

A Mobile Android Application to collect data for the treatment of Crohn's disease

Chuanfei Tang

2469509T

MSc Bioinformatics

Supervisor: Dr. Umer Zeeshan Ijaz

A report submitted in partial fulfilment of the requirements for the MSc

Bioinformatics Degree at The University of Glasgow

August 2020

Contents

Summary	1
Acknowledgements	1
Abbreviations	2
Introduction and Aims	2
Analysis	5
Program predesigns	5
1. User interface (UI) :.....	5
2. Program background:.....	5
3. Database:.....	6
Programming Language.....	6
Program used.....	6
Android Studio	6
MySQL	7
Google Apps Script	7
Postman	8
Google Spreadsheet	8
Product	8
Program structure	8
Android application.....	9
Google App script.....	12
Google Spreadsheet	13

Implementation.....	14
LoginActivity.....	14
MainActivity.....	16
AddActivity.....	20
InputActivity and InputActivity_M.....	25
Evaluation.....	26
Conclusion and Discussion	28
References.....	30

Summary

Crohn's disease (CD) is a kind of intestinal inflammatory disease with complex aetiology. The course of the disease is more protracted, recurrent attacks, not easy to cure. So far, for patients with active Crohn's disease (CD), a pure liquid diet called Exclusive Enteral Nutrition (EEN) is the best initial treatment. Although this treatment method is effective, this pure liquid diet is very restrictive, it is difficult for patients to adhere to the diet for a long time. For this reason, the BINGO team from the University of Glasgow hopes to develop a new diet that can be proved as effective as a pure liquid diet and does not involve stopping all solid foods. In this project, we have developed an android mobile application that can collect daily diet data of patients and healthy volunteers and store it in a cloud database for researchers to access and further analyze. Before that, the collection of such information was completed in the form of paper questionnaires, which was not convenient for the subjects to report the data, nor for the researchers to sort out and access.

Acknowledgements

I would like to thank my supervisor Dr. Umer Zeeshan Ijaz for his mentorship and all of his assistance throughout this process.

Abbreviations

1. Crohn's disease — CD
2. exclusive enteral nutrition — EEN
3. the **B**acteria, **I**mmunology, **N**utrition, **G**astroenterology and **O**MICS group —
BINGO group
4. User interface — UI

Introduction and Aims

Crohn's disease (CD) is an intestinal inflammatory disease of unknown origin, which may be related to infection, heredity, humoral immunity and cellular immunity.

Crohn's disease can affect any part of the digestive tract and extraintestinal organs. It cannot be cured at present and will accompany the patient for life. The clinical effects vary from person to person, the manifestations are abdominal pain, diarrhoea, intestinal obstruction, accompanied by fever, nutritional disorders and other extraintestinal manifestations. The course of Crohn's disease is recurrent.

When people with Crohn's disease develop symptoms, the disease is considered "active". When symptoms stop, it's called "remission". When the symptoms are relieved, it is called "relapse". Therefore, the purpose of the treatment of the disease is to induce and maintain the remission of disease activity, correct malnutrition, solve

complications, so as to improve the quality of life of patients. In recent years, because its incidence rate and prevalence of all races are increasing, Crohn's disease has attracted increasing attention from more and more clinicians.

Studies have shown that diet plays an important role in the aetiology of CD, treatments around diet are also being developed. Among the current treatments, A pure liquid diet (excluding any patient's normal food or drink) known as exclusive enteral nutrition (EEN) has been proved to be effective. EEN provides a person's daily calorie intake through a fluid diet, which can be provided by oral or tube feeding.

Studies have shown that long-term enteral nutrition supplementation can prolong the remission period and reduce the recurrence rate of Crohn's disease. In this case, patients can reduce the use of steroids and immunosuppressants, thereby reducing the potentially serious adverse events associated with these drugs, which is of great help to patients, especially to improve the growth of children with Crohn's disease.

However, EEN therapy also has some limitations. Such a pure liquid diet is very restrictive, and it is difficult for patients to adhere to the diet for a long time, especially for adults. Therefore,

the **B**acteria, **I**mmunology, **N**utrition, **G**astroenterology and **OMICS** (BINGO) group from the University of Glasgow is developing a more acceptable and tolerable new effective diet therapy (which is called CD-TREAT) by using the understanding of the pathogenesis of CD and the mechanism of EEN. It reproduces EEN by excluding

certain dietary components (such as gluten, lactose, and alcohol) and others (macronutrients, vitamins, minerals, and fibre) as close as possible to common foods. They have previously proved that CD-TREAT changes gut bacteria in healthy people in a similar way to a pure liquid diet. BINGO group's next plan is to recruit 10 adults and 10 teenagers with CD and provide them with a CD-TREAT diet for up to 12 weeks.

At present, the researchers of the BINGO group collect patients' daily diet by means of paper questionnaires, which brings two problems: 1. Patients are not always convenient to fill in the questionnaire, they may fill in the questionnaire after the end of the day, which may make it more difficult to recall what food they consumed. 2. Researchers need to manually collect, sort out, count and enter these data into the computer, which is very cumbersome and time-consuming. To this end, they hope to develop an Android application to facilitate researchers to track patients' daily diet and generate statistical analysis reports based on the collected data.

The application is expected to include two parts:

1. Patients can submit their own diet through the application, and the data will be directly sent to the remote database.
2. The application can analyze the collected data and generate various statistical data and charts for researchers to view and evaluate.

Analysis

Program predesigns

According to the requirements put forward by the research group, such a mobile program structure should include:

1. User interface (UI) :

Accept user input: ordinary users can submit data in this interface.

Display output: ordinary users can simply access their own historical data.

Administrator users can view the data of all users and various statistical reports in this interface.

2. Program background:

Submission algorithm: the data input by users can be temporarily stored and sorted out, and finally submitted to the cloud database after the user finishes all input.

Request algorithm: Request the data from the database according to the user's needs.

Statistical algorithm: Properly classify and process the requested data, and generate statistical reports and charts.

3. Database:

Storage: Store data from users.

Access: Send the corresponding data according to the user's request.

Programming Language

A variety of programming languages can be used to develop Android applications.

The widely used languages are Java and Kotlin. But there are also programs developed in C++, Python and other languages. Among them, Java is the most commonly used Android development language, and a large number of classes of Java core class library are also used in Android. At the same time, the research group put forward the requirement that the program can run across platforms. Considering the good cross-platform performance of Java, and I have received systematic Java programming training, the final decision of this project is to use java language.

