

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

Факультет компьютерных наук

Кафедра программирования и информационных технологий

**Курсовой проект**

Мобильное приложение для просмотра рецептов различных блюд и диет

Направление 09.03.02 «Информационные системы и технологии»

Программная инженерия в информационных системах

Зав. кафедрой \_\_\_\_\_ С.Д. Махортов, д.ф.- м.н., доцент \_\_.\_\_.20\_\_

Обучающийся \_\_\_\_\_ А. Н. Бредихин, 3 курс, д/о

Обучающийся \_\_\_\_\_ Н. Е. Кравцова, 3 курс, д/о

Обучающийся \_\_\_\_\_ В. Б. Новичихин, 3 курс, д/о

Руководитель \_\_\_\_\_ В. А. Ушаков, ассистент

Руководитель \_\_\_\_\_ В.С. Тарасов, ст. преподаватель

Воронеж 2024

## СОДЕРЖАНИЕ

Введение.....	3
1 Постановка задачи.....	4
1.1 Цели создания системы .....	4
1.2 Задачи проекта.....	4
2 Анализ предметной области .....	5
2.1 Терминология .....	5
2.2 Обзор аналогов .....	5
2.2.1 Kitchen Stories.....	5
2.2.2 Простые рецепты .....	6
2.2.3 Food.ru .....	8
2.3 Средства реализации.....	9
3 Реализация.....	11
3.1 Структура базы данных.....	11
3.2 Диаграмма состояния .....	12
3.3 Сценарии использования.....	14
4 Интерфейс приложения .....	16
4.1 Авторизация.....	16
4.2 Незарегистрированный пользователь .....	17
4.3 Зарегистрированный пользователь .....	23
4.4 Администратор.....	27
Заключение .....	30
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	31

## **Введение**

В наше время, когда технологии продолжают развиваться стремительными темпами, все больше людей предпочитают использовать мобильные приложения для решения различных задач, связанных с повседневной жизнью. Одной из важных сфер, где мобильные приложения могут значительно облегчить жизнь людей, является здоровое питание и диетология.

Несмотря на то, что сегодня существует множество различных источников информации, для многих людей актуальной остаётся проблема недостатка удобного ресурса для поиска и доступа к разнообразным рецептам еды, учитывая различные диетические пожелания.

Затрагивая тему современных реалий, почти у каждого человека имеются свои мобильные устройства, что значительно упрощает многие повседневные вещи. При наличии мобильного приложения с возможностью поиска рецептов по их названиям, диетам и добавлением их в избранное люди могут значительно облегчить себе процесс планирования питания и приготовления еды. К тому же, если у подобного приложения будет функционал, реализующий не только поиск, но и публикацию рецептов от пользователей, то это откроет широкие возможности для обмена кулинарными идеями и опытом. Для достижения поставленных целей для начала рассмотрим несколько популярных аналогов и затем перейдем к формулировке требований, на основе которых будет создаваться приложение.

## **1 Постановка задачи**

### **1.1 Цели создания системы**

Целью данной курсовой работы является создание мобильного приложения для просмотра рецептов различных блюд и диет с возможностью фильтрации по различным параметрам.

### **1.2 Задачи проекта**

При разработке приложения должны быть реализованы следующие задачи:

- возможность просмотра рецептов;
- поиск рецептов по названию, категории и диете;
- возможность пользователей добавлять в избранное понравившиеся рецепты;
- предоставление пользователям права создания собственных рецептов;
- публикация рецептов пользователей после одобрения администратора.

## **2 Анализ предметной области**

### **2.1 Терминология**

Клиент (клиентская сторона) – приложение, которое предоставляет пользователю возможность взаимодействовать со всей системой.

Мобильное приложение – программное изделие, разновидность прикладного программного обеспечения, предназначенная для работы на смартфонах, планшетах и других мобильных (портативных, переносных, карманных) устройствах.

Сервер (серверная часть) – компьютер, обслуживающий другие устройства (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач.

СУБД – система управления базами данных.

Фреймворк – программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

API – набор способов и правил, по которым различные программы общаются между собой и обмениваются данными. Spring – универсальный фреймворк с открытым исходным кодом для Java-платформы.

### **2.2 Обзор аналогов**

#### **2.2.1 Kitchen Stories**

Простое и удобное приложение с рецептами. Интерфейс приложения представлен на рисунке 1.

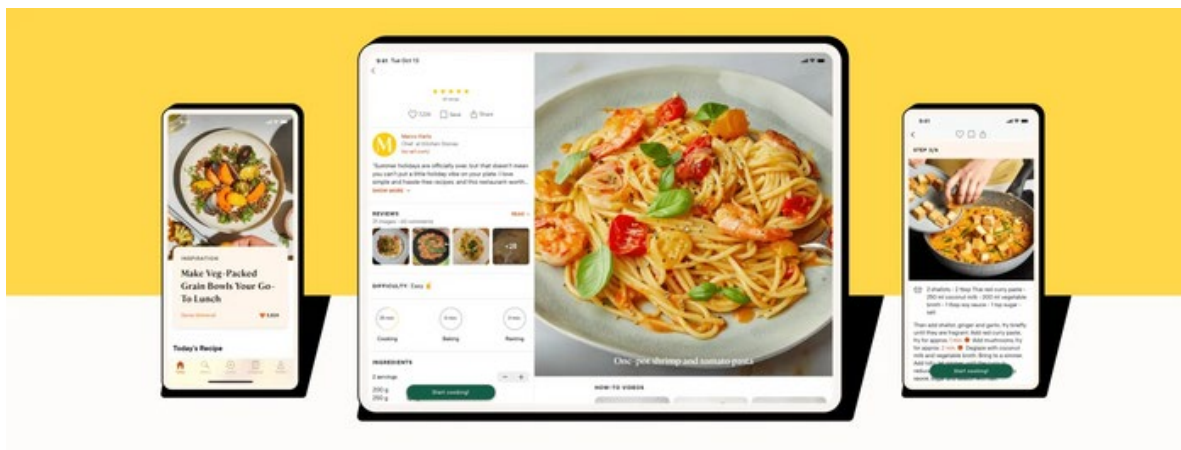


Рисунок 1 — Интерфейс приложения «Kitchen Stories»

Функциональные возможности приложения:

- есть возможность поиска рецептов по категориям: «Самые лучшие завтраки и закуски», «Сытные ужины», «Блюда для веганов» и так далее;
- есть возможность формировать список покупок из ингредиентов выбираемых блюд.

Плюсы:

- большой выбор категорий;
- можно сохранять рецепты в избранное;
- есть список покупок.

Минусы:

- нельзя добавлять собственные блюда.

### 2.2.2 Простые рецепты

Простое приложение с простыми рецептами. Интерфейс приложения представлен на рисунке 2.

