twitch.tv/dancojocar

youtube.com/dancojocar

Lecture #9
Firebase Services

Mobile Applications 2020-2021

twitch.tv/dancojocar

com/dancojocar

2007 Chat App 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020

https://startupandrew.com/posts/my-experiences-in-the-haitian-earthquake/ https://firebase.google.com/support/release-notes

twitch.tv/dancojocar

youtube.com/dancojocar

Add Firebase to Your Project

- Using the assistant, in Android Studio:

  1.Tools -> Firebase.

  2. Select the service.

  3. Connect to Firebase.

- Manually:
  - Create a project in console.firebase.google.com
  - Download the config file.
  - Add the SDK.

Add the SDK

Root-level build.gradle:

```
buildscript {
  // ...
  dependencies {
    // ...
    classpath ‘com.google.gms:google-services:4.3.4’ // google-services plugin
  }
}

allprojects {
  // ...
  repositories {
    // ...
    google() // Google's Maven repository
  }
}
```

Available Libraries

fi firebase.inappmessaging.display

firebase.perf

com.google.firebase.perf

com.google.firebase.perf.metrics

firebase.remoteconfig

com.google.firebase.remoteconfig

firebase.storage

com.google.firebase.storage

Inter-operational packages

com.google.firebase.auth.internal

com.google.firebase.ml.naturallanguage.languageid

com.google.firebase.ml.naturallanguage.smartreply

firebase.google.com/docs/reference/android/packages

twitch.tv/dancojocar

youtube.com/dancojocar

Realtime Database

- Realtime.
- Offline.
- Collaborate across devices with ease.
- Scale across multiple regions.

twitch.tv/dancojocar

youtube.com/dancojocar

Installation & Setup

Add the Realtime Database to your app

Secure Your Data

```
implementation 'com.google.firebase:firebase-database-ktx:19.5.1'
```

```
{{
  "rules": {
    "messages": {
      "message0": {
        "content": "Hello",
        "timestamp": 1405704370369
      },
      "message1": {
        "content": "Goodbye",
        "timestamp": 1405704395231
      },
      ...}
    }
  }
}
```

```
{
  "rules": {
    "messages": {
      ".read": "data.child('timestamp').val() > (now - 600000)",
      // new messages must have a string content and a number timestamp
      ".validate": "newData.hasChildren(['content', 'timestamp'])
                    && newData.child('content').isString()
                    && newData.child('timestamp').isNumber()"
    }
  }
}
```

// only the last ten minutes can be read

https://firebase.google.com/docs/database/security/securing-data

twitch.tv/dancojocar

youtube.com/dancojocar

Data Access

Read from your database

Write to your database

```
// Read from the database
myRef.addValueEventListener(object : ValueEventListener {
    override fun onDataChange(dataSnapshot: DataSnapshot) {
        // This method is called once with the initial value and again
        // whenever data at this location is updated.
        val value = dataSnapshot.getValue(String::class.java)
        Log.d(TAG, "Value is: $value")
    }

    override fun onCancelled(error: DatabaseError) {
        // Failed to read value
        Log.w(TAG, "Failed to read value.", error.toException())
    }
})
```

```
// Write a message to the database
val database = FirebaseDatabase.getInstance()
val myRef = database.getReference("message")

myRef.setValue("Hello, World!")
```

https://firebase.google.com/docs/database/android/read-and-write

twitch.tv/dancojocar

youtube.com/dancojocar

Update Data

```
private fun writeNewPost(
    userId: String,
    username: String,
    title: String,
    body: String
  ){
  // Create new post at /user-posts/$userid/$postid and at
  // /posts/$postid simultaneously
  val key = database.child("posts").push().key
  if (key == null) {
    Log.w(TAG, "Couldn't get push key for posts")
    return
  }

  val post = Post(userId, username, title, body)
  val postValues = post.toMap()

  val childUpdates = HashMap<String, Any>()
  childUpdates["/posts/$key"] = postValues
  childUpdates["/user-posts/$userId/$key"] = postValues

  database.updateChildren(childUpdates)
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

Using Transactions

```
private fun onStarClicked(postRef: DatabaseReference) {
  postRef.runTransaction(object : Transaction.Handler {
    override fun doTransaction(mutableData: MutableData): Transaction.Result {
      val p = mutableData.getValue(Post::class.java)
          ?: return Transaction.success(mutableData)
      if (p.stars.containsKey(uid)) {
        // Unstar the post and remove self from stars
        p.starCount = p.starCount - 1
        p.stars.remove(uid)
      } else {
        // Star the post and add self to stars
        p.starCount = p.starCount + 1
        p.stars[uid] = true
      }
      // Set value and report transaction success
      mutableData.value = p
      return Transaction.success(mutableData)
    }
    override fun onComplete(
      databaseError: DatabaseError?,
      b: Boolean,
      dataSnapshot: DataSnapshot?
    ) {
      // Transaction completed
      Log.d(TAG, "postTransaction:onComplete:" + databaseError!!)
```

Enabling Offline Capabilities

twitch.tv/dancojocar

youtube.com/dancojocar

Detecting Connection State

Keeping Data Fresh

