Mobile Device Programming Assignment 2

Lindie Chenou
Wednesday, October 7th, 2020

With the previous assignment, chapter 1-3 were covered by doing more practice. With this assignment chapter 4-6 will be practiced. This will steam from the app that has been built during the class lecture ContactList

The first problem asks to create an app called MealRater. This app will accept a user entry of a restaurant and a dish from that restaurant, then the user will be prompted to rate the dish that they just enter, and the rating will be display on the main screen below the dish name and restaurant name. To accomplish this task there needs to be four TextViews, two Edit Texts, and one button on the main screen. The rating window will have to be separate from the main activity. Similar to the calendar view from the assignment. There are two ways to prompt another screen or window to open on Android Studio. One way is by using the function intent, where the full screen will display the rating bar window. Another way which was required for the problem is dialog. Dialog fragment affectively brings the rating screen up while not covering the main screen where it was called. Operate like a pop up on the screen. The project needs an activity_main.xml, activity_rater.xml, Rater.java, and MainActivity.java. The mainActivity file will call the Rater.java. from the Rater.java, it will get the rating from the user input and set it to the display TextView for the result of the rating on the activity_main. xml. The result of problem 1 is listed below. `((MainActivity)getActivity()).transresult.setText(rating);`
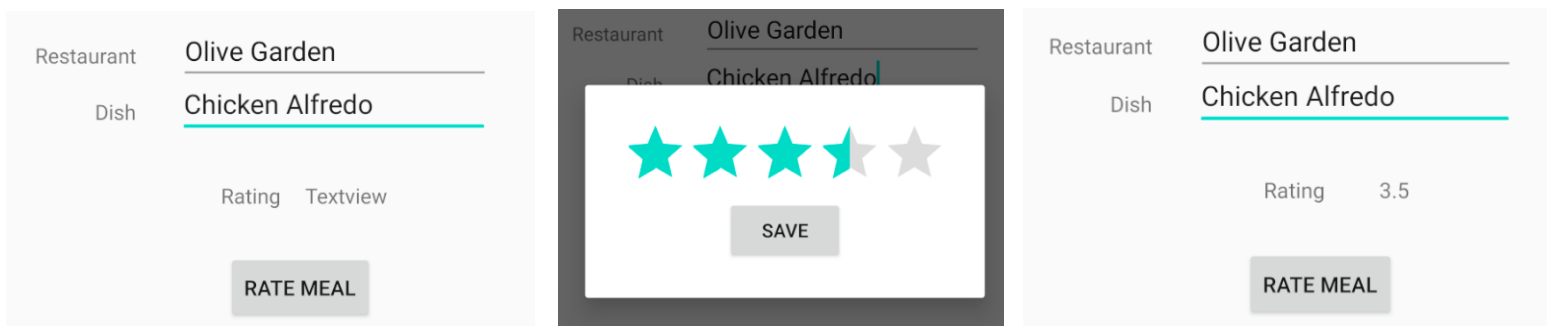
The second part of the project was building an app called RestaurantRater. RestaurantRater needs to allow the user the save the name and the address of a restaurant. The user also needs to

be able to save several dishes associated with that restaurant all saved in a database table. The user needs to be able to enter the dish name, dish type (Entrée, Dessert, etc.), and the rating of the dish on a scale of 1-5. There are one database files in which has two tables in it. On for the Restaurant attributes and a Dish table with its attributes. The column of the restaurant table is RestaurantId which is a primary key, the restaurant name, the restaurant StreetAddress, the restaurant city, the restaurant state, and the restaurant zipcode. The column of the Dish table is the DishId, the dish name, the dish type, the dish rating, and the RestaurantId which is passed from the restaurant table and it's a foreign key. There is three activity for this project, one MainActivity which the restaurant can be saved. The RestaurantActivity which will list out all the restaurant save on the database table. And lastly, The DishActivity which allows the user to add a dish corresponding to a restaurant.

