

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных технологий

Кафедра проектирования информационных компьютерных систем

Дисциплина "Современные технологии проектирования информационных систем"

К защите допустить:
Руководитель курсовой работы
старший преподаватель
кафедры
_____ А.В.Михалькевич
15.07.2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе
на тему

Автоматизированный помощник правильного питания

БГУИР КР 1-40 05 01-10 № 173 ПЗ

Студент _____ (подпись студента)

Курсовая работа
представлена на проверку
15.07.2025

(подпись студента)

Реферат

БГУИР КР 1-40 05 01-10 № 173 ПЗ, гр. 784371

, Автоматизированный помощник правильного питания, Минск: БГУИР - 2025.

Пояснительная записка 74276 с., 12 рис., 1 табл.

Ключевые слова: ООП, Курсовая работа, КП, Курсовой проект, Современные технологии проектирования информационных систем, СТПИС, диаграммы.

Предмет Современные технологии проектирования информационных систем, А.В.Михалькевич

Автоматизированный помощник правильного питания. Выполнила - Мавчун В.А. Руководитель - Михалькевич А.В. Курсовой проект состоит из пояснительной записки, графической части, кода программного продукта.

Automated nutritional assistant. Completed - Mavchun V.A. Head - Mikhalkevich A.V. The course project consists of an explanatory note, graphic part, software code product.

Содержание

[Введение](#)

[1 Описание предметной области](#)

[2 Описание основных процессов предметной области](#)

[3 Спецификация вариантов использования системы](#)

[4 Информационная модель системы и её описание](#)

[5 Обоснование выбора компонентов и технологий для реализации](#)

[6 Физическая организация и описание разработанных модулей](#)

[7 Модели представления системы и их описание](#)

[8 Руководство по разворачиванию системы](#)

[9 Результаты тестирования разработанной системы, и оценка выполнения задач](#)

[Заключение](#)

[Список использованных источников](#)

[Приложения](#)

Введение

Тема правильного питания очень актуальна, ведь на сегодняшний день так много людей следят за своим здоровьем. Чтобы вести здоровый образ жизни и держать себя в форме необходимо соблюдать все принципы правильного питания, но что, если не все так хорошо в нём осведомлены, а даже если и осведомлены, то расчёт энергетической ценности, а также выбор примерного или точного меню на каждый приём пищи обычному человеку очень сложен, даже из-за такой мелочи, как нехватка времени и правильных знаний. Почему здоровая пища заслуживает столько внимания? Потому что для достижения каких-либо результатов, например, в похудении, одних тренировок недостаточно. Дело в том, что на физические упражнения затрачивается около 5% времени (4-5 часов в неделю), а пища поступает каждый день по несколько раз. Вот почему питание является ключевым фактором в вопросе похудения, набора мышечной массы и здоровья в целом. Основные преимущества здорового питания: - укрепление костей; - укрепление иммунной системы; - снижение стресса; - эффективный вывод токсинов из организма; - улучшение качества сна; - снижение уровня холестерина и вредных жиров. Актуальность здоровой пищи очень важна в современных условиях постоянных

инфекций и заболеваний, вирусов и прочих болезней, которые за крайнее десятилетие получили мощное развитие и выкашивают множество людей по всему миру. Статистика доказывает, что только благодаря хорошему питанию многие болезни минуют человека или же переносятся им гораздо легче. И тут тоже с логической точки зрения всё ясно и понятно — ведь поскольку организм чрезвычайно насыщен различными полезными веществами и ингредиентами, он имеет максимально стойкий иммунитет, и вся его иммунная система прекрасно справляется с любыми неприятностями, даже очень сложными заболеваниями и инфекциями. Кроме того, в голове сложно держать множество полезных рецептов, а тем более их энергетическую ценность. Сложно также и каждый раз подсчитывать количество необходимых калорий для определённой цели и количество калорий, содержащихся в выбранном блюде. А в век информационных технологий запоминать рецепты, вести ежедневник с рационами и потреблёнными калориями неуместно. В таком случае весьма актуальным является использование автоматизированного помощника по правильному питанию, которое позволит не только автоматизировать процесс подбора меню, но и увеличить количество желающих следить за своим питанием в связи с тем, что данный процесс уже не будет требовать значительных затрат, времени и специальных знаний. Именно для этого и необходимо грамотно разработанное программное средство.

1 Описание предметной области

Целью курсовой работы является разработка программного средства, которое поможет автоматизировать процесс составления рациона на день, на основе информации о пользователях, их предпочтениях и целях, о калорийности блюд, продуктов, а также о соотношении белков, жиров и углеводов.

Одна из наиболее распространённых проблем в данное время - это трудности с лишним весом и огромным рядом заболеваний, связанных с ним. Причиной тому - сидячий образ жизни и несбалансированное питание. Таким образом, борьба с лишним весом является задачей каждого конкретного человека. И чтобы с ней бороться, необходимо вести здоровый образ жизни.

Здоровый образ жизни - это ежедневный комплекс мер, который состоит из правильного сбалансированного питания, занятий спортом, отказа от вредных привычек, позитивного и гармоничного взгляда на мир. Все это и является образом и стилем жизни!

И если спорт и вредные привычки - всё просто, то в вопросах правильного питания всегда существовало много противоречий и сложностей, с которыми нелегко справиться в одиночку.

Правильное питание — это питание, обеспечивающее рост, нормальное развитие и жизнедеятельность человека, способствующее укреплению его здоровья и профилактике заболеваний. Другими словами, это главное условие здорового образа жизни.

Система правильного питания - это самая настоящая база, на основе ее легко строится общее здоровое состояние организма, отличное настроение и внешняя привлекательность, заменить которую не в состоянии никакая косметика.

Если придерживаться элементарных правил правильного питания, не будет необходимости ходить к врачам-диетологам и скупать в огромных количествах книги о правильном питании, которые порой только путают и пугают, чтобы накормить себя и своих близких правильной и здоровой едой. А также большинство таких болезней как ожирение, сердечно-сосудистые заболевания, диабет, повышенное давление, рак обойдут стороной, а бодрость духа и энергия

никогда не иссякнут.

Важную роль в правильном питании играет расчёт энергетической ценности блюд и продуктов.

Энергетическая ценность - одна из наиболее важных характеристик продуктов питания, определяющая их пищевую - определяется количеством энергии, получаемой организмом от пищевых компонентов, входящих в потребляемую пищу, зависит от содержания в ней углеводов, жиров, белков и органических кислот.

Вместе с тем, достаточно сложно находить соответствующие рецепты, подбирать необходимые порции и ингредиенты, подходящие для достижения поставленной цели.

2 Описание основных процессов предметной области

3 Спецификация вариантов использования системы

Визуальное моделирование в UML можно представить как некоторый процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели исходной системы к логической, а затем и к физической модели соответствующей программной системы. Для достижения этих целей вначале строится модель в форме так называемой диаграммы вариантов использования (use case diagram), которая описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки.

Разработка диаграммы вариантов использования преследует цели:

- Определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы.
- Сформулировать общие требования к функциональному поведению проектируемой системы.
- Разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей.
- Подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не

говорится о том, каким образом будет реализовано взаимодействие актеров с системой.

Диаграмма вариантов использования представлена на рисунке 3.1.

