

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

Факультет Компьютерных наук
Кафедра информационных систем

Курсовая работа
по разработке мобильного приложения
«Voice Chef»

Обучающийся _____ Н. М. Щербаков
Обучающийся _____ Д. С. Кочура
Обучающийся _____ Е. А. Кураков
Обучающийся _____ М. К. Аксенова
Обучающийся _____ Н. В. Ларичев

Воронеж 2025

СОДЕРЖАНИЕ

Определения, обозначения и сокращения	4
ВВЕДЕНИЕ	6
1 Постановка задачи.....	7
1.1 Проблема.....	7
1.2 Цели создания приложения.....	7
1.3 Задачи	7
2 Анализ предметной области	9
2.1 Обзор конкурентов.....	9
2.1.1 Tasty.....	9
2.1.2 Yummlы	10
2.1.3 Kitchen Stories.....	11
2.1.4 Cookpad	12
2.2 Сравнение с конкурентами	13
2.3 Вывод по сравнению аналогов	14
3 Моделирование системы	15
3.1 Функциональные требования	15
3.2 Нефункциональные требования	16
3.3 Диаграмма классов.....	17
3.4 Диаграмма последовательностей	18
3.5 ER диаграмма	20
3.6 Диаграмма развёртывания	21
4 Реализация.....	22
4.1 Средства реализации.....	22
4.1.1 Backend технологии	22
4.1.2 Frontend технологии	22
4.1.3 Преимущества выбранных технологий	23
4.2 Реализация Backend	25
4.2.1 Архитектура Voice Chef.....	25
4.2.2 Взаимодействие компонентов	26
4.2.3 Технологические особенности	27
5 Дизайн и описание основных страниц.....	28

5.1 Брендбук.....	28
5.2 Экран загрузки.....	29
5.3 Экран авторизации.....	30
5.4 Главная	31
5.5 Поиск	32
5.6 Экран категории	33
5.7 Экран создания рецепта	34
5.8 Экран существующего рецепта	35
5.9 Профиль	36
ЗАКЛЮЧЕНИЕ	37
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	38

Определения, обозначения и сокращения

Frontend (FE) – это клиентская часть приложения (интерфейс, с которым взаимодействует пользователь), FE разработчик – человек, отвечающий за разработку «внешней» части приложения.

Backend (BE) – это программно-аппаратная часть приложения (логика приложения, скрытая от пользователя), BE разработчик – человек, отвечающий за разработку «внутренней» части приложения.

Quality assurance (QA) – это процесс обеспечение качества программного обеспечения, в контексте проекта – это обозначение человека, отвечающего за тестирование приложения.

Project manager (PM) – это человек, руководящий проектом и организующий работу команды.

DevOps — методология автоматизации технологических процессов сборки, настройки и развёртывания программного обеспечения, в контексте проекта, человек, отвечающий за развёртывания программного обеспечения.

Сервер – компьютер (или специальное компьютерное оборудование), выделенный и/или специализированный для выполнения определенных сервисных функций, таких как хранение информации, передача информации между связанными с ним устройствами.

Qwen – семейство больших языковых моделей, разработанных Alibaba.

Api – это механизмы, которые позволяют двум программным компонентам взаимодействовать друг с другом, используя набор определений и протоколов.

Python – это язык программирования, который широко используется в интернет-приложениях, разработке программного обеспечения, науке о данных и машинном обучении.

React native – это фреймворк для языка программирования JavaScript. Этот фреймворк позволяет писать на JS приложения для систем Android и iOS.

Typescript – это язык программирования, который является доработанной версией языка JavaScript, используемого разработчиками для создания интерактивных веб-страниц.

PostgreSQL – это свободно распространяемая объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом.

Miro – это инновационное рабочее пространство, в котором команды управляют проектами и разрабатывают продукты, в контексте проекта используется для построения диаграмм.

Figma – это графический редактор для совместного проектирования сайтов, приложений и других дизайнерских продуктов.

Jira – это инструмент управления рабочим процессом для команд разработчиков ПО, которые хотят систематизировать и отслеживать свою работу.

GitHub – это облачная платформа для хостинга IT-проектов и совместной разработки, использующая популярную систему контроля версий Git.

ВВЕДЕНИЕ

Мобильные приложения играют важную роль в повседневной жизни, облегчая выполнение бытовых задач. Одной из таких задач является приготовление пищи. Человек всегда старается приготовить что-то вкусное, однако для этого необходимо знать рецепт блюда. В открытом доступе есть большое количество рецептов, но найденный рецепт легко потерять, поэтому его нужно хранить в каком-то месте.

Есть много сайтов и приложений с рецептами, однако пользователи часто сталкиваются с неудобствами: необходимо переключаться между шагами рецепта, отвлекаться от процесса готовки, теряют найденные рецепты. Решением является мобильное приложение Voice Chef, объединяющее хранение, озвучивание и персонализацию рецептов.

Voice Chef — это hands-free помощник, позволяющий готовить, не прикасаясь к экрану. Пользователь может добавлять и настраивать рецепты, управлять процессом с помощью голоса, получать персональные рекомендации. Это делает приложение особенно полезным для семей с детьми, занятых пользователей и людей с ограничениями по зрению или моторике.

1 Постановка задачи

1.1 Проблема

При анализе рынка было выявлено, что существующие приложения с рецептами имеют некоторые недостатки.

Современные кулинарные приложения требуют постоянного визуального и тактильного взаимодействия, что неудобно в процессе готовки.

1.2 Цели создания приложения

Нашей командой разработки было поставлено несколько целей, основные пункты:

- разработать приложение для хранения рецептов блюд в одном месте, что позволит пользователю собирать и быстро находить свои любимые рецепты;
- реализовать возможность озвучки рецептов с мобильного устройства;
- создать персональные рекомендации контента с помощью AI, которые будут учитывать все предпочтения пользователя.

1.3 Задачи

Задачи, которые необходимо решить:

- анализ рынка мобильных приложений, связанных с приготовлением пищи;
- разработка архитектуры приложения с использованием современных технологий мобильной разработки;
- проектирование и реализация пользовательского интерфейса;
- реализация голосовой озвучки рецепта;
- реализация распознавания голосовых команд и управления воспроизведением рецептов;
- внедрение AI алгоритмов подбора рецептов на основе пользовательского поведения и имеющихся ингредиентов;
- проверка функциональности, стабильности и корректности работы всех компонентов.

