**РОСЖЕЛДОР**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

|  |  |  |  |
| --- | --- | --- | --- |
| **К защите:** |  |  | |
| **Заведующий кафедрой** | **Информационные** | |
| **технологии транспорта** | | |
|  | д-р техн. наук, проф. | |
|  |  | В. И. Хабаров | |
| *подпись* |  | *инициалы, фамилия* | |
|  |  |  | |
| *дата* |  |  | |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Тема:** | Разработка мобильного приложения по бронированию мест в ресторане | | | | | |
|  |  | | | | | |
|  | |  | БР.БИСТ.10.2022 |  |  |
|  | |  | *шифр документа* |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Выполнил** |  |  |  | **Руководитель** |
|  |  | А. Н. Кузовлева |  |  |  | канд. техн. наук, доц.  А. О. Коломеец |
| *подпись* |  | *инициалы, фамилия* |  | *подпись* |  | *инициалы, фамилия* |
|  |  |  |  |  |  |  |
| *дата* |  |  |  | *дата* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Консультанты по разделам** |  |  |  |  |
| 1 |  |  |  |  |
|  |  |  |  |  |
| 2 |  |  |  |  |
|  |  |  |  |  |
| 3 |  |  |  |  |
|  |  |  |  |  |
| Нормоконтролер работы |  |  |  | ст. преп.  Т. А. Распопина |
|  |  | *подпись* |  |  |
|  |  |  |  |  |
|  |  | *дата* |  |  |

**2022 г.**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

Факультет: Бизнес-информатики

Кафедра: Информационные технологии транспорта

Направление: 09.03.02 «Информационный системы и технологии»

Профиль: Интеллектуальные транспортные системы

|  |  |
| --- | --- |
|  | ***УТВЕРЖДАЮ****: зав. кафедрой «Информационные технологии транспорта»*  д-р техн. наук, проф.  В. И. Хабаров |
|  | *«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2022 г.* |

**З А Д А Н И Е**

**на выполнение выпускной квалификационной работы**

|  |  |  |  |
| --- | --- | --- | --- |
| студенту | Кузовлевой Анне Николаевне | | |
|  |  | | |
| 1. Тема «Разработка мобильного приложения по бронированию мест в ресторане» утверждена приказом №203/с от 30.05.2022 г. | | | |
| 2. Задание выдано «12» мая 2022 г. | | | |
| 3. Срок сдачи законченной работы на кафедру «17» июня 2022 г. | | | |
| 4. Исходные данные: данные, полученные в ходе прохождения преддипломной практики | | | |
| 5. Содержание расчетно-пояснительной записки | | | |
| Наименование разделов и вопросов | | Примерное количество страниц | График (сроки) выполнения |
| Введение | | 2 | 10.05.2022 |
| Анализ предметной области | | 15 | 10.05.2022 |
| Техническое задание | | 11 | 10.05.2022 |
| Проектирование информационной системы | | 5 | 28.05.2022 |
| Разработка ПО | | 14 | 28.05.2022 |
| Руководство пользователя | | 11 | 31.05.2022 |
| Заключение | | 1 | 02.06.2022 |

6. Содержание и объемы графической части

|  |  |  |
| --- | --- | --- |
| Наименование графического документа (чертежа, схемы, графика) | Количество  листов  формата А1 | График  (сроки)  выполнения |
| Презентация PowerPoint | 15 | 08.06.2022 |

|  |  |  |
| --- | --- | --- |
| Руководитель |  | А. О. Коломеец |
|  | *(подпись, фамилия, И.О.)* |  |
| Задание к использованию принял |  | А. Н. Кузовлева |
|  | *(подпись студента)* |  |

УДК 004.42

**АННОТАЦИЯ**

В работе 52 страницы, 51 рисунок, 7 таблиц, 16 источников.

Ключевые слова: *мобильное приложение, Android, бронирование, ресторан, администратор.*

Предмет исследования: мобильное приложение, предоставляющее возможность бронирования мест в ресторане.

Объект исследования: бронирование мест в ресторане.

Выпускная квалификационная работа посвящена разработке мобильного приложения, которое облегчит процесс бронирования мест в ресторане как для клиентов, так и для администратора предприятия.

Основанием для создания данного приложения явилась потребность в автоматизации процесса бронирования, способствующей экономии времени посетителей и работников предприятия.

**ABSTRACT**

The work contains 53 pages, 51 figures, 7 tables, 16 sources.

Keywords: *mobile application, Android, booking, restaurant, administrator.*

Subject of research: a mobile application that provides the opportunity to book seats in a restaurant.

Object of research: booking seats in a restaurant.

The final qualifying work is devoted to the development of a mobile application that will facilitate the process of booking seats in a restaurant for both customers and the administrator of the enterprise.

The basis for the creation of this application was the need to automate the booking process, which helps to save time for visitors and employees of the enterprise.

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

БД – база данных.

ГОСТ – государственный стандарт, который формулирует требования государства к качеству продукции, работ и услуг, имеющих межотраслевое значение.

ОС – операционная система.

СУБД – система управления базами данных.

ФИО – фамилия, имя и отчество.

IDE (Integrated Development Environment) – интегрированная среда разработки.

ER-diagram (entity-relationship diagram) – диаграмма сущность-связь.

Wi-Fi – технология беспроводной локальной сети с устройствами

ПК – персональный компьютер.

ПО – программное обеспечение.

POS-терминалы – (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Point Of Sale – точка продажи и от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Terminal – окончание) – это электронное программно-техническое устройство для приёма к оплате платёжных карт.

UI – пользовательский интерфейс.

HTML – (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) HyperText Markup Language – «язык [гипертекстовой](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82) разметки») – стандартизированный язык разметки документов для просмотра [веб-страниц](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) в [браузере](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80).

CSS – Cascading Style Sheets «каскадные таблицы стилей») –[формальный язык](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) описания внешнего вида документа ([веб-страницы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0)), написанного с использованием [языка разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8) (чаще всего [HTML](https://ru.wikipedia.org/wiki/HTML) или [XHTML](https://ru.wikipedia.org/wiki/XHTML))

SDK – (сокращение от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) software development kit – переводится как «комплект для разработки [программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)») – это набор инструментов для [разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) в одном устанавливаемом пакете.

СОДЕРЖАНИЕ

[Введение 6](#_Toc103894581)

[1 Анализ предметной области и моделирование бизнес-процессов 8](#_Toc103894582)

[1.1 Описание предметной области 8](#_Toc103894583)

[1.2 Обоснование актуальности разрабатываемого программного продукта 8](#_Toc103894584)

[1.3 Обзор ПО с похожим функционалом 9](#_Toc103894585)

[1.4 Моделирование бизнес-процессов 12](#_Toc103894586)

[1.5 Выводы по первому разделу 15](#_Toc103894587)

[2 Проектирование информационной системы 16](#_Toc103894588)

[2.1 Требования к мобильному приложению 16](#_Toc103894589)

[2.2 Обоснование выбора среды разработки 17](#_Toc103894590)

[2.3 Описание языка программирования JavaScript 20](#_Toc103894591)

[2.4 Разработка концептуальной модели базы данных 21](#_Toc103894592)

[2.5 Выводы по второму разделу 24](#_Toc103894593)

[3 Описание основных этапов разработки мобильного приложения по бронированию мест в ресторане 25](#_Toc103894594)

[3.1 Описание функциональных требований 25](#_Toc103894595)

[3.2 Разработка базы данных 25](#_Toc103894596)

[3.3 Проектирование интерфейса мобильного приложения 29](#_Toc103894597)

[3.4 Разработка программного обеспечения 32](#_Toc103894598)

[3.5 Выводы по третьему разделу 38](#_Toc103894599)

[4 Руководство пользователя 39](#_Toc103894600)

[Заключение 50](#_Toc103894601)

[Список использованных источников 51](#_Toc103894602)

# ВВЕДЕНИЕ

Информационные технологии развиваются каждый день и находят свое применение во многих сферах человеческой деятельности. В связи с этим, использование автоматизированных рабочих систем позволяет усовершенствовать процессы по работе с документами, сократить обязанности обслуживающего персонала, соответственно, увеличить производительность труда.

Рассмотрим, как информационные технологии участвуют в развитии ресторанного бизнеса. На сегодняшний день рабочие места официантов оборудованы компьютерами. Для посетителей предприятия общественного питания предоставляют бесплатный Wi-Fi, преимущества которого заключаются в возможности поработать во время приема пищи, организовать деловую встречу или пообщаться с друзьями и близкими.

Предприятия, занятые в сфере общественного питания, зачастую теряют значительную часть прибыли из-за неоптимального распределения мест при бронировании или посадке гостей. В связи с этим, автоматизация задачи по бронированию мест выгодна как для предприятия, так и для будущих посетителей.

Администрация увеличивает оперативность работы персонала, привлекает новых клиентов и повышает заполняемость. В таком случае, клиенту для оформления заявки необходимо предоставить ФИО для успешной идентификации по прибытию, номер телефона и количество персон.

Автоматизация бронирования позволит достигнуть повышения эффективности работы предприятия в целом за счет увеличение количества обслуживаемых клиентов, а также позволит улучшить значения показателей качества хранения, передачи и обработки информации, т.е. сократить время обработки и получения оперативных данных для принятия управленческих решений.

Актуальность выбранной темы определяется постоянно увеличивающимися темпами автоматизации во всех сферах человеческой деятельности и стремительно возрастающими возможностями вычислительных систем. В связи с чем и растет потребность в программных продуктах, позволяющих пользователю производить самообслуживание. Бронирование интернет-сервисом набирает все большую популярность, так как это является удобным способом сделать это быстро и без использования третьих лиц.

На данный момент уже накоплен некоторый опыт разработок в данной области, но от этого круг задач только расширяется. Имеется множество сервисов, осуществляющих бронирование, однако каждый из них имеет свои преимущества и недостатки. Точнее говоря, большинство создающихся программных комплексов ориентированы на конкретные задачи и определенные условия.

Целью выпускной квалификационной работы является разработка мобильного приложения по бронированию мест в ресторане.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* анализ существующих систем, обладающих схожим функционалом;
* определение функционала приложения на основе анализа существующих систем;
* изучение и выбор средств для осуществления разработки мобильного приложения для ОС Android;
* проектирование разработки приложения;
* разработка приложения для бронирования мест в ресторане;
* тестирование системы.

По окончанию работ разработано мобильное приложение по бронированию мест в ресторане, а также описана пользовательская документация, необходимая для его эксплуатации.

### 1 Анализ предметной области и моделирование бизнес-процессов

# 1.1 Описание предметной области

Ресторан представляет собой заведение, предназначенное для общественного питания и отдыха. Задачей предприятия является оказание услуг общественного питания, удовлетворяющих потребности клиента.

В связи с этим, возникает необходимость в комплексном управлении всеми ресурсами ресторана. Аппаратное (сенсорные POS-терминалы, принтеры, сканеры штрих-кода, считыватели карт и электронные весы) и программное обеспечение (1С: Ресторан) для автоматизации ресторана оказывают помощь всему персоналу предприятия питания.

Автоматизация процесса бронирования позволит вести аналитику предприятия, учитывая отзывы клиентов, уменьшить объем работы администратора, так как пользователи смогут заранее самостоятельно забронировать столик, уменьшить время ожидания посетителей в заведении.

В момент открытия ресторана вечером, когда приходят посетители, администратору бывает очень сложно сориентироваться, какие столики заняты, а какие свободны. Поэтому автоматизация является эффективным средством усовершенствования работы организации [1].

**1.2 Обоснование актуальности разрабатываемого программного продукта**

В данном разделе опишем возникающую проблему и пути ее решения.

Автоматизированное рабочее место администратора кафе предполагает наличие стандартного набора устройств: компьютера, монитора, компьютерной мыши и клавиатуры. Администратор кафе владеет информацией о столах, которые были заказаны. АРМ позволяет администратору видеть, какие столы уже заняты, а какие доступны для формирования заказа.

Администратор занимается сферой обслуживания клиентов и задействован в следующих бизнес-процессах:

* регистрация клиента;
* создание заказа с непосредственным бронированием столика;
* контроль обслуживания заказов официантами.

