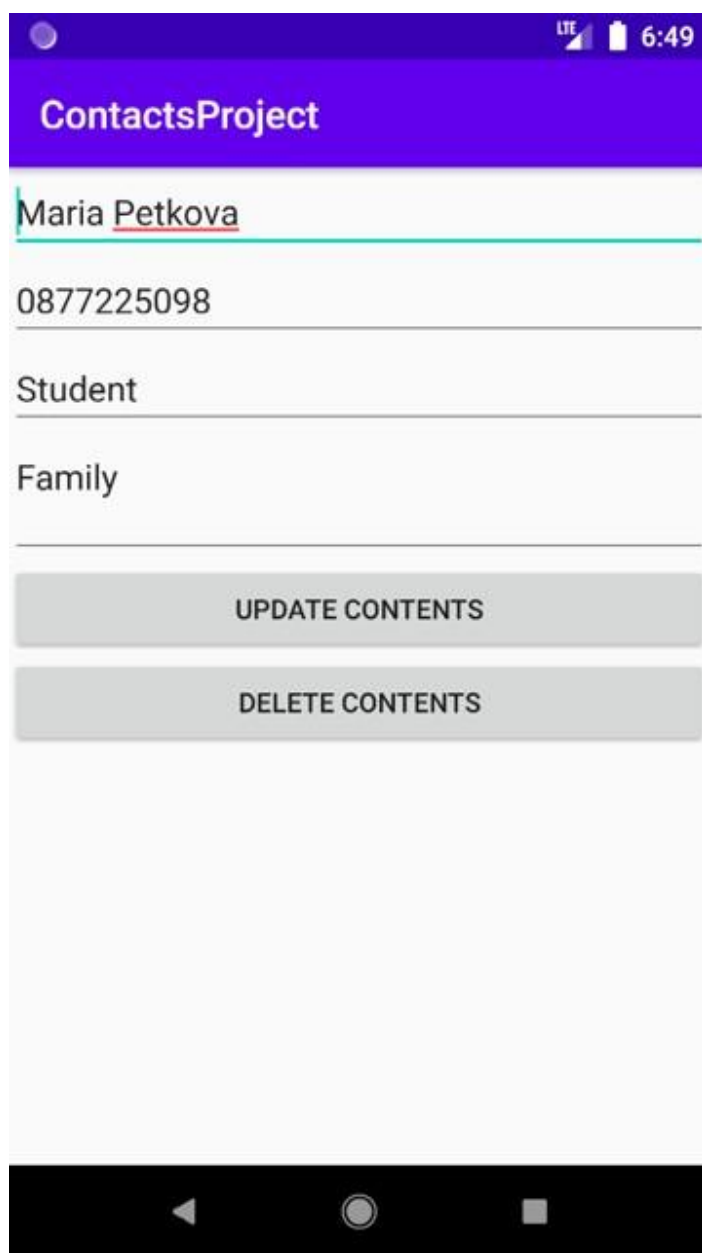


# ДОКУМЕНТАЦИЯ

## КУРСОВА РАБОТА ПО МОБИЛНИ ПРИЛОЖЕНИЯ

### „Телефонен указател“

Мария Петкова, Софтуерно инженерство, 3-ти курс, РБ  
1701321002



## КРАТКО ОПИСАНИЕ НА ПРОЕКТА

За цел на проекта избрах първа тема от предоставения ни списък с примерни теми за курсов проект:

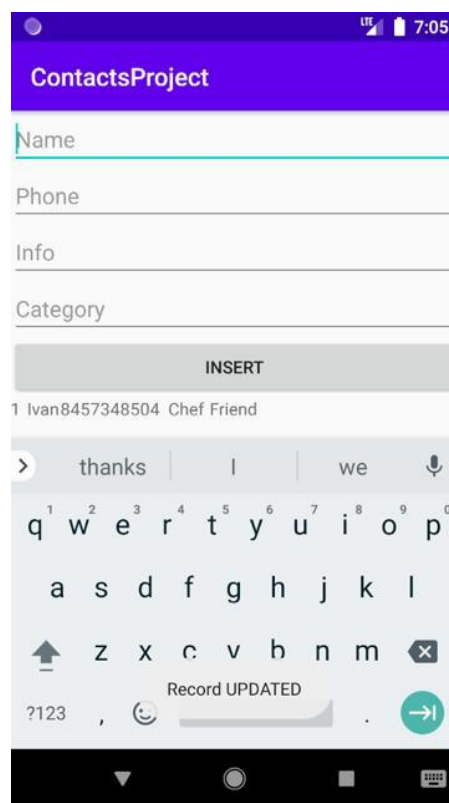
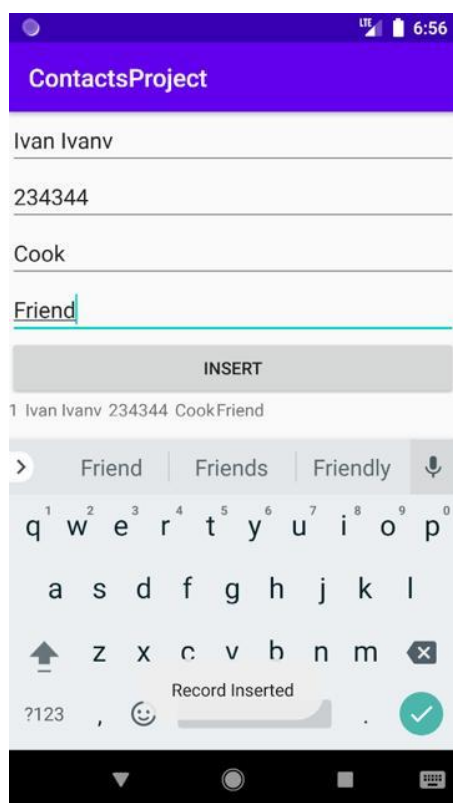
Да се създаде мобилно приложение, което:

1. Поддържа локална БД с телефонни номера. За всеки запис се пази: Име, телефонен номер, допълнително описание на контакта, категория потребител (познати, колеги, семейство, др.). Да има функционалност за преглед, редакция и добавяне на контакт.

За създаването на нов запис са необходими следните полета:

1. Име / имена на контакт;
2. Телефонен номер;
3. Допълнителна информация – например професия;
4. Категория – приятели, семейство, колеги;

След като данните бъдат въведени, следва натискане на бутона “INSERT”, намиращ се точно под текстовите полета. Записът ще се появи под бутона и ще бъде подреден в същата последователност както са подредени полетата → име – телефонен номер – допълнителна информация – категория. Ще бъде изведено кратко съобщение в дъното на екрана, щом записът е създаден и добавен в базата.



Ако потребителят желае да промени вече въведен запис, е достатъчно да натисне върху самия запис. Ще бъде отворена форма, подобна на началната, която предоставя възможност за редактиране на съществуващ запис и на изтриване на такъв.

Достатъчно е да се въведат нови данни и да се натисне бутона UPDATE. Потребителят отново ще получи съобщение за успешно редактиран запис. В примера се вижда редактирана информация за записа (полето Info).

При натискане на бутона Delete, потребителят получава съобщения за успешно премахнат запис. Опрецията изтриване е безвъзвратна.

# НАРЪЧНИК ЗА ПРОГРАМИСТА

Проектът се състои от 3 activity-та – главно (Main) Activity, UpdateActivity и ListView Activity.

