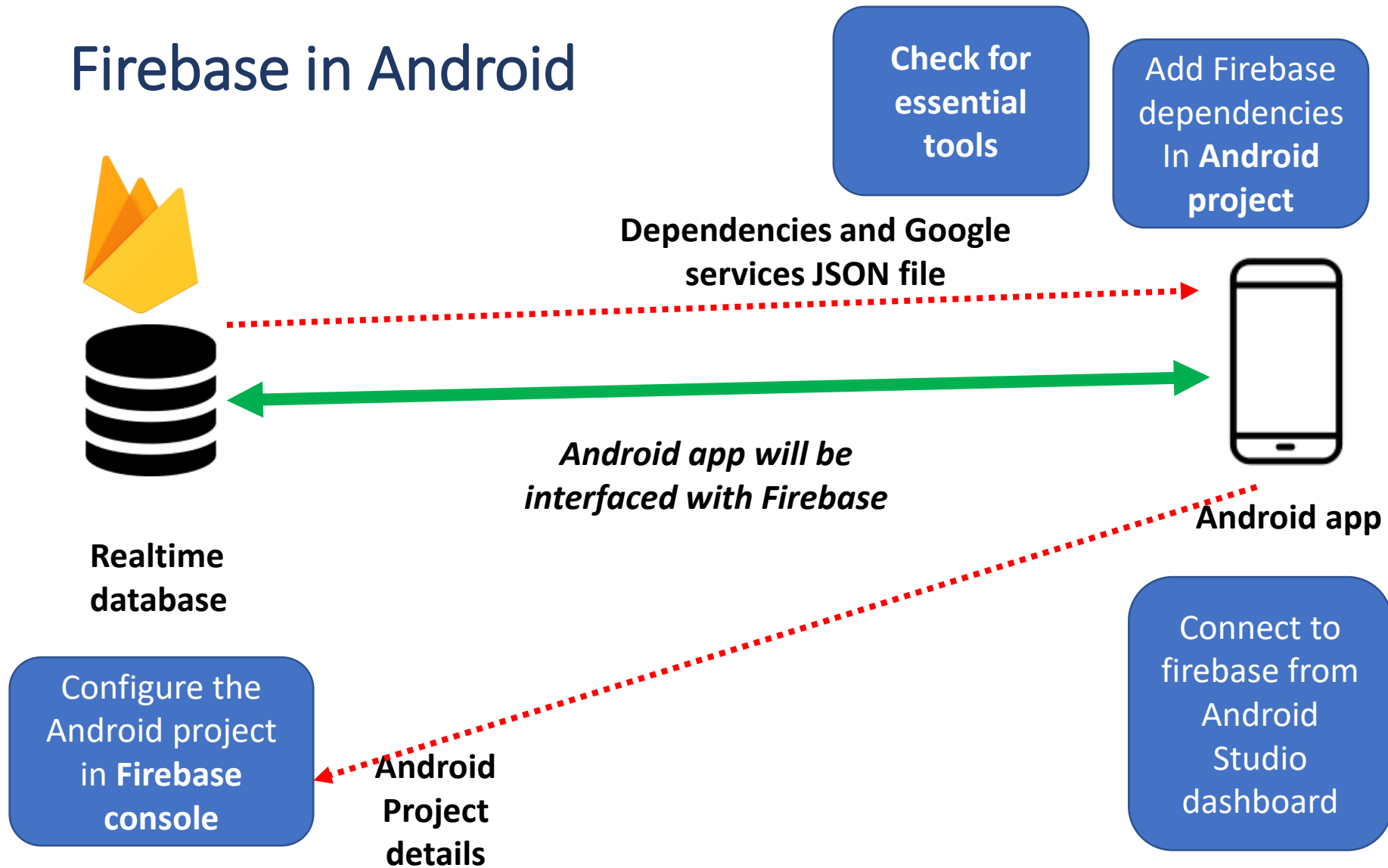


# Summer 2, 2019 - CS 4520/CS5520 – Mobile Application Development

**Pratheep Kumar Paranthaman, Ph.D.,**



# Firestore in Android

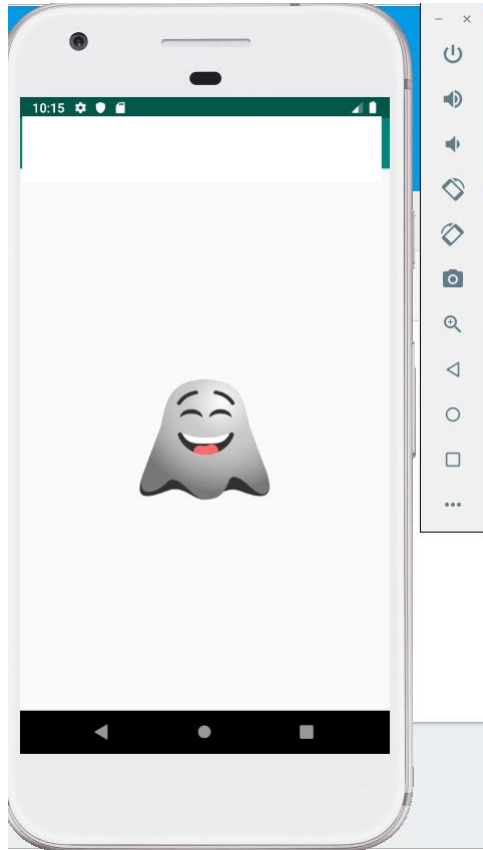


# Launch the App and check the Logcat

- At this point, there shouldn't be any errors on your