Министерство образования Новосибирской области ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

**Разработка мобильного приложения «Облачное хранение фотографий и их создание»**

Пояснительная записка к курсовому проекту

ПМ.01Разработка модулей программного обеспечения для компьютерных систем

МДК 01.03 Разработка мобильных приложений

НАТКиГ.200900.010.000ПЗ

Выполнил: Плахотный Г.А.

Новосибирск

2022

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc117529920)

[1 Исследовательский раздел 5](#_Toc117529921)

[1.1 Описание предметной области 5](#_Toc117529922)

[1.2 Образ клиента 5](#_Toc117529923)

[1.3 Сценарии 5](#_Toc117529924)

[1.4 Сбор и анализ прототипов 6](#_Toc117529925)

[2 Проектирование приложения 8](#_Toc117529926)

[2.1 UI/UX дизайн проекта 8](#_Toc117529927)

[3 Разработка мобильного приложения 18](#_Toc117529928)

[3.1 Разработка базы данных 19](#_Toc117529929)

[3.2 Разработка мультимедийного контента 19](#_Toc117529930)

[3.3 Описание используемых плагинов 30](#_Toc117529931)

[3.4 Описание разработанных процедур и функций 31](#_Toc117529932)

[4 Тестирование 37](#_Toc117529933)

[4.1 Протокол тестирования дизайна приложения 37](#_Toc117529934)

[4.2 Протокол тестирования функционала приложения 37](#_Toc117529935)

[ЗАКЛЮЧЕНИЕ 40](#_Toc117529936)

[СПИСОК ИСТОЧНИКОВ 41](#_Toc117529937)

## ВВЕДЕНИЕ

Мобильное приложение для облачного хранения изображений решает одну из главных проблем современных мобильных устройств, а именно — нехватку памяти. Практически у каждой компании, интернет-магазина, видеохостинга существует собственное мобильное приложение, количество которых постоянно увеличивается. Таковые приложения, а также их системные данные занимают драгоценную память мобильного устройства.

В результате, мобильное устройство, память которого переполнена различными приложениями, теряет одну из своих основных функций — создание фотографий. Данная функция позволяет пользователю запечатлеть важные моменты из своей жизни, сохранять визуальную информацию, заниматься творчеством.

Конечно, одним из решений проблемы нехватки памяти является постоянное совершенствование технического прогресса и увеличение объемов памяти, но рядовой пользователь не всегда поспевает за новыми моделями мобильных устройств и ему необходимо предоставить приложение, позволяющее оптимизировать хранилище фотографий, позволив, тем самым, продолжать создавать изображения.

Целью курсового проекта является создание мобильного приложения для облачного хранения фотографий, позволяющее пользователю загружать фотографии в облако из галереи мобильного устройства, просматривать данные фотографии на сервере.

Задачами курсового проекта в связи с указанной целью являются:

* изучение предметной области темы;
* рассмотрение приложения с точки зрения пользователя для нахождения необходимых функций приложения;
* выбор инструментов, наиболее подходящих для разработки данного типа приложения;
* выбор инструментов, наиболее подходящих для разработки серверной части;
* создание дизайн-проекта приложения;
* разработка приложения.

Объект исследования – приложения облачного хранения изображений.

Предмет исследования – изучение принципов работы и инструментов приложения.

# Исследовательский раздел

## 1.1 Описание предметной области

Зачастую, большинство людей используют свои мобильные устройства в качестве фотоаппарата, большое количество фотографий занимает место на мобильном устройстве пользователя, тем самым ограничивая возможность установки каких-либо дополнительных приложений. Также бывает и ситуация – множество необходимых приложений занимают практически все пространство памяти мобильного устройства и не позволяют пользователю создавать фотографии.

Приложение с возможностью сохранения фотографий и отправке их в облачное хранилище позволит пользователю экономить место на своем мобильном устройстве. Также данное мобильное приложение позволяет не переживать о потере каких-либо важных для клиента фотографий при утере устройства, ведь все фотографии будут сохранены в облачном хранилище.

Также приложение позволяет просматривать фотографии, находящиеся в облачном хранилище, не скачивая их в память устройства.

## 1.2 Образ клиента

Данное приложение рассчитано на любого клиента, с возрастом старше 6 лет, ведь фотографии может делать любой. Мобильное устройство на базе Android. Приложение поддерживает русский язык.

## 1.3 Сценарии

### 1.3.1 Первый сценарий

Геннадий — успешный бизнесмен, на телефоне которого находится большое количество различных приложений банковской сферы, у Геннадия родился сын, но из-за того, что на телефоне Геннадия все место занято различными банковскими приложениями, он не может сделать фотографии в данный момент. Установка приложения для хранения фотографий позволит Геннадию запечатлеть все важные моменты его жизни.

### 1.3.2 Второй сценарий

Валентин, гуляя по городу решил сфотографировать красивые архитектурные постройки, чтобы не заполнять память своего мобильного устройства, Валентин сделал фотографию и сразу загрузил его в облако, тем самым не засоряя хранилище своего телефона.

## 1.4 Сбор и анализ прототипов

### 1.4.1 Первый прототип

Название приложения: Яндекс Диск — облачное хранилище.

Описание: Яндекс Диск — это удобное защищенное облако для хранения всех ваших файлов. Безлимитное хранилище для фото с телефона.

Приложение позволяет сохранять фотографии. Есть фильтрация фотографий по различным критериям.

Приложение «Яндекс Диск» изображено на рисунке 1.1.

