|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**КУРСОВАЯ РАБОТА**

по дисциплине «Разработка мобильных приложений»

**Тема курсовой работы**

“Автоматизированная система работы автосервиса”

**Студент группы** ИКБО-26-20 Калугин Данила Сергеевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель курсовой работы** доцент Рысин М.Л.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена к защите «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Допущен к защите «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва – 2022

|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

Утверждаю

Заведующий кафедрой МОСИТ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Головин С.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы по дисциплине**

«Разработка мобильных приложений»

Студент Калугин Данила Сергеевич Группа ИКБО-26-20

**Тема работы:** Автоматизированная система работы автосервиса

**Исходные данные:**

**Разрабатываемый прототип должен** предоставлять возможности полностью интерактивной системы с понятным дружественным UI и обеспечивать необходимую функциональность, которая формируется в зависмости от заданной темы и предметной области изучаемых вопросов.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:**

Установка и настройка Android с применением виртуальных сред. Установка и настройка IDE (Android Studio, Eclipse, IntelliJ IDEA, Kivi и т.п.). Установка и настройка эмуляторов Android.

Анализ предметной области разрабатываемой программной системы, сбор и анализ требований, составление ТЗ согласно ГОСТ 19.201-78.

Изучение жизненного цикла программ и создание мобильного программного комплекса с применением языков программирования JAVA, Python, Kotlin,Swift, и др. (применяются по отдельному указанию руководителя), согласно темы курсовой работы. Реализация в создаваемом программном комплексе хранения данных в виде БД, файлов или пар ключ-значение. Обеспечение их создания, чтения, записи во внутреннем и внешнем хранилище.

Обеспечение безопасности информации при работе планового программного комплекса (обеспечение работы ролевой модели безопасности).

Тестирование и отладка кода созданного программного продукта, контрольные примеры работы.

Возможно портирование написанной программной системы на внешних хостах в сети Интернет.

Отчет по курсовой работе в виде пояснительной записки.

**Срок представления к защите курсовой работы:** «30» мая 2022 г.

**Задание на курсовую работу выдал** \_\_\_\_\_\_\_\_\_\_\_\_ Рысин М.Л.

«15» февраля 2022 г.

**Задание на курсовую работу получил**  \_\_\_\_\_\_\_\_\_\_\_\_ (Калугин Д.С.)

**Содержание**

[**Список сокращений и специальных терминов** 4](#_Toc104505550)

[**ВВЕДЕНИЕ** 5](#_Toc104505551)

[**Основная часть** 6](#_Toc104505552)

[**Анализ предметной области “Автоматизированная система работы автосервиса”** 6](#_Toc104505553)

[**Требования к системе** 6](#_Toc104505554)

[**Выбор системного и инструментального ПО** 7](#_Toc104505555)

[**Архитектура ПС** 8](#_Toc104505556)

[**Реализация ПС** 9](#_Toc104505557)

[**Контрольные примеры работы** 22](#_Toc104505558)

[**ЗАКЛЮЧЕНИЕ** 27](#_Toc104505559)

[**Список используемых источников** 28](#_Toc104505560)

# **Список сокращений и специальных терминов**

ПО – Программное обеспечение

ТЗ – Техническое задание

КР – Курсовая работа

БД – База данных

ОБД – Облачная база данных

СУБД – Система управления базами данных

ПК – Персональный компьютер

ЯП – Язык программирования

ИС – Информационная система

ОС – Операционная система

Android Studio – Интегрированная среда разработки для работы с платформой/операционной системой Android.

Java – Строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems.

Android SDK – Android Software Development Kit. Универсальное средство разработки мобильных приложений для операционной системы Android.

XML – eXtensible Markup Language. Расширяемый язык разметки. Используется для хранения и передачи данных.

JSON – JavaScript Object Notation. Текстовый формат обмена данными, основанный на JavaScript.

res – Каталог для хранения ресурсов проекта в Android Studio

IDE – Integrated development environment.Интегрированная среда разработки

# **ВВЕДЕНИЕ**

Актуальность темы КР заключается в том, что людям приходится чаще обращаться к услугам автосервисов для ремонта их личных и коммерческих транспортных средств. С ростом количества автомобилей на дорогах, возникает все больше работы по устранению неисправностей и обслуживанию этих автомобилей. Соответственно, с увеличением количества заказов, автосервисам требуются новые технологии для оказания более качественных и своевременных услуг клиенту. Такие технологии упростят мониторинг работы и позволят централизованно собирать информацию о проделанной работе.

Подводя итог, разработка данного позволит приблизить стандартную модель автосервиса к автоматизированному и информационному концепту.

Целью данной КР является разработка приложения на тему “Автоматизированная система работы автосервиса”.

Задачами КР являются:

* выбор задания и его утверждение;
* расширенное описание задания на естественном языке.
* разработка технического задания по выбранному заданию;
* Создание мобильного программного комплекса с применением ЯП JAVA.
* Тестирование и отладка разработанного мобильного приложения;
* разработка пользовательских мобильных интерфейсов;
* оформление пояснительной записки на курсовую работу;
* сдача на проверку;
* защита.

