Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

**разработка** **мобильного приложения**

**вИЗИТКИ**

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

МДК.01.03 Разработка мобильных приложений

НАТКиГ.201000.010.000ПЗ

Разработал:

студент группы ПР-20.101

Каргин А.А.

2022

**Содержание**

[Введение 3](#_Toc117112970)

[1 Исследовательский раздел 4](#_Toc117112971)

[1.1 Описание предметной области 4](#_Toc117112972)

[1.2 Образ клиента 5](#_Toc117112973)

[1.3 Сценарии 5](#_Toc117112974)

[1.4 Сбор и анализ прототипов 6](#_Toc117112975)

[2 Проектирование приложения 11](#_Toc117112976)

[2.1 UI/UX дизайн проекта 11](#_Toc117112977)

[2.2 Выбор технологии, языка и среды программирования 18](#_Toc117112978)

[3 Разработка мобильного приложения 19](#_Toc117112979)

[3.1 Разработка базы данных 19](#_Toc117112980)

[3.2 Разработка используемых плагинов 20](#_Toc117112981)

[3.3 Описание разработанных процедур и функций 21](#_Toc117112982)

[4 Тестирование 26](#_Toc117112983)

[4.1 Протокол тестирования дизайна приложения 26](#_Toc117112984)

[4.2 Протокол тестирования функционала приложения 26](#_Toc117112985)

[Заключение 29](#_Toc117112986)

[Библиография 30](#_Toc117112987)

[Приложение А 31](#_Toc117112988)

Введение

# Исследовательский раздел

## Описание предметной области

Люди постоянно обмениваются информацией, иногда это становится частью работы. Реклама. Это очень интересный инструмент, сейчас реклама всё больше переходит в цифровой мир, но при всём этом сейчас много кто использует бумагу чтобы, например чтобы поделиться контактными данными или предоставляемыми услугами.

Визитные карты — носитель контактной информации о человеке или организации, обычно изготавливающийся из бумаги или картона. Их часто используют чтобы предоставить конечному клиенту информацию о компании, сфере деятельности, перечне предоставляемых услуг, контактные телефоны, карту проезда, адрес web-страницы. Сейчас каждый человек имеет в руках мощный вычислительный и портативный аппарат — мобильный телефон. С его помощью человек может в кратчайшие сроки обмениваться информацией разными способами, как через интернет, так и через QR— коды или Bluetooth. Таким образом, имея программное обеспечение, позволяющее через QR— код поделиться контактной информацией с клиентом, количество проблем связанных с стоимостью и ненадежностью бумажного носителя были бы решены.

Создав приложение, позволяющее создать визитную карту, делиться ей при помощи QR— кода, и также хранить список полученных визитных карт, можно раз и навсегда решить проблемы бумажных визиток

## Образ клиента

Клиентами являются предприниматели малого и большого бизнеса, обычные, ведь приложение могут использовать даже студенты, чтобы поделиться друг с другом контактной информацией. Но наибольший спрос ожидается именно у предпринимателей, так как именно им чаще всего приходится сталкиваться с подобного рода проблемами.

По данным GEM 2019-2020гг. Средний возраст предпринимателей с устоявшимся бизнесом — 42 года.

## Сценарии

Прежде всего, при составлении сценария выявляется цель, которой будут достигать пользователи. В первую очередь целью является простота и доступность. Например: к начинающему дизайнеру Олегу пришёл потенциальный клиент и ему необходимо поделиться своими данными с клиентом, и он имея приложение для визитных карт может предложить ему установить его и без необходимости вручную переписывать данные, поделиться информацией при помощи QR-кода

Также такое приложение может уменьшить затраты на листовки у промоутеров, вместо раздачи листовок они смогут делиться данными через QR-код.

Ещё для примера такое приложение могут использовать студенты, в том случае, когда они только поступили в учебное заведение им может потребоваться обменяться контактами друг друга, и если у них будет установлено приложение, то процесс обмена во многом может упроститься. Староста пройдет и просканирует QR-код каждого одногруппника и получит их контакты, ссылки на социальные сети, и другую информацию.

