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

Факультет Компьютерных наук

Кафедра информационных систем

Техническое задание
на разработку мобильного приложения
«Voice Chef»

Исполнители

_____ Н.М. Щербаков

_____ Д.С. Кочура

_____ Е.А. Кураков

_____ М.К. Аксенова

Заказчик

_____ В.С. Тарасов

Воронеж 2025

СОДЕРЖАНИЕ

Термины 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 версия языка JavaScript, использующегося 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.

1 Общие положения

1.1 Наименование и условное обозначение приложения

Наименование приложения: «Приложение, которое озвучивает рецепты, подбирает ингредиенты и взаимодействует с пользователем голосом VoiceChef».

Условное обозначение приложения: «VoiceChef».

1.2 Заказчик и исполнители

Заказчик: старший преподаватель Тарасов Вячеслав Сергеевич.

Разработчик: 1 команда 2 группы

Состав команды и распределение ролей:

— Щербаков Никита – team lead, PM

— Кочура Даниил – аналитик, BE разработчик

— Кураков Евгений – QA, FE разработчик

— Аксенова Марина – DevOps, дизайнер

1.3 Перечень документов

— ГОСТ 7.32

— ГОСТ 34.602-2020

1.4 Обоснование создания приложения

1.4.1 Проблема и поставленные цели

В современном мире люди постоянно сталкиваются с необходимостью готовить себе пищу. 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.

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

Нашей командой разработки было поставлено несколько целей, основные пункты, которые должны присутствовать в разрабатываемом приложении:

— Хранение рецептов блюд в одном месте, что позволит пользователю собирать и быстро находить свои любимые рецепты;

— Создание рецептов с подробным описанием этапов приготовления;

— Облегчение процесса приготовления блюд путём озвучки рецептов.

1.4.2 Цели

— Создать удобное приложение для хранения рецептов

— Использовать в приложении ИИ

— Получить прибыль от создания приложения

1.4.3 Задачи

— Разработать и протестировать приложение

— Внедрить ИИ в систему рекомендаций

— Создать бизнес модель

1.4.4 Критерии успешности

— Хорошая организация разработки;

— Чётко сформулированные цели и задачи;

— Создание заложенной функциональности;

— Удовлетворение потребностей клиента;

— Получение выгоды от разработки проекта;

— Выполнение поставленных задач.

2 Требования к приложению

2.1 Функциональные требования

2.1.1 Неавторизованный пользователь

— Неавторизованный пользователь обязательно должен авторизоваться или зарегистрироваться в приложении при первом входе.

2.1.2 Авторизованный пользователь

— 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.

2.2 Нефункциональные требования

— Производительность: Приложение должно быстро обрабатывать команды и отвечать в течение 5 секунд;

— Кроссплатформенность: Поддержка Android и iOS;

— Безопасность: Шифрование пользовательских данных;

— Интуитивный интерфейс: Приложение интуитивно понятно для пользователей, простой и лаконичный дизайн.

3 Пользовательские сценарии

Состав и содержание работы по созданию приложения включает в себя следующие этапы.

Роли пользователей:

— Обычный пользователь. Может добавить 15 блюд и 3 рецепта, не может кастомизировать приложение.

— Пользователь с подпиской. Может добавить 45 блюд и 5 рецептов, может кастомизировать приложение.

3.1 Регистрация и авторизация

Как пользователь, я хочу зарегистрироваться в приложении, чтобы получить доступ к функционалу.

— Регистрация через e-mail;

— Вход по логину и паролю;

— Восстановление пароля.

3.2 Работа с рецептами

Как пользователь, я хочу

— Добавлять, редактировать и удалять рецепты;

— Назначать категории для рецептов;

— Просматривать рецепты, изображения и описание;

— Настраивать параметры (время приготовления, порции);

— Добавлять этапы приготовления и устанавливать таймеры.

