

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

«Информатика»

кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой

А. С. Кузнецов

подпись

инициалы, фамилия

« \_\_\_\_\_ » \_\_\_\_\_ 2019 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.04 – Программная инженерия

код – наименование направления

Разработка модуля «Абитуриент» для мобильного приложения «Я в СФУ»

тема

Руководитель

доцент, к. т. н.

А. В. Хныкин

подпись, дата

должность, ученая степень

инициалы, фамилия

Выпускник

М. О. Фабричкина

подпись, дата

инициалы, фамилия

Нормоконтролер

доцент, к. т. н.

О. А. Антамошкин

подпись, дата

должность, ученая степень

инициалы, фамилия

Красноярск 2019

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка модуля «Абитуриент» для мобильного приложения «Я в СФУ» » содержит 54 страницы текстового документа, 22 использованных источника, 25 рисунков, 1 приложение.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, БАЗА ДАННЫХ, ANDROID, АБИТУРИЕНТ, НАПРАВЛЕНИЯ ПОДГОТОВКИ, КАРТА КАМПУСА, ДОВУЗОВСКОЕ УПРАВЛЕНИЕ.

Целью работы является разработка модуля «Абитуриент» для мобильного приложения «Я в СФУ».

Для достижения поставленной цели решались следующие задачи.

- 1) Проанализирована предметная область.
- 2) Сделан обзор и анализ существующих решений.
- 3) Спроектирован и разработан модуль мобильного приложения:
  - а) разработана база данных;
  - б) реализована графическая часть клиента;
  - в) реализована функциональная часть клиента;
  - г) реализован сервер модуля «Абитуриент» мобильного приложения «Я в СФУ».
- 4) Проведено тестирование и устранение ошибок в разработанном модуле.

В ходе выполнения ВКР было получено свидетельство о государственной регистрации программы для ЭВМ №2019613682 от 21.03.2019 г. В настоящее время идут переговоры о возможности интеграции мобильного приложения «Я в СФУ» в информационное пространство СФУ.

## СОДЕРЖАНИЕ

Введение.....	5
1 Анализ предметной области .....	7
1.1 Анализ требований целевой аудитории к разрабатываемому программному продукту .....	7
1.2 Особенности предметной области и решаемых задач .....	10
1.3 Выбор средств разработки .....	13
1.3.1 Среда разработки.....	13
1.3.2 Реализация графической части клиента .....	15
1.3.3 Реализация функциональной части клиента .....	15
1.3.4 Реализация сервера .....	16
1.4 Основные возможности проектируемого модуля.....	17
1.5 Обзор существующих решений.....	18
1.6 Выводы.....	21
2 Проектирование программного продукта .....	22
2.1 Архитектура приложения.....	22
2.2 Структура базы данных .....	25
2.3 Выводы.....	27
3 Описание работы программного продукта.....	28
3.1 Серверная часть приложения.....	28
3.2 Клиентская часть приложения.....	31
3.2.1 Выбор типа пользователя и справочное меню.....	31
3.2.2 Направления подготовки .....	32
3.2.3 Институты .....	33
3.2.4 Общежития .....	34
3.2.5 Контакты .....	36
3.2.6 Новости .....	37
3.2.7 Карта кампуса.....	38
3.3 Выводы.....	40

4 Тестирование программного продукта .....	41
4.1 Тестирование удобства использования.....	41
4.2 Функциональное тестирование.....	42
4.3 Тестирование совместимости .....	45
4.4 Выводы.....	47
Заключение .....	49
Список сокращений .....	50
Список использованных источников .....	51
Приложение А Свидетельство о регистрации программы для ЭВМ .....	54

## ВВЕДЕНИЕ

IT-технологии стали неотъемлемой частью нашей жизни. Стационарные компьютеры, ноутбуки, планшеты и смартфоны – у каждого устройства из этих категорий наблюдается тенденция уменьшения размеров и наращивания мощностей. Чем меньше размер устройств, тем проще пользоваться ими не только дома, но также, без особых усилий, брать их с собой, что позволяет человеку всегда быть на связи, вследствие чего люди все чаще предпочитают использовать небольшие переносные устройства, в частности смартфоны, о чем сказано в пункте 1.2.

Актуальность разработки приложения «Я в СФУ» заключается в отсутствии адаптивности сервисов СФУ под мобильные устройства и в разрозненности и несистематизированности информации, которая, по проведенным исследованиям, действительно нужна абитуриентам (пункт 1.1).

Идея создания мобильного приложения для студентов и абитуриентов по отдельности не является новой, но на момент создания продукта в Play Market не были найдены полноценные аналоги приложения «Я в СФУ», которое совмещает в себе функционал, как для действующих студентов, так и для абитуриентов ВУЗа. Приложение «Я в СФУ» делится на разделы «Студент» и «Абитуриент». Существуют приложения, которые выполняют функционал только для одного из представленных разделов, и каждые имеют те или иные недостатки, рассмотренные далее в пункте 1.5.

Объектом исследования является целевая аудитория, в которую входят выпускники школ и студенты СФУ. Предметом исследования является процесс сбора и анализа информации с основного сайта СФУ, а также сбор и обработка информации с целевой аудитории.

Целью данной работы является создание модуля «Абитуриент» клиент-серверного мобильного приложения «Я в СФУ», с помощью которого произойдут следующие изменения:

- станет доступна систематизированная информация для абитуриента по институтам, общежитиям, новостям и контактам в одном мобильном приложении;

- появится карта кампуса с описанием объектов;

- появится возможность подбирать направление подготовки, используя специальные фильтры;

- станут доступны новости кампуса и довузовского управления СФУ в одном приложении.

Был определен список задач, которые должен решать реализуемый модуль, это такие задачи, как:

- сбор и систематизация информации об институтах, общежитиях, контактах, и местоположении объектов кампуса ВУЗа для абитуриентов в одном месте;

- сбор и обработка данных по направлениям профессиональной подготовки, а также реализация возможности применения фильтров для поиска подходящих направлений подготовки;

- вывод объектов кампуса университета на экран с применением Google Maps;

- отображение новостей ВУЗа и довузовского управления в мобильном приложении.

Клиент-серверное мобильное приложение «Я в СФУ» должно стать помощником и проводником абитуриентов и студентов университета, сопровождая их как от идеи поступления, так и на всем протяжении обучения. Слияние двух этих разделов в одном приложении делает его уникальным, повышая его ценность за счет отсутствия необходимости переустановки или поиска нового приложения при переходе из состояния абитуриента в состояние студента, достаточно авторизоваться в разделе «Студент», используя пару логин/пароль от всех сервисов СФУ, которую выдадут в начале обучения.

## **1 Анализ предметной области**

### **1.1 Анализ требований целевой аудитории к разрабатываемому программному продукту**

Зачастую, разработчики делают основной акцент на процессе программирования, уделяя недостаточно внимания общению с клиентом. Многие разработчики не умеют или не любят собирать требования, а клиенты часто не хотят тратить свое время на разработку требований к ПО. Недостаточный объем информации, поступающей от клиента, требования, сформулированные не полностью, их кардинальное изменение – это основные причины, вследствие которых командам, работающим в области информационных технологий, не удается вовремя и в рамках бюджета предоставить клиентам всю запланированную функциональность [1].

Для того, чтобы не допустить таких ошибок, важно грамотно подойти к управлению требованиями. Управление требованиями – это систематический подход к выявлению, организации и документированию требований к системе [2].

Для выявления требований целевой аудитории к разрабатываемому программному продукту, было проведено анкетирование с использованием Google Forms. Анкета содержала в себе 5 вопросов, 3 из которых касались раздела «Абитуриент». Далее мы рассмотрим результаты, которые были получены по итогам анкетирования. На рисунке 2 представлен сводный результат в процентном соотношении о статусе испытуемого.

Как видно из результатов, большинство испытуемых составили абитуриенты, требования которых мы и рассмотрим в рамках выполнения данной дипломной работы по разработке модуля «Абитуриент» клиент-серверного мобильного приложения «Я в СФУ».



Рисунок 1 – Сводный результат ответов о статусе анкетированного

У абитуриентов было два вопроса, каждый из которых представлял собой как поля с выбором ответа, для того чтобы сориентировать анкетлируемого, так и из полей для самостоятельного заполнения. В результате анкетирования были получены следующие результаты, представленные на рисунках 2 и 3.



Рисунок 2 – Результаты анкетирования по вопросу, посвященному основным возможностям модуля «Абитуриент»



**3. ПРЕДСТАВЬТЕ СИТУАЦИЮ, ВЫ АБИТУРИЕНТ, КОТОРЫЙ ПОСТУПАЕТ В УНИВЕРСИТЕТ В ДРУГОМ ГОРОДЕ И ПЛАНИРУЕТ ЗАСЕЛЯТЬСЯ В ОБЩЕЖИТИЕ. КАКУЮ ИНФОРМАЦИЮ ВЫ СЧИТАЕТЕ НЕОБХОДИМОЙ?**

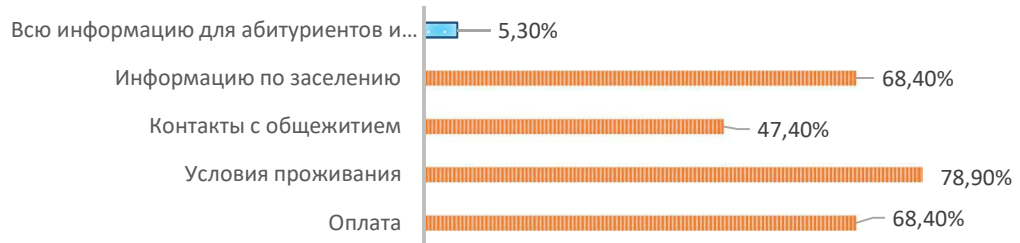


Рисунок 3 – Результаты анкетирования по вопросу, посвященному информации по общежитиям

Оранжевым цветом представлены фиксированные варианты ответов, голубым цветом, варианты, представленные анкетлируемыми самостоятельно. Рассмотрим результаты на рисунке 2. Как видно из результатов, целевая аудитория модуля «Абитуриент» наиболее заинтересована в получении информации о направлениях обучения, информации об общежитиях и информации о контактах.

По рисунку 3 видно, что в общежитиях большинство абитуриентов волнуют условия и оплата проживания, а также информация по заселению.

Итак, по результатам анкетирования были составлены основные требования к разрабатываемому модулю, а именно:

- модуль должен систематизировать и фильтровать данные о направлении подготовки бакалавров и специалистов университета согласно требованиям пользователя;

- необходимо отображать новости ВУЗа, касающиеся абитуриентов и приемной кампании;

- необходимо отображать все объекты кампуса университета на карте с описанием местоположения и назначения объекта;

- в модуле должна быть реализована возможность отображения списка институтов и расширенной информации по ним, такой как:

- фотографии;
- адрес;
- режим работы;
- контакты.

