

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



1797

Выпускная квалификационная работа

на тему

«Приложение для поиска музыкантов»

Обучающегося 4 курса
очной формы обучения
направление подготовки:

09.03.01 Информатика и вычислительная техника
направленность (профиль):

Технологии разработки программного обеспечения
Ханова Дмитрия Сергеевича

Руководитель выпускной квалификационной работы:
Кандидат педагогических наук, доцент
Авксентьева Елена Юрьевна

Санкт-Петербург
2021

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ 2

Введение 4

Глава 1. 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5

- 1.1. Постановка задачи 5
- 1.2. Обзор и сравнение аналогичных решений 6

Глава 2. 2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ 9

- 2.1. Функциональные требования 9
- 2.2. Нефункциональные требования 9
- 2.3. Описание входных и выходных данных 10
- 2.4. Моделирование требований к разработке 11
 - 2.4.1. Функциональная модель 11
 - 2.4.2. Диаграммы вариантов использования системы 11
 - 2.4.3. ER-диаграмма 13
- 2.5. Архитектура программного комплекса 14
- 2.6. Обоснование выбора технологий и программных средств разработки 15

Глава 3. Программная реализация проложения 17

- 3.1. Инструменты реализации 17
- 3.2. Реализация сервера приложения 18
 - 3.2.1. Работа с базой данных 18
 - 3.2.2. Реализация коммуникации пользователей 19
 - 3.2.3. Защита данных 20
 - 3.2.4. Рабочий сервер приложения 20
- 3.3. Реализация клиентской части приложения 21
 - 3.3.1. Реализация пользовательского интерфейса 21
 - 3.3.2. Поддержка разных версий платформы Android 21
- 3.4. Структура программы 22

Глава 4. Демонстрация работы приложения 26

- 4.1. Загрузка программы 26
- 4.2. Способ вызова программы 26
- 4.3. Режим работы программы 26
 - 4.3.1. Экран авторизации 27
 - 4.3.2. Главный экран приложения 28
 - 4.3.3. Работа с анкетами 29
 - 4.3.4. Экран сообщений 31
 - 4.3.5. Экран персональных данных 33
 - 4.3.6. Экран «Мои Анкеты» 34

Заключение 36**Список литературы 37**

ВВЕДЕНИЕ

АКТУАЛЬНОСТЬ

На сегодняшний день в России активно развивается музыкальный бизнес: появляется множество коллективов, одиночных музыкантов, музыкальных продюсеров и организаторов концертов. Продюсеры и организаторы ищут коллективы и одиночных музыкантов, последние в свою очередь могут искать первых, либо друг друга. Таким образом происходит постоянный поиск кадров.

Существует множество сервисов, которые позволяют обеспечить музыкантам и заинтересованным в них лицам связь друг с другом, но эти решения, зачастую, обладают неудобным для простого пользователя интерфейсом и множеством функций, которые только нагружают сервис, то есть, эти решения не являются специализированными именно на поиске музыкантов.

ЦЕЛЬ РАБОТЫ

Целью данной работы является разработка Android-приложения (далее Приложение), которое предоставляет платформу для поиска анкет музыкантов по определенным критериям.

Приложение позволяет пользователю создать свою анкету и, в зависимости от его требований, найти подходящую анкету другого пользователя.

ЗАДАЧИ РАБОТЫ

Для разработки приложения необходимо решить следующие задачи:

- сбор данных и формирование функциональных требований;
- уточнение функциональных требований (при необходимости);
- уточнение функциональной модели (при необходимости);
- создание архитектуры приложения;
- проектирование и реализация базы данных;
- проектирование пользовательского интерфейса;
- программная реализация сервера и клиента;
- выбор методов тестирования программного продукта

ГЛАВА 1.1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Постановка задачи

Разрабатываемое приложение будет решать следующие:

- возможность создавать анкету, в которой пользователь должен указать: тип анкеты, персональные данные, музыкальный инструмент (опционально, зависит от типа анкеты), навыки владения инструментом (опционально, зависит от типа анкеты), музыкальные предпочтения, музыкальный стиль (опционально, зависит от типа анкеты);
- поиск анкеты и предоставление информации о ее владельце: тип анкеты, город, музыкальное направление;
- уведомление пользователя о том, что его анкета была выбрана;
- предоставление платформы для возможности коммуникации между пользователями.
- Для реализации поставленных задач в системе предусмотрены
- следующие модули:
- модуль управления профилем;
- модуль поиска;
- модуль коммуникации;

Модуль управления профилем позволяет добавлять информацию о пользователях системы. Есть два типа пользователей системы: одиночные музыканты и музыкальные коллективы. Для одиночных музыкантов обязательна для заполнения следующая информация: персональные данные (имя, город, возраст), музыкальный инструмент и навыки владения им, музыкальные предпочтения. Для музыкальных коллективов: персональные данные (название коллектива, город), музыкальный инструмент и навыки владения им у искомого музыканта, музыкальный стиль / жанр. Таким образом в системе будут появляться

публичные данные, по которым будет происходить поиск и взаимодействие между пользователями.

Модуль поиска реализовывает поиск данных в системе по устанавливаемым фильтрам и категориям, что существенно упростит нахождение интересующих пользователей анкет.

Модуль коммуникации представляет собой мессенджер и позволит пользователям Приложения взаимодействовать друг с другом.

Разрабатываемое мобильное приложение будет доступно для Android. В дополнение к этому будет доступна веб-версия Приложения.

Таким образом, практическая значимость работы заключается в создании системы для объединения на одной площадке музыкантов и музыкальных коллективов самых разных жанров и умений. Приложение позволит пользователям быстро находить анкеты, которые будут отбираться исходя из их запросов.

1.2. Обзор и сравнение аналогичных решений

ВКонтакте – многофункциональная социальная сеть, объединяющая миллионы пользователей. Ресурс позволяет обмениваться сообщениями, организовывать группы по интересам, просматривать объявления, участвовать в обсуждениях. Имеет множество других функций.

Avito – интернет-сервис для размещения объявлений о товарах, вакансиях и резюме на рынке труда, а также услугах от частных лиц и компаний.

Splice – интернет-сервис для совместного создания музыки, продажи плагинов. Имеется возможность создания собственных сообществ, загрузки собственных произведений и плагинов. Имеет свой собственный плеер. Отсутствует русский язык.

Kompoz – интернет-сервис, позволяющий одновременно с другими пользователями создавать музыку с помощью облачных технологий. Имеется функционал купле-продажи музыкальных материалов. Отсутствует русский язык.

Vocalizr – интернет-сервис, предоставляющий музыкальным продюсерам найти вокалиста для своих композиций. На размещенные проекты проводятся кастинги.

В таблице 1 приведен результат анализа аналогичных решений.

Табл. 1. Сравнительная таблица аналогов

| Параметр | ВКонтакте | Avito | Splice | Kompoz | Vocalizr |
|--|-----------|-------|--------|--------|----------|
| Является ли сервис специализированным | - | - | + | + | + |
| Наличие русскоязычной локализации | + | + | - | - | - |
| Наличие мобильного приложения | + | + | - | - | - |
| Наличие бесплатной версии | + | + | + | + | + |
| Возможность составления анкеты пользователей | + | + | + | + | + |
| Поиск данных с использованием комбинируемых фильтров | + | + | - | - | + |
| Взаимодействие пользователей в системе (отправка / чтение сообщений) с возможностью загрузки медиафайлов | + | + | + | + | + |
| Нацеленность на офлайн сотрудничество пользователей | - | - | - | - | - |

В таблице 1 представлены позиции:

1. «+» – наличие функции у обозреваемого аналога.
2. «-» – отсутствие функции у обозреваемого аналога.

