Project Deliverable F: Prototype 2
GNG 2101

Submitted by Team C11

Khalil Aouadi, 300197227
Spencer Henry, 300073281
Mounira Nihad Zitouni, 300190536
Regina Mayani, 300207233
James Couture, 300076065

January 31st, 2021
University of Ottawa

Abstract

The following document for this deliverable is the second prototype of team C11's project. This document will contain the client's feedback on the concepts designed for deliverable D along with all the changes and improvements made to the concepts, the critical assumptions of the product, images and details of the prototype as well as the results from the tests that were run for this deliverable.

Introduction

After the third meeting with the client, the team moved on to deconstructing the feedback given by the client concerning the prototype concepts. Then, the team had the mission of providing a more detailed design to the current concept. The team created a medium-fidelity prototype to be able to test the product's most critical functionality and compare the prototype to the target specifications. By comparing the prototype to the target specifications, the team can determine improvements to the prototype for future iterations.

The objective of this document is to provide insight on the client meeting, prototype, testing and evaluations. Having these components will allow the team to take the next steps towards creating the finished product and reach an end goal.

Client feedback

For the third meeting, we presented a low-fidelity prototype to the client. The client was very pleased with the results. He found the prototype user-friendly and he didn't get lost in all the features presented to him. The team's objective of making the application easy to use was met with the low fidelity prototype. However, the client found that the choice of font was very hard to read, so changes to the application for ease of readability will be made. Unfortunately, the first prototype did not have the font of the initial design implemented. This means that the wanted font could not be tested for its clarity, but did let the team know that the font is something important and will need to be implemented in a future iteration. The second prototype, which is a medium-fidelity prototype, has colors and texts that are easier to read than prototype 1. The team still hopes to make the font more readable in future iterations. Another solution the team found was to make the text bigger to help the client read the small text.

During the client meeting, the team asked the client where he wanted the database to be placed. There were two options:

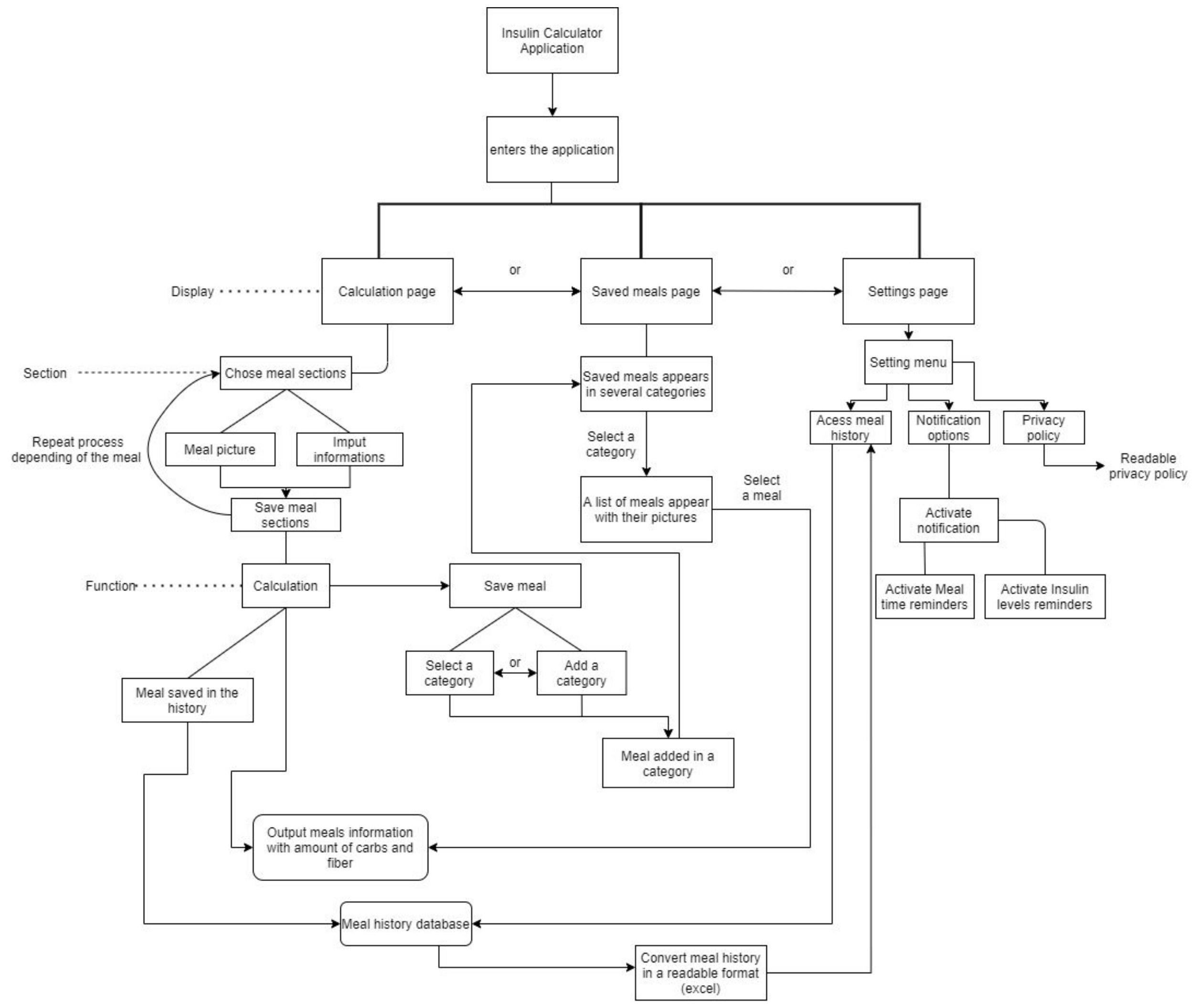
1) In his cellphone where it would weigh more and take more memory storage or,
2) in an outside database, which makes the weight lighter but the app would need an internet connection everywhere to access it.

