Lecture #11
Firebase & Apps Monetization

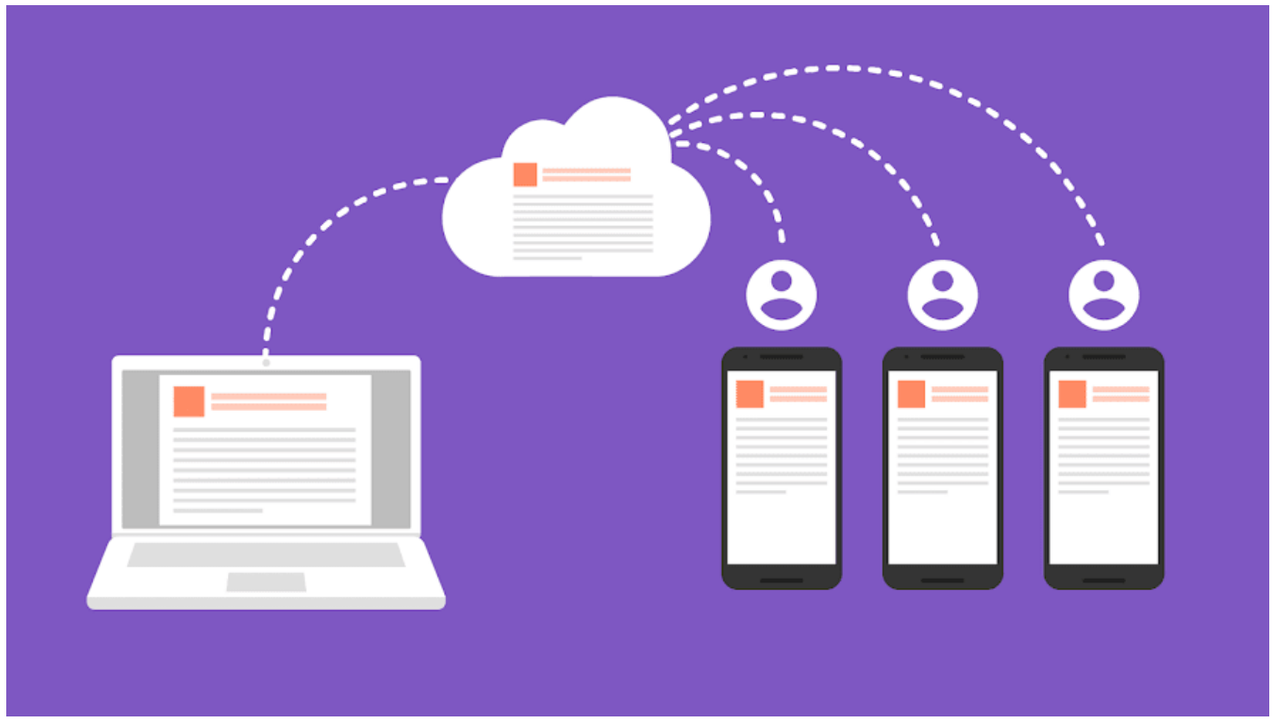
Mobile Applications 2018-2019

Available Libraries

Gradle Dependency Line	Service
com.google.firebase:firebase-core:16.0.6	Analytics
com.google.firebase:firebase-database:16.0.5	Realtime Database
com.google.firebase:firebase-firestore:17.1.4	Cloud Firestore
com.google.firebase:firebase-storage:16.0.5	Storage
com.crashlytics.sdk.android:crashlytics:2.9.7	Crashlytics
com.google.firebase:firebase-auth:16.1.0	Authentication
com.google.firebase:firebase-messaging:17.3.4	Cloud Messaging
com.google.firebase:firebase-config:16.1.2	Remote Config
com.google.firebase:firebase-invites:16.0.6	Invites and Dynamic Links
com.google.firebase:firebase-ads:17.1.2	AdMob
com.google.firebase:firebase-appindexing:17.1.0	App Indexing
com.google.firebase:firebase-perf:16.2.2	Performance Monitoring
com.google.firebase:firebase-functions:16.1.3	Cloud Functions for Firebase Client SDK
com.google.firebase:firebase-ml-vision:18.0.2	ML Kit (Vision)
com.google.firebase:firebase-ml-model-interpreter:16.2.4	ML Kit (Custom Model)

Realtime Database

- Realtime.
- Offline.
- Accessible from Client Devices.
- Scale across multiple databases.



Installation & Setup

Add the Realtime Database to your app

```
implementation 'com.google.firebase:firebase-database:16.0.5'
```

Installation & Setup

Add the Realtime Database to your app

```
implementation 'com.google.firebase:firebase-database:16.0.5'
```

Secure Your Data

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

Installation & Setup

Add the Realtime Database to your app

```
implementation 'com.google.firebase:firebase-database:16.0.5'
```

Secure Your Data

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

```
{
  "rules": {
    "messages": {
      "$message": {
        // only messages from the last ten minutes can be read
        ".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()"
      }
    }
  }
}
```

Installation & Setup

Add the Realtime Database to your app

```
implementation 'com.google.firebase:firebase-database:16.0.5'
```

Secure Your Data

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

```
{
  "rules": {
    "messages": {
      "$message": {
        // only messages from the last ten minutes can be read
        ".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()"
      }
    }
  }
}
```

Installation & Setup

Add the Realtime Database to your app

```
implementation 'com.google.firebase:firebase-database:16.0.5'
```

Secure Your Data

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

```
{
  "rules": {
    "messages": {
      "$message": {
        // only messages from the last ten minutes can be read
        ".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()"
      }
    }
  }
}
```

Installation & Setup

Add the Realtime Database to your app

```
implementation 'com.google.firebase:firebase-database:16.0.5'
```

Secure Your Data

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

```
{
  "rules": {
    "messages": {
      "$message": {
        // only messages from the last ten minutes can be read
        ".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()"
      }
    }
  }
}
```

Installation & Setup

Add the Realtime Database to your app

```
implementation 'com.google.firebase:firebase-database:16.0.5'
```

Secure Your Data

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

```
{
  "rules": {
    "messages": {
      "$message": {
        // only messages from the last ten minutes can be read
        ".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()"
      }
    }
  }
}
```

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

Data Access

Write to your database

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

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

Data Access

Write to your database

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

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

Read from 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())
    }
})
```

Data Access

Write to your database

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

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

Read from 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())
    }
})
```

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

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)
}
```

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!!)
```

```
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

```
FirebaseDatabase.getInstance().setPersistenceEnabled(true);
```

Enabling Offline Capabilities

```
FirebaseDatabase.getInstance().setPersistenceEnabled(true);
```

Keeping Data Fresh

```
DatabaseReference scoresRef = FirebaseDatabase.getInstance()
    .getReference("scores");
scoresRef.keepSynced(true);
```

Enabling Offline Capabilities

```
FirebaseDatabase.getInstance().setPersistenceEnabled(true);
```

Keeping Data Fresh

```
DatabaseReference scoresRef = FirebaseDatabase.getInstance()
    .getReference("scores");
scoresRef.keepSynced(true);

scoresRef.keepSynced(false);
```

DEMO

Enabling Offline Capabilities

```
FirebaseDatabase.getInstance().setPersistenceEnabled(true);
```

Keeping Data Fresh

```
DatabaseReference scoresRef = FirebaseDatabase.getInstance()
    .getReference("scores");
scoresRef.keepSynced(true);

scoresRef.keepSynced(false);
```

Detecting Connection State

```
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")
  }
})
```

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

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.

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.

Authenticate Using Google Sign-In

Authenticate Using Google Sign-In

- Dependencies

```
implementation 'com.google.firebase:firebase-auth:16.0.5'
implementation 'com.google.android.gms:play-services-auth:16.0.1'
```

Authenticate Using Google Sign-In

- Dependencies
- Integrate Google Sign-In

```
// Configure Google Sign In
val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
        .requestIdToken(getString(R.string.default_web_client_id))
        .requestEmail()
        .build()
```

Authenticate Using Google Sign-In

- Dependencies
- Integrate Google Sign-In

```
// Configure Google Sign In
val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
        .requestIdToken(getString(R.string.default_web_client_id))
        .requestEmail()
        .build()

private fun signIn() {
    val signInIntent = gso.signInIntent
    startActivityForResult(signInIntent, RC_SIGN_IN)
}
```

Authenticate Using Google Sign-In

- Dependencies

- Integrate Google Sign-In

```
public override fun onActivityResult(requestCode: Int, resultCode: Int,
      data: Intent?) {
  super.onActivityResult(requestCode, resultCode, data)

  // Result returned from launching the Intent from
  // GoogleSignInApi.getSignInIntent(...);
  if (requestCode == RC_SIGN_IN) {
    val task = GoogleSignIn.getSignedInAccountFromIntent(data)
    try {
      // Google Sign In was successful, authenticate with Firebase
      val account = task.getResult(ApiException::class.java)
      firebaseAuthWithGoogle(account!!)
    } catch (e: ApiException) {
      // Google Sign In failed, update UI appropriately
      Log.w(TAG, "Google sign in failed", e)
      // ...
    }
  }
}
```

Authenticate Using Google Sign-In

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

```
private lateinit var auth: FirebaseAuth
// ...
// Initialize Firebase Auth
auth = FirebaseAuth.getInstance()

// ...

public override fun onStart() {
    super.onStart()
    // Check if user is signed in (non-null) and update UI accordingly.
    val currentUser = auth.currentUser
    updateUI(currentUser)
}
```

Authenticate Using Google Sign-In

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

```
private fun firebaseAuthWithGoogle(acct: GoogleSignInAccount) {
  Log.d(TAG, "firebaseAuthWithGoogle:" + acct.id!!)