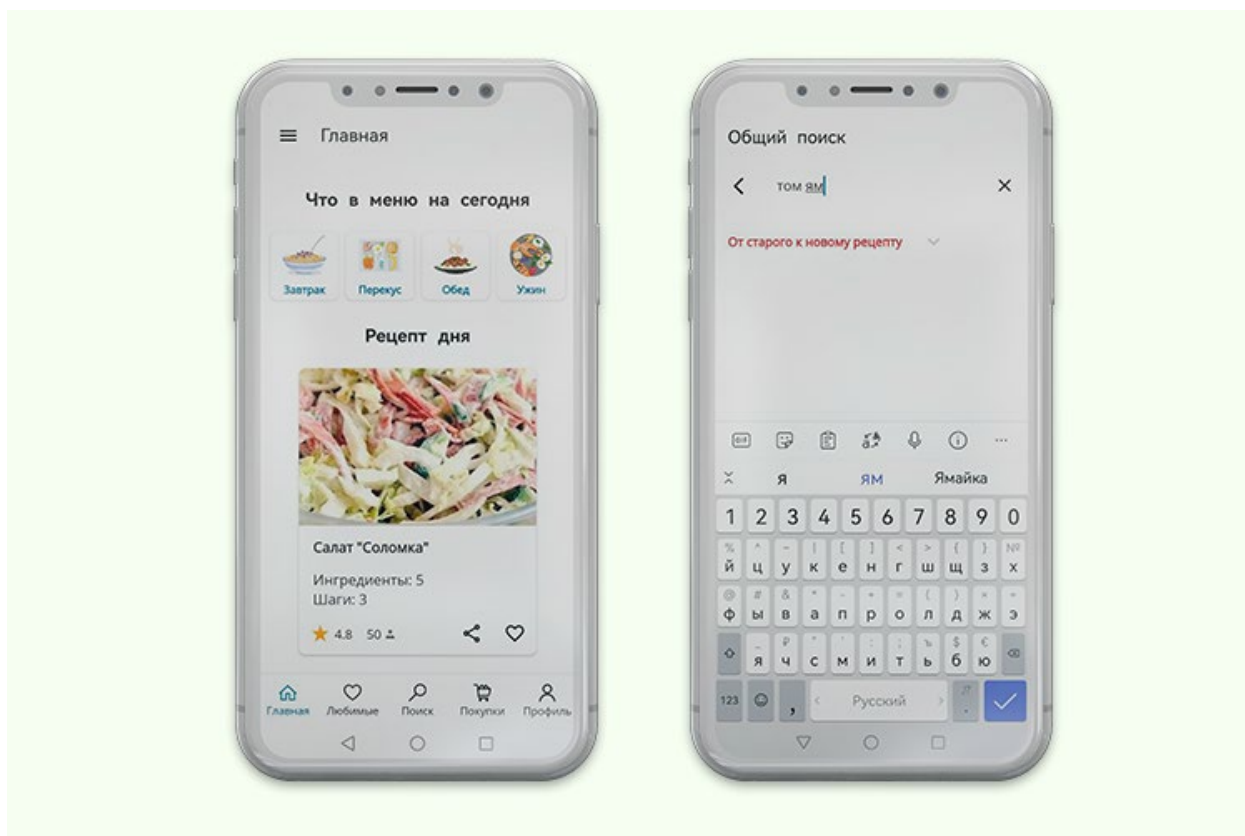


Рисунок 2 — Интерфейс приложения «Простые рецепты»

Функциональные возможности приложения:

- есть возможность добавить рецепт в избранное;
- есть возможность отмечать рецепты, которые уже были приготовлены;
- есть возможность добавить рецепт.

Плюсы:

- большой выбор категорий;
- можно сохранять рецепты в любимые и отмечать те, которые уже были приготовлены;

Минусы:

- база с рецептами небольшая.

### 2.2.3 Food.ru

Это удобное и быстрое приложение – своего рода мобильная версия сайта Food.ru. Интерфейс приложения представлен на рисунке 3.

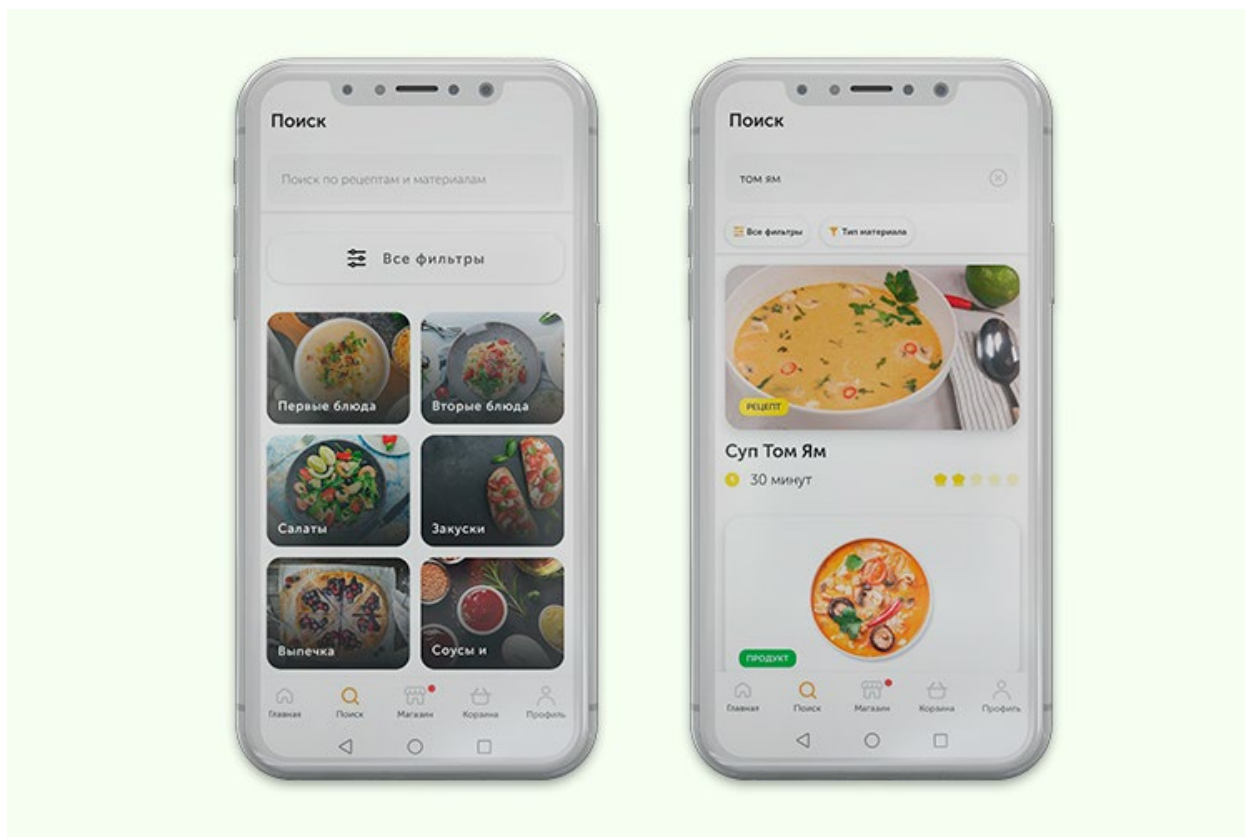


Рисунок 3 — Интерфейс приложения «Простые рецепты»

Функциональные возможности приложения:

- есть возможность создавать свои рецепты;
- есть возможность делиться рецептами;
- существует фильтрация по стоимости, типу, калорийности и времени приготовления.

Плюсы:

- есть лента новостей;
- удобный поиск;
- можно добавлять ингредиенты в список продуктов;
- можно делиться и лайкать рецепты.



Минусы:

- нужна регистрация для большинства функций.