Logcat. If there are any issues, then the configuration of Firebase has some problems.
- Check for the following
  - Necessary SDK tools
  - Proper versions of Firebase dependencies
    - Check this link - <https://firebase.google.com/docs/android/setup>

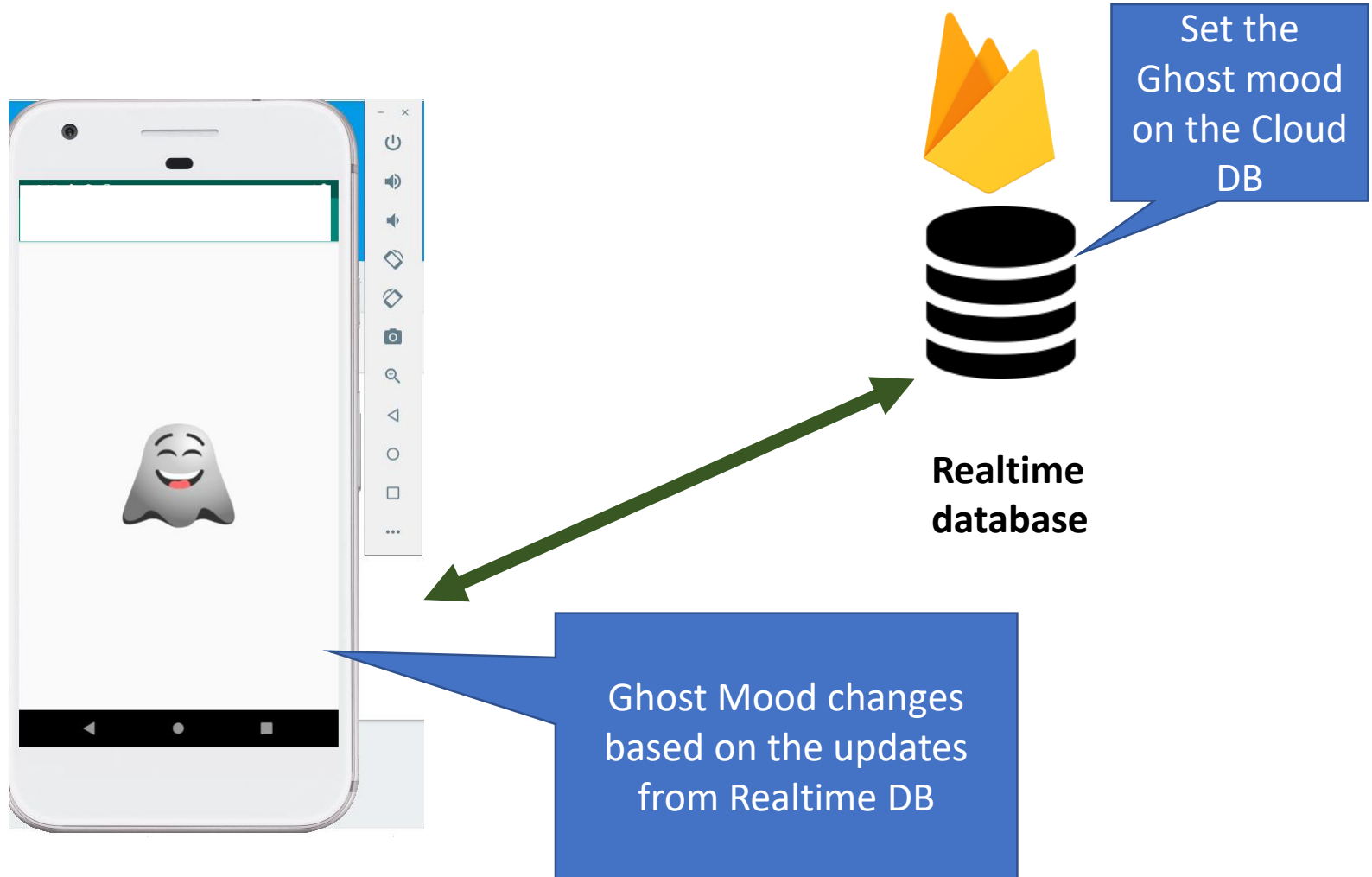
Let's experience Firebase!

