РОСЖЕЛДОР

Федеральное государственное бюджетное образовательное   
учреждение высшего образования

«Ростовский государственный университет путей сообщения»

(ФГБОУ ВО РГУПС)

Допустить к защите в ГЭК

И.о.зав. кафедрой «ВТ и АСУ»

\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В. Игнатьева

«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

**Тема**

**Разработка мобильного веб-приложения «Market place» для салонов красоты**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к выпускной квалификационной (бакалаврской) работе

АИБ 12.02.05 ПЗ

Направление подготовки «Информационные системы и технологии»,

профиль «Информационные системы и технологии на транспорте»

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Деришев

Руководитель работы

д.п.н., профессор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.И. Соколова

Нормоконтроль

ст. преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.Р. Осипова

Научный консультант

к.т.н., научный сотрудник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. А. Кулькин

2021

РОСЖЕЛДОР

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ростовский государственный университет путей сообщения»

(ФГБОУ ВО РГУПС)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра «ВТ и АСУ»

УТВЕРЖДАЮ

И.о.зав.кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В.Игнатьева

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

ЗАДАНИЕ

на выпускную квалификационную(бакалаврскую) работу

Студенту \_\_\_Деришеву Николаю Александровичу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

Группа АИБ-4-036

**1. Тема работы:** \_\_Разработка мобильного веб-приложения «Market place» для салонов красоты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утверждена приказом по университету № 95/ос от 26.01.2021 г.

Срок сдачи студентом законченной работы «9» июня 2021 г.

**2. Исходные данные к работе** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Литература по программированию на языке JavaScript\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Литература по программированию на языке Python\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Литература по проектированию информационных систем\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. Содержание расчетно-пояснительной записки (перечень вопросов, подлежащих разработке)**

1 Анализ предметной области разрабатываемого приложения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2 Обзор областей применения приложения «Market place»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3 Техническое задание\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Проектирование и конструирование веб-приложения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Расчет FP-метрик, LOC-метрик и COCOMO-метрик\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Разработка веб-приложения для автоматизации записи в салон\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7.Разработка интерфейса приложения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4. Перечень графического материала (с точным указанием названий слайдов презентации)**

1. Титульный лист\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Анализ предметной области\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Цели и задачи выпускной квалификационной работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Преимущества, особенности и области применения понятия «Маркетплейс»\_\_\_

5. Актуальность разрабатываемого мобильного веб-приложения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Разработка мобильного веб-приложения «Market place» для салонов красоты\_\_\_

7. Обоснование выбора языка и среды разработки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8. Интерфейс мобильного веб-приложения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания « 8 » февраля 2021 г.

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (инициалы, фамилия)

Задание принял к исполнению « 8 » февраля 2021 г.

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (инициалы, фамилия)

**Реферат**

Выпускная квалификационная работа содержит 87 листов пояснительной записки, включающей 41 рисунок, 22 таблицы, 11 источников и 1 приложение.

ВЕБ-ПРИЛОЖЕНИЕ, MARKETPLACE, JAVASCRIPT, PYTHON, БАЗА ДАННЫХ, POSTGRES, JSON, REACT NATIVE, ПРЕЦЕДЕНТЫ, КЛАССЫ, ИНТЕРФЕЙС, ORM, ANDROID STUDIO.

Объектом исследования является мобильная веб-разработка для автоматизации записи пользователя в салон красоты.

Цель работы – разработка мобильного веб-приложения «Market place» для салонов красоты в виде мобильного веб-приложения.

В выпускной квалификационной работе проведен анализ существующих приложений и их возможности и кратко описаны правила пользования приложением.

Приведена структура программы в виде списка модулей и характеристики каждого. Расчеты метрик позволили определить оценки по стоимости и затратам на разработку мобильного веб-приложения. На основе диаграмм прецедентов, последовательности, классов установлены отношения между [актёрами](https://ru.wikipedia.org/wiki/%D0%90%D0%BA%D1%82%D1%91%D1%80_(UML)) и [прецедентами](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82_(UML)), спроектированы и созданы классы, интерфейсы и отношения между ними.

Рассмотрены современные среды разработки, их возможности и преимущества. Выбраны оптимальные для разрабатываемого программного средства инструменты.

**Содержание**

[Введение 5](#_Toc74216808)

[1 Анализ предметной области разрабатываемого приложения 7](#_Toc74216809)

[1.1 Обзор областей применения приложения «Market place» 7](#_Toc74216810)

[1.2 Сравнительный аналитический обзор приложений «Market place» 11](#_Toc74216811)

[1.3 Техническое задание на разработку мобильного веб-приложения «Market place» для салонов красоты 15](#_Toc74216812)

[1.4 Обоснование выбора языка и среды разработки мобильного веб-приложения «Market place» для салонов красоты 17](#_Toc74216813)

[1.5 Вывод по первому разделу 20](#_Toc74216814)

[2 Проектирование мобильного веб-приложения «Market place» для салонов красоты 21](#_Toc74216815)

[2.1 Анализ требований на основе построения диаграмм прецедентов, классов, последовательности веб-приложения «Market place» для салонов красоты 21](#_Toc74216816)

[2.2 Структурные принципы проектирования мобильного веб-приложения «Market place» для салонов красоты 32](#_Toc74216817)

[2.3 Расчет функционально-ориентированных (FP), LOC-метрик, COCOMO-метрик и анализ рисков для мобильного веб-приложения «Market place» 33](#_Toc74216818)

[2.4 Вывод по второму разделу 44](#_Toc74216819)

[3 Программная реализация мобильного веб-приложения 45](#_Toc74216820)

[3.1 Взаимодействие мобильного веб-приложения на основе выбора базы данных 45](#_Toc74216821)

[3.2 Интерфейс мобильного веб-приложения «Market place» 47](#_Toc74216822)

[3.3 Руководство пользователя к мобильному веб-приложению «Market place» для салонов красоты 52](#_Toc74216823)

[3.4 Вывод по третьему разделу 59](#_Toc74216824)

[Заключение 61](#_Toc74216825)

[Список используемых источников 62](#_Toc74216826)

[Приложение А (обязательное) 63](#_Toc74216827)

[Листинг программного кода. 63](#_Toc74216828)

# **Введение**

В современном мире основные тенденции формирования электронного бизнеса базируются на развитии рынка Market place, тем самым влияя на высокий прогресс электронной коммерции в мире, что свидетельствует о высокоэффективности данной модели ведения бизнеса. Market place (далее Маркетплейс) – это самая популярная бизнес-модель в электронной коммерции, являющаяся двусторонней торговой площадкой, предоставляющей технологию и инфраструктуру для онлайн торговли.

Рассматриваемая тема выпускной квалификационной работы является актуальной, так как посвящена разработке мобильного веб-приложения «Market place» для салонов красоты. Использование модели маркетплейса в настоящее время актуально для самых разных отраслей и большинство компаний, которые становятся такими торговыми площадками, являются игроками электронной коммерции, крупными розничными торговцами, почтовыми и логистическими компаниями, финансовыми организациями и IT‑компаниями.

Целью выпускной квалификационной работы является разработка мобильного веб-приложения «Market place» для салонов красоты.

Для реализации поставленной цели необходимо выполнить следующие взаимосвязанные задачи:

1 Анализ предметной области. Осветить преимущества, особенности и области применения понятия «Маркетплейс». Провести обоснование проектных решений по видам обеспечения, в результате которых для реализации проекта были использованы средства разработки и моделирования.

2 Спроектировать мобильное веб-приложение на основе диаграмм прецедентов, классов, последовательности. Рассчитать функционально-ориентированные метрики (FP) и LOC-метрики с анализом рисков разрабатываемого мобильного веб-приложения.

3 Реализовать мобильное веб-приложение. На практике применить теоретические и практические методы системного проектирования сложных программных средств и построения моделей для создания мобильного веб-приложения «Market place» для салонов красоты.

Комплексный характер темы выпускной квалификационной работы определил необходимость использования трудов национальных и зарубежных разработчиков [4], таких как Маршал ван Альстайн – профессор, заведующий кафедрой информационных систем Бостонского университета, Джеффри Паркер – профессор Тулейнского университета и сотрудник MIT Center of Digital Business и другие, которые изучали и проектировали информационные системы маркетплейсов. Их труды были использованы при разработке и создании мобильного веб-приложения «Market place» для салонов красоты.

В первой главе был осуществлен анализ предметной области для разрабатываемого мобильного веб-приложения, рассмотрены программы-аналоги, составлено техническое задание на разработку, выполнено обоснование выбора среды и языка разработки.

Во второй главе рассмотрены основные этапы проектирования веб-приложения: моделирование системы путем построения диаграмм прецедентов, классов, последовательности, формулирование требований и расчет рисков к проектируемой информационной системе мобильного веб-приложения.

Третья глава работы содержит описание интерфейса разработанного мобильного веб-приложения «Market place» для салонов красоты и руководство пользователя к нему.

# **1 Анализ предметной области разрабатываемого приложения**

## **1.1 Обзор областей применения** **приложения «Market place»**

В настоящее время информационные технологии стали активно применяться в различных областях нашей жизни. Создание различных простых и сложных программных продуктов для разных областей торговли постоянно растет. С развитием информатизации рынок электронной коммерции претерпел существенные изменения. Наблюдается наглядный рост онлайн-торговли по прогнозам всех ведущих компаний, которые на основе разрабатываемых программ, приложений, сайтов и прочих программных продуктов стремительно развиваются. Данный факт приводит к все большому появлению маркетплейсов в электронной коммерции. Как правило, маркетплейсы стали значительными и популярными благодаря своим возможностям обеспечивать безопасные и легкие платежи между поставщиками и покупателями, и продвигать товары с предоставлением аналитической информации и юзабилити.

Согласно современным исследованиям, раньше основными участниками являлись интернет-магазины, на сегодняшний день первое место делят между собой маркетплейсы. Маркетплейсы не выступают в роли собственника товара, они не являются и не могут являться ими. Маркетплейсы предоставляют владельцам товаров, а именно производителям, дистрибьютерам, продавцам технологию и инфраструктуру для онлайн-торговли, другими словами маркетплейс – это торговая онлайн площадка.

На первый взгляд разница между маркетплейсом и крупным интернет-магазином не имеет принципиального значения. Но если провести параллель и разобрать обе площадки с точки зрения информационной системы, то интернет-магазин – это магазин одного производителя. Онлайн-альтернатива торгового центра – маркетплейс. Он, подобно гипермаркету или универмагу дает возможность арендовать продавцам пространство для торговли, а интернет-магазин предоставляет своим клиентам только услуги. Маркетплейс предоставляет совершенно разный спектр площадей для любого вида услуг, от продуктовых до косметических. Например, при продаже товаров через маркетплейс компания либо фирма получает личный кабинет на сайте, в котором будет самостоятельно размещать контент (фотографии, описание товаров, цены и т.д.), говоря другими словами получает своеобразный программный конструктор. Исходя из выше сказанного маркетплейс – это программная реализация магазина похожего на торговый центр, в которой продавцам предоставляются личные кабинеты, в которых они работают сами. Данный формат предоставления торговых услуг стал популярным во многом благодаря росту числа продавцов, желающих представлять свой товар или услугу в сети Интернет, но при этом не желающих нести риски и расходы по созданию и продвижению собственного интернет-магазина.

Согласно исследованию Forrester Research, в 2019-2020 годах 80 % потребителей хотя бы один раз совершали покупки онлайн, а 50 % делали это больше одного раза, при том что все покупки совершались через маркетплейсы. Около 71 % населения убеждены, что заказать и купить товары или услуги выгоднее через Интернет, чем в магазине.

Таким образом, актуальность разработки маркетплейса подкрепляется новыми возможностями реализации программных продуктов, которые в свою очередь повышают продажи любого магазина за счет удобного интерфейса и скорости обработки информации. Такая тенденция уже коснулась крупных и популярных интернет-магазинов со своей устоявшейся репутацией. Таким наплывом посетителей-покупателей похвастается не каждый интернет-магазин. На отечественном рынке разработка веб-приложений для электронной коммерция является относительно новым явлением как для разработчика и потребителя, так и для производителя. В новой разработке важно оценивать выгоды, которые может принести участие в электронном бизнесе, потенциальные проблемы, которые могут возникнуть в процессе реализации приложения. Вот почему так важно программно систематизировать преимущества маркетплейса для всех заинтересованных субъектов деятельности, которые разделены на потребителей, на производителей и разработчиков.

Разработчик любого программного продукта, приложения является самым важным человеком, поскольку он взаимодействует и с потребителем, и с производителем. При реализации программного приложения «Market place» следует оценивать важные особенности, которые влияют на выбор потребителя в пользу использования данного приложения, то есть, это выбор в пользу бесплатных приложений, доступность информации о товарах и услугах в режиме реального времени, использование торговых площадок. С точки зрения производителя, разработка приложений-маркетплейсов должна в себя включать предоставление новых видов услуг и освоение новых сегментов программного обеспечения, поддержка возможностей работы в режиме круглосуточного доступа к своевременно обновляющейся информации.

Международный рейтинг электронной коммерции не меняется в течение многих лет, и Китай остается лидером в этой гонке с оборотом в 562,7 млрд долларов, на ним следует Соединенные Штаты с оборотом 341,1 млрд долларов. Соотношение 10 крупнейших рынков маркетплейса в электронной коммерции за 2020 год представлен на рисунке 1.1. Отметим тот факт, что Россия не входит первую десятку стран, хотя в ней и существует достаточно большой потенциал прироста, поскольку средняя доля онлайн-продаж в мире показывает тенденцию к росту.

Рисунок 1.1 – Объем крупнейших *рынков* маркетплейса в мире за 2020 г.

Таким образом маркетплейсы в последнее время стали одними из лидеров роста российского рынка электронной коммерции. Популярность такого решения наглядно демонстрирует рейтинг пятнадцати крупнейших интернет-магазинов России, 60 % из которых в той или иной степени уже попробовали на себе роль маркетплейса (рисунок 1.2).

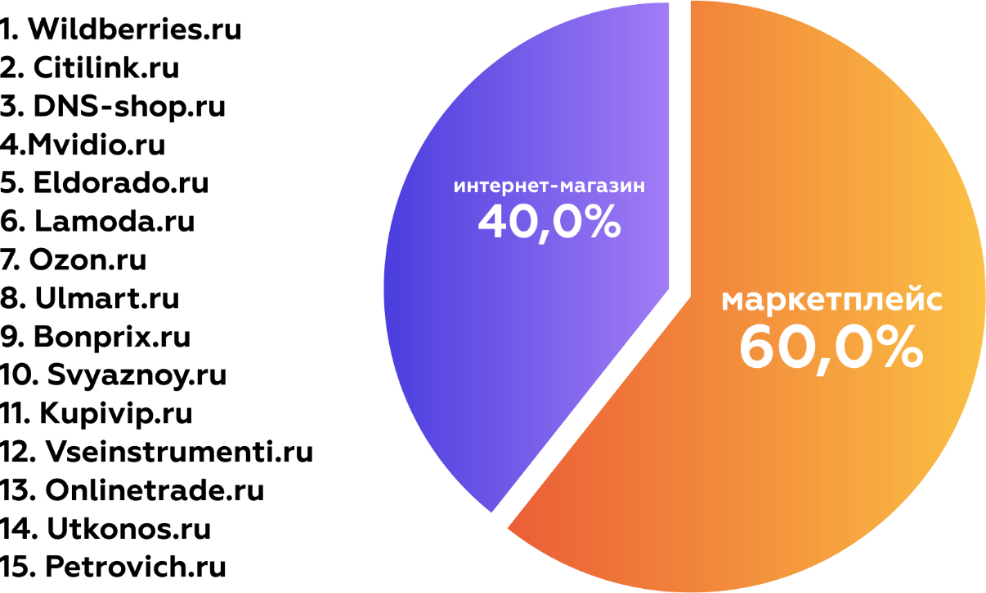


Рисунок 1.2 – Пятнадцать крупнейших интернет-магазинов России

Таким образом, можно сделать вывод, что в современном мире торговым организациям, чтобы идти в ногу со временем необходимо реконструировать принципы ведения бизнеса, идти на компромиссы, осваивать новые способы продажи товаров и услуг и менять свои правила. Таковым является выход в электронное пространство, путем предоставления услуг в Интернете с помощью сайтов или разработки мобильного веб-приложения для каждого интернет-магазина или компании. Разработка современного программного обеспечения направлена на поддержку развития любого вида бизнеса путем создания ресурсов в Интернете, налаживания эффективной коммуникации с широким кругом клиентов.