## 2.3 Средства реализации

Для реализации проекта были использованы следующие средства:

- Java – строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems;

- Spring Boot Framework – это веб-платформа Java на основе микрослужб с открытым кодом от Spring. Обладает большим набором инструментов [1];

- PostgreSQL – свободная объектно-реляционная система управления базами данных. Выбрана так как, поддерживает большое количество типов данных, имеет богатую функциональность, поддерживается многими серверами [2];

- Docker – платформа, предназначенная для помощи разработчикам в создании, совместном использовании и запуске современных приложений [3];

- Flyway – средство обеспечения версионности схем и управления миграциями базы данных. FlyWay имеет поддержку PostgreSQL и интегрируется с Spring Framework;

- Dart – это язык программирования, разработанный компанией Google, который используется для создания веб-приложений, мобильных приложений и других программных решений.

- Flutter – комплект средств разработки и фреймворк с открытым исходным кодом для создания мобильных приложений под Android и iOS, веб-приложений, а также настольных приложений под Windows, macOS и Linux с использованием языка программирования Dart, разработанный и развиваемый корпорацией Google.

- Miro – платформа для совместной работы распределенных команд;
- Swagger – фреймворк для спецификации REST API;
- Draw.io – бесплатное кроссплатформенное программное обеспечение для рисования графиков с открытым исходным кодом, разработанное на HTML5 и JavaScript. Его интерфейс можно использовать для создания диаграмм, таких как блок-схемы, каркасы, диаграммы UML, организационные диаграммы и сетевые диаграммы;
- Figma – онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени;
- Youtrack – облачная программа для управления проектами небольших групп;
- Github – веб-сервис для хостинга IT-проектов и их совместной разработки;
- Git – распределённая система управления версиями [4].

### 3 Реализация

В результате анализа было принято решение о разработке системы, описание которой приводится в данной главе.

#### 3.1 Структура базы данных

Структура базы данных приведена на рисунке 4.

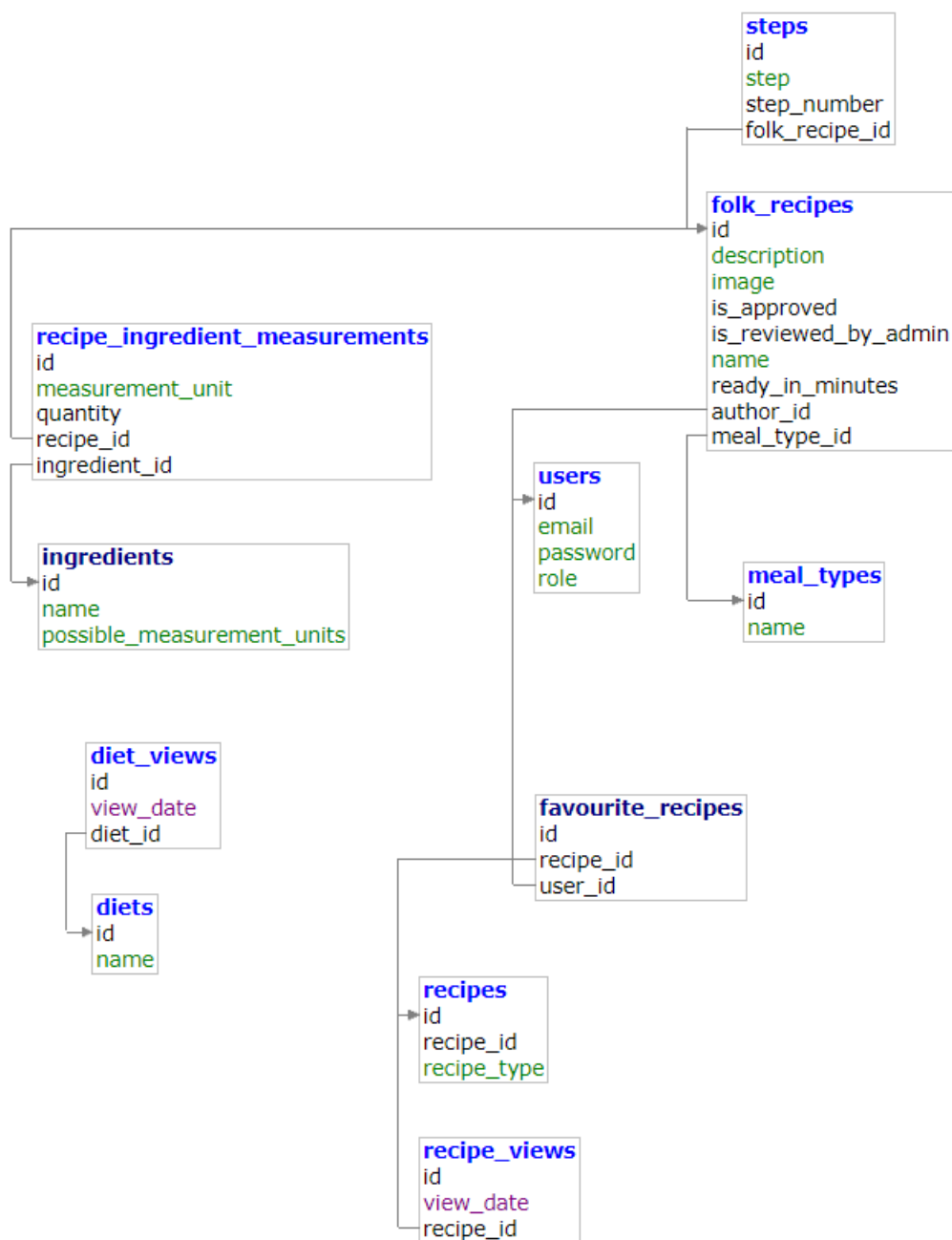


Рисунок 4 — Схема таблиц базы данных

Для хранения информации о рецептах используются следующие таблицы:

- Recipes – таблица для хранения id рецепта и его тип (пользовательский или с API);
- Recipe\_views – таблица для хранения времени просмотров рецептов;
- Users – таблица с пользователями приложения, в которой записаны email пароль и роль каждого пользователя;
- Favourite\_recipes – таблица с рецептами, которые пользователь добавил в избранное;
- Folk\_recipes – таблица с рецептами пользователей. Имеет следующие поля: название рецепта, его описание, фотография, количество минут на его приготовление, флаги с указанием была, ли произведена проверка этого рецепта и какое решение принял проверяющий, а также ссылки на автора рецепта и на категорию, к которой он относится;
- Recipe\_ingredient\_measurements – таблица хранит в себе информацию о количестве и граммовке каждого ингредиента в рецепте;
- Ingredients – таблица со всевозможными ингредиентами и их граммовками;
- Steps – таблица с шагами приготовления рецепта. Имеет следующие поля: описание шага, номер шага и ссылку на рецепт пользователя;
- Meal\_types – таблица содержит в себе названия всех категорий блюд;
- Diets – таблица содержит в себе названия всех диет;
- Diets\_views - таблица для хранения времени просмотров диеты.

### **3.2 Диаграмма состояния**

Диаграмма состояния позволяет определить все возможные сценарии поведения системы, выделить её ключевые состояния и переходы между

ними. Для данного проекта была спроектирована диаграмма для состояний пользовательского рецепта, представленная на рисунке 5.

