**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет»**

**Институт высоких технологий и пьезотехники**

**Кафедра прикладной информатики и инноватики**

**Луценко Константин Сергеевич**

**Мобильное приложение для управления клиентской базой специалистов индустрии красоты**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ**

**РАБОТА БАКАЛАВРА**

**по направлению 09.03.03 – Прикладная информатика**

**Научный руководитель –**

**профессор кафедры информационных и измерительных технологий**

**Яценко Д.В.**

**Ростов-на-Дону – 2025**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**

**ИНСТИТУТ ВЫСОКИХ ТЕХНОЛОГИЙ И ПЬЕЗОТЕХНИКИ**

**Кафедра прикладной информатики и инноватики**

**З А Д А Н И Е**

**на выпускную квалификационную работу**

**Студент** гр. 4–7 Луценко К.С.

**Тема:** Мобильное приложение для управления клиентской базой специалистов индустрии красоты

**Срок сдачи законченной работы** \_.\_.2025

**Исходные данные:**

Государственные стандарты

1. ГОСТ 7.1–2003 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления»;
2. ГОСТ 7.82–2001 «Библиографическая запись. Библиографическое описание электронных ресурсов»;
3. ГОСТ 19.001–77. Общие положения;
4. ГОСТ 19.781–90. Термины и определения;
5. ГОСТ 19.701–90 (ИСО 5807–85). Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения;
6. ГОСТ 19.103–77. Обозначения программ и программных документов;
7. ГОСТ 19.101–77. Виды программ и программных документов;
8. ГОСТ 19.106–78. Требования к программным документам, выполненным печатным способом;
9. ГОСТ 19.104–78. Основные надписи;
10. ГОСТ 19.301–79. Программа и методика испытаний. Требования к содержанию и оформлению;
11. ГОСТ 19.402–78. Описание программы;
12. ГОСТ 19.401–78. Текст программы. Требования к содержанию и оформлению;
13. ГОСТ 19.504–79. Руководство программиста. Требования к содержанию и оформлению;
14. ГОСТ 19.505–79. Руководство оператора. Требования к содержанию и оформлению;
15. ГОСТ 19.202–78. Спецификация. Требования к содержанию и оформлению.
16. Основное требование – разработать мобильное приложение, выполняющее основные требуемые функции.

***5. Перечень вопросов, подлежащих разработке:***

1. *Анализ предметной области.*
2. *Обоснование технических решений.*
3. *Проектирование элементов системы.*
4. *Реализация компонентов системы.*
5. *Разработка методики тестирования.*

***6. Перечень графического материала:***

*Подготовка графических материалов для презентации работы*

***7. Консультанты по работе:***

***8. Дата выдачи задания: \_\_.\_\_.2025***

***9. Руководитель Яценко Д.В.***

*Подпись ФИО*

***Задание принято к исполнению***

*Дата Подпись студента*

АННОТАЦИЯ

СОДЕРЖАНИЕ

[ГЛАВА 1. АНАЛИЗ, ОБЗОР РЕШЕНИЙ, ПОСТАНОВКА ЗАДАЧИ 7](#_Toc184494102)

[1.1 Первичная постановка задачи 7](#_Toc184494103)

[1.2 Декомпозиция задачи 7](#_Toc184494104)

[1.3. Обзор существующих аналогов 8](#_Toc184494105)

[1.4 Требования к системе 12](#_Toc184494106)

[1.5 Обзор решений 12](#_Toc184494107)

[1.5.1 Выбор средств для мобильной разработки 12](#_Toc184494108)

[1.5.2 Выбор решений серверной части 16](#_Toc184494109)

[1.5.3 Выбор сред разработки 19](#_Toc184494110)

[1.5.4 Выбор системы управления базами данных 20](#_Toc184494111)

[1.6. Выводы по главе 21](#_Toc184494112)

ВВЕДЕНИЕ

На сегодняшний день информационные технологии плотно проникли практически во все сферы человеческой деятельности, что привело к постоянно растущей потребности в разработке все более новых, уникальных и сложных информационных систем.

