# Приложение HealthyFood

Целью работы программы является автоматизация процесса составления расписания здорового питания. В программу также включены:

1. предоставление возможности персонализации предоставляемого плана в соответствии с целями и физическими параметрами пользователя
2. удобное управление бюджетом
3. поиск магазинов для покупки продуктов
4. контроль БЖУ и прочих параметров
5. предоставление списка рецептов, возможности их редактирования и добавления

## Сущности

* **Пользователь** – физическое лицо, которое может использовать приложение.
* **Магазин** – сущность, отвечающая за место, где можно купить продукт. Имеет название, адрес и ссылку список продуктов.
* **Продукт** – элементарная единица, участвующая в приготовлении ингредиента. Хранит информацию о БЖУ, калорийности, списке цен в возможных для покупки магазинах и лучшего магазина с самой выгодной ценой.
* **Ингредиент** – составная из одного или нескольких продуктов сущность, входящая в рецепт. Содержит информации о суммарном БЖУ продуктов, калорийности и о составе (списке продуктов).
* **Рецепт** – описание приготовления некоторого блюда. Включает список ингредиентов и процесс приготовления в виде текста или ссылки на страницу в интернете.
* **Блюдо** – нечто, что пользователь может приготовить или употребить в уже готовом виде. Состоит из рецепта (если таковой имеется), БЖУ, калорийности, предпочтительного времени употребления и суммарной стоимости ингредиентов.
* **Приём пищи** – сущность, сопоставляющая блюдо и время его употребления.
* **Паттерн** – шаблон дня недели с распределением какого-либо набора блюд по приёмам пищи, содержит суммарную стоимость, калорийность и суммарное БЖУ блюд.
* **Расписание** – набор паттернов дней на всю неделю.
* **Модель питания** – сущность, которая используется для учёта параметров и цели пользователя, а также хранения критериев создания паттернов дня.

## Свойства сущностей

#### Пользователь.

Редактируется самим пользователем приложения.

1. Имя.
2. Адрес электронной почты.
3. Вес.
4. Рост.
5. Телосложение. ENUM: «Эктоморф», «Мезоморф», «Эндоморф».
6. Пол. ENUM: “W”, “M”
7. Возраст.
8. Цель занятий. Имеет пользовательский тип ENUM. Может принимать значения «Похудение», «Набор массы», «Поддержание формы».
9. Недельный бюджет. Сумма денег, за которую не должны перейти траты на продукты за неделю.
10. Название модели питания.
11. Название текущего расписания.

#### Магазин.

Редактируется администратором.

1. Название.
2. Адрес.
3. Ссылка на магазин.
4. Список продуктов и цен. Массив из названий существующих в базе продуктов.

#### Продукт.

Редактируется администратором.

1. Название.
2. Лучший магазин. Магазин с самой выгодной ценой.
3. БЖУ. Словарь формата: {“белки”: число, “жиры”: число, “углеводы”: число}.
4. Калорийность.

### *Ингредиент.*

Редактируется администратором. Пользовательские ингредиенты могут задаваться в локальной базе данных самим пользователем.

1. Название.
2. БЖУ. Словарь формата: {“белки”: число, “жиры”: число, “углеводы”: число}. Складывается из БЖУ продуктов, входящих в состав.
3. Калорийность. Суммарная калорийность продуктов.
4. Список продуктов из имеющихся в базе.
5. Стоимость. Стоимость всех входящих продуктов.

### *Рецепт.*

Редактируется администратором. Пользовательские ингредиенты могут задаваться в локальной базе данных самим пользователем.

1. Блюдо.
2. Состав блюда. Список из имеющихся ингредиентов.
3. Процесс приготовления. Ссылка на страницу в интернете или текстовый формат.

### *Блюдо.*

Редактируется администратором. Пользовательские ингредиенты могут задаваться в локальной базе данных самим пользователем

1. Название.
2. БЖУ. Суммарное БЖУ всех ингредиентов. Словарь формата: {“белки”: число, “жиры”: число, “углеводы”: число}.
3. Калорийность. Суммарная калорийность ингредиентов.
4. Стоимость. Суммарная стоимость всех ингредиентов.
5. Предпочтительное время употребления. ENUM: {“Завтрак”, “Обед”, “Ужин”, “Перекус”}.