```
val connectedRef = FirebaseDatabase.getInstance().getReference(".info/connected")
connectedRef.addValueEventListener(object : ValueEventListener {
  override fun onDataChange(snapshot: DataSnapshot) {
    val connected = snapshot.getValue(Boolean::class.java) ?: false
    if (connected) {
      Log.d(TAG, "connected")
    } else {
      Log.d(TAG, "not connected")
    }
  }

  override fun onCancelled(error: DatabaseError) {
    Log.w(TAG, "Listener was cancelled")
  }
})
```

```
FirebaseDatabase.getInstance().setPersistenceEnabled(true);
scoresRef.keepSynced(true);
scoresRef.keepSynced(false);
```

https://firebase.google.com/docs/database/android/offline-capabilities

twitch.tv/dancojocar

youtube.com/dancojocar

Authentication

Key capabilities:

- Email and password based authentication.
- Federated identity provider integration:
  - Google, Facebook, Twitter, Github.
- Phone number authentication.
- Custom auth system integration.
- Anonymous auth.

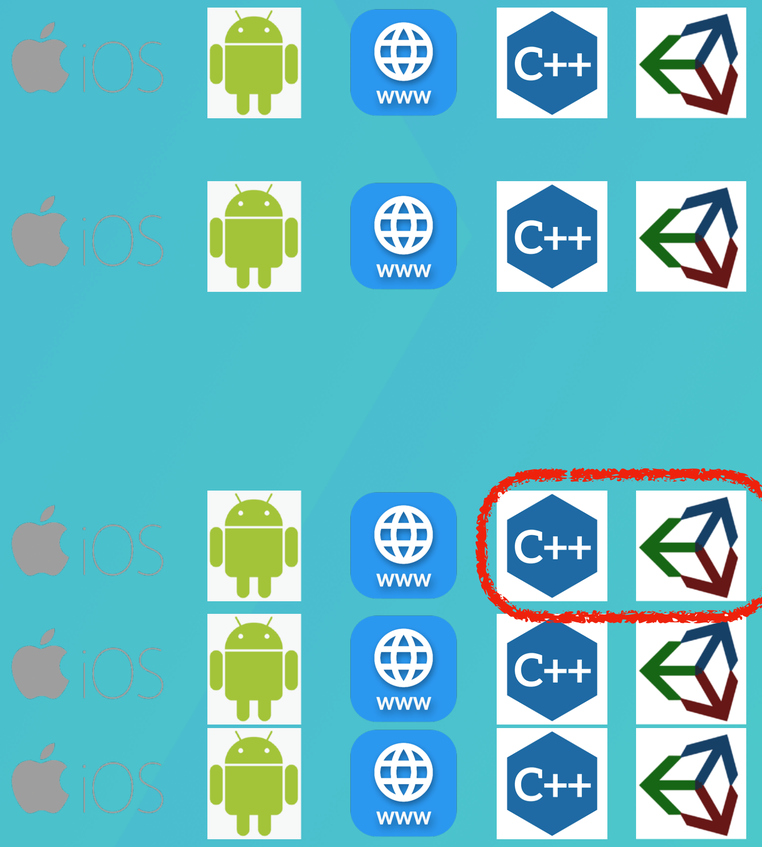
twitch.tv/dancojocar

youtube.com/dancojocar

Authentication

Key capabilities:

- Email and password based authentication.
- Federated identity provider integration:
  - Google, Facebook, Twitter, Github, Apple.
- Phone number authentication.
- Custom auth system integration.
- Anonymous auth.



Authenticate Using Google Sign-In

- Dependencies
- Integrate Google Sign-In
- Use shared auth
- Use the token

twitch.tv/dancojocar

youtube.com/dancojocar

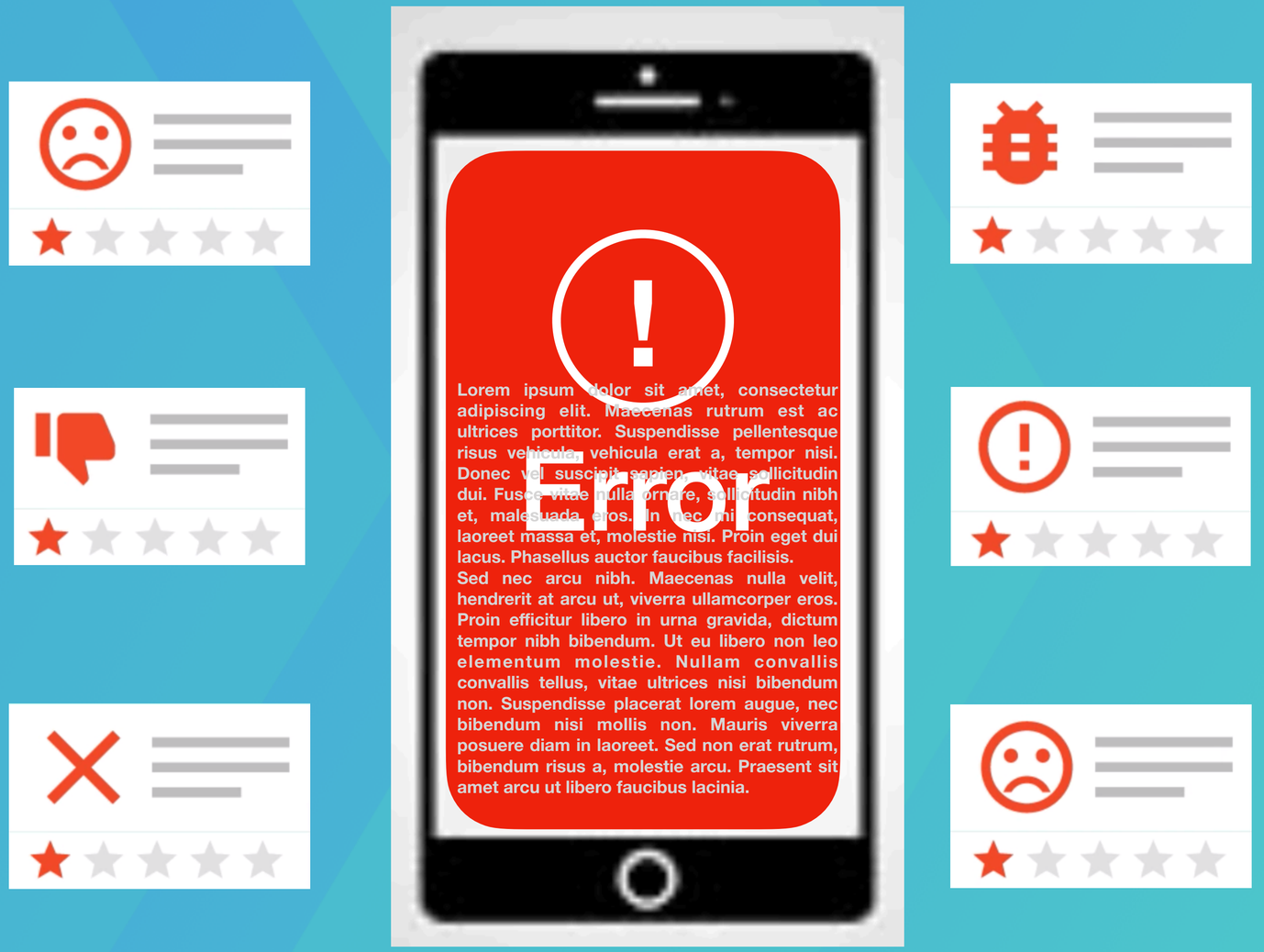
Sign out a User

DEMO