- необходимо формировать список общежитий и отображать расширенную информацию об общежитиях, включающую в себя такие составляющие, как:

- адрес общежития, с отображением его на карте;
- карусель с фотографиями общежития;
- стоимость проживания в общежитии;
- бытовые особенности общежития;
- руководство и правила заселения.

- необходимо формировать перечень контактной информации для пользователя.

## **1.2 Особенности предметной области и решаемых задач**

Смартфоны захватывают рынок, об этом можно судить исходя из распределения мирового Web-трафика за 2018 год по различным устройствам [3], представленного на рисунке 4.

Как видно из графика, более половины интернет трафика во всем мире проходит через смартфоны. Также за 2018 год средняя скорость мобильной передачи данных возросла на 30%. Более быстрое интернет-соединение помогает снизить уровень стресса, что следует из публикации компании Ericsson в своем отчете Ericsson Mobility Report [4].



Рисунок 4 – Распределение WEB-трафика по устройствам в 2018 году

Во многом современный смартфон сопоставим с компьютером, а в некоторых аспектах может и превосходить его. Сейчас уже никого не удивишь смартфоном с мощностями среднего компьютера, при этом размер позволяет брать его всегда с собой, не тратя больших усилий. Это стало одной из причин решения о реализации клиент-серверного мобильного приложения для студентов и абитуриентов СФУ. Сайт СФУ и сервис «Мой СФУ» содержат важную информацию, но большинство этой информации не требуется студенту или абитуриенту постоянно, также не всегда удобно заходить на эти ресурсы с персонального компьютера, ведь его иногда просто может не быть под рукой, а адаптивная верстка данных сервисов под мобильный вид отсутствует.

Из преимуществ смартфона перед персональным компьютером можно выделить такие факторы, как:

- компактность;
- переносимость;
- сравнительно низкая цена;
- удобное сенсорное управление.

Когда есть идея написания мобильного приложения, необходимо принять решение, какую платформу для приложения выбрать. Для того, чтобы сделать

выбор, необходимо обратиться к статистической информации по распределению операционных систем проданных смартфонов в России. Согласно сайту сбора глобальной статистики StatCounter [5], на момент сентября 2018 сложилась следующая ситуация, представленная на рисунке 5.

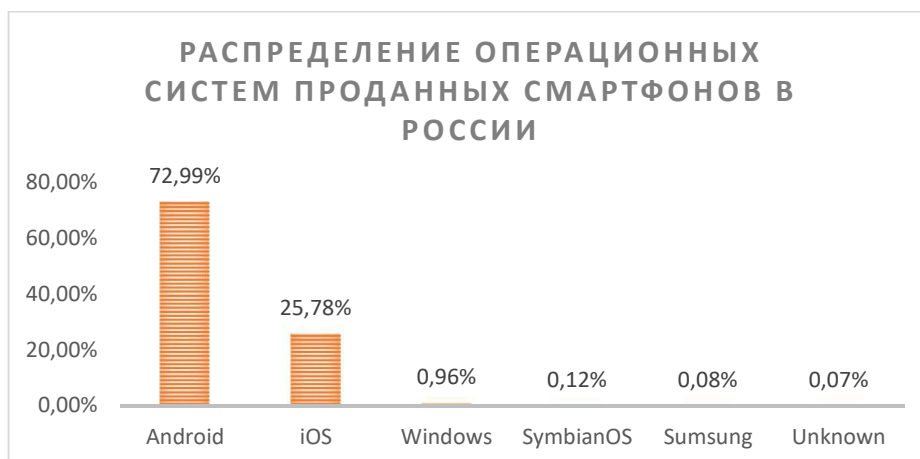


Рисунок 5 – Распределение операционных систем проданных смартфонов в России

Как видно из гистограммы, на рынке мобильных устройств в России преобладают устройства под управлением Android, в пользу которого и был сделан выбор.

После выбора операционной системы устройства разработки, необходимо решить, какую минимальную версию платформы поддерживать. Для определения минимальной версии обратимся к официальному сайту Android Developers [6]. На рисунке 6 представлены данные с панели распределения версий платформ.

Как можно судить из гистограммы 88,9% пользователей системы Android используют версию платформы 5 и выше. Также, начиная с версии Android 5.0 Lollipop, был введен Material Design, яркий и отзывчивый дизайн пользовательского интерфейса. Исходя из этих данных и из желания дать пользователям наиболее красивый и интерактивный интерфейс, было принято решение поддерживать версию платформы не ниже 5, что соответствует версии API от 21 выше.

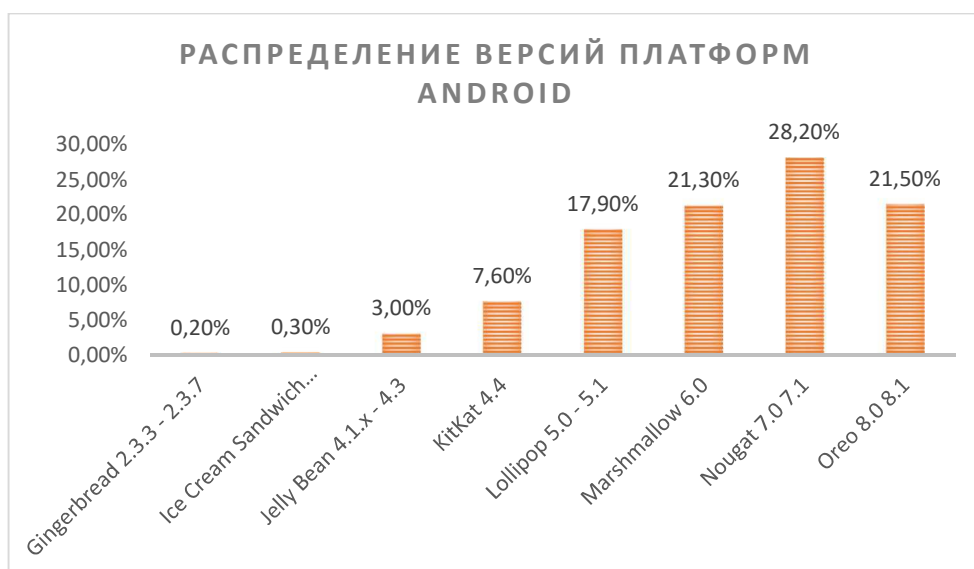


Рисунок 6 – Распределение версий платформ Android

## 1.3 Выбор средств разработки

### 1.3.1 Среда разработки

#### 1.3.1.1 Среда разработки мобильного приложения

Для нативной разработки мобильных приложений под Android OS, как правило, используют IntelliJ IDEA или Android Studio. Для создания клиентской части мобильного приложения была выбрана среда разработки IntelliJ IDEA, разработанная JetBrains, так как она включает в себя все функции Android Studio, а также дает возможность разрабатывать не только под Android OS, так как поддерживает большой перечень ЯП, что немаловажно в случае клиент-серверной мобильной разработкой. Данная среда включает в себя следующие преимущества [7]:

- умное автодополнение, инструменты для анализа качества кода, удобная навигация, расширенные рефакторинги и форматирование для Java, Groovy, Scala, Clojure и Erlang;
- профессиональный набор инструментов для разработки Android-приложений;
- поддержка JavaFX 2.0, интеграция с SceneBuilder;

- Дизайнер интерфейса для Swing;
- интеграция с автоматизированными инструментами сборки и управления проектом, включая Maven, Gradle, Ant и другими;
- инструменты для тестирования с поддержкой JUnit, TestNG, Spock, ScalaTest и spec2;
- интеграция с системами управления версиями, включая Git, Subversion, Mercurial и CVS.

IntelliJ IDEA позволяет быстрее и качественнее писать и менять код, подсказывает подходящие имена, находит подходящие методы, что позволяет быстрее и продуктивнее разрабатывать приложения.

### 1.3.1.2 Среда разработки сервера

Для работы с серверной частью мобильного приложения на Node.js была выбрана IDE WebStorm, также от компании JetBrains. WebStorm [8] – среда для разработки на JavaScript, которая подходит для client-side-разработки, создания приложений на Node.js и мобильных приложений на React Native.

Главное достоинство WebStorm – это удобный и умный редактор для JavaScript, HTML и CSS, который также поддерживает TypeScript, CoffeeScript, Dart, Less, Sass и Stylus и фреймворки, например, Angular, React и Vue.js.

WebStorm, обеспечивая подсветку и автодополнение кода, проверяет его на ошибки, помогает быстро перемещаться по проекту и безопасно вносить изменения с помощью рефакторингов. В WebStorm есть инструменты для отладки кода и интеграция с системами управления версиями.

Было принято решение использовать IDE одного семейства продуктов для реализации разработанного продукта, за счет их наибольшей совместимости при реализации и тестировании разрабатываемого ПО.

### **1.3.2 Реализация графической части клиента**

Для построения макетов приложения для Android традиционно используется язык XML. Начало XML было положено в 1996 году. Специалисты собрались с целью создания подмножества SGML, предназначенного специально для работы в WEB среде. Это подмножество получило название расширяемого языка разметки, в нем используются все преимущества структурной разметки и нет той сложности, которая присуща SGML.

XML предоставляет разработчикам инструменты, необходимые для выпуска новых видов приложений, которые охватывают не только простые Web-приложения, но и базы данных, электронные коммерческие системы, а также фактически любые системы отображения информации. Это возможно, поскольку, в отличие от HTML, XML содержит все для работы с данными. Он направлен на структурирование данных, а внешнее представление данных возложено на HTML-ориентированные таблицы стилей [9].

### **1.3.3 Реализация функциональной части клиента**

При выборе языка разработки было решено придерживаться нативной разработки на языке Java.

История Java восходит к 1991 году, когда группа инженеров из компании Sun под руководством Патрика Нотона (Patrick Naughton) и члена Совета директоров Джеймса Гослинга (James Gosling) занялась разработкой небольшого языка, который можно было бы использовать для программирования бытовых устройств, например, контроллеров для переключения каналов кабельного телевидения [10]. Сейчас Java является самым популярным и востребованным языком программирования в Enterprise проектах, что привело к устойчивому росту популярности и в более маленьких проектах. Java имеет следующие качества:

- язык Java независим от платформы, на которой выполняются программы: один и тот же код можно запускать под управлением операционных систем Windows, Solaris, Linux, Macintosh, что очень важно, так как работа нас проектом в целом, велась разработчиками на разных платформах;

- полностью объектно-ориентированный язык;

- память в языке Java освобождается автоматически с помощью механизма сборки мусора;

- Java обладает большой библиотекой программ для передачи данных на основе таких протоколов TCP/IP;

- многопоточность.

При разработке модуля язык Java использовался для получения данных с сервера, получения и записи данных в локальную базу данных, управления графическими элементами экрана и управлением взаимодействия с пользователем.

