Date: 15/06/2020

Time Taken: 2 hours

Done: Splash screen

Useful Links

How to Make Splash Screen in Android

Android Hide Title Bar and Full Screen Example

Step

1. Create a new `activity` for `splash activity`
2. Change launching `activity` with `splash activity` :

   Go to `manifest.xml` , change the `<intent-filter>...</intent-filter>` to splash activity

   ```
   ```java
   <activity android:name=".SplashActivity">
       <intent-filter>
           <action android:name="android.intent.action.MAIN" />
           <category android:name="android.intent.category.LAUNCHER" />
       </intent-filter>
   </activity>
   <activity android:name=".MainActivity">
   </activity>
   ```
   ```
3. Design the `splash activiy`
4. Add a timer that will trigger next activity

```java
public class SplashActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                startActivity(new Intent(SplashActivity.this, MainActivity.
                finish();
            }
        }, 2000); //means 2 seconds
    }
}
```

5. Hide Title Bar and Full Screen, add the code to `onCreate` function (`requestFeature()` must be called before adding content)

```java
//add before ContentView(...)
//will hide the title
requestWindowFeature(Window.FEATURE_NO_TITLE);
//hide the title bar
getSupportActionBar().hide();
//show the activity in full screen
this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,Window
```

Can Be Improved

1. The transition animation of the two activities

Date: 16/06/2020

Time Taken: 2 hours

Done: Clour theme design, Login UI

Useful Links

Material Design Color Tool

Material Theming

ColorSpace

Date: 17/06/2020

Time Taken: 2 hours

Done: Signup UI

Useful Links

TabLayout Tutorial With Example In Android Studio

Create swipe views with tabs using ViewPager

Create TabLayout Setps

1. In `activity.xml`

```Java
<androidx.viewpager.widget.ViewPager
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/pager"
    android:layout_weight="1"
    android:layout_width="match_parent"
    android:layout_height="0dp">

    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tab_layout"
        android:background="@color/colorPrimaryLight"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</androidx.viewpager.widget.ViewPager>
```

2. In `activity.java` setup pager

3. Create fragment

Date: 17/06/2020

Time Taken: 1 hours

Done: Administrator main UI, Driver main UI, Receiver main UI

Useful Links

How to set title for action bar in android?

Can Be Improved

1. Add `notification button` in action bar
2. Add `profile button` in action bar

Date: 17/06/2020

Time Taken: 4 hours

Done: Create Recyclerview and Cardview

Useful Links

Create a List with RecyclerView

Android RecyclerView, Android CardView Example Tutorial

Create a Card-Based Layout

Step

1. Open the build.gradle file for app module
2. Add the support library to the dependencies section

```java
dependencies {
    ...
    implementation 'androidx.recyclerview:recyclerview:1.0.0'
}
```

3. Add RecyclerView to layout

```Markup
<?xml version="1.0" encoding="utf-8"?>
<!-- A RecyclerView with some commonly used attributes -->
 <androidx.recyclerview.widget.RecyclerView
    android:id="@+id/my_recycler_view"
    android:scrollbars="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

4. Add CardView to a new layout (design the card content structure)

```Markup
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
...>

    <LinearLayout
    ...>

        <TextView
            ...>

        <TextView
            ...>

    </LinearLayout>

</androidx.cardview.widget.CardView>
```

5. Go to activity.java, obtain a handle to the object, connect it to a layout manager, and attach an adapter for the data to be displayed.

6. Add a list adapter in activity.java

Date: 18/06/2020

Time Taken: 4 hours

Done: Create Notification UI and My Account UI

1. Change the action bar title and logo icon

In the `onCreate()` function:

```java
// Change the actionbar title and icon
      getSupportActionBar().setDisplayShowHomeEnabled(true);
      getSupportActionBar().setLogo(R.drawable.ic_person_pin_black_24dp);
      getSupportActionBar().setDisplayUseLogoEnabled(true);
      getSupportActionBar().setTitle("Administrator");
```

> https://stackoverflow.com/questions/14483393/how-do-i-change-the-android-actionbar-title-and-icon

2. Create action bar menu

> https://www.journaldev.com/9357/android-actionbar-example-tutorial

1. Create menu XML resource file in res/menu folder

