

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В. Непомнящий
" ___ " _____ 2023 г.

БАКАЛАВРСКАЯ РАБОТА

090301 Информатика и вычислительная техника

Мобильное приложение «Fody»

Руководитель	_____	_____	доцент, канд. техн.наук.	С.Н. Титовский
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____		Н.Е. Евдокимов
	<i>подпись</i>	<i>дата</i>		
Нормоконтролёр	_____	_____		С.Н. Титовский
	<i>подпись</i>	<i>дата</i>		

Красноярск 2023

СОДЕРЖАНИЕ

Введение.....	4
1 Анализ предметной области	5
1.1 Постановка задачи.....	5
1.2 Обзор аналогов	5
1.2.1 Мобильное приложение «RecipeBook»	6
1.2.2 Мобильное приложение «Что готовим?»	7
1.2.3 Веб-сайт «ProCook»	9
1.3 Сравнительный анализ средств разработки мобильных приложений	10
1.3.1 Android Studio	10
1.3.2 Appcelerator	10
1.3.3 Ionic.....	11
1.4 Разработка интерфейса	11
1.5 Вывод по главе	13
2 Проектирование приложения.....	14
2.1 Разработка структуры базы данных	14
2.2 Структура приложения	15
2.3 Диаграмма вариантов использования	17
2.4 Структура проекта NestJS	18
2.5 Модуль авторизации	19
2.6 Модуль регистрации	20
2.7 Модуль личного кабинета пользователя	21
2.8 Модуль рецепта пользователя	21
2.9 Модуль списка рецептов пользователя.....	22
2.10 Модуль ингредиентов	22
2.11 Модуль рецепта	23
2.12 Модуль списка всех рецептов.....	23
2.13 Модуль списка найденных рецептов	23
2.14 Подмодуль создания рецепта.....	24

2.15 Подмодуль редактирования рецепта.....	24
2.16 Взаимодействие модулей	24
2.17 URL-адреса	25
2.18 Вывод по главе	26
3 Реализация.....	27
3.1 Выбор инструментов	27
3.1.1 Серверная часть.....	27
3.1.2 Клиентская часть.....	27
3.1.3 Система управления базами данных.....	27
3.1.4 ORM.....	28
3.2 Реализация серверной части	28
3.3 Реализация клиентской части	29
3.4 Описание работы мобильного приложения	30
3.4.1 Регистрация.....	30
3.4.2 Авторизация.....	31
3.4.3 Профиль пользователя.....	32
3.4.4 Список ингредиентов.....	33
3.4.5 Список рецептов пользователя	34
3.4.6 Список рецептов.....	35
3.4.7 Список найденных рецептов.....	36
3.4.8 Рецепт пользователя	37
3.4.9 Рецепт	38
3.4.10 Обновление и создание рецепта	39
3.5 Установка и запуск.....	Ошибка! Закладка не определена.
3.6 Вывод по главе	Ошибка! Закладка не определена.
Заключение	41
Список использованных источников	42

ВВЕДЕНИЕ

Множество людей сталкивались с ситуацией, в которой, имея большое количество продуктов, они не знали, что можно приготовить из них. Существуют специальные веб-сайты и мобильные приложения, которые помогают пользователям находить рецепты, используя продукты, которые уже есть у них дома. Мобильное приложение – это очень удобно, так как каждый день из рук мы не выпускаем именно телефон. Таким образом, мобильное приложение для поиска рецептов будет куда удобнее и практичнее для повседневного использования.

Такое приложение удобно, потому что оно позволяет пользователям быстро и легко находить рецепты, используя продукты, которые уже есть у них дома. Это экономит время и деньги, так как не нужно ходить в магазин за дополнительными продуктами, а также заказывать доставку еды. Такие сервисы помогают пользователям избежать выбрасывания еды, так как они предлагают рецепты, используя продукты, которые уже есть у пользователя в холодильнике. Это позволяет использовать продукты, прежде чем они испортятся, и таким образом сократить потери и избежать выбрасывания еды. Таким образом, такое приложение не только удобно и практично, но и способствует бережному отношению к ресурсам и окружающей среде.

1 Анализ предметной области

1.1 Постановка задачи

Цель работы – создание мобильного приложения «Fody». Основной целью разработки приложения является облегчение поиска блюда и его рецепта по имеющимся ингредиентам.

Разрабатываемое мобильное приложение должно соответствовать определенным требованиям и обладать рядом ключевых характеристик:

1) мобильное приложение должно быть интуитивно понятным и удобным;

2) у пользователя должен быть аккаунт, в котором он сможет посмотреть созданные им рецепты;

3) мобильное приложение должно предоставлять пользователю список рецептов по выбранным им ингредиентам, а также список всех существующих рецептов.

Для достижения цели работы должны быть выполнены следующие задачи:

- осуществить постановку задачи, выделить требования к приложению;
- произвести обзор существующих мобильных приложений;
- провести сравнительный анализ современных методов разработки мобильных приложений;
- определить требования и спроектировать мобильное приложение;
- реализовать мобильное приложение.

1.2 Обзор аналогов

На рынке существует несколько приложений, которые предлагают похожие услуги по поиску рецептов, используя имеющиеся у пользователя продукты. Хотя концепция этих приложений может быть схожей, каждое из

них имеет свои уникальные особенности и функции. Некоторые приложения могут предлагать более широкий выбор рецептов или более продвинутые функции поиска, в то время как другие могут иметь более простой и удобный интерфейс.

1.2.1 Мобильное приложение «RecipeBook»

Приложение с уникальной функцией под названием «Snap'n'Cook» («Потряси и готовь»). Данная функция позволяет пользователям выбрать продукты, которые имеются у них в холодильнике, и отметить их в поиске, а затем встряхнуть телефон. После этого приложение предложит рецепты, которые можно приготовить из выбранных продуктов [1] (Рисунок 1).

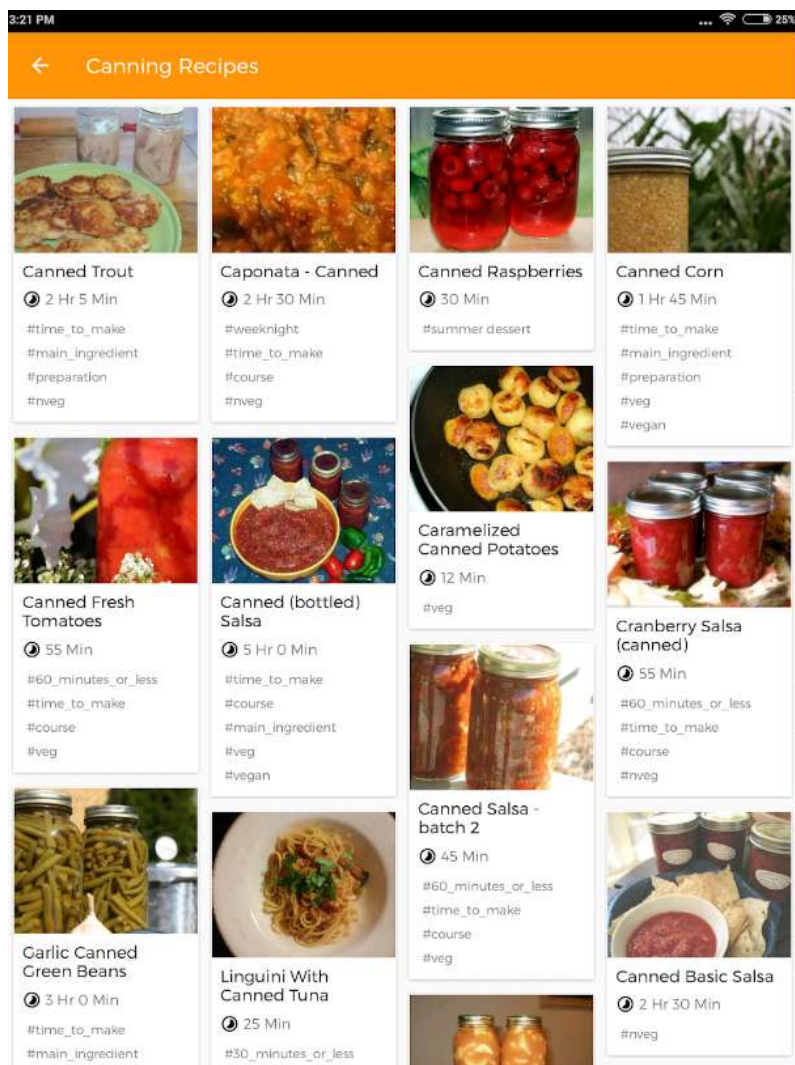


Рисунок 1 – Интерфейс мобильного приложения «RecipeBook»

Плюсы и минусы: приложение популярно за рубежом, но в нашей стране совершенно не пользуется спросом. Локализация приложения оставляет желать лучшего, а значит многие рецепты могут быть непонятными для пользователей из нашей страны.

1.2.2 Мобильное приложение «Что готовим?»