Для хранения данных на устройстве пользователя была выбрана база данных Realm. Одно из основных преимуществ Realm – это скорость. База данных Realm быстрее альтернативных ORM на Android примерно в 100 раз (по сравнению с SQLite – в 10 раз). А скорость взаимодействия с БД напрямую влияет на время отклика приложения [11].

### **1.3.4 Реализация сервера**

Для того, чтобы получить полномасштабный и функциональный проект, необходимо использовать сервер, для операций с получением и обработкой пользовательских данных, а также, обработкой запросов и ответов сервисов университета.

Сервер приложения «Я в СФУ» построен на базе Node.js по технологии REST. Node.js – среда выполнения JavaScript, основанная на JavaScript движке. Движок JavaScript, используемый Node (V8, разработанный компанией Google),



не только компилирует JavaScript во внутренний машинный код (подобно C или C++), но и делает это прозрачным образом, так что с точки зрения пользователя код ведет себя как чистый интерпретируемый язык программирования. Отсутствие отдельного шага компиляции уменьшает сложность обслуживания и развертывания [12].

JavaScript выполняет действие на стороне клиента, а Node – на сервере. С Node.js проще масштабировать проект. При одновременном массовом подключении к серверу Node.js работает асинхронно, расставляя приоритеты и распределяя ресурсы грамотнее.

Для обмена данными между сервером и клиентом используется сравнительно новый формат JSON, который основан на подмножестве языка программирования JavaScript, определенного в стандарте ECMA-262 3rd Edition-December 1999 [13]. Его структура делает его читаемым для человека, также за счёт своей лаконичности по сравнению с XML формат JSON является более подходящим при сериализации сложных структур, что и стало основной причиной выбора данного формата.

#### **1.4 Основные возможности проектируемого модуля**

Разработанный программный модуль имеет следующие возможности:

- отображение информации по новостям с основного новостного сайта и сайта довузовского управления ВУЗа;
- отображение контактных данных и реквизитов университета;
- подбор подходящих абитуриенту направлений подготовки, используя специальные фильтры по форме обучения, стоимости обучения и предметам, которые необходимы для поступления на выбранное направление подготовки;
- отображение кратких сведений об институтах университета с фотографиями и контактными данными приемной института;

- вывод на карту кампуса перечня площадок с отображением всех объектов учебного заведения, таких, как учебные корпуса, административные корпуса, общежития, спортивные сооружения и медицинский центр. Также при желании, пользователь может разрешить приложению определять его местоположение, что поможет ориентироваться в затруднительных ситуациях.

## **1.5 Обзор существующих решений**

Как было сказано ранее, идея разработки приложения для абитуриента не нова, существуют два таких приложения для абитуриентов СФУ, а именно:

- «Абитуриент СФУ» [14];
- «Поступай в СФУ» [15].

Приложение «Абитуриент СФУ» является официальным приложением университета, а приложение «Поступай в СФУ» принадлежит стороннему разработчику.

Так или иначе, оба приложения имеют те или иные недостатки, для более подробного обзора решений, рассмотрим каждое из них подробнее.

Приложение «Поступай в СФУ» имеет функционал, ограниченный следующими функциями:

- подбор специальностей по фильтрам;
- отображение информации о местоположении приемных институтов университета.

Пример применения фильтрации по названию, предметам и форме обучения можно увидеть на рисунке 7.

Название: \_\_\_\_\_

Предметы

Математика  Обществознание

Русский  Литература  Химия

Физика  Информатика  Биология

Иностраный язык  История  География

Форма обучения

Очная  Заочная  Очно-заочная

ПОИСК

Рисунок 7 – Фильтрация в приложении «Поступай в СФУ»

Далее рассмотрим приложение «Абитуриент СФУ». Приложение имеет более полный функционал в сравнении с предыдущим приложением, он включает в себя такие разделы, как:

- новости для абитуриентов и школьников;
- информация об университете;
- направления подготовки и их фильтрация;
- календарь абитуриента;
- АИС «Абитуриент»;
- карта кампуса.

Список доступных возможностей приложения «Абитуриент» СФУ можно увидеть на рисунке 8.

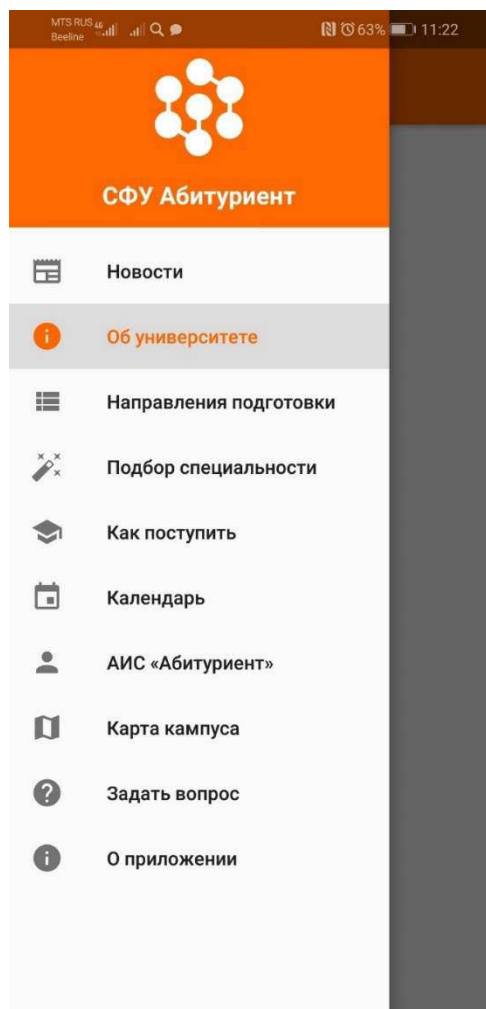


Рисунок 8 – Возможности приложения «Абитуриент СФУ»

Из недостатков по работе данного приложения можно выделить следующие аспекты:

- сбой приложения при попытке открыть новости;
- при выборе элемента карусели для его масштабирования всегда открывается либо первый элемент, либо элемент, открытый ранее;
- при перелистывании элемента карусели с измененным масштабом изображения на один сдвиг в ту или иную сторону, масштаб элемента не переходит в изначальное состояние, что доставляет некоторые неудобства использования;
- при отображении подобранных специальностей по платной форме обучения отсутствует стоимость обучения и количества платных мест.

## 1.6 Выводы

В результате анализа аналогов были рассмотрены 2 аналога модуля «Абитуриент» приложения «Абитуриент СФУ» и «Поступай в СФУ». Оба приложения являются бесплатными и свободно распространяются в Play Market. Среди них наибольший интерес представляет приложение «Абитуриент СФУ». Преимуществом этого приложения перед разработанным модулем является наличие синхронизации с АИС «Абитуриент». При рассмотрении разработанного модуля можно выделить функции, отсутствующие в официальном приложении, а именно:

- при отображении подобранных специальностей отображаются количество платных мест и стоимость обучения (при наличии), мест со специальными правами, средний и проходной баллы;

- контактная информация по каждому институту;

- подробная систематизированная информация об общежитиях с реальными фотографиями, информацией по особенностям быта, стоимости обучения и заселению, отражением адреса на карте.

Перечисленный выше функционал повышает ценность разработки модуля «Абитуриент» за счет отсутствия в нем недостатков аналогов и наличия возможностей, выходящих за рамки функционала аналогов.

## 2 Проектирование программного продукта

### 2.1 Архитектура приложения

Простейшая архитектура распределенного приложения, называемая архитектурой клиент-сервер, предполагает, что приложение состоит из двух частей: серверной, оказывающей услуги, и клиентской, пользующейся услугой.

Для клиент-серверного взаимодействия приложения «Я в СФУ» используется архитектурный стиль REST. REST представляет собой архитектурный стиль, который можно использовать для создания программных средств, в которых клиенты (агенты пользователей) могут отправлять запросы службам (конечным точкам) [16]. Рой Филдинг, один из главных авторов HTTP-протокола, ввёл термин REST в 2000 году. Системы с поддержкой REST, называются RESTful-системами. REST имеет ряд преимуществ перед другими архитектурными стилями, а именно:

- ресурсы, возвращаемые в ответ на запрос GET, можно кэшировать множеством разных способов;
- REST дает возможность включать в каждый ресурс все состояния, необходимые для обработки конкретного запроса;
- службы RESTful не должны иметь никаких побочных эффектов, при запросе ресурса с помощью команды GET;
- команды PUT и DELETE можно использовать несколько раз, и результат будет таким же, как при их первом использовании.

Управление передачей данных основывается на протоколе передачи данных HTTP. Основными и наиболее часто используемыми HTTP методами являются POST, GET, PUT и DELETE, менее используемые OPTIONS, HEAD и TRACE. Ниже указаны определения основных методов из спецификаций RFC 2068 [17].

Метод POST используется для запроса, при котором адресуемый сервер принимает объект, включенный в запрос, как новое подчинение ресурса, идентифицированного запрашиваемым URI в строке запроса.

Метод GET позволяет получать любую информацию (в форме объекта), идентифицированную запрашиваемым URI. Если запрашиваемый URI обращается к процессу, производящему данные, то в качестве объекта ответа должны быть возвращены произведенные данные, а не исходный текст процесса, если сам процесс не выводит исходный текст.

Запросы с методом PUT, которые содержат объект, сохраняются под запрашиваемым URI. Если URI обращается к существующему ресурсу, включенный объект следует рассматривать как модифицированную версию объекта, находящегося на первоначальном сервере. Если URI не указывает на существующий ресурс, и может интерпретироваться агентом пользователя как новый ресурс для запросов, первоначальный сервер может создать ресурс с данным URI.

Метод DELETE запрашивает первоначальный сервер об удалении ресурса, идентифицируемого запрашиваемым URI.

При разработке архитектуры клиент-серверного приложения очень важно правильно разделить работу между клиентской и серверной частью приложения. Можно сделать клиент «тонким», только отображающим результаты запроса. Это удобно для организации клиента, его можно разместить на простом телефоне с низкими системными характеристиками. Таким клиенту не нужно сложного программного обеспечения. Но тогда сервер становится «толстым», на него ложится большая нагрузка, ему приходится выполнять все запросы к источнику данных и всю обработку этих данных.

Можно, наоборот, сделать клиент «толстым», выполняющим всю обработку результатов запроса, а сервер «тонким», только рассылающим необработанные данные клиентам. В этом случае для клиента требуется мощный дорогостоящий смартфон.

Для разработанного приложения была выбрана архитектура с «тонким» клиентом и «толстым» сервером, что позволит охватить максимальный круг пользователей, за счет низких технических требований к аппаратуре пользователей.

На рисунке 9 представлена архитектура разработанного приложения.

