

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

Институт информационных технологий, математики и механики

Кафедра математического обеспечения и суперкомпьютерных технологий

Направление подготовки: «Программная инженерия»
Профиль подготовки: «Разработка программно-информационных систем»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

на тему
“Разработка программно-аппаратного комплекса для мониторинга показателей сердца человека”

Выполнил: студент группы
382008-1 Булгаков Даниил Эдуардович
Подпись

Научный руководитель:
доцент кафедры МОСТ, к.т.н.,
Борисов Николай Анатольевич
Подпись

Нижний Новгород
2024

Содержание

1. Введение	3
2. Описание предметной области	4
2.1 ЭКГ	4
2.1.1 Стандартные отведения	4
2.1.2 Усиленные отведения	5
2.1.3 Грудные отведения	6
2.1.4 Вывод	7
2.2 Подходы к мониторингу сердечной активности	8
3. Разработка проектного решения	9
3.1 Разработка аппаратной части комплекса (Hardware Level)	12
3.1.1 Выбор микроконтроллера	12
3.1.2 Выбор датчика для снятия ЭКГ	13
3.1.3 Тестирование CJMCU-333	14
3.1.4 Тестирование AD8232	15
3.1.5 Разработка модели	16
3.2 Разработка программной части комплекса (Software level)	21
3.2.1 Разработка требований к программному продукту (ПП)	21
3.2.2 Разработка программной части для модуля сбора данных на базе ESP32	22
3.2.3 Приложение для передачи данных (Application Layer)	25
3.2.4 Сервер для сбора, анализа и хранения данных (Server Layer)	26
3.2.5 База данных (Data Storage Layer)	32
3.2.6 Web Layer (Веб-приложение)	33
4. Практическая апробация и внедрение	35
5. Список литературы	36
6. Приложение	37

1. Введение

В последние десятилетия наблюдается значительный рост числа заболеваний сердечно-сосудистой системы, что делает мониторинг состояния сердца важной задачей в области медицины. Одним из наиболее распространенных методов диагностики и наблюдения за состоянием сердца является электрокардиография (ЭКГ). ЭКГ представляет собой графическую запись электрической активности сердца, которая позволяет выявлять различные аномалии, такие как аритмии, ишемия, инфаркты и другие патологии. Данный метод широко применяется благодаря своей информативности, неинвазивности и доступности.

Несмотря на то, что традиционные стационарные системы ЭКГ являются высокоэффективными, их использование ограничено условиями медицинских учреждений. Пациенты, особенно те, кто страдает хроническими заболеваниями, нуждаются в постоянном мониторинге сердечной активности, что затруднительно в условиях стационара. В этой связи актуальной становится разработка портативных систем для непрерывного мониторинга показателей сердца в повседневной жизни.

Цель данной дипломной работы заключается в разработке программно-аппаратного комплекса для мониторинга показателей сердца человека. Комплекс включает в себя модуль для снятия ЭКГ, приложение для передачи данных с модуля на сервер сбора, анализа и хранения данных.

2. Описание предметной области

2.1. ЭКГ

Электрокардиография — методика регистрации и исследования электрических полей, образующихся при работе сердца. Для получения значения разности потенциалов на участке тела человека используются электроды. Набор электродов, расположенных в определенных местах, формирует различные виды отведений:

- Стандартные отведения
- Усиленные отведения
- Грудные отведения

2.1.1. Стандартные отведения

Стандартные отведения регистрируют разность потенциалов между конечностями человека. Для получения данного типа отведений требуется три электрода: положительный, отрицательный и заземление. Так, правая и левая пара электродов руки образуют первое стандартное отведение - I, электроды правой руки и левой ноги – второе - II, третье отведение III - левая рука и левая нога. Третий электрод используется как заземление (Рис. 1).

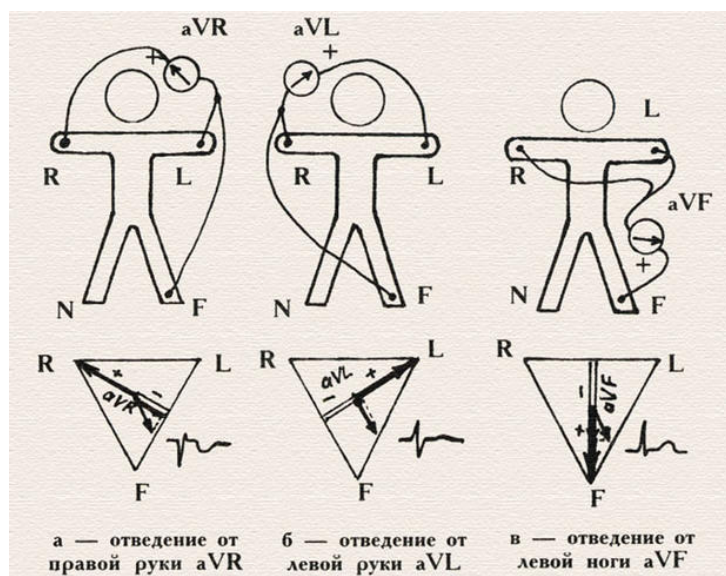


Рис. 1: Стандартные отведения.

