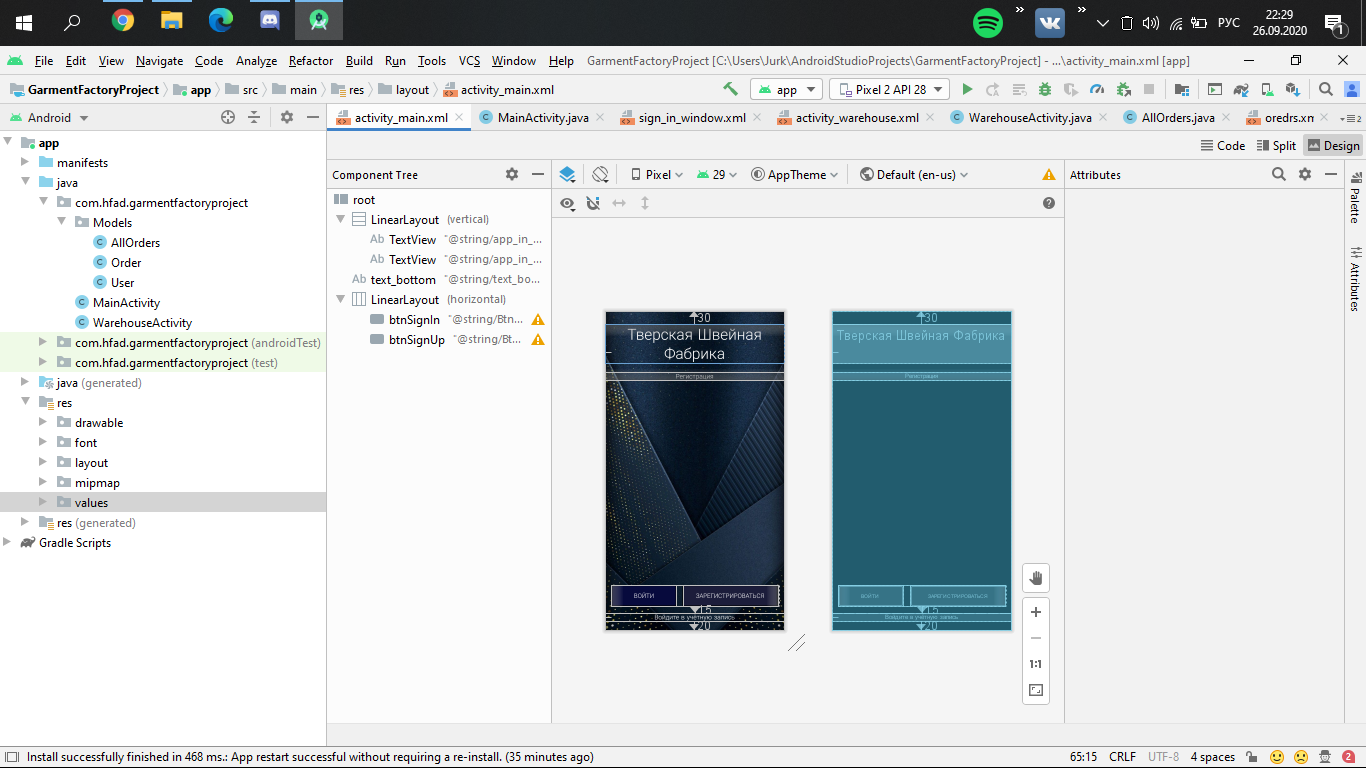
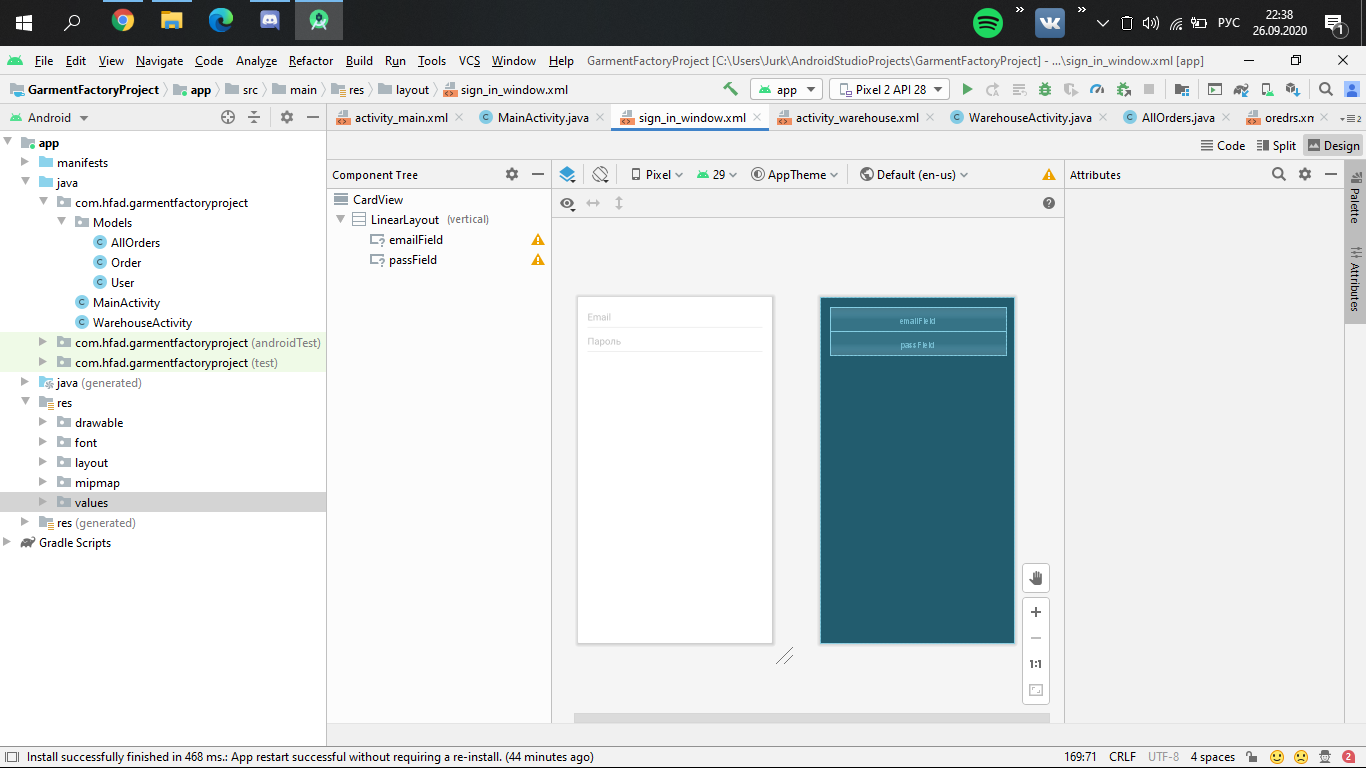
**Глава 4. Внедрение программного обеспечения**

