

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего образования

«Московский политехнический университет»

Факультет информационных технологий

Кафедра «Инфокогнитивные технологии»

Образовательная программа «Корпоративные информационные системы»

Курсовой проект

«FoodHelper»

Выполнили:

Студенты группы 191-361

Ларичев С.Ю.

Базавлюк О.И.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	4
1.1 ВЫБОР ТЕХНОЛОГИЙ И ИНСТРУМЕНТОВ	4
1.2 ВЫБОР РЕАЛИЗУЕМОГО ФУНКЦИОНАЛА.....	4
1.3 ВЫВОД ПО ТЕОРЕТИЧЕСКОЙ ЧАСТИ	5
2. ПРАКТИЧЕСКАЯ ЧАСТЬ.....	6
2.1 РАЗРАБОТКА ДИЗАЙНА	6
2.2 РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ	7
2.3 РАЗРАБОТКА И ТЕХНИЧЕСКИЕ РЕШЕНИЯ.....	11
2.4 СТРУКТУРА БИБЛИОТЕКИ КЛАССОВ ПРОЕКТА	12
2.5 ВЫВОД ПО ПРАКТИЧЕСКОЙ ЧАСТИ	13
ЗАКЛЮЧЕНИЕ	14

ВВЕДЕНИЕ

В современном мире остро стоит проблема лишнего веса, который негативно влияет на здоровье человека. Подобная проблема возникает от привычки переедать либо есть несбалансированную пищу. Помимо этого, существует проблема недоедания. Из-за быстрого темпа жизни человек не замечает, что он съел и не может подсчитать потребляемые калории. Из-за недоедания, человек быстро устает, начинает резко снижать вес, ухудшается здоровье, так как не поступает достаточное количество питательных веществ.

Самый простой способ привести свой организм в норму – это начать подсчитывать потребляемые калории и соотносить их с расходуемыми калориями, после чего, вес в обязательном порядке будет постепенно приходить в норму, а общее самочувствие улучшаться вне зависимости от цели.

Для того чтобы можно было удобно отслеживать потребляемые в обычной жизни блюда, было разработано приложение «FoodHelper», которое поможет следить за расходуемыми и потребляемыми в течении дня калориями. Используя приложение «FoodHelper» можно оптимизировать и улучшить рацион питания, отказавшись от излишков или, наоборот, ввести в рацион больше питательных веществ.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Выбор технологий и инструментов

При реализации проекта был выбран следующий стек технологий:

- SQLite – в качестве СУБД;
- С# - для реализации логической части, то есть backend;
- UWP – для реализации front части.

1.2 Выбор реализуемого функционала

Перед реализацией проекта был выбран основной функционал, который необходимо реализовать:

- Вывод статистики КБЖУ за день, за неделю и за 30 дней;
- Вывод статистики сожженных калорий за день, за неделю и за 30 дней;
- Добавление блюда, которое было съедено сегодня для дальнейшего расчета статистики;
- Добавление сожженных калорий за день;
- Возможность добавления рецепта с нефиксированным количеством ингредиентов;
- Возможность добавить ингредиент и его показатели;
- Возможность авторизации и регистрации;
- Возможность запоминания данных аккаунта, для того чтобы при последующем входе в приложение не нужно было авторизовываться;
- Показ съеденных за день блюд.

Данного функционала достаточно для первичного запуска и работы с приложением. В последующем функционал будет расширяться.

1.3 Вывод по теоретической части

Таким образом был подготовлен весь необходимый материал для разработки приложения, создана база данных, импортированы данные для проверки работы приложения, разработан дизайн.

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Разработка дизайна

Перед тем как приступить к реализации проекта, был разработан дизайн с помощью онлайн-сервиса для разработки интерфейсов Figma.

Дизайн представлен на рисунке 1.



Рисунок 1 — Дизайн-макет проекта

Более подробно макет можно рассмотреть по ссылке www.figma.com. Так макет является полностью кликабельным.

Цвета, выбранные для проекта представлены на рисунке 2.