3.3 Голосовое взаимодействие

Как пользователь, я хочу

— Запускать озвучивание рецепта голосом;

— Управлять озвучиванием (пауза, возобновление, стоп);

— Получать голосовые подсказки при готовке.

3.4 Ингредиенты и подбор продуктов

Как пользователь, я хочу

— Получать список ингредиентов по рецепту;

— Заменять или исключать ингредиенты.

3.5 Рекомендации

Как пользователь, я хочу

— Видеть подборки рецептов на основе предпочтений;

— Получать предложения рецептов на основе имеющихся продуктов.

3.6 Управление профилем

Как пользователь, я хочу

— Редактировать профиль;

— Выходить из аккаунта.

4 Перечень основных функциональных блоков

— Авторизация и регистрация;

— Работа с рецептами;

— Голосовое взаимодействие;

— Подбор ингредиентов;

— Рекомендательная система;

— Управление профилем.

5 Ограничения проекта

Сроки: 6 месяцев на разработку.

5.1 Технологический стек

5.1.1 Backend

— Python;

— Qwen;

— PostgreSQL;

— FastAPI.

5.1.2 Frontend

— React Native;

— TypeScript.

5.1.3 Инструменты

— Miro;

— Figma;

— Jira;
— GitHub.

5.2 Технические риски

5.2.1 Проблемы с производительностью и масштабируемостью

Риск:

Высокая нагрузка на сервер при увеличении числа пользователей, особенно при обработке голосовых команд и генерации рекомендаций.

Замедление работы API при выполнении сложных запросов к базе данных.

Возможные решения:

— Внедрить кеширование для часто запрашиваемых данных (список рецептов, рекомендации);
— Оптимизировать запросы к PostgreSQL с использованием индексов и эффективных структур данных;
— Разделить микросервисы (например, выделить сервер голосовой обработки в отдельный сервис);
— Использовать балансировщики нагрузки при высокой нагрузке;
— Гибкое масштабирование с помощью Docker + Kubernetes.

5.2.2 Ошибки обработки голосовых команд

Риск:

Низкое качество распознавания речи или ошибки в интерпретации голосовых команд пользователем.

Долгая задержка при преобразовании текста в речь или его обратной обработке.

Возможные решения:

— Использовать Qwen AI или альтернативные модели машинного обучения с возможностью обучения на пользовательских данных;
— Внедрить коррекцию ошибок ввода, анализируя наиболее частые ошибки в голосовых командах;
— Кешировать результаты голосового ввода для повторного использования без повторного запроса к AI-сервису;
— Использовать асинхронную обработку голосовых команд для уменьшения задержек.

5.2.3 Уязвимости безопасности и защита данных

Риск:

Возможность утечки личных данных пользователей (пароли, email, истории команд).

Незащищенный доступ к API, что может привести к злоупотреблению ресурсами.

Возможные решения:

— Использовать OAuth 2.0 / JWT для аутентификации пользователей;
— Хранить пароли в захешированном виде;
— Ограничить доступ к API с помощью Rate Limiting и защиты от брутфорс-атак;
— Регулярно проводить анализ уязвимостей;
— Настроить шифрование данных.

5.2.4 Проблемы с интеграцией API и внешних сервисов

Риск:

Перебои в работе Qwen AI могут привести к недоступности голосового управления.

Изменения в API сторонних сервисов (например, платежных систем или сервисов рекомендаций) могут привести к ошибкам.

Возможные решения:

— Реализовать резервные механизмы (например, переключение на альтернативный API, если основной недоступен);
— Использовать фоновую обработку API-запросов;
— Внедрить логирование ошибок API и автоматическое уведомление разработчиков о сбоях;
— Обновлять зависимости и тестировать изменения API в изолированной среде.

5.2.5 Сложности с CI/CD и развертыванием

Риск:

Ошибки при автоматическом развертывании могут привести к сбоям в продакшене.

Долгое время сборки и тестирования перед выпуском новых версий.

