**CONTENTS**

[**Summary of Development**](#_w1x7rfry9ool) **2**

[**Challenges Encountered**](#_7txv1m5qdr4) **3**

[**Test Cases Performed**](#_av8hq4j7o798) **3**

### 

### **Summary of Development**

The main task of the project is to create a login tab in the fun fragment of the application. The login tab contains elements like First Name, Last Name, Email, Password, Date of Birth, Gender, Agree to terms checkbox and a submit button. The app navigates to the confirmation page once the submit button is clicked displaying a text paragraph along with First Name and Last Name. Below the paragraph is a Recycler View which displays a scrollable list of food items.

Once the action is defined from one fragment to another in the navigation, java is generated for that action which defines the directions and the arguments needed for that action. Since we need to display two of the text inputs from one fragment to the next, two arguments should also be defined in the navigation. After the java is generated, the functions in that generated code can be used for navigation. The generated functions can be called during the click of the Submit Button using onClickListener function which takes us to the confirmation page displaying a text paragraph along with the First Name and Last Name.

As the requirement is to display a list of food items with Image, first a layout with an Image View and a TextView should be created. Then an Adapter should be created which contains a View Holder which retrieves the image and text view of the layout that was created. A Model Class should also be created which will have an integer and string which corresponds to an image and text view. Then the View Holder can be used to set the Image and Text and a List of the Model can be used to push the required content. Then in the java fragment of the confirmation screen, the recycler view can be populated by creating a list of model objects with values and passing the list the adapter that we created. A LayoutManager is used to set the layout of the recycler view. The Recycler view can be found by its id and the created Adapter and Layout Manager should be applied/set for that view. Also, the header text is also populated in this java fragment code by identifying the text view id of the confirmation screen and the required content to be displayed can be set using setText function.

The above mentioned summary is the way to create a tab which navigates to another fragment which in turn displays a list of items with a scrollable option. RecyclerView is one of the ways to efficiently load a list of items with a scrollable option.

### **Challenges Encountered**

* While executing/running, instead of having an “app” option, some other configuration was selected which led to the application creating some unit test case problems while launching.
* The navigation java codes were not generated as the navigation dependency was not added initially.
* Initially, only Constraint Layout was used for the sign up fun tab. It was not scrollable until Scroll View was added.
* Once an argument is added for passing from one fragment to another, the required java codes will not be generated unless we run the app. Then the generated java functions can be used for passing arguments from one fragment to destination.
* While trying the change ID’s for already existing input fields, views etc., the android studio was not recognising the changes in the ID until the project was cleaned.
* Did not return the size of the list for recycler view in getItemCount function in the Adapter which ultimately led to the problem of recycler view not loading.

### **Test Cases Performed**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case#** | **Test Case Description** | **Expected Result** | **Actual Result** |
| 1 | Check response with valid data in Fun tab | Navigate to confirmation screen | As expected |
| 2 | Check response with invalid data in Fun tab | Navigate to confirmation screen as validation are not added | As expected |
| 3 | Check if the all the input edit texts has hint properties in Fun tab | All input fields with hint properties in Fun tab | As expected |
| 4 | Check response of clicking submit button in Fun tab | Navigate to Confirmation Screen | As expected |
| 5 | Check if First Name and Last Name are displayed in Confirmation Screen | Displaying First and Last Name with welcome message | As expected |
| 6 | Check if the list of items in Confirmation Screen is Scrollable | List of food items with Scrollable option | As expected |
| 7 | Check if the images of food items in the Confirmation screen are loading properly | List of food items with images | As expected |
| 8 | Check if the font of hint text for each input field in Fun tab is different | Hint text of Input fields with different fonts in Fun tab | As expected |
| 9 | Check if the color of hint text for each input field in Fun tab is different | Hint text of Input fields with different colors in Fun tab | As expected |