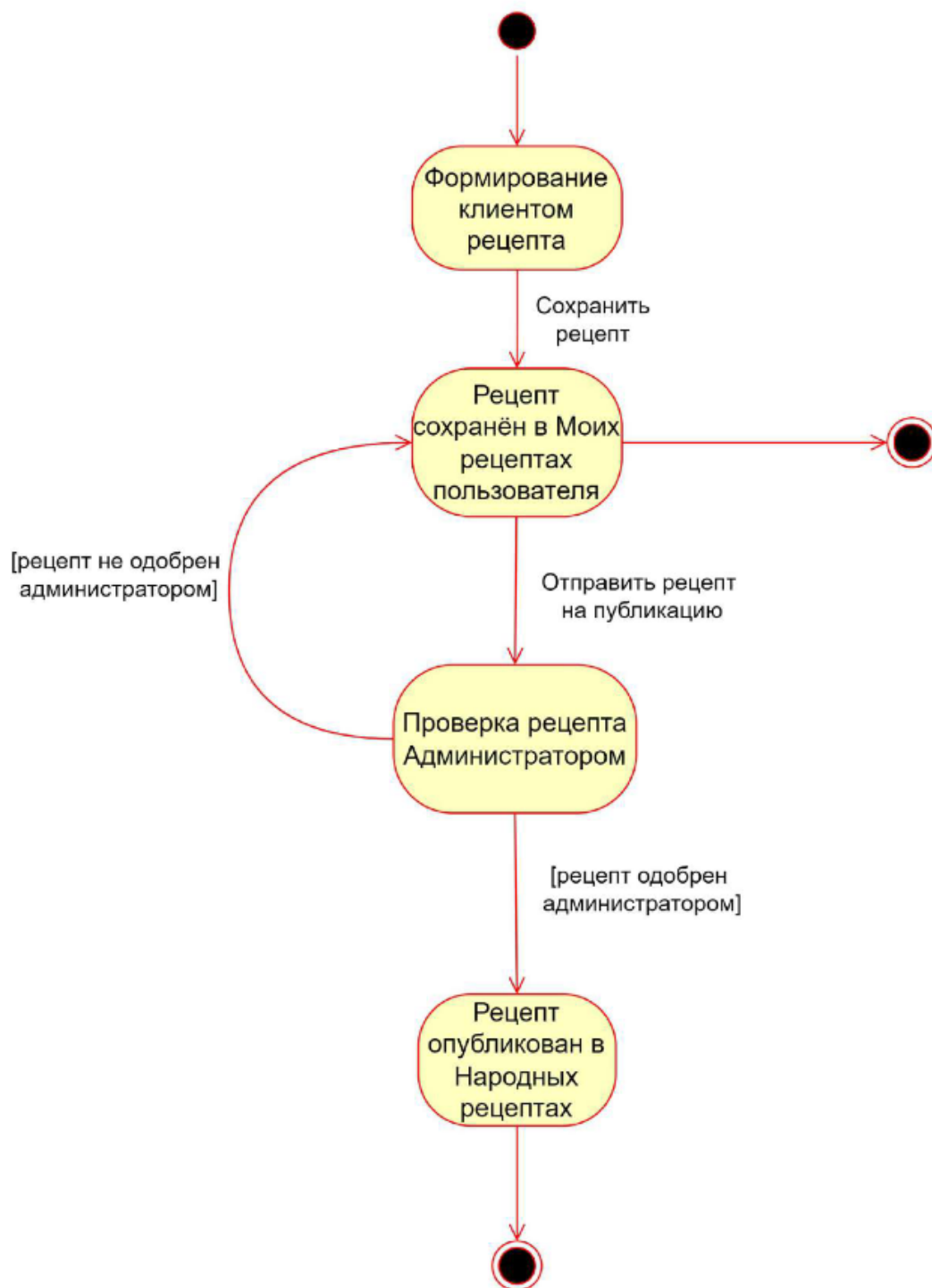


Рисунок 5 — Диаграмма состояний для рецепта пользователя

### 3.3 Сценарии использования

Сценарии использования для всех групп пользователей приведены на рисунке 6.

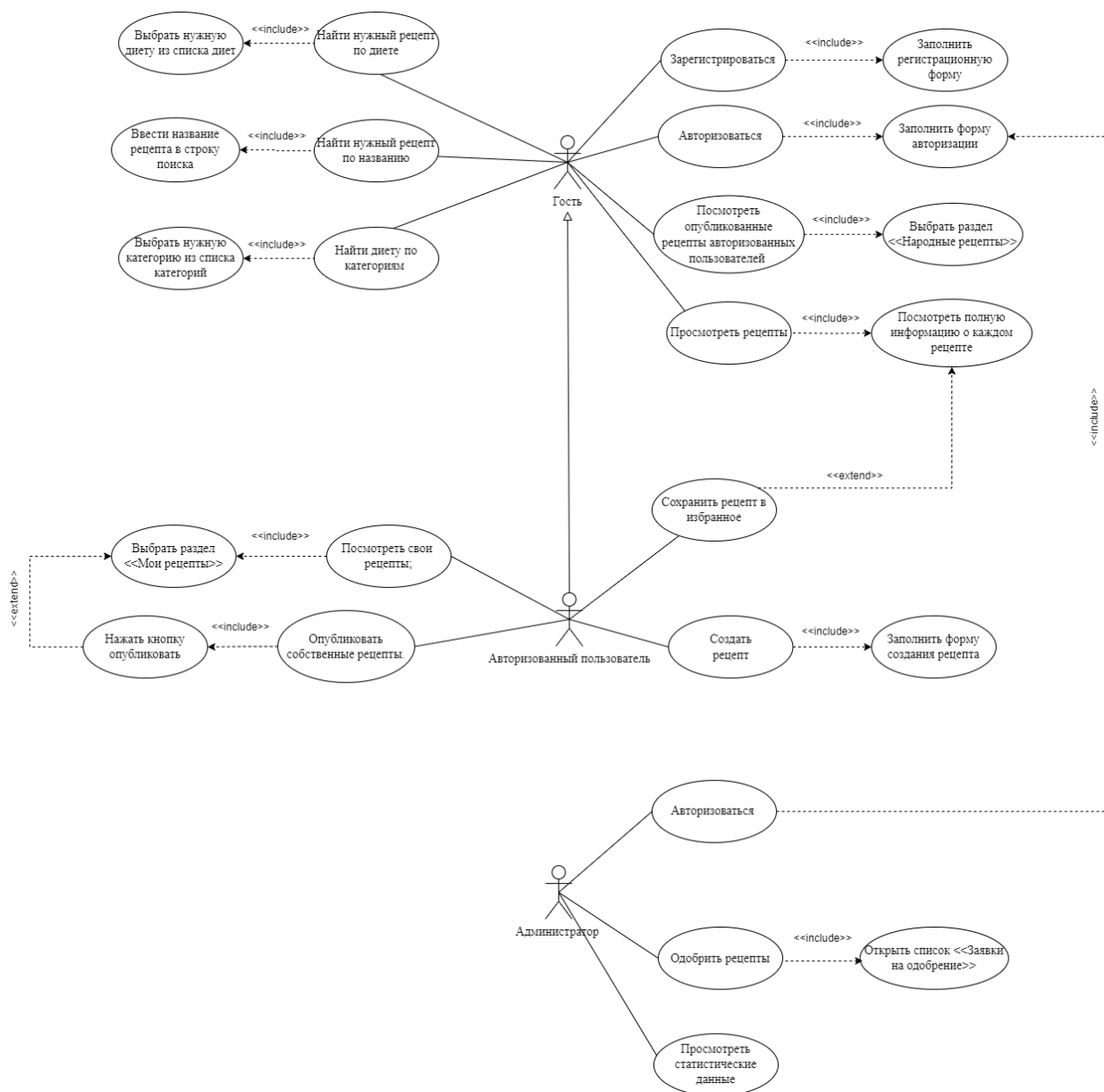


Рисунок 6 — Сценарии использования

Рассмотрим диаграмму для использования приложения разными группами пользователей. В данном случае необходимость составления диаграммы прецедентов обусловлена прежде всего тем, что Use-Case диаграмма — это прежде всего инструмент для моделирования системы, понимания ее функциональности и потребностей пользователей. Она может

помочь с определением основных действий, которые пользователь должен совершить в системе, чтобы достичь конкретных целей. Данная диаграмма также позволяет определить все возможные риски и проблемы, которые могут возникнуть в ходе использования системы.