Таким образом, нами была рассмотрена основная предметная область использования маркетплейсов, которая предоставляет совершенно различный спектр площадей для любого вида услуг, начиная от продуктов, заканчивая косметическими услугами. Осуществлен разбор влияния маркетплейсов на современное общество, а также выявлена актуальность разрабатываемого мобильного веб-приложения.

## **1.2 Сравнительный аналитический обзор приложений «Market place»**

Для выявления функциональных особенностей и предоставляемых услуг приложениями маркетплейсов рассмотрим их существующие аналоги.

В основе реализации информационной системы каждого маркетплейса лежит одна из трех базовых программных моделей [4] (таблица 1.1): C2C (customer-to-customer); B2C (business-to-customer); B2B (business-to-business).

Таблица 1.1 – Базовые модели маркетплейса

|  |  |  |
| --- | --- | --- |
| Тип | Описание | Пример |
| C2C (customer-to- customer) | Физические лица взаимодействуют друг с другом на равных. Их объединяют общие интересы, они сотрудничают на взаимовыгодных условиях и легко меняются ролями | BlaBlaCar |
| B2C (busines-to- customer) | Клиентами маркетплейса являются физические лица, а предпринематели выступают в качестве поставщиков | AliExpress |
| B2B (busines-to-busines) | объединение предпринимателей и позволяет им находить успешные решения для своего бизнеса. На таких площадках встречаются клиенты, которые ищут оптовых поставщиков или крупных подрядчиков, и продавцов, которым нужны новые рынки сбыта | AliBaBa |

В разработанной программной модели C2C разработчики подошли с обеих сторон в реализации проекта. В данном примере обе стороны – это физические лица, которые взаимодействуют друг с другом на равных. Примером является, такой популярный проект как BlaBlaCar (рисунок 1.3), данный вид маркетплейса позволяет водителям находить попутчиков.

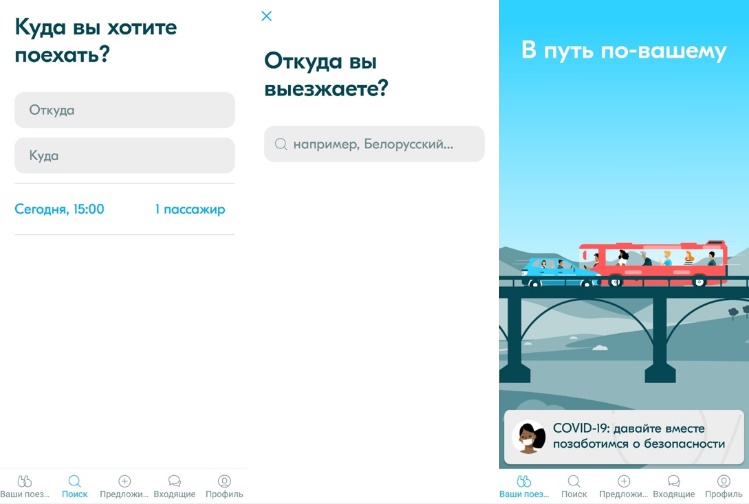


Рисунок 1.3 – Мобильное приложение маркетплейса BlaBlaCar

Иными словами, у одного человека имеется свободное место в автомобиле, у другого – деньги на бензин. Очевидно, что оба оказываются в выигрыше. Данный вид маркетплейса существует как мобильное приложение, которое можно скачать для двух платформ смартфонов (iOS, Android), воспользовавшись магазином того или иного «софта». Специального расширения на компьютер нет, но можно установить приложение на «базе» Android или iOS, используя специальную программу. Сейчас популярна программа ARC Welder, позволяющая открывать большинство расширений, предназначенных для смартфонов, через компьютер.

В программной модели маркетплейса B2C-типа разработчик оставляет клиентов как частные (физические) лица, а предприниматели действуют в роли поставщиков. Примером такой модели является AliExpress (рисунок 1.4), где на одном сайте можно найти тысячи товаров от футболок до мотоциклов. Их производят и выставляют на продажу множество разных компаний. Таким образом и клиент, и поставщик имеют свои личные кабинеты или же другими словами программный конструктор, в котором можно работать. Данный вид маркетплейса очень распространён, поэтому он имеет как сайт, так и мобильное приложение, последнее можно скачать на любом из доступных сервисах-магазинах, таких как AppStore или PlayMarket.

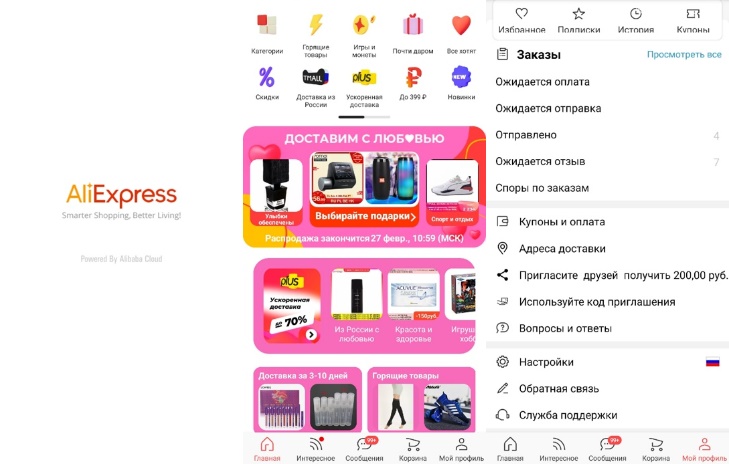


Рисунок 1.4 – Мобильное приложение маркетплейса Aliexpress

Торговая площадка AliExpress представляет полностью локализованное мобильное приложение «AliExpress», которое доступно как жителям России, так и стран СНГ. Оно позволяет вести разработку и обновлять платформу без привязки к глобальным процессам AliExpress благодаря специально развёрнутой локальной инфраструктуре. Главной задачей, над которой работала российская IT-команда, стала локализация продукта. Она создала инфраструктуру для локальной разработки и переноса основных функций на собственную платформу.

Существует и смешанная модель, она возникает посредством соединения B2C- и C2C-моделей в одной площадке. Программная модель B2B связывает предпринимателей и позволяет им находить удачные решения для своего бизнеса. Существует как сайт для данного проекта, так и мобильное приложение, которое можно скачать на доступных сервисах. Маркетплейс Airbnb использует быстроразвивающуюся платформу React Native, которая является относительно новым инструментом в среде Android, iOS, веб и кроссплатформенных фреймворков. Благодаря разработчикам данного маркетплейса, был разработан язык кроссплатформенного дизайна под названием DLS. Наличие такого унифицированного языка проектирования упростило написание кроссплатформенных функций для мобильных приложений.

На российском рынке существуют площадки – Яндекс.Маркет, Ozon и др., которые работают по принципу маркетплейса, но при этом просто являются витриной для сравнения цен на товары из онлайн-магазинов. Одним из самых популярных агрегаторов цен на товары был «Яндекс.Маркет», ошибочно называемый маркетплейсом, каковым он не являлся на тот момент. В отличие от другого проекта компании Яндекс, созданного совместно со Сбербанком – маркетплейса «Беру». На сегодняшний день торговая площадка «Беру» была интегрирована в «Яндекс.Маркет», что привело к появлению единого маркетплейса.

Программный интерфейс приложения «Яндекс.Маркет» представляет собой набор готовых процедур, функций и структур, с помощью которых разработчики могут создавать программы и приложения для работы и интеграции с сервисом. На настоящий момент код приложения «Яндекс.Маркет» системно не стабилизирован из-за постоянных смен команды программистов - нет ни статического анализа, ни метрик, ни подсчета CodeCoverage[[1]](#footnote-1) (покрытие кода). Большинство проектов, реализуемых на данной площадке, собираются или запускаются на Linux.

Маркетплейс Ozon использует две модели работы. Обе схемы подразумевают самостоятельное управление продажами: продавцы сами проходят регистрацию на площадке, создают контент, управляют ассортиментом и ценами, решают, как продвигать свои товары со своего склада – они собирают товары в заказ сами и передают их в службу доставки Ozon. В своей архитектуре Ozon использует Kubernetes[[2]](#footnote-2) и другие программные инструменты от Google, которые написаны на языке программирования Go. Это позволяет не разрабатывать свои решения, а дорабатывать готовые. Кроме того, в инструментах для высоконагруженных систем есть адаптеры для Go. Ozon стремится к созданию единой экосистемы на основе одного языка. Например, все продукты Alibaba построены на Java, таким образом огромный ретейлер создает кастомные версии всех необходимых продуктов.

Таким образом, рассмотрев три современные базовые программные модели реализации маркетплейсов – C2C (customer-to-customer); B2C (business-to-customer); B2B (business-to-business), их реализацию в приложениях, а также разобрав их полное техническое обоснование с точки зрения разработчика выяснили, что на сегодняшний день очень популярна смешанная модель реализации маркетплейсов. Она стала популярной в использовании за счет того, что в ней разработчик оставляет клиентов как физические лица, а в роли поставщиков выступают предприниматели.

**1.3** **Техническое задание на разработку мобильного веб-приложения «Market place» для салонов красоты**

Рассмотрев и определив важные аспекты предметной области мобильной разработки маркетплейсов, разработаем техническое задание для правильной реализации мобильного веб-приложения «Market place» для салонов красоты.

Проведя анализ сегментирования целевой аудитории для салонов красоты, выяснили, что подавляющее число посетителей – это женщины и мужчины 25-45 лет. Но за последнее время выросло доля подростков 14-18 лет. Таким образом целевая аудитория приложения составляет от 14 до 45 лет.

Информационная система разрабатываемого нами маркетплейса для салонов красоты должна представляться в виде мобильного веб-приложения с пользовательским интерфейсом, серверной частью и базой данных.

Разрабатываемое мобильное веб-приложение «Market place» для салонов красоты, должно предоставлять сервис по созданию, хранению, обработки и выводу информации об услугах, салонов, контактной информации и новостях.

Критерии пользователей. Разрабатываемое мобильное веб-приложение является прототипом и содержит только клиентскую часть, где клиенту-пользователю предоставляется возможности, описанные ниже.

Для доступа к сервису и возможности иметь личный кабинет нужно быть зарегистрированным пользователем. Для регистрации пользователя требуется отдельная форма, которая требует обязательного заполнения полей:

* фамилия и имя;
* номер телефона.

После успешного входа в приложение, система перенаправляет пользователя на главную Lendingpage страницу, на которой ему представляется возможность доступа к информации и пользованию сервисом. Выделяется личный кабинет, где пользователь в разделе «Настройки» имеет право изменять и управлять своими данными и созданной информацией.

Краткая страница описания конкретного салона должна демонстрировать следующую информацию:

* текстовое описание салона;
* услуги;
* карта.

На главной странице выводятся краткие данные всех салонов и есть возможность перейти по ссылке на страницу подробного описания выбранного салона.

На странице описания конкретного салона необходимо предоставить подробную информацию о салоне:

* текстовое описание салона;
* услуги и их цены;
* рейтинг салона;
* описание особенностей салона;
* галерея, карта;
* контактная информация;
* отзывы.

На странице поиска необходимо выводить информацию о всех интересующих салонах, а также осуществлять поиск по определенным категориям.

Страница понравившихся салонов должна демонстрировать краткую информацию:

* текстовое описание салона;
* услуги;
* карта.

На странице уведомлений должны выводиться все уведомления записей в выбранный салон.

После выбора соответствующего салона система должна открывать новую форму для записи, включающую:

* услуги, предоставляемые салоном;
* дата и время;
* салон;
* адрес;
* стоимость услуг.

Функции, которое необходимо автоматизировать:

* Поиск и выбор салонов – данная функция предназначена для поиска салонов, информация о которых хранится в базе данных. Пользователь делает запрос и моментально получает необходимую информацию.
* Выбор услуг – данная функция предназначена для самостоятельного просмотра и выбора услуг предоставляемые салонами, а также просмотра цены услуг и характеристик салонов.
* Выбор даты и времени – данная функция предназначена для самостоятельного выбора только доступной даты с временем записи.
* Запись – автоматическая запись, без вмешивания третьих лиц.

Таким образом, на основе важных аспектов предметной области мобильной разработки маркетплейсов, разработали техническое задание для правильной реализации мобильного веб-приложения «Market place» для салонов красоты.

## **1.4 Обоснование выбора языка и среды разработки мобильного веб-приложения «Market place» для салонов красоты**

Для создания мобильного веб-приложения «Market place» для салонов красоты использовалось два языка программирования Javascript и Python.

Javascript – динамический скриптовый язык программирования высокого уровня. Он отличается мультипарадигменностью. Речь идет о поддержке объектно-ориентированного модуля. Чаще всего язык используется для создания интерактивных веб-страниц и приложений.

Javascript популярен благодаря своим свойствам, а именно:

1. Поддержка скриптов всеми популярными браузерами; полная интеграция с вёрсткой страниц (HTML+CSS) и серверной частью (backend).
2. Позволяет частично обрабатывать веб-страницы на компьютерах пользователя без запросов к серверу. Это экономит время и трафик, снижает нагрузку на сервер.
3. Заполнение форм, выбор действий, активация кнопок, проверки ввода, реагирование на наведение / клики мыши и т.п. это даёт потрясающий уровень юзабилити. Сначала код может показаться сложным, но к его синтаксису, логике быстро привыкаешь. Отображение действий ощутимо добавляет энтузиазма.

Python – мультипарадигмальный язык программирования. Полностью поддерживающий объектно-ориентированное, структурное, обобщенное, функциональное программирование. Python упрощает написание кода делает разработку быстрой, потому что обладает динамической типизацией, автоматическим видением памяти, привязкой типов данных и др.

В качестве среды разработки выбрал программу Android Studio – интегрированная среда (IDE) для разработки мобильного софта в операционной системе Андроид.

В программу встроен эмулятор, позволяющий проверить корректную работу приложения на устройствах с разными экранами, с различными соотношениями сторон. Отличительная особенность эмулятора – просмотр приблизительных показателей производительности при запуске приложения на самых популярных устройствах.

Среда разработки для приложений Android Studio последней версии стала по настоящему удобной даже для начинающих разработчиков. В программе реализованы все современные средства для упаковки кода, его маркировки.

Локализация приложений становится существенно проще с функцией SDK, которая также входит в перечень достоинств Android Studio. Связь с целевой аудиторией приложения после его релиза поможет реализовать инструмент от компании Google – Google Cloud Messaging. Среда разработки поддерживает работу с несколькими языками программирования, к которым относятся самые популярные – C/C++, Java. Тестирование корректности работы новых игр, утилит, их производительности на той или иной системе, происходит непосредственно в эмуляторе. В Android Studio находится достаточно большая библиотека с готовыми шаблонами и компонентами для разработки ПО. Для неопытных/начинающих разработчиков специально создано руководство по использованию Android Studio, размещенное на официальном сайте утилиты.

Для разработки любого вида дизайна приложений, специалисты рекомендуют использовать графический редактор, который умеет работать со слоями, чтобы грамотно взаимодействовать с изображениями и другими элементами дизайн-макета или прототипа. Почти весь современный софт обладает солидным набором инструментов, в которых можно легко запутаться, поэтому следует выбирать программные продукты с отличным функционалом для разработки дизайна, но в то же время, с интуитивно понятным интерфейсом. Таковым является продукт Adobe Photoshop – это программа профессиональных дизайнеров для всех, кот связан с отрисовкой макета приложения, сайта, редактирования или обработки графических изображений. В данном программном продукте можно рисовать графику сайтов и приложений, добавлять картинки, тексты, тонко настраивая элементы с помощью панели инструментов. Удобство сервиса заключается в автоматической генерации стилей для продукта. Пользователю достаточно добавить картинку, отрегулировать опции в панели и сразу будет виден результат в стилях.

Интерфейс – это иными словами оформление, то, как выглядит приложение либо сайт внешне, его оболочка. Причём это не просто то, как он выглядит, а именно его системная структура. Интерфейсы являются основой взаимодействия всех современных информационных систем.