  val credential = GoogleAuthProvider.getCredential(acct.idToken, null)
  auth.signInWithCredential(credential)
    .addOnCompleteListener(this) { task ->
      if (task.isSuccessful) {
        // Sign in success, update UI with the signed-in user's information
        Log.d(TAG, "signInWithCredential:success")
        val user = auth.currentUser
        updateUI(user)
      } else {
        // If sign in fails, display a message to the user.
        Log.w(TAG, "signInWithCredential:failure", task.exception)
```

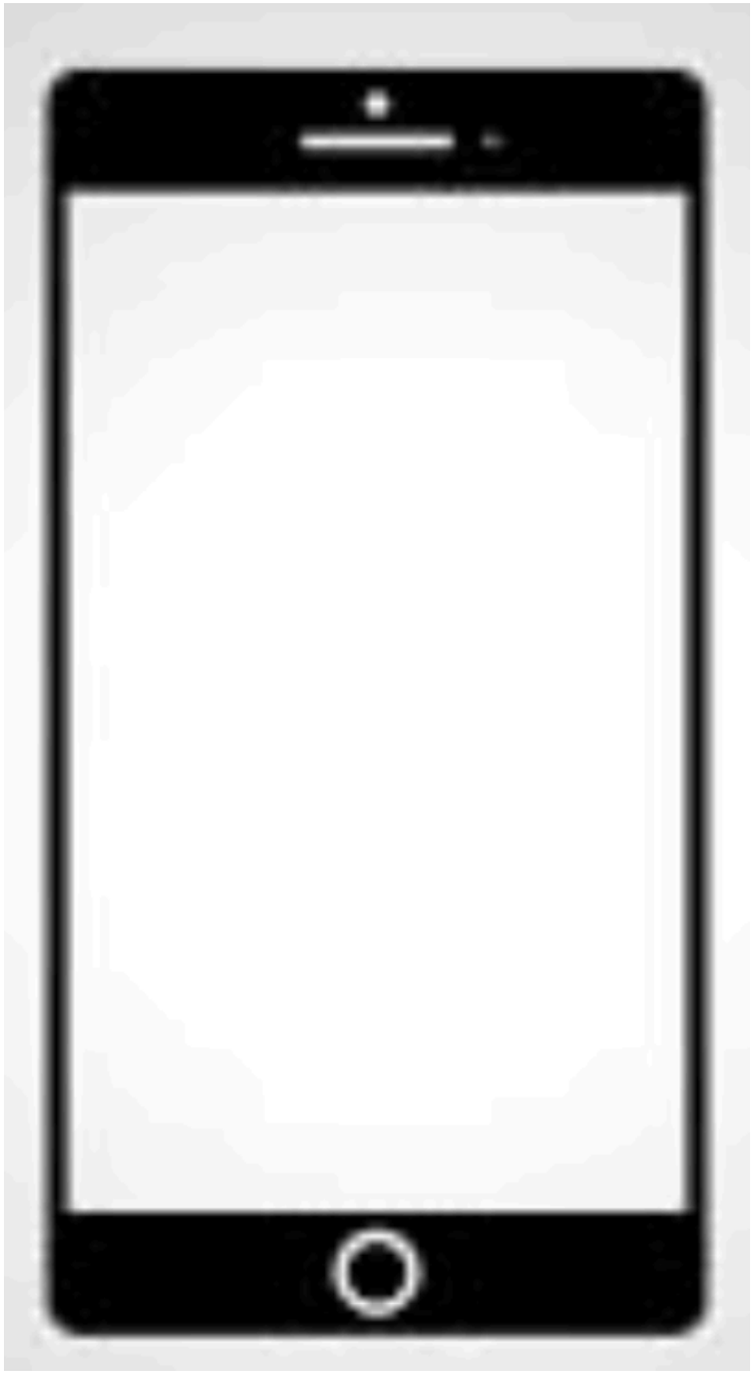
Authenticate Using Google Sign-In

```
private fun firebaseAuthWithGoogle(acct: GoogleSignInAccount) {
  Log.d(TAG, "firebaseAuthWithGoogle:" + acct.id!!)

  val credential = GoogleAuthProvider.getCredential(acct.idToken, null)
  auth.signInWithCredential(credential)
    .addOnCompleteListener(this) { task ->
      if (task.isSuccessful) {
        // Sign in success, update UI with the signed-in user's information
        Log.d(TAG, "signInWithCredential:success")
        val user = auth.currentUser
        updateUI(user)
      } else {
        // If sign in fails, display a message to the user.
        Log.w(TAG, "signInWithCredential:failure", task.exception)
        Snackbar.make(main_layout, "Authentication Failed.",
Snackbar.LENGTH_SHORT).show()
        updateUI(null)
      }