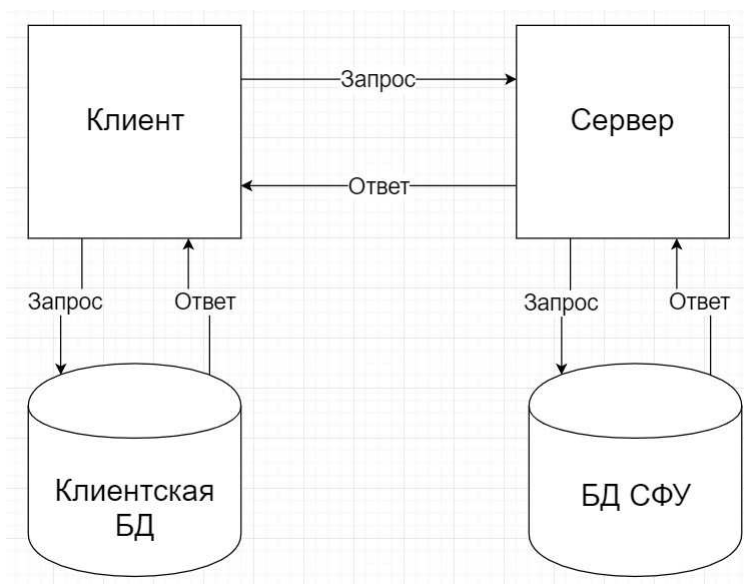


Рисунок 9 – Архитектура «клиент-сервер» приложения «Я в СФУ»

Клиент представляет собой мобильное приложение на базе Android OS на языке программирования Java. Посредством использования библиотеки Retrofit 2, которая упрощает взаимодействие с REST API сервера. Клиент отправляет запрос к серверу, разработанному на Node.js. Сервер, следуя командам клиента, получает необходимую информацию с сервисов СФУ, обрабатывает ее, приводя к необходимому виду и возвращает информацию на клиент, таким образом, сервер берет на себя основную рабочую нагрузку, тем самым реализуя архитектуру с «тонким» клиентом и «толстым» сервером.

Клиент распоряжается информацией согласно определенным сценариям, либо сохраняя ее в локальную базу данных под управлением Realm, либо не хранит, загружая ее каждый раз с сервера по запросу пользователя.

Модуль «Абитуриент» является частью приложения «Я в СФУ» и его архитектура является идентичной.



## 2.2 Структура базы данных

В проекте косвенно используется база данных СФУ для получения информации, посредством парсинга страниц сервисов СФУ, и нативная NoSQL клиентская база данных Realm для хранения информации на устройстве пользователя.

Realm – это объектно-ориентированная кросс-платформенная база данных, которая сохраняет данные локально на устройстве. Realm доступна для основных мобильных языков, таких как Swift и Objective-C (iOS), Java (Android), C# (Xamarin, .NET) и JavaScript (React Native и Node.js). База данных Realm является легкой и высокопроизводительной, способной обрабатывать очень большие нагрузки данных и выполнять запросы за доли секунды. Приложение, использующее realm, может обновлять данные так быстро, как это необходимо, чтобы обеспечить приятный, привлекательный пользовательский интерфейс. Это также идеальный компаньон для написания реактивных приложений из-за природы «live object» базы данных [18].

Главными особенностями базы данных Realm является ее быстрдействие и хорошая программная оболочка для обращения к ней. Также в Realm присутствуют такие особенности Realm объектов, как:

- получение объектов из базы очень быстрое, десериализации как таковой нет, чтение с диска происходит только при обращении к конкретному полю;
- существует требование делать все поля приватными и обращаться через геттеры;
- метод `copyFromRealm()` – позволяет получать отвязанные, полностью собранные объекты, как обычная ORM;
- в `debugger` все поля будут `null`. Для получения какого-либо значения нам нужно обращаться через геттер;
- изменения распространяются моментально в рамках одного потока;

- фильтрация объектов осуществляется по полям, названия полей указываются в виде строки, без использования SQL;

- так же таблицы в базе не связаны индексами и, в случае необходимости, связать таблицы, связь осуществляется по сущностям.

На рисунке 10 представлены таблицы и их связи, используемые в приложении «Я в СФУ».

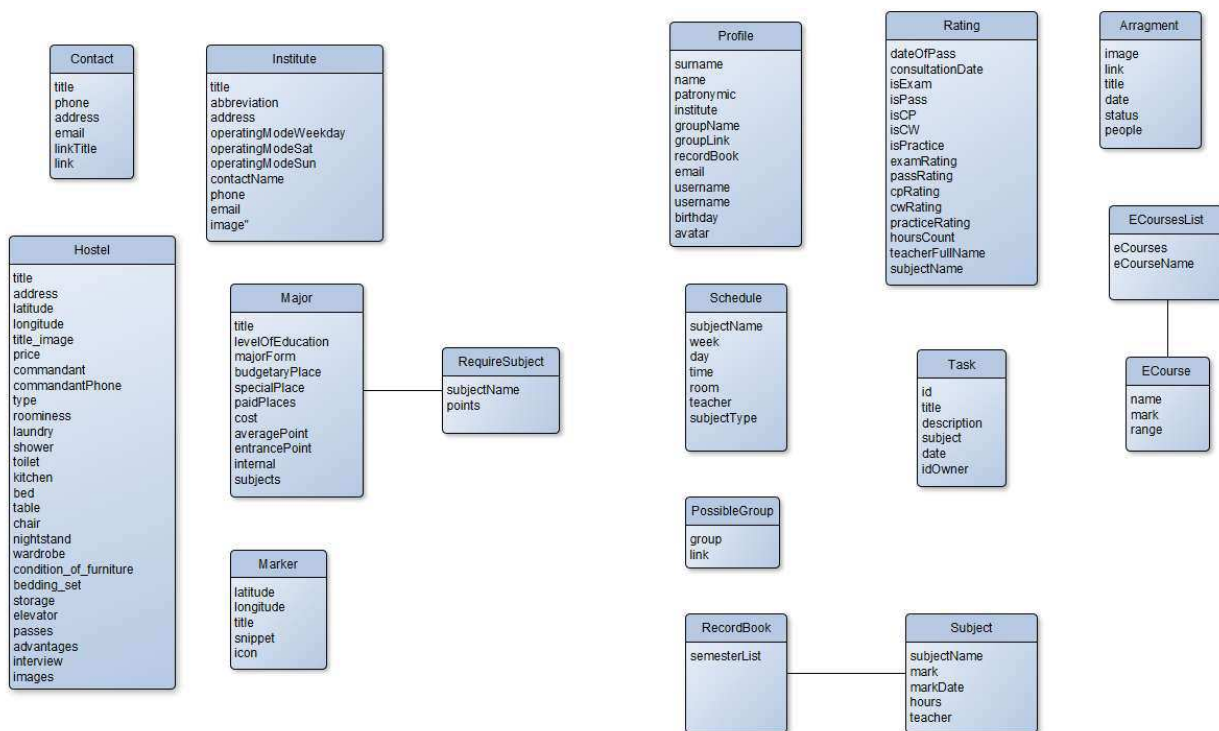


Рисунок 10 – Схема локальной базы данных

В левой части рисунка 10 представлены таблицы, отвечающие за хранение информации раздела «Абитуриент», это такие таблицы, как:

- Contact, которая отвечает за хранения информации о контактах СФУ;
- Institute, которая содержит описание институтов ВУЗа;
- Hostel, которая содержит в себе информацию об общежитиях;
- Major, которая содержит перечень информации о направлениях подготовки СФУ, она связана с таблицей RequireSubject по сущности subjects;

- RequireSubject, данная таблица содержит перечень предметов ЕГЭ, которые необходимы для поступления на интересующее направление подготовки;

- Marker, содержит сведения о местоположении объектов СФУ.

Наличие локальной базы данных обеспечивает быстрый доступ к данным в случае необходимости, также экономит мобильный трафик, так как скачивание данных на устройство происходит один раз и далее данные подгружаются из локальной базы данных на устройстве пользователя.

### **2.3 Выводы**

В результате работы была спроектирована архитектура «клиент-сервер» мобильного приложения под Android OS. Для обмена данными между сервером и клиентом архитектура была выстроена в стиле REST. Обмен данными предусматривает JSON формат. Для взаимодействия с REST использовалась библиотека Retrofit 2, что позволяет убрать часть рутинной работы при взаимодействии с HTTP запросами.

На сервере вся информация хранится в виде текстовых документов в формате JSON. По причине отсутствия доступа API к сайтам СФУ и структурных подразделений, получение информации происходит путем парсинга необходимых web-страниц.

Часть данных хранится исключительно на сервере и подгружается динамически в процессе работы приложения, например, раздел «Новости», требующий частого обновления актуальной информации по запросу пользователя. Более часто используемые данные хранятся на устройстве. В проекте используется локальная NoSQL база данных Realm, модуль «Абитуриент» включает в себя 6 таблиц.

## 3 Описание работы программного продукта

### 3.1 Серверная часть приложения

Как было сказано ранее в пункте 1.3.2, для организации передачи данных используется архитектура REST. В модуле «Абитуриент» для передачи данных между сервером и клиентом используются следующие POST запросы:

- getInstitutes, этот запрос отправляет на клиент список институтов;
- getContacts, данный запрос отвечает за отправку списка контактов;
- getHostels, отправляет список общежитий;
- getMarkers, отправляет перечень меток объектов кампуса СФУ;
- getMajors, запрос отправляет список направлений подготовки;
- getNews, запрос, отвечающий за отправку новостей СФУ.

На сервере находятся следующие JSON файлы, содержащие в себе необходимую информацию для отправки на клиент:

- contacts.json, в котором содержится информация о контактных лицах;
- hostels.json, в нем содержится информация по общежитиям;
- institutes.json, в котором содержатся данные по институтам ВУЗа;
- majors.json, содержит перечень направлений подготовки;
- markers.json с метками объектов кампуса СФУ.