## Сбор и анализ прототипов

У данного мобильного приложения немного конкурентов, ввиду новизны идеи. В магазине приложений Google Play было найдено всего несколько таких приложений. Рассмотрим дизайн, ошибки и достижения найденного приложения.

Первым приложением, которое выдаёт поиск Google Play было “Создать визитка”. На рисунке (Рисунок 1) можно видеть главную картинку этого приложения из Google Play

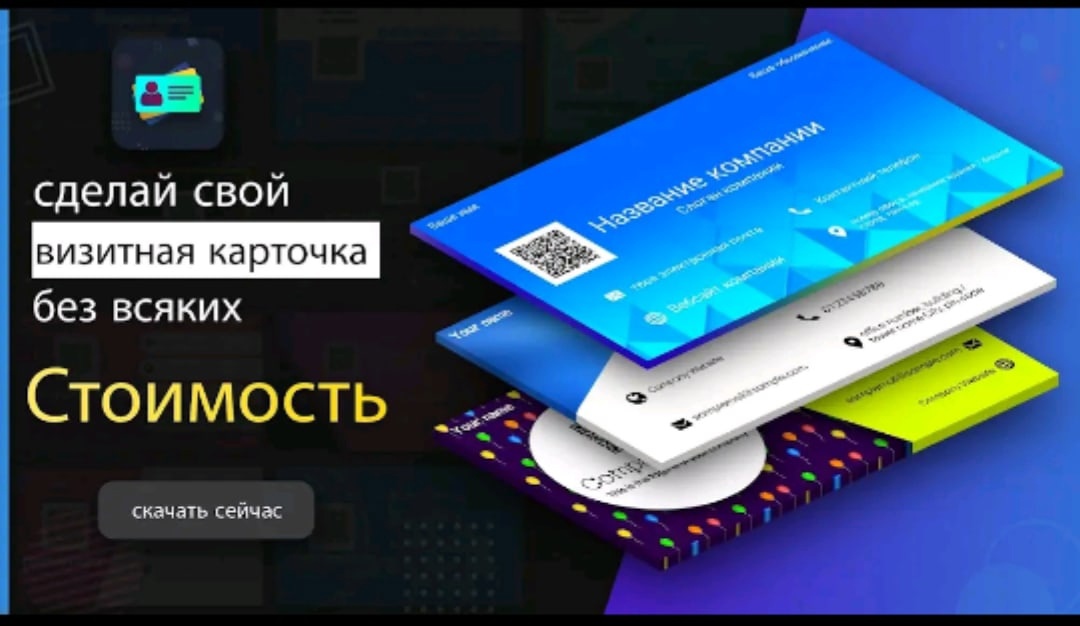


Рисунок 1 – Главная картинка в Google Play

Сама картинка уже создаёт странное впечатление о приложении, видно, что перевод выполнен некачественно. После установки приложения и его открытия. На рисунке (Рисунок 2) можно видеть главный экран приложения.