  // ...
  }
}
```

DEMO

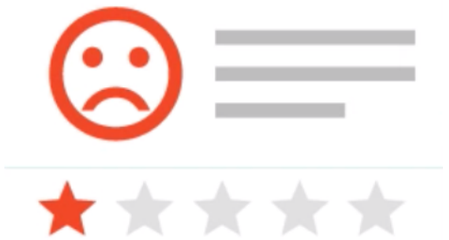
Sign out a User

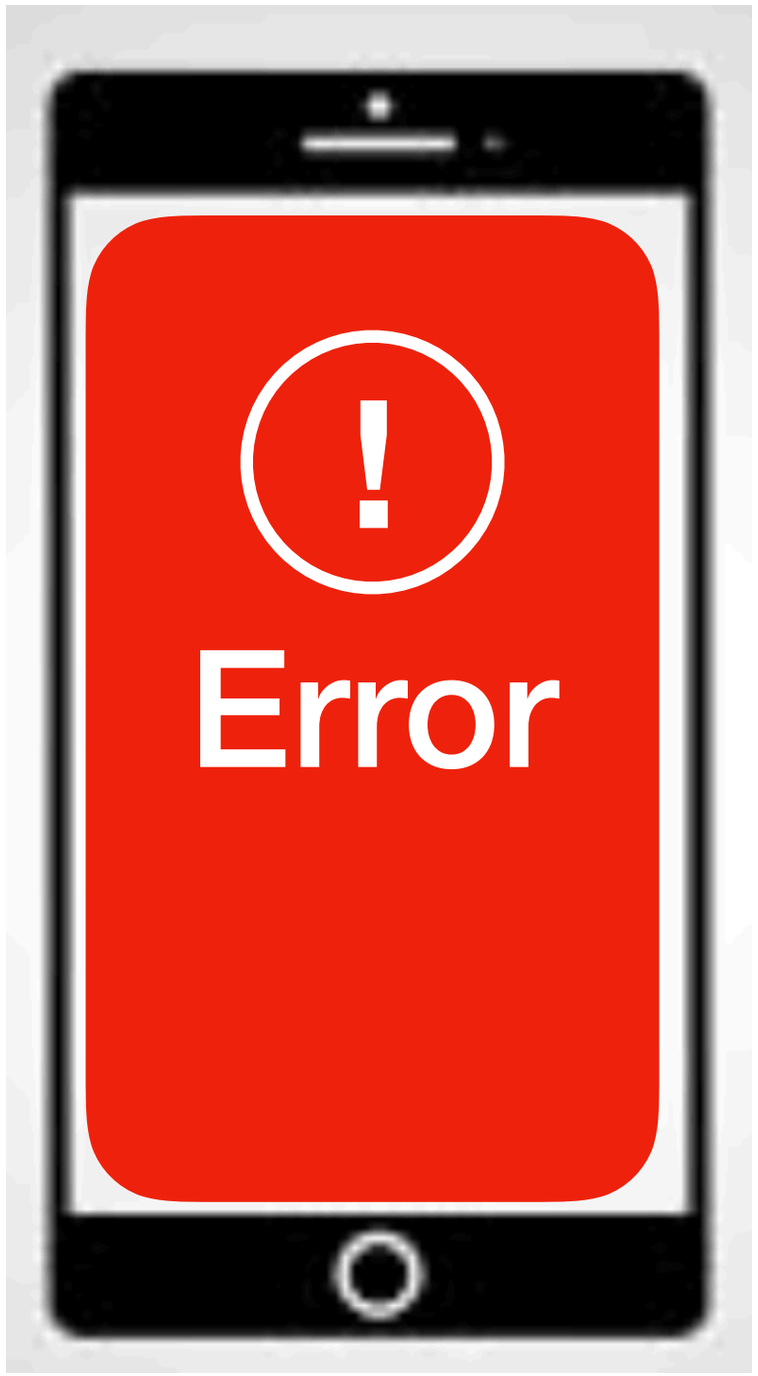
```
FirebaseAuth.getInstance().signOut()
```

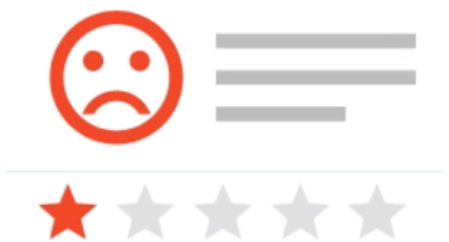


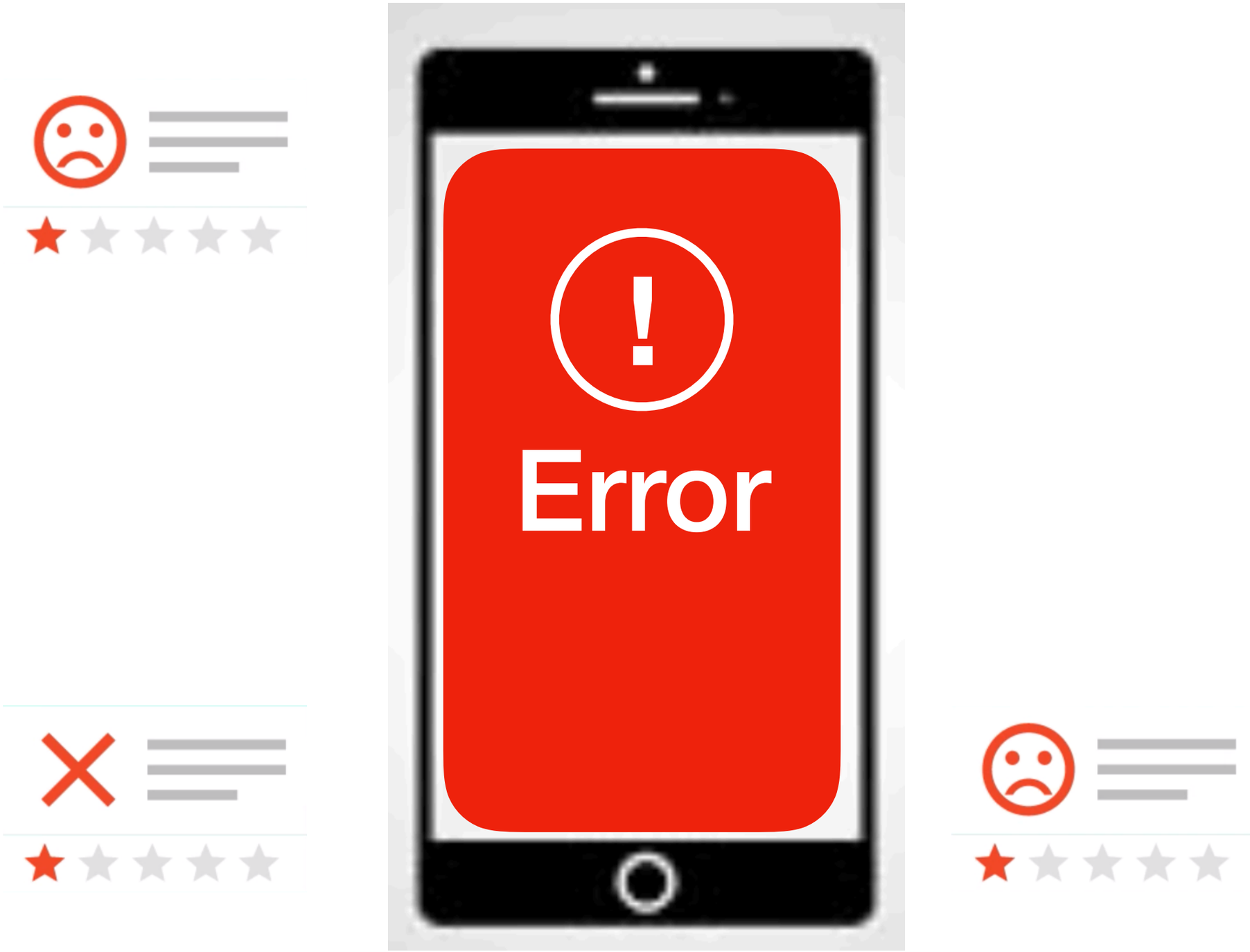

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas rutrum est ac ultrices porttitor. Suspendisse pellentesque risus vehicula, vehicula erat a, tempor nisi. Donec vel suscipit sapien, vitae sollicitudin dui. Fusce vitae nulla ornare, sollicitudin nibh et, malesuada eros. In nec mi consequat, laoreet massa et, molestie nisi. Proin eget dui lacus. Phasellus auctor faucibus facilisis.
Sed nec arcu nibh. Maecenas nulla velit, hendrerit at arcu ut, viverra ullamcorper eros. Proin efficitur libero in urna gravida, dictum tempor nibh bibendum. Ut eu libero non leo elementum molestie. Nullam convallis convallis tellus, vitae ultrices nisi bibendum non. Suspendisse placerat lorem augue, nec bibendum nisi mollis non. Mauris viverra posuere diam in laoreet. Sed non erat rutrum, bibendum risus a, molestie arcu. Praesent sit amet arcu ut libero faucibus lacinia.

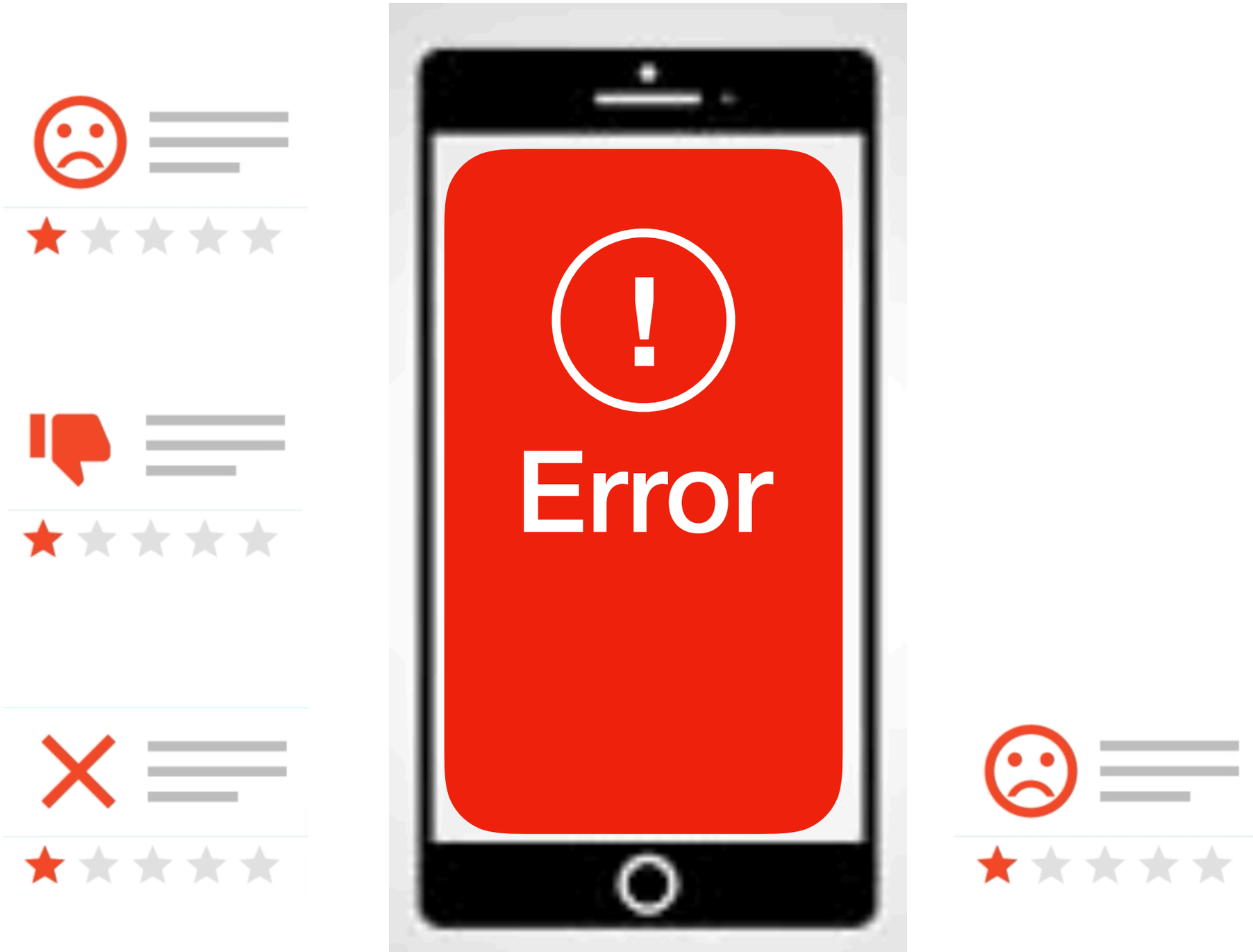

!
Error

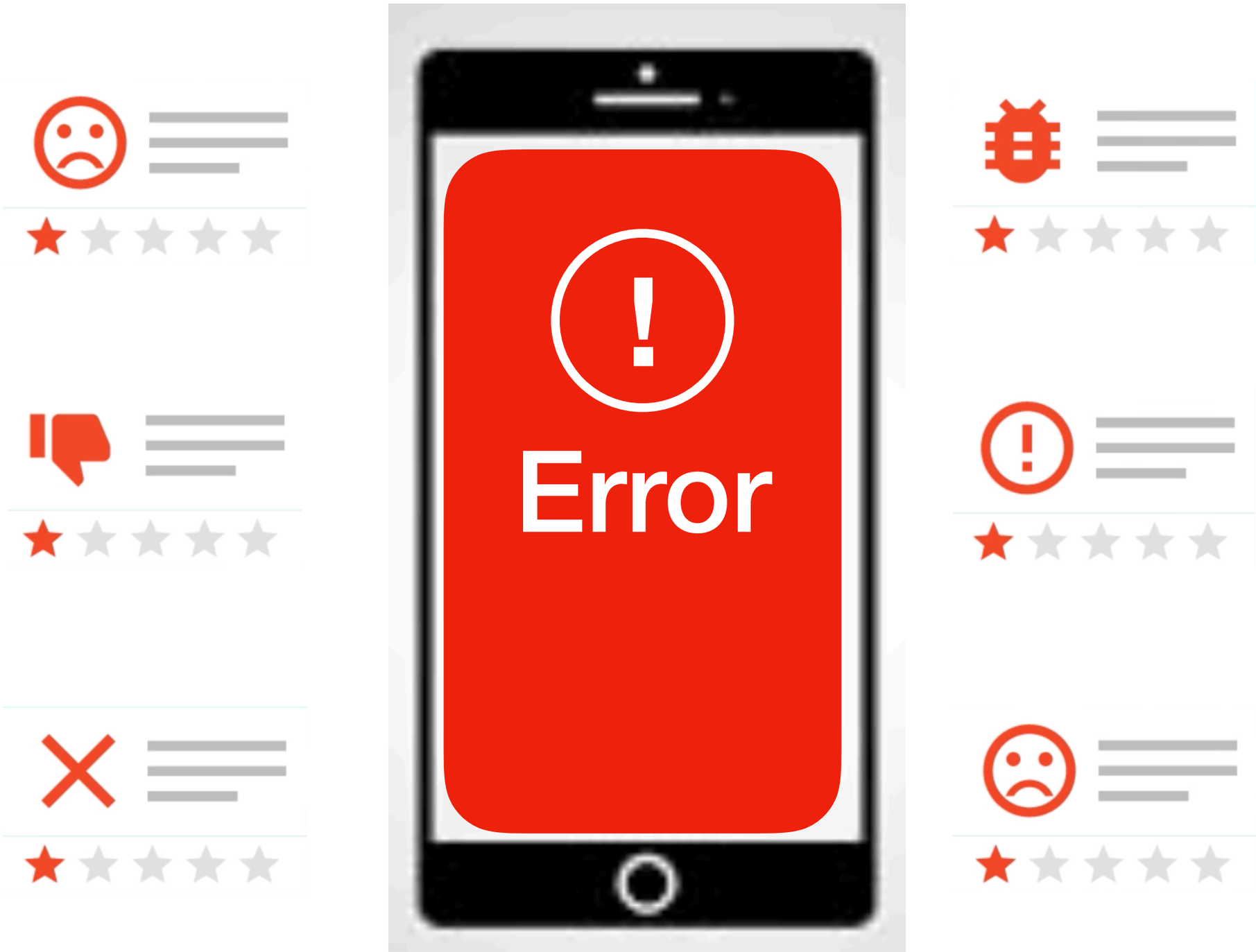


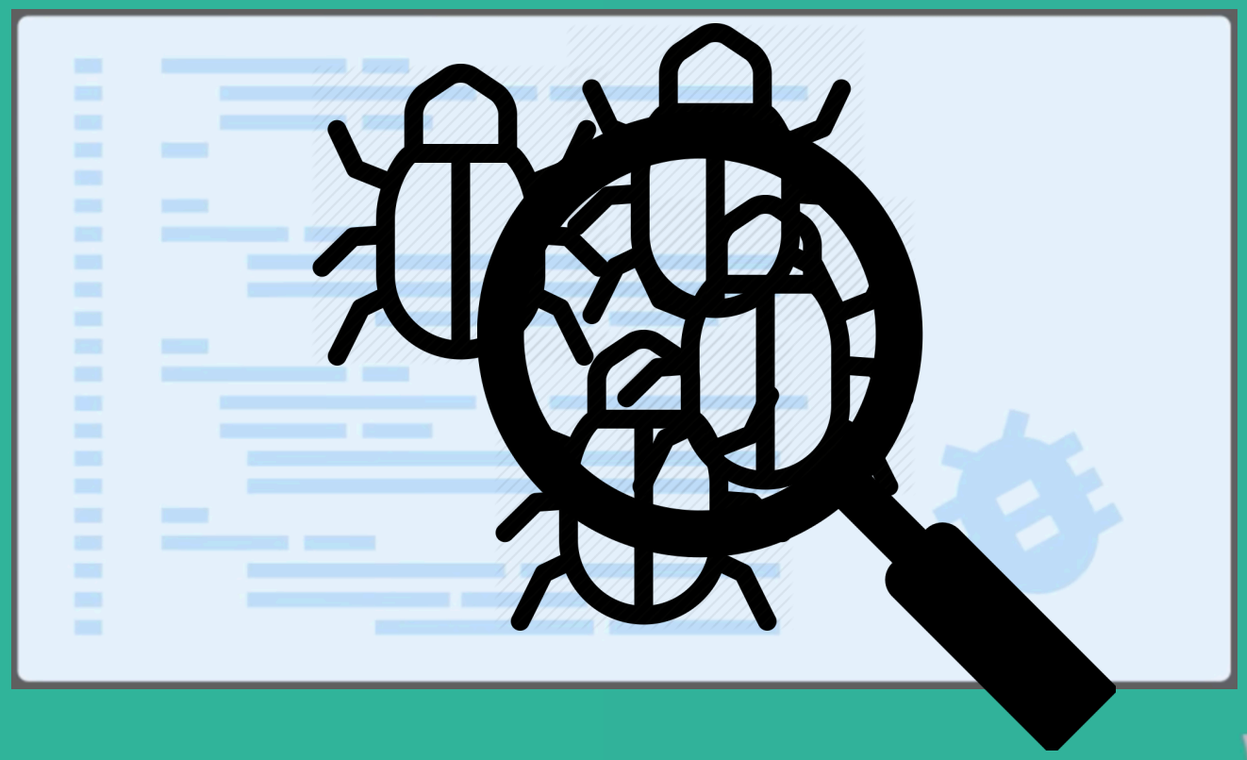

!
Error



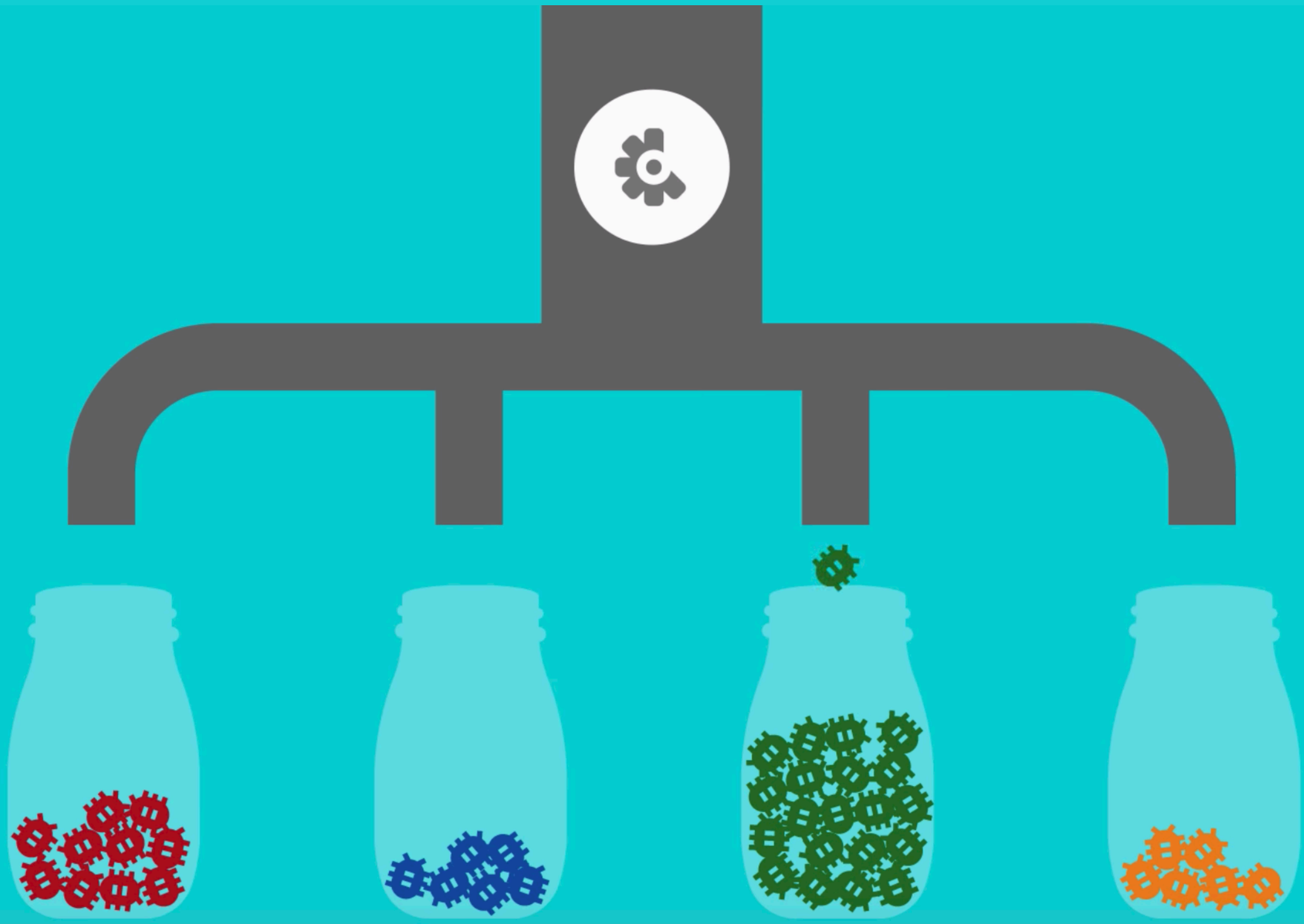

!
Error


!
Error


Error



Crashlytics



Enable the SDK