4.1. Разработка клиентской части (модулей) и интерфейса приложения

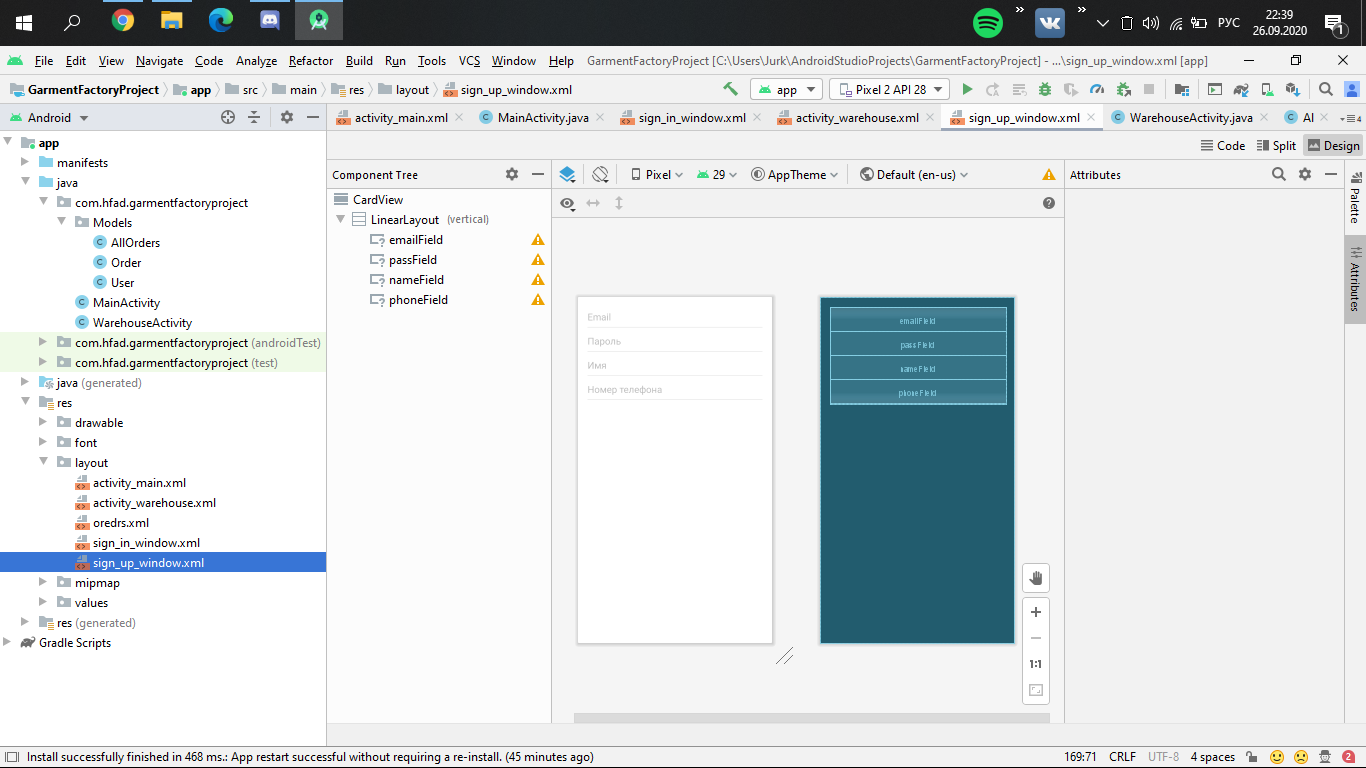
1. Главная форма с кнопками авторизации и регистрации, с дальнейшим перенапровлением на другую форму

**package** com.hfad.garmentfactoryproject;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.Button;  
import android.widget.RelativeLayout;  
  
import com.google.android.gms.tasks.OnFailureListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.material.snackbar.Snackbar;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.hfad.garmentfactoryproject.Models.User;  
import com.rengwuxian.materialedittext.MaterialEditText;  
  
import java.util.concurrent.ConcurrentSkipListMap;  
  
public class MainActivity **extends** AppCompatActivity {  
  
 Button **btnSignIn**, **btnSignUp**;  
 FirebaseAuth **auth**;  
 FirebaseDatabase **db**;  
 DatabaseReference **users**;  
  
 RelativeLayout **root**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **btnSignIn** = findViewById(R.id.***btnSignIn***);  
 **btnSignUp** = findViewById(R.id.***btnSignUp***);  
  
 **root** = findViewById(R.id.***root***);  
  
 **auth** = FirebaseAuth.*getInstance*();  
 **db** = FirebaseDatabase.*getInstance*();  
 **users** = **db**.getReference(**"Users"**);  
  
 **btnSignUp**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 showSighUpWindow();  
 }  
 });  
 **btnSignIn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 showSignInWindow();  
 }  
 });  
 }  
  
 *//авторизация*  
**private void** showSignInWindow(){  
 AlertDialog.Builder dialog = **new** AlertDialog.Builder(**this**);  
 dialog.setTitle(**"Войти"**);  
 dialog.setMessage(**"Введите данные для входа"**);  
  
 LayoutInflater inflater = LayoutInflater.*from*(**this**);  
 View SignInWindow = inflater.inflate(R.layout.***sign\_in\_window***, **null**);  
 dialog.setView(SignInWindow);  
  