Возможные решения:

— Использовать GitHub Actions для автоматизации тестирования и развертывания;
— Автоматизировать мониторинг и откаты изменений в случае сбоев;
— Использовать контейнеризацию с Docker для стандартизации среды разработки и продакшена.

6 Дизайн и описание основных страниц

6.1 Брендбук

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

Рисунок 1 - Иконка приложения

Рисунок 2 - Брендбук

6.2 Экран загрузки

При входе в приложение пользователь увидит экран предварительной загрузки, на нём изображено название приложения и лоадер.

Рисунок 3 - Загрузочный экран

6.3 Главная

На главной странице приложения располагаются категории.

Вверху страницы есть кнопка «+», при нажатии на неё откроется экран создания категории.

В центре расположен список категорий, под названием категории расположен список блюд. Последний элемент в списке – кнопка, при нажатии на которую пользователь перейдёт на экран создания блюда.

Рисунок 4 - Главная страница

6.4 Экран категории

На экране категории, вверху страницы, есть кнопка «+», при нажатии на неё откроется экран добавления блюда.

В центре расположен список блюд, под названием блюда расположен список рецептов. Последний элемент в списке – кнопка, при нажатии на которую пользователь перейдёт на экран создания рецепта.

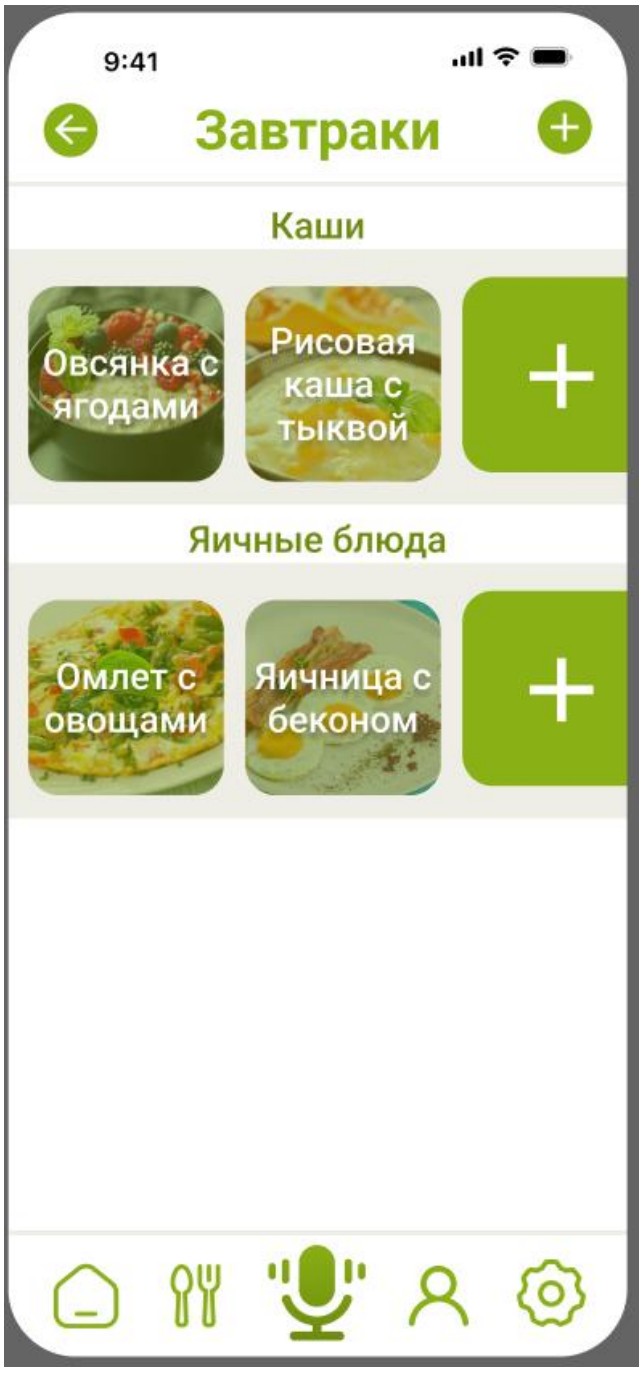

Рисунок 5 - Экран категории

6.5 Экран создания рецепта

На экране создания рецепта в правом верхнем углу кнопка создать, при нажатии на неё рецепт создаётся.

