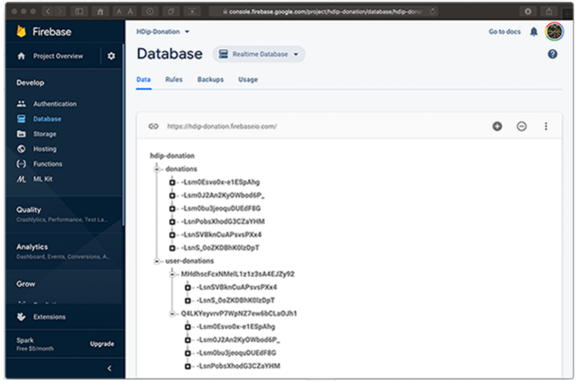
Mobile Application Development

Produced by David Drohan (ddrohan@wit.ie)

Department of Computing & Mathematics
Waterford Institute of Technology
http://www.wit.ie

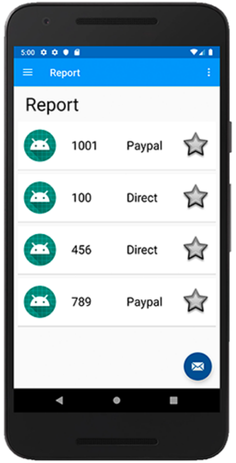
Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Donation-V5 Walkthrough



https://hdip-donation.firebaseio.com/

```
hdip-donation
  donations
    -Lsm0Esvo0x-e1ESpAhg
    -Lsm0J2An2KyOWbod6P_
    -Lsm0bu3jeoquDUEdF8G
    -LsnPobsXhodG3CZaYHM
    -LsnSVBknCuAPsvsPXx4
    -LsnS_0oZKDBhK0lzDpT
  user-donations
    MHdhscFcxNMelL1z1z3sA4EJZy92
      -LsnSVBknCuAPsvsPXx4
      -LsnS_0oZKDBhK0lzDpT
    Q4LKYeyvrvP7WpNZ7ew6bCLaOJh1
      -Lsm0Esvo0x-e1ESpAhg
      -Lsm0J2An2KyOWbod6P_
      -Lsm0bu3jeoquDUEdF8G
      -LsnPobsXhodG3CZaYHM
```



Report

1001	Paypal
100	Direct
456	Direct
789	Paypal

Steps to integrate Firebase Database into your App

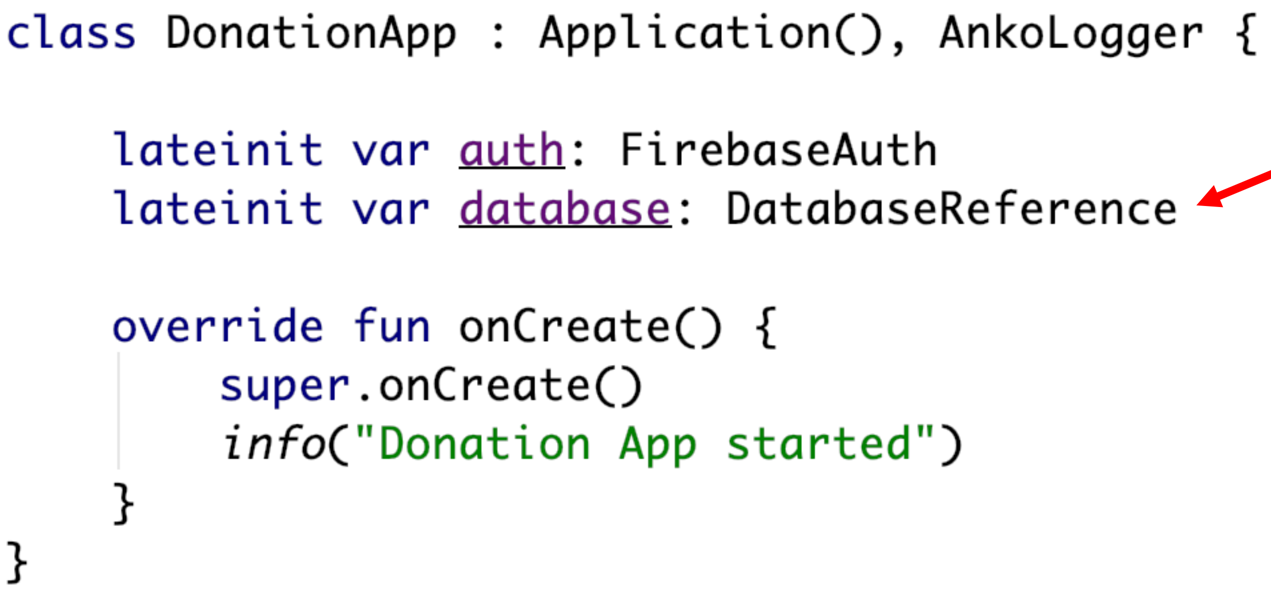
1. Create your Firebase Project
2. Setup your Database & Rules
3. Setup Firebase Database in your App
4. Introduce Access Flow – CRUD operations on Donations