Получение информации с новостного сайта СФУ [19] происходит путем его парсинга (web scraping) с помощью библиотеки Cheerio. Парсер собирает данные со страницы новостей, структурирует их и отправляет на клиент в соответствии с моделью данных. Функция, отвечающая за парсинг страницы новостей, изображена на рисунке 11. Алгоритм работы парсинга простой, сначала необходимо прогрузить нужную страницу по ее URL адресу, далее получить все новостные статьи по классу «news-items» тега «article». Далее из каждой статьи необходимо получить ссылку, категорию, дату размещения, заголовок, описание и URL изображения новости.

```

function getSFUNews(res) {
  let URL = 'http://news.sfu-kras.ru';
  let news = [];

  request(URL, function (error, response, body) {
    if (!error) {
      let $ = cheerio.load(body, {decodeEntities: false});
      teasers = $(".news-items article");

      teasers.each(function (i) {
        let article = new News();
        article.setLink = "http://news.sfu-kras.ru" + $("#news > div > article:nth-child("
          + (i + 1) + ") > a").attr("href");
        article.setCategory = $("#news > div > article:nth-child(" + (i + 1) +
          ") > a > span.news-item-category").text();
        article.setDate = $("#news > div > article:nth-child(" + (i + 1) + ") > a > time").text();
        article.setTitle = $("#news > div > article:nth-child(" + (i + 1) + ") > a > h3").text();
        article.setDescription = $("#news > div > article:nth-child(" + (i + 1) + ") > div p").text();
        article.setImageUrl = "http://news.sfu-kras.ru" + $("#news > div > article:nth-child("
          + (i + 1) + ") > a > span.news-item-image > img").attr("src");

        news.push(article);
      })
    }
    res.status(200)
      .json({
        news
      });
  });
}

```

Рисунок 11 – Функция парсинга сайта новостей

Для того, чтобы структурировать все данные, полученные при парсинге и из файлов JSON, а также использовать библиотеку для сетевого взаимодействия Retrofit 2 при их получении на клиенте, нужно создать модель данных для всех HTTP запросов, такая же модель данных должны быть представлена в функциональной части на клиенте приложения. Пример модели данных представлен на рисунке 12, где представлена модель данных новостей. Остальные модели данных имеют схожую структуру.

```

class News {

    constructor(link, title, imageUrl, category, date, description) {
        this.setLink = link;
        this.setTitle = title;
        this.setImageUrl = imageUrl;
        this.setCategory = category;
        this.setDate = date;
        this.setDescription = description;
    }

    set setLink(newValue) {
        this.link = newValue;
    }

    get getLink() {
        return this.link
    }
    set setTitle(newValue) {
        this.title = newValue;
    }
    set setImageUrl(newValue) {
        this.imageUrl = newValue;
    }
    set setCategory(newValue) {
        this.category = newValue;
    }
    set setDate(newValue) {
        this.date = newValue;
    }
    set setDescription(newValue) {
        this.description = newValue;
    }
}

module.exports = News;

```

Рисунок 12 – Модель данных новостей

Так, при использовании Retrofit 2, снижается часть рутинной работы по обеспечению клиент серверного взаимодействия приложения. Сервер берет на себя всю работу по обработке данных и отправляет на клиент «чистые» данные, где они могут сразу же использоваться. Таким образом, мы делаем сервер «толстым», а клиент «тонким», что позволяет устанавливать и использовать приложения на телефонах с минимальными заявленными характеристиками, за счет чего становится возможным обеспечить максимальный охват пользовательской клиентуры.

## 3.2 Клиентская часть приложения

### 3.2.1 Выбор типа пользователя и справочное меню

При первом запуске приложения необходимо будет выбрать тип пользователя (рисунок 13а). Далее, при каждом запуске будет открываться раздел необходимый конкретному пользователю, при необходимости раздел по умолчанию можно сменить.

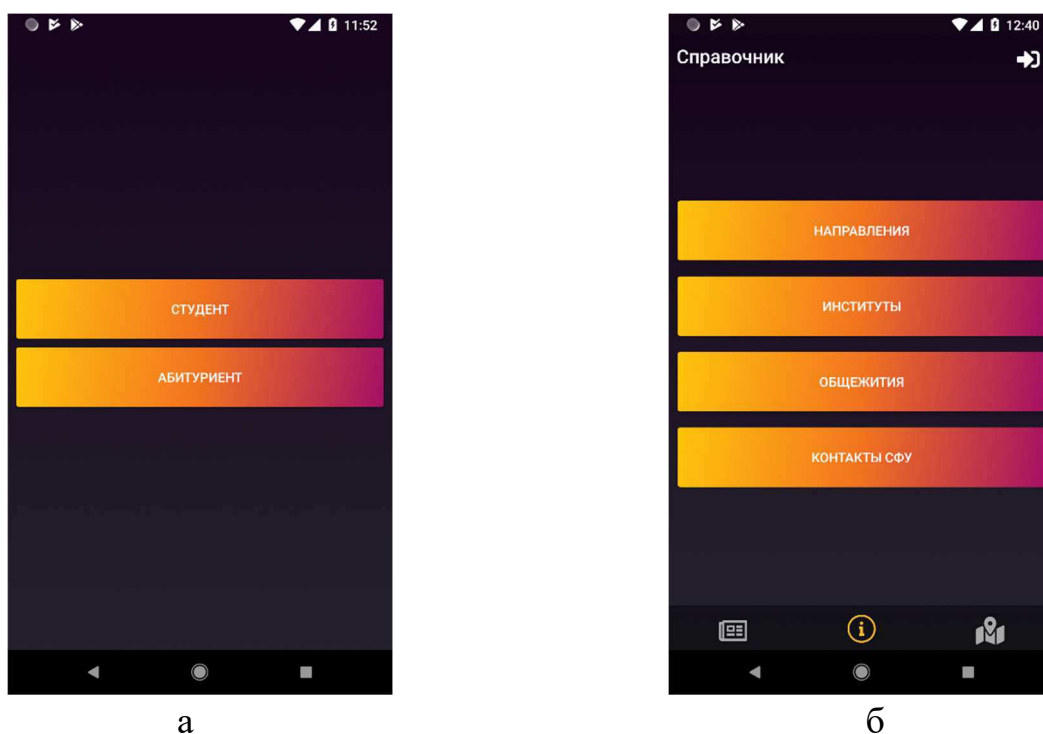


Рисунок 13 – Интерфейс приложения: а) выбор типа пользователя; б) меню справочника абитуриента

После выбора раздела открывается меню справочника абитуриента (рисунок 13б). В верхнем правой углу экрана находится кнопка выхода из текущего раздела. По центру экрана располагаются такие пункты, как:

- направления;
- институты;
- общежития;

- контакты.

В нижней левой части экрана находится кнопка перехода на экран новостей, в нижней правой – кнопка перехода на карту кампуса.

### 3.2.2 Направления подготовки

В приложении реализован фильтр по направлениям подготовки. Данная возможность позволяет абитуриентам ознакомиться с существующими в ВУЗе направлениями подготовки и подобрать нужное в соответствии со своими требованиями и возможностями. Реализованные фильтры представлены на рисунке 14.

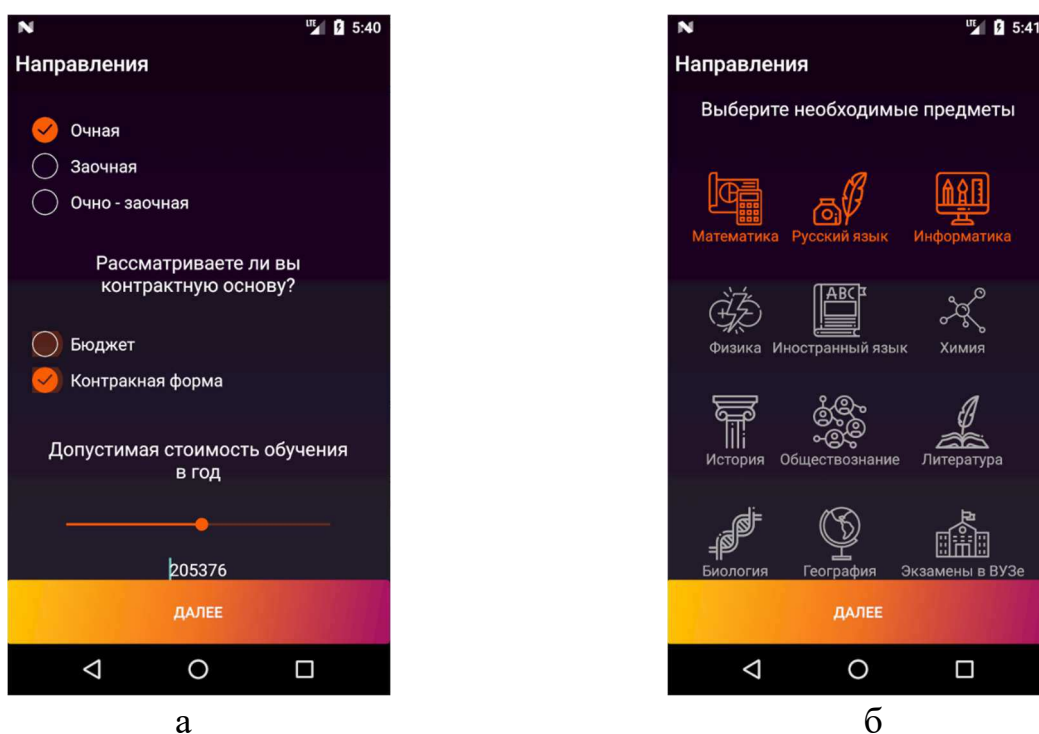


Рисунок 14 – Интерфейс приложения: а) выбор формы и стоимости обучения; б) выбор экзаменационных предметов

Инструменты, представленные на рисунке 14, позволяют применить следующие фильтры:

- выбор формы обучения (очная, заочная, очно-заочная);
- основа обучения (бюджет, контрактная форма);



- максимально допустимая стоимость обучения (при контрактной форме обучения);

- выбор вступительных экзаменов.

Далее, после применения фильтров, на экран выводится перечень соответствующих им направлений подготовки (рисунок 15).

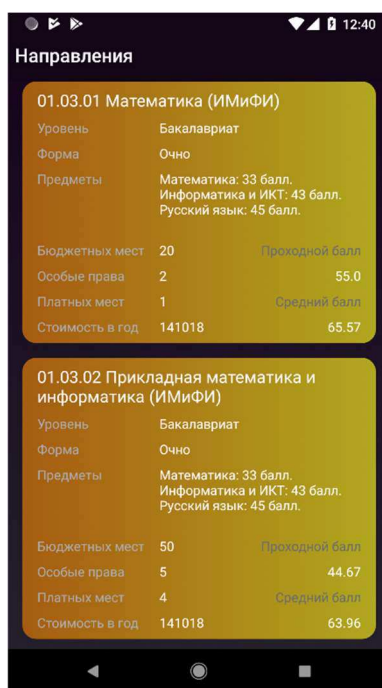


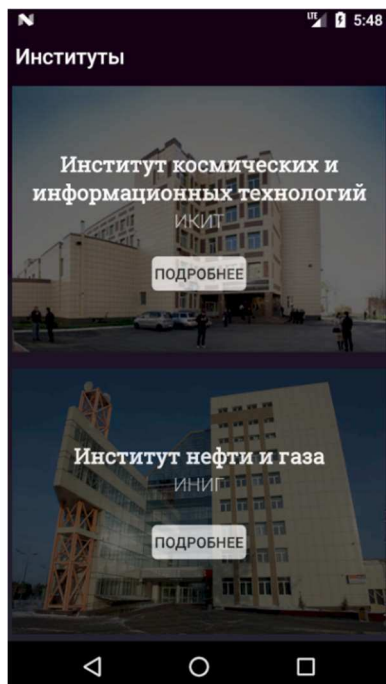
Рисунок 15 – Информация о направлениях подготовки

### 3.2.3 Институты

В приложении также представлена краткая информация по институтам Сибирского федерального университета. На момент первой половины 2019 года в состав университета входит 21 институт на территории Красноярска и 3 филиала за его пределами. Это следующая информация:

- наименование;
- адрес;
- режим работы;
- контакты.

