Министерство образования Новосибирской области

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

**разработка** **мобильного приложения для ведения здорового образа жизни**

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

МДК.01.03 Разработка мобильных приложений

НАТКиГ.200200.010.000ПЗ

Разработал:

студент группы ПР-20.101

Антух С.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

Мобильное приложение для ведения здорового образа жизни поможет клиенту более точно следить за своими результатами, а так же поможет просматривать свою физическую активность и бить свои же рекорды.

Целью курсового проекта является создание мобильного приложения для ведения здорового образа жизни.

Задачами курсового проекта в связи с указанной целью являются:

* изучение предметной области;
* рассмотрение приложения с точки зрения пользователя для выявления необходимых функций приложения;
* создание интерфейса и необходимого функционала приложения;
* тестирование приложения.

Объект исследования — приложение для ведения здорового образа жизни.

Предмет исследования — изучение принципов функционирования и инструментов приложения.

# Исследовательский раздел

## Описание предметной области

Жизнь человека и ее дальнейшее течение зависит от многих причин, самая первостепенная из которых — состояние здоровья его организма. Существует множество определений и мнений о том, что такое здоровье, одно из которых: нет болезней и физиологических дефектов — значит здоровье на высшем уровне. Согласно Всемирной организации здравоохранения, здоровье — это состояние полного физического, социального, психологического и духовного благополучия человека, соответствующее его возрасту и полу. Оно является основным условием полноценной жизни и главной человеческой потребностью. Наше здоровье определяет способность к труду и преодолению проблем, обеспечивает гармоничное развитие личности и продолжение рода, является предпосылкой к познанию счастья и окружающего нас мира. Здоровье важно не только для отдельного человека, но и для всего общества в целом. Для обеспечения активной и долгой жизни необходимо разумно подходить к сохранению и укреплению своего здоровья. Здоровый образ жизни — это образ жизни человека, который направлен на улучшение и сохранение здоровья, профилактику болезней и укрепление организма. На данный момент 7,1% населения стало избавляться от пагубных привычек и начали ответственно и осознанно подходить к вопросу здоровья и правильного образа жизни. Поэтому, в настоящее время, ведение здорового образа жизни среди разных поколений становится менее популярным, и чтобы хотя бы немного повысить процент людей ведущих здоровый образ жизни, а так же облегчить ведение здорового образа жизни, было принято решение о создании фитнесс приложения. Отличный помощник при решении данных проблем может выступить мобильный телефон. В наше время мобильные телефоны стали самыми необходимыми в жизни людей. Они есть почти у каждого человека. Сейчас это портативные, мощные, удобные и полезные помощники, которые изменились до неузнаваемости. Если раньше телефон был мало функционален и использовался только как средство для связи, то теперь он может использоваться как портативный персональный компьютер или смартфон, который будет всегда под рукой. Существует множество программных решений для ведения здорового образа жизни. У каждого человека, который хочет вести или ведет его, есть различного рода приложения, которые помогают ему это делать — приложения с подсчётом шагов, пройденного расстояния, расчёта соотношении массы тела, счётчиками калорий, шагов и так далее.

К сожалению, приложения, которое поможет начать, ввести и поддерживать здоровый образ жизни, содержащее все самое структурированное, эффективное и полезное под рукой, нет. Если даже и есть, то обязательно чего-то не хватает, либо оно неудобное, либо платное. Чтобы устранить эту проблему, в рамках данной курсовой работы было решено разработать мобильное приложение на платформе Android, которое будет содержать не только всю необходимую информацию, но и будет помогать следить за своим здоровьем и результатами. Целью курсовой работы является: разработка мобильного приложения «Fitness Tracker» для информационно-методической поддержки здорового образа жизни, представляющее собой справочник с данными своих результатов.