The APPDBHelper file is where the database table is created for the Restaurant and Dish. Also, the Dish and Restaurant file which is there to do the setter and the getter of the components of each attribute. the RestaurantDataSource and the DishDataSource are the files that would set, edit, and delete the values from the dish or restaurant table. And lastly, the RestauranAdapter and DishAdapter are the files that would set the values on a ListView. They are the physical representation of what it's in the database table.

On the RestaurantDataSource three functions would be used to make the restaurant database table functional. First, the createRestaurant function which will take the input of the String name, String address, String city, String state, and String zipcode. Those values will be entered into the table. The next function is getAllRestaurant. This will get all the rows in the restaurant table to be displayed. And lastly the getRestaurantById; it's set up so that the dish table can get the corresponding restaurant table that it's adding to.

The second part of the RestaurantRater app is to allow the user to edit the dish to enter from a restaurant and also allow the user to delete any dish they choose. Also having the list of all corresponding dish below the restaurant address. DishDataSource file is where the dish can be created, edit, and delete from the database table. Like the Restaurant, to add a dish to the table use the createDish function, which accepts a String name, a String dishType, a String dishrating, and a restaurantId which has to be pass from when the restaurant it's clicked from the list. To update the dish from the list the program uses the updateDish function from the DishDataSource files. The deleteDish function is to delete the dish from the Dish table. Lastly to display all of the dishes attached to a specific restaurantId is by using the getDishofRestaurant.

