

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук
Кафедра программирования и информационных технологий

Мобильное приложение для бронирования мест в ресторанах и барах
Restobook

Курсовой проект
по дисциплине
Технологии программирования
09.03.04 Программная инженерия
Информационные системы и сетевые технологии

Преподаватель _____ В.С. Тарасов, ст. преподаватель _____.20__
Обучающийся _____ П.А. Путин, 3 курс, д/о
Обучающийся _____ М.В. Ефремов, 3 курс, д/о
Обучающийся _____ М.М.З. Насайр, 3 курс, д/о
Обучающийся _____ Ж.Ф. Ноэль, 3 курс, д/о
Руководитель _____ Е.Д. Проскуряков, ассистент

Воронеж 2024

Содержание

Определения, обозначения и сокращения	4
Введение.....	5
1 Постановка задачи.....	6
1.1 Обзор аналогов	6
1.1.1 ReMarked.....	7
1.1.2 Restoplace.....	8
1.1.3 GuestMe.....	9
2 Анализ предметной области	11
2.1 Глоссарий предметной области.....	11
2.2 Процесс бронирования столика.....	11
2.3 Учитываемые сведения о брони.....	12
2.4 Учет броней	12
2.5 Диаграмма сущность-связь	13
3 Реализация.....	14
3.1 Средства реализации	14
3.1.1 MVVM.....	14
3.1.2 Java	15
3.1.3 PostgreSQL.....	15
3.1.4 Keycloak	15
3.1.5 Nginx	15
3.1.6 Docker.....	15
3.1.7 AppMetrica	16
3.2 Логика приложения	16
3.3 Функциональные требования к разрабатываемой системе	16
3.3.1 Вход в приложение:.....	16
3.3.2 Учёт броней	17
3.3.3 Учёт сотрудников ресторана	17
3.3.4 Учёт столов и мест в ресторане.....	18
3.3.5 Регистрация нового ресторана	18
3.4 Нефункциональные требования к разрабатываемой системе	19
3.4.1 Требования по безопасности	19
3.4.2 Дополнительные требования.....	19

3.5 Структура классов серверной части приложения	19
3.6 Структура классов моделей серверной части	20
3.7 Хранение данных	21
3.8 Развёртывание приложения	22
3.9 Реализация интерфейса	22
3.9.3 Реализация интерфейса мобильного приложения.....	22
4 Аналитика	30
Заключение	31
Список использованных источников	32
ПРИЛОЖЕНИЕ А VRMN диаграмма	33
ПРИЛОЖЕНИЕ Б Диаграмма прецедентов	34
ПРИЛОЖЕНИЕ В Диаграммы состояний.....	35

Определения, обозначения и сокращения

Термин	Определение
Администратор вендора	сотрудник поставщика разрабатываемой системы, принимающий от администраторов ресторанов заявки на создание учётной записи в разрабатываемом мобильном приложении и администрирующей сервер приложения
Администратор ресторана	сотрудник ресторана, подавший заявку на создание учётной записи в мобильном приложении и в последующем имеющий доступ к редактированию информации о сотрудниках и столах
Клиентская сторона	компьютер, использующий ресурсы сервера и предоставляющий пользователю возможность взаимодействия с системой
Логин	частный случай идентификатора, который используется для представления конкретного пользователя в системе
Сотрудник ресторана	человек, имеющий учётную запись в мобильном приложении
Сервер, серверная часть	компьютер, обслуживающий другие компьютеры (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач
Цифровизация	это процесс превращения аналоговых данных и рабочих процессов в цифровой формат
Popup	всплывающее окно, которое появляется внезапно, без запроса пользователя, и отображается поверх экрана, на котором находился пользователь
REST API (REST, Representational State Transfer)	стиль архитектуры программного обеспечения для построения масштабируемых веб-приложений
TabBar	панель вкладок в нижней части экрана, позволяющая быстро переключаться между разделами приложения
Onboarding	обучающая функциональность в приложении, появляющаяся при первом запуске для ознакомления пользователя с продуктом

Введение

Современное деловое сообщество сталкивается с растущей необходимостью в цифровизации бизнес-процессов для повышения эффективности и конкурентоспособности. Одной из наиболее актуальных областей, где цифровизация может принести значительные выгоды, является сфера общественного питания.

В наше время все больше людей стремятся питаться в ресторанах и кафе, и, соответственно, возрастает необходимость в удобной системе бронирования столов. Зачастую клиенты испытывают сложности с поиском свободного места, а рестораторы сталкиваются с проблемой организации работы без потери потенциальных клиентов из-за неэффективного управления столиками.

Целью нашего проекта является создание приложения Restobook, которое поможет рестораторам и их клиентам в оптимизации процесса бронирования столов. Мы уверены, что цифровизация этой области поможет увеличить эффективность работы ресторанов и улучшить опыт клиентов.

Для достижения этой цели мы планируем проанализировать статистику использования двух популярных приложений для бронирования столов - Restoplace и GuestMe. Мы изучим, какие функции данных приложений наиболее востребованы пользователями, какие проблемы они сталкиваются при использовании, и какие возможности можно предложить для улучшения процесса бронирования.

На основе полученных данных мы разработаем удобный и интуитивно понятный интерфейс приложения Restobook, который будет выгодно отличаться от конкурентов. Мы уверены, что наше приложение принесет реальную пользу как рестораторам, так и их клиентам, и поможет оптимизировать процесс бронирования столов в общественных заведениях.

1 Постановка задачи

Целью данного курсового проекта является создание мобильного приложения для сотрудников ресторанов и баров по автоматизации учёта броней посетителей. Данное мобильное приложение разрабатывается с целью:

- в период с июня по август 2024 года предоставить доступ к системе трём ресторанам;
- в течение месяца после начала использования приложения сотрудниками ресторана скорость обработки заявок на бронь должна увеличиться на 10 процентов в сравнении с периодом до подключения;
- удовлетворённость сотрудников ресторанов процессом обработки заявок на бронирование после месяца использования приложения должна составлять не меньше 8, где 1 – не удовлетворён, 10 – полностью удовлетворён.

Приложение должно обладать следующей функциональностью:

- учёт броней;
- учёт сотрудников ресторана;
- учёт столов и мест в ресторане;

1.1 Обзор аналогов

В ходе исследования рынка приложений по бронированию мест в ресторанах и барах было выявлено 3 прямых конкурента. Таблица 1 содержит результаты проведённого конкурентного исследования.

Таблица 1 — Результаты конкурентного исследования

Характеристика	ReMarked	Restoplace	GuestMe
Просмотр занятости столов	+	+	+
Просмотр очереди брони конкретного стола	-	+	-
Информация о госте при подаче заявки	-	-	+
Интеграция с системами автоматизации ресторана	+	-	+
Стоимость	Договорная	1100 Р/мес.	2900 Р/мес.

1.1.1 ReMarked

Облачный сервис, клиент которого работает на телефонах, планшетах и ноутбуках. Заказы можно принимать через виджет, телеграм-бот и приложение. Система бронирования ReMarked рассчитана как на сетевые проекты, так и на одиночные заведения с высокой проходимостью. Позволяет клиентам автоматически резервировать места на сайте, гости получают уведомления/напоминания о брони через Whatsapp, есть возможность вносить депозиты и отправлять данные о резерве на кассу. Рисунок 1, рисунок 2 и рисунок 3 демонстрируют элементы интерфейса приложения ReMarked.

