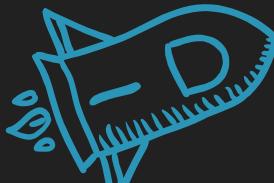


Orbital 2018: Intermediate Android Development



<https://tinyurl.com/orbitalnote>

Before we begin:

1. Overview of Mobile App Development
2. Basic Android Development Workshop
3. Android Studio Installed
4. Google Account (For Firebase)



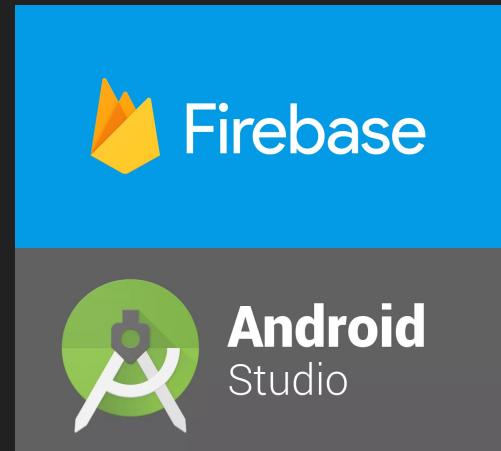
Workshop Overview

1. Part 1: User Authentication

- a. Firebase
- b. LoginActivity
- c. SignupActivity
- d. MainActivity

2. Part 2: CRUD (Create, Read, Update, Delete)

- a. RecyclerView for displaying Notes
- b. Add Notes
- c. Edit Notes
- d. Delete Notes

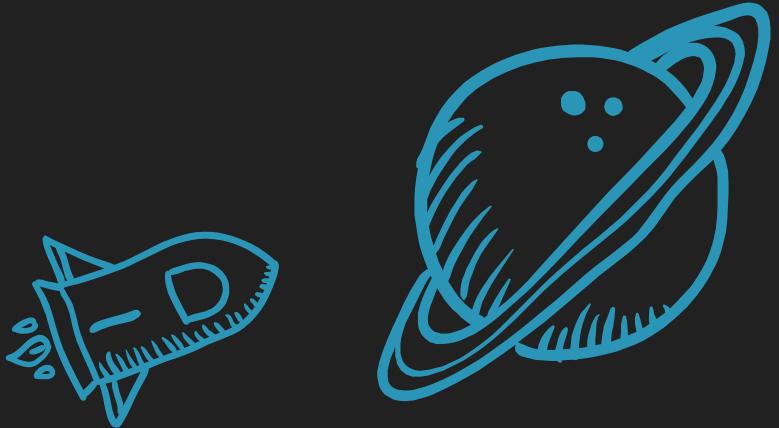


OrbitalNotes (Note Taking App)

1. Simple Cloud-based note taking application
2. Android Mobile Application
3. Backend running on Firebase
4. Features:
 - a. Signup
 - b. Login
 - c. Add Notes
 - d. Edit Notes
 - e. Delete Notes
 - f. View Notes



Part 1: User Authentication



Create New Project

Create New Project

 Create Android Project

Application name
OrbitalNotes

Company domain
example.com

Project location
/Users/luishengjie/workspace/driven-android/OrbitalNotes ...

Package name
com.example.orbitalnotes Edit

Include C++ support
 Include Kotlin support

Cancel Previous Next Finish

Target Android Devices

Create New Project

The dialog box is titled "Target Android Devices". It has a sidebar on the left with icons for Phone and Tablet, Wear, TV, Android Auto, and Android Things. The "Phone and Tablet" section is selected, indicated by a checked checkbox. It shows "API 16: Android 4.1 (Jelly Bean)" as the target. Below it, a note says: "By targeting API 16 and later, your app will run on approximately 99.2% of devices." There is also an option to "Include Android Instant App support" which is unchecked. The other sections (Wear, TV, Auto, Things) each have their own dropdown menus for selecting different API levels.

Target Android Devices

Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

Phone and Tablet

API 16: Android 4.1 (Jelly Bean)

By targeting API 16 and later, your app will run on approximately 99.2% of devices. [Help me choose](#)

Include Android Instant App support

Wear

API 21: Android 5.0 (Lollipop)

TV

API 21: Android 5.0 (Lollipop)

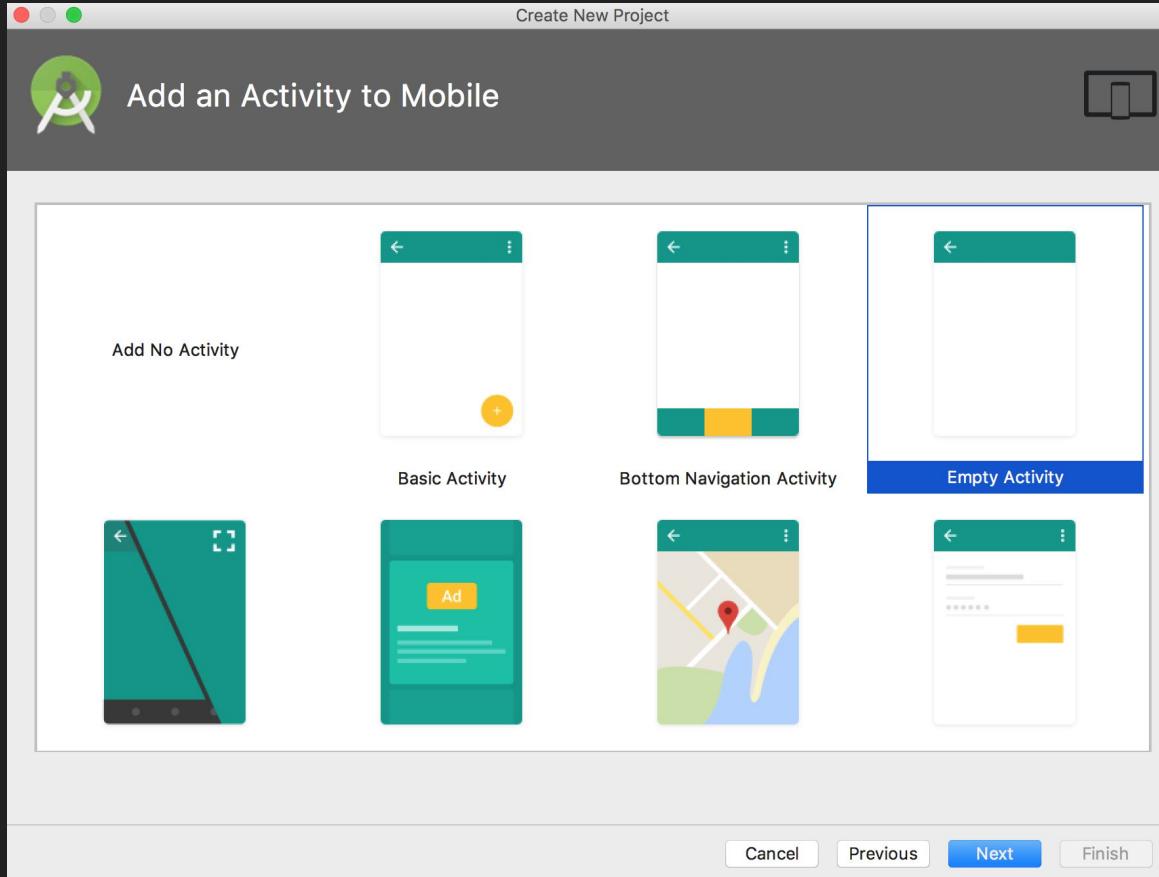
Android Auto

Android Things

API 24: Android 7.0 (Nougat)

