Министерство образования Новосибирской области

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

**Разработка мобильного приложения, осуществляющего захват и запись видео с экрана Android**

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

МДК.01.03 Разработка мобильных приложений

НАТКиГ.200900.010.000ПЗ

Разработал:

студент группы ПР-20.101

Иванов А.C.

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

Мобильное приложение для СТО делает взаимодействие с предприятием более удобным для клиента: в случае возникновения проблемы с автомобилем у пользователя не должно возникнуть трудности, куда обратиться. Клиент может записаться на обслуживание с помощью приложения, выбрав нужные услуги, дату и время. …

Вопрос, касающийся нахождения автосервиса, предоставляющего качественные услуги, является очень актуальным, ведь как машин, так и автосервисов с каждым днём становится всё больше, что затрудняет поиск необходимого.

Целью курсового проекта является создание мобильного приложения для компании по производству женской одежды.

Задачами курсового проекта в связи с указанной целью являются:

* изучение предметной области темы;
* рассмотрение приложения с точки зрения пользователя для выявления необходимых функций приложения;

Объект исследования – приложения для автосервисов.

Предмет исследования – изучение принципов функционирования и инструментов приложения.

# Исследовательский раздел

## Описание предметной области

В наше время практически у каждого человека есть смартфон, которым он в той или иной степени пользуется. В первую очередь не стоит забывать, что это устройство является карманным персональным компьютер, который дополнен функциональностью мобильного телефона. Это значит, что на смартфоне можно выполнять достаточное большинство необходимых действий. Периодически у любого владельца смартфона или планшета возникает необходимость поделиться с другим человеком видео из игры, закрытого профиля в социальной сети или системного интерфейса. Встроенной программы на Android для записи экрана чаще всего нет, но необходимость воспользоваться захватом и записью экрана может возникнуть в любой момент.

Создание записей с экрана телефона — очень удобная функция. Можно обмениваться наглядными советами, работой в приложениях, фишками устройства или других приложений, записывать урок и т.д. Все это может повысить полезность смартфона и упростить жизнь пользователю.

Для захвата и записи видео с экрана смартфона могут понадобиться такие функции, как:

* запись при нажатии соответствующей кнопки в приложении;
* включение оверлея и осуществление записи при его помощи;
* просмотр сохраненных видео;
* изменение настроек записи;
* редактирование записанных видео.

Функции захвата и записи видео с экрана смартфона отличаются от функций захвата и записи видео с экрана персонального компьютера тем, что в компьютерной функции можно выбрать конкретную область захвата, что предоставляет более удобную реализацию записи.

Мобильное приложение для записи видео с экрана упростит жизнь многим пользователям смартфона.

## Образ клиента

По статистике 94,9% интернет-пользователей возрастом от 16 до 64 года владеют смартфонами. То есть средний возраст пользователей смартфонами в России равен 40 годам. К такому выводу пришли специалисты мобильной рекламной платформы byyd в 2021 году.

Захват и запись видео с экрана могут потребоваться владельцу смартфона любого возраста, поэтому ограничения по возрасту отсутствуют. Так как активными пользователями смартфонов являются более молодые пользователи, то они составляют большую часть тех, кто хотел был воспользоваться захватом и записью видео с экрана смартфона.

## Сценарии

Мария Викторовна преподает математику. Ей необходимо записать обучающее видео по новой теме, потому что сегодня поставили дистанционное обучение. Но именно сегодня в доме отключили свет, и у неё нет возможности включить компьютер. А более чем достаточно будет скачать приложение и записать видео с экрана телефона, на котором будет решать математические задания в другом приложении.

Максим много играет в мобильные игры, поэтому его друзья часто просят его снять для них видео, чтобы они могли научиться играть. Для этого ему нужно воспользоваться захватом и записью экрана с его смартфона, но ему необходимо, чтобы видеозапись была неограниченна по времени и передавала звук игры, а встроенного приложения в его телефоне нет. Поэтому ему нужно скачать приложение, чтобы снять обучающее видео.

Василий является разработчиком электронной книги для мобильного устройства. Ему необходимо показать функционал своего приложения, чтобы заинтересовать пользователей и показать, что можно делать в приложении. Установка приложения поможет записать видео с функционалом разрабатываемого им приложения.

## Сбор и анализ прототипов

Существует большое количество приложений, осуществляющих захват и запись видео с экрана смартфона. Так как приложения получают свою популярность за счет функции записи и захвата экрана, то при выборе прототипов стоит руководствоваться популярностью приложений.

### 1.4.1 XRecorder

XRecorder имеет яркое цветовое решение приложения, преобладают оранжевый, чёрный и белый цвета.