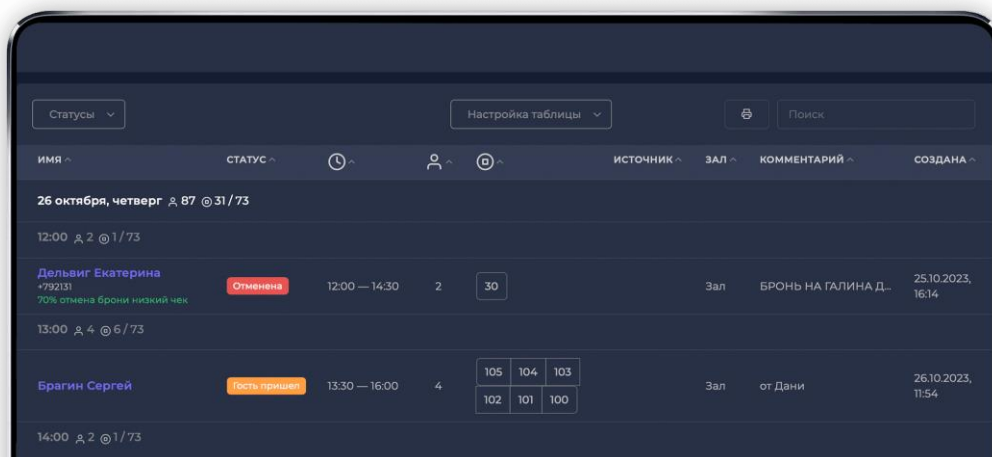


Рисунок 1 — Список броней

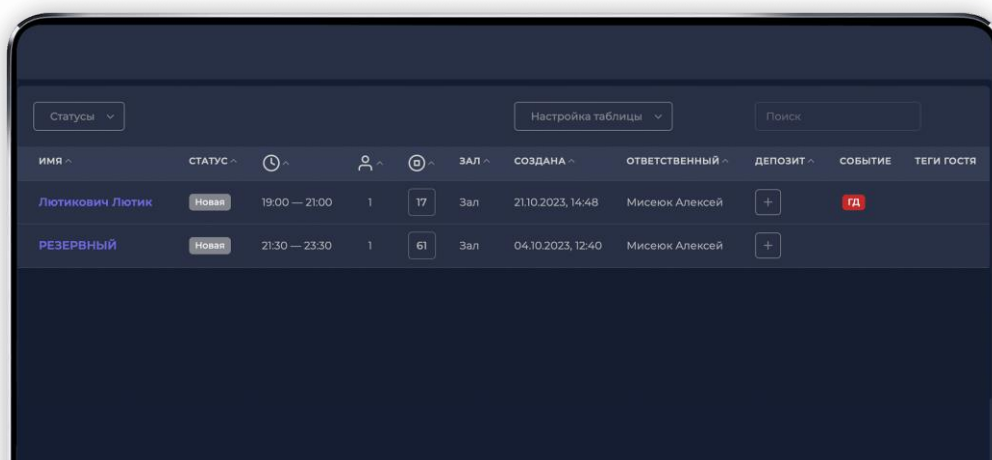


Рисунок 2 — Лист ожидания

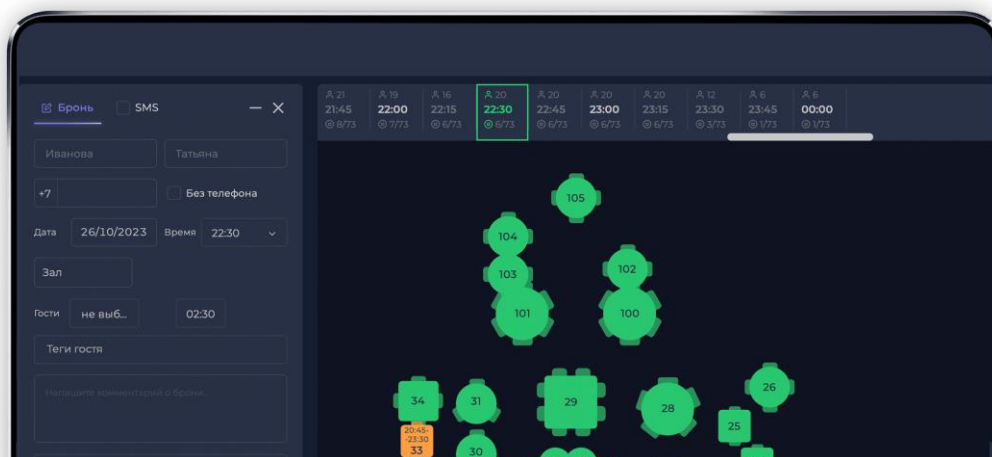


Рисунок 3 — Создание брони

1.1.2 Restoplace

Restoplace — это программа, которая позволяет управлять посадкой гостей в ресторанах и кафе с помощью приложения для хостес. Система

содержит основной функционал — подключение депозитов за бронирование столиков, напоминание по СМС для гостя о забронированном столе, групповое бронирование, подробная статистика о загруженности заведения и всех заявках гостей. В терминале программы Рестоплейс хостес может совершать пересадку гостей между столами, бронировать сразу несколько столиков на одного гостя, бронировать банкетные залы, продлевать, сдвигать время. Рисунок 4 демонстрирует интерфейс приложения Restoplace.

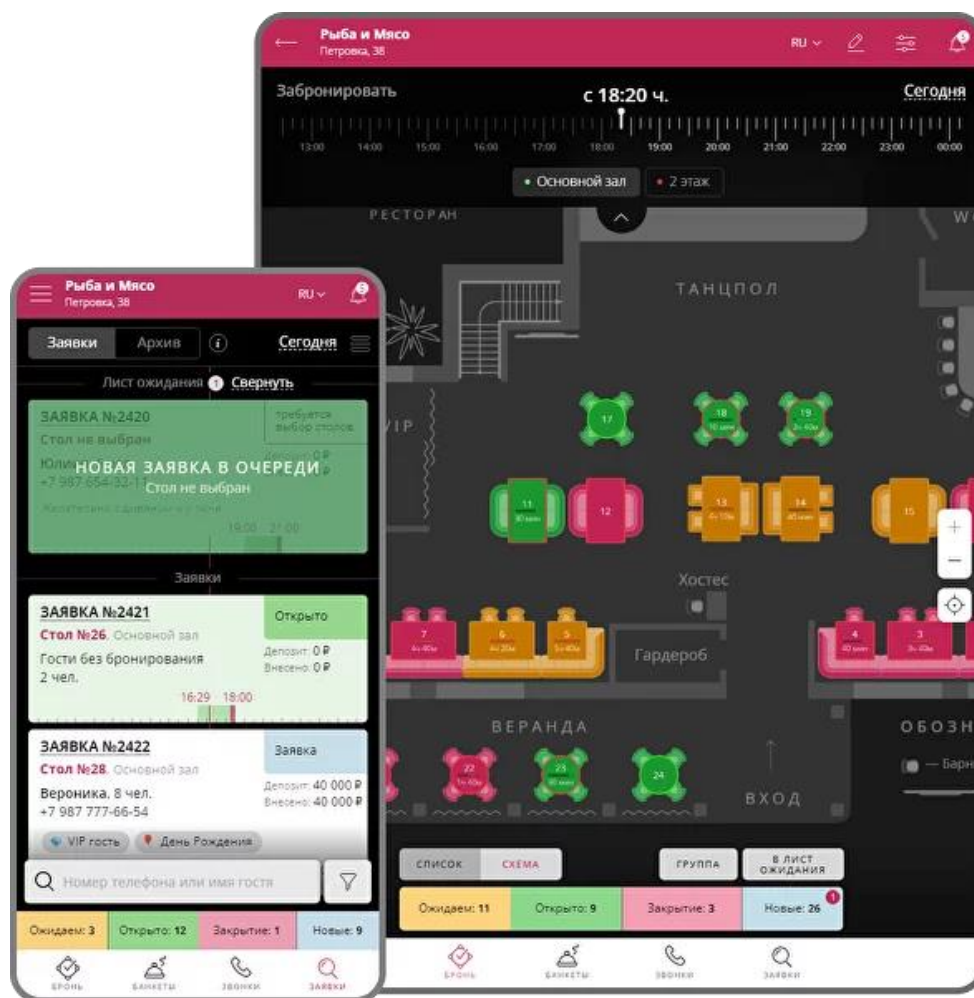
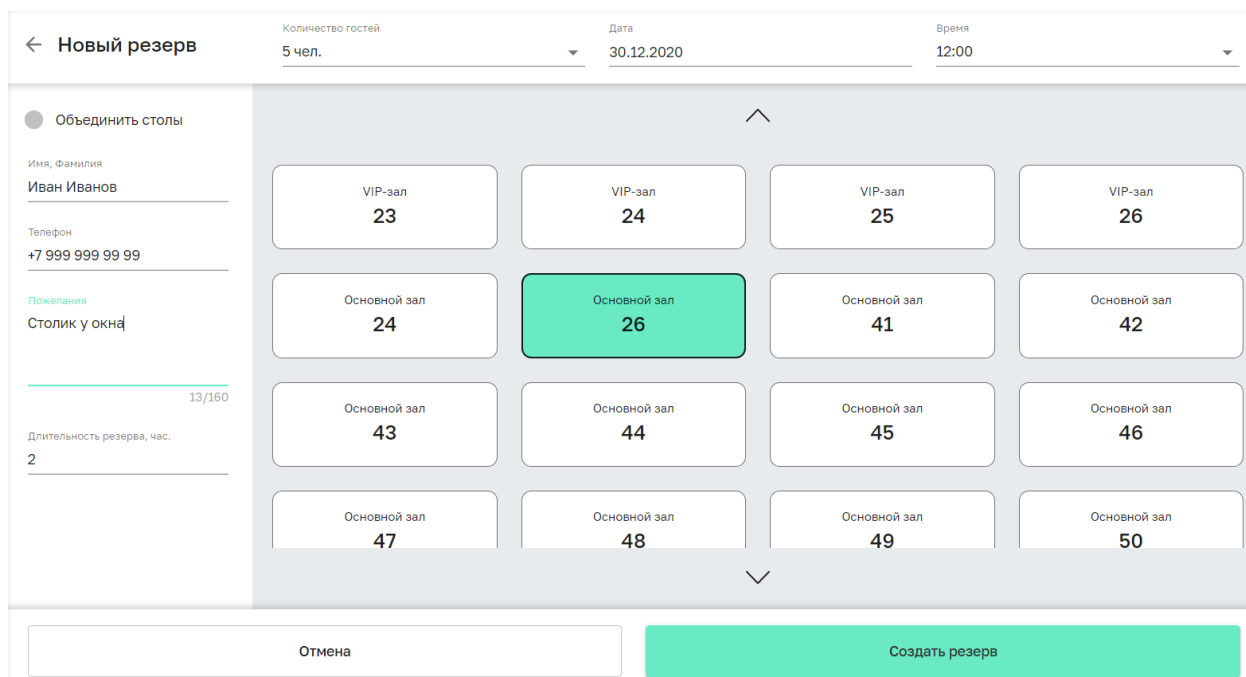


Рисунок 4 — Демонстрация интерфейса приложения Restoplace

1.1.3 GuestMe

Рестораны, которые уже работают на iiko, могут не искать отдельный сервис для бронирования столов, а просто подключить дополнительный модуль — GuestMe. Он поддерживает основные функции аналогичных инструментов: мгновенная интеграция с системой автоматизации, быстрое получение информации о зале и гостях из iiko, при поступлении звонка от

гостя персонал сразу видит всю информацию о нем. Есть удобная схема залов для оформления резерва, работа с банкетными. Виджеты бронирования для любых площадок — сайт, социальные сети, карты. Невозможность овербукинга. Функцию бронирования также можно отключать в некоторых дни или часы. Рисунок 5 демонстрирует интерфейс приложения GuestMe.