Если рассматривать процедуру работы предпринимателей, осуществляющих взаимодействие с клиентской базой, то можно заметить, что раньше достаточное количество времени затрачивалось на ведение различного рода документации, одной из которых, например, является журнал записи. Ранее требовалось найти способ установить связь с человеком, при условии того, что не все используют соцсети и мессенджеры, а затем проследить за правильностью записи и ее очередностью. Сегодня же, процесс стал значительно проще, так как появились средства автоматизации, которые позволяют снять огромную «ручную» работу с мастера и передать ее вычислительной системе [1].

С переносом подобных рутинных процессов в информационные системы решается множество проблем ведения бумажной записи: человеческие ошибки, результатом которых является наложение записей двух разных клиентов друг на друга или подобные; трудозатраты на осуществление и выбор удобного варианта записи; необходимость напоминаний многим из клиентов о том, когда и на какую процедуру они записаны.

Актуальность темы дипломной работы обусловлена активным ростом применения программных средств взаимодействия с клиентской базой. Такое мобильное приложение стало бы значимой альтернативой платным сервисам, которые трудны и недоступны самозанятым и клиентам, позволив первым упростить работу, а вторым ускорить и облегчить нахождение нужного мастера и запись к нему.

ГЛАВА 1. АНАЛИЗ, ОБЗОР РЕШЕНИЙ, ПОСТАНОВКА ЗАДАЧИ

1.1 Первичная постановка задачи

Задачей выпускной квалификационной работы является разработка мобильного приложения для работы с клиентской базой мастеров индустрии красоты. Итогом работы должно стать приложение под IOS/Android, призванное облегчить взаимодействие между мастером и клиентом, а также снять немалую нагрузку с мастера и автоматизировать рутинные задачи, позволив предпринимателю увеличить охват клиентов и оказать как можно больше услуг.

Приложение должно позволять:

* Клиентам: найти требуемую услугу, подходящего мастера и записаться к нему, предварительно оценив местоположение и отзывы об услуге. Получить уведомление об услуге заранее. Также должна быть возможность связаться с мастером непосредственно внутри приложения, а также инструменты управления записью.
* Мастерам: контролировать записывающихся клиентов, взаимодействовать с ними. Собирать статистику и отчетность об оказанных услугах. Должны существовать возможности продвижения для повышения заинтересованности клиентов.

Основная идея – создать простое, понятное и доступное для каждого кроссплатформенное приложение под любой смартфон. Необходимо обеспечить простоту интерфейса и достаточное число инструментов для удобства как клиентов, так и мастеров.

1.2 Декомпозиция задачи

Выделенную ранее задачу можно разделить на следующие этапы:

1. Поиск аналогов, исследование их достоинств и недостатков, взятие на заметку некоторых идей по реализации преимуществ и нейтрализации неудобств.
2. Анализ основных требований к приложению.
3. Проектирование мобильного приложения, разработка требований к нему.
4. Проектирование серверной архитектуры, разработка требований к API.
5. Разработка программного кода двух принципиальных частей системы.
6. Тестирование мобильной и серверной части, а также их взаимодействия. Устранение возникших проблем.
7. Демонстрация результатов, проверка на выполнение всех требований.
8. Развертывание серверной части, тестирование на целевой аудитории.

1.3. Обзор существующих аналогов

Первоочередным действием при начале разработки информационной системы является обзор и анализ конкурентов. Проведение данного этапа позволяет учесть все недостатки похожих приложений, а наиболее удачные реализации функций улучшить и внедрить и в свой продукт. Таким образом, достигается реализация максимально удобного и ориентированного на пользователя приложения, обладающего уникальными инструментами для работы.

Для оценки конкурентов будут рассмотрены следующие сервисы: YClients, SimplyBook, DIKIDI, Masters, Мой Профи. Исследованы все преимущества и недостатки для построения конкурентоспособного приложения для своей целевой аудитории.

YClients – российская платформа онлайн-записи для различного вида бизнеса [2].

Данная система позволяет не только автоматизировать рутинные процессы, но и вести бизнес в целом – рассчитывать зарплату, подключать онлайн-кассу и другое. Платформа завоевывает обширную долю рынка Российской Федерации и активно конкурирует с DIKIDI, который будет рассмотрен ниже.

