

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2
По курсу “Проектирование баз знаний”

Выполнил студент группы 821703:

Щур А. А.

Проверил:

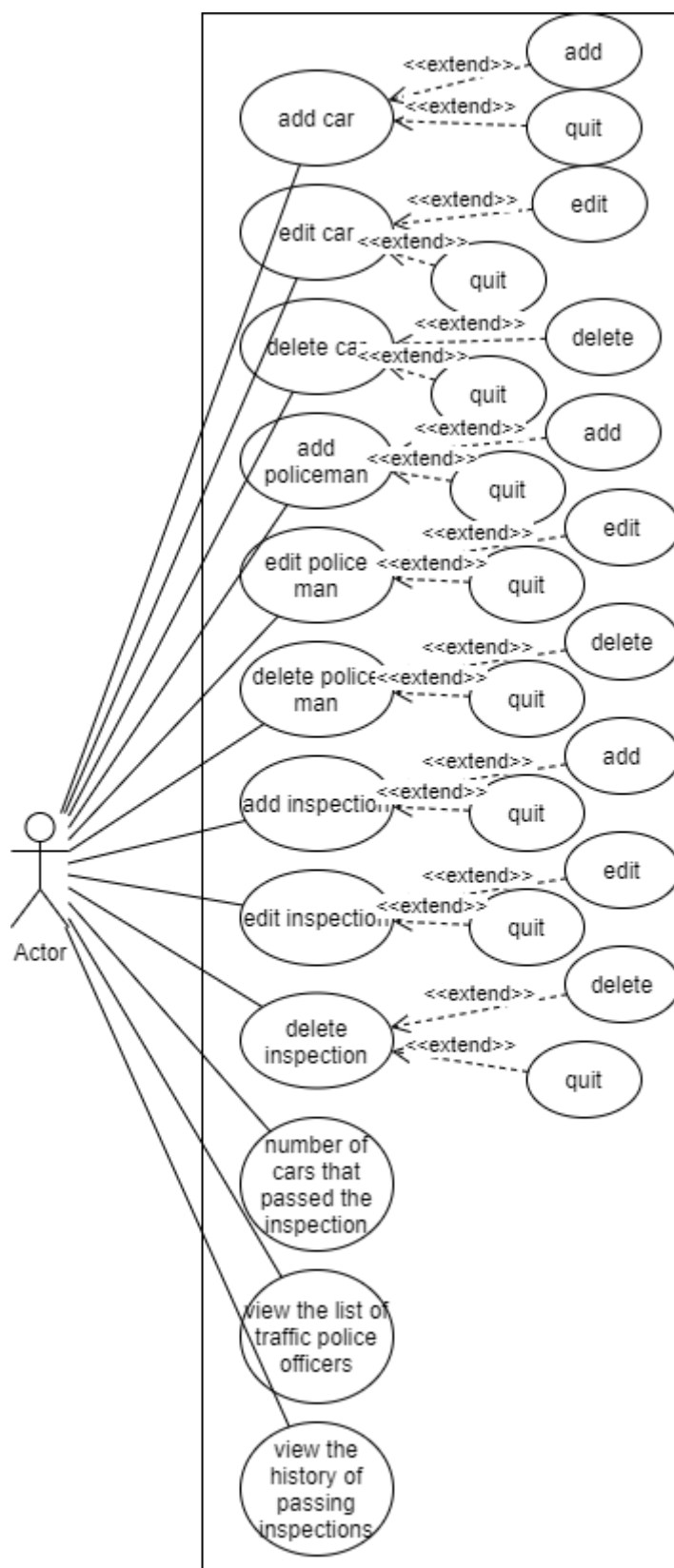
Синельников П.М.