Тогда для получения кардиограммы достаточно вычислить разность потенциалов между указанными сигналами.

Отведение	Вычисление
1-ое отведение	LA-RA
2-ое отведение	LL-RA
3-е отведение	LL-LA

LA - левая рука, RA - правая рука, LL - левая нога.

Нетрудно заметить, что в случае, когда нам требуется получить значения сразу по трем отведениям, то аппаратно потребуются считывать только два из них, так как третье можно вычислить путем сложения/вычитания двух других, к примеру:

$$1\text{-ое отведение} + 3\text{-е отведение} = 2\text{-е отведение}$$

Данные отведения позволяют регистрировать следующие типы заболеваний:

- Ишемия миокарда (недостаточное поступление кислорода в сердечную мышцу). Это может проявляться в виде изменений в зубцах ST и T, а также снижения амплитуды зубцов.
- Аритмии, такие как фибрилляция предсердий или желудочковые экстрасистолы. Это может проявляться в виде изменений в ритме и частоте сердечных сокращений, а также в форме зубцов на ЭКГ.
- Блокады проводимости сердца, такие как блокада правой ножки пучка Гиса. Это может проявляться в виде изменений в продолжительности и форме зубцов на ЭКГ.

2.1.2. Усиленные отведения

Усиленные отведения по принципу очень схожи со стандартными отведениями, для них также требуется три электрода. Однако они регистрируют разность потенциалов между одной из конечностей, на которой помещён активный положительный электрод данного отведения и суммарный электродом двух других конечностей. Существуют три таких отведения:

- aVR - усиленное отведение правой руки
- aVL - усиленное отведение левой руки
- aVF - усиленное отведение левой ноги

Для вычисления можно использовать как сигналы с конечностей, так и значения стандартных отведений, используя таблицу.

Отведение	Вычисление	Аналог
aVR	$RA - 0.5 \cdot (LA + LL)$	$-0.5 \cdot (I + II)$
aVL	$LA - 0.5 \cdot (LL + RA)$	$0.5 \cdot (I - III)$
aVF	$LL - 0.5 \cdot (LA + RA)$	$0.5 \cdot (II + III)$

LA - левая рука, RA - правая рука, LL - левая нога

I, II, III - типы стандартных отведений

Данные отведения используются для оценки электрической активности сердца в переднезаднем направлении. Они могут помочь выявить инфаркт миокарда.

2.1.3. Грудные отведения

Грудные отведения регистрируют разницу потенциалов между позитивным электродом, установленным в определённой точке грудной клетки (всего их 6) и единым для остальных пяти электродом Вильсона, потенциал которого равняется нулю (Рис. 2).

Данные отведения используются для оценки электрической активности сердца в горизонтальной плоскости. Они могут помочь выявить заболевания миокарда, такие как инфаркт миокарда и аномалии развития сердца.

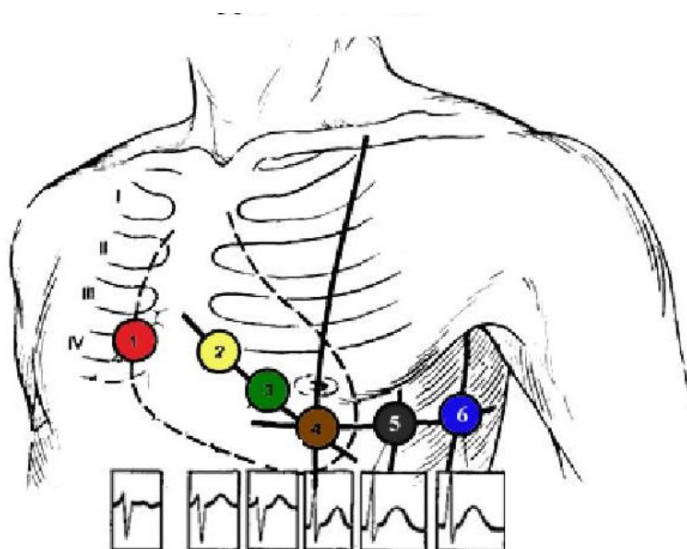


Рис. 2: Грудные отведения.