На стартовом экране приложения есть значок в виде холодильника. При нажатии на него открывается список продуктов, в котором пользователь должен выбрать имеющиеся продукты галочками. Затем достаточно нажать кнопку «Сформировать рецепты», и приложение предложит блюда, которые

можно приготовить из выбранных продуктов. Это удобная и практичная функция, которая помогает пользователям быстро находить рецепты, используя имеющиеся у них продукты [1] (Рисунок 2).

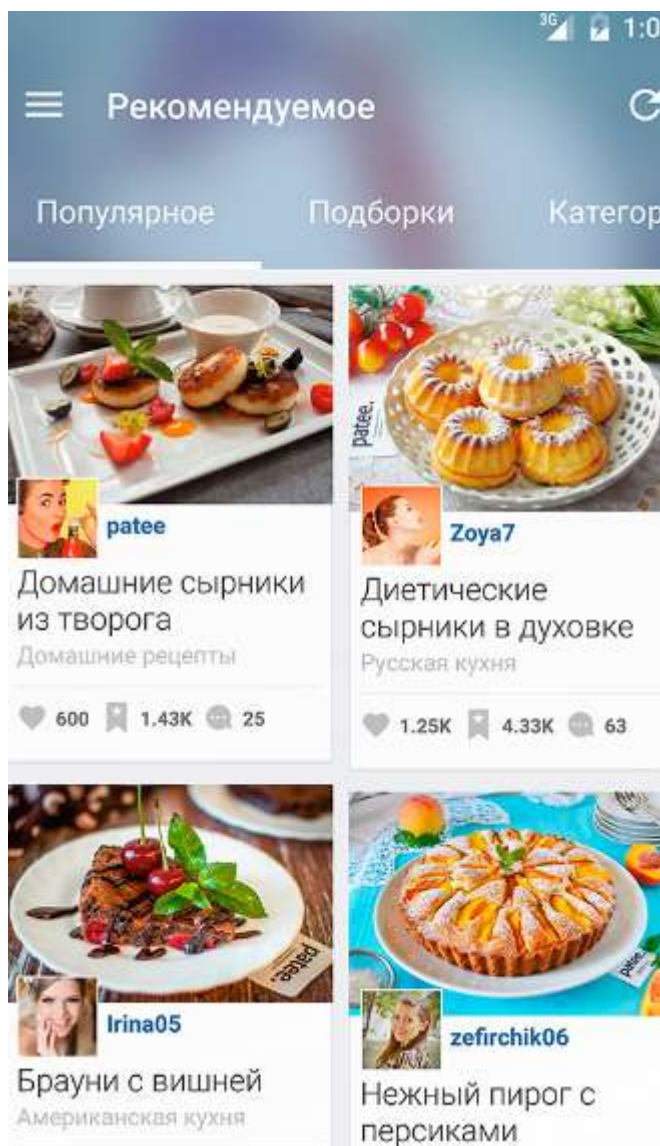


Рисунок 2 – Интерфейс мобильного приложения «Что готовим?»

Плюсы и минусы: приложение имеет удобный и понятный интерфейс, но количество доступных рецептов в нем ограничено. Также в приложении присутствует реклама и возможность оформления премиум-подписки, что может оттолкнуть большое количество пользователей. Приложение может быть полезным для тех, кто ищет простые и быстрые рецепты для повседневного использования.

1.2.3 Веб-сайт «ProCook»

На этом сайте представлено множество разделов, которые позволяют пользователям выбирать рецепты по различным критериям. Один из разделов посвящен выбору ингредиентов, что позволяет находить рецепты, используя продукты, которые уже есть у пользователя дома. Кроме того, на сайте есть раздел с тостами к праздникам и многое другое [1] (Рисунок 3).

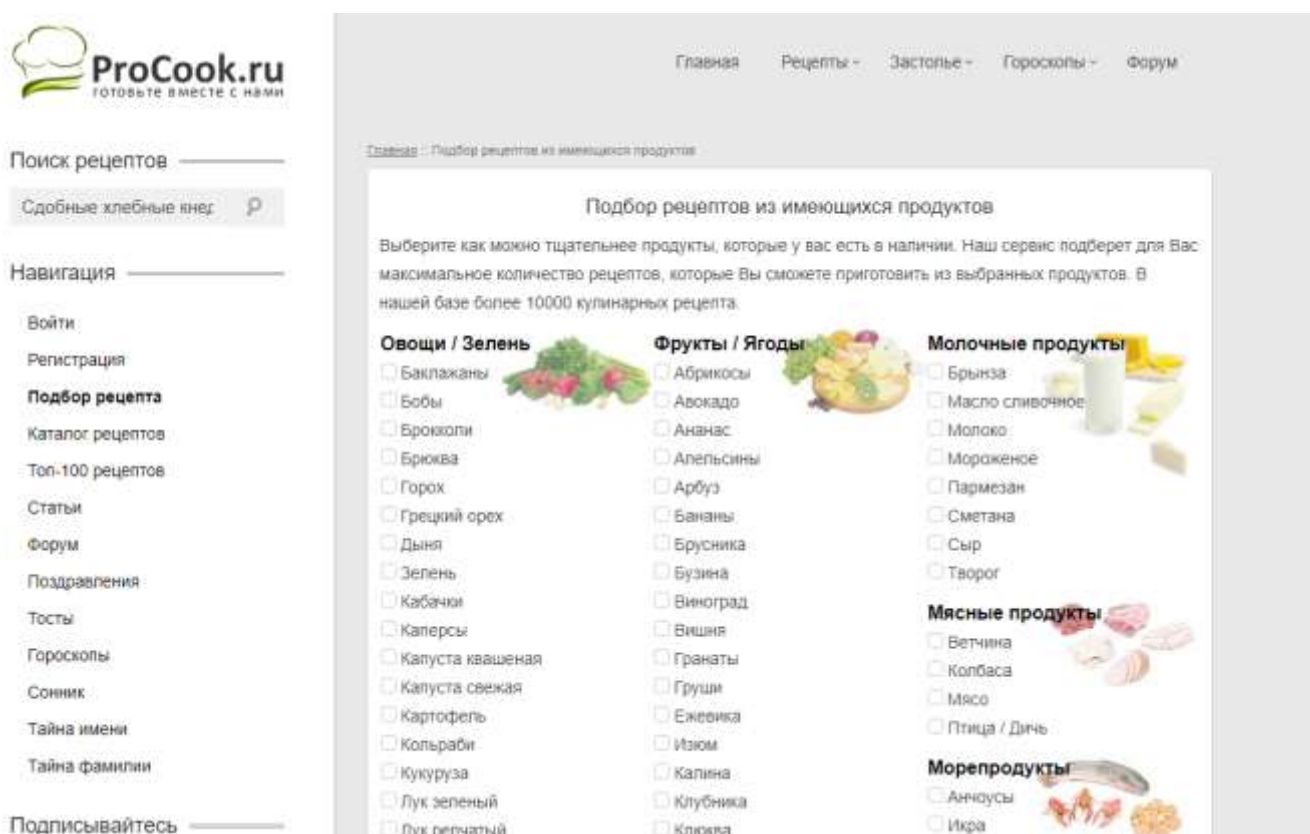


Рисунок 3 – Интерфейс веб-сайта «ProCook»

Плюсы и минусы: на сайте реализован удобный выбор ингредиентов, который позволяет пользователям быстро находить рецепты. Однако из-за большого количества разделов пользователь может запутаться и потратить много времени на поиск нужного раздела. Кроме того, сайт перестал обновляться уже большое количество времени, что может снизить его актуальность и полезность для пользователей.

Рассмотрев существующие аналоги, можно смело сказать, что ни одно из данных приложений и web-сайтов не подходит для целевой аудитории.

1.3 Сравнительный анализ средств разработки мобильных приложений

1.3.1 Android Studio

Android Studio – это официальная интегрированная среда разработки (IDE) для создания нативных приложений для Android от Google. Она включает в себя множество инструментов и функций, которые помогают разработчикам создавать качественные приложения для Android [2].

Особенности:

- редактор кода с подсветкой синтаксиса, автодополнением и другими функциями, которые упрощают написание кода;
- мощный эмулятор Android, который позволяет тестировать приложения на различных устройствах и версиях Android;
- поддержка отладки и профилирования приложений для выявления и исправления ошибок и улучшения производительности.

1.3.2 Appcelerator

Appcelerator позволяет разработчикам создавать приложения с меньшим числом строк кода. Этот инструмент разработки приложений поддерживает приложения для iOS, Android, Windows и браузеров [2].

Особенности:

- повышает скорость разработки мобильных приложений;
- расширяет пределы емкости “облачных” хранилищ;
- позволяет создавать мобильные приложения для всех поддерживаемых операционных систем;

– создан на основе открытых стандартов и полного доступа к базовой мобильной ОС.

1.3.3 Ionic

Ionic — это среда разработки мобильных приложений. Он широко используется для разработки гибридных мобильных приложений. Это полезный инструмент для создания мобильных приложений с использованием веб-технологий, таких как CSS, HTML5 и SASS [2].