2 Анализ предметной области

2.1 Обзор конкурентов

2.1.1 Tasty

Формат: Короткие, динамичные видео-рецепты, оптимизированные для просмотра в социальных сетях.

Сильные стороны:

- высококачественный видеоконтент с яркой подачей;
- большая библиотека рецептов, охватывающая разные кухни мира;
- простота восприятия рецептов — визуальные инструкции не требуют текста;
- сильный бренд: проект от BuzzFeed с миллионной аудиторией и активным сообществом.

Слабые стороны:

- отсутствие голосового сопровождения и пошаговой озвучки;
- пользователь не может управлять видео без прикосновений, что неудобно во время готовки;
- нет персонализации под ингредиенты пользователя или диетические предпочтения.

Монетизация:

- партнёрские рецепты с брендами (например, Heinz, McCormick);
- встраиваемая реклама в контент;
- собственная линейка товаров (Tasty cookware).

2.1.2 Yummly

Формат: Интеллектуальный подбор рецептов на основе пользовательских предпочтений, ингредиентов, техники приготовления.

Сильные стороны:

- сильная персонализация: система учитывает диеты, аллергию, вкусы;
- возможность составлять список покупок прямо из рецепта;
- интеграция с техникой Whirlpool: пользователь может отправлять рецепт прямо на духовку.

Слабые стороны:

- голосовой интерфейс реализован только для устройств Whirlpool, недоступен массовому пользователю;
- интерфейс перегружен функциональностью — возможен барьер входа;
- многие функции доступны только после регистрации и настройки.

Монетизация:

- подписка на Yummly Pro: доступ к premium-рецептам от шеф-поваров;
- партнёрства с брендами и ретаргетинг по продуктовым предпочтениям.

2.1.3 Kitchen Stories

Формат: Пошаговые рецепты с фотографиями, видеороликами, инструкциями и статьями о кулинарии.

Сильные стороны:

- премиум-контент от редакции: красивые фотографии, понятные инструкции;
- интерфейс, удобный для начинающих и опытных поваров;
- хорошо структурированные рецепты с таймерами, списками покупок и питательной ценностью.

Слабые стороны:

- отсутствие пользовательского контента: нельзя добавлять свои рецепты;
- нет голосового сопровождения, что ограничивает использование в процессе готовки;
- контент более “глянцевый”, чем практичный — не всегда подходит для повседневной кухни.

Монетизация:

- платная подписка на premium-функции;
- партнёрские проекты с брендами (например, IKEA, Zwilling).

2.1.4 Cookpad

Формат: Социальная платформа для обмена пользовательскими рецептами, аналог «Инстаграма для поваров».

Сильные стороны:

- огромная база рецептов, ежедневно пополняемая пользователями;
- возможность комментировать, ставить лайки и сохранять рецепты в избранное;
- простота публикации: можно загрузить фото с описанием прямо с телефона.

Слабые стороны:

- качество рецептов варьируется: много дубликатов, нет модерации или проверки;
- интерфейс может быть перегружен и не структурирован;
- нет озвучки, голосового управления или профессионального контента.

Монетизация:

- freemium-модель: базовый функционал бесплатный, подписка открывает доступ к некоторым функциям и рекомендациям;
- реклама и партнёрский контент.

2.2 Сравнение с конкурентами

Таблица 1 - Сравнение с конкурентами

Сервис	Функционал	Модель оплаты	Особенности
Voice Chef	Голосовая озвучка рецептов, пошаговая навигация, подбор по ингредиентам, добавление собственных рецептов	Freemium: базовая версия бесплатна, Pro-подписка (расширенные лимиты и функции)	Удобен для готовки с занятыми руками. Пользователь сам добавляет рецепты. Персонализация. AI-рекомендации на основе поведения. Простота, доступность для пользователей с ограничениями.
Tasty	Видео-рецепты с пошаговой демонстрацией	Бесплатно, монетизация через рекламу и партнёрства	Красочный визуальный контент, высокая узнаваемость. Нет озвучки и голосовой навигации. Упор на видео.
Yummly	Персонализированные рецепты, подбор по ингредиентам, список покупок	Freemium + подписка Yummly Pro	Интеграция с техникой Whirlpool. Голосовое управление ограничено только “умной” техникой.
Kitchen Stories	Пошаговые рецепты, видео, статьи, таймеры	Бесплатно + подписка на премиум-контент	Удобный интерфейс. Контент от редакции. Нет пользовательских рецептов и озвучки шагов.
Cookpad	Платформа с пользовательскими рецептами	Бесплатно + премиум-функции по подписке	Активное сообщество, простой формат публикации. Нет голосового управления.

2.3 Вывод по сравнению аналогов

Voice Chef выгодно отличается от конкурентов уникальной функцией голосового озвучивания рецептов и пошаговой навигации, что делает его удобным для использования во время готовки. В отличие от конкурентов, приложение ориентировано на hands-free формат, что особенно актуально для занятых пользователей и людей с ограничениями. Ниша голосовых кулинарных помощников практически свободна, а значит, при грамотной реализации и продвижении Voice Chef имеет все шансы занять своё место на рынке и предложить ценность, которую не покрывают существующие решения.

3 Моделирование системы

3.1 Функциональные требования

Система требует обязательно авторизоваться или зарегистрироваться в приложении при первом входе.

Система позволяет:

- взаимодействовать с приложением голосом;
- добавлять категории блюд;
- добавлять рецепты блюд;
- просматривать список категорий блюд;
- просматривать список рецептов блюд;
- открывать рецепт блюда;
- просматривать рецепт блюда;
- изменять рецепт блюда;
- запускать озвучку рецепта;
- останавливать озвучку рецепта;
- возобновлять озвучку рецепта;
- отключать озвучку рецепта;
- настраивать название блюда;
- настраивать изображение блюда;
- настраивать название рецепта;
- настраивать изображение рецепта;
- настраивать количество порций;
- настраивать время приготовления;
- добавлять и настраивать этапы приготовления рецепта и время, затрачиваемое на них;
- просматривать список рекомендуемых рецептов;
- редактировать свой профиль;
- выйти из профиля.

