

**НАПРЕДНИ КОНЦЕПТИ У БАЗАМА ПОДАТАКА**

**+**

**РАЗВОЈ АПЛИКАЦИЈА ЗА МОБИЛНЕ СИСТЕМЕ**

МЕНТОР: СТУДЕНТ:

проф. др Лазар Копања Драгослав Бабић, МИТ21/19

мр Небојша Андријевић Марко Козомора, МИТ16/19

НОВИ САД, ДЕЦЕМБАР 2020

САДРЖАЈ:

[1 УВОД 3](#_Toc59997510)

[1.1 Идјеа за пројекат 3](#_Toc59997511)

[1.2 Алати / технологије за израду пројекта 3](#_Toc59997512)

[1.2.1 MongoDB – База података 4](#_Toc59997513)

[1.2.2 Spring Boot – Backend 4](#_Toc59997514)

[1.2.3 Android studio – JAVA – Frontend 4](#_Toc59997515)

[2 АПЛИКАЦИЈА – Евиденција вожње 5](#_Toc59997516)

[3 ТЕХНИЧКИ ИЗВЕШТАЈ 6](#_Toc59997517)

[3.1 Увод 6](#_Toc59997518)

[3.2 Дефинисање проблема 6](#_Toc59997519)

[3.3 Експеримент 6](#_Toc59997520)

[4 ЗАКЉУЧАК 12](#_Toc59997521)

[5 ЛИТЕРАТУРА 13](#_Toc59997522)

# УВОД

## Идјеа за пројекат

Модернизација и усавршавање постојећих система за обраду и прикупљање података омогућила је фирмама на тржишту да унапреде своје пословање уз релативно мала улагања. Приступачност техничке опреме дала је власницима компанија (фирми) да за релативно мали новац многоструко олакшају посао својим запосленима и повећају њихову продуктивност. У светлу горе наведеног дошли смо на идеју да израдом апликације за Android платформу, унапредимо и убрзамо посао једног диспечера у неком предузећу .

У великом броју предузећа које се баве транспортом, као и у предузећима која поседују службене аутомобиле, то представља огроман и одговоран посао. Ситна грешка која се направи приликом отварања или затварања путног налога, папира који је потребан да се изврши вожња службеним возилом, могла би да створи велике проблеме приликом контроле предузећа.



*Слика 1. – Сто једног диспечера*

Сваки корисник службеног возила би имао свој кориснички налог којим би се пријављивао на апликацију и тиме смањио могућност настанка грешке (није могуће да један корисник има 2 или више отворених вожњи/налога).

## Алати / технологије за израду пројекта

У наредном поглављу су наведене наведени алати/технологије којима је израђен пројекат. За базу података је коришћен MongoDB, за Backend SpringBoot а за Frontend JAVA. Апликација је писана у Android Studio.

### MongoDB – База података

MongoDB је NoSQL база података оријентисана на документе која се користи за велико складиштење података. Уместо да користи табеле и редове као у традиционалним релационим базама података, МонгоДБ користи колекције и документе. ... Збирке садрже скупове докумената и функције што је еквивалент релацијским табелама базе података

### Spring Boot – Backend

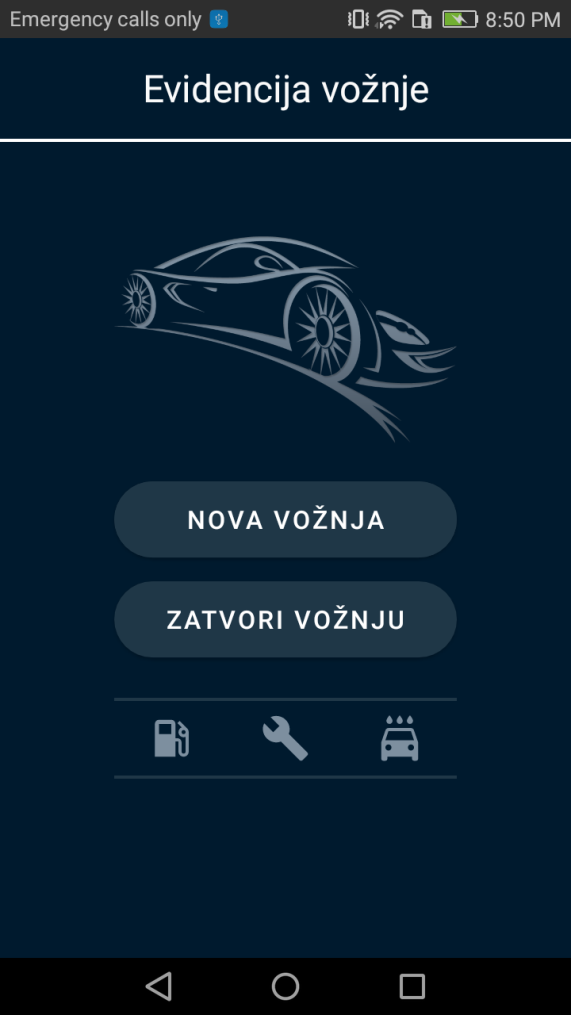
SpringBoot је framework отвореног кода заснован на Јави који се користи за креирање микро услуге. Развио га је Пивотал Теам и користи се за изградњу самосталних и за производњу спремних апликација. SpringBoot пружа добру платформу за програмере Јаве да развију самосталну апликацију коју можете једноставно покренути. Можете започети са минималним конфигурацијама без потребе за комплетним подешавањем конфигурације Спринг.