Особенности:

- на 100% бесплатный проект с открытым исходным кодом;
- позволяет создавать прогрессивные веб-приложения и собственные мобильные приложения для каждого крупного магазина приложений;
- предлагает отличные инструменты и сервисы для создания интерактивных приложений;
- поставляется со многими компонентами CSS и JavaScript для разработки мобильных приложений.

1.4 Разработка интерфейса

Концепт интерфейса мобильного приложения (Рисунок 4).



Рисунок 4 – Концепт интерфейса мобильного приложения

При первом запуске мобильного приложения пользователь попадает на главный экран, где находится весь список ингредиентов, при выборе которых пользователь попадет на страницу со списком с соответствующими рецептами.

При нажатии на кнопку «Профиль» авторизованный пользователь может посмотреть данные о себе и выйти из своего аккаунта.

После успешной авторизации пользователь может посмотреть, удалить или редактировать созданные им рецепты.

При нажатии на кнопку «Рецепты» пользователю открывается страница со всеми доступными рецептами.

1.5 Вывод по главе

В этой главе были определены цели и задачи, которые необходимо выполнить при реализации выпускной квалификационной работы. Проведенный обзор аналогов позволил определить требования к функционалу и особенностям разрабатываемого мобильного приложения. Сравнение существующих средств разработки мобильных приложений помогло определить наиболее подходящие инструменты и технологии для создания мобильного приложения в соответствии с требованиями и целями работы. Также разработан концепт интерфейса и описаны основные возможности приложения.

2 Проектирование приложения

2.1 Разработка структуры базы данных

В результате анализа предметной определены следующие сущности: users, recipe, connection, ingredient.

В результате нормализации определены атрибуты сущностей (Таблица 1).

Таблица 1 – Атрибуты сущностей

Сущность			
Атрибут	Назначение	Тип	Ограничения
Users			
Id	Идентификатор	Integer	Primary key
Username	Имя пользователя	Text	Not null
Password	Пароль пользователя	Text	Not null
Recipe			
Id	Идентификатор	Integer	Primary key
Name	Название рецепта	Text	Not null
Title	Описание рецепта	Text	Not null
UserId	Идентификатор пользователя, создавшего рецепт	Integer	Foreign key
Ingredient			
Id	Идентификатор	integer	Primary key
Name	Название ингредиента	Text	Not null
Connection			
Id	Идентификатор	Integer	Primary key
IdRecipe	Идентификатор рецепта	Integer	Foreign key
InIngredient	Идентификатор ингредиента	Integer	Foreign key

Структура базы данных была разработана с использованием логической модели базы данных (Рисунок 5).

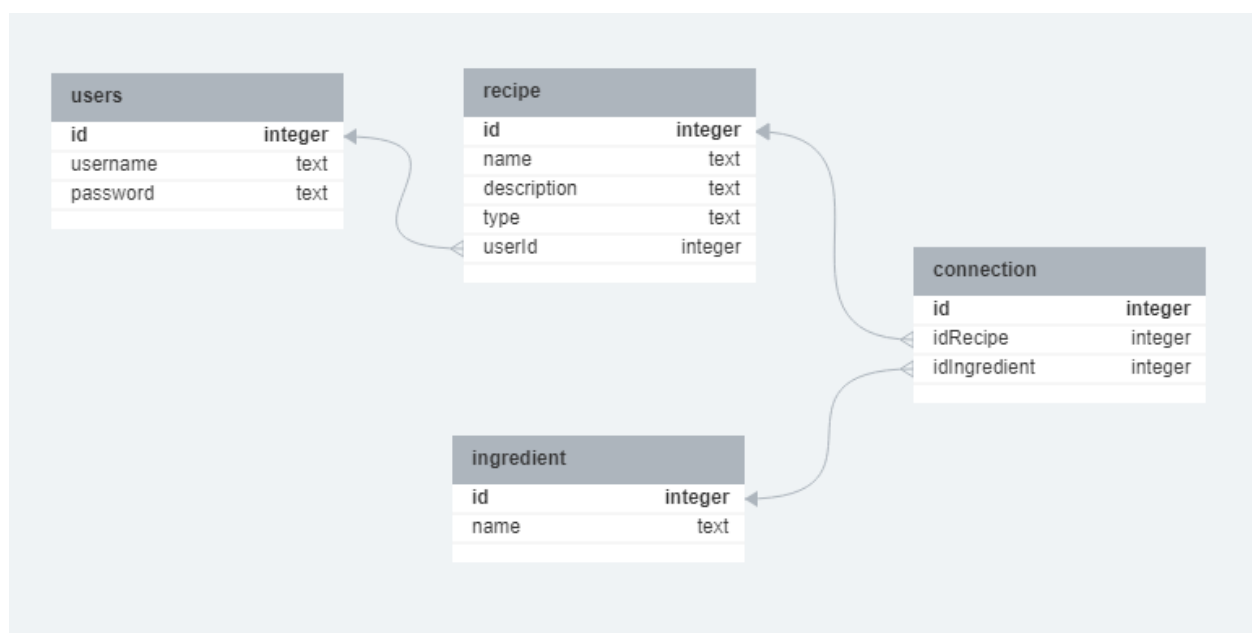


Рисунок 5 – Логическая модель базы данных

Была создана связующая таблица между таблицами Рецепты и Ингредиенты, чтобы установить отношение «многие ко многим» между ними. Это означает, что один рецепт может содержать несколько ингредиентов, а один ингредиент может использоваться в нескольких рецептах.

Связующая таблица содержит столбцы-ссылки на первичные ключи каждой из связываемых таблиц и позволяет связать между собой несколько записей из одной таблицы с несколькими записями из другой таблицы.

Использование связующей таблицы позволяет эффективно управлять данными о рецептах и ингредиентах и обеспечивает гибкость при построении запросов и анализе данных. Это очень удобно для поиска всех рецептов, которые содержат определенные ингредиенты, или все ингредиенты, которые используются в определенном рецепте [7].

2.2 Структура приложения

Структура приложения определяет, как организованы и взаимодействуют между собой различные компоненты приложения. Она позволяет

разработчикам четко определить ответственность каждого компонента и обеспечить эффективное взаимодействие между ними.

Использование хорошо продуманной структуры приложения имеет множество преимуществ:

– упрощает разработку: четкое разделение ответственности между компонентами приложения упрощает разработку и тестирование каждого компонента отдельно;

– улучшает производительность: хорошо спроектированная структура приложения может обеспечить эффективное использование ресурсов и улучшить производительность приложения;

– облегчает сопровождение: четкая структура приложения облегчает понимание кода и упрощает его сопровождение и дальнейшее развитие.

Структура мобильного приложения (Рисунок 6).

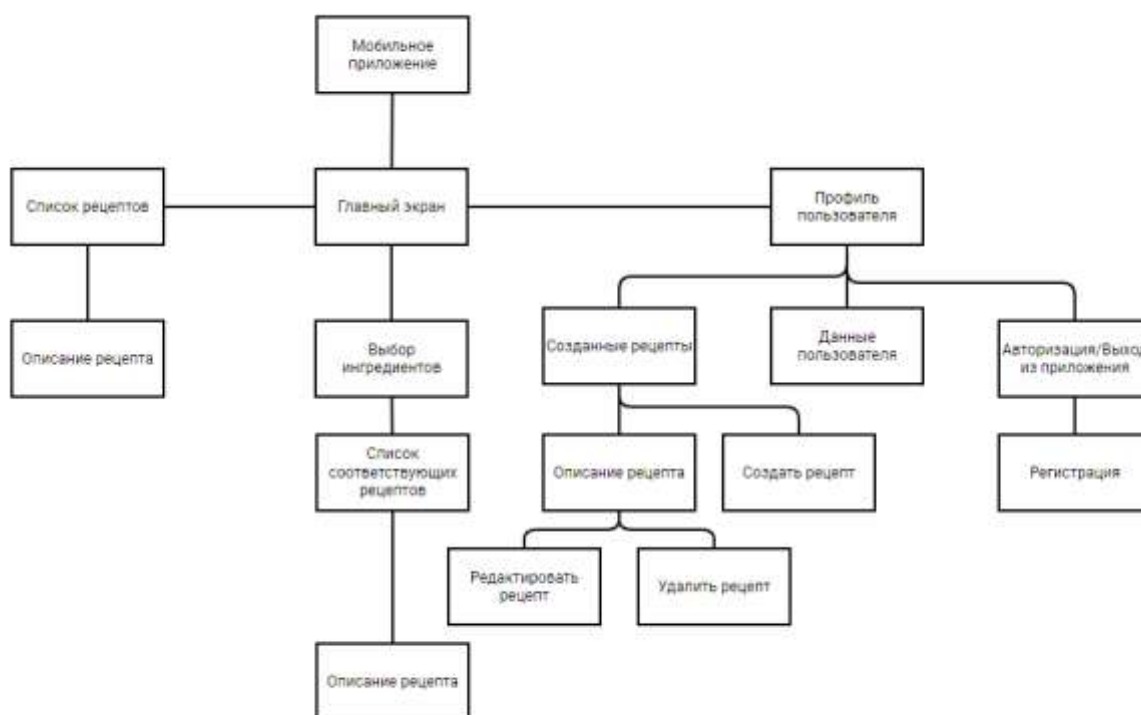


Рисунок 6 – Структура разрабатываемого мобильного приложения

2.3 Диаграмма вариантов использования

Чтобы лучше понять необходимый функционал приложения и его компоненты, можно построить диаграмму вариантов использования. Приложение должно предоставлять пользователю такие возможности, как регистрация нового пользователя, вход в систему, просмотр списка созданных им рецептов, поиск рецептов по ингредиентам, фильтрация рецептов по типу блюда и названию, просмотр информации о рецептах, а также создание, удаление и редактирование рецептов.