Program used

Android Studio

Android studio is an Android development environment based on IntelliJ IDEA.

Similar to Eclipse ADT, Android Studio provides integrated Android development tools for development and debugging. Its functions include syntax highlighting, automatic completion, debugging integration, etc., which can greatly reduce the difficulty of

developing programs.

MySQL

MySQL is one of the most popular open-source database management systems, written in C and C++. Because of its small size, fast speed, simple and easy to use, low overall cost, especially the characteristics of open source, the development of small and medium-sized websites generally chooses MySQL as the website database. MySQL provides APIs for many programming languages, including Java. In the early stage of this project, MySQL is used as a database and tested locally.

Google Apps Script

Google Apps script is an add-on component for Google sheets, docs, slides and forms. It is used to access and modify the above Google services from the outside. It can implement the automatic process of the above services and integrate with the external API to support other programs to interact with them. In the middle stage of the project development, due to the abandonment of MySQL database and the use of Google Spreadsheet as the replacement database, the program is used to interact with the database.

Postman

Postman is the client of a powerful web debugging tool. Postman provides users with web API & HTTP request debugging function. Postman can send any type of HTTP request (get, head, post, put...) with any number of parameters + headers. It is a practical debugging tool. In the process of project development, due to the use of Google Apps script as a network application interface, after the completion of code writing, it is necessary to test and debug a script to verify the correctness and usability of the interface.

Google Spreadsheet

Google Spreadsheet is a free online spreadsheet application. Users can access the spreadsheet and the data stored in it anytime and anywhere through the web browser. Based on its characteristics of free, easy to access and support script interaction, it was used as a database in the middle of project development.

Product

Program structure

This complete program consists of three components: Android application, Google App script as the interface to access the database, and Google Spreadsheet as the

database.

Android application

When an Android program is created, Android studio builds a basic structure for it.

Designers can develop applications on this structure. Take my program as an

example:

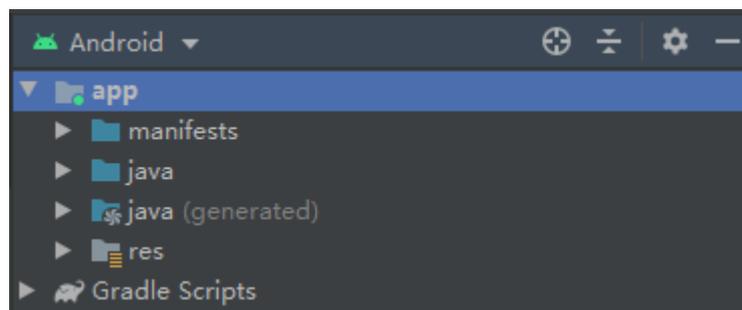


Figure 1. Android program structure

In the figure, you can see that an Android program is composed of multiple files and folders, which are used for different functions. The specific analysis is as follows:

Manifests

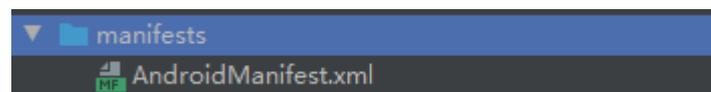


Figure 2. Manifests component

Used to store the AndroidManifest XML file. This file is the configuration file for the entire project. The four components defined in the program need to be registered in this file. In addition, permissions can be added to the program in this file. The information configured in the manifest file will be added to the Android system.

When the program runs, the system will find the configuration information in the

manifest file, and then open the corresponding components according to the configuration information.

Java

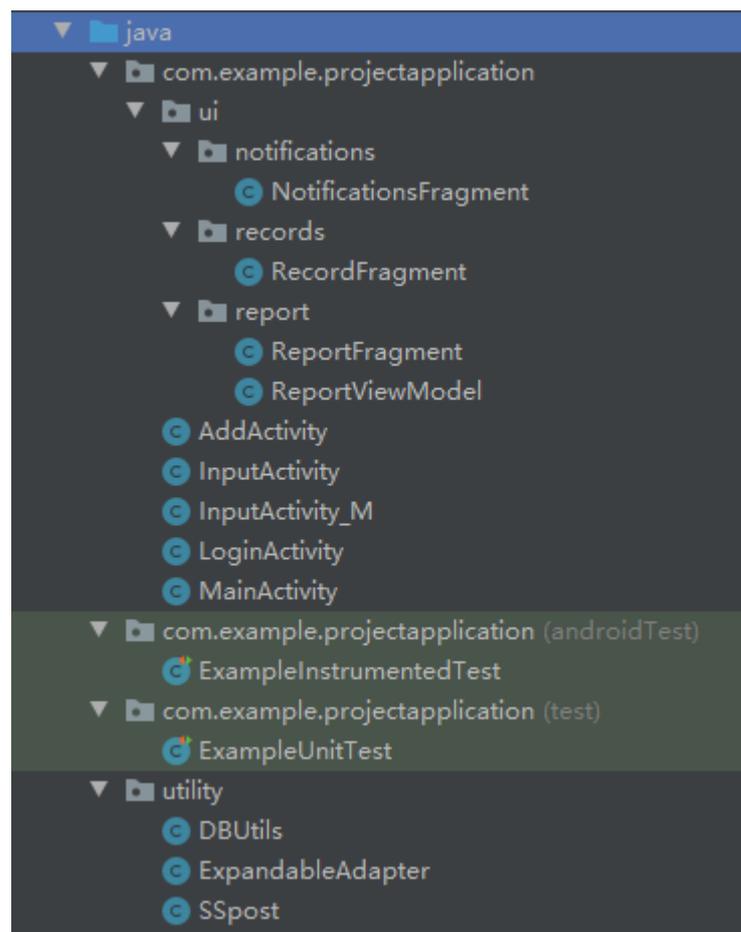


Figure 3. Java component

It is used to store all Java code. Multiple packages can be created in this folder, and different files or activities can be stored in each package.

