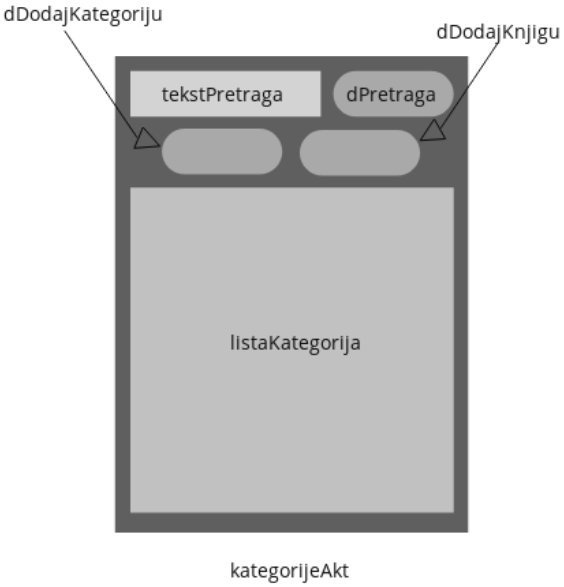
Projekat iz Razvoja mobilnih aplikacija

Spirala 1

Cilj projekta je da se razvije aplikacija za upravljanje listom pročitanih knjiga, planiranje čitanja knjiga i vođenja kratkih napomena o knjigama. Razvoj projekta će biti spiralan tako da se svaka spirala nastavlja na prethodnu. Niz funkcionalnosti koje aplikacija ima će vremenom rasti i sa posljednjom spiralom bi se trebala dobiti upotrebljiva mobilna aplikacija.

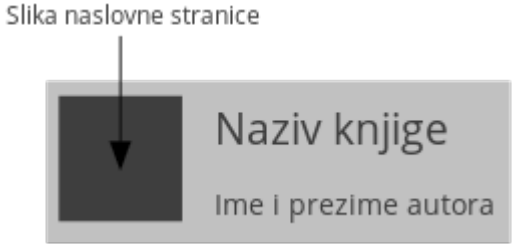
1.) **[2 boda]** Početna aktivnost sa nazivom ***KategorijeAkt*** treba da sadrži edittext sa id-em ***tekstPretraga***, listview sa id-em ***listaKategorija***, tri dugmeta sa id-evima ***dPretraga***, ***dDodajKategoriju*** i ***dDodajKnjigu***. Layout neka izgleda kao na *Slici 1*. Dugme *dDodajKategoriju* treba biti onemogućeno (*setEnabled(false)*). Kada korisnik unese tekst u polje i pritisne dugme *dPretraga* lista se treba filtrirati, filtriranje radite sa metodama *adapter.getFilter().filter(string)*. Ukoliko je rezultat prazna lista dugme dDodajKategoriju se treba omogućiti. Kada korisnik klikne na dugme *dDodajKategoriju* uneseni tekst se treba dodati u listu.



Slika 1.

2.) **[2 boda]** Napravite aktivnost sa nazivom ***DodavanjeKnjigeAkt*** koja treba da sadrži imageView sa id-em ***naslovnaStr***, editText sa id-em ***imeAutora***, editText sa id-em ***nazivKnjige***, dugme sa id-em ***dNadjiSliku***, dugme sa id-em ***dUpisiKnjigu*** i spinner u kojem biramo kojoj kategoriji knjiga pripada sa id-em ***sKategorijaKnjige***. Spinner popunite sa podacima o kategorijama iz prvog zadatka. Ova aktivnost se treba prikazati kada korisnik klikne na dugme *dDodajKnjigu* iz prošlog zadatka. Kada se klikne na dugme *dUpisiKnjigu* u niz trebate dodati knjigu sa unesenim podacima. Implementirajte i dugme ***dPonisti*** koje vraća korisnika na početnu aktivnost bez dodavanja nove knjige. Dizajn layouta ove aktivnosti uradite samostalno.

3.) **[3 boda]** Kada korisnik klikne na element liste iz zadatka 1 treba se otvoriti aktivnost ***ListaKnjigaAkt***. Ova aktivnost treba da sadrži listView sa id-em ***listaKnjiga*** i u njoj trebate prikazati sve knjige koje pripadaju odabranoj kategoriji. Implementirajte i dugme ***dPovratak*** koje će vratiti korisnika na početnu aktivnost. Layout elementa liste treba biti kao na Slici 2. Id-evi widget-a elementa liste neka budu ***eNaslovna*** za sliku naslovne stranice, ***eNaziv*** za naziv knjige i ***eAutor*** za ime i prezime autora.



Slika 2.