```
FirebaseAuth.getInstance().signOut()
```

twitch.tv/dancojocar

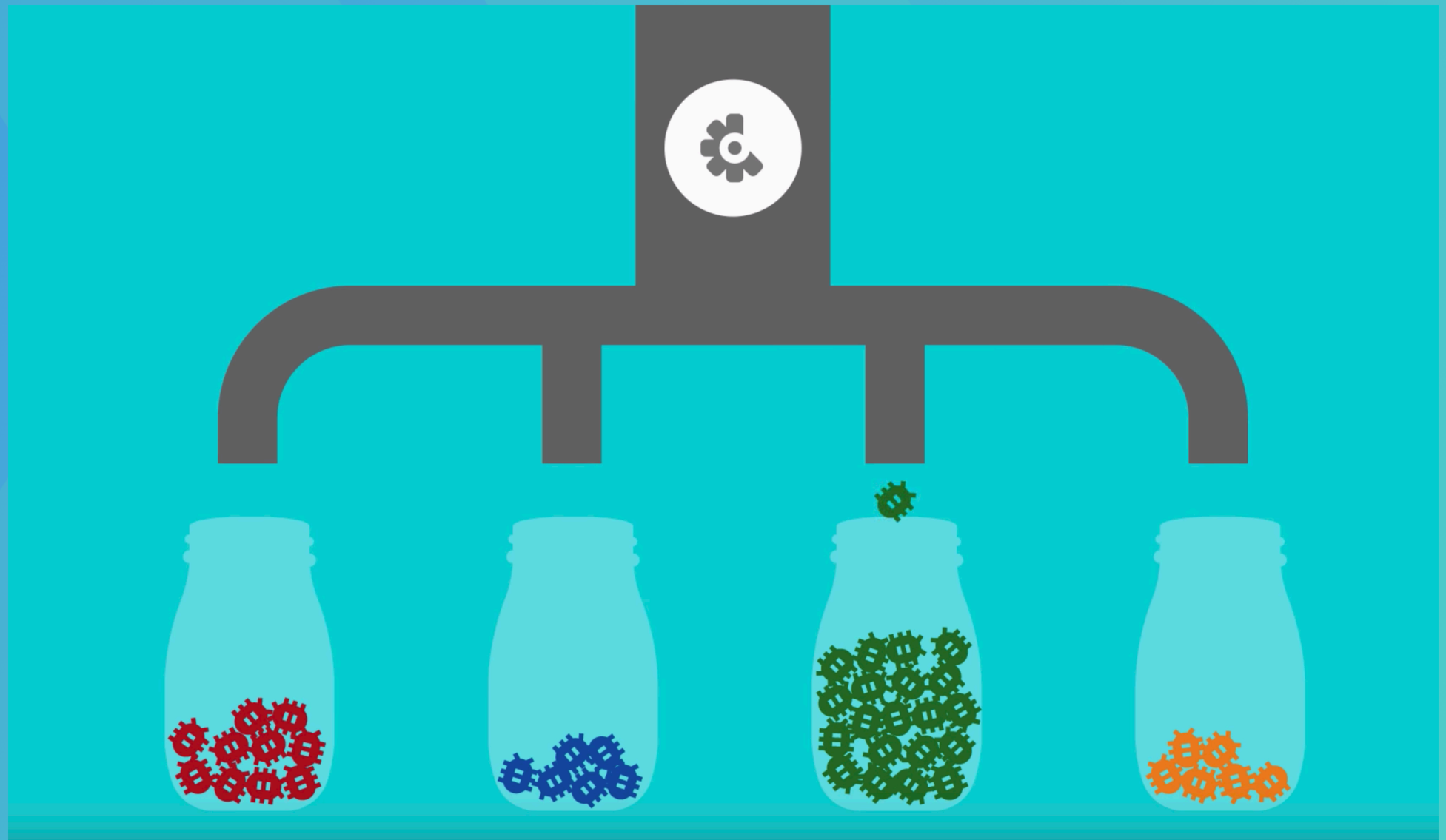
youtube.com/dancojocar



twitch.tv/dancojocar

youtube.com/dancojocar

Crashlytics



Enable the SDK

twitch.tv/dancojocar

youtube.com/dancojocar