```
buildscript {
    repositories {
        // ...
        // Add repository
        maven {
            url 'https://maven.fabric.io/public'
        }
    }
    dependencies {
        // ...
        // Check for v3.1.2 or higher
        classpath 'com.google.gms:google-services:4.2.0'
        // Add dependency
        classpath 'io.fabric.tools:gradle:1.26.1'
    }
}
allprojects {
    // ...
    repositories {
        // ...
        // Add repository
        maven {
            url 'https://maven.google.com/'
        }
    }
}
```

Enable the SDK

```
apply plugin: 'com.android.application'
apply plugin: 'io.fabric'

dependencies {
    // ...
    // Check for v11.4.2 or higher
    implementation 'com.google.firebase:firebase-core:16.0.6'

    // Add dependency
    implementation 'com.crashlytics.sdk.android:crashlytics:2.9.7'
}
```

Test Implementation

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

Customize

AndroidManifest.xml

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

Customize

AndroidManifest.xml

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

Enable collection for selected users:

```
Fabric.with(this, Crashlytics())
```

Customize

AndroidManifest.xml

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

Enable collection for selected users:

```
Fabric.with(this, Crashlytics())
```

Crash report and `Log.println:`

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

Customize

`AndroidManifest.xml`

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

Enable collection for selected users:

```
Fabric.with(this, Crashlytics())
```

Crash report and `Log.println`:

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

Crash report only:

```
Crashlytics.log("message")
```

Customize

`AndroidManifest.xml`

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

Enable collection for selected users:

```
Fabric.with(this, Crashlytics())
```

Crash report and `Log.println`:

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

Crash report only:

```
Crashlytics.log("message")
```

Add custom keys:

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

Customize

`AndroidManifest.xml`

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

Enable collection for selected users:

```
Fabric.with(this, Crashlytics())
```

Crash report and `Log.println:`

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

Crash report only:

```
Crashlytics.log("message")
```

Add custom keys:

```
Crashlytics.setInt("current_level", 3)
Crashlytics.setString("last_UI_action", "teamed the dragon!")
```

Customize

AndroidManifest.xml

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

Enable collection for selected users:

```
Fabric.with(this, Crashlytics())
```

Crash report and `Log.println:`

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

Crash report only:

```
Crashlytics.log("message")
```

Add custom keys:

```
Crashlytics.setInt("current_level", 3)
Crashlytics.setString("last_UI_action", "teamed the dragon!")
```

Set user IDs

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

Customize

AndroidManifest.xml

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

Enable collection for selected users:

```
Fabric.with(this, Crashlytics())
```

Crash report and `Log.println:`

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

Crash report only:

```
Crashlytics.log("message")
```

Add custom keys:

```
Crashlytics.setInt("current_level", 3)
Crashlytics.setString("last_UI_action", "teamed the dragon!")
```

Set user IDs

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

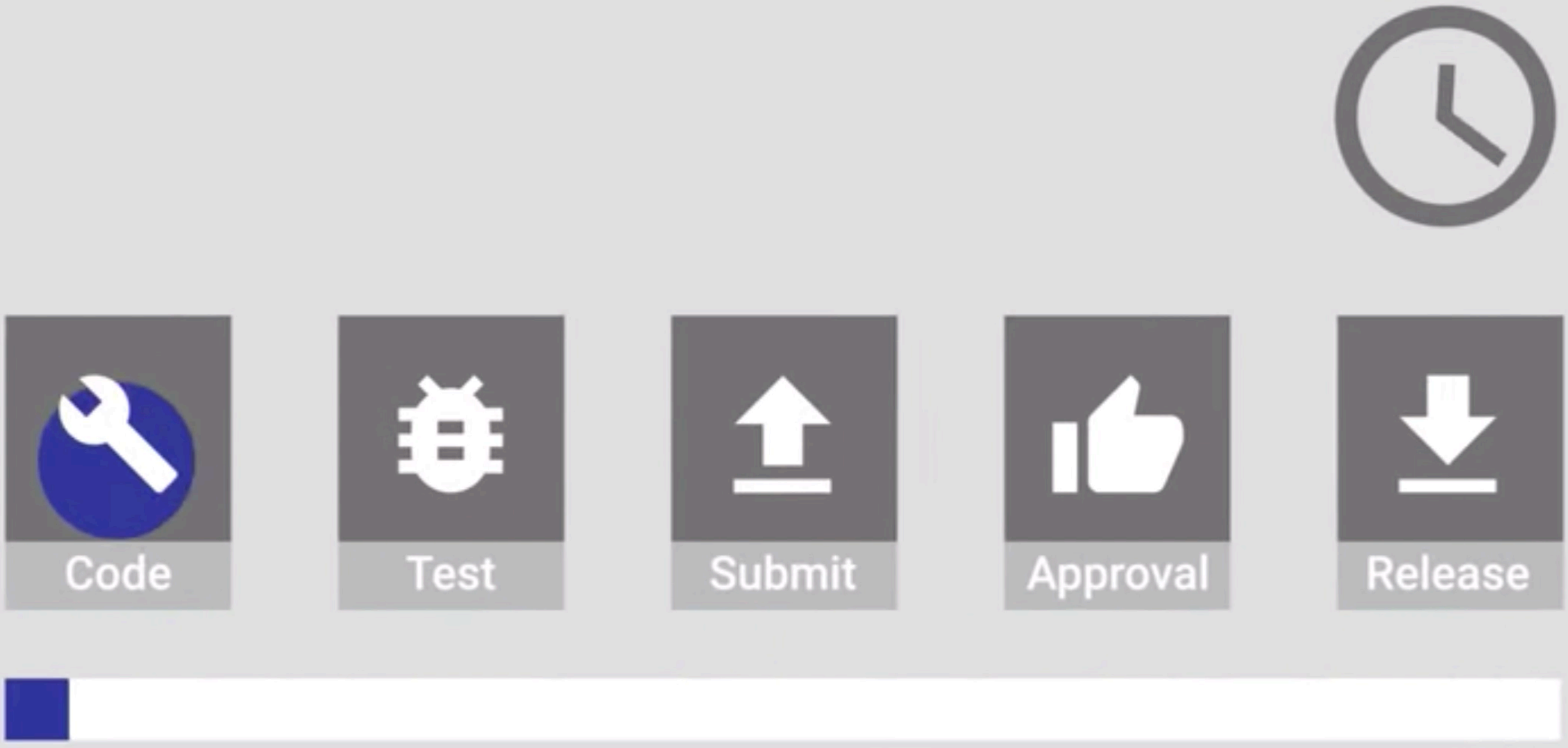
Log non-fatal exceptions

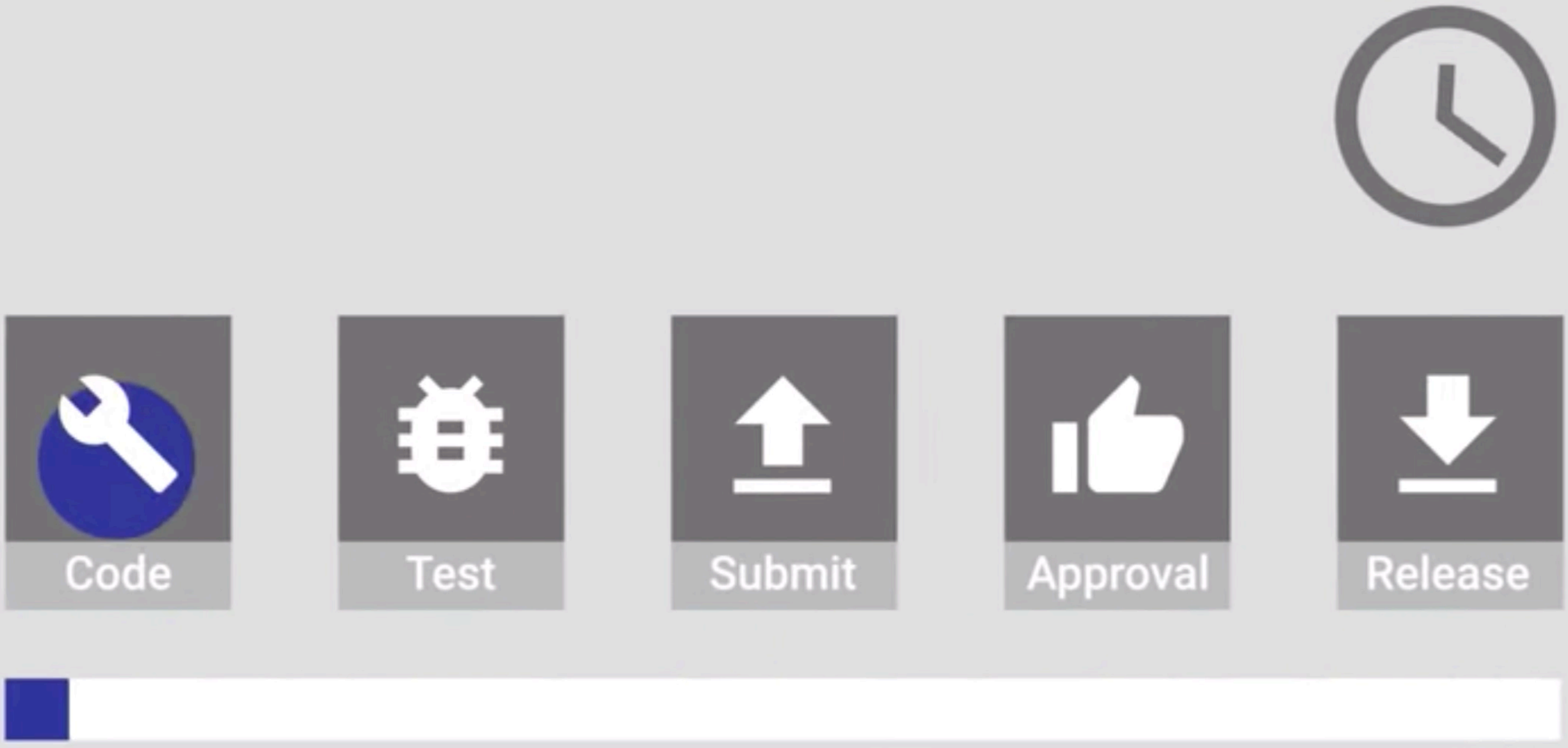
```
Crashlytics.logException(e)
```

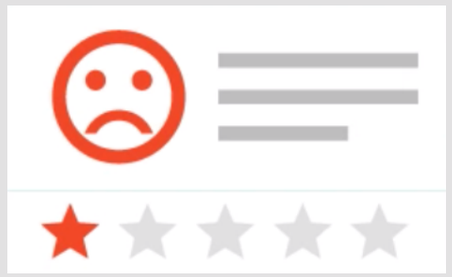
https://firebase.google.com/docs/crashlytics/customize-crash-reports

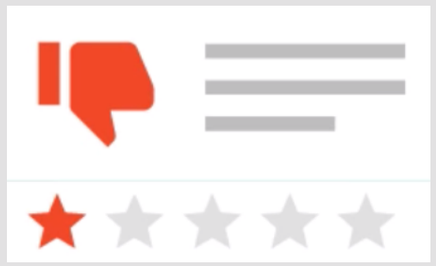

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas rutrum est ac ultrices porttitor. Suspendisse pellentesque risus vehicula, vehicula erat a, tempor nisi. Donec vel suscipit sapien, vitae sollicitudin dui. Fusce vitae nulla ornare, sollicitudin nibh et, malesuada eros. In nec mi consequat, laoreet massa et, molestie nisi. Proin eget dui lacus. Phasellus auctor faucibus facilisis.
Sed nec arcu nibh. Maecenas nulla velit, hendrerit at arcu ut, viverra ullamcorper eros. Proin efficitur libero in urna gravida, dictum tempor nibh bibendum. Ut eu libero non leo elementum molestie. Nullam convallis convallis tellus, vitae ultrices nisi bibendum non. Suspendisse placerat lorem augue, nec bibendum nisi mollis non. Mauris viverra posuere diam in laoreet. Sed non erat rutrum, bibendum risus a, molestie arcu. Praesent sit amet arcu ut libero faucibus lacinia.

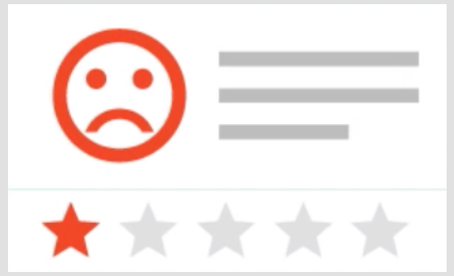

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas rutrum est ac ultrices porttitor. Suspendisse pellentesque risus vehicula, vehicula erat a, tempor nisi. Donec vel suscipit sapien, vitae sollicitudin dui. Fusce vitae nulla ornare, sollicitudin nibh et, malesuada eros. In nec mi consequat, laoreet massa et, molestie nisi. Proin eget dui lacus. Phasellus auctor faucibus facilisis.
Sed nec arcu nibh. Maecenas nulla velit, hendrerit at arcu ut, viverra ullamcorper eros. Proin efficitur libero in urna gravida, dictum tempor nibh bibendum. Ut eu libero non leo elementum molestie. Nullam convallis convallis tellus, vitae ultrices nisi bibendum non. Suspendisse placerat lorem augue, nec bibendum nisi mollis non. Mauris viverra posuere diam in laoreet. Sed non erat rutrum, bibendum risus a, molestie arcu. Praesent sit amet arcu ut libero faucibus lacinia.


Code
Test
Submit
Approval
Release


Code
Test
Submit
Approval
Release



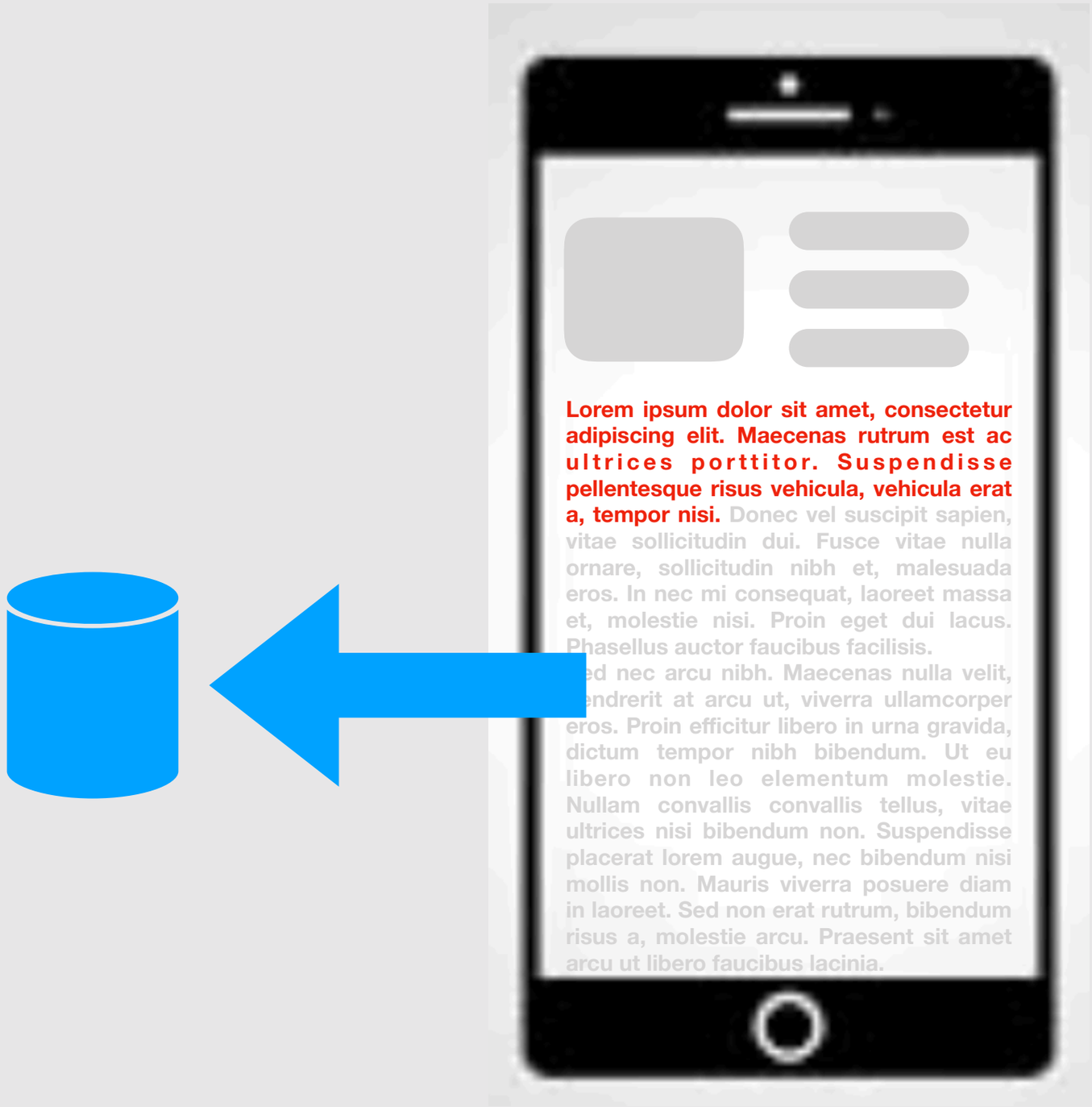





Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas rutrum est ac ultrices porttitor. Suspendisse pellentesque risus vehicula, vehicula erat a, tempor nisi. Donec vel suscipit sapien, vitae sollicitudin dui. Fusce vitae nulla ornare, sollicitudin nibh et, malesuada eros. In nec mi consequat, laoreet massa et, molestie nisi. Proin eget dui lacus. Phasellus auctor faucibus facilisis.