Res

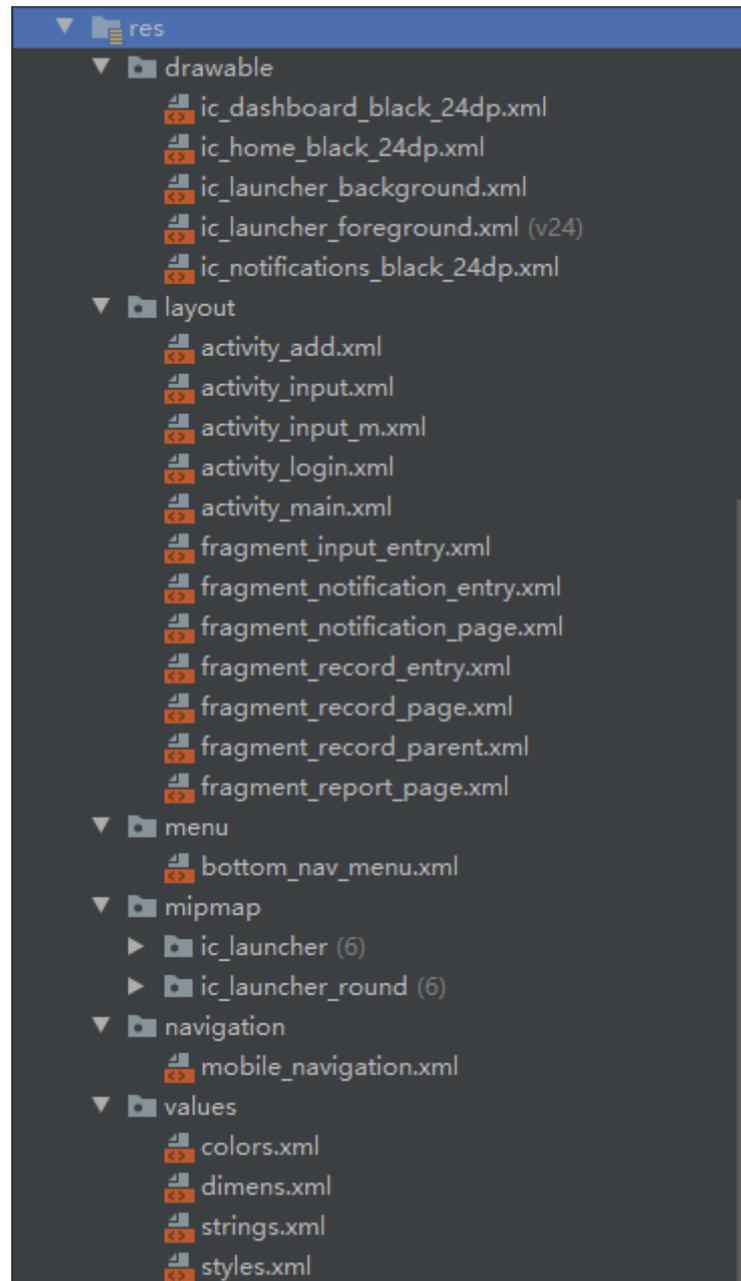


Figure 4. res component

It is used to store the resources used by Android programs, such as picture layout files, strings, etc.

The drawable directory is used to store images and XML files.

The layout directory is used to store layout files.

The mipmap directory is usually used to store application icons. The system will match icons of corresponding sizes according to the screen resolution.

The values directory is used to place defined strings.

Gradle Scripts

It is used to store the related files created by the project and need not be modified

Google App script

In order to use Google Apps script as the interface between our Android application and Google Spreadsheet, it needs to take the form of network application. To deploy Google Apps script as a network application, the script needs to have a doGet() function and a doPost() function.

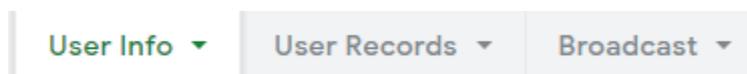
```

1  function doGet(request) {
2      return getItems();
3  }
4
5  function doPost(request){
6      return getItems();
7  }
8
9  function getItems(){
10
11     var output = ContentService.createTextOutput();
12
13     //open the Spread sheet by passing id
14     var ss= SpreadsheetApp.openById("1ZICBBNBLraWkj00ncFwBXZLLfjRfonazYjKk1I0x9g");
15     var sheet=ss.getSheetByName("User Records");
16
17     var records={};
18
19     var rows = sheet.getRange(2, 1, sheet.getLastRow()-1, sheet.getLastColumn()).getValues();
20     data = [];
21
22     for (var p = 0, l = rows.length; p < l; p++) {
23         var row = rows[p],
24             record = {};
25         record['UserID'] = row[0];
26         record['Date'] = row[1];
27         record['MealType']=row[2];
28         record['Meal']=row[3];
29         record['Quantity']=row[4];
30
31         data.push(record);
32     }
33
34     records.items = data;
35     var result=JSON.stringify(records);
36
37     output = ContentService.createTextOutput(result);
38     output.setMimeType(ContentService.MimeType.JSON);
39     return output;
40 }

```

Figure 5. An example of Google Apps script deployed as a web application

Google Spreadsheet



In this Android application, Google Spreadsheet was used as a database. This database consists of three sheets: User information sheet, which was used to store user's user ID, password and user group. The record sheet was used to store the dietary data submitted by users. The Broadcast sheet was used to send notifications to users or user groups.

Implementation

This program contains a total of five activities, of which the main activity consists of three subfragments. Each activity represents a single screen with a user interface.

When there are multiple activities in the program, these activities can wake up each other and form a complex program with multi-layer activities. The following figure shows the interaction logic between the activities.