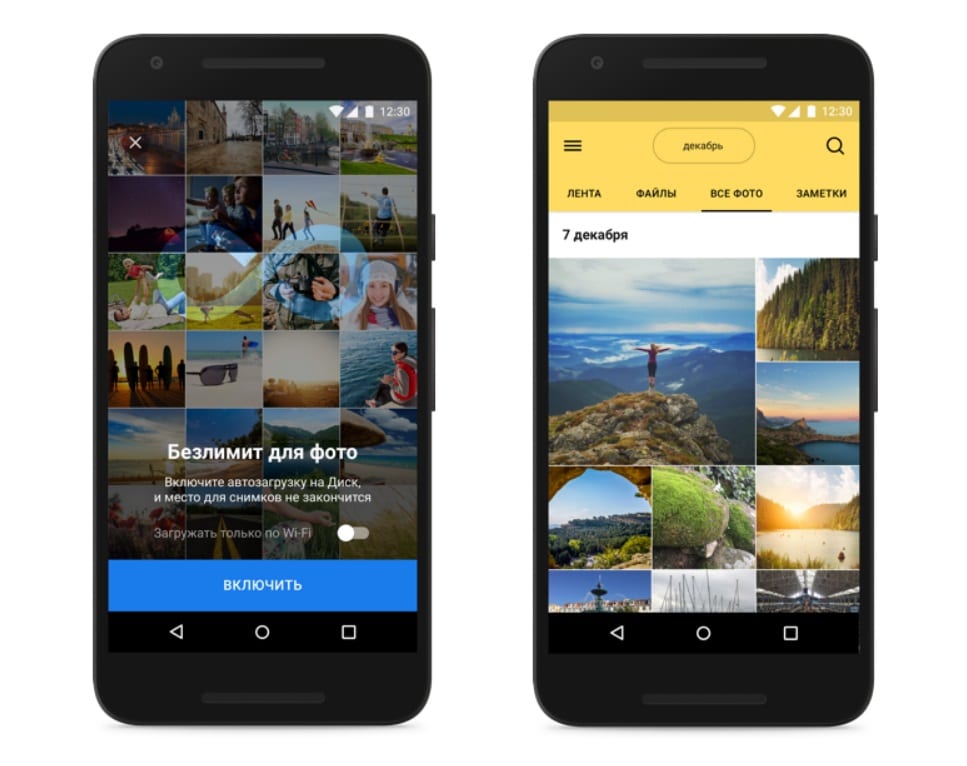


Рисунок 1.1 — Приложение «Яндекс Диск»

### 1.4.2 Второй прототип

Название приложения: Google Фото.

Описание: это удобный сервис, который позволяет хранить и автоматически упорядочивать фотографии, а также делиться ими.

Приложение «Google Фото» изображено на рисунке 1.2.

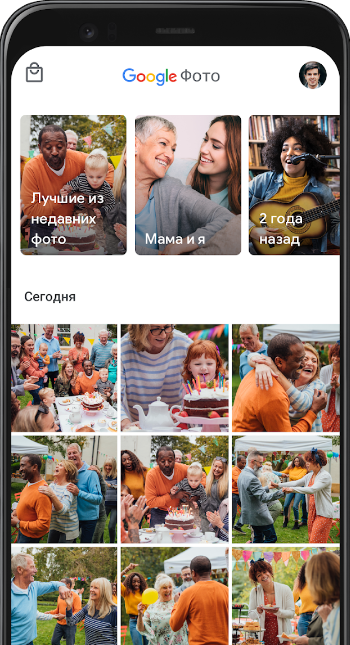


Рисунок 1.2 — Приложение «Google Фото»

# Проектирование приложения

## 2.1 UI/UX дизайн проекта

Выбор приложения для разработки дизайн-проекта будущего приложения пал на веб-сервис Figma. Выбор обоснован тем, что Figma является веб-сервисом и работает из браузера, также очевидным плюсом является приятный и понятный дизайн интерфейса, а также наличием плагинов, облегчающих перенос разметки из дизайн-проекта в будущее приложение.

Для проекта были определены основные экраны:

* экран регистрации;
* экран авторизации;
* экран восстановления пароля (заполнения почты для восстановления);
* экран ввода полученного кода для восстановления;
* экран изменения пароля;
* главный экран;
* экран стартовых настроек
* экран настроек;
* экран с избранными изображениями;
* экран с начальными настройками.

Предполагалось создать минималистичный дизайн мобильного приложения, чтобы не отвлекать внимания пользователя на фотографии различными элементами управления. Поэтому в приложении используются следующие цвета: белый, черный, серый, фиолетовый. Стоит обратить внимание на то, что используется мягкий оттенок фиолетового цвета, который вполне приятно сочетается с другими элементами приложения.

Также в приложении используется градиент двух цветов: мягко-голубого и фиолетового, который практически не бросается в глаза, а заметен лишь при намеренном просмотре элементов управления. В некоторых элементах управления используется основной цвет и его полутон: фиолетовый и светло-фиолетовый.

На рисунке 2.1.1 представлена страница авторизации.

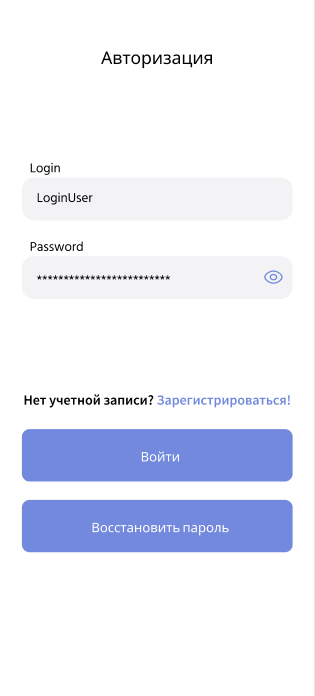


Рисунок 2.1.1 — экран авторизации

На данной странице мобильного приложения пользователю предоставлена возможность ввести данные для авторизации в мобильном приложении. При отсутствии учетной записи, пользователю предоставлена возможность перейти на страницу регистрации. Также, присутствует кнопка, предназначенная для перехода на страницу восстановления пароля.

На рисунке 2.1.2 представлена страница регистрации в приложении.

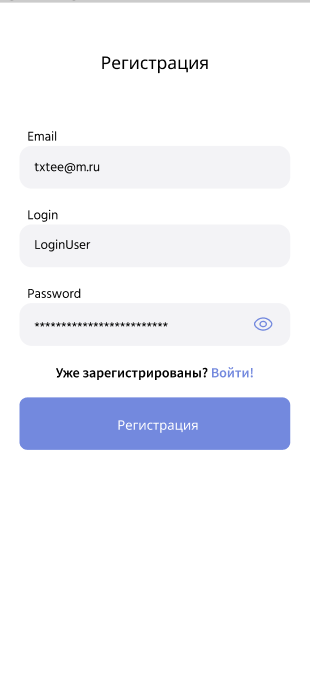


Рисунок 2.1.2 — экран регистрации

На данной странице мобильного приложения пользователю предоставлена возможность зарегистрироваться. Также, при наличии у пользователя аккаунта, ему предоставлена возможность перейти на страницу авторизации.

На рисунке 2.3 представлены страницы восстановления пароля.

