Projekt Banka

-Programiranje mobilnih uređaja-

Dominik Milinković, 3.e

Elektrotehnička i ekonomska škola Nova Gradiška

Sadržaj

1. [Uvod](#_Uvod) .......................................................................................................................1
2. [Postavljanje radnog mjesta](#_Postavljanje_radnog_mjesta) ....................................................................................1
3. [Main Activity](#_Main_Activity) ..........................................................................................................2
   1. [Korisnik Activity](#_Korisnik_Activity) ..............................................................................................4
      1. [data class korisnici](#_data_class_korisnici) ...............................................................................7
      2. [KorisnikAdapter](#_KorisnikAdapter) ...................................................................................7
      3. [TekstiItem](#_TekstiItem) ............................................................................................8
   2. [Transakcija aktivnost](#_Transakcija_aktivnost) ......................................................................................9
      1. [data class Transa](#_data_class_Transa) .................................................................................11
      2. [TransAdapter](#_TransAdapter) ......................................................................................12
      3. [ItemTrans](#_ItemTrans) ..........................................................................................13

# Uvod

Zadatak je modelirati banku s korisnicima, računima i transakcijama koristeći klase, FireBase, RecyclerView.

# Postavljanje radnog mjesta

Kada otvorimo Android Studio i odaberemo Empty View Activity, treba odabrati ime i package name za naš projekt (npr. hr.prezime.imeProjekta).

Na internet pregledniku otvorimo FireBase, ulogiramo se i u kreiranju projekta prvo unesemo ime projekta i kliknemo na dalje, pa udaberemo Default Account for Firebase i kliknemo na CreateProject.

Kada se završi, dobit ćemo stranicu gdje piše Get started i udaberemo ikonu Androida.

Onda će nas prvo tražiti Package Name projekta koje smo odabrali pri kreiranju projekta unutar Android Studia i SHA-1 kod.

Kliknemo na Gradle Bar s desne strane i kliknemo na Run kako bismo upisali signingReport i onda će nam se ispisati nekoliko podataka i potražimo gdje piše SHA1 i kopiramo kod i zalijepimo ga unutar FireBase-a i kliknemo Register App.

Zatim preuzmemo google-services.json i zalijepimo unutar Android Studia. Treba otići na Project karticu i kliknuti na ime našeg projekta kako bi se razgranatalo i unutar app zalijepiti .json.

Nakon toga, unutar Tools kartice kliknuti na FireBase i otvorit će se Assistant gdje potražimo karticu Realtime Database i odaberemo Get started with Realtime Database.

Aplikacija je već povezana sa FireBase, stoga idemo na drugi korak koji je dodati Realtime Databasr SDK u našu aplikaciju. Samo kliknemo na Add i prihvatimo Accept Changes

Unutar build.gradle treba upisati buildFeatures { viewBinding true} i Sync now koristi se kako bi se omogućila funkcionalnost View Bindinda.

# Main Activity

A screen shot of a computer program

Description automatically generated

XML kod koji prikazuje početnu stranicu aplikacije na kojoj se nalaze 2 button i 1 RecyclerView.

U prvom buttonu (korisnik) tekst „Korisnik“ će biti veličine 20sp. Širina je postavljena na 0dp jer želim da taj element popuni dostupan prostor, a visina je wrap content što znači da će visina biti onolika koliko je potrebno da se prikaže sadržaj unutar elementa,

Constraintao sam vrh buttona na vrh parenta s razmakom (marginom) od 10dp i constraintao sam lijevi dio buttona s lijevim djelom parenta s marginom od 20dp.

Drugom buttonu će tekst „Transakcije“ također biti 20sp, a ista mu je i visina i širina i constraint na vrh parenta kao prvom buttonu. Razlika je što je lijevi dio constraintan na desni dio prvog button s marginom od 25dp i drugaciji je naravno id jer svaki id mora biti jedinstven.