```java
<menu xmlns:android="https://schemas.android.com/apk/res/android"
    xmlns:app="https://schemas.android.com/apk/res-auto"
    xmlns:tools="https://schemas.android.com/tools" tools:context=".MainAct

    <item
        android:id="@+id/add" android:icon="@android:drawable/ic_menu_add"
    <item
        android:id="@+id/reset" android:icon="@android:drawable/ic_menu_rev
    <item
        android:id="@+id/about" android:icon="@android:drawable/ic_dialog_i
    </item>
    <item
        android:id="@+id/exit"  app:showAsAction="never" android:title="@st
    </item>
</menu>
```

2. In MainActivity.java, in the MainActivity Class:

```java
public class MainActivity extends AppCompatActivity {

    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is pre
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) { switch(item.getIt
        case R.id.add:
            count=(TextView)findViewById(R.id.textView);
            count.setText("Add is clicked");
            return(true);
        case R.id.reset:
            count=(TextView)findViewById(R.id.textView);
            count.setText("Nothing is selected");
            return(true);
        case R.id.about:
            Toast.makeText(this, R.string.about_toast, Toast.LENGTH_LONG).s
            return(true);
        case R.id.exit:
            finish();
            return(true);

    }
        return(super.onOptionsItemSelected(item));
    }
}
```

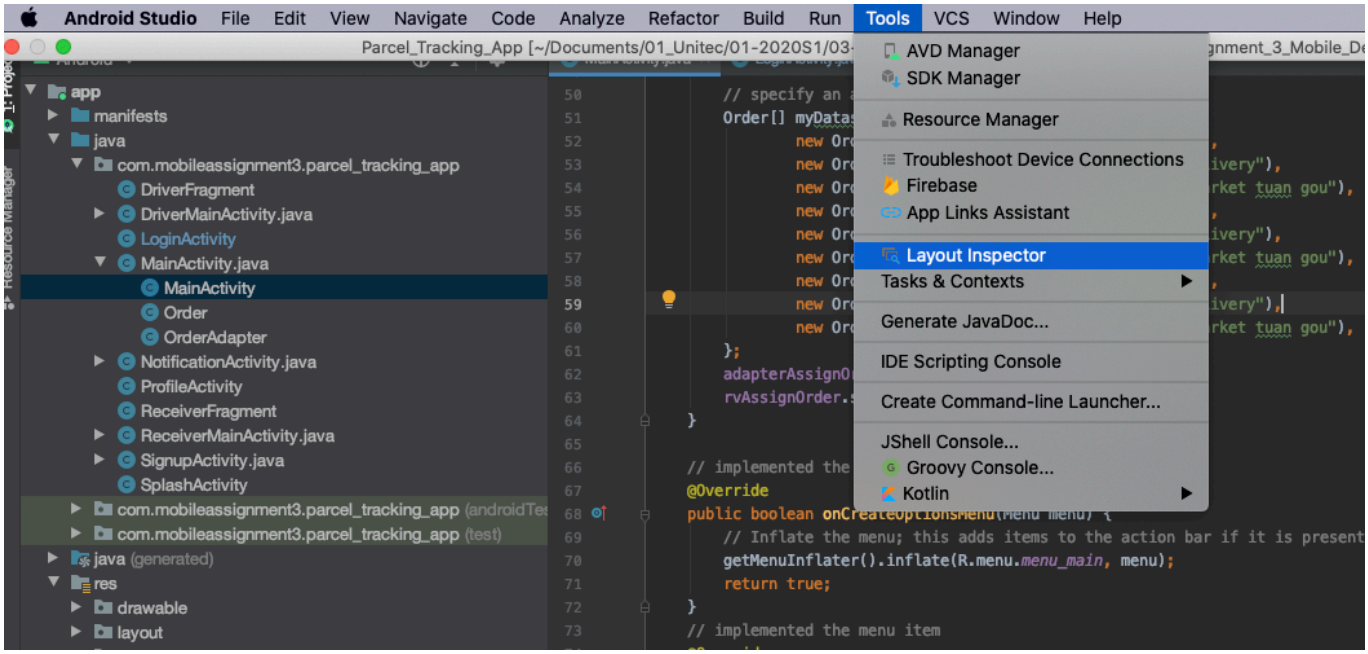
3. Set OnClick Listener on Action Bar Title

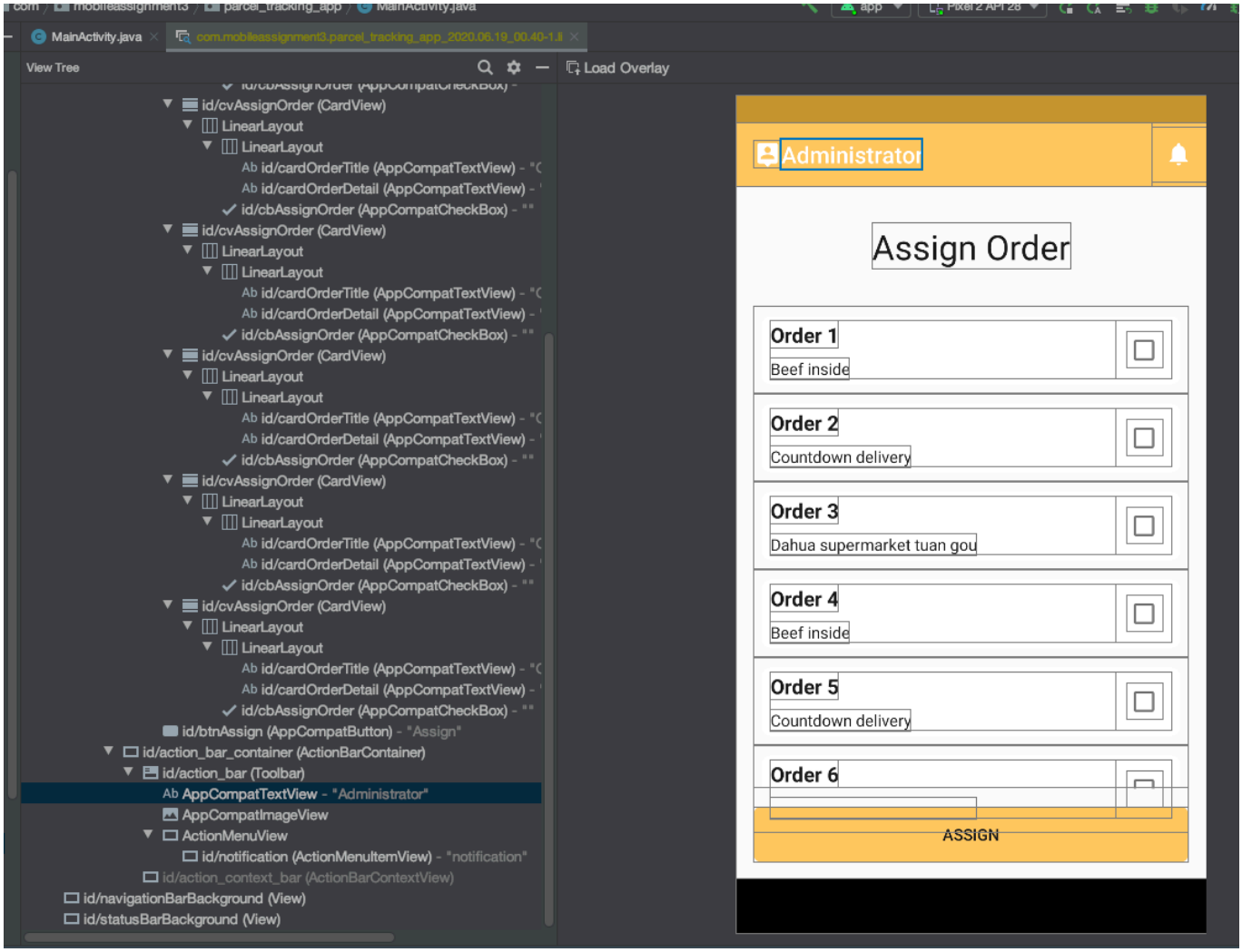
> https://stackoverflow.com/questions/24838155/set-onclick-listener-on-action-bar-title-in-android

1. Find the action bar title id

   Open the activity in emulator, Click on `Tools >>>> Layout Inspector`, in the open window, select the activity which has the action bar, in the open window can see every item of that activity with

details, like `id`



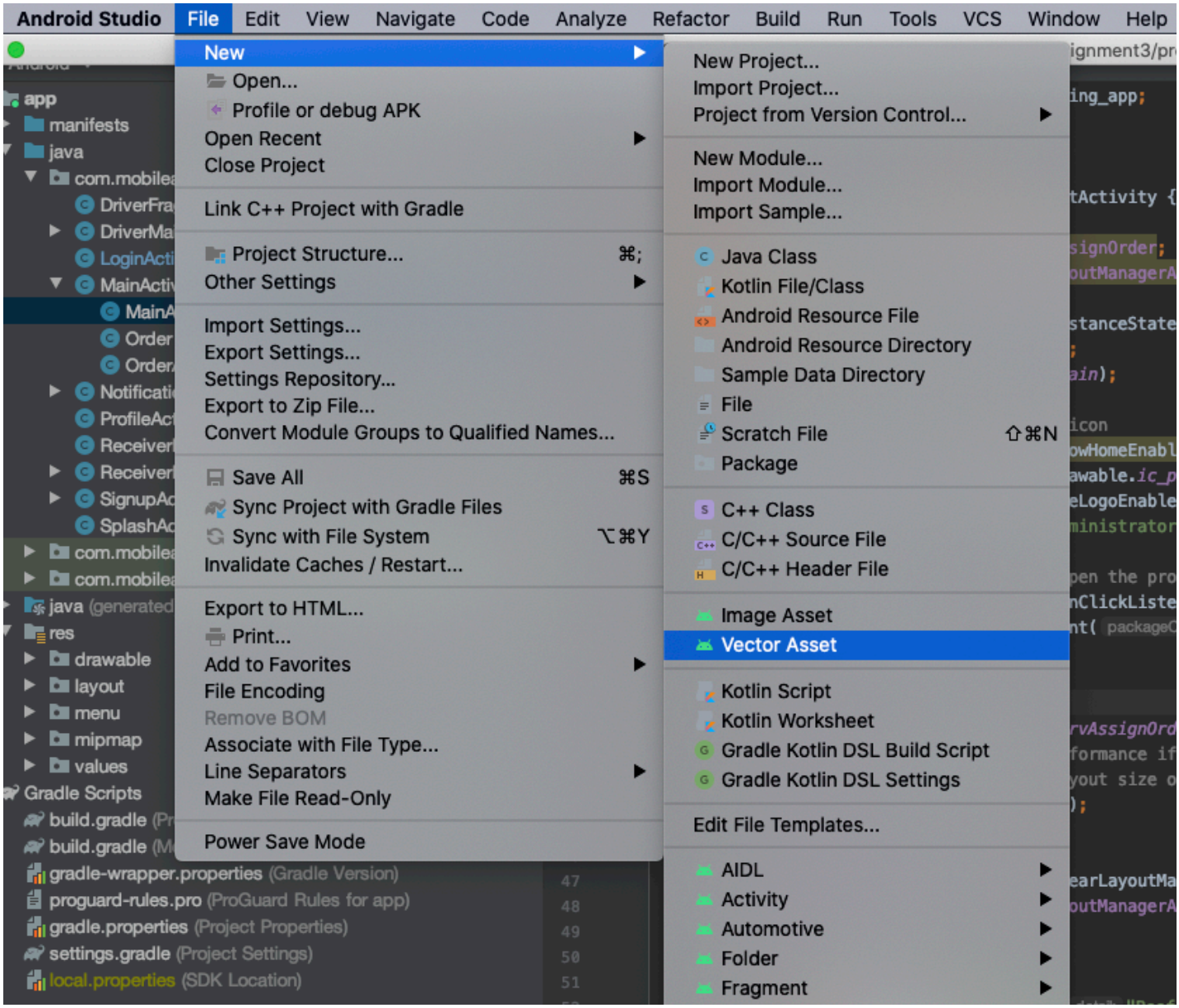


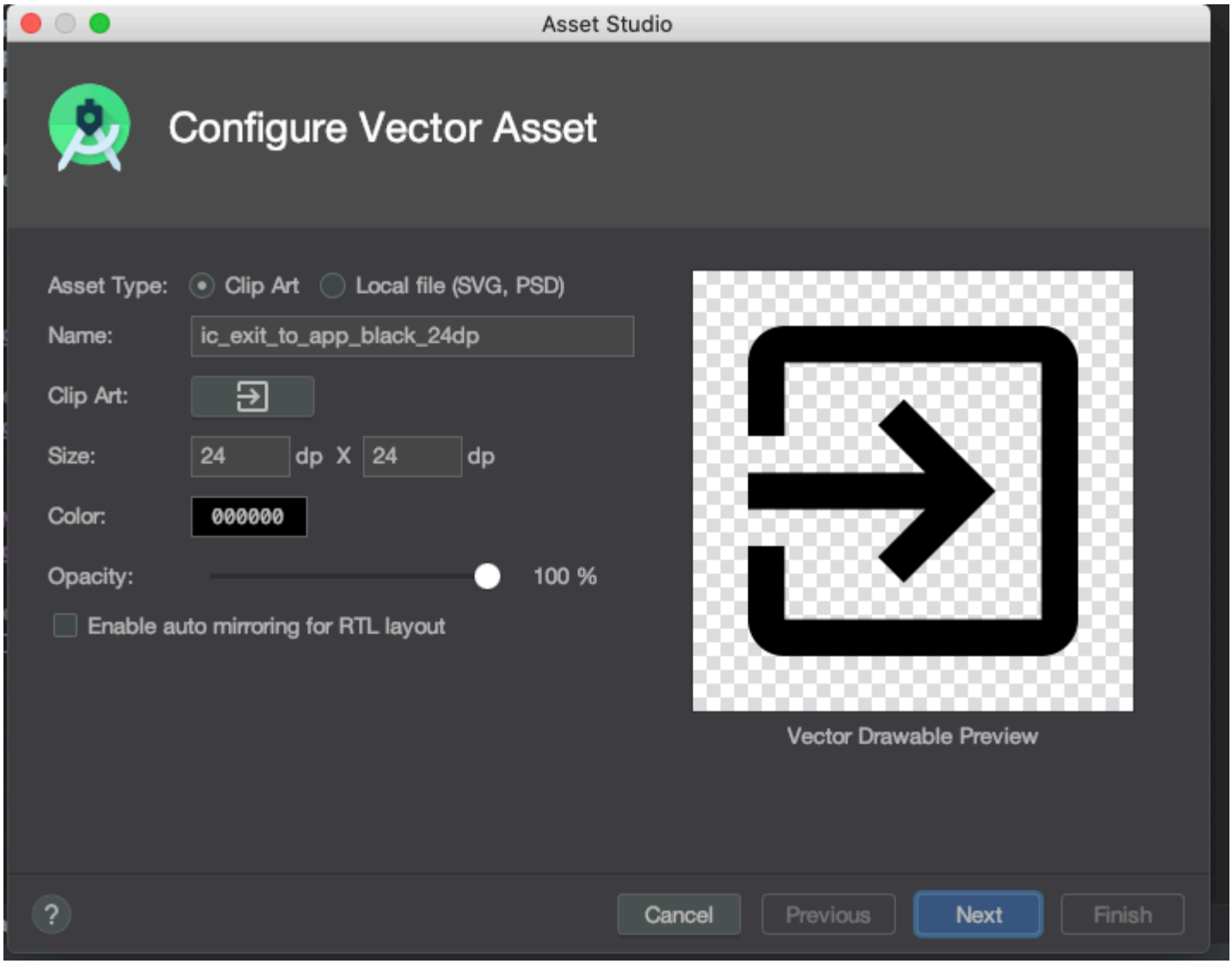
2. In the MainActivity, in the `OnCreate()` function:

