Ahsanullah University of Science and Technology

Department of Computer Science and Enineering

CSE3224: Information System Design and Software Engineering Lab

Spring 2020

Project Report

Charity Network

Lab Section: A2

Group: 6

Submitted to

Dr.-Ing. Nusrat J. Lisa	**Nibir Chandra Mandal**
Assistant Professor	Lecturer
CSE, AUST	CSE, AUST

Submitted by

Tahiya Ahmed Chowdhury	170204048
Jannatul Ferdaus	170204051
Tamanna Nazmin	170204052

Contents

1 Contribution of Tahiya Ahmed Chowdhury (ID - 170204048)

The Classes and Methods under the classes are documented below:

1.1 Sign up

1.1.1 Firebase Connectivity Authenticating Issues

1. fAuth = FirebaseAuth.getInstance();
2. FirebaseInstanceId.getInstance().getInstanceId()

Used In : onCreate() method of Registration.
Functionality :
- The method is used for checking firebase authentication.
- It will check connectivity issues. If there is no internet connection app will not run at all.
- When the connectivity issue will be solved user can Sign in.

1.1.2 Send Verification Link to Email

3. isVallidEmail(CharSequence target)
4. fAuth.getCurrentUser().sendEmailVerification()

Used In : register.onClick() action of Registration.
Functionality :
- The first method is used for checking whether the email is valid or not.
- If the email is valid then it sends a verification link to users email.

1.2 Sign in

1.2.1 Sign in for verified users

5. fAuth.getCurrentUser().isEmailVerified()
6. fAuth.signInWithEmailAndPassword(email,password)

Used In : loginButton.setOnClickListener() action of Login.
Functionality :
- The first method is used for checking whether the email is verified or not.
- In second method, if the email is verified and the password matches then it lets user login.

1.2.2 Reset Password

7. fAuth.sendPasswordResetEmail()

Used In : resetPassword.setOnClickListener() action of Login.
Functionality :
- The method is used to generate and send a password reset link to user's email.

1.3 Define User Category

8. sendUserToDesignatedHomePage()

Used In : onCreate() method of IntermediateLogin.
Functionality :
• There are three types of user in our app: donor,organization and volunteer. After logging in, the donor is sent to Donation Category UI and the other users are sent to organization Homepage and volunteer Homepage respectively.

1.4 Donation Category

9. sendUserToDesignatedDonationCategory()

Used In : onCreate() method of DonationCategory.
Functionality :
• After logging in, the user who wants to donate chooses a donation category i.e food,medicine,cloth,etc and they are sent to respective pages.

1.5 Integrating google map

1.5.1 Search Location

10. public void onMapReady(GoogleMap googleMap)
11. searchView.setOnQueryTextListener()
12. private void setLocation()

Used In : public class SearchLocation extends FragmentActivity implements OnMapReadyCallback
Functionality :
• The onMapReady function opens the google map in a fragment activity.
• Then user searches the location that he wants to select via searchView.
• The location is shown to the user in google map as his search result and then he can select that location.setLocation method will work here.

1.5.2 Location picker for posts

13. protected void onActivityResult()

Used In : createPost.java
Functionality :
• The location chose on SearchLocation activity is now catched on createPost activity.

1.5.3 Location picker for events

14. protected void onActivityResult()

Used In : createEvent.java
Functionality :
- Similar to the onActivityResult() method of createPost, fetches the location from Search Location activity.

1.5.4 Extract address from location using Reverse Geocoding

15. private void getCityName()

Functionality :
- By appling reverse geocoding, the address of the location selected from google map is determined for storing it in database.

1.6 Fetch user location and display track

1.6.1 Request user's location

16. private void getCurrentLocation()
17. fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient()
18. onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults)

Used In : Eventdetails.java and PostDetails.java
Functionality :
- The currentLocation of user is accessed using fusedLocationProvider class.
- But if the location permission is not granted by user then Permission is requested using the onRequestPermission method.

1.6.2 Get user's location

19. private void getCurrentLocation()
20. private void getCurrentLocation()
21. LocationManager locationManager = (LocationManager) getSystemService(Context.LOCATION$_SERVICE$);

Used In : Eventdetails.java and PostDetails.java
Functionality :
- After location permisssion have been granted, the current Location of user is fetched using the LocationManager instance in getCurrentLocation() method.

1.6.3 Display path from user's location to event address

22. private void getCurrentLocation()
23. initiateLocationRetreiving()
24. DisplayTrack()

Used In : Event details.java
Functionality :

- The currentLocation of user is fetched via getCurrentLocation() method.
- The location of the event whose details the user is requesting is fetched via initiateLocationRetreiving() method.
- Finally the path or track from users current location and fetched location is displayed in google map using the DisplayTrack() method.

1.6.4 Display path from user's location to organization address

Exactly similar to events details, the address of donation sending location is fetched and path is show on google map from users current location.

2 Contribution of Jannatul Ferdaus (ID - 170204051)

The features that are done by this id in the project is described here:

2.1 Profile

2.1.1 Fetch information from database

```
25. protected void onCreate(Bundle savedInstanceState)
```

Functionality :
- The name, email, location and contact of user is fetched from the database.
- Shows information in profile page according to the category i.e. donor, volunteer, organization.

2.2 Edit Profile

The code is differently done for donor, volunteer and organization using a single database table.

2.2.1 Fetch information from database

```
26. protected void onCreate(Bundle savedInstanceState)
```

Functionality :
- All the information of user which is collected during registration is fetched from the database.
- Shows details in edit profile page in a secured labeled space according to the category i.e. donor, volunteer, organization in order to edit and update by the user who's logged in.