На данной странице можно управлять записанными видео. Ниже приведен Рисунок 1 с изображением страницы с видео.

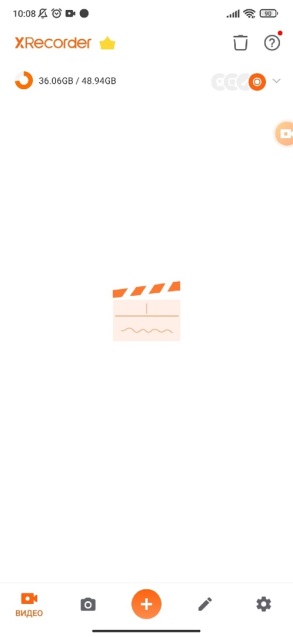


Рисунок 1 — Страница с видео

У приложения есть шапка, инструменты, информация о количестве доступной памяти на телефоне и навигационное меню в нижней части экрана. Шапку приложения составляет название приложения и кнопки для перехода в корзину, удаления и информации. В навигационном меню имеются кнопки для перехода между экранами «ВИДЕО», «ФОТО», «ИЗМЕНИТЬ», «НАСТРОЙКИ».

Ниже на Рисунок 2 представлена страница «ФОТО». На данной странице можно управлять сделанными фото.

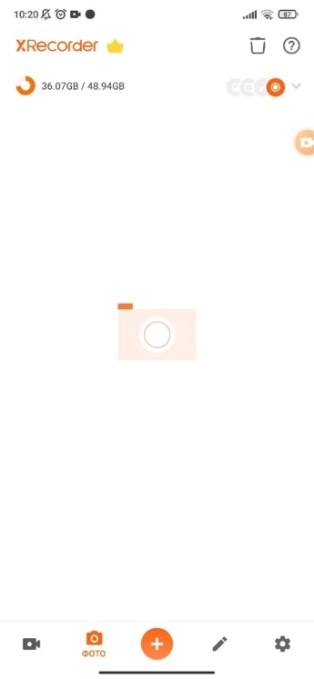


Рисунок 2 — Страница «ФОТО»

На Рисунок 3 ниже представлена страница «ИЗМЕНИТЬ». На данной странице можно выполнять различные действия с видео и фото.

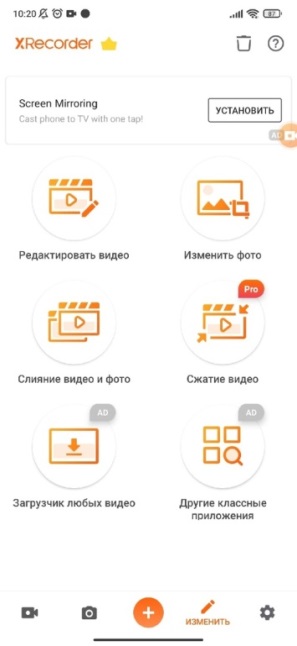


Рисунок 3 — Страница «ИЗМЕНИТЬ»

Ниже на Рисунок 4 представлена страница «НАСТРОЙКИ». Данная страница позволяет изменять некоторые настройки данного приложения.

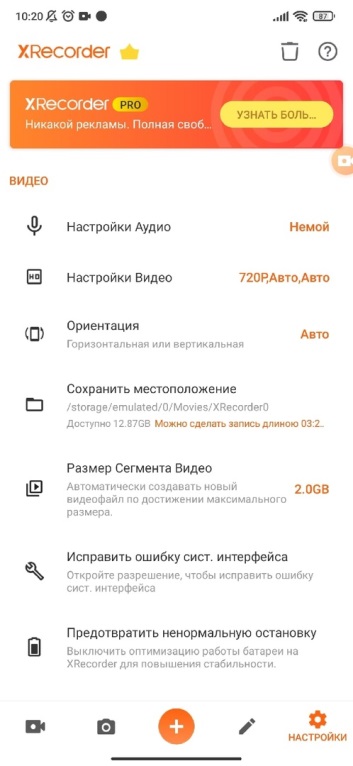


Рисунок 4 — Страница «НАСТРОЙКИ»

Кроме того, у приложения есть основная функция записи, которая представлена в виде оверлея и показана на Рисунок 5.