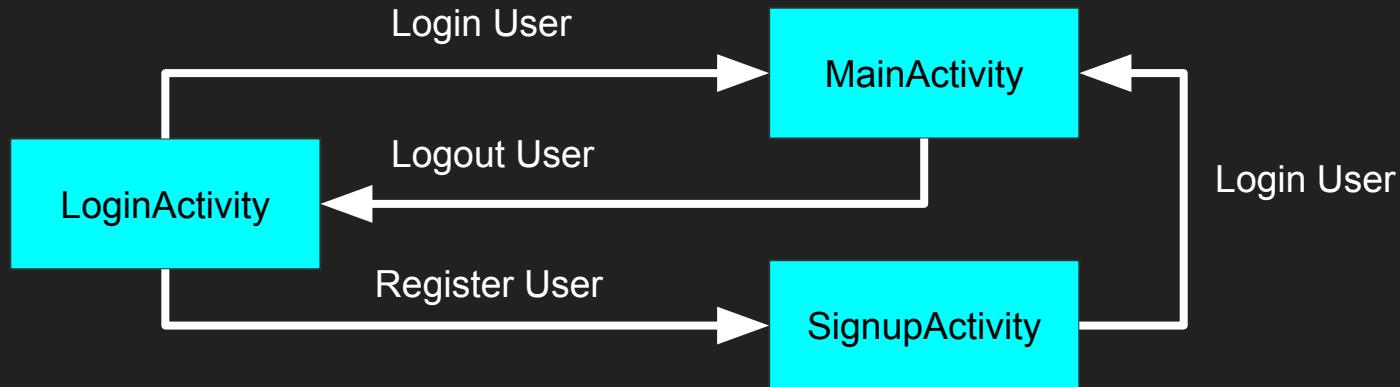
Cancel Previous Next Finish

Select Empty Activity



Layout for User Authentication

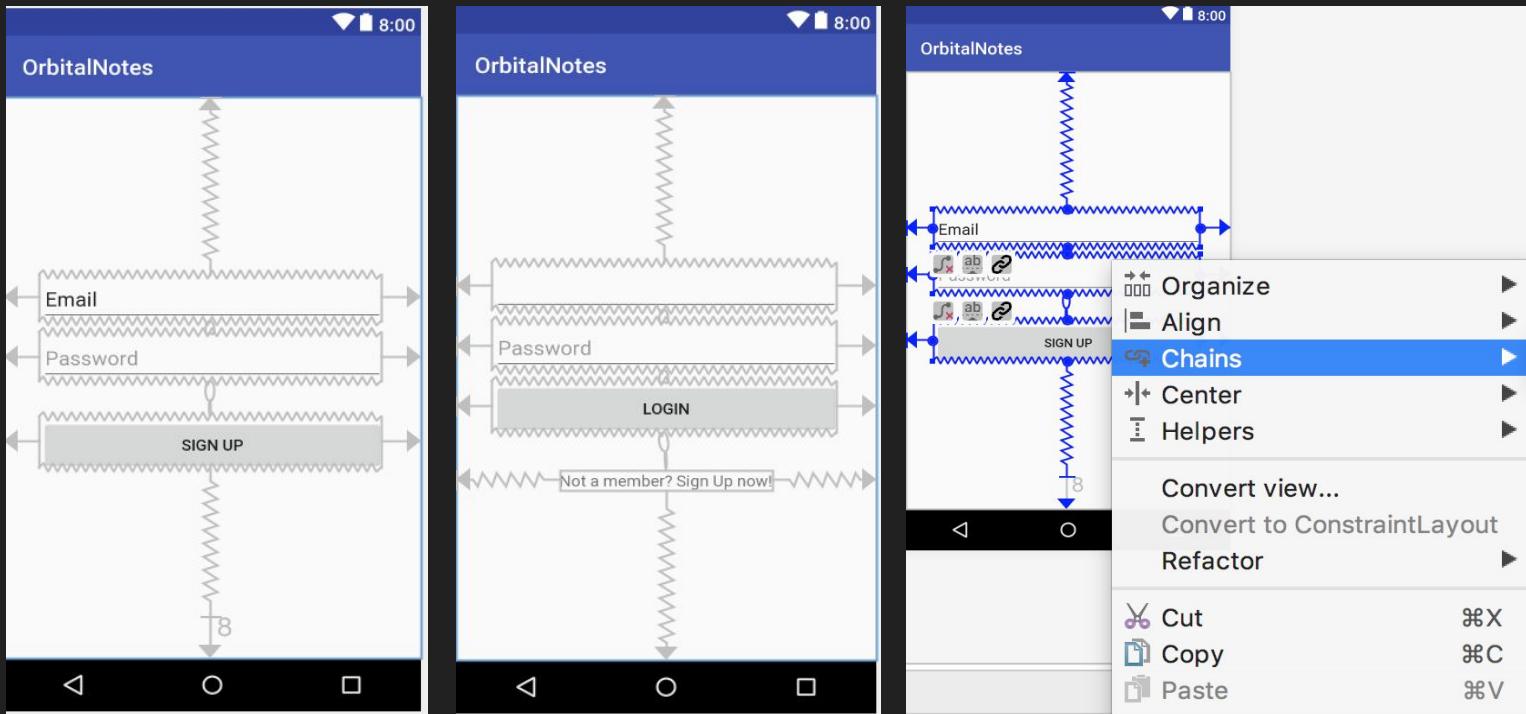
1. MainActivity (Main Page)
2. LoginActivity (Login Page)
3. SignupActivity (Signup Page)



Layout for User Authentication

1. Create 2 new empty Activities
 - o LoginActivity
 - o SignupActivity
2. Add the respective widgets into their .xml files
3. Initialise widgets in the .java files

Layout for User Authentication



Setup Firebase

<https://console.firebaseio.google.com>

The image shows the Firebase console interface. On the left, the main dashboard features a 'Welcome to Firebase!' section with a brief introduction and links to 'Learn more', 'Documentation', and 'Support'. Below this is a section titled 'Your projects using Firebase' with a large blue button labeled '+ Add project' and a smaller box labeled 'Explore a demo project'. A green arrow points from the '+ Add project' button towards the 'Add a project' dialog on the right. The dialog is titled 'Add a project' and contains fields for 'Project name' (set to 'OrbitalNotes'), 'Project ID' (set to 'orbitalnotes2018'), and 'Country/region' (set to 'Singapore'). It also includes a note about Analytics data sharing and 'CREATE PROJECT' and 'CANCEL' buttons at the bottom. The top right corner of the dialog has a close 'X' button.

Firebase

Welcome to Firebase!

Tools from Google for developing great apps, engaging with your users, and earning more through mobile ads.

[Learn more](#) [Documentation](#) [Support](#)

Your projects using Firebase

+ Add project

Explore a demo project

Add a project

Project name

OrbitalNotes

Project ID

orbitalnotes2018

Country/region

Singapore

By default, your Analytics data will enhance other Firebase features and Google products. You can control how your analytics data is shared in your settings at anytime.
[Learn more](#)

CANCEL CREATE PROJECT

Setup Firebase

<https://console.firebaseio.google.com>

Add Firebase to your Android app

1 Register app 2 Download config file 3 Add Firebase SDK

Android Studio instructions Alternatives: [Unity](#) [C++](#)

- [Download google-services.json](#)
- Switch to the Project view in Android Studio to see your project root directory.
- Move the google-services.json file you just downloaded into your Android app module root directory.

google-services.json

Already added the dependencies?
[Skip to the console](#)

CONTINUE

Enable FirebaseAuth

Add Firebase to your Android app

1 2 3

Register app Download config file Add Firebase SDK

The Google services plugin for [Gradle](#) loads the google-services.json file you just downloaded. Modify your build.gradle files to use the plugin.

1. Project-level build.gradle (<project>/build.gradle):

```
buildscript {  
    dependencies {  
        // Add this line  
        classpath 'com.google.gms:google-services:3.2.0'  
    }  
}
```

2. App-level build.gradle (<project>/<app-module>/build.gradle):

```
dependencies {  
    // Add this line  
    compile 'com.google.firebaseio:firebase-core:12.0.1'  
}  
...  
// Add to the bottom of the file  
apply plugin: 'com.google.gms.google-services'  
includes Analytics by default ⓘ
```

compile **'com.google.firebaseio:firebase-core:12.0.1'**
compile **"com.google.firebaseio:firebase-auth:12.0.1"**

Add both lines to
App-level build.gradle



Enable FirebaseAuth

build.gradle(app)

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation 'com.android.support.constraint:constraint-layout:1.1.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
    compile 'com.google.firebaseio:firebase-core:12.0.1'
    compile "com.google.firebaseio:firebase-auth:12.0.1"
    compile 'com.google.firebaseio:firebase-database:12.0.1'
     // CardView
    compile 'com.android.support:cardview-v7:27.1.1'
    implementation 'com.android.support:design:27.1.1'
}
```

Enable FirebaseAuth

Enable Sign-in method for Email/Password

Sign-in providers	
Provider	Status
Email/Password	Enabled
Phone	Disabled
Google	Disabled
Play Games	Disabled
Facebook	Disabled
Twitter	Disabled
Github	Disabled
Anonymous	Disabled

AndroidManifest.xml

Add permission for Internet

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.orbitalnotes">

    <uses-permission android:name="android.permission.INTERNET" /> ← Add permission
    for Internet

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".SignupActivity">
        </activity>

        <activity android:name=".LoginActivity"></activity>
    </application>

</manifest>
```

Firebase Authentication

Firebase methods used for User Authentication

```
FirebaseAuth.getInstance().createUserWithEmailAndPassword(email,  
password).addOnCompleteListener(SignupActivity.this, new OnCompleteListener<AuthResult>(){  
});
```

1. Firebase implementation to create a new user with Email and Password
2. Make sure that sign-in method for email and password in Firebase is configured.