Steps to integrate Firebase Database into your App

1. Create your Firebase Project
2. Setup your Database & Rules
3. Setup Firebase Database in your App
4. Introduce Access Flow – CRUD operations on Donations

https://github.com/firebase/quickstart-android

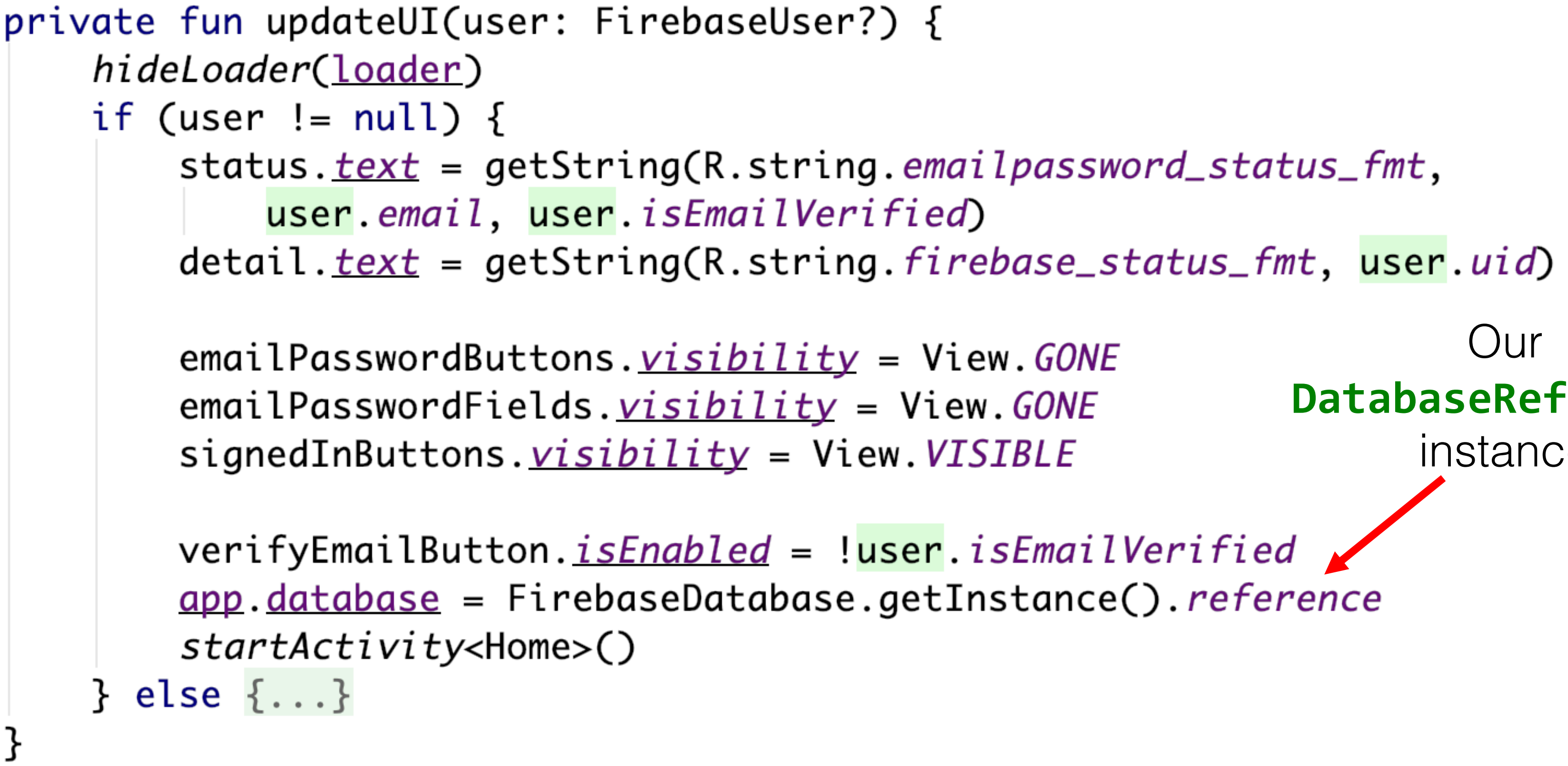
Introduce Access Flow – **DonationApp**



```
class DonationApp : Application(), AnkoLogger {

    lateinit var auth: FirebaseAuth
    lateinit var database: DatabaseReference

    override fun onCreate() {
        super.onCreate()
        info("Donation App started")
    }
}
```