Реализация представлена на рисунке 16.



а



б

Рисунок 16 – Интерфейс приложения: а) список институтов; б) информация об институте

### 3.2.4 Общежития

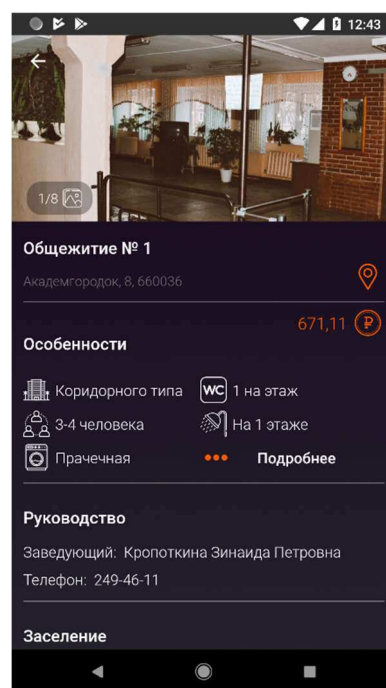
Систематизированный раздел общежитий является уникальным. Это единственное приложение, где собрана и обработана информация по общежитиям СФУ из различных источников и в интересах пользователя, выявленных путем анкетирования аудитории потенциальных абитуриентов. На рисунке 17а приведен полный перечень общежитий, всего их на первую половину 2019 года построено и эксплуатируется в количестве 30 объектов.

На рисунках 17б и 18а изображен экран информации о выбранном общежитии, эта информация включает в себя карусель изображений, соответствующих фотографиям выбранного общежития, номер общежития, адрес с возможностью отображения его на карте, пункт «Особенности», включающий в себя краткое описание санитарно-технического состояния помещений, при нажатии на кнопку «Подробнее» можно увидеть полный

перечень особенностей общежития (рисунок 18б), ниже указаны руководство общежития и правила заселения.

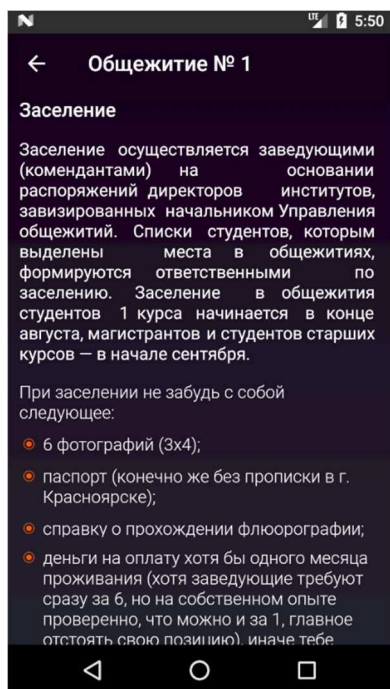


а

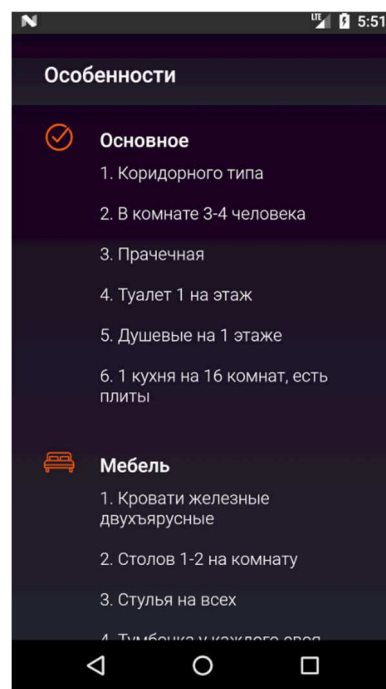


б

Рисунок 17 – Интерфейс приложения: а) список общежитий; б) информация о выбранном общежитии



а



б

Рисунок 18 – Интерфейс приложения: а) информация о выбранном общежитии; б) особенности выбранного общежития

При нажатии на изображение в карусели, оно открывается в отдельном окне, где его можно масштабировать (рисунок 19а). Перелистывание слайдов карусели осуществляется с анимацией перехода, изображенной на рисунке 19б.



а



б

Рисунок 19 – Интерфейс приложения: а) масштабирование карусели; б) перелистывание слайдов карусели

### 3.2.5 Контакты

Абитуриенту бывает нужно связаться с некоторыми отделами ВУЗа для решения тех или иных вопросов, например, с приемной комиссией по вопросам поступления, абитуриенту в силу его неопытности может быть сложно найти необходимые контакты на сайте университета, поэтому является рациональным разместить данную информацию в мобильном приложении. На рисунке 20 изображен интерфейс перечня контактов, туда входят такие пункты, как:

- «Контакты, реквизиты»;
- «Общий отдел (канцелярия)»;
- «Приемная комиссия»;

- «Приемная ректора»;
- «Пресс-служба».

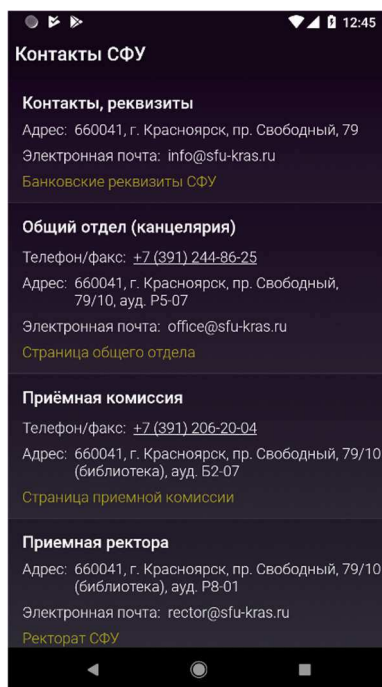
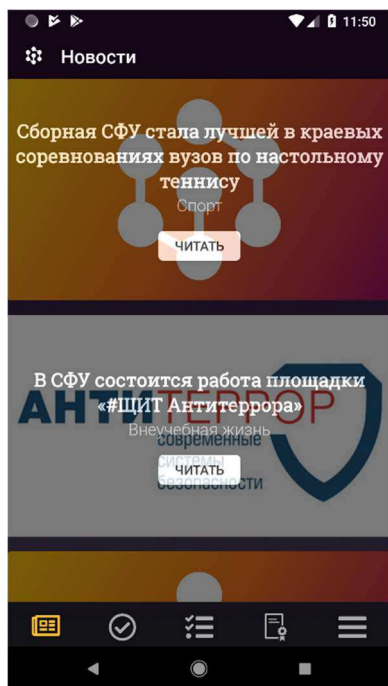


Рисунок 20 – Информация о контактах

Каждый из пунктов, по мере наличия, содержит такую информацию, как адрес, телефон/факс, электронная почта и ссылка на web-страницу.

### 3.2.6 Новости

При переходе на страницу новостей сразу подгружается и отражается список новостей ВУЗа (рисунок 21а). При нажатии на кнопку «Читать» открывается отдельный экран с краткой аннотацией новости и с возможностью перехода на полную страницу новости (рисунок 21б).



а



б

Рисунок 21 – Интерфейс приложения; а) список новостей; б) выбранная новость

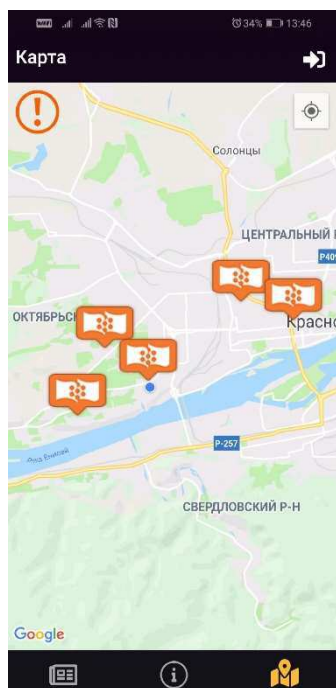
### 3.2.7 Карта кампуса

При открытии карты кампуса открывается карта с отдаленным масштабом с метками пяти площадок кампуса СФУ (рисунок 22а).

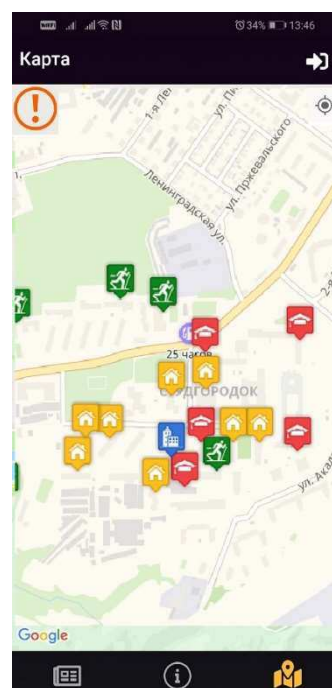
При увеличении масштаба (приближении) карты, путем двойного нажатия или «обратного» щипкового жеста, метка с площадкой «разбивается» на множество объектов этой площадки (рисунок 22б). Также масштабирование возможно при нажатии на метке площадки, тогда происходит приближении камеры также с разбиением на более мелкие объекты и с централизацией карты на выбранной площадке.

При уменьшении масштаба (отдалении) карты объекты площадок «схлопываются» в одну метку площадки.





а



б

Рисунок 22 – Интерфейс приложения: а) карта кампуса с отдаленным масштабом; б) карта кампуса с приближенным масштабом

Обозначение меток карты кампуса представлена на рисунке 23.

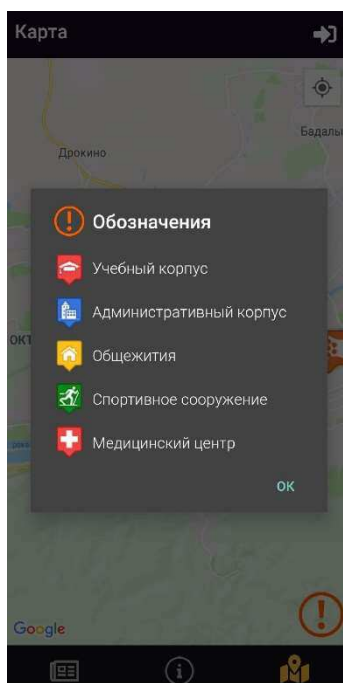


Рисунок 23 – Обозначение меток карты кампуса

### 3.3 Выводы

Модуль был разработан в соответствии с архитектурой «клиент-сервер». В зависимости от той или иной задачи сервер либо хранит данные, либо получает их с официальных сервисов СФУ. Запросы на получение данных реализованы методом POST HTTP-протокола. По требованию пользователя клиент отправляет запрос на сервер. В ответ приходит текстовый файл формата JSON, который библиотекой Retrofit разбирается на составные части согласно модели данных. В таком виде данные используются согласно поставленной клиенту задаче.