← Новый резерв

Количество гостей: 5 чел.

Дата: 30.12.2020

Время: 12:00

Объединить столы

Имя, Фамилия: Иван Иванов

Телефон: +7 999 999 99 99

Пожелания: Столик у окна

13/160

Длительность резерва, час: 2

VIP-зал 23	VIP-зал 24	VIP-зал 25	VIP-зал 26
Основной зал 24	Основной зал 26	Основной зал 41	Основной зал 42
Основной зал 43	Основной зал 44	Основной зал 45	Основной зал 46
Основной зал 47	Основной зал 48	Основной зал 49	Основной зал 50

Отмена

Создать резерв

Рисунок 5 — Окно создания нового резерва

2 Анализ предметной области

2.1 Глоссарий предметной области

Гость – это посетитель ресторана.

Книга резервации – это полиграфическое изделие, используемое для учёта броней.

Бронь – это платная услуга, которая позволяет гостю заранее зарезервировать место в заведении. Стоимость брони не включается в счет заказа, но гарантирует, что место будет доступно в определенное время и день.

Ожидающая бронь – это бронь, информация о которой внесена в книгу резервации, но гость ещё не пришёл.

Открытая бронь – это бронь, гость которой уже пришёл в ресторан.

Закрытая бронь – это бронь, гость которой освободил зарезервированный столик.

2.2 Процесс бронирования столика

Работа ресторана с точки зрения бронирования столов всегда проходит по одному алгоритму (см. рисунок 6):

- а) принять звонок или заявку от гостя;
- б) узнать дату, время, количество людей и желаемое время брони;
- в) выбрать подходящий под критерии из пункта б столик;
- г) ввести данные в программу или книгу резервации. Если это требуется, то оставить комментарий к брони.

Также у гостя есть возможность отменить бронь.



Рисунок 6 — Диаграмма активности работы сотрудника ресторана с бронью

2.3 Учитываемые сведения о брони

При резервации брони в программу или книгу резервов вносятся следующие данные:

- дата и время брони;
- имя клиента;
- ФИО сотрудника, создавшего бронь;
- номера забронированных столиков;
- статус брони;
- дополнительную информацию в виде комментария.

2.4 Учет броней

Для авторизованного сотрудника ресторана система должна предоставлять доступ к:

- просмотру забронированных столов;
- просмотру очереди броней;
- созданию, редактированию и удалению информации о брони.

При создании и редактировании брони система должна позволять учитывать следующие сведения:

- дата и время брони;
- имя клиента;
- ФИО сотрудника, создавшего бронь;
- номера забронированных столиков;
- статус брони;
- дополнительную информацию в виде комментария.

Как результат, эта информация должна сохраниться в системе и быть доступной для всех сотрудников ресторана.

2.5 Диаграмма сущность-связь

На основе анализа предметной сущности были выделены сущности «Сотрудник», «Стол», «Ресторан», «Бронь» (см. рисунок 7).

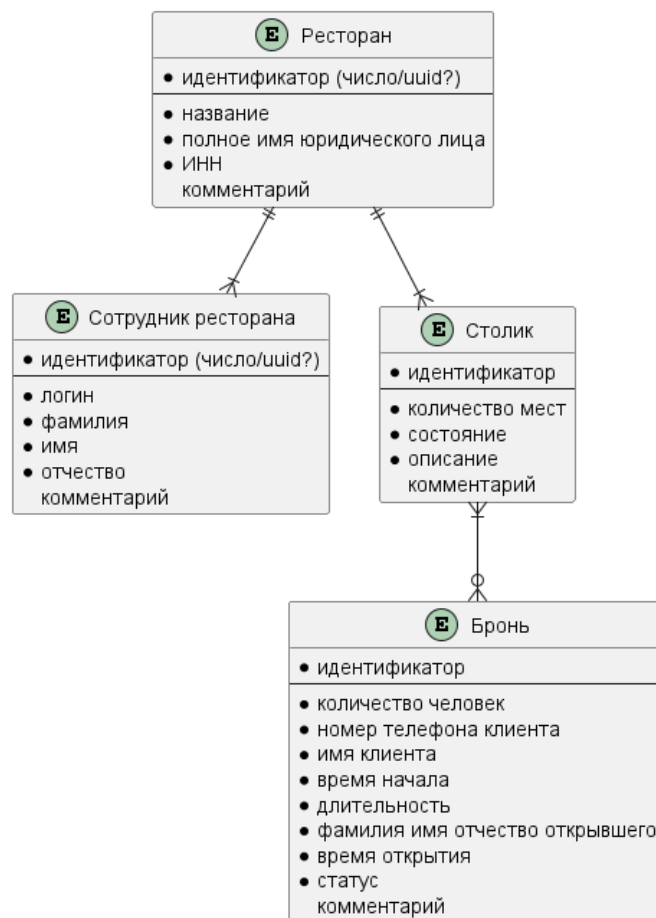


Рисунок 7 — Диаграмма сущностей и связей системы

3 Реализация

3.1 Средства реализации

Для реализации клиентской части веб-приложения использовался фреймворк Flutter версии 3.19.1 и язык программирования Dart 3.3.0.

Flutter – это написанный на Dart кроссплатформенный фреймворк для разработки мобильных, веб- и десктопных приложений. К основным плюсам Flutter можно отнести:

- ускоренную разработку;
- высокую производительность;
- развитое и активное сообщество разработчиков.

Ускорение разработки достигается за счёт использования виджетов, что позволяет применять объектно-ориентированный подход к любому элементу интерфейса.

Высокая производительность достигается за счёт графического движка Skia, обеспечивающего компиляцию исходного кода приложения непосредственно в нативный код платформы, а также за счёт рендеринга пользовательского интерфейса на видеокарте.

Благодаря популярности Flutter существует развитая система сторонних программных пакетов, позволяющих расширить функциональность фреймворка [1].

3.1.1 MVVM

MVVM (Model-View-ViewModel) – это архитектурный шаблон, используемый при разработке приложений для разделения пользовательского интерфейса и логики выполнения.

За счёт привязки данных модели и представления через `viewmodel` обеспечивается автоматическое обновление пользовательского интерфейса при изменении данных модели. Это позволяет облегчить и ускорить разработку приложений.

Также за счёт гибкости данного шаблона появляется возможность изменять пользовательский интерфейс без необходимости большого изменения логики выполнения.

3.1.2 Java

Для реализации серверной части приложения использовался язык программирования Java 21 и фреймворк Spring Boot 3.2.4.

Язык программирования Java обладает простым синтаксисом, обширной стандартной библиотекой для работы со структурами данных и сетью, а также удобными инструментами сборки, например, Maven, позволяющими быстро подключать необходимые зависимости.

Фреймворк Spring Boot 3 [2] обеспечивает лёгкость создания веб-приложений за счёт развитой системы аннотаций Java.

3.1.3 PostgreSQL

Для хранения данных используется реляционная база данных PostgreSQL 16.2. За счёт производительности в долгосрочной перспективе позволит значительно увеличить клиентскую базу без необходимости смены технологии.

3.1.4 Keycloak

Для аутентификации пользователей используется Keycloak 23.0.7. В будущем этот сервис обеспечит нам возможность лёгкой реализации входа через сторонние сервисы.

3.1.5 Nginx

В качестве веб-сервера для клиентской части использовался Nginx версии 1.25.

Данный веб-сервер обеспечивает высокую скорость обработки запросов при работе со статическим контентом, не требующим постоянного обновления.

3.1.6 Docker

Для развёртывания приложения использовался Docker версии 26.1.1.

Docker – это инструмент контейнеризации приложений, позволяющий обеспечить переносимость приложений и ускорить разработку. За счёт изолированности процессов приложение не сможет затронуть внешнее окружение, что обеспечивает повышенную безопасность.

Инструмент Docker compose позволяет легко настроить запуск нескольких контейнеров за счёт декларативного описания запускаемых сервисов и сетей.