Рисунок 2 – Главный экран приложения

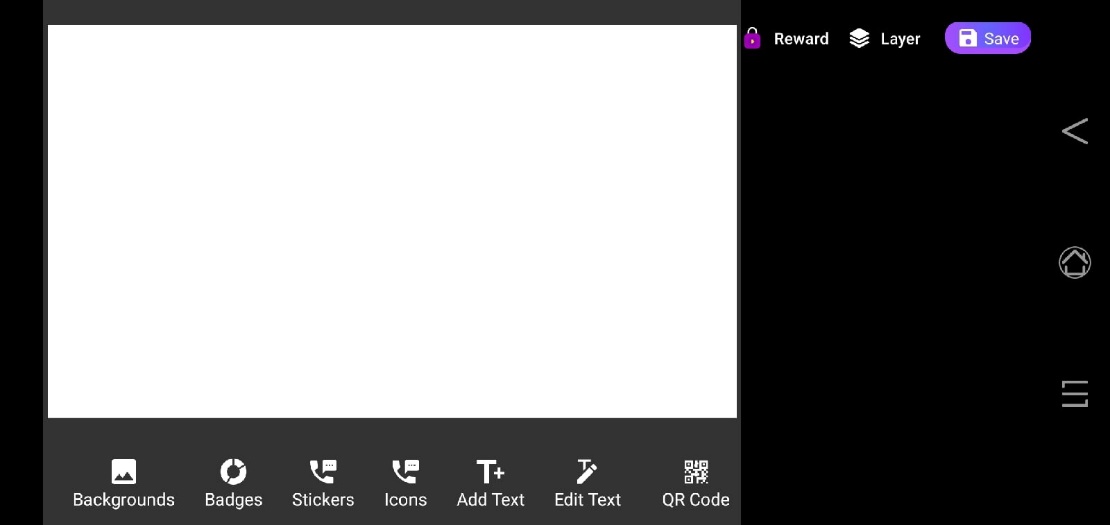
Сразу можно отметить, что пользоваться приложением можно только в альбомной ориентации, что не совсем удобно. Было бы правильней продумать возможность поворота экрана. Используя главное меню можно посмотреть уже созданные или создать «визитку» из шаблона. В приложении отсутствует функционал для того, чтобы делиться визитной карточкой через QR-код или Bluetooth. На картинке ниже (Рисунок 3) показан редактор для создания визитных карточек.

Рисунок 3 – Окно создания визитных карт

Редактор сделан достаточно хорошо, есть много элементов, но практически отсутствуют русские шрифты.

Исходя из всех минусов, можно сказать, что приложение не обладает ни достаточным функционалом, ни дизайном чтобы быть в реальном использовании

# Проектирование приложения

## UI/UX дизайн проекта

Дизайн проекта разработан в программе Figma.

Для проекта были определены основные экраны:

* экран первоначального заполнения данных о пользователе;
* экран стартового обучения;
* экран главной страницы;
* экран настроек;
* экран управления данными о пользователе;
* экран управления визитными картами;
* экран создания визитных карт;
* экран сканирования QR-кода.

Тёмный дизайн сейчас является весьма популярен у молодёжи, и имеет несколько преимуществ: такое решение поможет экономить заряд батареи на устройствах с AMOLED дисплеями, также темная тема является более безопасной для человеческого глаза. Крупные шрифты, большие элементы интерфейса будут удобны для использования людям с небольшими отклонениями зрения, а для обычного пользователя нажатие на элементы интерфейса не будет сопровождаться неудобством попадания по ним. Стоит заметить, что данные цвета не выглядят скучно, а весьма лаконично и позволяют пользователю сконцентрироваться именно на работе с приложением, не отвлекаясь на элементы дизайна

Ниже на рисунке (Рисунок 4) представлен логотип для приложения

Рисунок 4 – Логотип приложения

На логотипе, в примитивном представлении, изображены несколько абстрактных визитных карт и надпись «NEGO», которая образована от латинского «NEGOTIUM», что означает «дело»

На рисунке (Рисунок 5) ниже представлена страница первоначального заполнения данных о пользователе.

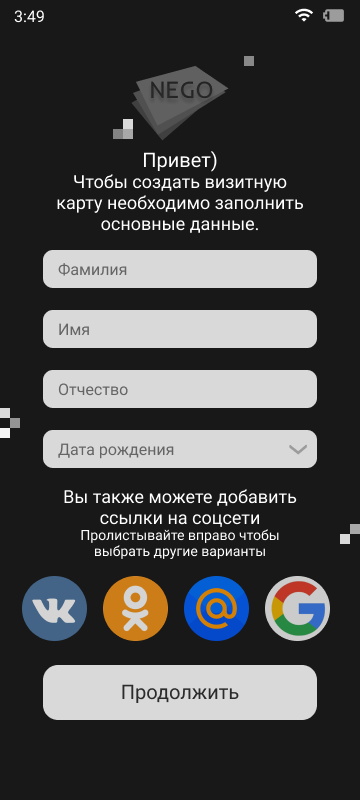


Рисунок 5 – Страница «Первоначальное заполнение данных о пользователе»

На данной странице мобильного приложения можно заполнить данные, которые потом можно будет использовать для быстрого заполнения визитных карт.