2.1.4. Вывод

Таким образом, регистрация двух отведений ЭКГ, будь то стандартные или усиленные, является ключевым этапом в диагностике сердечных заболеваний. Этот метод позволяет получить обширную информацию о состоянии сердца пациента, включая анализ электрической активности в различных направлениях.

2.2. Подходы к мониторингу сердечной активности

В настоящее время существуют различные подходы к мониторингу сердечной активности, включая:

1. Традиционные стационарные ЭКГ системы:

Высокая точность и надежность, но требуют нахождения в клинике.

2. Портативные ЭКГ устройства:

Легкие и удобные в использовании, но часто имеют ограниченные возможности по длительности работы и качеству связи.

3. Носимые устройства (например, смарт-часы с функцией ЭКГ):

Удобны для повседневного использования, но часто менее точны и имеют ограничения по функциональности.

Выбор портативных ЭКГ устройств обусловлен стремлением к сочетанию высокой точности измерений с максимальной мобильностью и удобством использования, что в конечном итоге позволит обеспечить непрерывный и эффективный мониторинг сердечной активности пациента в реальном времени для своевременного реагирования на изменения состояния здоровья пациента.

3. Разработка проектного решения

Архитектура системы играет ключевую роль в обеспечении надежного и эффективного функционирования программно-аппаратного комплекса для мониторинга показателей сердца. При её проектировании было принято решение использовать принцип многослойной архитектуры. Это решение было обосновано следующими причинами:

1. Разделение ответственностей:

Многослойная архитектура позволяет четко разделить компоненты системы на логические уровни, каждый из которых отвечает за свои специфические функции. Например, уровень сбора данных отвечает за получение и обработку информации от датчиков, уровень приложения - за визуализацию и интеракцию с пользователем, а уровень сервера - за обработку и хранение данных.

2. Повторное использование кода:

Благодаря модульной структуре многослойной архитектуры компоненты системы становятся более независимыми и переиспользуемыми. Это позволяет избежать дублирования кода и обеспечить более эффективную разработку и поддержку программного продукта.

3. Гибкость и масштабируемость:

Многослойная архитектура обеспечивает гибкость в изменении и модификации системы. Новые функциональные возможности могут быть легко добавлены или изменены без влияния на другие компоненты. Кроме того, такая архитектура легко масштабируется при необходимости увеличения производительности или добавления новых узлов.

4. Улучшенная поддержка и тестирование:

Четкое разделение компонентов упрощает процесс поддержки и тестирования системы. Каждый уровень может быть протестирован отдельно, что позволяет выявлять и устранять ошибки на ранних этапах разработки.

Архитектура проекта разделена на пять основных слоев (Рис. 3):