Рассмотрим, какие функции предполагается автоматизировать:

* регистрация посетителя самостоятельно;
* создание заказа с непосредственным бронированием столика;
* заполнение справочников клиентов, блюд, столиков, организаций, персонала;
* построение отчетной документации.

На текущий момент данная работа выполняется следующим образом: клиент звонит по телефону, желая заказать столик, администратор смотрит журнал заказов и подбирает подходящий столик, фиксирует заказ в журнале, записывает номер телефона.

В момент открытия заведения вечером, когда приходят посетители, администратору бывает очень сложно сориентироваться, какие столики заняты, а какие свободны. Поэтому автоматизация является эффективным средством усовершенствования работы администратора.

**1.3 Обзор ПО с похожим функционалом**

В качестве вариантов автоматизации существуют готовые типовые решения, предназначенные для полной автоматизации деятельности ресторана.

Предприятие для внедрения в практику своей работы онлайн-бронирования может использовать один из трех вариантов: использовать веб-сервис, подключить готовый модуль на веб-сайт, заказать разработку собственной системы онлайн-бронирования.

Существует линейка программных продуктов «Трактиръ», которая включает в себя решения для автоматизации процесса обслуживания гостей и управления оперативной деятельностью ресторана, ведения управленческого и бухгалтерского учета, а также интернет-сервисы, связанные с ресторанным бизнесом. Преимущества программных продуктов: учет рабочего времени персонала, бронирование столов, управление доставкой, аналитика, отчеты [1].

Недостатком данной системы является отсутствие следующих функций: отсутствие приложения для гостей, программы лояльности, нет управления правами пользователя, рисунок 1.1.

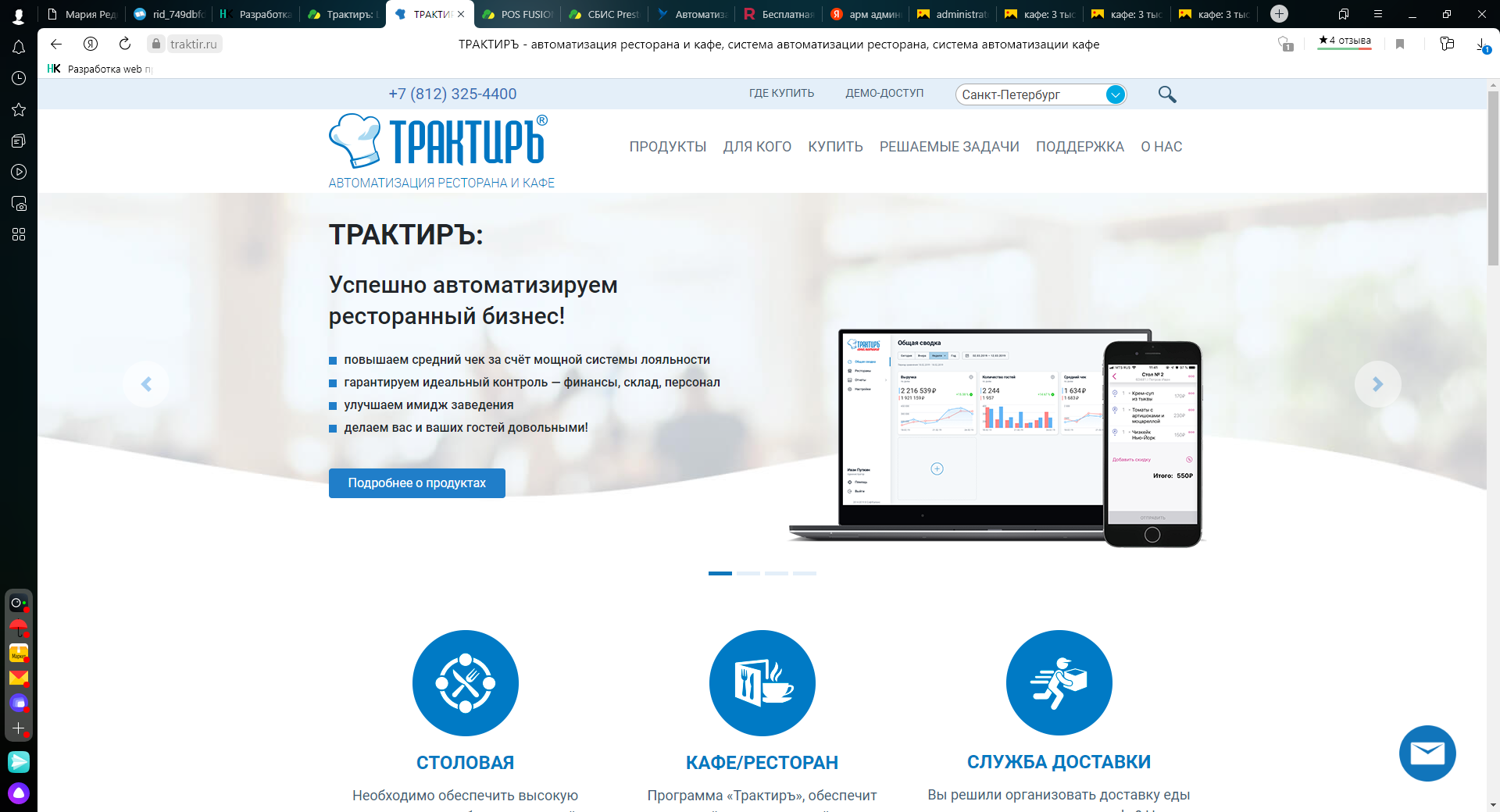


Рисунок 1.1 – Информационная система «Трактиръ»

В качестве еще одного примера рассмотрим сервис бронирования «Restoplace», рисунок 1.2. Сервис бронирования столов создан для автоматизации работы хостес в ресторане. Сервис Restoplace позволяет гостям легко бронировать столики на любое свободное время. Просто создайте виджет с фотографиями своих столиков, установите на свой сайт и в социальные сети, появится кнопка «Забронировать столик». Создайте пользователя с правами «хостес» и передайте логин, пароль своему сотруднику. Теперь вы сможете видеть, то, как работает ваше заведение в режиме реального времени [2].

Минусом данного сервиса является ориентированность на любое предприятие, что затрудняет представление пользователя об интересующем заведении и накладывает ограничения на применение программ лояльности.

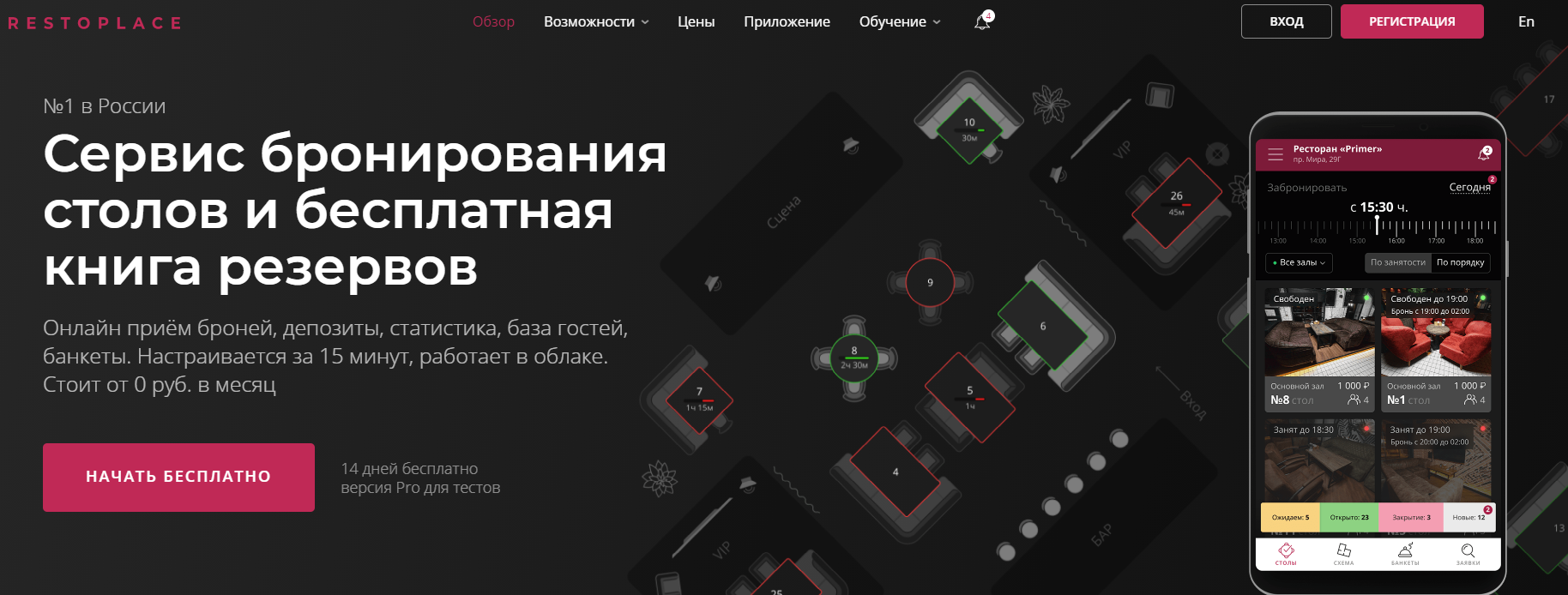


Рисунок 1.2 – Информационная система «Restoplace»

Еще одним аналогом данной системы является система автоматизации ресторанов, столовых и служб доставки «СБИС Presto», рисунок 1.3.

Сервис предоставляет возможности автоматизации работы персонала. Реализует следующие функции: ведение складского учета, управление и бронь залов, заказ продуктов, связь с кухней [3].



Рисунок 1.3 – Информационная система «СБИС Presto»

Представленная линейка аналогов не предполагает использование мобильного приложения, позволяющего пользователю совершать бронирование без привлечения третьих лиц.

# 1.4 Моделирование бизнес-процессов

Для описания объектов-сущностей используются CASE-диаграммы.

Диаграммы вариантов использования представляют собой графическое представление взаимодействия пользователя и программного приложения. Диаграммы строятся во время изучения технического задания, она состоит из графической диаграммы, описывающей действующие лица и прецеденты (варианты использования), а также спецификации, представляющего собой текстовое описание конкретных последовательностей действий (потока событий), которые выполняет пользователь при работе с системой. Диаграмма вариантов использования является необходимым средством при анализе требований, выдвигаемых по отношению к проектируемой системе [4].

На рисунке 1.4 представлена диаграмма вариантов использования для актера «Посетитель».

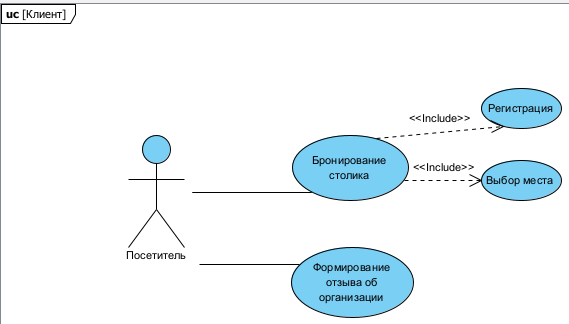


Рисунок 1.4 – Диаграмма вариантов использования актера «Посетитель»

С помощью диаграммы последовательности описаны процессы «Бронирование столика» и «Формирование отзыва» на рисунках 1.5, 1.6.

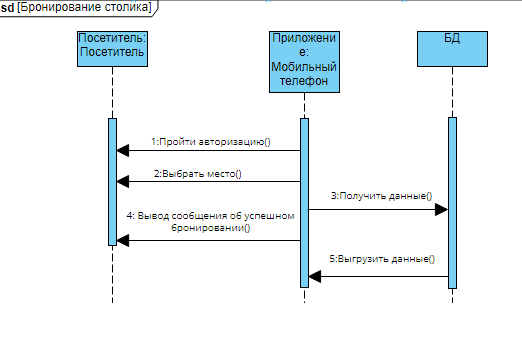


Рисунок 1.5 – Дигарамма последовательности «Бронирование столика»

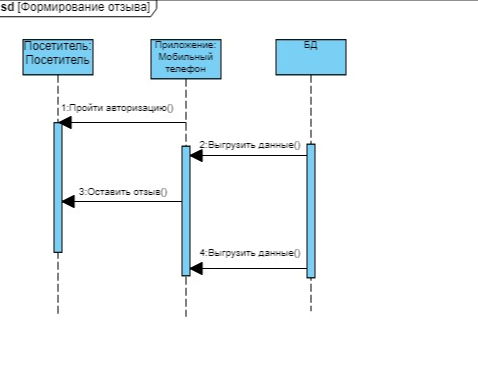


Рисунок 1.6 – Дигарамма последовательности процесса «Формирование отзыва»

Диаграмма последовательности позволяет отобразить временные особенности передачи и приема сообщений объектами.