На рисунке (Рисунок 6) ниже представлена страница стартового обучения.



Рисунок 6 – Страница «Стартового обучения»

На данной странице пользователя уведомляют о функциях основных элементов интерфейса. Можно увидеть, что снизу находится меню, при помощи которого можно выполнять навигацию по приложению

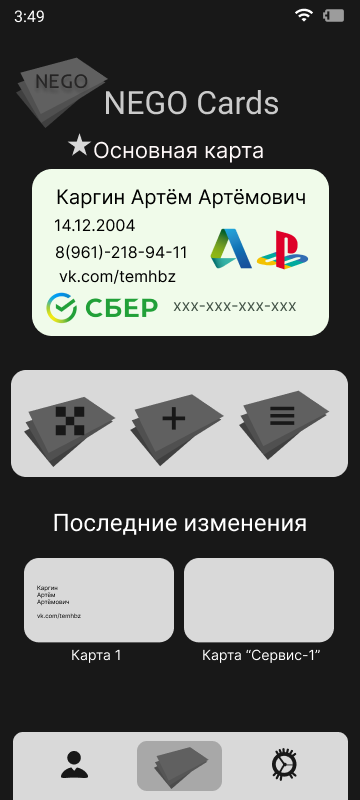
На рисунке (Рисунок 7) ниже представлена главная страница.

Рисунок 7 – Страница «Главная страница»

На данной странице выводится главная информация приложения имеется функционал для взаимодействия с визитными картами, также выводятся последние редактируемые карты, и та карта, которую пользователь пометил как избранную

На рисунке (Рисунок 10) ниже представлена страница сканирования QR-кода.



Рисунок 10 – Страница «Чата»

На данной странице имеется возможность отсканировать QR-код визитной карты.

## Выбор технологии, языка и среды программирования

Средой программирования выбрана программа Android Studio. И всё написание кода будет выполняться на Kotlin, так как это достаточно новый язык программирования и даёт мне возможности «нативной» разработки, что является необходимостью ведь в данном приложении будет выполняться работа с внутренними компонентами устройства. На начальном этапе только с камерой, но в продолжительной перспективе будет необходима работа с NFC или bluetooth

# Разработка мобильного приложения

## Разработка базы данных

К проекту был подключён сервис Firebase посредством помощника Firebase в программе Android Studio и прописывания кода в файлах проекта.

С помощью Authentication Firebase была реализована аутентификация пользователей (Рисунок 13), условием которой является наличие пароля и электронной почты у пользователя. Так же при заполнении полей регистрации происходит проверка заполнения: валидация пароля и электронной почты.

Рисунок 13 – Аутентификация пользователей

В ходе анализа были выявлены сущности: запись на приём, чат, пользователи, услуги.

На рисунке (Рисунок 14) ниже представлена таблица «Пользователи», которая включает в себя контакт пользователя и VIN его автомобиля.



Рисунок 14 – Таблица «Пользователи»

На рисунке (Рисунок 15) ниже представлена таблица «Запись на приём», которая включает в себя название услуги, дату записи и VIN обслуживаемого автомобиля.

Рисунок 15 – Таблица «Запись на приём»

## Разработка используемых плагинов

Ниже приведен листинг (Листинг 1), на котором изображено название эмулятора для запуска приложения в Android Studio.



Листинг 1 – Эмулятор

Так же были использованы, как встраиваемые плагины, так и дополнительные. Ниже приведен листинг, (Листинг 2) используемых плагинов.



Листинг 2 – Плагины

Добавленные плагины предназначаются для подключения Firebase, RealTime Database, Auth и навигационного меню.