Our **DatabaseReference** instance

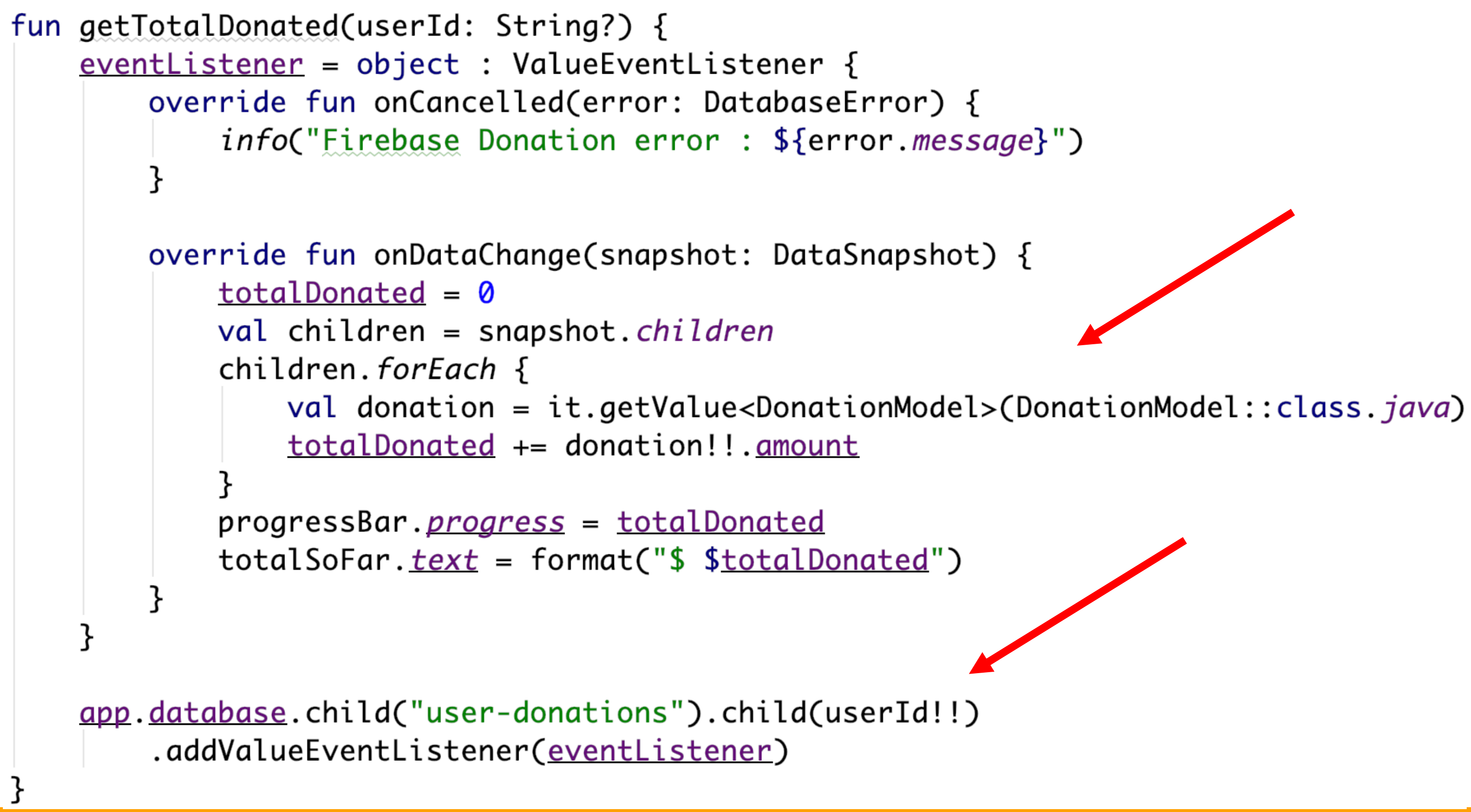
Introduce Access Flow – **Login**



```
private fun updateUI(user: FirebaseUser?) {
    hideLoader(loader)
    if (user != null) {
        status.text = getString(R.string.emailpassword_status_fmt,
            user.email, user.isEmailVerified)
        detail.text = getString(R.string.firebase_status_fmt, user.uid)

        emailPasswordButtons.visibility = View.GONE
        emailPasswordFields.visibility = View.GONE
        signedInButtons.visibility = View.VISIBLE

        verifyEmailButton.isEnabled = !user.isEmailVerified
        app.database = FirebaseDatabase.getInstance().reference
        startActivity<Home>()
    } else {...}
}
```