3.1.7 AppMetrica

Для сбора данных о работе приложения, пользовательских событиях и аналитики используется сервис AppMetrica [3]. За счёт удобного SDK и гибкой настройки данный сервис позволяет легко настроить отправку событий пользователей, а также строить на основе полученных данных аналитические отчёты, например, воронки конверсий.

3.2 Логика приложения

В системе пользователь имеет одну из ролей:

- неавторизованный пользователь;
- администратор ресторана;
- сотрудник ресторана;
- администратора вендора.

3.3 Функциональные требования к разрабатываемой системе

Обзор функциональных требований показан на рисунке Б.1. Далее приведено подробное описание функциональных требований.

3.3.1 Вход в приложение:

- при первом включении приложения пользователю должен быть показан onboarding;
- приложение должно позволять войти по логину и паролю;
- пользователю должна быть доступна информация о способе подачи заявки на регистрацию в приложении.

3.3.2 Учёт броней

Для авторизованного сотрудника ресторана система должна предоставлять доступ к:

- просмотру забронированных столов;
- просмотру очереди броней;
- созданию, редактированию и удалению информации о брони.

При создании и редактировании брони система должна позволять учитывать следующие сведения:

- дата и время брони;
- имя клиента;
- ФИО сотрудника, создавшего бронь;
- номера забронированных столиков;
- статус брони (см. рисунок в.1);
- дополнительную информацию в виде комментария.

Как результат, эта информация должна сохраниться в системе и быть доступной для всех сотрудников ресторана.

3.3.3 Учёт сотрудников ресторана

Каждому сотруднику ресторана система должна предоставлять доступ к:

- просмотру личного кабинета;
- возможности смены пароля.

Администратору ресторана система должна предоставлять доступ к:

- просмотру информации о сотрудниках ресторана;
- созданию, редактированию и удалению информации о сотруднике ресторана.

При создании и редактировании учётной записи нового сотрудника система должна позволять учитывать следующие сведения:

- фамилию, имя и отчество сотрудника;
- логин сотрудника;
- дополнительную информацию в виде комментария.

Как результат, эта информация должна сохраниться в системе и быть доступной для администратора ресторана и владельца этой учётной записи.

3.3.4 Учёт столов и мест в ресторане

Каждому сотруднику ресторана система должна предоставлять доступ к просмотру информации о столах в ресторане.

Администратору ресторана система должна предоставлять доступ к созданию, редактированию и удалению информации о столах.

При создании и редактировании информации о столе система должна позволять учитывать следующие сведения:

- номер стола;
- количество мест за столом;
- состояние стола (см. рисунок в.2);
- дополнительную информацию в виде комментария.

Как результат, эта информация должна сохраниться в системе и быть доступной для всех сотрудников ресторана.

3.3.5 Регистрация нового ресторана

Система должна предоставлять администратору вендора возможность регистрировать новые рестораны, учитывая следующие сведения:

- название ресторана;
- полное наименование юридического лица, владеющего рестораном;
- ИНН;
- дополнительную информацию в виде комментария;
- фамилию, имя и отчество администратора ресторана.

Администратор вендора получает сведения из заявки, которые пользователь присылает ему на электронную почту (см. рисунок а.1). Как результат, эта информация должна сохраниться в системе и быть доступной для администратора вендора.

3.4 Нефункциональные требования к разрабатываемой системе

3.4.1 Требования по безопасности

- Обмен данных между клиентом и сервером должен осуществляться по протоколу HTTPS;
- пароли пользователей должны храниться в базе данных в хешированном виде; для хеширования должен использоваться алгоритм PBKDF2;
- сотрудники ресторана должны иметь доступ на просмотр и изменение информации только в своём ресторане.

3.4.2 Дополнительные требования

При первом запуске мобильного приложения пользователь должен иметь возможность ознакомиться с основными функциями приложения.

3.5 Структура классов серверной части приложения

Классовая структура серверной части приложения организована по принципу слоистой архитектуры с выделением классов контроллеров, обеспечивающих обработку HTTP запросов; сервисов, содержащих бизнес-логику приложения, и репозитория Spring data JPA, обеспечивающих доступ к данным в базе данных (см. рисунок 8).

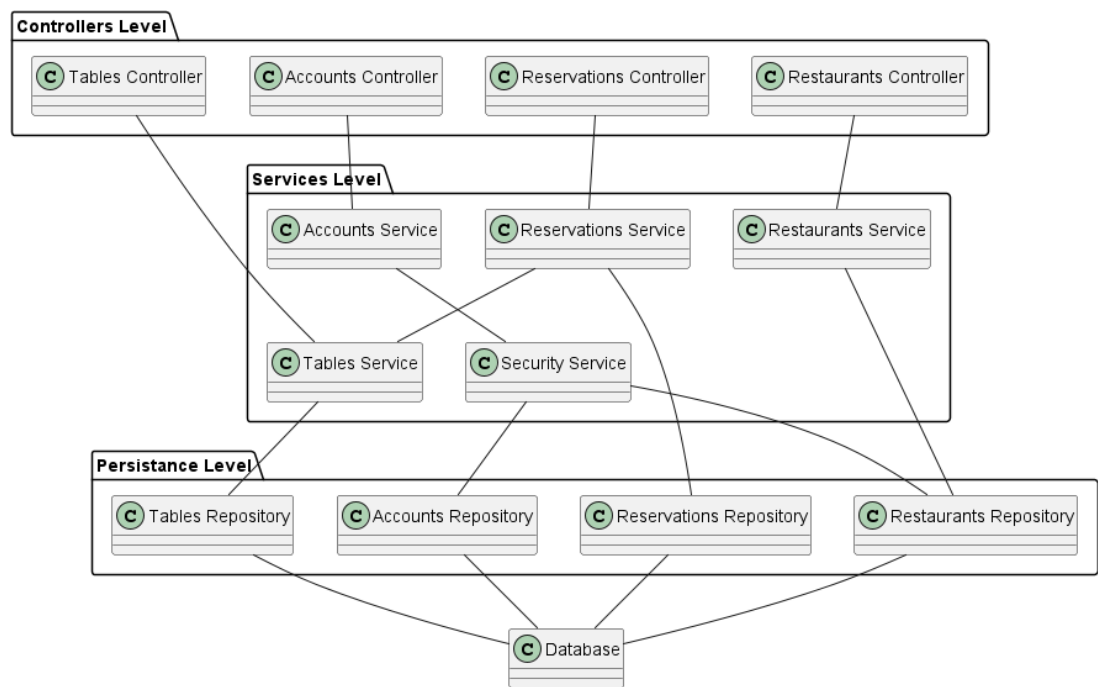


Рисунок 8 — Диаграмма классов серверной части приложения

3.6 Структура классов моделей серверной части

Сущности, выявленные в ходе анализа предметной области, представлены в программе серверной части классами моделей (см. рисунок 9).

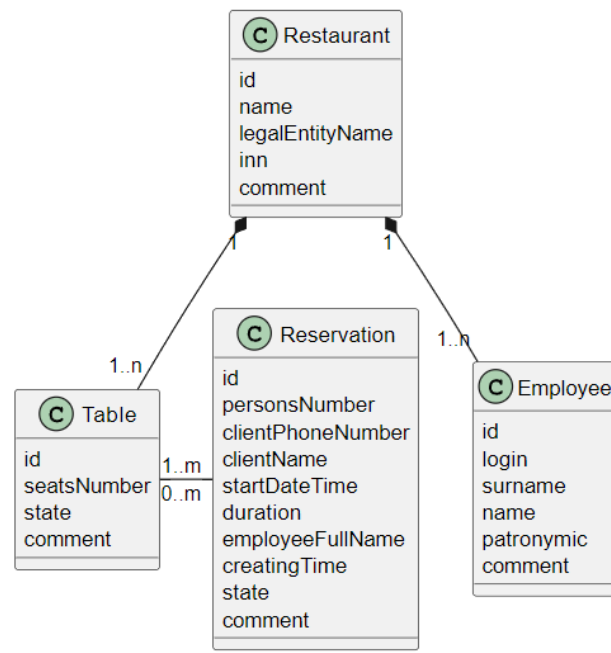


Рисунок 9 — Классы модели

3.7 Хранение данных

Данные приложения хранятся в реляционной базе данных PostgreSQL. Физическая схема базы данных показана рисунке 10. Данные для аутентификации пользователей хранятся в базе данных Keycloak.