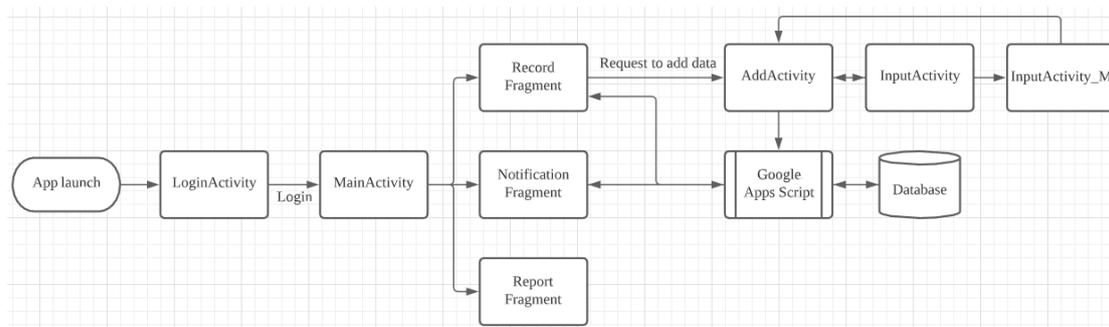


Figure 6. Activity interaction

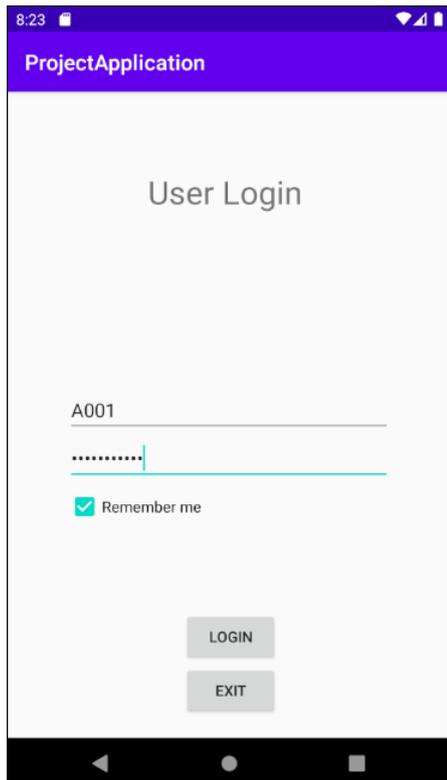
LoginActivity

When the application is started, LoginActivity is started first. Its function is very simple, which is used to confirm the user's identity. If the user cannot provide the correct user ID and password, they cannot enter the application.

At the beginning of the design, in addition to the login and logout buttons, a registration button was also considered. However, it is difficult to implement a good registration function. For example, if the user is required to register by

email or phone number, the application should have the function of sending the verification code to the email / mobile phone. Considering the actual needs of researchers, the number of participants is not very large, and researchers may need to further group registered users in the background (such as the experimental group and control group). The final decision is to manually assign user ID, password and group to each user through background database (i.e. Google Spreadsheet).

The implementation of login is that the user first needs to fill in the user ID and password. If any column is empty, a prompt will pop up when you click login to ask the user to fill in all the items. After filling in and clicking login, LoginActivity will run Google Apps script, which will send a request to the spreadsheet, retrieve all user information in the spreadsheet, generate a JSON file and return it to the application. Then, the activity parses the JSON file, extracts all the user data, and compares it with the user input. If the user's user ID and password match with it, MainActivity will be launched.



	A	B	C	D
1	User ID	Password	Group	
2	A1	123	group A	
3	A2	234	group A	
4	B1	456	group B	
5	Admin	567	Admin	
6				

Figure 7. Login interface and background database

MainActivity

MainActivity is the core part of the application. It consists of three subfragments:

1. Records fragment :

Records fragment is used to display and add records entries. It mainly contains a ListView component (used to display all the history entries of the current user in the database) and a floating button. Users can click the "+" floating button in the lower right corner to jump to AddActivity.

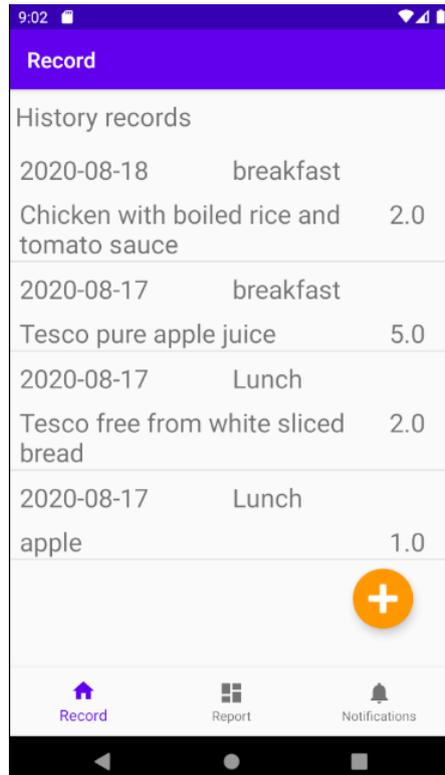
The way to retrieve the user history entry from the database is similar to that when logging in. The activity uses Google Apps script to retrieve the data of all

users in the form of JSON files from the spreadsheet. These data are parsed locally and filtered according to the current user. Only the current user's data is displayed on ListView. What is worth mentioning is the way ListView works. ListView is widely used as a component of Android programs to display lists. Three components are needed to make ListView work normally.

1.1 Data source. The data source used in this program is an ArrayList composed of HashMap. Each HashMap is a complete data entry.

1.2. ListView adapter. There are many kinds of adapters that can be used in ListView, such as SimpleAdapter, BaseAdapter, etc. Among them, SimpleAdapter is simple and easy to use, but it is limited and inflexible. BaseAdapter is powerful and can be customized according to the development requirements. However, developers need to rewrite the methods in this class by themselves, which is complex. In this application, the original plan is to use the BaseAdapter, and part of the code has been rewritten. However, due to the lack of time and complexity of the code, the plan was abandoned and SimpleAdapter was used instead. In the application code, you can see the customized BaseAdapter class, which is ExpandableAdapter class in Figure 3.

1.3. ListView component and its subentry fragment, which is responsible for the appearance of the list and the internal items.



	A	B	C	D	E	F	G
1	UserID	Date	MealType	Meal	Quantity	Unit	
2	A2	2020-08-19	breakfast	Free from White Bread	2	1 slice 37g	
3	A2	2020-08-19	breakfast	7-up	2	100ml	
4	A1	2020-08-18	breakfast	Chicken with boiled rice and tomato	2	Small size	
5	A1	2020-08-17	breakfast	Tesco pure apple juice	5	100ml	
6	A1	2020-08-17	Lunch	Tesco free from white sliced bread	2	1 slice 33g	
7	A1	2020-08-17	Lunch	apple	1	One	
8	A2	2020-08-17	breakfast	apple	1	One	
9							