Firebase Authentication

```
FirebaseAuth.getInstance().getCurrentUser()
```

1. Returns null if there is no user logged in
2. Used to check if user is logged in

```
FirebaseAuth.getInstance().signInWithEmailAndPassword(email, password)  
.addOnCompleteListener(LoginActivity.this, new OnCompleteListener<AuthResult>() {}
```

1. Firebase method which allows user to sign in with their email and password

Signup Activity

SignupActivity I

```
public class SignupActivity extends AppCompatActivity {

    private EditText mEditTextEmail;
    private EditText mEditTextPw;
    private Button mBtnLogin;
    private FirebaseAuth auth;

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

        // Initialise widgets
        mEditTextEmail = findViewById(R.id.editTextEmail);
        mEditTextPw = findViewById(R.id.editTextPassword);
        mBtnLogin = findViewById(R.id.buttonSignup);

        // Firebase Auth Instance
        auth = FirebaseAuth.getInstance();
    }
}
```

Linking UI component (View) to code

Button Click Event

```
mBtnSignup.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        //Something happens  
    }  
})
```

1. Android's method for View (Button)
2. Method listens for “click” event from user then do something

SignupActivity II

```
mBtnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String email = mEditTextEmail.getText().toString().trim();
        String password = mEditTextPw.getText().toString().trim();
        // Check if email is empty
        if (TextUtils.isEmpty(email)) {
            Toast.makeText(context: SignupActivity.this, text: "Enter Email Address", Toast.LENGTH_SHORT).show();
        }

        // check if password is empty
        if (TextUtils.isEmpty(password)) {
            Toast.makeText(context: SignupActivity.this, text: "Enter Password", Toast.LENGTH_SHORT).show();
        }

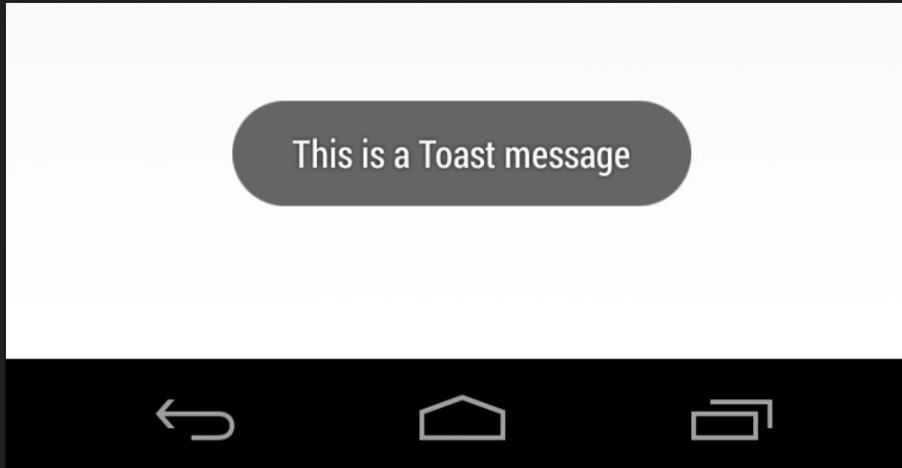
        // Set your own additional constraints

        // Create a new user
        auth.createUserWithEmailAndPassword(email, password).
            addOnCompleteListener( activity: SignupActivity.this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (!task.isSuccessful()) {
                        Toast.makeText(context: SignupActivity.this, text: "Authentication Failed", Toast.LENGTH_SHORT).show();
                    } else {
                        // Signup successful, got to main activity
                        startActivity(new Intent(packageContext: SignupActivity.this, MainActivity.class));
                        // End the activity
                        finish();
                    }
                }
            });
    }
});
```

Toast

```
Toast.makeText(getApplicationContext(), "This is a Toast message", Toast.LENGTH_SHORT).show();
```

1. Android method to display “pop-up” message temporarily
2. Toast.LENGTH_SHORT (Pop-up message last for 2000 milliseconds)
3. Toast.LENGTH_LONG (Pop-up message last for 3500 milliseconds)



Main Activity

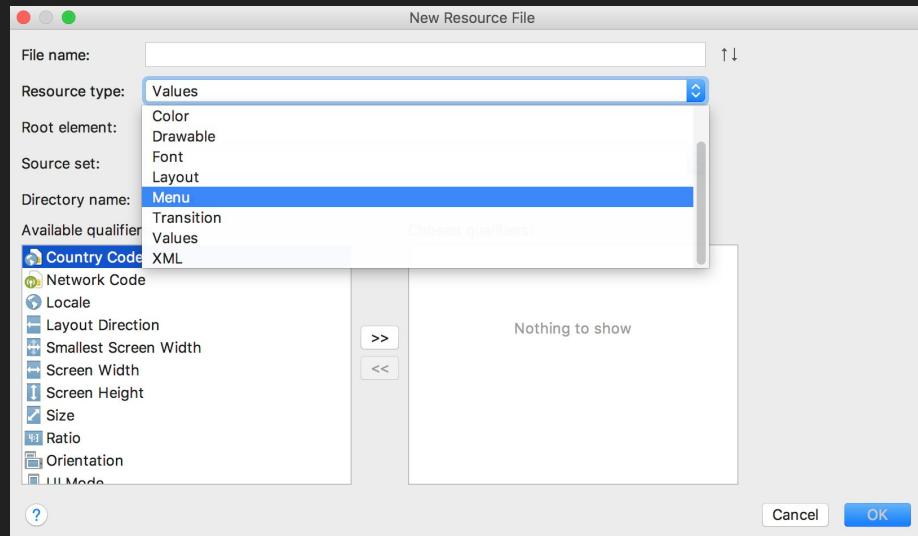
MainActivity

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Check if user is logged in
        // If user is not logged in, direct user to login page
        if (FirebaseAuth.getInstance().getCurrentUser() == null) {
            Intent intent = new Intent( packageContext: MainActivity.this, LoginActivity.class);
            startActivity(intent);
        }
    }
}
```

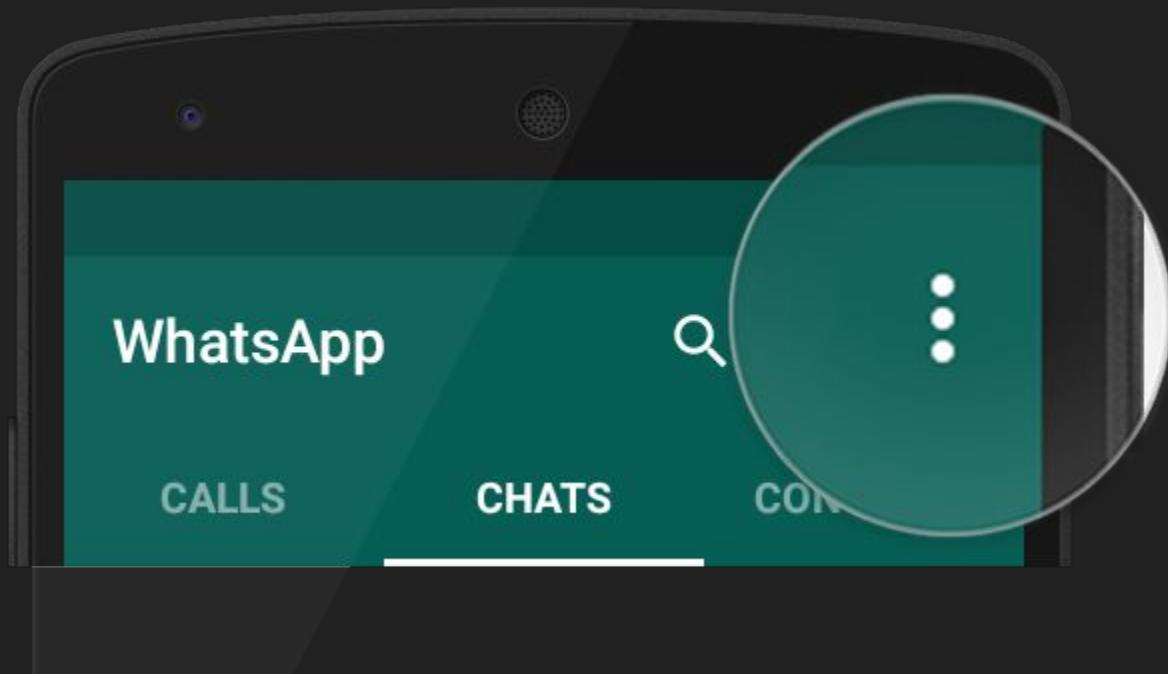
Create Custom Menu for Activity

Create a new layout resource file



```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/logout"
        android:title="Logout"
        app:showAsAction="always"
    />
</menu>
```

Android Menu



MainActivity

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Check if user is logged in
        // If user is not logged in, direct user to login page
        if (FirebaseAuth.getInstance().getCurrentUser() == null) {
            Intent intent = new Intent( mContext: MainActivity.this, LoginActivity.class);
            startActivity(intent);
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main_menu, menu);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.logout:
                userLogout();
                return true;
        }
        return super.onOptionsItemSelected(item);
    }

    // Signout User and return user to LoginActivity
    private void userLogout(){
        FirebaseAuth.getInstance().signOut();
        Intent intent = new Intent( mContext: MainActivity.this, LoginActivity.class);
        startActivity(intent);
        finish();
    }
}
```