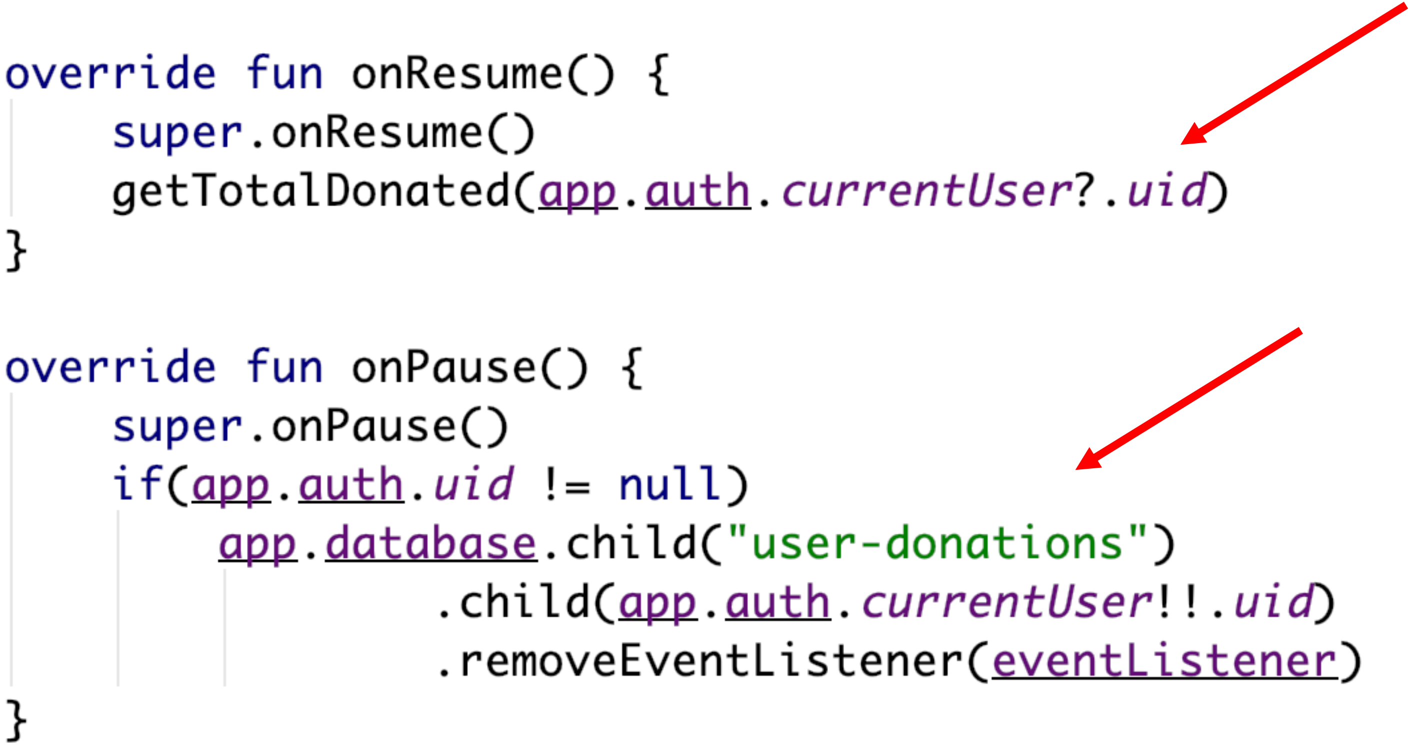
Our **DatabaseReference** instance

Introduce Access Flow – **Donate**



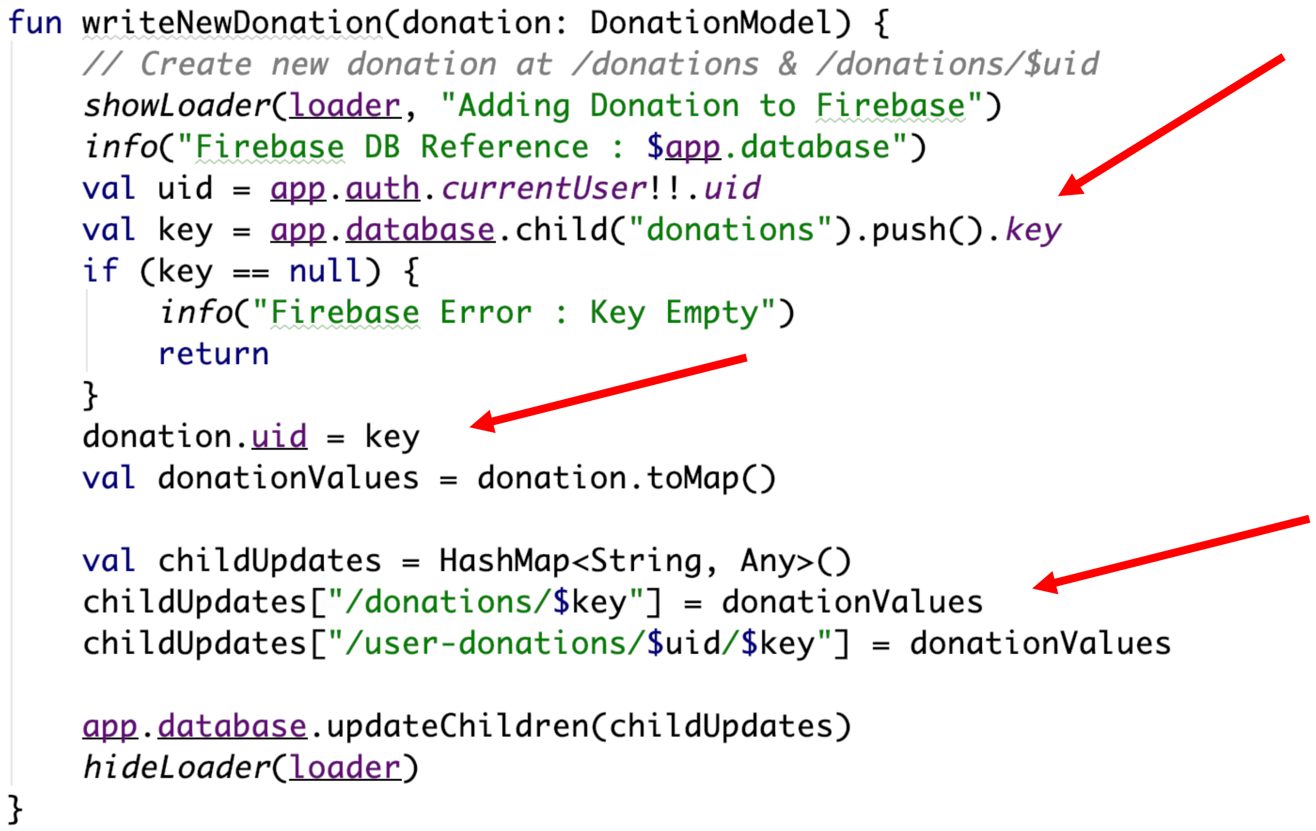
```
fun getTotalDonated(userId: String?) {
    eventListener = object : ValueEventListener {
        override fun onCancelled(error: DatabaseError) {
            info("Firebase Donation error : ${error.message}")
        }

        override fun onDataChange(snapshot: DataSnapshot) {
            totalDonated = 0
            val children = snapshot.children
            children.forEach {
                val donation = it.getValue<DonationModel>(DonationModel::class.java)
                totalDonated += donation!!.amount
            }
            progressBar.progress = totalDonated
            totalSoFar.text = format("$ $totalDonated")
        }
    }

    app.database.child("user-donations").child(userId!!)
        .addValueEventListener(eventListener)
}
```