4.) **[2 boda]** U zadatak 2 dodajte funkcionalnost koja će otvoriti dijalog za odabir slike kada korisnik klikne na dugme *dNadjiSliku*. Koristite implicitni intent i akciju *Intent.ACTION_GET_CONTENT*. Kada korisnik odabere sliku prikažite je u *naslovnaStr*. *Ukoliko hardkodirate nekoliko različitih slika, bez da ponudite mogućnost korisniku da odabere koju sliku želi, možete dobiti maksimalno 0.5/2 boda na ovaj zadatak. Slika se treba prikazivati i u elementu listaKnjiga.*

5.) **[1 bod]** Kada korisnik klikne na knjigu iz liste *listaKnjiga* taj element liste obojite svijetlo plavom bojom (0xffaabbed). Boja treba ostati zapamćena i ako se vratite na početnu aktivnost.

Napomena: *Rok za rad na spirali 1 je 31.03.2018 u 23:59. Vaš projekat se treba nalaziti na Bitbucket repozitoriju do navedenog roka. Svi projekti koji nisu pravilno postavljeni, nepotpuni ili nepravovremeni će se bodovati sa 0 bodova. Uputstva za postavljanje projekta će biti na c2 stranici predmeta.*

Projekat iz Razvoja mobilnih aplikacija

Spirala 2

Prepravite aplikaciju iz prethodne spirale tako da sada umjesto više aktivnosti sada koristite jednu aktivnost i više fragmenata.

Zadatak 1. [3 boda] Napravite fragment **ListeFragment** koji sadrži layout kao početna aktivnost iz prethodne spirale. Pored ovog layouta dodajte još dva dugmeta na vrhu layouta. Jedno dugme sa id-em **dKategorije**, a drugo sa id-em **dAutori**. Kada se klikne na dugme *dKategorije* u listview elementu se učitavaju kategorije (kao u spirali 1), u ovom slučaju dugmadi *dDodajKategoriju, dPretraga i dDodajKnjigu* se ponašaju kao u spirali 1. Kada se klikne na dugme *dDodajKnjigu* otvara se fragment ***DodavanjeKnjigeFragment*** koji sadrži isti layout kao aktivnost *DodavanjeKnjigeAkt* stim da dugme *dPonisti* sada vraća na fragment ListeFragment. Kada se klikne na dugme *dAutori* u listview se trebaju učitati autori knjiga koje su dodane, bez ponavljanja (ako je neki autor napisao više knjiga). Element liste u slučaju autora sadrži ime i prezime autora i broj knjiga koje je taj autor napisao. Dugmadi *dPretraga* i *dDodajKategoriju* i polje *tekstPretraga* se trebaju sakriti sa *setVisibility(View.GONE)*.

Zadatak 2. [2.5 boda] Napravite fragment **KnjigeFragment** koji sadrži layout kao aktivnost *ListaKnjigaAkt*. U početnoj aktivnosti na početku neka se prikazuje fragment *ListeFragment* kada korisnik klikne na neki od elemenata iz liste zamjenite fragment *ListeFragment* sa fragmentom *KnjigeFragment* u kojem ćete prikazati knjige iz kategorije ili knjige autora. Dugme *dPovratak* zamjenjuje nazad fragment *KnjigeFragment* sa fragmentom *ListeFragment.*

Zadatak 3. [3 boda] Napravite novi layout za ekrane šire od 450dp u kojem će se prikazivati uporedo *ListeFragment* i *KnjigeFragment* po principu master/detail flow-a. Kada se klikne na dugme dDodajKnjigu otvorite fragment *DodavanjeKnjigeFragment* preko cijelog layouta.

Zadatak 4. [1.5 bod]

a) Napravite prevod svih labela na engleski i bosanski jezik
b) Sve boje zapišite u values/colors.xml i koristite ih iz ovog resursa

Napomena: *Rok za rad na spirali 2 je* **11.04.2018 do 23:59**. *Spirala se treba nalaziti na istom repozitoriju kao i prethodna spirala. Novu spiralu postavite na master branch, a prethodnu spiralu postavite na branch spirala1.*

Projekat iz Razvoja mobilnih aplikacija

Spirala 3

Prije implementacije zadataka implementirajte sljedeće:

- Klasa **Knjiga** treba da sadrži sljedeće atribute: **id** *tipa string,* **naziv** *tipa string*, **autori** *tipa ArrayList<Autor>*, **opis** tipa string, **datumObjavljivanja** *tipa string*, **slika** *tipa URL i* **brojStrinica** *tipa int* (klasa treba da ima konstruktor koji prima parametre u istom redosljedu kako su atributi nabrojani, te gettere i settere za sve atribute npr. getNaziv i setNaziv)
- Klasa **Autor** treba da sadrži sljedeće atribute: **imeiPrezime** *tipa string* i **knjige** *tipa ArrayList<string>* (klasa treba da ima konstruktor koji prima imeiPrezime autora te id knjige koji će se dodati u niz knjige, potrebno je implementirati gettere i settere kao i metodu **dodajKnjigu(string id)** koja dodaje dati id u listu knjige (ako već ne postoji)
- Prethodne klase obavezno trebaju imati navedene atribute, ali niste ograničeni na samo ove atribute. Tj. možete imati i dodatne atribute ako vam je lakše da realizujete zadatke, ali ne smijete mijenjati konstruktore koji su navedeni ranije (možete dodati nove konstruktore).