The client chose the first option. For the team, it implied that it wasn't needed to use firebase but, instead, SQLite can be simply used to make the local database. Then, we have proposed to the client two options regarding the disposition of the different pictures. The first option was that there would be a picture that englobes all the meals at once and the second was different pictures for the different constituents of the meal. The client told us that they would be interested in the two options and they would rather want to be able to choose. Therefore, for a future prototype, the team will program a sidebar that will be hidden on the main page with a key to press on. If the key is pressed, the sidebar would slide in the middle and show the different pictures of meals and/or constituents of a meal when selecting foods for the meals. Overall, the client was very pleased with the prototype.

Design and flowchart updates

For the design, the client liked the design and the ideas we had for the final prototype, however, the client found it hard to read the text in the application so a change to the font and size is necessary. As for the functional decomposition flowchart, we added more information on the subsystem and detailed the various functions.

Figure 1.0: Flowchart/Diagram of the Decomposition of the Application



Most Critical Product Assumptions

In this section, the most critical product assumptions will be discussed and analyzed. It is important to consider that these assumptions are mostly relevant to the prototype being developed for this deliverable and future iterations of prototypes.
The past assumptions from deliverable D are still valid for this prototype. However, the most critical product assumptions that are being evaluated for prototype 2 are as follows, proper interactions between functions (functionality factor) and the feasibility for the team to be able to complete the project in a timely manner.

The functionality factor is the first critical product assumption that will be analyzed. The application must make sure that both the SQLite local database and the objects are able to be synced and properly communicate the information in order to calculate and properly portray the information on the application for the client. The team has assumed that the SQLite package is compatible with Android Studio and has also assumed that ArrayList will work with both with ease. Making the second prototype, the interaction between SQLite and the java files can be evaluated and analyzed for future improvements.

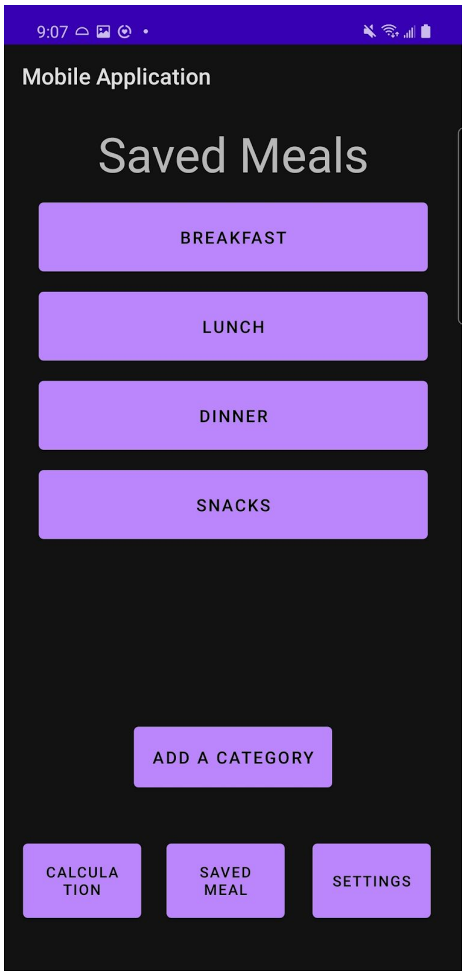
The second critical assumption is that the team will be able to make all the functions and implementations before the presentation date. The team has decided to evaluate the amount of time it will take to program and finish functions. The team will decide the most urgent features to program (database and classes). Depending on how many functions can be implemented in time, more time and resources can be invested in the visual quality of the application.