When both a restaurant and dish or click the open activity will show all the attributes that were added to for that restaurant or dish. When clicking on a restaurant from the ListofRestauranActivity it should open the MainActivity with the name, StreetAddress, State, City, and Zipcode already being on the EditText. Same thing when the DishActivity is open from a click on the Dish, with having the dish name, dish type, and the dish rating being on the EditText.
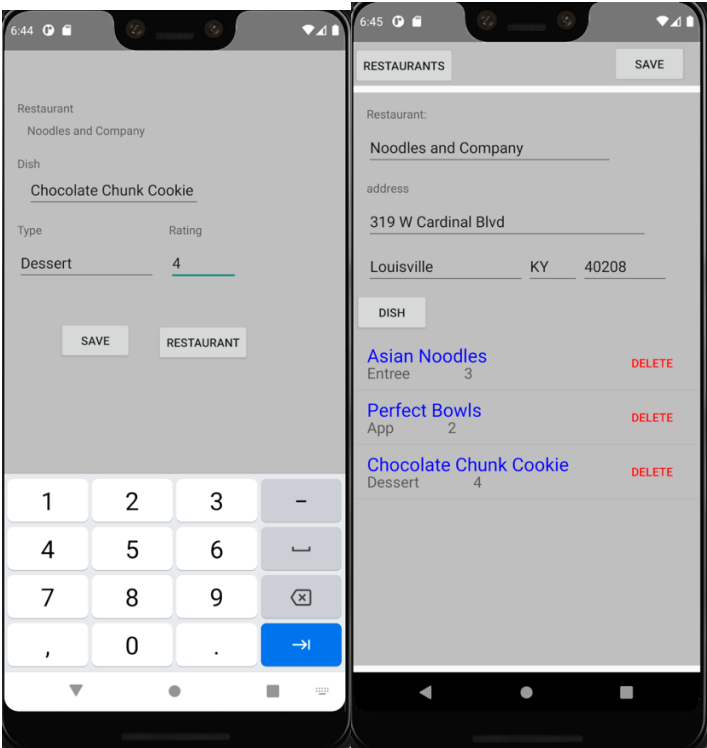
These first set of ScreenShot are the display on how when the app has just been downloaded.
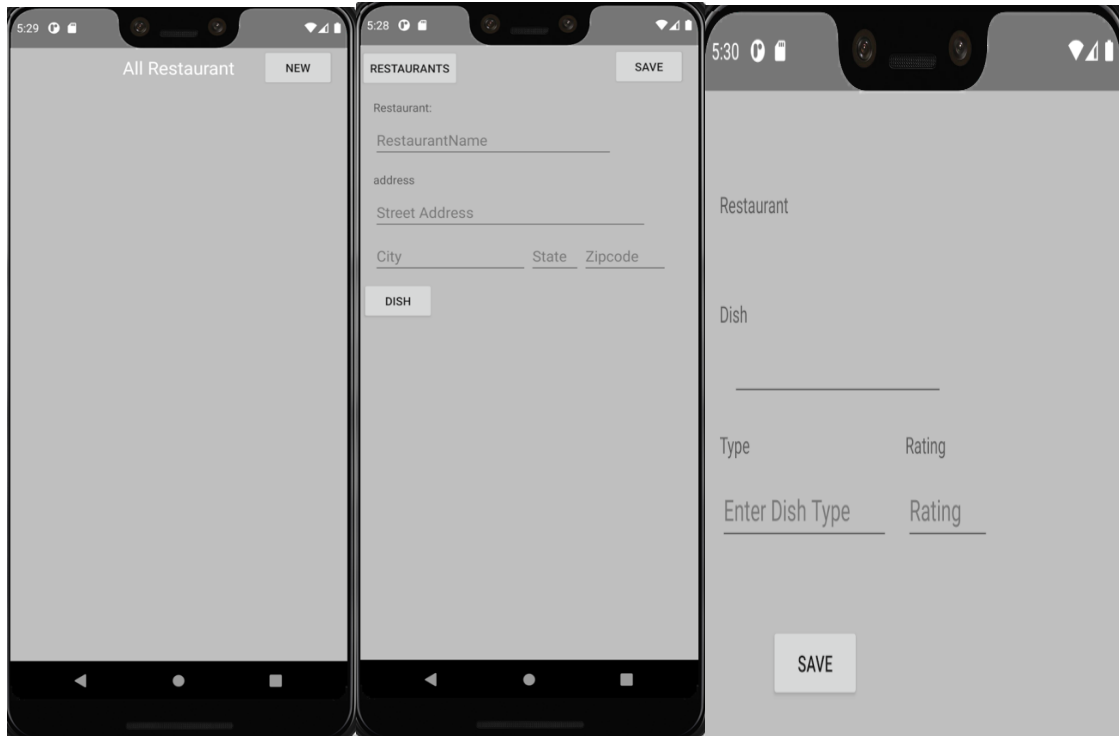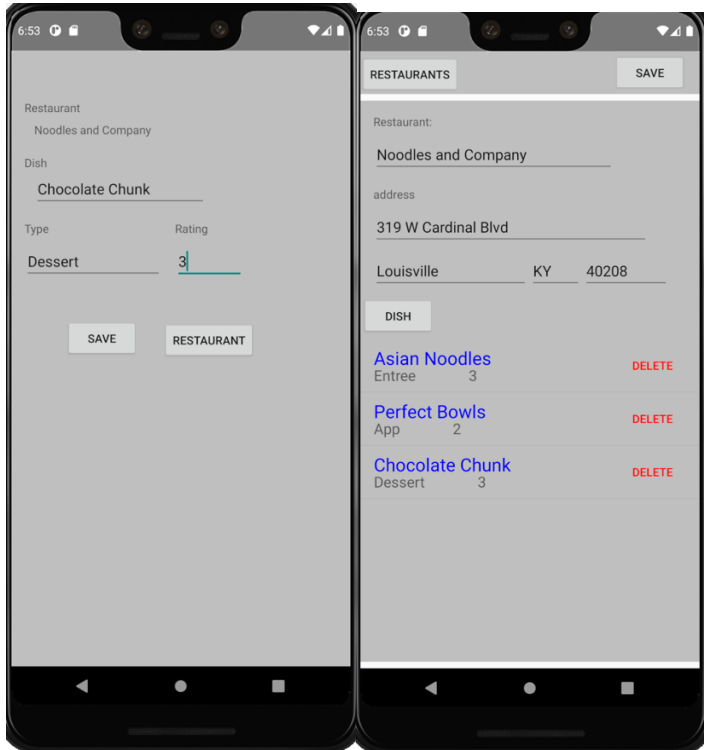

The insertion of a Restaurant and it's attributes. Also the view of the Restaurant when the

