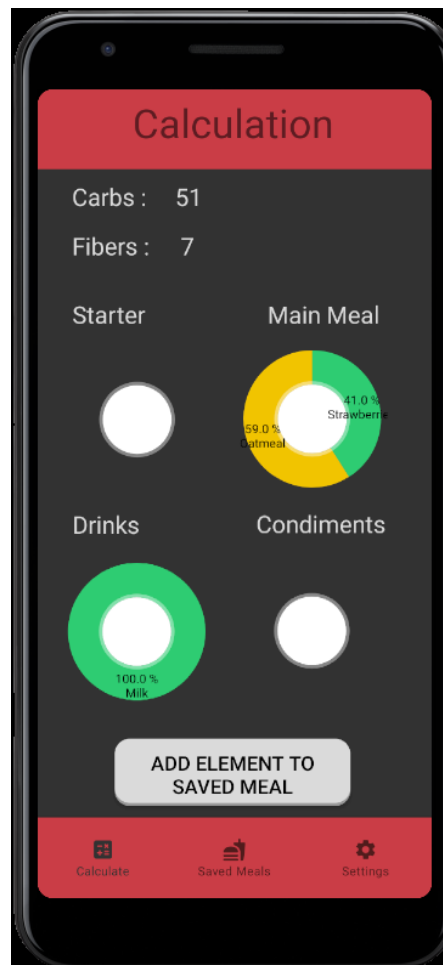
How to use it?:

- 1) Enter the name of the food in the name section input
- 2) Enter the grammes of carbohydrates in the carbs input
- 3) Enter the grammes of fibers in the fiber input
- 4) Select in which category the food belongs to (starter, main meal, drinks or condiments)
- 5) Click on the add elements
- 6) Repeat steps 1 to 5 for other food in the meal
- 7) When done adding elements, Click on calculate

Once the user clicks on the calculate button, it will show the calculation page.

3.4.2 Calculation page

Figure 5 : Calculation page



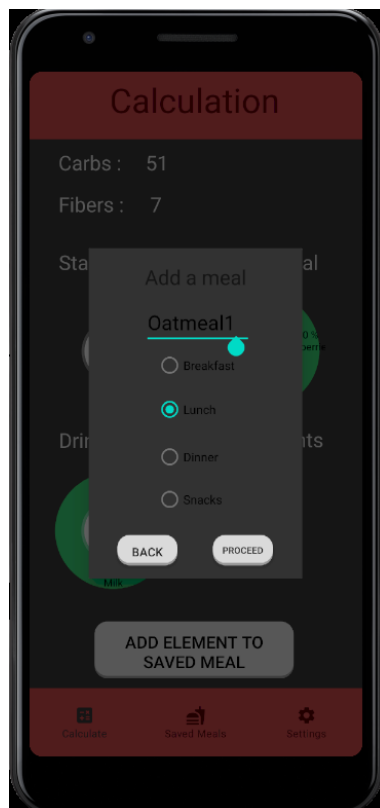
Caption: This screenshot represents the calculation page where all the information useful for calculations are inputted

The purpose of the calculation page is to have a visual representation of the carbohydrates distribution in each meal. At the top of the page on the left, the total amount of carbohydrates is shown next to the word “carbs”. Also, the total amount of fibers is right under the carbs.

There are a total of four pie charts (Starter, Main Meal, Drinks and Condiments). They are all showing their different carbohydrates’ distribution. The colors don’t mean anything, it is for visual representation purposes only.

To save the meal, the user clicks the “Add Elements to Saved Meals” button to add to favorites.

Figure 6 : Add a meal to save meal category pop up



Caption : This is the pop-up that appears when the user clicks to save a meal.

When wanting to save a meal, Figure... will pop up. This pop up asks the user to choose in which meal time the meal will be added to (Breakfast, Lunch, Dinner or Snacks). Once done, the user clicks on the “Proceed” button or, else, click on the “Back” button to come back to the calculation page.

3.4.3 Saved Meals

Figure 7 :Saved Meals page

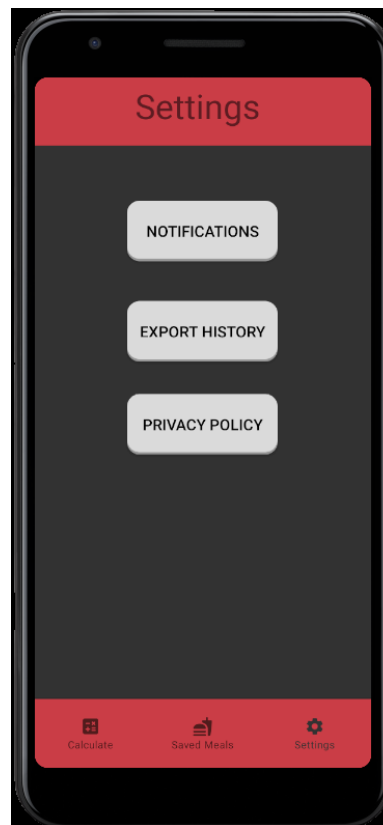


Caption : This screenshot is of the saved meals page. It is possible to see the different meals that were saved by the user (i.e. Burger, Lasagna and Oatmeal1).

The purpose of the saved meals page is to save the meals that are eaten the most often. The user clicks on one of the meals, for example Lasagna, and uses it for the calculations.

3.4.4 Settings

Figure 8 : Notification page



Caption: The screenshot of the setting page that allows the user to customize his notification, who can access his information, and his export history.