## **4 Интерфейс приложения**

Для реализации клиентской части проекта была выбрана связка Dart и Flutter. Этот набор инструментов обеспечивает разработчиков широкими возможностями для создания мобильных приложений на Android. Flutter позволяет использовать SDK для доступа к аппаратным возможностям устройства, взаимодействия с различными сервисами и API, разработки пользовательского интерфейса и многого другого.

### **4.1 Авторизация**

Пользователь имеет возможность авторизоваться по почте (рисунок 7) и в соответствии с ролью, которую имеет его аккаунт, получит в своё распоряжение соответствующий функционал.

Соответственно функционал будет различным для администратора и простого пользователя, предоставляя им разные возможности.



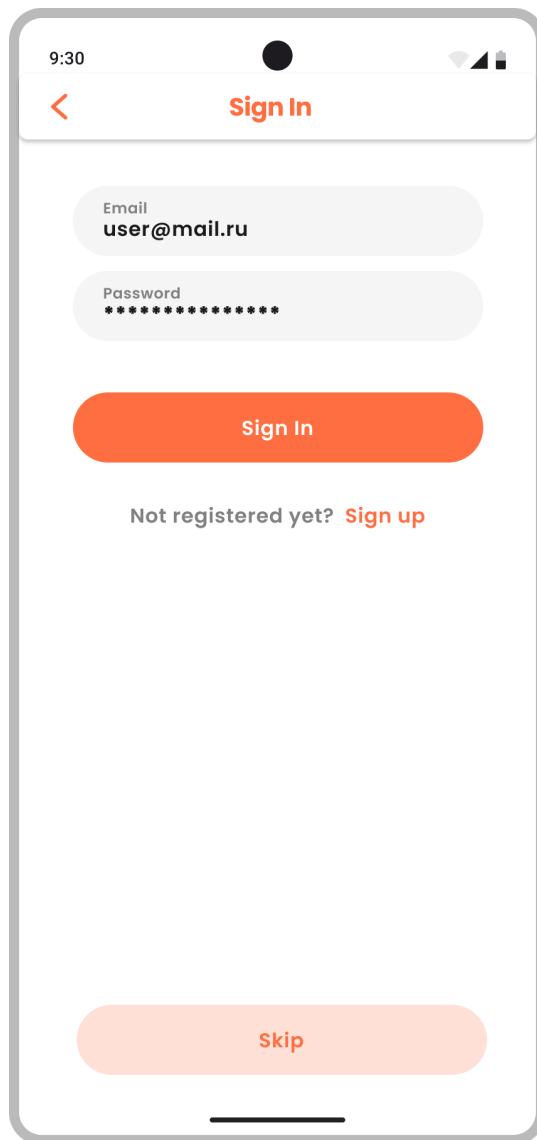


Рисунок 7 — Экран авторизации

## 4.2 Незарегистрированный пользователь

У незарегистрированного пользователя предусмотрен следующий набор возможностей:

- регистрация (рисунок 8);

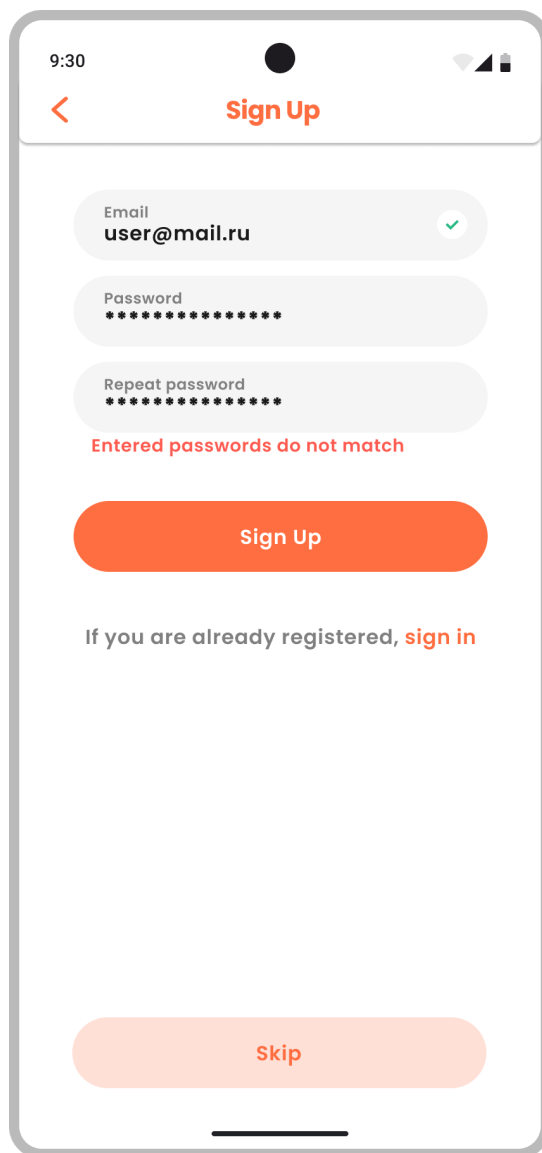


Рисунок 8 — Экран регистрации

— поиск рецептов по названию (рисунок 9);

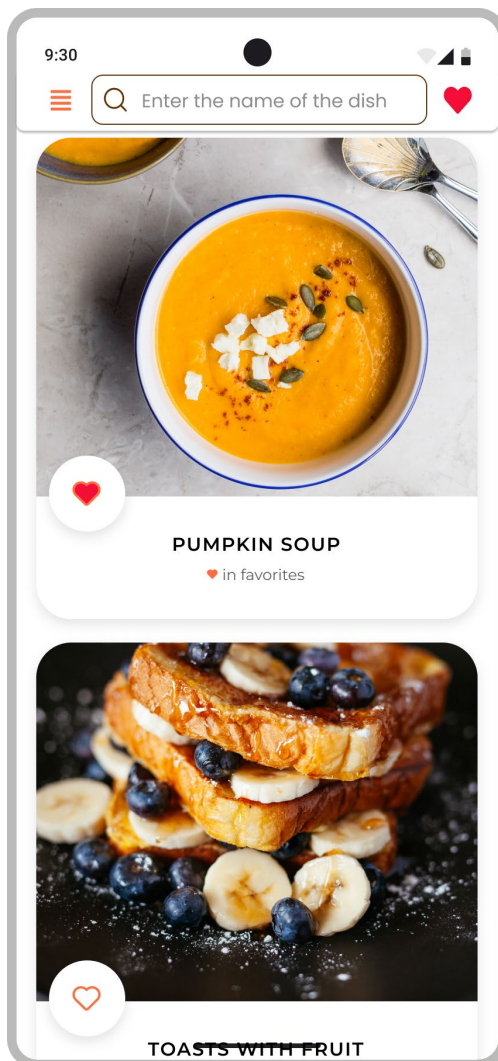


Рисунок 9 — Экран с полем для поиска рецептов по названию

— выбор категории блюд (рисунок 10);