```java
// Click the action bar title to open the profile activity
findViewById(R.id.action_bar).setOnClickListener(new View.OnClickListener()
    @Override
    public void onClick(View v) {
        Intent myIntent = new Intent(MainActivity.this, ProfileActivity.cla
        startActivity(myIntent);
    }
});
```

4. Use the build in Icon

Click on `File >>>> New >>>> Vector Asset` , can select the icon and it will be added into the `res/drawable`





Date: 18/06/2020

Time Taken: 1 hours

Done: Discussion Class Diagram

1.1 administrator, driver, receiver all have the same attribute as the user: username, email, password

1.2 user has a group/role attribute: administrator/receiver/driver

1.3 delivery attribute(basic, necessary): delivery address, receiver, status, assugned driver, tracking number

1.4 delivery status: `ready_to_delivery_today` , `out_for_delivery` , `on_the_way` , `delivered`

2.1 administrator function: `get(delivery.ready_to_delivery_today)` , `assign(delivery.ready_to_delivery_today)`

2.2 after the administrator assign the delivery, the delivery status change to `out_for_delivery`

3.1 driver function: `get(delivery.assigned_to_me)` , `sendMessage(delivery_of_the_receiver,receiver, estimated_time)`

3.2 after the driver send the delivery message to the receiver, the delivery status change to `on_the_way`

4.1 receiver function: `get(delivery.receiver_is_me)` , `receiveMessage(when mine delivery will come to me)` , `view(tracking_delivery)` , `confirm(receive_the_delivery)`

4.2 after the receiver confirm the delivery, the delivery status change to `delivered`

Date: 19/06/2020

Time Taken: 1 hours

Done: Logo design

Useful Links

FreeLogoDesign

Date: 23/06/2020

Time Taken: 2 hours

Done: Google Map Directions API

Useful Links

Google Maps Platform Web Services Directions API

Android GoogleMap Example to Draw Route Between Two Locations

Date: 24/06/2020

Time Taken: 0.5 hour

Done: Logout and clear current Activity stack and launch a new Activity

Useful Links

[Clear your current Activity stack and launch a new Activity]

If want to clear current Activity stack and launch a new Activity (for example, logging out of the app and launching a log in Activity), there appears to be two approaches.

1. Target (API >= 16)

Calling `finishAffinity()` from an Activity

1. Target (11 <= API < 16)

