Пояснительная записка к контрольному домашнему заданию по дисциплине «программирование».

Образовательная программа «бизнес-информатика»

Бакалавриат, 1-й курс.

Шаяхмедова Луиза ББИ168

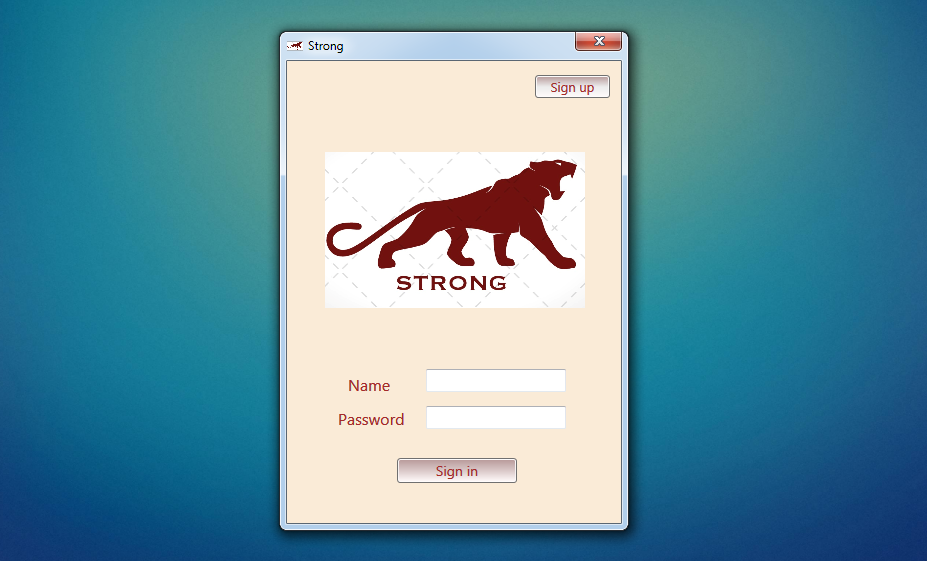
1. Тема: Разработка приложения для контроля питания.
2. Адрес репозитория: <https://github.com/louise98/probniy>
3. Одиночный проект.
4. Аннотация: Целевая аудитория программы – пользователи, желающие контролировать свой вес и объем употребленных калорий в день. Зарегистрироваться может несколько пользователей. В зависимости от цели, программа рассчитывает оптимальный объем калорий в день для данного пользователя. Пользователь имеет возможность добавлять пищу, просматривать историю, прогресс, процентное соотношение употребленной еды, а также изменять настройки.
5. Используемые технологии и библиотеки:

Данный проект можно разделить на две части – бэкенд и WPF приложение.

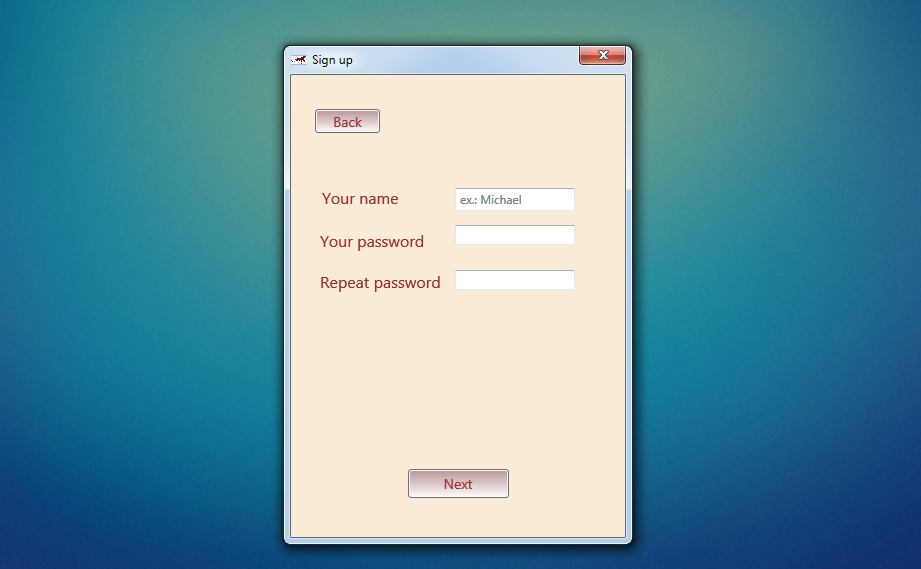
Рассмотрим их отдельно.