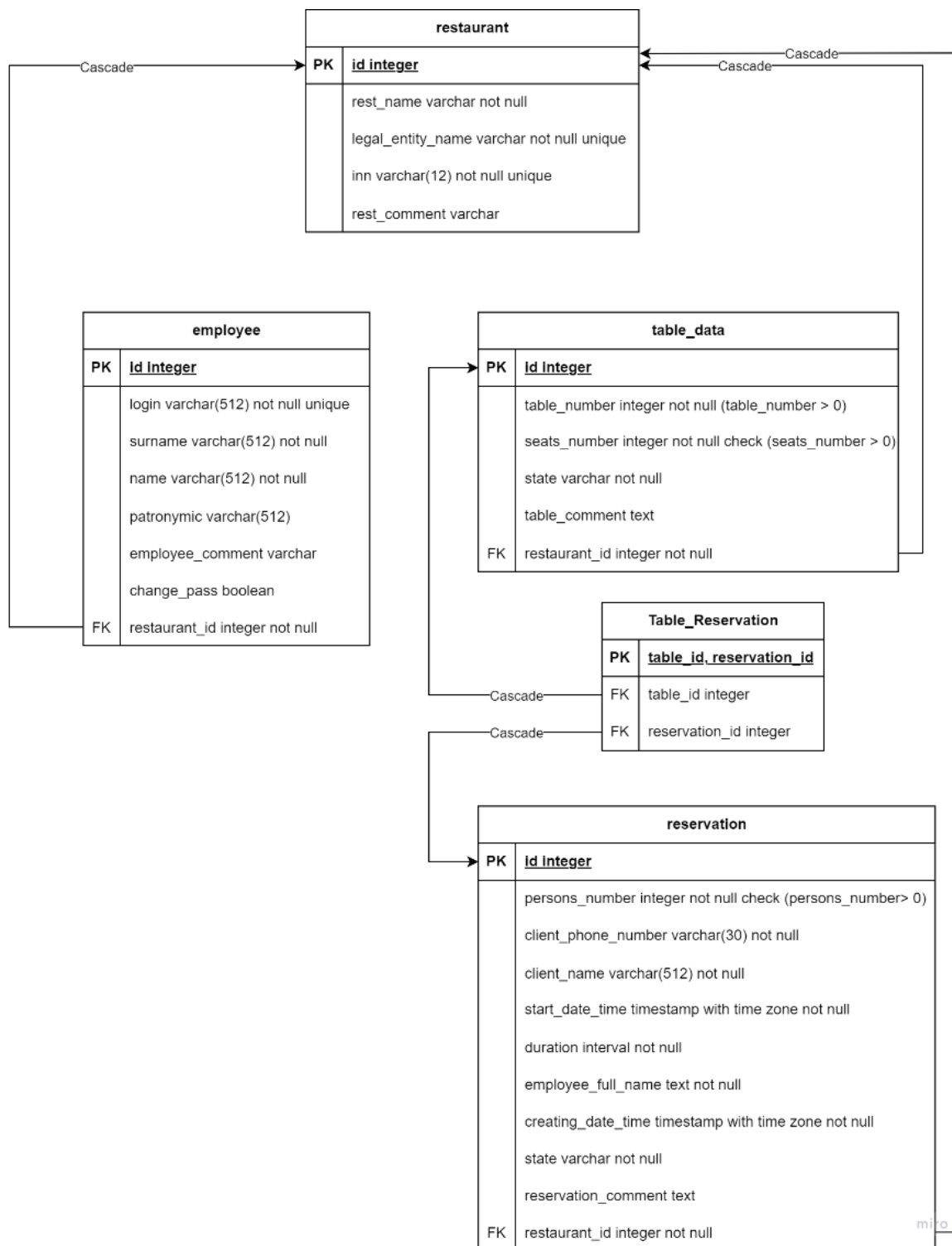


Рисунок 10 — Физическая схема базы данных приложения Restobook

3.8 Развёртывание приложения

Серверное приложение поставляется в виде контейнеров Docker. Для запуска приложения используется Docker compose. Для обеспечения работы https протокола используется Nginx (см. рисунок 11).

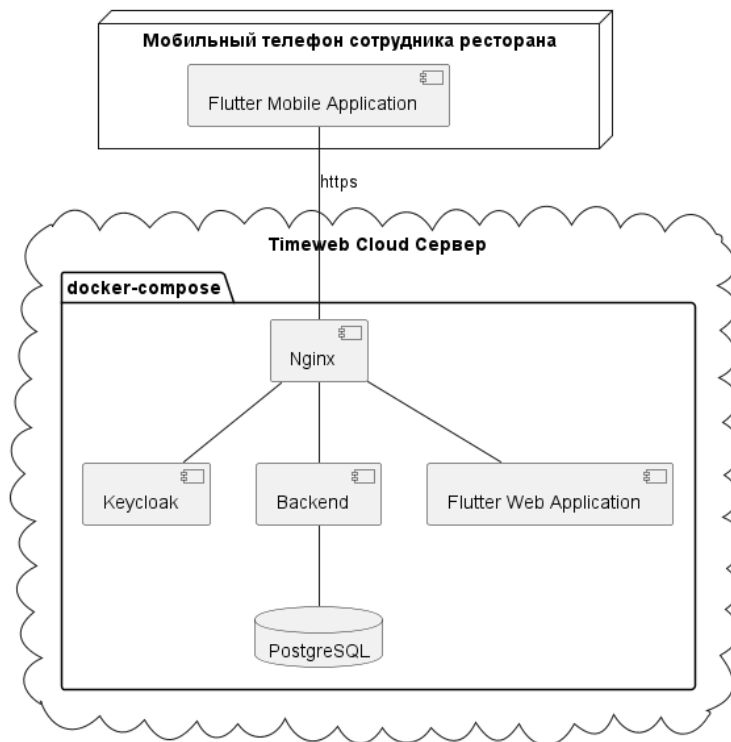


Рисунок 11 — Диаграмма развёртывания приложения

3.9 Реализация интерфейса

3.9.3 Реализация интерфейса мобильного приложения

Экраны мобильного приложения оформлены в едином стиле с использованием ограниченного набора шрифтов, одинаково и корректно отображаются на устройствах с операционной системой Android 11 и выше.

При первом заходе в приложение пользователю показывается onboarding, содержащий краткую информацию по использованию приложения (см. рисунок 12)

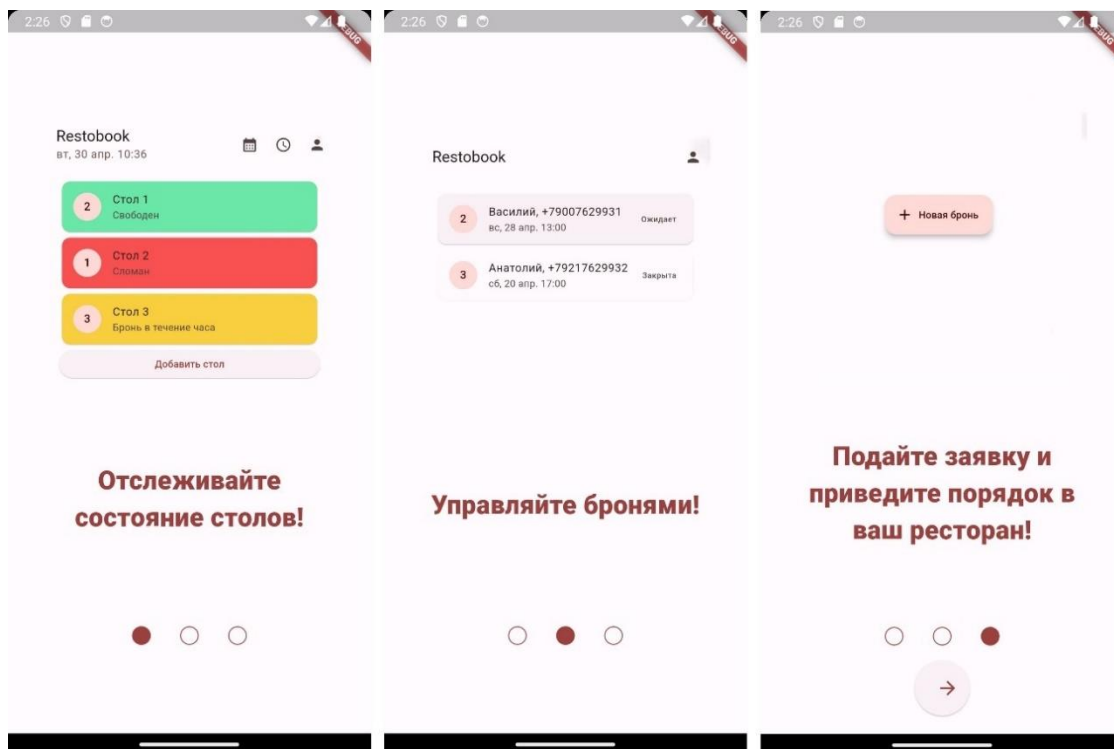


Рисунок 12 — Onboarding при первом входе в приложение

Если у пользователя нет учётной записи, инструкцию по получению логина и пароля он может узнать на специальных экранах, перейдя по кнопке с экрана входа (см. рисунок 13).