### Android studio – JAVA – Frontend

Јава је објектно оријентисани програмски језик заснован на класама који је дизајниран да има што мање зависности од имплементације. То је програмски језик опште намене намењен програмерима апликација да пишу једном, покрећу се било где, што значи да се компајлирани Јава код може покретати на свим платформама које подржавају Јаву без потребе за поновном компилацијом. Јава апликације се обично компајлирају у бајт код који се може изводити на било којој Јава виртуелној машини без обзира на основну рачунарску архитектуру. Синтакса Јава слична је C и C++, али има мање могућности нижег нивоа од било које од њих. Јава рунтиме пружа динамичке могућности (као што су рефлексија и модификација кода извршавања) које обично нису доступне у традиционалним компајлираним језицима. Од 2019. године Јава је била један од најпопуларнијих програмских језика који се користе према GitHub-у, посебно за клијент-сервер веб апликације, са пријављених 9 милиона програмера.

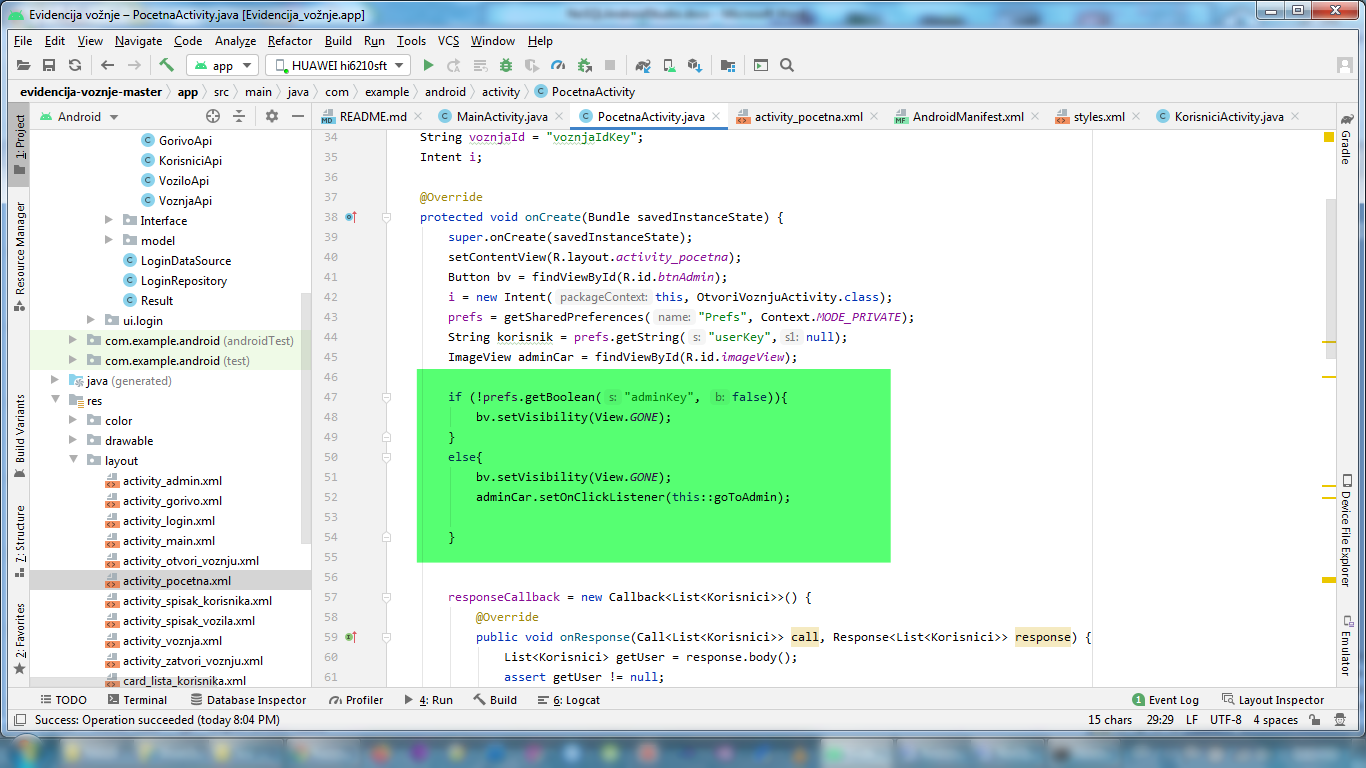
# АПЛИКАЦИЈА – Евиденција вожње

На основу имена апликације се види чему је апликација намењена, да евидентира вожње корисника. Замисао је да апликација буде инсталирана на неки андроид уређај који је преко CanBus-а повезан са путним рачунаром возила. Логовањем на апликацију корисник добија приступ главном менију (Слика 2).



*Слика 2. – Изглед main форме*

Уколико су кориснику омогућена администраторска права, кликом на слику аутомобила, приступа се администраторском мени-у. Ако корисник апликације нема админ права, додир слике неће имати никакву функцију. На слици 3 је приказан део кода који омогућује приступање администраторском делу апликације (маркиран зеленом).