Структура КР:

* Список сокращений и специальных терминов
* Введение
* Основная часть
* Заключение
* Список используемых источников

# **Основная часть**

## **Анализ предметной области “Автоматизированная система работы автосервиса”**

Услуги автосервисов предоставляются их же владельцами. В зависимости от специализации, автомастерские предлагают различные услуги. В список самых популярных услуг, которые предлагают автомобилистам, входят:

* Проведение диагностики на предмет неисправностей;
* Техническое обслуживание автомобилей;
* Различные виды ремонта;
* Покраска деталей;
* Химчистка салонов автомобилей;
* Различные улучшения для автомобиля;
* Замена старых запчастей на новые;
* Мойка автомобиля;
* Шиномонтаж.

## **Требования к системе**

Функциональными требованиями к разрабатываемому мобильному приложению являются следующие пункты:

* Логичный и понятный пользовательский интерфейс;
* Удобство работы пользователя с приложением;
* Связь приложения с БД для хранения данных о пользователе и поступивших заявках;
* Система должна быть доступна для всех устройств с ОС Android 7.1 и выше

## **Выбор системного и инструментального ПО**

Для разработки мобильного приложения выбор пал на Android Studio, т. к. данное ПО целенаправленно заточено под разработку мобильных приложений, а также Android Studio подходит мне по системным требованиям.

Мои системные требования:

* Оперативная память: 16 GB ОЗУ;
* Процессор: AMD Ryzen 5 1600x;
* Свободное место на диске: 400GB;
* Видеокарта: GeForce GTX1050 Ti;
* Разрешение экрана: 2540x1440.

Системные требования Android Studio показаны на рисунке 1.

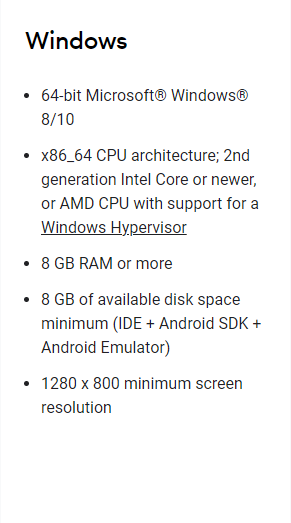


Рисунок 1. Системные требования Android Studio

В качестве платформы для использования БД при разработке мобильного приложения использовалась Firebase. Firebase — это мобильная платформа Google, которая помогает быстро разрабатывать высококачественные приложения, в неё входят такие функции как: Authentication и Realtime Database, которые я использовал в своём мобильном приложении.

Для подключения Firebase к вашему проекту в Android Studio, следует проделать следующие простые действия:

* Зайти в свой аккаунт Google через Android Studio;
* Во вкладке Tools в Android Studio выбрать пункт Firebase;
* Выбрать нужные вам функции.

Всё остальное будет интуитивно понятно, что также является большим плюсом данной платформы.

## **Архитектура ПС**

Вообще говоря, не существует общепринятого термина «архитектура программного обеспечения». Хорошая архитектура — это прежде всего выгодная архитектура, делающая процесс разработки и сопровождения программы более простым и эффективным. Программу с хорошей архитектурой легче расширять и изменять, а также тестировать, отлаживать и понимать.

На самом деле можно сформулировать список вполне разумных и универсальных критериев:

* Эффективность;
* Гибкость;
* Расширяемость;
* Масштабируемость процесса разработки;
* Тестируемость;
* Поддержка приложения.

Все эти пункты я старался использовать при разработке собственного мобильного приложения.

## **Реализация ПС**

Далее будут представлена реализация кода основных файлов проекта.

**Authorization.java**

package com.example.autoservice;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.Objects;  
  
public class Authorization extends AppCompatActivity {  
  
 FirebaseAuth mAuth = FirebaseAuth.*getInstance*();  
 DatabaseReference database = FirebaseDatabase.*getInstance*().getReference();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_auth*);  
  
 Button to\_register\_btn = findViewById(R.id.*ToRegisterButton*);  
  
 Button to\_auth\_btn = findViewById(R.id.*AuthButton*);  
  
 EditText login = findViewById(R.id.*TextEmailAuth*);  
 EditText password = findViewById(R.id.*TextPasswordAuth*);  
  
 to\_register\_btn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent(Authorization.this, Registration.class);  
 startActivity(intent);  
 }  
 });  
  
 to\_auth\_btn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 mAuth.signInWithEmailAndPassword(login.getText().toString(), password.getText().toString()).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 database = FirebaseDatabase.*getInstance*().getReference("Users").child(mAuth.getCurrentUser().getUid()).child("status");  
 database.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 String user\_status = ((snapshot.getValue(String.class)));  
 if (Objects.*equals*(user\_status, "Admin")) {  
 Intent intent = new Intent(Authorization.this, FrontPageAdmin.class);  
 startActivity(intent);  
 }  
 if (Objects.*equals*(user\_status, "Worker")){  
 Intent intent2 = new Intent(Authorization.this, FrontPageWorker.class);  
 startActivity(intent2);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
 }  
 }  
 });  
 }  
 });  
 }  
}