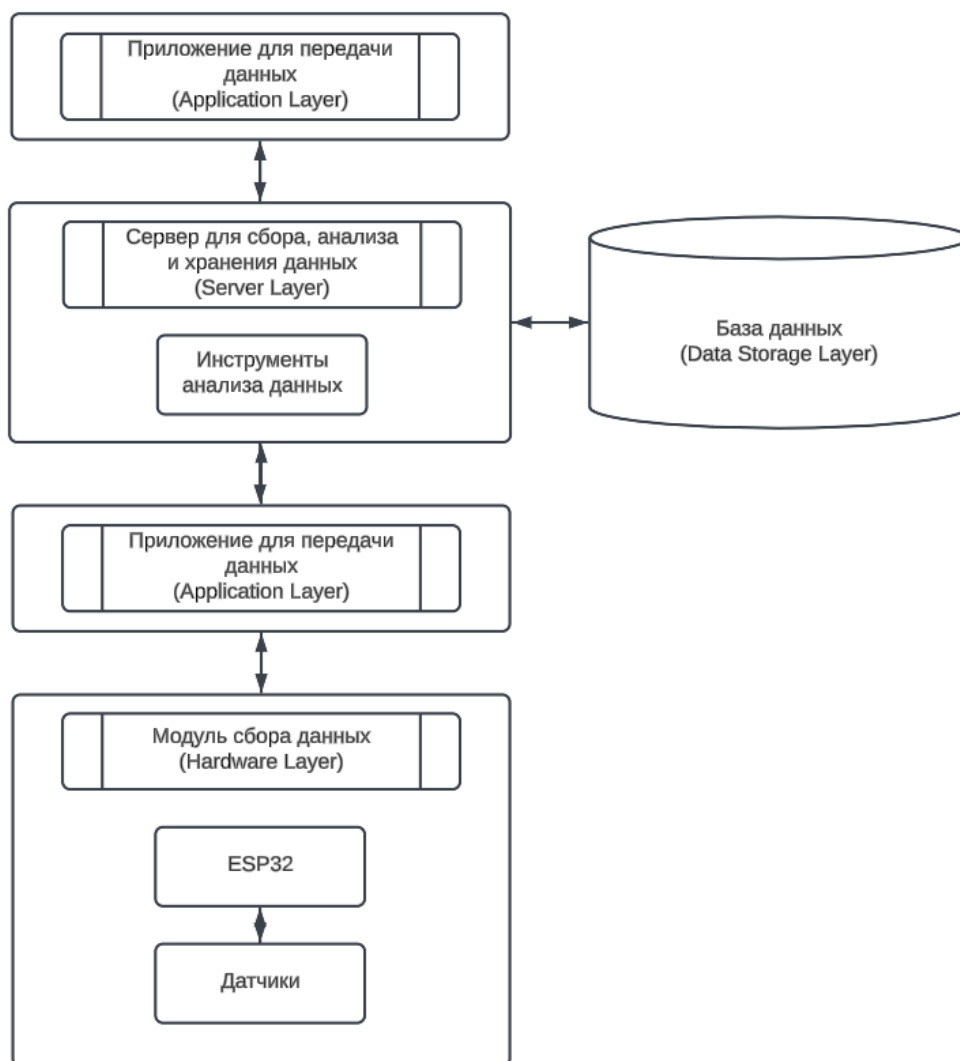


Рис. 3: Архитектура.

1. Модуль сбора данных (Hardware Layer):

Он включает в себя аппаратное обеспечение для снятия показаний ЭКГ, такое как микроконтроллер ESP32 и усилитель сигнала AD8232. Микроконтроллер ESP32 отвечает за считывание данных с электродов и преобразование их в цифровой формат. Усилитель сигнала AD8232 усиливает сигнал ЭКГ для более точного считывания.

2. Приложение для передачи данных (Application Layer):

Отвечает за прием, обработку и передачу данных с модуля сбора данных на сервер для анализа и хранения. В приложении реализована логика работы с данными, включая их форматирование, упаковку и отправку на сервер.

3. Сервер для сбора, анализа и хранения данных (Server Layer):

Этот слой осуществляет прием, анализ и хранение данных, полученных от приложения для передачи данных. В рамках сервера реализованы функциональности по обработке данных, анализу показателей сердечной активности и хранению результатов.

4. База данных (Data Storage Layer):

Обеспечивает долговременное хранение данных и предоставляет интерфейсы для их извлечения и манипуляций. Данные о показателях сердечной активности сохраняются в базе данных для последующего доступа и анализа. Для обеспечения надежности и масштабируемости, база данных может быть построена на основе реляционных или NoSQL технологий.

5. Web Layer (Веб-приложение):

Этот слой предоставляет пользовательский интерфейс для взаимодействия с системой через веб-браузер. Пользователи могут просматривать данные, настраивать параметры мониторинга и получать заключения по результатам исследования ЭКГ.

3.1. Разработка аппаратной части комплекса (Hardware Level)

В аппаратной части программно-аппаратного комплекса для мониторинга показателей сердца человека решающее значение имеет выбор компонентов, обеспечивающих сбор и передачу данных. Микроконтроллеры, сенсоры и другие устройства должны быть тщательно подобраны с учетом их функциональности, надежности и совместимости с основными целями проекта.

В этом разделе будет рассмотрен процесс выбора и обоснование использования конкретных компонентов, фокусируясь на микроконтроллере, датчиках для снятия ЭКГ и других важных устройствах для поддержки работоспособности модуля в целом.

3.1.1. Выбор микроконтроллера

При выборе микроконтроллера рассматривались следующие характеристики:

- Размер платы
- Энергопотребление
- Вычислительная мощность (кол-во ядер)
- Количество пинов

С учетом всех факторов, были выбраны микроконтроллеры ESP32 и ESP32-3C. Ниже приведены основные сравнительные характеристики.

	ESP32	ESP32-3C
CPU	Xtensa LX6	RISC-V
Кол-во ядер	2	1
Пины GPIO	34	22
Потребление	До 325 мА	До 240 мА
Размеры	31 x 18 x 3.0 мм	24 x 16 x 3.1 мм

По итогу решено использовать микроконтроллер ESP32-3C поскольку он обладает меньшими размерами и более низким энергопотреблением, однако придется столкнуться с проблемами производительности поскольку выбранный микроконтроллер имеет одно ядро и нельзя будет отправлять полученные данные в фоновом режиме на втором ядре.