Преимущества:

* Платформа поможет решить подавляющее множество проблем при ведении бизнеса. Осуществлять складской учет, создавать программы лояльности, организовывать интеграции с другими сервисами.
* Заключила партнерство с крупными соцсетями, банками и другими площадками, что позволяет увеличить вовлеченность клиентов к услугам и бизнесу [3].
* Подходит практически любой сфере бизнеса и предлагает инструменты для удобного менеджмента.

Недостатки:

* Достаточно дорогостоящая для начинающих специалистов.
* Высокий порог входа: из-за расширения возможностей интерфейс стал нагруженным и не всегда интуитивно понятен. Для решения этой проблемы созданы курсы, однако они очень дороги.
* Разработчики не прислушиваются к массовым требованиям о внедрении необходимых функций, допускают множество ошибок в реализации.

SimplyBook – приложение онлайн-записи от SimplyBook.me с системой бронирования для сайтов и онлайн-календарем [4].

Преимущества:

* Достаточно прост для использования со стороны клиента.
* Даёт возможность создания собственного шаблонного сайта для записи на базе домена SimplyBook.
* Интеграция с различными API, возможность уведомлений клиентов по SMS.

Недостатки:

* Зарубежный, поэтому в любое время может стать недоступен для использования.
* Изначально создан для англоязычной аудитории и плохо переведен на русский язык. Незнание английского языка может вызывать проблемы при использовании и настройке приложения.
* Множество проблем с работоспособностью приложения, виджетов. Неудобство интерфейса.
* Отсутствие качественной и быстрой технической поддержки.
* Достаточно дорог для малого бизнеса, оплату из России осуществить сложнее из-за политики зарубежных платежных систем.

DIKIDI – бесплатный международный сервис онлайн-записи, автоматизации, управления и продвижения бизнеса в сфере услуг. [5]

Преимущества:

* Прост в освоении и настройке, не требует долгого обучения.
* Интеграции с картами, 1С, SMS-уведомлениями и другими сервисами.
* Есть бесплатный тариф, предоставляющий базовый функционал.

Недостатки:

* Отсутствие качественной технической поддержки.
* Приложение часто работает с ошибками и сбоями.
* Некоторые функции доступны только в web-версии, а реализация в мобильной версии затянута.

Rient – бесплатный сервис онлайн-записи + CRM для сферы услуг [6].

Преимущества:

* Бесплатный базовый функционал.
* Удобная подробная статистика.
* Инструменты CRM [7], обеспечивающие управление не только взаимоотношениями с клиентами, но и бизнесом.

Недостатки:

* Отсутствие мобильного приложения.

Мой профи – мобильное приложение для планирования сеансов и продвижения мастеров красоты [8].

Преимущества:

* Ориентировано на частных мастеров, как следствие предлагает конкурентно низкую стоимость подписки.
* Простое в использовании, не требует обширной предварительной настройки. Для любого пользователя, имеющего в наличии мобильный телефон не составит труда записаться, либо организовать онлайн-запись на свои услуги.
* Кроссплатформенность: разработчики создали и сайт, и мобильное приложение, что позволяет записаться с любого устройства, будь то персональный компьютер, планшет или смартфон.

Недостатки:

* Устаревший интерфейс, не привлекающий к элементам управления.
* Множество отзывов об ухудшении работоспособности приложения при его обновлении, ошибки. Разработчики не всегда вовремя исправляют проблемы, что приводит к снижению числа пользователей.
* Проблемы с получением уведомлений, периодически возникают сбои и напоминания как для мастеров, так и для клиентов, перестают работать.

1.4 Требования к системе

Итогом разработки должна стать информационная система, соответствующая следующим требованиям:

1. Система должна быть доступна на мобильных устройствах с любой операционной системой;
2. Приложение должно обладать разным функционалом для мастеров и клиентов;
3. Позволяет клиенту подобрать необходимую услугу, записаться на неё и, при необходимости, отменить;
4. Позволяет мастеру осуществлять контроль за записями клиентов, регулировать их;
5. Реализовывает возможность связи посредством встроенного чата;
6. Собирает удобную статистику для мастера;

1.5 Обзор решений

Были исследованы решения для мобильной и серверной части, чтобы достичь максимального удобства и при этом достаточной производительности приложения, обеспечив при этом хороший пользовательский опыт. Далее представлен обзор решений для каждой части системы.