The purpose of the setting page is to customize the application for the user.

The user can turn on/off the settings, export his meals history and read the privacy policy. The privacy policy implies information on how the users personal datas are protected from the public.

3.5 Exiting the System

To exit the application, the user has to click the home or back button of the android phone. Also, it is possible to exit by closing the tab in the android application.

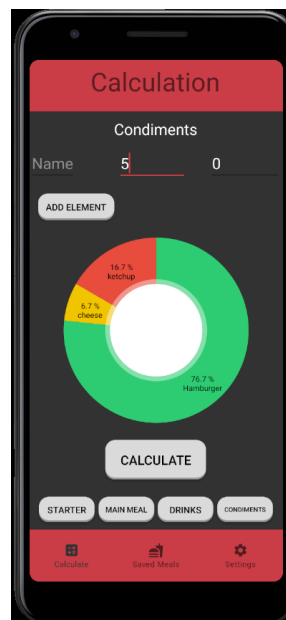
4 Using the System

This section will cover every feature provided from the Mealtime Insulin Calculator. Any important considerations will be covered in the below section.

4.1 Calculate

The purpose of this function is to add all the food's carbohydrates and fibers in order to calculate the total amount of carbohydrates. For this section, it is important for the user to input the food name, grams of carbohydrates and grams of fibers. If any of the following information is not added, the food will not be added to be calculated. The user can verify if the food is added by observing the pie chart and see if their food was indeed added to the pie chart. The pie chart gives a visual representation of the carbohydrates for the meal.

Figure 9 : Main page of the application



Caption : A screenshot of the main page showing where to input the name, grams of carbohydrates and fibers at the top of the image.

4.1.1 ADD ELEMENT

This function adds the information of Name, Carbohydrate, and Fiber as a Food. The local database stores the information of each meal in order to be able to keep track of amounts of carbohydrates and fibers.

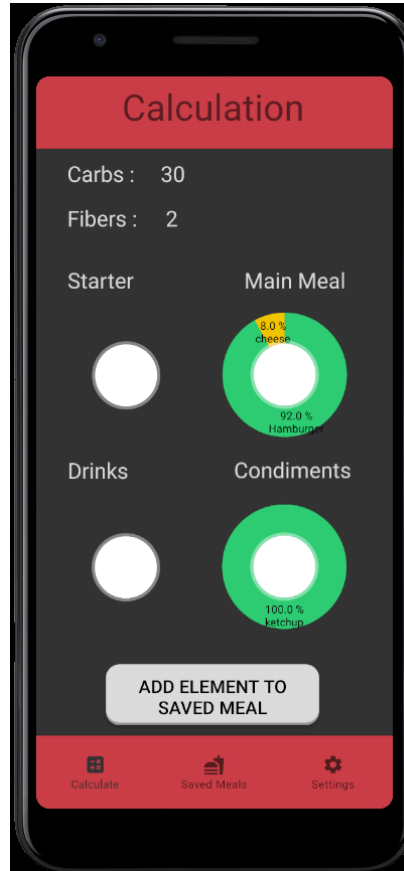
4.1.2 CALCULATE

This function will take the information stored in the database from the add element function to take the total amount of carbohydrates and fibers. This function will then send the user to a different page with all the calculations.

4.2 Calculation

The calculation section is a section that shows the total grams of carbohydrates and total grams of fibers. This section also includes 4 different pie charts displaying the different types of foods selected by the user when they added their foods in the previous section (4.1). This page has the option to save the meal in the database.

Figure 10 : Calculation Page

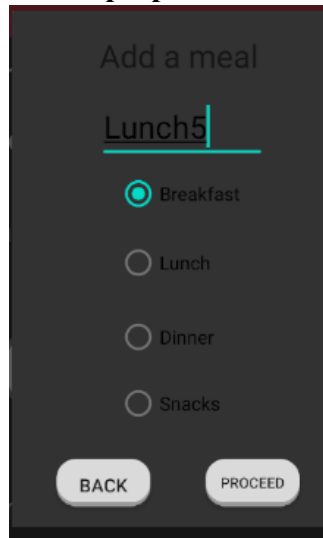


Caption : This is a screenshot of the calculation page where it is possible to see the different pie charts.

4.2.1 ADD ELEMENT TO SAVED MEAL

This function opens a prompt to allow the user to save their meals in the wanted category. This section asks the user to input the name of the meal they want to save and the category they want the meal to be saved. Pressing back will cancel adding the meal, while proceed will add the saved meal to the database.

Figure 11 : Pop-up when adding meal



Caption: A screenshot of the process by which the user can enter the type of meal he has eaten.

4.3 Saved Meal

This section allows the user to go and choose previous meals to save time. This allows the user skipping recreating meals that were already created in the past. The page allows the user to select the category they saved their meals.