Диаграмма вариантов использования (Рисунок 7).

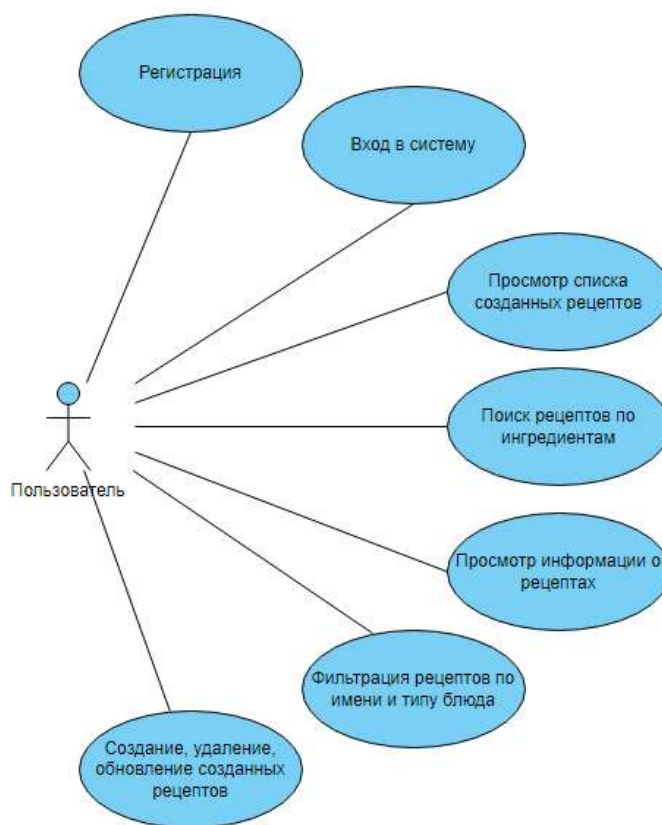


Рисунок 7 – Диаграмма вариантов использования объекта «Пользователь»

2.4 Структура проекта NestJS

Для обеспечения четкой организации кода и упрощения его расширения и поддержки в работе применяется модульная структура проекта. Приложение разбито на отдельные модули, каждый из которых отвечает за определенную функциональность. Модульная структура [3] (Рисунок 8).

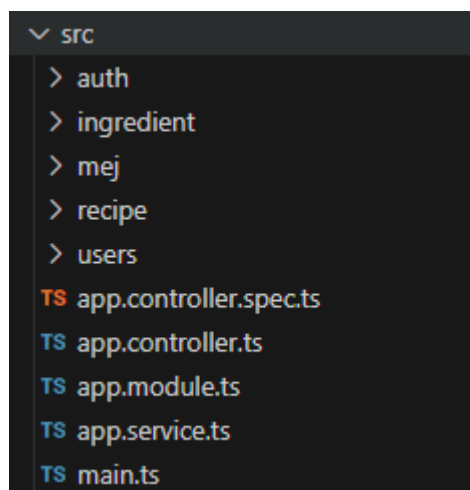


Рисунок 8 – Структура NestJS

Контроллеры в NestJS отвечают за обработку входящих запросов (Request) и возврат ответов (Response) клиенту. Назначение контроллера – принимать запросы для приложения.

Корневой модуль является отправной точкой, которую Nest использует для построения графа приложения – внутренней структуры данных, которую Nest использует для разрешения отношений и зависимостей между модулями.

Сервис содержит методы для выполнения определенных задач, таких как работа с базой данных или выполнение сложных вычислений. Сервисы могут быть внедрены в другие классы, такие как контроллеры, чтобы использовать их функциональность.

Файл «main.ts» содержит асинхронную функцию, которая выполняет начальную загрузку приложения [9].

2.5 Модуль авторизации

Модуль авторизации является важным компонентом приложения, который обеспечивает безопасность и конфиденциальность информации пользователя. Для входа в аккаунт пользователь должен ввести свои учетные данные, включая логин и пароль. После этого приложение отправляет запрос на сервер для проверки достоверности предоставленной информации. В случае успешной проверки пользователь получает доступ к некоторым функциям приложения. В случае неправильного ввода учетных данных пользователю будет предложено повторить попытку.

Диаграмма последовательности авторизации (Рисунок 9).

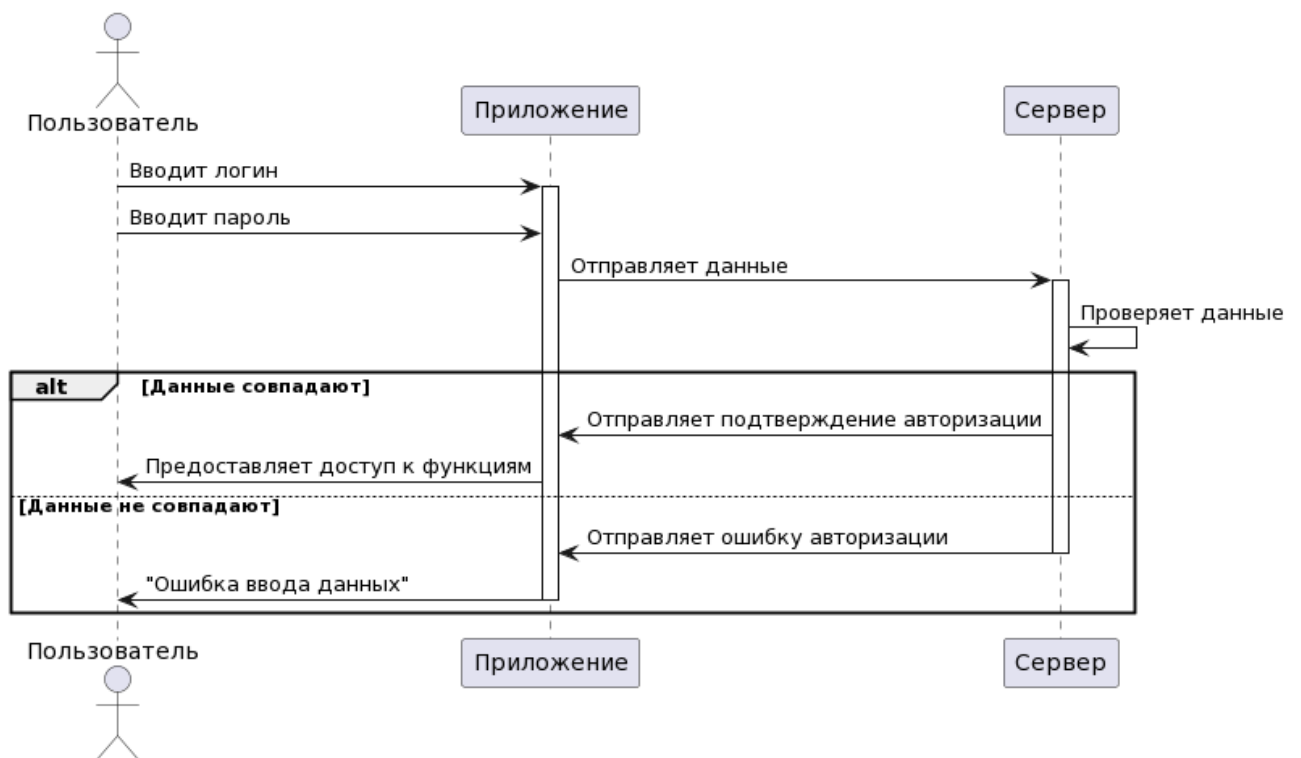


Рисунок 9 – Диаграмма последовательности авторизации

Данный модуль содержит следующие методы:

- login() – проверяет правильность введенных данных;
- findByUserName() – возвращает данные пользователя с указанным ЛОГИНОМ;

– passwordComparison() – возвращает логическое значение о результате сравнения паролей.

2.6 Модуль регистрации

Модуль регистрации отвечает за создание нового аккаунта пользователя. Для того, чтобы воспользоваться некоторыми функциями приложения, пользователю необходимо пройти процесс регистрации. Пользователь должен придумать логин и пароль. Если логин не занят и пароль соответствует правилам, то данные нового аккаунта появятся в базе данных, и пользователь сможет войти в него. Этот процесс обеспечивает безопасность и конфиденциальность информации пользователя, предотвращая несанкционированный доступ к личным данным.