3.1.2. Выбор датчика для снятия ЭКГ

При выборе модели датчика сердечного ритма были определены следующие критерии, которым он должен соответствовать:

1. Датчик должен поддерживать считывание на частоте в 100 Гц
2. Датчик не должен иметь больших размеров
3. Измерения датчика должны быть разборчивыми
4. Цена датчика должна быть бюджетной

После проведения анализа было решено остановиться на следующих датчиках:

- AD620
- AD623
- AD8232
- CJMCU-333

Далее проходил этап их сравнения на тестовом стенде.

Тестирующий стенд был разработан для одновременного анализа данных со всех четырех датчиков. Основным питающим элементом была литиевая батарея, чтобы избежать зашумления сигнала, которое может возникнуть при питании от сети.

Приведем описание для датчиков, которые были главными претендентами на выбор.

3.1.3. Тестирование CJMCU-333

CJMCU333 — маломощный прецизионный инструментальный усилитель. Датчик имеет универсальную конструкцию с тремя операционными усилителями, небольшой размер и малое энергопотребление. Один внешний резистор устанавливает любой коэффициент усиления от 1 до 1000 (Рис. 4).

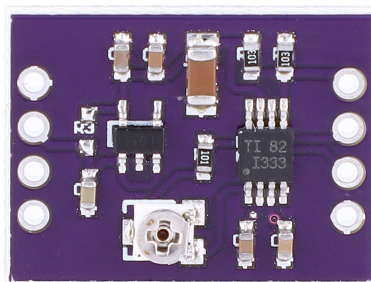


Рис. 4: CJMCU-333

После калибровки и тестирования, ЭКГ выглядело следующим образом (Рис. 5, 6).

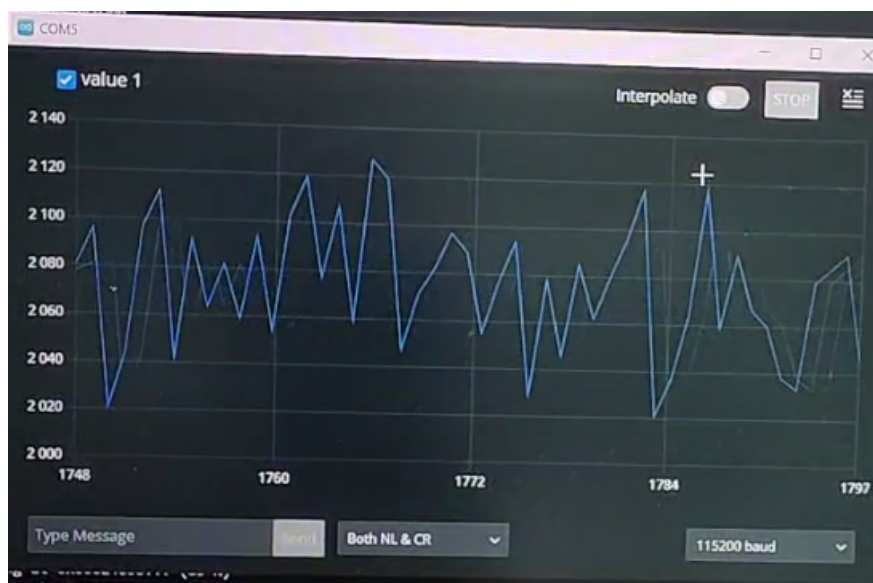


Рис. 5: CJMCU-333 ЭКГ Пример-1.