3.2 Нефункциональные требования

- производительность: Приложение должно быстро обрабатывать команды и отвечать в течение 5 секунд;
- поддержка Android 7;
- безопасность: Шифрование пользовательских данных;
- оптимизированные запросы к PostgreSQL с использованием индексов и эффективных структур данных;
- использование Qwen AI для системы рекомендаций на основе предпочтений пользователя.

3.3 Диаграмма классов

Диаграмма классов показывает основные сущности (классы) системы и связи между ними.

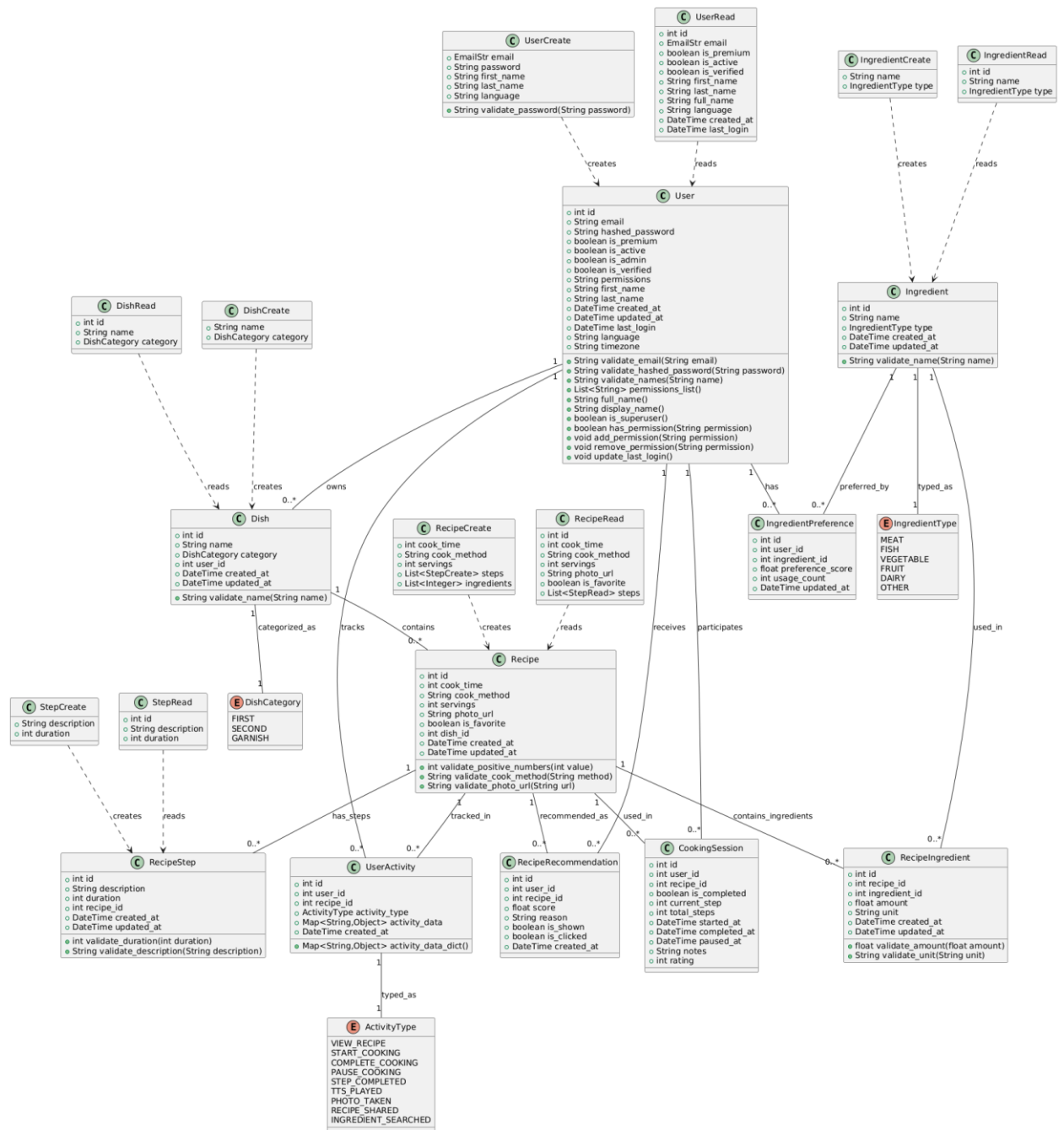


Рисунок 1 - Диаграмма классов

3.4 Диаграмма последовательностей

Диаграмма последовательностей отображает порядок взаимодействия между объектами при выполнении определённого сценария.