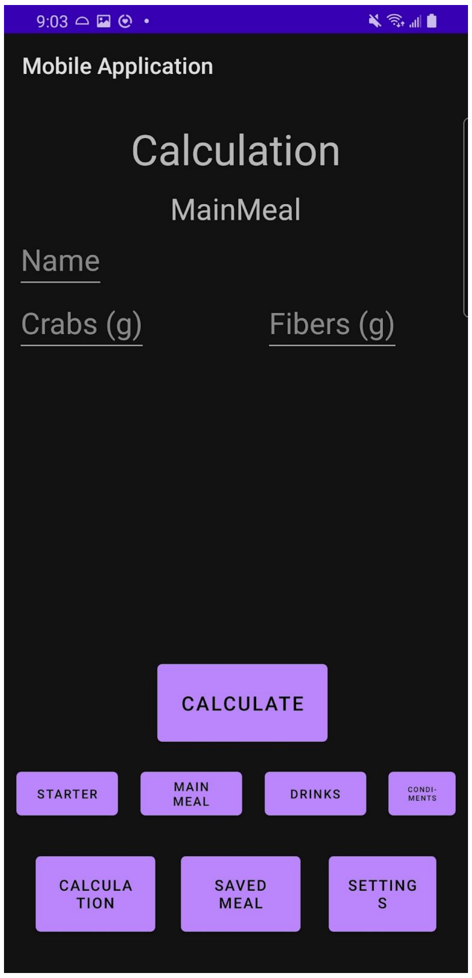
Prototype analysis

For the second prototype, the team has used Java to implement functions/methods to manipulate the data that is being communicated with SQLite to properly store and display information on the application. The initial PC version was simplistic but without any mobile app implementations. Later the code was transferred into Android Studio to be able to create an android app environment in order to get closer to one of the client needs, accessible on an android phone. The design of the writing is different than the shown first prototype, but will not be evaluated during the analysis of the second prototype. It was expected that the prototype would be able to add Foods to a Meal and be able to properly go through the pages and be able to visualise the implementations of the database. Since the second prototype is part of the code for the vision of the final product, the prototype is considered a medium-fidelity prototype.

