

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**ИСПОЛЬЗОВАНИЕ SPRING FRAMEWORK И ANDROID STUDIO
ДЛЯ РАЗРАБОТКИ ПРИЛОЖЕНИЙ ПО СОЗДАНИЮ
ИНДИВИДУАЛЬНЫХ РАЦИОНОВ ПИТАНИЯ
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Семенова Егора Алексеевича

Научный руководитель:

Старший преподаватель

кафедры ИиП

Лапшева Е.Е.

подпись, дата

Зав. кафедрой ИиП,

к.ф.-м.н., доцент

Огнева М.В.

подпись, дата

Саратов 2022

ВВЕДЕНИЕ

В нынешнее время — в век технологий и инноваций, потребность общества в удобстве и комфорте растёт с каждым годом. У каждого человека есть мобильный телефон, с помощью которого можно не только коммуницировать с обществом, но и оптимизировать работу или обучение. Доказано, что каждый четвертый пользователь смартфона проводит с ним более семи часов в день.

Использование веб-сайтов теряет свою актуальность, ведь на смену им пришли мобильные приложения. Их удобства заключаются в том, что вся информация отобрана под конкретного пользователя, что ускоряет процесс получения данных.

Актуальность настоящей работы подтверждается тем, что в современных реалиях людям тяжело следить за собой и правильно питаться, в результате чего большая часть населения страдает лишним весом. Благо во время информационных технологий любой процесс можно автоматизировать с помощью цифровых методик. Автоматизация выбора рациона питания может избавить человека от траты лишнего времени на подбор индивидуального питания и выбора блюда на прием пищи. Это поможет сэкономить не только драгоценное время, но и укрепить здоровье и помочь людям с ожирением избавиться от проблем с лишним весом.

Цель бакалаврской работы: написание мобильного приложения для подбора индивидуального рациона питания и REST API приложения, которое будет брать на себя большую часть нагрузки и хранить в себе всю базу данных.

Для достижения поставленной цели следует выполнить следующие задачи:

- найти оптимальный алгоритм для индивидуального расчета калорий;
- сформировать инструментарий (язык программирования, среды разработки и дополнительные библиотеки);
- выбрать подходящую Базу данных и заполнить ее данными;
- разработать RESTful приложение;

- разработать мобильное приложение;
- сравнить функционал разработанного приложения с уже существующими в интернете.

Методологические основы Spring framework и использования инструментов языка Java представлены в работах К. Волса, Б. Эккеля, А. Дэвиса.

Практическая значимость бакалаврской работы состоит в том, что разработанные приложения могут быть использованы как альтернатива существующим мобильным приложениям по подбору рационов питания, так как они удобны в использовании и имеют функционал, отсутствующий у конкурентов.

Структура и объём работы. Бакалаврская работа состоит из введения, двух разделов, заключения, списка использованных источников и семи приложений. Общий объём работы – 79 страниц, из них 40 страниц – основное содержание, включая 6 рисунков и 1 таблицу, список использованных источников информации – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел « Основные понятия и использованные технологии» посвящен выбору и описанию использованных технологий для создания REstful и андроид приложений

В пунктах 1.1 — 1.3 приводятся необходимые определения веб-приложения и HTTP протокола, по которому происходит общение с ними.

Веб-приложение — клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера.

HTTP-запрос — это способ, с помощью которого платформы интернет-коммуникаций, такие как веб-браузеры, запрашивают информацию, необходимую им для загрузки веб-сайта.

REST или Representational state transfer, что переводится как передача состояния представления — это архитектурный стиль проектирования API с использованием протокола HTTP.

REST — это набор архитектурных ограничений, а не протокол или стандарт. Разработчики API могут реализовывать REST различными способами.

В пункте 1.4 рассматривается язык программирования Java и такие среды разработки как IntelliJ Idea и Android Studio.

В пункте 1.5 приводится обзор Spring Framework, служащий каркасом для написания веб-приложений.

Spring Framework — это платформа Java, которая обеспечивает всестороннюю поддержку инфраструктуры для разработки приложений Java. Spring управляет инфраструктурой, поэтому разработчик может сосредоточиться на своем приложении.

Spring Framework состоит из нескольких частей, организованных примерно в 20 модулей. Эти модули сгруппированы в Core Container, Data Access/Integration, Web, AOP(Aspect Oriented Programming), Instrumentation и Test.