## Описание разработанных процедур и функций

В приложении разработаны следующие функции:

* регистрация и авторизация;
* переход между страницами с помощью навигационного меню;
* запись на приём;
* просмотр услуг автосервиса;
* появление календаря по нажатию на после с выбором даты;
* чат.

Ниже представлен листинг (Листинг 4) кода, который отвечает за регистрацию пользователя в системе.

public class MainActivity extends AppCompatActivity {  
  
 EditText inputemail, inputpassword, cconformpassword;  
 Button registerBtn, vxodBtn;  
 String emailPattern = "[a-zA-z0-9.\_-]+@[a-z]+\\.+[a-z]+";  
 ProgressDialog progressDialog;  
  
 FirebaseAuth mAuth;  
 FirebaseUser mUser;  
  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 vxodBtn = findViewById(R.id.*vxodBtn*);  
 inputemail = findViewById(R.id. *inputemail*);  
 inputpassword = findViewById(R.id. *inputpassword*);  
 cconformpassword = findViewById(R.id.*cconformpassword*);  
 registerBtn = findViewById(R.id. *registerBtn*);  
 progressDialog = new ProgressDialog(this);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 mUser = mAuth.getCurrentUser();  
  
 vxodBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 startActivity(new Intent(MainActivity.this, login.class));  
 }  
 });  
  
  
 registerBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 PerforAuth();  
 }  
 });  
 }  
  
  
  
 private void PerforAuth() {  
 String email = inputemail.getText().toString();  
 String password = inputpassword.getText().toString();  
 String conformpassword = cconformpassword.getText().toString();  
  
 if(password.isEmpty() || password.length()<6)  
 {  
 inputpassword.setError("Введите пароль, не менее 6 символов");  
 } else if(!password.equals(conformpassword))  
 {  
 cconformpassword.setError("Пароли не совпадают");  
 }  
 else{  
 progressDialog.setMessage("Пожалуйста, подождите");  
 progressDialog.setTitle("Регистрация");  
 progressDialog.setCanceledOnTouchOutside(false);  
 progressDialog.show();  
  
  
  
 mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful())  
 {  
 progressDialog.dismiss();  
 sendUserToNextActivity();  
 Toast.*makeText*(MainActivity.this, "Регистрация прошла успешно", Toast.*LENGTH\_LONG*).show();  
  
 }  
 else {  
 progressDialog.dismiss();  
 Toast.*makeText*(MainActivity.this, ""+task.getException(), Toast.*LENGTH\_LONG*).show();  
  
 }  
 }  
 });  
 }  
 }  
  
 private void sendUserToNextActivity() {  
 Intent intent = new Intent(MainActivity.this, Home.class);  
 intent.setFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*| Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(intent);  
 }  
  
  
}

Листинг 4 – Регистрация пользователя

Во время регистрации происходит проверка пароля пользователя на соответствие: пароль состоит не менее из 6 символов и совпадает с повторным ведением пароля. А также, если пользователь с данным адресом электронной почты уже зарегистрирован, пользователь получит предупреждение об этом.

Ниже на листинге (Листинг 5) кода представлен метод перехода на другую активность.

vxodBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 startActivity(new Intent(MainActivity.this, login.class));  
 }  
 });  
  
  
 registerBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 PerforAuth();  
 }  
 });  
}

Листинг 5 – Переход на другие активности

Ниже на листинге (Листинг 6) представлен код, отвечающий за авторизацию пользователя в системе.

public class login extends AppCompatActivity {  
  
 EditText inputemail, inputpassword;  
 Button loginBtn;  
 Button createNewAccount;  
 String emailPattern = "[a-zA-z0-9.\_-]+@[a-z]+\\.+[a-z]+";  
 ProgressDialog progressDialog;  
  