Интерфейс приложения «Я в СФУ» выполнен в стиле Material Design. Material Design держится на четырех основных принципах [20]:

- тактильные поверхности. В Material Design интерфейс складывается из осязаемых слоёв так называемой «цифровой бумаги»;

- полиграфический дизайн. Если считать слои частями «цифровой бумаги», то в том, что касается «цифровых чернил», используется подход из традиционного графического дизайна;

- осмысленная анимация. В Material Design необходимо с помощью анимации в слоях и в «цифровых чернилах» давать пользователям подсказки о работе интерфейса;

- адаптивный дизайн в работе на устройствах с разным расширением.

Главным в этой концепции является то, что свойства и поведение объектов на экране являются такими же предсказуемыми и понятными человеку, как и любое простое событие в физическом мире.

Всего в результате работы было спроектировано и реализовано 15 окон модуля «Абитуриент». Дизайн раздела «Абитуриент» соответствует дизайну всего приложения «Я в СФУ» и гармонично совмещает в себе корпоративные цвета СФУ и современный темный фон.



## 4 Тестирование программного продукта

### 4.1 Тестирование удобства использования

Тестирование удобства пользования (Usability testing) [21] – это не что иное, как тестирование дружелюбности приложения для пользователя. При тестировании удобства использования проверяется, легко ли новому пользователю разобраться в приложении. В общем, при тестировании удобства использования тестируется системная навигация.

Тест удобства использования удостоверяется в простоте и эффективности использования продукта при использовании стандартных практик тестирования удобства использования.

Сценарии тестирования удобства использования:

- содержание окон мобильного приложения верное, без грамматических и орфографических ошибок;
- все шрифты соответствуют требованиям;
- оптимальность выбранной цветовой палитры;
- все тексты правильно выравнены;
- все поля правильно выравнены;
- между полями, колонками, рядами и сообщениями об ошибках оставлено достаточно свободного места;
- все кнопки должны иметь стандартный формат и размер;
- неактивные поля должны быть серыми;
- адаптация приложения при разных разрешениях экрана;
- панель скролла должна появляться только тогда, когда она требуется;
- заголовок должен отображаться на каждой странице;
- данные в выпадающих списках не обрезаются из-за размеров поля.

Была собрана фокус-группа независимых пользователей, состоящая из 50 человек, каждый из которых дал свои оценки по вышеупомянутым пунктам

по десятибалльной шкале, чтоб более точно проанализировать то, как пользователи оценивают продукт. Средние значения результатов тестирования модуля «Абитуриент» приведены в таблице 1.

Таблица 1 – Средние значения результатов тестирования

	Цвета	Шрифты	Предупреждающие сообщения	Ссылки и изображения	Заголовки	Удобство кнопок	Удобство полей и форм	Навигация
Среднее значение	8,25	7,75	7,625	7,875	8,25	8,5	7,75	8,125

Исходя из средних значений для каждого пункта тестирования, можно сделать вывод, что для пользователей мобильное приложение является достаточно удобным в использовании.

Также были проведены проверки контента мобильного приложения на наличие грамматических и орфографических ошибок, информативность, структурированность и логическую связанность информации.

В результате тестирования и доработки приложения, были устранены все найденные грамматические и орфографические ошибки. Данные структурированы и логически связаны для понимания пользователем, страницы с информацией в достаточной мере информативны.

Таким образом, по результатам работы можно сделать вывод, что тестирование на удобство пройдено успешно.

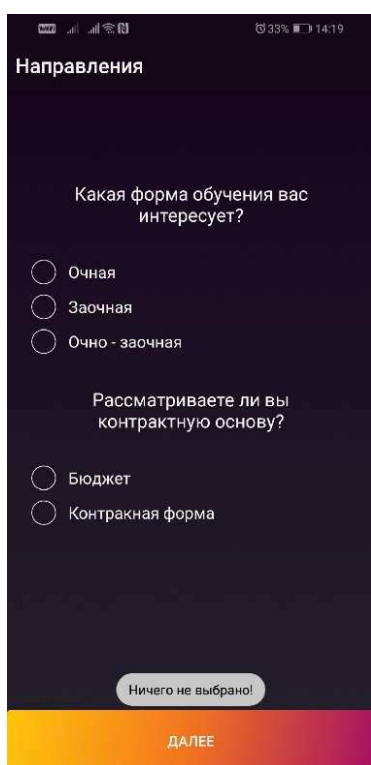
## 4.2 Функциональное тестирование

Функциональное тестирование мобильных приложений охватывает тестирование взаимодействия с пользователем, тестирование транзакций, а также проверку продукта на соответствие функциональной спецификации.

Одним из важнейших пунктов функционального тестирования, является проверка полей ввода информации и наличие информационных сообщений при ошибках или некорректного поведения пользователя.

В модуле «Абитуриент» есть несколько элементов отвечающих за отправку данных на сервер, они используются при настройке параметров фильтрации.

При подборе направления подготовки пользователю необходимо настроить фильтры для нахождения наиболее подходящих результатов. В том случае, если пользователь не выберет ни один фильтр, то он получит следующее предупреждение, представленное на рисунке 24а.



а



б

Рисунок 24 – Интерфейс приложения: а) предупреждение пользователя; б) выбор максимального количества предметов

Далее необходимо выбрать перечень предметов. Максимальное количество предметов для выбора 3, по достижению этого количества остальные «невыбранные» предметы становятся неактивными, что выражается в их сером цвете (рисунок 24б).

В случае если направлений, соответствующих установленным критериям найдено не будет, пользователь увидит информационное сообщение с предложением вернуться к настройке параметров фильтрации (рисунок 25).

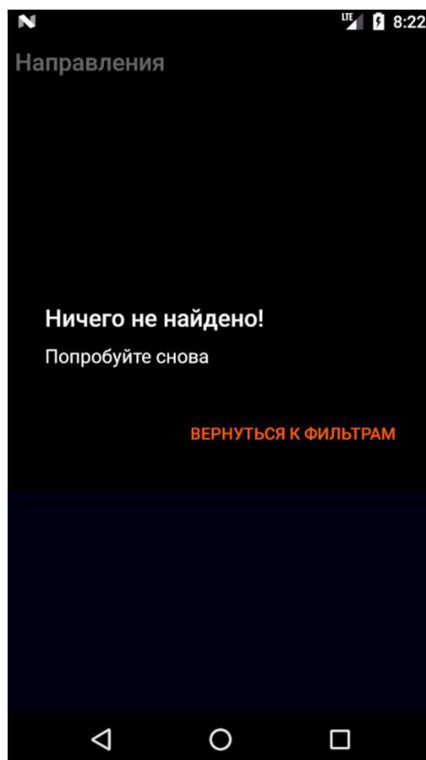


Рисунок 25 – Настройка параметров фильтрации.

Работа приложения в момент запуска/завершения работы, полностью соответствует спецификациям и работает корректно. Также была проверена работа приложения при влиянии внешних событий. В случае входящего звонка приложение без сбоев переходит в фоновый режим. При завершении звонка работа приложения восстанавливается в нормальном режиме в состоянии до момента совершения звонка. Аналогично не выявлено проблем при принятии и отправлении SMS-сообщений с панели быстрого доступа.

Сценарии работы прокрутки страницы адекватны, вспомогательный элемент скролл для прокрутки страницы становится видимым только в случае необходимости.

В модуле «Абитуриент» корректно работает навигация между ключевыми реализованными функциями «Новости», «Меню» и «Карты», между которыми

организован быстрый переход, а также между остальными пунктами функции «Меню».

Была также протестированная установка приложения на устройства, системные характеристики которых соответствовали заявленным спецификациям, установка проходит без системных сбоев в нормальном режиме.

Весь функционал, заявленный в спецификациях, выполняется корректно, не вызывая системных сбоев.

### **4.3 Тестирование совместимости**

Тестирование совместимости [21] используется, чтобы убедиться, что приложение совместимо с другими элементами системы, в которой оно работает, например, операционными системами или аппаратным обеспечением смартфона.

Цель тестирования совместимости – оценка того, насколько хорошо программный продукт работает под определенной версией Android OS, с другим ПО или разными системными требованиями.

Сценарии тестирования совместимости:

- убедиться, что приложение достаточно адаптировано под разные разрешения экрана, предусмотренных ТЗ;

- убедиться, что картинки корректно отображаются на разных разрешениях экрана, предусмотренных ТЗ;

- убедиться, что шрифты верно отображаются на разных разрешениях экрана, предусмотренных ТЗ;

- убедитесь, что приложение корректно работает на разных версиях Android OS, предусмотренных ТЗ.

В таблице 2 представлены результаты тестирования различных элементов на Android OS под перечисленные выше требования.

Таблица 2 – Результаты тестирования совместимости

Версия Android OS	Lollipop (5.0)	Marshmallow (6.0)	Nougat (7.0)	Oreo (8.1)	Pie (9.0)
Выбор типа пользователя	корректно	корректно	корректно	корректно	корректно
Выбор пункта меню	корректно	корректно	корректно	корректно	корректно
Просмотр новостей	корректно	корректно	корректно	корректно	корректно
Использование карты кампуса	корректно	корректно	корректно	корректно	корректно
Работа с направлениями подготовки	корректно	корректно	корректно	корректно	корректно
Просмотр институтов	корректно	корректно	корректно	корректно	корректно
Просмотр общежитий	корректно	корректно	корректно	корректно	корректно
Просмотр контактов	корректно	корректно	корректно	корректно	корректно

Долгое время производители считали соотношение сторон 16:9 единственно приемлемым, но с увеличением длины диагонали, начиная от 5.5 дюйма, устройство с соотношением 16:9 кажется громоздким, поэтому в 2017 [22] появились первые смартфоны с соотношением экрана 18:9, которые продолжают набирать популярность благодаря своей эргономичности. В таблице 3 показаны результаты тестирования на экранах с различным соотношением сторон.

Таблица 3 – Результаты тестирования отображения

Соотношение сторон экрана	16:9	18:9
Переходы между страницами	корректно	корректно
Отображения надписей, кнопок, полей	корректно	корректно
Корректные расстояния между элементами	корректно	корректно
Верное отображение текста	корректно	корректно

Таким образом, по результатам тестирования на совместимость приложения для различных соотношений сторон экрана и разных версий Android OS можно сделать вывод, что мобильное приложение корректно

отображается, а все функции выполняются верно, независимо от того, с каким соотношением сторон экранов необходимо иметь дело.