Исходя из результатов обзора аналогичных решений, можно сделать следующие выводы:

- ВКонтакте сервис с самым большим функционалом, но он не является специализированным;
- Большая часть рассмотренных сервисов предлагает онлайн-формат создания музыки, а не офлайн-встреч, формирования коллективов и т. д.;
- Большая часть рассмотренных сервисов не имеет русскоязычной локализации, что существенно затрудняет их использование для русскоязычных пользователей;

В ходе анализа предметной области и обзора аналогичных решений было установлено, что разрабатываемая система будет иметь уникальный функционал, отсутствующий в аналогичных системах. **Приложение** нацелено на офлайн сотрудничестве пользователей и является специализированным.

ГЛАВА 2.2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

2.1. Функциональные требования

Разрабатываемое приложение должно удовлетворять следующим функциональным требованиям:

- пользователь должен иметь возможность регистрироваться и авторизоваться в системе;
- пользователь должен иметь возможность редактировать данные в анкете;
- система должна предоставлять пользователю анкеты, исходя из критериев поиска;
- Система должна предоставлять пользователям возможность обеспечить коммуникацию друг с другом;

2.2. Нефункциональные требования

В результате анализа функциональных требований и представленных задач, были сформулированы следующие нефункциональные требования:

- серверная часть приложения должна быть написана на языке программирования PHP;
- клиентская часть приложения должна быть реализована с использованием фреймворка Cordova;
- приложение должно быть скомпилировано под платформу Android.

2.3. Описание входных и выходных данных

Входными данными для разрабатываемого комплекса являются данные авторизации пользователей, данные анкет и тексты сообщений.

Выходными данными являются сформированные страницы музыкантов / музыкальных коллективов и диалоги между пользователями.

Подробный перечень входных и выходных данных программного комплекса представлен в таблице 2.

Таблица 2 – Входные и выходные данные

| Функция | Входные данные | Выходные данные |
|---|---|---|
| Авторизация: шаг №1 | Адрес электронной почты, пароль | Код авторизации |
| Авторизация: шаг №2 | Код авторизации, пароль | Экраны приложения, информация о пользователе |
| Добавление / редактирование данных пользователя | Фамилия, имя, отчество, место проживание, телефон, e-mail, о себе, фотография | Занесение данных в систему хранения, отображение в системе |
| Добавление анкеты | Тип, название / имя, место, инструмент, направление | Занесение данных в систему хранения, отображение в системе |
| Поиск анкеты | Данные для поиска | Информация по запросу |
| Отправка сообщения | Текст сообщения, идентификаторы отправителя и получателя | Занесение данных в систему хранения, отображение сообщения в диалоге |

2.4. Моделирование требований к разработке

Рисунок 1. Функциональная модель Приложения

2.4.1. Функциональная модель

При построении функциональной модели программы клиента использовалась DFD-методология (Data Flow Diagrams). Эта методология графического структурного анализа, описывающая внешние по отношению к системе источники, а также логические функции, потоки данных и хранилища данных, к которым осуществляется доступ. Разрабатываемая система должна удовлетворять функциональной модели, представленной на Рисунке 1.

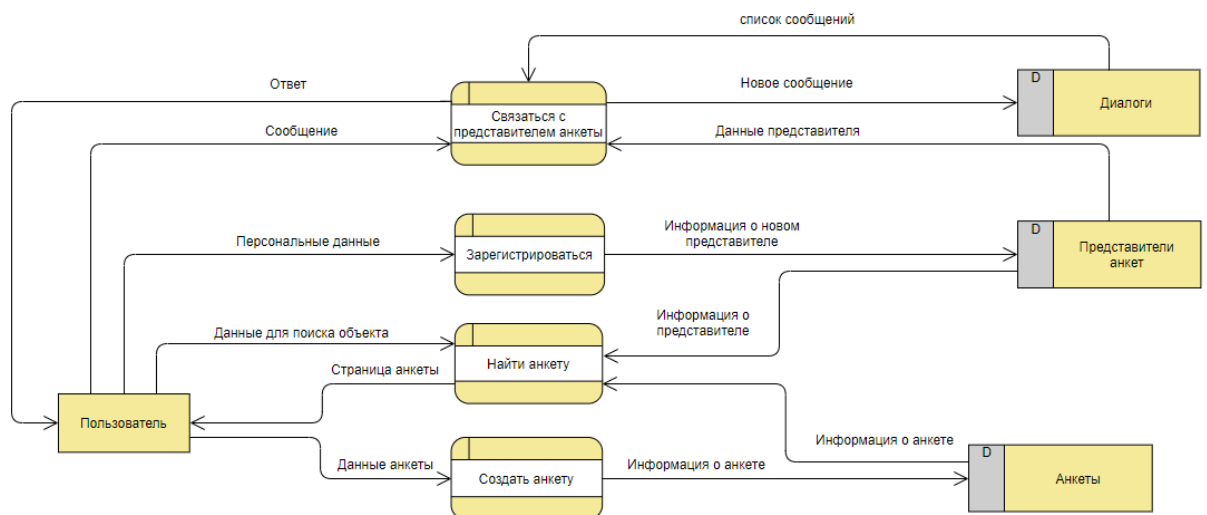


Рисунок 1. Функциональная модель Приложения

2.4.2. Диаграммы вариантов использования системы

Для наглядного описания функциональных возможностей и поведения системы была использована диаграмма вариантов использования (UML Use-Case).

На диаграмме изображены все возможные варианты действий, которые пользователь может осуществить во время использования системы. В диаграмме вариантов использования выделены следующие актеры – Пользователь и Неопределенный пользователь. Результаты представлены на Рисунке 2.

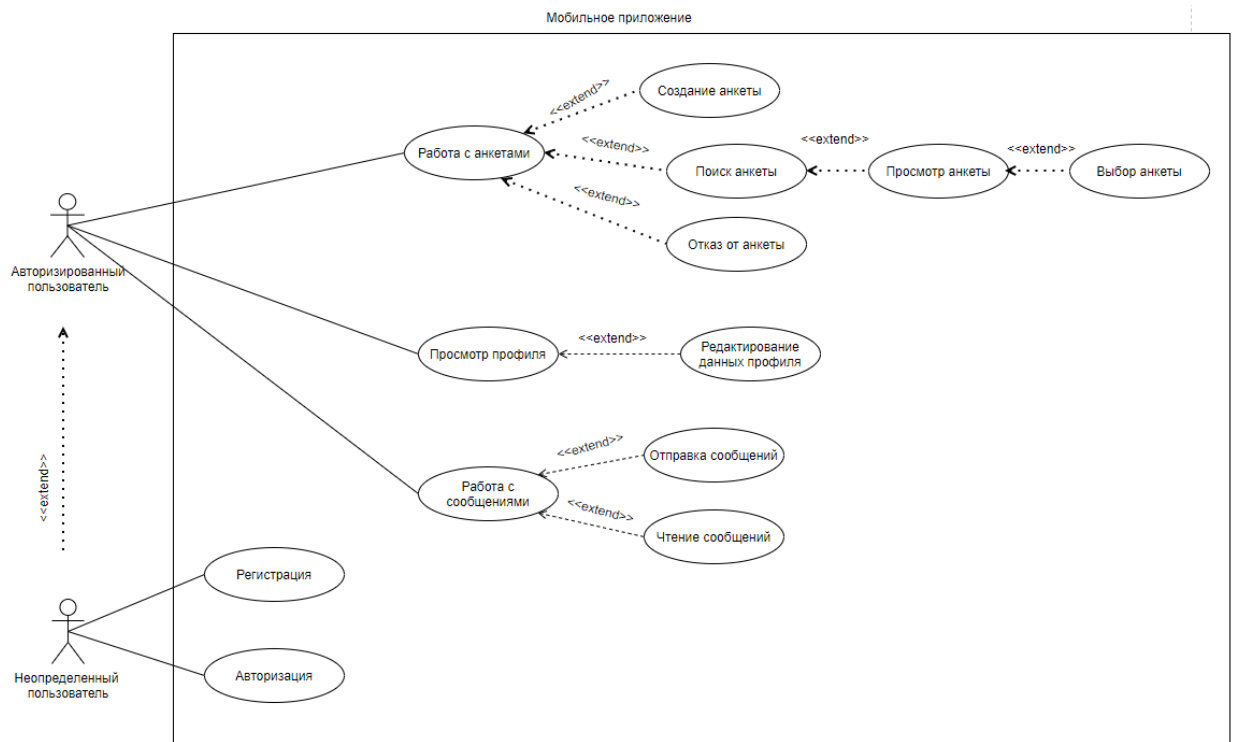


Рисунок 2. Диаграмма вариантов.