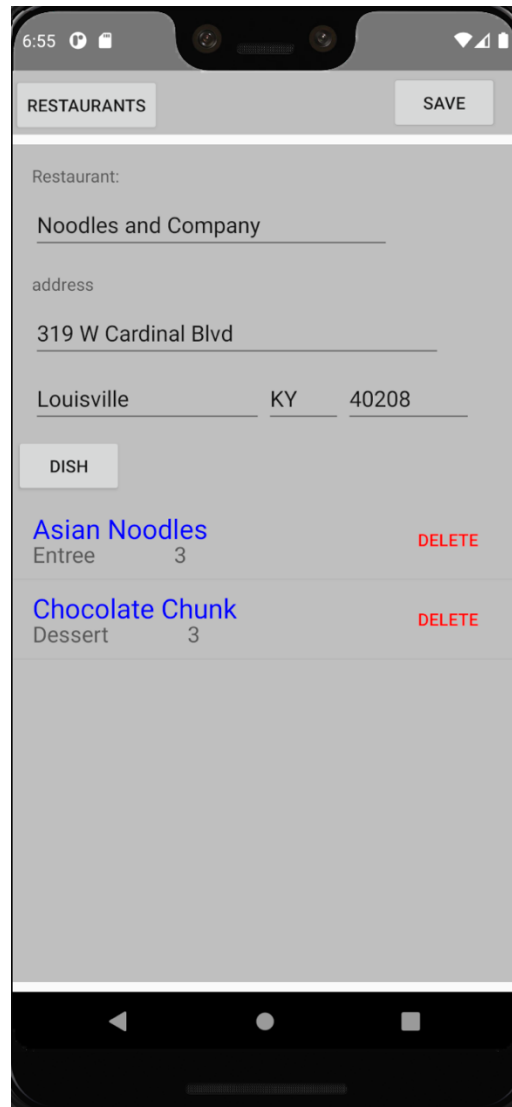restaurant has been added to the Restaurant Table
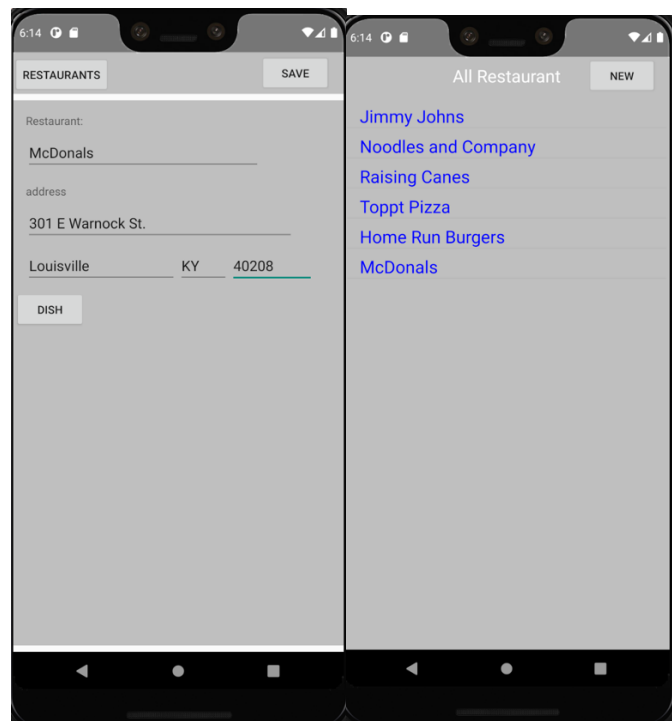
Adding dish to a specific restaurant and showing the Dishes from that restaurant.

Editing the Chocolate Chunk Cookie entry from the dish table.

And Lastly deleting a dish from a restaurant and the dish table.

In conclusion, there were major grounds made to understand building a well full functioning app on Android Studio. Starting with the first problem using the dialog function. Which is to implement a pop up on the main screen. That is helpful for a calendar view, rating bar, or even to send out alarms or alerts on the input from the user. Part 2 & 3 are mostly involved in having multiple database table functions. Also understanding how to connect between multiple classes of the project and sending information between those classes. In all, with the RestaurantRater

application that is the first fully functioning application that could be very useful to a user that

happens to dine out a lot that has been completed in the class.