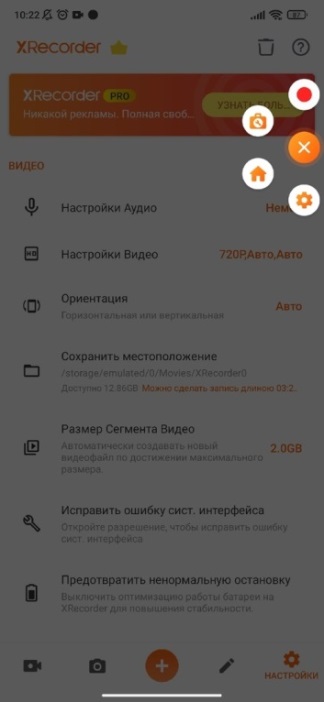


Рисунок 5 — Оверлей записи видео с экрана

### 1.4.2 Mobizen Screen Recorder

Данное приложение имеет удобный интерфейс и высокую степень настройки. Можно настроить качество видео, количество кадров, обрезать видео или добавить что-нибудь к нему, а так же написать название и указать путь. У данного приложения присутствует интуитивно понятный оверлей, который облегчает запись для пользователя.

Приложение «Mobizen Screen Recorder» изображено на рисунке 1.2.

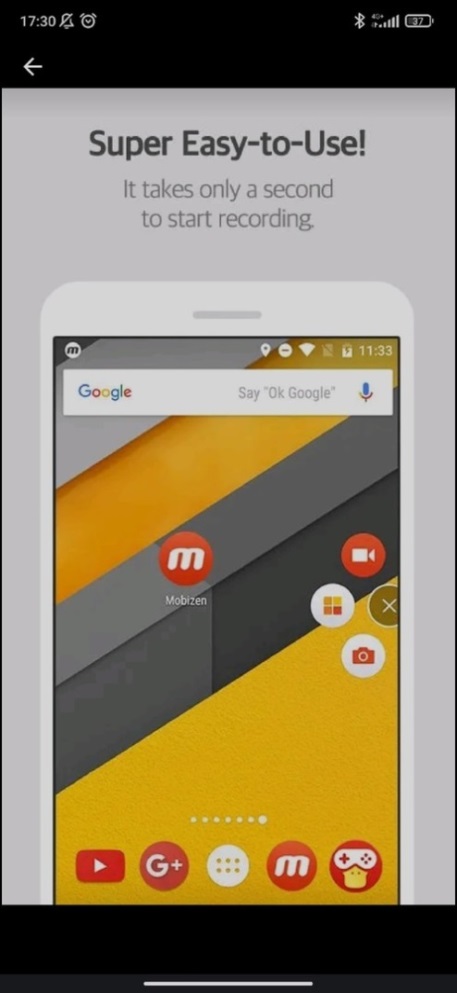


Рисунок 1.2 — Приложение «Mobizen Screen Recorder»

### 1.4.3 Запись Экрана

Самое простое приложение из рассмотренных. Оно состоит только из оверлея, который имеет понятный и простой интерфейс. Данное приложение не заставляет пользователя разбираться в функционале приложения, достаточно начать запись и в нужный момент ее закончить.

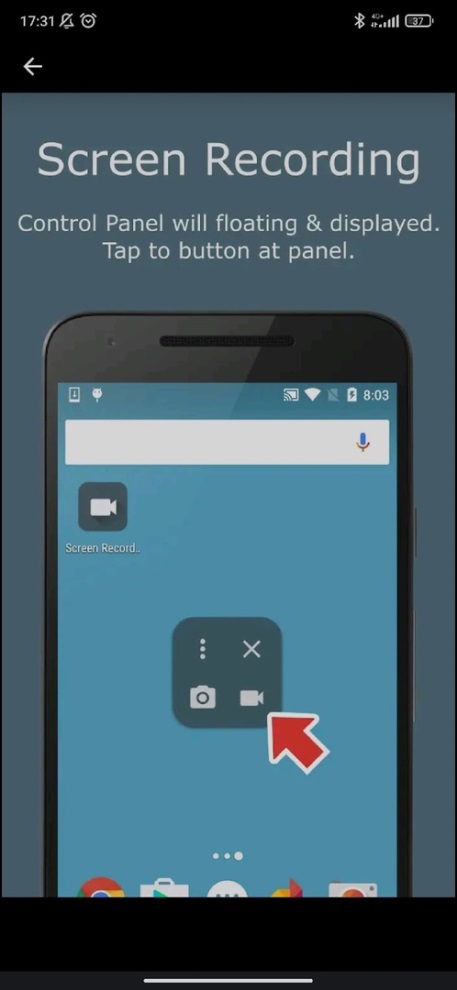


Рисунок 1.3 — Приложение «Запись Экрана»

### 1.4.4 Запись Экрана — снимок экрана

Данное приложение содержит в себе все необходимое для комфортной работы с фото и видео на телефоне. Приложение позволяет записывать видео без ограничения по времени, а так же гарантирует, что не будет собирать личную информацию от пользователей без разрешения. Можно настроить фронтальную камеру, которая будет снимать видео параллельно записи видео с экрана, а затем соединить оба видео при помощи редактора.