Неопределенный пользователь может пройти *процедуру регистрации* в Приложении, указав в советующей форме номер телефона. При первом запуске приложения и авторизации происходит создание профиля нового пользователя, логином которого является номер телефона.

Неопределенный пользователь может пройти *процедуру авторизации*, указав в соответствующей форме зарегистрированный номер телефона и генерируемым кодам авторизации, приходящим по смс

Авторизованный пользователь может *работать с анкетами*. Работа с анкетами может заключаться в:

- создании анкеты;
- поиске анкеты, который заключается в просмотре анкеты, и выборе анкеты, то есть отмечать нужные из предложенных ему Приложением;

— отказ от анкеты.

Авторизированный пользователь может *просматривать профиль*. Также, пользователь может редактировать данные профиля — менять информацию о себе.

Авторизированный пользователь может *работать с сообщениями*. Работа с сообщениями может заключаться в отправке сообщений другим пользователям и просмотра их сообщений (Эта функция становится возможной, когда пользователи приняли анкеты друг друга).

2.4.3. ER-диаграмма

Модель данных должна удовлетворять модели, представленной в виде ER-диаграммы, показанной на Рисунке 3.

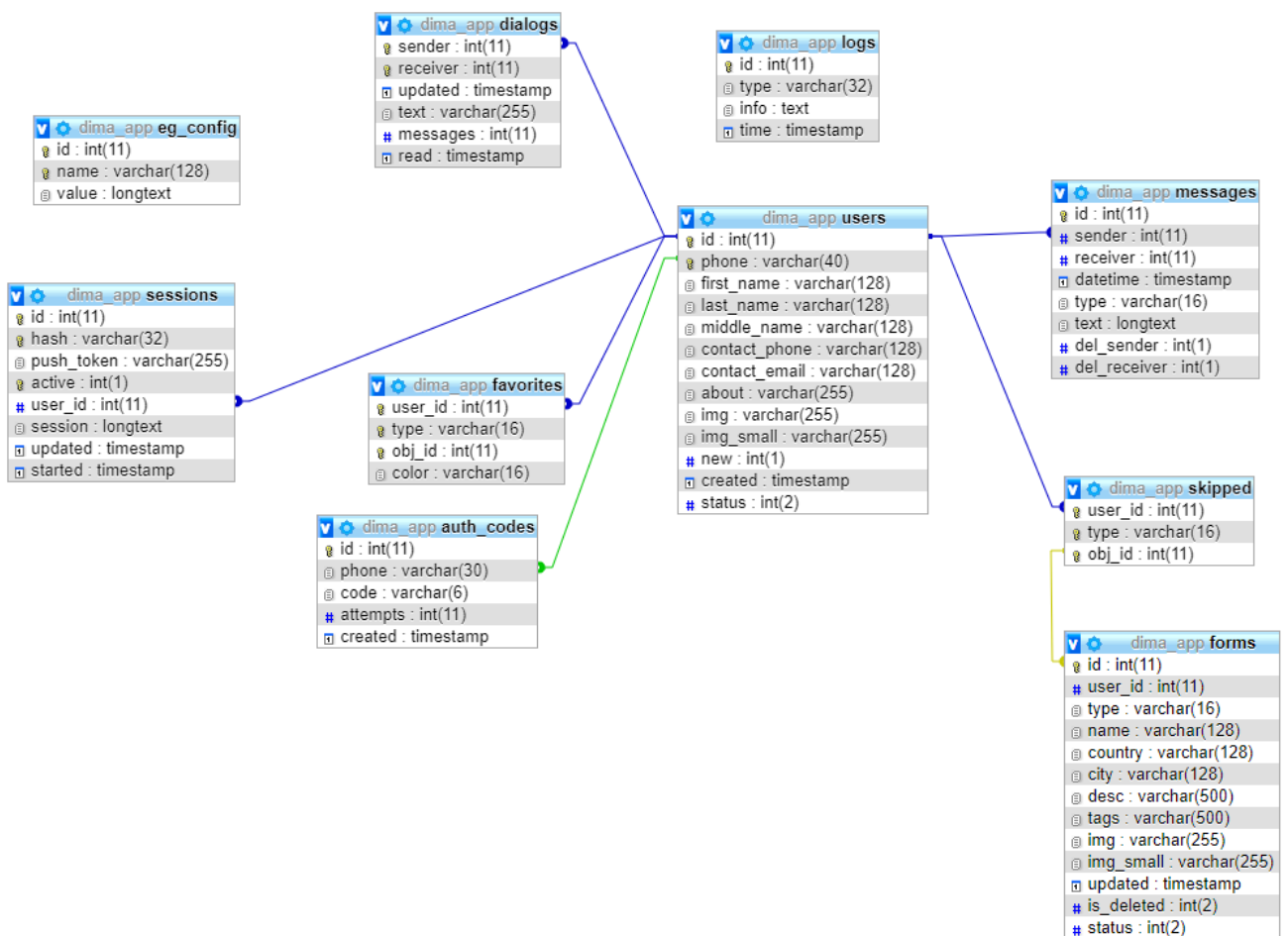


Рисунок 3. ER-диаграмма.

2.5. Архитектура программного комплекса

Разрабатываемый программный продукт будет реализован как мобильное приложение с использованием архитектуры «клиент-сервер».

Архитектура «клиент-сервер» предполагает наличие двух компонентов: сервера, способного принимать запросы и интерпретировать их, производя специфицируемые запросом действия с БД, после чего результаты отправляются клиенту. Задачей клиента является отображение результатов с помощью интерфейса пользователя.

Приложение должно быть спроектировано согласно паттерну MVC (Model View Controller) с однонаправленным потоком данных.

Модель (Model) содержит в себе всю логику приложения, она хранит и обрабатывает данные, при этом не взаимодействуя с пользователем напрямую.

Представление (View) отображает данные, которые ему передали. В веб-приложении оно обычно состоит из HTML-шаблонов страниц.

Контроллер (Controller) отвечает за выполнение запросов, пришедших от пользователя.

Данная концепция заключается в том, что все три модуля производят обработку пользовательских данных в строгом порядке. Например, действия контроллера не могут привести к изменению вида, минуя модель.

2.6. Обоснование выбора технологий и программных средств разработки

В качестве среды разработки была выбрана IDE от JetBrains PhpStorm, поскольку она предоставляет все необходимые для разработки функции:

1. Интеллектуальный редактор PHP кода с подсветкой синтаксиса, автодополнением кода, расширенными настройками форматирования кода, предотвращением ошибок.
2. HTML, CSS, JavaScript редактор. Отладка и модульное тестирование для JS. Поддержка HTML5, CSS, SCSS. Полный набор инструментов для фронтенд-разработки.
3. Удаленное развертывание приложений и автоматическая синхронизация с использованием FTP, SFTP, FTPS и др.
4. Инструменты работы с базами данных, SQL редактор.

Для разработки мобильного приложения будет использована Apache Cordova. Apache Cordova — это платформа разработки мобильных приложений с открытым исходным кодом. Она позволяет использовать стандартные веб-технологии, такие как HTML5, CSS3 и JavaScript для кросс платформенной разработки, избегая родного языка разработки для каждой из мобильных платформ. Таким образом, приложение будет доступно для устройств на Android и iOS, имея общую кодовую базу. Такой выбор был сделан, поскольку по требованию заказчика разработка должна быть проведена в течение короткого промежутка времени.