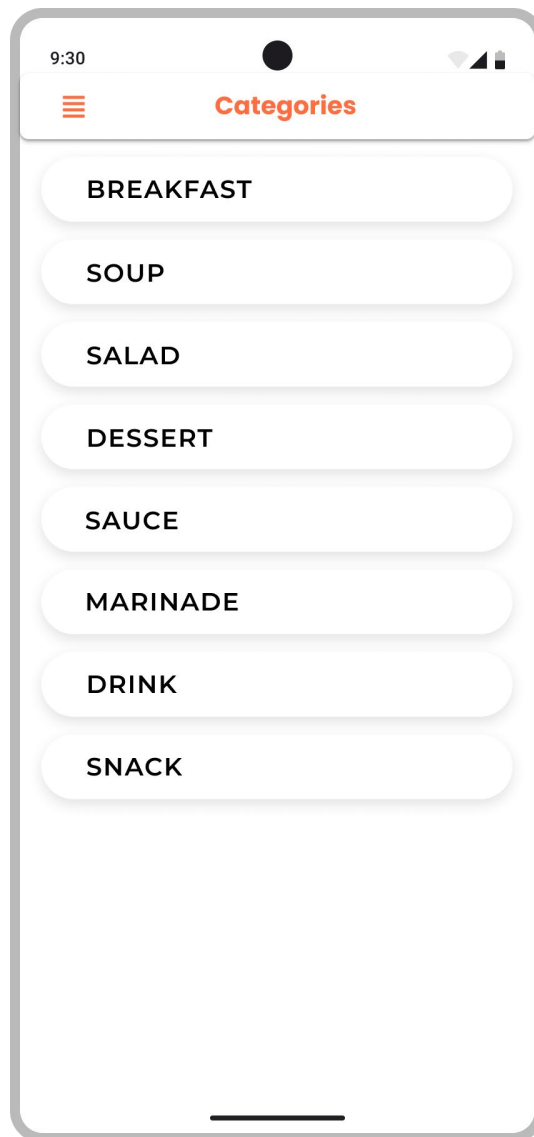


Рисунок 10 — Экран выбора категории блюд

— выбор диеты (рисунок 11);

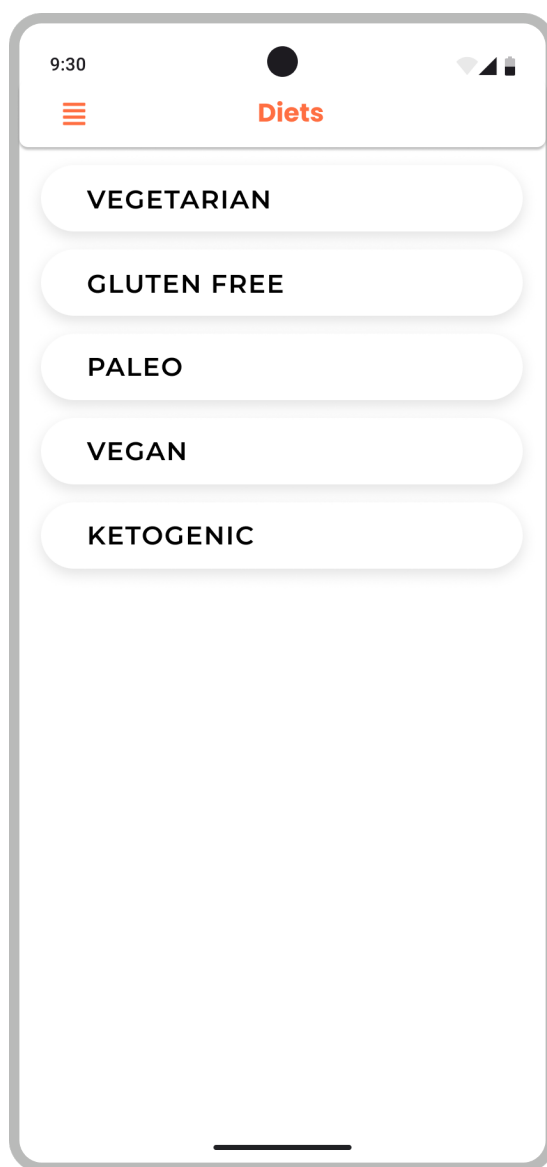


Рисунок 11 — Экран выбора диеты

— доступ к опубликованным рецептам авторизованных пользователей приложения – народным рецептам (рисунок 12);

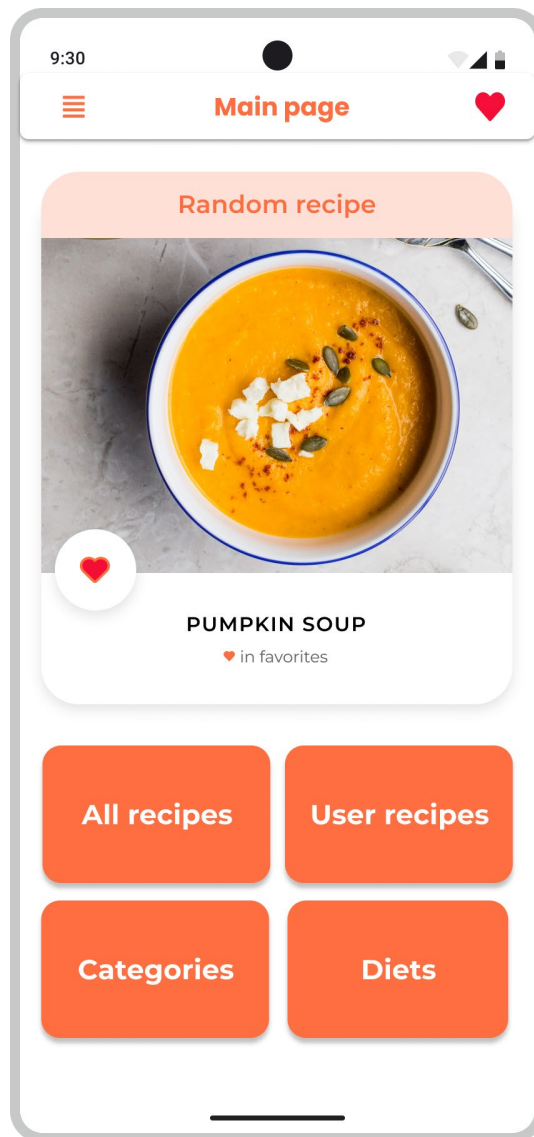


Рисунок 12 — Экран с кнопкой перехода к народным рецептам

— просмотр рецепта (рисунок 13).