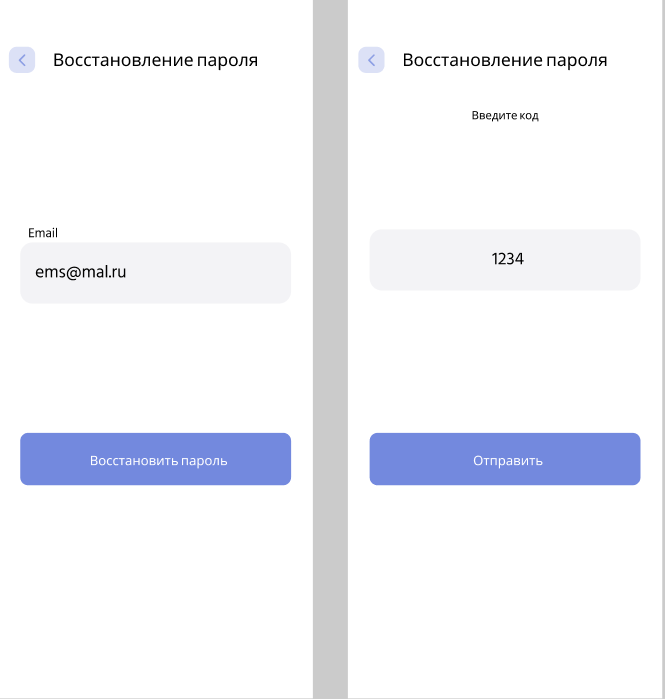


Рисунок 2.1.3 — страницы восстановления пароля

На первой странице пользователю предлагается ввести почту, необходимую для восстановления. На второй странице пользователь должен ввести код, присланный на почту. После, пользователь переходит на страницу, содержащую поле для ввода нового пароля.

На рисунке 2.1.4 представлена страница для ввода нового пароля.

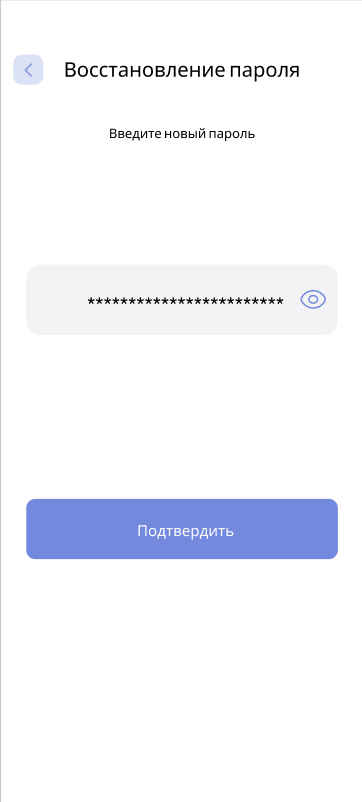


Рисунок 2.1.4 — экран восстановления пароля

Пользователь, после авторизации или регистрации, переходит на страницу «знакомства», где вводит основную информацию о себе.

Данная страница представлена на рисунке 2.1.5.

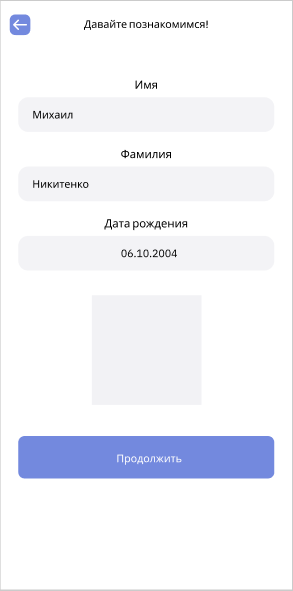


Рисунок 2.1.5 — экран знакомства с пользователем

На данной странице пользователь вводит основные данные о себе: фамилию, имя, дату рождения. А также добавляет фотографию своего профиля, которая носит исключительно эстетический характер и видна только пользователю.

После пользователь попадает на страницу со своими фотографиями, которая является главной страницей приложения.

На рисунке 2.1.6 представлена страница с фотографиями.

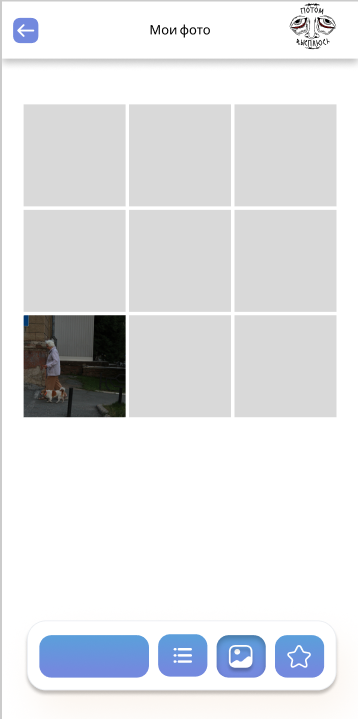


Рисунок 2.1.6 — экран с фотографиями пользователя

На данном экране расположены фотографии, добавленные пользователем в облако. Также снизу страницы расположена навигационная панель, позволяющая перемещаться между разделами: «сделать фото», «настройки», «мои фото», «избранное».

Экран с настройками изображен на рисунке 2.1.7

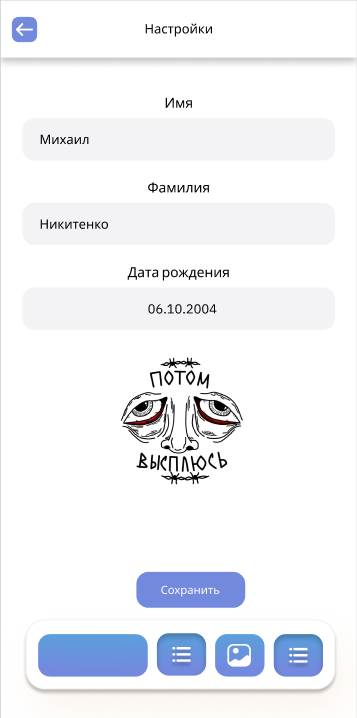


Рисунок 2.1.7 — экран настроек профиля

На данном экране расположены основные поля, предназначенные для настроек информации о профиле пользователя. А также предоставлена возможность сменить фотографию профиля.

На рисунке 2.1.8 изображен экран с избранными изображениями.

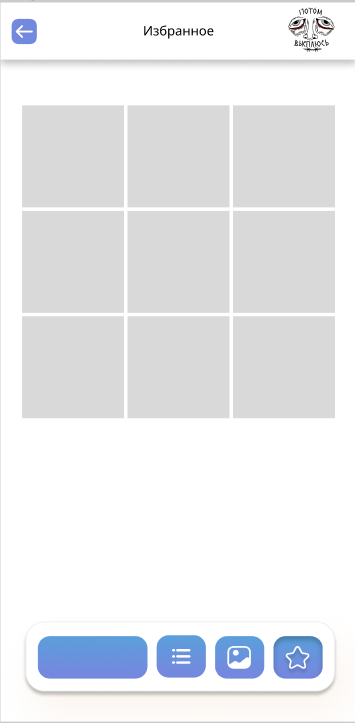


Рисунок 2.1.8 — экран с избранными изображениями

Экран идентичен экрану со всеми изображениями, но содержит только те изображения, которые избрал пользователь. При нажатии на какое-либо изображение, пользователю открывается окно с информацией о данном изображении.

На рисунке 2.1.9 изображен экран с информацией о фотографии.



Рисунок 2.1.9 — экран с информацией о фотографии

На данном экране пользователь может подробнее рассмотреть выбранную фотографию, а также узнать дату создания изображения.