**activity\_auth.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.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"  
 tools:context=".Authorization">  
  
 <EditText  
 android:id="@+id/TextEmailAuth"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="124dp"  
 android:ems="10"  
 android:hint="@string/email"  
 android:inputType="textEmailAddress"  
 android:minHeight="48dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.507"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:autofillHints="emailAddress" />  
  
 <EditText  
 android:id="@+id/TextPasswordAuth"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="68dp"  
 android:ems="10"  
 android:hint="@string/password"  
 android:inputType="textPassword"  
 android:minHeight="48dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.507"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TextEmailAuth"  
 android:autofillHints="password" />  
  
 <Button  
 android:id="@+id/AuthButton"  
 android:layout\_width="213dp"  
 android:layout\_height="48dp"  
 android:layout\_marginTop="68dp"  
 android:text="@string/authorization"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.507"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TextPasswordAuth" />  
  
 <Button  
 android:id="@+id/ToRegisterButton"  
 android:layout\_width="213dp"  
 android:layout\_height="48dp"  
 android:layout\_marginTop="36dp"  
 android:text="@string/to\_register"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.507"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/AuthButton" />  
</androidx.constraintlayout.widget.ConstraintLayout>

**Registration.java**

package com.example.autoservice;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
public class Registration extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_registration*);  
  
 EditText name = findViewById(R.id.*TextName*);  
 EditText email = findViewById(R.id.*TextEmail*);  
 EditText password = findViewById(R.id.*TextPassword*);  
 Button register\_btn = findViewById(R.id.*RegisterButton*);  
  
 DatabaseReference database = FirebaseDatabase.*getInstance*().getReference();  
 FirebaseAuth mAuth = FirebaseAuth.*getInstance*();  
  
 register\_btn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 mAuth.createUserWithEmailAndPassword(email.getText().toString(), password.getText().toString()).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()){  
 String value\_name = name.getText().toString();  
 database.child("Users").child(mAuth.getCurrentUser().getUid()).child("name").setValue(value\_name);  
 database.child("Users").child(mAuth.getCurrentUser().getUid()).child("email").setValue(email.getText().toString());  
 database.child("Users").child(mAuth.getCurrentUser().getUid()).child("password").setValue(password.getText().toString());  
 database.child("Users").child(mAuth.getCurrentUser().getUid()).child("status").setValue("Worker");  
  
 Intent intent = new Intent(Registration.this, Authorization.class);  
 startActivity(intent);  
 Toast.*makeText*(Registration.this,"Успешная регистрация",Toast.*LENGTH\_LONG*).show();  
 }  
 else {  
 Toast.*makeText*(Registration.this,"Ошибка регистрации",Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 });  
 }  
 });  
 }  
}

**activity\_registration.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.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"  
 tools:context=".Registration">  
  
 <EditText  
 android:id="@+id/TextName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="124dp"  
 android:ems="10"  
 android:hint="@string/name"  
 android:inputType="textPersonName"  
 android:minHeight="48dp"  
 android:textColorHint="#757575"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:importantForAutofill="no" />  
  
 <EditText  
 android:id="@+id/TextEmail"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="64dp"  
 android:ems="10"  
 android:hint="@string/email"  
 android:inputType="textEmailAddress"  
 android:minHeight="48dp"  
 android:textColorHint="#757575"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TextName"  
 android:importantForAutofill="no" />  
  
 <EditText  
 android:id="@+id/TextPassword"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="64dp"  
 android:ems="10"  
 android:hint="@string/password"  
 android:inputType="textPassword"  
 android:minHeight="48dp"  
 android:textColorHint="#757575"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TextEmail"  
 android:importantForAutofill="no" />  
  
 <Button  
 android:id="@+id/RegisterButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/register"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TextPassword"  
 app:layout\_constraintVertical\_bias="0.308" />  
</androidx.constraintlayout.widget.ConstraintLayout>

**FrontPageAdmin.java**

package com.example.autoservice;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.ListView;  
import android.widget.Toast;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
import java.util.Objects;  
  
public class FrontPageAdmin extends AppCompatActivity {  
  
 FirebaseAuth mAuth = FirebaseAuth.*getInstance*();  
 DatabaseReference database = FirebaseDatabase.*getInstance*().getReference();  
  
 ArrayList<String> arr = new ArrayList<>();  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_front\_page\_admin*);  
 setTitle("Admin");  
  
 database = FirebaseDatabase.*getInstance*().getReference("Users").child(mAuth.getCurrentUser().getUid()).child("name");  
 database.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 String username = (String) snapshot.getValue(String.class);  
 Toast.*makeText*(FrontPageAdmin.this,"Добро пожаловать, " + username + "!",Toast.*LENGTH\_LONG*).show();  
  