Figure 12: Saved Meal page Categories

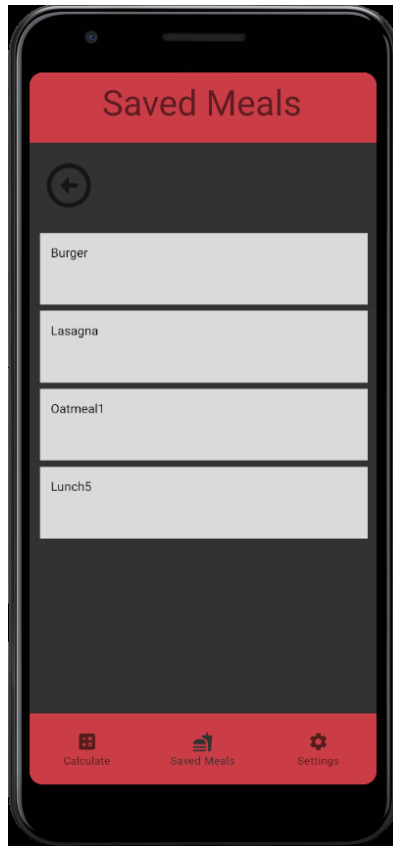


Caption: This is the different categories the user can save his meals (i.e. Breakfast, Lunch, Diner or Snack)

4.3.1 Saved Meal Options

When clicking on a category, the user can then select the meal they want to select. When selecting the wanted meal, the user is brought back to the calculate page in order to add any foods they want, or to be able to calculate the total amount of carbohydrates.

Figure 13 : Saved Meals in a Category

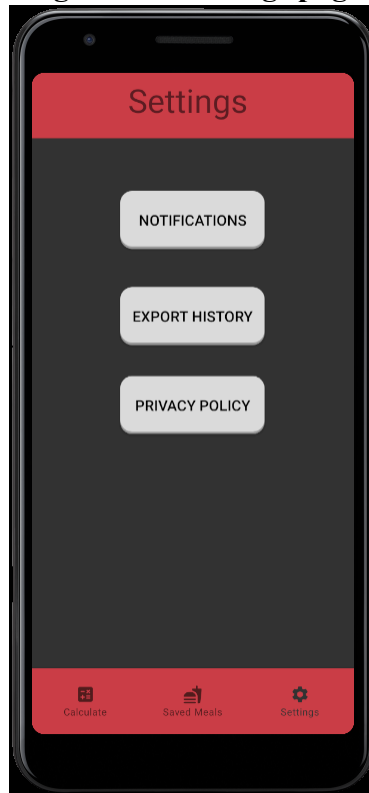


Caption: A screenshot of the saved meal of the user so that he can use them later on.

4.4 Settings

This section is made for future implementations. The current prototype does not have any real functionality implemented. The different options that are planned to be added in the future are still described below. There are three different options: Notifications, Export History, and Privacy policy.

Figure 14 : Settings page

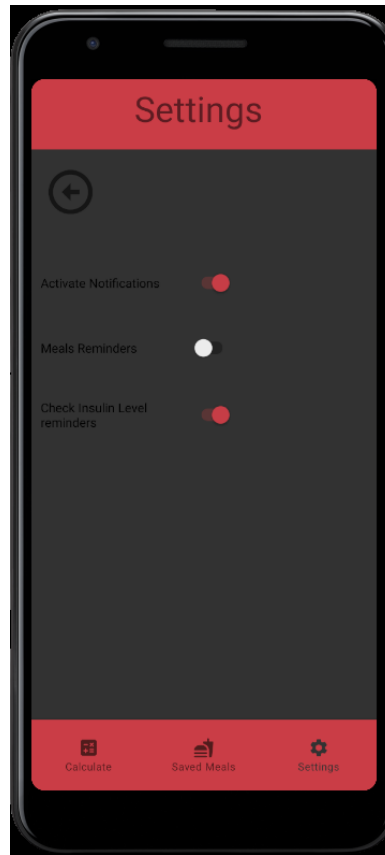


Caption : this is the settings page. On the setting page there are the notifications, the export history and the privacy policy.

4.4.1 Notifications

The Notification options contain three future implementations. The three options can either be turned on or off. Notifications add the ability to have reminders about adding condiments to meals. This helps the user remember to add hidden carbohydrates like ketchup (a food that is sometimes forgettable when adding up carbohydrates).

Figure 15 : notifications Page



Caption: A screenshot of the notifications page for the Mealtime Insulin Calculator app

4.4.2 Export History

The second option is Meals Reminder, which gives reminders to eat after a certain amount of time has gone without using the application. After a long time of not eating, a diabetics blood glucose can start changing. The reminder can help the client remember to eat and take insulin.

4.4.3 Privacy Policy

The last option is Check Insulin Level Reminders, which similarly to the previous option, gives reminders to the user about taking their blood glucose after not using the application for a while.

5 Troubleshooting & Support

The following sub-sections provide detailed instructions on how the error messages and behavior of the application work. Then how the maintenance of the application is done as well as how to contact the support for help from the team..

5.1 Error Messages or Behaviors

The application uses a certain type of messages called “Toast messages“ to handle unwanted behaviors. These messages appear at the bottom of the screen for a few seconds and they tell the user what needs to be done so that the application can run properly. The application uses those messages especially when there is information to enter, for example when the user wants to calculate all their entries, they have to click on a button to move to the page showing the total number of carbohydrates and fibers. But if the user forgets to enter an entry, the application cannot proceed properly and will then display a message stating that a meal must be added before clicking on the button. The same thing happens when trying to save a meal without its name as the application needs the meal name to store it properly. A message is also shown when the user tries to access a feature that isn't there yet, it would state that the feature isn't there yet so that the user isn't confused.

See Appendix II

5.2 Maintenance

Since the product is an application that isn't available on the play store, the user should check the github from time to time to see if the application has new versions that would add features or fix bugs. If the user finds any new bugs, the user is recommended to look if a new update is available.

5.3 Support

For general information and questions, the user should read the privacy policy first and the readable section on github. If that doesn't answer their questions, they can contact support via email at mealttimeInsulinCalculator@gmail.com.