## 2.2 Выбор технологии, языка и среды программирования

Средой программирования была выбрана программа Android Studio, причиной выбора данной среды разработки является то, что данная среда разработки мобильных приложений является основной в нашем колледже.

Среда разработки поддерживает два языка программирования: Java и более современный Kotlin. В качестве языка разработки была выбран язык программирования Java, этот выбор обусловлен тем, что язык программирования Java, в определенных аспектах, имеет большую схожесть с языком программирования C#, который является основным языком программирования в нашем колледже.

Используемый сервис для базы данных – phpMyAdmin.

phpMyAdmin представляет собой веб-интерфейс для администрирования СУБД MySQL. Данный веб-интерфейс был выбран по причине использования его в курсовой работе на предыдущем курсе. Разработка серверной части велась на языке программирования PHP.

Веб-интерфейс phpMyAdmin располагается на бесплатном хостинге от beget.tech. Взаимодействие с веб-хостингом в курсовой работе позволяет намного лучше приблизиться к процессу “настоящей” разработки мобильного приложения.

Для разработки иконки мобильного приложения использовался Adobe Photoshop CC 2019.

# Разработка мобильного приложения

## 3.1 Разработка базы данных

В качестве веб-интерфейса для администрирования был выбран phpMyAdmin, расположенный на веб-хостинге beget.tech.

Для взаимодействия с мобильным приложением необходимо было добавить две сущности:

* сущность users;
* сущность recoveryPassword;
* сущность photos.

Сущность users содержит всю необходимую информацию о пользователе мобильного приложения: Имя, Фамилию, адрес электронной почты, логин, пароль, дату рождения, а также фотографию профиля пользователя. Данная информация заполняется пользователем, сразу же после регистрации, в стартовом окне настроек профиля. Позже, пользователь также может изменить информацию профиля в окне настроек.

Структура сущности users представлена на рисунке 3.1.1.



Рисунок 3.1.1 — структура сущности users

При заполнении пользователем данных в окне регистрации, происходит создание записи в сущности на сервере. Код, отвечающий за получение данных из запроса и создание записи представлен на листинге 1.

## <?php

## //коды ошибок

## //004 - аккаунт с таким email-адресом уже зарегистрирован.

## require\_once 'DBConnection/connect.php';

## if($\_SESSION['connect']){

## $firstname = mysqli\_real\_escape\_string($connect,$\_GET['firstname']);

## $lastname = mysqli\_real\_escape\_string($connect,$\_GET['lastname']);

## $bday = mysqli\_real\_escape\_string($connect,$\_GET['bday']);

## $mainPhoto = mysqli\_real\_escape\_string($connect,$\_GET['mainPhoto']);

## $password = mysqli\_real\_escape\_string($connect,$\_GET['password']);

## $login = mysqli\_real\_escape\_string($connect,$\_GET['login']);

## $email = mysqli\_real\_escape\_string($connect,$\_GET['email']);

## $users = mysqli\_query($connect, "select id from users where email = '$email'");

## $users = mysqli\_fetch\_all($users);

## foreach($users as $users){}

## if(count($users) > 0){

## echo '004';

## }

## else{

## mysqli\_query($connect, "INSERT INTO `users`(`firstName`, `lastName`, `bday`, `mainPhoto`, `password`, `login`, `email`) VALUES ('$firstname', '$lastname', '$bday', '$mainPhoto', '$password', '$login', '$email')");

## }

## }else{

## echo '004';

## }

## ?>

Листинг 1 — получение данных из запроса и добавление записи

Для безопасности работы серверной части приложения запросы были экранированы, чтобы избежать случайного или намеренного вызова команд через запрос.

При авторизации пользователя вызывается класс, сверяющий данные, полученные из запроса с данными, хранящимися на сервере. Отвечающий за это код представлен в листинге 2.

<?php

require\_once 'DBConnection/connect.php';

if($\_SESSION['connect']){

$$login = mysql\_real\_escape\_string($connect,$\_GET['login']);

$$password = mysql\_real\_escape\_string($connect,$\_GET['password']);

$user = mysqli\_query($connect, "SELECT \* FROM users WHERE login = '$login' and password = '$password'");

$rows = array();

while ($row = mysql\_fetch\_assoc($user)) {

$rows[] = $row;

}

if(count($rows) > 0){

echo json\_encode($rows);

}

else{

echo '004';

}

}

?>

Листинг 2 — код для проверки данных при авторизации

Сущность recoveryPassword содержит всю необходимую информацию для работы системы восстановления пароля: id пользователя, дату и ip-адресс пользователя. Структура сущности recoveryPassword представлена на рисунке 3.1.2.

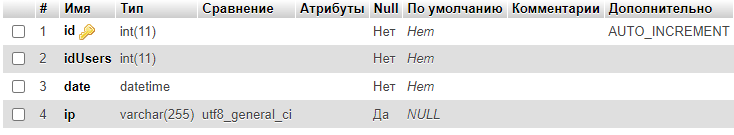


Рисунок 3.1.2 — структура сущности recoveryPassword

При запросе восстановления пароля пользователем, при помощи запроса на сервер, отправляются данные, введенные пользователем, где они проверяются на схожесть с теми, которые были заполнены в сущности users, а именно электронной почты. Если таковая запись существует, то генерируется код проверки, а также письмо, которое отправляется пользователю на указанный адрес электронной почты. Получив данное письмо, пользователь вводит код в окне восстановления пароля, если введенный пользователем код и код на сервере совпадают, то пользователю предоставляется возможность изменить пароль.

Код восстановления пароля представлен в листинге 3.

<?php

require\_once 'DBConnection/connect.php';

if($\_SESSION['connect']){

$email = mysqli\_real\_escape\_string($connect,$\_GET['to']);

$users = mysqli\_query($connect, "select id from users where email = '$email'");

$users = mysqli\_fetch\_all($users);

foreach($users as $users){}

if(count($users) > 0){

$subject = "Код восстановления пароля в ";

$app = mysqli\_real\_escape\_string($connect,$\_GET['app']);

$code = mysqli\_real\_escape\_string($connect,$\_GET['code']);

$subject .= $app;

$helpemail = mysqli\_real\_escape\_string($connect,$\_GET['helpemail']);

$from = "smtp.beget.com";

$headers = "From:" . $from;

$headers .= "MIME-Version: 1.0\r\n";

$headers .= "Content-Type: text/html; charset=UTF-8\r\n";

$message ='

<div bgcolor="#ffffff" background="#m\_-6545216809324096674\_ffffff" style="padding:0px;margin:0px;background:#ffffff;background-color:#ffffff">

<table width="550" cellspacing="0" cellpadding="0" border="0" align="center" style="width:550px;text-align:center;padding:0 25px">