  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 ArrayAdapter<String> adapter= new ArrayAdapter<>(FrontPageAdmin.this, android.R.layout.*simple\_list\_item\_1*, arr);  
  
 ListView applications\_list=findViewById(R.id.*ListViewApplicationsAdmin*);  
  
 applications\_list.setAdapter(adapter);  
  
 applications\_list.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
  
 }  
 });  
  
 database = FirebaseDatabase.*getInstance*().getReference("Applications");  
  
 database.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
 String value;  
  
 value=((snapshot.getKey()));  
 arr.add(value);  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot snapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 Button add\_application = findViewById(R.id.*AddApplicationButton*);  
  
 add\_application.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent(FrontPageAdmin.this, MakeApplication.class);  
 startActivity(intent);  
 }  
 });  
 }  
}

**activity\_front\_page\_admin.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.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"  
 tools:context=".FrontPageAdmin">  
  
 <ListView  
 android:id="@+id/ListViewApplicationsAdmin"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/AddApplicationButton"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <Button  
 android:id="@+id/AddApplicationButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="80dp"  
 android:text="@string/add\_application"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>

**FrontPageWorker.java**

package com.example.autoservice;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
import android.widget.Toast;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.ChildEventListener;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
  
public class FrontPageWorker extends AppCompatActivity {  
  
 FirebaseAuth mAuth = FirebaseAuth.*getInstance*();  
 DatabaseReference database = FirebaseDatabase.*getInstance*().getReference();  
  
 ArrayList<String> arr = new ArrayList<>();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_front\_page\_worker*);  
  
 database = FirebaseDatabase.*getInstance*().getReference("Users").child(mAuth.getCurrentUser().getUid()).child("name");  
 database.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 String username = (String) snapshot.getValue(String.class);  
 Toast.*makeText*(FrontPageWorker.this,"Добро пожаловать, " + username + "!",Toast.*LENGTH\_LONG*).show();  
 setTitle(username);  
  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 ArrayAdapter<String> adapter= new ArrayAdapter<>(FrontPageWorker.this, android.R.layout.*simple\_list\_item\_1*, arr);  
  
 ListView applications\_list=findViewById(R.id.*ListViewApplicationsWorker*);  
  
 applications\_list.setAdapter(adapter);  
  
 applications\_list.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
  
 }  
 });  
  
 database = FirebaseDatabase.*getInstance*().getReference("Applications");  
  
 database.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
 String value;  
  
 value=((snapshot.getKey()));  
 arr.add(value);  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot snapshot) {  
  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
 }  
}

**activity\_front\_page\_worker.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.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"  
 tools:context=".FrontPageWorker">  
  
 <ListView  
 android:id="@+id/ListViewApplicationsWorker"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginBottom="32dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0" />  
</androidx.constraintlayout.widget.ConstraintLayout>

**MakeApplication.java**

package com.example.autoservice;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
public class MakeApplication extends AppCompatActivity {  
  
 DatabaseReference database = FirebaseDatabase.*getInstance*().getReference();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_make\_application*);  
  
 setTitle("Создание заявки на ремонт");  
  
 EditText application\_name = findViewById(R.id.*TextApplicationName*);  
 EditText car\_make = findViewById(R.id.*TextCarMake*);  
 EditText car\_number = findViewById(R.id.*TextCarNumber*);  
 EditText work\_description = findViewById(R.id.*TextWorkDescription*);  
 Button makeApp\_btn = findViewById(R.id.*MakeApplicationButton*);  
  
 makeApp\_btn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String AppName = application\_name.getText().toString();  
 String CarMake = car\_make.getText().toString();  
 String CarNumber = car\_number.getText().toString();  
 String WorkDescription = work\_description.getText().toString();  
  
 database.child("Applications").child("Заявка: " + AppName).child("Car make").setValue(CarMake);  
 database.child("Applications").child("Заявка: " + AppName).child("Car number").setValue(CarNumber);  
 database.child("Applications").child("Заявка: " + AppName).child("Work description").setValue(WorkDescription);  
  
 Intent intent = new Intent(MakeApplication.this, FrontPageAdmin.class);  
 startActivity(intent);  
  
 Toast.*makeText*(MakeApplication.this,"Заявка успешно создана",Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }  
}

**activity\_make\_application.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.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"  
 tools:context=".MakeApplication">  
  
 <EditText  
 android:id="@+id/TextApplicationName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 android:autofillHints=""  
 android:ems="10"  
 android:hint="@string/application\_name"  
 android:inputType="text"  
 android:minHeight="48dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <EditText  
 android:id="@+id/TextCarMake"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:autofillHints=""  
 android:ems="10"  
 android:hint="@string/car\_make"  
 android:inputType="text"  
 android:minHeight="48dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TextApplicationName" />  
  
