Министерство образования Новосибирской области ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

**Разработка мобильного приложения для ведения дневника питания**

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

НАТКиГ.210300.043.000ПЗ

Выполнил: Бурцев В.В.

2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 4

1 Исследовательский раздел 5

1.1 Описание предметной области 5

1.2 Образ клиента 5

1.3 Сценарии 5

1.4 Сбор и анализ прототипов 5

2 Проектирование приложения 6

2.1 UI/UX дизайн проекта 6

2.2 Выбор технологии, языка и среды программирования 8

3 Разработка мобильного приложения 9

3.1 Разработка базы данных 9

3.2 Разработка мультимедийного контента 9

3.3 Описание используемых плагинов 9

3.4 Описание разработанных процедур и функций 9

4 Тестирование 10

4.1 Протокол тестирования дизайна приложения 10

4.2 Протокол тестирования функционала приложения 10

ЗАКЛЮЧЕНИЕ 11

СПИСОК ИСТОЧНИКОВ 12

Приложение А 14

ВВЕДЕНИЕ

По данным федеральной службы государственной статистики с 2018 по 2021 год проблемы с лишним весом испытывает 40 и 50 процентов россиян соответственно. Основываясь на эти данные, можно сделать вывод, что этот процент растёт с каждым годом. Чтобы избавиться от лишнего веса необходимо настроить здоровое питание. Важно отметить, здоровое питание это не просто контроль за калориями и жесткими диетами, а в обеспечении организма полноценным рационом, в который входят все необходимые для него продукты: мясо, злаки, овощи, продукты. Здоровое питание важная составляющая здорового образа жизни человека. Именно поэтому, вопросы по темам лишнего веса и здорового питания являются как никогда актуальными.

Целью курсового проекта является разработка мобильного приложения для ведения дневника питания.

Для достижения поставленной цели определены следующие задачи:

1. изучить предметную область;
2. определить основной функционал приложения;
3. выбрать технологии, язык и среду программирования;
4. разработать дизайн дневника питания;
5. разработать базу данных для приложения;
6. разработать функционал приложения;
7. протестировать полученный продукт.

Объект исследования − рацион питания людей.

Предметом исследования является способ контроля рациона питания

Практическая значимость курсового проекта заключается в пользе данного приложения для людей, желающих поменять привычки питания. При помощи данного приложения человек начнёт обращать своё внимание на пищу, которую он употребляет, что поспособствует формированию здорового питания.

1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

* 1. Описание предметной области

На данный момент в России все больше людей становятся сторонниками здоровым образом жизни, а его следствием является здоровое питание. По данным исследования Всероссийского центра изучения общественного мнения от 30 мая 2020 года почти 60% опрошенных россиян следят за своим питанием.

Современный человек ценит своё время, а процесс подсчета употреблённых белков, жиров, углеводов и калорий весьма затратный относительно времени. Трудно представить, чтобы такой человек самостоятельно сел и подсчитал употреблённые им питательные вещества. Дабы сделать этот процесс менее трудоёмким и времязатратным, на данный момент разрабатывается приложение, нацеленное на решение этой проблемы. При помощи этого приложение пользователь сможет добавить часто употребляемую им пищу, указать для неё все питательные вещества на 100 грамм. В дальнейшем от пользователя потребуется указывать вес, данной пищи, а после приложение само подсчитает белки, жиры, углеводы и калории.

* 1. Образ клиента

Данное приложение нацелено на российскую аудиторию, которая интересуется здоровым образом жизни, и обращает внимание на свой рацион питания. Как правило, здоровым питание человек начинает интересоваться уже в осознанном возрасте.

* 1. Сценарии

Разрабатываемое мобильное приложение предназначено для людей, которые хотят ежедневно вести заметки об употребляемой им пищи и при этом не тратить на это большое количество времени.

* 1. Сбор и анализ прототипов

2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

2.1 UI/UX дизайн проекта

Для проектирования дизайна приложения выбран онлайн-сервис для разработки интерфейсов «Figma».