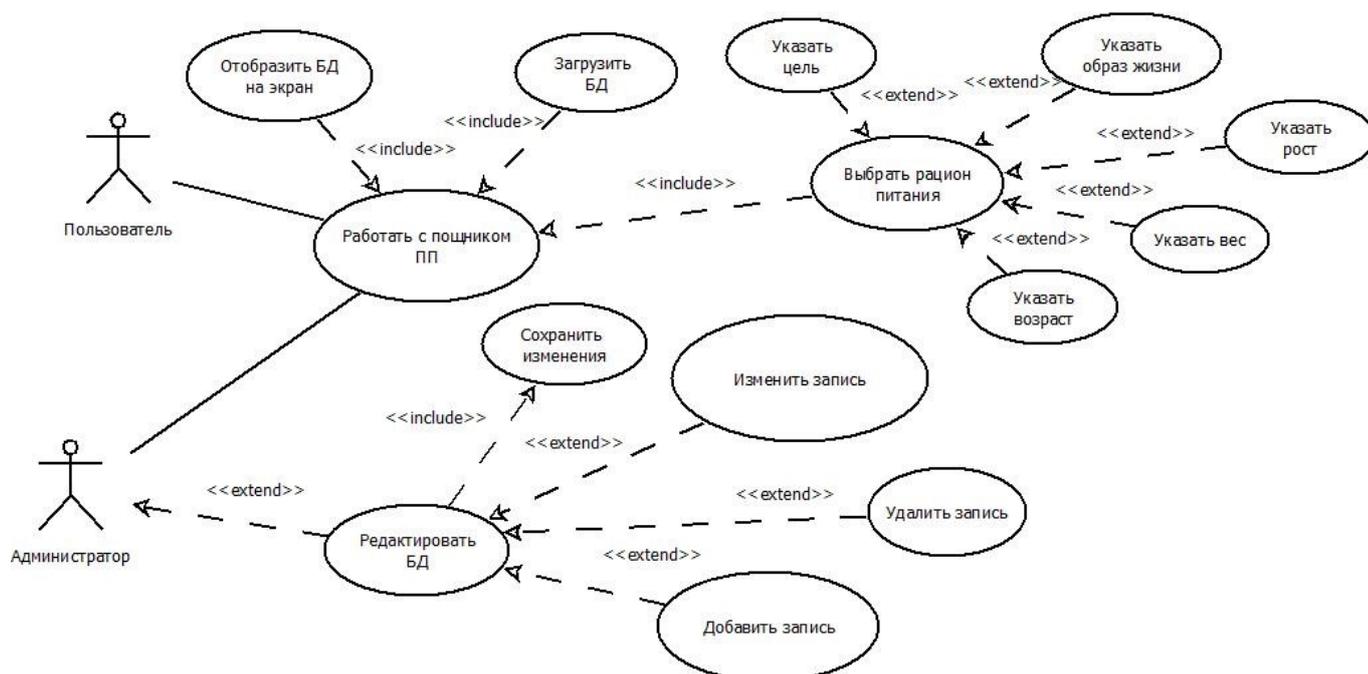


Рисунок 3.1 - Диаграмма вариантов использования

В качестве актера данной диаграммы выступает два субъекта: администратор и пользователь.

Пользователь в данном случае обращается к соответствующему действию «Работать с помощником ПП». Между актером и вариантом использования «Запустить приложение» возникает отношение ассоциации. На следующем этапе разработки данной диаграммы вариант использования «Работать с помощником ПП» может быть уточнен на основе введения в рассмотрение трех дополнительных вариантов использования. Это позволяет выделить такие действия, как «Отобразить БД на экран», «Загрузить БД» и «Выбрать рацион». Вполне очевидно, что указанные действия раскрывают поведение исходного варианта использования в смысле его конкретизации, и поэтому между ними будет иметь место отношение включения. Затем «Выбрать рацион» включает в себя дополнительные варианты использования «Указать цель», «Указать образ жизни», «Указать рост», «Указать Вес», «Указать возраст». Данные варианты использования «Отобразить задания» связан с родительским вариантом использования отношением расширения.

Между администратором, который обращается к действию «Работать с помощником ПП», возникает отношение ассоциации. Отношение расширения связывает администратора с таким действием, как «Редактировать БД». «Редактировать данные» включает в себя варианты использования с отношением включения - «Сохранить изменения», с отношением расширения - «Изменить запись», «Удалить запись», «Добавить запись».

4 Информационная модель системы и её описание

Диаграммы сущность-связь позволяют использовать наглядные графические обозначения для моделирования сущностей и их взаимосвязей и предлагают некоторый набор стандартных обозначений для определения данных и отношений между ними. С помощью этого вида диаграмм можно описать отдельные компоненты концептуальной модели данных и совокупность взаимосвязей между ними, имеющих важное значение для разрабатываемой системы.

Первым шагом является извлечение информации из описания предметной области и выделение сущностей. Сущность – это класс однотипных объектов, информация о которых должна быть учтена в модели.

Исходя из описания предметной области в разрабатываемом программном средстве «Автоматизированный помощник правильно питания» можно выделить следующие сущности: блюдо, виды блюд, рецепт, приём пищи, пользователь, выбор пользователя, комментарии, статьи, как показано на рисунке 4.1.



Рисунок 4.1 - Сущности

Вторым шагом является идентификация связей.

Связь – это некоторая ассоциация между двумя сущностями. Связи позволяют по одной сущности находить другие сущности, связанные с нею.

Связь типа один-к-одному означает, что один экземпляр первой сущности (левой) связан с одним экземпляром второй сущности (правой). Связь один-к-одному чаще всего свидетельствует о том, что на самом деле мы имеем всего одну сущность, неправильно разделенную на две.

Связь типа один-ко-многим означает, что один экземпляр первой сущности (левой) связан с несколькими экземплярами второй сущности (правой). Это наиболее часто используемый тип связи. Левая сущность (со стороны "один") называется родительской, правая (со стороны "многo") – дочерней.

Связь типа много-ко-многим означает, что каждый экземпляр первой сущности может быть

связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Тип связи много-ко-многим является временным типом связи, допустимым на ранних этапах разработки модели. В дальнейшем этот тип связи должен быть заменен двумя связями типа один-ко-многим путем создания промежуточной сущности.

Каждая связь может иметь одну из двух модальностей связи:

- может;
- должен.

Модальность "может" означает, что экземпляр одной сущности может быть связан с одним или несколькими экземплярами другой сущности, а может быть и не связан ни с одним экземпляром.

Модальность "должен" означает, что экземпляр одной сущности обязан быть связан не менее чем с одним экземпляром другой сущности.

Связь может иметь разную модальность с разных концов.

Описанный графический синтаксис позволяет однозначно читать диаграммы, пользуясь следующей схемой построения фраз:

<Каждый экземпляр СУЩНОСТИ 1> <МОДАЛЬНОСТЬ СВЯЗИ> <НАИМЕНОВАНИЕ СВЯЗИ> <ТИП СВЯЗИ> <экземпляр СУЩНОСТИ 2>.

Каждая связь может быть прочитана как слева направо, так и справа налево.

В данном случае, каждый пользователь может производить какое-то количество выборов, а каждый выбор должен принадлежать одному пользователю. Каждый выбор может состоять из нескольких блюд, а каждое блюдо, в свою очередь, может входить в несколько выборов. Так же выбор пользователя обязан содержать приёмы пищи, а приёмы пищи могут относиться к нескольким пользователям. Каждое блюдо должно состоять из рецепта, а каждый рецепт обязан относиться к блюду. Блюдо обязано относиться к виду, а вид блюда, тем самым, может содержать несколько блюд. Пользователь может оставлять комментарии к статье в информационных блоках, а вот комментарий должен принадлежать одному пользователю, а также одной статье, когда статья может содержать много комментариев.

Связывание сущностей представлено на рисунке 4.2.

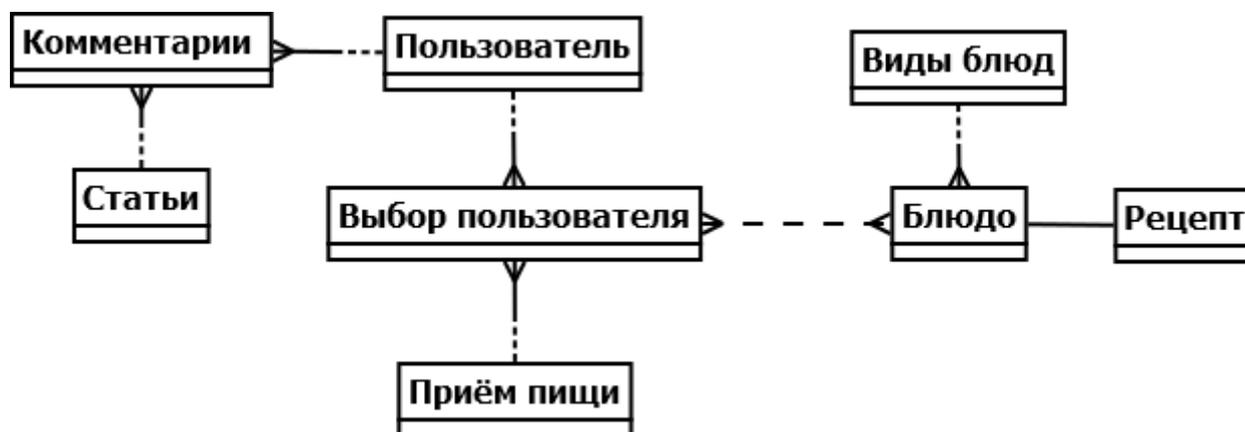


Рисунок 4.2 - Связи

Последним шагом моделирования является идентификация атрибутов.