Zadatak a. Implementirajte klasu **DohvatiKnjige** koja je nasljeđena iz AsyncTask klase. Kao parametre za izvršavanje zadatka ova klasa prima nazive knjiga. Zadatak je da prođete kroz sve nazive knjiga i dohvatite listu knjiga koje imaju dati naziv. Koristite Google Books API i sljedeću metodu web servisa https://www.googleapis.com/books/v1/volumes?q=intitle:Naziv. Za svaki prosljeđeni naziv trebate pozvati web servis, a na kraju formirajte listu svih knjiga koje ste skupili sa ovog web servisa. Svaki put postavite da servis vrati najviše 5 rezultata (koristeći parametar maxResults). Konstruktor DohvatiKnjige prima jedan parametar, a to je implementacija interfejsa **IDohvatiKnjigeDone** koji ima samo jednu metodu **onDohvatiDone(ArrayList<Knjige>)**. Nakon što skupite sve knjige pozovite metodu onDohvatiDone. Ovaj interfejs neka bude deklarisan unutar klase DohvatiKnjige (kao interfejs OnMuzicarSearchDone iz vježbe 4) i koristite ga za komunikaciju sa fragmentom koji će pozivati DohvatiKnjige.

Zadatak b. Implementirajte klasu **DohvatiNajnovije** koja je nasljeđena iz AsyncTask klase. Kao parametre za izvršavanje zadatka ova klasa prima ime i prezime autora. Zadatak je da koristeći isti web servis kao u zadatku 1 pronađete zadnjih 5 najnovijih knjiga koje je dati autor napisao. Konstruktor DohvatiNajnovije prima jedan parametar, a to je implementacija interfejsa **IDohvatiNajnovijeDone** koji ima samo jednu metodu **onNajnovijeDone(ArrayList<Knjige>)**. Nakon što dobijete odgovor od web servisa pozovite metodu onNajnovijeDone sa knjigama koje je web servis vratio. Ovaj interfejs neka bude deklarisan unutar klase DohvatiNajnovije i koristite ga za komunikaciju sa fragmentom koji će pozivati DohvatiNajnovije.

Zadatak c. Implementirajte IntentService **KnjigePoznanika**. Ovaj servis u intent extra podacima prima sljedeće: **idKorisnika** *tipa string* i **receiver** *tipa ResultReceiver.* Kada servis primi intent poziva Google Books API web servis https://developers.google.com/books/docs/v1/using#WorkingBookshelves. Kao rezultat KnjigePoznanika reveiver-u treba vratiti listu svih knjiga, sa svih bookshelves koji su javni za korisnika sa datim id-em (userId je isto što i idKorisnika). Unutar klase definišite tri statusa: STATUS_START sa vrijednosti 0, STATUS_FINISH sa vrijednosti 1 i STATUS_ERROR sa vrijednosti 2. Prije pozivanja web servisa receiveru pošaljite STATUS_START sa Bundle.EMPTY, nakon uspješno obrađenog odgovora od web servisa receiveru pošaljite

STATUS_FINISH sa Bundle koji ima podatak **listaKnjiga** koja je parcelableArrayList sa knjigama koje ste dobili pozivajući web servis, a ako je došlo do greške pošaljite STATUS_ERROR sa Bundle.EMPTY.

Zadatak U početnoj aktivnosti (KategorijeAkt) dodajte dugme sa id-em **dDodajOnline** sa tekstom *dodaj knjigu online*. Klik na ovo dugme otvara fragment **FragmentOnline**. FragmentOnline sadrži spinner sa kategorijama koji ima id **sKategorije**, edit text sa **id-em tekstUpit,** spinner **sRezultat,** dugme sa id-em **dRun** i tekstom *Pretraga*, dugme sa id-em **dAdd** i tekstom dodaj knjigu i dugme sa id-em **dPovratak** koje vraća nazad na početni izgled aplikacije. Klik na dugme dRun radi sljedeće:

a.1) [1 bod] Ako je u tekstUpit upisana jedna riječ pozovite DohvatiKnjige sa datom riječi

a.2) [2 boda] Ako je u tekstUpit upisano više stringova koji su odvojeni sa ; pozovite DohvatiKnjige gdje svaki string predstavlja jedan naziv knjige

b) [3 boda] Ako je u tekstUpit upisano autor:Neko Ime. Pozovite DohvatiNajnovije gdje je parametar string koji slijedi nakon ‘autor:’

c) [4 boda] Ako je u tekstUpit upisano korisnik:nekiUserId. Pozovite KnjigePoznanika gdje je parametar string koji slijedi nakon ‘korisnik:’

Nakon što se završi pretraga u spinner sRezultat upišite nazive knjiga koje je web servis vratio. Klik na dugme dAdd dodaje knjigu koja je odabrana u spinneru sRezultat u kategoriju koja je odabrana u spinneru sKategorije. Sliku za ove knjige zasada hardkodirajte. Knjige koje web servis vrati moraju imati sve podatke popunjene (atribut slika popunite sa vrijednosti iz imageLinks.thumbnail).