Initialise the menu created

When Logout is pressed

Login Activity

LoginActivity |

```
public class LoginActivity extends AppCompatActivity {
    private TextView mTextViewSignup;
    private Button mButtonLogin;
    private EditText mEditTextEmail;
    private EditText mEditTextPw;

    private FirebaseAuth auth;

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

        // Firebase Auth Instance
        auth = FirebaseAuth.getInstance();

        // Initialise widgets
        mButtonLogin = findViewById(R.id.buttonLogin);
        mEditTextEmail = findViewById(R.id.editTextEmail);
        mEditTextPw = findViewById(R.id.editTextPassword);
        mTextViewSignup = findViewById(R.id.textViewSignup);

        mTextViewSignup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), SignupActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

Brings user to the
SignUp Activity



LoginActivity II

```
mButtonLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String email = mEditTextEmail.getText().toString().trim();
        String password = mEditTextPw.getText().toString().trim();
        // Check if email is empty
        if (TextUtils.isEmpty(email)) {
            Toast.makeText(context: LoginActivity.this, text: "Enter Email Address", Toast.LENGTH_SHORT).show();
        }

        // check if password is empty
        if (TextUtils.isEmpty(password)) {
            Toast.makeText(context: LoginActivity.this, text: "Enter Password", Toast.LENGTH_SHORT).show();
        }

        // Set your own additional constraints

        //authenticate user
        auth.signInWithEmailAndPassword(email, password)
            .addOnCompleteListener( activity: LoginActivity.this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (!task.isSuccessful()) {
                        // error occurred
                        Toast.makeText(context: LoginActivity.this, text: "Authentication Failed", Toast.LENGTH_SHORT).show();
                    } else {
                        Intent intent = new Intent(packageContext: LoginActivity.this, MainActivity.class);
                        startActivity(intent);
                        finish();
                    }
                }
            });
    }
});
```

Time to test!

1. Start your emulator
(might take awhile)
2. Sign up on OrbitalNote
3. Login to OrbitalNote



New User is Added

The screenshot shows the Firebase Authentication console interface. On the left, there's a sidebar with sections for DEVELOP, STABILITY, and ANALYTICS. The DEVELOP section is expanded, showing links for Authentication, Database, Storage, Hosting, and Functions. The Authentication link is highlighted with a blue icon. The main content area displays a table of users. A green box highlights the first row, which contains the following data:

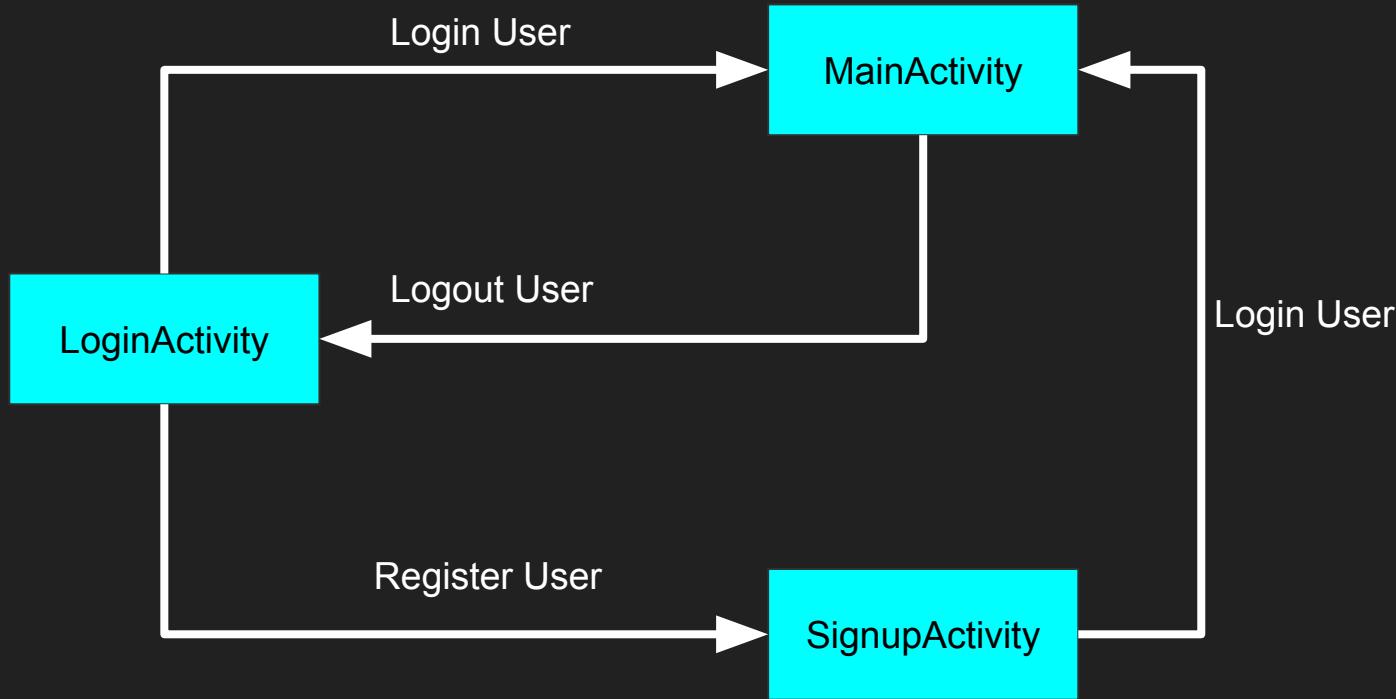
Identifier	Providers	Created	Signed In	User UID ↑
luishengjie8@gmail.com	✉	Apr 22, 2018	Apr 22, 2018	MBxY31j9j6VEiDqOfmiCuKvdgXq2

A green arrow points from the text "New User Created" at the bottom center to the highlighted row in the table.

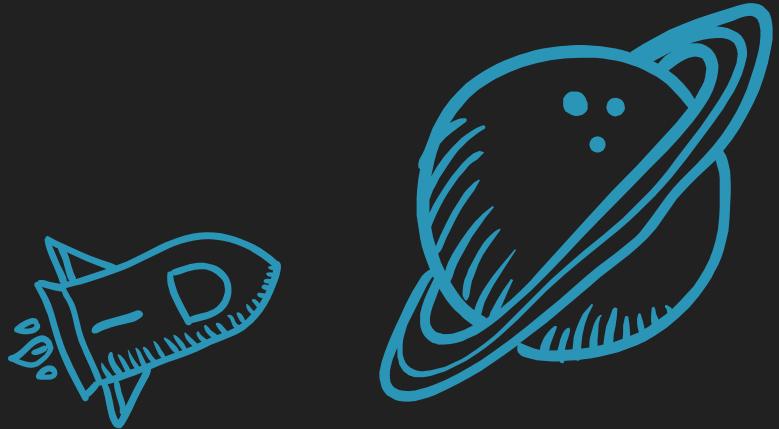
New User Created

<https://console.firebaseio.google.com>

Review on User Authentication

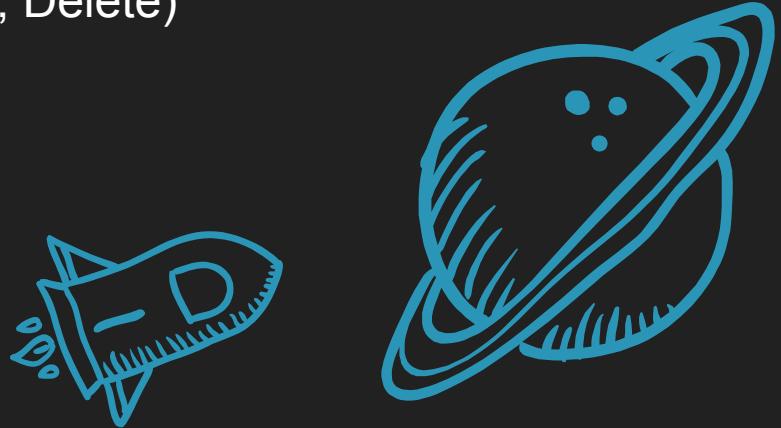


Time for a break

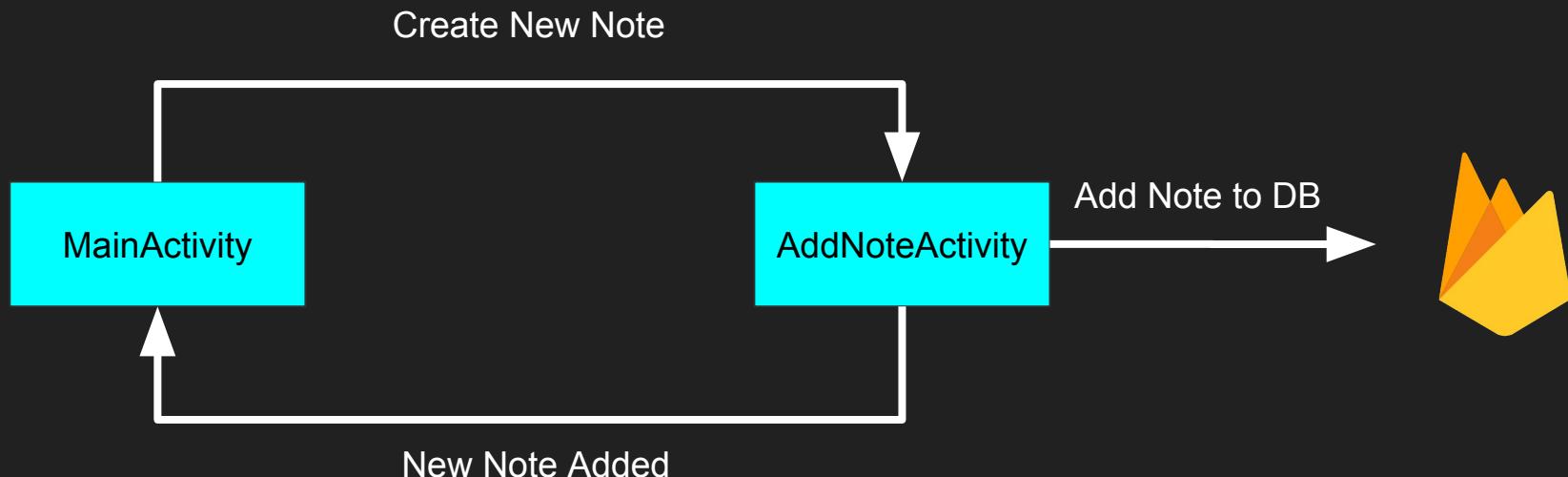


Part 2: CRUD

(Create, Read, Update, Delete)



Create a New Note



Note class