```
buildscript {
    repositories {
        // Add the following repositories:
        google()  // Google's Maven repository
    }
    dependencies {
        // ...
        classpath 'com.google.gms:google-services:4.3.4'  // Google Services plugin
        // Add dependency
        classpath 'com.google.firebase:firebase-crashlytics-gradle:2.4.1'
         // Crashlytics plugin
    }
}
allprojects {
    repositories {
        // Check that you have the following line (if not, add it):
        google()  // Google's Maven repository
    }
}
```

Enable the SDK


```
apply plugin: 'com.android.application'
// Google Services Gradle plugin
apply plugin: 'com.google.gms.google-services'

// Apply the Crashlytics Gradle plugin
apply plugin: 'com.google.firebase.crashlytics'

dependencies {
  // …
  // Import the BoM for the Firebase platform
  implementation platform('com.google.firebase:firebase-bom:26.1.1')

  // Declare the dependencies for the Crashlytics and Analytics libraries
  // When using the BoM, you don't specify versions in Firebase library dependencies
  implementation 'com.google.firebase:firebase-crashlytics-ktx'
  implementation 'com.google.firebase:firebase-analytics-ktx'
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

Test Implementation

```
val crashButton = Button(this)
crashButton.text = "Crash!"
crashButton.setOnClickListener {
    throw RuntimeException("Test Crash") // Force a crash
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

DEMO

Customize

AndroidManifest.xml

```
<meta-data
    android:name="firebase_crashlytics_collection_enabled"
    android:value="false" />
```

Enable collection for selected users:

```
val crashlytics = FirebaseCrashlytics.getInstance()
Crashlytics.setUserIdentifier("myAppUserId")
```

Crash reports and log only:

```
Crashlytics.log(Log.DEBUG, "message", "message")
```

Add custom keys:

```
Crashlytics.setInt(key, 1 /* int value */)
Crashlytics.setBool(key, true /* boolean value */)
Crashlytics.setFloat(key, 1.0f /* float value */)
Crashlytics.setDouble(key, 1.0 /* double value */)
Crashlytics.setString(key, "foo" /* string value */)
Crashlytics.setString("key_UI_action", "clicked the dragon!")
```

Set user IDs