Napomena: *Rok za rad na spirali 3 je* **22.05.2018 do 23:59**. *Spirala se treba nalaziti na istom repozitoriju kao i prethodna spirala. Novu spiralu postavite na master branch, a prethodnu spiralu postavite na branch spirala2.*

Projekat iz Razvoja mobilnih aplikacija

Spirala 4

Zadatak 1. [2 boda] Element liste listaKnjiga ispravite tako da ima sljedeća polja:
TextView sa id-em **eDatumObjavljivanja**
TextView sa id-em **eOpis**
TextView sa id-em **eBrojStranica**
Polja popunite sa vrijednostima koje ste dobili iz web servisa, a za sliku naslovne stranice uzmite sliku sa url-a. Za prikazivanje slike možete koristiti http://square.github.io/picasso/

Zadatak 2. [3 boda] U element liste knjiga dodajte i dugme dPreporuci. Klik na ovo dugme otvara novi fragment **FragmentPreporuci** u kojem se prikazuje spinner **sKontakti** sa listom emailova svih kontakata i podaci o knjizi koja je preporučena, kao i dugme **dPosalji**. Klik na dPosalji šalje email sa sljedećim tekstom:

Zdravo** {IME}**,
Pročitaj knjigu** {IME KNJIGE} **od** {IME AUTORA}**!

Email se treba poslati na odabrani email, {IME} odgovara imenu odabranog kontakta, {IME KNJIGE} je naziv knjige koju preporučujete i {IME AUTORA} je ime prvog autora knjige.

Zadatak 3. [5 bodova]
Napravite BazaOpenHelper koji nasljeđuje SQLiteOpenHelper. Kreirajte bazu sa sljedećim tabelama:

- **Kategorija** sa dvije kolone: **_id** - primarni ključ i kolona **naziv** tipa TEXT
- **Knjiga** sa kolonama: **_id** - primarni ključ, **naziv** tipa TEXT, **opis** tipa TEXT, **datumObjavljivanja** tipa TEXT, **brojStranica** tipa INTEGER, **idWebServis** tipa TEXT koja predstavlja id sa web servisa, **idkategorije** tipa INTEGER
- **Autor** sa kolonama: **_id** - primarni ključ, **ime** tipa TEXT
- **Autorstvo** sa kolonama: **_id** - primarni ključ, **idautora** tipa INTEGER i **idknjige** tipa INTEGER

a) [1 bod] Pri dodavanju nove kategorije upišite je u tabelu i iskoristite podatke iz tabele Kategorija za prikaz liste kategorija, napravite metodu unutar ove klase ***long dodajKategoriju(String naziv)*** - vraća id dodane kategorije ili -1 ako je došlo do greške.
Napomena: ista kategorija se ne smije ponavljati više puta u tabeli Kategorija

b) [1.5 bod] Pri dodavanju nove knjige upišite je u tabelu Knjiga, ukoliko autor ne postoji u tabeli Autor dodajte ga. U tabelu Autorstvo upišite id autora i id knjige. Ukoliko knjiga ima više autora ponovite za sve autore. Napravite metodu ***long dodajKnjigu(Knjiga knjiga)*** - vraća id dodane knjige ili -1 ako je došlo do greške.
Napomena: isti autor se ne smije više puta pojavljivati u tabeli Autor, isto vrijedi i za knjigu

c) [1 bod] Kada se klikne na neku kategoriju napravite upit koji će vratiti knjige date kategorije iz baze i rezultat iskoristite pri prikazivanju liste knjiga kategorije. Napravite metodu ***ArrayList<Knjiga> knjigeKategorije(long idKategorije)***

d) [1.5 bod] Kada se klikne na nekog autora napravite upit koji će vratiti knjige datog autora iz baze i rezultat iskoristite pri prikazivanju knjiga autora. Napravite metodu ***ArrayList<Knjiga> knjigeAutora(long idAutora)***

Napomena: *Rok za rad na spirali 4 je* **30.05.2018 do 23:59**. *Spirala se treba nalaziti na istom repozitoriju kao i prethodna spirala. Novu spiralu postavite na master branch, a prethodnu spiralu postavite na branch spirala3.*