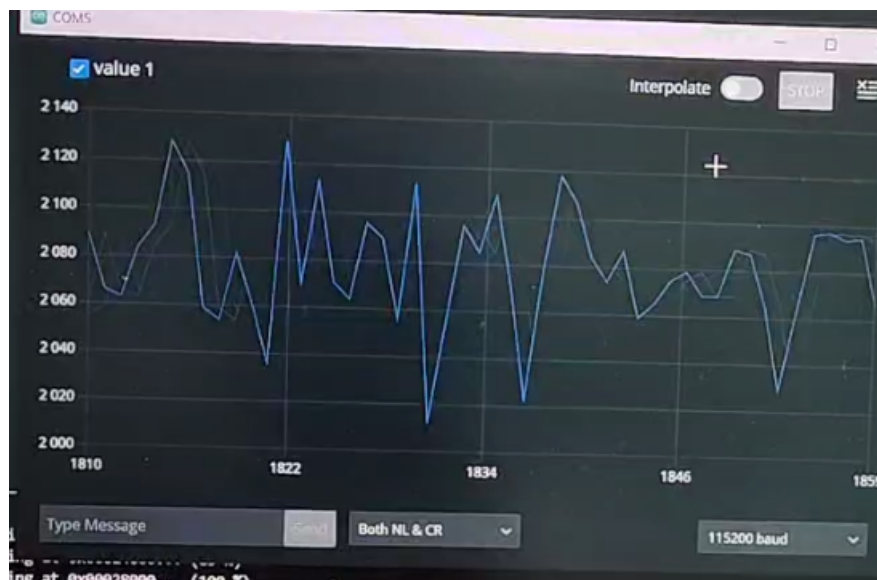


Рис. 6: CJMCU-333 ЭКГ Пример-2.

По графику невозможно отследить периодичность. Кривая имеет большое количество артефактов и шума. Разобрать данную ЭКГ-кардиограмму не представляется возможным. Результаты тестирования были неудовлетворительными, поэтому данный датчик применяться в проекте не будет.

3.1.4. Тестирование AD8232

AD8232 — это интегрированный блок формирования сигнала для ЭКГ и других приложений измерения биопотенциала. Он предназначен для извлечения, усиления и фильтрации слабых сигналов биопотенциала в условиях шумов, например, создаваемых движением или удаленным размещением электродов (Рис. 7).

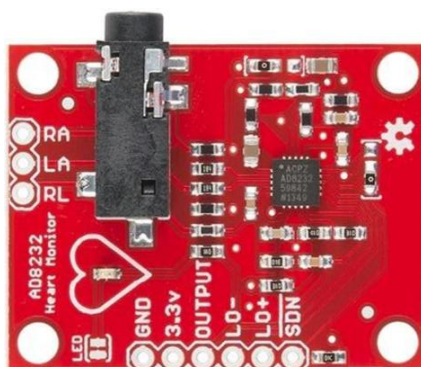


Рис. 7: AD8232

При тестировании график ЭКГ выглядел следующим образом (Рис. 8).

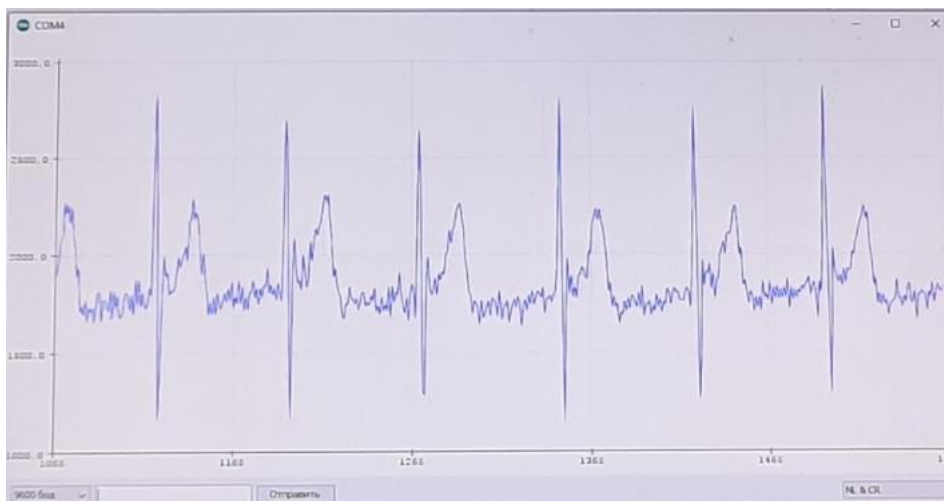


Рис. 8: График AD8232 1-ое отведение.

По результатам первых тестов можно заметить, что показатели получаются довольно точными и чистыми. Кривая имеет выраженную периодичность, а также точно прослеживаются зубцы. В качестве эксперимента было проведено считывание ЭКГ по второму отведению (Рис. 9).



Рис. 9: График AD8232 2-ое отведение.

А также по значениям первого и второго отведений было вычислено третье отведение с последующим отображением значений в виде графика (Рис. 9).

Таким образом, из всех протестированных датчиков был выбран именно AD8232, поскольку только с его помощью удалось достичь таких качественных результатов.

3.1.5. Разработка модели

При разработке модуля необходимо учитывать возможность его размещения на спортивной майке, а значит он должен иметь относительно небольшие размеры и маленький вес.



Рис. 10: График AD8232 3-ое отведение.

Так как модуль необходимо будет заряжать и отправлять с него данные на сервер было принято решение разбить его на два подмодуля. Так, в первом подмодуле будут располагаться электроды и датчики считывания ЭКГ, а во втором то, что необходимо для сохранения, обработки и передачи полученных данных.