Таким образом, в качестве языка программирования для разработки мобильного веб-приложения «Market place» для салонов красоты были выбраны JavaScript и Python, а в качестве разработки дизайна пользовательского интерфейса данного приложения AdobePhotoshop [10] и Visual Studio. Приложение взаимодействует с БД используя ORM, которая создает некую виртуальную базу, запросы которой, преобразуется из программного кода Python в SQL команды.

## **1.5 Вывод по первому разделу**

В данном разделе был осуществлен анализ предметной области, для разрабатываемого мобильного веб-приложения «Market place» для салонов красоты. Определена актуальность данной разработки, которая подкрепляется новыми возможностями реализации программных продуктов, поскольку на отечественном рынке разработка веб-приложений для электронной коммерция является относительно новым явлением.

Рассмотрели три современные базовые программные модели реализации маркетплейсов – C2C (customer-to-customer); B2C (business-to-customer); B2B (business-to-business), их реализацию в приложениях, а также разобрали аналоги приложений и их полное техническое обоснование с точки зрения разработчика. На основе предметной области мобильной разработки маркетплейсов, разработали техническое задание для правильной реализации мобильного веб-приложения «Market place» для салонов красоты.

Осуществили обоснование выбора языков и сред разработок. Для создания мобильного веб-приложения «Market place» для салонов красоты использовалось два языка программирования Javascript и Python.

В качестве среды разработки выбрал программу Android Studio – интегрированная среда (IDE) для разработки мобильного софта в операционной системе Андроид. Для отрисовки интерфейса приложения использовался Adobe Photoshop –программа профессиональных дизайнеров, для создания макета приложения, сайта, редактирования или обработки графических изображений.

# **2 Проектирование мобильного веб-приложения «Market place» для салонов красоты**

## **2.1 Анализ требований на основе построения диаграмм прецедентов, классов, последовательности веб-приложения «Market place» для салонов красоты**

На основе технического задания разрабатываемого мобильного веб-приложения в разделе 1, были определены прецеденты[[3]](#footnote-3), которые могут возникнуть в информационной системе нашей разработки (таблица 2.1). Прецедент [1] является отдельным сервисом информационной системы, устанавливающий один из способов ее применения и описывает классический способ взаимодействия клиента с системой.

Таблица 2.1 – Прецеденты системы

|  |  |
| --- | --- |
| Пользователь | Информационная система |
| Авторизация | |
| Как пользователь вносит контактную информацию в виде номера телефона как логин для входа в приложение | Проверяет введение пользователем данные. Сохраняет информацию в базе данных |
| Настройки | |
| Пользователь вводит в систему Имя, Фамилию, и если изменился его номер телефона, меняет его | Система принимает подлинность новых данных и сохраняет их |
| Краткий обзор | |
| Пользователь может просмотреть краткую информацию о салоне, а именно его расположение, вид предоставляемых услуг и название | Дает доступ для просмотра краткой информации о салоне, в виде раскрывающегося списка |
| Выбрать салон | |
| Пользователь выбирает салон, нажимая на соответствующий выбор | Перенаправляет пользователя на новую форму для дальнейшей работы |
| Запись | |
| Пользователь выбирает нужные ему услуги из перечня услуг, просматривает дополнительную доступную информацию | Система проверяет выбранные пользователем данные, и автоматически при выборе услуг подсчитывает стоимость |
| Подтверждение записи | |
| Пользователь выбирает день, время записи. Нажимает на подтверждение | Система предоставляет пользователю доступные дни для записи и время на этот день. перенаправляет его главную страницу и отсылает на приложение уведомление об успешной записи |

Продолжение таблицы 2.1

|  |  |
| --- | --- |
| Переход на главную страницу | |
| Пользователь имеет право выйти из приложения, либо продолжить запись | Система обновляет данные, пока пользователь не выйдет из приложения |

Представленные в таблице 2.1 прецеденты [1] можно отобразить в виде детализированных отношений включения (include) и расширения (extend) в диаграмме прецедентов [5] (рисунок 2.1). Главное ее предназначение – это представление функциональности и поведения, делающее возможным клиенту и программисту совместно рассматривать проектируемую либо уже разработанную информационную систему.

При построении концепции с помощью диаграммы прецедентов информационной системы [2] мы сформулировали информационную систему на ее сферы, определили функциональные лица (актеров), их связь в информационной системе, а также спрогнозировали функциональность системы.

Прецедент «Авторизация» подразумевает включающий прецедент «Номер телефона».

Прецедент «Настройки» расширяется прецедентами «Изменение номера телефона», «Ввод имени, фамилии».

Прецедент «Краткий обзор салона» имеет отношения расширения «Просмотреть описание» «Просмотреть услуги» «Просмотреть расположение». А также неразрывно связан с прецедентом «Переход на главную страницу». В котором пользователю доступен выход из предыдущего прецедента.

Прецедент «Выбрать салон» обуславливает определённый выбор салона пользователем, в котором осуществляется прецедент «Просмотр информации» включающий себя «Выбор услуги» с автоматическим прецедентом «Подсчет цены услуг».

Прецедент «Запись» включает «Выбрать день», «Выбрать время», «Подтверждение информации». После прохождения всех прецедентов записи, пользователю доступен прецедент «Переход на главную страницу».

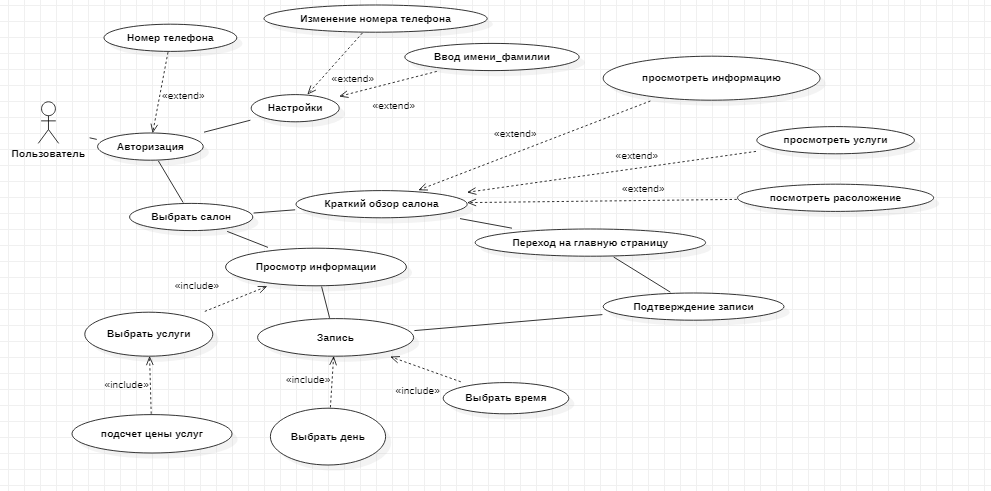


Рисунок 2.1 – Диаграмма прецедентов веб-приложения «Market place»

*Главный сценарий «Назначение работы»*

Пользователь: Запускает приложение на мобильном устройстве

ИС: Загружает приложение

Пользователь: Авторизуется, вводит номер телефона

ИС: Открывается главный экран приложения (Lending page), в котором пользователю доступен просмотр ленты (выбор салонов красоты, барбершопов).

Пользователь: Делает выбор, просматривая Lending page. Нажимает на выбранный салон

ИС: Открывается Lending page выбранного салона:

* 1. Информация о салоне;
  2. Деятельность;
  3. Контактная информация;
  4. Отзывы;
  5. Услуги на выбор;
  6. Прайс лист;

Пользователь: Может выбрать интересующие его услуги, сразу же просматривая их цену

ИС: В нижней части окна ведется подсчет выбранных им услуг.

Пользователь: Подтверждает свой выбор, нажимая на соответствующую кнопку.

ИС: Перенаправляет пользователя на форму записи

Пользователь: В форме записи должен обязательно выбирать определенный промежуток дней, в которые он хотел бы обратиться в салон.

ИС: Подтверждает выбор времени посещения.

ИС: Автоматически предлагает пользователю свободные часы для записи, в тот промежуток дней, который определил пользователь.

Пользователь: Подтверждает выбор, нажимает кнопку «записаться».

ИС: Автоматически перенаправляет пользователя на главную страницу приложения, где он может либо продолжить просмотр и в последующем записаться в другие салоны, либо завершить работу.

*Сценарий «прерванная регистрация услуги записи»*

Пользователь: Запускает приложение на мобильном устройстве

ИС: Загружает приложение

Пользователь: Авторизуется, вводит номер телефона

ИС: Открывается главный экран приложения (Lending page), в котором пользователю доступен просмотр ленты (выбор салонов красоты, барбершопов).

Пользователь: Делает выбор, просматривая Lending page. Нажимает на выбранный салон

ИС: Открывается Lending page выбранного салона:

* 1. Информация о салоне;
  2. Деятельность;
  3. Контактная информация;
  4. Отзывы;
  5. Услуги на выбор;
  6. Прайс лист;

Пользователь: Может выбрать интересующие его услуги, сразу же просматривая их цену

ИС: В нижней части окна ведется подсчет выбранных им услуг.

Пользователь: Подтверждает свой выбор, нажимая на соответствующую кнопку.

ИС: В нижней части окна ведется подсчет выбранных им услуг.

Пользователь: Подтверждает свой выбор, нажимая на соответствующую кнопку.

ИС: Перенаправляет пользователя на форму записи

Пользователь: В форме записи должен обязательно выбирать определенный промежуток дней, в которые он хотел бы обратиться в салон.

ИС: Подтверждает выбор времени посещения.

ИС: Автоматически предлагает пользователю свободные часы для записи, в тот промежуток дней, который определил пользователь.

Пользователь: Нажимает на кнопку «отменить».

ИС: Автоматически перенаправляет пользователя на главную страницу приложения.

Таким образом, мы рассмотрели основные требования к разработке мобильного веб-приложения «Market place» для салонов красоты опираясь на диаграмму прецедентов. Разработали главный сценарий работы данного, предусматривая сценарий об ошибке. Результатом стала информационная система, которая должна представляться в виде мобильного веб-приложения маркетплейса с пользовательским интерфейсом, серверной частью и базой данных.

При построении концепции с помощью диаграммы классов информационной системы воспользовались моделью предметной области. **Модель предметной области [5] –** это часть объектно-ориентированного анализа, которая показывает основные концептуальные классы [2] (представление идеи или объекта) с точки зрения разработчика в терминах предметной области. В UML модель предметной области представлена в виде диаграммы классов, которая отображает классы.

В данной выпускной квалификационной работе использовался метод, основанный на выделении существительных [5], который состоит из выбора существительных из текстового описания предметной области.

При рассмотрении предметной области выделим следующие классы, необходимые для правильного функционирования системы разрабатываемого мобильного веб-приложения «Market place» для салонов красоты:

Список полученных концептуальных классов: «User», «Administrator», «Client», «Salon», «Appointment», «Confirmation», «Calendar».

Рассмотрим подробную структуру каждого класса.

«User» – класс, объединяющий характеристики всех пользователей.

Должен обладать уникальным ID.

«Administrator», «Client» – классы, которые наследуют поля класса «User».

«Salon» – класс салона. Каждый объект данного класса представляет поля с информацией об одном салоне. Должен обладать уникальным ID.

«Appointment» – класс записи в салон. Каждый объект данного класса представляет поля с соответствующей информацией о записи в салон. Должен обладать уникальным ID.

«Confirmation» – класс подтверждения записи. Каждый объект данного класса представляет поля, заполненные ранее пользователем.

«Calendar» – класс календаря. Каждый объект данного класса представляет поля с информацией о сформированном клиентом бронировании записи в салон.

Подробнее рассмотрим поля и методы каждого класса.

Класс «User»:

* уникальный ID – ID: int;
* номер телефона – phone: string;
* имя – name: string;
* фамилия – surname: string;

Классы «Administrator», «Client» содержат те же поля что и класс «User», так как они наследованы от него.

Класс «Salon»:

* уникальный ID – ID: intger;
* название – name: string;
* адрес – address: string;
* маршрут: - route: string;
* название услуг – service\_name: string;
* цена услуг – service\_price: double;
* отзывы – reviews: string;
* описание – description: string;
* контакты – contacts: string;
* галерея – gallery: JPG.

Класс «Appointment»:

* уникальный ID – ID: intger;
* название услуг – service\_name: string;
* дата – date: string;
* время – time: double;
* Класс Confirmation:
* дата – date: string;
* время – time: double;
* название услуг – service\_name: string;
* название – name: string;
* адрес – address: string;
* Стоимость – cost: int
* уникальный ID – ID\_Appointment: int.

Класс «Calendar»:

* уникальный ID – ID\_Salon: int;
* дата – date: string;
* время – time: double;

Атрибуты классов, и их отношения, связи между приведенными классами представлены на UML-диаграмме классов на рисунке 2.1.

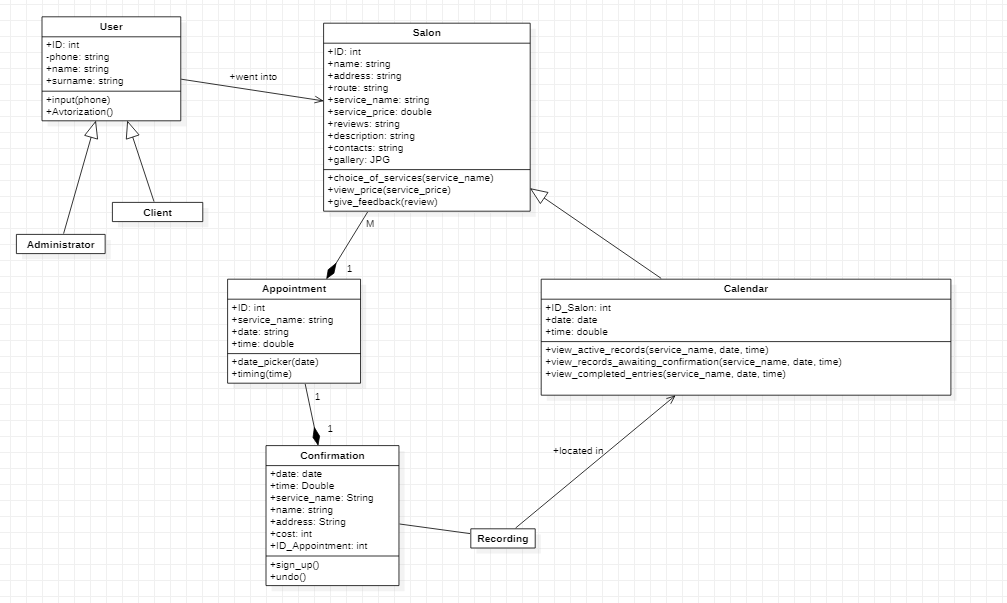


Рисунок 2.2 – Диаграмма классов веб-приложения «Market place»

Таким образом, была построена м**одель объектно-ориентированного анализа для мобильного веб-приложения** «Market place» для салонов красоты**,** отображающая основные классы в терминах предметной области. На основе языка uml разработали диаграмму классов для данного приложения, которая отображает классы понятий реального мира, а также ассоциации между классами и атрибуты концептуальных классов.

Для построения системных операций использовали диаграммы последовательности. Системная операция – это операция, включенная в интерфейс открытой системы для обработки входящих системных событий, которую система выполняет как черный ящик.

Описание операции «New User Registration» (регистрация нового пользователя) для создаваемого мобильного веб-приложения «Market place» для салонов красоты представлена в таблице 2.2.

Таблица 2.2 – Регистрация нового пользователя

|  |  |
| --- | --- |
| Операция | Прохождение регистрации |
| Ссылки | Прецедент «Авторизация» |
| Предусловия | В системе нет зарегистрированных пользователей |
| Постусловия | Добавлен новый пользователь |

Более подробно данную системную операцию описывает диаграмма последовательности (рисунок 2.3).

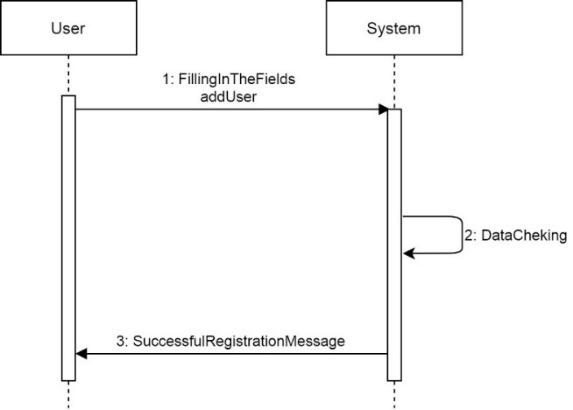


Рисунок 2.3 – Диаграмма последовательности для операции «Регистрация нового пользователя»

Описание операции «Entry to the Salon» (запись в салон) для разрабатываемого приложения представлена в таблице 2.3.