 <EditText  
 android:id="@+id/TextCarNumber"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:autofillHints=""  
 android:ems="10"  
 android:hint="@string/car\_number"  
 android:inputType="text"  
 android:minHeight="48dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TextCarMake" />  
  
 <EditText  
 android:id="@+id/TextWorkDescription"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:autofillHints=""  
 android:ems="10"  
 android:hint="@string/work\_description"  
 android:inputType="text"  
 android:minHeight="48dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TextCarNumber" />  
  
 <Button  
 android:id="@+id/MakeApplicationButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:layout\_marginBottom="80dp"  
 android:text="@string/make\_application"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TextWorkDescription"  
 app:layout\_constraintVertical\_bias="0.497" />  
</androidx.constraintlayout.widget.ConstraintLayout>

## **Контрольные примеры работы**

На следующих рисунках можно будет увидеть контрольные примеры работы мобильного приложения, а также его связь с БД (Рисунок 2-11):

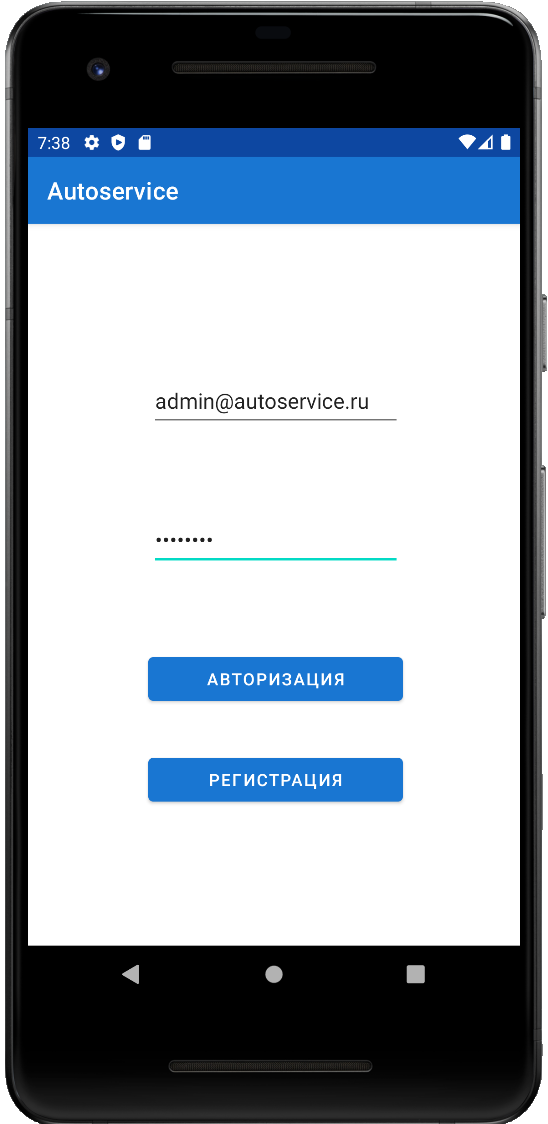


Рисунок 2. Стартовый экран приложения

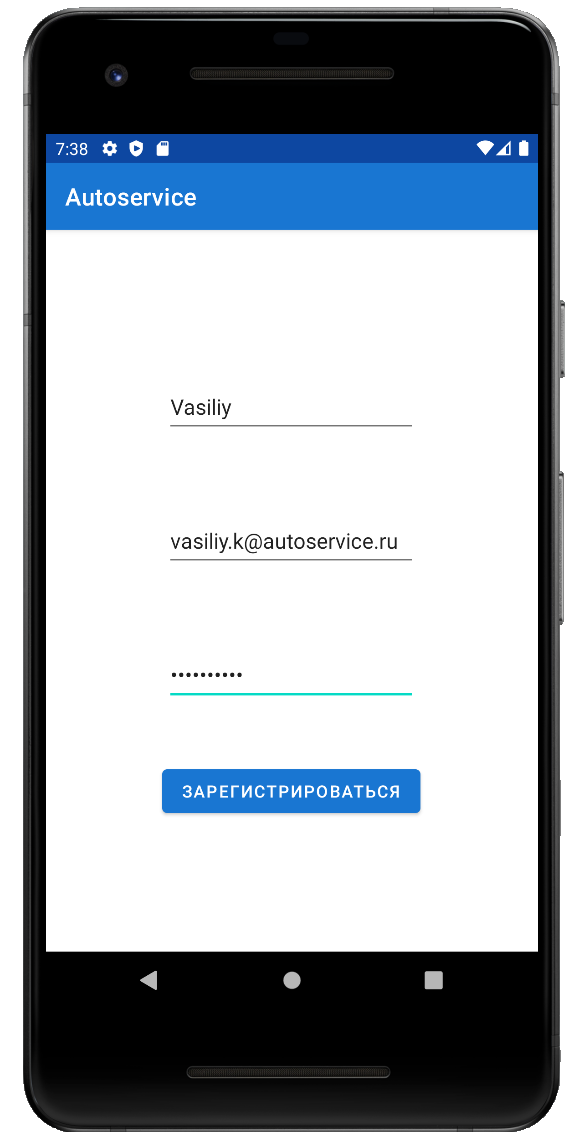


Рисунок 3. Процесс регистрации в приложении

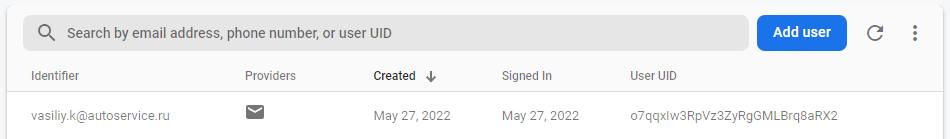


Рисунок 4. Authentication платформы Firebase

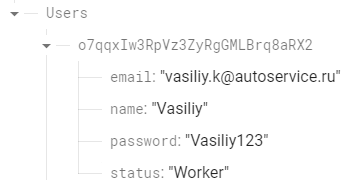


Рисунок 5. Realtime Database платформы Firebase

Как видно из рисунков выше, приложение правильно работает с функцией Authentication и Database.