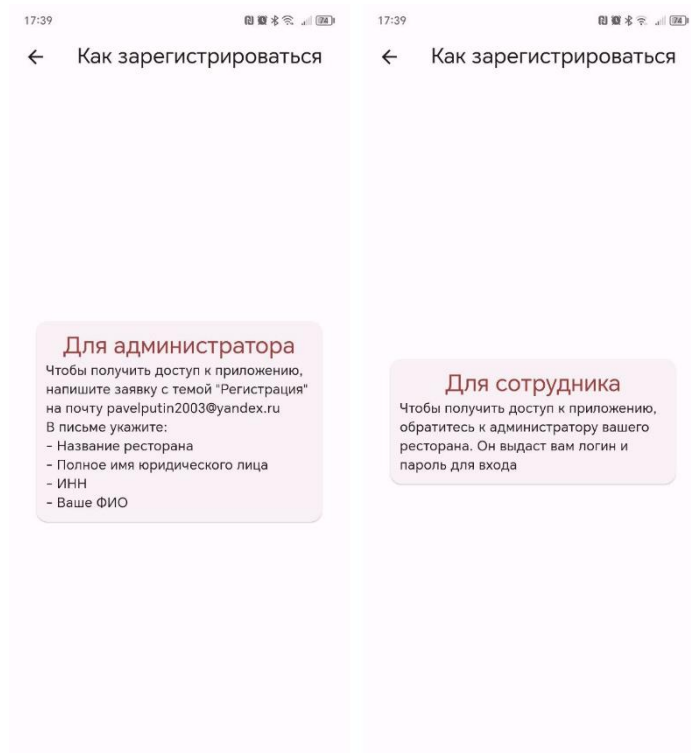


Рисунок 13 — Экран с информацией о регистрации для администратора (слева) и обычного сотрудника (справа)

Если пользователь не сменил пароль после того, как ему предоставили данные учётной записи, в приложении будет показан рорир с приглашением сменить пароль (см. рисунок 14)

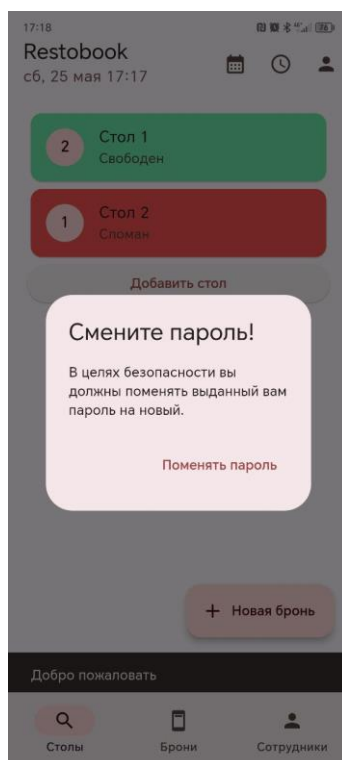


Рисунок 14 — Рорир с предложением сменить пароль

Первым экраном, который видит пользователь, является экран обзора столов (см. рисунок 15). На нём отображается список столиков, помеченных цветной индикацией:

- зелёный цвет – столик свободен;
- красный цвет – столик занят или сломан;
- жёлтый цвет – ближайшая бронь столика будет через час.

Главный экран, в зависимости от того, является вошедший пользователь администратором ресторана или нет, имеет кнопку перехода к обзору сотрудников и кнопку добавления стола.

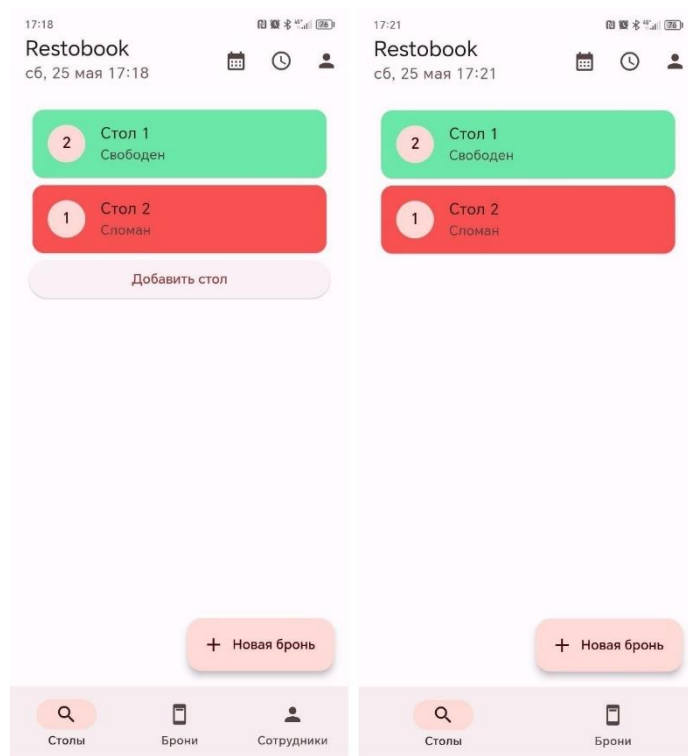


Рисунок 15 — Экран обзора столов для администратора ресторана (слева) и обычного сотрудника (справа)

Также приложения позволяет выбрать время, для просмотра занятости столов с помощью (см. рисунок 16).

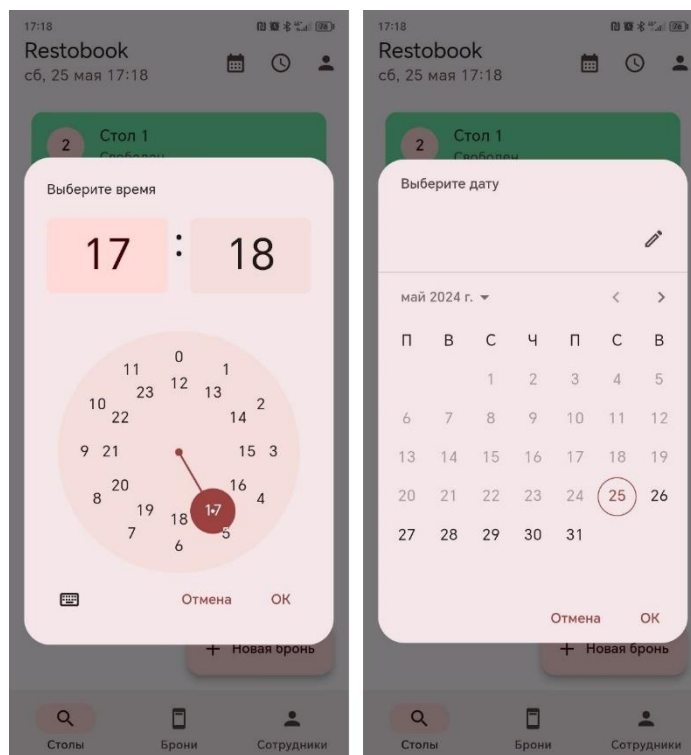


Рисунок 16 — Элементы управления выбора времени и даты занятости столов

По нажатию на кнопку «Новая бронь» открывается форма для создания брони (см. рисунок 17).



17:18

← Новая бронь

Количество гостей

Номер клиента

Имя клиента

Время

Дата

Длительность (в минутах)

Столики

Комментарий

Применить

Рисунок 17 — Форма создания брони

Приложение позволяет просматривать список броней столиков. При этом закрытые брони отображаются внизу списка и имеют более бледный оттенок фона (см. рисунок 18).



Рисунок 18 — Обзор броней

При нажатии на определённую бронь открывается экран с подробной информацией о выбранной брони и возможностью её редактирования (см. рисунок 19).

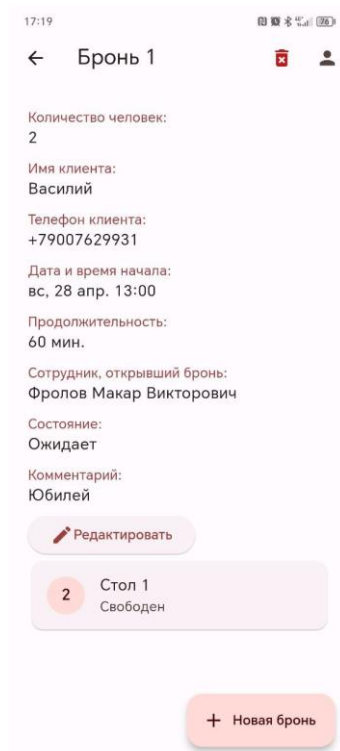


Рисунок 19 — Экран с информацией о брони

При нажатии на определённый стол открывается экран с подробной информацией о выбранном столе, возможностью его редактировать и посмотреть список связанных с ним броней (см. рисунок 20).

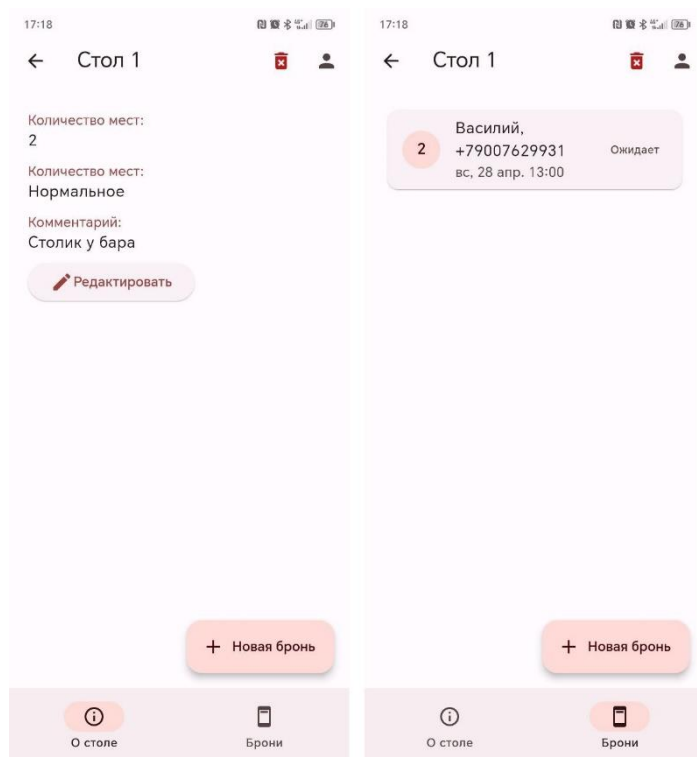


Рисунок 20 — Экраны с информацией о столики (слева) и списком связанных со столиком броней (справа)

Для администратора ресторана также доступна вкладка с обзором сотрудников (см. рисунок 21), с возможностью открыть форму редактирования сотрудника или экран с подробной информацией о выбранном сотруднике. При выборе в списке самого себя, открывается профиль.