На рисунке 1.7 изображена диаграмма состояний объекта «Заказ» во взаимодействии с посетителем. На рисунке 1.8 представлена диаграмма обзора взаимодействия процесса «Совершение заказа».

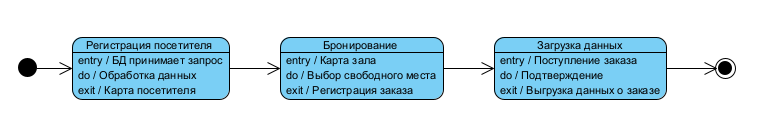


Рисунок 1.7 – Диаграмма состояний объекта «Заказ»

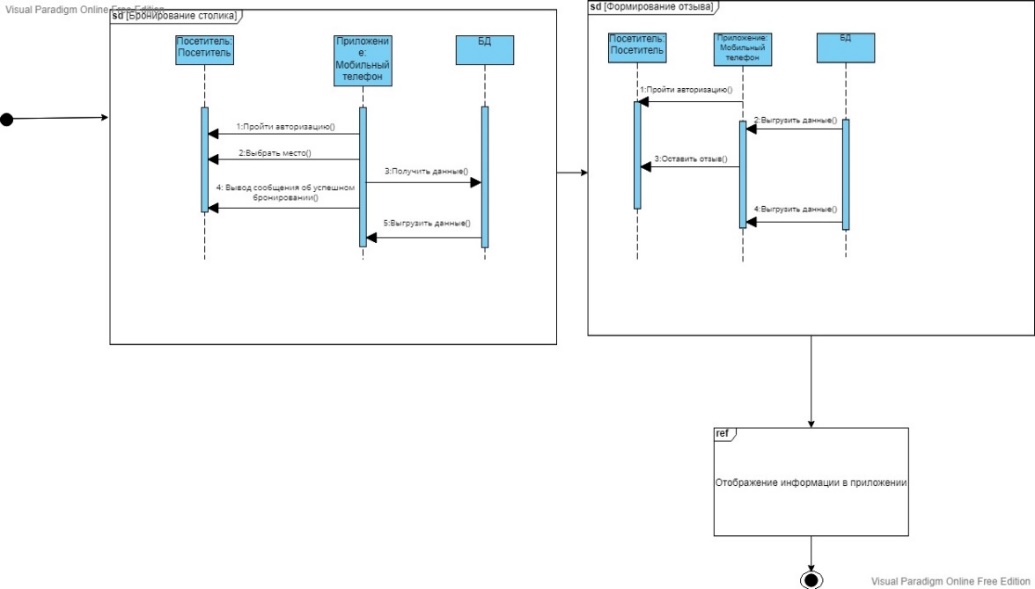


Рисунок 1.8 – Диаграмма обзора взаимодействия

Диаграмма классов (от англ. «class diagram») предназначена для представления внутренней структуры программы в виде классов и связей между ними [4]. Диграмма классов предметной области представлена на рисунке 1.9.

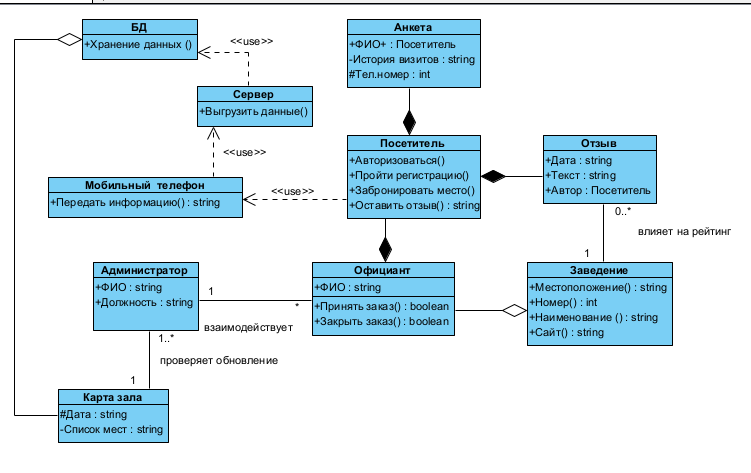


Рисунок 1.9– Диаграмма классов предметной области «Ресторан»

# 1.5 Выводы по первому разделу

В первом разделе было представлено описание предметной области, приведено обоснование актуальности разрабатываемого программного продукта.

На основании полученной информации были смоделированы бизнес-процессы с использованием нотации UML. Построена диаграмма вариантов использования, диаграмма состояний, диаграммы последовательностей, диаграмма обзора взаимодействия и диаграмма классов.

На сегодняшний день современному человеку необходимо быть коммуникабельными и мобильными, поэтому смартфоны с каждым годом становятся все популярнее, а их возможности постоянно расширяются. Системы автоматизации ресторанного бизнеса и бронирования мест станут еще более функциональными, а число использующих их заведений будет расти, если в них будет реализована функциональность для решения связанных задач – например, бронирование столиков и сбор отзывов посетителей.

Был проведен анализ аналогичных решений, в других проектах больше реализованы функции на получение информации для рабочего персонала предприятия, но отсутствует взаимодействие с пользователем и возможность самообслуживания (самостоятельного бронирования мест) для будущих клиентов.

Представленный анализ рынка аналогов также позволяет подтвердить актуальность предлагаемого решения.

Во втором и третьем разделе представлено описание этапов разработки мобильного приложения по бронированию мест в ресторане.

# 2 Проектирование информационной системы

# 2.1 Требования к мобильному приложению

Язык интерфейса приложения: русский. Разработанное программное обеспечение должно быть совместимо с устройствами на базе ОС Android 4.0 (и старше).

По восприятию дизайн приложения должен быть адаптирован под любого пользователя.

Должны быть использованы преимущественно светлые и контрастные цветовые решения. На страницах не должно быть большого объема текста.

В дизайне не должны присутствовать:

* мелькающие баннеры;
* много сливающегося текста;
* тёмные и агрессивные цветовые сочетания и графические решения.

Фон экрана должен быть однородным и повторяющимся. В том случае,

если требуется использовать текстурирование – текстура точно так же должна быть однородной и легко дублируемой.

Пример дизайна страниц с подбором цветовой гаммы представлен на рисунке 2.1. Данное решение было создано в графическом онлайн-сервисе Figma.

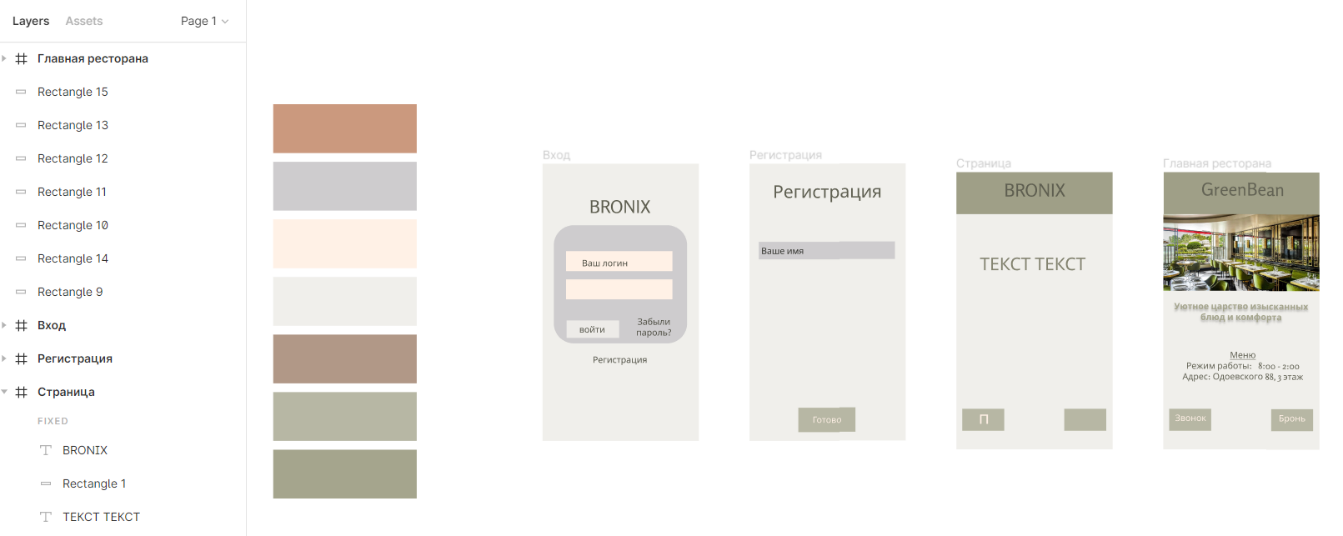


Рисунок 2.1 – Пример дизайна

# 2.2 Обоснование выбора среды разработки

Среда разработки – это внедрение комплексных программных средств с целью создания ПО для гаджета (сокращенная аббревиатура «IDE»). В некоторых случаях может вмещать в себе элементы интегрирования с управленческими системами и разный инструментарий, упрощающий построение графического изображения с нуля.

Существуют различные среды разработки, позволяющие разрабатывать приложения для устройств, использующие операционную систему Android. Однако самыми популярными являются следующие IDE:

* IntelliJ Idea;
* Eclipse;
* React Native;
* Android Studio.

Программа IntelliJ Idea разработана российской компанией JetBrains. Подобно Эклипс, эта среда позволяет создавать приложения и программы на нескольких программных языках. Также IntelliJ Idea не перегружает ПК. Достоинства:

* более оперативная отладка значений;
* предусмотрен автозаполнитель методов;
* есть рефакторинг;
* интерфейс более понятный и лаконичный;
* подходит для тех, кто программирует на Java.

Единственный недостаток – за IntelliJ Idea нужно платить. Хотя тот, кто понимает ситуацию на рынке, не будет считать это минусом [5].

Следующая среда разработки Eclipse является интегрированной и позволяет подключать дополнения, использование которых позволяет значительно расширить функционал данной программы. IDE является кроссплатформенной – она работает под операционными системами Linux, Windows, Mac OS X и Solaris.

С одной стороны, то, что IDE Eclipse позволяет создавать разнообразные продукты, очень удобно. Данная среда разработки успешно применяется для огромного количества самых разнообразных задач.

С другой стороны, при разработке приложений Android из-за гибкости Eclipse возникает ряд неудобств, в сравнении со средой разработки, ориентированной исключительно на создание данных приложений. К таким неудобствам можно отнести, например, необходимость самостоятельно дополнительно устанавливать эмулятор устройства с ОС Android, необходимого для тестирования разрабатываемого приложения без использования реального устройства [6].

Android Studio – среда разработки под Android, основанная на IntelliJ IDEA. Подобно Android IDE, она предоставляет интегрированные инструменты для разработки и отладки. Дополнительно ко всем возможностям, ожидаемым от IntelliJ, в Android Studio реализованы:

* поддержка сборки приложения, основанной на Gradle;
* специфичный для Android рефакторинг и быстрое исправление дефектов;
* lint инструменты для поиска проблем с производительностью, с юзабилити, с совместимостью версий и других;
* возможности ProGuard (утилита для сокращения, оптимизации и обфускации кода) и подписи приложений;
* основанные на шаблонах мастера для создания общих Android конструкций и компонентов;
* WYSIWYG редактор, работающий на многих размерах экранов и разрешений, окно предварительного просмотра, показывающее запущенное приложение сразу на нескольких устройствах и в реальном времени;
* встроенная поддержка облачной платформы Google.

IDE Android Studio, как уже было сказано, предназначена именно для разработки приложений на ОС Android, ориентирована исключительно на данную систему, а потому достаточно удобна, понятна и функциональна. После загрузки среды, в ней можно найти большое множество различных макетов для создания пользовательского интерфейса (UI), с которого чаще всего начинается разработка приложения.