Атрибут сущности - это именованная характеристика, являющаяся некоторым свойством сущности. Наименование атрибута должно быть выражено существительным в единственном числе (возможно, с характеризующими прилагательными).

Сущность «Блюдо» состоит из следующих атрибутов: код блюда, название, вид блюда, энергетическая ценность, количество. Сущность «Виды блюд» состоит из атрибутов: код вида, название. Сущность «Приём пищи» состоит из атрибутов: код приёма, приём пищи, проценты. Сущность «Рецепт» состоит из: код рецепта, код блюда, рецепт, изображение. Сущность «Пользователь» состоит из следующих атрибутов: код пользователя, имя, фамилия, логин, пароль, вес, рост, возраст, физическая активность, цель. Сущность «Выбор пользователя» состоит из атрибутов: код выбора, код пользователя, код блюда, код приёма пищи, дата выбора. Сущность «Комментарии» состоит из: код комментария, код пользователя, комментарий, код статьи. Сущность «Статьи» состоит из атрибутов: код статьи, текст, изображение, название, описание.

Диаграмма сущность-связь для автоматизированного помощника правильного питания предоставлена на рисунке 4.3.

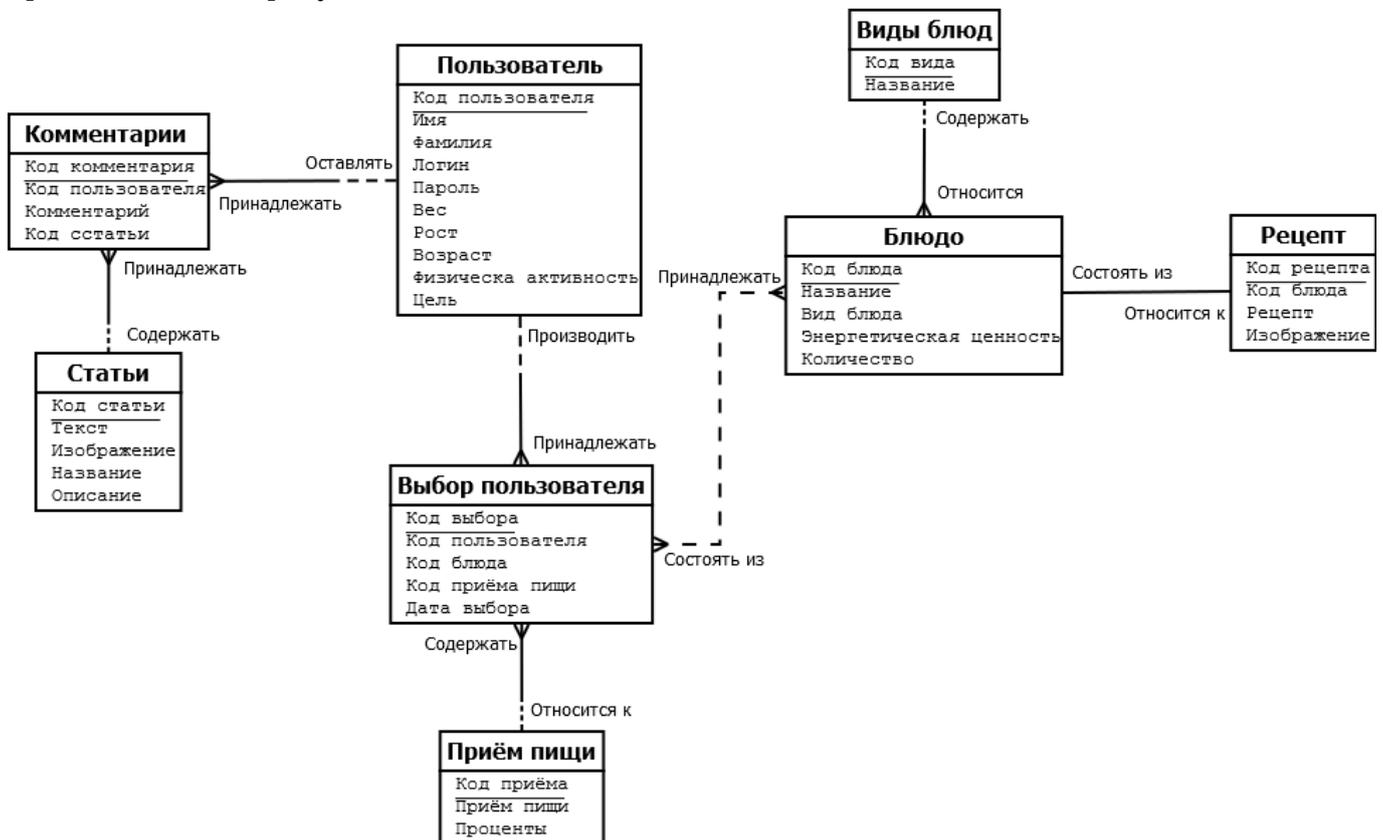


Рисунок 4.3 - Диаграмма сущность-связь

В результате была спроектирована база данных правильного питания. База данных включает в себя информацию энергетической ценности блюд и подбору рецептов индивидуально для каждого.

5 Обоснование выбора компонентов и технологий для реализации

В качестве языка для разработки приложения был выбран язык программирования для создания web-приложений PHP.

PHP — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

Важным преимуществом языка PHP перед такими языками, как языков Perl и C заключается в возможности создания HTML документов с внедренными командами PHP.

Преимущества PHP:

- является свободным программным обеспечением, распространяемым под особой лицензией (PHP license);
- легок в освоении на всех этапах;
- поддерживается большим сообществом пользователей и разработчиков;
- имеет развитую поддержку баз данных;
- имеется огромное количество библиотек и расширений языка;
- может использоваться в изолированной среде;
- предлагает нативные средства организации веб-сессий, программный интерфейс расширений;
- является довольно полной заменой проприетарной среды ASP (Active Server Pages) от Microsoft;
- может быть развёрнут почти на любом сервере;
- портирован под большое количество аппаратных платформ и операционных систем.

Кроме того, PHP может работать на всех основных операционных системах. Скорость в разработке также важна для веб-мастера, так как PHP может отделять HTML код сценария элементов, это значительно сокращает время разработки проектов. Во многих случаях на этапе строительства проекта можно в отдельности создавать HTML код и PHP код, а после объединить их в общий файл. Это не только упрощает жизнь как для программиста, но также устраняет препятствия, которые стоят на пути эффективной и гибкой конструкции.

В области веб-программирования, в частности серверной части, PHP — один из популярных сценарных языков.

А так как необходимо было создать динамическое веб-приложение, то лучше языка, чем PHP не найти.

Для создания реляционной базы данных был использован такой непроецедурный язык, как SQL, в связке с PHP работает прекрасно.

SQL символизирует собой Структурированный Язык Запросов. Это - язык, который дает возможность создавать и работать в реляционных базах данных, являющихся наборами связанной информации, сохраняемой в таблицах.

В нашем все более и более взаимосвязанном компьютерном мире, пользователь снабжённый

таким языком, имеет огромное преимущество в использовании и обобщении информации из ряда источников с помощью большого количества способов.

Такой язык был выбран исходя из следующих его преимуществ:

- независимость от конкретной СУБД. Несмотря на наличие диалектов и различий в синтаксисе, в большинстве своём тексты SQL-запросов, содержащие DDL и DML, могут быть достаточно легко перенесены из одной СУБД в другую. Существуют системы, разработчики которых изначально ориентировались на применение по меньшей мере нескольких СУБД;

- наличие стандартов. Наличие стандартов и набора тестов для выявления совместимости и соответствия конкретной реализации SQL общепринятому стандарту только способствует «стабилизации» языка;

- декларативность. С помощью SQL программист описывает только то, какие данные нужно извлечь или модифицировать. То, каким образом это сделать, решает СУБД непосредственно при обработке SQL-запроса. Однако не стоит думать, что это полностью универсальный принцип — программист описывает набор данных для выборки или модификации, однако ему при этом полезно представлять, как СУБД будет разбирать текст его запроса.

В качестве веб-сервера использовался Apache.

Apache - Web-сервер с открытым исходным кодом.