If the user needs technical assistance, the client must first give information about their device, the version of their device and the version of the application they are using. Then they must properly explain what the problem is and how to replicate it. Then they would have to explain their usage of the application before the issue appeared so that the team can replicate the problem to understand and solve the issue.

6 Product Documentation

6.1 Software

6.1.1 BOM (Bill of Materials)

Table 1: Bill of materials

Material	Price (\$)
PC/Laptop	0 (Everyone in the project already had access to a computer)
Java SE	0

6.1.2 Equipment list

- Android Studio (<https://developer.android.com/studio/>)
- SQLite (<https://sqlite.org/index.html>)
- JAVA SE (<https://www.oracle.com/java/technologies/javase-downloads.html>)
- MPAndroidChart (<https://github.com/PhilJay/MPAndroidChart>)

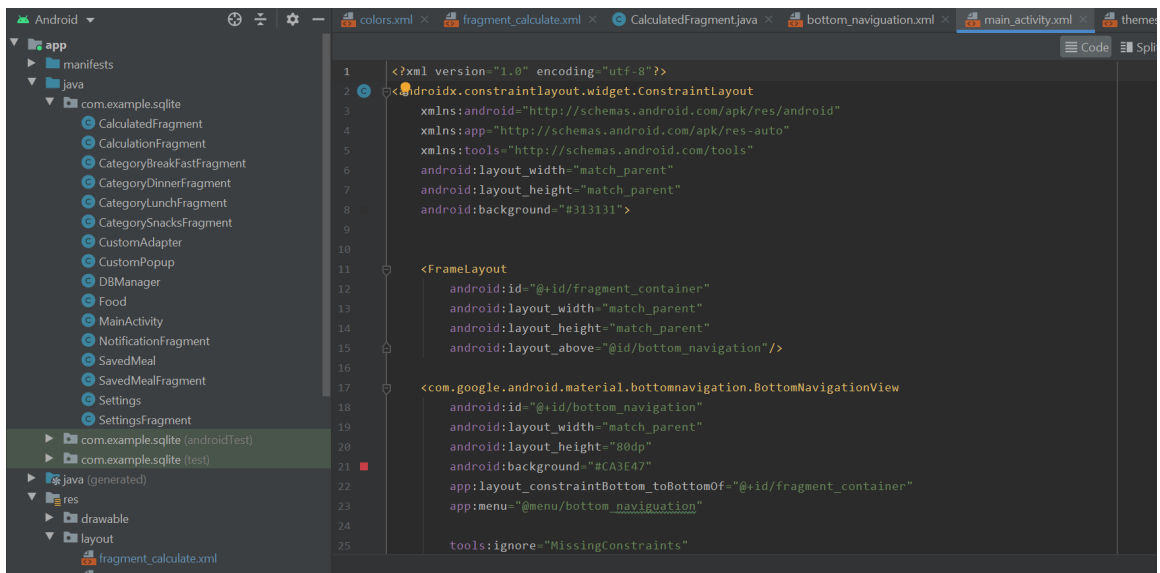
6.1.3 Instructions

All parts of the software are accessible as open source in the following link:
<https://github.com/Mealtime-Insulin-Calculator/Mobile-application>.

The JAVA files are all found in the following link:
<https://github.com/Mealtime-Insulin-Calculator/Mobile-application/tree/main/app/src/main/java/com/example/sqlite>.

In order to build the prototype, the files can be opened using Android Studio. The following should look like the following screenshot:

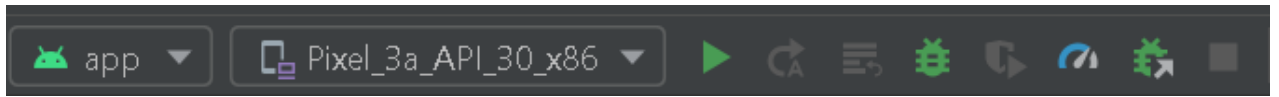
Figure 16 : Screenshot of the files opening with Android Studio



Caption: Screenshot shows various sections of code for the calculator on Android Studio

Using Android Studio, an emulator can be used directly from Android Studio in order to test the prototype.