Некоторые специалисты считают, что данная среда разработки имеет ряд недоработок, которые необходимо исправить. Но даже они признают, что IDE Android Studio весьма конкурентоспособна [7].

React Native использует встроенные компоненты интерфейса и API-интерфейсы. Фреймворк позволяет разрабатывать мобильные приложения, неотличимые от нативных приложений, использующих [Java](https://proglib.io/p/5-books-enterprise-java-developer), [Kotlin](https://proglib.io/p/do-you-need-kotlin), [Swift](https://proglib.io/p/swift-guide) или [Objective C](http://www.ohandroid.com/objective-c-android.html).

За прошедшие годы React Native приобрел значительную популярность и используется компаниями Facebook, Instagram, Tesla, Intuit, Bloomberg, Uber, Yahoo и Walmart.

Помогает разработчикам [значительно сократить затраты и время разработки](https://www.g2.com/survey_responses/react-native-review-3497394) приложений.

Среда разработки React Native нацелена на результат, сопоставимый с нативной разработкой. React Native предоставляет компоненты для текста, изображений, ввода с клавиатуры, прокручиваемых списков, индикатора выполнения, анимации, буфера обмена, ссылок. Эти компоненты значительно ускоряют процесс разработки приложений, а функция «Hot Reloading» также экономит много времени, поскольку позволяет перезагрузить приложение, не перекомпилировав весь код.

Инструменты разработки, такие как [Nuclide](https://nuclide.io/) для написания кода, [Yoga](https://yogalayout.com/) для построения макетов, [Sentry](https://sentry.io/welcome/) для мониторинга ошибок и сбоев и [React Developer Tools](https://reactnative.dev/docs/debugging#react-developer-tools) для отладки, делают процесс разработки React Native намного проще и быстрее [8].

Выбор данной платформы в качестве среды разработки мобильного приложения основан на вышеперечисленных преимуществах.

# 2.3 Описание языка программирования JavaScript

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

Javascript популярен не случайно, а благодаря своим достоинствам.

Незаменимость для веб-разработки*.* Поддержка скриптов всеми популярными браузерами, полная интеграция с вёрсткой страниц (HTML+CSS) и серверной частью (backend).

Скорость работы и производительность. Javascript позволяет частично обрабатывать веб-страницы на компьютерах пользователя без запросов к серверу. Это экономит время и трафик, снижает нагрузку на сервер.

Простота и рациональность применения. Для сложных заданий есть варианты решения, можно подобрать лучший, адаптировать.

Удобство пользовательских интерфейсов. Заполнение форм, выбор действий, активация кнопок, проверки ввода, реагирование на клики мыши.

Лёгкость освоения. Сначала код может показаться сложным, но к его синтаксису, логике быстро привыкаешь. Отображение действий ощутимо добавляет энтузиазма.

Как любой язык программирования, Javascript имеет некоторые недостатки (ограничения).

Нет возможности чтения и загрузки файлов. Это ограничение функциональности на стороне клиента. Главная причина – соображения безопасности.

Нестрогая типизация и вольная трактовка. Язык игнорирует явные нестыковки. Имеет место разная интерпретация данных. Нет возможности раннего выявления ошибок. Все недочёты выявляются уже на этапе работы.

Доступность для злоумышленников. В свободный скриптовый язык проще всего встроить фрагмент вредоносного кода, который может навредить пользователю. Надежда только на антивирус.

Надо отметить, что сообщество пользователей Javascript активно улучшает язык, устраняя многие недостатки. Узких мест становится всё меньше. Браузеры постоянно совершенствуют работу с JS. Выход HTML 5 дал новый толчок к расширению возможностей скриптов.

TypeScript – это язык программирования для веб-разработки, основанный на [JavaScript](https://blog.skillfactory.ru/glossary/javascript/). Делает код понятнее и надежнее, добавляет статическую типизацию (переменные привязаны к конкретным типам данных), а также может быть скомпилирован в JavaScript [9].

**2.4 Разработка концептуальной модели базы данных**

ER-модель (от англ. entity-relationship model, модель «сущность – связь») – модель данных, позволяющая описывать концептуальные схемы предметной области. ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями [10].

Для данной предметной области можно спроектировать следующие сущности: Заведение, Персонал, Столики, Клиент, Заказ, Бронь столика, Карта зала.

В таблицах 2.1 – 2.7 дано описание каждой сущности.

Таблица 2.1 – Описание сущности «Персонал»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательное | PK | Описание |
| Код\_сотрудника | integer |  |  | Уникальный идентификатор |
| ФИО | varchar(50) |  |  | Фамилия, имя, отчество |
| Должность | varchar(30) |  |  | Занимаемая должность |

Таблица 2.2 – Описание сущности «Заведение»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательное | PK | Описание |
| Наименование | integer |  |  | Уникальный идентификатор |
| Код\_карты | integer |  | FK | Код карты зала |
| Телефон | integer |  |  | Телефон заведения |
| Адрес | varchar  (60) |  |  | Адрес заведения |
| Номер\_отзыва | integer |  | FK | Отзыв о заведении из ЛК |

Таблица 2.3 – Описание сущности «Столики»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательное | PK | Описание |
| Номер\_столика | integer |  |  | Уникальный идентификатор |
| Количество мест | integer |  |  | Количество свободных мест |
| Код\_брони | integer |  | FK | Идентификатор брони столика |

Таблица 2.4 – Описание сущности «Клиент»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательное | PK | Описание |
| Код\_клиента | integer |  |  | Уникальный идентификатор |
| Код\_лк | integer |  | FK | Личный идентификатор |
| ФИО | varchar(50) |  |  | Фамилия, имя клиента |

Таблица 2.5 – Описание сущности «Заказ»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательное | PK | Описание |
| Код\_заказа | integer |  |  | Уникальный идентификатор |
| Код\_клиента | integer |  | FK | Идентификатор сущности Клиент |
| Дата | date |  |  | Дата заказа |

Таблица 2.6 – Описание сущности «Бронь столика»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательное | PK | Описание |
| Код\_брони | integer |  |  | Уникальный идентификатор |
| Код\_клиента | integer |  | FK | Идентификатор сущности Клиент |
| Дата | datetime |  |  | Дата и время брони |

Таблица 2.7 – Описание сущности «Карта зала»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательное | PK | Описание |
| Код\_карты | integer |  |  | Уникальный идентификатор |
| Номер\_столика | integer |  | FK | Идентификатор сущности Столики |

На рисунке 2.2 показана ER-диаграмма, представляющая сущности Персонал, Заведение, Бронь столика, Столики, Карта зала, Клиент, Заказ и их взаимосвязи. Числа, описывающие типы бинарных связей (1:1, 1:M, M:N), обозначают максимальное количество сущностей на каждой стороне связи. ER-диаграммы устроены по тому же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи – глаголов [11].

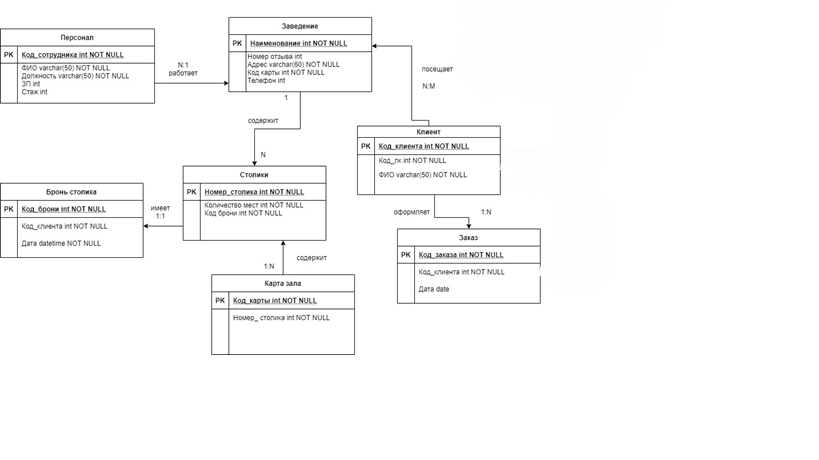


Рисунок 2.2 – Концептуальная модель базы данных

# 2.5 Выводы по второму разделу

В данном разделе были рассмотрены наиболее популярные среды разработки мобильных приложений. Представлено описание языка разработки JavaScript (TypeScript) и краткое описание каждой платформы, на основании которого был сделан выбор.

Основываясь на требования к разрабатываемому продукту и представленное описание платформ, в качестве инструмента разработки выбрана среда React Native.

Также была разработана концептуальная модель базы данных, позволяющая представить подробное описание предметной области. Построена ER-модель информационной системы «Бронирование столиков в ресторане», включающая семь сущностей.

В виде таблиц приведено семантическое описание сущностей, описаны взаимосвязи между сущностями.

Предлагаемая система представляет собой мобильное приложение, способное удовлетворить потребности клиентов. Приложение должно быть способно обрабатывать процессы, такие как бронирование столика, регистрация клиента, предварительное формирование брони. Таким образом, это поможет облегчить работу администратора и сэкономить время клиентов.

Подводя итог, следует отметить, что разрабатываемое приложение должно быть простым в использовании и предоставлять возможности для удобного и эффективного бронирования мест в ресторане.

# 3 Описание основных этапов разработки мобильного приложения по бронированию мест в ресторане

# 3.1 Описание функциональных требований

Предлагаемая система представляет собой мобильное приложение, способное удовлетворить потребности клиентов. Приложение должно быть способно обрабатывать процессы, такие как бронирование столика, регистрация клиента, предварительное формирование заказа. Таким образом, это поможет облегчить работу администратора и сэкономить время клиентов.

Выделим следующие ключевые требования к разрабатываемому программному обеспечению, предназначенному для бронирования столиков и мест на предприятии общественного питания (ресторан):

* отображение расположения столиков в зале со статусом (свободный/забронированный);
* возможность выбора столика для бронирования пользователем;
* просмотр меню заведения;
* отзывы посетителей. Возможность оставить оценку посещенному заведению и просмотреть отзывы других клиентов;
* история заказов. Хранение информации о ранее совершенных бронированиях конкретного пользователя для получения скидки постоянного клиента;
* возможность восстановления пароля по e-mail.

# 3.2 Разработка базы данных

База данных Firebase Realtime – это база данных, размещенная в облаке. Данные хранятся в формате JSON и синхронизируются в реальном времени для каждого подключенного клиента. Когда вы создаете кроссплатформенные приложения с помощью наших платформ Apple, Android и JavaScript SDK, все ваши клиенты совместно используют один экземпляр базы данных реального времени и автоматически получают обновления с новейшими данными.

База данных Firebase Realtime позволяет создавать многофункциональные приложения для совместной работы, предоставляя безопасный доступ к базе данных непосредственно из клиентского кода. Данные сохраняются локально, и даже в автономном режиме события в реальном времени продолжают запускаться, предоставляя конечному пользователю оперативную работу. Когда устройство восстанавливает соединение, база данных реального времени синхронизирует локальные изменения данных с удаленными обновлениями, которые произошли, когда клиент был в автономном режиме, автоматически объединяя любые конфликты. База данных реального времени – это база данных NoSQL, поэтому ее оптимизация и функциональность отличаются от реляционной базы данных.

Все данные базы данных Firebase Realtime хранятся в виде объектов JSON. Вы можете думать о базе данных как о дереве JSON, размещенном в облаке. В отличие от базы данных SQL здесь нет таблиц или записей. Когда вы добавляете данные в дерево JSON, они становятся узлом в существующей структуре JSON со связанным ключом. Вы можете предоставить свои собственные ключи, такие как идентификаторы пользователей или семантические имена, или они могут быть предоставлены с помощью [push()](https://firebase.google.com/docs/reference/android/com/google/firebase/database/DatabaseReference#push()) [12].

На рисунке 3.1 представлено подключение базы данных к приложению.



Рисунок 3.1 – Подключение БД

База данных представлена на рисунке 3.2. Она содержит информацию о столах, историю заказов клиента, данные о клиентах, отзывы и меню.