 FirebaseAuth mAuth;  
 FirebaseUser mUser;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
  
 inputemail = findViewById(R.id.*inputemail*);  
 inputpassword = findViewById(R.id.*inputpassword*);  
 loginBtn = findViewById(R.id.*loginBtn*);  
 createNewAccount = findViewById(R.id.*createNewAccount*);  
  
 progressDialog = new ProgressDialog(this);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 mUser = mAuth.getCurrentUser();  
  
  
 createNewAccount.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 startActivity(new Intent(login.this, MainActivity.class));  
 }  
 });  
  
 loginBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 perforLogin();  
 }  
 });  
  
  
 }  
  
 private void perforLogin() {  
  
 String email = inputemail.getText().toString();  
 String password = inputpassword.getText().toString();  
  
 if (password.isEmpty() || password.length() < 6) {  
 inputpassword.setError("Введите пароль, не менее 6 символов");  
 } else {  
 progressDialog.setMessage("Пожалуйста, подождите");  
 progressDialog.setTitle("Вход");  
 progressDialog.setCanceledOnTouchOutside(false);  
 progressDialog.show();  
  
 mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 progressDialog.dismiss();  
 Toast.*makeText*(login.this, "Вход выполнен", Toast.*LENGTH\_LONG*).show();  
 }else  
 {  
 progressDialog.dismiss();  
 Toast.*makeText*(login.this, ""+task.getException(), Toast.*LENGTH\_LONG*).show();  
  
 }  
 }  
 });  
 }  
  
 }  
  
 private void sendUserToNextActivity() {  
 Intent intent = new Intent(login.this, Home.class);  
 intent.setFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*| Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(intent);  
 }  
  
}

Листинг 6 – Авторизация пользователя

Авторизация была выполнена с помощью сервиса Firebase Auth.

# Тестирование

## Протокол тестирования дизайна приложения

Для проведения тестирования дизайна приложение было выбрано устройство Honor 10i c разрешение экрана 2340×1080.

Проверка была проведена на:

* оптимальный размер кнопок;
* читабельный размер шрифта;
* приятное глазу цветовое сочетание;
* корректное расположение элементов;

## Протокол тестирования функционала приложения

Для проверки правильности ввода пароля при регистрации были проведены UnitTests, которые изображены ниже на листинге (Листинг 3):

public class PasswordUtils {  
  
 private static final String *VALID\_PASSWORD\_REGEX* = "((?=.\*\\d)(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[@#$%=:\\?]).{8,12})";  
  
 @Test  
 public void validatePassword\_EmptyString() {  
 *// setup* String password = "";  
  
 *// execute* boolean actual = PasswordUtils.*validatePassword*(password);  
  
 *// assert  
 assertFalse*(actual);  
 }  
  
 @Test  
 public void validatePassword\_Null() {  
 *// setup* String password = null;  
  
 *// execute* boolean actual = PasswordUtils.*validatePassword*(password);  
  
 *// assert  
 assertFalse*(actual);  
 }  
  
 @Test  
 public void validatePassword\_Missing\_OneNumber() {  
 *// setup* String password = "Abcdefg#";  
  
 *// execute* boolean actual = PasswordUtils.*validatePassword*(password);  
  
 *// assert  
 assertFalse*(actual);  
 }  
  
 @Test  
 public void validatePassword\_LengthTooLong() {  
 *// setup* String password = "Abcdefg5#abcdefgabcd";  
  
 *// execute* boolean actual = PasswordUtils.*validatePassword*(password);  
  
 *// assert  
 assertFalse*(actual);  
 }  
  
 @Test  
 public void validatePassword\_Missing\_OneUpperCaseLetter() {  
 *// setup* String password = "abcdefg5#";  
  
 *// execute* boolean actual = PasswordUtils.*validatePassword*(password);  
  
 *// assert  
 assertFalse*(actual);  
 }  
   
  
  
 public static boolean validatePassword(String password) {  
 return (password != null && password.matches(*VALID\_PASSWORD\_REGEX*));  
 }  
}

Листинг 3 – UnitTests

Так же для тестирования функционала были разработаны TestCase.