Ограничения:

* У каждого блюда должен быть рецепт.

### *Приём пищи.*

Генерируется автоматически в процессе подбора блюд под приёмы пищи, указанные пользователем.

1. Блюдо.
2. Время употребления. ENUM: {“Завтрак”, “Обед”, “Ужин”, “Перекус”}.

### *Модель питания.*

Создаётся автоматически исходя из параметров, заданных пользователем.

1. Название.
2. Дневное БЖУ. Суммарное БЖУ всех блюд в течение дня. Словарь формата: {“белки”: число, “жиры”: число, “углеводы”: число}.
3. Дневные калории. Суммарная калорийность блюд в течение дня.
4. Бюджет на день.

### *Паттерн.*

Генерируется автоматически исходя из данных пользователя, подбирая оптимальный набор блюд. Может быть отредактирован пользователем.

1. Название.
2. Приёмы пищи. Список приёмов пищи, составленный под времена, который пользователь указал при создании паттерна.
3. БЖУ. Суммарное БЖУ всех блюд.
4. Стоимость. Суммарная стоимость всех блюд.
5. Калорийность. Суммарная калорийность.

### *Расписание.*

Создаётся пользователем путём выбора паттернов на каждый день недели.

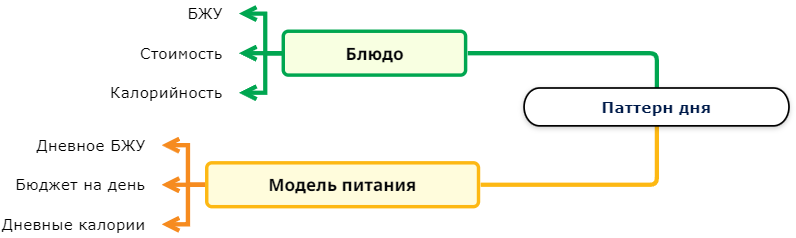
1. Название.
2. Паттерны. Список ссылок на паттерны на каждый день недели.
3. Стоимость. Суммарная стоимость по всем паттернам.

Ограничения:

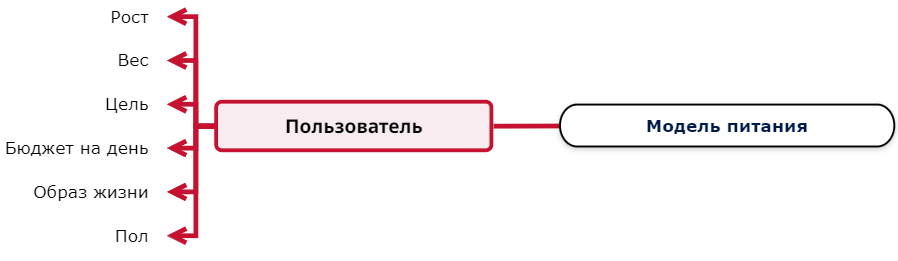
* Каждому паттерну должен соответствовать день недели.

## Связи в расчётах.

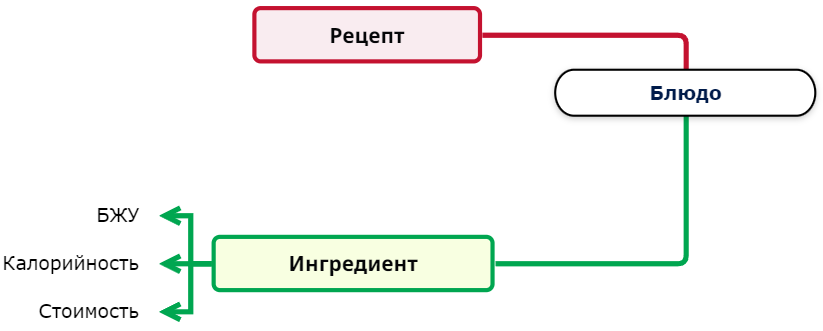
1. Расписание определяется набором паттернов дня по дням недели.
2. Паттерн дня определяется блюдами, а также моделью питания при автоматической генерации. При автогенерации суммарная калорийность, стоимость и суммарное БЖУ не должны превышать значений, указанных в модели питания.



