

Technische Dokumentation

EVENT-PLAN

„Entwicklung mobiler Anwendungen“

Ausgabedatum: 31.01.2021

Herausgeber: Cedric Tamo Nenkam

Mat. 677984

1. About Quigo

1.1. Motivation

- Ich bin mehrmals spontan gereist und konnte deshalb meine beiden Gepäckstücke von je 23 kg nicht nutzen.
- In unseren verschiedenen WhatsApp-Gruppen gibt es oft Leute, die Pakete von einem Land ins andere schicken wollen, und das sehr dringend.
- Und schließlich kannst du etwas Geld verdienen, um dein Flugticket zu amortisieren.

Quigo löst all diese Probleme. Du reist und hast kein Gepäck, also stell dein Gepäck online, verkaufe es, verdiene ein wenig Geld und hilf anderen Leuten, die dringende Pakete von einem Ort zum anderen schicken wollen.

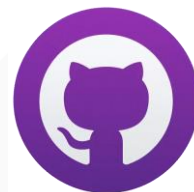
1.2.Tools

Quigo wird auf die Plattform

- Android Studio entwickelt,
- und basiert ganz Komplet auf die Programmiersprache Kotlin,
- Firebase eine Entwicklungs-Plattform für Mobile und Web-Anwendungen von Google
- entwickelt wird auch hier für die Speicherung, Authentifizierung et weiteres verwende
- Das Projekt ist auf GitHub gespeichert