Test-Case №1

Название тест кейса: Регистрация

Предусловия тест кейса: приложение открыто02

Шаги тест кейса:

* 1. шаг тест кейс №1:открыть страницу регистрации;
  2. шаг тест кейса №2: заполнить все поля;
  3. шаг тест кейса №3: нажать на кнопку «регистрация».

Ожидаемый результат тест кейса:

Пользователь зарегистрирован в системе.

Статус кейса: выполнено.

Test-Case №2

Название тест кейса: Авторизация

Предусловия тест кейса: пользователь зарегистрирован

Шаги тест кейса:

* 1. шаг тест кейс №1: открыть страницу авторизации;
  2. шаг тест кейса №2: заполнить все поля;
  3. шаг тест кейса №3: нажать на кнопку «Вход».

Ожидаемый результат тест кейса:

Пользователь войдёт в систему.

Статус кейса: выполнено.

Test-Case №3

Название тест кейса: Переход на другую активность

Предусловия тест кейса: пользователь вошёл в систему

Шаги тест кейса:

1. шаг тест кейс №1: открыть страницу с навигацией;
2. шаг тест кейса №2: нажать на желаемую кнопку.

Ожидаемый результат тест кейса:

Пользователь перешёл на необходимую ему страницу.

Статус кейса: выполнено.

Заключение

В результате работы было разработано клиентское мобильное приложение для автосервиса.

Во время разработки возникли трудности с подключением сервиса Firebase к проекту, проблема была решена изменением Android на AndroidX. AndroidX – это переработанная библиотека, чтобы сделать имена пакетов более понятными.

В дальнейшем планируется совершенствовать приложение и добавить уже к имеющемуся функционал.

Библиография

1. Android Developers [электронный ресурс]: Документация Android Stusio – https://developer.android.com/docs/
2. Help Center [электронный ресурс]: Документация Java: https://docs.oracle.com/en/java/
3. Автостат [электронный ресурс]: Структура российских автовладельцев по возрасту и маркам https://www.autostat.ru/infographics/40710/
4. Firebase [электронный ресурс]: Документация Firebase - https://firebase.google.com/docs?hl=uk
5. Material Design [электронный ресурс]: Bottom Navigation: https://material.io/components/bottom-navigation/android
6. Git [электронный ресурс]: Документация Git: https://git-scm.com/
7. Figma [электронный ресурс]: Прототипирование проектов https://www.figma.com/

Приложение А

Техническое задание

Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б.С. Галущака»

**разработка мобильного приложения для автосервиса**

НАТКиГ.202100.010.000ПЗ

Выполнил:

студент группы

ПР-20.106

Петрищева А.В.

2022

**Содержание**

Введение 34

1 Назначения разработки 35

2 Требования к мобильному приложению 36

2.1 Требования к функциональным характеристикам 36

2.2 Требования к надёжности 36

2.3 Условия эксплуатации 37

2.4 Требования к составу и параметрам технических средств 2337

2.5 Требования к информационной и программной совместимости 2337

2.6 Требования к защите информации 38

2.7 Требования к маркировке и упаковке 2338

3 Требования к программной документации 2338

4 Технико-экономические показатели 2338

5 Стадии и этапы разработки 23638

6 Порядок контроля и приёмки 2339

**Введение**

Настоящее техническое задание распространяется на разработку информационной системы «Разработка мобильного приложения для автосервиса», используемого для записи на приём, просмотра услуг, чата с оператором.

Наименование приложения: «Garage54».

Краткая характеристика области применения: приложение для автосервиса – полезное оснащение телефона автовладельца. При необходимости пользователь может просмотреть имеющиеся в автосервисе услуги, записаться на приём и обратиться за помощью к оператору.

Условные обозначения и сокращения:

БД – База данных;

ИС – Информационная система;

ОС – Операционная система;

СТО – Станция технического обслуживания.

Основанием для проведения разработки является Протокол №6 от 21 февраля 2022 года.