RecyclerView-u su širina i visina jednako postavljeni kao i na oba buttona. Vrh RecyclerView-a sam constraintao na dno buttona korisnik s marginom od 20dp i marginom od lijevog dijela parenta s marginom 20dp. Scrollbars je postavljen vertikalno što znači da će se moći skrloati prema dolje.

A screen shot of a computer program

Description automatically generated

Kotlin kod

lateinit var binding: ActivityMainBinding – ova linija deklarira varijablu binding tipa ActivityMainBinding. lateinit se koristi za varijable koje neće biti inicijalizirane prilikom stvaranja instance klase, nego će se kasnije inicijalizirati u metodi OnCreate

binding=Activity... setContentView – inicijalizira se binding varijabla i postavlja View za aktivnost ActivityMainBinding.inflate(layoutInflater) stvara instancuActivityMainBinding klase za trenutnu aktivnost, setContentView postavlja View na pogled koji je vraćen iz binding.root

binding.korisnik/transakcije.setOnClickListener – kada se klikne na gumb (kojemu je id ili korisnik ili transakcije), stvara se Intent za pokretanje nove aktivnosti (ili korisnik ili transakcije)

# Korisnik Activity

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Kod u XML-u za korisnika nakon što smo u Main Activity-u kliknuli na button korisnik.

Ovaj layout ima 3 EditTexta (za ime, prezimei broj mobitela), 1 DatePicker (za datum rodjenja), 1 button stvori koji kad kliknemo se korisnik treba stvoriti u RecyclerView ispod button.

Padding – predstavlja unutarnji razmak elementa na 15dp. Unutarnji razmak je prostor između granice elementa i njegovog sadržaja

Za EditText brMob koristim inputType:Phone jer mi je potrebno da se unose samo brojevi za broj mobitela, a maxLength je postavljen na 10 što znači da ne može biti više od 10 znamenki jer broj mobitela ima 10 znamenki

Za DatePicker stavljeno je datePickerMode=“spinner“ što znači da će se vrtnjom posebno odabirati dan, mjesec i godina, a calenderViewShown može biti ili true ili false, ako je true, onda će kalendar biti prikazan ispod spinnera, a ovdje je stavljeno false jer ne želim da bude prikazan kalendar.

Kada se klikne na button Stvori, korisnik s imenom, prezimenom, brojem mobitela i datumom rođenja treba biti prikazan u RecyclerView-u ispod buttona.

A screenshot of a computer program

Description automatically generated

Kotlin kod

private val database... – ova linija stvara referencu na Firebase bazu podataka, Firebase.Database.getInstance() vraća instancu Firebase baze podataka, a getReference vraća referencu (u ovom slučaju) Korisnici čvor u bazi podataka.

private var korisnik1... – ova linija deklarira varijablu korisnik1 kao ArrayList objekta klase (u ovom slučaju) korisnici

binding.stvori... – postavlja OnClickListener za button stvori. Kada se klikne na button, stvara se novi korisnici objekt s podatcima unesenim od strane korisnika i dodaje se u korisnik1 listu. Zatim se cijela lista korisnik1 postavlja kao vrijednost Korisnici čvora u Firebase bazi podataka.

database.addValueEventListener... – postavlja ValueEventListener na Korisnici čvor u Firebase-u, kada se podatci na tom čvoru promijene, metoda onDataChange se poziva s DataSnapshot objektom koji sadrži trenutne podatke. U toj metodi se korisnik1 lista prvo briše, a zatim popunjava novim podatcima iz baze podataka i onda se postavlja layoutManager i adapter za RecyclerView RV.

# data class korisnici

A screen shot of a computer

Description automatically generated

Definira se nova data class pod nazivom korisnici

Glavna joj je svrha držanje podataka

var znači varijabla (npr. var ime: String – deklarira se varijabla ime tipa string)

# KorisnikAdapter

A computer screen shot of a program

Description automatically generated

Definira se nova klasa korisnikAdapter koja nasljeđuje RecyclerView.Adapter.

RecyclerView.Adapter je apstraktna klasa koja služi kao most između RecyclerView i podataka koje RecyclerView prikazuje.

korisnikAdapter prima 2 argumenta: textList, koji je ArrayList objekata klase korisnici, i th, koji je Context objekt.