В качестве основного логотипа мобильного приложения выбрано изображение с яблоком (Рисунок 1), т.к. оно минималистичное и ассоциируется с полезной пищей.



Рисунок 1 – Логотип приложения

Определена основная цветовая схема приложения для светлой темы, которая представлена в таблице 1. Именно эти цвета у человека ассоциируются со здоровье, здоровым питанием, потому что зелёный цвет и его оттенки − это естественный цвет, который человек часто встречает в окружающем мире.

Таблица 1 – Цветовая схема приложения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| #298043 | #9EFFBB | #52FF86 | #4F805E | #41CC6A |

На основе данной цветовой схемы   
разработан дизайн, который состоит из 6 экранов. При запуске программы отображается загрузочный экран (Рисунок 2).



Рисунок 2 – Загрузочный экран приложения

Основным экраном приложения является экран, на котором отображается список записей употребляемой пищи (Рисунок 3).

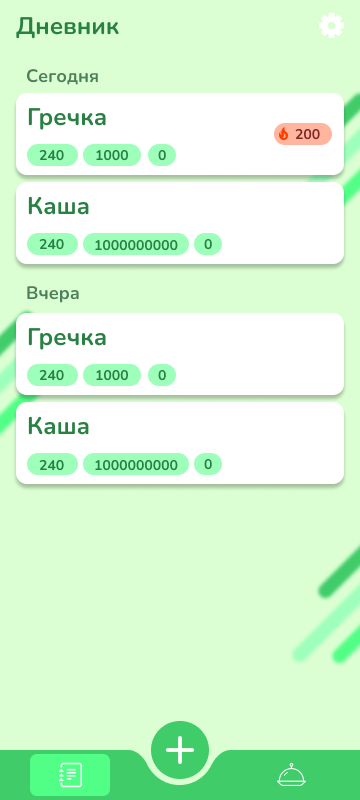


Рисунок 3 – Экран записей употребляемой пищи

Для добавления записи на основной экран, необходимо перейти на экран добавления записи (Рисунок 4). На данном экране пользователь может выбрать из списка ранее добавленных блюд нужное, после чего вписать вес употреблённой пищи. Далее, для добавления записи нужно нажать на кнопку.

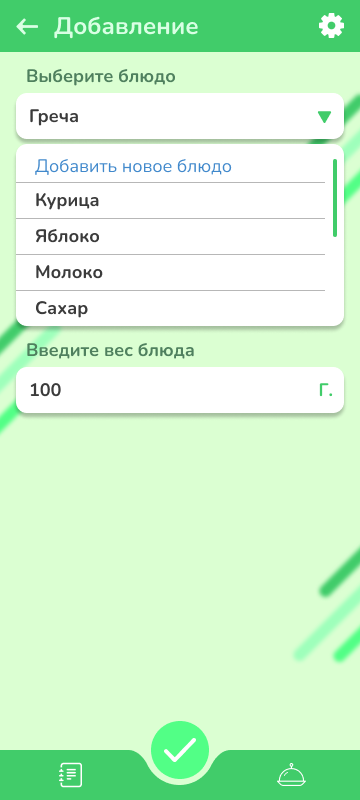


Рисунок 4 – Экран добавления записи в дневник

Чтобы пользователь мог вести записи личного дневника питания разработан экран для вывода списка добавленных блюд (Рисунок 5) и экран добавление этих самых блюд (Рисунок 6).