Ниже присутствуют кнопки для добавления фото, этапов, ингредиентов. Поля для названия рецепта, описания этапов, ввода количества порций, время приготовления, ввода ингредиента и его количества. Внизу страницы выбор категории.

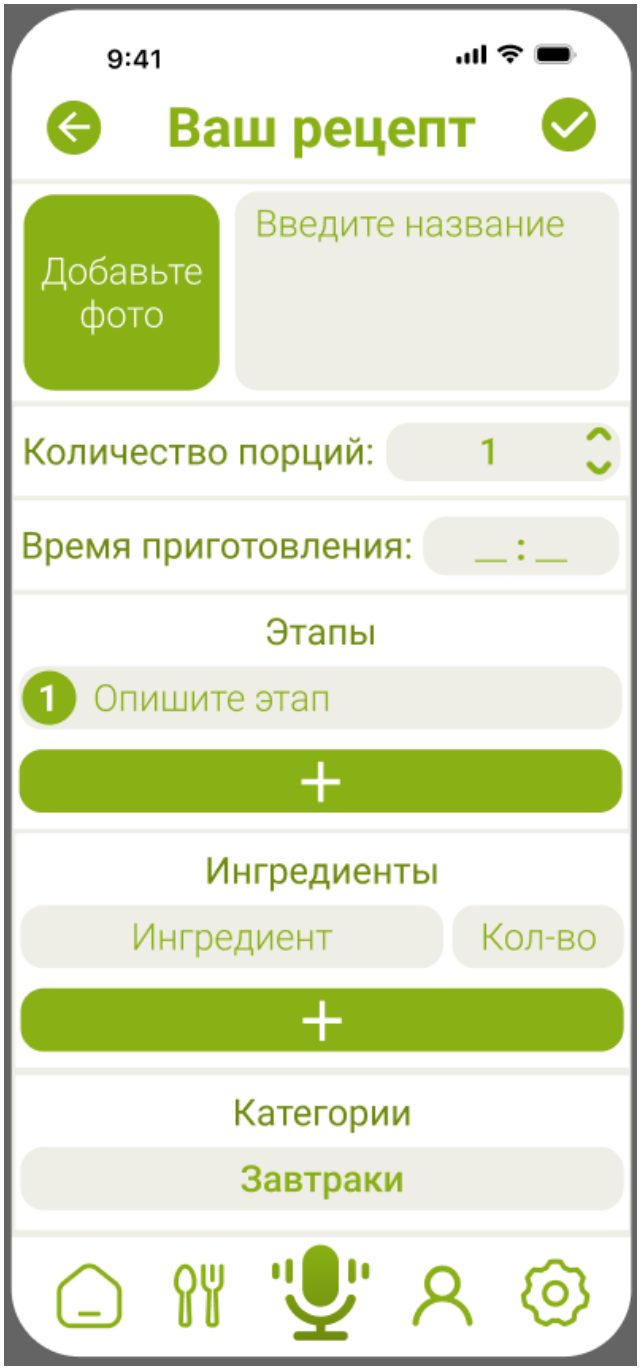

Рисунок 6 - Экран создания рецепта

7 Проработка API

7.1 Авторизация

— POST /auth/register — регистрация пользователя.

— POST /auth/login — вход в систему.

7.2 Рецепты

— GET /recipes — получить список рецептов.

— POST /recipes — добавить новый рецепт.

— PUT /recipes/{id} — обновить рецепт.

— DELETE /recipes/{id} — удалить рецепт.

7.3 Озвучивание рецептов

— POST /recipes/{id}/speak — запустить озвучивание.