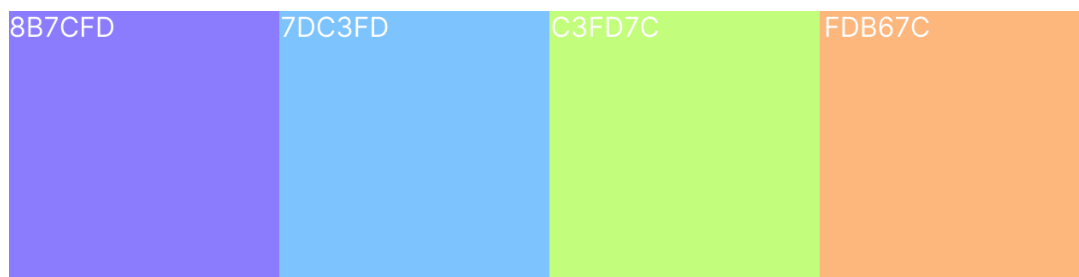


Рисунок 2 — Цветовая палитра проекта

Выбор цветовой гаммы обусловлен влиянием цветов на человека:

- Голубой – успокаивает, не дает переедать;
- Зеленый – поддерживает в тонусе;
- Оранжевый – улучшают настроение, избавляют от депрессивных состояний;

- Фиолетовый – способствует гармонии души и тела, положительно влияет на эмоционально неуравновешенных людей, снимает усталость.

2.2 Реализация приложения

Дизайн проекта реализован на универсальной платформе Windows (UWP). На главной странице (рисунок 3) реализован резиновый дизайн, то есть при изменении размера окна, адаптивно изменяются размеры блоков, то есть блоки хранятся в таких адаптивных контейнерах как, Grid и StackPanel. Все универсальные стили кнопок, текстовых блоков хранятся в файле App.xaml.

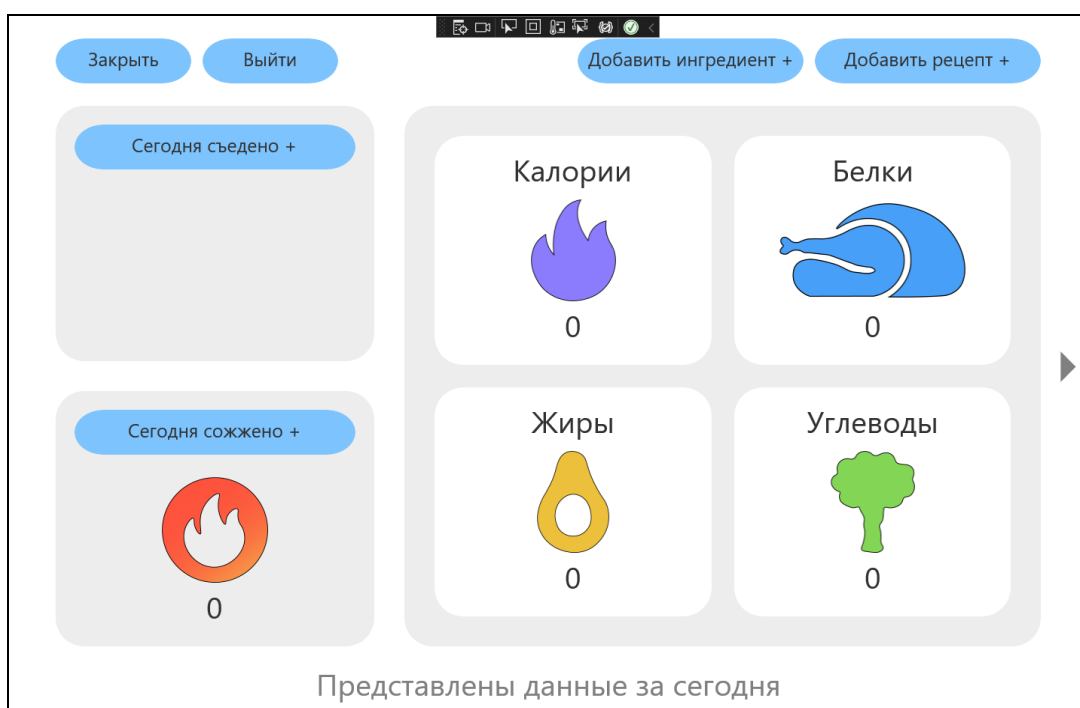


Рисунок 3 — Главная страница

Главная страница отображает статистику пользователя за день, неделю, месяц в блоках, которые обновляются, при добавлении съеденного блюда, либо изменении периода отображения. Статистика представляет собой: КБЖУ, количество сожженных калорий, а также список съеденных блюд. На главную страницу можно перейти из фреймов Auth () (рисунок 4) и Register () (рисунок 5), также реализовано запоминание пользователя в файле приложения, при котором в приложении сразу открывается фрейм главной страницы. С главной страницы можно: добавлять блюда () (рисунок 6), сожженные калории () (рисунок 7), добавлять новые ингредиенты () (рисунок 8) и новые рецепты () (рисунок 9).

