Developers Guide

By: Joshua Wiess – 2833991

1. Installation

To retrieve the source code, download it from Joshua-Wiess/Software-Engineering-Project: Meal Generator by Joshua Wiess 2833991.

To use the app, first start by downloading Android Studio which can be navigated to using the following link: Download Android Studio & App Tools - Android Developers. Next, go through the installation setup without changing any of the preset information. Once it is installed, open Android Studio. It will then guide you through the Setup Wizard in which you will keep all the preset settings selected once again. During the license agreement section, you will have to accept both the android-sdk-license and the intel-android-extra-license. Once the application is open, select to open a project, then navigate to where the "TermProject" folder is located and open it. If it pops up with an error associated with the SDK path being changed, just click "Ok" and wait for the gradle build to be loaded which can take a significant amount of time. There is a bar in the bottom right corner of the screen showing that this is in progress. You will know it is done when the green play button is accessible on the top of the screen. Clicking this will start the app and using the navigator on the left side of the screen provides access to all the necessary code. Once the emulator boots up, the app will automatically open and testing can begin.

2. Database

The database should not need modification as it already handles user login, authentication, tracks likes, and tracks dislikes. If it does need modification, then there are three tables with one corresponding to user login information, another being for liked foods, and the last being for disliked foods.

Starting with the "Userbase" table, this table contains an ID for the user that connects it to the liked and disliked tables, the user's login name, and the user's password. The "LikedFoods" table connects to the userbase table through the user ID and it stores the meal type of a food and the name of the food that the user liked. The

“Disliked Foods” table serves a similar purpose and stores the same information but is in relation to foods that the user disliked. The methods for managing the database have straightforward names indicating their purpose.

3. Login Page (Main Activity)

The login page consists of two main methods with one being for managing what happens when login is pressed and the other managing what happens when the registration button is pressed. The login button gets the strings from the username and password boxes, then passes those to the userValidation method that is part of the database. This checks what was input against what is currently in the UserBase table and looks for a match. If there is a match, then the ThirdActivity is started which is the food suggestion activity. In the case of there not being a match, then a toast shows on the screen to indicate an unsuccessful login. When the registration button is pressed, the second activity is initialized, and the user is taken to the registration page. The onActivityResult method waits for the user to return from the registration page, and then takes the information that is returned (username and password) and uses the insertNewUser method that is part of the database. This then creates an account for the user that will be stored on the device.

4. Registration (Second Activity)

The registration activity only contains one additional method with it being in relation to the “Submit” button. This method waits for the user to press the button then stores the username and password into two variables which is then passed back to the login activity.

5. Food Suggestion (Third Activity)

The third activity is where the bulk of the user’s time will be spent. The main variables that are in this activity are the breakfastFoods, lunchFoods, and dinnerFoods arraylist. These contain the names of the foods that will be suggested to the user. If a food is to be added to the program, then it needs to be added to those arraylist. Another important variable is the imageFiles map which links the pictures of the foods to the

names that are stored in the arraylist. To add another food, the picture of the food needs to be added to the map. The last variable that needs addressing is the nutritionDictionary which stores the nutritional information in an array of strings with a key that corresponds to the name of the food from the arrayList. This variable is the last one that would need to be modified when foods are to be added.

The activity starts with loops checking that there are no disliked foods being shown to the user and it does this by removing them for the arraylist so they are no longer taken into consideration by the program. The next set of code is in relation to what happens when the breakfast, lunch, and dinner buttons are pressed. They essentially check the arrayList and put foods in the ImageViews so that they can be shown to the user. All images have an ID associated with them and accessing the image requires this ID which can be obtained from getting the value from the map. There is a textbox hidden in the activity that stores the meal type so the app can keep track of what is being shown on the screen.

The next chunk of important code is the onClickListeners for the top, middle, and bottom images. These on click listeners allow the user to tap on the food being shown on the screen to obtain more information about the food. These also control the like and dislike features of the app which interact with the database through the insertLikedFood and insertDislikedFood methods respectively.

Lastly, there is an onClickListener which is connected to the meal generation button that starts the meal suggestion activity and will send the user's ID so that the app can know what information to pull from the like and dislike tables.

6. Meal Suggestion (Activity Four)

The fourth activity has five main variables which are the breakfastFoodsArray, lunchFoodsArray, dinnerFoodsArray, imageFiles, and the cursor (c). The activity starts by pointing a cursor at the start of the results returned by searching the LikedFoods table for the userID. Then it uses a while loop to go through ten of the liked foods for breakfast, lunch, and dinner that the user liked and populates the arrays with this information. The activity then goes through that array loading the images associated with the liked foods into the left and right ImageViews for each meal. When an image is inserted into an ImageView, an onClickListener is attached to the image so that the user can tap on the food to get the name of the food.

7. Important Notes

All of the activities pass the userID to one another when a new activity is started. The userID allows for the storing and retrieval of information associated with the user so that they can use the app in the future. The userID is a unique key that has no duplicates in the table.

Inserting other images into the app to be used involves navigating to the drawable folder that is a subdirectory inside the res folder. The simplest way to add an image is right clicking on the drawable folder, opening it in explorer, and then dragging and dropping the image files into the folder.

The XML files should not need any modifications and are all labeled with what it is exactly that they do. Important attention should be paid to the third activity as the foods are shown in a scrollable list. This allows for the easy addition of more ImageViews to show the user more foods. Adding these ImageViews can be done programmatically so as to add the number of ImageViews that are desired.

Lastly, removal of foods that are liked or disliked from the food suggestion activity occurs when the app is closed and reopened. Consequently, the user will continue to see foods that they made a decision on until they fully close and reopen the app. Fully close in this context means that the app is not still open in the background.