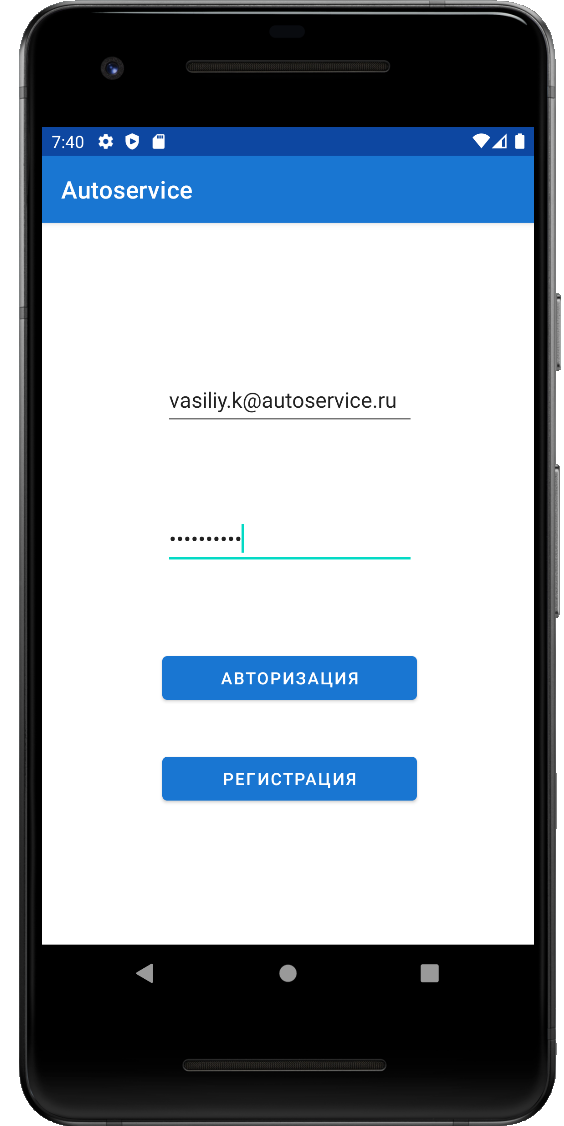


Рисунок 6. Процесс входа в приложение

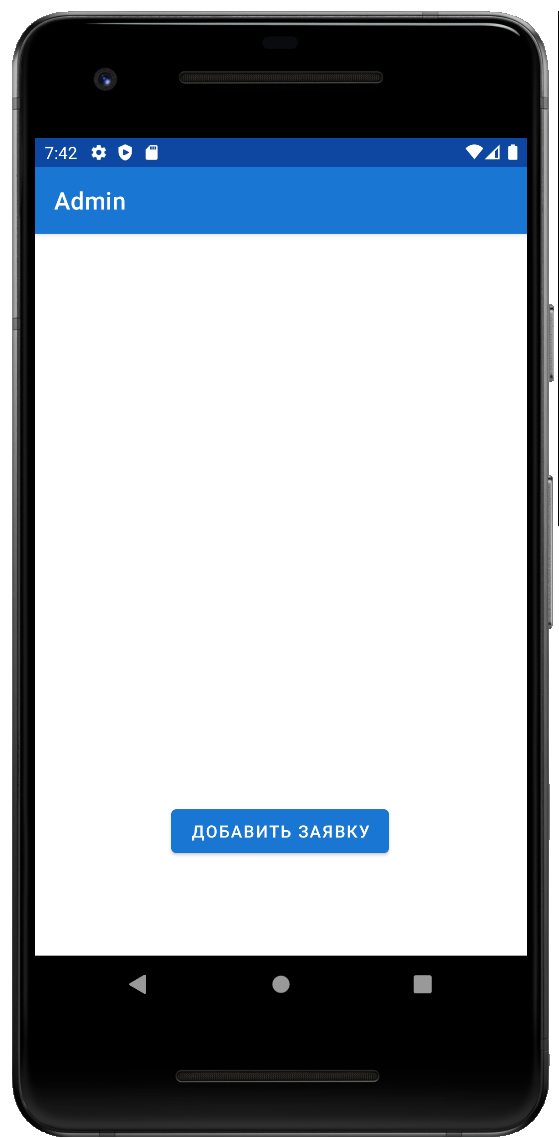


Рисунок 7. Основной экран входа после авторизации администратора

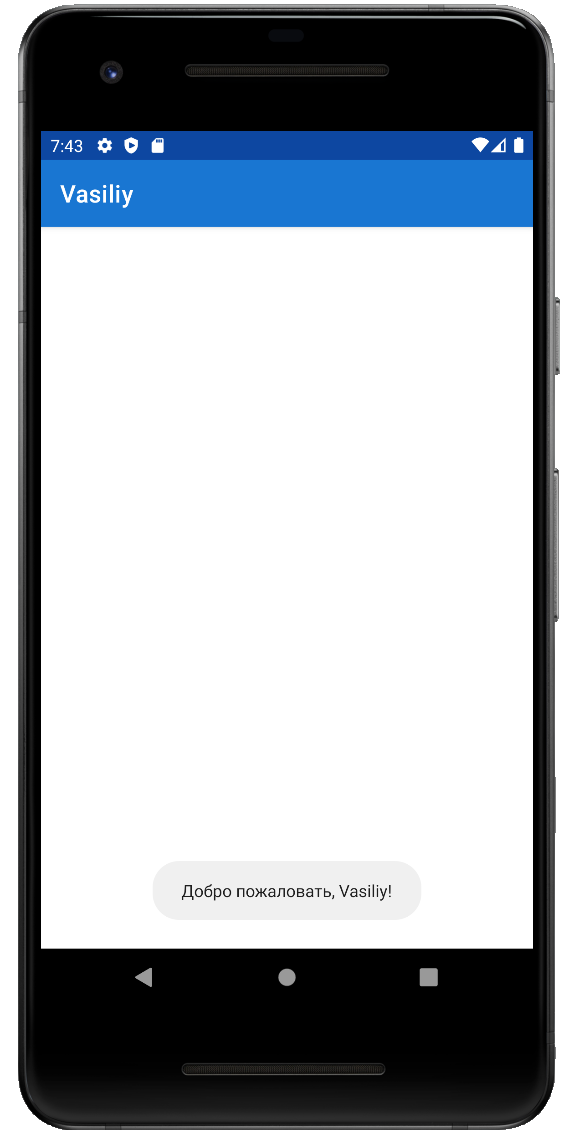