МИНСК
2020

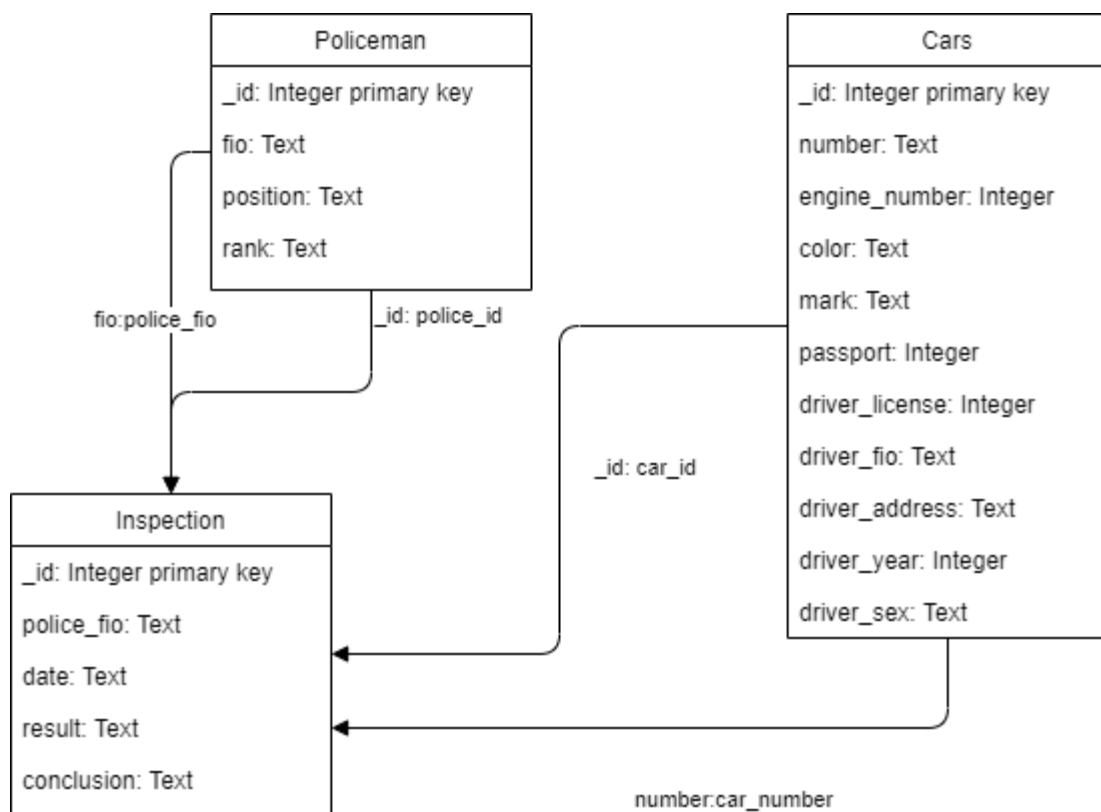
Оглавление

Концептуальное проектирование	3
Физическое проектирование	4
Инструментальные средства	6
Словесное описание предметной области	7
Основные элементы интерфейса	8
Тексты основных запросов, функций, процедур и триггеров	11

1.Концептуальное проектирование



2. Физическое проектирование



3. Инструментальные средства

Для написания программы был выбран язык программирования kotlin. В качестве СУБД была выбрана SQLite. Всё писалось приложением для операционной системы Android

4. Словесное описание предметной области

Вариант 3. «Технический осмотр автомобилей»