## Exercise 6 – Realtime DB Intro



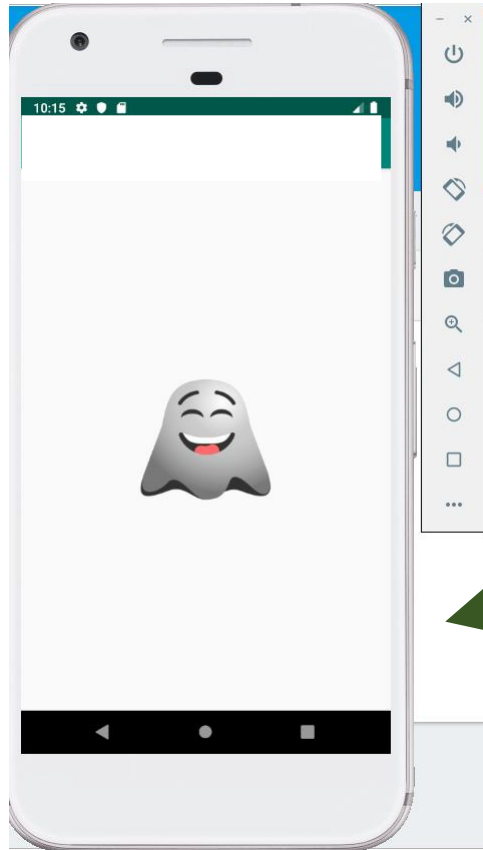
- **Ghost Mood App – revisit**
  - Change the mood on Realtime database and make it to reflect on the App.

# Exercise 6 – Realtime DB Intro



# How to read data from Firebase in realtime?





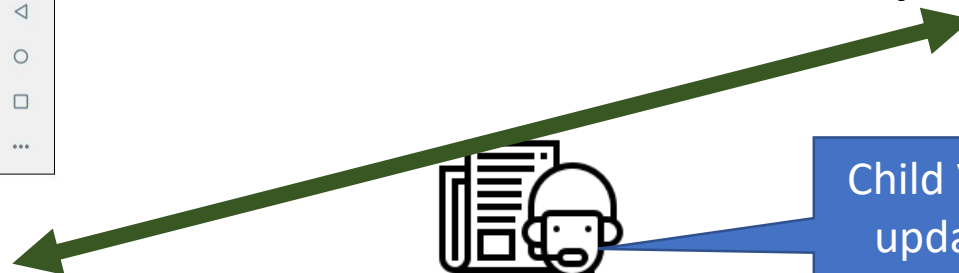
1. Get the reference of root
2. Get the reference of child
3. Set a **listener** on Child