Sed nec arcu nibh. Maecenas nulla velit, hendrerit at arcu ut, viverra ullamcorper eros. Proin efficitur libero in urna gravida, dictum tempor nibh bibendum. Ut eu libero non leo elementum molestie. Nullam convallis convallis tellus, vitae ultrices nisi bibendum non. Suspendisse placerat lorem augue, nec bibendum nisi mollis non. Mauris viverra posuere diam in laoreet. Sed non erat rutrum, bibendum risus a, molestie arcu. Praesent sit amet arcu ut libero faucibus lacinia.


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas rutrum est ac ultrices porttitor. Suspendisse pellentesque risus vehicula, vehicula erat a, tempor nisi. Donec vel suscipit sapien, vitae sollicitudin dui. Fusce vitae nulla ornare, sollicitudin nibh et, malesuada eros. In nec mi consequat, laoreet massa et, molestie nisi. Proin eget dui lacus. Phasellus auctor faucibus facilisis.
Sed nec arcu nibh. Maecenas nulla velit, hendrerit at arcu ut, viverra ullamcorper eros. Proin efficitur libero in urna gravida, dictum tempor nibh bibendum. Ut eu libero non leo elementum molestie. Nullam convallis convallis tellus, vitae ultrices nisi bibendum non. Suspendisse placerat lorem augue, nec bibendum nisi mollis non. Mauris viverra posuere diam in laoreet. Sed non erat rutrum, bibendum risus a, molestie arcu. Praesent sit amet arcu ut libero faucibus lacinia.


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas rutrum est ac ultrices porttitor. Suspendisse pellentesque risus vehicula, vehicula erat a, tempor nisi. Donec vel suscipit sapien, vitae sollicitudin dui. Fusce vitae nulla ornare, sollicitudin nibh et, malesuada eros. In nec mi consequat, laoreet massa et, molestie nisi. Proin eget dui lacus. Phasellus auctor faucibus facilisis.
Sed nec arcu nibh. Maecenas nulla velit, hendrerit at arcu ut, viverra ullamcorper eros. Proin efficitur libero in urna gravida, dictum tempor nibh bibendum. Ut eu libero non leo elementum molestie. Nullam convallis convallis tellus, vitae ultrices nisi bibendum non. Suspendisse placerat lorem augue, nec bibendum nisi mollis non. Mauris viverra posuere diam in laoreet. Sed non erat rutrum, bibendum risus a, molestie arcu. Praesent sit amet arcu ut libero faucibus lacinia.


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas rutrum est ac ultrices porttitor. Suspendisse pellentesque risus vehicula, vehicula erat a, tempor nisi. Donec vel suscipit sapien, vitae sollicitudin dui. Fusce vitae nulla ornare, sollicitudin nibh et, malesuada eros. In nec mi consequat, laoreet massa et, molestie nisi. Proin eget dui lacus. Phasellus auctor faucibus facilisis.
ed nec arcu nibh. Maecenas nulla velit, endrerit at arcu ut, viverra ullamcorper eros. Proin efficitur libero in urna gravida, dictum tempor nibh bibendum. Ut eu libero non leo elementum molestie. Nullam convallis convallis tellus, vitae ultrices nisi bibendum non. Suspendisse placerat lorem augue, nec bibendum nisi mollis non. Mauris viverra posuere diam in laoreet. Sed non erat rutrum, bibendum risus a, molestie arcu. Praesent sit amet arcu ut libero faucibus lacinia.

Realtime