1.5.1 Выбор средств для мобильной разработки

Можно выделить две основные категории языков программирования для разработки приложений на смартфоны – нативные и кроссплатформенные. Нативные языки программирования – платформы, созданные специально под определенную операционную систему. Для IOS это Swift, для Android – Java, Kotlin. В то время как нативные языки позволяют более тонко работать с ресурсами той или иной системы, позволяя достичь максимальной производительности и практически полного контроля, кроссплатформенные языки позволяют не тратить время на разработку двух приложений для каждой операционной системы, а направить весь вектор разработки на особенности самого функционала приложения.

В нашем случае, основная целевая аудитория – девушки, которые активно пользуются смартфонами. Согласно исследованию сервиса Яндекс.Радар по состоянию на 9 ноября 2024 года доля устройств на основе операционной системы Android чуть менее 80%, в то время как устройства под управлением IOS составили более 20% [9]. На графике можно увидеть, как менялись предпочтения в операционных системах смартфонов за 10 лет (Рисунок 1).

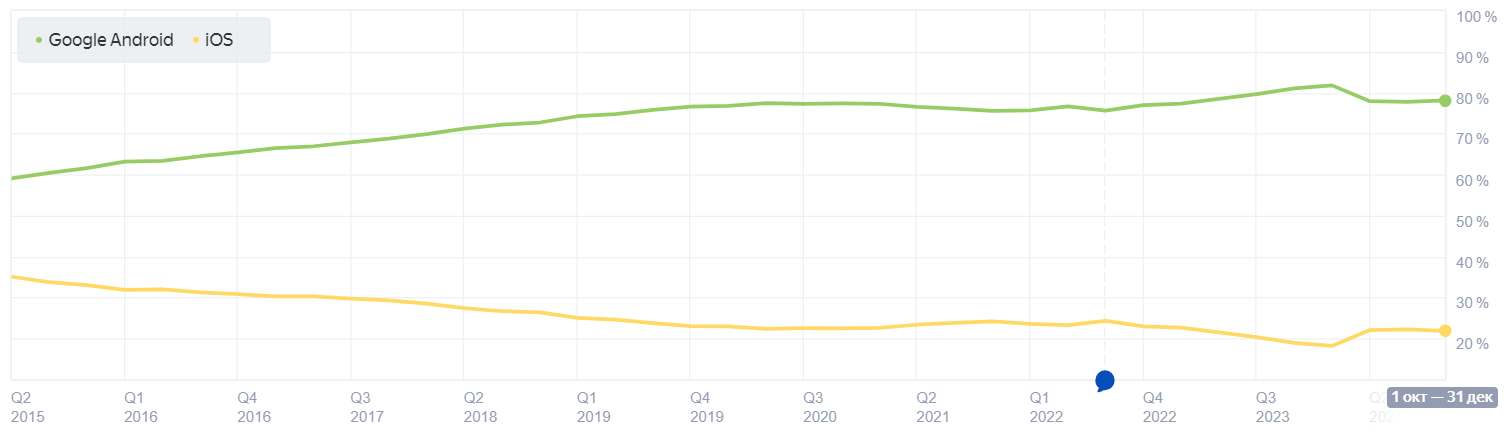


Рисунок 1 – Использование операционных систем на смартфонах в России в 2015-2024 годах.

Поэтому основное внимание нужно уделить этим двум операционным системам, что влечет за собой важность использования именно кроссплатформенных инструментов для экономии трудозатрат при разработке.

Рассмотрим самые популярные кроссплатформенные языки программирования React Native и Flutter.

React Native – платформа, созданная Facebook для создания мобильных приложения, используя в основе JavaScript и React, позволяющая создавать одну кодовую базу для разных операционных систем.

Был выпущен в 2015 году с открытым исходным кодом и в первые пару лет завоевал огромное доверие разработчиков. На нем были написаны такие мобильные приложения, как Skype, Discord, Facebook, Airbnb и другие.