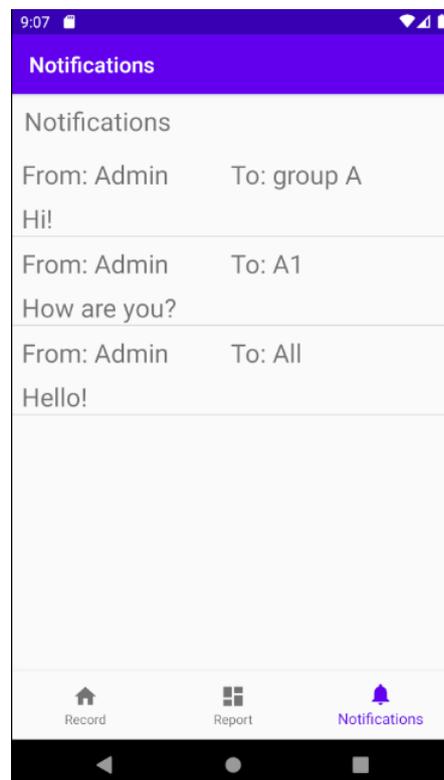
Figure 8. Record fragment and background database

2. Report fragment:

This component is planned to be able to process, generate and display user statistics. If the user is an ordinary user, only this user's statistical data will be displayed. If the current user is an administrator, the administrator can further filter and view the data and generate statistical reports of all users. Due to lack of time, the function did not complete.

3. Notification fragment:

This function is not one of the requirements of the project. However, being able to send information to users may prove to be a useful function. Since similar functions have been implemented in record fragment, most of the previous code can be reused here.



<i>fx</i>	A	B	C	D
1	From	To	Message	
2	Admin	group A	Hi!	
3	Admin	A1	How are you?	
4	Admin	All	Hello!	
5				

Figure 9. Notification fragment and background database

AddActivity

AddActivity is mainly composed of five parts:

Date Picker: Used to let the user select the date. If not selected, the date of the day will be filled in automatically by default. At the beginning of the project, a text input box was used as the date input, and the user needs to input the date according to the specified format. However, this method is obviously very inconvenient and error-prone, and it is improved to use a Date Picker.

RadioButton Group: Used to let users choose the time of the meal (breakfast, lunch, dinner). If no selection is made, breakfast will be selected automatically by default.

ListView: It is used to display the food that has been added. There is an X button at the end of the entry to delete the current item. As mentioned earlier, the operation of ListView depends on the adapter. The initial project used the same SimpleAdapter as in the previous windows. However, during the test, it was found that the user may accidentally input the wrong data. As the SimpleAdapter did not support deleting the clicked entry in ListView, the user had to restart the entire application to reset the list in order to delete the wrong item. The final solution is to use the BaseAdapter and rewrite the getView method. The ViewHolder class is used in the getView method, which can support button

listening.

Two buttons: Add and Send. Click the Add button will open the InputActivity for inputting food data and click the send button will send all the entries in the current list to the database and jump back to MainActivity.

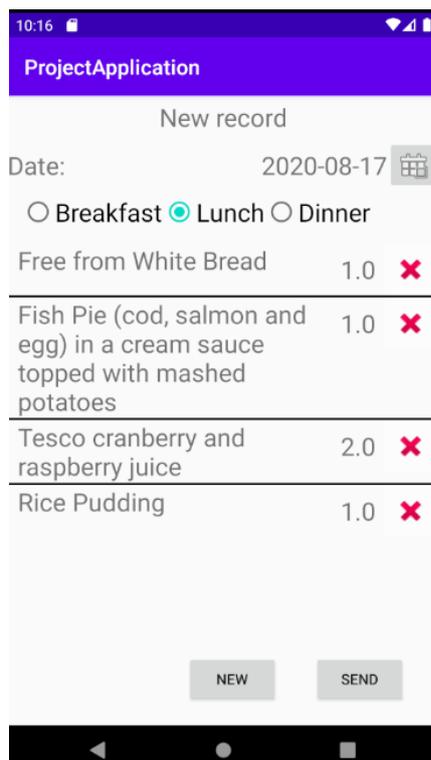


Figure 10. AddActivity

What is worth explaining here is the start method of Android activity. In Android, every application is regarded as a task. When an activity is opened in an application, the default start method is to create a new activity instance and store it in the corresponding task of the application. Since an application is often composed of many activities, the activities will be stacked layer by layer, which is called activity stack. Therefore, although the application seems to be a coherent

whole, it is actually composed of individual newly created activities. This brings a problem. After data input is completed in the InputActivity, clicking the finish button will create a new AddActivity instance. The new instance itself is not associated with the previous AddActivity, so it will not inherit the data of that activity.

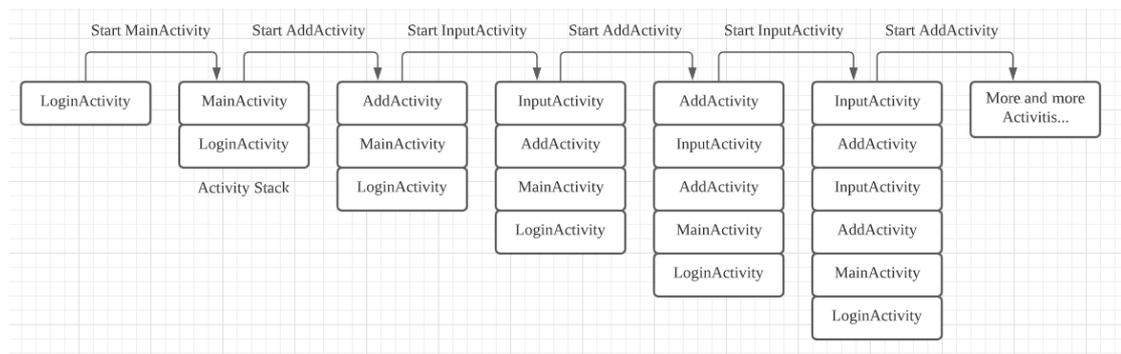


Figure 11. Standard start mode

The solution is to start AddActivity in singleTask mode instead of the default standard mode. In singleTask mode, only one AddActivity instance will be created, and when other activities try to create a new AddActivity instance, all activities above the existing AddActivity in the stack will be automatically destroyed to make AddActivity the current activity. This not only solves the data inheritance problem mentioned above (because the same activity is used, the data will still be retained), but also solves the looping nesting problem that may occur with multiple activities interaction.