Рисунок 8. Основной экран входа после авторизации работника

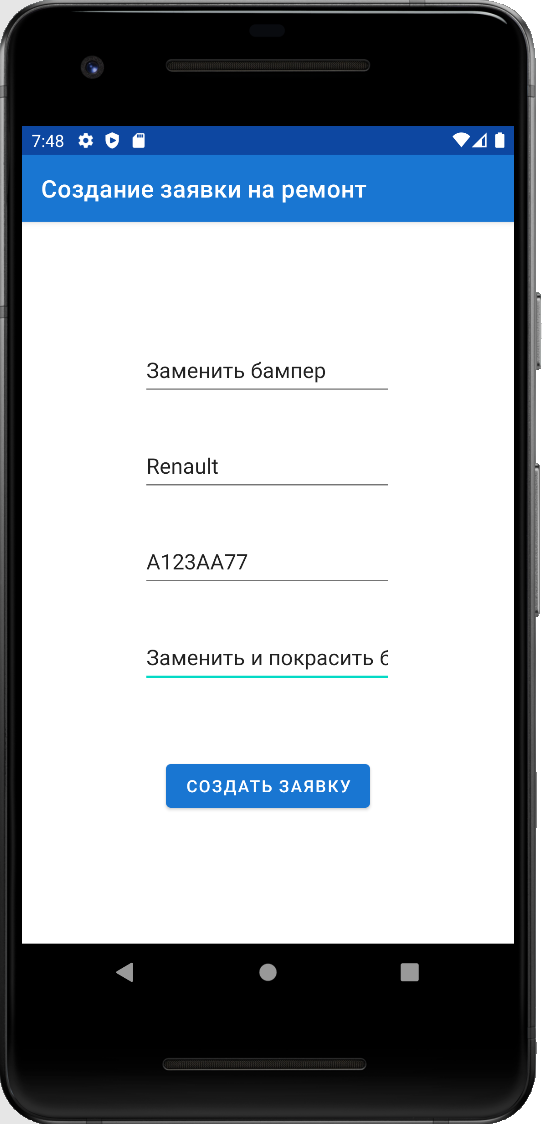


Рисунок 9. Экран для добавления заявки на ремонт

На этой странице администратор может добавить заявку на ремонт автомобиля клиента.