<tbody><tr>

<td align="center" valign="top" style="text-align:center;vertical-align:top">

<table height="50" width="550" cellspacing="0" cellpadding="0" border="0" style="max-width:550px;width:100%;height:50px">

<tbody><tr>

<td align="left" valign="top" style="text-align:center;vertical-align:top">

<p style="font-size:40px; color: black;">'.$app.'</p>

</td>

</tr>

</tbody></table>

</td>

</tr>

<tr>

<td align="center" valign="center" style="text-align:center;vertical-align:center">

<table width="550" cellspacing="0" cellpadding="0" border="0" background="#m\_-6545216809324096674\_F5F5F7" bgcolor="#F5F5F7" style="background:#f5f5f7;background-color:#f5f5f7;max-width:550px;width:100%;border-radius:20px">

<tbody><tr>

<td colspan="2" align="left" valign="top" style="padding:30px 25px 20px;text-align:left;vertical-align:top">

<font face="sans-serif, Tahoma, Arial, Helvetica" size="2" color="#000000" style="color:#000000;font-size:24px;line-height:30px">

<strong>

Здравствуйте!</strong>

</font>

</td>

</tr>

<tr>

<td align="left" valign="top" style="padding:18px 25px;text-align:left;vertical-align:top">

<font face="sans-serif, Tahoma, Arial, Helvetica" size="2" color="#333333" style="color:#333333;font-size:16px;line-height:26px">

Вы или кто-то другой запросил сброс пароля для Вашего аккаунта.

</font>

</td>

</tr>

<tr>

<td align="left" valign="top" style="padding:5px 25px;text-align:left;vertical-align:top">

<font face="sans-serif, Tahoma, Arial, Helvetica" size="2" color="#333333" style="color:#333333;font-size:16px;line-height:26px">

Для смены пароля, вам потребуется перейти обратно в приложение и ввести код:

</font>

</td>

</tr>

<tr>

<td align="left" valign="top" style="padding:8px 25px 28px;text-align:center;vertical-align:top">

<p style="display:inline-block;text-decoration:none;padding:15px 25px;background-color:#ffda54;background:#ffda54;border-radius:12px" target="\_blank">

<font face="sans-serif, Tahoma, Arial, Helvetica" size="2" color="#0D1531" style="color:#0d1531;font-size:16px;line-height:26px">

'.$code.'

</font>

</p>

</td>

</tr> </tbody></table>

</td>

</tr>

<tr>

<td align="center" valign="bottom" style="text-align:center;vertical-align:bottom">

<table width="550" cellspacing="0" cellpadding="0" border="0" style="max-width:550px;width:100%;border-radius:20px">

<tbody><tr>

<td align="left" valign="top" style="text-align:center;vertical-align:top">

<p style="font-size:40px; color: black;">'.$app.'</p>

</td>

</tr>

<tr>

<td align="center" valign="top" style="padding:0 25px;text-align:center;vertical-align:top">

<font face="sans-serif, Tahoma, Arial, Helvetica" size="2" color="#333333" style="color:#333333;font-size:14px;line-height:24px">

Работаем круглосуточно

</font>

</td>

</tr>

<tr>

<td align="center" valign="top" style="padding:0 25px;text-align:center;vertical-align:top">

<a href="mailto:'.$helpemail.'" style="text-decoration:none" target="\_blank">

<font face="sans-serif, Tahoma, Arial,

Helvetica" size="2" color="#333333" style="color:#333333;font-size:14px;line-height:24px">

<span class="il">'.$helpemail.'</span> </font>

</a>

</td>

</tr>

<tr>

<td align="center" valign="top" style="padding:15px 75px 10px;text-align:center;vertical-align:top">

<font face="sans-serif, Tahoma, Arial, Helvetica" size="2" color="#666666" style="color:#666666;font-size:12px;line-height:15px">

Это автоматическое сообщение — не отвечайте на него. Если&nbsp;у&nbsp;вас&nbsp;возникли вопросы, пишите нам.

</font>

</td>

</tr>

</tbody></table>

</td>

</tr>

</tbody></table><div class="yj6qo"></div><div class="adL">

</div></div>';

mail($email,$subject,$message, $headers);

$id = $users[0];

$date = date("Y-m-d H:i:s");

if (!empty($\_SERVER['HTTP\_CLIENT\_IP'])) {

$ip = $\_SERVER['HTTP\_CLIENT\_IP'];

} elseif (!empty($\_SERVER['HTTP\_X\_FORWARDED\_FOR'])) {

$ip = $\_SERVER['HTTP\_X\_FORWARDED\_FOR'];

} else {

$ip = $\_SERVER['REMOTE\_ADDR'];

}

mysqli\_query($connect, "INSERT INTO `recoveryPassword`(`idUsers`, `date`, `ip`) VALUES($id, '$date', '$ip')");

}

else{

echo '005';

}

}

?>

Листинг 3 — код для восстановления пароля и генерации письма

## 3.2 Разработка мультимедийного контента

Дизайн-проект приложения разрабатывался в веб-сервисе Figma.

Рабочий интерфейс веб-сервиса представлен на рисунке 3.2.1.