Introduce Access Flow – **Donate**



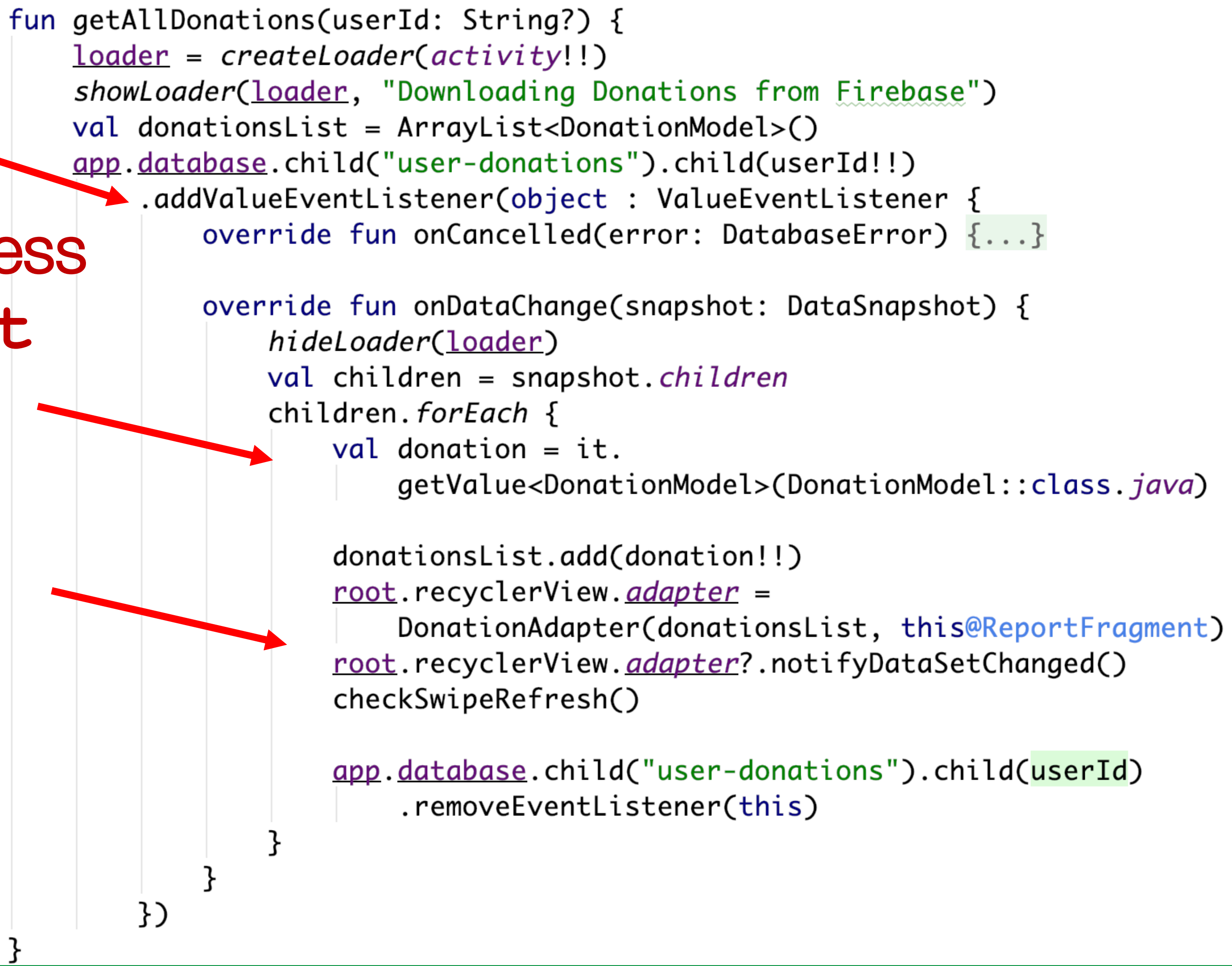
```
override fun onResume() {
    super.onResume()
    getTotalDonated(app.auth.currentUser?.uid)
}

override fun onPause() {
    super.onPause()
    if(app.auth.uid != null)
        app.database.child("user-donations")
                .child(app.auth.currentUser!!.uid)
                .removeEventListener(eventListener)
}
```