## Образ клиента

В первую очередь приложение рассчитано для спортсменов, для которых фиксация своих результатов является очень важной. Первыми кто установят данное приложение, будут именно они, после данное приложение будут устанавливать люди, которые ведут спортивный образ жизни, или люди, которые хотят следить за своим здоровьем.

## Сценарии

Спортсмен Егор увлекается лёгкой атлетикой, часто бегает по всяким маршрутам или гуляет с друзьями, для него фиксирование своих результатов это главный объект исследования, потому что он с каждым разом хочет бить свои рекорды. Поэтому установка приложения будет хорошим решением для фиксации своих результатов и поддержания здорового образа жизни.

Василий страдает излишком веса, он очень хочет похудеть, сходив к доктору ему дали рекомендацию с необходимым количеством съедаемых калорий и их сжигания. Единственным решением для Василия является занятие ходьбой. Поэтому установка приложения поможет Василию подсчитывать сжигаемые калории и рассчитывать соотношения своего веса до приведения его в норму.

Кирилл ведёт активный образ жизни, часто гуляет с друзьями, очень много проводит времени, условно на ногах. Поэтому установка приложения поможет Кириллу просматривать количество шагов и общее расстояние, которое он прошёл за день.

## Сбор и анализ прототипов

У «Fitness Tracker» мобильного приложения конкурентов множество, так как ходьба является основной деятельностью человека.

**Шагомер — счётчик Шагов от Leap Fitness Group** имеет яркое цветовое решение приложения.

Ниже приведен рисунок (Рисунок 1) с изображением главной страницы.



Рисунок 1 – Главная страница

О приложении:

Для подсчета шагов этот шагомер использует встроенный датчик. Никакого GPS, а значит, заряд держится дольше. Приложение отслеживает сожженные калории, пройденное расстояние, время на ногах и т. д. Вся информация представлена на понятных графиках.

Одно нажатие на кнопку старта — и шаги уже считаются. Даже тогда, когда телефон заблокирован и лежит в сумке, кармане или нарукавной повязке.

**Шагомер** — **счетчик шагов от Trusted Android Apps** имеет яркое цветовое решение приложения.

Ниже приведен рисунок (Рисунок 2) с изображением главной страницы.



Рисунок 2 – Главная страница

О приложении:

Оставайтесь активными с шагомером — приложением для отслеживания шагов и счетчика калорий.

Для всех фанатов здоровья, которые ищут идеальное приложение для шагомера. Бесплатный счетчик шагов и счетчик калорий для похудения — ответ на ваш поиск. Лучший шагомер для Android — это просто точный трекер шагов для любого уровня навыков. Счетчик шагов и приложение для счетчика калорий упрощают ведение здорового образа жизни. Бесплатный шагомер для похудения отслеживает ваши шаги, не разряжая аккумулятор мобильного телефона.

**Шагомер** — **счетчик шагов** **от** **Hitchhike Tech** имеет яркое цветовое решение приложения.

Ниже приведен рисунок (Рисунок 3) с изображением главной страницы.

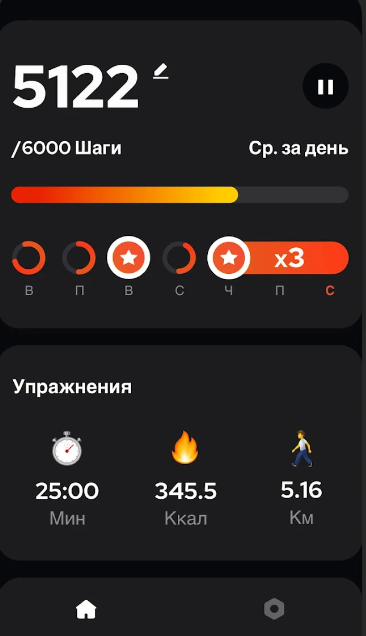


Рисунок 3 – Главная страница

О приложении:

«Счетчик шагов» использует встроенный полнофункциональный трекер физической активности, четко и автоматически отслеживает ваши шаги, сожженные калории, пройденное расстояние и продолжительность ходьбы. GPS-трекинг не используется, что уменьшает расход заряда. Подключение к сети Wi-Fi не требуется, так что вы можете отслеживать ходьбу.