Приложение «Запись Экрана – снимок экрана» изображено на рисунке 1.4.

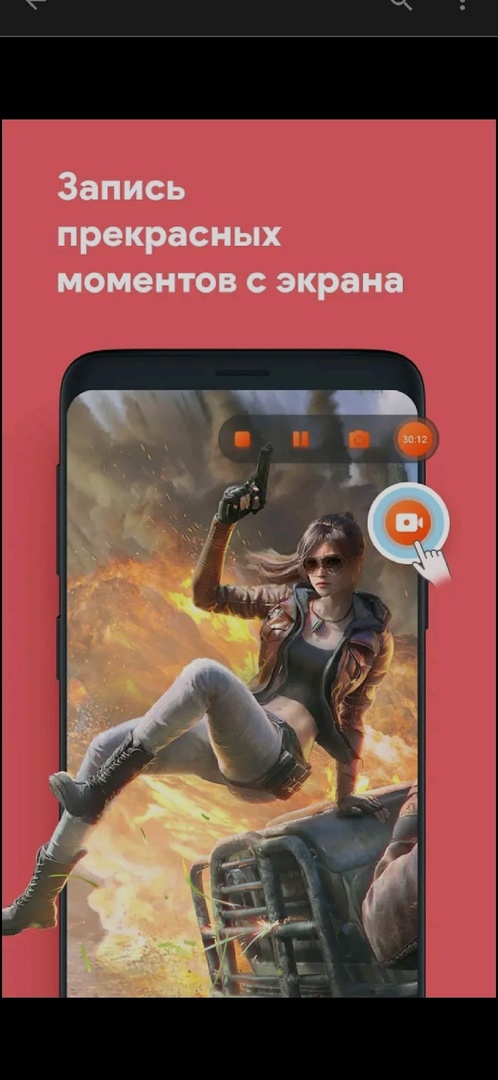


Рисунок 1.4 — Приложение «Запись Экрана – снимок экрана»

### 1.4.5 Запись Экрана — снимать видео

«Записывайте действия на экране, записывайте видео и игры без временных ограничений одним касанием, а затем совершенствуйте результаты с помощью разных инструментов, например кисти, камеры с лицом, быстрых снимков экрана и т.п. Сделайте один щелчок, чтобы поделиться видео с друзьями, родственниками и подписчиками одним нажатием.» Именно так рекомендует себя данное приложение, которое собрало в себе все преимущества предыдущих приложений. Приложение имеет удобный и интуитивно понятный интерфейс, а так же множество полезных функций, помогающих при работе с видео и его записи.

Приложение «Запись Экрана – снимать видео» изображено на рисунке 1.4.

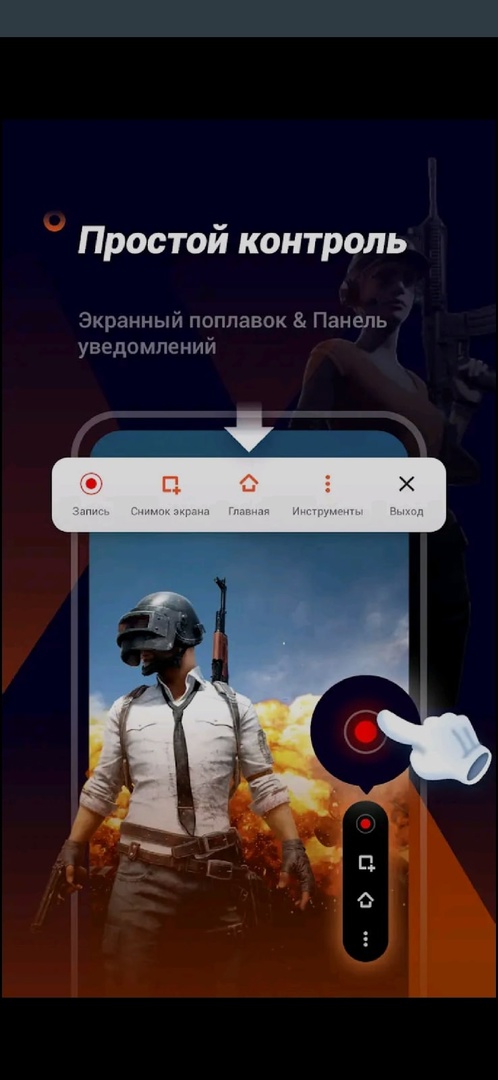


Рисунок 1.4 — Приложение «Запись Экрана – снимать видео»

Проанализировав данные приложения, можно сделать вывод.