Можно выделить основные причины популярности данного инструмента:

1. Экономия трудозатрат. Компания тратит средства и время на одну группу разработчиков, которые делают одно приложение для двух операционных систем, вместо двойных затрат во времени и средствах под каждую ОС отдельно;
2. В основе лежит React, который в свою очередь построен с использованием JavaScript, что обеспечивает низкий уровень входа в разработку, особенно, если есть предыдущий опыт в двух вышеупомянутых средствах;
3. Фреймворк позволил разработчикам веб-платформ без труда переквалифицироваться дополнительно в разработчиков и мобильных решений.

Flutter – инструментарий для создания скомпилированных приложений на основе виджетов, что даёт в результате визуально коренные элементы для каждой из операционных систем. Представлен компанией Google в конце 2018 года.

Поначалу фреймворк был воспринят с осторожностью, ведь на рынке был уже достаточно популярный и понятный React Native. Однако с течением лет ситуация изменилась и сейчас можно утверждать, что средства – прямые конкуренты друг другу. Можно сравнивать данные инструменты по следующим критериям:

1. Производительность. Разработчики Flutter предусмотрели его компиляцию в собственные библиотеки ARM или x86, что заметно влияет на производительность. React Native не компилируется в собственный код, и у него все еще есть слой JavaScript, что делает его менее производительным, чем Flutter. Однако Facebook знают о проблеме и занимаются ее решением, как результат, разработка движка Hermes, значительно повышающего производительность платформы на мобильных системах. Flutter в любом случае немного будет выигрывать в производительности, за счет того, что в качестве основы использует движок на C++, а React Native никак в ближайшее время не сможет избавиться от bridge архитектуры, устанавливая соединение с JavaScript посредством JSON. Однако разница в производительности несущественна, что позволяет выбирать любой из понравившихся инструментов.
2. Визуальная составляющая. Flutter использует виджеты, которые гарантированно будут выглядеть так же, как и элементы, построенные на нативных средствах разработки. React Native же работает напрямую с нативными элементами, что помимо правильного внешнего вида даёт возможность обновления в реальном времени.
3. IDE. Для React Native представлен более широкий спектр средств: практически все среды разработки, доступные для React поддерживают и React Native. В случае с Flutter не все так гладко, пока его поддерживают VS Code, Intellij и Android Studio.
4. API и пользовательский интерфейс. Для Flutter представлено больше встроенных средств, что даёт большую надежность и возможность выбора того, что необходимо. В случае с React Native встроенных библиотек не так много, лишь основные необходимые, однако, написано множество сторонних библиотек и даже фреймворков поверх React Native, которые реализуют тот же функционал, что и библиотеки Flutter, хоть местами и не так стабильны [10].

На основе вышеперечисленных рассуждений можно сделать вывод, что на данном этапе инструменты идут наравне, каждый имея свои плюсы и минусы. Поэтому выбор зависит только от предпочтений в языке разработки и ключевых особенностях разрабатываемого приложения. В нашем случае приложение визуально не нагруженное, поэтому React Native подойдет в качестве простого и понятного средства для разработки. Для облегчения сборки и запуска будем использовать фреймворк Expo, предоставляющий возможность видеть изменения в коде приложения в реальном времени благодаря развертыванию сервера и открытия его в приложении Expo Go. Так же Expo предоставляет собственные библиотеки и заготовки для разработки, что ускоряет создание приложения.

1.5.2 Выбор решений серверной части

Для серверной части буду использовать Java Spring Boot. Он написан на Java - одном из самых важных и широко используемых языков программирования в мире. В отличие от ряда других языков программирования, влияние которых с течением времени ослабевало, влияние Java с годами становилось только сильнее. Разработчики языка выделяют следующие особенности данного языка, которые и делают его настолько популярным:

