

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет им. Н.И.
Лобачевского»

Институт информационных технологий, математики и механики
Кафедра Математического обеспечения и суперкомпьютерных технологий
Направление подготовки «Инженерия программного обеспечения»

ОТЧЁТ
по учебной практике на тему
**«Разработка программно-аппаратного комплекса для мониторинга показателей сердца
человека»**

Выполнил:
студент группы 382008-1
Булгаков Д.Э.
Подпись

Проверил:
к.т.н., доц.
Борисов Н.А.
Подпись

Нижний Новгород
2024

Содержание

1. Введение	3
2. Постановка задачи.	4
3. Проведенная работа.	5
3.1 Стек технологий.	5
3.2 Сценарии использования.	7
3.3 Разработка и реализация архитектуры.	8
4. Заключение.	9
5. Список литературы.	10
6. Приложение.	11

1. Введение

Заболевания сердца и сосудов, ставшие ведущей причиной смертности по всему миру, находятся в центре внимания медицинского сообщества. Ишемическая болезнь сердца, артериальная гипертензия и другие патологии сердечно-сосудистой системы требуют серьезного подхода к диагностике и профилактике.

Сердечно-сосудистая система, сложная в своей структуре и функциональности, подвергается различным воздействиям, которые могут привести к серьезным нарушениям. В этом контексте регулярные обследования приобретают ключевое значение. Они не только предоставляют возможность выявления начальных стадий заболеваний до появления явных симптомов, но и открывают перспективы для раннего вмешательства и эффективной профилактики. Такой подход становится неотъемлемой частью стратегии поддержания здоровья сердечно-сосудистой системы в условиях современного образа жизни.

2. Постановка задачи.

1. Выбрать стек технологий, на основе которых будет написан back-end сервера.

- Определить язык программирования для сервера.
- Определить какую библиотеку использовать для написания сервера.
- Выбрать подходящую базу данных.

2. Определить сценарии использования сервера.

Сценарии использования разделены на следующие платформы:

- Web сайт на основе Vue.js
- Electron приложение.

Общие сценарии можно обобщить и выделить в отдельную компоненту.

3. Разработать архитектуру сервера.

4. Реализовать архитектуру.

3. Проведенная работа.

3.1. Стек технологий.

Перед началом проектирования архитектуры сервера, требуется выбрать язык программирования и фреймворк. На рассмотрении были следующие варианты:

- Node.js с фреймворком Express.js
- Golang с фреймворками Gin или Martini
- Python с фреймворками Flask или Django

Необходимо сравнить основные характеристики каждого решения, такие как производительность, удобство и простота разработки, доступность библиотек и инструментов, а также сообщество и поддержка. Прежде всего, было решено остановиться на языке программирования Python по следующим причинам:

- **Простота и читаемость кода:**

Python известен своим чистым и выразительным синтаксисом, который делает код легким для чтения, понимания и поддержки.

- **Широкие возможности для веб-разработки:**

Python имеет обширную экосистему библиотек и фреймворков для веб-разработки. Flask, Django, FastAPI - все они предоставляют мощные инструменты для создания веб-приложений любого уровня сложности.

- **Кросс-платформенность и портативность:**

Python работает на множестве операционных систем, что обеспечивает гибкость и портативность разработки. Это означает, что код, написанный на Python, может быть легко перенесен с одной платформы на другую без необходимости внесения значительных изменений.

- **Богатая стандартная библиотека:**

Python поставляется с обширной стандартной библиотекой, которая включает в себя множество модулей для работы с сетью, обработки данных, взаимодействия с базами данных и многое другое.

Осталось определиться с фреймворком Python. Было решено использовать Flask, т.к. сервер является учебным, небольшим проектом, он не требует таких глобальных и серьезных инструментов, которые предоставляет Django. Также Flask обладает более интуитивно понятным интерфейсом, что позволит меньше времени потратить на изучение инструмента.

Также необходимо выбрать базу данных, которая лучше всего подойдет к выбранному фреймворку. На рассмотрении были следующие варианты:

- PostgreSQL
- SQLite
- MySQL
- MongoDB
- Redis

По итогу, было решено использовать PostgreSQL из-за его надежности, расширяемости, а также доступности. Кроме того, PostgreSQL активно развивается сообществом и имеет обширную документацию, что делает его привлекательным выбором для учебных проектов.

3.2. Сценарии использования.

3.3. Разработка и реализация архитектуры.

4. Заключение.

5. Список литературы.

6. Приложение.