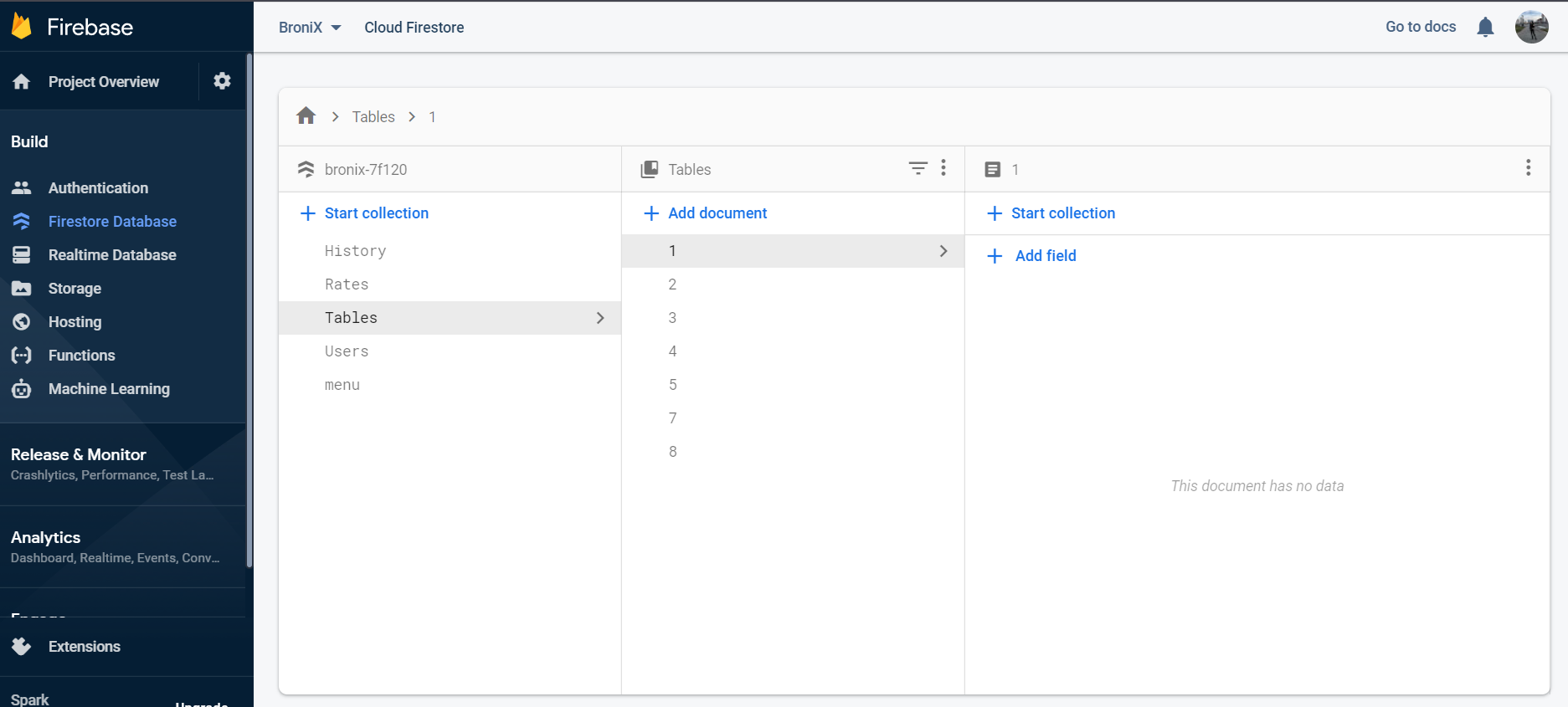


Рисунок 3.2 – База данных

На рисунке 3.3 приведен пример содержания сущности «Меню». Блюда всех категорий содержат название , стоимость и КБЖУ.

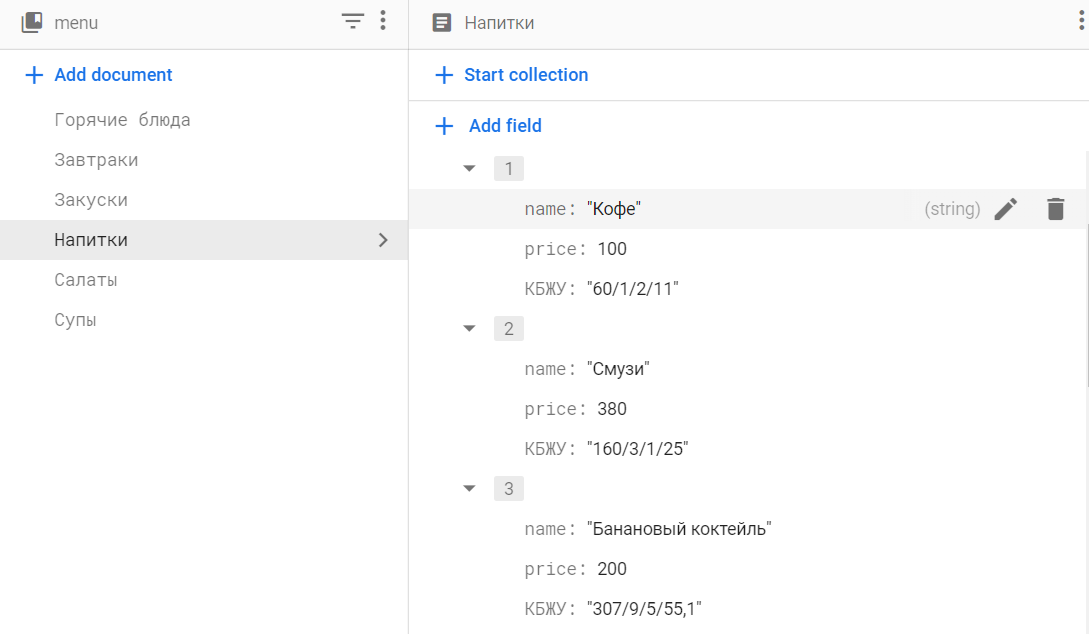


Рисунок 3.3 – Категория «Напитки»

На рисунках 3.4, 3.5 представлено заполнение данных по категориям соответствующими типами данных.

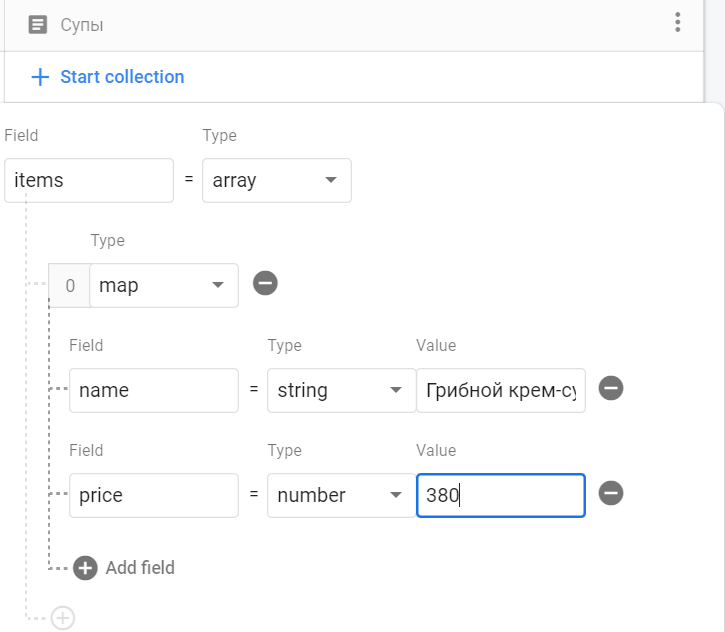


Рисунок 3.4 – Категория «Супы»

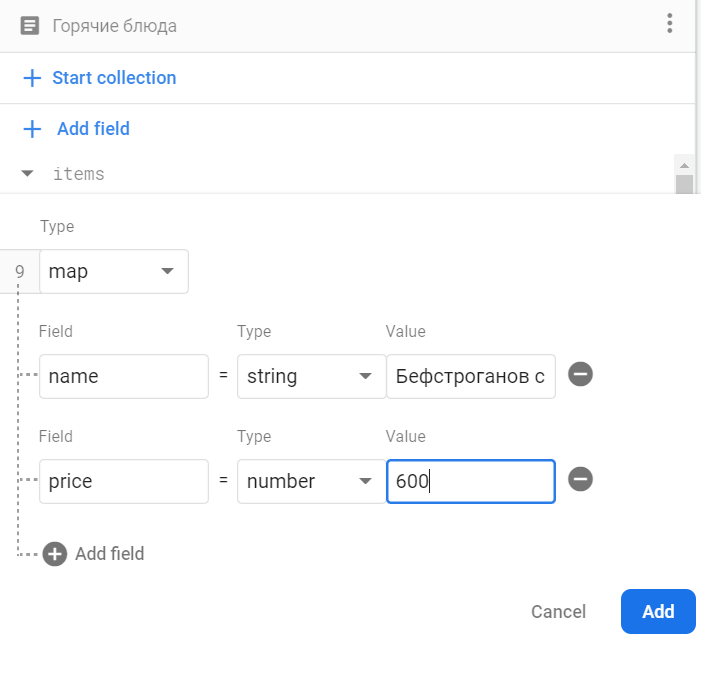


Рисунок 3.5 – Категория «Горячие блюда»

На рисунке 3.6 показано представление сущности «Отзывы».

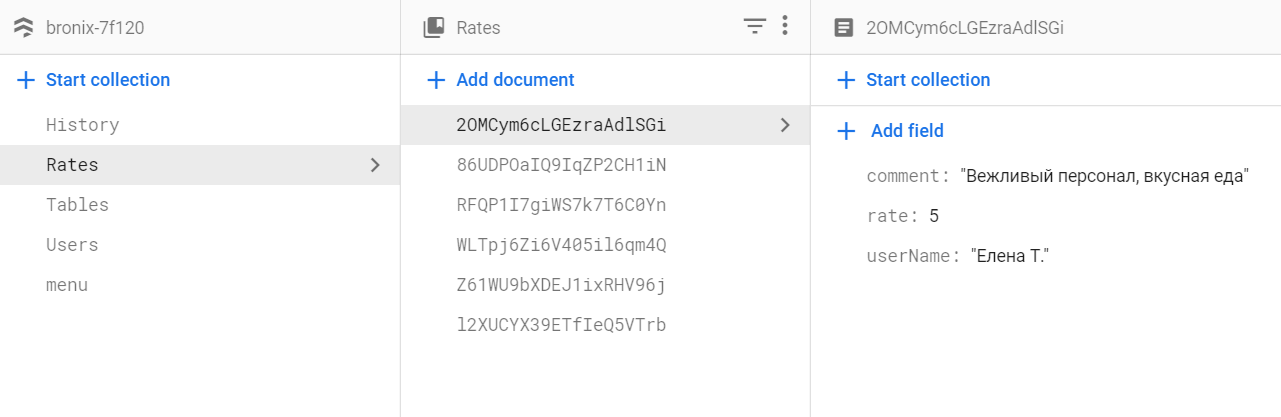


Рисунок 3.6 – Отзывы

# 3.3 Проектирование интерфейса мобильного приложения

Основная цветовая гамма приложения, представленная в подразделе 2.1, реализована в соответствии с рисунком 3.7.

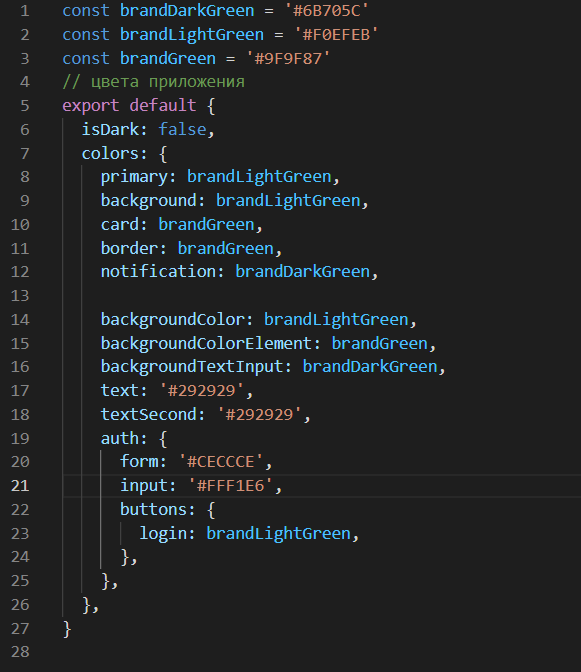


Рисунок 3.7 – Цвета приложения

Основная страница содержит фото ресторана «RestaurantImage», описание заведения «Description» и кнопки «Button».