Figure 17: Screenshot of Android Studio - Emulator section



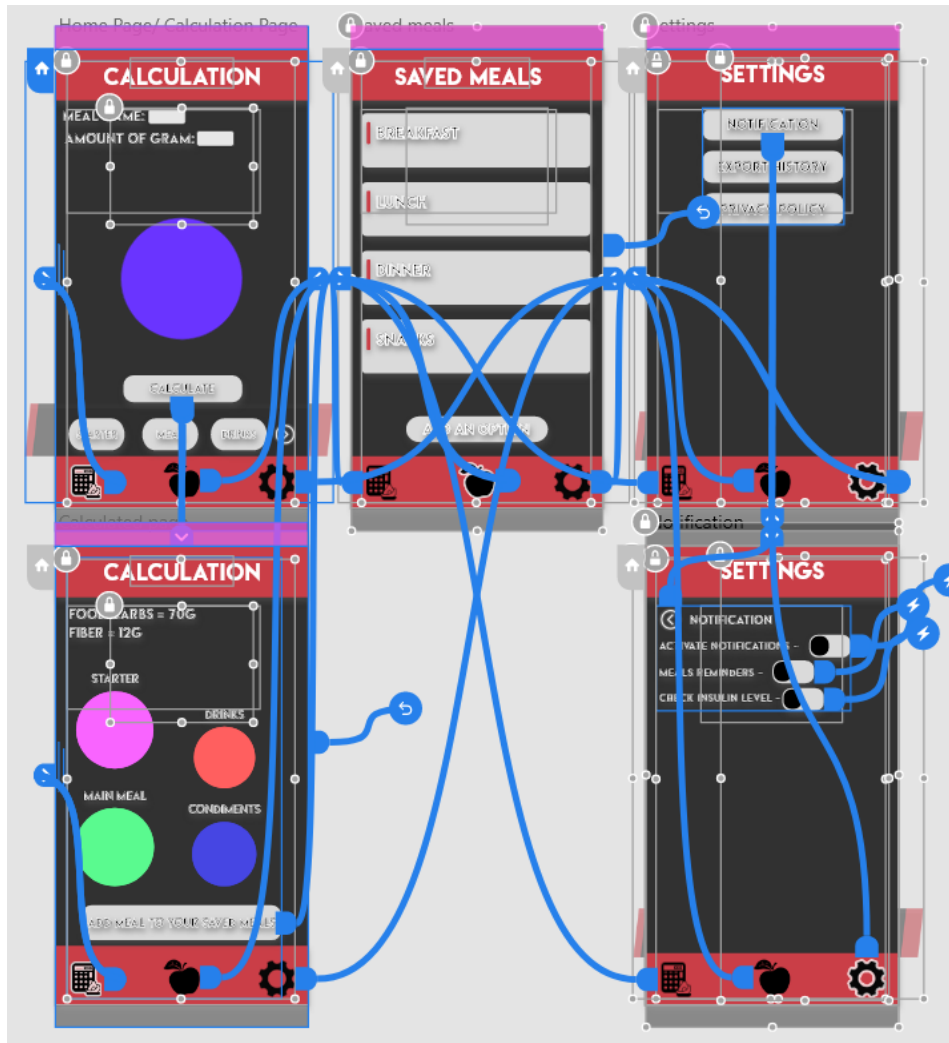
Caption: Android Studio provides different models of phones to select from for the emulator, as well as various tools for the emulator.

The green play button allows the user to activate the emulator and run the code on the emulator. If the emulator is not working on the device, the user can plug in their device to their computer, and select their mobile device with the selection box on the left of the green play button. This will allow the user to test the application directly on their mobile device. Not only will it test the application on the mobile device, it will also automatically install it, allowing the user to use the application without being connected to the computer anymore.

6.2 Testing & Validation

6.2.1 Prototype 1

Figure 18 : Picture of the Prototype Used for Analysis



Caption: Prototype 1 is a wireframe that shows the connection between each function in the application.

Note: The link to the prototype:

<https://xd.adobe.com/view/ca0e435d-6d6b-4bbb-a74f-1edb37a89926-9f5d/?fullscreen>.

Prototype 1 was used to test the foreseen simplicity and practicality of the prototype. This section will incorporate a table summarizing the analysis of the prototype and analysis of the prototype.

Table 2 : Comparison between the Target, the Prototype before Meal Bank, and Prototype after Meal Bank.

Target Specification	Foreseen (before)	Prototype (early after)	Prototype (after)
Simplicity (# of questions)	4	6-7	2-3
Practicality (Subj (low-high/1-5))	3-4	2	4
Customer needs (# of customer need)	7	N/A	8

Note: To be noted that the number of questions is considered on a per food basis. Green represents exceeded expectations. Yellow represents satisfied requirements. Red represents unsatisfied requirements.

Below, the different aspects of the table will be discussed in detail for better understanding to improve the project in order to better future iterations of prototypes. The first section will refer to the values given to each section of the table while the second section will discuss the considerations for future prototypes.

The foreseen(before) column represents the values that are aimed at the target specification. These values were used from previous analyses of other groups that have done this project as well as professional applications on the play store.

The prototype (early after) represents the usage of the application with no prior use of the application. It can be noted that the prototype (after) column reflects on the prototype (early after) in the sense that the values go from worse to better. For simplicity, more questions have to be asked before the application stores any foods in its memory. Because of this, a higher number of questions is necessary. These questions include the food name, carbohydrates, fibers, picture,

selection, carbohydrate meal specific correction, and fibre meal specific correction. The first four questions are specific to storing the meal information, while the last two questions are centered on entering food into a meal.

Note: fibre meal specific correction is considered as an uncertainty, due to the unlikelihood of changing this value hence the margin 6-7.

The practicality of the prototype (early after) is not as good as the team wishes the prototype to be. Using the application constantly adding foods/meals would make the idea of the application absolute because the application would be adding workload to the client when taking care of their diabetes. By having a couple of extra questions, the usage of the application can be made easier. The addition of the meals while eating a meal is less practical than the goal of the application giving it a value of 2 (lower than the targeted 3-4).

The prototype (after) column represents the usage of the application after having added every meal that the client will be eating on a daily basis. Only two-three questions need to be asked to the client in order to have the information necessary for each food being the food name, carbohydrate meal specific correction, and fibre meal specific correction.

Note: fibre meal specific correction is considered as an uncertainty, due to the unlikelihood of changing this value hence the margin 2-3.

The final value given is the prototype (after)'s value for practicality. For this section, the targeted value is compared to past group teams and play store applications. The value that the team gave those applications come to 2-3 in most cases. The goal for the application is to do a bit better (3-4). When all the necessary meals are entered in the meal history bank, the application becomes more practical for the user and comes to the highest range of the target the team have for the application. The application also shows pictures of the foods, increasing the practicality of the application to a 4.