Подмодуль ЭКГ

Экспериментальным путем было выяснено, что наиболее подходящей моделью датчика сердечного ритма является AD8232. Так как нам требуется считывать два типа отведений минимум, а каждый датчик может считывать максимум один, нам потребуется использовать их сразу два. Прimitивная схема данного подмодуля выглядит следующим образом (Рис. 11).

На вход в данный подмодуль поступает четыре провода:

- SDN1
- GND
- 3v3
- SDN2

Провода типа SDN отвечают за перевод подключенного к ним датчика в энергосберегающий режим. GND и 3v3 отвечают за питание датчиков.

На выход идет шесть проводов:

- два провода типа LO1

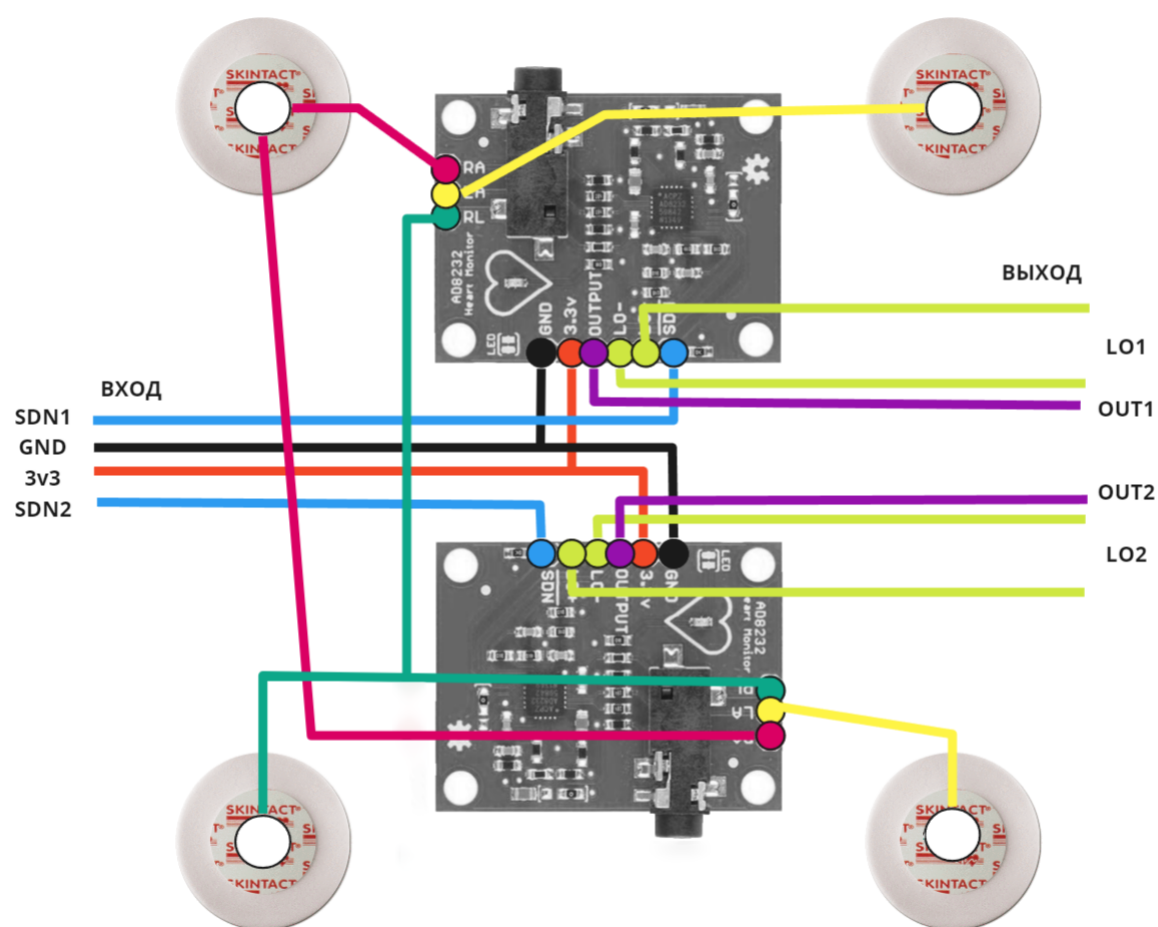


Рис. 11: Подмодуль ЭКГ.

- два провода типа LO2
- OUT1
- OUT2

Провода типа LO передают сигнал о том, что электроды подключены к датчику и с них считывается сигнал. Провод OUT1 отправляет значение ЭКГ по первому типу отведения, OUT2 по второму.