В пункте 1.6 рассматривается Apache Maven фреймворк.

Apache Maven предназначен для автоматизации процесса сборки проектов на основе описания их структуры в файле на языке POM (Project Object Model), который является подмножеством формата XML. Maven позволяет выполнять компиляцию кодов, создавать дистрибутив программы, архивные файлы jar/war и генерировать документацию. Простые проекты Maven может собрать в командной строке. Название программы Maven вышло из языка идиш, смысл которого можно выразить как «собиратель знания».

В пункте 1.7 рассматривается база данных PostgreSQL.

PostgreSQL — это популярная свободная объектно-реляционная система управления базами данных. PostgreSQL базируется на языке SQL и поддерживает многочисленные возможности.

В пункте 1.8 рассматривается кодировка base64, служащая для кодировки изображений в текстовый формат.

Base64 — стандарт кодирования двоичных данных при помощи только 64 символов ASCII. Алфавит кодирования содержит латинские символы A-Z, a-z, цифры 0-9 (всего 62 знака) и 2 дополнительных символа, зависящих от системы реализации. Каждые 3 исходных байта кодируются 4-мя символами.

В пункте 1.9 рассматривается Hibernate ORM фреймворк.

Hibernate — библиотека для языка программирования Java, предназначенная для решения задач объектно-реляционного отображения (ORM), самая популярная реализация спецификации JPA (Java Persistence API).

Позволяет сократить объемы низкоуровневого программирования при работе с реляционными базами данных; может использоваться как в процессе проектирования системы классов и таблиц «с нуля», так и для работы с уже существующей базой.

В пункте 1.10 рассматривается библиотека Lombok.

Lombok работает путем подключения к процессу сборки и автоматической генерации байт-кода Java в файлах .class в соответствии с рядом аннотаций проекта, которые вводятся в написанный код.

В пункте 1.11 рассматривается библиотека Volley.

Volley — это HTTP-библиотека, которая делает работу в сети очень простой и быстрой для Android приложений. Она была представлена во время Google I/O 2013 и разработана Google из-за отсутствия в Android SDK сетевого класса, способного работать, не мешая работе пользователя. Хотя Volley является частью проекта Android с открытым исходным кодом, в январе 2017 года Google объявил, что Volley переместится в отдельную библиотеку. Он управляет обработкой и кэшированием сетевых запросов и экономит драгоценное время разработчиков от повторного написания одного и того же кода сетевого вызова/кеша. Volley не подходит для больших операций загрузки или потоковой передачи, так как он удерживает все ответы в памяти во время синтаксического анализа.

Итоги. Изученный в рамках 1 раздела инструментарий позволяет в полной мере решить задачи разработки RESTful приложения и разработки андроид приложения.

Второй раздел «Создание приложений» посвящен реализации RESTful приложения на основе Spring Framework, которое на основе данных о пользователе с помощью HTTP запросов генерирует индивидуальный рацион питания; и андроид приложения, которое использует API предыдущего и предоставляет удобный интерфейс пользователю.

Проект REST приложения использует четыре основных пакета: controller, entity, repository и service. Пакет controller содержит класс, отвечающий за единственный эндпоинт, который принимает данные пользователя и возвращает ему рецепты. Пакет entity содержит набор всех необходимых классов для хранения и передачи данных такие как рецепты, продукты или данные пользователя. Пакет repository представляют собой паттерн Repository. Это слой

абстракции, инкапсулирующий в себе всё, что относится к способу хранения данных. Пакет `service` содержит в себе логику приложения и призван разделять ее от слоя данных.

Сервис `RecipeBuilderService` генерирует на основе данных о пользователе рацион питания на всю неделю по формуле Харриса-Бенедикта.

Андроид приложение использует такие пакеты, как:

- `Manifest`, который хранит в себе единственный файл `AndroidManifest.xml`, где хранятся сведения о всех страницах андроид приложения и некоторые его настройки и разрешения, такие как разрешение на доступ в интернет.
- `Model`, который содержит такие классы и перечислимые типы, как `Activity`, `Gender`, `Recipe`, `RecipeType`, `UserData` и `Week`. Каждый из этих классов соответствует одноименному классу пакет `Entity` ранее описанного REST API приложения, за исключением перечислимых типов.
- `Util`, который имеет лишь один утилитарный класс `RationUtil`, содержащий два метода: **`getProductsText`** — служит для того, чтобы на основе коллекции `Map`, которая хранит в себе элементы в паре ключ - значение, создавать строку, удобную для чтения, которая хранит в себе наименование продуктов и их точные граммовки; **`createNewActivity`** — создает новую страницу и открывает ее на экране мобильного телефона пользователя.
- `Activity`, который содержит классы, унаследованные от `AppCompatActivity`.
- `Layout`, который является стандартным пакетом `Android Studio` приложений и создается автоматически при создании проекта.