Using *GNG2101 Report: Deliverable B Mealtime Insulin Calculator - Needs Assessment, Problem Statement, Metrics, Benchmarking, and Target Specifications*' table 2: Need statements obtained from the client and evaluated according to their importance, the number of client needs met can be calculated. The target specifications have given this value a 7. It would not make sense to evaluate the prototype (early after) column, because there is no difference to the prototype (after) as both are the same applications. The other target specifications looked at have some differences to be mentioned. Every client need was looked at (in table 2.0 below), given the value of 8 for the current satisfied client needs met.

Table 3 : Representation of the Client Needs Met by the Prototype

#	Needs	Importance (1-5)
1	The application works on the most recent version of android.	5
2	The application calculates the amount of carbohydrates from the meal information.	5
3	The application must be able to calculate the amount of carbohydrates for beverages, condiments, and all the other foods that could contain carbs in less than 2 minutes.	5
4	The application provides a simple, user-friendly design for the interface, one that is easy to navigate.	4
5	Application incorporates a meal history that stores information about previous meals to make the application usage faster and more accurate to the carbohydrate intake.	4
6	The meal history can be exported to an excel file.	4
7	Push notifications give notifications for mealtimes and reminders about hidden sugars.	3
8	Overlapping picture of the meal and a pie chart to show the different foods.	1
9	The application should be available to everyone on Android with a cost of \$0.	3

Note: Green represents met client needs. Yellow represents exceptions, details explained in text above. Red represents client needs not met.

Prototype Evaluation

Using the values gathered in table 1.0, an evaluation can be made to improve future prototypes. The following section will discuss what information can be gathered, analyzed, and changes to be made.

Looking at the prototype (early after) column, each row is marked in red showing further analysis to see potential concerns/improvements. The prototype (after) column shows either exceeding or up to standard. From this point, the table is a mixture of good aspects and not as good aspects.

Further analysis of the table shows promising results for the project. The initial red can be explained from the initial trade-off (more questions) resulting in fewer questions in the long run than the anticipated number values. The team expects the user to use the meals stored in the meal bank more than once. The client has mentioned that storing the meals would be nice as their meals are not too diverse. Giving this information, it can be assumed that foods will be used multiple times, giving overall usage of the application fewer questions with more application usage time. The second part that can be analyzed is the practicality that can be observed in the table. For this section, the practicality depends highly on the implementation of adding the meals in a meal bank. Once again, the high initial pay-offs are estimated to be worth the higher practicality over long periods of time using the application.

Looking at the customer needs, a couple of considerations have to be made. First of all, one of the client needs was morphed into a different function that the client wants as well. This morphed need is the need for an overlapping pie chart over the food. Instead, the team decided to use pie charts for visual simplicity where the client reacted with positive feedback to this novel

change. The reason it is marked as 8, is considering this slight change of perspective and acceptance of the client during the last client meet.

Prototype 1 is accessible using the following link:

<https://xd.adobe.com/view/ca0e435d-6d6b-4bbb-a74f-1edb37a89926-9f5d/?fullscreen>

6.2.2 Prototype 2

This section will be given to test the foreseen functionality and feasibility of the prototype. This section will incorporate a table summarizing the analysis of the prototype and analysis of the prototype. Please refer to *Project Deliverable C: Group C11 Conceptual Design, Project Plan* for more details on the target specifications.

Table 4 : Comparison between the Target, the Prototype before Meal Bank, and Prototype current Meal Bank.

Target Specification	Foreseen (before)	Prototype (current)
Likelihood of success(Subject (low-high/1-5))	4-5	4.5
Feasibility(binary True/False)	True	True
Time to Create Function (estimation in days)	<8	5-6

Note: To be noted that the number of questions is considered on a per food basis. Green represents exceeded expectations. Yellow represents satisfied requirements. Red represents unsatisfied requirements.

Below, the different aspects of the table will be discussed in detail for better understanding to improve the project in order to better future iterations of prototypes. The first section will refer to the values given to each section of the table while the second section will discuss the considerations for future prototypes. The first section will refer to the values given to each section of the table while the second section will discuss the considerations for future prototypes.

The foreseen(before) column represents the values that are aimed at the target specification. More information can be found in *Project Deliverable C: Group C11 Conceptual Design, Project Plan*. These values were used from the prototype testing and past evaluated values.

Feasibility includes the ease the team has to implement functions and to be able to surmount challenges when going through new software or libraries in order to get the final product. In this product's case, some of these initial challenges/barriers include SQLite, Android Studio and Pi Chart libraries. Currently, the SQLite and Android Studio have directly been discovered by the team and thus can be evaluated for feasibility.

The Foreseen (before) column represents the initial vision of the final product. The prototype (current) column represents the usage of the application after having added every meal that the client will be eating on a daily basis. The database will be full of the client's personal meal choices. In this case, the usage of the application will take less time and will function better for the user.

When considering the likelihood of success, it is important to have a point of reference to properly evaluate a subjective value, otherwise, the subjective value will have no meaning and thus no application to future implementations. In this case, it is important to note that the point of reference is the envisioned ease of implementation of the functions and structure of the project. The value was given a 4-5 meaning that the team would have ease in implementing functions and

making the project work. With the second prototype, there were no major issues that made the project impossible to progress. Considering the time it took to create the classes to store information of foods and meals, and then implementing an SQLite database that can store this information, the team is comfortable putting a value of 4.5 for the likelihood of success for the current implementations.