Таблица 2.3 – Запись в салон

|  |  |
| --- | --- |
| Операция | Запись в салон |
| Ссылки | Прецедент «Запись» |
| Предусловия | В системе есть пользователь |
| Постусловия | В систему добавлена запись в салон |

Более подробно данную системную операцию, с точки зрения обращения пользователя к системе описывает диаграмма последовательности (рисунок 2.4).

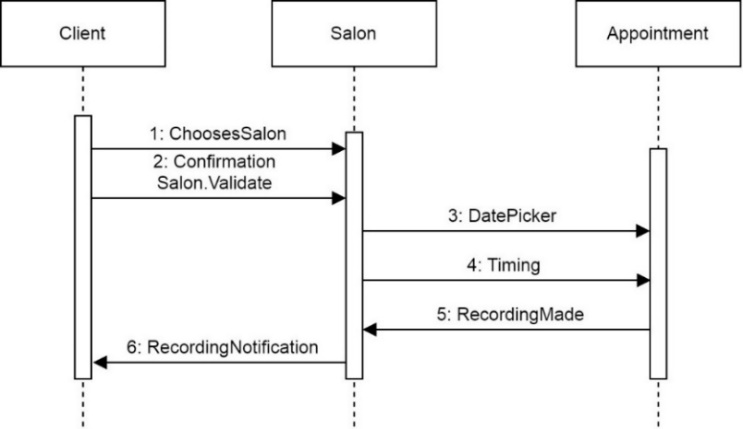


Рисунок 2.4 – Диаграмма последовательности для операции «Запись в салон»

Следующим основным этапов является подробное построение диаграмм последовательностей, которые должны показывать отношение из множества объектов системы с пользователем в процессе выполнения системных требований.

Построим диаграмму последовательностей для операций «Регистрация нового пользователя» и «Запись в салон» (рисунки 2.5 – 2.6).

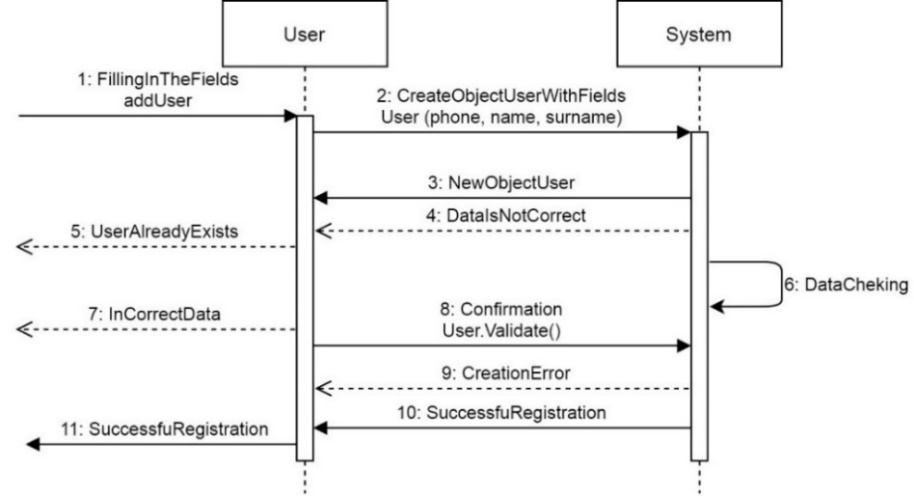


Рисунок 2.5 – Диаграмма последовательности для операции «Регистрация нового пользователя»

На диаграмме (рисунок 2.5) показана последовательность действий при регистрации нового пользователя в системе. При получении запроса система вызывает метод addUser [3], который создает объект User (листинг кода приложения представлен в приложении А), который в свою очередь проверяет корректность введённых данных и сохраняет данные.

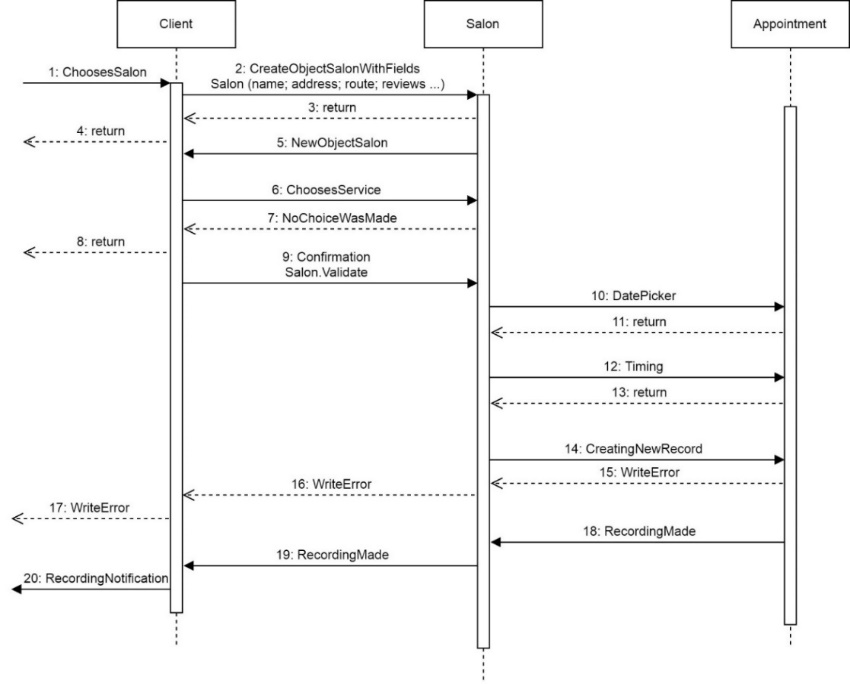


Рисунок 2.6 – Диаграмма последовательности для операции «Запись в салон»

На диаграмме (рисунок 2.6) показана последовательность действий при записывании в салон. При получении запроса система вызывает метод ChoosesSalon класса Salon. В этом методе сначала создаётся новый объект [8] класса Salon, затем вызывается метод validate у этого объекта [9]. Метод validate проводит проверку свойств объекта Salon (смотри приложение А). Если запись прошла успешно, клиент получает уведомлении о записи.

Таким образом мы построили диаграмму последовательности мобильного веб-приложения «Market place» для салонов красоты, описывающую основные функции, которые выполняются на основе взаимодействия пользователя с системой.

## **2.2 Структурные принципы проектирования мобильного веб-приложения «Market place» для салонов красоты**

Функциональная структура разрабатываемого нами мобильного веб-приложения «Market place» для салонов красоты состоит из следующих модулей:

* модуль авторизации;
* модуль отображения салонов красоты, барбершопов;
* модуль записи в салон;
* модуль контроля записи в салон;
* модуль отслеживания местоположения;
* модуль синхронизации данных;
* сервисный модуль.

Модуль авторизации обеспечивает регистрацию под конкретного пользователя на период его работы. Авторизация производится путем подключения к беспроводным каналам связи. При авторизации пользователя происходит загрузка его данных и список доступных салонов.

Модуль отображения салонов красоты обеспечивает отображение салонов, барбершопов на мобильном устройстве, а также их расположение.

Модуль записи в салон предоставляет клиенту (пользователю) интерфейсы для осуществления просмотра, выбора и записи в салон. Выходной информацией данного модуля являются данные для автоматического формирования базы данных.

Модуль контроля записи в салон предоставляет администратору интерфейсы для просмотра и редактирования зарегистрированных пользователей.

Модуль отслеживания местоположения выполняет функцию формирования и передачи навигационных данных с заданной периодичностью на сервер смежной системы для определения местоположения пользователя, находящегося в любом месте, для отрисовки маршрута к салону с точностью, обеспечиваемой средствами спутниковой навигации.

Модуль синхронизации обеспечивает синхронизацию данных между приложением на мобильном устройстве и базой данных в период работы пользователя с использованием беспроводных каналов связи.

Сервисный модуль мобильного рабочего места обеспечивает возможность работы со встроенными в мобильное устройство GPS модулям.

Таким образом, в данном подразделе рассмотрели основные структурные модули, необходимые для мобильного веб-приложения «market place» для салонов красоты - модули авторизации, отображения салонов красоты, барбершопов, записи в салон; контроля записи, отслеживания местоположения, синхронизации данных и сервисного модуля.

## **2.3 Расчет функционально-ориентированных (FP), LOC-метрик, COCOMO-метрик и анализ рисков для мобильного веб-приложения «Market place»**

Для измерения разрабатываемого мобильного веб-приложения «Market place» для салонов красоты рассчитаем функционально-ориентированные метрики [7], которые основываются на размерно-ориентированные метрики на LOC-оценках. LOC-оценка – это количество строк в программном продукте.

В таблице 2.4 описывается количество внешних вводов. Внешний ввод – элементарный процесс, перемещающий данные из внешней среды в приложение.

Таблица 2.4 – Ранг и оценка сложности внешних вводов мобильного веб-приложения «Market place» для салонов красоты

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Транзакции: Внешние вводы | | | | | |
| Название ввода | Поля ввода и элементы данных | Кол-во элементов данных | Ссылки на файлы | Ранг | Кол-во вводов | Общая сложность (общ. ранг) |
| Авторизация | Поля: что  Кнопка: продолжить | 2 | 0-1 | Низкий =3 | 1 | 3\*1=3 |

Продолжение таблицы 2.4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выбрать салон | Поля: что, где, Кнопка: салон, где, нравится, услуги | 6 | 0-1 | Низкий =3 | 1 | 3\*1=3 |
| Поиск салона | Поля: что, где, кнопка: салон, где, нравится, услуги | 6 | 0-1 | Низкий =3 | 1 | 3\*1=3 |
| Добавить бронь | Поля: что, когда, во сколько, кто  Кнопка продолжить + сообщение подтверждения | 6 | 0-1 | Низкий =3 | 1 | 3\*1=3 |
| Уведомление | Поля: что, где, дата, время | 4 | 0-1 | Низкий =3 | 1 | 3\*1=3 |

Таблица 2.5 описывает количество внешних выводов. Внешний вывод [5] *–* это элементарный процесс, который перемещает данные, вычисленные в приложении, во внешнюю среду. В этом контексте выводы означают отчеты, экраны, распечатки, сообщения об ошибках.

Таблица 2.5 – Ранг и оценка сложности внешних выводов мобильного веб-приложения «Market place» для салонов красоты

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Транзакции: Внешние выводы | | | | | |
| Название вывода | Поля вывода и элементы данных | Кол-во элементов данных | Ссылки на файлы | Ранг | Кол-во выводов | Общая сложность (общ. ранг) |
| Просмотреть бронь | Поля: что, где, когда, во сколько и кнопка ОК | 5 | 0-1 | Низкий =4 | 1 | 4\*1=4 |
| Напоминание | Поля: что, где, когда, во сколько и кнопка ОК | 5 | 0-1 | Низкий =4 | 1 | 4\*1=4 |

Таблица 2.6 описывает количество внешних запросов. Внешний запрос [5] *–* это процесс, работающий как с входными, так и с выходными данными. Результатом являются данные, возвращаемые внутренними логическими файлами внешнего интерфейса.

Таблица 2.6 – Ранг и оценка сложности внешних запросов мобильного веб-приложения «Market place» для салонов красоты

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Транзакции: Внешние запросы | | | | | |
| Название запроса | Поля ввода и элементы данных | Кол-во элементов данных | Ссылки на файлы | Ранг | Кол-во запросов | Общая сложность (общ. ранг) |
| Просмотреть все уведомления (3 шт.) | Поля: что, где, когда, во сколько, ОК \*3шт. | 15 | 0-1 | Низкий =3 | 1 | 3\*1=3 |
| Просмотреть все понравившиеся салоны (3 шт.) | Поля: что, где \*3шт. | 6 | 0-1 | Низкий =3 | 1 | 3\*1=3 |

Таблица 2.7 описывает количество внутренних логических файлов. Внутренний логический файл [5]– это логически распознаваемая пользователем группа связанных данных, расположенных в приложении и обслуживаемых через внешние вводы.

Таблица 2.7 – Ранг и оценка сложности внутренних логических файлов мобильного веб-приложения «Market place» для салонов красоты

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Внутренние логические файлы | | | | | |
| Название файла | Поля ввода и эл-ты данных | Кол-во эл-ов данных | Кол-во эл-ов данных записей | Ранг | Кол-во файлов | Общая сложность (общ. ранг) |
| Логический файл «главная страница» | что, где, рейтинг | 3 | 1 | Низкий 7 | 1 | 7\*1=7 |
| Логический файл «календарь» | Дата, время, салон, услуга \*3шт. | 12 | 1 | Низкий =7 | 1 | 7\*1=7 |
| Логический файл «поиск» | Что, где, рейтинг | 3 | 1 | Низкий =7 | 1 | 7\*1=7 |
| Логический файл «нравится» | Что, где, рейтинг | 3 | 1 | Низкий =7 | 1 | 7\*1=7 |
| Логический файл «уведомления» | Что, где, когда, во сколько | 4 | 1 | Низкий =7 | 1 | 7\*1=7 |

В таблице 2.8 каждая из указанных характеристик связана со сложностью реализации. Для этого характеристике присваивается низкий, средний либо высокий ранг, а затем формируется рейтинг ранга.

Таблица 2.8 – Исходные данные для расчета FP-метрик

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя характеристики | Ранг, сложность, количество | | | |
|  | Низкий | Средний | Высокий | Итого |
| Внешние вводы | 5x3 = 15 | 0x4 = х | 0x6 = х | = 15 |
| Внешние выводы | 2x4 = 8 | 0x5 = х | 0x7 = х | = 8 |
| Внешние запросы | 2х3 = 6 | 0x4 = х | 0x6 = х | = 6 |
| Внутренние логические файлы  Внешние интерфейсные файлы | 5x7 = 35  0x5 = 0 | 0x 10= х  0x7 = х | 0x15 = х  0x10 = х | = 35  = 0 |
| Общее количество S = | | | | = 67 |

Количество функциональных указателей *FP* вычисляется по формуле 2.1

где *Fi* – коэффициенты регулировки сложности.

*S* – Общая сумма рангов.

Каждый коэффициент регулировки сложности *Fi* может принимать следующие значения:

0 – нет влияния (не важно), 1 – случайное (слабое), 2 – небольшое, 3 – среднее, 4 – важное, 5 – основное.

Значения выбираются эмпирически в результате ответа на 14 вопросов, которые характеризуют системные параметры будущего ПО (таблица 2.9).

Исходя из (таблица 2.9) диапазон изменения *количества функциональных указателей* высчитывается по формуле 2.2:

Таблица 2.9 – Определение системных параметров приложения

|  |  |  |  |
| --- | --- | --- | --- |
| № | Системный параметр | Описание | *Fi* |
| 1 | Передачи данных | Насколько много средств связи требуется для передачи или обмена информацией с приложением или системой? | *F1=3* среднее |
| 2 | Распределенная обработка данных | Как обрабатываются распределенные данные и функции обработки? | *F2=3* среднее |
| 3 | Производительность | Нуждается ли пользователь в фиксации времени ответа или производительности? | *F3=3* среднее |
| 4 | Распространенность используемой конфигурации | Насколько распространена текущая аппаратная платформа, на которой будет выполняться приложение? | *F4=1* случайно |
| 5 | Скорость транзакций | Как часто выполняются транзакции? (каждый день, каждую неделю, каждый месяц) | *F5=5* каждый день |
| 6 | Оперативный ввод данных | Какой процент информации надо вводить в режиме онлайн? | *F6=4* почти всю |
| 7 | Эффективность работы конечного пользователя | Приложение проектировалось для обеспечения эффективной работы конечного пользователя? | *F7=4* важное |
| 8 | Оперативное обновление | Как много внутренних файлов обновляется в онлайновой транзакции? | *F8=3* среднее |
| 9 | Сложность обработки | Выполняет ли приложение интенсивную логическую или математическую обработку? | *F9=3* среднее |
| 10 | Повторная используемость | Приложение разрабатывалось для удовлетворения требований одного или многих пользователей? | *F10=4* важное |
| 11 | Легкость инсталляции | Насколько трудны преобразование и инсталляция приложения? | *F11=2*небольшое |
| 12 | Легкость эксплуатации | Насколько эффективны и/или автоматизированы процедуры запуска, резервирования и восстановления? | *F12=3* среднее |
| 13 | Разнообразные условия размещения | Была ли спроектирована, разработана и поддержана возможность инсталляции приложения в разных местах для различных организаций? | *F13=3* среднее |
| 14 | Простота изменений | Была ли спроектирована, разработана и поддержана в приложении простота изменений? | *F14=3* среднее |
|  |  |  | 44 |

Для разрабатываемого веб-приложения «Market place» для салонов красоты количество функциональных указателей будет следующим:

FP-метрики легко пересчитать в LOC-метрики. (LOC– Lines Of Code). LOC-метрика – это количество строк в программном продукте. А результаты такого пересчета зависят от используемого для реализации приложения языка программирования.