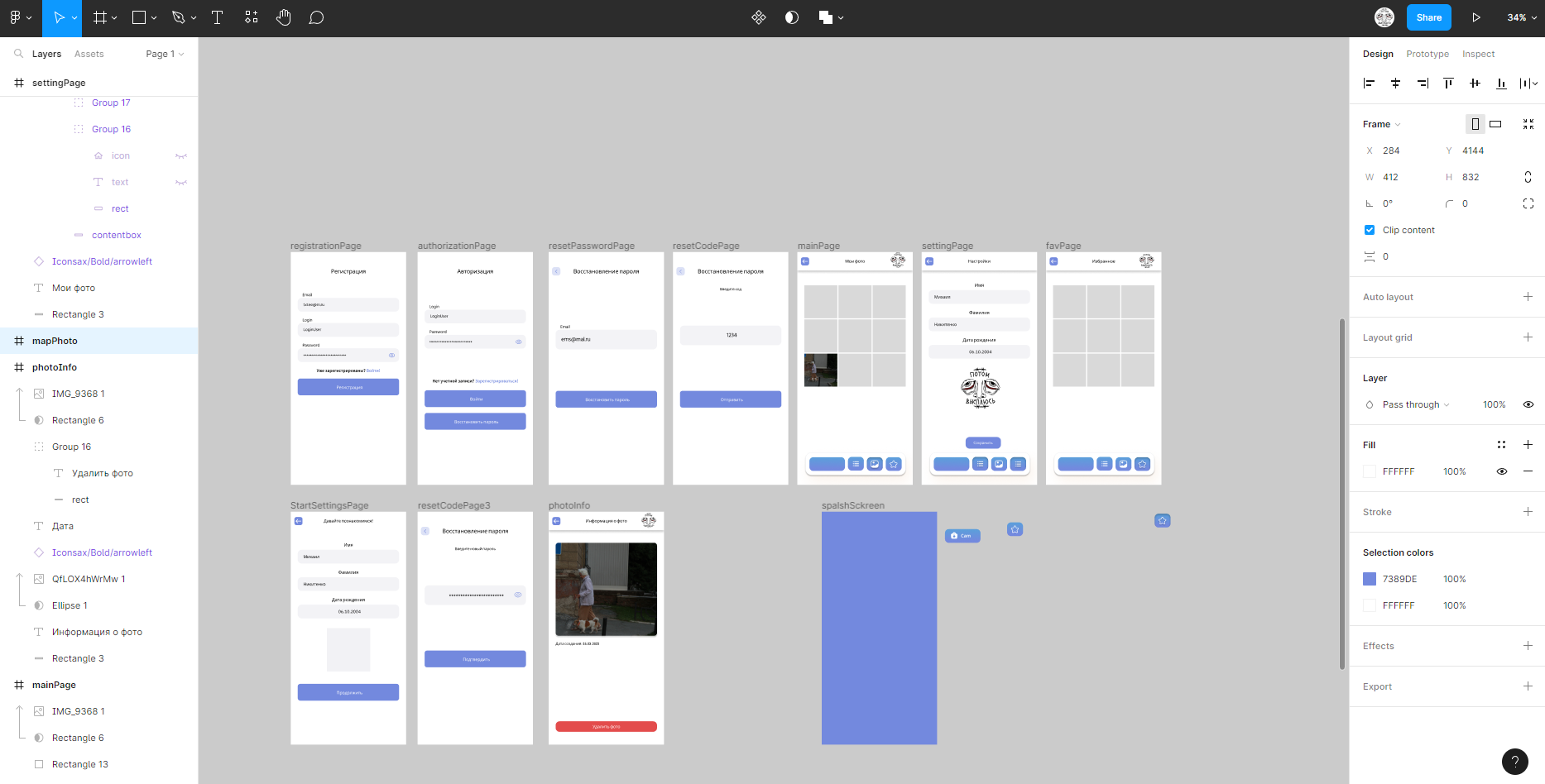


Рисунок 3.2.1 — рабочий интерфейс Figma

Размер окна приложения в дизайн-проекте Figma совпадал с размером окон приложения в Android Studio, что позволило использовать специальный плагин, помогающий перенести разметку из Figma в layout-разметку окна в Android Studio. Разработка дизайна в Android Studio велась при помощи использования относительной разметки — RelativeLayout, что позволяло свободно размещать элементы по заданным XOY координатной плоскости. В листинге 4 представлена разметка окна авторизации.

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

>

<TextView

android:fontFamily="@font/nunito\_medium"

android:layout\_gravity="center\_horizontal|center"

android:gravity="center"

android:textSize="20sp"

android:textColor="@color/black"

android:layout\_marginLeft="0dp"

android:layout\_marginTop="38dp"

android:layout\_width="427dp"

android:layout\_height="77dp"

android:layout\_centerHorizontal="true"

android:text="@string/authorization"

/>

<TextView android:id="@+id/login"

android:fontFamily="@font/nunito\_medium"

android:textSize="15sp"

android:textColor="@color/black"

android:layout\_marginLeft="42dp"

android:layout\_marginTop="186dp"

android:layout\_width="178.15dp"

android:layout\_height="30.5dp"

android:text="@string/login\_txt"

/>

<TextView android:id="@+id/password"

android:fontFamily="@font/nunito\_medium"

android:textSize="15sp"

android:textColor="@color/black"

android:layout\_marginLeft="42dp"

android:layout\_marginTop="276dp"

android:layout\_width="155.5dp"

android:layout\_height="30.5dp"

android:text="@string/password\_txt"

/>

<EditText

android:id="@+id/tb\_login"

android:background="@drawable/input\_field"

android:layout\_marginStart="25dp"

android:layout\_marginTop="206dp"

android:layout\_centerHorizontal="true"

android:layout\_width="362dp"

android:layout\_height="49dp"

android:paddingStart="15dp"

/>

<EditText

android:id="@+id/tb\_password"

android:background="@drawable/input\_field"

android:layout\_marginStart="25dp"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="296dp"

android:layout\_width="362dp"

android:layout\_height="49dp"

android:paddingStart="15dp"

/>

<TextView android:id="@+id/have\_acc"

android:fontFamily="@font/nunito\_medium"

android:textStyle="bold"

android:layout\_gravity="center\_horizontal|center"

android:gravity="center"

android:textSize="16sp"

android:textColor="@color/button\_color\_1"

android:layout\_marginLeft="22dp"

android:layout\_marginTop="450dp"

android:layout\_width="375dp"

android:layout\_height="31dp"

android:text="@string/not\_account"

android:layout\_centerHorizontal="true"

android:onClick="toRegistrationPage"

/>

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_enter"

android:background="@drawable/btn\_register"

android:layout\_marginStart="25dp"

android:layout\_marginTop="494dp"

android:layout\_width="362dp"

android:layout\_height="60dp"

android:text="@string/enter\_txt\_btn"

android:layout\_centerHorizontal="true"

android:textColor="@color/white"

android:textSize="14sp"

android:onClick="authorization"

/>

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/recover\_passw"

android:background="@drawable/btn\_register"

android:layout\_marginStart="25dp"

android:layout\_marginTop="575dp"

android:layout\_width="362dp"

android:layout\_height="60dp"

android:layout\_centerHorizontal="true"

android:text="@string/recover\_password"

android:textColor="@color/white"

android:textSize="14sp"

android:onClick="toResetPasswordPage"

/>

</RelativeLayout>

Листинг 4 — xml-разметка страницы авторизации

Также для мобильного приложения, в Adobe Photoshop CC 2019, была создана иконка, ее изображение представлено на рисунке 3.2.2.



Рисунок 3.2.2 — иконка мобильного приложения

## 3.3 Описание используемых плагинов

На рисунке 3.3.1 изображен эмулятор для запуска и тестирования приложения в Android Studio.

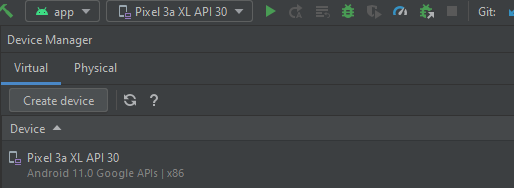


Рисунок 3.3.1 — эмулятор Android Studio

На рисунке 3.3.2 изображены используемые в проекте плагины, большая часть из них — плагины, установленные по умолчанию.