#### **4.4 Выводы**

В процессе проверки качества программного продукта было проведено несколько видов тестирования, а именно:

- тестирование удобства пользования;
- функциональное тестирование;
- тестирование совместимости.

В процессе тестирования удобства пользования проверялась простота и эффективность использования продукта. В результате тестирования удобства пользования выявлено, что для пользователей разработанный модуль является удобным в использовании. При проверке контента модуля на наличие грамматических и орфографических ошибок, информативность, структурированность и логическую связанность информации, все найденные замечания были устранены.

Функциональное тестирование подразумевало проверку на соответствие заявленному функционалу и наличие информативных сообщений при внутренних ошибках приложения или некорректного поведения пользователя.

Весь функционал, заявленный в спецификации, выполняется корректно, не вызывая системных сбоев. Так же приложение корректно восстанавливает свою работу после влияния внешних событий. Все предусмотренные спецификациями информационные сообщения работают и отображаются корректно.

Тестирование совместимости призвано проверить корректность работы приложения в определенных разновидностях аппаратного и программного окружения, предусмотренных спецификациями. В данной работе предусматривается поддержка минимальной версии платформы Android OS не ниже 5, что соответствует версии API от 21 выше. Также необходимо было

строить адаптивный интерфейс приложения для корректного отображения устройств как с соотношением 16:9, так и с широкоформатным разрешением 18:9.

В ходе проверки поддержки разных версий Android OS были проверены следующие версии: Lollipop (5.0), Marshmallow (6.0), Nougat (7.0), Oreo (8.1), Pie (9.0). По всем пунктам проверки тестирование прошло успешно. Тестирование адаптивности интерфейса под различные расширения и соотношения экрана также прошло успешно, интерфейс является адаптивным и динамически подстраивающийся под заданные размеры экрана смартфона.

По полученным в ходе анализа данным можно сделать вывод о том, что разработанный модуль «Абитуриент» приложения «Я в СФУ» отвечает всем требованиям, которые были предъявлены в начале разработки, более детально описанным в пунктах 1.1 и 1.2.

В ходе проверки взаимодействия модулей системы между собой, а также их интеграции в одну общую систему было выявлено, что совокупность всех разработанных и внедренных в приложение модулей работает стабильно.



## ЗАКЛЮЧЕНИЕ

По результатам работы, был реализован модуль «Абитуриент», и совместно с усилиями остальных разработчиков было получено стабильно работающее клиент-серверное мобильное приложение «Я в СФУ», состоящее из трех модулей и объединяющее функционал для студента и абитуриента в одно информационное пространство, что делает его уникальным в подобного рода приложениях. Другими словами, все поставленные задачи были решены и цель достигнута.

В данном приложении заинтересованы как довузовское управление Сибирского федерального университета, так и учебный отдел. В настоящее время идут переговоры о дальнейшем развитии и интеграции приложения в информационное пространство СФУ.

В дальнейшем в модуле «Абитуриент» планируются некоторые улучшения, в частности, сбор информации в виде интервью обучающихся каждого направления подготовки для предоставления абитуриентам более расширенной информации. Также положительное решение о сотрудничестве с СФУ даст доступ к сервису АИС Абитуриент, что станет важным дополнением функционала модуля.

В масштабах приложения «Я в СФУ» планируется проектирование и реализация третьего раздела «Преподаватель», который будет включать в себя функционал, полезный для преподавателей, например, реестр научных трудов преподавателя с возможностью подачи его в различные конкурсы/гранты, расписание преподавателя и другой необходимый функционал. Также планируется разработать приложение для системы iOS и внедрить возможность персонализации интерфейса приложения.

Результатом выполнения ВКР было также получение свидетельства о государственной регистрации программы для ЭВМ №2019613682 от 21.03.2019 г., представленное в приложении А.

## СПИСОК СОКРАЩЕНИЙ

- АИС – Автоматизированная информационная система
- БД – База данных
- ВУЗ – Высшее учебное заведение
- ЕГЭ – Единый государственный экзамен
- ПО – программное обеспечение
- СФУ – Сибирский федеральный университет
- ТЗ – Техническое задание
- API – Application Programming Interface
- JSON – JavaScript Object Notation
- HTML – HyperText Markup Language (язык гипертекстовой разметки)
- HTTP – HyperText Transfer Protocol (протокол передачи гипертекста)
- ORM – Object-Relational Mapping (объектно-реляционное отображение)
- OS – Operating System (операционная система)
- REST – Representational state transfer (передача состояния представления)
- SGML – Standard Generalized Markup Language (стандартный обобщённый язык разметки)
- URL – Uniform Resource Locator (единый указатель ресурса)
- WWW – World Wide Web (всемирная паутина)
- XML – eXtensible Markup Language (расширяемый язык разметки)

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вигерс, К. И. Разработка требований к программному обеспечению.: Пер. с англ. / К. И. Вигерс. – М.: Издательско-торговый дом «Русская Редакция», 2004. – 576 с.
2. Леффингуэлл, Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход.: Пер. с англ. / Д. Леффингуэлл, Д. Уидриг. – М.: Издательский дом «Вильямс», 2002. – 448с.
3. Desktop vs Mobile vs Tablet Market Share Worldwide [Электронный ресурс]: StatCounter предоставляет статистические данные по выборке, превышающей 10 миллиардов просмотров страниц в месяц, собранных со всей сети StatCounter из более чем 2 миллионов веб-сайтов. // StatCounter GlobalStats. – Режим доступа: <http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/2018>
4. Streaming delays mentally taxing for smartphone users: Ericsson Mobility Report [Электронный ресурс]: Пресс-релиз от 17.02.2016. // Ericsson. – Режим доступа: <https://www.ericsson.com/en/press-releases/2016/2/streaming-delays-mentally-taxing-for-smartphone-users-ericsson-mobility-report>
5. Mobile Operating System Market Share Russian Federation [Электронный ресурс]: StatCounter предоставляет статистические данные по выборке, превышающей 10 миллиардов просмотров страниц в месяц, собранных со всей сети StatCounter из более чем 2 миллионов веб-сайтов. // StatCounter GlobalStats. – Режим доступа: <http://gs.statcounter.com/os-market-share/mobile/russian-federation>
6. Distribution dashboard [Электронный ресурс]: Сайт посвященный Android Platform. // Android Developers. – Режим доступа: <https://developer.android.com/about/dashboards>
7. IntelliJ IDEA [Электронный ресурс]: Продукты и разработки. // Jet Brains. – Режим доступа: <https://jetbrains.ru/products/idea/>

8. WebStorm – умная IDE для JavaScript-разработчиков [Электронный ресурс]: Продукты и разработки. // Jet Brains. – Режим доступа: <https://jetbrains.ru/products/webstorm/>
9. Сол, К. XML: структурирование данных для Web: Введение [Электронный ресурс] / К. Сол // Журнал BPM World. – 1998. – Режим доступа: <https://iso.ru/ru/press-center/journal/1715.phtml>
10. Светлова, И. История языка программирования Java [Электронный ресурс] / И. Светлова // Oracle Patches. – 2018. – Режим доступа: <https://oracle-patches.com/coding/3415-история-языка-программирования-java>
11. Realm — кроссплатформенная мобильная база данных [Электронный ресурс]: Блог компании Jetrubby. // Jetrubby. – Режим доступа: <https://jetrubby.com/ru/blog/realm-mobilnaya-baza-dannyh/>
12. Браун, И. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript. / И. Браун. – СПб.: Питер, 2017. – 336 с.
13. Введение в JSON [Электронный ресурс]: ECMA-404 Стандарт обмена данными JSON. // JSON. – Режим доступа: <http://json.org/json-ru.html>
14. Абитуриент СФУ [Электронный ресурс]. Магазин приложений, игр, книг, музыки и фильмов компании Google и других компаний. // Google Play. – Режим доступа: <https://play.google.com/store/apps/details?id=ru.sfukras.abiturient>
15. Поступай в СФУ [Электронный ресурс]: Магазин приложений, игр, книг, музыки и фильмов компании Google и других компаний. // Google Play. – Режим доступа: <https://play.google.com/store/apps/details?id=com.ilyabaikalow.abiturienthelper>
16. Фландерс, Д. Введение в службы RESTful с использованием WCF [Электронный ресурс] / Д. Фландерс // MSDN Magazine. – 2009. – Режим доступа: <https://msdn.microsoft.com/ru-ru/magazine/dd315413.aspx>
17. Филдинг, Р. RFC 2068 — Протокол Передачи Гипертекста - HTTP/1.1 [Электронный ресурс] / Р. Филдинг, Дж. Геттис, Дж. Могул, Г. Фристик, Т. Бернерс-Ли // RFC – 1997. – Режим доступа: <https://rfc2.ru/2068.rfc/original>

18. What is Realm Platform? [Электронный ресурс]: Синхронизированная документация по Realm// Realm Sync Docs. – Режим доступа: <https://docs.realm.io/sync/what-is-realm-platform>

19. Новости [Электронный ресурс]: Новостной сайт СФУ// Новости СФУ. – Режим доступа: <http://news.sfu-kras.ru>

20. Material Design: на Луну и обратно [Электронный ресурс]: Русскоязычный веб-сайт в формате коллективного блога с элементами новостного сайта, созданный для публикации новостей, аналитических статей. // [habr](https://habr.com/ru/company/redmadrobot/blog/252773/). – Режим доступа: <https://habr.com/ru/company/redmadrobot/blog/252773/>

21. Web Application Testing Checklist: Example Test Cases for Website [Электронный ресурс]: Guru99 это площадка для обучения и получения нового опыта// Guru99. – Режим доступа: <https://www.guru99.com>

22. Первые в мире смартфоны с экраном 18:9 – что в них такого? [Электронный ресурс]: Новости высоких технологий hi-tech Mail.ru. // Hi-tech – Режим доступа: <https://hi-tech.mail.ru/review/lg-q6/>

## ПРИЛОЖЕНИЕ А

### Свидетельство о регистрации программы для ЭВМ



Рисунок А. 1 – Свидетельство о государственной регистрации программы для ЭВМ




Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий  
институт

«Информатика»  
кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой

  
А. С. Кузнецов  
подпись      инициалы, фамилия

« 05 » 07 2019 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.04 – Программная инженерия  
код – наименование направления

Разработка модуля «Абитуриент» для мобильного приложения «Я в СФУ»  
тема

Руководитель

  
подпись, дата


доцент, к. т. н.

должность, ученая степень

А. В. Хныкин

инициалы, фамилия

Выпускник

  
подпись, дата

М. О. Фабричкина

инициалы, фамилия

Нормоконтролер

  
подпись, дата

доцент, к. т. н.

должность, ученая степень

О. А. Литамошкин

инициалы, фамилия

Красноярск 2019