В главното Activity се създава базата данни и си инициализират текстовите полета, в които се въвеждат данните. Тук е имплементирана операцията по добавяне на нов запис като директна заявка към вече готовата база.

```
protected EditText editName, editTel, editInfo, editCategory;
protected Button btnInsert;
// protected TextView viewResults;
protected ListView simpleList;

protected void initDB() throws SQLException {
    SQLiteDatabase db=null;
    db=SQLiteDatabase.openOrCreateDatabase(
        path: getFilesDir().getPath()+"/contacts.db",
        factory: null
    );
    String q="CREATE TABLE if not exists CONTACTS( ";
    q+="ID integer primary key AUTOINCREMENT, ";
    q+="name text not null, ";
    q+="tel text not null, ";
    q+="info text not null, ";
    q+="category text not null, ";
    q+="unique(name, tel) ); ";
    db.execSQL(q);
    db.close();
}
```

В същото activity има и два EventListener-a – единият следи за натискане на бутона Insert, а другият следи за клик върху вече въведените данни. Вторият отваря панела за Update и Delete.

```

btnInsert.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SQLiteDatabase db=null;
        try{
            db=SQLiteDatabase.openOrCreateDatabase(
                path: getFilesDir().getPath()+"/contacts.db",
                factory: null

            );
            String name=editName.getText().toString();
            String tel=editTel.getText().toString();
            String info=editInfo.getText().toString();
            String category=editCategory.getText().toString();
            String s="INSERT INTO CONTACTS(name, tel, info, category) ";
            s+="VALUES(?, ?, ?, ?)";
            db.execSQL(s, new Object[]{name, tel, info, category});
            Toast.makeText(getApplicationContext(),
                text: "Record Inserted",
                Toast.LENGTH_LONG

```

В Update Activity се намира логиката, която стои зад бутоните „UPDATE” и „DELETE“. В първия случай, се пуска нова заявка към вече готовата база за намиране на запис по ID в базата и за пренаписването му.

```

btnUpdate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SQLiteDatabase db=null;
        try{
            db=SQLiteDatabase.openOrCreateDatabase(
                path: getFilesDir().getPath()+"/contacts.db",
                factory: null

            );
            String name=euName.getText().toString();
            String tel=euTel.getText().toString();
            String info=euInfo.getText().toString();
            String category=euCategory.getText().toString();
            String s="UPDATE CONTACTS SET name=?, tel=?, info=?, category=? WHERE ";
            s+="ID=? ";

            db.execSQL(s, new Object[]{name, tel, info, category, ID});
            Toast.makeText(getApplicationContext(),
                text: "Record UPDATED",
                Toast.LENGTH_LONG
            ).show();
        }catch (Exception e){
            Toast.makeText(getApplicationContext(),
                e.getMessage(),
                Toast.LENGTH_LONG
            ).show();
        }finally {

```

При бутона за изтриване отново търсенето става по ID, като този път намереният запис бива изтрит.

```
btnDelete.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        SQLiteDatabase db=null;  
        try{  
            db=SQLiteDatabase.openOrCreateDatabase(  
                path: getFilesDir().getPath()+"/contacts.db",  
                factory: null  
            );  
            String s="DELETE FROM CONTACTS WHERE ";  
            s+="ID=? ";  
  
            db.execSQL(s, new Object[]{ID});  
            Toast.makeText(getApplicationContext(),  
                text: "Record DELETED",  
                Toast.LENGTH_LONG  
            ).show();  
        }catch (Exception e){  
            Toast.makeText(getApplicationContext(),  
                e.getMessage(),  
                Toast.LENGTH_LONG  
            ).show();  
        }finally {  
            if(db!=null){  
                db.close();  
            }  
        }  
    }  
});
```

След завършването на всяка една от заявките, връзката към базата е прекъсната.

## ЗАКЛЮЧЕНИЕ

Възможности за подобрене:

- Да се създаде собствено лого на приложението, което да се зарежда на телефона на потребителя вместо базовата иконка на приложение, създадено с Android Studio.
- Да се създаде Splash Screen за плавен преход между стартирането на приложението и зареждането на началната му страница.
- Да има възможност за плавен преход между панелите на приложението;
- Да има възможност за директно набиране на контакта чрез телефонния му номер.
- Да има възможност за добавяне на снимка на съответния контакт.