Metoda onCreateViewHolder se poziva kada RecyclerView treba novi ViewHolder za prikaz stavke. Ova metoda stvara novi ViewHolder koristeći TekstiItemBinding.inflate za inflaciju stavke izgleda.

Metoda onBindViewHolder se poziva za ažuriranje ViewHolder s podatcima za određenu poziciju. Ova metoda poziva bindItem na ViewHolder za vezanje podataka za korisnika na trenutnoj poziciji.

Metoda getItemCount vraća ukupan broj stavki u podatcima u setu. Ova metoda vraća veličinu textList.

Definira se i unutarnja klasa ViewHolder koja nasljeđuje RecyclerView.ViewHolder. ViewHolder opisuje stavku View-a i podatke o njenom mjestu untar RecyclerView. bindItem je metoda koja veže podatke za korisnika na stavku pogleda.

# TekstiItem

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Visina je postavljena na wrap content i odnosi se na parenta.

U korisnikAdapter metoda onCreateViewHolder stvara novi ViewHolder koristeći TekstiItemBinding.inflate za inflaciju stavke izgleda.

# Transakcija aktivnost

XML kod

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Kotlin kod

private val database... – ova linija stvara referencu na Firebase bazu podataka, Firebase.Database.getInstance() vraća instancu Firebase baze podataka, a getReference vraća referencu (u ovom slučaju) Transakcije čvor u bazi podataka.

private var tekst1... – ova linija deklarira varijablu tekst1 kao ArrayList objekta klase (u ovom slučaju) Transa

binding.prenesi... – postavlja OnClickListener za button Prenesi. Kada se klikne na button, stvara se novi Transa objekt s podatcima unesenim od strane korisnika i dodaje se u tekst1 listu. Zatim se cijela lista tekst1 postavlja kao vrijednost Transakcije čvora u Firebase bazi podataka.

database.addValueEventListener... – postavlja ValueEventListener na Transakcije čvor u Firebase-u, kada se podatci na tom čvoru promijene, metoda onDataChange se poziva s DataSnapshot objektom koji sadrži trenutne podatke. U toj metodi se tekst1 lista prvo briše, a zatim popunjava novim podatcima iz baze podataka i onda se postavlja layoutManager i adapter za RecyclerView transakcijee.

# data class Transa

A screen shot of a computer

Description automatically generated

Definira se nova data class pod nazivom Transa

Glavna joj je svrha držanje podataka

# TransAdapter

A computer screen shot of a program

Description automatically generated

Definira se nova klasa TransAdapter koja nasljeđuje RecyclerView.Adapter.

RecyclerView.Adapter je apstraktna klasa koja služi kao most između RecyclerView i podataka koje RecyclerView prikazuje.

TransAdapter prima 2 argumenta: textList, koji je ArrayList objekata klase Transa, i th, koji je Context objekt.

Metoda onCreateViewHolder se poziva kada RecyclerView treba novi ViewHolder za prikaz stavke. Ova metoda stvara novi ViewHolder koristeći ItemTransBinding.inflate za inflaciju stavke izgleda.

Metoda onBindViewHolder se poziva za ažuriranje ViewHolder s podatcima za određenu poziciju. Ova metoda poziva bindItem na ViewHolder za vezanje podataka za transakciju na trenutnoj poziciji.

Metoda getItemCount vraća ukupan broj stavki u podatcima u setu. Ova metoda vraća veličinu textList.

Definira se i unutarnja klasa ViewHolder koja nasljeđuje RecyclerView.ViewHolder. ViewHolder opisuje stavku View-a i podatke o njenom mjestu untar RecyclerView. bindItem je metoda koja veže podatke za transakciju na stavku pogleda.

# ItemTrans

A screen shot of a computer

Description automatically generated

Visina je postavljena na wrap content i odnosi se na parenta.

U TransAdapter metoda onCreateViewHolder stvara novi ViewHolder koristeći ItemTransBinding.inflate za inflaciju stavke izgleda.