Диаграмма последовательности регистрации (Рисунок 10).

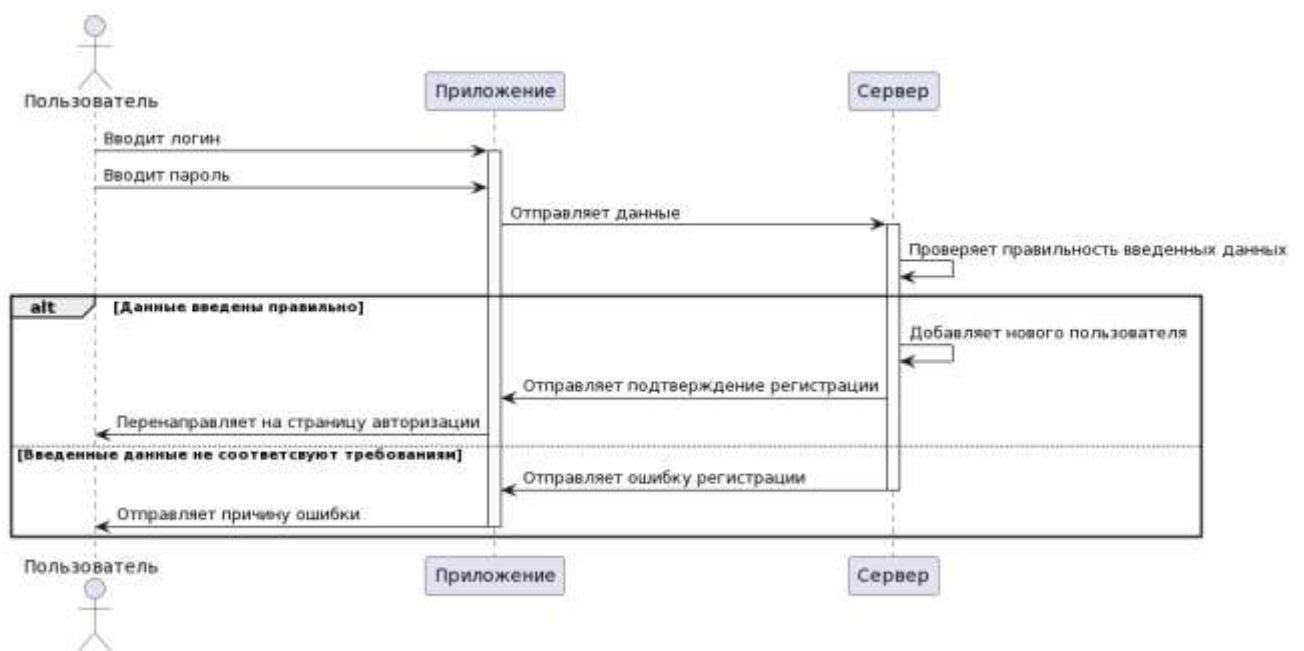


Рисунок 10 – Диаграмма последовательности регистрации

Данный модуль содержит следующие методы:

– register() – если логин не занят, то создать нового пользователя;

- findByUserName() – найти пользователя по логину;
- createUser() – сохранить данные пользователя;
- hashPassword() – шифрование пароль.

2.7 Модуль личного кабинета пользователя

При успешной авторизации в модуле предоставляется возможность перейти на страницу созданных рецептов, выйти из учетной записи, а также увидеть информацию о пользователе. Доступ к этому модулю осуществляется через меню навигации, которое расположено в нижней части мобильного приложения.

Данный модуль содержит следующие методы:

- getById() – получить данные пользователя с указанным id;
- check() – получить данные пользователя;
- logout() – выход из учетной записи.

2.8 Модуль рецепта пользователя

Данный модуль предназначен для удаления и обновления рецепта, а также отображения информации о рецепте пользователя. Каждый рецепт содержит название, ингредиенты, тип блюда и подробную инструкцию по приготовлению блюда.

Данный модуль содержит следующие методы:

- deleteRecipe() – удалить рецепт;
- getIngredientNames() – получить названия ингредиентов с указанными id;
- getByRecipeId() – получить данные о рецепте с указанным id;
- getByIds() – получить данные о ингредиентах с указанными id.

Взаимодействие с модулем рецептов пользователя (Рисунок 11).

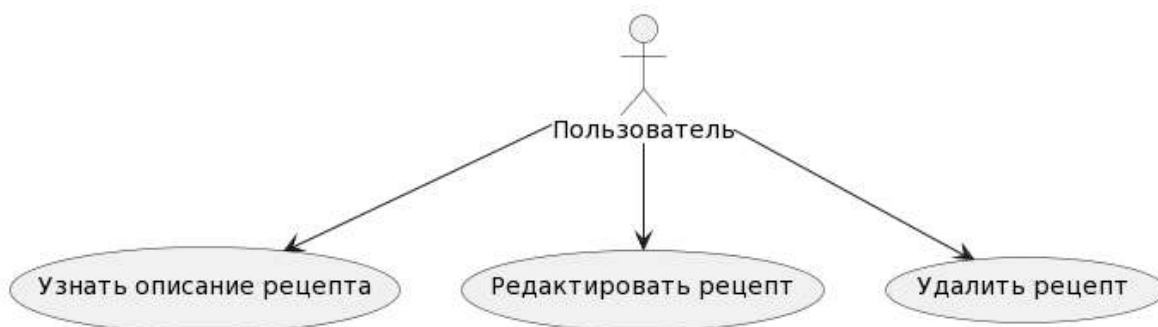


Рисунок 11 – Использование модуля рецепта пользователя

2.9 Модуль списка рецептов пользователя

Этот модуль предназначен для предоставления пользователю списка созданных им рецептов и обеспечения перехода на страницу создания нового рецепта.

Данный модуль содержит следующие методы:

– findRecipesByUser() – получить данные о рецептах пользователя.

2.10 Модуль ингредиентов

Данный модуль позволяет пользователю выбирать ингредиенты, которые он хочет использовать для поиска рецептов.

Диаграмма поиска рецептов по заданным ингредиентам (Рисунок 12).



Рисунок 12 – Диаграмма последовательности поиска рецептов

Данный модуль содержит следующие методы:

- getAllIngredient() – получить данные о всех ингредиентах;
- getByIds() – получить данные о ингредиентах с указанными id.

2.11 Модуль рецепта

Данный модуль предназначен для отображения информации о выбранном рецепте. Рецепт содержит название, ингредиенты, тип блюда и подробную инструкцию по приготовлению блюда.

Данный модуль содержит следующие методы:

- getIngredientNames() – получить названия ингредиентов с указанными id;
- getByRecipeId() – получить данные о рецепте с указанным id;
- getByIds() – возвращает данные о ингредиентах с указанными id.

2.12 Модуль списка всех рецептов

Данный модуль отвечает за отображение списка всех рецептов, а также за поиск по названию и фильтрации рецептов по типу блюда.

Данный модуль содержит следующие методы:

- getAllRecipe() – получить данные о всех рецептах.

2.13 Модуль списка найденных рецептов

Этот модуль предназначен для отображения списка рецептов, содержащих выбранные пользователем ингредиенты, и предоставления возможности фильтрации рецептов по типу блюда.

Данный модуль содержит следующие методы:

- getAllRecipe() – получить данные о всех рецептах.

2.14 Подмодуль создания рецепта

Подмодуль создания рецепта предоставляет пользователю возможность создать новый рецепт. Пользователь может указать: название рецепта, основную информацию о рецепте, список ингредиентов и тип блюда.

Данный модуль содержит следующие методы:

- createRecipe() – создать рецепт;
- getAllIngredient() – получить данные о всех ингредиентах.

2.15 Подмодуль редактирования рецепта

Подмодуль редактирования рецепта предоставляет пользователю возможность изменять информацию о существующих рецептах. В рамках этого подмодуля пользователь может редактировать название рецепта, изменять список ингредиентов и корректировать основную информацию о рецепте.

Данный модуль содержит следующие методы:

- updateRecipe() – обновить данные о рецепте;
- getAllIngredient() – получить данные о всех ингредиентах.

2.16 Взаимодействие модулей

Для того, чтобы получить более глубокое понимание взаимодействия между модулями в системе, можно создать схему, отображающую связи между ними. Это позволит визуализировать структуру системы. Такой подход помогает лучше понять архитектуру системы и обнаружить возможные узкие места или проблемы в ее работе.

Схема взаимодействия между модулями (Рисунок 13).