```
package com.example.orbitalnotes;

public class Note {
    private String title;
    private String message;

    public void setMessage(String message) { this.message = message; }

    public void setTitle(String title) { this.title = title; }

    public String getTitle() { return title; }

    public String getMessage() { return message; }

}
```

Add Note Activity

AddNoteActivity

activity_add_note.xml



menu_add_note.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/addnote"
        android:title="Add"
        app:showAsAction="always"
    />
</menu>
```

AddNoteActivity

```
public class AddNoteActivity extends AppCompatActivity {
    private EditText mEditTextTitle;
    private EditText mEditTextMessage;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_note);

        // Initialise Widgets
        mEditTextTitle = findViewById(R.id.editTextTitle);
        mEditTextMessage = findViewById(R.id.editTextMessage);

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_add_note, menu);
        return super.onCreateOptionsMenu(menu);
    }

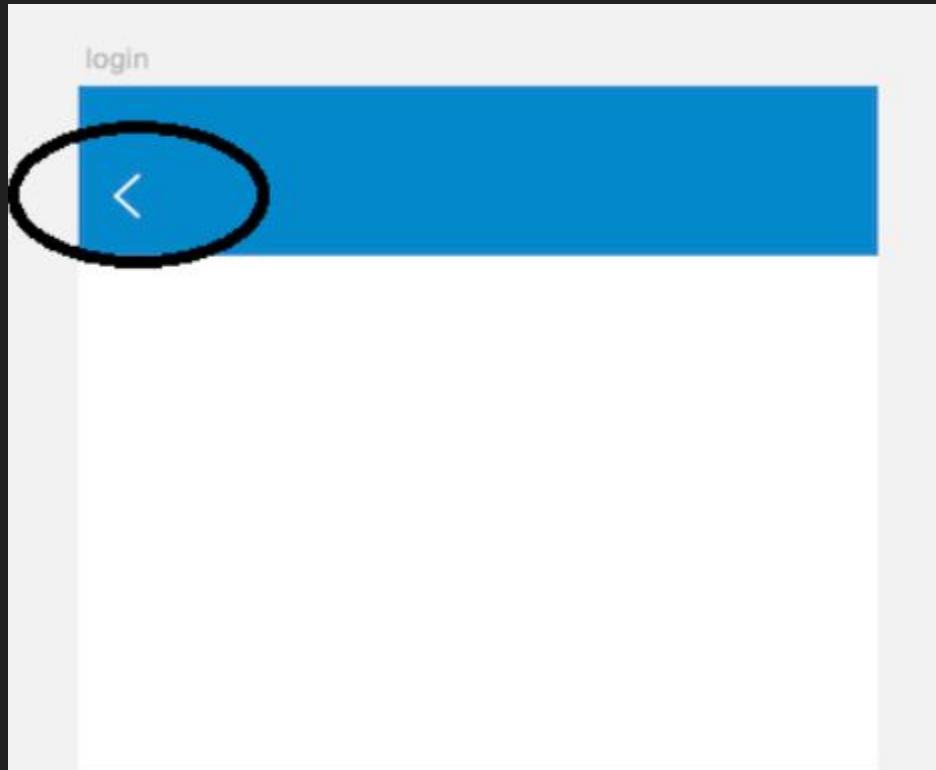
    @Override
    public boolean onSupportNavigateUp() {
        backToMainActivity();
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.addnote:
                // Add Note to Firebase
                addNewNote();
                // Back to MainActivity
                backToMainActivity();
                return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

When the back Button
Pressed

When Add Button
Pressed

onSupportNavigateUp



AddNoteActivity

```
private void addNewNote() {
    String title = mEditTextTitle.getText().toString().trim();
    String message = mEditTextMessage.getText().toString().trim();
    // Write a message to the database
    DatabaseReference mDatabase = FirebaseDatabase.getInstance().getReference("notes");
    // Note object to store title and message
    Note note = new Note();
    note.setTitle(title);
    note.setMessage(message);
    String noteId = mDatabase.push().getKey();
    mDatabase.child(noteId).setValue(note);
}

private void backToMainActivity() {
    Intent intent = new Intent(packageContext: AddNoteActivity.this, MainActivity.class);
    startActivity(intent);
    finish();
}
```

Main Activity

MainActivity

```
public class MainActivity extends AppCompatActivity {
    private FloatingActionButton fabAddNote;

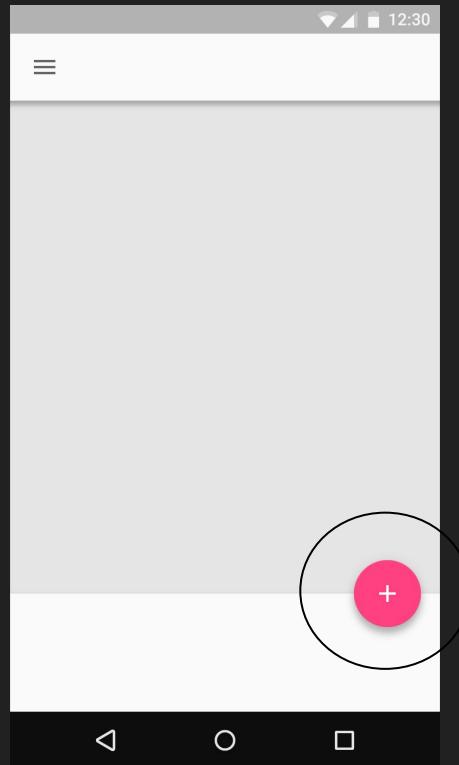
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Check if user is logged in
        // If user is not logged in, direct user to login page
        if (FirebaseAuth.getInstance().getCurrentUser() == null) {
            Intent intent = new Intent( packageContext: MainActivity.this, LoginActivity.class);
            startActivity(intent);
        }

        // Initialise widgets
        fabAddNote = findViewById(R.id.fabAddNote);
        fabAddNote.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                writeNewNote();
            }
        });
    }

    private void writeNewNote() {
        Intent intent = new Intent( packageContext: MainActivity.this, AddNoteActivity.class);
        startActivity(intent);
        finish();
    }
}
```

Floating Action Button

“A **floating action button (FAB)** performs the primary, or most common, action on a screen. It appears in front of all screen content, typically as a circular shape with an icon in its center. FABs come in three types: regular, mini, and extended.” - Google Material Design



New Note is Added (Firebase)

<https://console.firebaseio.google.com>

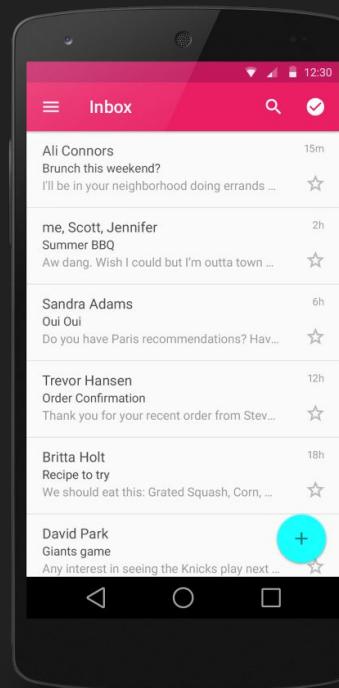
The screenshot shows the Firebase Database interface. On the left, a sidebar menu includes 'Authentication', 'Database' (selected), 'Storage', 'Hosting', and 'Functions'. Below these are sections for 'STABILITY' (Crashlytics, Performance, Test Lab) and 'ANALYTICS' (Dashboard, Events, Audiences, Attribution). The main area displays the database structure for the project 'orbitalnotes2018'. Under the 'notes' node, there is a single child node with the key '-LAdwfM_quc2ZRIGiCv7'. This node contains two properties: 'message' with the value 'helloworld' and 'title' with the value 'hello'.