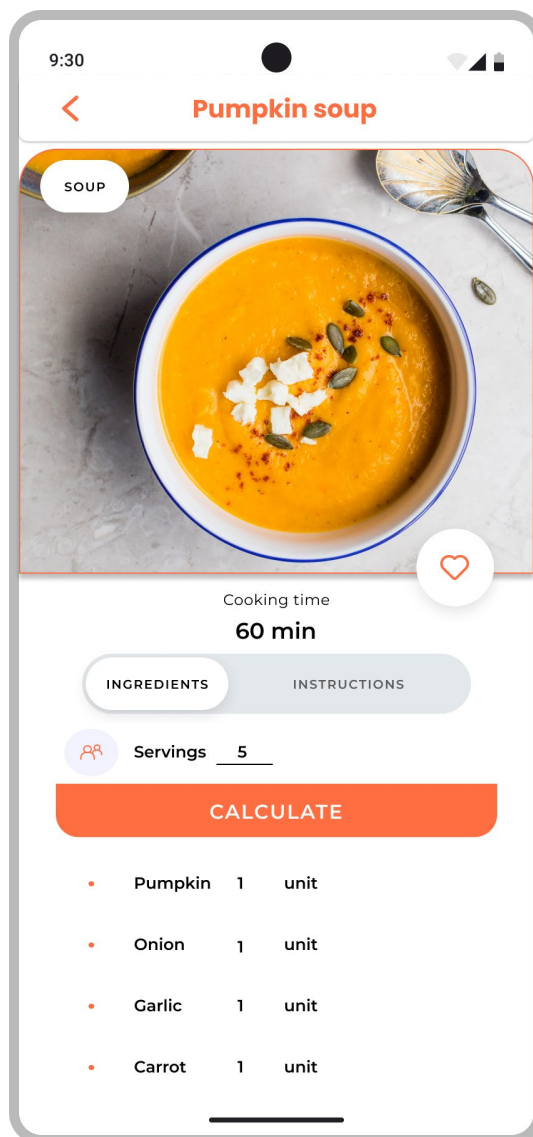


Рисунок 13 — Экран просмотра рецепта

### 4.3 Зарегистрированный пользователь

У зарегистрированного пользователя, помимо возможностей не зарегистрированного пользователя, предусмотрен следующий набор возможностей:

- вход в личный кабинет (рисунок 14);

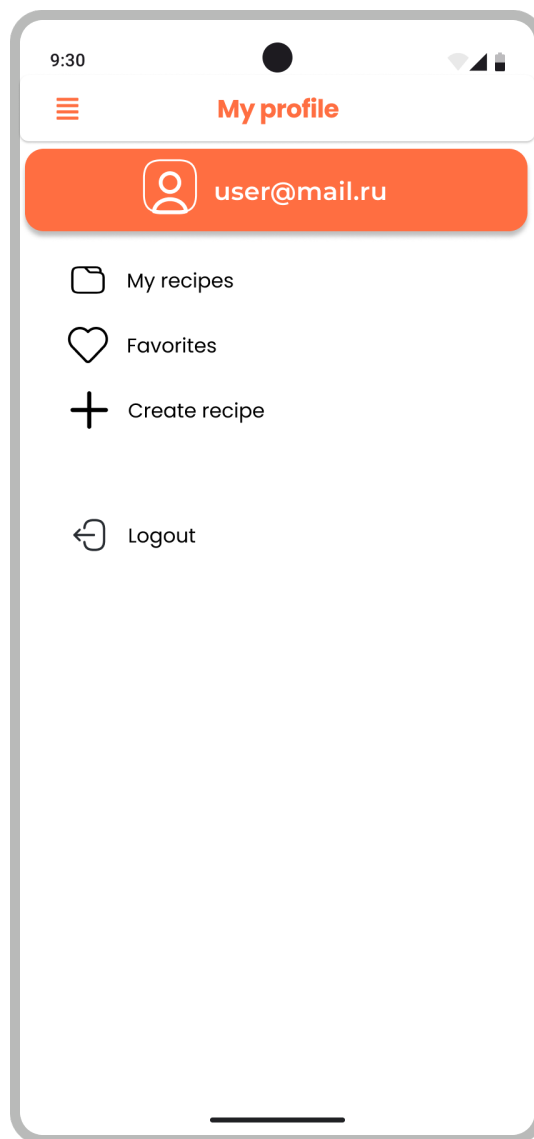


Рисунок 14 — Экран личного кабинета

— добавление в избранное понравившиеся рецепты (рисунок 15);



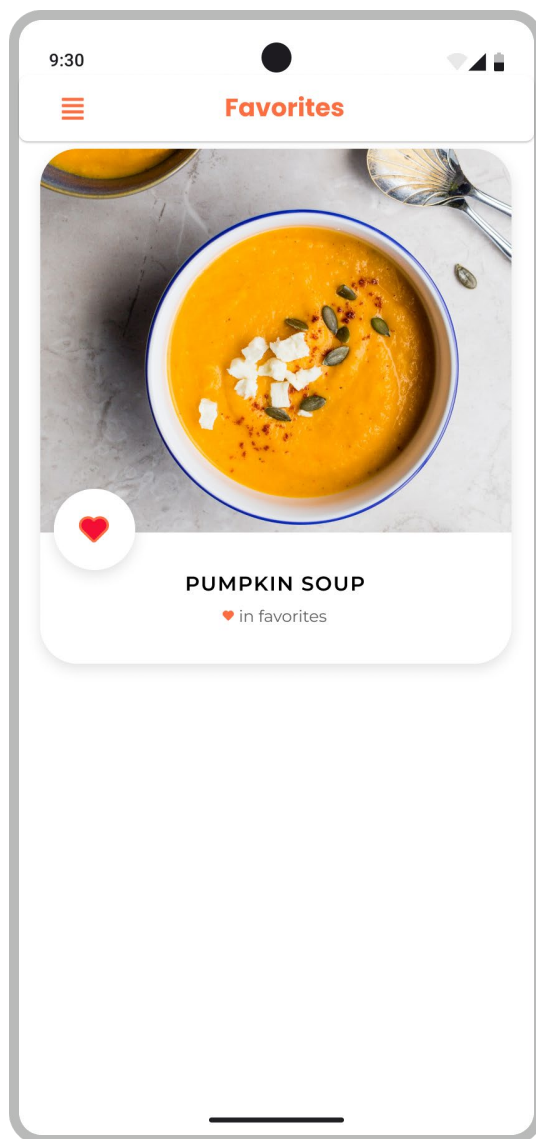


Рисунок 15 — Экран избранного

— создание рецепта и их публикация после одобрения администратора (рисунок 16);

9:30

< Create recipe

UPLOAD PHOTO

RECIPE NAME

Squash caviar

DISH CATEGORY

DINNER

DESCRIPTION

A dish of stewed zucchini.

COOKING TIME

40 min

NUMBER OF SERVINGS

6

INGREDIENTS

ONION	1	unit	—
ZUCCHINI	3	unit	—
GARLIC	1	unit	—

+ ADD INGREDIENT

PREPARATION INSTRUCTIONS

Prepare all the necessary ingredients. —

Cut the zucchini into large cubes. —

+ ADD STEP

☒ **Post the recipe**  
Your recipe will be available to all users after moderation

CANCEL SAVE

Рисунок 16 — Экран создания рецепта

— просмотр своих ранее созданных рецептов (рисунок 17).