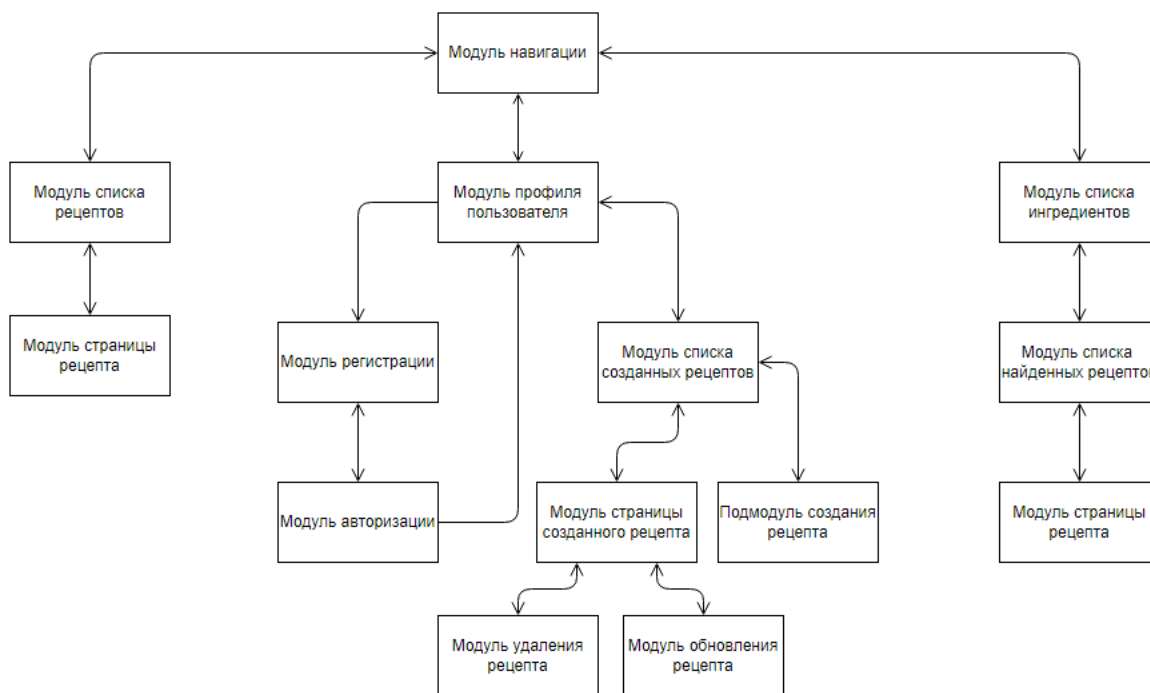


Рисунок 13 – Схема взаимодействия модулей

2.17 URL-адреса

Единый указатель ресурса (Uniform Resource Locator) – это стандартный формат адреса веб-страницы или другого ресурса в Интернете.

Выбраны следующие URL-адреса:

- Get /users/:id – получить информацию о пользователе;
- Delete /users/delete/:id – удалить пользователя;
- Post /auth/register – регистрация пользователя;
- Post /auth/login – авторизация пользователя;
- Post /auth/logout – выход из учетной записи;
- Post /auth/check – получить информацию о текущем пользователе;
- Get /recipe/all – получить список всех рецептов;
- Get /recipe/myrecipe – получить список рецептов пользователя;
- Get /recipe/found – получить список рецептов по выбранным ингредиентам;

- Get /recipe/:id/ingredients – получить список ингредиентов в рецепте;
- Get /recipe/:id – получить информацию о рецепте;
- Post /recipe/create – создать новый рецепт;
- Put /recipe/update/:id – обновить информацию о рецепте;
- Delete /recipe/delete/:id – удалить рецепт;
- Get /ingredient/all – получить список всех ингредиентов;
- Get /ingredient/names – получить имена ингредиентов;
- Get /ingredient/:id – получение информации о ингредиенте.

2.18 Вывод по главе

Во второй главе разработана структура базы данных с помощью логической модели. Для упрощения процесса разработки была описана структура мобильного приложения и диаграмма вариантов использования пользователем. Обозначен функционал для каждого модуля. Рассмотрено взаимодействие модулей. Выбраны URL-адреса.

3 Реализация

3.1 Выбор инструментов

3.1.1 Серверная часть

Для серверной части наиболее целесообразно использовать NestJS. NestJS – это фреймворк для создания масштабируемых, быстрых и легко тестируемых веб-приложений [6].

Одно из преимуществ NestJS – это его модульная структура, которая упрощает разделение проекта на отдельные блоки. Он также содержит встроенный контейнер DI (Dependency Injection), который используется для создания более эффективных и модульных приложений [10].

3.1.2 Клиентская часть

Для реализации клиентской части приложения был выбран React Native. React Native использует технологию модульного программирования, в которой функции реализованы в виде отдельных блоков, называемых модулями. Этот подход обеспечивает гибкую среду разработки приложений, а также улучшает взаимодействие между разработчиками. Сейчас он очень популярен, а значит найти решение проблемы будет не таким сложным [8].

3.1.3 Система управления базами данных

SQLite – это компактная встраиваемая СУБД, которая имеет множество преимуществ для использования в мобильных приложениях. Она широко доступна, стабильна и надежна. Большинство платформ используют SQLite для локального хранилища. Базы данных SQLite хранятся в одном файле на диске, что делает их портативными и эффективным форматом обмена данными [5].

3.1.4 ORM

В качестве ORM был выбран TypeORM. TypeORM – это простой в использовании ORM для создания новых приложений, подключающихся к базам данных [4].

3.2 Реализация серверной части

Реализация серверной части (Рисунок 14).

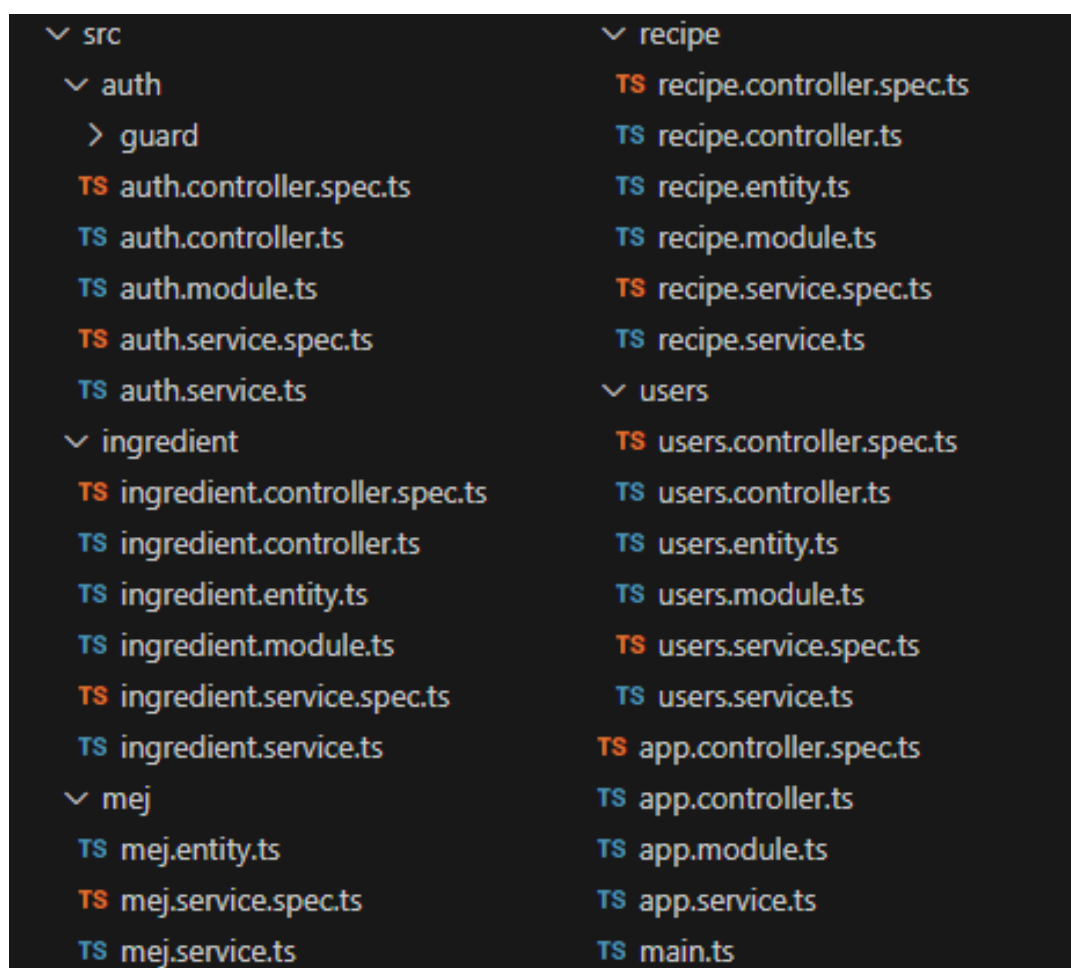


Рисунок 14 – Серверная часть

Описание контроллеров (Таблица 2).