 **final** MaterialEditText email = SignInWindow.findViewById(R.id.***emailField***);  
 **final** MaterialEditText password = SignInWindow.findViewById(R.id.***passField***);  
  
 dialog.setNegativeButton(**"Отменить"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** which) {  
 dialogInterface.dismiss();  
 }  
 });  
  
 dialog.setPositiveButton(**"Войти"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** which) {  
 **if**(TextUtils.*isEmpty*(email.getText().toString())){  
 Snackbar.*make*(**root**, **"Введите ваш email"**, Snackbar.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
  
 **if**(password.getText().toString().length() < 5){  
 Snackbar.*make*(**root**, **"Введите пароль более 5 символов"**, Snackbar.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
  
 **auth**.signInWithEmailAndPassword(email.getText().toString(), password.getText().toString())  
 .addOnSuccessListener(**new** OnSuccessListener<AuthResult>() {  
 @Override  
 **public void** onSuccess(AuthResult authResult) {  
 startActivity(**new** Intent(MainActivity.**this**, WarehouseActivity.**class**));  
 finish();  
 }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 Snackbar.*make*(**root**, **"Ошибка авторизации."** + e.getMessage(), Snackbar.***LENGTH\_SHORT***);  
 }  
 });  
 }  
 });  
  
 dialog.show();  
 }  
  
 **private void** showSighUpWindow() {  
 AlertDialog.Builder dialog = **new** AlertDialog.Builder(**this**);  
 dialog.setTitle(**"Зарегистрироваться"**);  
 dialog.setMessage(**"Введите все данные"**);  
  
 LayoutInflater inflater = LayoutInflater.*from*(**this**);  
 View SignUpWindow = inflater.inflate(R.layout.***sign\_up\_window***, **null**);  
 dialog.setView(SignUpWindow);  
  
 **final** MaterialEditText email = SignUpWindow.findViewById(R.id.***emailField***);  
 **final** MaterialEditText password = SignUpWindow.findViewById(R.id.***passField***);  
 **final** MaterialEditText name = SignUpWindow.findViewById(R.id.***nameField***);  
 **final** MaterialEditText phone = SignUpWindow.findViewById(R.id.***phoneField***);  
  
 dialog.setNegativeButton(**"Отменить"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** which) {  
 dialogInterface.dismiss();  
 }  
 });  
  
 dialog.setPositiveButton(**"Зарегистрироваться"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** which) {  
  
 *//проверки*  
**if**(TextUtils.*isEmpty*(email.getText().toString())){  
 Snackbar.*make*(**root**, **"Введите ваш email"**, Snackbar.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
  
 **if**(TextUtils.*isEmpty*(name.getText().toString())){  
 Snackbar.*make*(**root**, **"Введите ваше имя"**, Snackbar.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
  
 **if**(TextUtils.*isEmpty*(phone.getText().toString())){  
 Snackbar.*make*(**root**, **"Введите ваш номер"**, Snackbar.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
  
 **if**(password.getText().toString().length() < 5){  
 Snackbar.*make*(**root**, **"Введите пароль более 5 символов"**, Snackbar.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
  
 *// регистрация*  
**auth**.createUserWithEmailAndPassword(email.getText().toString(), password.getText().toString())  
 .addOnSuccessListener(**new** OnSuccessListener<AuthResult>() {  
 @Override  
 **public void** onSuccess(AuthResult authResult) {  
 User user = **new** User();  
 user.setEmail(email.getText().toString());  
 user.setName(name.getText().toString());  
 user.setPassword(password.getText().toString());  
 user.setPhone(phone.getText().toString());  
  
 **users**.child(FirebaseAuth.*getInstance*().getCurrentUser().getUid())  
 .setValue(user)  
 .addOnSuccessListener(**new** OnSuccessListener<Void>() {  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 Snackbar.*make*(**root**, **"Пользователь зарегистрирован"**, Snackbar.***LENGTH\_SHORT***);  
 }  
 });  
 }  
 });  
 }  
 });  
  
 dialog.show();  
  