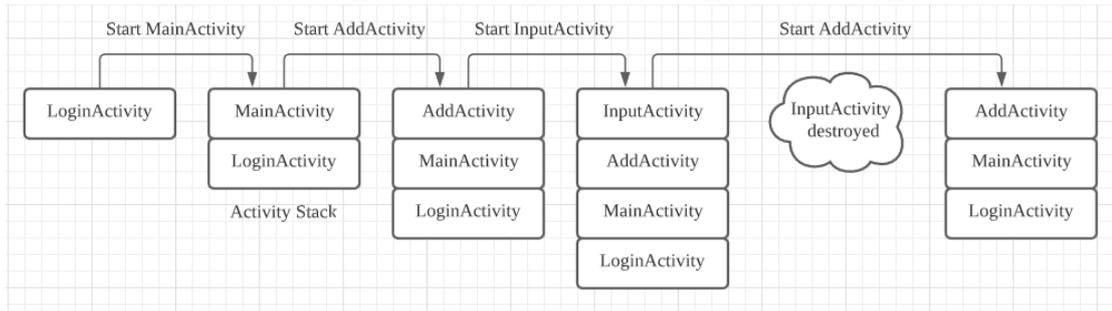


Figure 12. SingleTask start mode (AddActivity)

However, using singleTask start mode brings another problem. Android programs transfer values to each other through Intent, and the values passed by Intent are received at onCreate() in the activity life cycle. Due to the use of singleTask, only one AddActivity will be created, so only when it is created for the first time can the value be correctly received through the onCreate() method. In order to solve this problem, the onNewIntent() method was rewritten, so that the application can get the value correctly when it receives the new intention.

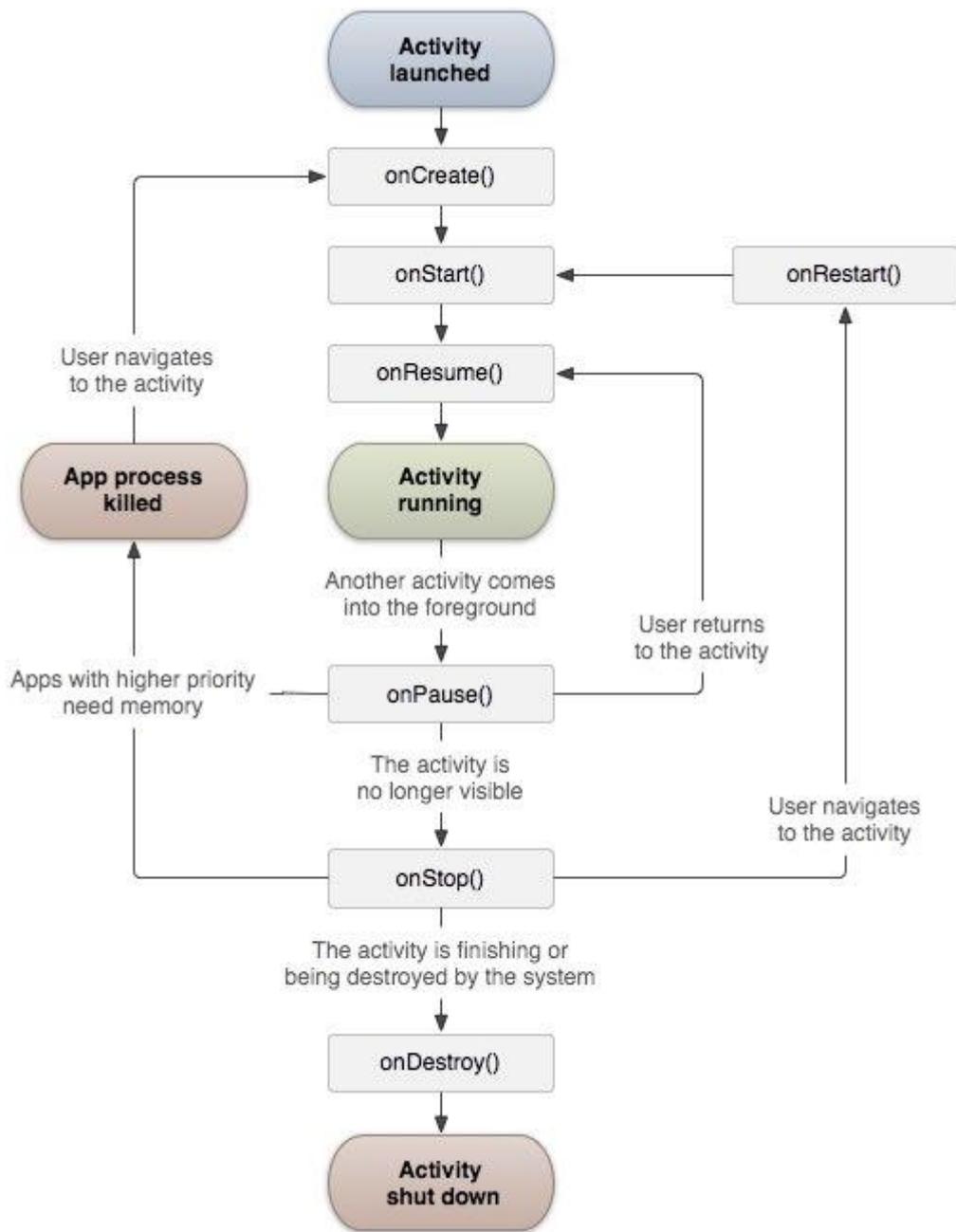


Figure 13. The life cycle of activity (picture from Android website)