```Java
Intent intent = new Intent(this, LoginActivity.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK
startActivity(intent);
finish();
```

Date: 25/06/2020

Time Taken: 1 hour

Done: Let the user do not need to login again if not logout

Steps

In the `PermissionActivity`, chenk whether there is a user.

```Java
if (checkPermission(Manifest.permission.ACCESS_FINE_LOCATION, this.ACCESS_FINE_
    FirebaseController controller = new FirebaseController();
    FirebaseUser currentUser = controller.getCurrentFirebaseUserObject();
    if (currentUser == null) { // Not logged in, go to LoginActivity
        Intent myIntent = new Intent(PermissionActivity.this, LoginActivity.cla
        startActivity(myIntent);
        finish();
    } else { // Login session still valid, go to activity according to user rol
        controller.updateUIafterLogin(this, true);
    }
}
```

Date: 25/06/2020

Time Taken: 3 hour

Done: Login instrumented unit test

Useful Links

Build instrumented unit tests

Steps

1. In app's module-level build.gradle file, specify these libraries as dependencies:

```java
dependencies {
    ...
    androidTestImplementation 'androidx.test:runner:1.1.0'
    androidTestImplementation 'androidx.test:rules:1.1.0'
    }
```

2. Specify AndroidJUnitRunner as the default test instrumentation runner in app's module-level build.gradle file:

```java
android {
    defaultConfig {
        ...
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
}
```

3. Create the login instrumented unit test

```java

public class LoginInstrumentedTest {
    private static final String EMAIL = "driver@driver.com";
    private static final String PASSWORD = "12345678";
    private FirebaseController firebaseController;
    private Context appContext;
    @Before
    public void setup() {
        firebaseController = new FirebaseController();
        appContext = InstrumentationRegistry.getInstrumentation().getTargetCont
    }

    @Test
    public void useAppContext() {
        assertEquals("com.mobileassignment3.parcel_tracking_app", appContext.ge
    }

    @Test
    public void loginTest() {
        firebaseController.loginUser(EMAIL, PASSWORD, new OnCompleteListener<Au
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                assertEquals(true, task.isSuccessful());

                // Should have user after logged in
                FirebaseUser user = firebaseController.getCurrentFirebaseUserOb
                assertNotNull(user);
                assertEquals(EMAIL, user.getEmail());

                firebaseController.getUser(new OnSuccessListener<User>() {
                    @Override
                    public void onSuccess(User user) {
                        // Check user role type
                        assertEquals(User.DRIVER, (Object)user.typeArray.get(0)
                    }
                });

            }
        });
    }
}
```

Date: 26/06/2020

Time Taken: 4 hour

Done: Use Firestore to send message

Useful Links

[Add data to Cloud Firestore]

[Get realtime updates with Cloud Firestore]

Steps

1. In `FirebaseController`, set the data of a document within a collection, explicitly specifying a document identifier.