 }  
}

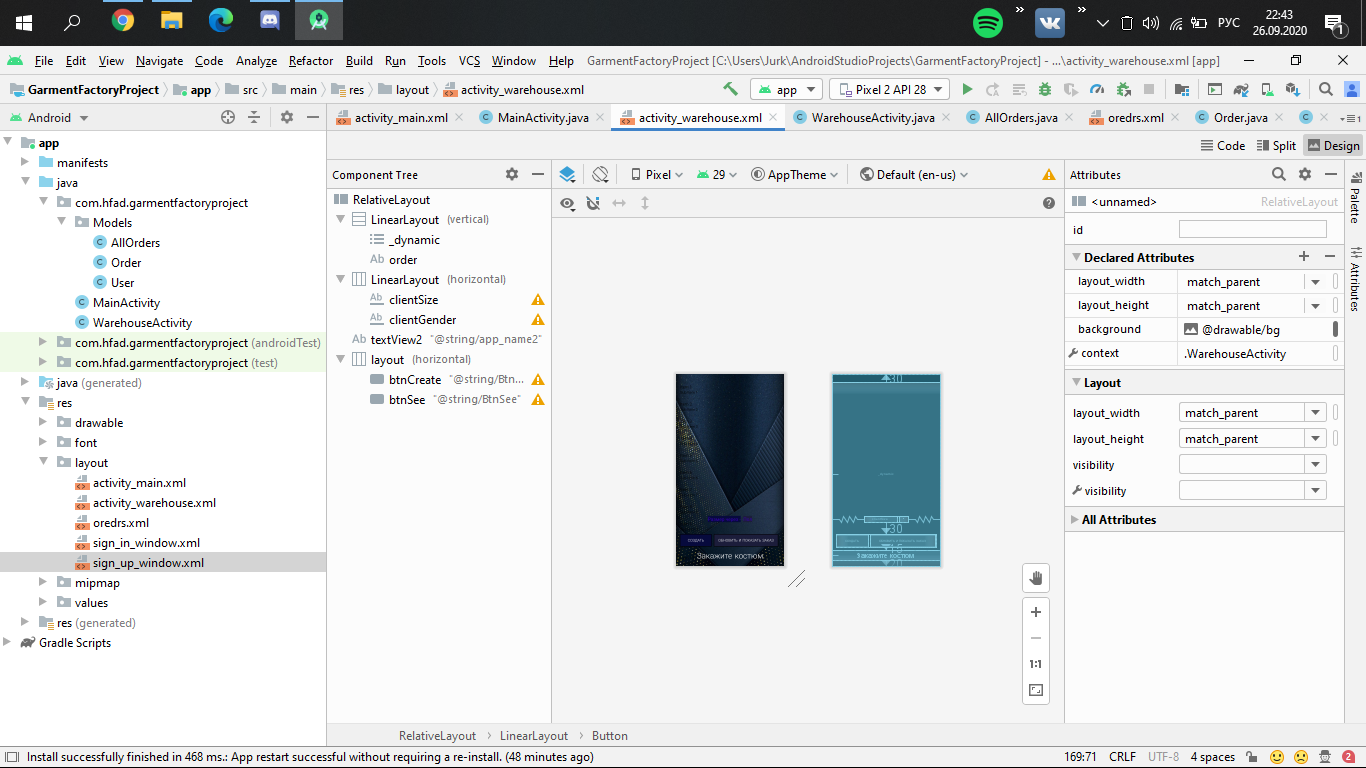
1. Форма авторизации



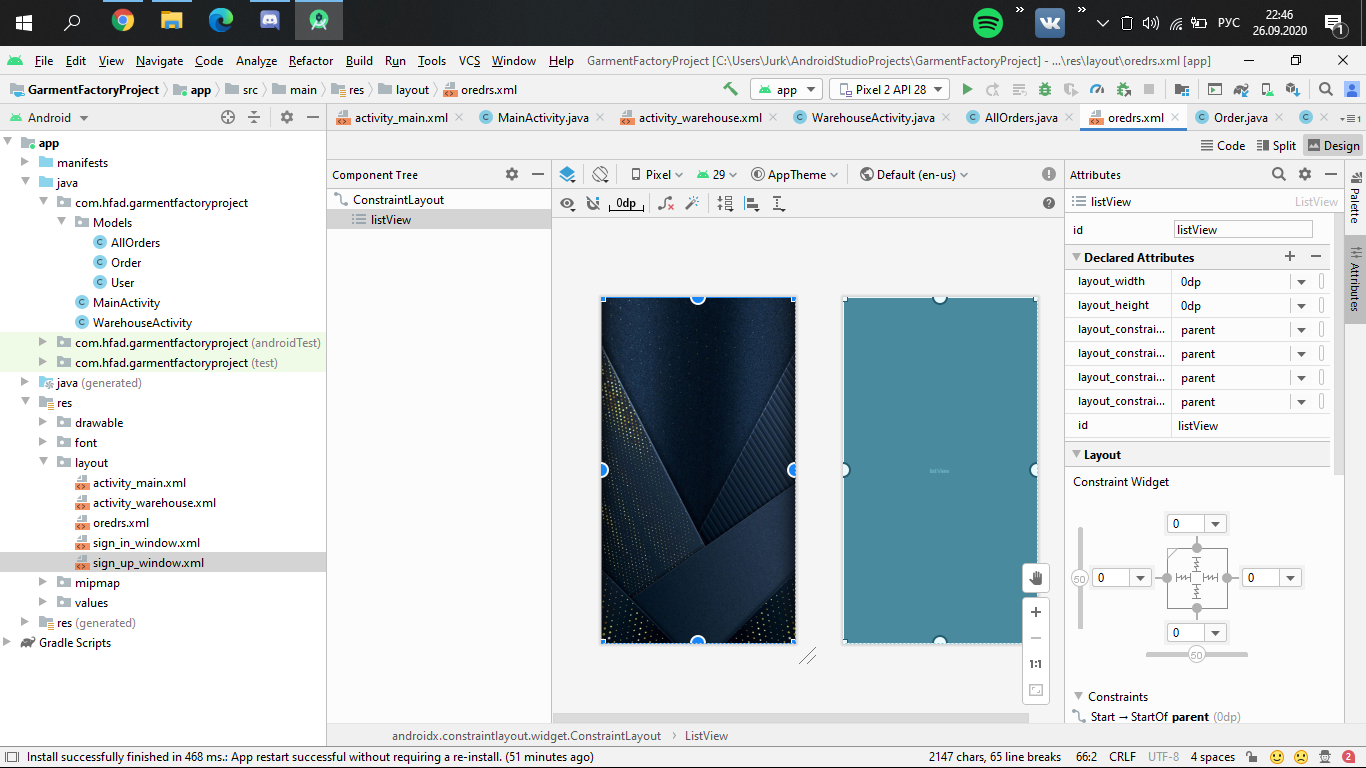
1. Форма регистрации



1. Форма созданий заказов и кнопка перенаправления на форму со всеми вашими заказами

**package** com.hfad.garmentfactoryproject;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.hfad.garmentfactoryproject.Models.AllOrders;  
import com.hfad.garmentfactoryproject.Models.Order;  
  
  
public class WarehouseActivity **extends** AppCompatActivity {  
  
 **private** EditText **editSize**, **editGender**;  
 **private** DatabaseReference **mDataBase**;  
 **private** String **ORDER\_KEY** = **"Order"**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_warehouse***);  
 init();  
 }  
  
 **private void** init() {  
 **editSize** = findViewById(R.id.***clientSize***);  
 **editGender** = findViewById(R.id.***clientGender***);  
 **mDataBase** = FirebaseDatabase.*getInstance*().getReference(**ORDER\_KEY**);  
 }  
  
 *//кнопка для сохранения*  
**public void** onClickSave(View view) {  
 String id = **mDataBase**.getKey();  
 String size = **editSize**.getText().toString();  
 String gender = **editGender**.getText().toString();  
 Order newOrder = **new** Order(id, size, gender);  
 *//проверка на пустые поля*  
**if** (!TextUtils.*isEmpty*(size) && !TextUtils.*isEmpty*(gender)) {  
 **mDataBase**.push().setValue(newOrder);  
 }  
 **else**{  
 Toast.*makeText*(**this**, **"Пустое поле"**, Toast.***LENGTH\_SHORT***).show();  
 }  
  