Основные достоинства Apache- надёжность, безопасность и гибкость настройки. Apache позволяет подключать различные модули, добавляющие в него новые возможности - например, можно подключить модуль, обеспечивающий поддержку PHP или любого другого Web-ориентированного языка программирования.

Поддержка PHP для Apache и MySQL вызвало дальнейший рост его популярности. Те, кто знаком с Apache уже могли заметить, что в настоящее время это наиболее часто используемый веб-сервер в мире.

Так вот сочетание Apache, PHP и MySQL, позволяют создавать полнофункциональные веб-приложения.

Для создания интерфейса были выбраны HTML и CSS (каскадные таблицы стилей).

HyperText Markup Language — «язык гипертекстовой разметки») — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML).

Преимущества:

- сохраняет информацию о выбранных автором шрифтах, кодировках, стилях, цвете текста;
- читается на различных компьютерных платформах;
- многие программы (в их числе бесплатные) умеют с ним работать;
- это открытый стандарт;

Недостатки:

- графика хранится отдельно;
- часто представляет из себя не отдельный файл, а целую группу файлов;
- занимает больше места на диске, чем простые текстовые файлы.

Стили являются удобным, практичным и эффективным инструментом при вёрстке веб-страниц и оформления текста, ссылок, изображений и других элементов.

Каскадные таблицы стилей были выбраны исходя из ряда преимуществ:

- CSS позволяет сохранить время. Можно, единожды создав стиль CSS, использовать его со множеством веб-страниц;

- страницы загружаются быстрее;

- изменение дизайна веб-страниц становится проще – единственное что нужно сделать крупное изменение – нужно изменить один стиль, и это изменение затронет все элементы, использующие этот стиль, на веб-странице автоматически;

- больше возможностей – в CSS гораздо больше стелевых атрибутов по сравнению с HTML, поэтому с помощью CSS можно создать куда более привлекательный дизайн в отличии от простого HTML;

- поддержка различных устройств – таблицы стилей позволяют оптимизировать внешний вид контента страницы к любому типу устройств. Используя один и тот же HTML документ, можно предоставлять различные версии дизайна веб-сайта для различных устройств таких как PDA (карманных компьютеров), смартфонов, планшетов.

В качестве текстового редактора для написания текста программы был выбран Notepad++.

Notepad++ — свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса большого количества языков программирования и разметки. Поддерживает открытие более 100 форматов.

Основные преимущества и возможности данного редактора:

- полная бесплатность;

- работа с большим количеством вкладок одновременно;

- легкость установки и минимальный размер установочного файла;

- подсветка синтаксиса;

- работа с большим количеством языков и форматов файлов;

- возможность экспорта файла в другой формат;

- использование различных шрифтов для вывода фрагментов кода определенного типа;

- наличие горячих клавиш, позволяющих получить быстрый доступ к основным опциям и возможностям редактора (копировать, вставить, удалить, сохранить, закрыть, создать и так далее);

- возможность расширенного поиска информации в файле;

- возможность смены кодировки файла;

- возможность расширения функционала.

Многие вебмастера, верстальщики и программисты используют в своей работе такую программу, как Notepad++. Прежде всего, это продвинутый текстовый редактор, обладающий большим количеством опций и возможностей. Исходя из всего этого он и был выбран.

6 Физическая организация и описание разработанных модулей

Физическая структура сайта описывает размещение файлов и папок на диске компьютера. Правильная организация файлов и папок файловой системы помогают в дальнейшем держать

все в порядке.

Данное программное средство состоит из довольно большого количества файлов, каждый из которых выполняет свою функцию. В корневой папке `pp`, изображённой на рисунке 6.1, располагаются файлы `php`, создающие серверную логику и хранящие все функции, которые необходимы пользователю для работы с данным программным средством.

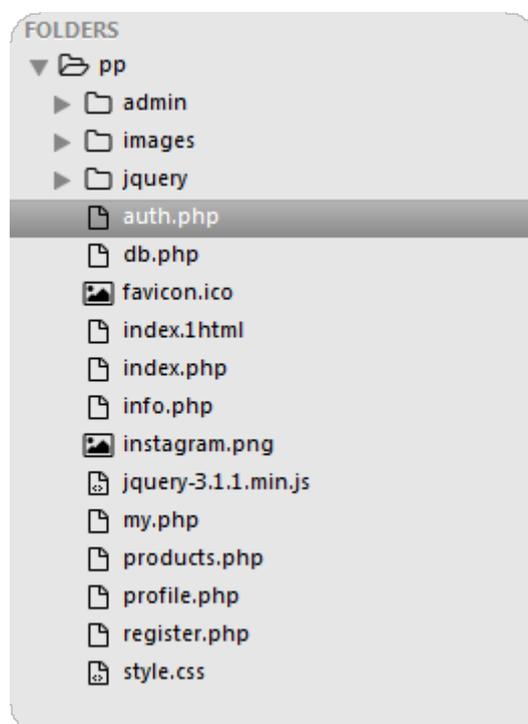


Рисунок 6.1 – Корневой каталог

Также там расположены файлы стилей `css`. Данные файлы предназначены для оформления страницы, запрошенной пользователем. Благодаря этим файлам реализован адаптивный дизайн программного средства, что дает возможность его использования на мобильных устройствах.

В папке `admin` содержатся файлы `php`, в которых реализуются функции администрирования сайта. Благодаря им администратор имеет право добавлять, удалять и редактировать данные. А также пользоваться сайтом как обычный пользователь.

В папке `image` располагаются все изображения сайта, которые включают в себя изображения рецептов, главные изображения на фоне сайта.

В папке `jquery` и `js` находятся скрипты к разным элементам дизайна сайта. С их помощью можно добиться приятного интерфейса, который будет привлекать пользователей.

В папке `menu` содержатся все стилевые файлы и изображения необходимые для оформления меню сайта.

Для большего удобства была создана папка `inf`, в которой располагается файл `article.php`, который несёт в себе информацию информационных блоков и все стилевые файлы для него.

Физическая структура ресурса разрабатывается, исходя из удобства размещения файлов. Однако более или менее точное сохранение порядка следования логических разделов в физической структуре сайта позволяет избежать путаницы при последующем дополнении и

обновлении материалов.

Модульное программирование — это организация программы как совокупности небольших независимых блоков, называемых модулями, структура и поведение которых подчиняются определенным правилам. Использование модульного программирования позволяет упростить тестирование программы и обнаружение ошибок. Аппаратно-зависимые подзадачи могут быть строго отделены от других подзадач, что улучшает мобильность создаваемых программ.

Модуль — функционально законченный фрагмент программы.

Принцип модульности является средством упрощения задачи проектирования ПС и распределения процесса разработки ПС между группами разработчиков. При разбиении ПС на модули для каждого модуля указывается реализуемая им функциональность, а также связи с другими модулями. Удобство использования модульной архитектуры заключается в возможности обновления (замены) модуля, без необходимости изменения остальной системы.

В данном программном средстве реализованы следующие модули:

- db.php - обеспечивает соединение с базой данной с помощью драйвера Mysql;
- auth.php - содержит пользовательские функции для авторизации. Пользователь имеет возможность войти в свой профиль, введя логин и пароль. Если профиля не существует, об этом сообщается. В случае, если пользователь не зарегистрирован ему нужно подключиться к следующему модулю register.php, который содержит пользовательские функции для регистрации;

- register.php - содержит пользовательские функции для регистрации. Пользователь может зарегистрироваться, введя свой новый логин и пароль для входа в профиль. Данные будут занесены и сохранены в базу;

- profile.php - отображает страницу с актуальными данными пользователя. На странице находится информация с личными данными. В неё входит выбранный вес пользователя, его рост, возраст, образ жизни (количество занятий физкультурой в неделю), выбранная цель (похудение, поддержание веса, набор мышечной массы), а также необходимое количество потребления калорий в день. Данные можно изменять и сохранять заново.

- my.php - реализует класс пользователя и передает актуальную модель клиенту. Один из главных модулей, в нём учитывается выбор меню пользователя относительно каждого приёма пищи (завтрака, обеда, ужина, одного из перекусов) и отображается уже в едином целом, как предоставление рациона на день, уже с посчитанными граммами и рецептами каждого из блюд;

- products.php - отображает список всех продуктов из базы данных и их параметры. Продукты отображаются в виде таблицы, удобно и понятно для пользователя;

- index.php - содержит функции для администрирования. После его подключения администратор получит html страницу, на которой будут доступны для работы все таблицы со всем содержимым. Ему будут доступны базовые функции для работы с базой данных: добавление новых рецептов или продуктов, их редактирование и удаление. Также работа с информационными блоками сайта;

- information.php - содержит все информационные блоки на сайте, хранит комментарии к данным статьям.