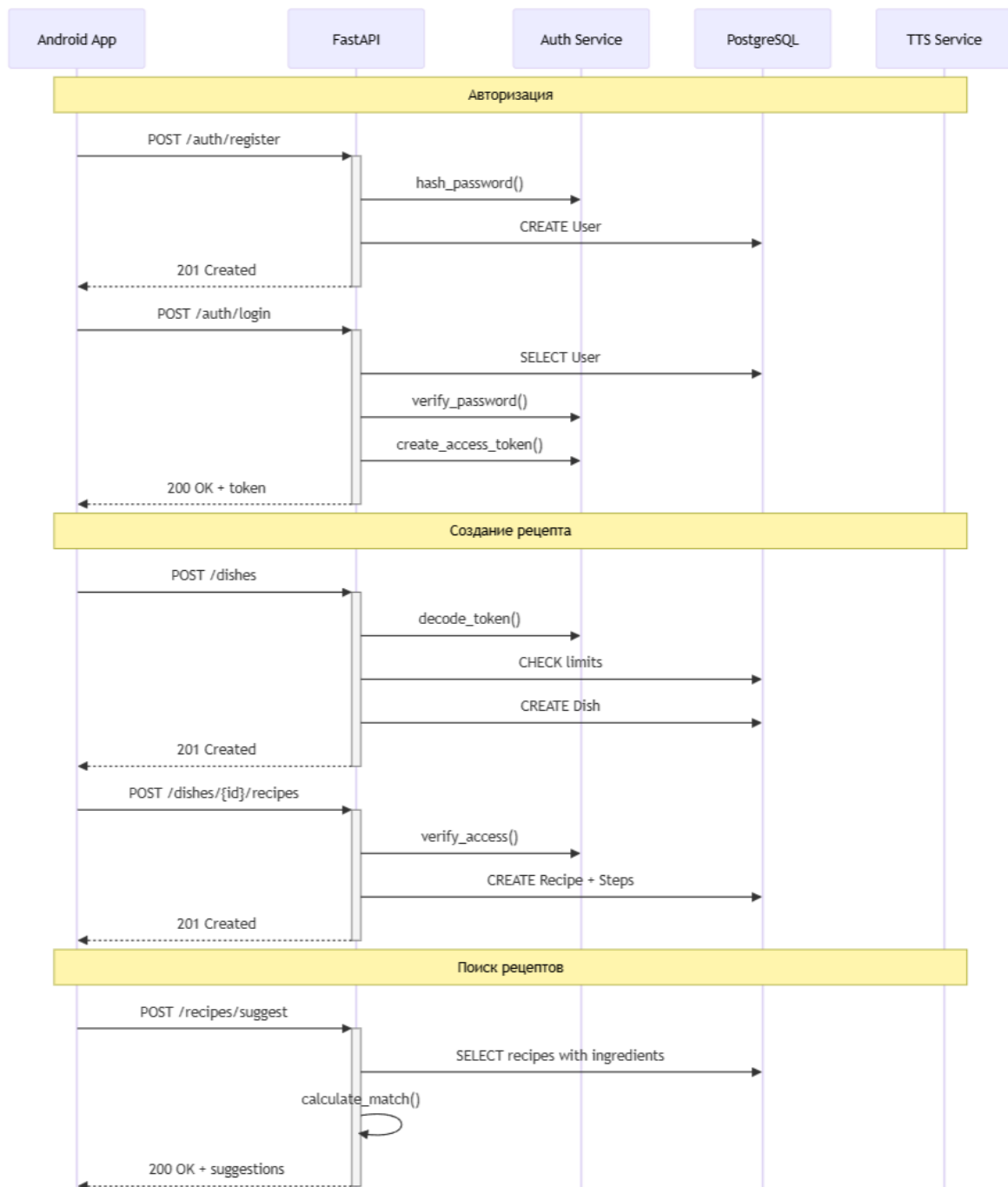


Рисунок 2 - Диаграмма последовательностей

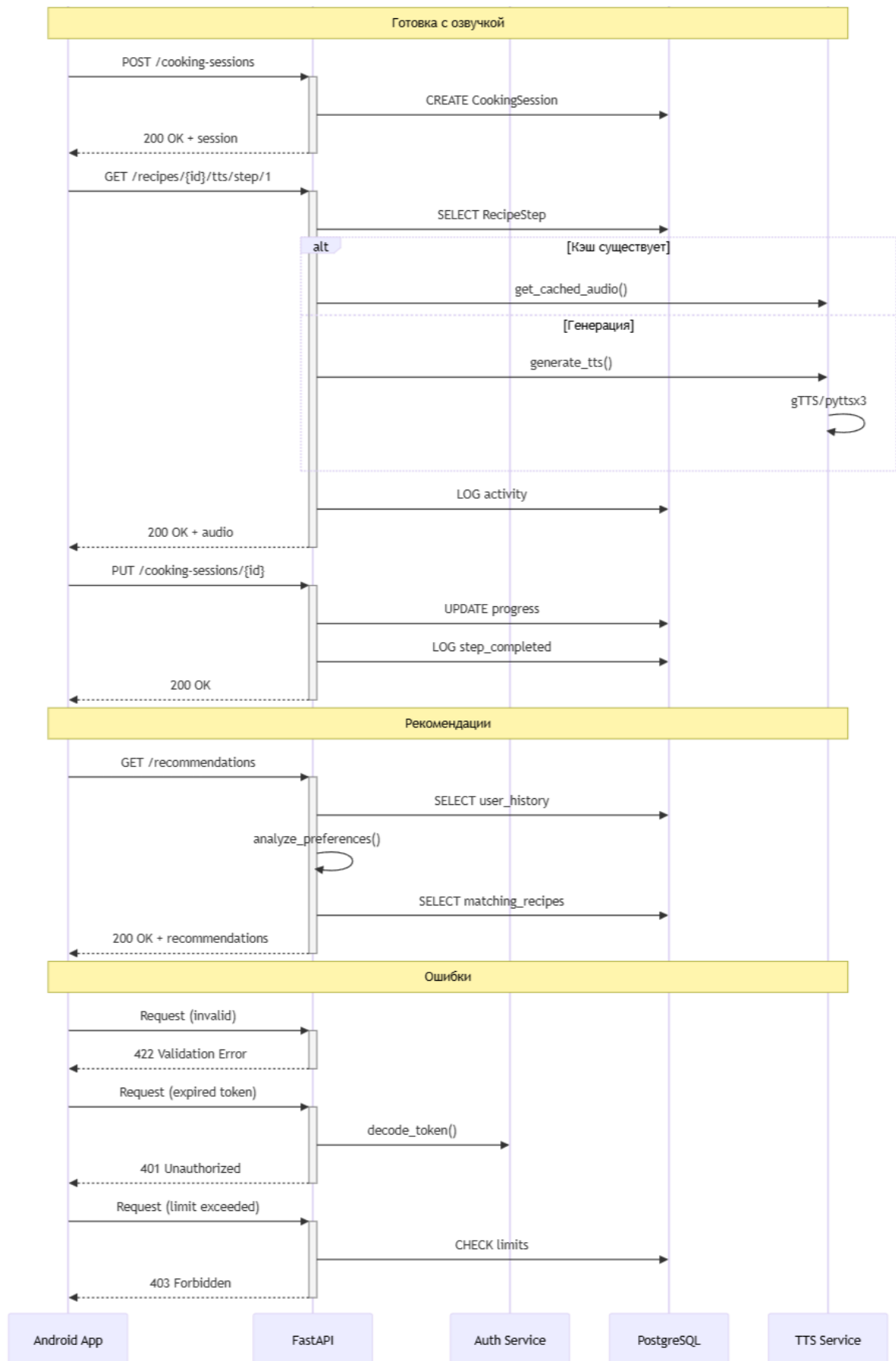


Рисунок 3 - Продолжение диаграммы последовательностей

3.5 ER диаграмма

ER диаграмма оказывает логическую структуру базы данных — таблицы (сущности), поля и связи между ними.