Для разработки клиентской части будет использоваться язык программирования JavaScript с библиотекой jQuery, визуальная составляющая интерфейса будет реализована при помощи HTML 5 в связке с CSS 3.

Для реализации серверной части будет использоваться язык программирования PHP.

Для управления базой данных будет использоваться серверная СУБД MySQL:

1. MSQL – это бесплатная СУБД с большим количеством пользователей.
2. Безопасность – большое количество функций, обеспечивающих безопасность, которые поддерживаются по умолчанию.
3. Масштабируемость – MySQL легко работает с большими объемами данных и легко масштабируется.
4. Скорость – упрощение некоторых стандартов SQL позволяет MySQL значительно увеличить производительность.

ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРОЛОЖЕНИЯ

3.1. Инструменты реализации

Для создания мобильного приложения была использована платформа разработки мобильных приложений Apache Cordova. В качестве среды разработки была использована IDE PhpStorm. Серверная часть системы была реализована при помощи языка PHP в связке с СУБД MySQL 5.6. Клиентская часть приложения была реализована на языке JavaScript с библиотекой jQuery, графический интерфейс был реализован с помощью HTML 5 и CSS 3.

На Рисунке 4 приведен программный код языка PHP, который функцию отправки кода авторизации на телефон пользователя для регистрации / авторизации в Приложении.

```
22     public function code() {
23         if (!$_POST['phone']) {
24             Tpl::i()->json([
25                 'result' => 'error',
26                 'error' => '"phone" required.'
27             ], false, false);
28         }
29
30         if ($this->session->info['user_id']) {
31             Tpl::i()->json([
32                 'result' => 'error',
33                 'error' => 'Already logged in.'
34             ], false, false);
35         }
36
37         $code = $this->user->get_auth_code($_POST['phone']);
38         if (!$code)
39             Tpl::i()->json([
40                 'result' => 'error',
41                 'error' => 'Codes daily limit.'
42             ], false, false);
43         if ($code)
44             Sms::send($_POST['phone'], "Код авторизации Becarre: {$code}");
45
46         $this->page['code'] = rand(100000, 999999); //not real code. just for fun
47
48         Tpl::i()->json();
49     }
50 }
```

Рисунок 4. Код функции.

3.2. Реализация сервера приложения

Для корректной работы сервера были установлены следующие программные средства:

- СУБД MySQL 5.6;
- интерпретатор PHP версии 7.2;
- веб-сервер Apache версии не ниже 2.4;
- nginx 1.14.1.

3.2.1. Работа с базой данных

Работа с базой данных осуществлялась при помощи расширения MySQLi, которое предоставляет полный функционал MySQL 4.1 и выше. Создание / изменение таблиц данных происходит посредством отправки SQL-запросов к базе данных, генерируемых моделями при помощи языка PHP.

```

110 public function check_auth_code($phone, $code) {
111     $phone = prepare_phone($phone);
112
113     $time = date(YMDHIS, strtotime(Cfg::$c['auth']['codes_ttl'] . " seconds ago"));
114     $true_code = (array)$this->db->q(
115         "SELECT * FROM auth_codes WHERE phone='$phone' AND created > '$time' ORDER BY created DESC LIMIT 1"
116     )->fetch_assoc();
117
118     if ($true_code['id']) {
119         $this->db->q("UPDATE auth_codes SET attempts=attempts+1 WHERE id={$true_code['id']}");
120     }
121
122     return $true_code['attempts'] <= Cfg::$c['auth']['code_entry_attempts'] AND $true_code['code'] == $code;
123 }
124

```

Рисунок 5. Пример использования SQL-запроса к базе данных.

На Рисунке 5. приведен листинг функции `check_auth_code`, которая используется для проверки корректности ввода кода авторизации пользователем. На строках 114–116 и 119 продемонстрировано формирование SQL-запроса. В первом случае отправляется запрос к таблице “auth_code” с целью выяснения, выписывался ли когда-либо код авторизации пользователю с номером телефона, который хранится в переменной `$phone`. Во втором же случае, к таблице отправляется запрос, который обновляет количество попыток ввода кода для данного телефона.

3.2.2. Реализация коммуникации пользователей

При регистрации пользователь указывает только номер мобильного телефона, после чего на него высылается сгенерированный сервером код авторизации – строка длиной в 6 цифр. При авторизации в приложении процедура проходит аналогичным образом.

СМС-рассылка осуществляется с использованием сервиса SMSC.

В случае, если при отправке сообщения возникла какая-либо ошибка, сервер заносит ее в лог-файл и отправляет разработчику электронное сообщение об ошибке.

```

4  class Sms extends Model
5  {
6      public static function send($phone, $text, $charset = 'utf-8'){
7          Cfg::i()->load('cfg/sms');
8          $phone = prepare_phone($phone);
9          $query = http_build_query(array(
10             'login' => Cfg::$c['sms']['login'],
11             'psw' => Cfg::$c['sms']['password'],
12             'phones' => $phone,
13             'mes' => $text,
14             'charset' => $charset
15         ));
16
17         self::mail_log(implode("\n", [
18             "Телефон: $phone",
19             "Текст: $text",
20             "-----",
21             "Ip: {$_SERVER['REMOTE_ADDR']}"
22         ]));
23
24         return file_get_contents("http://smc.ru/sys/send.php?$query");
25     }
26
27     public static function mail_log($text){
28         $mails = [
29             'khanovdima02@gmail.com'
30         ];
31
32         foreach ($mails as $mail){
33             mail($mail, 'Becarre sms log', $text);
34         }
35     }
36 }

```

Рисунок 6. Листинг кода отправки СМС.

На Рисунке 6. приведен листинг класса «Sms», который отвечает за отправку СМС на мобильный номер пользователя. На строках 9–15 в переменную \$query записывается сгенерированная строка GET-параметров, включающая в себя:

- логин и пароль разработчика приложения в сервисе SMSC;
- номер мобильного телефона, на который будет осуществлена отправка сообщения с кодом авторизации;
- сгенерированный код авторизации;
- кодировка сообщения, для корректного его отображения.

На 24 строке с помощью метода «file_get_contents» осуществляется GET-запрос сервису SMSC.

3.2.3. Защита данных

Защита от несанкционированного доступа: вход в систему осуществляется по номеру телефона и коду авторизации, приходящему на указанный номер. Реализована система сессий, для которых генерируется хэш из криптографически безопасных псевдослучайных байтов (функция random_bytes). Количество попыток авторизации ограничено. Обмен данными осуществляется по протоколу HTTPS, SSL-сертификат предоставляется центром сертификации LetsEncrypt.

3.2.4. Рабочий сервер приложения

В качестве рабочего сервера было решено арендовать хостинг-сервис TimeWeb, так как он находится на территории РФ (По законодательству Российской Федерации персональные данные пользователя должны храниться на территории государства), а также выгоден в тарифном плане: за счет того, что приложение не занимает полноценный экземпляр операционной системы, а ее часть (Хостинг размещает на одном сервере большое количество сайтов от разных пользователей).

3.3. Реализация клиентской части приложения

3.3.1. Реализация пользовательского интерфейса

Система реализуется с web-интерфейсом для мобильных устройств с обеспечением многопользовательского режима работы и удалённым доступом.

При разработке приложения соблюдались рекомендации по реализации интерфейса для мобильных Android.

Интерфейсы разделов типизированы, цветовая палитра имеет оттенки фиолетового и серого цветов. При возникновении ошибок в работе подсистемы на экран выводится сообщение с наименованием ошибки и с рекомендациями по её устранению.

3.3.2. Поддержка разных версий платформы Android

Для каждого проекта, реализованном для платформы Android, возникает необходимость обозначить самую старшую версию платформы, на которой приложение будет работать корректно. Чем поддерживаемая версия старше, тем больше пользователей сможет пользоваться приложением, но, при этом, это грозит возможным неполным функционалом с работой API младших версий.