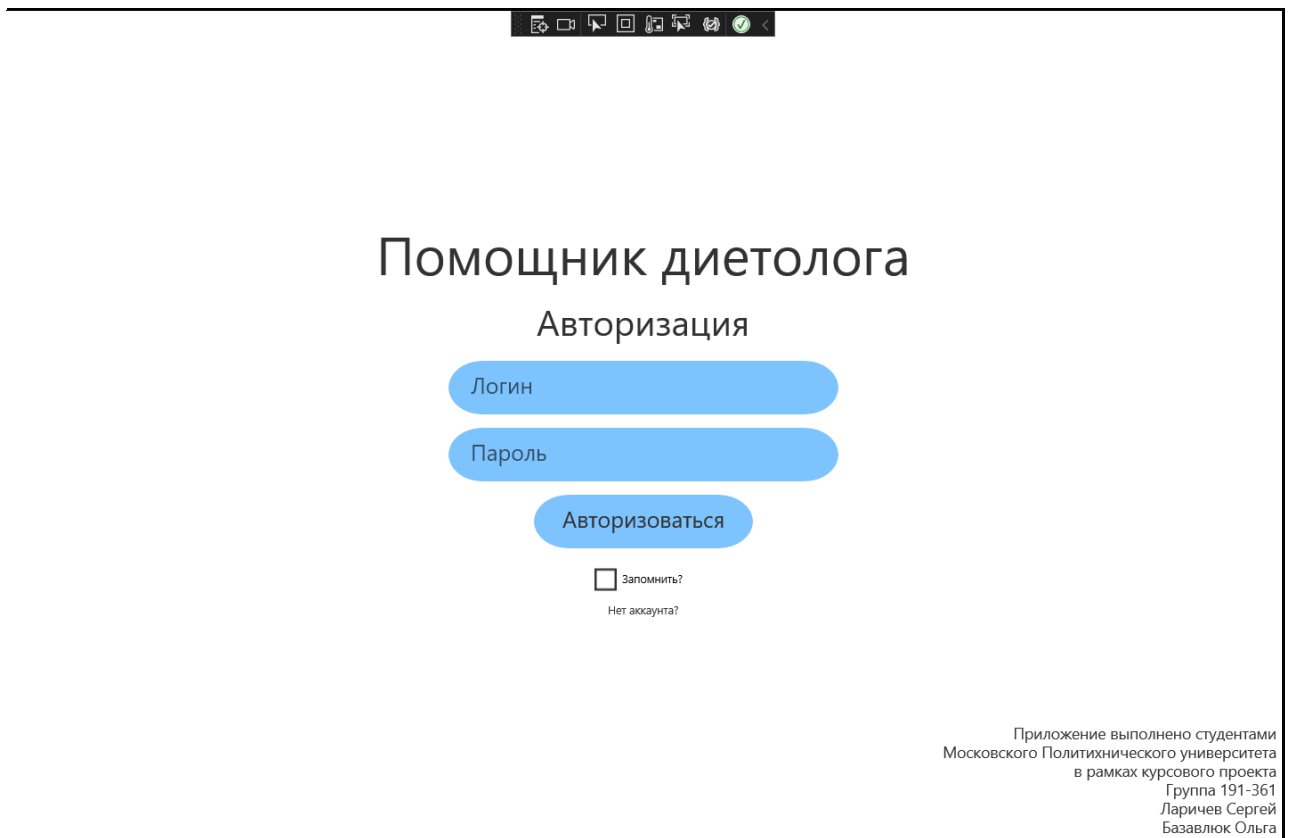


Рисунок 4 — Страница авторизации

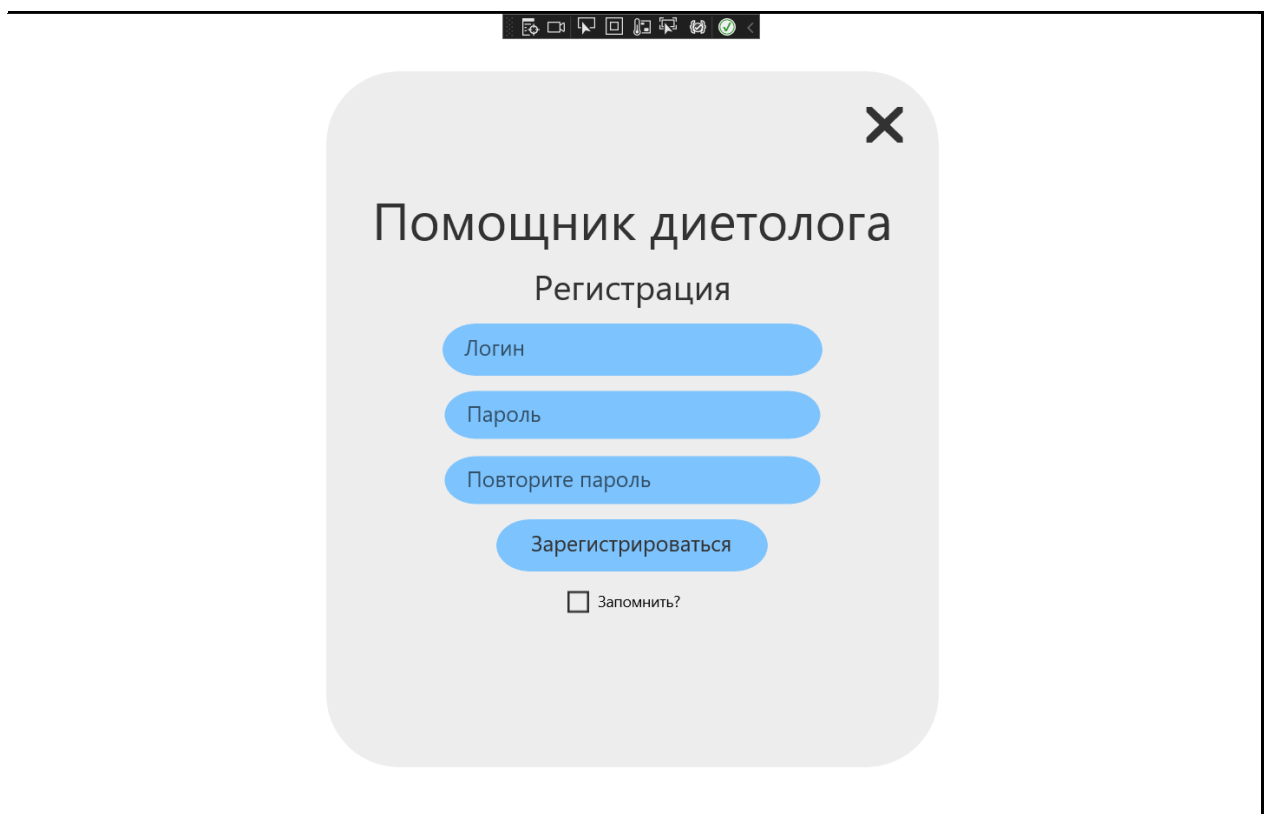


Рисунок 5 — Страница регистрации

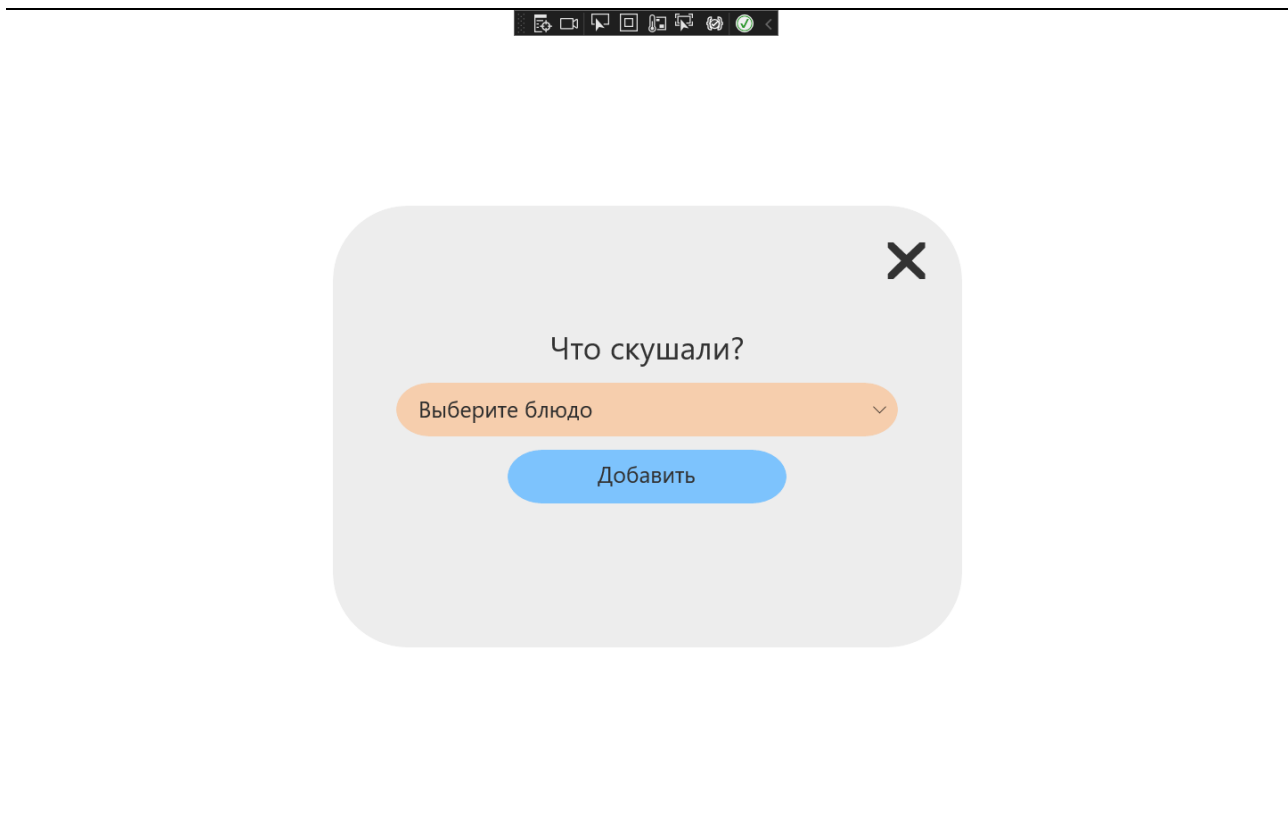


Рисунок 6 — Страница добавления блюда

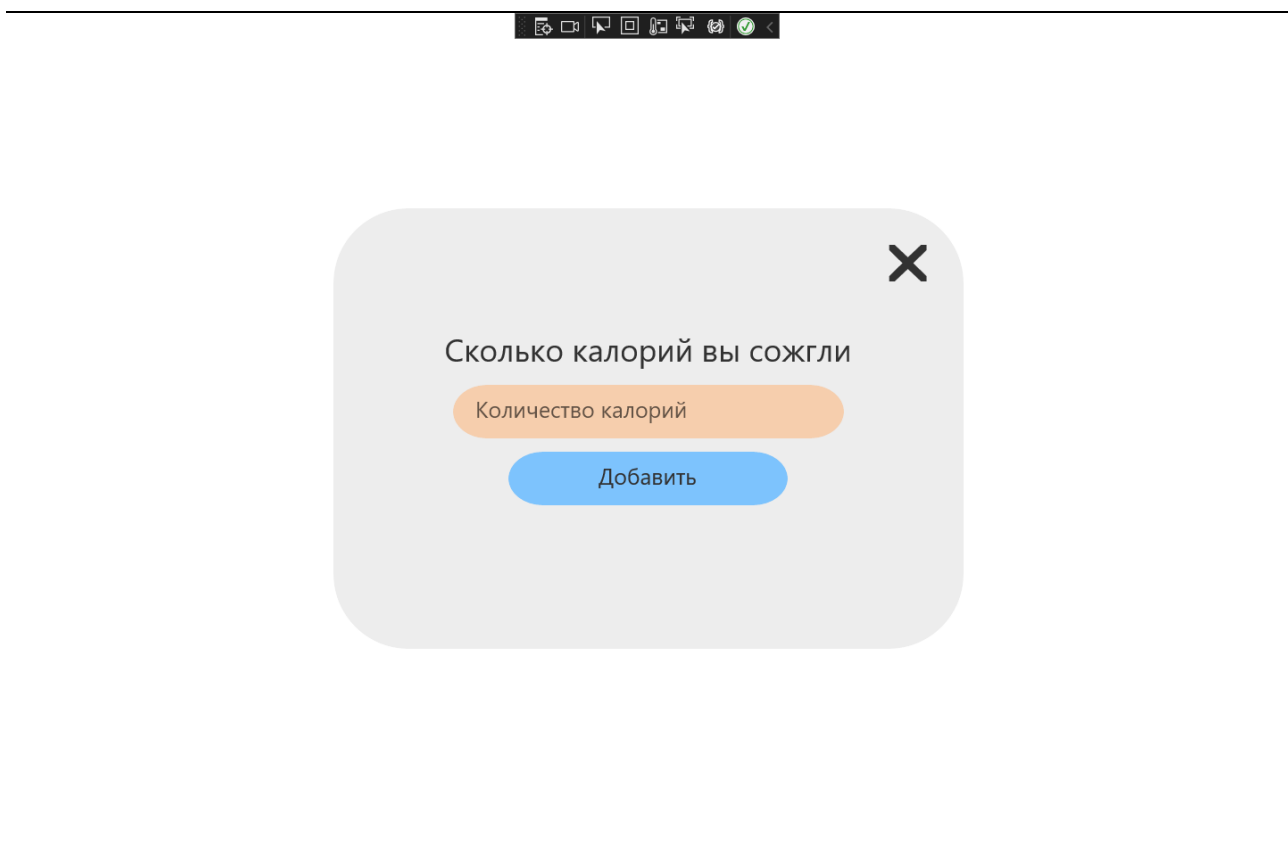


Рисунок 7 — Страница добавления сожженных калорий

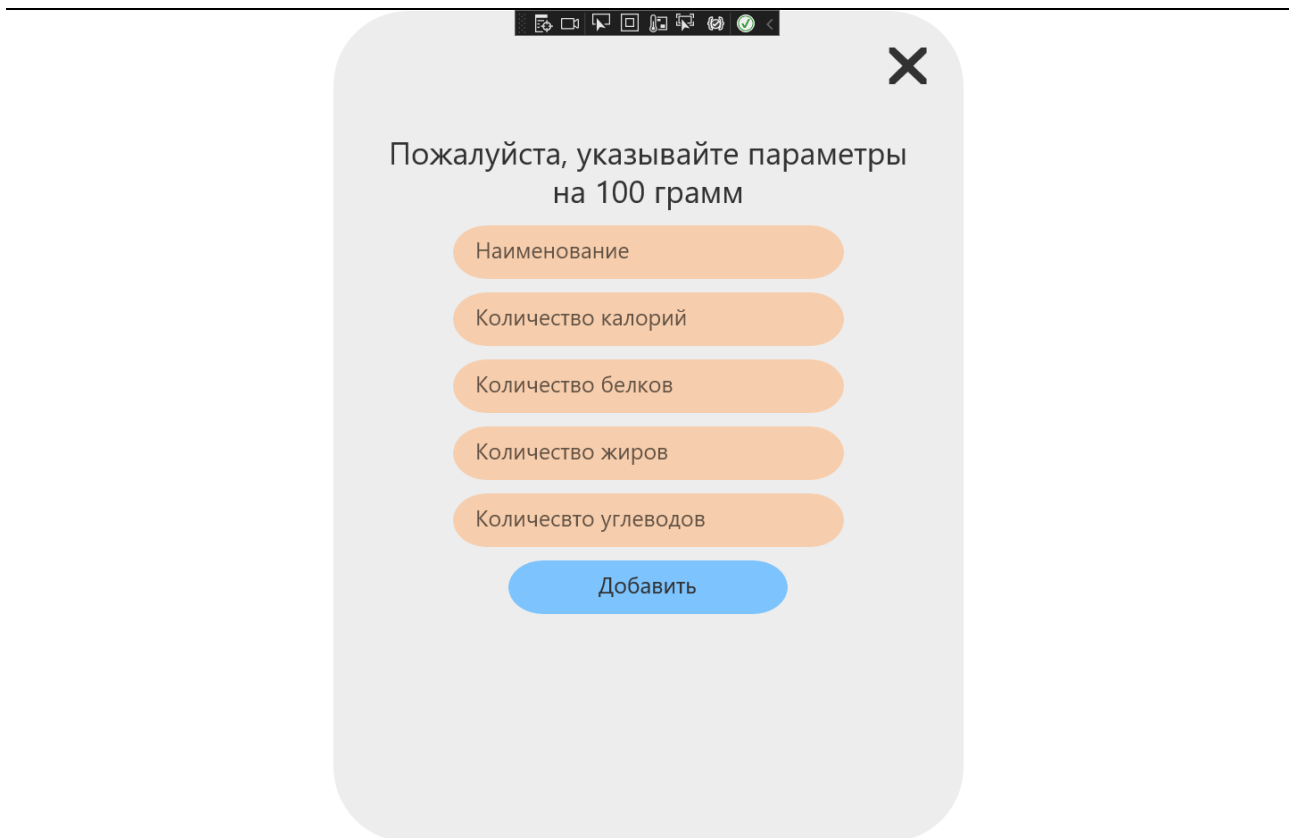


Рисунок 8 — Страница добавления ингредиента



Рисунок 9 — Страница добавления рецепта

На странице авторизации вводятся логин и пароль пользователя, также можно запомнить пользователя и перейти на страницу регистрации, если