 }  
  
 *//кнопка открытия всех заказов (вылетает)*  
**public void** onClickRead(View view) {  
 Intent i = **new** Intent(WarehouseActivity.**this**, AllOrders.**class**);  
 startActivity(i);  
 }  
  
}

1. Форма со всеми вашими заказами



**package** com.hfad.garmentfactoryproject.Models;  
  
import android.os.Bundle;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
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 com.hfad.garmentfactoryproject.R;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class AllOrders **extends** AppCompatActivity {  
  
 **private** ListView **listView**;  
 **private** ArrayAdapter<String> **adapter**;  
 **private** List<String> **listData**;  
  
 **private** DatabaseReference **mDataBase**;  
 **private** String **ORDER\_KEY** = **"Order"**;  
  
 **protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***oredrs***);  
 init();  
 getDataFromDB();  
 }  
  
 **private void** init() {  
 **listView** = findViewById(R.id.***listView***);  
 **listData** = **new** ArrayList<>();  
 **adapter** = **new** ArrayAdapter<>(**this**,android.R.layout.***simple\_list\_item\_1***, **listData**);  
 **listView**.setAdapter(**adapter**);  
 **mDataBase** = FirebaseDatabase.*getInstance*().getReference(**ORDER\_KEY**);  
 }  
 *//читаем базу данных*  
**private void** getDataFromDB() {  
 ValueEventListener vListener = **new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **if**(**listData**.size() > 0)**listData**.clear();  
 **for**(DataSnapshot ds : dataSnapshot.getChildren()) {  
 Order orders = ds.getValue(Order.**class**);  
 **assert** orders != **null**;  
 **listData**.add(orders.**size**);  
 }  
 **adapter**.notifyDataSetChanged();  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError error) {  
  
 }  
 };  
 **mDataBase**.addValueEventListener(vListener);  
 }  
  
}

1. Код класса Order (содержит информацию о заказе)

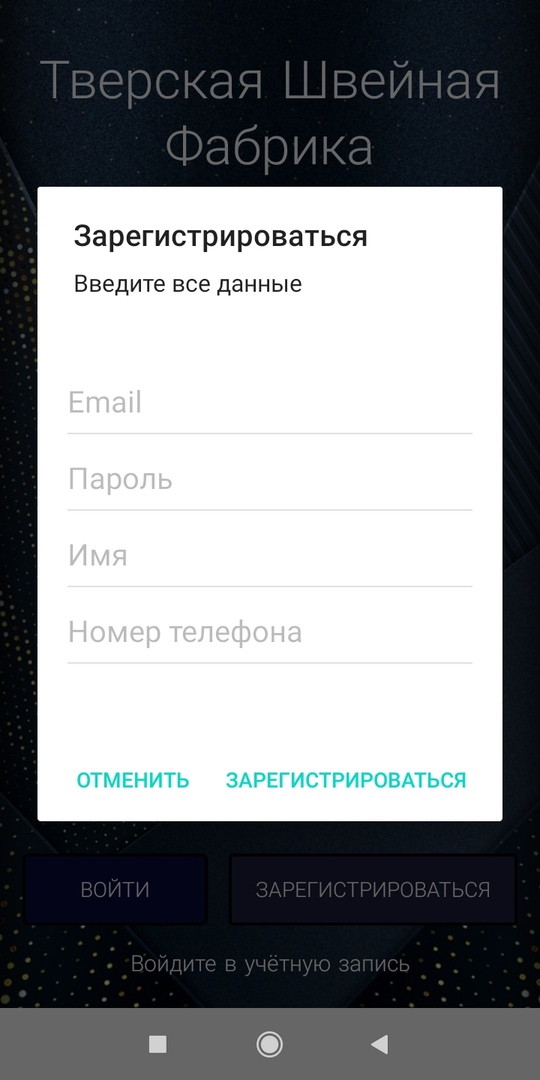
**package** com.hfad.garmentfactoryproject.Models;  
  
public class Order {  
 **public** String **id**, **size**, **gender**;  
  
 **public** Order() {  
 }  
  
 **public** Order(String id, String size, String gender) {  
 **this**.**id** = id;  
 **this**.**size** = size;  
 **this**.**gender** = gender;  
 }  
}

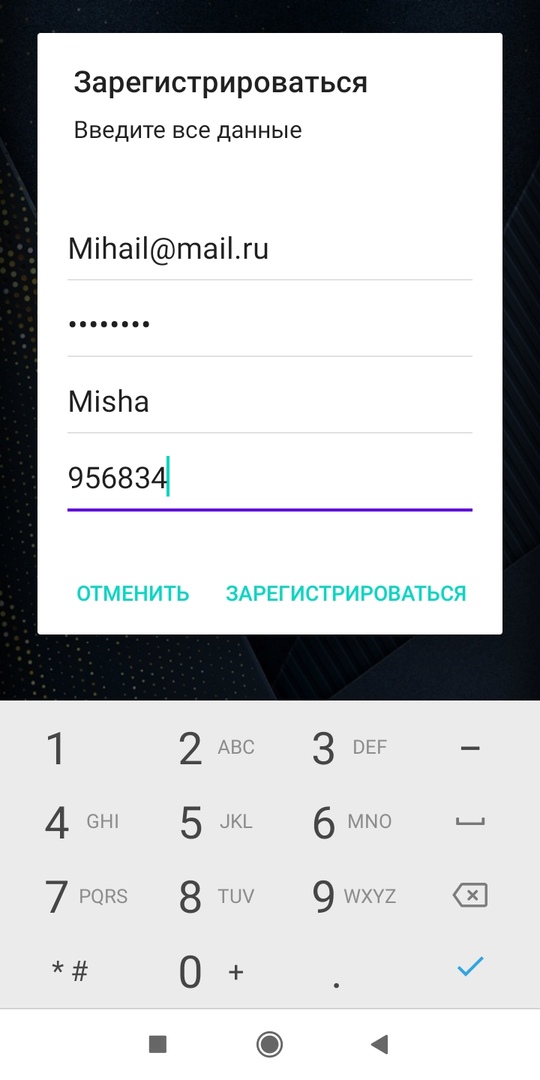
1. Код класса User (содержит информацию о пользователе, и записывает их в таблицу)