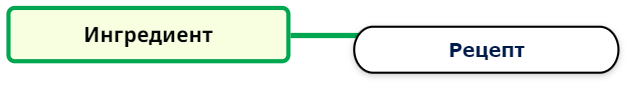
1. Модель питания определяется пользователем.



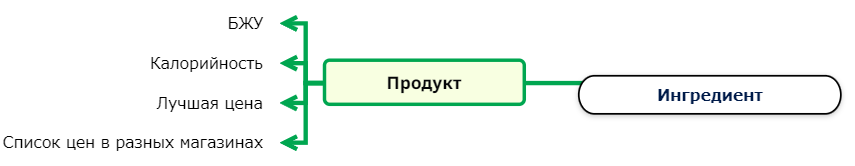
1. Блюдо определяется рецептом и ингредиентами.



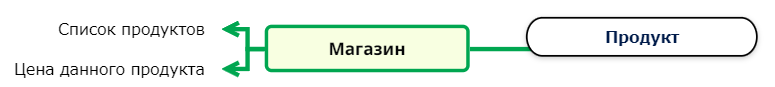
1. Рецепт определяется лишь ингредиентами.



1. Ингредиент определяется продуктами, которыми в него входят. По умолчанию за цену продуктов (для подсчёта стоимости ингредиента) берётся лучшая цена продукта, но можно выбрать другой магазин для покупки с другой ценой.



1. Продукт использует поля магазина. Для подсчёта лучшей цены берётся первая порядковая статистика среди цен во всех магазинах, где есть данный продукт.



## Сценарии использования приложения (Use-cases).

1. Изменение информации о пользователе. Пользователь вводит или меняет свои данные и цель. Происходит расчёт модели питания: вычисляется суммарная стоимость блюд, включенных в день, их БЖУ и калорийность.
2. Составление расписания. Когда пользователь составляет расписание, он выбирает сохранённые паттерны и распределяет их по неделе. Паттерны, которые пользователь убирает из расписания, остаются в разделе сохранённых.
3. Создание паттерна. Если пользователь хочет создать новый паттерн, он сперва указывает, какое количество раз и какие приёмы пищи (Завтрак, Обед, Ужин, Перекус) он собирается делать. Затем пользователь выбирает либо генерацию паттерна, либо делает его вручную. При автогенерации происходит подбор блюд таким образом, чтобы:
   1. Суммарное БЖУ и общая калорийность блюд имели значения около того, что рассчитала модель
   2. Суммарная стоимость не превышала бюджет на день
   3. Блюда соответствовали времени приёма пищи

При ручном создании паттерна, перечисленные выше правила могут нарушаться.

В процессе создания паттерна производиться промежуточный подсчёт суммарных стоимости, БЖУ и калорийности, которые выводятся пользователю.

1. Изменение паттерна. Пользователь удаляет или меняет паттерн. Процесс изменения аналогичен ручному созданию с частично заполненной информацией о паттерне.
2. Просмотр информации пользователем. Пользователь может просматривать всю доступную информацию о паттернах, модели питания, блюдах, ингредиентах, продуктах и магазинах.
3. Выбор магазина. Когда пользователь просматривает информацию о продукте, ему показываются варианты его покупки в разных магазинах. Также предлагается магазин с самой выгодной ценой.
4. Изучение рецепта. Когда пользователь просматривает рецепт происходит открытие текстового файла из базы данных или переход по ссылке в интернете.
5. Изменение локальной базы данных. Пользователь может добавлять/изменять или удалять свои рецепты. В этих случаях в локальной базе пользователя меняются данные о рецепте. Аналогично пользователь может работать с ингредиентами и блюдами.
6. Изменение глобальной базы данных. Администратор может изменять глобальную базу данных, добавляя, изменяя или удаляя рецепты/ингредиенты/продукты или магазины.