Introduce Access Flow – **Donate**



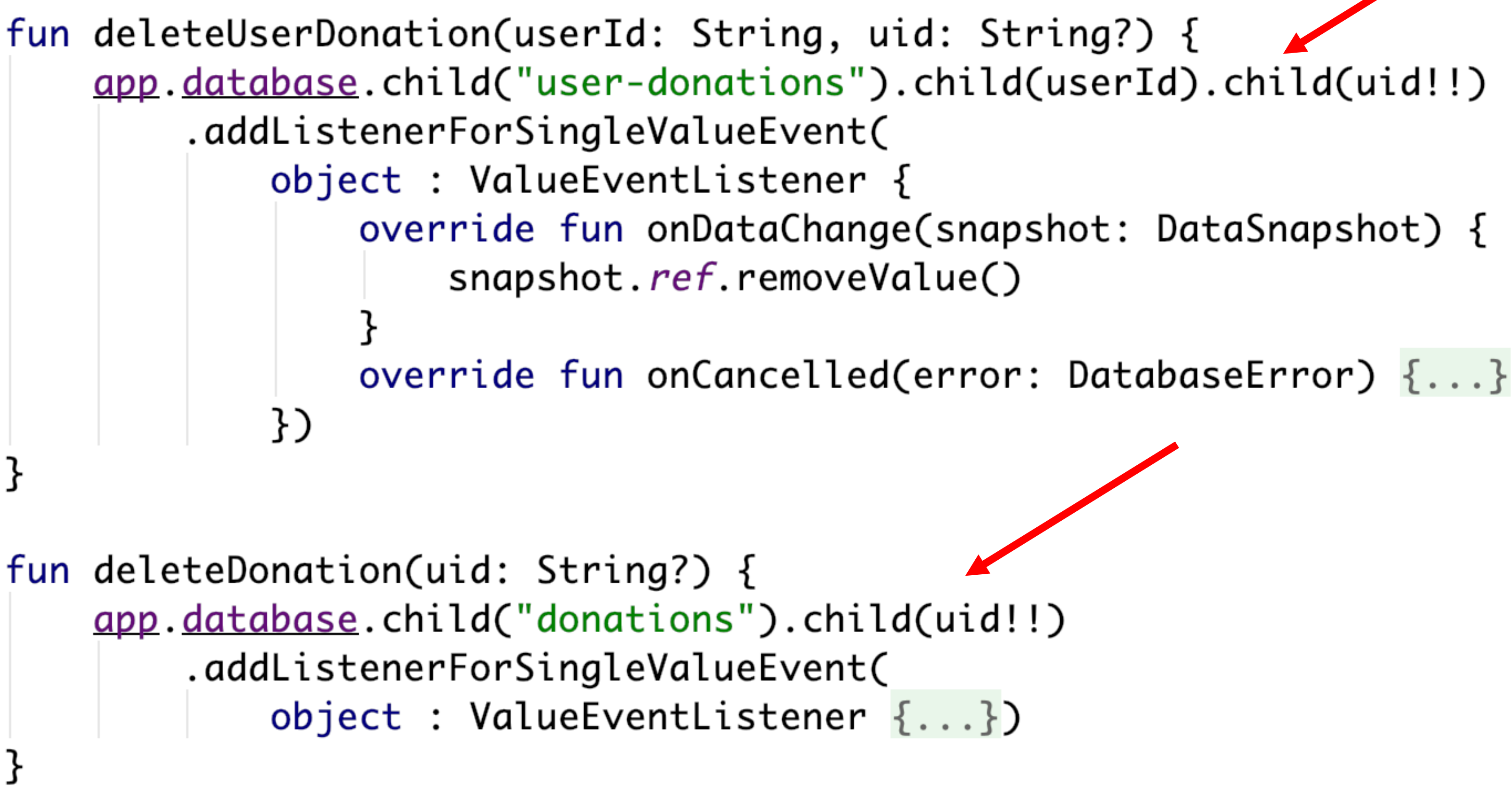
```
fun writeNewDonation(donation: DonationModel) {
    // Create new donation at /donations & /donations/$uid
    showLoader(loader, "Adding Donation to Firebase")
    info("Firebase DB Reference : $app.database")
    val uid = app.auth.currentUser!!.uid
    val key = app.database.child("donations").push().key
    if (key == null) {
        info("Firebase Error : Key Empty")
        return
    }
    donation.uid = key
    val donationValues = donation.toMap()

    val childUpdates = HashMap<String, Any>()
    childUpdates["/donations/$key"] = donationValues
    childUpdates["/user-donations/$uid/$key"] = donationValues

    app.database.updateChildren(childUpdates)
    hideLoader(loader)
}
```