Программный код часто разбивается на несколько файлов, каждый из которых компилируется отдельно от остальных. Такая модульность программного кода позволяет значительно уменьшить время перекомпиляции при изменениях, вносимых лишь в небольшое количество исходных файлов, и упрощает групповую разработку.

7 Модели представления системы и их описание

Алгоритм представлен в виде диаграммы деятельности.

Диаграмма деятельности — UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого. Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

Диаграмма представлена на рисунке 7.1.

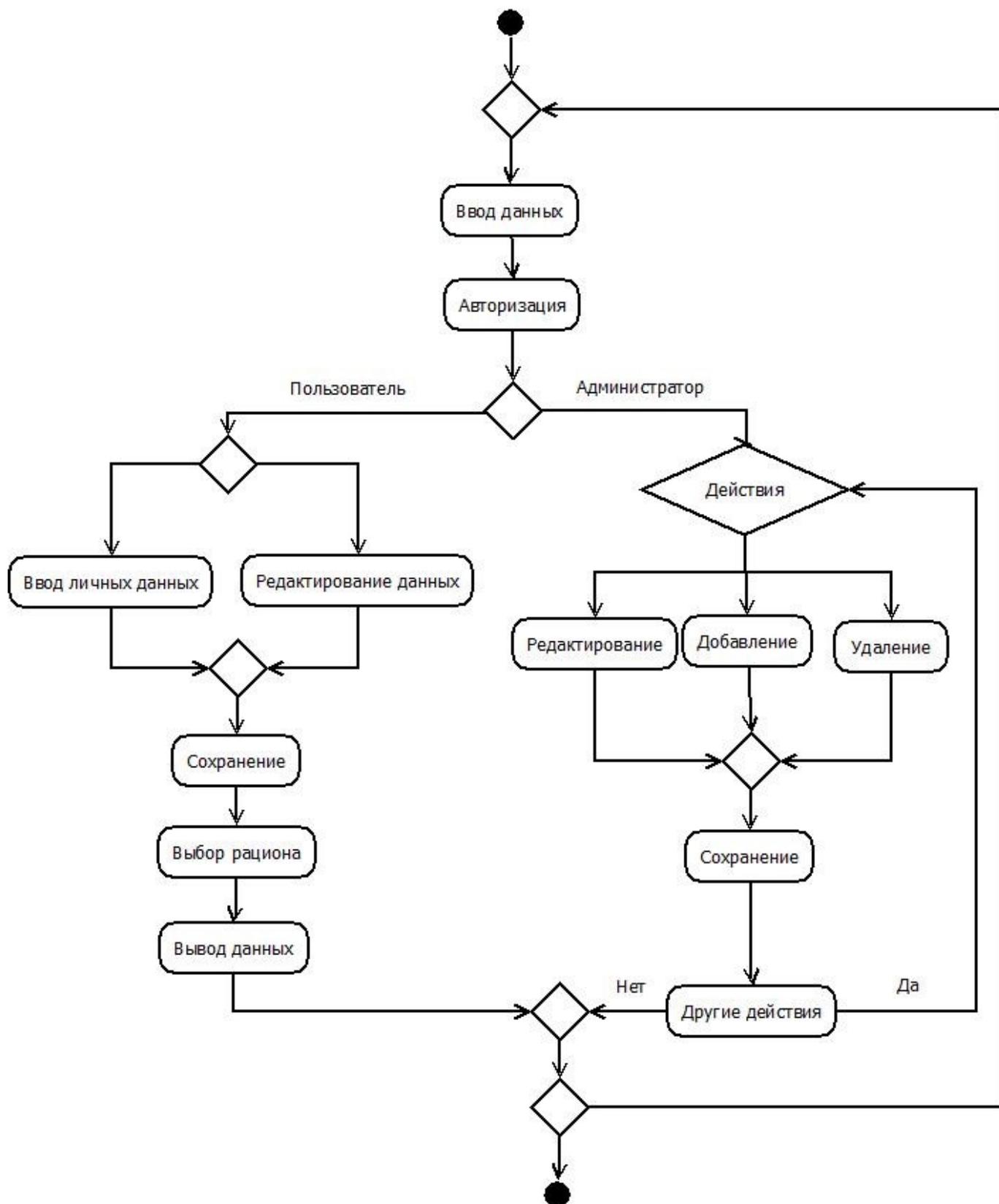


Рисунок 7.1 - Диаграмма деятельности

При запуске программного средства осуществляется вход либо регистрация, если пользователя ещё нет в базе. Вход может быть от имени администратора, который в свою очередь имеет возможность редактировать данные в базе, удалять их либо добавлять новые. И от пользователя. Пользователь в свою очередь при входе должен ввести свои личные данные, в которые входят вес, рост, возраст, образ жизни и стоящая перед ним цель. Исходя из этого программа рассчитает дневную норму потребления калорий и выведет выбор блюд для

определённого приёма пищи из которых, пользователь должен будет выбрать нужное ему. После выбора всех приёмов и блюд, программное средство покажет всё меню и количество допустимых грамм для употребления каждого из них на каждый приём пищи исходя их процентного соотношения, необходимого для каждого из них. Пользователь также сможет редактировать в любое время свои данные и сохранять изменения.

Схема работы программы представлена на рисунке 7.2.

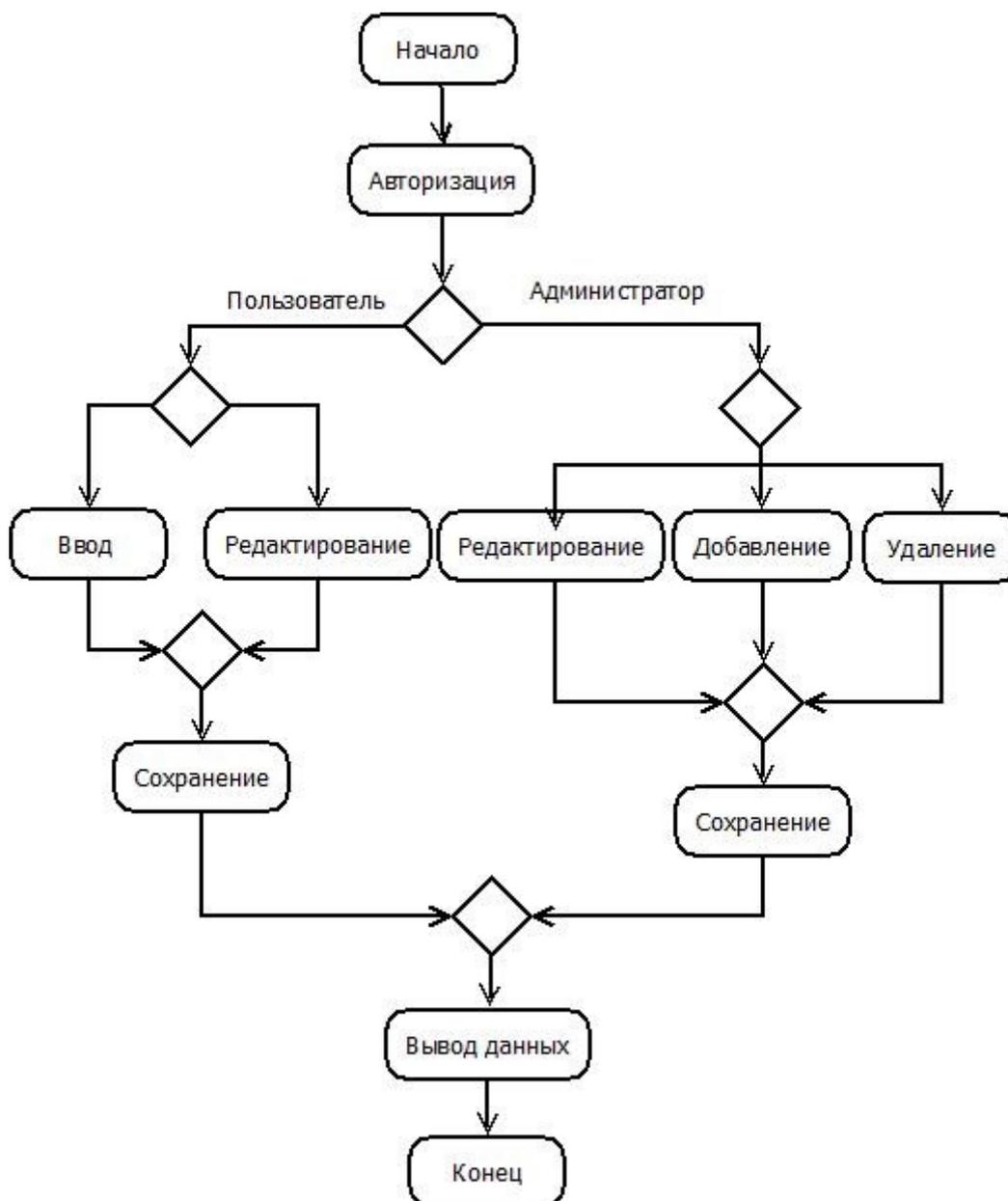


Рисунок 7.2 - Блок-схема

Рацион хранится в базе ровно одни сутки, после наступления нового дня пользователь снова может выбирать себе меню

8 Руководство по развертыванию системы

Программное средство «Автоматизированный помощник правильного питания» предназначен для быстрого подсчёта дневного потребления калорий и выбора рациона на

день.