Наименование темы разработки – «Разработка мобильного приложения для автосервиса».

Условное обозначение темы разработки – «Автосервис – Garage54».

**1** **Назначение разработки**

Основное назначение мобильного приложения заключается в:

* осведомлении клиентов об услугах;
* возможности клиента записи на приём.

Лица, которые могут работать с данной Системой:

пользователь – просматривает услуги, регистрируется, авторизуется, оформляет запись;

администратор – добавляет и редактирует услуги, просматривает записи.

**2 Требования к мобильному приложению**

2.1 Требования к функциональным характеристикам

Требования к составу выполняемых функций:

* регистрация и авторизация пользователей;
* просмотр имеющихся услуг;
* запись на приём;
* чат с оператором.

Входные данные должны быть организованы в виде вводимого, в специальную форму, текста, соответствующего определённому шаблону. Данные, вводимые вручную, проверяются на корректность.

**2.2 Требования к надёжности**

Обеспечение устойчивого функционирования должно выполняться несколькими действиями:

* организация бесперебойного питания оборудования пользователя;
* использование лицензионного программного обеспечения;
* организация стабильного интернет-соединения.

Приложение должно контролировать входную информацию:

* соблюдение типов данных при заполнении полей;
* операции изменения, удаления и сохранения.

Время восстановления после отказа, вызванного сбоем электропитания технических средств, не фатальным сбоем ОС или файловой системы, не должно превышать 15 минут при соблюдении условий эксплуатации технических и программных средств и правильной настройки операционной системы.

**2.3 Условия эксплуатации**

Обслуживание ИС включает в себя:

1. информационное обслуживание – ввод и редактирование информации БД;
2. системное администрирование БД ИС.

**2.4 Требования к составу и параметрам технических средств**

Для работы приложения на устройстве требуется: объем свободной памяти не менее 3 ГБ, версия устройства Android 10 и выше.

**2.5 Требования к информационной и программной совместимости**

Проектирование взаимодействия с файловой системой должно быть выполнено в рамках разработки курсового проекта. При разработке взаимодействия с файловой системой должен быть использован язык программирования Java.

**2.6 Требования к защите информации**

Доступ к информации БД предоставляется только администратору базы данных.

**2.7** **Требования к маркировке и упаковке**

Требования к маркировке и упаковке не предъявляются.

**3 Требования к программной документации**

Состав программной документации должен включать в себя:

* техническое задание;
* проектную документацию.

**4 Технико-экономические показатели**

Экономические преимущества разработки и ориентировочная экономическая эффективность не рассчитывается.

**5 Стадии и этапы разработки**

Таблица 1 – Стадии разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название этапа | Срок, даты | Отчётность |
| 1 | Исследование предметной области | 25.03.2022-28.03.2022 | Пояснительная записка |
| 2 | Выбор моделей, описывающих предметную область | 01.04.2022-03.04.2022 | Пояснительная записка |
| 3 | Разработка технического задания | 03.04.2022-05.04.2022 | Техническое задание |
| 4 | Изучение Java и Firebase | 01.04.2022-  18.04.2022 | Пояснительная записка |
| 5 | Анализ требований и уточнение спецификаций | 14.04.2022 | Спецификации программного обеспечения |
| 6 | Проектирование структуры мобильного приложения, проектирование компонентов (технический проект) | 15.04.2022-29.04.2022 | Схема структурная системы и спецификации компонентов |
| 8 | Кодирование клиентской части | 01.05.2022-20.06.2022 | Программный  Продукт |
| 9 | Тестирование компонентов.  Сборка и комплексное тестирование | 20.06.2022 | Тексты программных компонентов |
| 10 | Разработка программной  документации | 24.06.2022 – 26.06.2022 | Программная  документация |
| 12 | Защита | 27.06.2022 |  |

**6 Порядок контроля и приёмки**

Виды испытаний – защита курсового проекта.

Общее требования к приёмке:

* техническое задание;
* пояснительная записка;
* программный продукт;
* презентация.