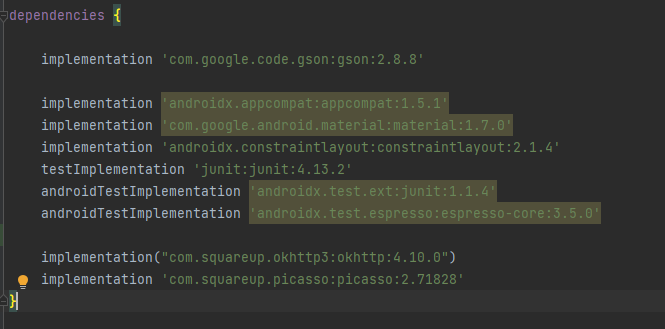


Рисунок 3.3.2 — эмулятор Android Studio

Добавленными являются две нижние библиотеки — okhttp3 и picasso.

Библиотека okhttp3 отвечает за создание http запросов с данными на сервер, а Picasso, в свою очередь, отвечает за работу с изображениями.

## 3.4 Описание разработанных процедур и функций

В приложении разработаны следующие функции:

* регистрация и авторизация;
* запрос восстановления пароля;
* восстановление пароля;
* изменение информации профиля;
* переход между страницами с помощью навигационного меню;
* добавление фотографий;
* просмотр фотографий.

Листинг кода, отвечающего за регистрацию пользователя в системе представлен в листинге 5.

public void registerAcc(View v) {

EditText emailEditText = (EditText) findViewById(R.id.edit\_text\_email);

String email = emailEditText.getText().toString();

EditText nameEditText = (EditText) findViewById(R.id.edit\_text\_login);

String login = nameEditText.getText().toString();

EditText passwordEditText = (EditText) findViewById(R.id.edit\_text\_password);

String password = passwordEditText.getText().toString();

if (isValidEmail(email)) {

OkHttpClient client = new OkHttpClient();

String url = "https://smtpservers.ru/projects/praktikaMobile/registerUsers?password=" + password + "&login=" + login + "&email=" + email;

Request request = new Request.Builder()

.url(url)

.build();

Call call = client.newCall(request);

call.enqueue(new Callback() {

@Override

public void onResponse(Call call, Response response) throws IOException {

String responseBody = response.body().string();

if(responseBody.equals("004")){

Handler handler = new Handler(Looper.getMainLooper());

handler.post(new Runnable() {

@Override

public void run() {

Toast.makeText(getApplicationContext(), "Аккаунт с таким email-адрессом уже зарегистрирован!", Toast.LENGTH\_SHORT).show();

}

});

}

else {

startSettingsPage.Login = login;

toStartSettingsPage(v);

}

}

@Override

public void onFailure(Call call, IOException e) {

String error = e.toString();

}

});

}

}

Листинг 5 — код, отвечающий за регистрацию

Во время регистрации происходит проверка почты пользователя на правильность введения при помощи использования регулярного выражения. Код метода проверки представлен на листинге 6.

public static boolean isValidEmail(String email) {

// Проверяем на null

if (email == null) {

return false;

}

// Задаем паттерн регулярного выражения для email

Pattern pattern = Pattern.compile("^[A-Za-z0-9+\_.-]+@[A-Za-z0-9.-]+$");

// Сравниваем строку с паттерном

Matcher matcher = pattern.matcher(email);

return matcher.matches();

}

Листинг 6 — код, отвечающий за проверку электронной почты

Проверка же на уже зарегистрированный адрес электронной почты происходит на сервере и, если таковой адрес уже зарегистрирован, то сервер отправляет код ошибки 004, а программа, в свою очередь, выводит сообщение о том, что аккаунт с таким email-адресом уже зарегистрирован.

Ниже на листинге 7 кода представлен метод перехода на другие активности.

public void to Start Settings Page(View v)

{

Intent i = new Intent(getApplicationContext(),startSettingsPage.class);

startActivity(i);

}

public void to Authorization Page(View v){

Intent i = new Intent(getApplicationContext(), authorizationPage.class);

startActivity(i);

}

Листинг 7 — код, отвечающий за переход на другие активности

Ниже на листинге 8 кода представлен метод отправки сообщения при восстановлении пароля.

protected void SendMessage(String email, View v){

OkHttpClient client = new OkHttpClient();

Random random = new Random();

int randomNumber = random.nextInt(8999) + 1000;

finalRandomNumber = randomNumber;

String url = "https://smtpservers.ru/projects/praktikaMobile/sendMessage?to="+email+"&code="+randomNumber+"&app=photochka&helpemail=help@smtpservers.ru";

Request request = new Request.Builder()

.url(url)

.build();

Call call = client.newCall(request);

call.enqueue(new Callback() {

@Override

public void onResponse(Call call, Response response) throws IOException {

String responseBody = response.body().string();

if(responseBody.equals("005")){

Handler handler = new Handler(Looper.getMainLooper());

handler.post(new Runnable() {

@Override

public void run() {

Toast.makeText(getApplicationContext(), "Аккаунт с таким email не зарегистрирован!", Toast.LENGTH\_SHORT).show();

}

});

}

else{

Handler handler = new Handler(Looper.getMainLooper());

handler.post(new Runnable() {

@Override

public void run() {

SQLiteDatabase db = getBaseContext().openOrCreateDatabase("app.db",MODE\_PRIVATE,null);

db.execSQL("CREATE TABLE IF NOT EXISTS RecoveryPassword (id INTEGER PRIMARY KEY AUTOINCREMENT, email TEXT, code INTEGER)");

String exception = String.format("INSERT into RecoveryPassword(email, code) VALUES('%s', %s)", email, finalRandomNumber);

db.execSQL(exception);

db.close();

Toast.makeText(getApplicationContext(), "Код отправлен!", Toast.LENGTH\_SHORT).show();

toResetPasswordPage(v);

}

});

}

}

@Override

public void onFailure(Call call, IOException e) {

String error = e.toString();

}

});

}

Листинг 8 — код, отвечающий за отправку сообщения при восстановлении пароля

# Тестирование

## 4.1 Протокол тестирования дизайна приложения

Для проведения тестирования дизайна приложение было выбрано устройство Redmi Note 9 c разрешением экрана 2340×1080.

Проверка была проведена на:

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

## 4.2 Протокол тестирования функционала приложения

Для проверки правильности ввода электронной почты при регистрации были проведены UnitTests, которые изображены ниже на листинге 9:

public class EmailUtils {  
  
 private static final String *VALID\_EMAIL\_REGEX* = "(^[A-Za-z0-9+\_.-]+@[A-Za-z0-9.-]+$);  
  
 @Test  
 public void validateEmail\_EmptyString() {  
 *// setup* String email = "";  
  
 *// execute* boolean actual = EmailUtils.*validateEmail*(email);  
  
 *// assert  
 assertFalse*(actual);  
 }  
@Test  
 public void validateEmail\_Missing\_Symbol() {  
 *// setup* String password = "emailmail.ru";  
  
 *// execute* boolean actual = EmailUtils.*validatePassword*(email);  
  
 *// assert  
 assertFalse*(actual);

Листинг 9 — UnitTests

Также для тестирования функционала были разработаны TestCase.