* Простота – язык Java был спроектирован так, чтобы быть легким в изучении и эффективным в использовании профессиональным программистом. Если у вас есть определенный опыт программирования, то освоить Java не составит особого труда;
* надежность – Java является строго типизированным языком: ваш код проверяется на этапе компиляции. Тем не менее, код также проверяется и во время выполнения. Многие трудно обнаруживаемые ошибки, которые часто приводят к возникновению сложных для воспроизведения ситуаций во время выполнения, в Java попросту невозможны. Предсказуемость поведения написанного вами кода в несходных условиях - ключевая особенность Java;
* объектная ориентация – обильно заимствуя из многих продуктивных объектно-ориентированных сред, существующих на протяжении последних нескольких десятилетий, в Java удалось найти баланс между парадигмой "все является объектом" сторонников чистоты стиля и более прагматичной моделью "не путайтесь под ногами: объектная модель в Java проста и легко расширяема, в то время как элементарные типы, такие как целочисленные, были сохранены высоко производительными сущностями, которые не являются объектами;
* многопоточность – в Java поддерживается многопоточное программирование, которое дает возможность писать программы, выполняющие много действий одновременно. Исполняющая среда Java поддерживает элегантное, но вместе с тем сложное решение для синхронизации множества процессов, которое делает возможным построение бесперебойно работающих интерактивных систем;
* нейтральность к архитектуре – разработчики преследовали идеологию работоспособности написанного кода везде и всегда;
* интерпретируемость и высокая производительность – язык Java делает возможным создание межплатформенных программ за счет их компиляции в промежуточное представление, называемое байт-кодом Java. Байт-код может выполняться на любой системе, где внедрена машина JVM;
* распределенность – язык поддерживает TCP/IP и удаленный вызов методов;
* динамичность – программы на Java содержат существенный объем информации о типах времени выполнения, используемой для проверки и разрешения доступа к объектам во время выполнения, что делает возможным безопасное и надлежащее динамическое связывание кода. Это критически важно для надежности среды Java, в которой небольшие фрагменты байт-кода могут динамически обновляться в функционирующей системе [11].

Spring Boot – проект, упрощающий настройку и разработку приложений Spring. Построен на основе Spring Framework. Призван упростить разработку приложений с минимальной конфигурацией и ускорить процесс создания веб-приложений.

Преимущества Spring Boot:

1. На основе зависимостей, добавленных в проект, делает предположения о желаемом и производит настройку на основе этого, убирая ненужные части стандартной конфигурации Spring;
2. Имеет встроенный сервер Tomcat, Jetty или Undertow, позволяющий разрабатывать автономные приложения, собирающиеся в JAR-файл для легкого развертывания на любой машине с JVM.

Hibernate ORM (объектно-реляционное отображение) – фреймворк, упрощающий взаимодействие с базами данных для Java-приложений. Сопоставляет объектно-ориентированную модель предметной области с реляционной базой данных и дает возможность обращаться с данными, как с сущностями, которые разработчик сам реализует в виде желаемых таблиц в базе данных.

Преимущества Hibernate:

1. Заменяет прямой доступ к базе данных на функции обработки объектов высокого уровня, что, в свою очередь, облегчает понимание и работу разработчкикам;
2. Сопоставляет таблицы в базах данных с классами в Java и упрощает операции с базой;
3. Автоматически обеспечивает сохраняемость, разработчику больше не нужно заботиться о базовых операциях работы с базой данных, затрачивая время на написание явного кода взаимодействия с СУБД;
4. Предоставляет собственный мощный язык запросов HQL, в который заложены идеологии наследования, полиморфизма и ассоциации [12].

Таким образом, связка Java Spring Boot и Hibernate с необходимыми дополнительными библиотеками позволяют разрабатывать мощные веб-приложения, легко взаимодействующие с базой данных, а также реализующие все преимущества языка программирования Java.

1.5.3 Выбор сред разработки

Для разработки мобильной части выбрана Visual Studio Code.

Visual Studio Code, или VSCode, — текстовый редактор/редактор исходного кода, разработанный Microsoft с использованием Electron Framework для Windows, Linux и MacOS. Это один из самых популярных редакторов исходного кода на рынке. Вот некоторые особенности, которые отличают его от обычного текстового редактора:

* IntelliSense. VSCode выходит за рамки обычной подсветки синтаксиса и предлагает IntelliSense — интеллектуальные дополнения, адаптированные к коду и его тонкостям, которые дополнительно можно отрегулировать под свой вкус с помощью плагинов;
* Запуск и отладка. VSCode позволяет запускать ваш код (с отладкой или без неё с помощью веб-редактора) в самом приложении, а также обращаться к терминалу Windows напрямую из приложения для запуска некоторых средств, если это необходимо;
* Встроенный Git. В VSCode упрощена работа с Git, что позволяет контролировать коммиты, подготавливать файлы и выполнять другие действия с Git прямо из редактора. Реализовано это в виде удобного раздела, делающего работу прозрачной и быстрой, достаточно авторизоваться в Git и готово;
* Расширение: написано множество плагинов, расширяющих базовый функционал приложения и внедряющих множество полезных инструментов, что позволяет создать уникальную среду разработки под интересы каждого разработчика [13].

Для разработки серверной части выбрана IntelliJ IDEA.

IDEA разработана частной компанией JetBrains. Это интегрированная среда разработки (IDE), написанная на Java для разработки программного обеспечения и проектов. В ней существует бесплатная Community версия, дающая базовый функционал.

Возможности IntelliJ IDEA:

* Интеллектуальные подсказки к коду. В реальном времени во время написания IDE предугадывает, что вы хотите написать, давая подробную информацию о каждом классе или методе. Достаточно выбрать, что необходимо и не затрачивать время на самостоятельное написание. Так же данная функция помогает вспомнить забытые методы или просмотреть неизвестные с целью воспользоваться ими;
* Поиск повторяющегося кода. Все разработчики борются за повторное использование кода. IDE ищет повторяющийся код и помогает его заменить;
* Быстрые исправления. Если разработчик допускает ошибку в написании кода, IDE это отслеживает и предлагает варианты решения в реальном времени;
* Поддержка Git. Аналогично VSCode позволяет легко и быстро настроить взаимодействие с Git;
* Поддержка плагинов. Позволяет устанавливать плагины из магазина, что позволяет делать IDE уникальной под ваши вкусы и предпочтения [14].

1.5.4 Выбор системы управления базами данных

В качестве СУБД выбрана PostgreSQL – мощная объектно-реляционная система баз данных с открытым исходным кодом, которая использует и расширяет язык SQL в сочетании со многими функциями, позволяющими безопасно хранить и масштабировать работу с данными. История создания PostgreSQL началась в 1986 году в рамках проекта POSTGRES Калифорнийского университета в Беркли и насчитывает более 35 лет активной разработки на базовой платформе.

PostgreSQL заслужил прочную репутацию благодаря своей проверенной архитектуре, надежности, целостности данных, надежному набору функций, расширяемости и приверженности сообщества разработчиков программного обеспечения с открытым исходным кодом постоянному предоставлению производительных и инновационных решений. PostgreSQL работает на всех основных операционных системах, с 2001 года поддерживает ACID и имеет мощные дополнения, такие как популярный расширитель геопространственных баз данных PostGIS.

PostgreSQL обладает множеством функций, призванных помочь разработчикам создавать приложения, администраторам – защищать целостность данных и создавать отказоустойчивые среды, а также управлять данными независимо от того, насколько они масштабны [15].

1.6. Выводы по главе

В данной главе рассмотрена идеология разрабатываемого приложения, поставлена цель разработки и произведена декомпозиция задачи, что позволяет построить последовательный и четкий план разработки. Сформированы функциональные требования к системе.

Рассмотрены аналоги, что позволило понять, каким функционалом должно обладать приложение и какие недостатки конкурентов устранять, чтобы обладать популярностью у пользователей и выполнять необходимые функции.

Выбраны языки программирования, фреймворки, средства управления базами данных и среды разработки для мобильной и серверной частей приложения. Рассмотрены React Native, Expo, Java, Spring Boot, Hibernate, преимущества каждого из них и обоснован их выбор, чтобы обеспечить необходимые характеристики приложения, такие как визуальная составляющая, производительность, безопасность и другие. В случае с средами разработки рассмотрен функционал и возможности, влияющие на скорость и производительность разработки. Выбраны VSCode и IntelliJ IDEA, предоставляющие весь необходимый функционал для комфортной работы.