Рисунок 5 – Экран списка добавленных блюд

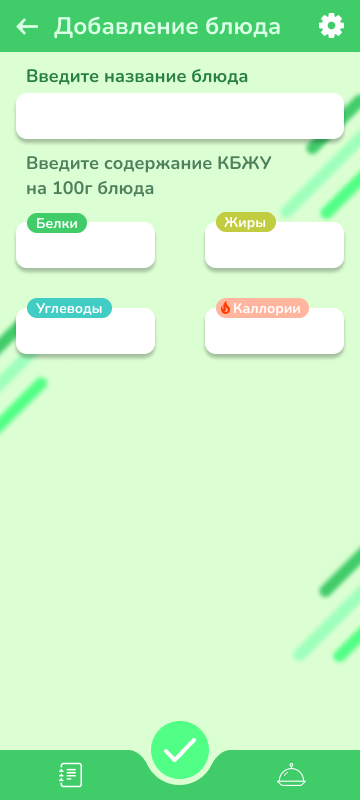


Рисунок 6 – Экран добавления блюда

На экране добавления блюда (Рисунок 6) пользователю необходимо ввести уникальное название блюда, а также содержание белков, жиров, углеводов и калорий в блюде на 100 грамм.

Последним экраном в приложении является экран настроек   
(Рисунок 7), где пользователь сможет выбрать цветовую схему приложения, которая ему более комфортна и информацию о приложении.

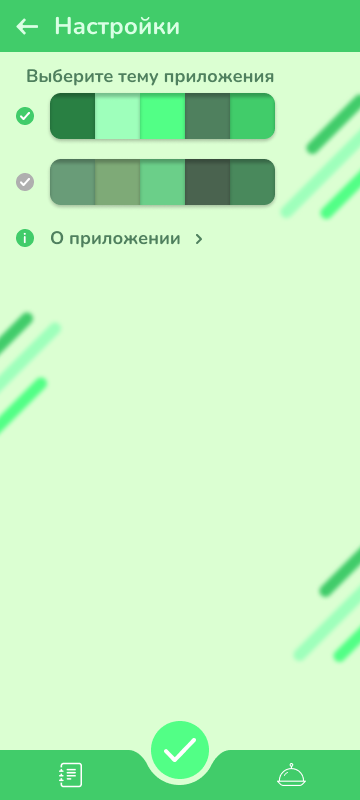


Рисунок 7 – Экран настроек

Для улучшения пользовательского опыта в использовании данного приложения предусмотрены плавные анимации перехода на каждый экран.

2.2 Выбор технологии, языка и среды программирования

Вся разработка проекта ведётся в IDE Android Studio. Используется именно эта среда разработки, т.к. разрабатываемое мобильное приложение является нативным.

Для решения поставленной задачи выбран язык программирования Kotlin, т.к. на данный момент является современным и обладает всеми необходимыми технологиями и библиотеками, которые могут решить данные задачи.

3 РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

3.1 Разработка базы данных

В качестве разрабатываемой базы выбрана локальная система управления базами данных − SQLite. Для её разработки на данный момент существует библиотека Room. В данном приложении используются две сущности: «FoodDbEntity» и «JournalDbEntity», реализации которых приведены на рисунке 8 и 9 соответственно.

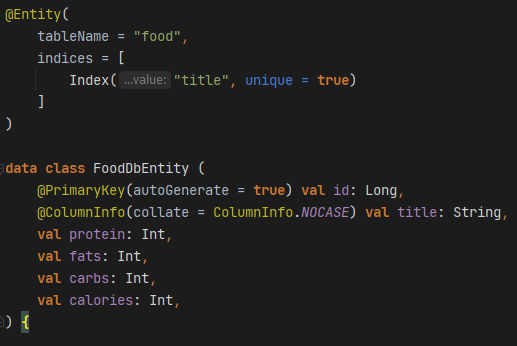


Рисунок 8 – Сущность базы данных «FoodDbEntity»



Рисунок 9 – Сущность базы данных «JournalDbEntity»

Как видно из выше приведенных рисунков их реализация немного отличается, т.к. сущность «JournalDbEntity» имеет внешний ключ из таблицы «FoodDbEntity».

После создания сущностей базы данных необходимо прописать к ним запросы, чтобы иметь возможность манипулировать данными из приложения. Для реализации данной функции необходимо прописать следующие запросы: «FoodDao» для «FoodDbEntity» (Рисунок 10) и «JournalDao» для «JouranlDbEntity» (Рисунок 11).