```java
ParcelMessage data = new ParcelMessage(title, message, user.getEmail(), rec
db.collection("messages").document(receiverEmail)
        .set(data)
        .addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Log.d(TAG, "DocumentSnapshot successfully written!");
                if (listener != null) {
                    listener.onSuccess(aVoid);
                }
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Log.w(TAG, "Error writing document", e);
                if (failureListener != null) {
                    failureListener.onFailure(e);
                }
            }
        });
```

2. In `FirebaseController`, listen to realtime update

```java
public void listenToMessage(String receiverEmail, final long timestamp, fin
    final DocumentReference docRef = db.collection("messages").document(rec
    docRef.addSnapshotListener(new EventListener<DocumentSnapshot>() {
        @Override
        public void onEvent(@Nullable DocumentSnapshot snapshot, @Nullable
            if (e != null) {
                Log.w(TAG, "Listen failed.", e);
                return;
            }

            if (snapshot != null && snapshot.exists()) {
                Log.d(TAG, "Current data: " + snapshot.getData());

                if (listener != null) {
                    ParcelMessage message = snapshot.toObject(ParcelMessage
                    if (message.timestamp >= timestamp) {
                        listener.onSuccess(message);
                    }
                }
            } else {
                Log.d(TAG, "Current data: null");
            }
        }
    });
}
```

3. In `DriverMainActivity`, set onClick listener

```java
new FirebaseController().sendMessageToReceiver("Delivery Notification", dri
    new OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid) {
            Toast.makeText(mContext, "Message sent successfully!", Toast.LE
        }
    }, new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText(mContext, "Oops, message sent failed!", Toast.LE
        }
    });
```

4. In `ReceiverMainActivity`, set a create alert dialog function to show the message:

```java
public void onCreateDialog(ParcelMessage message) {
        if (!isRunning) {
            Log.w("ReceiverMainActivity", "App paused, don't show dialog or
            return;
        }
        SharedPreferences.Editor editor = getPreferences(Context.MODE_PRIVA
        editor.putLong("last_message_update", new Date().getTime());
        editor.apply();
        // Use the Builder class for convenient dialog construction
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle(message.title)
                .setMessage(message.content)
                .setPositiveButton("Yay!!", new DialogInterface.OnClickList
                    public void onClick(DialogInterface dialog, int id) { }
                });
        // Create the AlertDialog object and return it
        builder.create().show();
    }
```

Date: 27/06/2020

Time Taken: 2 hour

Done: Update deliveryJob status after driver sent the message

Useful Links

Add data to Cloud Firestore

Steps

1. Follow the documation to create a updat function

```java
public void updateDeliveryJobStatus(){
    DocumentReference document = new FirebaseController().db.collection("us
    document.update("deliveryJobList", deliveryJobArray) // No way to updat
            .addOnSuccessListener(new OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void aVoid) {
                    Log.w("Driver", "updateDeliveryJobStatus OK");
                }
            })
            .addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    Log.w("Driver", "Error updating document", e);
                }
            });
}
```

2. In the sendMessageToReceiver function, update the deliveryJob status and use `notifyItemChanged` to tell the adapter to update the list item view

```java

new FirebaseController().sendMessageToReceiver("Delivery Notification", dri
        new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Toast.makeText(mContext, "Message sent successfully!", Toas
                deliveryJob.setStatus(DeliveryJob.ON_THE_WAY);
                deliveryJobArray.set(position, deliveryJob);
                notifyItemChanged(position); // notify to refresh view, to
                updateDeliveryJobStatus();
            }
        }, new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(mContext, "Oops, message sent failed!", Toas
            }
        })
```

3. In the `onBindViewHolder`, check the deliveryJob status and change the background color

```java
if (deliveryJobArray.get(position).getStatus() == DeliveryJob.ON_THE_WAY){
            holder.cardView.setCardBackgroundColor(Color.LTGRAY);
}
```