В пункте 2.4.3 было проведено сравнение разработанного андроид приложения с существующими реализациями в интернете. Для анализа были

выбраны 3 приложения: Lifesum, Правильное питание, Диеты - правильное питание.

Все выбранные для анализа приложения платные, что является их основным недостатком, потому что не каждый пользователь имеет возможность их приобрести. Помимо этого, выбранные для анализа приложения не имеют такой функционал, как разработанное в ходе работы приложение, по созданию списка продуктов на неделю, с которым удобно ходить в магазин и покупать все необходимое. Создание данного списка можно выделить как основное достоинство разработанного приложения.

Из данного анализа также можно выделить, что разработанное мобильное приложение является лишь оболочкой над RESTful приложением, которое хранит в себе всю базу данных, не занимая лишнее место на смартфоне пользователя, а также берет на себя некоторую работу, что является непосредственно его достоинством. Также RESTful приложением может пользоваться кто угодно с любого устройства с выходом в интернет с использованием HTTP-запросов. Иными словам можно сказать, что любой разработчик сможет воспользоваться API для создания рационов, а также создавать свои приложения или сервисы, которые будут с ним взаимодействовать.

Итоги. В рамках третьего раздела разработаны веб-приложение и андроид приложения, которое использует HTTP запросы для общения с предыдущим. Они обладают преимуществами и недостатками, также было проведено сравнение с существующими андроид приложениями.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы была достигнута основная цель, а именно - реализовано RESTful приложение по созданию рационов питания и андроид приложение, которое использует API предыдущего.

При достижении поставленной цели были выполнены следующие задачи:

- использована формула Харриса-Бенедикта для индивидуального расчета калорий;
- выбран объектно-ориентированный язык программирования Java;
- изучены такие технологии, как Spring Framework, Apache Maven, Apache Freemarker, Hibernate, Lombok, Volley;
- проведено ознакомление и заполнения базы данных PostgreSQL;
- проведен анализ схожих мобильных приложений, в результате которого были выявлены достоинства и план развития разработанного мобильного приложения.

Разработанные приложения помогают экономить время и усилия, а также поддерживают здоровье, предоставляя удобный интерфейс по созданию рецептов на основе индивидуальных параметров пользователей.

Проект разработанного REST API приложения был развернут на локальной машине, что исключает возможности опробовать его всем пользователям интернета. Такие дополнения, как возможность добавлять новые рецепты администраторам приложения, а также убирать из рациона некоторые продукты, на которые у пользователя может быть аллергия или неусваиваемость, могли бы повысить качество данной программы.

Основные источники информации

1. Что такое объектно-ориентированное программирование? [Электронный ресурс]. — URL: <https://timeweb.com/ru/community/articles/obektno-orientirovannoe-programmirovanie> (Дата обращения 18.05.2022)
2. Зацепление, связанность и связность в объектно-ориентированном программировании [Электронный ресурс]. — URL: <https://intellect.icu/zatseplenie-zavisimost-i-svyazannost-svyaznost-v-obektno-orientirovannom-programmirovanii-oop-7010> (Дата обращения 18.05.2022)
3. Martin R. Design Principles and Design Patterns / Robert C. Martin // Object Mentor. — 2000. — Vol. 2 — Pp. 4-16.
4. Мартин Р. Чистый код. Создание анализ и рефакторинг / Р. Мартин; пер. с англ. — СПб.: Питер, 2020 — 464 с.: ил. — (Серия «Библиотека программиста»)
5. Эккель Б. Философия Java. 4-е изд / Б. Эккель. — СПб.: Питер, 2015. — 1168 с.: ил. — (Серия «Java»)
6. Uzayr S. Mastering Java / Sufyan bin Uzayr. — Boca Raton, CRC Press, 2022. — 518 p.
7. Downey T. Guide to Web Development with Java. 2nd edn. / T. Downey. — Miami, Springer, 2022. — 517 p.