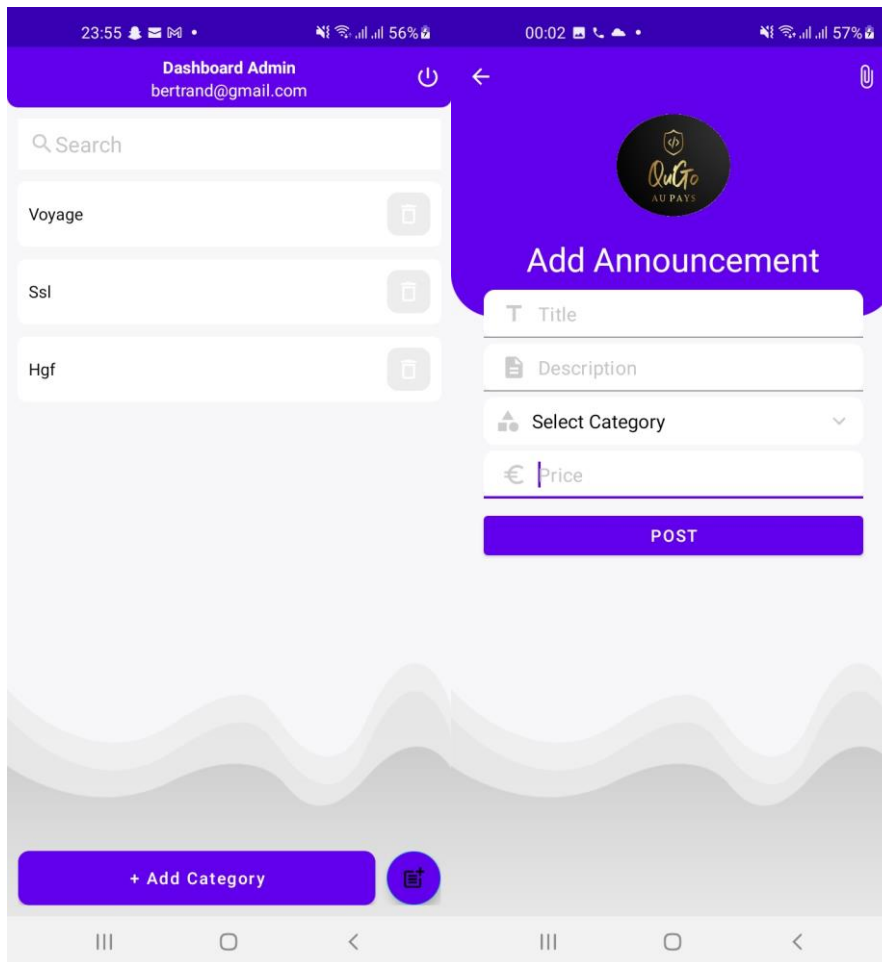


Kotlin



2. Anforderungen

2.1. Multiple Screens



2.2. User Generated Content (User can enter data / content that is stored)

```
private fun addCategoryFirebase() {
    //show Progress
    progressDialog.show()

    //get timestamp
    val timestamp = System.currentTimeMillis()

    //setup data to add in firebase db
    val hashMap = HashMap<String, Any>()
    hashMap["id"] = "$timestamp"
    hashMap["category"] = category
    hashMap["timestamp"] = timestamp
    hashMap["uid"] = "${firebaseAuth.uid}"

    //add to firebase db: Database Root > Categories > categoryId > category info
    val ref = FirebaseDatabase.getInstance().getReference(path: "Categories")
    ref.child(pathString: "$timestamp")
        .setValue(hashMap)
        .addOnSuccessListener { //it Void!
            // added successfully
            progressDialog.dismiss()
            Toast.makeText(context: this, text: "Category Created Successfully...", Toast.LENGTH_SHORT).show()
            binding.categoryEt.setText("")
        }
        .addOnFailureListener { e ->
            //failed to add
            progressDialog.dismiss()
            Toast.makeText(context: this, text: "Failed to add due to ${e.message}", Toast.LENGTH_SHORT).show()
        }
}
```

2.3. App displays a list / grid of objects (recycler view)

```
private fun loadCategories() {
    //init arraylist
    categoryArrayList = ArrayList()

    //get all categories from firebase database... Firebase DB > Categories
    val ref = FirebaseDatabase.getInstance().getReference(path: "Categories")
    ref.addValueEventListener(object: ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            //clear list before starting adding data into it
            categoryArrayList.clear()
            for (ds in snapshot.children) {
                //get data as model
                val model = ds.getValue(ModelCategory::class.java)

                //add to arraylist
                categoryArrayList.add(model!!)
            }
            //setup adapter
            adapterCategory = AdapterCategory(context: this@DashboardAdminActivity, categoryArrayList)
            //set adapter to recyclerView
            binding.categoriesRV.adapter = adapterCategory
        }

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

2.4.App displays some content in a web view (as a HTML page)

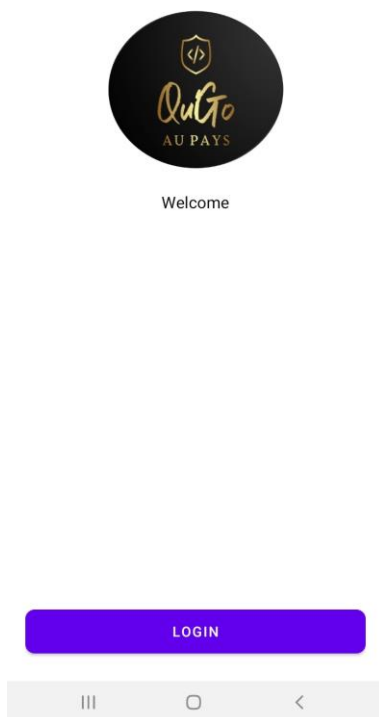
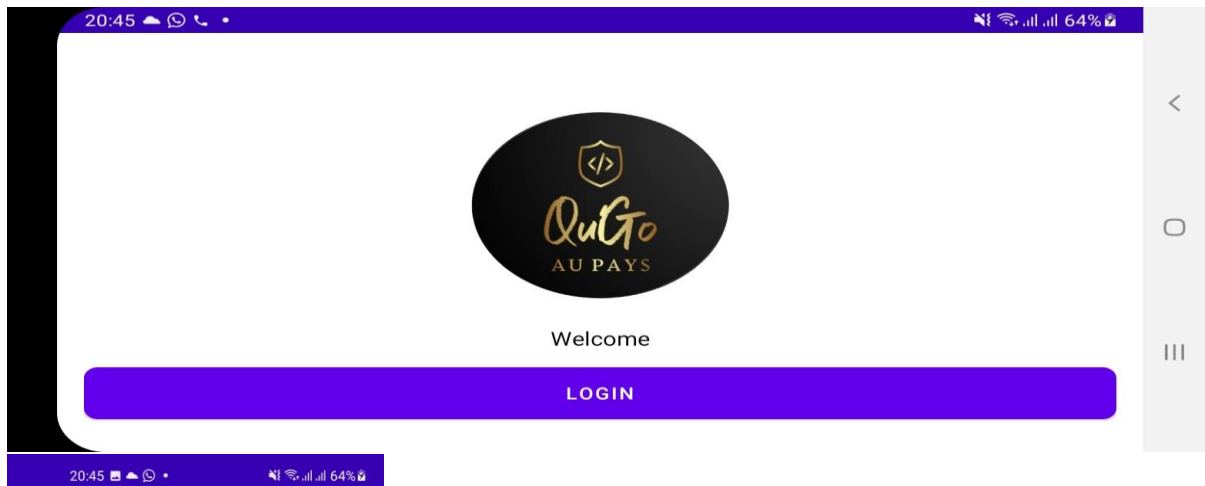
```
//handle click, share post
binding.shareBtn.setOnClickListener {
    val shareBody = "Coming Soon on playStore: https://play.google.com/store/apps?gl=DE"
    val shareSub = postTitle