InputActivity and InputActivity_M

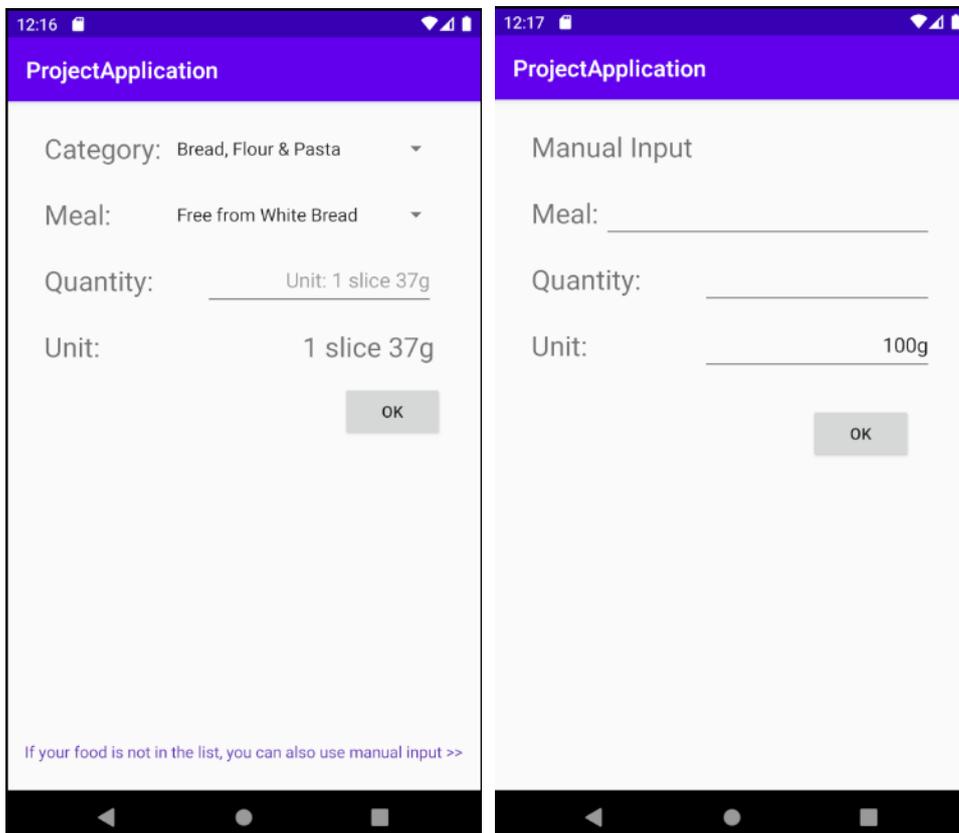


Figure 14. InputActivity and InputActivity_M

InputActivity consists of two drop-down menus, a text input box, a confirm button and a clickable text at the bottom. The two drop-down menus are marked with "category" and "meal". Svolos, a member of the BINGO group, provided an excel sheet showing the food and its corresponding category they wanted to be displayed in this menu. The user needs to select a category first. According to the different categories selected, the "meal" drop-down menu at the bottom will provide different food options. After finding the corresponding food, the user needs to fill in the consumption manually. Finally, if users consume food that

doesn't appear in the list, they can jump to InputActivity_M by clicking the text at the bottom of the screen, where M stands for manual here.

Meal names	Meal	Size	kcal	Fat	SFA	Prot	CHO	Sugrs	DF
	Bread, Flour & Pasta								
	Free from White Bread	1 slice 37g	92	1.6	0.1	1.7	15.8	0.4	3.7
	Schar Wholesome white loaf	1 slice 27g	64	0.9	0.1	0.9	12	0.9	2
	Tesco free from white sliced bread	1 slice 33g	92	1.7	0.1	0.7	17.2	0.4	2.3
	Genius white loaf	1 slice 36g	91	1.9	0.1	0.6	18	1.4	3.1
	Schar Gluten Free Pizza Base	150g = 1 pizza base	440	7.2	0.9	3.5	8.1	8.1	6.9
	Doves Farm Gluten Free White Bread Flour	100g	351	0.9	0.2	5.3	79.8	0.3	1.4
	Free From Gluten Fusilli pasta	100g	172	0.6	0.2	2.8	38.6	0.2	0.9
	Cereal & Dairy								
	Gofree cornflakes	100g	382	0.9	0.2	6.2	35	22	4
	Gofree Rice pops	100g	385	1.2	0.4	7.5	85	9.3	1.5
	Free from Honey Hoops	100g	381	1	0.3	6	86	19	2
	GoFree Honeyflakes Dry	100g	382	0.9	0.2	6.2	85	22	4
	Lactofree long life semi skimmed milk	100ml	39	1.5	1	3.5	2.7	2.7	0
	Lactofree long life whole milk	100ml	56	3.6	2.3	3.3	2.6	2.6	0
	Lactofree Cheddar	100g	416	35	22	25	0.5	0.5	0
	Lactofree Spreadable butter	100g	676	75	34	0.5	0.5	0.5	0
	Lactofree Soft cheese	100g	199	17	11	8.7	3	3	0
	Fruit & Fruit juice & 7up								
	Tesco pure apple juice	100ml	46	0.5	0.1	0.5	9.5	9.5	0.5
	Tesco Orange juice smooth	100ml	47	0	0.1	0.5	10.5	10.5	0
	Tesco pineapple juice	100ml	53	0	0.1	0.2	12.3	12.3	0.2
	Tesco Apple and mango juice	100ml	50	0.1	0	0.1	11.7	11.7	0.1
	Tesco apple and raspberry juice	100ml	11	0	0	0	2.5	2.5	0
	Tesco cranberry and raspberry juice	100ml	19	0	0	0	4.4	4.1	0
	Tesco cranberry juice	100ml	20	0	0	0	4.5	3.9	0
	Fruit snack (apple peeled)	One size 100g	45	0.1	0	0.4	11.2	11.8	1.6
	Fruit snack (pear peeled)	One size 100g	41	0.1	0	0.3	10.4	10.4	1.7
	7-up	100ml	29	0	0	0	7	7	0
	Pizza & Sandwiches								
S1	Sandwich with chicken, lettuce, cucumber, tomato	One size	351	11.6	4.1	25.4	31.4	3.4	4.9

Figure 14. A screenshot of the excel sheet provided by BINGO group

Evaluation

At present, the application has been tested on the virtual machine for several times, and the data used in the test are real patient diet data collected by BINGO group in previous experiments. The data reporting and receiving functions of the application work well. All data are successfully sent from the virtual machine to the database and can be correctly received by the application again. Due to the development progress, the application has not been put into actual use for the time being. It is expected that the application will have good performance when the number of users is small.

The database access is very convenient, researchers can access and copy the data at any time, eliminating the time consumed in the previous process of "filling in the questionnaire —— collecting the questionnaire —— entering the questionnaire". However, due to abandoning MySQL database and using Google Spreadsheet as the database, and being unfamiliar with Google Apps script, the function of automatically analyzing data and generating reports according to different users was not implemented.