Интерфейс ПС удобен и понятен в использовании. Хорошо подобраны цвета, ссылки и другие компоненты страниц хорошо структурированы, нет ничего лишнего.

При загрузке сайта, первым делом открывается страница, где необходимо войти в свой личный профиль либо зарегистрироваться, иначе дальнейшей работы с сайтом не может быть. Стартовая страница с авторизацией представлена на рисунке 8.1.

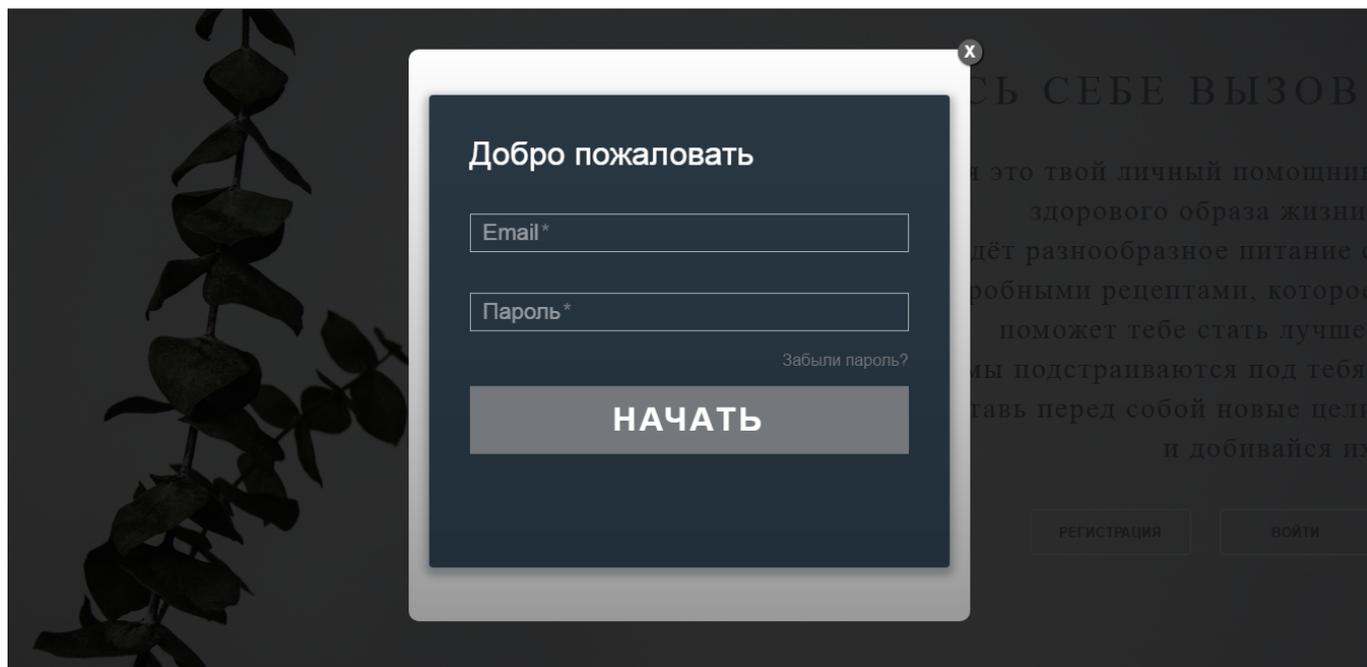


Рисунок 8.1 – Стартовая страница

После регистрации и входа в свой профиль необходимо ввести личные данные: рост, вес, возраст, стиль жизни и свою цель, после чего подтвердить информацию, рисунок 8.2. После чего программа подсчитает пользователю его дневную норму потребления энергетической и пищевой ценности, исходя из которой будет строиться весь дальнейший рацион. Данные можно будет отредактировать в любой момент.

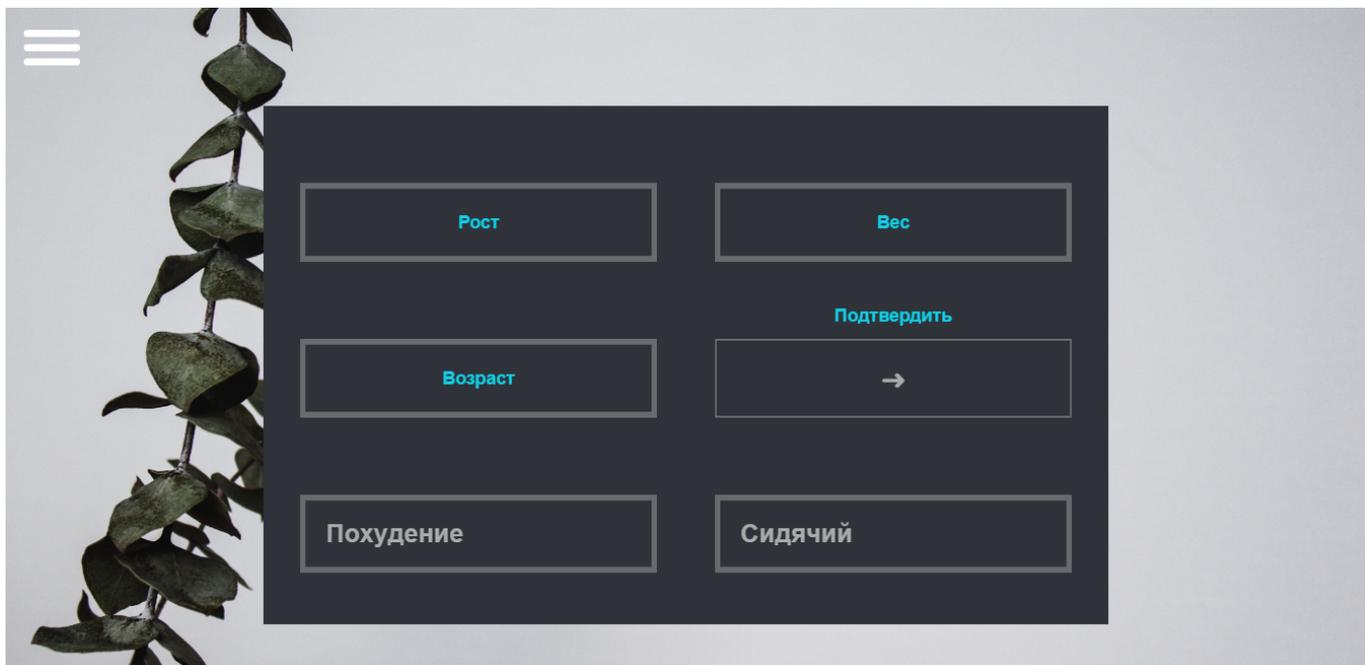


Рисунок 8.2 – Личные данные

При входе на главную страницу, рисунок 8.3, пользователю предоставляется выбор любых предложенных блюд для каждого приёма пищи с указанием энергетической и пищевой ценности в ста граммах. Отмечая желаемое блюдо и нажимая кнопку «Выбрать» сайта переходит к последующему приёму, записывая при этом выбор в базу данных.