**Шагомер** — **счетчик шагов от ITO Technologies** имеет яркое цветовое решение приложения.

Ниже приведен рисунок (Рисунок 4) с изображением главной страницы.



Рисунок 4 – Главная страница

О приложении:

Шагомер — это совершенно бесплатный счетчик шагов и трекер каллорий: отслеживайте свою ежедневную активность, дистанцию бега, сожженные калории и другие фитнес-показатели!

Шагомер работает в режиме оффлайн, что позволяет вам контролировать результаты своей активности даже без интернета.

**Samsung Health от Samsung Electronics** имеет яркое цветовое решение приложения.

Ниже приведен рисунок (Рисунок 5) с изображением главной страницы.

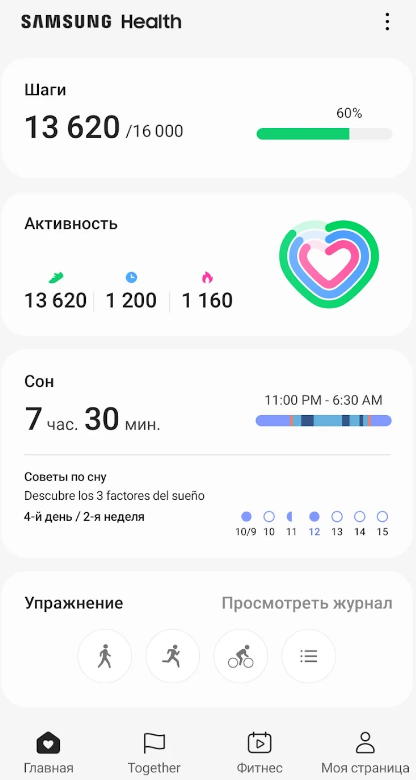


Рисунок 5 – Главная страница

О приложении:

Samsung Health обладает различными функциями, которые помогут вам управлять своим здоровьем. Так как приложение позволяет автоматически регистрировать многие процессы активности, создание здорового образа жизни стало проще и легче, чем когда-либо.

# Проектирование приложения

## UI/UX дизайн проекта

Дизайн проекта разработан в программе Figma.

Для проекта были определены основные экраны:

* главный экран;
* экран с количеством пройденных шагов;
* экран с количеством сожженных калорий;
* экран пройденным расстоянием;

Выбор цветовой гаммы формировался на результатах статистики пользователей. Большинство пользователей выбирают темные оттенки. В первую очередь, производители смартфонов говорят, что темная тема хороша на OLED-экранах, которых в последнее время становится все больше. Это действительно смотрится эффектно. Так же считается, что экран с темной темой не просто хорошо выглядит, но и бережет глаза. Но стоит заметить, что данный цвет, разбавленный изображениями, не выглядят скучно, а весьма лаконично.

Поэтому принято решение в разработке макета отдать предпочтение чёрным цветам.

Ниже на рисунке (Рисунок 6) представлен логотип «Fitness Tracker».



Рисунок 6 – Логотип «Fitness Tracker»

На логотипе изображена тематическая приложению иллюстрация.

На рисунке (Рисунок 7) ниже представлена страница приветствия.

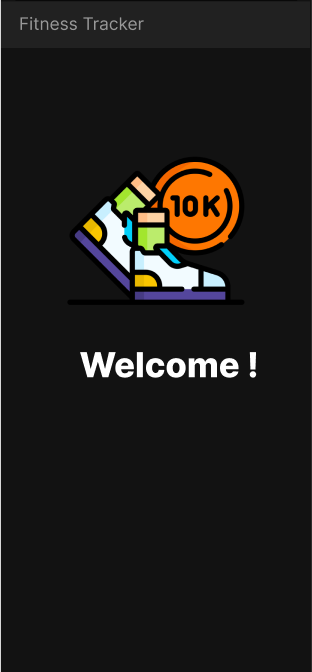


Рисунок 7 – Страница «Приветствия»

На данной странице мобильного приложения предоставлена возможность увидеть логотип приложения.