Для разрабатываемого веб-приложения «Market place» для салонов красоты был выбран язык программирования JavaScript и Python. Данные языки программирования имеют достаточно похожий функционал, поэтому, в среднем на 1 функциональную точку на языке JavaScript или Pythonприходится 53 строк кода. В данном программном продукте количество строк высчитывается по формуле 2.3.

Таким образом мы произвели основной расчет функционально-ориентированных метрик разрабатываемого мобильного веб-приложения «Market place» для салонов красоты и размерно-ориентированных метрик, основанных на LOC-оценках. Количество строк в данном в разрабатываемом приложении будет составлять 3885 строк программного кода.

Для расчета затрат на разработку приложения, используется модель раннего этапа проектирования.

Основное уравнение этой модели имеет следующий вид (формула 2.4):

где *А* – масштабный коэффициент *А* = 2,5;

*Размер* – размер ПО (количество строк) выраженный в тысячах LOC;

*Мe –* множитель поправки зависит от 7 формирователей затрат, характеризующих продукт, процесс и персонал;

*В* – показатель степени, отражающий нелинейную зависимость затрат от размера проекта (от длины кода LOC).

Значение показателя степени *В* изменяется в диапазоне 1,01... 1,26, зависит от 5 масштабных факторов *Wi* (таблица 2.10) и вычисляется по формуле 2.5.

Общая характеристика масштабных факторов *Wi* позволяет определить оценки этих факторов. Оценки принимают 6 значений: от очень низкой (5) до сверхвысокой (0).

Таблица 2.10 – Характеристика масштабных факторов *Wi*

|  |  |  |
| --- | --- | --- |
| Масштабный фактор (*Wi*) | Пояснение | *Wi* |
| 1) Предсказуемость, наличие прецедентов PREC | Отражает предыдущий опыт организации в реализации проектов этого типа. Очень низкий (=5) означает отсутствие опыта. Сверхвысокий (=0) означает, что организация полностью знакома с этой прикладной областью (ранее такое уже выполняли) | 5 нет опыта |
| 2) Гибкость разработки FLEX | Отражает степень гибкости процесса разработки. Очень низкий означает, что используется заданный процесс. Сверхвысокий означает, что клиент установил только общие цели | 3 среднее |
| 3) Разрешение архитектуры / Разрешение рисков в архитектуре RESL | Отражает степень выполняемого анализа риска. Очень низкий (=5) означает малый анализ. Сверхвысокий (=0) означает полный и сквозной анализ риска | 3 -среднее |
| 4) Связность группы TEAM | Отражает, насколько хорошо разработчики группы знают друг друга и насколько удачно они совместно работают. Очень низкий (=5) означает очень трудные взаимодействия. Сверхвысокий, (=0) означает интегрированную группу, без проблем взаимодействия | 0 – один в группе, сам студент выполняет |
| 5) Зрелость процесса РМАТ | Означает зрелость процесса в организации. Вычисление этого фактора может выполняться по вопроснику СММ | 3 -среднее |
|  |  | =14 |

Сумма этих значений для разрабатываемого мобильного веб-приложения «Market place» для салонов красоты равна:

14,

поэтому конечное значение степени будет следующим:

*=*1,15.

Таблица 2.11 позволяет определить оценки этих факторов *Wi.*

Таблица 2.11 – Оценка масштабных факторов *Wi*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Масштабный фактор (*Wi*) | | Очень низкий 5 | | Низкий 4 | Номинальный 3 | Высокий 2 | Очень высокий 1 | Сверхвысокий 0 |
| 1 PRЕС | | Полностью непредсказуемый проект. Никогда такое не проектировали | | Главным образом, в значительной степени непредсказуемый | Отчасти  непредсказуемый | Большей частью  знакомый | В значительной  степени знакомый | Полностью знакомый.  Повторное использование |
| 2 FLEX | Точный, строгий процесс разработки | | Редкое расслабление в работе | | Некоторое расслабление в работе | Большей частью согласованный процесс | Некоторое согласование процесса | Заказчик определил только общие цели |
| 3 RESL | Малое разрешение риска (20%) | | Некоторое (40%) | | Частое (60%) | Большей частью (75%) | Почти всегда (90%) | Полное (100%) |
| 4 TEAM | Очень трудное взаимодействие | | Достаточно трудное взаимодействие | | Среднее  взаимодействие | Главным образом  корпоративность | Высокая  корпоративность | Безукоризненное  взаимодействие |
| 5 РМАТ | Взвешенное среднее значение от количества ответов «Yes» на вопросник СММ Maturity | | | | | | | |

Вернемся к обсуждению основного уравнения модели раннего этапа проектирования.

Множитель поправки *Мe* зависит от набора формирователей затрат *EMi*, перечисленных в таблице 2.10 и определяется по формуле 2.6.

Для каждого формирователя затрат *EMi* определяется оценка в интервале от 0.5 до 1.5, где 0.5 соответствует очень низкому значению влияния, а 1.5 – сверхвысокому значению, 1 – номинальное влияние (нет влияния). Перемножение всех множителей затрат *EMi* формирует множитель поправки:

Таблица 2.12 – Формирователи затрат *EMi* для раннего этапа проектирования

|  |  |  |
| --- | --- | --- |
| Обозначение | Название | *EMi* |
| PERS | Возможности (способности) персонала (Personnel Capability) | средние способности =1 |
| RCPX | Надежность и сложность продукта (Product Reliability and Complexity) | Среднее = 1 |
| RUSE | Требуемое повторное использование (Required Reuse)  Необходимость повторного использования | Среднее = 0.5 |
| PDIF | Трудность (сложность) платформы (Platform Difficulty) | Python, js = 1,5 |
| PREX | Опытность персонала (Personnel Experience) | Мало опытный (студент)= 1.5 |
| FСIL | Средства поддержки (Facilities)  Возможности | Среднее = 1 |
| SCED | График (Schedule)  Сроки | Сроки не жесткие = 0.5 |
| Итого: |  | =1\*1\*0.5\*1.5\*1.5\*1\*0.5=0.5625 |

Множитель поправки *Мe* может быть в диапазоне от67 = 279 936 до 1

Для мобильного веб-приложения «Market place»:

Таким образом, произведя расчет COCOMO-метрик, которые отвечают за оценку стоимости разработки веб-приложения «Market place» для салонов красоты, сделали вывод, что за шесть с половиной месяцев один человек может выполнить данный программный продукт.

Качественный анализ рисков (таблицы 2.13 – 2.20) включает классификацию выявленных рисков [7]. Качественный анализ рисков включает:

* Определение вероятности реализации рисков.
* Определение тяжести последствий реализации рисков.
* Определение ранга риска по матрице «вероятность-последствия».
* Определение близости наступления риска.
* Оценка качества использованной информации.

Таблица 2.13 – Шесть основных рисков

|  |  |  |  |
| --- | --- | --- | --- |
| Причина | Условия | Последствия | Ущерб |
| Требования не ясны | Отсутствие описания сценариев использования системы | Задержка начала разработки ПО. Большой объем переработок | Задержки сроков выпуска готовой продукции и дополнительные трудозатраты |
| Ошибка разработчиков | Программный код низкого качества | Большое количество ошибок. Высокие затраты на их исправление | Задержки сроков выпуска готовой продукции и дополнительные трудозатраты |
| Закладывание логической бомбы | Отсутствие хороших отношений в команде, кто имеет дело к проекту | Задержка начала разработки ПО. Большое число ошибок | Задержки сроков выпуска готовой продукции и дополнительные трудозатраты |
| С ростом информации ПО медленнее начинает работать | Не увеличиваются параметры сервера | Большое количество ошибок. Высокие затраты на их исправление | Задержки сроков выпуска готовой продукции и дополнительные трудозатраты |
| Плагиат | Плагиат идеи с официальных источников | Разборки в судебном порядке | Задержки сроков выпуска готовой продукции и дополнительные трудозатраты |
| Текучесть кадров | Частая смена участников | Низкая произв-сть при вводе новых участников в проект | Задержки сроков выпуска готовой продукции и дополнительные трудозатраты |

Произведем расчёт Качественного анализа рисков для мобильного веб-приложения «Market place» для салонов красоты, результаты представлены в таблицах 2.14-2.20.

Таблица 2.14 – Вероятность возникновения рисков

|  |  |  |  |
| --- | --- | --- | --- |
| Причина | Вероятность | Воздействие | Ранг |
| Требования не ясны | Очень вероятно | Катастрофические | 9 |
| Ошибка разработчиков | Очень вероятно | Критичные | 6 |
| Закладывание логической бомбы | Возможно | Критичные | 4 |
| С ростом информации маркетплейс медленнее начинает работать | Возможно | Критичные | 4 |
| Плагиат | Возможно | Критичные | 4 |
| Текучесть кадров | Возможно | Критичные | 4 |

Таблица 2.15 – Риск: ошибка разработчиков

|  |  |
| --- | --- |
| Номер: 1 | Категория: Технологический |
| Причина: Ошибка разработчиков | Симптомы: Разработчики будут использовать новые программные продукты |
| Последствия: Низкая производительность. | Воздействие: Увеличение сроков и трудоемкости разработки |
| Вероятность: Очень вероятно | Степень воздействия: Критичная |
| Близость: Очень скоро | Ранг: 6 |

Таблица 2.16 – Риск: требования не ясны

|  |  |
| --- | --- |
| Номер: 2 | Категория: Логический |
| Причина: Требования не ясны | Симптомы: Разработчики плохо ориентируются в предметной области |
| Последствия: Задержка начала разработки ПО. | Воздействие: Увеличение сроков и трудоемкости разработки |
| Вероятность: Очень вероятно | Степень воздействия: Катастрофическая |
| Близость: Очень скоро | Ранг: 9 |

Таблица 2.17 – Риск: закладывание логической бомбы

|  |  |
| --- | --- |
| Номер: 3 | Категория: Технологический |
| Причина: Закладывание логической бомбы | Симптомы: Разработчики заново анализируют весь программный продукт |
| Последствия: Задержка разработки ПО. | Воздействие: Увеличение сроков и трудоемкости разработки |
| Вероятность: возможно | Степень воздействия: Критичная |
| Близость: скоро | Ранг: 4 |

Таблица 2.18 – Риск: рост информации

|  |  |
| --- | --- |
| Номер: 4 | Категория: Технологический |
| Причина: С ростом информации маркетплейс медленнее начинает работать | Симптомы: Не увеличиваются параметры сервера |
| Последствия: Задержка разработки. Большие затраты на их исправление. | Воздействие: Увеличение сроков и трудоемкости разработки |
| Вероятность: возможно | Степень воздействия: Критичная |
| Близость: скоро | Ранг: 4 |

Таблица 2.19 – Риск: плагиат

|  |  |
| --- | --- |
| Номер: 5 | Категория: Функциональный |
| Причина: Плагиат | Симптомы: Реализация не своих идей |
| Последствия: Разборки в судебном порядке | Воздействие: Увеличение сроков и трудоемкости разработки |
| Вероятность: возможно | Степень воздействия: Критичная |
| Близость: не скоро | Ранг: 4 |

Таблица 2.20 – Риск: Текучесть кадров

|  |  |
| --- | --- |
| Номер: 6 | Категория: Технологический |
| Причина: Текучесть кадров | Симптомы: Частая смена участников |
| Последствия: Низкая производительность при вводе новых участников в проект. | Воздействие: Увеличение сроков и трудоемкости разработки |
| Вероятность: возможно | Степень воздействия: Критичная |
| Близость: скоро | Ранг: 4 |

Таким образом мы рассмотрели основные риски, а именно определили вероятности реализации рисков, тяжесть последствий, ранг риска по матрице «вероятность-последствия», определили близость наступления риска, дали оценку качества, к использованной информации, которые могут возникнуть при создании мобильного веб-приложения «Market place» для салонов красоты.

## **2.4 Вывод по второму разделу**

В данном разделе был выполнен аналитический обзор требований при разработке мобильного веб-приложения «Market place» для салонов красоты на основе диаграммы прецедентов, который заключался в определении главных ролей, актеров, прецедентов и их взаимодействии, а также в построении самой диаграммы прецедентов.

Был выполнен анализ структурных принципов проектирования приложения и модели предметной области мобильного веб-приложения «Market place» для салонов красоты, на основе которых, были построены диаграммы классов и последовательности, и определены главные функциональные задачи. Осуществлены расчеты функционально- и размерно-ориентированных метрик для мобильного веб-приложения «Market place» (FP, LOC, COCOMO) и выполнен анализ рисков при его разработке.

# **3 Программная реализация мобильного веб-приложения**

## **3.1 Взаимодействие мобильного веб-приложения на основе выбора базы данных**

Для реализации мобильного веб-приложения «Market place» для салонов красоты использовалась база данных (далее БД) Postgres. Postgres – это не просто реляционная, а объектно-реляционная СУБД. Главной особенностью данной БД является ее фундаментальная поддержка настраиваемых объектов и их поведения, включая типы данных, функции, операции, домены и индексы. Postgres поддерживает обширный список данных. В дополнение к числовым, с плавающей точкой, текстовым, логическим и другим типам данных и их многочисленным вариациям, данная БД поддерживает uuid (стандарт идентификации), денежные, перечисляемые, геометрические, двоичные типы, сетевые адреса, битовые строки, текстовый поиск, xml, json, массивы, а также некоторые внутренние типы для идентификации объектов. Данная БД уже была спроектирована другими разработчиками и загружена, как проект на внешние носители, откуда была использована.

Для взаимодействия приложения с БД используется ORM (Object Relation Mapping) или объектно-реляционное отображение. Этот метод программирования, служит для преобразования несовместимых типов данных в объектно-ориентированные языки программирования. Суть данного метода заключается в том, что создается некая абстракция – «виртуальная объектная база», запросы которой, преобразуется в SQL команды. ORM фреймворк может быть написан на каком-либо объектно-ориентированном языке (PHP, Python, Ruby). Классы будут соответствовать таблицам в базе, а экземпляры этих классов – конкретным строкам таблицы. SQLAlchemy – это библиотека на языке Python для работы с реляционными СУБД с применением технологии ORM. Служит для синхронизации объектов Python и записей реляционной БД. SQLAlchemy позволяет описывать структуры баз данных и способы взаимодействия с ними языке Python без использования SQL.

Разрабатываемое мобильное веб-приложение основано на JSON файлах. Название JSON означает JavaScript Object Notation (представление объектов JavaScript). Использование именно формата json удобно тем, что информация в данном формате – это массивы и объекты Javascript, к которым можно легко получить доступ с клиентской стороны веб-приложения. Любая база данных включает в себя набор функций для записи, чтения, обновления и удаления данных из таблиц. JSON имеет следующие преимущества: он компактен, его предложения легко составлять и читать как человеком, так и компьютером и его легко преобразовать в структуру данных для большинства языков программирования (числа, строки, логические переменные, массивы и так далее). Многие языки программирования имеют функции и библиотеки для чтения и создания структур JSON.

В нашей разработке мобильного веб-приложения используется одна из распространенных форм JSON – пересылка данных от сервера к браузеру. Обычно данные JSON доставляются с помощью [AJAX](http://en.wikipedia.org/wiki/Ajax_%28programming%29), который позволяет обмениваться данными браузеру и серверу без необходимости перезагружать страницу. К примеру, в разрабатываемом мобильном веб-приложении пользователь нажимает миниатюру продукта, услуги в мобильном приложении, магазине. JavaScript, выполняющийся на браузере, генерирует запрос AJAX к скрипту PHP, запущенному на сервере, передавая ID выбранного продукта. Скрипт PHP получает название продукта, описание, цену и другую информацию из базы данных. Затем составляет из данных строку JSON и отсылает ее браузеру. JavaScript, выполняющийся на браузере, получает строку JSON, декодирует ее и выводит информацию о продукте на странице для пользователя.