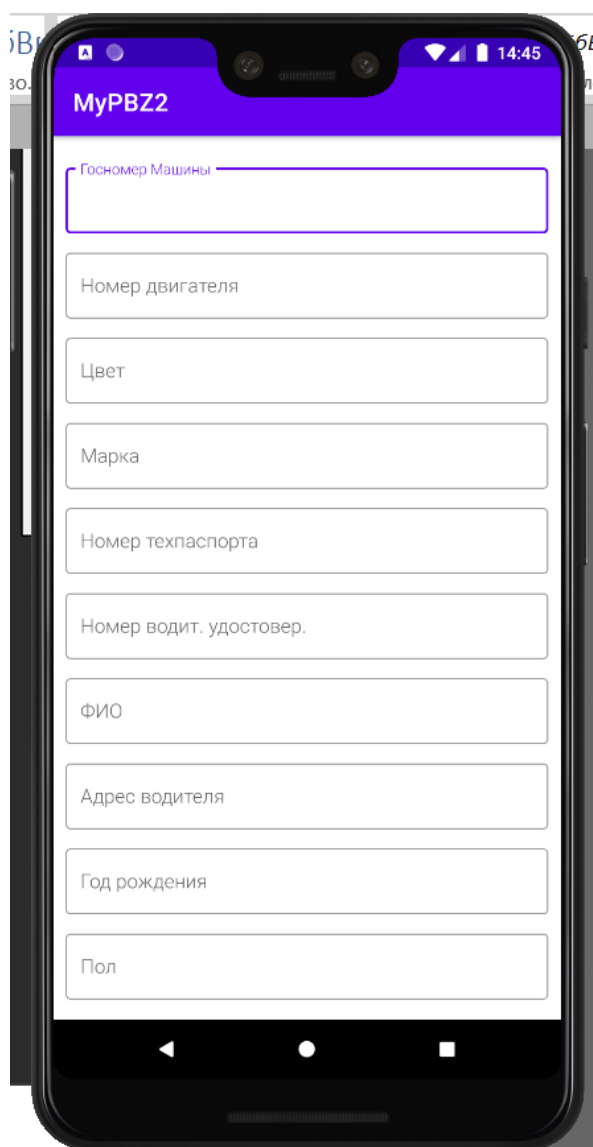
Предприятие – учреждение ГИБДД

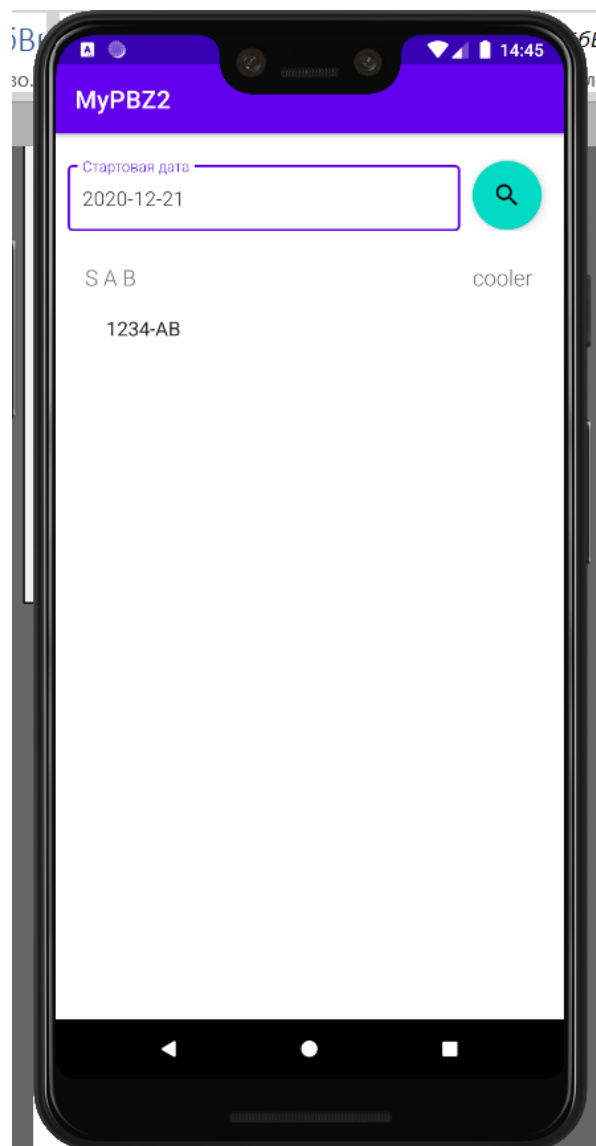
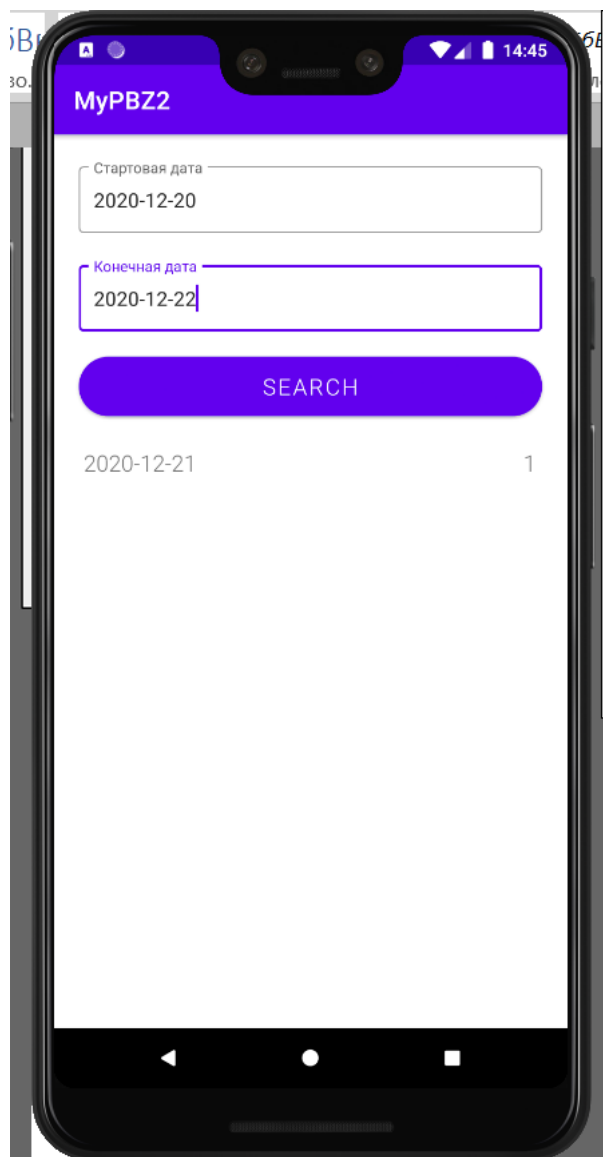
Словесное описание предметной области: При проведении технического осмотра автомобиля необходимо фиксировать следующие данные: госномер автомобиля, проходящего технический осмотр, номер двигателя, цвет, марка, номер технического паспорта, номер водительского удостоверения, ФИО владельца, адрес прописки, год рождения, пол. Данные фиксируются на дату прохождения текущего осмотра, необходимо хранить историю осмотров – дата прохождения, результат. Необходимо также фиксировать ФИО, должность, звание сотрудника ГАИ, проводившего осмотр, заключение осмотра. Каждый день технический осмотр могут проходить много автомобилей, проводить осмотр могут разные сотрудники, но каждый сотрудник проводит за день не более 10 осмотров.