**package** com.hfad.garmentfactoryproject.Models;  
  
public class User {  
 **private** String **name**, **email**, **password**, **phone**;  
  
 **public** User() {}  
  
 **public** User(String name, String email, String password, String phone) {  
 **this**.**name** = name;  
 **this**.**email** = email;  
 **this**.**password** = password;  
 **this**.**phone** = phone;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
  
 **public** String getEmail() {  
 **return email**;  
 }  
  
 **public void** setEmail(String email) {  
 **this**.**email** = email;  
 }  
  
 **public** String getPassword() {  
 **return password**;  
 }  
  
 **public void** setPassword(String password) {  
 **this**.**password** = password;  
 }  
  
 **public** String getPhone() {  
 **return phone**;  
 }  
  
 **public void** setPhone(String phone) {  
 **this**.**phone** = phone;  
 }  
}

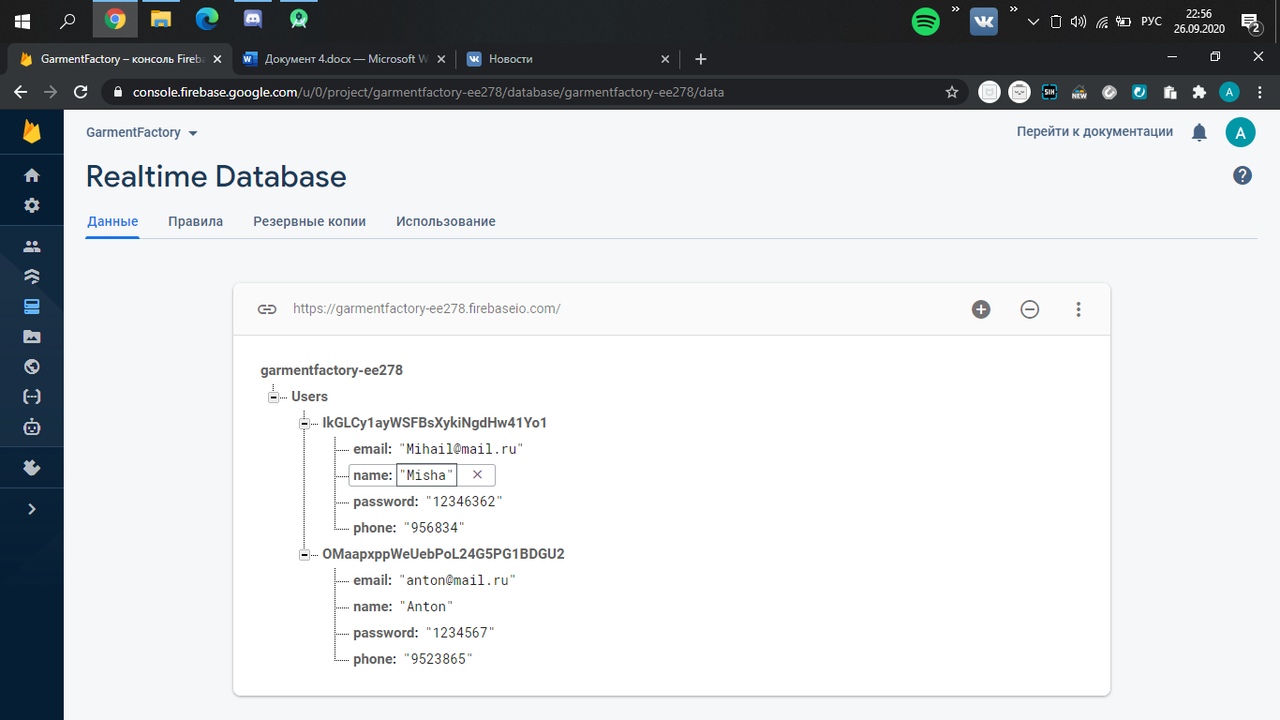
4.2 Оформление документации и тестирование созданного приложения