*Слика 3. – Изглед кода из PocetnaActivity.java*

# ТЕХНИЧКИ ИЗВЕШТАЈ

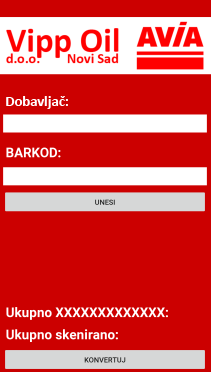
## Увод

У овом раду описује се израда софтверског пакета за извршавање пријема робе за маркет у фирми „VIPP OIL“ и контрола са доставном књигом.

## Дефинисање проблема

Због велике количине различитих артикала којим фирма располаже, честе су грешке приликом мануалног сравњивања доставне књиге које доводе до неслагања робе на лагеру и стања по књигама. Из тог разлога договорено је да се изради апликација која би тај процес аутоматизовала и могућност грешке свела на минимум.

## Експеримент

 Због исплативости реализације пројекта одлучено је да се пише апликација за Android оперативни систем (доступност хардвера у виду паметних телефона) којим би се радило сравњење робе и апликација за Windows платформу којим би се прикупљени подаци анализирали. За израду андроид апликације коришћен је Android Studio а Microsoft Visual Studio 2010 за израду апликације на рачунару.

Осмишљен је кориснички интерфејс за апликацију (Слика 4). Корисник пре почетка рада пописа треба да дефинише шифре добављача од којих се поручује роба (нпр Нелт – 001, Мерката – 002, Бастапромет – 003, итд).

Стартовањем апликације на паметном телефону контрола пријема робе може да почне. У поље Добављач корисник уноси троцифрени предефинисани број за добављача од којег планира да контролише пријем робе, затим се позиционира на поље баркод и почиње са скенирањем баркодова који се налазе на артиклима. Апликација генерише базу података у којој се налазе скенирани баркодови и број скенирања истих, као и укупан број скенираних артикала. Те податке исписује на радној површини апликације и тиме кориснику омогућује додатну контролу приликом скенирања.