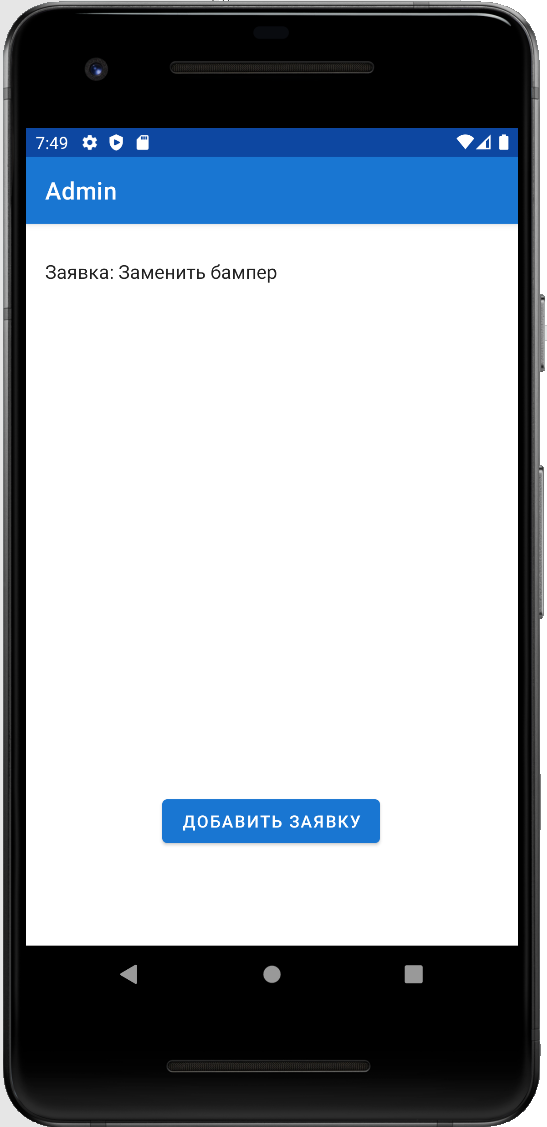


Рисунок 10. Главная страница администратора



Рисунок 11. Главная страница работника

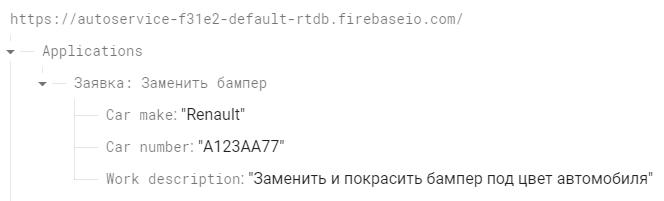


Рисунок 12. Realtime Database платформы Firebase

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной курсовой работы были выполнены следующие пункты:

* Была выбрана и утверждена задача;
* Было произведено расширенное описание задание на естественном языке.
* Было разработано техническое задание по выбранному заданию и были определены требования к системы;
* Было создано мобильное приложением с применением ЯП JAVA.
* Были произведены тестирование и отладка разработанного мобильного приложения;
* Была оформлена пояснительной записки на курсовую работу;
* Курсовая работа была сдана на проверку;
* Курсовая работа была защищена.

Выполнение всех вышеуказанных пунктов позволило мне достигнуть поставленной цели и выполнить все изначальные задачи.

# **Список используемых источников**

1. Авдеев В.А. Организация ЭВМ и перифирия с демонстрацией имитационных моделей. – М.: ДМК, 2014. – 708 с.
2. Антамошкин, О. А. Программная инженерия. Теория и практика. Учебник. M: НИЦ Инфра-М, 2012. – 368 с.
3. Дейтел, Х., М. Операционные системы. Основы и принципы. Т. 1 – М.: Бином, 2016. - 1024 c.
4. Дейтел, Х., М. Операционные системы. Т. 2. Распределенные системы, сети, безопасность. М.: Бином, 2016. - 704 c
5. Таненбаум, Э. Современные операционные системы / Э. Таненбаум. СПб.: Питер, 2013. - 1120 c.
6. С.А.Орлов. Программная инженерия. Учебник для вузов. 5-е издание обновленное и дополненное.М: Издательский дом «Питер»,2017. 812 с.
7. Дейв Тейлор. Сценарии командной оболочки. Linux, OS X и Unix. 2-е издание. Издательский дом «Питер»,2017.624 с.
8. SWEBOK V2, 2004 г
9. SWEBOK V3, 2013 г
10. <https://firebase.google.com/docs?hl=uk> – Firebase Documentation
11. <https://code.tutsplus.com/ru/tutorials/get-started-with-firebase-for-android--cms-27248> - “Начинаем работать с Firebase на Android”
12. [https://support.google.com/firebase/?hl=ru#topic=6399725](https://support.google.com/firebase/?hl=ru%23topic=6399725) – Справка - Firebase