В Мобильном веб-приложении «Market place» для салонов красоты был использован фреймворк React Native, что облегчает написание приложения с рендерингом для мобильных устройств. Фреймворк основан на React, библиотеке JavaScript, которую Facebook использует для разработки пользовательских интерфейсов. Благодаря данному инструменту появилась возможность писать одновременно для нескольких платформ, поскольку написанный код может совместно использоваться между платформами.

## **3.2 Интерфейс мобильного веб-приложения «Market place»**

Используя выбранное программное обеспечение, которое мы подробно рассмотрели ранее, рассмотрим интерфейс [11] для мобильного веб-приложения «Market place» для салонов красоты.

Для реализации интерфейса учитывались цвета, формы, пропорции, когнитивная психология [6]. Однако, несколько принципов, которые использовались в данной разработке все же отмечу:

*Цвет*. Цвета делятся на теплые (желтый, оранжевый, красный), холодные (синий, зеленый), нейтральные (серый). Для разрабатываемого нами мобильного веб-приложения использовались теплые цвета

*Форма*. При разработке веб-приложения использовался прямоугольник со скругленными углами. Или круг. Полностью прямоугольные ЭИ потеряли свою эстетику и остаются историей в книгах по дизайну.

*Иконки*. Иконки в приложении очевидны, сделаны не слишком маленькими иначе по ним было бы очень трудно попасть.

*Привычки пользователя*. При реализации интерфейса к мобильному веб-приложению «Market place» для салонов красоты, учитывали привычки пользователя т.е. интерфейс должен иметь как можно больше аналогий, с известными пользователю вещами.

*Система должна отвечать*. В мобильном веб-приложении «Market place», была выполнена анимация клика т.е. система отзывается, на нажатие той или иной кнопки.

Интерфейс мобильного веб-приложения «Market place» для салонов красоты, можно увидеть на рисунках 3.1 – 3.8.

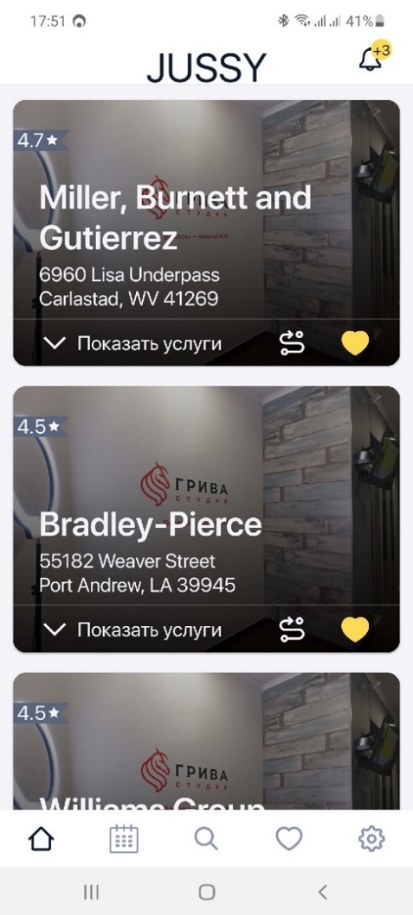


Рисунок 3.1 – Интерфейс главной страницы мобильного веб-приложения

Главная страница приложения обозначается иконкой в виде домика, что обозначает домашнюю страницу, которая находится в нижней левой части экрана. На главной странице отображается краткая информация обо всех салонах.

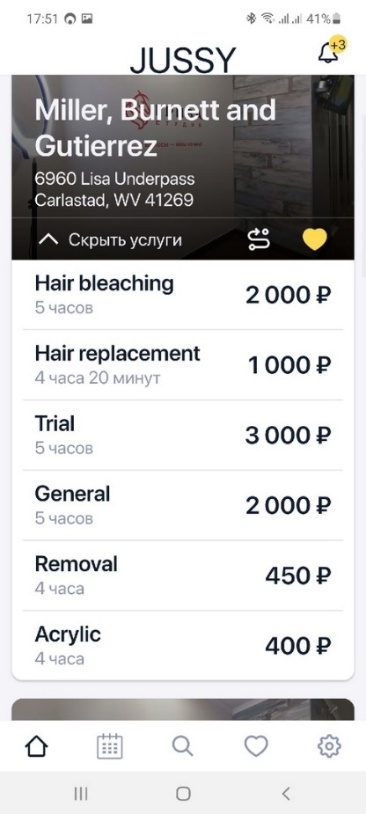


Рисунок 3.2 – Интерфейс раскрывающиего списка любого салона

На главной странице приложения для ознакомительного просмотра прайс-листа каждого салона существует выпадающий список.

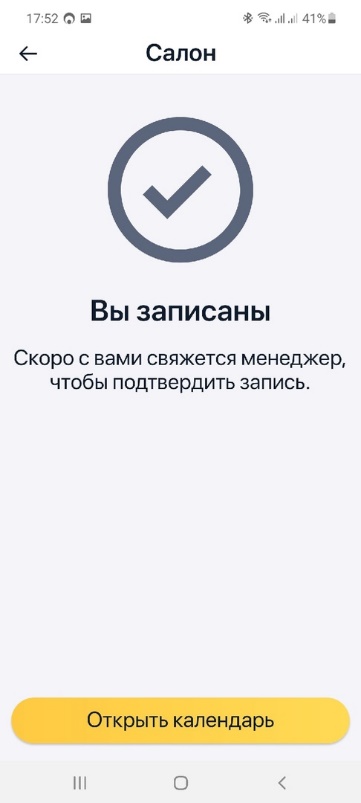
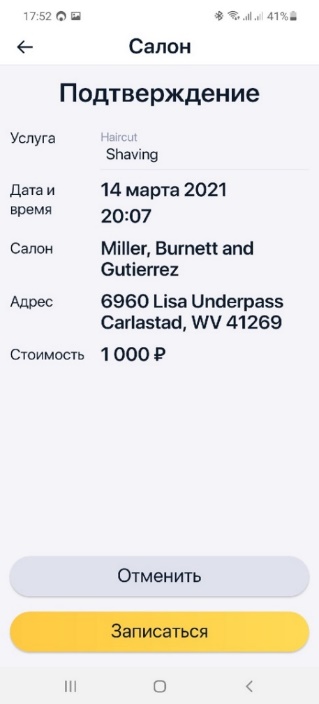
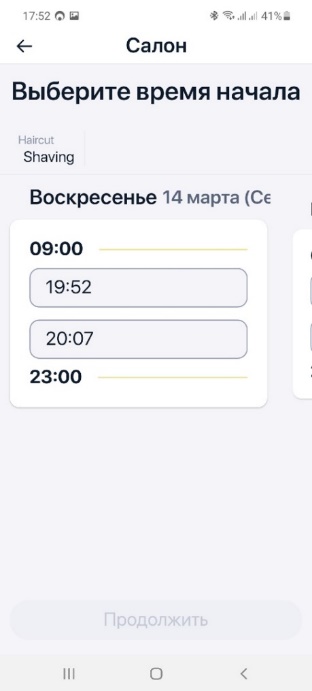
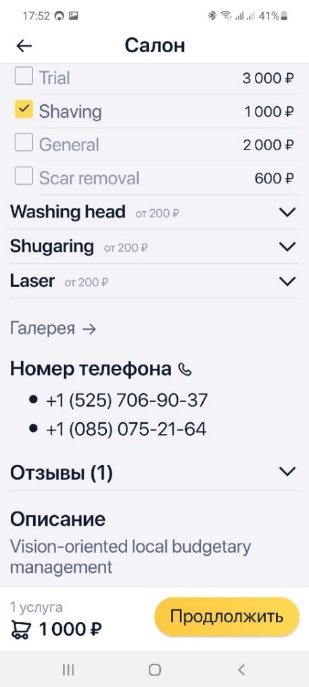


Рисунок 3.3 – Интерфейс записи в салон

Для записи в салон, пользователю необходимо его выбрать. Откроется подробное описание салона, с полным прайс-листом, галереей, контактными данными и отзывами. Для продолжения записи необходимо пользователю выбрать услуги, предоставляемые салоном, нажать кнопку «Продолжить», тем самым откроется новая форма, на которой пользователю следует выбрать свободную дату и время для записи. После подтверждения пользователь может посмотреть отчет о записи в салон и проверить, все ли ему подходит. Если все подходит, то он нажимает кнопку «Записаться», в противном случае – «Отменить». При успешной записи открывается новая форма для перехода на страницу «Календарь».

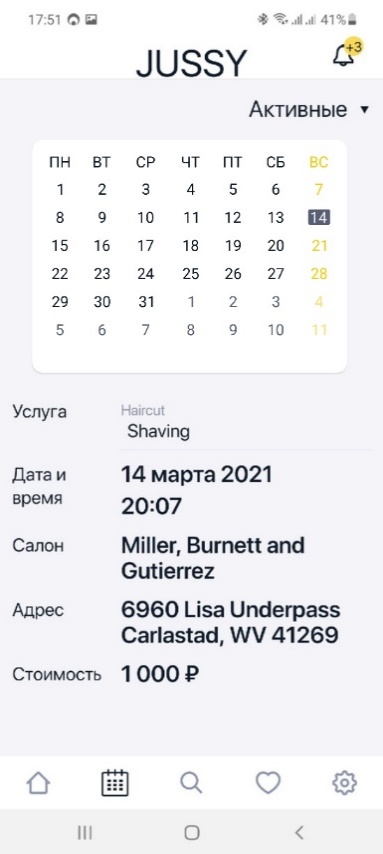


Рисунок 3.4 – Интерфейс календаря

Страница «Календарь» отображается иконкой в виде календарика, она следует сразу после главной страницы в панели управления. На данной странице пользователю доступны все активные записи, которые он осуществил.

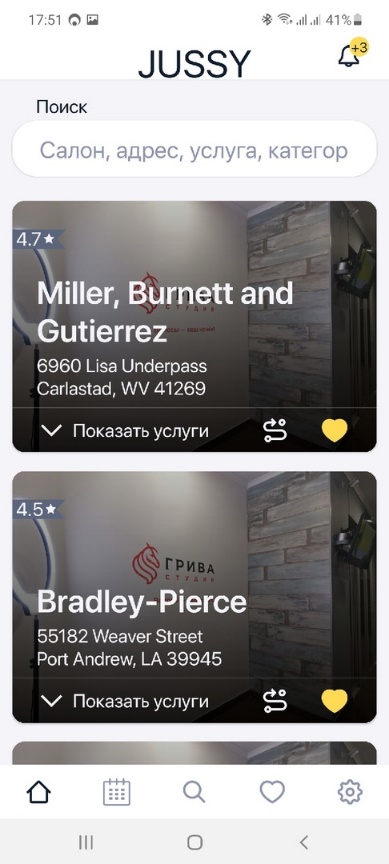


Рисунок 3.5 – Интерфейс поиска салона по категориям

Страница «Поиск салона по категориям» отображается иконкой в виде лупы, что символично обозначает поиск. Данная иконка находится в центре панели управления приложения. На данной странице, пользователь имеет возможность найти интересующие его салоны по определенным категориям.

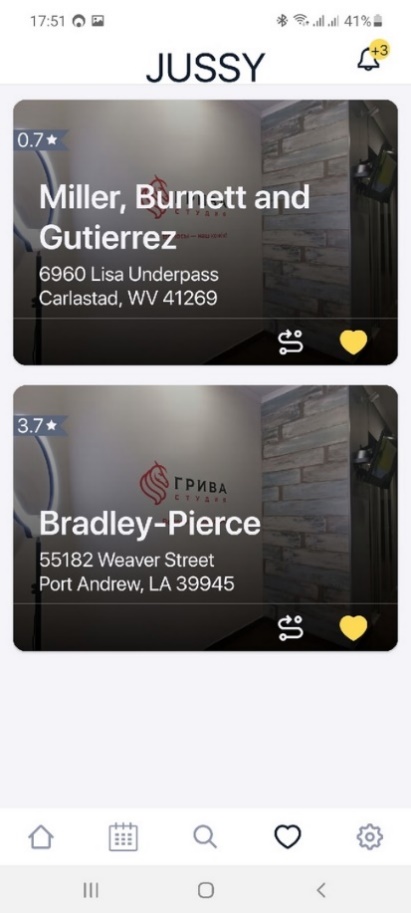


Рисунок 3.6 – Интерфейс меню салонов с отметкой «Нравится»

На панели управления приложения есть страница для салонов с отметкой «Нравится», данная отметка изображается в виде закрашенного сердечка, она существует для того, чтобы пользователь не тратил время на поиск салонов, которые ему понравились, поэтому они добавлены в отдельный раздел.

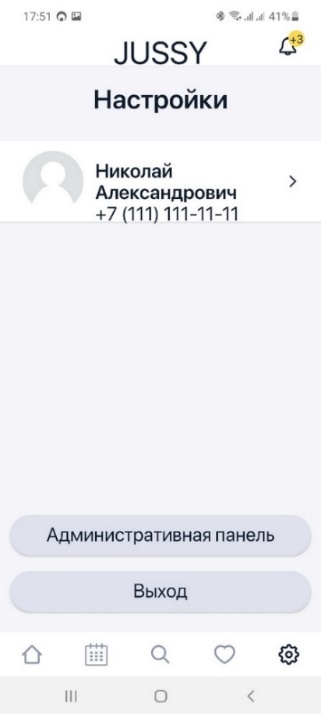


Рисунок 3.7 – Интерфейс личного кабинета

Страница «личный кабинет» находится на вкладке «Настройки», которая символично изображается иконкой в виде шестеренки. В данном разделе клиент может добавить фотографию в профиль, изменить при необходимости номер телефона, имя и фамилию или же совсем выйти из приложения.

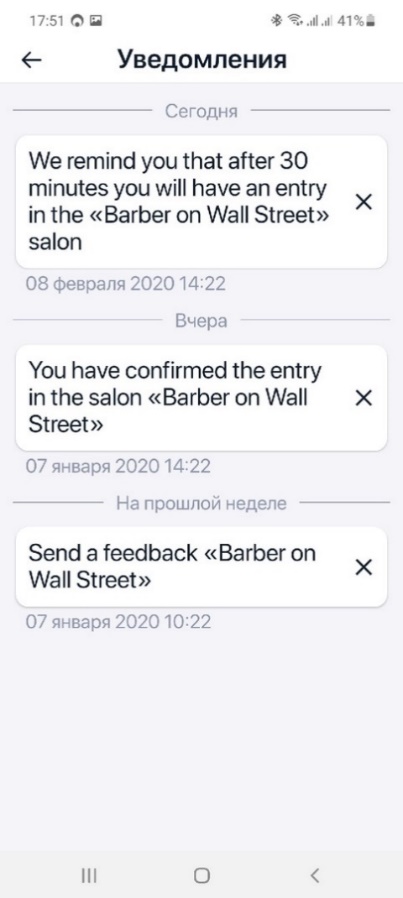


Рисунок 3.8 – Интерфейс уведомлений

Страница «Уведомления» мобильного приложения отображается иконкой в виде колокольчика в верхнем правом углу приложения. На данной странице отображаются все уведомления, которые связаны с записью в салон, напоминании о записи.

Таким образом, было разработано мобильное веб-приложение «Market place» для салонов красоты. В данном подразделе были приведены примеры интерфейсы всех главных функций приложения их и описание.

## **3.3 Руководство пользователя к мобильному веб-приложению «Market place» для салонов красоты**

Для запуска мобильного веб-приложения «Market place» нажмите на рабочем столе смартфона на его иконку (рисунок 3.9).