2.2.2 Update information in database

```
27. update.setOnClickListener(new View.OnClickListener()
```

Functionality :
- When user click on update button, all the edited information will be updated in database.
- No fields should remain empty.
- All the cases which may cause error are handled very well using proper condition in back-end section.

2.2.3 Show updated information in profile

```
28. protected void onCreate(Bundle savedInstanceState)
```

Functionality :

- After pressing on update button, user will be taken to profile page automatically and the updated information will be showed.
- On the other hand, previous information will be lost from the database and also from profile page.

2.3 Wishlist

2.3.1 Store information in database

```
29. public wishlistModel()
30. public wishlistAdapter()
```

Functionality :

- If donor wants to donate later then, he can click on "add to wishlish" button and the donation details will be stored in database.

2.3.2 View information

```
31. private void getIncomingIntent()
32. private void setValue()
```

Functionality :

- The stored information in database including donation title, time, date and other details will be showed in wishlist page one after another.

2.4 Log Out

```
33. logout.setOnClickListener()
```

Functionality :

- If user i.e. donor, volunteer, organization clicks on logout button from profile page, he will be logged out from the current page and user will be taken to log in page.

2.5 Bottom Navigation

```
34. h1.setOnClickListener()
```

Functionality :

- Bottom navigation for three types of user i.e. donor, volunteer, organization is differently designed and connected according to their design.
- User can conduct only the pages which are allowed for him. For example, explore, event, wishlist, profile is only for donor, whereas pages for volunteer and organization is different.

2.6 Prototyping

35. private void setUser()

Functionality :
- The whole connection of all the pages of the project is done smoothly.
- Some pages are used for all users like edit profile. To stay away from bugs, many conditions are applied in back-end such that the user won't toggle if he presses back button from the common page.

3 Contribution of Tamanna Nazmin (ID - 170204052)

The contribution of this id in the project is described here:

3.1 Organization Dashboard

3.1.1 Add new donation post

36. protected void onCreate(Bundle savedInstanceState)
37. ValidatePostInfo()

Functionality :
- The user who is an organization can add a new post seeking for donation.
- The validatePostInfo() method makes sure that the post is valid and also it sotres all the donation related information including organization name,category,title,description,location,last date of donation etc in the database.

3.1.2 Edit and Delete Posts

38. public class postsAdapter extends RecyclerView.Adapter<postsAdapter.myViewHolder>
39. postModel()
40. getIncomingIntent()
41. setValue()

Functionality :
- The user who is an organization can edit and delete their created post.
- After the post is added, it is received by getIncomingIntent() and then passed in setValue() method.
- And the recyclerView class adds organizations events to their myEvents page.

3.2 Volunteer dashboard

3.2.1 Create new event

42. ValidatePostInfo()

Functionality :
- The user who is an volunteer can add a new event.
- The validatePostInfo() method makes sure that the event is valid i.e event date has not passed the current date and also store all event related information in the database.

3.2.2 Edit and Delete Events

43. public class eventModel
44. public class myEventAadapter extends RecyclerView.Adapter<myEventAadapter.myViewHolder>
45. getIncomingIntent()
46. setValue()

Functionality :

- The user who is an volunteer can edit and delete his events.
- After the event is added, it is received by getIncomingIntent() and then passed in setValue() method.
- And the myEventAdapter recycler view class will let him browse through his own events.

3.3 Event features for Donor

3.3.1 Browse events

47. eventAdapter extends RecyclerView.Adapter<eventAdapter.myViewHolder> implements Filterable
48. public class EventDetails
49. public class eventModel

Functionality :

- The user who is an donor can view events posted by volunteers.
- He can also view event details i.e date,time,address,etc.

3.3.2 Add events to going list

50. protected void onCreate(Bundle savedInstanceState)
51. public class eventModel

Functionality :

- The user can mark an event as going if he is interested.

3.3.3 Events notification

52. public class notificationAdapter extends RecyclerView.Adapter<com.example.charity.notificationAdapter.myView
53. public class NotificationModel

Functionality :

- The notification model holds all the events that the user has marked as going.
- And the notificationAdapter RecyclerView class notifies user of any upcoming events that he has decided on going.

3.4 Donation category

3.4.1 Food

54. public class foodAdapter extends RecyclerView.Adapter<foodAdapter.myViewHolder>implements Filterable
55. public class foodCategory

Functionality :

- Show Posts from Database using recycler View based on food category.
- Posts those are over dated are removed automatically.

3.4.2 Book

56. public class bookAdapter extends RecyclerView.Adapter<bookAdapter.myViewHolder>implements Filterable
57. public class bookCategory

Functionality :

- Show Posts from Database using recycler View based on book category.
- Posts those are over dated are removed automatically.

3.4.3 Cloth

58. public class clothAdapter extends RecyclerView.Adapter<clothAdapter.myViewHolder>implements Filterable
59. public class clothCategory

Functionality :

- Show Posts from Database using recycler View based on cloth category.
- Posts those are over dated are removed automatically.

3.4.4 Medicine

60. public class medicineAdapter extends RecyclerView.Adapter<medicineAdapter.myViewHolder>implements Filterable
61. public class medicineCategory

Functionality :

- Show Posts from Database using recycler View based on medicine category.
- Posts those are over dated are removed automatically.

3.4.5 Others

62. public class otherAdapter extends RecyclerView.Adapter<otherAdapter.myViewHolder>implements Filterable
63. public class otherCategory

Functionality :
- Show Posts from Database using recycler View based on other categories except food,book,medicine and cloth.
- Posts those are over dated are removed automatically.

3.5 About us

64. onCreate()

Used in : aboutUs.java
Functionality :
- Displaying information about our app.