— POST /recipes/{id}/pause — поставить на паузу.

— POST /recipes/{id}/resume — возобновить.

— POST /recipes/{id}/stop — остановить.

7.4 Ингредиенты

— GET /ingredients — получить список ингредиентов.

— POST /recipes/{id}/ingredients — добавить ингредиенты в рецепт.

7.5 Рекомендации

— GET /recommendations — получить список рекомендованных рецептов.

7.6 Управление профилем

— GET /profile — получить данные профиля.

— PUT /profile — обновить профиль.

— POST /auth/logout — выйти из системы.

8 Начальная архитектура

8.1 UML Диаграммы

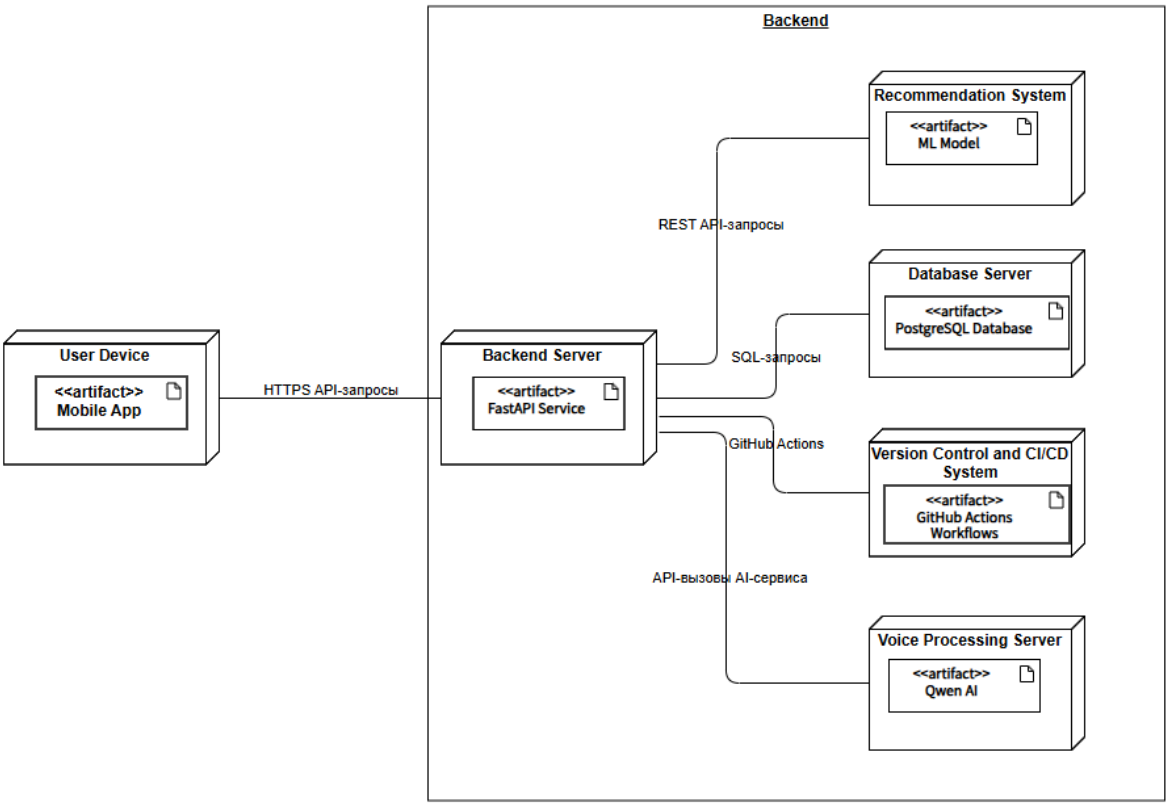

Рисунок 7 - Диаграмма развёртывания

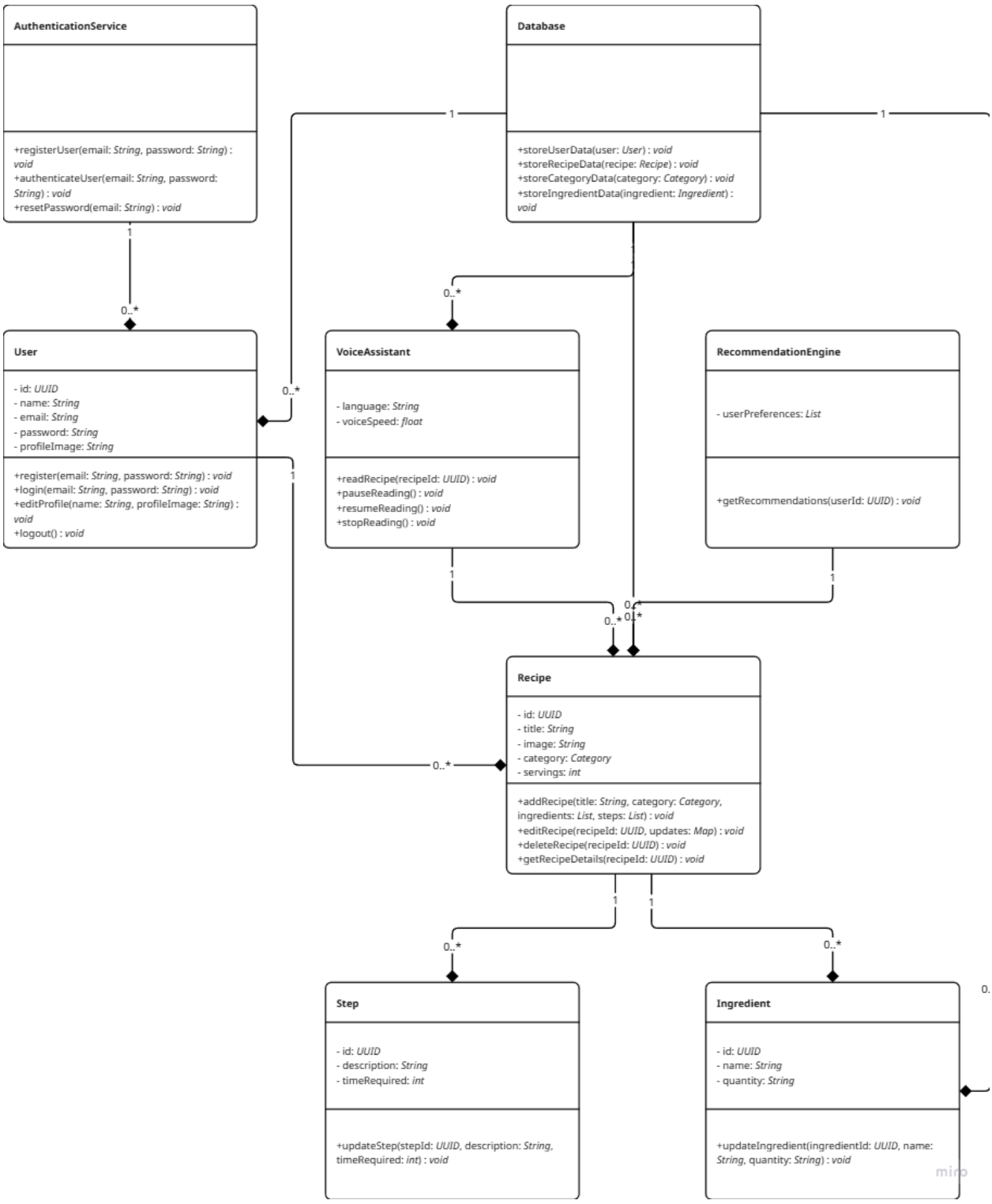