```
orbitalnotes2018
  notes
    -LAdwfM_quc2ZRIGiCv7
      message: "helloworld"
      title: "hello"
```

View New Note Added (Android)

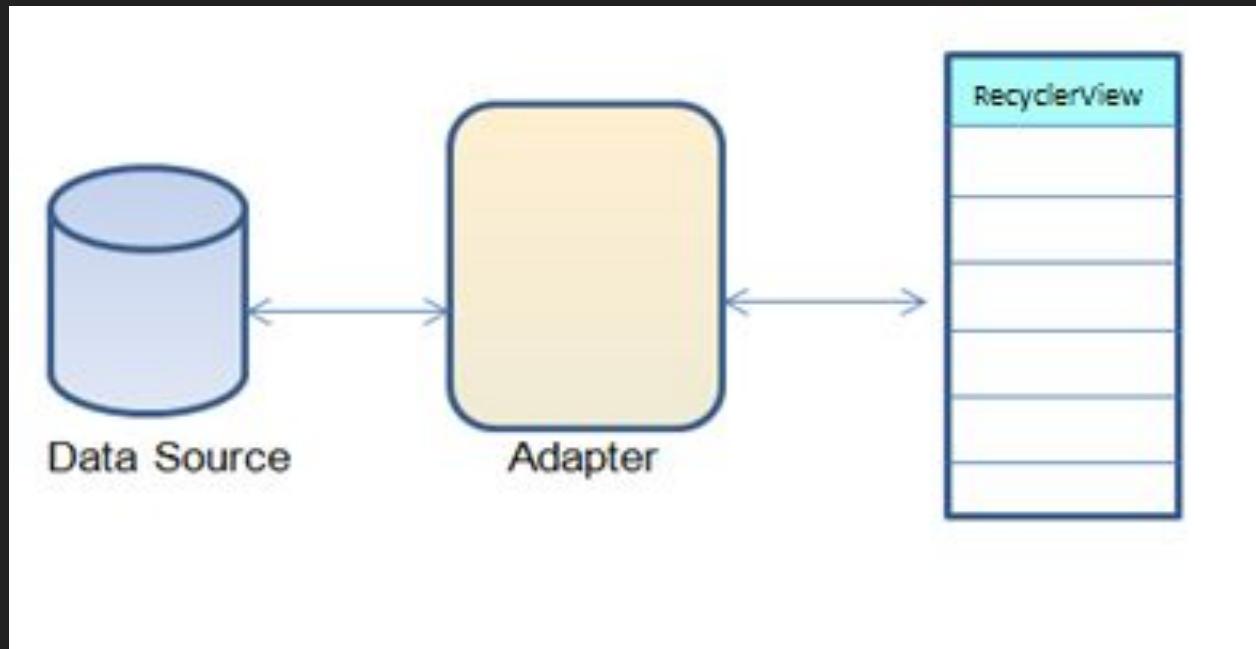
RecyclerView

1. Creates views for items
2. Replaces content of views with new data item when current item not visible
3. Used as a scrollable list



View Adapter

Connecting RecyclerView with the data from Firebase.



card_note.xml

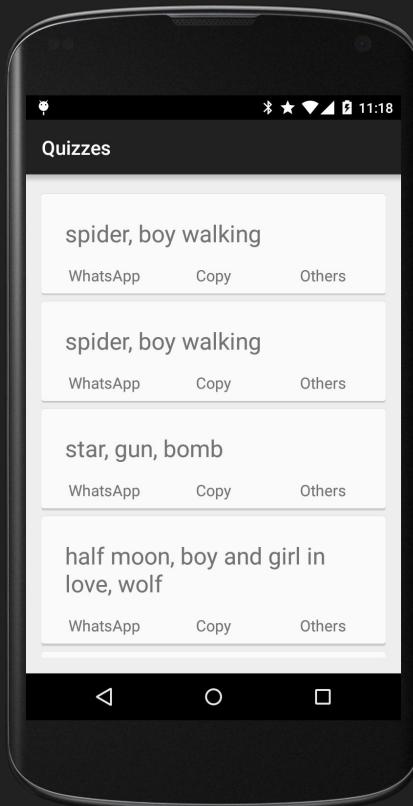
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <android.support.v7.widget.CardView
        android:id="@+id/cardViewNote"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="4dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
            android:padding="8dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:paddingRight="8dp"
                android:text="Title: "/>
            <TextView
                android:id="@+id/cardNoteTitle"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="Card Title"/>
        </LinearLayout>
    </android.support.v7.widget.CardView>
</LinearLayout>
```

CardView



NotesRvAdapter

```
public class NotesRvAdapter extends RecyclerView.Adapter<NotesRvAdapter.NotesViewHolder> {
    private Context mContext;
    private List<Note> notesList;

    public NotesRvAdapter(Context context, List<Note> notes) {
        mContext = context;
        notesList = notes;
    }

    @NonNull
    @Override
    public NotesViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.card_note, parent,
                false);
        NotesViewHolder notesViewHolder = new NotesViewHolder(itemView);

        return notesViewHolder;
    }

    @Override
    public void onBindViewHolder(@NonNull NotesViewHolder holder, int position) {
        final Note note = notesList.get(position);
        holder.mTitle.setText(note.getTitle());
        holder.mCardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(view.getContext(), EditNoteActivity.class);
                view.getContext().startActivity(intent);
                ((Activity)mContext).finish();
            }
        });
    }

    @Override
    public int getItemCount() {
        return notesList.size();
    }

    public class NotesViewHolder extends RecyclerView.ViewHolder {
        public TextView mTitle;
        public CardView mCardView;

        public NotesViewHolder(View view) {
            super(view);
            mTitle = view.findViewById(R.id.cardNoteTitle);
            mCardView = view.findViewById(R.id.cardViewNote);
        }
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fabAddNote"
        android:layout_width="56dp"
        android:layout_height="56dp"
        android:layout_marginBottom="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginRight="16dp"
        android:clickable="true"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:srcCompat="@android:drawable/ic_input_add" />

    <android.support.v7.widget.RecyclerView
        android:id="@+id/recycleViewNotes"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

MainActivity

```
public class MainActivity extends AppCompatActivity {
    private FloatingActionButton fabAddNote;
    private RecyclerView mRvNotes;
    private ArrayList<Note> noteList;
    private NotesRvAdapter notesRvAdapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Check if user is logged in
        // If user is not logged in, direct user to login page
        if (FirebaseAuth.getInstance().getCurrentUser() == null) {
            Intent intent = new Intent(packageContext: MainActivity.this, LoginActivity.class);
            startActivity(intent);
        }
        // Initialise widgets
        fabAddNote = findViewById(R.id.fabAddNote);
        fabAddNote.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                writeNewNote();
            }
        });
        // Part 2: CRUD display
        noteList = new ArrayList<>();
        mRvNotes = findViewById(R.id.recycleViewNotes);
        mRvNotes = findViewById(R.id.recycleViewNotes);
        mRvNotes.setHasFixedSize(true);
        notesRvAdapter = new NotesRvAdapter(context: this, noteList);
        mRvNotes.setAdapter(notesRvAdapter);
        LinearLayoutManager linearLayoutManager = new LinearLayoutManager(context: this);
        mRvNotes.setLayoutManager(linearLayoutManager);
        getFirebaseData(new NotesCallback() {
            @Override
            public void onCallBack(Note note) {
                noteList.add(note);
                notesRvAdapter.notifyDataSetChanged();
            }
        });
    }
}
```

MainActivity

```
private void getFirebaseData(final NotesCallback notesCallback) {
    DatabaseReference reference = FirebaseDatabase.getInstance().getReference();
    DatabaseReference notesRef = reference.child("notes");
    notesRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            // Result will be holded Here
            for (DataSnapshot dataSnap : dataSnapshot.getChildren()) {
                Note note = new Note();
                String title = String.valueOf(dataSnap.child("title").getValue());
                String message = String.valueOf(dataSnap.child("message").getValue());
                note.setTitle(title);
                note.setMessage(message);
                notesCallback.onCallBack(note);
            }
        }