В качестве самой старшей версии была выбрана 5.0 (Lollipop, API 21), так как согласно данным, которые приводит Android Studio – интегрированная среда разработки для платформы Android, она будет работать примерно на 94,1% Android-устройств (см. рис. 7).

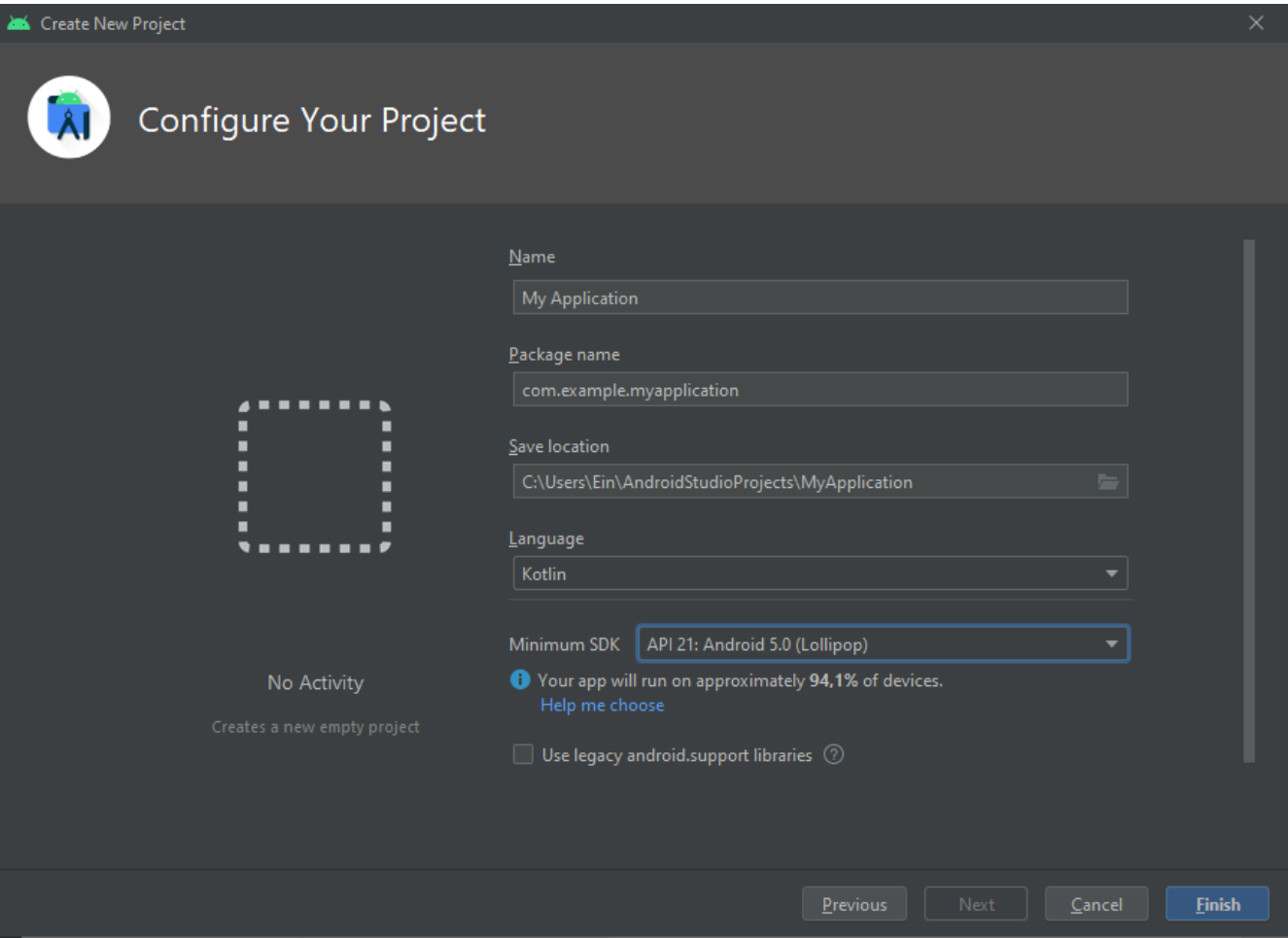


Рисунок 7. Статистика Android-версий.

3.4. Структура программы

Основные файлы и распределение функций программы представлены в Таблице 3.

Таблица 3 – структура программы

| | | | |
|----------------------------------|-----|--|--|
| assets/application/js/ app.js | App | Основной файл с главными методами приложения | is_admin(); api(method, params, callback); showError(message, callback); init(); run(); selectMenu(stage, type_view, not_loaded_data); |
|----------------------------------|-----|--|--|

| | | | |
|----------------|---------|---|---|
| | | | showDialog(h1, desc, ok_callback, ok_text, cancel_callback, cancel_text, more_callback, more_text); showAlert(h1, desc); |
| app_catalog.js | Catalog | Инициализация анкеты пользователей. Взаимодействие с элементами анкеты. | init(); run(); changeType(type); searchFormShowMore(); searchFormClear(); searchResultClear(); clickSearchButton(); searchRun(); clickItem(self); viewItem(id, type); clickFavorite(self, e); |
| app_login.js | Login | Авторизация пользователей, выход из системы | showDisplay(step, noclear); submitStep1(); submitStep2(); logoutClick(); logout(); |
| app_profile.js | Profile | Редактирование и загрузка профиля | init(); showProfile(); load(); saveProfile(); clickToLoadImage(self); changeInputImage(self); |
| app_good.js | Good | Работа с пользовательскими анкетами: создание, редактирование, удаление, сохранение | create(); savegood(); getList(); loadList(); clickGood(self); showEdit(id); cancelEdit(); removeItem(); |

Продолжение Табл.3

| 1 | 2 | 3 | 4 |
|-------------------------------|---------|--|---|
| app_message.js | Message | Работа с мессенджером. Загрузка списка диалогов и отображение сообщений в конкретном диалоге. Отправка сообщений | init(); makeTplDialogs(); getDialogs(); showChatClick(); showChatDisplay(user_id, username, img, back_stage); backToDialogs(); loadDialog(after_message_id, before_message_id, append_type); appendMessageToChat(message, append_type); dialogAjaxLoader(); dialogsAjaxLoader(); sendClick(); |
| /core/controller/api/auth.php | auth | Контроллер в API, отвечающий за авторизацию пользователя и создание сессий | code(); check_session(); activate_session(); session_push_token(); login(); logout(); |
| goods.php | goods | Контроллер в API, отвечающий за работу с анкетами | get(); list(); edit(); remove(); |
| message.php | message | Контроллер в API, отвечающий за получение списка диалогов и сообщений, отправку и поиск сообщений | send(); deleteMessage(); getDialog(); getDialogs(); deleteDialog(); search(); getNewMessagesCount(); |
| upload.php | upload | Контроллер в API, отвечающий за загрузку изображений в систему | img(); |

Продолжение Табл.3

| 1 | 2 | 3 | 4 |
|----------|------|--|------------------------------|
| user.php | user | Контроллер в API, отвечающий за работу с профилем и пользователями (представителями) в системе | get(); edit(); list(); |

В структуре программы также присутствуют следующие директории:

- core/model/ – хранит модели, содержащие методы для
- контроллеров в API;
- tpl/application – хранит шаблоны для графического интерфейса

ГЛАВА 4. ДЕМОНСТРАЦИЯ РАБОТЫ ПРИЛОЖЕНИЯ

4.1. Загрузка программы

Для работы мобильного приложения требуется загрузить и установить его на Android устройство. В данный момент **Приложение** еще не размещено на Google Play Store, поэтому пользователю предлагается воспользоваться стандартными средствами установщика системы Android.

4.2. Способ вызова программы

Загрузка производится путем открытия установленного приложения

4.3. Режим работы программы

Пользователь взаимодействует с программой через графический интерфейс.