Код дизайна основной страницы приложения представлен на рисунке 3.8.

Дизайн страниц регистрации, авторизации и восстановления пароля представлен на рисунке 3.9. Все элементы содержатся в основном слое View. Стиль текста у страниц одинаковый, так как приложение должно содержать единую структуру.

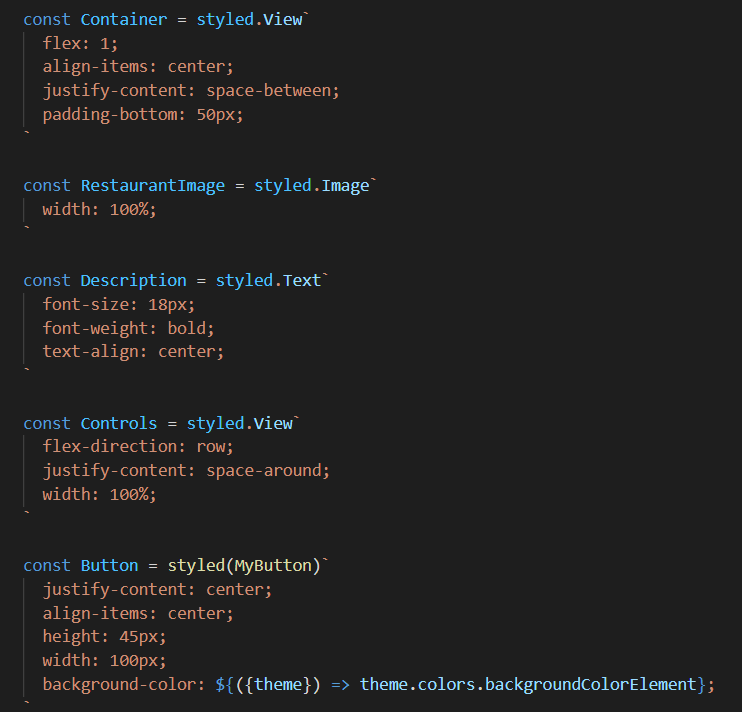


Рисунок 3.8 – Основная страница

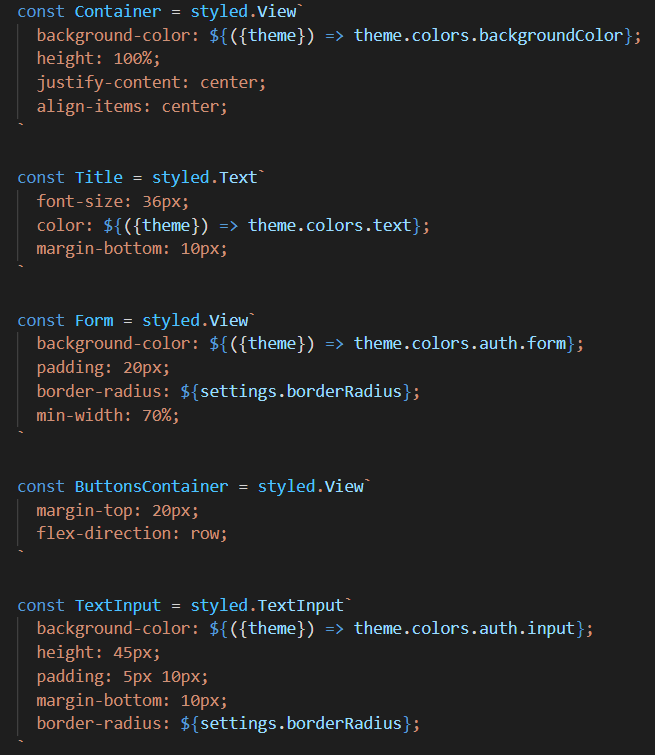


Рисунок 3.9 – Начальные страницы

Меню приложения содержит изображения иконок, которые были загружены в проект.

Пункт меню «Профиль пользователя» содержит страницу с данными пользователя: фото, имя, почтовый адрес. Данные можно изменить.

Код дизайна объекта представлен на рисунке 3.10.

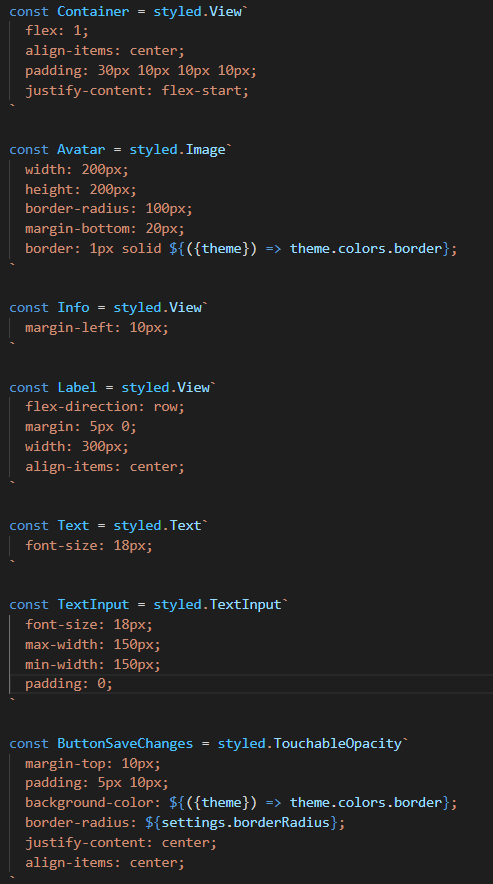


Рисунок 3.10 – Профиль пользователя

Код дизайна страницы меню ресторана представлен на рисунке 3.11.

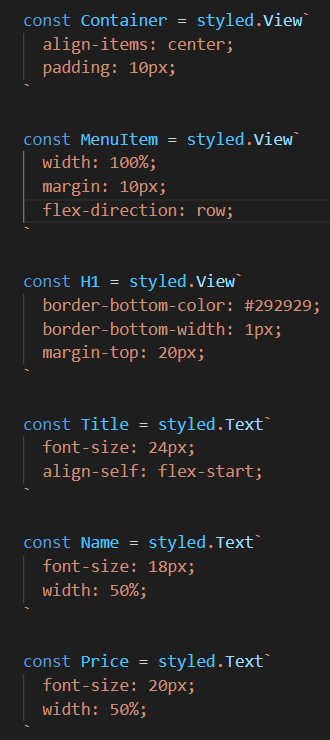


Рисунок 3.11 – Дизайн меню

# 3.4 Разработка программного обеспечения

Перед началом работы была произведена установка Visual Studio Code, Node.js и такие расширения, как React Native Tools, React Native Redux [13].

Создали основной класс приложения App и подключили необходимые библиотеки, как показано на рисунке 3.12. В качестве основного слоя используем SafeAreaView, так как он адаптируется под экран телефона [14].

Для хранения данных используется база данных реального времени FireBase.

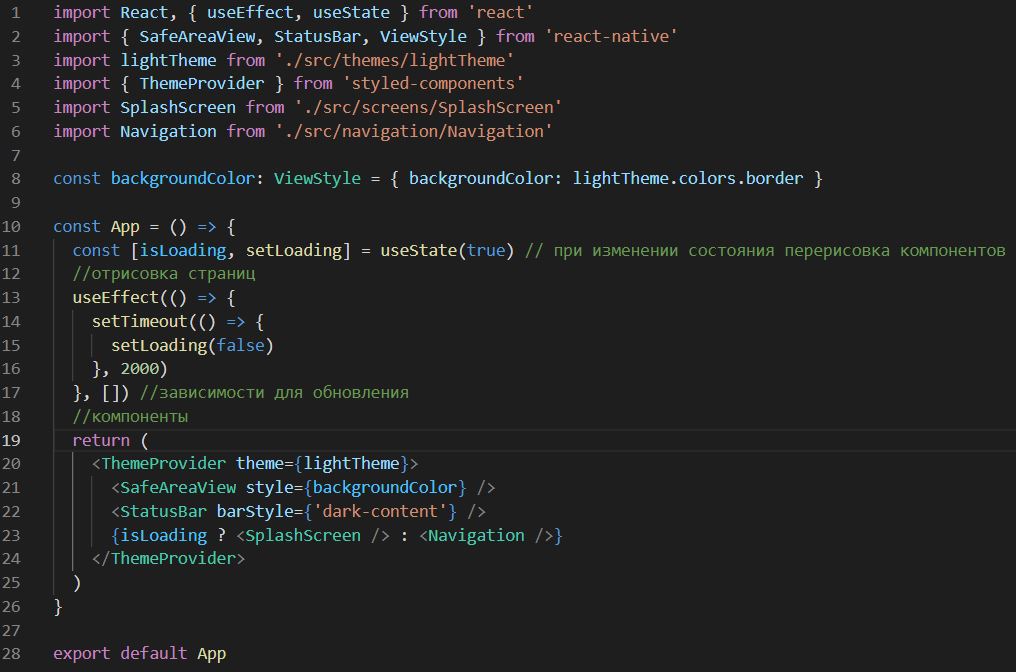


Рисунок 3.12 – Создание класса App

Для входа в приложение пользователь вводит логин и пароль, код проверки корректности введенных данных и вывод сообщения об ошибке представлен на рисунках 3.13 – 3.15.



Рисунок 3.13 – Проверка введенных данных

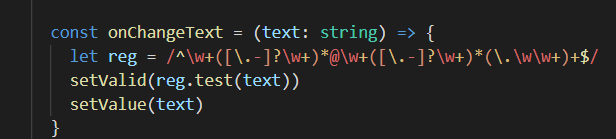


Рисунок 3.14 – Использование регулярных выражений

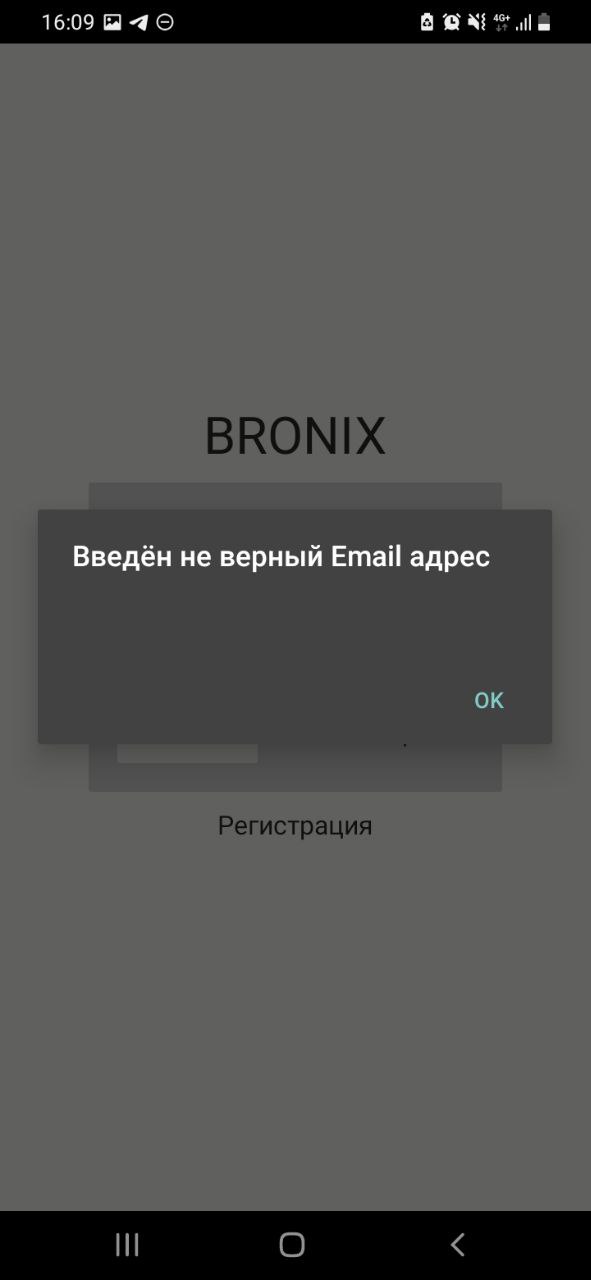


Рисунок 3.15 – Вывод сообщения об ошибке

После успешного входа в приложение, на основном экране пользователю доступны две кнопки. При нажатии на «Звонок» будет произведен вызов разработчику ПО (handlePressCall). После нажатия кнопки «Бронь» будет осуществлен переход к странице бронирования с выбором даты посещения (setOpenBookingPopup).