    val shareIntent = Intent(Intent.ACTION_SEND)
    shareIntent.type = "text/plain"

    shareIntent.putExtra(Intent.EXTRA_SUBJECT, shareSub)
    shareIntent.putExtra(Intent.EXTRA_TEXT, shareBody)

    startActivity(Intent.createChooser(shareIntent, "Share To:"))
}
```

2.5.App works in portrait an landscape mode



2.6.App is working offline (with a local data base)

```
package com.home.quigo

import android.app.Application
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase

class offline: Application() {
    override fun onCreate() {
        super.onCreate()
        Firebase.database.setPersistenceEnabled(true)
    }
}
```

2.7.Unit Tests

```
//specify the signature
fun validateRegistrationInput(
    username: String,
    password: String,
    confirmPassword:String
):Boolean{
    if(username.isEmpty() || password.isEmpty()){
        return false
    }
    if(username in existingUser){
        return false
    }
    if(password != confirmPassword){
        return false
    }
    if(password.length < 6){
        return false
    }
    if(password.count{it.isDigit()}<2){
        return false
    }
    return true
}

package com.home.quigo

object PostUtil {
    /*
     * Test Fails if ...
     * ....Title is empty
     * ....Description is empty
     * ....price is empty
     */
    fun validatePostInput(
        title: String,
        description: String,
        price:String
    ):Boolean{
        if(title.isEmpty() || description.isEmpty() || price.isEmpty()){
            return false
        }
        return true
    }
}
```

2.8. User login (e.g. Firebase Auth)

```
private fun loginUser() {
    // 3) Login - Firebase Auth

    //show progress
    progressDialog.setMessage("Logging In...")
    progressDialog.show()

    //
    firebaseAuth.signInWithEmailAndPassword(email, password)
        .addOnSuccessListener { @AuthResult()
            //Login success
            checkUser()
        }
        .addOnFailureListener { e->
            //Login failed
            progressDialog.dismiss()
            Toast.makeText(context, this, text: "Login failed due to ${e.message}", Toast.LENGTH_SHORT).show()
        }
}

private fun checkUser() {
    // 4) check user type - Firebase Auth and open dashboard accordingly */
    progressDialog.setMessage("Checking User ...")

    val firebaseUser = firebaseAuth.currentUser!!

    val ref = FirebaseDatabase.getInstance().getReference("users")
    ref.child(firebaseUser.uid)
        .addListenerForSingleValueEvent(object : ValueEventListener {


            override fun onDataChange(snapshot: DataSnapshot) {
                progressDialog.dismiss()

                //get user type e.g user or admin
                val userType = snapshot.child("userType").value
                if (userType == "user") {
                    //simple user, open user dashboard
                    startActivity(Intent(context, this@LoginActivity, DashboardActivity::class.java))
                    finish()
                }
                else if (userType == "admin") {
                    //admin, open admin dashboard
                    startActivity(Intent(context, this@LoginActivity, DashboardAdminActivity::class.java))
                    finish()
                }
            }

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

2.9. Animations

```
Handler().postDelayed(Runnable {
    checkUser()
}, delayMillis: 2000) //delay for 2 seconds
}
```



2.10. Background Task