аккаунта еще нет. При нажатии на кнопку "Авторизоваться", путем SQL запроса, происходит поиск введенных данных в поля и, если такой пользователь существует, осуществляется переход на главную страницу.

Страница регистрации сделана аналогично авторизации, за исключением того, что пользователь добавляется в базу посредством SQL запроса.

На странице добавления блюда из выпадающего списка выбирается рецепт и нажимается кнопка "Добавить". В выпадающий список рецепты попадают из таблицы рецептов с помощью выборки SQL запросом. Далее с помощью SQL запроса производится проверка есть ли данное блюдо у данного пользователя за сегодняшний день, если есть, то производится обновление поля количества. Если блюда нет, то производится добавление данных.

Страница добавления сожженных калорий сделана аналогично странице добавления блюда, за исключением того, что пользователь вводит количество сожженных калорий.

На странице добавления ингредиента производится ввод наименование ингредиента, далее его КБЖУ и нажимается кнопка "добавить". Далее с помощью SQL запроса данные добавляются в базу данных.

На странице добавления рецепта производится ввод наименования рецепта, далее в выпадающем списке выбирается ингредиент и рядом вводится количество грамм в данном рецепте. По кнопке "Добавить ингредиент" добавляется еще один выпадающий список и поле для ввода количества. По кнопке "Добавить рецепт" с помощью SQL запроса в базу добавляется новый рецепт и его составляющие.

2.3 Разработка и технические решения

На основе реализуемых функций разработана база данных, которая содержит 6 таблиц, 1 представление и связи между ними, сохраняющие целостность данных. Представление – это виртуальная таблица, которая содержит данные из других таблиц и при добавлении данных в них, данные в представлении обновляются. В данном случае, это необходимо для того, чтобы

хранить КБЖУ каждого рецепта и при добавлении нового рецепта данная статистика рассчитывалась и добавлялась в таблицу. Это позволяет упростить разработку приложения и избежать ошибок при вычислении. Разработанная база данных изображена на рисунке 10.