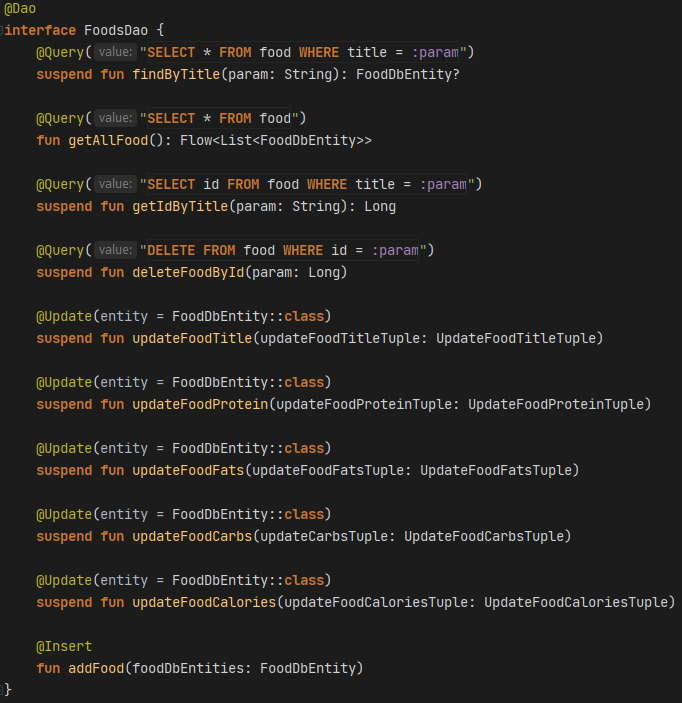


Рисунок 10 – Запрос «FoodDao»

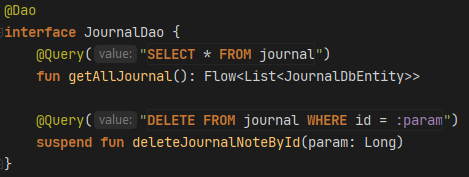


Рисунок 11 − Запрос «JournalDao»