```
Crashlytics.setUserIdentifier("user123456789")
```

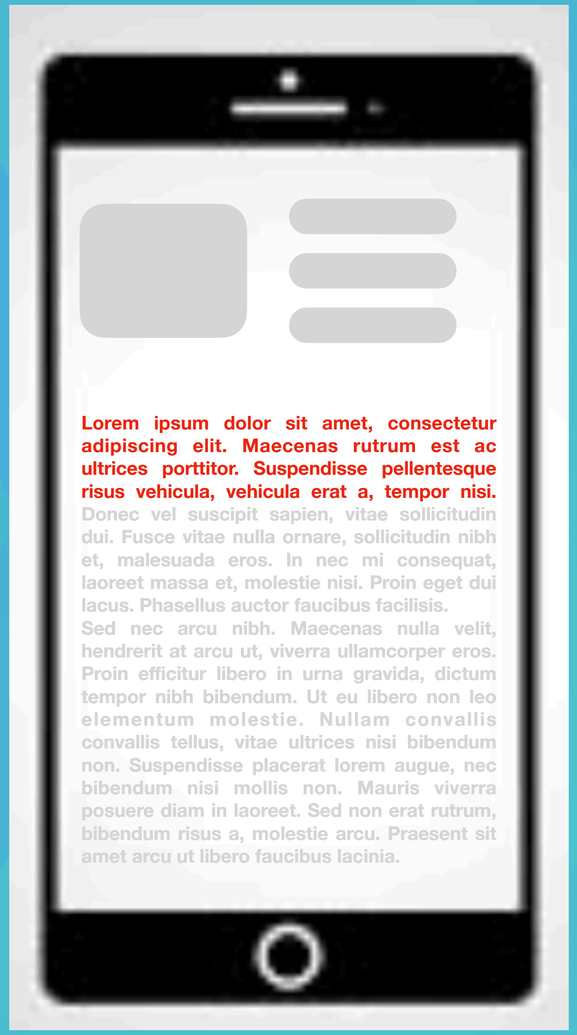
Log non-fatal exceptions