Таблица 2 – Контроллеры

Контроллер	Назначение
auth.controller	Обработка входящих запросов и формирование ответов при регистрации, авторизации, проверки данных пользователя, выхода из учетной записи.
ingredient.controller	Обработка входящих запросов и формирование ответов для объекта «ingredient».
recipe.controller	Обработка входящих запросов и формирование ответов для объекта «recipe».
users.controller	Обработка входящих запросов и формирование ответов для объекта «users».
app.controller	Базовый контроллер с одним обработчиком маршрута.

3.3 Реализация клиентской части

Реализация клиентской части (Рисунок 15).

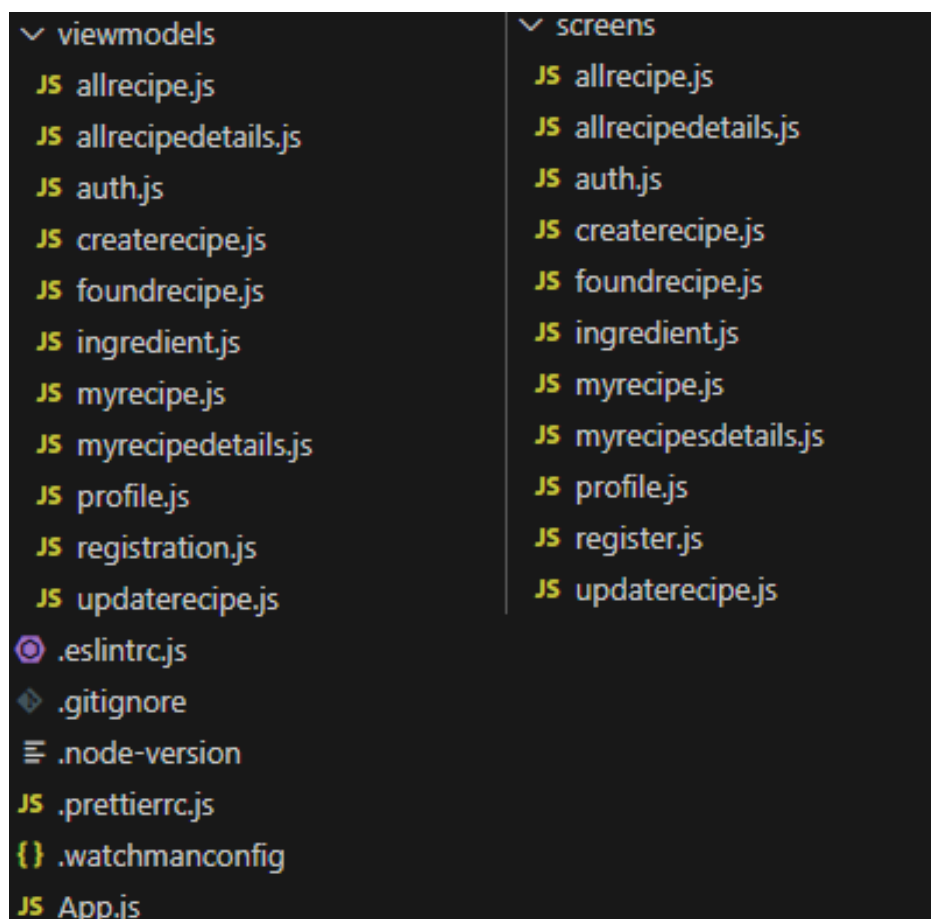


Рисунок 15 – Клиентская часть

ViewModels – это объекты, которые содержат данные и логику, связанные с отображением данных на экране. Они помогают отделить логику управления данными от компонентов пользовательского интерфейса, которые отвечают за отображение данных [12].

Screens – это компоненты, которые используются для определения отдельных экранов в мобильном приложении. Они используются в сочетании с навигационными библиотеками, чтобы управлять переходами между экранами и предоставлять пользователю возможность перемещаться по приложению [11].

3.4 Описание работы мобильного приложения

3.4.1 Регистрация

На странице регистрации пользователь может увидеть:

- поле «Имя пользователя»;
- поле «Пароль»: уникальная комбинация цифр и букв, предназначенная для подтверждения личности;
- кнопка для создания учетной записи;
- кнопка для перехода на страницу авторизации.

Интерфейс регистрации (Рисунок 16).

Имя пользователя

Пароль

Регистрация

Уже есть аккаунт? **Войти**

Рисунок 16 – Регистрация

3.4.2 Авторизация

На странице авторизации у пользователя также есть поля «Логин», «Пароль» и кнопка «Войти».

Интерфейс авторизации (Рисунок 17).

The image shows a simple authorization form. It consists of two stacked rectangular input fields with rounded corners and orange borders. The top field is labeled 'Имя пользователя' (Username) and the bottom field is labeled 'Пароль' (Password). Below these fields is a rounded rectangular button with an orange border and the text 'Войти' (Login) in orange.

Рисунок 17 – Авторизация

3.4.3 Профиль пользователя

На данной странице пользователь может:

- посмотреть личные данные;
- посмотреть созданные рецепты: после нажатия на кнопку «Посмотреть созданные рецепты» пользователь попадает на страницу со списком созданных им рецептов;
- выйти из учетной записи.

Интерфейс профиля пользователя (Рисунок 18).

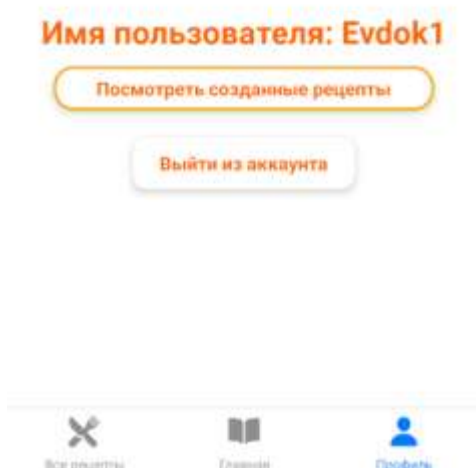


Рисунок 18 – Профиль пользователя

3.4.4 Список ингредиентов

Данная страница является основной в приложении, на ней пользователь может:

- выбрать нужные ингредиенты, которые представлены списком названий;
- найти рецепты с выбранными ингредиентами: после нажатия на кнопку «Найти рецепты» пользователь попадает на страницу со списком найденных рецептов.

Интерфейс списка ингредиентов (Рисунок 19).



Рисунок 19 – Страница ингредиентов

3.4.5 Список рецептов пользователя

На этой странице находится список рецептов, которые создал пользователь. После перехода на данную страницу пользователь может:

- найти рецепт вручную: если нажать на рецепт, то открывается отдельная страница с инструкцией по приготовлению;
- создать рецепт: пользователь переходит на страницу, где нужно ввести название рецепта, ингредиенты, тип блюда и подробную инструкцию о его приготовлении.

Интерфейс списка рецептов пользователя (Рисунок 20).

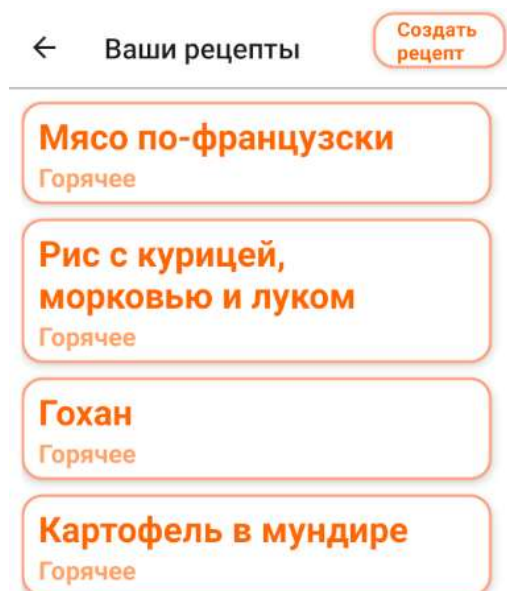


Рисунок 20 – Рецепты пользователя

3.4.6 Список рецептов

На странице находится список всех рецептов, которые внесены в базу данных. После перехода на данную страницу пользователь может:

- найти рецепт вручную: если нажать на рецепт, то открывается отдельная страница с инструкцией по приготовлению;
- найти рецепт по названию;
- отфильтровать рецепты по типу блюда.

Интерфейс списка всех рецептов (Рисунок 21).