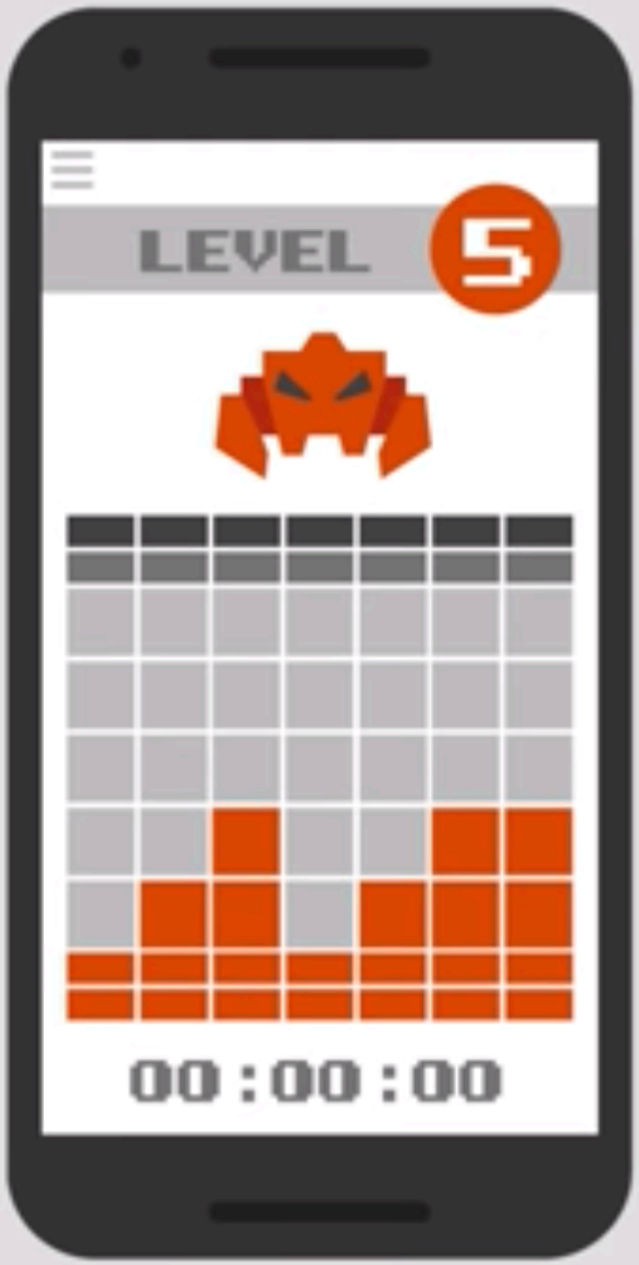

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas rutrum est ac ultrices porttitor. Suspendisse pellentesque risus vehicula, vehicula erat a, tempor nisi. Donec vel suscipit sapien, vitae sollicitudin dui. Fusce vitae nulla ornare, sollicitudin nibh et, malesuada eros. In nec mi consequat, laoreet massa et, molestie nisi. Proin eget dui lacus. Phasellus auctor faucibus facilisis.
ed nec arcu nibh. Maecenas nulla velit, endrerit at arcu ut, viverra ullamcorper eros. Proin efficitur libero in urna gravida, dictum tempor nibh bibendum. Ut eu libero non leo elementum molestie. Nullam convallis convallis tellus, vitae ultrices nisi bibendum non. Suspendisse placerat lorem augue, nec bibendum nisi mollis non. Mauris viverra posuere diam in laoreet. Sed non erat rutrum, bibendum risus a, molestie arcu. Praesent sit amet arcu ut libero faucibus lacinia.

Realtime

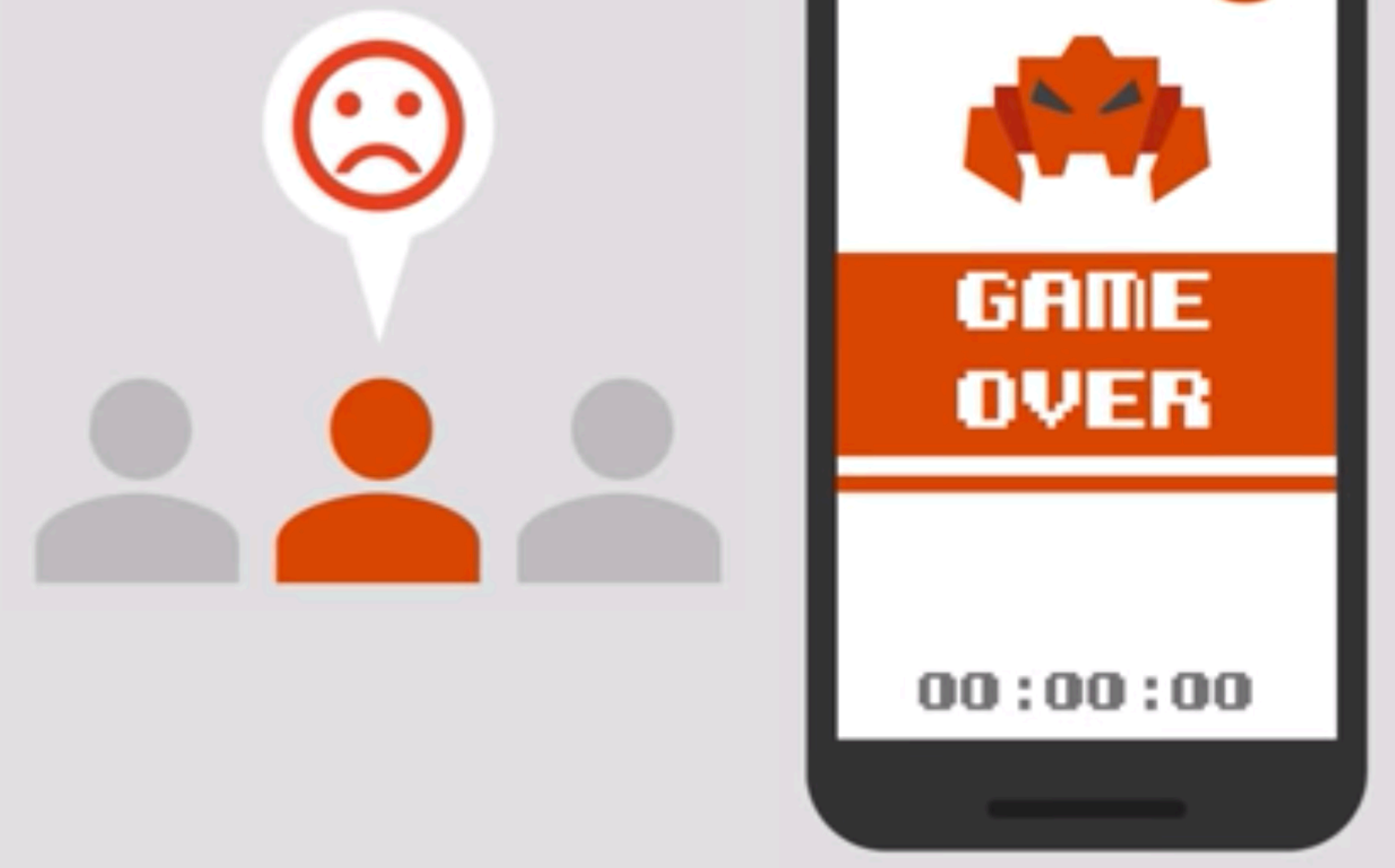

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas rutrum est ac ultrices porttitor. Suspendisse pellentesque risus vehicula, vehicula erat a, tempor nisi. Donec vel suscipit sapien, vitae sollicitudin dui. Fusce vitae nulla ornare, sollicitudin nibh et, malesuada
ros. In nec mi consequat, laoreet massa
molestie nisi. Proin eget dui lacus.
us auctor faucibus facilisis.
rcu nibh. Maecenas nulla velit,
at arcu ut, viverra ullamcorper
roin efficitur libero in urna gravida,
um tempor nibh bibendum. Ut eu
ibero non leo elementum molestie.
Nullam convallis convallis tellus, vitae ultrices nisi bibendum non. Suspendisse placerat lorem augue, nec bibendum nisi mollis non. Mauris viverra posuere diam in laoreet. Sed non erat rutrum, bibendum risus a, molestie arcu. Praesent sit amet arcu ut libero faucibus lacinia.

Realtime


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas rutrum est ac ultrices porttitor. Suspendisse pellentesque risus vehicula, vehicula erat a, tempor nisi. Donec vel suscipit sapien, vitae sollicitudin dui. Fusce vitae nulla ornare, sollicitudin nibh et, malesuada
ros. In nec mi consequat, laoreet massa
molestie nisi. Proin eget dui lacus.
us auctor faucibus facilisis.
rcu nibh. Maecenas nulla velit,
at arcu ut, viverra ullamcorper
roin efficitur libero in urna gravida,
um tempor nibh bibendum. Ut eu
ibero non leo elementum molestie.
Nullam convallis convallis tellus, vitae ultrices nisi bibendum non. Suspendisse placerat lorem augue, nec bibendum nisi mollis non. Mauris viverra posuere diam in laoreet. Sed non erat rutrum, bibendum risus a, molestie arcu. Praesent sit amet arcu ut libero faucibus lacinia.


LEVEL
5
00 : 00 : 00

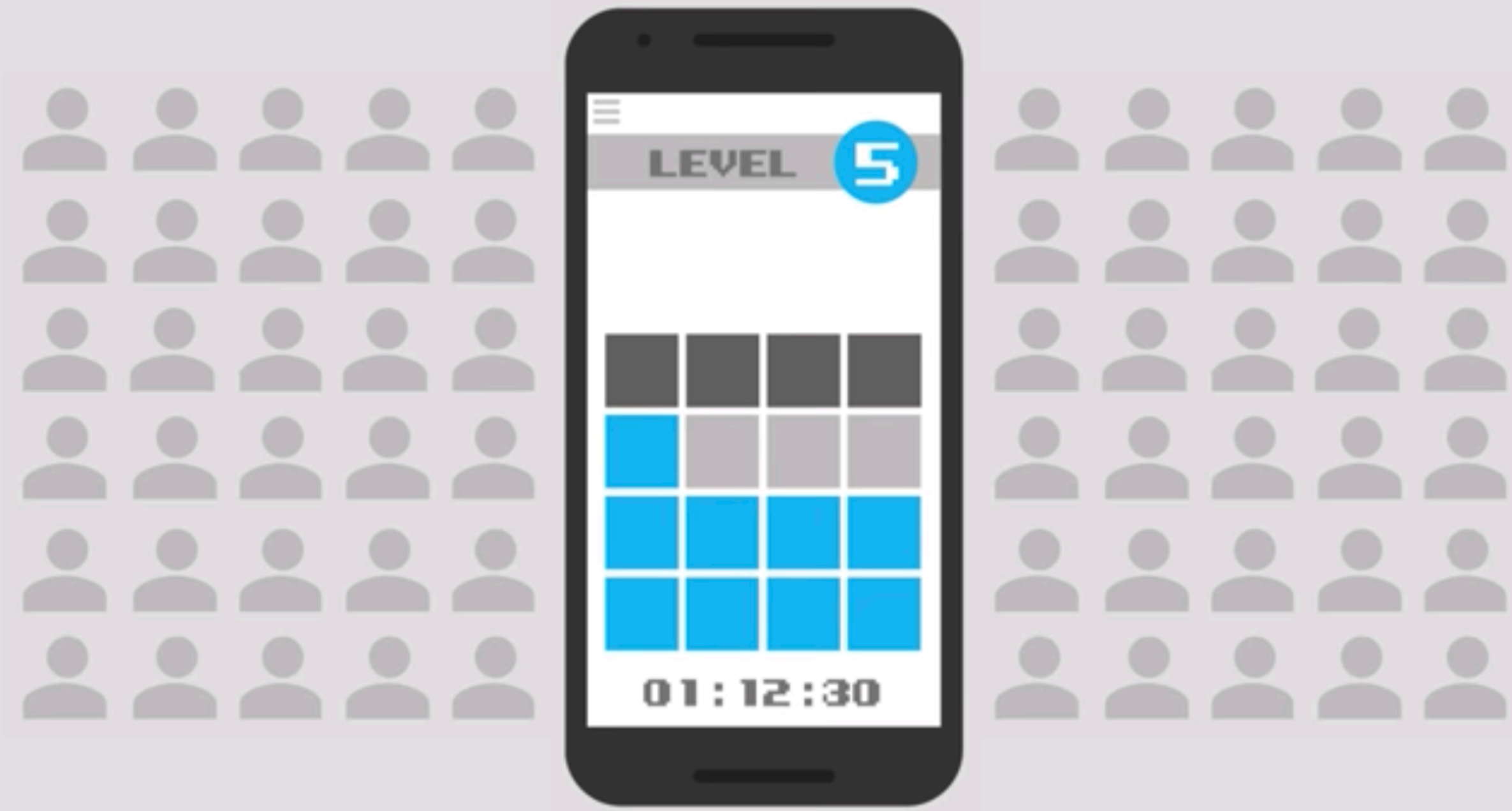

LEVEL
5
GAME
OVER
00 : 00 : 00

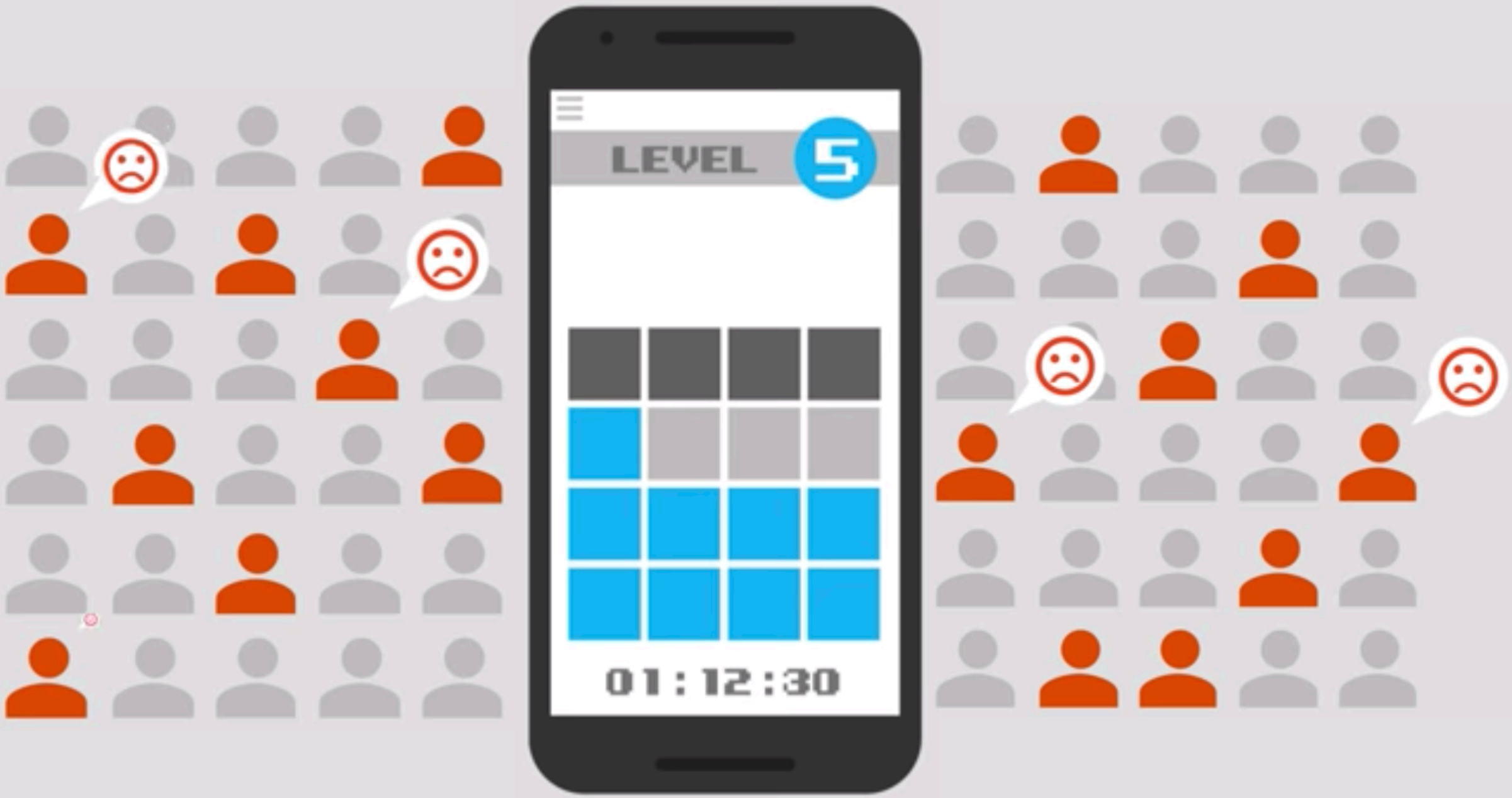

LEVEL
5
GAME
OVER
00 : 00 : 00

Realtime

Realtime

Realtime


LEVEL
5
01:12:30


LEVEL
5
01:12:30

A

B

Remote Config

Using Remote Config

1 Add Firebase to your app