When considering the feasibility, it is important to note that only the foreseeable future can be considered. In the initial envisioning of the project, the team has assessed that it would be true that the group could complete the tasks at hand. For the current prototype, being able to properly implement the functions and having so many resources on the internet that have already helped a lot with the current development, leads the team to believe that the feasibility of the project is still true. Due to the accessibility to the information needed for this project, the feasibility of the current prototype can be evaluated to be true.

When looking at the time it takes to create functions, the prototype shows much faster development. The initial idea would be that the time for programming the functions themselves would take about eight days leaving the rest of the time to explore Android Studio and new libraries. The second prototype has shown that most of the main functions were programmed in less than 4 days. Another 1-2 days is expected in order to make changes as the development makes progress. This value is a mix of the given information gathered from making the second prototype and the projection of future need for implementations. Technically, the functions are done and connecting the work with Android Studio and SQLite is the majority of the rest of the project (which is not counted towards creating the functions themselves). The 1-2 days is added just in case changes are needed. For these reasons, the time to create a function was given a value of 5-6 days.

Prototype Evaluation

Using the values gathered in table 1.0, an evaluation can be made to improve future prototypes. The following section will discuss what information can be gathered, analyzed, and changed for the final prototype.

From both the foreseen (Before) and prototype (current) columns, every row is marked showing that the specification either met or exceeded the expectations the team had. This analysis shows promising results towards the final prototype for the project.

Looking at the first target specification, the 4.5 indicates that challenges seen in the interactions of the implementations of the software are within the group's skill range. Knowledge or capabilities are not foreseeable barriers to the development of the product. The interactions between SQLite, classes, and Android Studio are doable and give good feedback on the implementations and different perspectives for future implementations. In the project, the class SavedMeal was changed from an ArrayList<String> to an ArrayList<Food>, showing an improvement in implementation (using the Food class instead of String).

The second target specification analysis shows if the project is feasible. Being able to know if certain functions are feasible in the first place is important. If the SQLite would be seen as not working at all, or the group would lack the comprehension to implement, the project would become unfeasible until a new course of action and a new prototype (with a different approach) can be created. In this case, seeing that the second prototype is feasible, means that there is nothing to change in that aspect. The purpose of a feasibility check is to make sure that the project is going towards progress. If it would have been negative, the prototype would have been further analyzed in more detail on why it failed and a new prototype would have been created.

The last target specification analyzed was the time to create function target specification. This target specification was set with 8 days as a maximum in the hopes to leave the rest to learn how to create the proper implementations for the project. So far, considering that most of the

functions are done and thus mostly the implementations of the functions are needed, the project is seeing progress at a steady rate.

For the critical assumptions, it is important to note that the previous target specification analysis summarizes the findings for those critical assumptions. The first critical assumption (functionality factor) was evaluated using likelihood of success and feasibility, thus the conclusion of the two target specification analysis is the conclusion to the first critical assumption analysis. The second critical assumption (implementation time for the project) was evaluated using the time to create functions target specification.

6.2.3 Final prototype

The final prototype was created through the improvements of the previous prototypes. It is to be said that this, thus, application does not need any special requirements for sustainable usage. Issues might happen with the size of the apk file. The final prototype is made in order to get a product that can be used for evaluation. The final prototype is essentially the addition of prototype 1 and 2 together. The evaluation on both prototypes is applicable and relevant to the final prototype. The only difference that can be concluded from the final prototype is the satisfaction of the client. One of the team members has type one diabetes and was able to test the application. The client also gave feedback on the final prototype.

By looking at the feedback, the team is able to conclude the assumption made from the team. The improvement from the previous prototype is successful is the assumption being made for the final prototype.

Both the client and group member with diabetes got to see the application first hand and its features. Both agree that the application does what it is meant to do. The client was very satisfied and really liked the pie chart giving a visual representation. For this prototype, the prototype is analyzed with a true or false value. Because the prototype is focused on concluding the success of

the implementation and making sure the design is making great positive progress and stays feasible and within the ability of the team to create, the analysis for the final prototype will be straight forward.

Table 5 : The number of questions

Unit of Measurement	Expected values	Obtained Values
# of questions	2-3	3
Functions added (amount)	4	4
Likelihood of success (Subj. 1-5/low-high)	5	4.29

In the table above, it can be observed that every category is in range. Each of these will be described in the paragraphs below.

The number of questions being evaluated, are the questions needed to add a food to the calculation. Looking at the following figure, it can be observed that there are only three questions in the final prototype :the name of the food, the grams of carbohydrates, and the grams of fibers.

Figure 19 : Calculation Page Questions



Caption : Screenshot showing that only three questions are used in order to calculate a food into the meal.

It is important to consider that the amount of questions is consistent with previous prototypes and previous evaluations of what would be wanted to be implemented. The final prototype is useful to get a conclusion if the previous evaluations are accurate. In this case, before there were no pie charts, this prototype confirms that having only three questions is still viable for a properly implemented pie chart prototype version.

The second part being evaluated is the added functions. For the final prototype of the project, the team knew that there would not be enough time to put 100% of the implementations the client would have liked. Some implementations would have included creating an export system, pie chart with pictures of the foods and on. The team concluded that they needed to focus on realistic expectations and amount of final features. The features that were focused on were: Pie charts, adding meals, saving meals, reusing meals. In the final prototype, the team was able to achieve this goal to make all of these features work. The other implementations that the team

concluded that time would not allow to have so many implementations, were added in the future works section. Looking at the previous table, the final prototype concludes that what the team expected to be able to be done, was feasible. This conclusion confirms that the project is in the right direction.