1. Регистрация

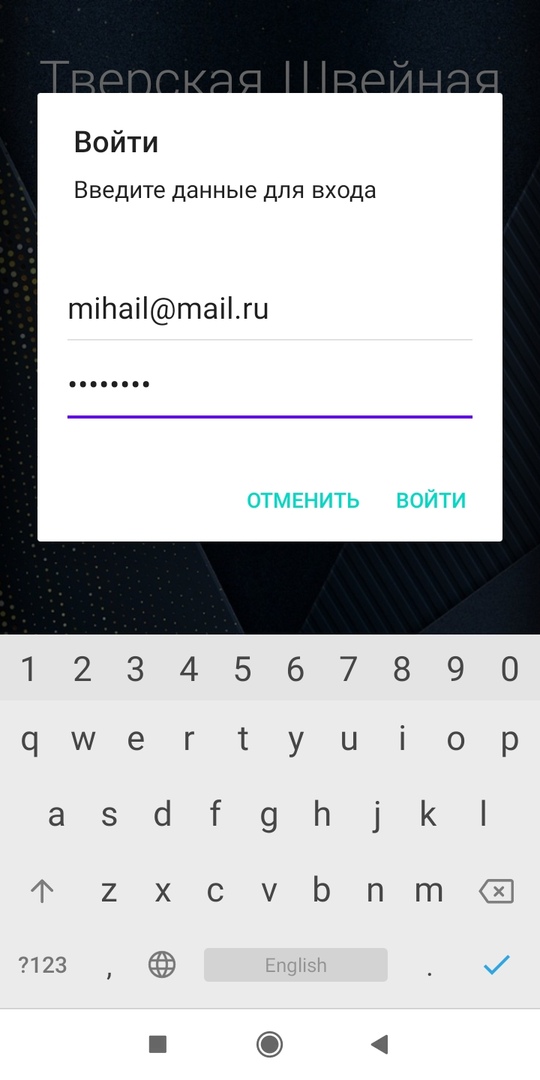




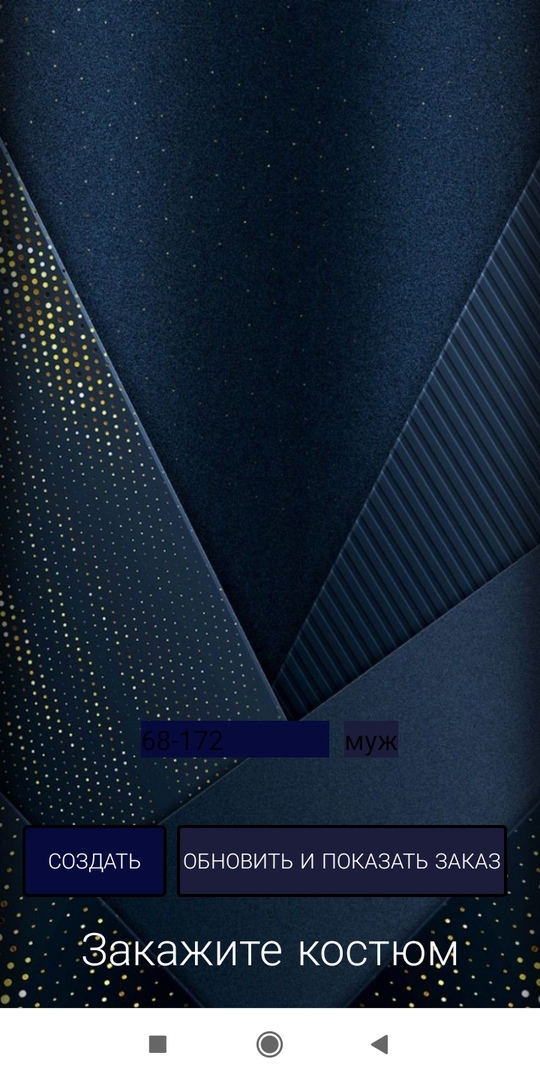
1. Появляется запись в базе данных



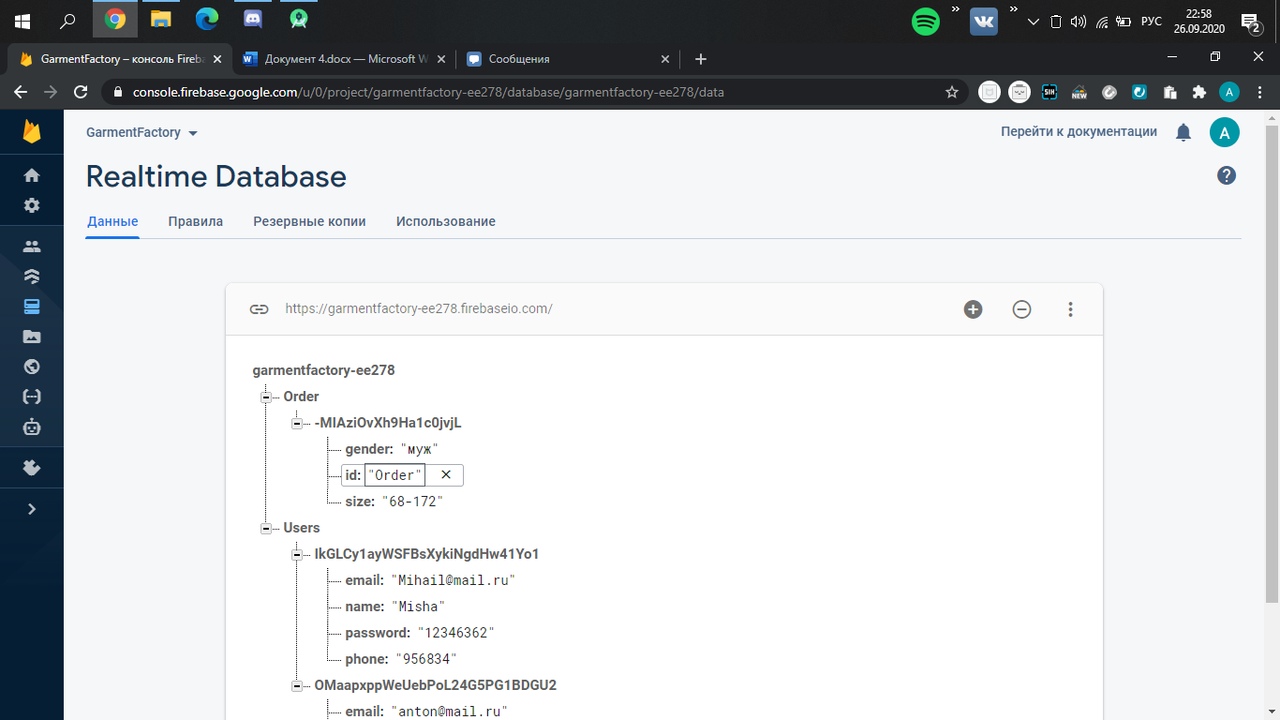
1. Авторизация



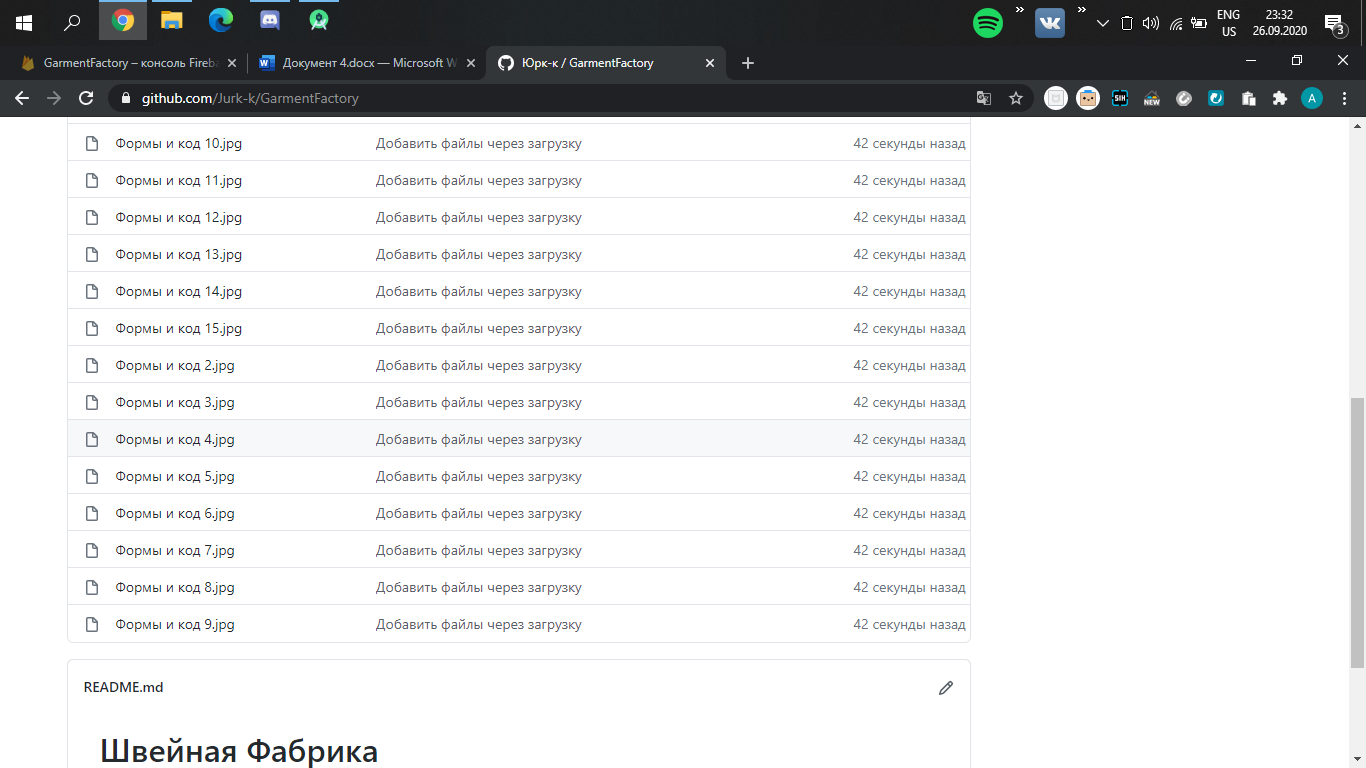