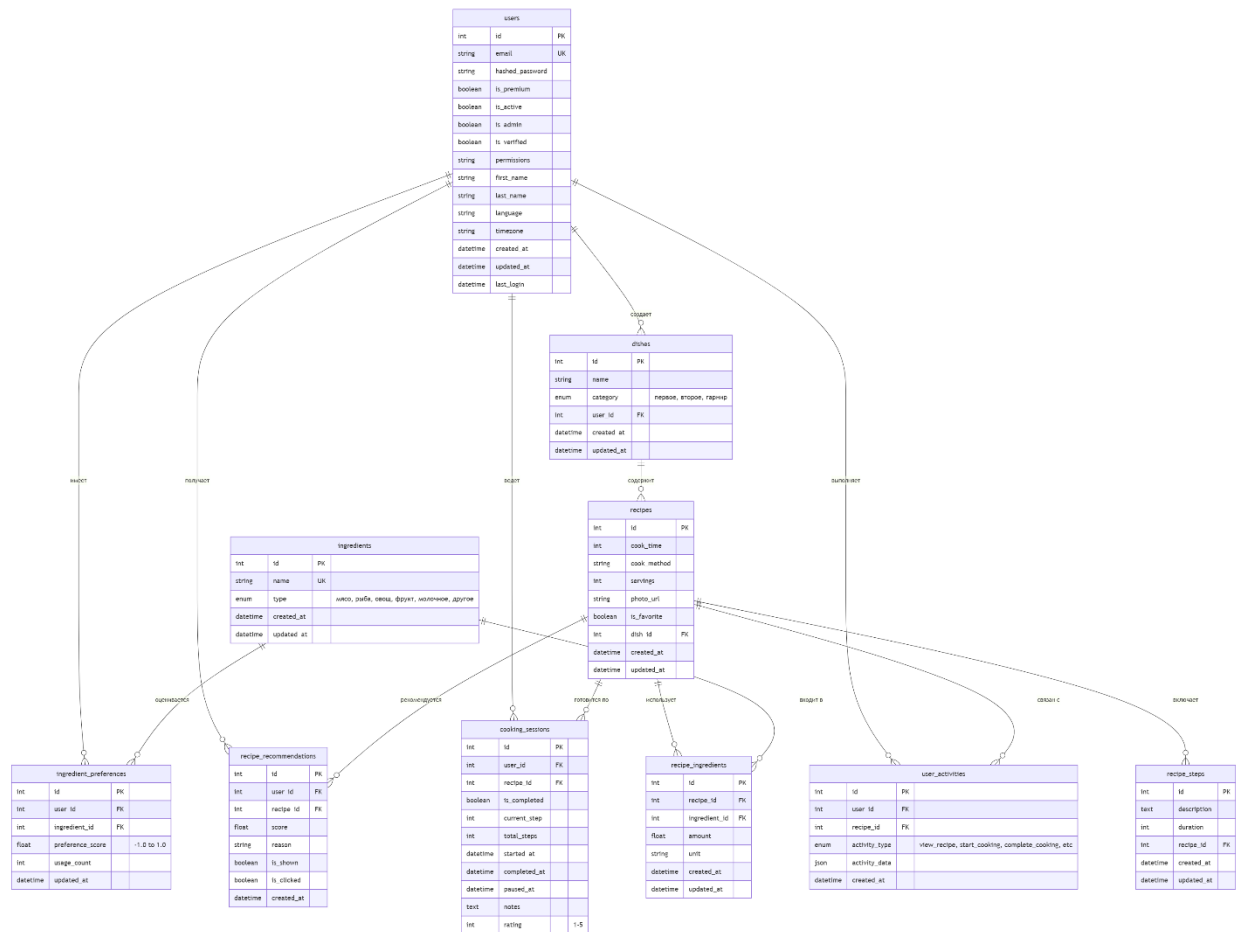


Рисунок 4 - ER диаграмма

3.6 Диаграмма развёртывания

Диаграмма развёртывания иллюстрирует размещение компонентов системы на физической или виртуальной инфраструктуре.

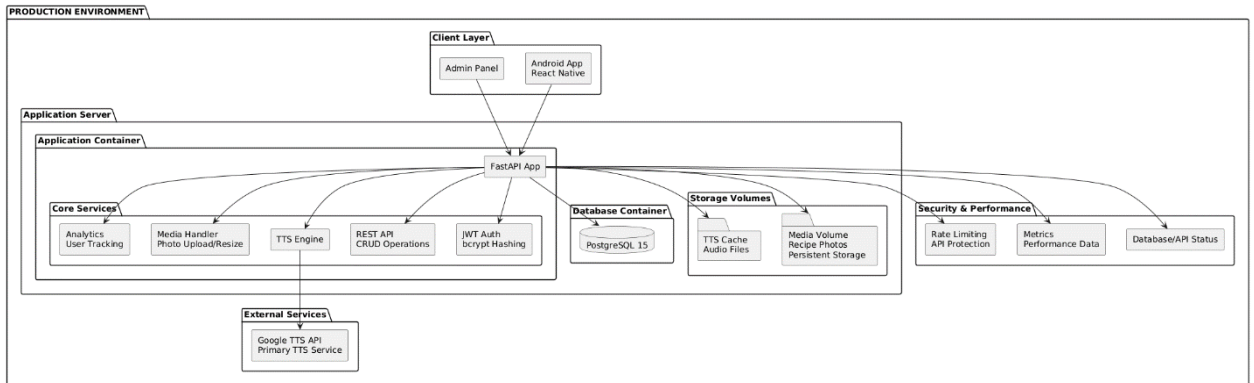


Рисунок 5 - Диаграмма развёртывания

4 Реализация

4.1 Средства реализации

4.1.1 Backend технологии

Технологии, используемые при разработке Backend:

- язык программирования Python версия 3.12;
- фреймворк FastAPI для создания высокопроизводительного REST API;
- СУБД PostgreSQL 15 для надежного хранения данных;
- инструмент для создания документации API OpenAPI/Swagger с автогенерацией;
- SQLAlchemy ORM для работы с базой данных;
- Alembic для управления миграциями схемы БД;
- Pydantic для валидации данных и сериализации;
- Passlib с bcrypt для безопасного хеширования паролей;
- Python-jose для работы с JWT токенами.