Root



Child

ghostmood	1
Key	Value



Listener

Child Value  
updated

# Manifest Permission

```
<uses-permission android:name="android.permission.INTERNET" />
```

# Retrieve data from Firebase

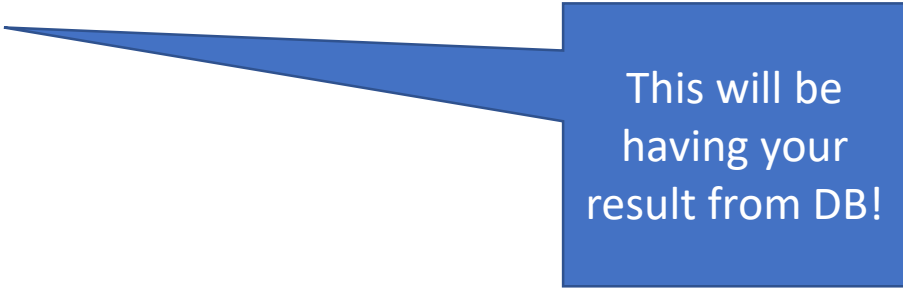
- 1. Get the database root reference

```
DatabaseReference databaseRootReference = FirebaseDatabase.getInstance().getReference();
```

- 2. Get the database child reference

```
DatabaseReference childReference = databaseRootReference.child("childName");
```

- 3. Set an asynchronous Listener(*addValueEventListener()*) on CHILD
  - onDataChange -> **Dat snapshot**
  - onCancelled



This will be  
having your  
result from DB!

# Security Measures

- It's essential to use the security features of the Firebase system for better managing the infrastructure.
- **Authentication:**
  - By default **unauthenticated** users won't be able to access your Firebase.

**request.auth != null**

Configure the security rules under Rules tab :

allow read, write: if request.auth != null;

# Security Measures

- How do you create authentication for the users?
  - Firebase Authentication provides a couple of options
    - Email authentication
    - Phone Authentication
    - Anonymous
    - Third-party
- Using one of the above methods will result in following points:
  - Registers your user to your APP
  - Provides authentication key

# Security Measures

- **Authorization:**

- For Realtime Database/ storage, you can set the user authorization for the data access:
  - Read
  - Write

```
service firebase.storage {  
  match /b/{bucket}/o {  
    match /{allPaths=**} {  
      allow read, write: if request.auth != null;  
    }  
  }  
}
```



Only  
authenticated  
users will be  
able to  
read/write

# Security Measures

- **Data Validation:**

- You can set specific constraints on the data that gets into the Cloud storage.
- The constraints can be: filename, content type or size

```
service firebase.storage {  
  match /b/{bucket}/o {  
    match /images/{imageId} {  
      // Only allow uploads of any image file that's less than 5MB  
      allow write: if request.resource.size < 5 * 1024 * 1024  
        && request.resource.contentType.matches('image/.*');  
    }  
  }  
}
```

**Example:**  
Restriction on  
File size for  
uploads

# Firestore

- **Authentication** – manages user registration and authentication
- ~~Realtime database – Cloud hosted NoSQL Database~~
- **Storage** – storage management for various files(images, documents)
- Hosting – Global web hosting
- MLKit – SDKs for ML tasks