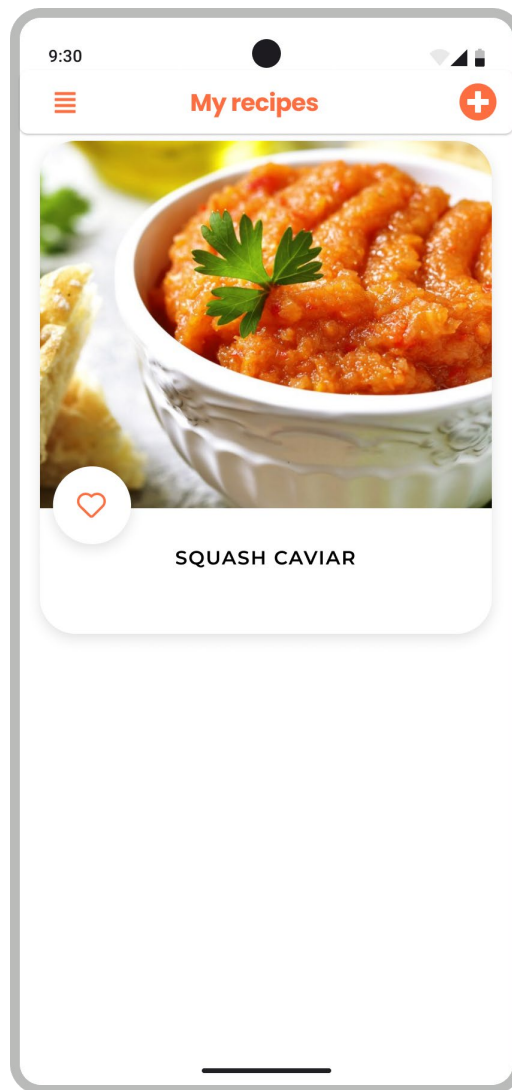


Рисунок 17 — Экран рецептов, созданных пользователем

#### 4.4 Администратор

У администратора предусмотрен следующий набор возможностей:

- проверка рецептов, созданных пользователями (рисунок 18);

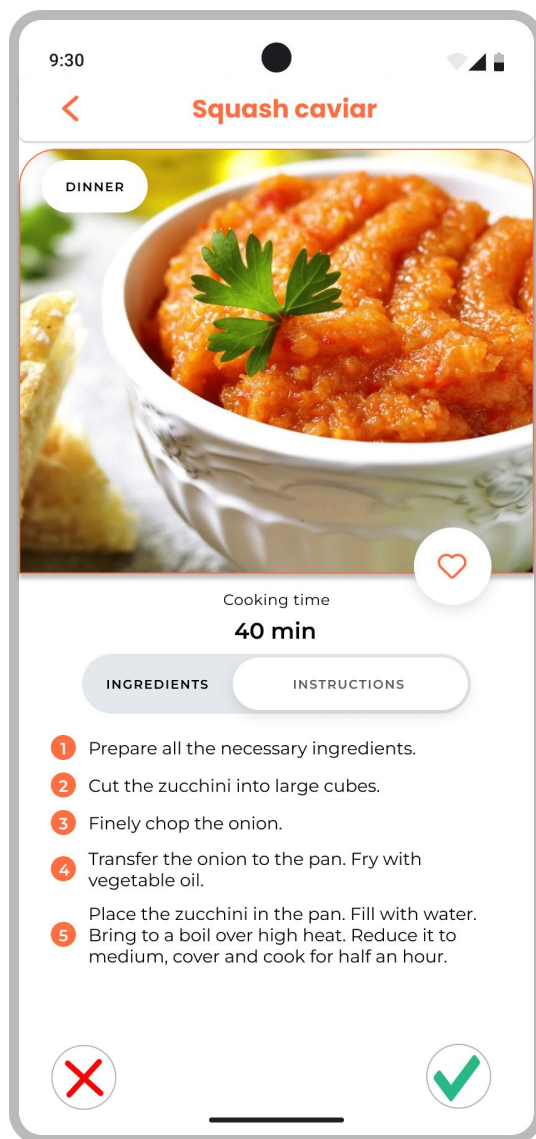


Рисунок 18 — Экран проверки рецепта

— просмотр статистических данных (рисунок 19).

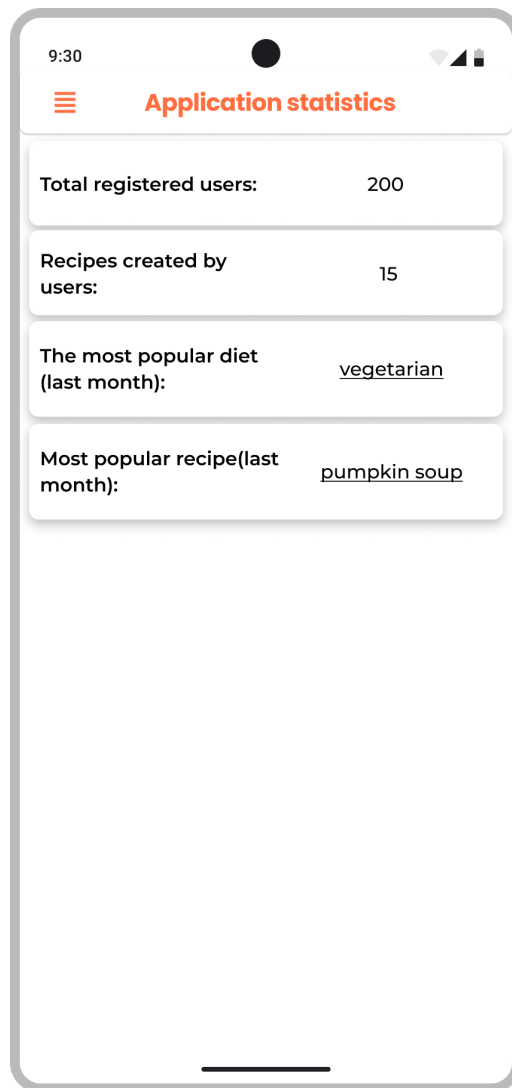


Рисунок 19 — Экран со статистическими данными

## Заключение

При выполнении работы было рассмотрено несколько аналогичных решений для просмотра рецептов различных блюд и диет, и, после их анализа и выделения недостатков, были выработаны основные идеи проекта, требования к функционалу и особенности, способные сделать разрабатываемое приложение лишённым этих недостатков.

Для разработки приложения были разработаны дизайн-макеты интерфейса, выбраны средства для разработки приложения, построены все необходимые диаграммы и написана сопутствующая документация.

Для контроля версий приложения было создано 3 репозитория GitHub – для документации, клиентской и серверной части соответственно.

При разработке приложения были реализованы следующие задачи:

- возможность просмотра рецептов;
- поиск рецептов по названию, категории и диете;
- возможность пользователей добавлять в избранное понравившиеся рецепты;
- предоставление пользователям права создания собственных рецептов;
- публикация рецептов пользователей после одобрения администратора.

Разработанное приложение удовлетворяет всем поставленным ранее требованиям.

Все основные поставленные задачи были выполнены в полном объёме.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Spring Framework Documentation [Электронный ресурс]. – Режим доступа: <https://spring.io/projects/spring-boot>. – Spring Boot. – (Дата обращения 28.04.2024).
2. PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с.
3. The Docker Book: Containerization is the new virtualization Kindle Edition / James Turnbull. - James Turnbull, 2014. – 388 с.
4. Pro Git: Everything you need to know about Git / Scott Chacon, Ben Straub. – 2-е изд. – изд. Apress, 2014. – 458 с.