На рисунке (Рисунок 8) ниже представлена страница с количеством пройденных шагов.

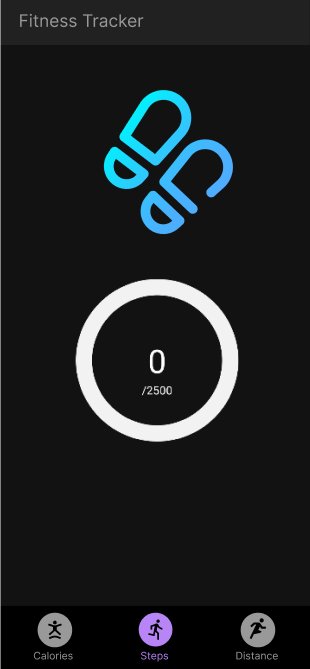


Рисунок 8 – Страница «Количество шагов»

На данной странице пользователь имеет возможность просмотреть, какое количество шагов он сделал. Для этого ему достаточно переключиться в нижнем меню на «Steps»

На рисунке (Рисунок 9) ниже представлена страница c сожжёнными калориями.

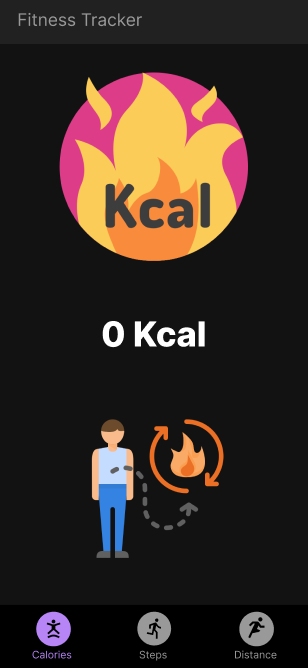


Рисунок 9 – Страница «Сожжённые калории»

На данной странице имеется возможность просмотра общего количества сожжённых калорий пользователем. Для этого ему достаточно переключиться в нижнем меню на «Calories»

На рисунке (Рисунок 10) ниже представлена страница с пройденным расстоянием.

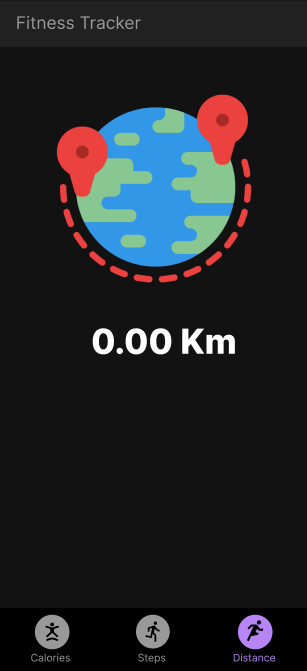


Рисунок 10 – Страница «Пройденное расстояние»

На данной странице представлена возможность просмотра пройденного расстояния. Для этого ему достаточно переключиться в нижнем меню на «Distance»

## Выбор технологии, языка и среды программирования

Средой программирования выбрана программа Android Studio версии: 2021.3.1.16. Языком программирования является Kotlin. Использовать Kotlin проще и удобнее, чем Java. В нем программный код получается в среднем на 40% короче. А еще он позволяет не допускать некоторых ошибок, которые могут возникнуть в ходе выполнения программы. Когда код более простой и понятный, ошибки сложнее сделать и легче обнаружить, поэтому их число стремительно снижается. Тратится меньше времени на разработку и тестирование.

# 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.