Дополнительные сервисы:

- Google TTS API для высококачественной генерации речи;
- aiofiles для асинхронной работы с файловой системой;
- python-multipart для обработки загрузки файлов.

Для развертывания и администрирования приложения используются следующие средства:

- Docker и Docker Compose для контейнеризации и оркестрации сервисов;
- PostgreSQL в Docker контейнере для изоляции БД;
- Nginx в качестве reverse проху и для обслуживания статических файлов;
- система логирования для мониторинга работы приложения.

4.1.2 Frontend технологии

Для реализации клиентской части приложения были использованы следующие средства:

- платформа React Native для кроссплатформенной разработки;
- язык программирования JavaScript/TypeScript;
- Expo CLI для ускоренной разработки и тестирования.

4.1.3 Преимущества выбранных технологий

В качестве преимуществ выбранных технологий можно отметить следующие пункты.

Для Python и FastAPI:

- высокая производительность - FastAPI обеспечивает скорость, сравнимую с Node.js и Go;
- автоматическая генерация документации - OpenAPI схема создается автоматически из кода;
- встроенная валидация данных через Pydantic с детальными сообщениями об ошибках;
- асинхронная поддержка для эффективной обработки I/O операций;
- современный Python синтаксис с поддержкой type hints;
- богатая экосистема библиотек для работы с TTS, обработкой файлов и аналитикой;
- простое тестирование с встроенным TestClient и поддержкой pytest.

Для PostgreSQL:

- надежность и ACID совместимость для критически важных данных;
- расширенные возможности (JSON поддержка, полнотекстовый поиск, индексы);
- масштабируемость с поддержкой репликации и шардинга;
- открытый исходный код без лицензионных ограничений;
- отличная производительность для сложных запросов и аналитики;
- богатые типы данных включая массивы, JSON, временные ряды.

Для React Native:

- кроссплатформенность - один код для Android и iOS;
- производительность нативных приложений благодаря мосту к нативным API;
- обширная экосистема пакетов и компонентов;
- Hot Reload для быстрой разработки и отладки;
- возможность использования нативного кода при необходимости;

Для Docker и контейнеризации:

- изоляция приложений и их зависимостей;
- воспроизводимость окружения на любой инфраструктуре;
- упрощенное развертывание через Docker Compose;
- масштабируемость сервисов независимо друг от друга;
- контроль ресурсов и мониторинг производительности;
- обновления без простоя через rolling updates.

Для архитектурных решений:

- монолитная архитектура с модульной структурой обеспечивает простоту разработки на начальном этапе с возможностью будущего разделения на микросервисы;
- RESTful API предоставляет стандартизированный интерфейс для мобильного приложения;
- JWT авторизация обеспечивает stateless аутентификацию без необходимости серверных сессий;
- слоевая архитектура (Router-Service-Repository) обеспечивает четкое разделение ответственности и упрощает тестирование.

Выбранный технологический стек обеспечивает высокую производительность, безопасность и масштабируемость приложения Voice

Chef, а также позволяет быстро итерировать и добавлять новые функции в соответствии с потребностями пользователей.

4.2 Реализация Backend

4.2.1 Архитектура Voice Chef

Наше приложение Voice Chef построено на основе монолитной архитектуры с модульной структурой и состоит из следующих компонентов:

- **Authentication Service** тот обеспечивает безопасную регистрацию, авторизацию пользователей и управление JWT токенами. Поддерживает разграничение прав доступа между обычными пользователями, премиум пользователями и администраторами.
- **Recipe Management Service** предназначен для управления блюдами, рецептами, шагами приготовления и ингредиентами. Обеспечивает CRUD операции с валидацией данных и контролем лимитов пользователей.
- **Media Service** обеспечивает загрузку, обработку и хранение медиафайлов (фотографии рецептов). Поддерживает различные лимиты размера файлов для обычных и премиум пользователей.
- **TTS Service**, основная функция данного сервиса заключается в генерации аудиофайлов для озвучки шагов рецептов. Использует Google TTS API как основной сервис и pyttsx3 как fallback решение для работы в автономном режиме.
- **Analytics Service** – сервис для сбора и анализа пользовательской активности, генерации персональных рекомендаций на основе предпочтений и истории готовки пользователей.
- **Admin Service**, административная панель предназначена для управления системой, мониторинга пользователей, модерации контента и получения статистики использования приложения.
- **Google TTS API** – внешний сервис для высококачественной генерации речи, используемый для озвучки шагов рецептов.

4.2.2 Взаимодействие компонентов

Взаимодействие клиента с серверной частью.

Мобильное приложение и Authentication Service.

Регистрация, авторизация и управление токенами осуществляются через стандартные HTTP REST API запросы.

Мобильное приложение и Recipe Management Service.

Через HTTP API осуществляется работа с блюдами, рецептами и ингредиентами, включая создание, редактирование, поиск и подбор рецептов.

Мобильное приложение и Media Service.

Загрузка и получение фотографий рецептов через HTTP-запросы с поддержкой multipart/form-data.

Мобильное приложение и TTS Service.

Получение аудиофайлов озвучки шагов рецептов через HTTP с возвращением аудио потока.

Взаимодействие между сервисами.

Recipe Management и Analytics Service.

Передача данных о пользовательской активности для анализа предпочтений и генерации рекомендаций через внутренние HTTP вызовы.

TTS Service и Google TTS API.

Интеграция с внешним API Google TTS для генерации высококачественной речи через HTTP запросы.

Analytics Service и Recipe Management.

Сбор данных о взаимодействии пользователей с рецептами для персонализации рекомендаций.

Admin Service и All Services.

Административная панель получает агрегированные данные от всех сервисов для отображения статистики и управления системой.

4.2.3 Технологические особенности

Для взаимодействия между различными частями системы применяются следующие протоколы:

- HTTP REST API – основной протокол для всех взаимодействий между мобильным приложением и серверными сервисами;
- HTTPS обеспечение безопасности передачи данных с SSL/TLS шифрованием;
- File Streaming – эффективная передача аудиофайлов и изображений с поддержкой кэширования.