```
Crashlytics.logException(e)
```

https://firebase.google.com/docs/crashlytics/customize-crash-reports

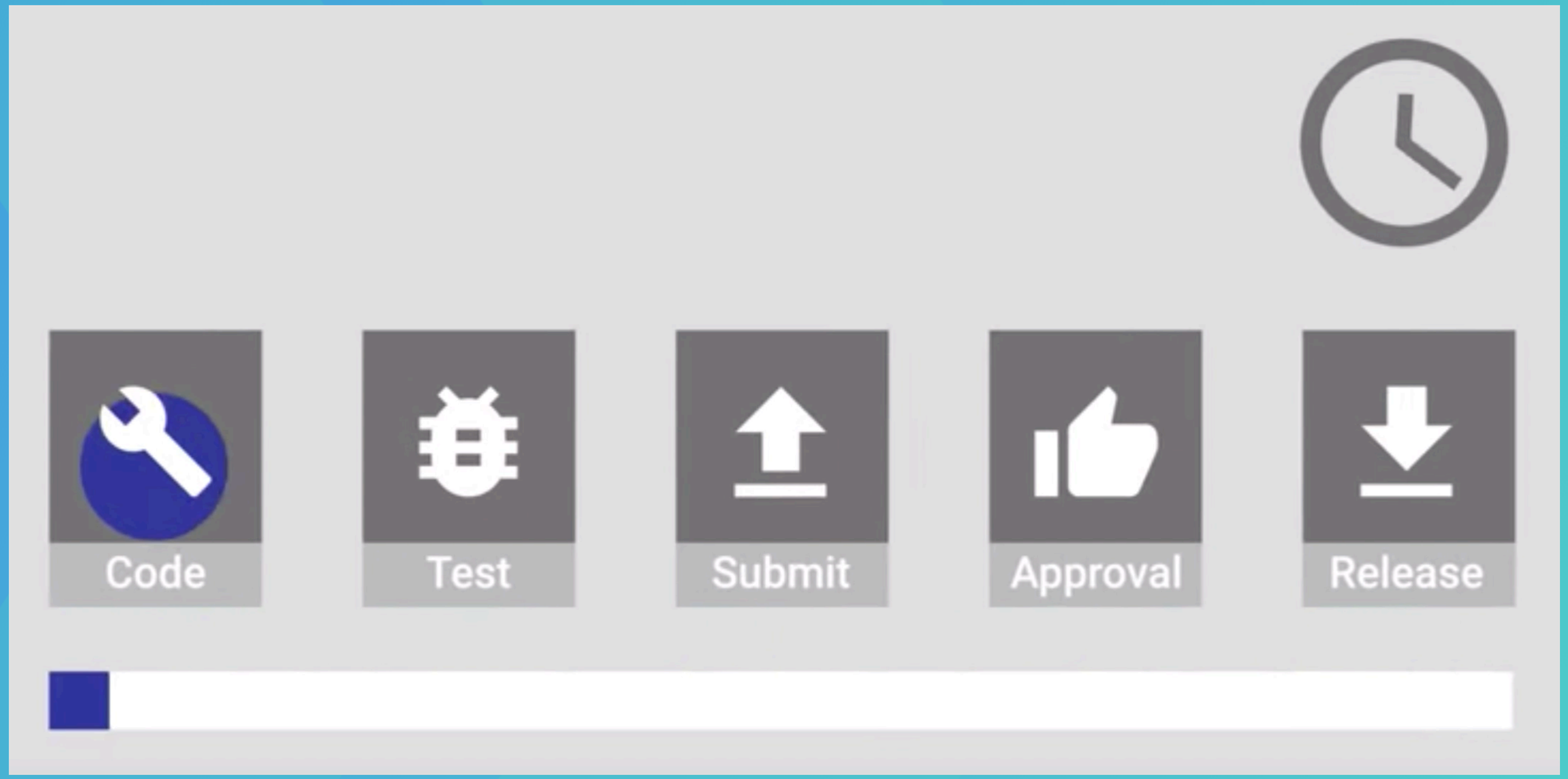
twitch.tv/dancojocar

youtube.com/dancojocar



twitch.tv/dancojocar

youtube.com/dancojocar



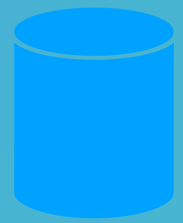
twitch.tv/dancojocar

youtube.com/dancojocar

twitch.tv/dancojocar

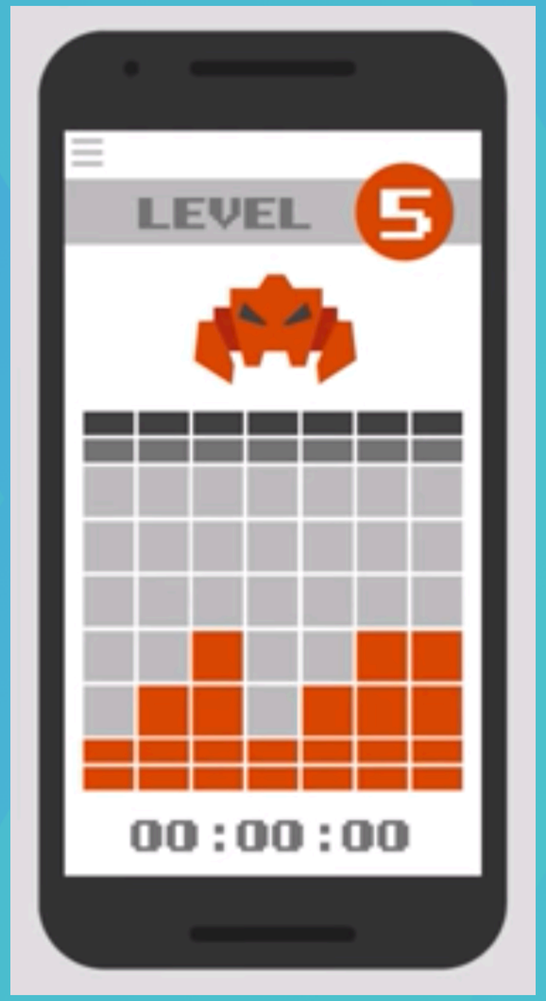
youtube.com/dancojocar

Realtime



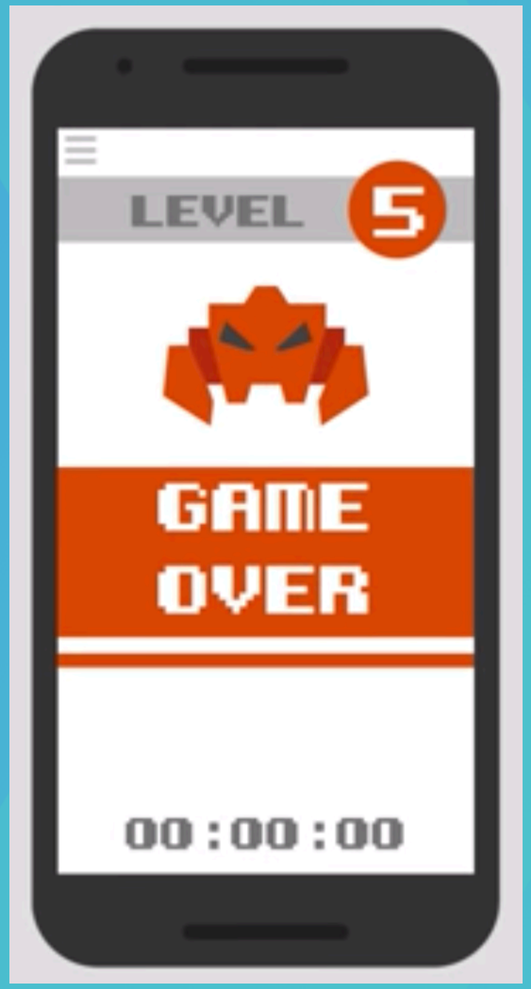
twitch.tv/dancojocar

youtube.com/dancojocar



twitch.tv/dancojocar