# User authentication system

# Firebase Authentication

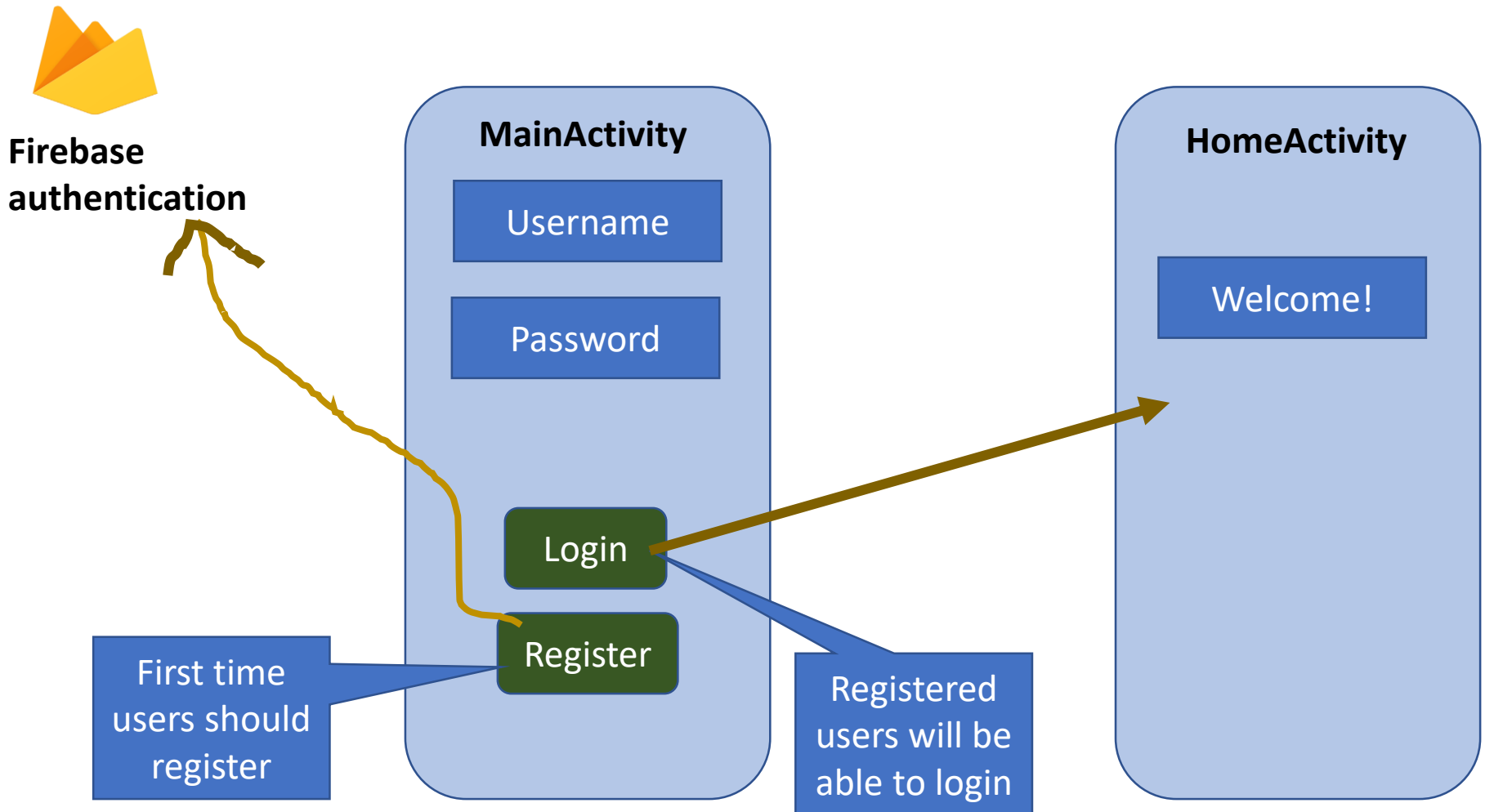
- Manages user registration
- Using Firebase authentication you register your users
  - Firebase Authentication provides a couple of options
    - Email authentication
    - Phone Authentication
    - Anonymous
    - Third-party
- All registered users will be authenticated to use your App:
  - You can still control the read/write functionalities