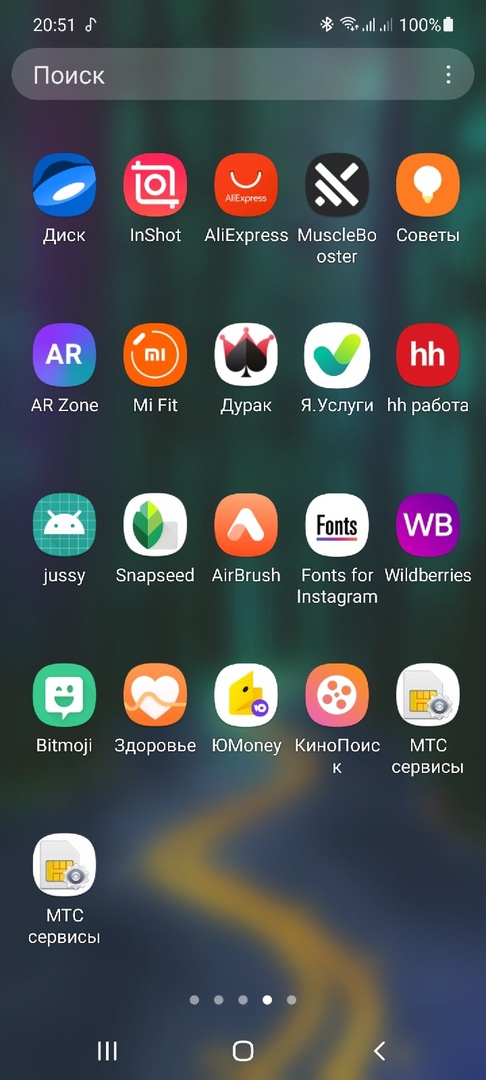


Рисунок 3.9 – Иконка мобильно веб-приложения

На экране появится страница «Авторизация» (рисунок 3.10), которая требует регистрацию/авторизацию в мобильном веб-приложении «Market place» для салонов красоты. Введите требуемые данные.

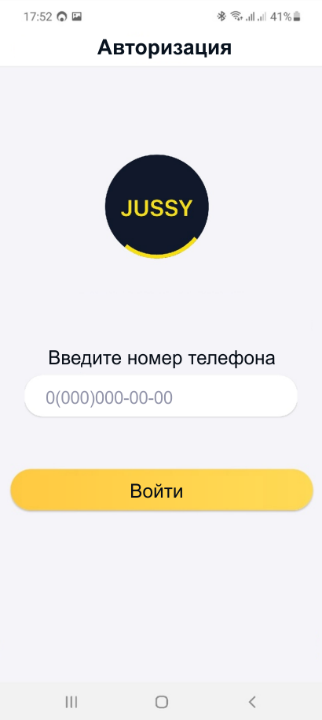


Рисунок 3.10 – Страница «Авторизация»

Откроется «Главная страница» мобильного веб-приложения (рисунок 3.11). В нижней панели экрана выберите кнопку «Настройки» (рисунок 3.12). Заполните личный профиль, сохраните (рисунок 3.13).

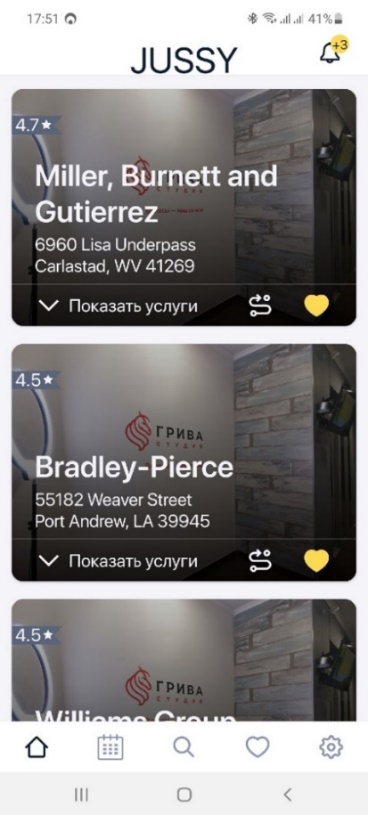


Рисунок 3.11 – Страница «Главная страница»

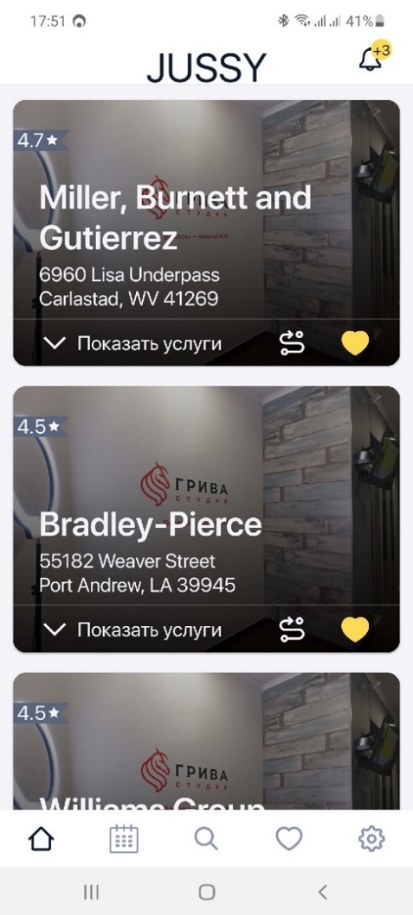


Рисунок 3.12 – Кнопка «Настройки»

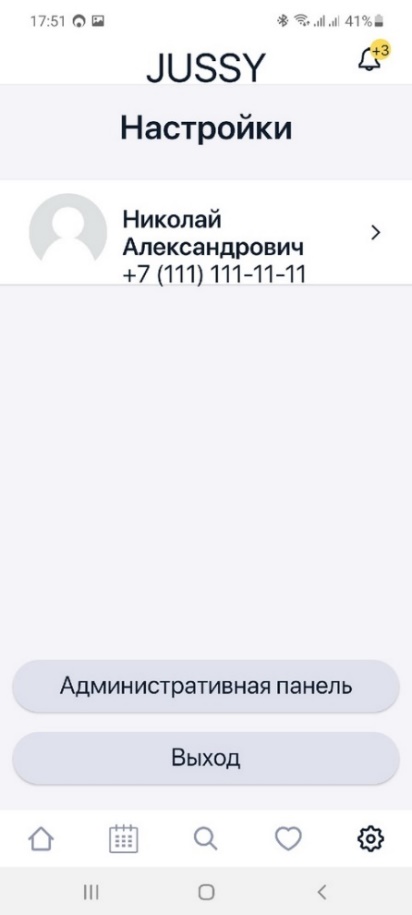


Рисунок 3.13 – Страница «Личный кабинет»

Нажмите в нижней панели экрана кнопку «Поиск» (рисунок 3.14). В текстовой строке введите данные для поиска салона (название, адрес, услуга), в который хотите записаться или же выберите понравившийся салон из предложенного списка ниже (рисунок 3.15).

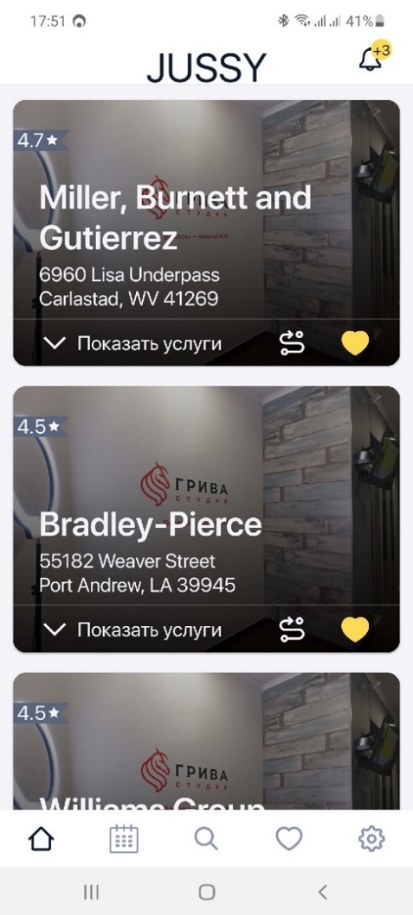


Рисунок 3.14 – Кнопка «Поиск»

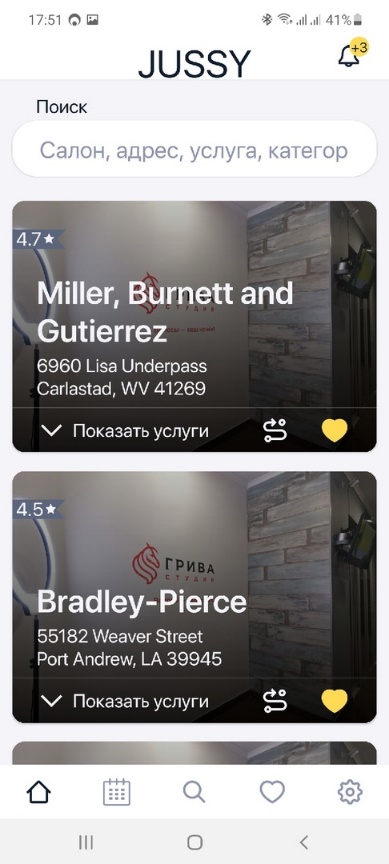


Рисунок 3.15 – Список салонов

Для просмотра краткой информации салона нажмите на расположенную на картинке салона кнопку «Раскрывающейся список» (рисунок 3.16). Откройся прайс-лист салона (рисунок 3.17).

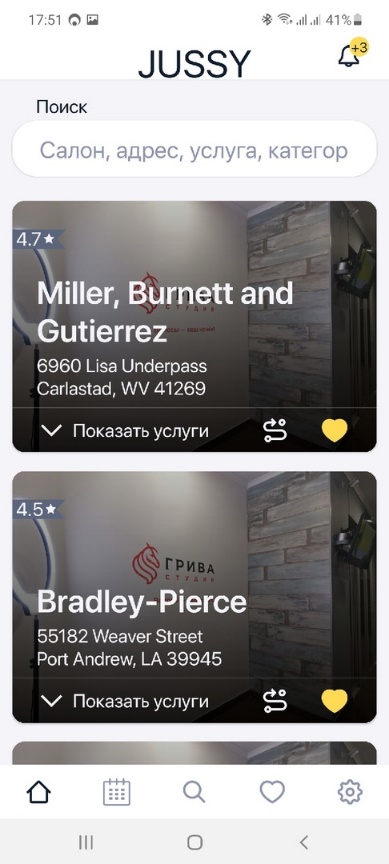


Рисунок 3.16 – Кнопка «Раскрывающейся список»

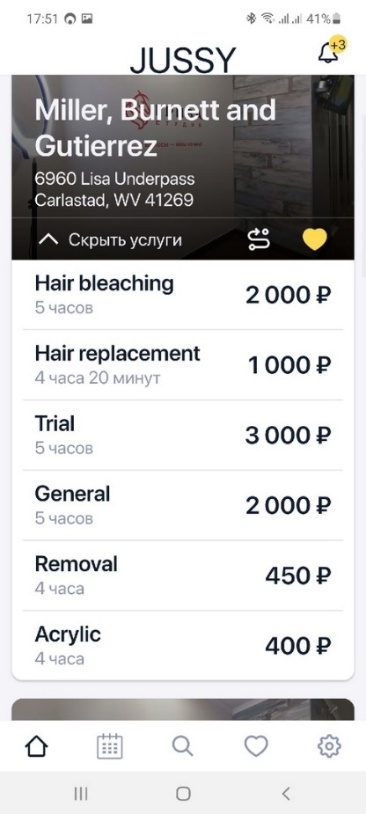


Рисунок 3.17 – Прайс-лист салона

Для просмотра местоположения салона нажмите на расположенную на картинке салона кнопку «Построение маршрута» (рисунок 3.18). Откроется страница «Googlemaps» (рисунок 3.19).

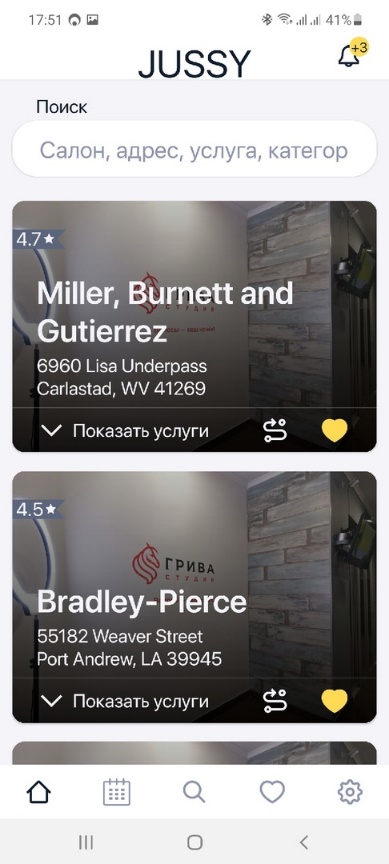


Рисунок 3.18 - Кнопка «Построение маршрута»

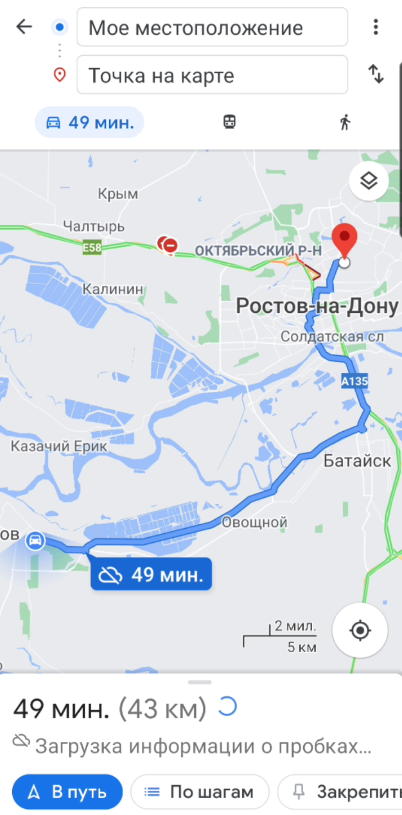


Рисунок 3.19 – Страница «Googlemaps»

Для выделения салона в категорию «Нравится» нажмите на расположенную на картинке салона кнопку «Нравится» (рисунок 3.20). Тогда данный салон отправится в категорию понравившихся. Для того, чтобы открыть список понравившихся салонов нажмите на кнопку «Нравится» в нижней панели экрана. Откроется страница отмеченных салонов (рисунок 3.21). Выберите салон.

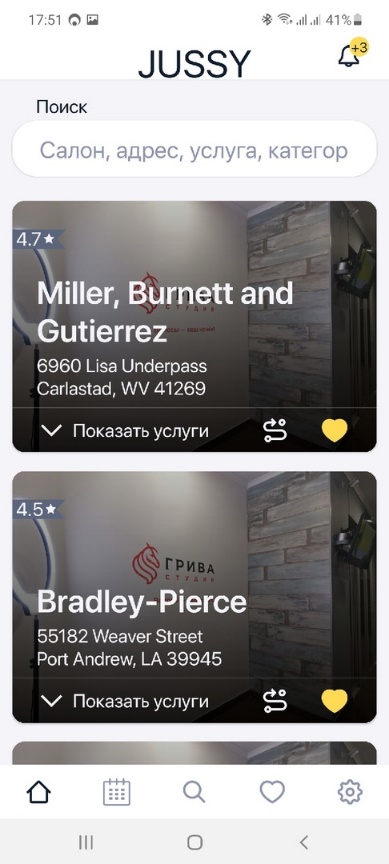


Рисунок 3.20 – Кнопка «Нравится»

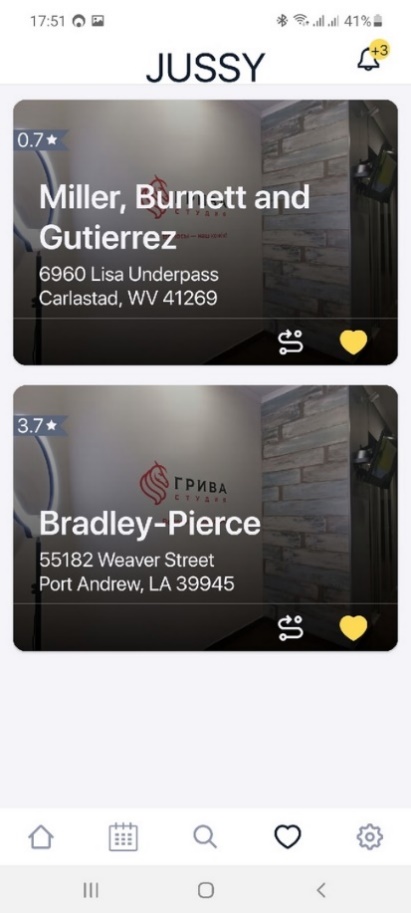


Рисунок 3.21 – Страница понравившихся салонов

После выбора салона на экране смартфона откроется страница «Салон» (рисунок 3.22). На этой странице доступна вся информация о салоне (рейтинг, описание, контактные данные, местоположение, галерея, услуги, отзывы). Выберите интересующие услуги, в нижней части экрана после выбора услуг ведется подсчет цены. По окончании выбора услуг нажмите кнопку «Продолжить» (рисунок 3.23).