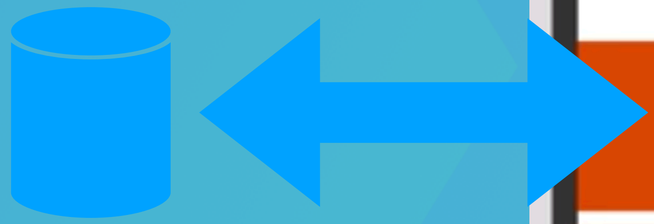
youtube.com/dancojocar

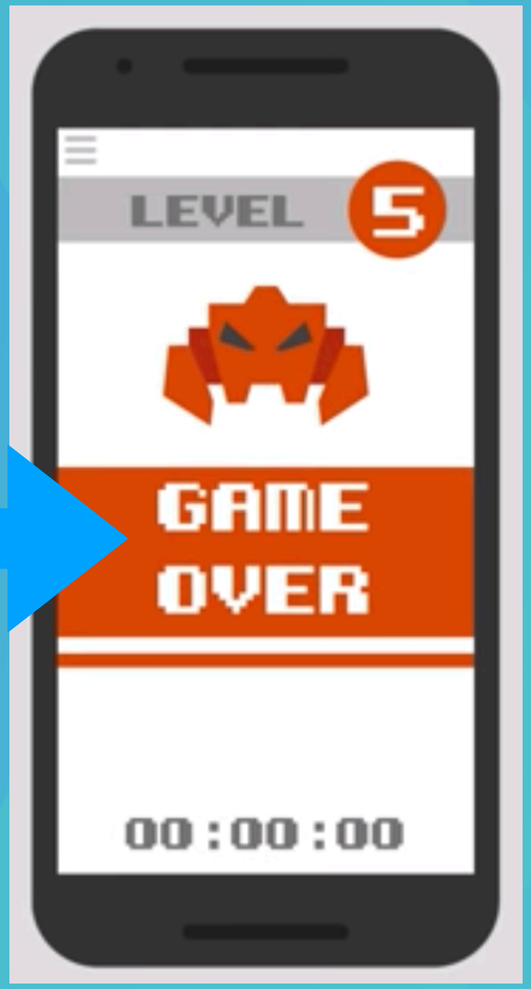


twitch.tv/dancojocar

youtube.com/dancojocar

Realtime

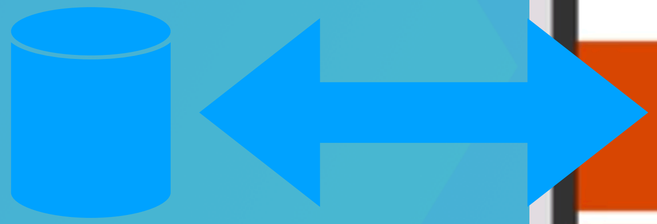


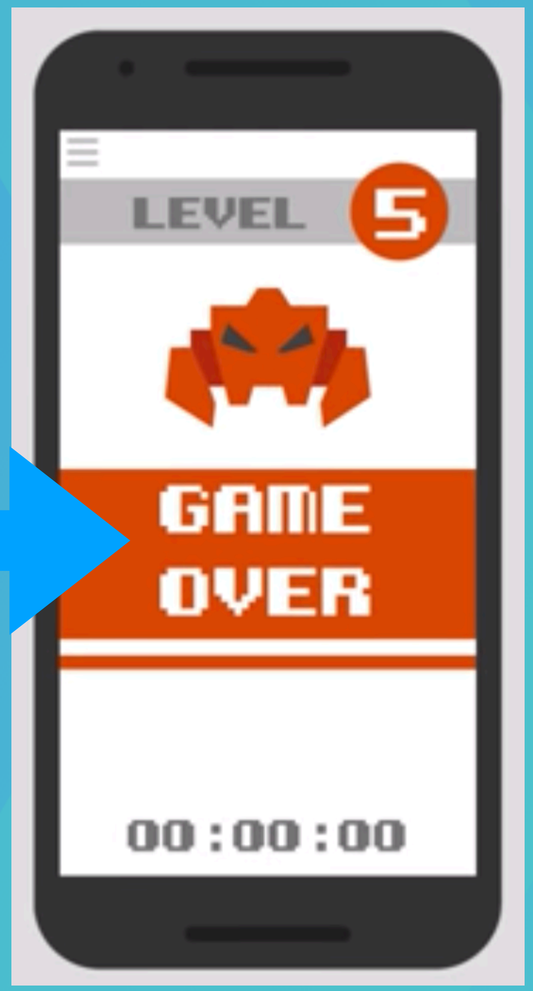


twitch.tv/dancojocar

youtube.com/dancojocar

Realtime



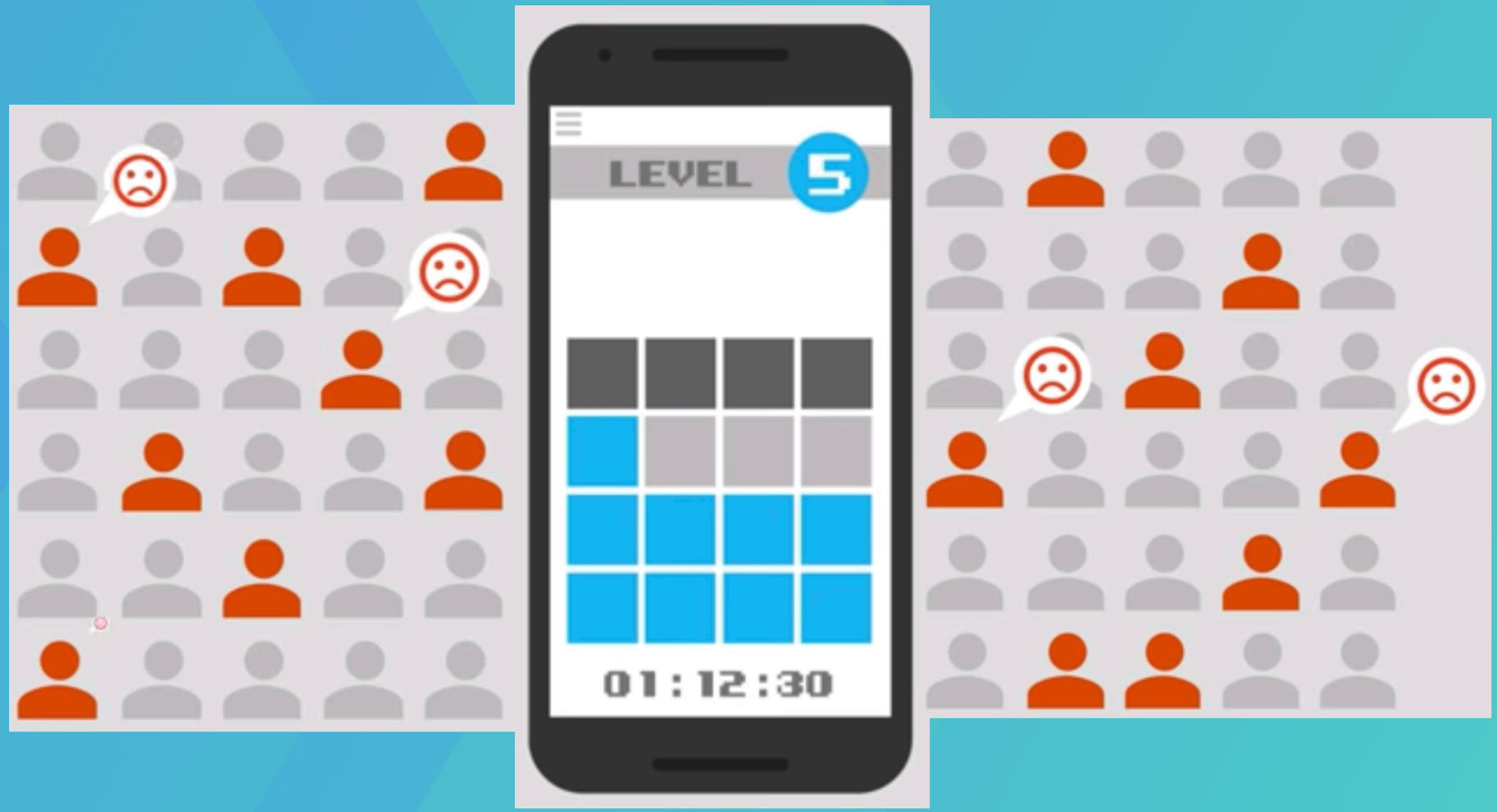


twitch.tv/dancojocar

youtube.com/dancojocar

twitch.tv/dancojocar

youtube.com/dancojocar



Remote Config

Using Remote Config

twitch.tv/dancojocar

youtube.com/dancojocar

1. Add Firebase to your app