Для создания бэкэнд части использовался стэк Python/Flask. На самом деле, сайт не является полноценным бэкэндом для приложения, он нужен только для генерации графиков. Сайт может генерировать два вида графиков – pie chart и обычный flow chart. Для генерации графиков используется библиотека matplotlib.

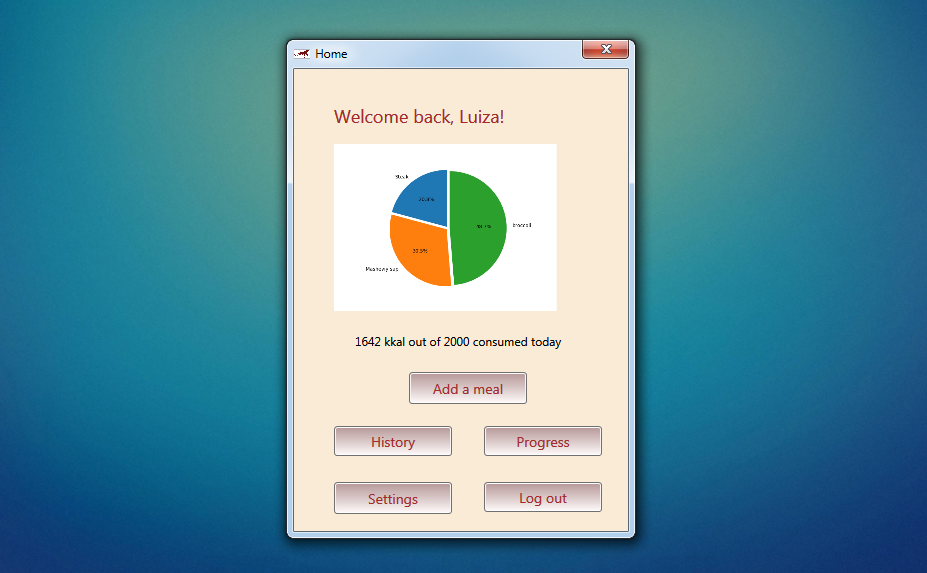
В WPF приложении используются стандартные элементы WPF. Реализован CRUD, сериализация, авторизация, веб-запросы для получения графиков.

1. Интерфейс программы:

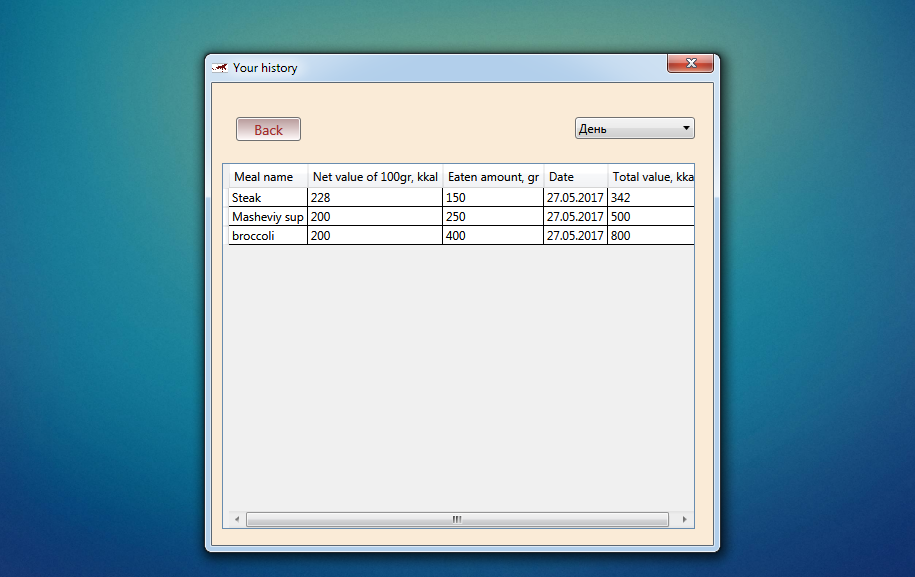
Окно входа



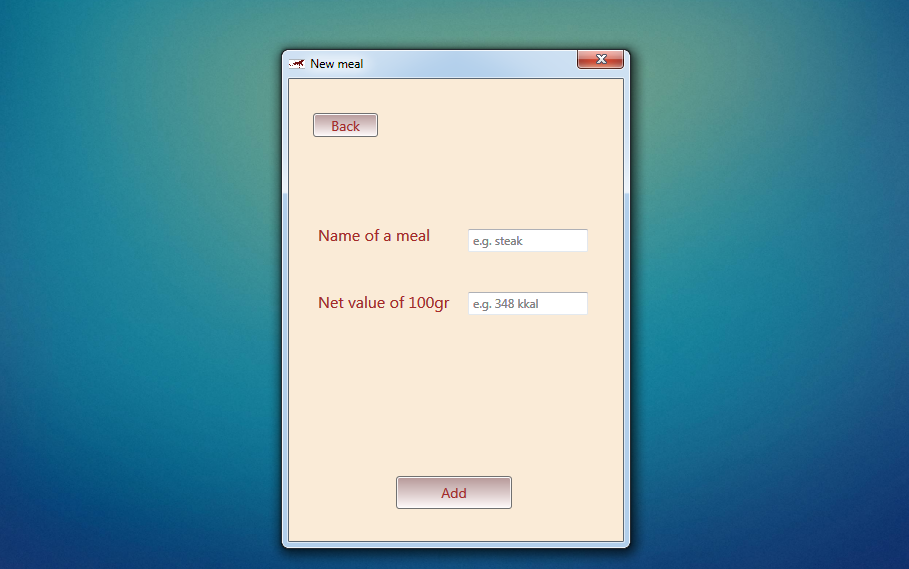
Окно регистрации нового пользователя



Главное окно приложения (домашняя страница пользователя)



Окно просмотра истории употребленных блюд пользователя



Окно добавления нового блюда в программу



Окно настроек

1. Состав классов с кратким описанием каждого класса:

User

Состоит из полей возраст, пол, цель, имя, пароль и другая личная информация. Нужен для авторизации пользователя в приложении, все дальнейшие действия привязываются к конкретному пользователю. Сериализуется в файл при обновлении, десериализуется при запуске нужных экранов. Глобальная база юзеров хранится в Data.users. Каждая сессия приложения привязана к определенному пользователю из базы, это значение хранится в Data.currentUser как ссылка на элемент Data.users. Есть один конструктор, некоторые поля являются свойствами, если к ним нужен read/write доступ. Имеет поля типа enumeration. Один из основных компонентов – User.history – история всех приемов пищи пользователя.

Meal

Класс, определяющий прием пищи. Является сериализуемым. Имеет поля название, пищевая ценность на 100 грамм, принятая порция, дата приема и вычислимое свойство Total, показывающее общую калорийность приема пищи.

Data

Класс, содержащий глобальную информацию, используемую кросс-оконно. Главные поля – currentUser (текущий юзер), users (база всех пользователей), meals (база блюд, добавляемых пользователями)

HelperFunction

Агрегирует статические функции для валидации вводимых данных, создания юзера и сериализации/десериализации.

1. Список использованных источников:

Ефремов С.Г. – оффлайн и онлайн консультации.