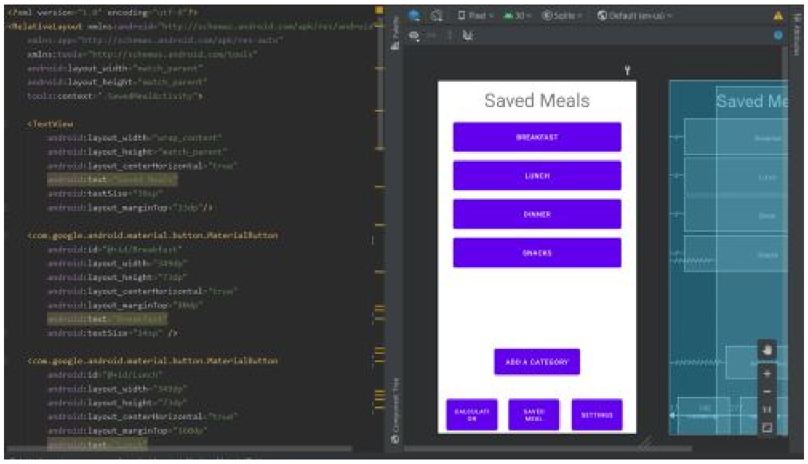
Prototype

Figures 2.0, 3.0: Pictures of the Prototype Used for Analysis



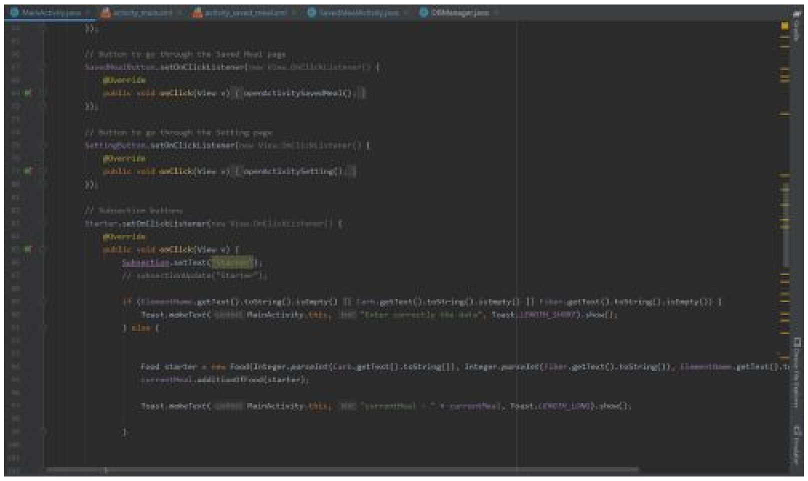


The design for the current application is very simplistic since the team put more of their focus on the code for the application then the aesthetics to ensure that the application gets done on time. Only two sections out of three have been implemented, the calculation page and the saved meal page. The calculation page can take 3 elements, which is the name of the food, the amount of Carbohydrates and Fibers it has, by then clicking on the subsection corresponding to the food, the application creates an object of that food that will be stored in one of the categories available in the saved meal page.

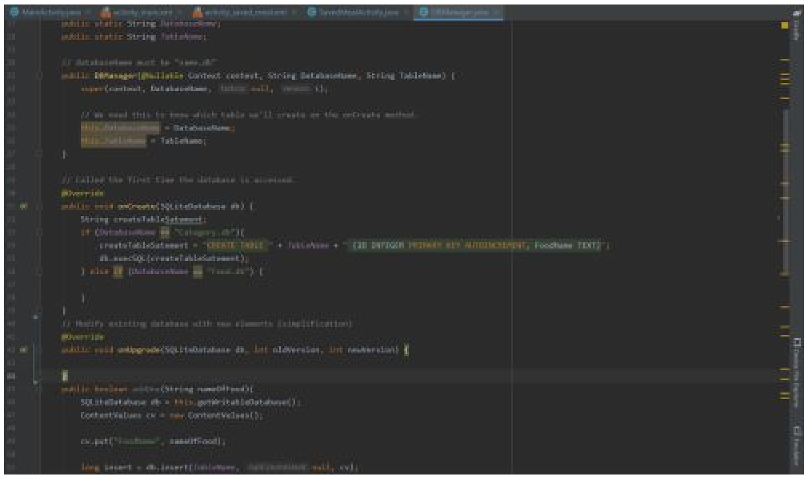


```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SavedMealActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_centerHorizontal="true"
        android:text="Saved Meals"
        android:textSize="30sp"
        android:layout_marginTop="15dp"/>

    <com.google.android.material.button.MaterialButton
        android:id="@+id/Breakfast"
        android:layout_width="349dp"
        android:layout_height="73dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="80dp"
        android:text="Breakfast"
        android:textSize="14sp" />

    <com.google.android.material.button.MaterialButton
        android:id="@+id/Lunch"
        android:layout_width="349dp"
        android:layout_height="73dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="160dp"
        android:text="[illegible]"
```

This is the design tool available in android studio. We use xml to make the elements and to align them as we wish.



```java
        });

        // Button to go through the Saved Meal page
        SavedMealButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { openActivitySavedMeal(); }
        });

        // Button to go through the Setting page
        SettingButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { openActivitySetting(); }
        });

        // Subsection buttons
        Starter.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Subsection.setText("Starter");
                // subsectionUpdate("Starter");

                if (ElementName.getText().toString().isEmpty() || Carb.getText().toString().isEmpty() || Fiber.getText().toString().isEmpty()) {
                    Toast.makeText( context: MainActivity.this,  text: "Enter correctly the data", Toast.LENGTH_SHORT).show();
                } else {


                    Food starter = new Food(Integer.parseInt(Carb.getText().toString()), Integer.parseInt(Fiber.getText().toString()), ElementName.getText().t
                    currentMeal.additionOfFood(starter);

                    Toast.makeText( context: MainActivity.this,  text: "currentMeal : " + currentMeal, Toast.LENGTH_LONG).show();

                }
```



```java
    public static String DatabaseName;
    public static String TableName;

    // DatabaseName must be "name.db"
    public DBManager(@Nullable Context context, String DatabaseName, String TableName) {
        super(context, DatabaseName,  factory: null,  version: 1);

        // We need this to know which table we'll create on the onCreate method.
        this.DatabaseName = DatabaseName;
        this.TableName = TableName;
    }

    // Called the first time the database is accessed.
    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTableSatement;
        if (DatabaseName == "Category.db"){
            createTableSatement = "CREATE TABLE " + TableName + " (ID INTEGER PRIMARY KEY AUTOINCREMENT, FoodName TEXT)";
            db.execSQL(createTableSatement);
        } else if (DatabaseName == "Food.db") {

        }
    }
    // Modify existing database with new elements (simplification)
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    }
    public boolean addOne(String nameOfFood){
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv = new ContentValues();

        cv.put("FoodName", nameOfFood);

        long insert = db.insert(TableName,  nullColumnHack: null, cv);
```