```
implementation 'com.google.firebase:firebase-config-ktx'
```

2. Get the Remote Config singleton object

```
FirebaseRemoteConfig.getInstance()
```

5. Set parameter values in the service (as needed)

6. Fetch and activate values from the service (as needed)

https://firebase.google.com/docs/remote-config/use-config-android

Make $

- In-app purchases.
- Subscriptions.
- Advertising.
- Paid apps.
- E-commerce.

https://developer.android.com/distribute/best-practices/earn/monetization-options

twitch.tv/dancojocar

youtube.com/dancojocar

AdMob

Update your AndroidManifest.xml

Add your AdMob app ID to the SDK

```
implementation 'com.google.firebase:firebase-ads:19.6.0'
```

```
<manifest>
  <application>
    <!-- Sample AdMob App ID: ca-app-pub-3940256099942544~3347511713 -->
    <meta-data
      android:name="com.google.android.gms.ads.APPLICATION_ID"
      android:value="[ADMOB_APP_ID]"/>
  </application>
</manifest>
```

Initialize the SDK

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    // Sample AdMob app ID: ca-app-pub-3940256099942544~3347511713
    MobileAds.initialize(this, "YOUR_ADMOB_APP_ID")
}
```

twitch.tv/dancojocar

youtube.com/dancojocar

Ad Formats

DEMO

- Banner
- Interstitial
- Native
- RewardedVideo

Google Play Billing Overview

twitch.tv/dancojocar

youtube.com/dancojocar

DEMO

- Types of in-app products:
  - One-time products.
  - Subscriptions.
- In-app product configuration options:
  - Title.
  - Description.
  - Product ID.
  - Price / Default Price.

https://developer.android.com/google/play/billing/billing_overview

twitch.tv/dancojocar

youtube.com/dancojocar

Lecture outcomes

- Use Firebase Realtime Database.
- How to use Remove Config with A-B testing.
- Authenticate your users using Firebase, with GoogleSignIn and EmailPassword methods.
- How to use Firebase Realtime Database.
- Using AdMob to display ads.