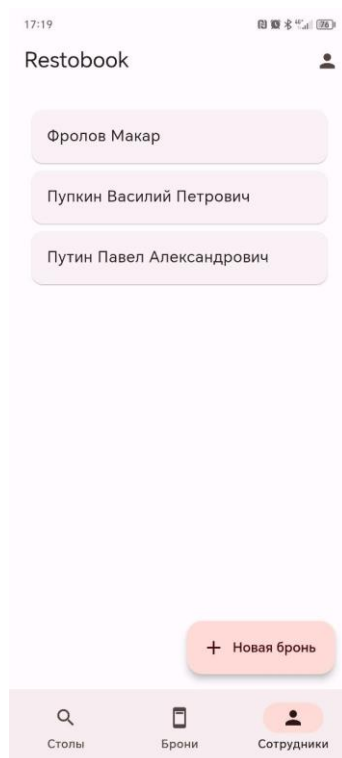


Рисунок 21 — Обзор сотрудников ресторана

В профиле сотрудника отображается подробная информация, а также приложение позволяет изменить пароль или выйти из аккаунта (см. рисунок 22).

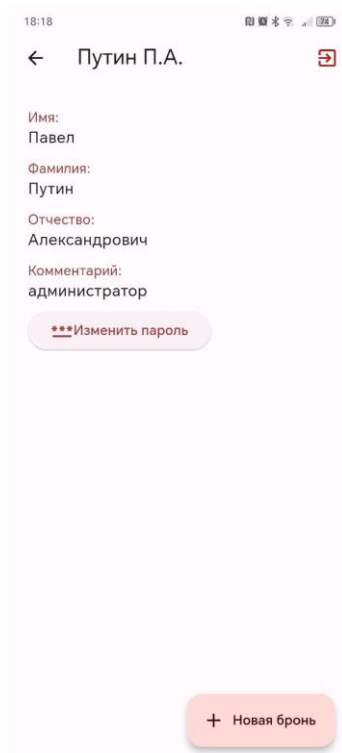


Рисунок 22 — Профиль сотрудника

4 Аналитика

Для тестирования работоспособности приложения доступ к нему был предоставлен студентам факультета компьютерных наук. Приложение было установлено 14 людьми. В результате эксплуатации, были получены данные для воронки по основному сценарию – «Создание и редактирование брони». Из них видно, что до этапа редактирования брони дошло 15% человек (см. рисунок 23). Из этого можно сделать вывод, что некоторые элементы формы редактирования брони, например, изменение её состояния, стоит сделать более удобными для пользователя.

Конверсия шагов

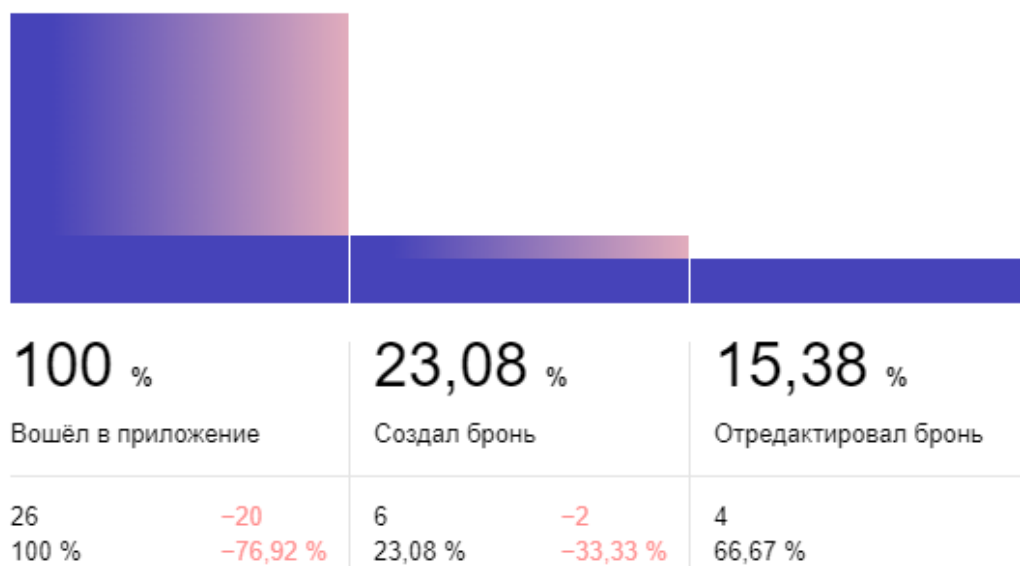


Рисунок 23 — Воронка создания и редактирования брони

Заключение

В ходе реализации данного проекта были выполнены поставленные задачи. Разработанное приложение удовлетворяет предъявленным требованиям, а именно:

- обеспечивает возможность создавать и редактировать брони, вести их учет;
- обеспечивает возможность создавать закрытые сотрудников;
- обеспечивает сохранение данных о брони в системе для доступа сотрудникам ресторана.

В результате проведённого тестирования было установлено, что готовое приложение успешно прошло все выполненные тесты на различные сценарии использования, тем самым подтвердив свою работоспособность.

Список использованных источников

1. Pub.dev [Электронный ресурс]. – Режим доступа: URL: <https://pub.dev/> - Заглавие с экрана (Дата обращения 20.05.2024)
2. Документация SpringBoot [Электронный ресурс]. – Режим доступа: URL: <https://docs.spring.io/spring-boot/docs/current/reference/html/> – Заглавие с экрана. – (Дата обращения 23.05.2024).
3. Документация AppMetrica [Электронный ресурс]. – Режим доступа: URL: <https://appmetrica.yandex.ru/docs/ru/> – Заглавие с экрана. – (Дата обращения 25.05.2024).