5 Дизайн и описание основных страниц

5.1 Брендбук

После анализа рынка для приложения были выбраны определённые цвета: оттенки зелёного, белый. Иконка в виде микрофона с шапкой шефа.



Рисунок 6 - Иконка приложения



Рисунок 7 - Брендбук

5.2 Экран загрузки

При запуске приложения пользователь видит экран предварительной загрузки, на котором расположены:

- название приложения (логотип или текст);
- анимация загрузки (лоадер);
- экран отображается до полной загрузки главной страницы.



Рисунок 8 - Экран загрузки

5.3 Экран авторизации

При первом входе в приложение или входе без авторизации открывается экран, на котором расположены:

- поля ввода: логин, имя, пароль;
- кнопка «Авторизоваться»;
- кнопка «Восстановить пароль».

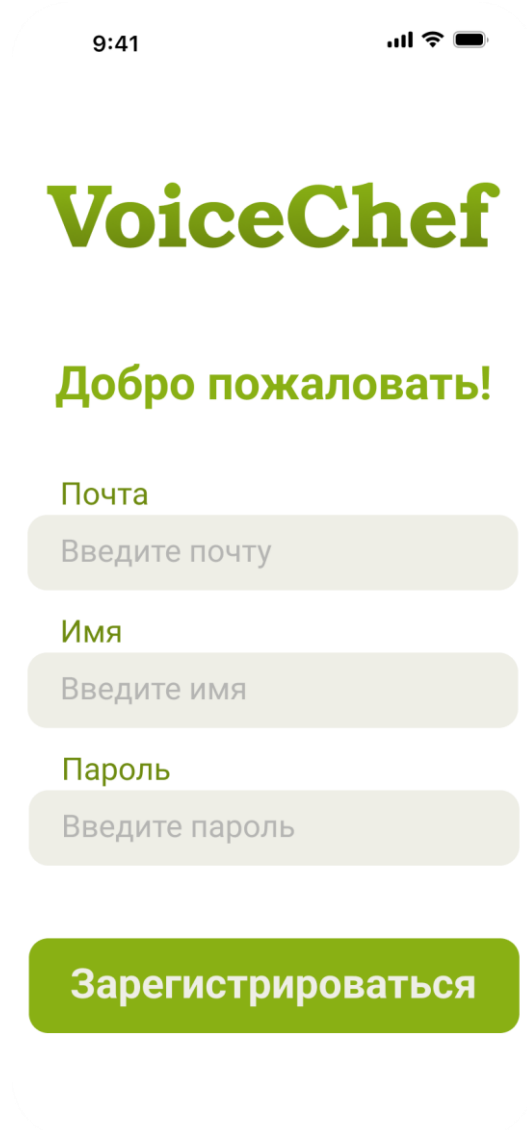


Рисунок 9 - Экран авторизации

5.4 Главная

Основной экран после входа в приложение, на котором расположены:

- вверху: кнопка «+» для создания новой категории;
- в центре: список категорий;
- под каждой категорией – список блюд;
- в конце списка – кнопка «Добавить блюдо», переход на экран создания блюда.



Рисунок 10 - Экран авторизации

5.5 Поиск

Этот экран позволяет искать рецепты, на нём расположены:

- вверху: поле ввода текста;
- под полем: список рекомендованных рецептов;
- при вводе запроса и нажатии на иконку лупы – выводятся результаты поиска, заменяя рекомендации.

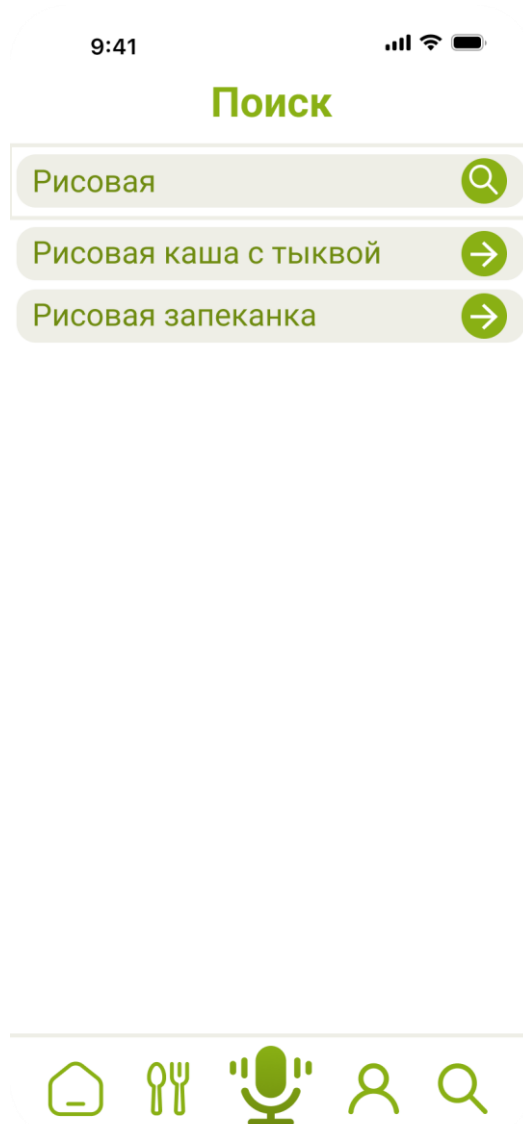


Рисунок 11 - Поиск

5.6 Экран категории

Открывается при выборе категории, на этом экране расположены:

- вверху: кнопка «+» – переход на экран добавления блюда;
- в центре: список блюд данной категории;
- под каждым блюдом – список рецептов;
- последний элемент – кнопка «Добавить рецепт» (открывает экран создания рецепта).