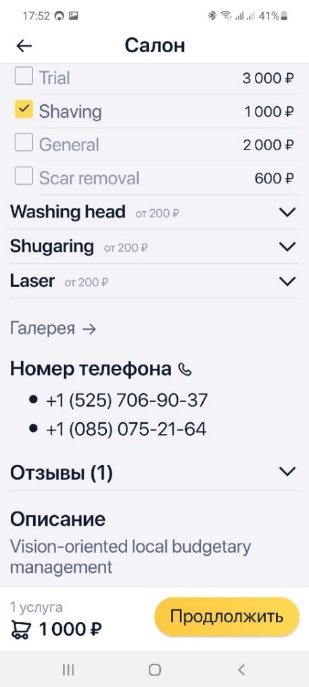


Рисунок 3.22 – Страница «Салон»

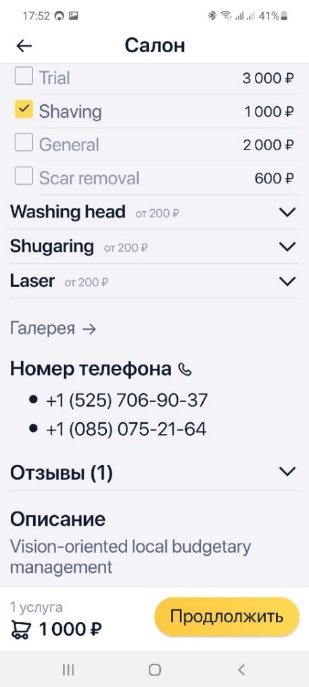


Рисунок 3.23 – Кнопка «Продолжить»

Откроется станица «Запись» (рисунок 3.24). Выберите предложенную дату и время. Нажмите на кнопку «Продолжить».

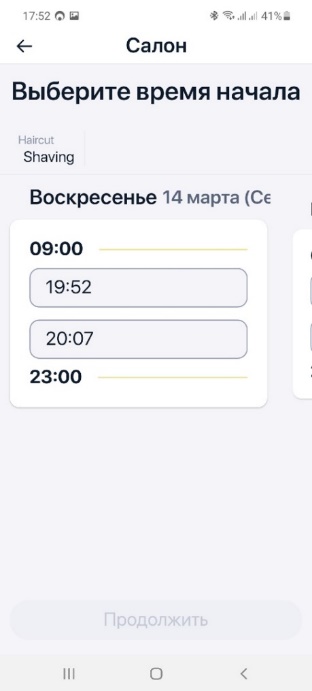


Рисунок 3.24 – Страница «Запись»

Откроется станица «Подтверждение» (рисунок 3.25). На данной странице предоставлена вся информация о записи в салон (название, адрес, услуги, дата и время, стоимость). Для подтверждения нажмите кнопку «Записаться» (рисунок 3.26), для отмены записи кнопку «Отменить» (рисунок 3.27).

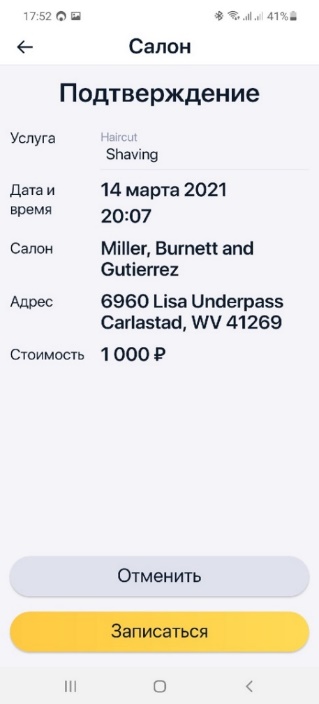


Рисунок 3.25 – Страница «Подтверждение»

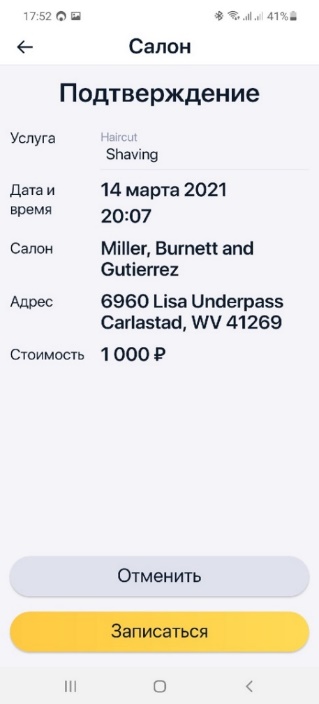


Рисунок 3.26 – Кнопка «Записаться»

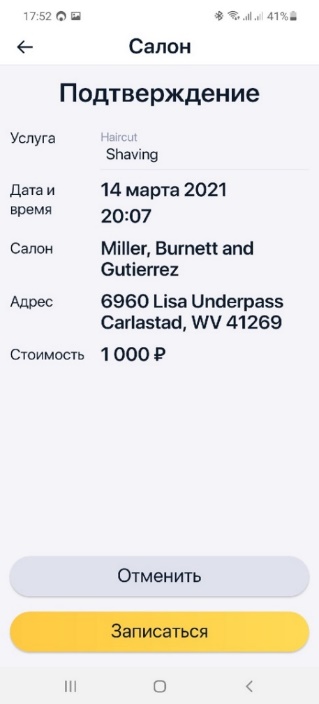


Рисунок 3.27 – Кнопка «Отменить»

При успешной записи откроется страница «Подтверждение записи» (рисунок 3.28). В нижней части экрана нажмите кнопку «Открыть календарь» (рисунок 3.29).

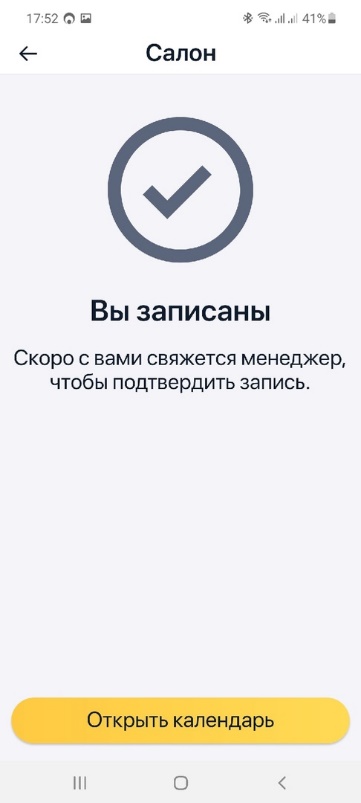


Рисунок 3.28 – Страница «Подтверждение записи»

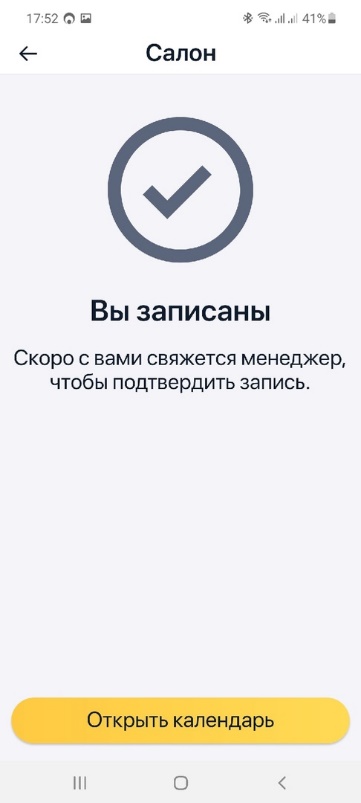


Рисунок 3.29 – Кнопка «Календарь»

Откроется страница «Календарь» (рисунок 3.30), где можно увидеть отмеченную дату записи в салон.

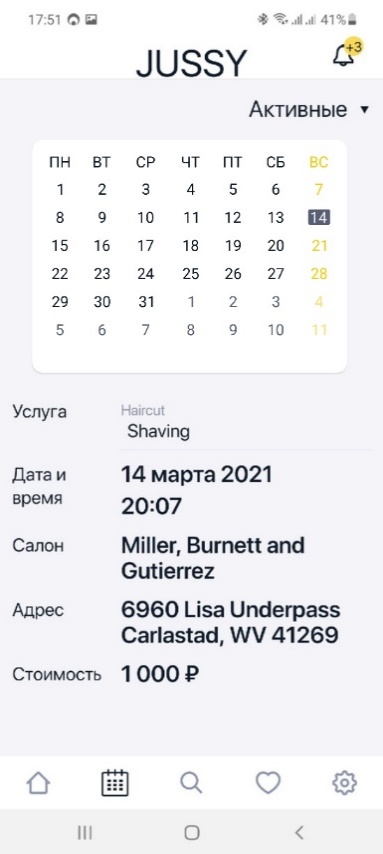


Рисунок 3.30 – Страница «Календарь»

Для того, чтобы открыть страницу «Календарь» в нижней панели экрана нажмите на кнопку «Календарь» рисунок (3.31).

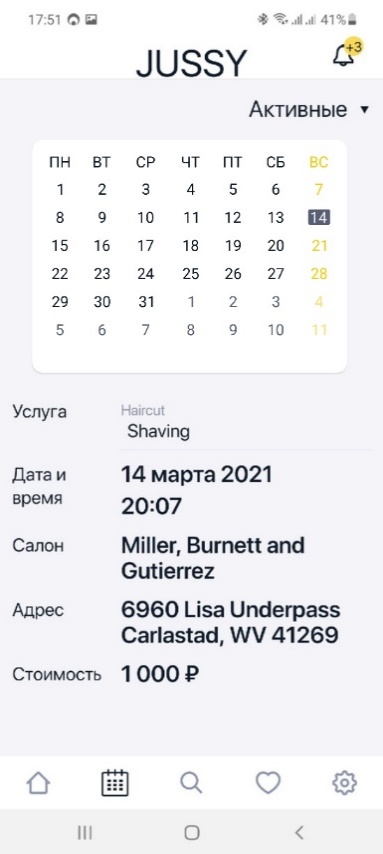


Рисунок 3.31 – Страница «Календарь»

После подтверждения придет уведомление об успешной записи в салон. Все уведомления можно посмотреть, нажав на кнопку «Уведомления» (рисунок 3.32), находящаяся в верхнем правом углу.

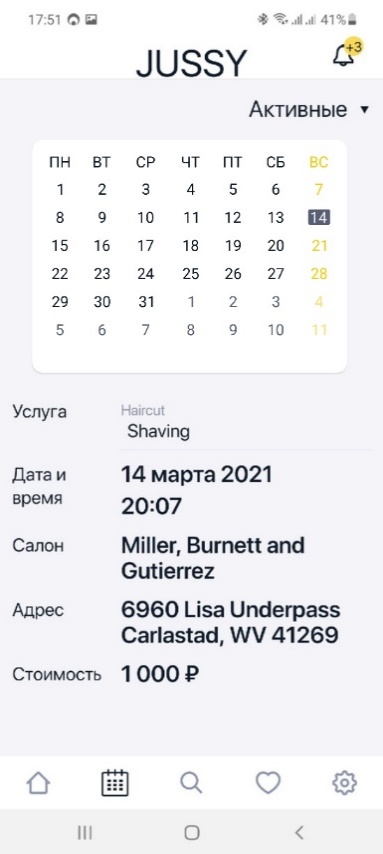


Рисунок 3.32 – Кнопка «Уведомления»

## **3.4 Вывод по третьему разделу**

В данном разделе рассмотрели взаимодействие БД с приложением. Для реализации мобильного веб-приложения «Market place» для салонов красоты использовалась БД Postgres, которая является объектно-реляционной СУБД. Для взаимодействия приложения с БД используется ORM – методика, предназначенная для преобразования несовместимых типов данных в объектно-ориентированные языки программирования. В разработке приложения использовалась библиотека на языке Python – SQLAlchemy, которая нужна для работы с реляционными СУБД. Технология ORM служит для синхронизации объектов Python и записей реляционной БД. Разрабатываемое мобильное веб-приложение основано на JSON файлах. В мобильном веб-приложении «Market place» для салонов красоты был использован фреймворк React Native, что облегчает написание приложения с рендерингом для мобильных устройств.

Разработали интерфейс мобильного веб-приложения «Market place» для салонов красоты и написали руководство пользователя к данному приложению.

# **Заключение**

В результате выполнения выпускной квалификационной работы было разработано мобильное веб-приложение «Market place» для салонов красоты. Данная разработка автоматизирует процессы поиска и записи в выбранный салон и вывод результатов (оповещение о записи).

В процессе написания выпускной квалификационной работы были выполнены следующие задачи:

* анализ предметной области. Рассмотрены преимущества и особенности применения понятия «Маркетплейс». проведено обоснование проектных решений по видам обеспечения, в результате которых для реализации проекта были использованы средства разработки и моделирования;
* спроектирована мобильное веб-приложение на основе диаграмм прецедентов, классов, последовательностей. Рассчитаны функционально-ориентированные метрики и LOC-метрики с анализом рисков разрабатываемого мобильного веб-приложения;
* Реализовано мобильное веб-приложение. Применены теоретические и практические методы системного проектирования сложных программных средств и построения моделей для создания мобильного веб-приложения «Market place» для салонов красоты.

Подводя итоги написания выпускной квалификационной работы, можно сказать, что все поставленные задачи решены, следовательно, главная цель, поставленная в выпускной квалификационной работе, также достигнута.

# **Список используемых источников**

1. **Буч Г,** Язык UML. Руководство пользователя. / Г. Буч, Дж. Рамбо, И. Якобсон – М.: Форум, 2018. – 496 с.
2. **Братигенко** **В.В,** Проектирование информационных систем. / В.В. Братигенко – Иркутск: БГУЭП, 2019. – 84 с.
3. **Герберт Шилдт,** Java 8. Руководство для начинающих. /Г. Шилдт – Москва: Вильямс, 2018. – 720 с.
4. **Джеффри Паркер,** Революция платформ. Как сетевые рынки меняют экономику – и как заставить их работать на вас. / Дж. Паркер, М. Альстин, С. Чаудари – М.: МИФ, 2017. – 304 с.
5. **Жуков В. В,** Технологии программирования: учебно-методическое пособие для курсовой работы / В.В. Жуков – Ростов н/Д.: ФГБОУ ВО РГУПС, 2017 – 27с.
6. **Лебедев Артемий,** Коководство. Четвертое издание. / Артемий Лебедев – М.: Студия Артемия Лебедева, 2019. -536 с.
7. **Орлов С.** Технологии разработки программного обеспечения: Учебник. / С. Орлов – СПб.: Питер, 2020. – 464 с.
8. **Патрик Нимейер,** Программирование на Java. / А.М. Кудан – Москва: Эксмо, 2018. – 1214 с.
9. **Роберт Мартин,** Чистый код. / Р. Мартин – СПб.: БХВ-Санкт-Петербург, 2018. – 464 с.
10. **Тучкевич Е.И,** Adobe Photoshop CC 2019. Мастер-класс. / Е. И. Тучкевич – Спб.: БХВ-Санкт-Петербург, 2020. – 496 с.
11. **Унгер Расс,** UX-дизайн. Практическое руководство по проектированию опыта взаимодействия. / Унгер Расс, Чандлер Кэролайн. – Спб.: Символ-Плюс, 2019. – 336 с.

# **Приложение А (обязательное)**

## Листинг программного кода.

Страницы 63-87

import { field, string, succeed } from "jsonous";

import { assocPath } from "ramda";

import AsyncStorage from "@react-native-community/async-storage";

import { momentFromServerTimestampDecoder } from

1. CodeCoverage – это метрика тестирования ПО, для выявления процента исходного кода программы [↑](#footnote-ref-1)
2. Kubernetes – проект с исходным кодом, для управления кластером контейнеров Linux как единой системой [↑](#footnote-ref-2)
3. Прецедент – это функциональность системы, позволяющая пользователю получить некий значимый для него, ощутимый и измеримый результат. [↑](#footnote-ref-3)