4.3.1. Экран авторизации

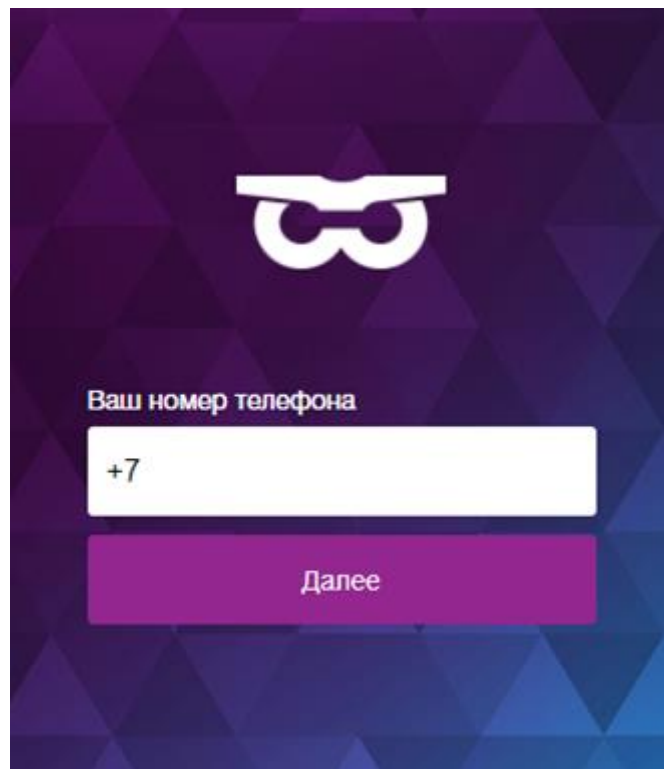


Рисунок 8. Экран авторизации.

Для того, чтобы авторизоваться / зарегистрироваться в приложении, пользователь должен ввести номер мобильного телефона и нажать на кнопку «Далее», после чего система переадресует пользователя на экран подтверждения кода авторизации (см. рис. 9).

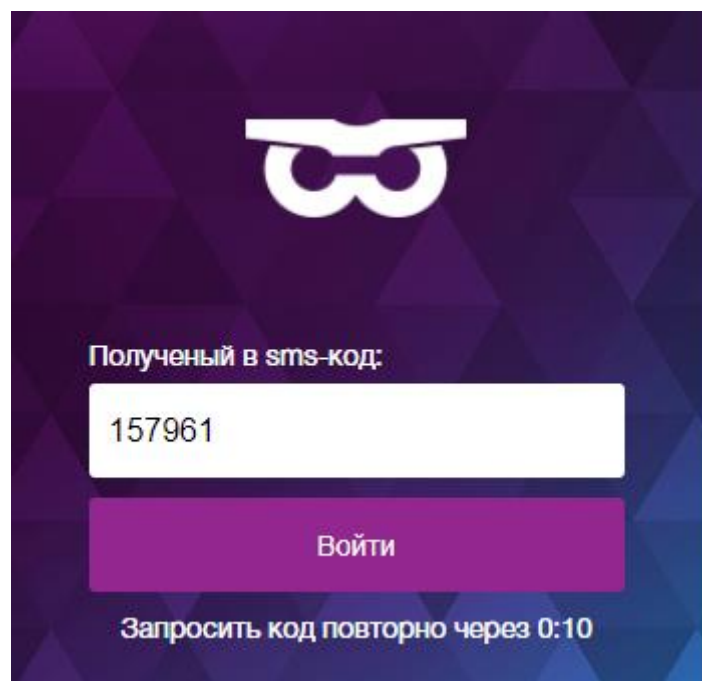


Рисунок 9. Экран подтверждения авторизации.

4.3.2. Главный экран приложения

В случае, если регистрация / авторизация прошла успешно, система переадресует пользователя на главный экран приложения (см. рис. 10), где расположены следующие элементы: анкеты пользователей, поиск, фильтры.

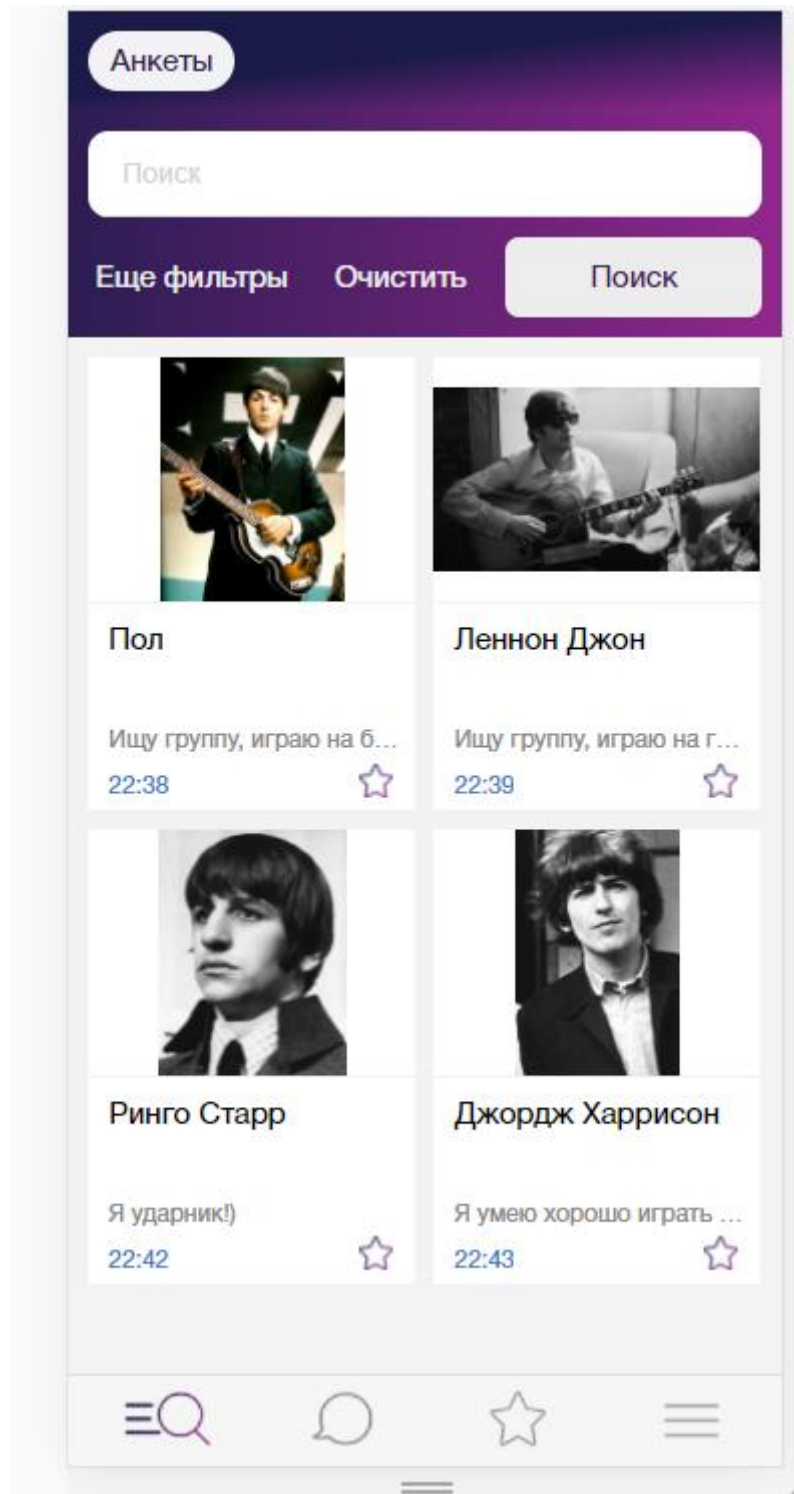


Рисунок 10. Главный экран приложения.