        @Override
        public void onCancelled(DatabaseError databaseError) {
            // Handle db error
        }
    });
}
```

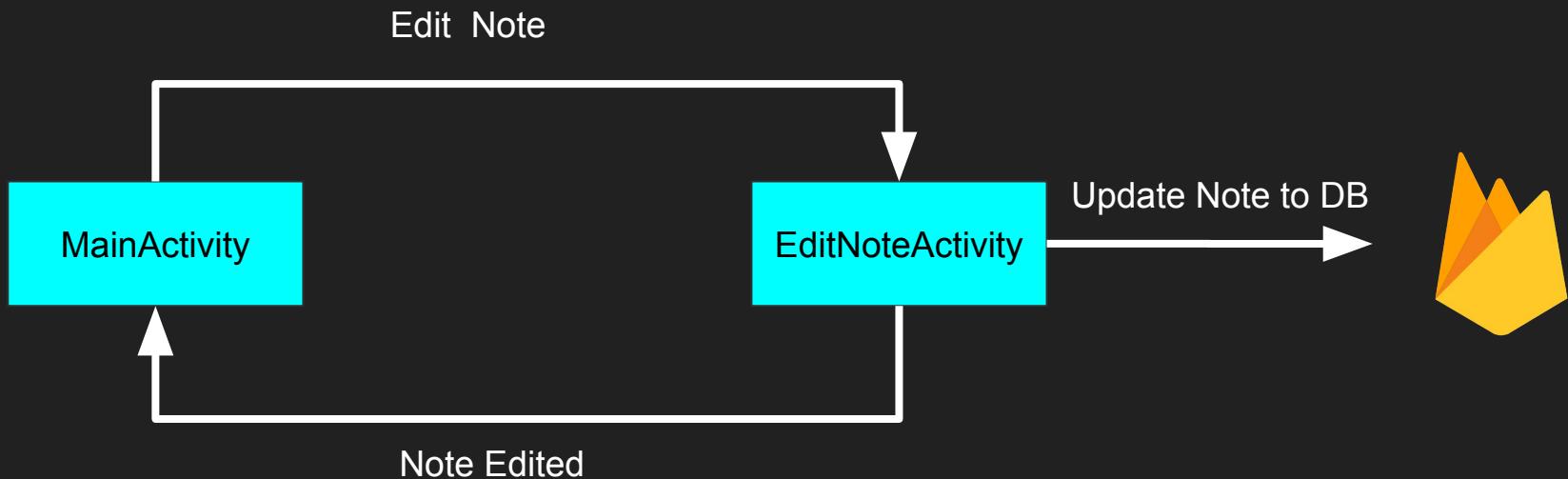
NotesCallback (Interface)

Create your own callback to wait for Firebase to return you the data

```
package com.example.orbitalnotes;

public interface NotesCallback {
    void onCallBack(Note note);
}
```

Edit Note



Implement Parcelable Note

Parcelable Interface used to allow objects to be passed from one activity to another

```
public class Note implements Parcelable{
    private String title;
    private String id;
    private String message;

    public Note(){

    }

    public void setId(String id) {
        this.id = id;
    }

    public void setMessage(String message) { this.message = message; }

    public void setTitle(String title) { this.title = title; }

    public String getTitle() { return title; }

    public String getMessage() { return message; }

    public String getId() {
        return id;
    }
}
```

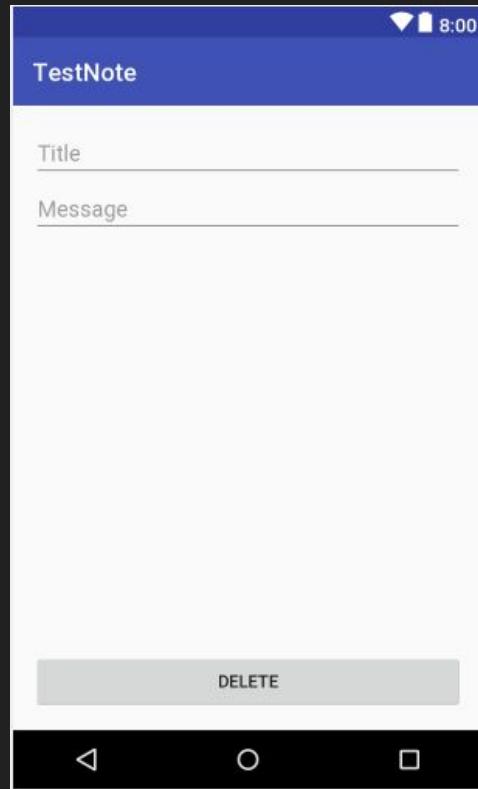
Implement Parcelable Note

```
@Override  
public int describeContents() {  
    return 0;  
}  
  
@Override  
public void writeToParcel(Parcel dest, int flags) {  
    dest.writeString(id);  
    dest.writeString(title);  
    dest.writeString(message);  
}  
  
public Note(Parcel source) {  
    this.id = source.readString();  
    this.title = source.readString();  
    this.message = source.readString();  
}  
  
public static final Parcelable.Creator<Note> CREATOR  
    = new Parcelable.Creator<Note>() {  
        public Note createFromParcel(Parcel in) { return new Note(in); }  
  
        public Note[] newArray(int size) { return new Note[size]; }  
    };
```

NotesRvAdapter

```
@Override
public void onBindViewHolder(@NonNull NotesViewHolder holder, final int position) {
    final Note note = notesList.get(position);
    holder.mTitle.setText(note.getTitle());
    holder.mCardView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(view.getContext(), EditNoteActivity.class);
            Bundle bundle = new Bundle();
            bundle.putParcelable("note", note);
            intent.putExtras(bundle);
            view.getContext().startActivity(intent);
            ((Activity)mContext).finish();
        }
    });
}
```

activity_edit_note.xml



EditNoteActivity |

```
public class EditNoteActivity extends AppCompatActivity {
    private EditText mEditTextTitle;
    private EditText mEditTextMessage;
    private String noteId;

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

        // Initialise Widgets
        mEditTextTitle = findViewById(R.id.editTextTitle);
        mEditTextMessage = findViewById(R.id.editTextMessage);

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);

        // Get Note from bundle
        Bundle b = this.getIntent().getExtras();
        if (b != null){
            Note note = b.getParcelable( key: "note");
            mEditTextTitle.setText(note.getTitle());
            mEditTextMessage.setText(note.getMessage());
            noteId = note.getId();

        } else {
            new NullPointerException();
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_add_note, menu);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onSupportNavigateUp() {
        backToMainActivity();
        return true;
    }
}
```

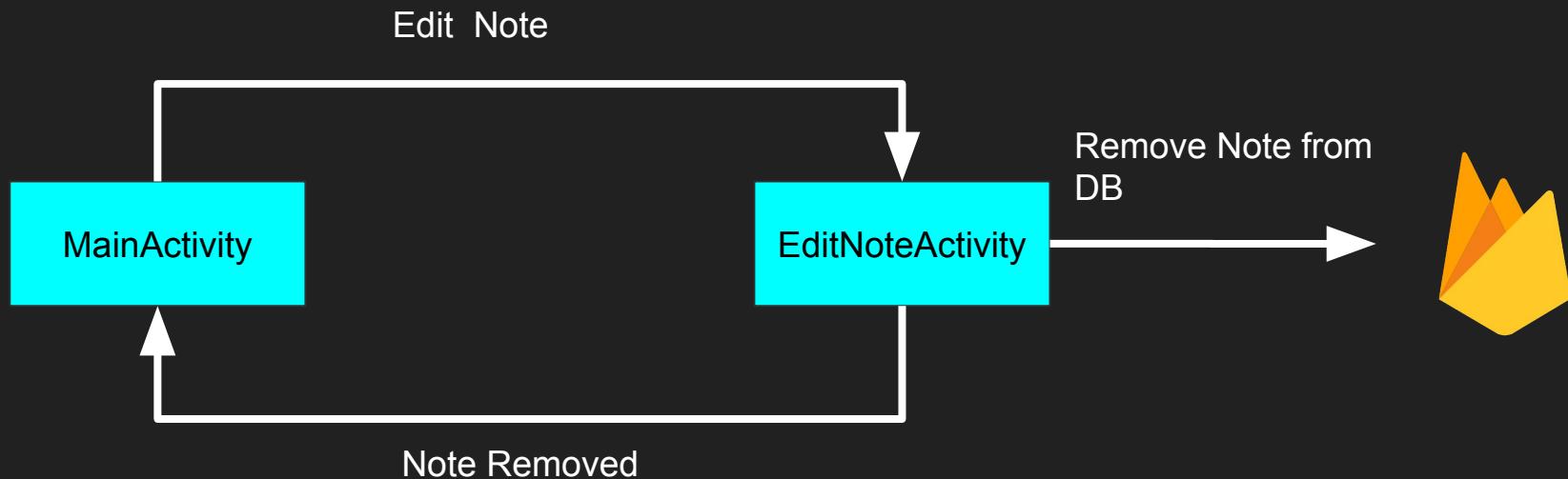
EditNoteActivity II

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.addnote:
            // Add Note to FireBase
            updateNote();
            // Back to MainActivity
            backToMainActivity();
            return true;
    }
    return super.onOptionsItemSelected(item);
}

private void updateNote() {
    Note note = new Note();
    note.setTitle(mEditTextTitle.getText().toString());
    note.setMessage(mEditTextMessage.getText().toString());
    note.setId(noteId);
    DatabaseReference mDatabase = FirebaseDatabase.getInstance().getReference("notes");
    mDatabase.child(noteId).setValue(note);
}

private void backToMainActivity() {
    Intent intent = new Intent(getApplicationContext(), MainActivity.class);
    startActivity(intent);
    finish();
}
```