Обработка нажатия кнопок представлена на рисунке 3.16.

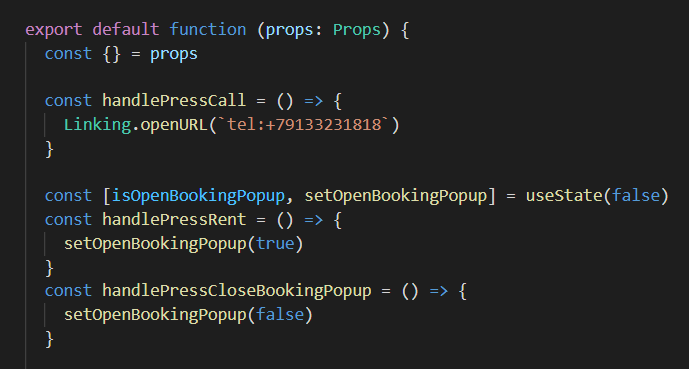


Рисунок 3.16– Обработка нажатия кнопок

Забронированный пользователем стол меняет цвет с зеленого на серый. Проверка стола на занятость (DatePicker) представлена на рисунке 3.17.

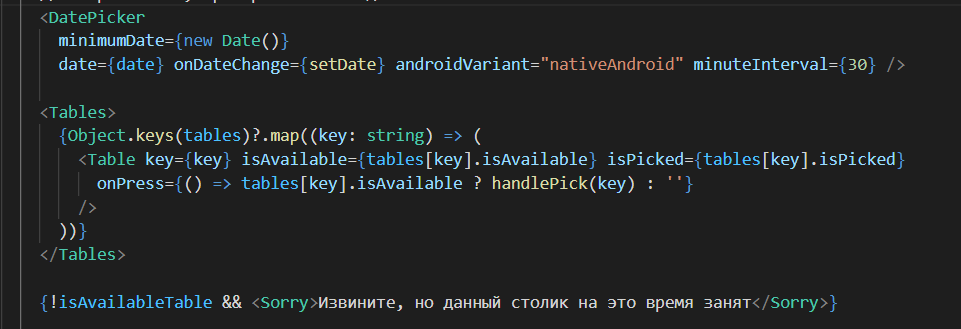


Рисунок 3.17 – Обработка выбора столика

Взаимодействие пользователя с приложением представлено в виде возможности создания отзывов о заведении.

Для достижения данной цели был создан класс RateScreen. После создания отзыва данные отправляются в БД и хранятся в коллекции «Rates» [15].

Метод onChangeText реагирует на изменения, SetRate содержит информацию об оценке, рисунок 3.18.



Рисунок 3.18 – Создание отзыва

Изменения в базе данных представлены на рисунке 3.19. Метод setAverageRate получает среднюю оценку, метод setReviews – комментарии.



Рисунок 3.19 – Изменения БД

Отображение комментария представлено на рисунке 3.20.

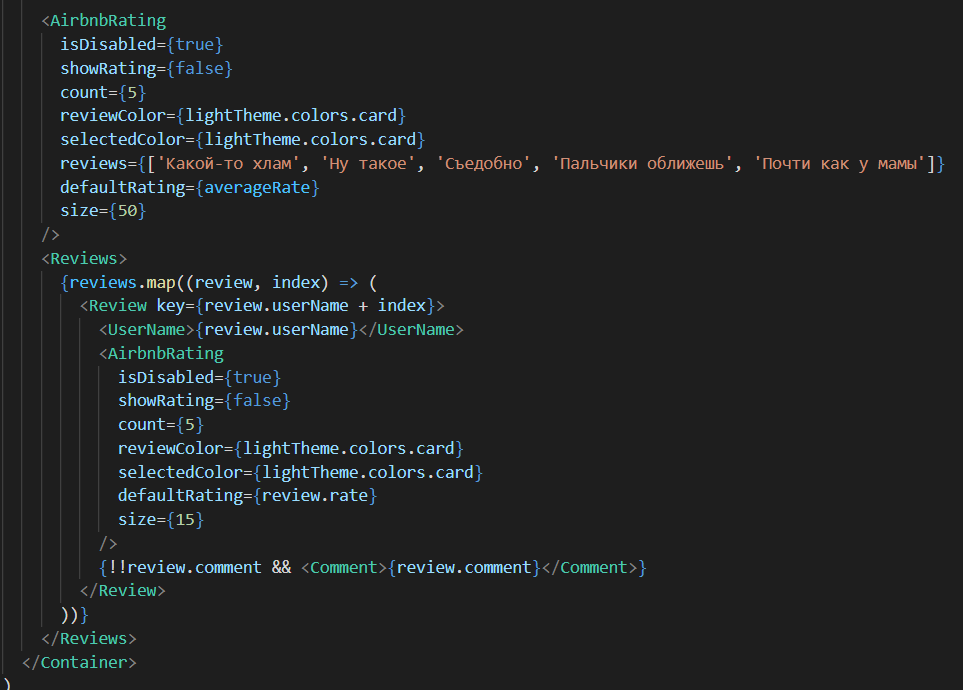


Рисунок 3.20 – Представление отзыва

Пункт меню «Профиль пользователя» содержит страницу с данными пользователя: фото, имя, почтовый адрес. Фото загружается в виде ссылки. Реализация представлена на рисунке 3.21.

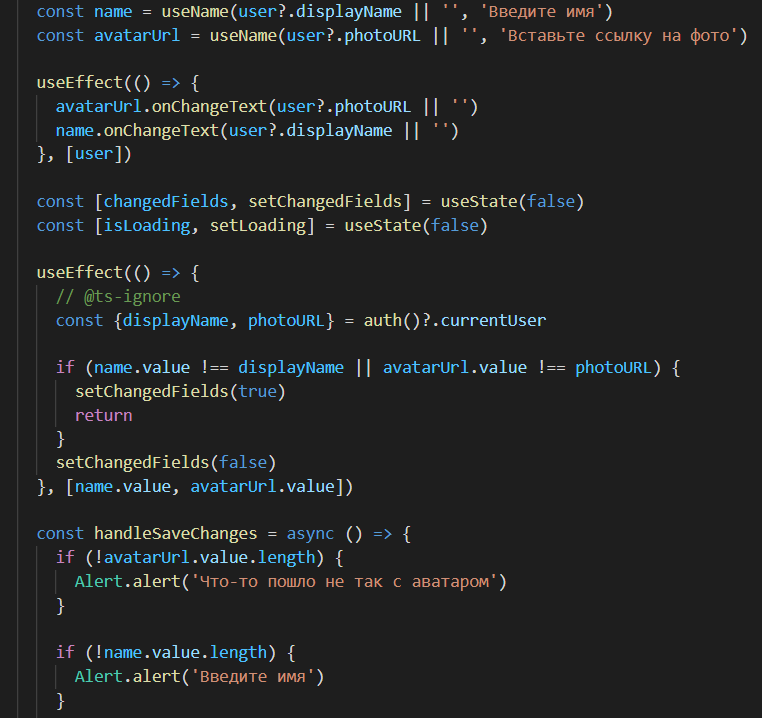


Рисунок 3.21 – Профиль пользователя

При нажатии на кнопку «Выйти» в главном меню пользователь выйдет из своего личного кабинета и перейдет к авторизации, рисунок 3.22.



Рисунок 3.22 – Выход из профиля

# 3.5 Выводы по третьему разделу

В данном разделе был подробно рассмотрен процесс разработки мобильного приложения по бронированию мест в ресторане. Данное описание основано на анализе предметной области, моделировании основных бизнес-процессов и требованиях, представленных в разделах 1 и 2.

Итогом является успешная установка мобильного приложения «BroniX», как показано на рисунке 3.23. На рисунке 3.24 представлена стартовая страница приложения.



Рисунок 3.23 – Установленное приложение



Рисунок 3.24 – Запуск приложения

# 4 Руководство пользователя

Для эксплуатации мобильного приложения от пользователей не должно требоваться специальных технических навыков, знания технологий или программных продуктов, за исключением общих навыков работы со смартфонами [16].

Чтобы запустить мобильное приложение, необходимо загрузить его на смартфон с ОС Android. После запуска откроется главный экран, на котором расположены поля для авторизации. Пользователю необходимо ввести свои логин и пароль, а затем нажать кнопку «Войти», рисунок 4.1.



Рисунок 4.1 – Вход в приложение

Если пользователь ввел неверный логин, то после нажатия на кнопку «Войти» на главном экране появится уведомление об ошибке, рисунок 4.2.

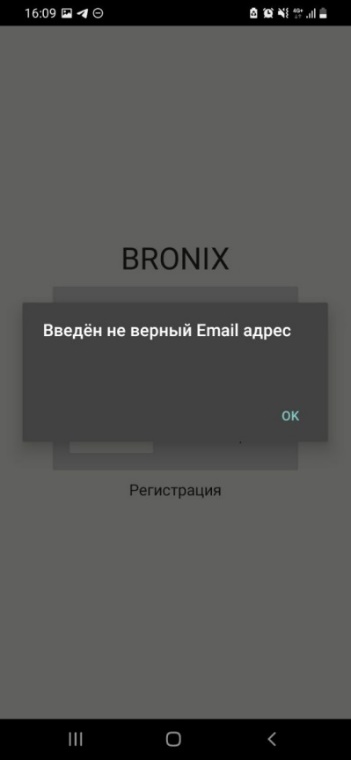


Рисунок 4.2 – Уведомление об ошибке

Если пользователь не зарегистрирован, то ему необходимо ввести данные для регистрации, рисунок 4.3.

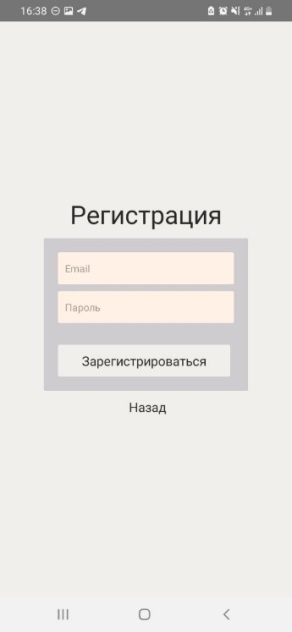


Рисунок 4.3 – Регистрация пользователя

После успешной авторизации пользователь увидит главную страницу приложения с меню в левом верхнем углу и двумя кнопками: «Бронь», «Звонок», рисунки 4.4, 4.5.



Рисунок 4.4 – Основная страница

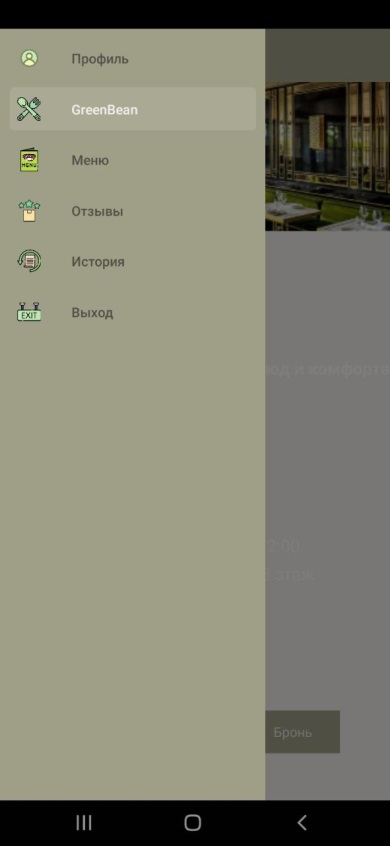


Рисунок 4.5 – Пункты меню

При нажатии на звонок будет произведен вызов разработчику ПО. После нажатия кнопки «Бронь» будет осуществлен переход к странице бронирования. Пользователю необходимо выбрать дату посещения и свободный столик, который отображается зеленым, рисунок 4.6.



Рисунок 4.6 – Карта зала

После успешного бронирования будет выведено сообщение, рисунок 4.7.