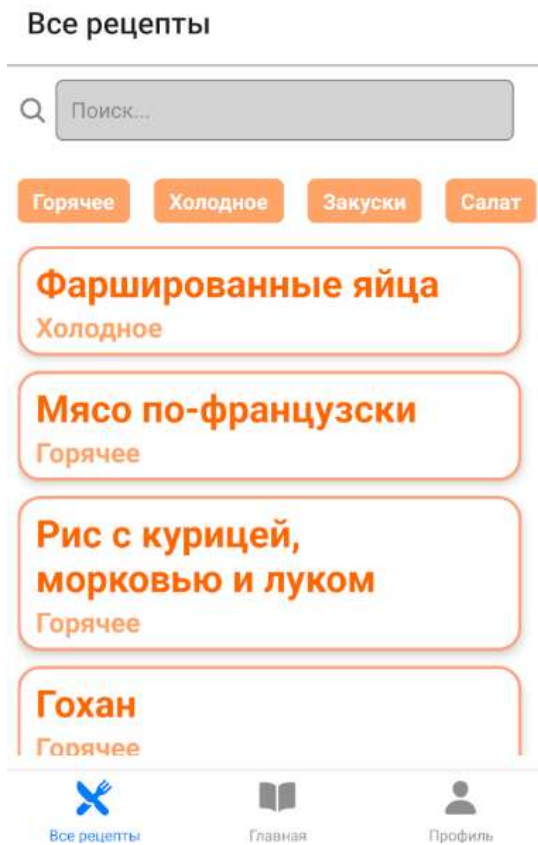


Рисунок 21 – Все рецепты

3.4.7 Список найденных рецептов

На этой странице находится список найденных рецептов, которые включают в себя указанные ингредиенты. После перехода на данную страницу пользователь может:

- найти рецепт вручную: если нажать на рецепт, то открывается отдельная страница с инструкцией по приготовлению;
- отфильтровать рецепты по типу блюда.

Интерфейс списка найденных рецептов (Рисунок 22).

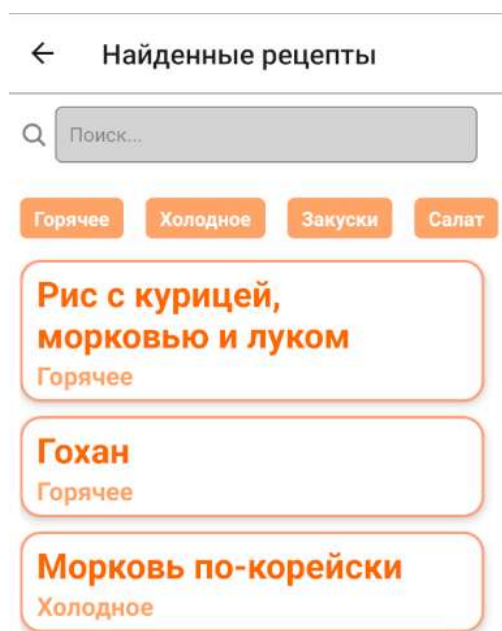


Рисунок 22 – Найденные рецепты

3.4.8 Рецепт пользователя

На странице находится название рецепта, тип блюда, ингредиенты, описание приготовления. Пользователь может:

- удалить рецепт;
- обновить рецепт: пользователь переходит на страницу обновления рецепта.

Интерфейс рецепта пользователя (Рисунок 23).

Удалить рецепт

Обновить рецепт

Мясо по-французски

Горячее

Картофель

Говядина

Сыр

Лук

Майонез

Шаг 1 Очистите картофель, вымойте его и нарежьте его кубиками средней величины. Мясо также нарежьте кубиками (2 см). Лук нарежьте кольцами.

Шаг 2 Выложите на противень мясо, равномерно распределяя по всей поверхности. Поперчите (0,25 ч. ложки), посолите (0,5 ч. ложки). Посыпьте частью нарезанного кольцами репчатого лука. После этого залейте мясо с луком 100 г

Рисунок 23 – Рецепт пользователя

3.4.9 Рецепт

На странице находится название рецепта, тип блюда, ингредиенты, описание приготовления.

Интерфейс рецепта (Рисунок 24).



Рисунок 24 – Рецепт

3.4.10 Обновление и создание рецепта

Эти страницы отвечают за управление данными рецепта:

- название рецепта;
- описание рецепта;
- ингредиенты рецепта;
- тип блюда.

Интерфейс обновления и создания рецепта (Рисунок 25).

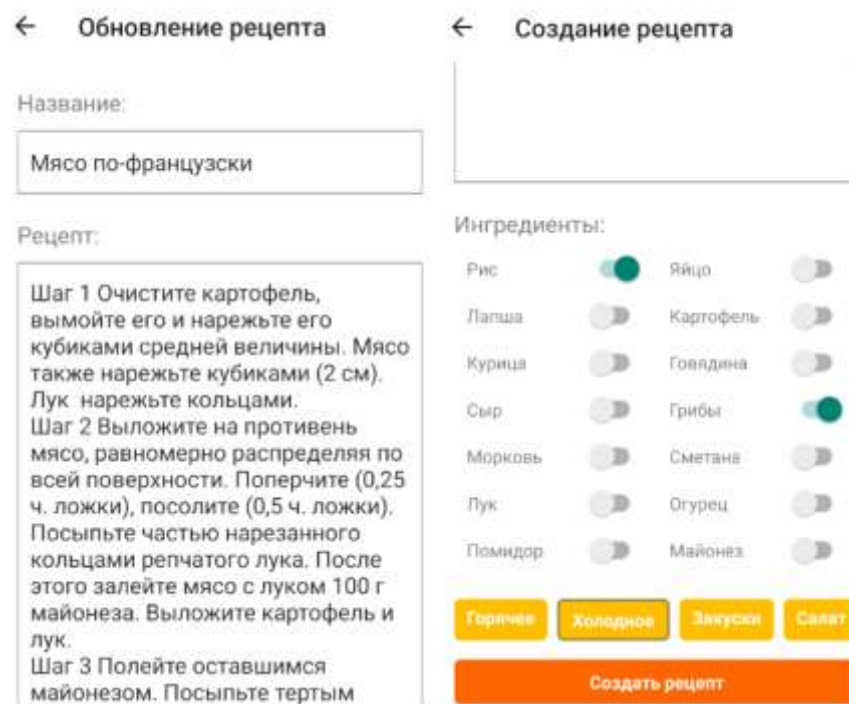


Рисунок 25 – Обновление и создание рецепта

3.5 Вывод по главе

В данной главе:

- описаны технологии для разработки мобильного приложения;
- описаны серверная и клиентская части мобильного приложения;
- описаны основные функции и элементы приложения;
- реализовано мобильное приложение согласно спецификации требований.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы, было разработано мобильное приложение, позволяющее пользователю найти рецепты, используя имеющиеся у него ингредиенты. В процессе создания приложения были выполнены следующие шаги:

- проанализированы аналоги разработанного мобильного приложения;
- проанализированы средства разработки мобильного приложения;
- разработана схематичная структура мобильного приложения;
- определены требования к функционалу приложения, включая возможность поиска рецептов по ингредиентам, просмотр рецептов и отображения созданных рецептов;
- функционал приложения был разбит на модули и для каждого модуля был обозначен необходимый функционал;
- рассмотрено взаимодействие модулей;
- реализован дизайн пользовательского интерфейса, соответствующий современным требованиям к мобильным приложениям.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Все курсы онлайн : русскоязычный интернет ресурс о кулинарных мобильных приложениях [Электронный ресурс]. — URL: <https://vse-kursy.com/read/303-15-luchshih-besplatnyh-kulinarnyh-prilozhenii.html> (дата обращения: 12.12.2022).

2. Clickfraud : русскоязычный интернет ресурс о разработке мобильных приложений [Электронный ресурс]. — URL: <https://clickfraud.ru/luchshee-programmnoe-obespechenie-dlya-razrabotki-mobilnyh-prilozhenij-v-2022-godu/> (дата обращения: 13.12.2022).

3. NestJs : официальный сайт [Электронный ресурс]. — URL: <https://nestjs.com/> (дата обращения: 11.02.2023).

4. TypeORM : официальный сайт [Электронный ресурс]. — URL: <https://typeorm.io/> (дата обращения: 11.02.2023).

5. SQLite : официальный сайт [Электронный ресурс]. — URL: <https://www.sqlite.org/index.html> (дата обращения: 11.02.2023).

6. DigitalOcean : блог с множеством статей о разработке на NestJs [Электронный ресурс]. — URL: <https://www.digitalocean.com/> (дата обращения: 11.02.2023).

7. Lucidchart : официальный сайт [Электронный ресурс]. — URL: <https://www.lucidchart.com/pages/ru/> (дата обращения: 01.03.2023).

8. React Native : официальный сайт [Электронный ресурс]. — URL: <https://reactnative.dev> (дата обращения: 10.03.2023).

9. Habr : русскоязычный веб-сайт, который сочетает в себе элементы коллективного блога по темам и новостного портала [Электронный ресурс]. — URL: <https://habr.com/ru/companies/timeweb/articles/663234/#защитникипредохранители--guards> (дата обращения: 10.03.2023).

10. WANAGO.IO : англоязычный блог о разработке приложений с использованием NestJS. — URL: <https://wanago.io/2020/05/25/api-nestjs-authenticating-users-bcrypt-passport-jwt-cookies/> (дата обращения: 20.03.2023).

11. 8host : руководство о применении стилей к приложениям React Native [Электронный ресурс]. — URL: <https://www.8host.com/blog/stilizaciya-v-prilozheniyax-react-native/> (дата обращения: 15.04.2023).

12. MobX : официальный сайт [Электронный ресурс]. — URL: <https://mobx.js.org/README.html> (дата обращения 17.04.2023).