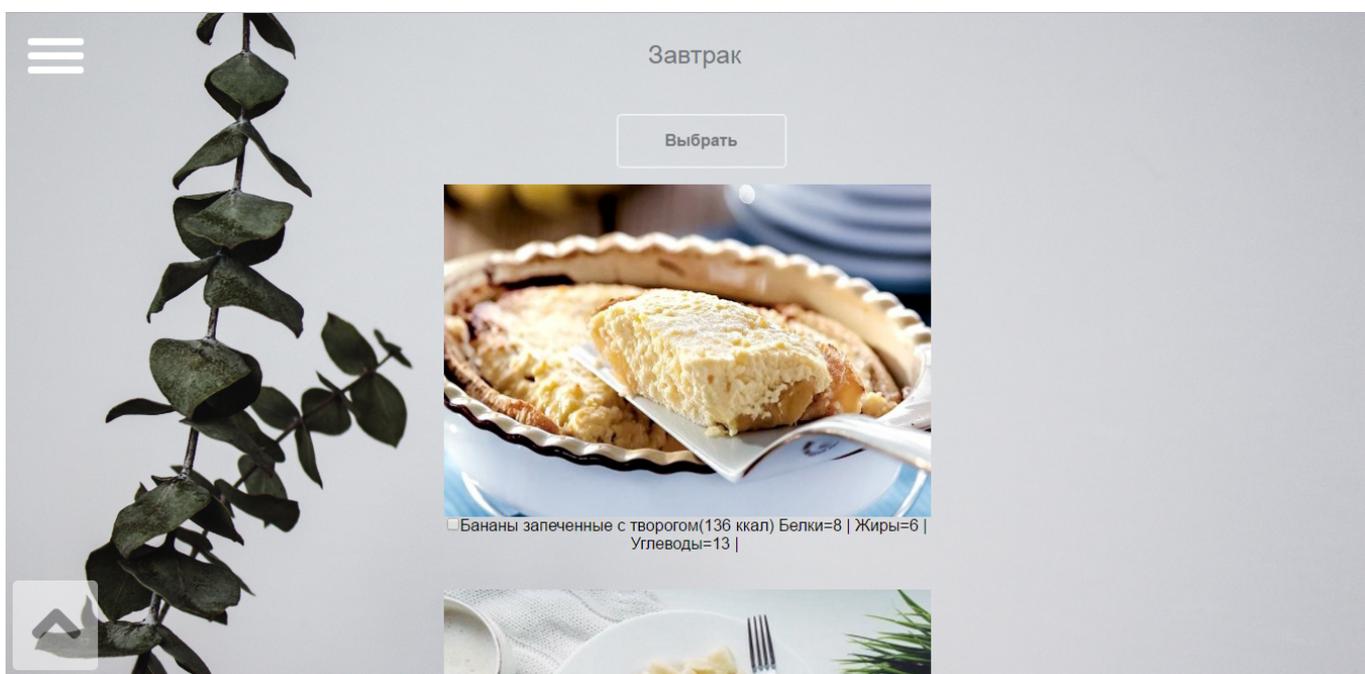


Рисунок 8.3 – Главная страница

Уже после всего выбора блюд, выводится всё меню, где указано количество потребления калорий на каждый приём и количество граммов каждого выбранного блюда, для достижения поставленной цели пользователем. При нажатии на сформированное меню появляется новое диалоговое окно с рецептами и изображением блюд.

Пользователям будут доступны статьи с интересной и полезной информацией, представленные ввиду блоков с кратким описанием, рисунок 8.4.

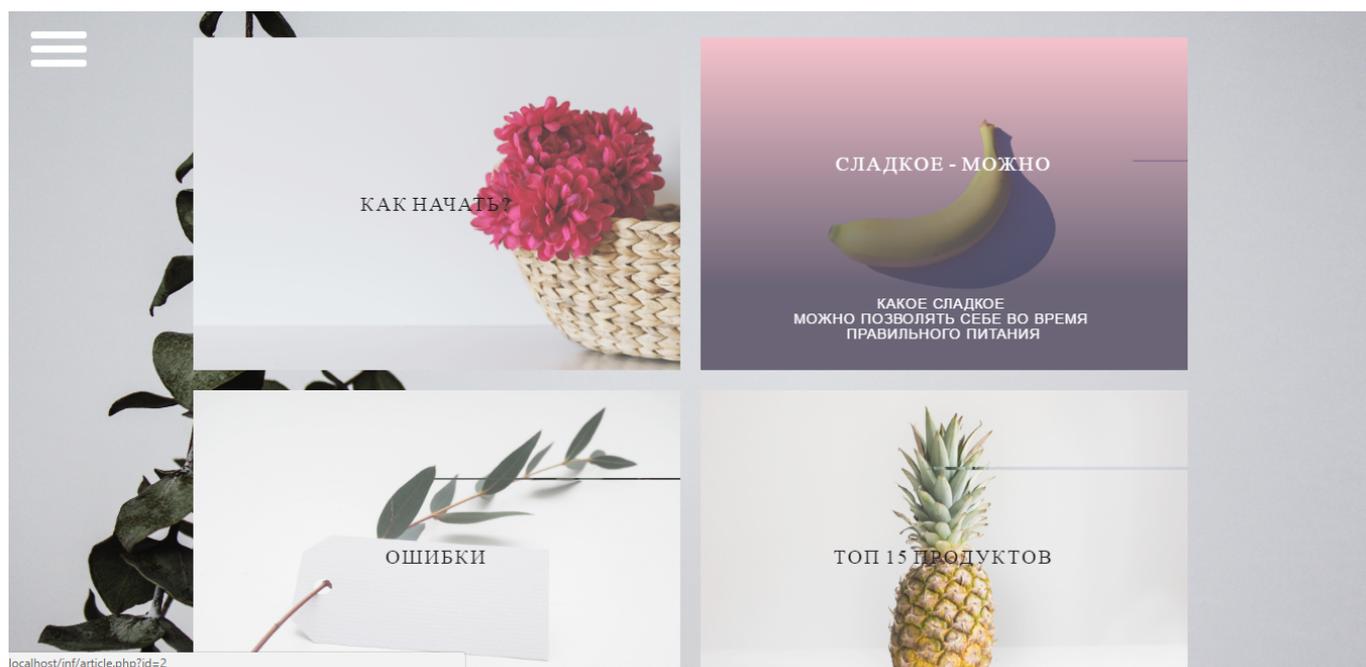


Рисунок 8.4 – Информационные блоки

При входе в любую статью у пользователя будет возможность ознакомиться с интересными фактами, а также оставлять свой комментарий на странице и общаться с другими пользователями, рисунок 8.5

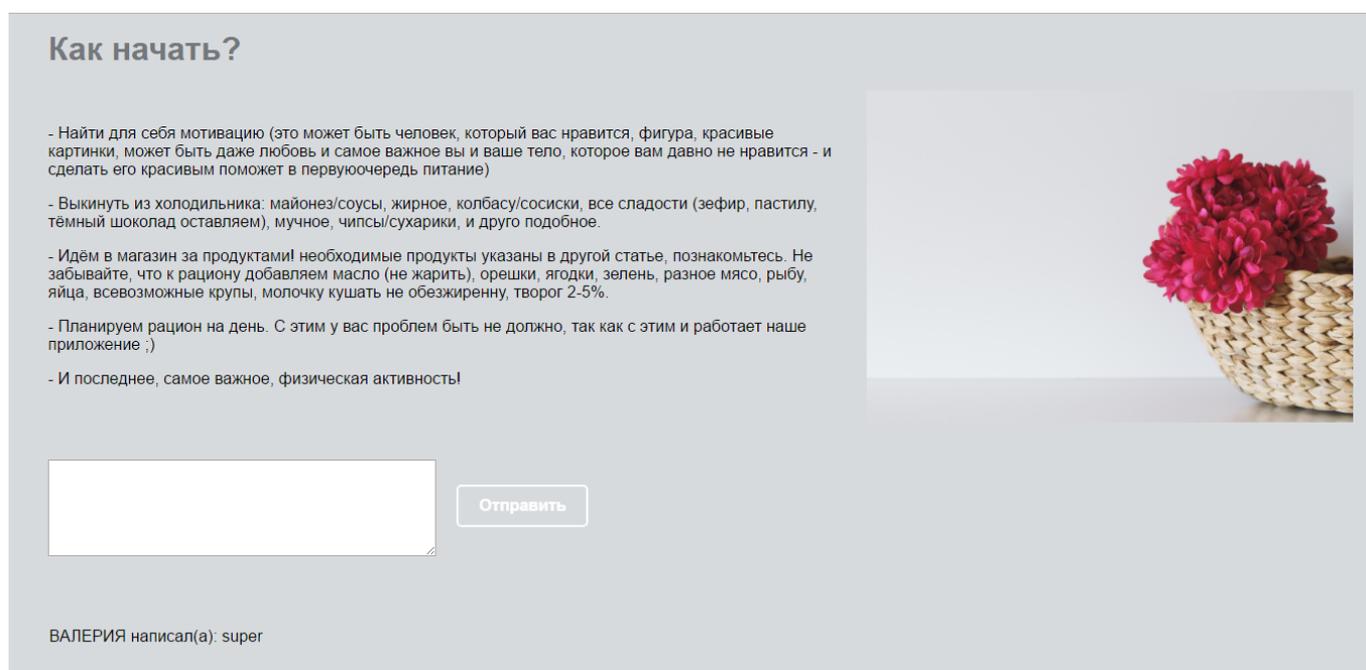


Рисунок 8.5 – Статья

У сайта разработан собственный логотип, который располагается вверху страницы профиля. Удобные переходы по ссылкам, красивая навигация и правильно подобранные картинки

создают сайту определённый стиль и привлекают любых пользователей.