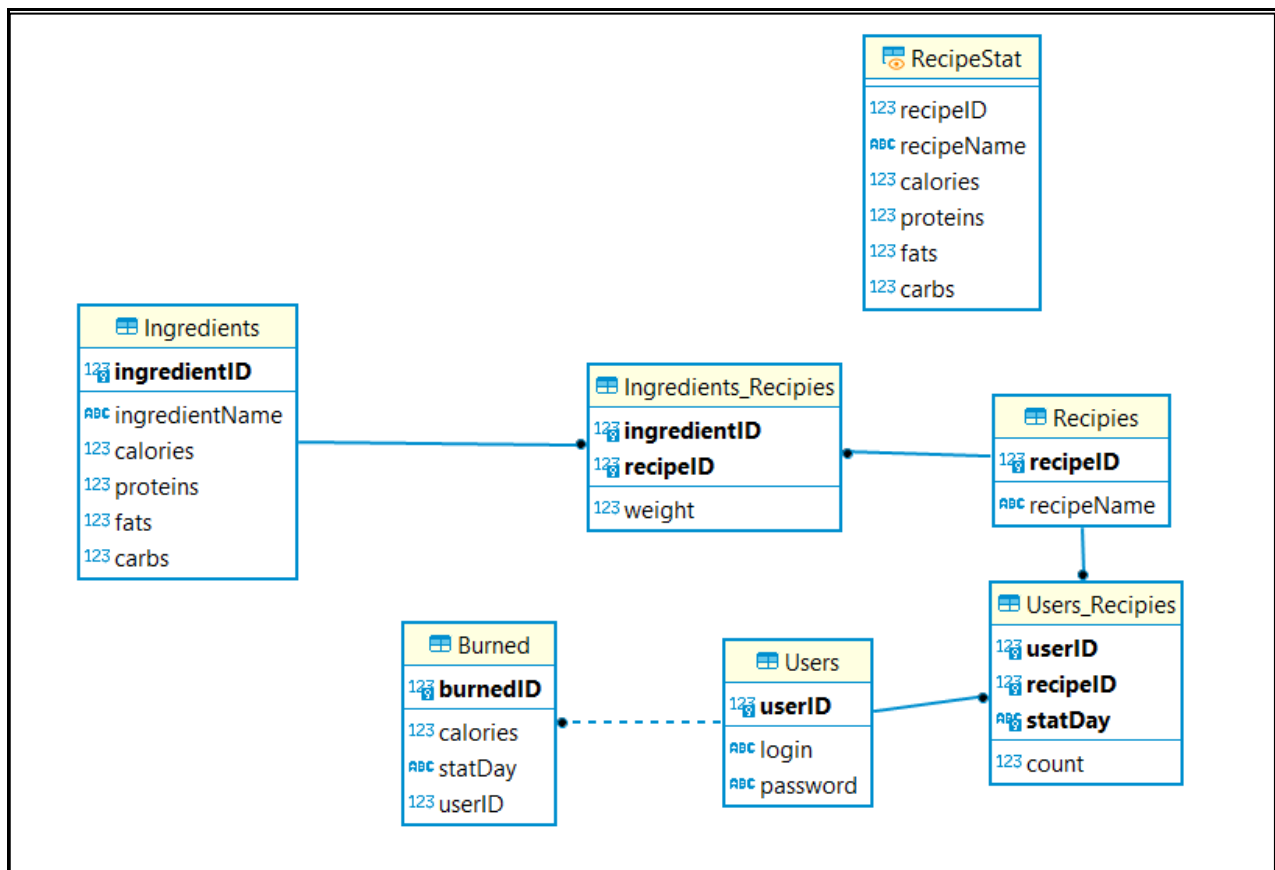


Рисунок 10 — Схема базы данных

Сама база данных создавалась с помощью запросов, далее, через менеджер СУБД DBeaver в файл базы данных производился импорт данных. Данные импортировались из csv файлов, подготовленных на основе схемы базы данных.

2.4 Структура библиотеки классов проекта

Так как SQLite — локальная СУБД, при инициализации БД, посредством запроса, происходит создание таблиц и представления, если они отсутствуют. База данных хранится в хранилище данных приложения. Также библиотека содержит методы, которые реализуют запросы к таблицам и позволяют получить данные как из таблиц, так и из представления. Далее, в библиотеке содержатся такие структуры, как User и Stats, которые содержат в себе данные о пользователе и его статистике, соответственно.

2.5 Вывод по практической части

В результате выполнения практической части было реализовано приложение, состоящее из запросов к базе данных, которые позволяют обрабатывать запросы, исходящие от фронтенд. Помимо этого, в рамках практической части была разработана база данных и дизайн приложения.

ЗАКЛЮЧЕНИЕ

Так как проект выполнялся в команде, то разработка велась с периодическим сохранением в Git репозиторий, который находится по ссылке <https://github.com/s3rg0sh4/FoodHelperApp>.

В рамках курсового проекта был реализован весь необходимый функционал для работы с приложением и решена проблема отслеживания съеденных калорий и полезных элементов в течении дня.