Необходимо реализовать выполнения следующих функций:

- Добавление/редактирование/удаление информации о владельцах автомобилей и их транспортных средствах.
- Добавление/редактирование/удаление информации сотрудников ГАИ.
- Добавление/редактирование/удаление информации о проведенном осмотре.
- Расчет количество автомобилей, прошедших техосмотр за заданный промежуток времени с разбивкой по дням.
- Просмотр списка сотрудников ГАИ, проводивших осмотр на заданную дату: ФИО, звание сотрудника, госномера автомобилей, которые он осматривал.
- Просмотр истории прохождения осмотров заданным автомобилем (номер двигателя) – дата прохождения, результат.

5. Основные элементы интерфейса







7.Тексты основных запросов, функций, процедур и триггеров

```
cursor = db.rawQuery( sql: "SELECT *\n" +  
    "FROM ${DBHelper.TABLE_CONTACT3}\n" +  
    "WHERE DATE(${DBHelper.TL_DATE}) <= ? AND DATE(${DBHelper.TL_DATE}) >= ? ",  
    arrayOf(endDate.text.toString(), startDate.text.toString()))
```

```
button.setOnClickListener {  
    cursor = db.rawQuery("SELECT * FROM ${DBHelper.TABLE_CONTACT1} WHERE ${DBHelper.CAR_ENGINE_NUMBER} = ?",  
        arrayOf(searchField.text.toString()))  
}
```

```
button.setOnClickListener { it: View!  
    cursor = db.rawQuery( sql: "SELECT * FROM ${DBHelper.TABLE_CONTACT3} WHERE DATE(${DBHelper.TL_DATE}) = ? ",  
        arrayOf(dateChoose.text.toString()))  
    if (cursor.moveToFirst()) {
```

```
private fun save(){  
    val cv = ContentValues()  
    cv.put(DBHelper.CAR_NUMBER, addCarNumber.text.toString())  
    cv.put(DBHelper.CAR_ENGINE_NUMBER, addCarEngine.text.toString().toInt())  
    cv.put(DBHelper.CAR_COLOR, addCarColor.text.toString())  
    cv.put(DBHelper.CAR_MARK, addCarMark.text.toString())  
    cv.put(DBHelper.CAR_PASSPORT, addTechPassport.text.toString().toInt())  
    cv.put(DBHelper.CAR_DRIVER_LICENSE, addCarDriverLicense.text.toString().toInt())  
    cv.put(DBHelper.CAR_DRIVER_FIO, addDriverFIO.text.toString())  
    cv.put(DBHelper.CAR_DRIVER_ADDRESS, addDriverAddress.text.toString())  
    cv.put(DBHelper.CAR_DRIVER_YEAR, addDriverYear.text.toString().toInt())  
    cv.put(DBHelper.CAR_DRIVER_SEX, addDriverSex.text.toString())  
    arg = if (arg>0){  
        db.update(DBHelper.TABLE_CONTACT1, cv, whereClause: "${DBHelper.CAR_ID} = $arg", whereArgs: null)  
    }  
    else{  
        db.insert(DBHelper.TABLE_CONTACT1, nullColumnHack: null, cv).toInt()  
    }  
    close()  
}
```