Shown above is a snippet from the code being developed for the application. Tthe first screenshot showcases the main activity thread which is launched as soon as the application starts. The second screenshot is a part of the class responsible for managing the database.

Note: The prototype link can be provided if need be

<u>Design Testing</u>

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

Target Specification	Foreseen (before)	Prototype (current)
Likelihood of success(Subj (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. The conclusion will thus be that as of March 7th, the team should have enough time to complete all the implementations by the presentation day in early April.

Future Implementations

Using the prototype to confirm two assumptions, the team was able to determine a couple of changes that should be made to better satisfy the target specifications.

Overall, the prototype shows that the team’s ideas are on the right track with either getting values that are within target specifications or better than the target specifications.

The first critical assumption, the functionality factor, confirms that the project is going in the right direction. The team should still consider creating different solutions in case an implementation is not feasible in a certain way. It is important to create second plans for things like pie charts if a certain library does not work.

The second critical assumption, implementation of the project, can also help find more intel on future iterations. Looking at the analysis of the second prototype, the team will consider putting more time into the implementation of the functions. The team has been too focused on making the functions, which works, but the prototype shows that there are functionalities missing from the vision of the final product. The missing functionalities in the prototype can be explained by the high yield in function design but lower yield in adapting the function design producing a bottle-neck effect. The team will need to take more time for implementation and less time for function design, as it takes less time than anticipated to create the function and will help to yield a more finished product sooner.

The third consideration for future change would be to program the notification buttons. In the low fidelity prototype, one of the main features that the client appreciated was the notification buttons. However, the team hasn’t programmed anything related to changing the settings or giving notifications yet. Unfortunately, this prototype was focused on making sure the project was possible with the current ideas of design. Therefore the notification implementations will be implemented in the next prototype.

A fourth consideration is the accessibility to the client’s camera. The client wants photos of his meals in the application. The current prototype does not have this option. Once this feature is programmed, it will be possible for the team to decide the placement of the photos. There are two options: The first one would be to place the pictures in the database. The second option would be to create a path to outside pictures and use those pictures inside the application. The team has yet to decide but will decide within the next few meetings.

The fifth future implementation that can be made is the export file. The client wanted to be able to export his personal information from the application. The team has not yet programmed this feature of the application. Also, The team will need to decide how the information will be presented on the exported document. It will be programmed at the end as it is not a priority.

The last consideration is the addition of the visual. The team will use android studio to program the main visual aspects of the application. The visual was already established from the previous prototype and design pictures. Therefore, the application will be a replica of the past prototype.

In conclusion, a couple of changes can be made from using the prototype. These changes include having multiple ways to implement functions, change in workload focus, programming the notification buttons, access to the camera, export file and add the visual. All of the changes can be solved with the team's current information. For the visual, if need be, the opinion of the client will be requested through email.

Conclusion

Through the analysis of the project, concerns were found considering the functionality, feasibility and the number of customer needs met for the prototype.

The client meetings the team has had up to this point have been extremely valuable. We intend to continue getting feedback from the client by providing APK files with the medium-fidelity application and maintaining good communication with the client to ensure that the high-fidelity application gets completed and satisfies the team and the target specifications. Before design day, the team will continue to improve the current state of the code for the application to be able to implement all the elements that are still missing. We will also continue testing the application for bugs and changes to be made and work on the design in Android Studio. Finally, we will prepare for the design day presentation using the feedback we got for the presentation for deliverable E.

Wrike Link

https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=vg1IgxM4n8qAhl1hnTeKZCgG9v1wgoAb%7CIE2DGNBSHA2TCLSTGE3A

Division of the work :

- James created a good amount of the code with the help of Mounira.
- Khalil, with the help of James, has been working on implementing the code inside Android Studio.
- Regina, Mounira and Spencer mostly worked on the report
- Spencer updated the Wrike.
- James and Spencer reviewed the report