*Слика 4.*

Поља за унос локације и баркода су дефинисана као EditText поља а натписи (лабеле) „Dobavljač“, „BARKOD“, „Ukupno“ и „Ukupno skenirano“ као TextView. Следи код из фајла задуженог за изглед апликације (activity\_main.xml):

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@android:color/holo\_red\_dark"  
 tools:context=".MainActivity"**>  
  
 <**ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:src="@drawable/baner"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/baner"** />  
  
 <**TextView  
 android:id="@+id/tvLabelaDob"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:focusable="false"  
 android:text=" Dobavljač:"  
 android:textColor="@android:color/background\_light"  
 android:textSize="26dp"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView"** />  
  
 <**EditText  
 android:id="@+id/etDob"  
 android:layout\_width="0dp"  
 android:layout\_height="35dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:background="@android:color/background\_light"  
 android:ems="10"  
 android:inputType="number"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvLabelaDob"** />  
  
 <**TextView  
 android:id="@+id/tvLabelaBarKod"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:focusable="false"  
 android:text=" BARKOD:"  
 android:textColor="@android:color/background\_light"  
 android:textSize="26dp"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/etDob"** />  
  
 <**EditText  
 android:id="@+id/etBarKod"  
 android:layout\_width="0dp"  
 android:layout\_height="35dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="12dp"  
 android:layout\_marginEnd="8dp"  
 android:background="@android:color/background\_light"  
 android:ems="10"  
 android:inputType="number"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvLabelaBarKod"** />  
  
 <**Button  
 android:id="@+id/btn\_add"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:text="Unesi"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/etBarKod"** />  
  
 <**TextView  
 android:id="@+id/tvLabelaUkupnoSpec"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="8dp"  
 android:focusable="false"  
 android:text=" Ukupno XXXXXXXXXXXXX: "  
 android:textColor="@android:color/background\_light"  
 android:textSize="26dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toTopOf="@+id/tvLabelaUkupno"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"** />  
  
 <**TextView  
 android:id="@+id/tvLabelaUkupno"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="8dp"  
 android:focusable="false"  
 android:text=" Ukupno skenirano: "  
 android:textColor="@android:color/background\_light"  
 android:textSize="26dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toTopOf="@+id/btn\_convert"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"** />  
  
 <**Button  
 android:id="@+id/btn\_convert"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:text="Konvertuj"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"** />  
  
</**android.support.constraint.ConstraintLayout**>

Апликација након скенирања једног ентитета ресетује поље баркод и курсор остаје позициониран на том пољу како би скенирање могло наставити. Када корисник жели да промени добављача то треба да уради након завршеног скенирања артикла одабиром (уписивањем) нове/друге шифре добављача у поље добављач. Податак који се приликом успешног скенирања уписује у базу података изгледа као: **001|3334445559|10.03.2019 15:00** где први део низа представља шифру добављача, други баркод а трећи датум и време скенирања. Када се заврши скенирање артикала корисник треба да притисне дугме „Konvertuj“ чиме ће податке који се налазе у бази података уписати у новокреирани „.csv“ документ и тиме их учинити доступним за даљу обраду. Добијени документ се увлачи у апликацију која се налази на рачунару (Windows платформа) која додатно обрађује документ и упоређује га са стањем доставне књиге коју је добављач дужан да пошаље у електронској форми.

У наставку иде део кода задужен за упис података у базу (DatabaseHelper.java):

**package** com.mkstudio.avia.inventaravia.database;  
  
**import** android.content.ContentValues;  
**import** android.content.Context;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.database.sqlite.SQLiteOpenHelper;  
  
**public class** DatabaseHelper **extends** SQLiteOpenHelper {  
  
 **public static final** String ***DATABASE\_NAME*** = **"TempAviaDatabase.db"**;  
 **public static final** String ***TABLE\_NAME*** = **"tblPrijemRobe"**;  
 **public static final** String ***COL\_01*** = **"ID"**;  
 **public static final** String ***COL\_02*** = **"colBarKod"**;  
 **public static final** String ***COL\_03*** = **"colDobSifra"**;  
 **public static final** String ***COL\_04*** = **"colDatumSken"**;  
  
 **public** DatabaseHelper(Context context) {  
 **super**(context, ***DATABASE\_NAME***, **null**, 1);  
 }  
  
 @Override  
 **public void** onCreate(SQLiteDatabase db) {  
 db.execSQL(**"CREATE TABLE "** + ***TABLE\_NAME*** + **" ("** + ***COL\_01*** + **" INTEGER PRIMARY KEY AUTOINCREMENT,"** + ***COL\_02*** + **" NUMERIC,"** + ***COL\_03*** + **" TEXT,"** + ***COL\_04*** + **" TEXT)"**);  
 }  
  