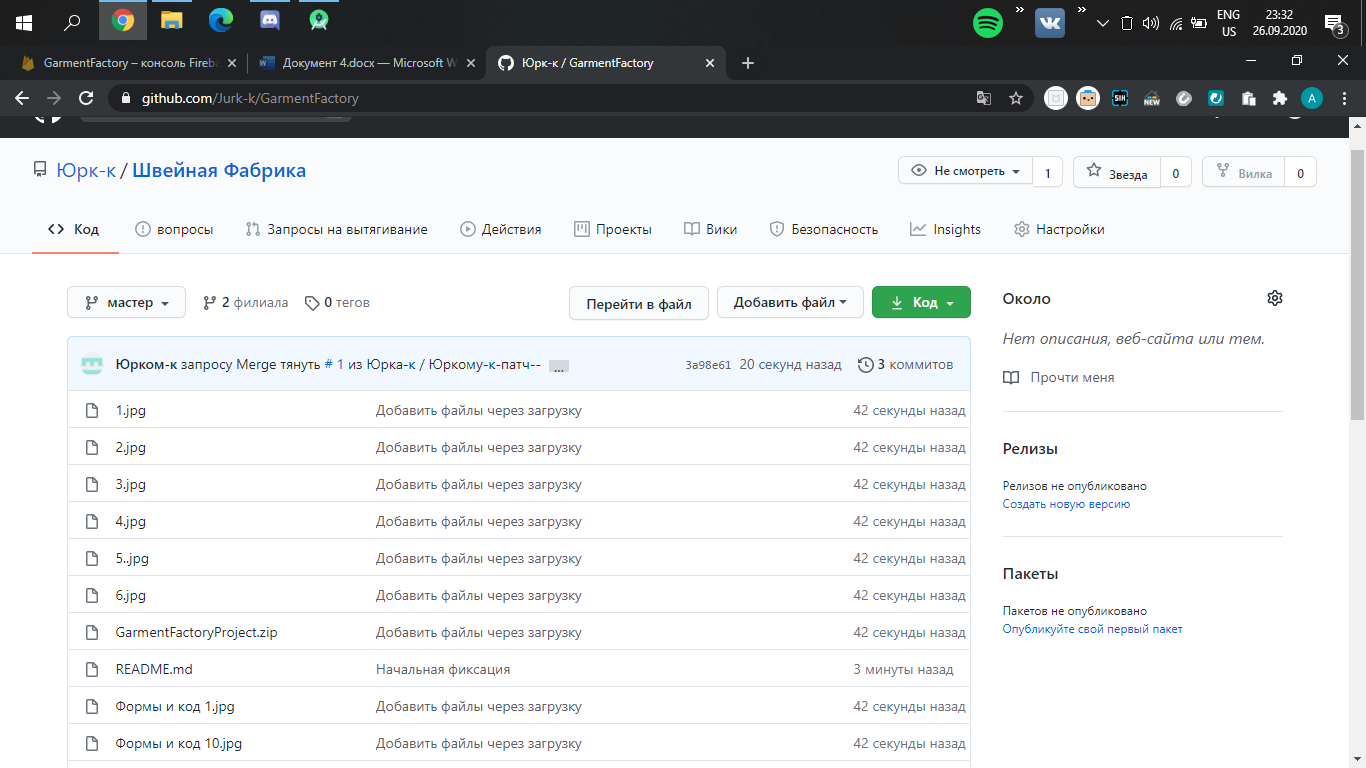
1. Создание заказа



1. Появляется запись заказа в базе данных



4.3 Размещение проекта. Использование системы контроля версий



https://github.com/Jurk-k/GarmentFactory