4.3.3. Работа с анкетами

Пользователь взаимодействует с анкетами (см. рис 11.). При нажатии на кнопку «Пропустить» выбранная анкета больше не будет появляться в поиске, а на ее месте появится следующая из списка. Если пользователь нажмет кнопку «Связаться», то система перенаправит его на отдельный экран (см. рис. 12), где должна происходить коммуникация между пользователями.



Рисунок 11. Анкета пользователя.



Рисунок 12. Экран диалога.

Также, пользователь имеет возможность просмотреть аккаунт своего собеседника. Для этого ему нужно нажать на блок «данные контакта» (см. рис. 12), после чего система перенаправит его на экран, где содержится информация о пользователе (см. рис. 13).

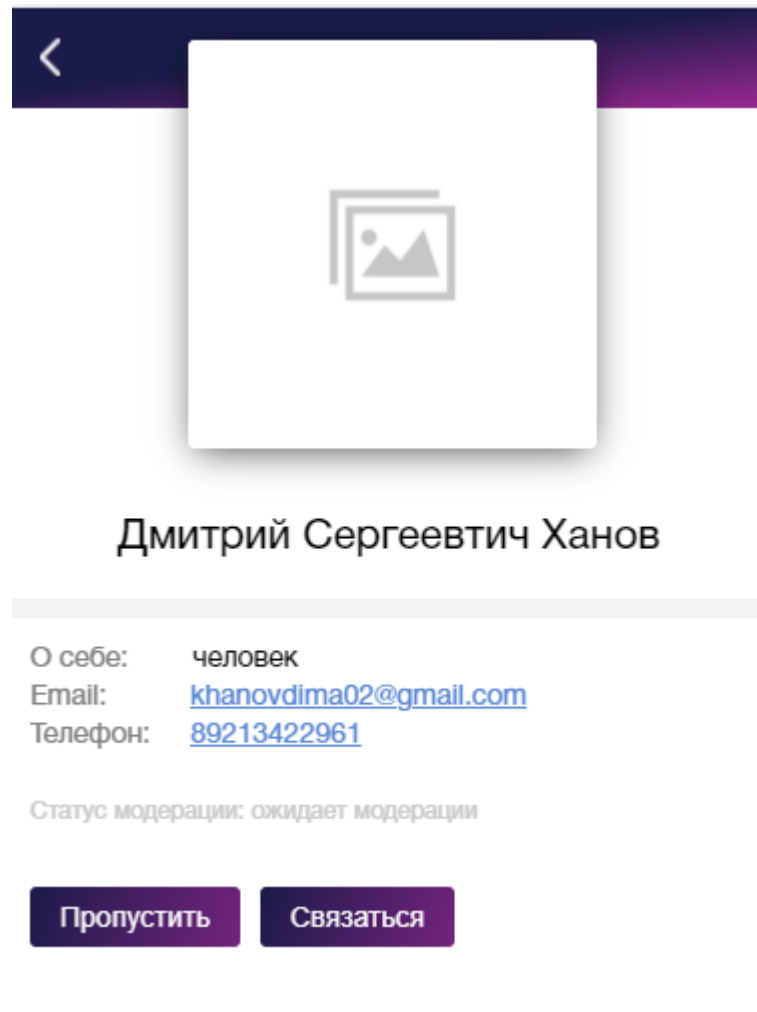


Рисунок 13. Данные пользователя.

4.3.4. Экран сообщений

Пользователь может посмотреть входящие сообщения, перейдя на соответствующий экран приложения. Для это ему следует нажать кнопку, изображенной на Рисунке 14. в красной рамке. после чего система перенаправит его на нужный экран (см. рис. 14).

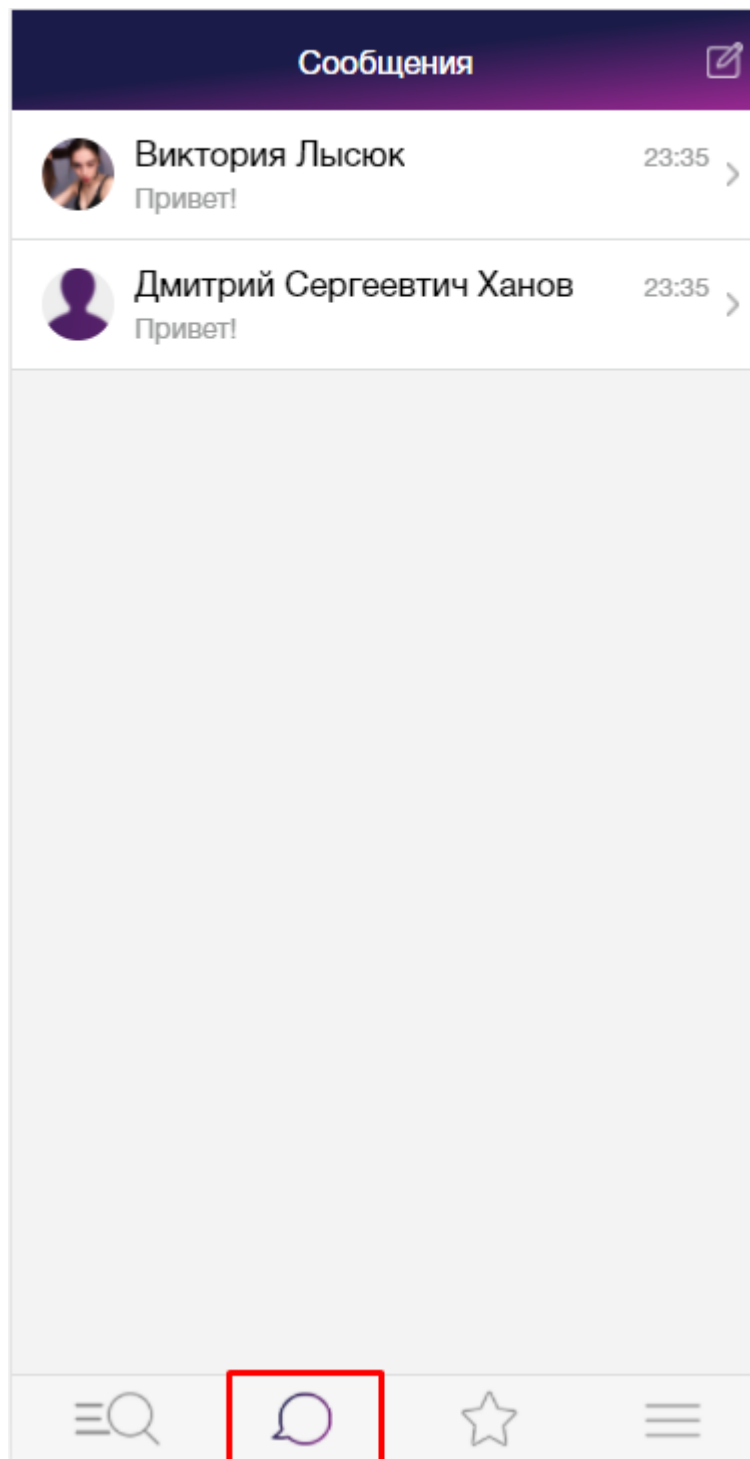


Рисунок 14. Экран списка диалогов.

Для того, чтобы пользователю перейти в интересующий его диалог, следует нажать советуемый элемент списка.

4.3.5. Экран персональных данных

На данном экране пользователь может:

- просмотреть / изменить информацию о себе;
- связаться с тех. поддержкой;
- просмотреть свои анкеты;
- настроить приложение;
- выйти из системы.

Пользователь может просмотреть / изменить информацию о себе, перейдя на соответствующий экран приложения. Для это ему следует нажать кнопку, изображенной на Рисунке 15. в красной рамке. после чего система перенаправит его на нужный экран (см. рис. 15).