# Firestore User Authentication System

# User Authentication

- User Registration
- User Authentication

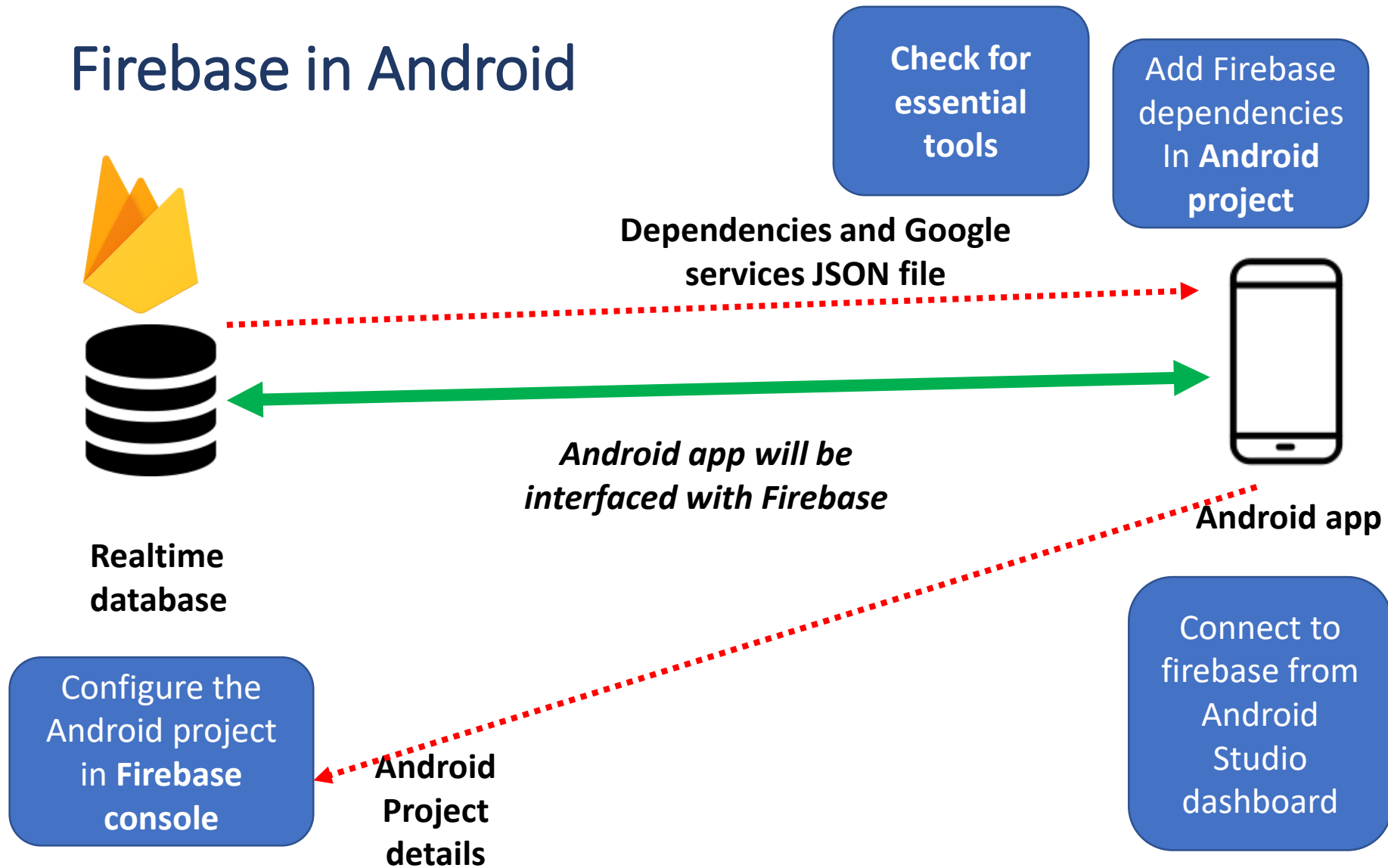
# Exercise 7 -Firebase user authentication system