It is very novel to use Google Apps script to interact with Google Spreadsheet, and then use Google Spreadsheet as a database. In the middle stage of the project, it is found that there is no available server to run SQL database, so this method is adopted to implement the database function. However, on the other hand, the documents and learning materials about Google Apps Script are very scarce, and it takes a lot of time to implement the interaction between Google script and spreadsheet. The learning materials and literature available are limited to a few video tutorials and the script instructions provided by Google itself. However, the former is usually difficult to learn systematic programming knowledge, and the content of the tutorial often seems to be telling the audience how to copy and paste the code, while the Google script instructions only enumerate the method names and the purposes of each method, so it is impossible to have a systematic understanding of the usage too.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1				Prescribed	2020/3/7	2020/3/8	2020/3/9	2020/3/10	2020/3/11	2020/3/12	2020/3/13	2020/3/14	2020/3/15	2020/3/16	2020/3/17	2020/3/18	2020/3/19	2020/3/20	2020/3/21	2020/3/22
2		ETH paraiban	ETH paraiban	Prescribed	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15	Day 16
17	Lactofree semi-skimmed			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	Lactofree whole milk (mls)			547.64	400	520	400	900	614.4	721.5	660	520	864.4	360	1000	754.4	400	1142.9	870	
19	Gofree cornflakes			15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	90
20	Gofree Rice pops			15	0	0	0	0	0	0	0	45	45	45	90	45	45	0	0	0
21	Free From Honey Hoops			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	Gofree Honeyflakes Dry			15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	90
23	Lactofree Cheddar			33.87666667	75	60	30	70	117.9	44.3	40	45	87.9	45	40	117.9	0	143.6	75	
24	Lactofree Spreadable butter			1.857142857	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	Lactofree Soft cheese			39.375	45	30	60	-15	45	30	45	45	30	30	-15	45	30	30	30	105
26	Free From White Bread			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	Tesco free from white rolls 4 pack			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	Schar Wholesome white loaf			79.1	84	84	84	0	84	168	0	84	84	84	0	84	84	84	84	84
29	Tesco free from white sliced bread			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	Genius white loaf			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	Schar Gluten Free Pizza Base			9.375	0	0	0	75	0	0	75	0	0	75	0	0	75	0	75	0
32	Doves Farm Gluten Free White Bread Flour			6.473333333	25	0	0	0	15	5	0	25	15	0	0	15	25	10	0	
33	Free From Gluten Fuusti pasta			54.66666667	0	0	160	160	240	100	160	0	240	0	160	240	0	0	160	
34	Fruit snack (apple peeled)			50	100	0	100	0	100	0	0	0	0	100	0	0	0	0	0	0
35	Fruit snack (pear peeled)			50	0	100	0	0	0	100	100	0	100	0	0	0	0	0	0	100
36	Tesco pure apple juice			150	0	0	0	200	200	400	200	200	200	200	200	0	0	0	0	0
37	Tesco Orange juice smooth			150	400	0	400	0	200	0	200	200	200	0	0	400	200	0	200	
38	Tesco pinaapple juice			150	200	0	0	400	0	200	200	200	200	0	400	200	200	0	400	
39	Tesco Apple and mango juice			150	0	600	0	0	200	0	0	0	0	400	200	0	0	0	0	
40	Tesco apple and raspberry juice			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
41	Tesco cranberry and raspberry juice			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
42	Tesco cranberry juice			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
43	7-up			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
44	Chicken Breast			115.672619	240	0	0	27	147	147	18	0	120	0	0	120	0	90	0	
45	Lettuce			3.125	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
46	Cucumber			6.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
47	*****			6.125	0	100	100	176	0	100	176	0	0	100	176	0	0	176	100	

Figure 15. A screenshot of the sample data provided by BINGO group

Conclusion and Discussion

At present, the application has implemented the following functions: user login, submitting diet data to database, viewing submitted historical data. The function that has not been implemented is to analyze data and generate statistical reports.

Due to the limited understanding of Google Apps Script, a lot of work that should be done on the database side has to be transferred to the client-side. For example, when logging in, the normal login method is that the client sends the user information to the database, and the database compares the information. If the same entry is found, it returns true. In this application, the database sends all the user information to the client, and then the client compares it. This is neither fast nor safe. The same implementation method is also used when the client receives the

database entries. The client pulls all the entries from the database, and then filters the entries according to the current user and presents them to the users. If the sample size is large or there are many database entries, this communication method is undoubtedly very inefficient. Therefore, it is recommended to copy the entries in the database to other places regularly when using the spreadsheet as a database, so as to prevent the lag caused by too much data being transferred.

To further optimize the current application, first is to learn Google Apps Script deeply, use Google script to classify and filter the database, and implement the work that should be completed by the server, to optimize the transmission efficiency. At the same time, although according to the current sample size, the spreadsheet is fully competent for the database responsibility, the use of a real MySQL database is still a good choice. As the most commonly used database in the world, it is easy to use, powerful and has rich references and tutorials, which can prove useful when more complex functions need to be implemented in the future Basics.

According to BINGO group members, their current way of processing data is to use Excel sheets to process data. Although Excel is powerful, a large part of the process needs to be done manually. To implement the function of automatic data analysis and statistical report generation, I think a powerful database with an automatic pipeline is the solution. At the same time, I think it's better to leave all the data processing to the server (if possible), while the mobile application only needs to

receive the data and present it.

References

1. Svolos V, Hansen R, Nichols B, et al. Treatment of Active Crohn's Disease With an Ordinary Food-based Diet That Replicates Exclusive Enteral Nutrition. *Gastroenterology*. 2019;156(5):1354-1367.e6.
doi:10.1053/j.gastro.2018.12.002
2. Quince C, Ijaz UZ, Loman N, et al. Extensive Modulation of the Fecal Metagenome in Children With Crohn's Disease During Exclusive Enteral Nutrition. *Am J Gastroenterol*. 2015;110(12):1718-1730.
doi:10.1038/ajg.2015.357
3. Akobeng AK, Zhang D, Gordon M, MacDonald JK. Enteral nutrition for maintenance of remission in Crohn's disease. *Cochrane Database Syst Rev*. 2018;8(8):CD005984. Published 2018 Aug 11.
doi:10.1002/14651858.CD005984.pub3
4. Veauthier B, Hornecker JR. Crohn's Disease: Diagnosis and Management. *Am Fam Physician*. 2018;98(11):661-669.
5. Baumgart DC, Sandborn WJ. Crohn's disease [published correction appears in *Lancet*. 2013 Jan 19;381(9862):204]. *Lancet*. 2012;380(9853):1590-1605. doi:10.1016/S0140-6736(12)60026-9