Рисунок 8 - Диаграмма классов

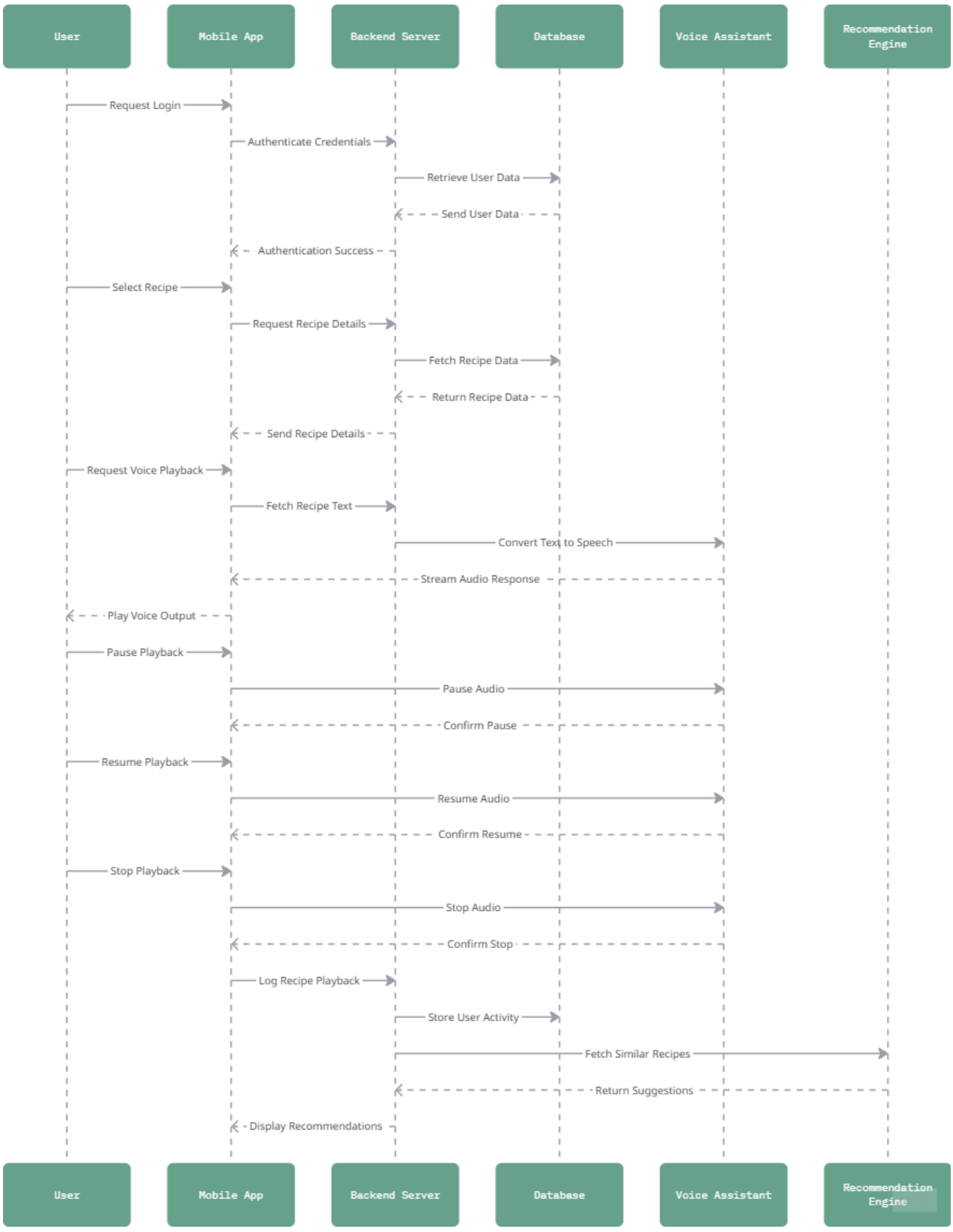

Рисунок 9 - Диаграмма последовательностей

8.2 ER Диаграмма

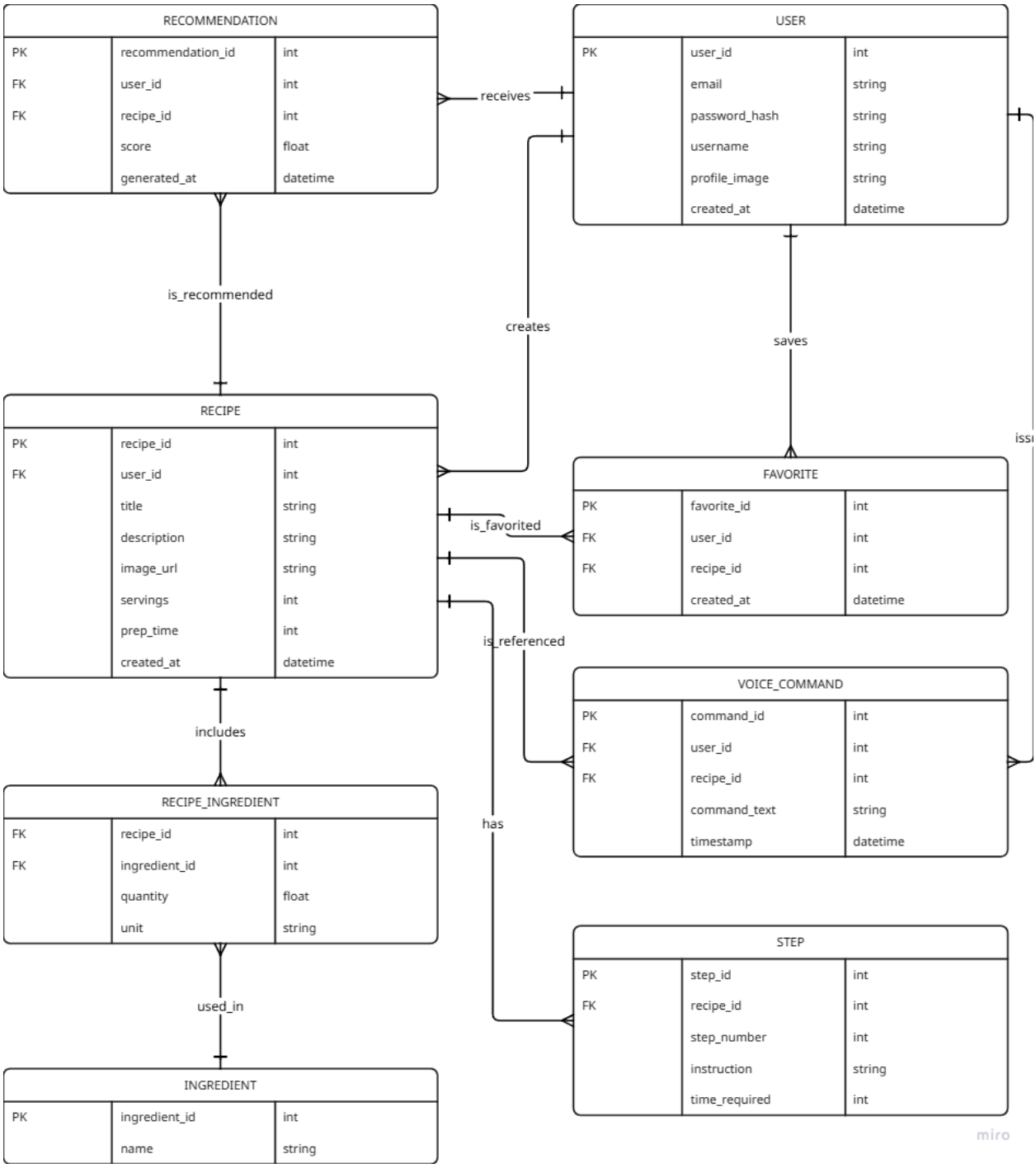

Рисунок 10 - ER Диаграмма