# User registration

Let's create a new project and  
practice configuration

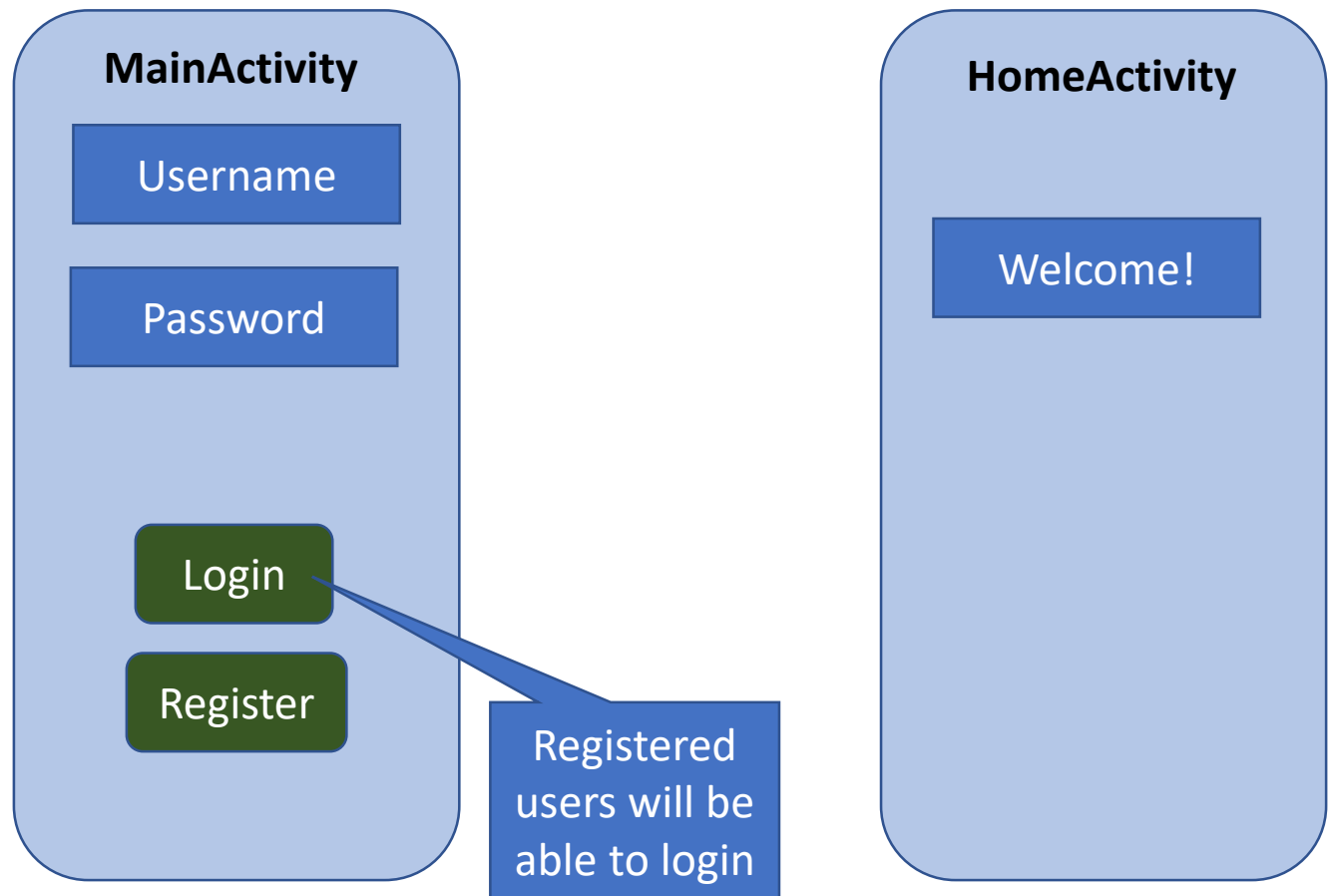
# Firestore in Android





# Layout creation

# Exercise 7 -Firebase user authentication system



# Configure Firebase Console for Authentication

# Steps for enabling user Authentication

- Part 1 – Firebase console
  - Enable Email Sign-in method from Authentication tab
    - Authentication -> Sign-in method -> Email -> Enable
- Part 2 – Android Studio
  - Include the dependencies :
    - Google Services
    - Firebase core
    - Firebase Authentication

```
classpath 'com.google.gms:google-services:4.2.0'
```

```
implementation 'com.google.firebase:firebase-core:17.0.0'
```