Изучена техника безопасности при разработке.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Прохорова М. А. О ВОЗМОЖНОСТЯХ CRM - СИСТЕМ ДЛЯ МАЛОГО БИЗНЕСА // Экономика и социум. 2016. №4-2 (23). URL: https://cyberleninka.ru/article/n/o-vozmozhnostyah-crm-sistem-dlya-malogo-biznesa (дата обращения: 26.10.2024).
2. YCLIENTS — онлайн-запись и автоматизация | CRM-система для салонов красоты и других типов бизнеса. — Текст : электронный // YCLIENTS : [сайт]. — URL: https://www.yclients.com/ (дата обращения: 26.10.2024).
3. Магазин интеграций партнерских сервисов YCLIENTS. — Текст : электронный // YCLIENTS : [сайт]. — URL: https://www.yclients.com/marketplace/overview (дата обращения: 26.10.2024).
4. SimplyBook.me - бесплатная система предварительной записи. — Текст : электронный // SimplyBook : [сайт]. — URL: https://simplybook.me/ru/ (дата обращения: 26.10.2024).
5. DIKIDI - сервис онлайн записи клиентов, автоматизация и продвижение в сфере услуг. — Текст : электронный // DIKIDI : [сайт]. — URL: https://dikidi.ru/ (дата обращения: 26.10.2024).
6. Rient — онлайн-запись клиентов + CRM. — Текст : электронный // Rient : [сайт]. — URL: https://rient.ru/ (дата обращения: 26.10.2024).
7. Черкашин, П. А. Стратегия управления взаимоотношениями с клиентами (CRM) : учебное пособие / П. А. Черкашин. — 2-е изд. — Москва : ИНТУИТ, 2016. — 420 с. — ISBN 978-5-94774-643-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100766 (дата обращения: 26.10.2024). — Режим доступа: для авториз. пользователей.
8. Лучшие частные мастера красоты России. — Текст : электронный // moiprofi : [сайт]. — URL: https://moiprofi.ru/ (дата обращения: 26.10.2024).
9. Мобильные ОС в России | Яндекс.Радар. — Текст : электронный // Яндекс.Радар : [сайт]. — URL: https://radar.yandex.ru/mobile?period=all&group=quarter (дата обращения: 15.11.2024).
10. Колчанова Светлана Анатольевна СРАВНЕНИЕ ФРЕЙМВОРКОВ FLUTTER И REACT NATIVE, ИСПОЛЬЗУЕМЫХ В РАЗРАБОТКЕ ГИБРИДНЫХ ПРИЛОЖЕНИЙ // E-Scio. 2022. №4 (67). URL: https://cyberleninka.ru/article/n/sravnenie-freymvorkov-flutter-i-react-native-ispolzuemyh-v-razrabotke-gibridnyh-prilozheniy (дата обращения: 26.11.2024).
11. Шилдт, Г. Java : Полное руководство / Шилдт Герберт. — 12-е изд. : Пер. с англ. — СПб : ООО "Диалектика", 2023. — 1344 c. — Текст : непосредственный (дата обращения: 26.11.2024).
12. Здитовец А. Л. ОСНОВНЫЕ ТЕХНОЛОГИИ И ФРЕЙМВОРКИ ДЛЯ БЕКЕНД-РАЗРАБОТКИ НА JAVA // Инновационная наука. 2023. №8-2. URL: https://cyberleninka.ru/article/n/osnovnye-tehnologii-i-freymvorki-dlya-bekend-razrabotki-na-java (дата обращения: 26.11.2024).
13. Visual Studio vs Visual Studio Code - What to Choose in 2024?. — Текст : электронный // GeeksForGeeks : [сайт]. — URL: https://www.geeksforgeeks.org/visual-studio-vs-visual-studio-code/ (дата обращения: 01.12.2024).
14. Eclipse Vs IntelliJ IDEA Vs NetBeans for Java Development. — Текст : электронный // GeeksForGeeks : [сайт]. — URL: https://www.geeksforgeeks.org/eclipse-vs-intellij-idea-vs-netbeans-for-java-development/ (дата обращения: 01.12.2024).
15. PostgreSQL: The World's Most Advanced Open Source Relational Database. — Текст : электронный // PostgreSQL : [сайт]. — URL: https://www.postgresql.org/ (дата обращения: 01.12.2024).