Introduce Access Flow – **Report**



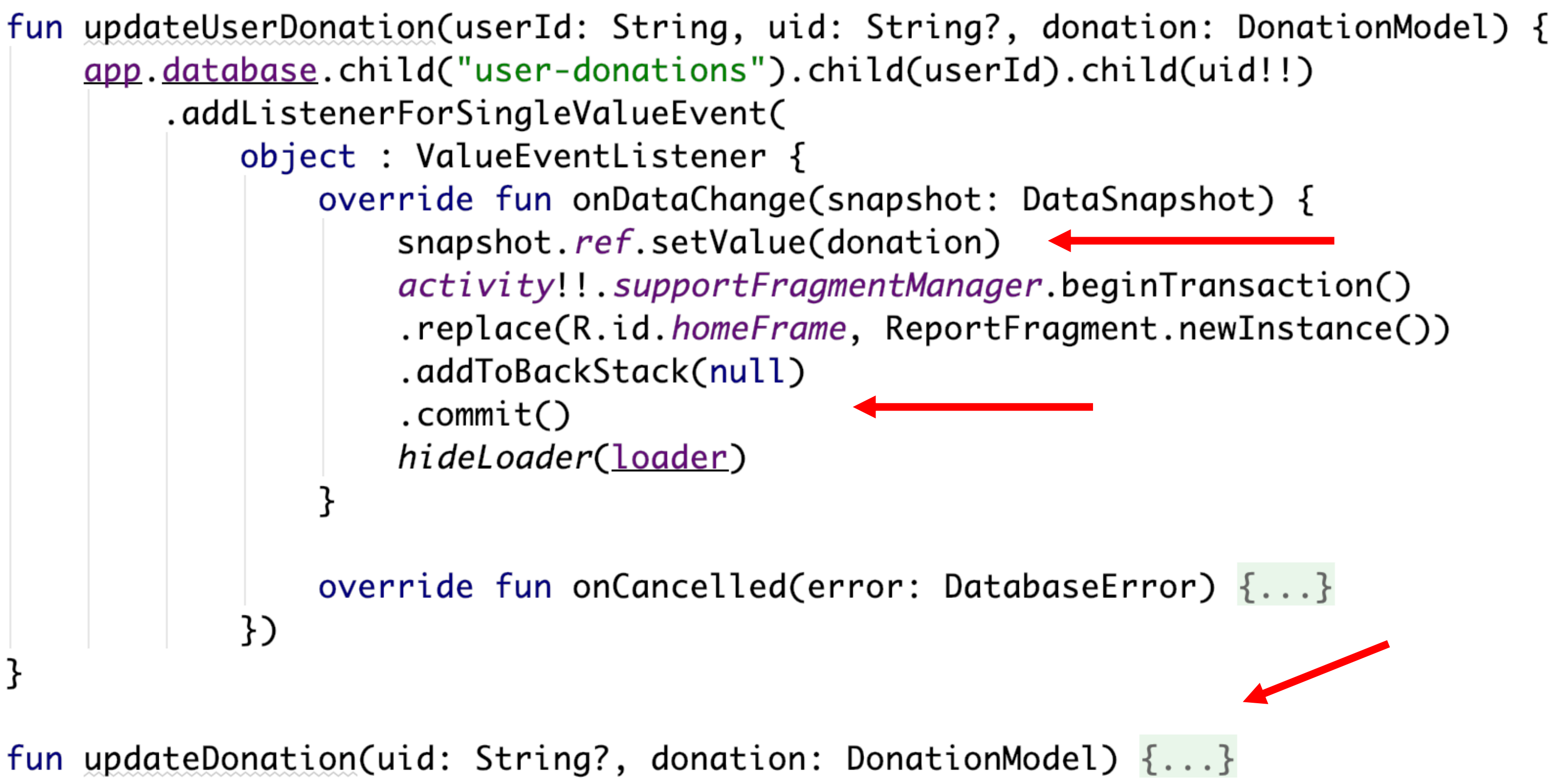
```
fun getAllDonations(userId: String?) {
    loader = createLoader(activity!!)
    showLoader(loader, "Downloading Donations from Firebase")
    val donationsList = ArrayList<DonationModel>()
    app.database.child("user-donations").child(userId!!)
        .addValueEventListener(object : ValueEventListener {
            override fun onCancelled(error: DatabaseError) {...}

            override fun onDataChange(snapshot: DataSnapshot) {
                hideLoader(loader)
                val children = snapshot.children
                children.forEach {
                    val donation = it.
                        getValue<DonationModel>(DonationModel::class.java)

                    donationsList.add(donation!!)
                    root.recyclerView.adapter =
                        DonationAdapter(donationsList, this@ReportFragment)
                    root.recyclerView.adapter?.notifyDataSetChanged()
                    checkSwipeRefresh()

                    app.database.child("user-donations").child(userId)
                        .removeEventListener(this)
                }
            }
        })
}
```

Introduce Access Flow – **Report**



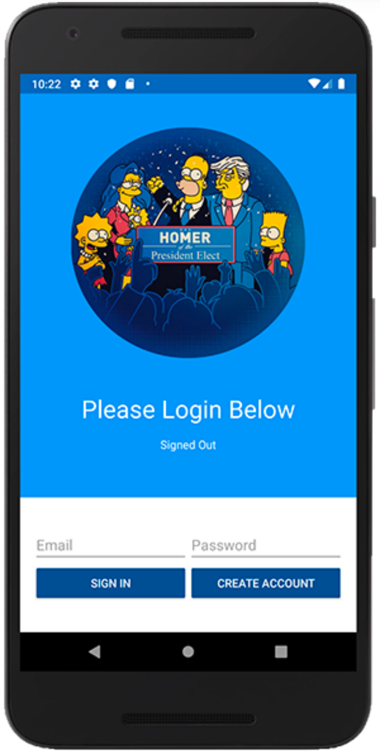
```
fun deleteUserDonation(userId: String, uid: String?) {
    app.database.child("user-donations").child(userId).child(uid!!)
        .addListenerForSingleValueEvent(
            object : ValueEventListener {
                override fun onDataChange(snapshot: DataSnapshot) {
                    snapshot.ref.removeValue()
                }
                override fun onCancelled(error: DatabaseError) {...}
            })
}

fun deleteDonation(uid: String?) {
    app.database.child("donations").child(uid!!)
        .addListenerForSingleValueEvent(
            object : ValueEventListener {...})
}
```