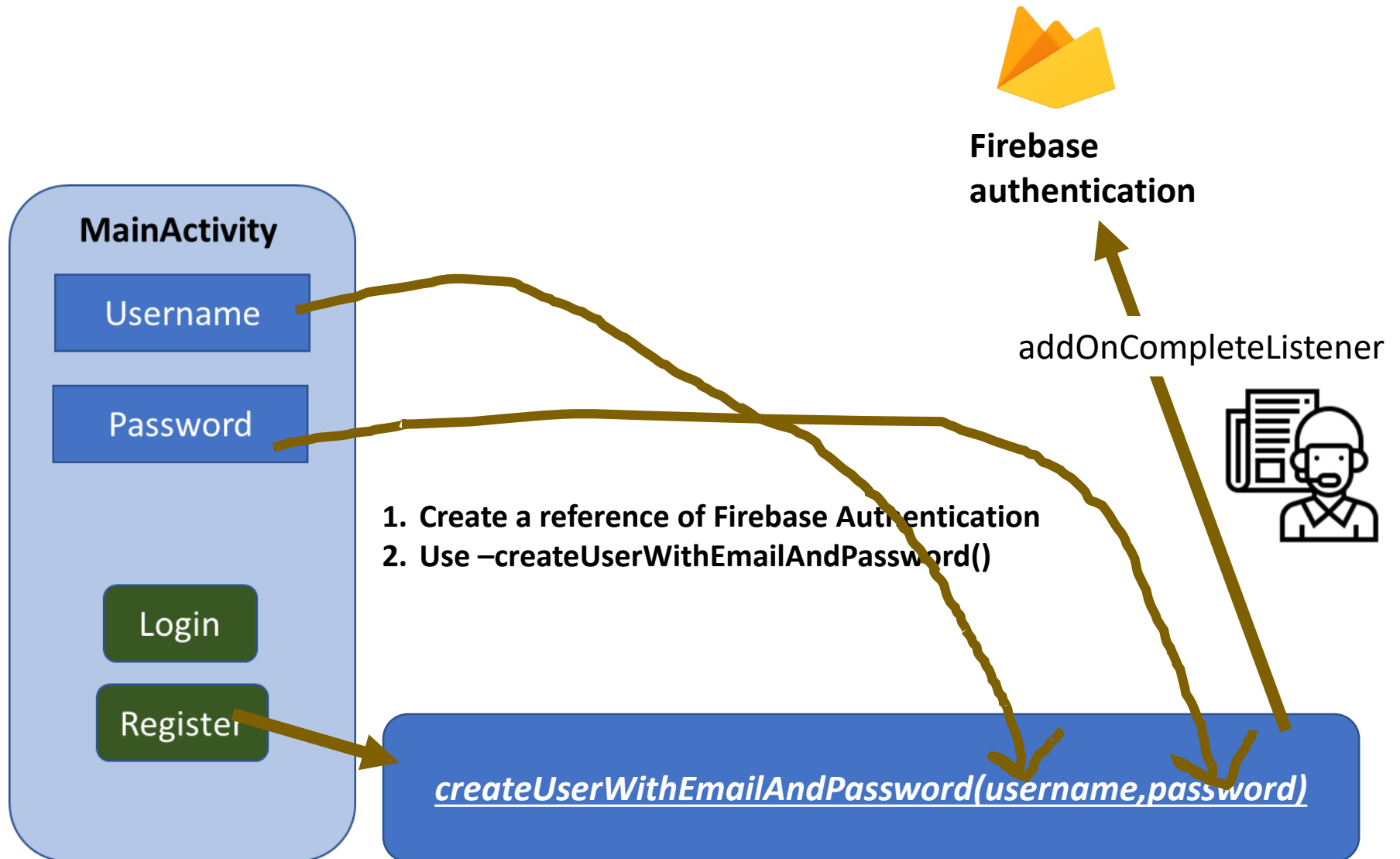
```
implementation 'com.google.firebase:firebase-auth:18.1.0'
```

```
apply plugin: 'com.google.gms.google-services'
```

# User Registration -Implementation

# Implementation of user registration system

- Grab the user data(username and password)
- **Create a Reference for Firebase authentication  
(FirebaseAuth.getInstance())**
- **Use -> createUserWithEmailAndPassword() to setup registration**
- **Set EventListeners to the task**
  - **onCompleteListener**
  - **onFailureListener**



# User Authentication