```
implementation 'com.google.firebase:firebase-config:16.1.2'
```

Using Remote Config

1 Add Firebase to your app

```
implementation 'com.google.firebase:firebase-config:16.1.2'
```

2 Get the Remote Config singleton object

```
FirebaseRemoteConfig.getInstance()
```

Using Remote Config

1 Add Firebase to your app

```
implementation 'com.google.firebase:firebase-config:16.1.2'
```

2 Get the Remote Config singleton object

```
FirebaseRemoteConfig.getInstance()
```

3 Set in-app default parameter values

Using Remote Config

1 Add Firebase to your app

```
implementation 'com.google.firebase:firebase-config:16.1.2'
```

2 Get the Remote Config singleton object

```
FirebaseRemoteConfig.getInstance()
```

3 Set in-app default parameter values

4 Get parameter values to use in your app

Using Remote Config

1. Add Firebase to your app

```
implementation 'com.google.firebase:firebase-config:16.1.2'
```

2. Get the Remote Config singleton object

```
FirebaseRemoteConfig.getInstance()
```

3. Set in-app default parameter values

4. Get parameter values to use in your app

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

Using Remote Config

1. Add Firebase to your app

```
implementation 'com.google.firebase:firebase-config:16.1.2'
```

2. Get the Remote Config singleton object

```
FirebaseRemoteConfig.getInstance()
```

3. Set in-app default parameter values

4. Get parameter values to use in your app

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

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

Using Remote Config

DEMO

1. Add Firebase to your app

```
implementation 'com.google.firebase:firebase-config:16.1.2'
```

2. Get the Remote Config singleton object

```
FirebaseRemoteConfig.getInstance()
```

3. Set in-app default parameter values

4. Get parameter values to use in your app

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.

Make $

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

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

AdMob

Import the Mobile Ads SDK

```
implementation 'com.google.firebase:firebase-ads:17.1.2'
```

AdMob

Import the Mobile Ads SDK

```
implementation 'com.google.firebase:firebase-ads:17.1.2'
```

Update your AndroidManifest.xml

```
<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>
```

AdMob

Import the Mobile Ads SDK

```
implementation 'com.google.firebase:firebase-ads:17.1.2'
```

Update your AndroidManifest.xml

```
<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")
}
```

Ad Formats

Ad Formats

- Banner

```
<com.google.android.gms.ads.AdView
    xmlns:ads="http://schemas.android.com/apk/res-auto"
    android:id="@+id/adView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_alignParentBottom="true"
    ads:adSize="BANNER"
    ads:adUnitId="ca-app-pub-3940256099942544/6300978111">
</com.google.android.gms.ads.AdView>
```

Ad Formats

- Banner

```
<com.google.android.gms.ads.AdView
    xmlns:ads="http://schemas.android.com/apk/res-auto"
    android:id="@+id/adView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_alignParentBottom="true"
    ads:adSize="BANNER"
    ads:adUnitId="ca-app-pub-3940256099942544/6300978111">
 </com.google.android.gms.ads.AdView>
```

```
val adView = AdView(this)
adView.adSize = AdSize.BANNER
adView.adUnitId = "ca-app-pub-3940256099942544/6300978111"
// TODO: Add adView to your view hierarchy.
```

https://developers.google.com/admob/android/banner

Ad Formats

- Banner
- Interstitial

```
class MainActivity : Activity() {

    private lateinit var mInterstitialAd: InterstitialAd

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        MobileAds.initialize(this,
            "ca-app-pub-3940256099942544~3347511713")

        mInterstitialAd = InterstitialAd(this)
        mInterstitialAd.adUnitId = "ca-app-pub-3940256099942544/1033173712"
        mInterstitialAd.loadAd(AdRequest.Builder().build())
    }
}
```

https://developers.google.com/admob/android/interstitial

Ad Formats

- Banner
- Interstitial
- Native

```
val adLoader = AdLoader.Builder(this, "ca-app-pub-3940256099942544/2247696110")
        .forUnifiedNativeAd { ad : UnifiedNativeAd ->
            // Show the ad.
        }
        .withAdListener(object : AdListener() {
            override fun onAdFailedToLoad(errorCode: Int) {
                // Handle the failure by logging, altering the UI, and so on.
            }
        })
        .withNativeAdOptions(NativeAdOptions.Builder()
                // Methods in the NativeAdOptions.Builder class can be
                // used here to specify individual options settings.
                .build())
        .build()
```

https://developers.google.com/admob/android/native/start

Ad Formats

- Banner
- Interstitial
- Native
- Rewarded Video

```
import com.google.android.gms.ads.AdRequest
import com.google.android.gms.ads.MobileAds
import com.google.android.gms.ads.reward.RewardedVideoAd

class MainActivity : AppCompatActivity(), RewardedVideoAdListener {

private lateinit var mRewardedVideoAd: RewardedVideoAd

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        MobileAds.initialize(this, "ca-app-pub-3940256099942544~3347511713")

        // Use an activity context to get the rewarded video instance.
```

DEMO

Ad Formats

```
import com.google.android.gms.ads.AdRequest
import com.google.android.gms.ads.MobileAds
import com.google.android.gms.ads.reward.RewardedVideoAd

class MainActivity : AppCompatActivity(), RewardedVideoAdListener {

    private lateinit var mRewardedVideoAd: RewardedVideoAd

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        MobileAds.initialize(this, "ca-app-pub-3940256099942544~3347511713")

        // Use an activity context to get the rewarded video instance.
        mRewardedVideoAd = MobileAds.getRewardedVideoAdInstance(this)
        mRewardedVideoAd.rewardedVideoAdListener = this
    }
    ...
}
```

https://developers.google.com/admob/android/rewarded-video

DEMO

Google Play Billing Overview

- 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

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.