Положительными качествами разрабатываемого приложения должны быть:

* оснащенность необходимыми функциями;
* удобный оверлей, помогающий пользователю;
* правильная цветовая гамма, состоящая из трёх основных цветов.

Минусы приложения, которые стоит избегать при разработке:

* большое количество функций, которые присутствуют в редакторе видео;
* страницы для работы с фотографиями.

# Проектирование приложения

## UI/UX дизайн проекта

Дизайн проекта был разработан в программе Figma, где были определены основные экраны:

* Экран приветствия;
* экран оверлея;
* экран информации;
* главный экран.

Цветовая гамма была выбрана посредством изучения информации о популярности цветов. Даже в самой простой однотонной схеме обычно присутствует как минимум два оттенка одного цвета. Правильно подобранные цветовые паттерны стимулируют участие пользователей и узнаваемость бренда.

Так как у пользователей разные вкусовые предпочтения, то принято решение использовать темную и светлую тему в зависимости от системной темы устройства пользователя. Так же принято решение отдать предпочтение спокойному активному оранжевому цвету, который будет хорошо гармонировать, как с черным, так и с белым цветами.

Ниже на Рисунке 2.1 представлен логотип «PL Record». На логотипе изображена тематическая приложению иллюстрация.



Рисунок 2.1 – Логотип «PL Record»

На Рисунке 2.2 ниже представлена страница приветствия. На данной странице мобильного приложения предоставлена возможность увидеть логотип приложения.



Рисунок 2.2 – Страница «Приветствия»

На Рисунке 2.3 ниже представлена главная страница. На данной странице пользователь имеет возможность использовать функцию захвата и записи экрана, а также функция для просмотра ранее записанных видео.

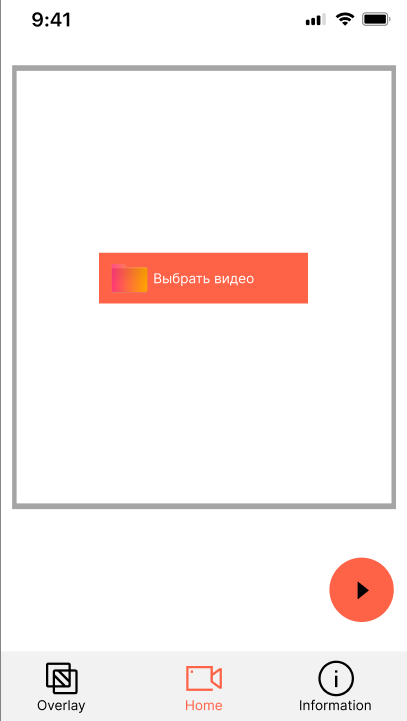


Рисунок 2.3 – Главная страница

На Рисунке 2.4 ниже представлена страница «Информация». На данной странице находится информация о приложении и его разработчике.



Рисунок 2.4 – Страница «Информация»

На Рисунке 2.5 ниже представлена страница «Оверлей». На данной страницы пользователь может воспользоваться функцией «Оверлей», которая позволяет начинать и заканчивать съемку видео при свернутом приложении.

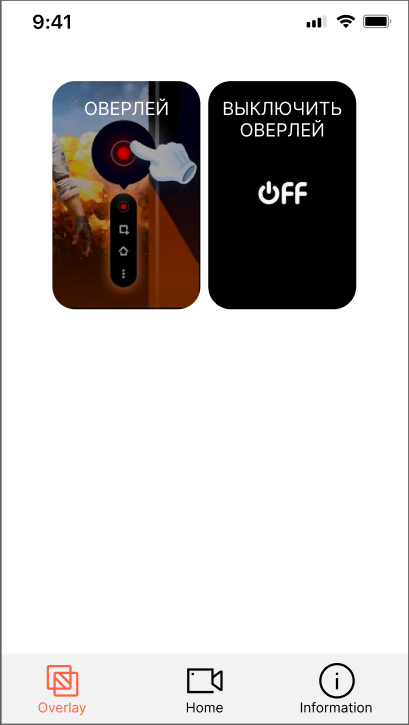


Рисунок 2.5 – Страница «Оверлей»

## Выбор технологии, языка и среды программирования

Средой программирования выбрана программа Android Studio версии: 2021.3.1.16. Языком программирования является Java.

# 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.