 **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 db.execSQL(**"DROP TABLE IF EXISTS "** + ***TABLE\_NAME***);  
 onCreate(db);  
 }  
  
 **public** Cursor getAllDataMain() {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 Cursor res = db.rawQuery(**"SELECT \* FROM "** + ***TABLE\_NAME***, **null**);  
 **return** res;  
 }  
  
 **public boolean** insertTempData (String barkod\_broj, String dobavljac\_sifra, String datum){  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
 contentValues.put(***COL\_02***, barkod\_broj);  
 contentValues.put(***COL\_03***, dobavljac\_sifra);  
 contentValues.put(***COL\_04***, datum);  
 **long** results = db.insert(***TABLE\_NAME***, **null**, contentValues);  
 **if** (results == -1)  
 **return false**;  
 **else  
 return true**;  
 }  
  
 **public** Integer counterArtikl () {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 Cursor res = db.rawQuery(**"SELECT \* FROM "** + ***TABLE\_NAME***, **null**);  
 **if** (res.getCount()<= 0){  
 res.close();  
 **return** 0;  
 }  
 **else** {  
 **int** counter = res.getCount();  
 res.close();  
 **return** counter;  
 }  
 }  
  
 **public** Integer counterArtiklSpec (String barkod) {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 String selectQuery = **"SELECT \* FROM "** + ***TABLE\_NAME*** + **" WHERE "** + ***COL\_02*** + **" = ?"**;  
 Cursor res = db.rawQuery(selectQuery, **new** String[]{barkod});  
 **if** (res.getCount()<= 0){  
 res.close();  
 **return** 0;  
 }  
 **else** {  
 **int** counter = res.getCount();  
 res.close();  
 **return** counter;  
 }  
 }  
  
 **public** Integer deleteDataMain () {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 **return** db.delete(***TABLE\_NAME***, **null**,**null**);  
 }  
}

Део кода из главног дела програма (MainActivity.java):

**public class** MainActivity **extends** AppCompatActivity **implements** View.OnClickListener {  
  
 DatabaseHelper **myDB**;  
 Button **bKonvertuj**, **bPrihvati**;  
 TextView **tvLabelaDobavljac**,**tvLabelaBarkod**, **tvLabelaUkupno**, **tvLabelaUkupnoSpec**;  
 EditText **etDobavljac**, **etBCArt**;  
 String **strBarkod**, **strDobavljac**;  
 **boolean hasPermission**;  
 **private static final int *REQUEST\_WRITE\_STORAGE*** = 112;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **myDB** = **new** DatabaseHelper(**this**);  
  
 **bKonvertuj** = (Button) findViewById(R.id.***btn\_convert***);  
 **bPrihvati** = (Button) findViewById(R.id.***btn\_add***);  
  
 **tvLabelaDobavljac**= (TextView) findViewById(R.id.***tvLabelaDob***);  
 **tvLabelaBarkod**= (TextView) findViewById(R.id.***tvLabelaInvBr***);  
 **tvLabelaUkupnoSpec**= (TextView) findViewById(R.id.***tvLabelaUkupnoSpec***);  
 **tvLabelaUkupno**= (TextView) findViewById(R.id.***tvLabelaUkupno***);  
  
 **etDobavljac** = (EditText) findViewById(R.id.**etDob**);  
 **etBCArt** = (EditText) findViewById(R.id.***etBarKod***);  
  
 **bPrihvati**.setOnClickListener(**this**);  
 **bKonvertuj**.setOnClickListener(**this**);  
  
 **hasPermission** = (ContextCompat.*checkSelfPermission*(**this**,  
 Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***) == PackageManager.***PERMISSION\_GRANTED***);  
 **if** (!**hasPermission**) {  
 ActivityCompat.*requestPermissions*(**this**,  
 **new** String[]{Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***},  
 ***REQUEST\_WRITE\_STORAGE***);  
 } **else**{  
 *// do nothing* }  
 }

# ЗАКЉУЧАК

Успешном сарадњом са коментором и запосленим радницима на пумпи одрађена апликација је знатно убрзала и олакшала битан део посла везан за пословање пумпе. Кроз израду апликација успео сам знатно да унапредим своја знања везана за програмирање, али и генерално рад у тиму.

# ЛИТЕРАТУРА

1. <https://www.guru99.com/what-is-mongodb.html>
2. <https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm>
3. <https://en.wikipedia.org/wiki/Java_(programming_language)>