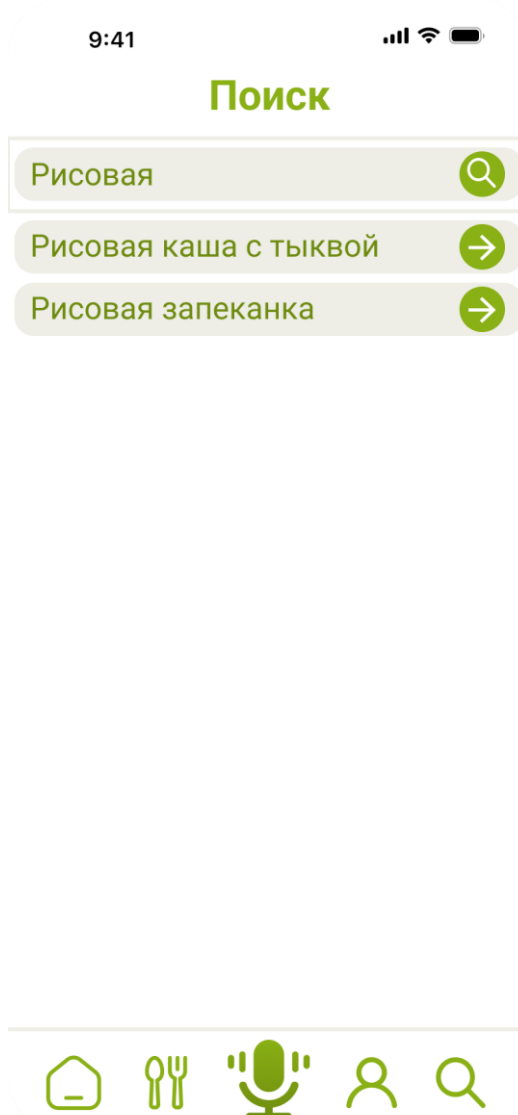


Рисунок 12 - Экран категории

5.7 Экран создания рецепта

Данный экран позволяет добавить новый рецепт, на нём расположены:

- вверху справа: кнопка «Создать»;
- кнопки: добавить фото, добавить этапы, добавить ингредиенты;
- поля: название рецепта, описание этапов, количество порций, время приготовления, ингредиенты и их количество;
- внизу: выбор категории.

9:41

Ваш рецепт

Добавьте фото

Введите название

Количество порций: 1

Время приготовления: __ : __

Ингредиенты

Ингредиент	Кол-во
+	

Этапы

1 Опишите этап

+

Категории

Завтраки

Home Fork and Knife Microphone Profile Search

Рисунок 13 - Экран создания рецепта

5.8 Экран существующего рецепта

Экран просмотра и редактирования рецепта, на котором расположены:

- отображаются: фото, название, категория, ингредиенты, этапы;
- все элементы доступны для редактирования;
- внизу экрана (тапбар): кнопка микрофона – позволяет взаимодействовать с приложением голосом: включить, приостановить, возобновить или остановить озвучку рецепта.



Рисунок 14 - Экран существующего рецепта

5.9 Профиль

Этот экран содержит пользовательские данные:

- поля: логин, имя, пароль;
- кнопки: «Выйти» из аккаунта, «Удалить аккаунт».

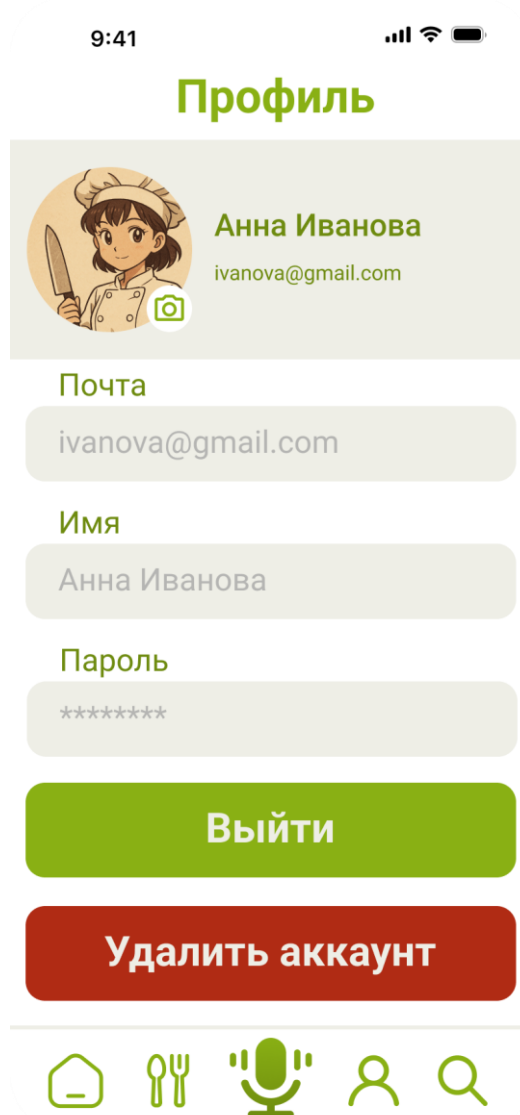


Рисунок 15 - Профиль

ЗАКЛЮЧЕНИЕ

В рамках данной курсовой работы было разработано мобильное приложение Voice Chef, предназначенное для голосового сопровождения рецептов и взаимодействия с пользователем в процессе приготовления пищи. Проект решает актуальные проблемы, с которыми сталкиваются современные пользователи: необходимость постоянно отвлекаться на экран, отсутствие персонализированного подхода и слабая адаптация под людей с ограниченными возможностями.

Проект имеет высокий потенциал к масштабированию, как в техническом, так и в коммерческом плане. Возможна локализация под международные рынки, интеграция с умными колонками и кухонной техникой, а также расширение функционала для новых сегментов пользователей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация FastAPI [Электронный ресурс]. – Режим доступа: <https://fastapi.tiangolo.com/ru/tutorial/> – Заглавие с экрана. – (Дата обращения 15.05.2025).
2. Документация Docker [Электронный ресурс]. – Режим доступа: <https://docs.docker.com/> – Заглавие с экрана. – (Дата обращения 20.05.2025).
3. Документация Swagger [Электронный ресурс]. – Режим доступа: <https://swagger.io/docs/open-source-tools/swagger-editor/> – Заглавие с экрана. – (Дата обращения 10.05.2025).
4. Документация JWT [Электронный ресурс]. – Режим доступа: <https://jwt.io/introduction> – Заглавие с экрана. – (Дата обращения 05.05.2025).