TestCase 1:

Название — регистрация.

Предусловия — приложение запущено.

Шаги:

* открыть страницу регистрации;
* заполнить все поля;
* нажать на кнопку «регистрация».

Ожидаемый результат тест кейса: пользователь зарегистрирован в системе.

Статус кейса — выполнено.

TestCase 2:

Название — авторизация.

Предусловия — приложение запущено, пользователь зарегестрирован.

Шаги:

* открыть страницу авторизации;
* заполнить все поля;
* нажать на кнопку «авторизоваться».

Ожидаемый результат тест кейса: пользователь вошел в систему.

Статус кейса — выполнено.

TestCase 3:

Название — переход на другое окно.

Предусловия —пользователь авторизован в системе.

Шаги:

* нажать на желаемую кнопку.

Ожидаемый результат тест кейса: пользователь перешел на необходимую страницу.

Статус кейса — выполнено.

## ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы было разработано мобильное приложение, позволяющее пользователю взаимодействовать с облачным хранилищем фотографий, а именно — добавлять фотографии и просматривать их.

Однако, разработка мобильного приложения является одной из нескольких частей курсовой работы. Важную роль сыграло взаимодействие клиентского приложения с сервером, находящимся на веб-хостинге. Были созданы классы на php, которые являлись “интерфейсом” и получали данные из http-get запроса, исходящего от приложения и записывали информацию в базу данных.

Во время разработки возникали трудности во время работы с изображениями — их передаче на сервер, отображение в приложении. Большую часть трудностей можно связать с малым количеством информации по теме, а также ее заметному устареванию.

СПИСОК ИСТОЧНИКОВ

1. ГОСТ 2.105–95 ЕСКД. Общие требования к текстовым документам. — М.: Изд-во стандартов, 1996. — 37 с.
2. ГОСТ Р 7.05–2008 СИБИД. Библиографическая ссылка. — М.: Стандартинформ, 2008 — 22 с.
3. ГОСТ 19.101–77 ЕСПД. Виды программ и программных документов. — М.: Стандартинформ, 2010 — 4 с.
4. ГОСТ 19.105–78 ЕСПД. Общие требования к программным документам. — М.: Изд-во стандартов, 1987. — 2 с.
5. ГОСТ 19.404–79 ЕСПД. Пояснительная записка. Требования к содержанию. — М.: Изд-во стандартов, 1987. — 2 с.
6. ГОСТ 2.106–96 ЕСКД. Требования к программным документам, выполненным печатным способом. — М.: Изд-во стандартов, 1996. — 37 с.
7. ГОСТ 19.401–78 ЕСПД. Текст программы. Требования к содержанию. — М.: Стандартинформ, 2010 — 4 с.
8. ГОСТ 19.402–78 ЕСПД. Описание программы. — М.: Стандартинформ, 2010 — 3 с.
9. ГОСТ 19.503–79 ЕСПД. Руководство системного программиста. Требования к содержанию и оформлению. — М.: Стандартинформ, 2010 — 4 с
10. phpMyAdmin [Электронный ресурс]: Документация phpMyAdmin – Режим доступа к руководству: <https://www.phpmyadmin.net/>
11. Android Developers [электронный ресурс]: Документация Android Stusio – https://developer.android.com/docs/
12. Help Center [электронный ресурс]: Документация Java: https://docs.oracle.com/en/java/
13. Git [электронный ресурс]: Документация Git: https://git-scm.com/
14. 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 ХРАНЕНИЯ ФОТОГРАФИЙ**

НАТКиГ.202100.010.000ПЗ

Выполнил:

студент группы

ПР-20.102к

Плахотный Г.А.

2022

ОГЛАВЛЕНИЕ

[Введение ………………………………………………………………………….23](#_Toc121334558)

[1 Назначение разработки ………………………………………………………..24](#_Toc121334559)

[2 Требования к мобильному приложению……………………………………..25](#_Toc121334560)

[2.1 Требования к функциональным характеристикам…………………………25](#_Toc121334561)

[2.2 Требования к надёжности …………...………………………………………25](#_Toc121334562)

[2.3 Условия эксплуатации……………………………………………………….26](#_Toc121334563)

[2.4 Требования к составу и параметрам технических средств………………..26](#_Toc121334564)

[2.5 Требования к информационной и программной совместимости…………26](#_Toc121334565)

[2.6 Требования к защите информации………………………………………….26](#_Toc121334566)

[2.7 Требования к маркировке и упаковке………………………………………26](#_Toc121334567)

[3 Требования к программной документации 27](#_Toc121334568)

[4 Технико-экономические показатели 28](#_Toc121334569)

[5 Стадии и этапы разработки 29](#_Toc121334570)

[6 Порядок контроля и приёмки 30](#_Toc121334571)

Введение

Настоящее техническое задание распространяется на разработку информационной системы «Разработка мобильного приложения для сохранения фотографий», используемого для отправки фотографий в облачное хранилище из галереи мобильного устройства, просмотра фотографий облачного хранилища в мобильном приложении, создании фотографий напрямую из мобильного приложения и отправки их в облачное хранилище.

Наименование приложения: «SMPhoto».

Краткая характеристика области применения: приложение для облачного хранения фотографий – полезное оснащение телефона любого пользователя. Пользователь может просмотреть фотографии, находящиеся в облачном хранилище, создавать фотографии в мобильном приложении и отправлять их в облачное хранилище.

Условные обозначения и сокращения:

БД – База данных;

ИС – Информационная система;

ОС – Операционная система.

Основанием для проведения разработки является Протокол № от месяца 202 года.

Наименование темы разработки – «Разработка мобильного приложения для облачного хранения фотографий».

Условное обозначение темы разработки – «Облако – SMPhoto».

1 Назначение разработки

Основное назначение мобильного приложения заключается в:

* просмотре фотографий, сохраненных на сервере;
* возможности клиента создавать фотографии и отправлять их на сервер;
* возможности клиента загружать фотографии из галереи мобильного устройства.

Лица, которые могут работать с данной Системой:

* пользователь – просматривает фотографии, регистрируется, авторизуется, отправляет фотографии на сервер.

2 Требования к мобильному приложению

## 2.1 Требования к функциональным характеристикам

Требования к составу выполняемых функций:

* регистрация и авторизация пользователей;
* просмотр имеющихся на сервере фотографий;
* добавление фотографий;
* создание фотографий в приложении.

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 Kotlin.

## 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

Стадии и этапы разработки указаны в таблице 1.

Таблица 1 – Стадии разработки

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

6 Порядок контроля и приёмки

Виды испытаний – защита курсового проекта.

Общее требования к приёмке:

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