The last test that was done consists to analyze the likelihood of success. For this value, the team used a formula to calculate numbers to represent potential success of the project. The expected value is set to be the max in the following formula:

$$\frac{(3 - \text{Best case scenario for \# of questions}) + (\text{\# of functions added})}{\text{Total expected value}} * 5$$

Where the total expected value is :

$$\begin{aligned} & (5 - \text{Best case scenario for \# of questions}) + (\text{\# of functions added}) \\ & = (5-2) + 4 \\ & = 7 \end{aligned}$$

The formula is set up so that the expected value will be a value of 5. The project is expected to be feasible. Using the previous formula, the obtained value give the following:

$$\begin{aligned} & \frac{(5 - \text{\# of questions}) + (\text{\# of functions added})}{\text{Total expected value}} * 5 \\ & = \frac{(5-2)+4}{7} * 5 \\ & = 5 \quad * \text{The value calculated for the expected value using the best case scenario} \end{aligned}$$

The next calculation shows the result of the obtained value for likelihood.

$$\begin{aligned} & \frac{(5 - \text{\# of questions}) + (\text{\# of functions added})}{\text{Total expected value}} * 5 \\ & = \frac{(5-3)+4}{7} * 5 \\ & = 4.29 \quad * \text{The value calculated for the obtained value} \end{aligned}$$

These numbers are relevant because they demonstrate the best case scenario. The difference in the answer comes from the amount of questions. Even though the value is lower, the team does not consider it to be any concern due to its still relatively high value. The value is close to the previously hoped subjective value of 4.5. The team decided to implement a way to calculate the progress. This value is relevant because it is a conclusive value that allows the team to non-subjectively conclude that the likelihood is high (This formula can stay consistent/modified with newer implementations in order to evaluate the relative likelihood of success). The 4.29 value stands in the range of 4-5 likelihood of success.

From these tests, the most important consideration to take from the final prototype is to make sure that the team keeps working together. The previous numbers/tests show that the team is

successful in completing the objectives. Although the team is able to finish all work in time, sometimes completing the work can come close to the deadline. It can thus be concluded from the numbers and tests that the final prototype is valid and closer to the final product.

7 Conclusions and Recommendations for Future Work

Through the development of the mobile application the team learned some really useful lessons concerning time management, especially considering the fact that the team needed to learn new tools that were never used before and cost more time than simply starting the application creation with tools already used by the team. The team decided to use android studio, but other technologies such as progressive web application or software like flutter were available too. Those are avenues the team should have explored before settling on android studio as it could have greatly changed the workflow for the better. If the team had in his disposition more time in the process some improvement on the existing features could have been done and other features discussed during the elaboration of the application could have been added. Since, the time was limited only the main features were implemented in the application. Some features ideas elaborated by the team were options for customizing the application to the user linking, the ability to have meals linked to an account so that the user can access his meal on another device or even a calendar mode for the saved meals.

8 Bibliography

Build software better, together. (2021a). GitHub.

<https://github.com/Mealtime-Insulin-Calculator/Mobile-application>

Build software better, together. (2021b). GitHub.

<https://github.com/Mealtime-Insulin-Calculator/Mobile-application/tree/main/app/src/main/java/com/example/sqlite>

C11 -Mealtime Insulin Calculator | MakerRepo. (2021). C11.

<https://makerepo.com/JCMSpencer/826.c11-mealtime-insulin-calculator>

Download Android Studio and SDK tools |. (2021). Android Developers.

<https://developer.android.com/studio/>

Montegriffo, N. (2020, November 2). *What is an APK file and how to install APKs on Android?* NextPit.

<https://www.nextpit.com/android-for-beginners-what-is-an-apk-file>

P. (2021). *PhilJay/MPAndroidChart*. GitHub. <https://github.com/PhilJay/MPAndroidChart>

SQLite Home Page. (2021). Insulin Calculator Team. <https://sqlite.org/index.html>

APPENDICES

9 APPENDIX I: Design Files

Table 6: Referenced Documents

Document Name	Document Location and/or URL	Issuance Date
PD D	Maker Repo	February 7th
PD F	Maker Repo	March 7th

10 APPENDIX II: Other Appendices

Table 7 : Error Messages or Behaviors and how to correct them

Scenario	Error message	Causes	Corrective actions
Forced update (Latest version)	App update required	The user needs to update the application before continuing	Update the application
App update (skip and update later)	New version available	The user can apply the update now or after to have the newer version	Update the application when needed
System error	Sorry! Something went wrong.	the system has encountered a bug	None
Scheduled maintenance	Schedule maintenance in progress	Need to update and verify the good functioning of the application	None
Adding food with missing information	Oups...you forgot to add an information	The user forgot to add an information	Add the missing information
Forgetting meal name	Oups...you forgot the meal name	Saving the meal but the user hasn't entered the meal name	Add the meal name
Settings notification	This feature is still in developpement	The settings are yet to be programmed. Therefore, the user can not use them	None

Maker Repo:

<https://makerepo.com/JCMSpencer/826.c11-mealtime-insulin-calculator>

Github App Repository:

<https://github.com/Mealtime-Insulin-Calculator/Mobile-application>