ПРИЛОЖЕНИЕ А BPMN диаграмма

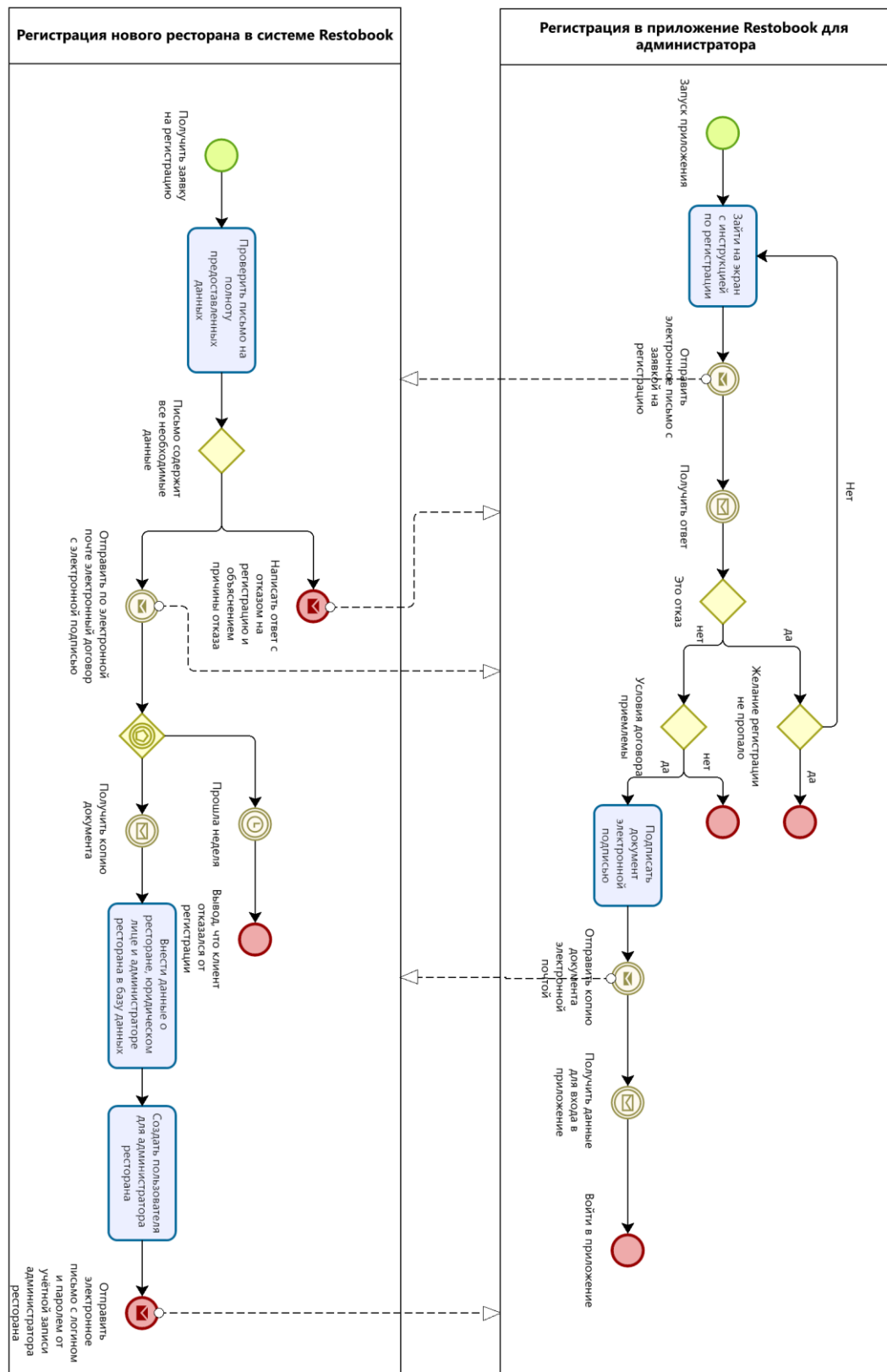


Рисунок А.1 — BPMN диаграмма регистрации администратора ресторана

ПРИЛОЖЕНИЕ Б

Диаграмма прецедентов

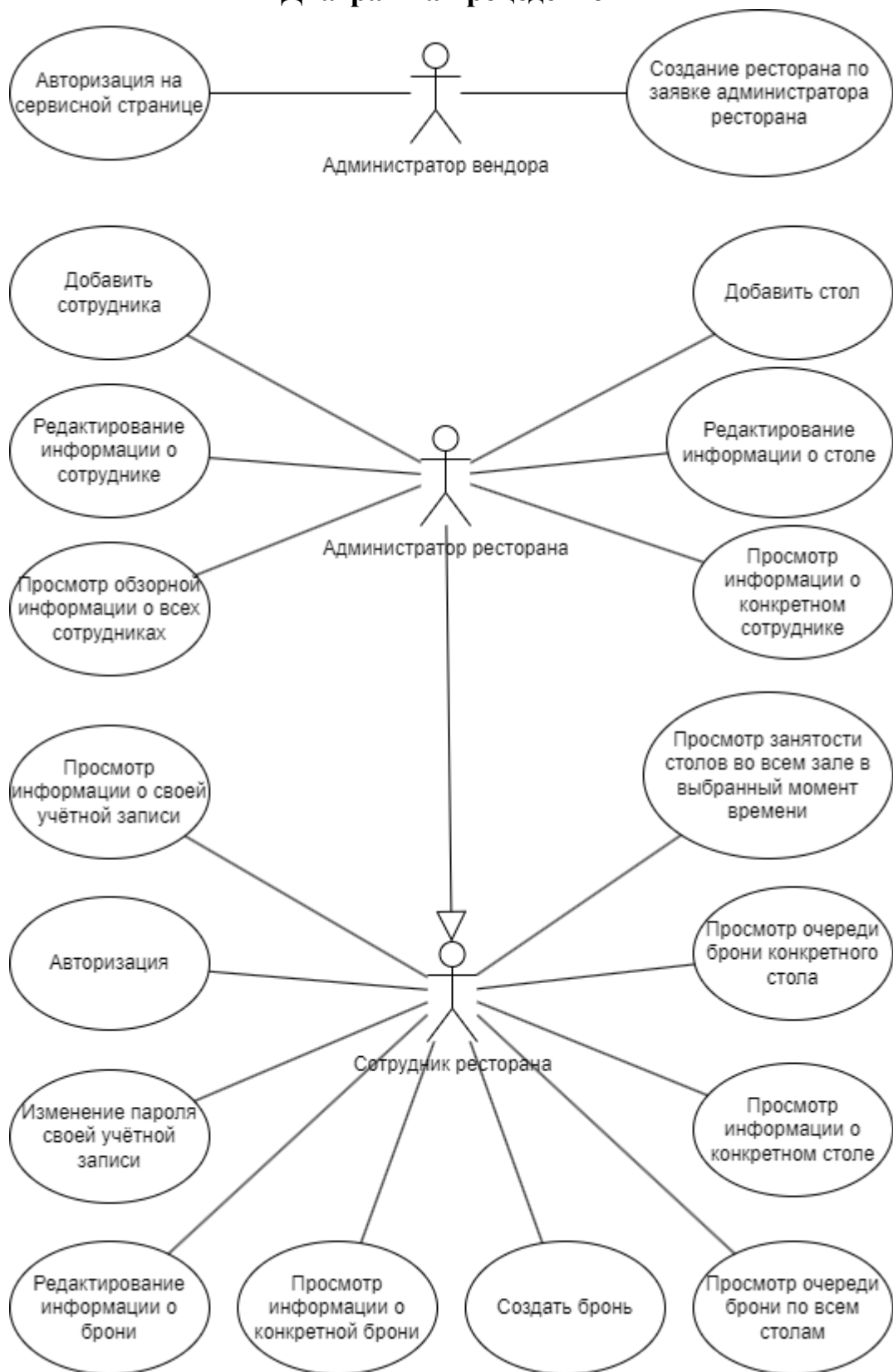


Рисунок Б.1 — Диаграмма прецедентов

ПРИЛОЖЕНИЕ В

Диаграммы состояний

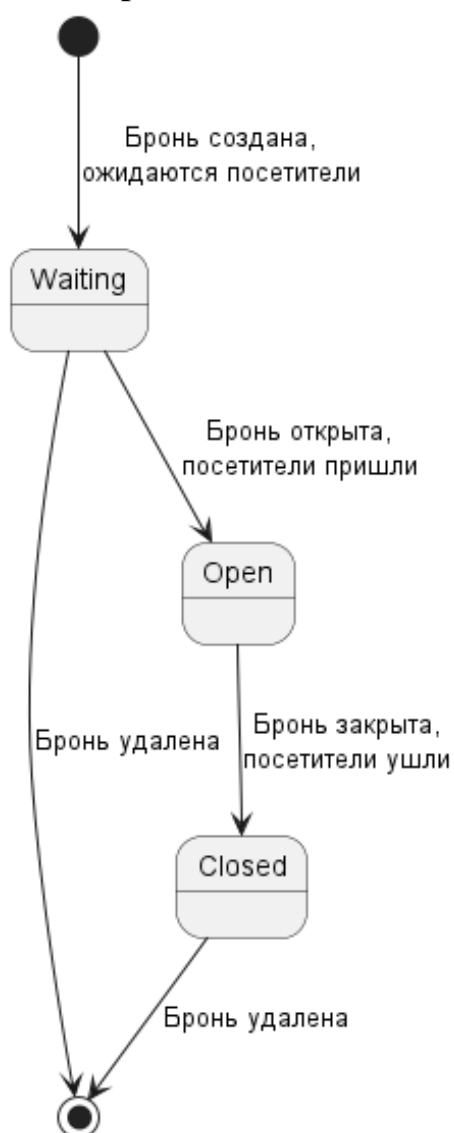


Рисунок В.1 — Диаграмма состояний для брони

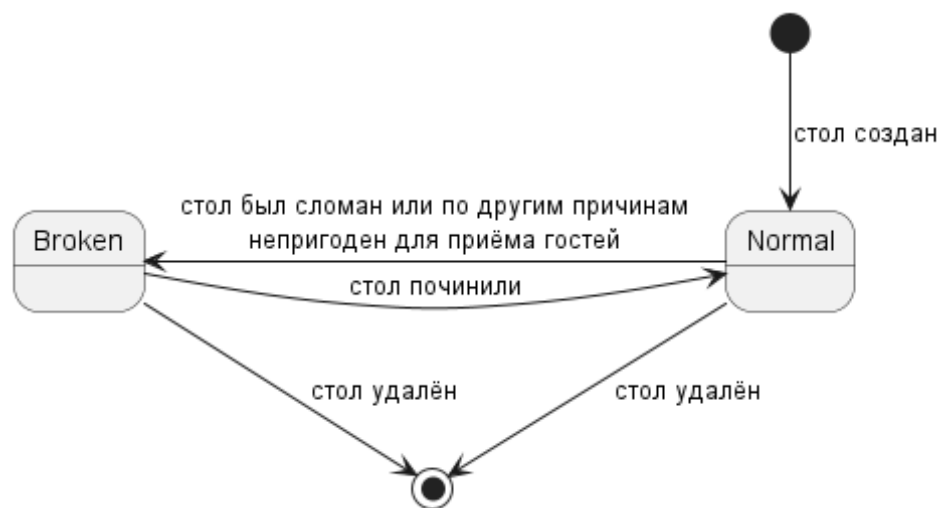


Рисунок В.2 — Диаграмма состояний для столов