9 Результаты тестирования разработанной системы, и оценка выполнения задач

Отладка ПС - это деятельность, направленная на обнаружение и исправление ошибок в ПС с использованием процессов выполнения программ.

Тестирование программного средства - это процесс выполнения программ на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ, это любая деятельность, направленная на обнаружение ошибок в программном продукте.

Указанный набор данных называется тестовым или просто тестом.

Процесс отладки включает:

- действия, направленные на выявление ошибок (тестирование);
- диагностику и локализацию ошибок (определение характера ошибок и их местонахождение);
- внесение исправлений в программу с целью устранения ошибок.

Успех отладки ПС в значительной степени предопределяет рациональная организация тестирования. При отладке ПС отыскиваются и устраняются, в основном, те ошибки, наличие которых в ПС устанавливается при тестировании.

Нельзя гарантировать, что тестированием ПС практически выполнимым набором тестов можно установить наличие каждой имеющейся в ПС ошибки. Поэтому возникает две задачи. Первая задача: подготовить такой набор тестов и применить к ним ПС, чтобы обнаружить в нем по возможности большее число ошибок. Однако, чем дольше продолжается процесс тестирования (и отладки в целом), тем большей становится стоимость ПС. Отсюда вторая задача: определить момент окончания отладки ПС (или отдельной его компоненты). Признаком возможности окончания отладки является полнота охвата пропущенными через ПС тестами (т.е. тестами, к которым применено ПС) множества различных ситуаций, возникающих при выполнении программ, и относительно редкое проявление ошибок в ПС на последнем отрезке процесса тестирования. Последнее определяется в соответствии с требуемой степенью надежности ПС, указанной в спецификации его качества.

Существует несколько признаков, по которым принято производить классификацию видов тестирования. Обычно выделяют следующие:

- по объекту тестирования;
- по знанию системы;
- по степени автоматизации;
- по степени изолированности;
- по времени проведения тестирования;
- по признаку позитивности сценариев;
- по степени подготовленности к тестированию.

Все виды тестирования программного обеспечения, в зависимости от преследуемых целей,

можно условно разделить на следующие группы:

- функциональные;
- нефункциональные;
- связанные с изменениями.

Функциональные тесты базируются на функциях и особенностях, а также взаимодействии с другими системами, и могут быть представлены на всех уровнях тестирования: компонентном или модульном, интеграционном, системном и приемочном.

Функциональные требования включают в себя:

- функциональная пригодность;
- точность;
- способность к взаимодействию;
- соответствие стандартам и правилам;
- защищённость.

Функциональные виды тестирования рассматривают внешнее поведение системы. Одни из самых распространенных видов функциональных тестов:

- функциональное тестирование;
- тестирование безопасности;
- тестирование взаимодействия.

В процессе разработки данного программного средства также было проведено функциональное тестирование. Результаты тестирования представлены в таблице 9.1

Таблица 9.1 - Результаты тестирования

Действие	Ожидаемый результат	Полученный результат	Принятые меры
Регистрация пользователя	Пользователь зарегистрирован	Ошибка при регистрации	Проверка правильности ввода
Авторизация пользователя	Пользователь авторизован	Пользователь авторизован успешно	
Ввод личной информации	Отображение и сохранение личных данных пользователя	Данные сохранены	
Изменение личной информации	Информация изменена	Информация изменена	
Выбор рациона	Отображение рациона на экран	Рацион не отображён	Исправлена ошибка отображения базы данных
Отображение рецептов	Отображение рецептов на экран	Рецепты отображены	
Добавление статей	Новая статья добавлена	Статья добавлена	
Удаление данных	Данные удалены	Данные не удалены	Исправлена ошибка написания запроса

В ходе тестирования были выявлены следующие ошибки:

- некоторые поля не были защищены от ввода некорректных данных;
- неправильное подключение к базе данных;
- неправильный синтаксис написания запроса.

Основная цель выделения отладки и тестирования как отдельных этапов создания

программы заключается в том, чтобы обратить внимание обязательности обеих стадий и на необходимость специального планирования временных затрат по каждой из них в отдельности.

Очевидно, что процесс управления тестированием ПО затрагивает все этапы жизненного цикла разработки. Он подразумевает сравнение действительного состояния продукта и того состояния, которое было запланировано и задокументировано в плане тестирования продукта. Процесс тестирования, анализа и мониторинга помогает спланировать и изменить последующие задачи наилучшим путем.

Таким образом, ошибки, выявленные на этапе тестирования, были устранены, а ПС «Автоматизированный помощник правильного питания» работает в соответствии с заданным алгоритмом.

Заключение

Данный автоматизированный помощник позволит обеспечить быстрое составление рациона на текущий день, сохраняя при этом данные пользователя в базе. Это и является главным преимуществом автоматизированного помощника правильного питания. В программе будет представлено полное разнообразие рациона, сбалансированная методика подбора блюд, никаких жёстких диет, теория и база знаний, сохранение рациона в базе, возможность выбора цели (похудение, поддержание веса, набор мышечной массы), удобство интерфейса, а самое главное то, что программа будет совершенно бесплатной. Автоматизированный помощник правильного питания позволяет помочь пользователям в выборе рациона на день, подсчёта энергетической ценности и соотношения белков, жиров и углеводов в выбранных блюдах, а также продуктах, узнать о основных принципах правильного питания и показывает, что не так сложно этому соответствовать. Кроме того, была написана документация к пользованию разработанным приложением, содержащая информацию о методах веб-интерфейса, их параметрах, назначении и виде возвращаемых данных. Подробно были рассмотрены задачи программы, их решение, основные функции, гарантирующие удобство работы с данным программным средством и алгоритм работы программы.

Список использованных источников

1. [печатное издание] **1. Зервас, К. Web 2.0: создание приложений на PHP / К. Зервас - М.: «Вильямс», 2009. - 544 с.**
2. [печатное издание] **2. Костарев, А. Ф. PHP 5 / А. Ф. Костарев - СПб.: «БХВ-Петербург», 2008. - 1104 с.**
3. [печатное издание] **3. Кузнецов, М. Р., Симдянов, И. В. Самоучитель PHP 5/6. / М. Р. Кузнецов, И. В. Симдянов - СПб.: «БХВ-Петербург», 2009. - 672 с.**
4. [печатное издание] **4. Кузнецов, М. Р., Симдянов, И. В. PHP. Практика создания Web-сайтов. / М. Р. Кузнецов, И. В. Симдянов - СПб.: «БХВ-Петербург», 2008.**
5. [печатное издание] **5. Монастырский, К. О. Основы абсолютного здоровья и долголетия. Функциональное питание. / К.О. Монастырский - М.: 2012.**
6. [печатное издание] **Тейлор, А. SQL для чайников, 8-е издание / А. Тейлор - М.: «Диалектика», 2014. - 416 с.**
7. [печатное издание] **Тутельян, В.А. Справочник по диетологии. / В. А. Тутелян - М.: Медицина, 2002. - 544 с.**

Приложения

1. [Задание] **Задание** [5ed0eb4361c9a_Лист задания.docx](#)
2. [Приложение] **Приложение** [5ed0f17b1c188_Приложение А \(Листинг кода\).docx](#)