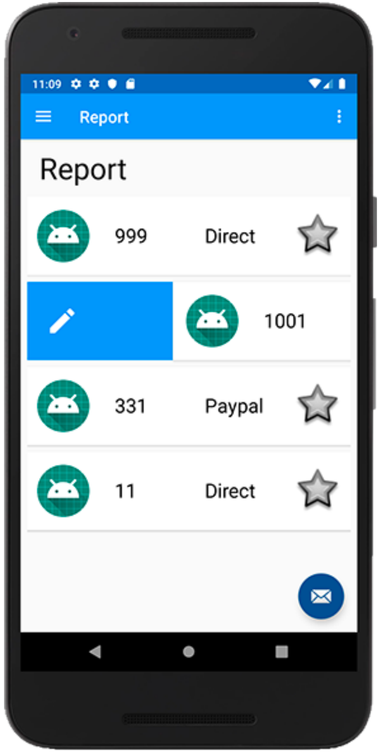
Introduce Access Flow – `Edit`

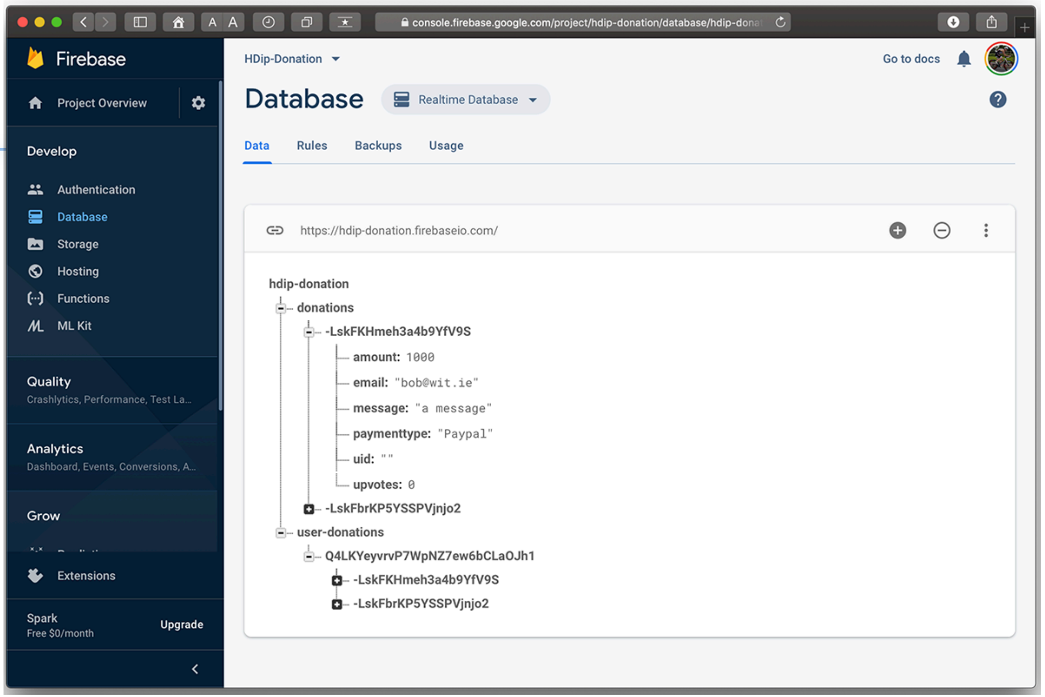


```
fun updateUserDonation(userId: String, uid: String?, donation: DonationModel) {
    app.database.child("user-donations").child(userId).child(uid!!)
        .addListenerForSingleValueEvent(
            object : ValueEventListener {
                override fun onDataChange(snapshot: DataSnapshot) {
                    snapshot.ref.setValue(donation)
                    activity!!.supportFragmentManager.beginTransaction()
                    .replace(R.id.homeFrame, ReportFragment.newInstance())
                    .addToBackStack(null)
                    .commit()
                    hideLoader(loader)
                }

                override fun onCancelled(error: DatabaseError) {...}
            })
}

fun updateDonation(uid: String?, donation: DonationModel) {...}
```

Firebase + Mobile App







```
hdip-donation
  donations
    -LskFKHmeh3a4b9YfV9S
      amount: 1000
      email: "bob@wit.ie"
      message: "a message"
      paymenttype: "Paypal"
      uid: ""
      upvotes: 0
    -LskFbrKP5YSSPVjnjo2
  user-donations
    Q4LKYeyvrvP7WpNZ7ew6bCLaOJh1
      -LskFKHmeh3a4b9YfV9S
      -LskFbrKP5YSSPVjnjo2
```

References

https://console.firebase.google.com/
https://firebase.google.com/
https://firebase.google.com/docs/database/android/lists-of-data
https://github.com/firebase/quickstart-android

Thanks.