Далее необходимо написать реализацию к этим запросам

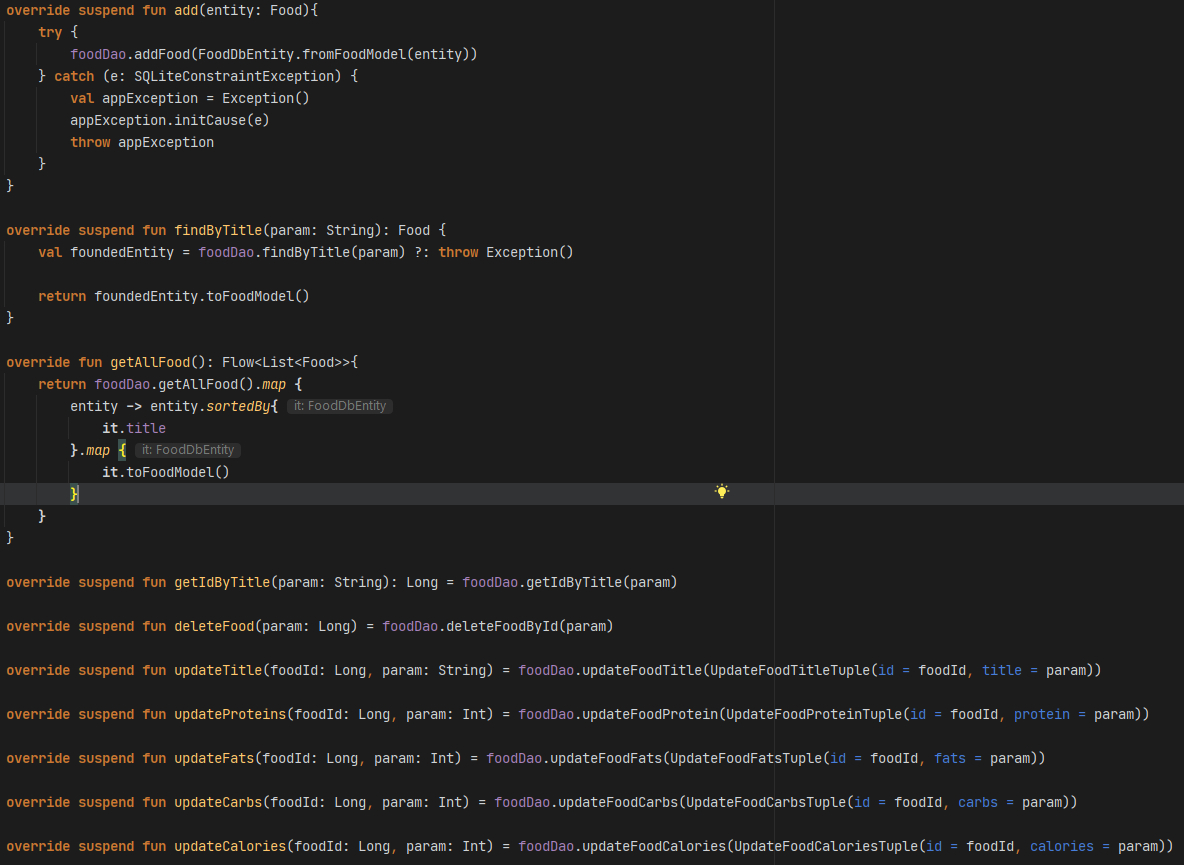


Рисунок 12 – Реализация запросов

И последним шагом в разработке базы данных является реализация класса (Рисунок 13), который даст возможность получать данные из разработанной локальной базы данных.

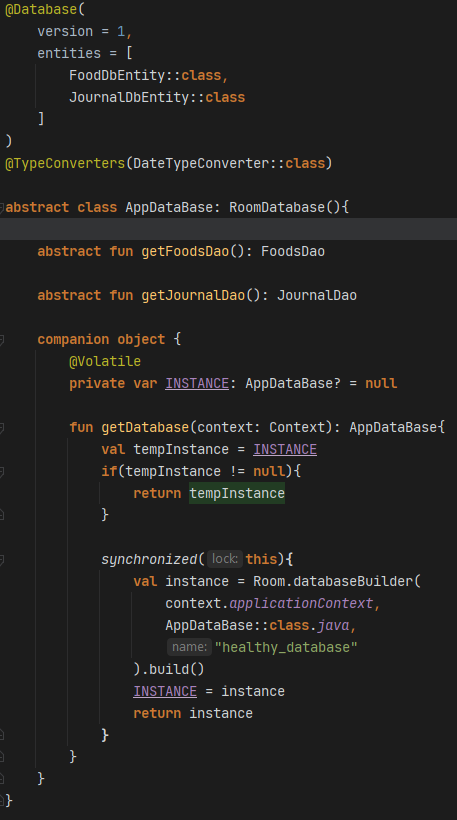
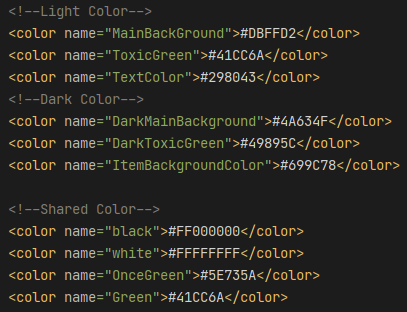


Рисунок 13 – Реализация класса «AppDataBase»

3.2 Разработка мультимедийного контента

Вся верстка приложения ведётся на языке разметки «Xml». Перед началом разработки дизайна все цвета приложения, размер шрифта необходимо вынести в отдельные файлы, как представлено на рисунке 14.



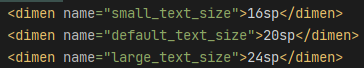


Рисунок 14 – Вынесение основных значений

Для использования значков в приложении и задних фонов для своих элементов их необходимо выносить в отдельные файлы с расширением «Xml», в папку drawable (Рисунок 15).