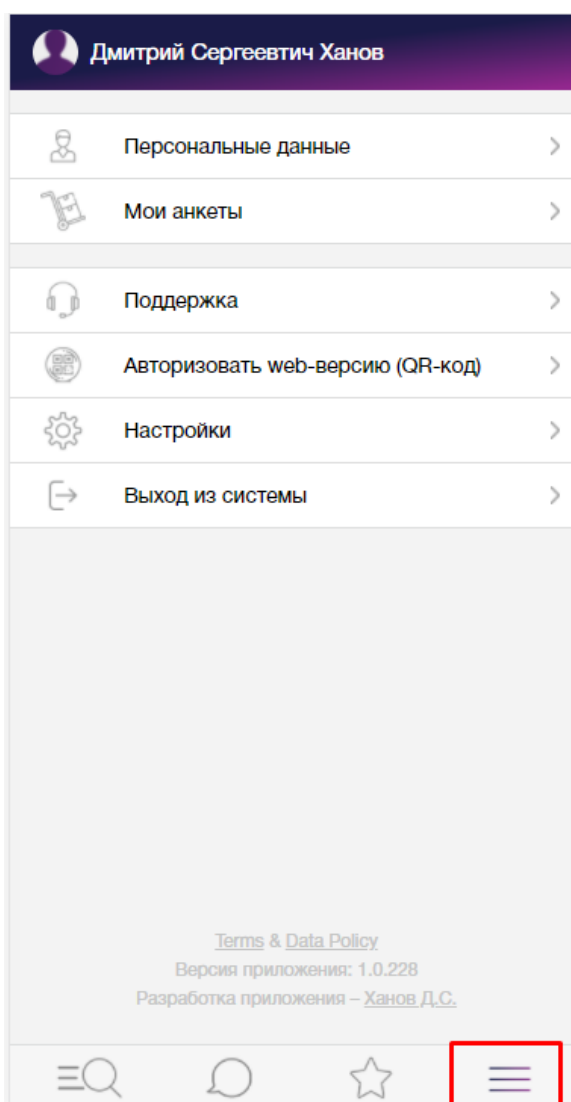


Рисунок 15. Экран персональных данных

4.3.6. Экран «Мои Анкеты»

Для осуществления работы с анкетами пользователь должен перейти на экран «Мои анкеты» (см. рис. 15), после чего система переадресует его на соответствующий экран. (см. рис. 16). На данном экране пользователь может:

- добавить анкету;
- просмотреть / редактировать / имеющиеся анкеты.

Для того, чтобы создать анкету, пользователь должен нажать на кнопку, которая изображена на Рисунке 16 в красной рамке, после чего система переадресует его соответствующий экран (см. рис 17).

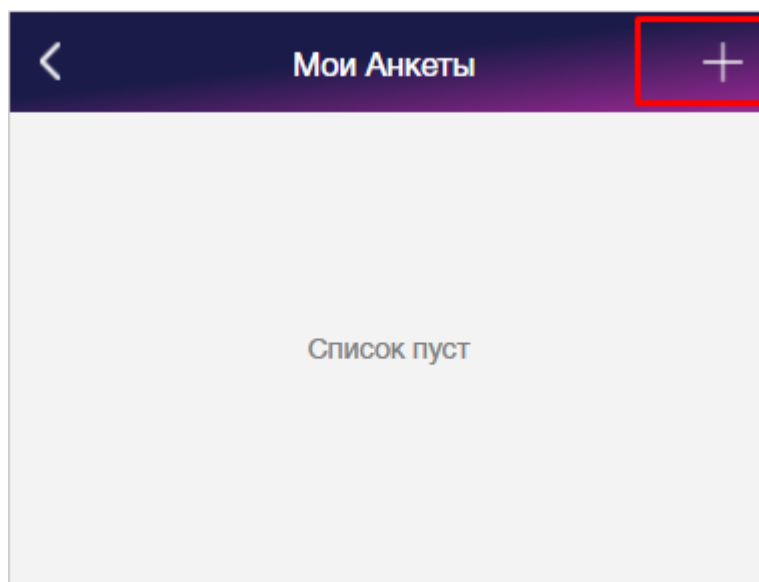


Рисунок 16. Экран «Мои анкеты».


| Отмена | Редактирование | Готово |
|---|----------------|--------|
| Название/Имя | | |
| Страна | | |
| Город | | |
| Теги для поиска | | |
| Описание | | |
|  | | |
| Удалить | | |

Рисунок 17. Экран создания анкеты.

На данном экране пользователю предложено внести информацию о себе, теги, по котором будет осуществляться поиск его анкеты, и описание. Также пользователь может добавить фотографию (поддерживаемые форматы: .png, .jpeg) нажав блок с изображением фотоаппарата, который находится после блока «Описание». Чтобы завершить создание анкеты, пользователю следует нажать на кнопку «Готово», после чего система перенаправит его на экран «Мои анкеты».

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы было разработано мобильное приложение для поиска музыкантов, также были выполнены следующие пункты:

- изучена предметная область;
- проведен анализ аналогичных решений;
- обоснована целесообразность разработки;
- сформулированы функциональные требования к разработке;
- проведено функциональное моделирование системы;
- обоснован выбор средств реализации;
- реализована разработка;

Из-за ограниченного срока выполнения работы остались нереализованными следующие пункты, которые планируется добавить в последующей работе над **Приложением**:

- поддержка устройств под управлением операционной системы IOS;
- интеллектуальный подбор анкет, который будет реализован с помощью нейросетей;
- возможность добавления в анкеты аудиофайлов.

СПИСОК ЛИТЕРАТУРЫ

1. 47 готовых решений для создания Web-сайта (+ DVD-ROM) / А.Г. Богданов и др. - Москва: Огни, 2009. - 272 с.
2. Apache Cordova API Reference [Электронный ресурс] – URL: <https://cordova.apache.org/docs/en/2.4.0/> (Дата обращения: 20.02.2021).
3. htmlbook.ru [Электронный ресурс] – URL: <http://htmlbook.ru/html5> (Дата обращения 24.02.2021).
4. jQuery русская документация [Электронный ресурс] – URL: <https://jquery-docs.ru/> (Дата обращения: 02.03.2021).
5. kodaktor.ru [Электронный ресурс] – URL: <https://kodaktor.ru/g/vkr> (Дата обращения 15.03.2021).
6. MDN Web Docs [Электронный ресурс] – URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (Дата обращения: 23.02.2021).
7. MySQL [Электронный ресурс] – URL: <https://dev.mysql.com/doc/> (Дата обращения: 23.02.2021).
8. PHP Hypertext Preprocessor [Электронный ресурс] – <https://www.php.net/manual/ru/intro.mysql.php> (дата обращения: 21.02.2021).
9. SWIRT Профессиональная разработка технической документации [Электронный ресурс] – URL: <https://www.swrit.ru/gost-espd.html> (Дата обращения 16.04.2021).
10. W3C [Электронный ресурс] – URL: <https://www.w3.org/Style/CSS/specs.ru.html> (Дата обращения 24.02.2021).
11. Бер, Бибо jQuery. Подробное руководство по продвинутому JavaScript / Бибо Бер. - М.: Символ-плюс, 2013. - 540 с.

- 12.Веллинг, Л. Разработка веб-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томсон. - 4-е изд. - М. и др. : Вильямс, 2010 (СПб.). - 847 с.
- 13.ГОСТ Р 52653-2006 Информационно-коммуникационные технологии в образовании. Термины и определения. – М.: Стандартинформ, 2018. – 13 с.
- 14.Консультант Плюс [Электронный ресурс] – URL: http://www.consultant.ru/document/cons_doc_LAW_61801/ (Дата обращения 28.04.2021).
- 15.Минник, К. JavaScript для чайников / К. Минник, Е. Холланд. - М. [и др.] : Диалектика, 2016. - 313 с.
- 16.Флэнаган, Дэвид JavaScript. Карманный справочник / Дэвид Флэнаган. - М.: Вильямс, 2015. - 320 с.