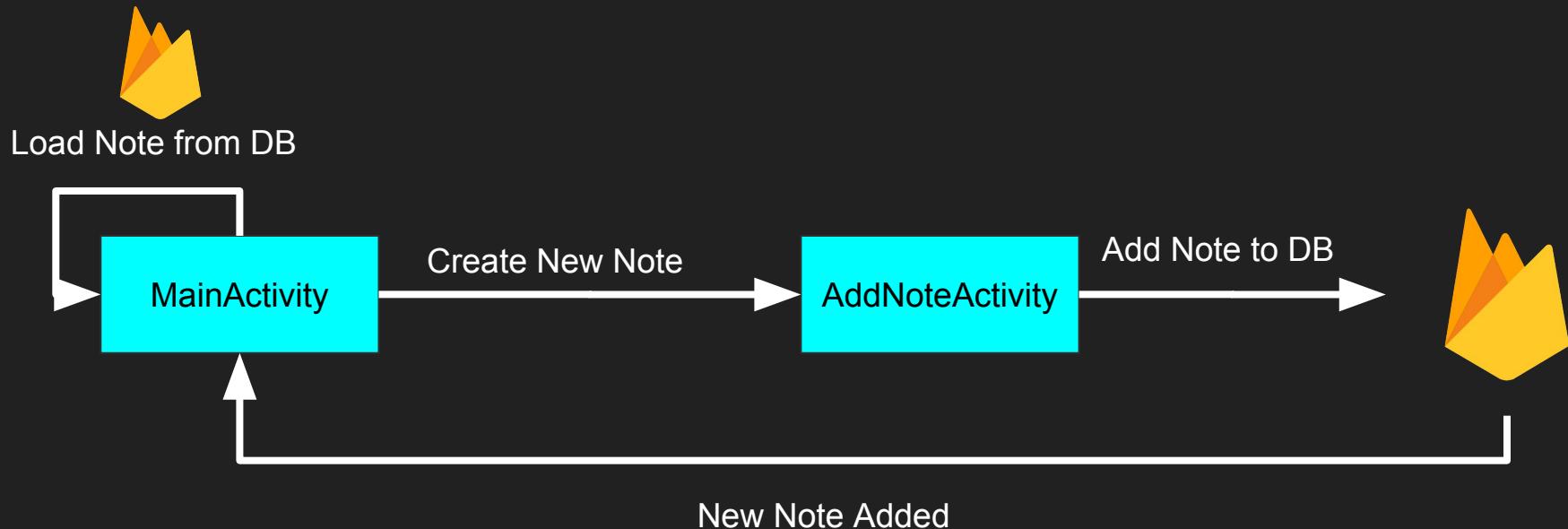
Delete Note



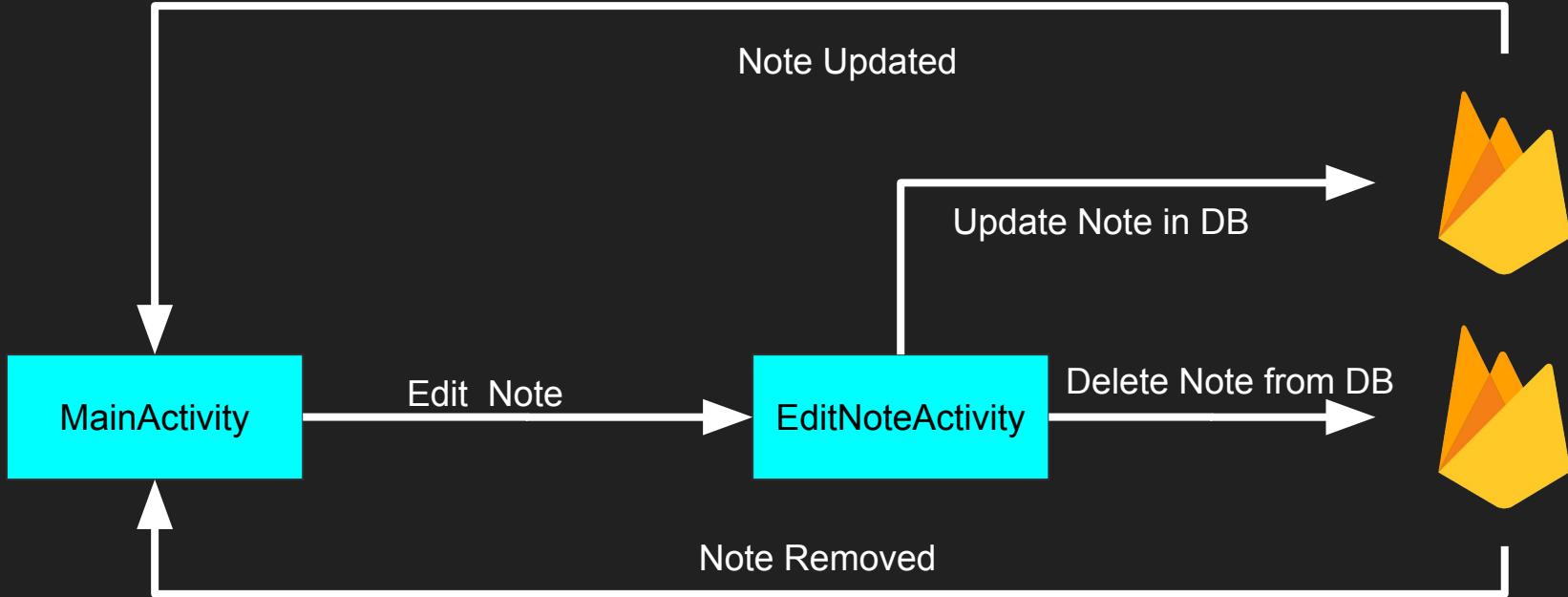
Delete a Note

```
private void deleteNote() {  
    DatabaseReference mDatabase = FirebaseDatabase.getInstance().getReference(s: "notes");  
    mDatabase.child(noteId).removeValue();  
}
```

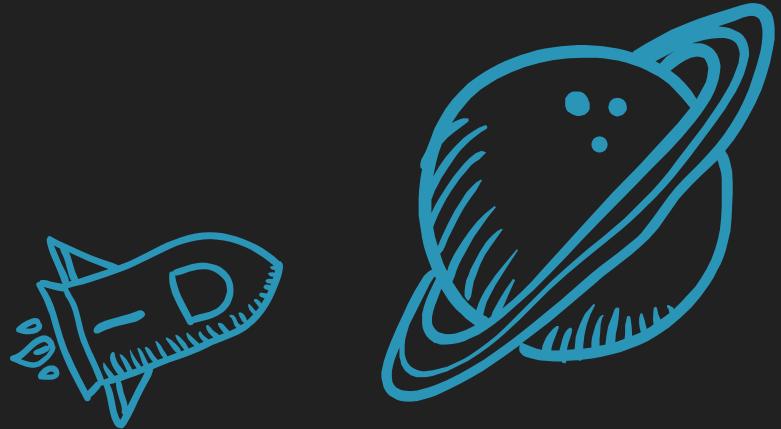
Review on CRUD



Review on CRUD



Improving OrbitalNotes!



Changing app icon

How to change the app icon?

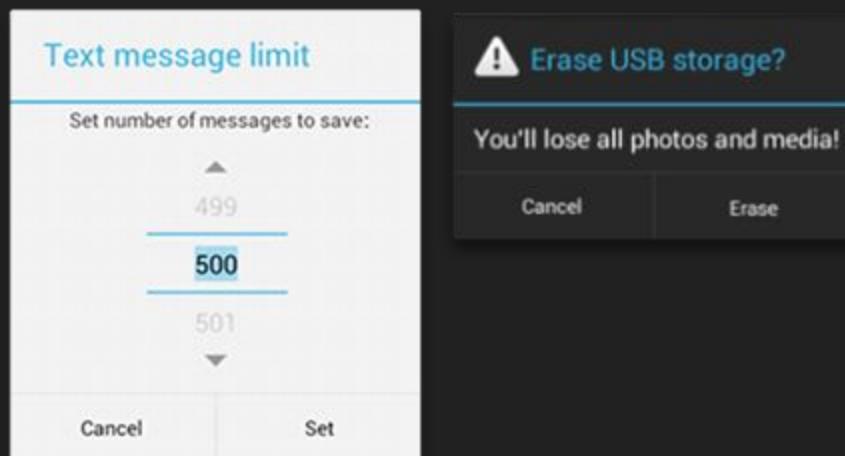
1. Expand the project root folder in the Project View
2. Right Click on the app folder
New->Image Asset
3. In the pop up that appears select the new logo you would like to use (image/clip art/text).



Dialog box

Pop-up dialog box - AlertDialog

<https://developer.android.com/guide/topics/ui/dialogs>



SharedPreference

How to store data locally?

Setting value in preference

```
SharedPreferences.Editor editor = getSharedPreferences(MY_PREFS_NAME,  
MODE_PRIVATE).edit();
```

```
editor.putString("noteTitle", "CS1234");
```

```
editor.putInt("noteID", 8);
```

```
editor.apply();
```

SharedPreference

How to store data locally?

Getting value in preference

```
SharedPreferences prefs = getSharedPreferences(MY_PREFS_NAME,  
MODE_PRIVATE);
```

```
String title = prefs.getString("noteTitle", "No Title"); // "No Title" is the default value.
```

```
int id = prefs.getInt("noteID", 0); // 0 is the default value.
```

Thank you