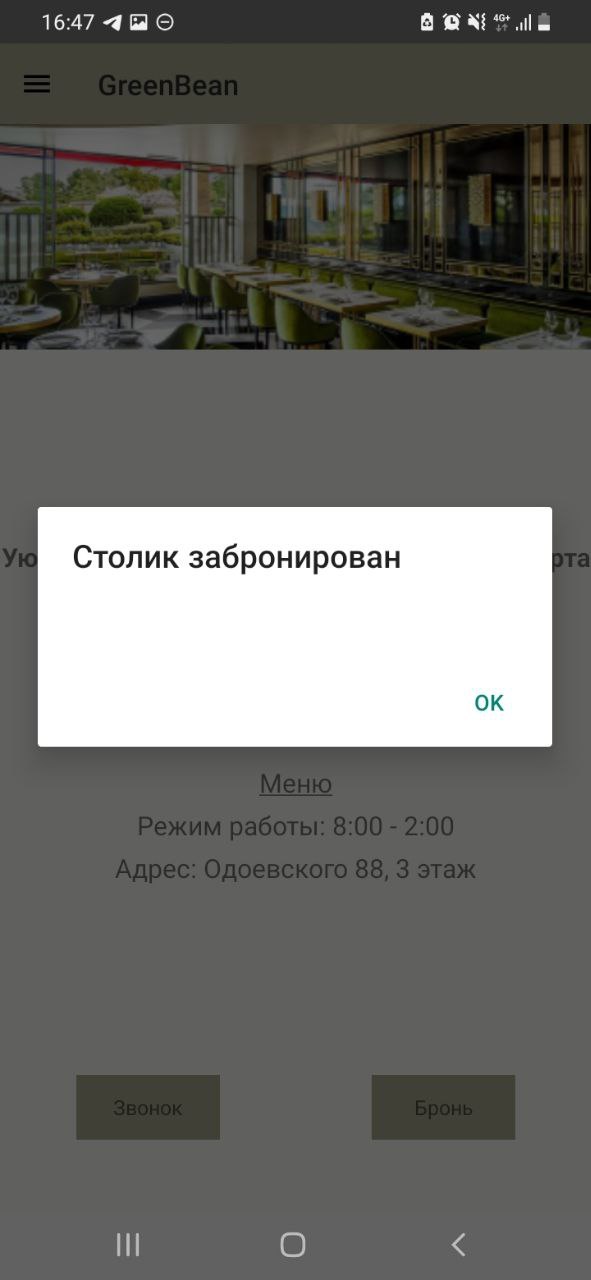


Рисунок 4.7 – Уведомление о бронировании

Затем бронирование отразится в истории заказов клиента, рисунок 4.8.



Рисунок 4.8 – История заказов

После посещения заведения пользователь может оставить свой отзыв или посмотреть рейтинг заведения, перейдя в соответствующий пункт меню, рисунки 4.9, 4.10.

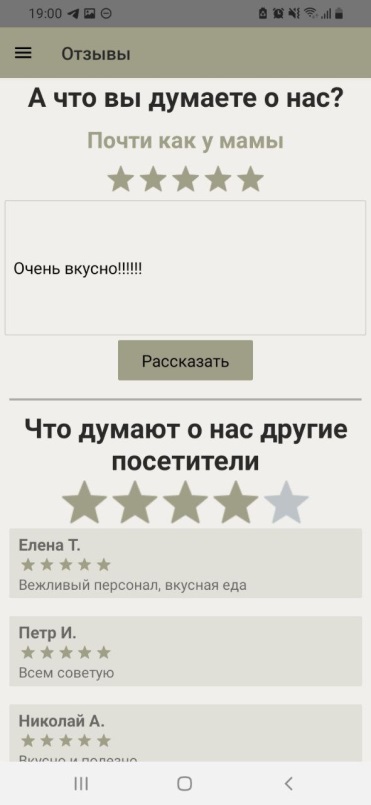


Рисунок 4.9 – Формирование отзыва



Рисунок 4.10 – Просмотр отзывов

Перейдя в категорию «Меню», можно посмотреть меню ресторана, рисунки 4.11 – 4.13.



Рисунок 4.11 – Меню «Горячие блюда»



Рисунок 4.12 – Меню «Завтраки»



Рисунок 4.13 – Меню «Салаты»

Пункт меню «Профиль пользователя» содержит страницу с данными пользователя: фото, имя, почтовый адрес.

Данные можно редактировать, рисунок 4.14.

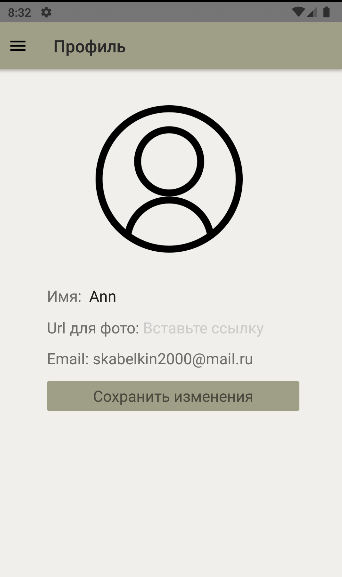


Рисунок 4.14 – Профиль пользователя

Если пользователь не помнит пароль, то после нажатия «Забыли пароль» на странице авторизации, откроется окно восстановления пароля, который будет выслан на e-mail, рисунок 4.15. Пример письма представлен на рисунке 4.16.

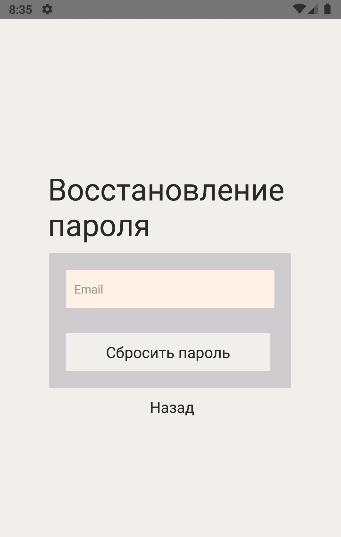


Рисунок 4.15 – Восстановление пароля

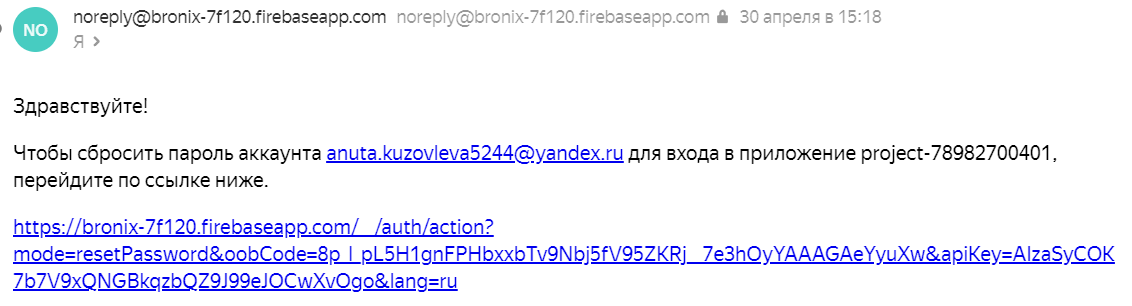


Рисунок 4.16 –Уведомление о восстановлении пароля

При нажатии на кнопку «Выйти» в главном меню пользователь выйдет из своего личного кабинета и перейдет к авторизации.

# ЗАКЛЮЧЕНИЕ

Смартфоны с операционной системой Android являются наиболее популярными, а также доступными устройствами на сегодняшний день. Многие производители мобильных устройств используют именно эту систему, поэтому выбор устройств на данной ОС достаточно велик. Именно поэтому целью выпускной квалификационной работы была выбрана разработка мобильного приложения для устройств с ОС Android для бронирования мест в ресторане.

Приложение содержит несколько деятельностей (экранных форм), одна из которых определяется как главная и отображается при первом запуске приложения. Деятельности приложения независимы друг от друга.

Были описаны методы и технологии проектирования приложения, а также развернутое обоснование выбора среды разработки и используемого языка программирования.

В качестве среды разработки была выбрана платформа React Native, поскольку она предоставляет множество инструментов для разработки. Разработка выполнялась на языке программирования JavaScript(TypeScript).

В ходе выполнения выпускной квалификационной работы была спроектирована БД для разрабатываемой ИС и описаны основные бизнес-процессы.

В выпускной квалификационной работе приведены скриншоты разработанного мобильного приложения по бронированию мест в ресторане.

В рамках выпускной квалификационной работы было разработано мобильное приложение для бронирования мест в ресторане устройств с ОС Android и руководство пользователя, которое позволит новому пользователю быстро сориентироваться в навигации приложения.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лучшие программы для кафе и ресторанов [Электронный ресурс]. – URL: <https://a2is.ru/catalog/programmy-dlya-kafe-i-restoranov> (Дата обращения: 18.03.2022).
2. Бесплатная программа для ресторана и кафе [Электронный ресурс]. – URL: <https://restoplace.cc/> (Дата обращения: 18.03.2022).
3. Полная автоматизация ресторанного бизнеса «СБИС Presto» [Электронный ресурс]. – URL: <https://sbis.ru/presto?utm_source=yandex&utm_> medium=cpc&utm\_campaign=yd\_presto\_poisk\_brend\_rf&utm\_content=presto%7Chead\_1%7Ctext\_1%7Cnone%7Cgid%7C4412084157%7Caid%7C10071847887%7C%7C%7C24461628046%7C%7Cdesktop&utm\_term=сбис%20престо&yadclid=30956672&yadordid=157925848&yclid=8752978544216768511 (Дата обращения: 19.03.2022).
4. Иванов Д.А. Моделирование на UML: учебно-методическое пособие/Д.А. Иванов, Ф.К.Новиков. – СПБ: СПБГУ ИТМО, 2018. – 200с.
5. Аксенов К. В. Обзор современных средств для разработки мобильных приложений // Новые информационные технологии в автоматизированных системах. 2017. №17. URL: https://cyberleninka.ru/article/n/obzor-sovremennyh-sredstv-dlya-razrabotki-mobilnyh-prilozheniy (Дата обращения: 19.03.2022).
6. Обзор платформы Eclipse [Электронный ресурс] – URL: <http://hightech.in.ua/content/art-eclipse-platform> (Дата обращения: 19.03.2022).
7. Васильева К. Н., Хусаинова Г. Я. Обзор программных средств для разработки мобильных приложений // Colloquium-journal. 2020. №2 (54). URL: https://cyberleninka.ru/article/n/obzor-programmnyh-sredstv-dlya-razrabotki-mobilnyh-prilozheniy (Дата обращения: 20.03.2022).
8. React Native. Learn once, write anywhere. [Электронный ресурс] – URL: <https://reactnative.dev/> (Дата обращения: 21.03.2022).
9. Современный учебник JavaScript. [Электронный ресурс] – URL: <https://learn.javascript.ru/> (Дата обращения: 21.03.2022).
10. Долженко А.И. Разработка приложений на базе WPF и Silverlight / А.И. Долженко. - 2-е изд., исправ. - М.: Национальный Открытый Университет «ИНТУИТ», 2019. - 453 с.
11. Павлова Е.А. Технологии разработки современных информационных систем на платформе Microsoft .NET: курс / Е.А. Павлова. - М.: Интернет-Университет Информационных Технологий, 2017. - 112 с. - (Основы информационных технологий). - ISBN 978-5-9963-0003-7.
12. База данных Firebase в реальном времени. [Электронный ресурс] –URL: <https://firebase.google.com/docs/database> (Дата обращения: 24.03.2022).
13. Проектирование информационных систем [Электронный ресурс]. – URL: https://compress.ru/article.aspx?id=11764 (Дата обращения: 25.03.2022).
14. Амелин К.С., Граничин О.Н., Кияев В.И., Корявко А.В. Введение в разработку приложений для мобильных платформ / Амелин К.С., Граничин О.Н., Кияев В.И., Корявко А.В. – СПб.: Издательство ВВМ, 2018. – 507 с.
15. Этапы разработки мобильных приложений [Электронный ресурс]. – URL: https://magora-systems.ru/etapy-razabotki-prilojeniy/ (Дата обращения: 28.03.2022).
16. Котляров В.П. Основы тестирования программного обеспечения / В.П. Котляров, Т.В. Коликова. – М.: Интернет-Университет Информационных Технологий, 2017. - 288 с. - (Основы информационных технологий). - ISBN 5-94774-406-4.