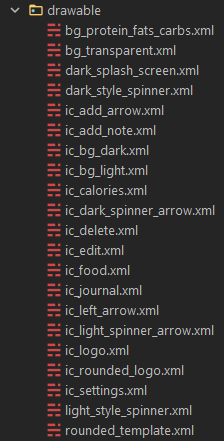


Рисунок 15 – Вынесение значков и задних фонов в отдельные файлы

Каждый экран приложения также выносится в отдельный файл (Рисунок 16) с расширением «Xml». Все экраны находятся в одной папке с название layout.

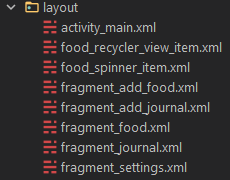


Рисунок 16 – Экраны приложения

Чтобы реализовать светлую и темную тему в приложении необходимо параллельно прописывать стили для каждого файла themes, представленных на рисунке 17.



Рисунок 17 – Файлы themes для светлой и темной темы

Пример стиля написания стиля для задней поверхности элемента светлой и темной темы на рисунке 18 и 19 соответственно.

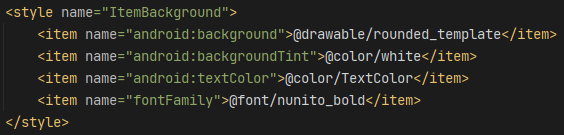


Рисунок 18 – Стиль задней поверхности элемента для светлой темы приложения

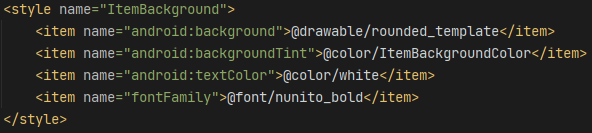


Рисунок 19 – Стиль задней поверхности элемента для темной темы приложения

3.3 Описание используемых плагинов

При создании проекта по умолчанию в «build.gradle (:app)» устанавливаются следующие зависимости представленные на рисунке 14.

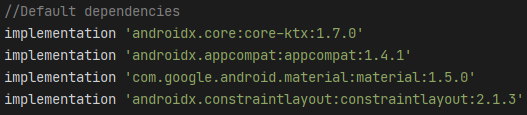


Рисунок 14 – Предустановленные зависимости

Для проведения тестов приложения важно установить зависимости, представленные на рисунке 15.

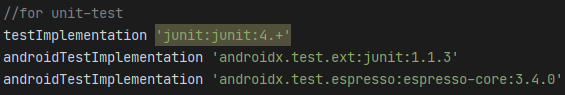


Рисунок 15 – Зависимости для проведения Unit тестов

Чтобы реализовать локальную базу данных, необходимо установить библиотеку Room (Рисунок 16).

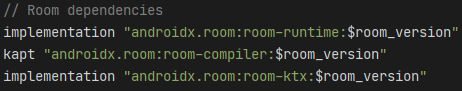


Рисунок 16 – Установка библиотеки Room

Для реализации паттерна проектирования MVVM нужно прописать зависимости представленные на рисунке 17.

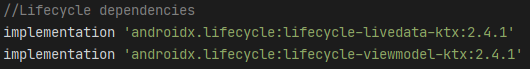


Рисунок 17 – Библиотеки для реализации паттерна проектирования

Чтобы успешно реализовать заданный интерфейс приложения, а также сделать его масштабируемым под разные расширения экранов необходимо установить следующие зависимости (Рисунок 18)

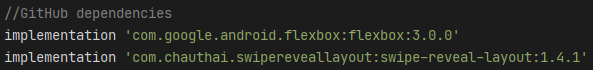


Рисунок 18 – Зависимости для реализации интерфейса

Для реализации асинхронности в мобильном приложении необходимо установить библиотеку Coroutines (Рисунок 19)



Рисунок 19 – Установка зависимости Coroutines

3.4 Описание разработанных процедур и функций