Подмодуль с микроконтроллером

Основная задача данного подмодуля является считывание данных с датчиков ЭКГ, промежуточное хранение и их последующая отправка. Для сохранения большого объема информации на определенное время требуется внешний накопитель. Так как модуль является автономным, для него требуется батарея с возможностью зарядки.

Таким образом, минимальный набор требуемых элементов выглядит следующим образом:

- ESP32-3C
- Преобразователь напряжение в 3В
- Модуль с SD-картой
- Модуль для зарядного устройства
- Литий-ионная батарея
- Магнитный коннектор

Исходя из этого, была разработана примитивная схема данного подмодуля (Рис. 12).

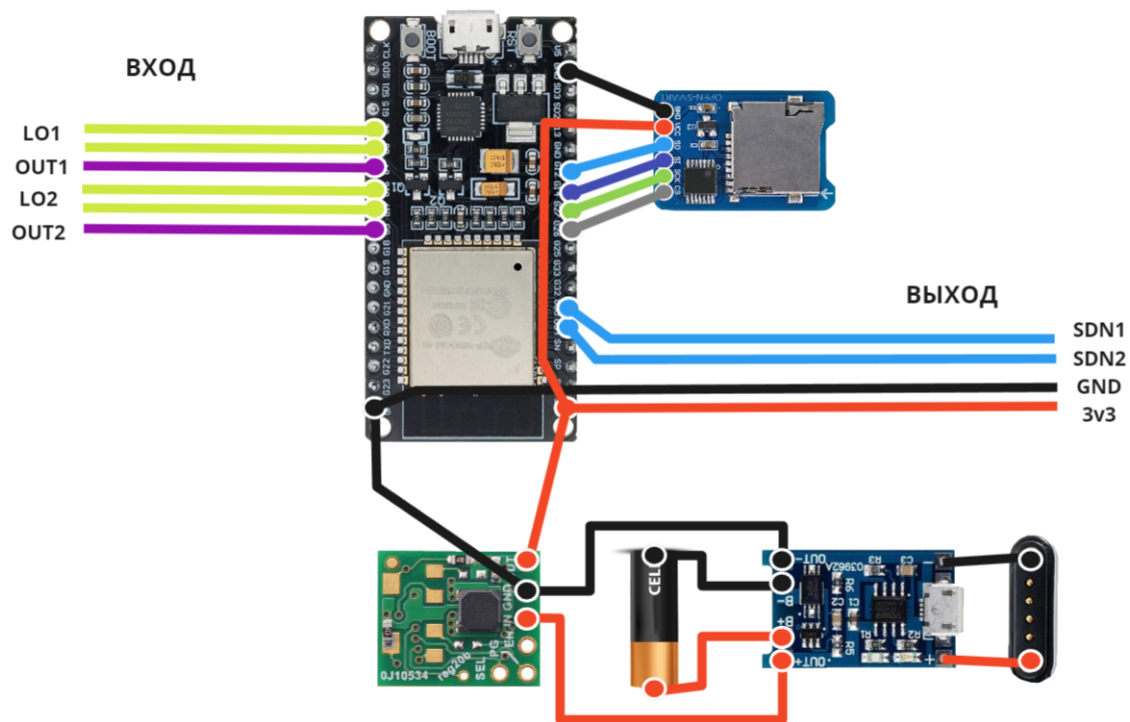


Рис. 12: Подмодуль с микроконтроллером.

Входные и выходные провода соединятся с подмодулем ЭКГ, поэтому ознакомится с назначением каждого провода можно в вышеупомянутом разделе.

Для удобства была разработана тестовая 3D модель, в которую были размещены все необходимые компоненты.

3.2. Разработка программной части комплекса (Software level)

В программной части программно-аппаратного комплекса для мониторинга показателей сердца человека не менее важен выбор правильных технологий и методов разработки. Разработка приложения для передачи данных, серверной части и веб-интерфейса требует грамотного подхода и выбора наиболее подходящих инструментов.

В данном разделе будет рассмотрен процесс выбора технологий и методов разработки, обоснование принятых решений и анализ их влияния на функциональность и эффективность всего комплекса мониторинга, а также определение функциональных и нефункциональных требований.

3.2.1. Разработка требований к программному продукту (ПП)

Требования к программному продукту можно разделить на функциональные и нефункциональные.

Функциональные требования:

1. Сбор данных ЭКГ:

ПП должен уметь собирать данные ЭКГ с помощью модуля ESP32 и AD8232.

2. Передача данных:

Данные ЭКГ должны передаваться от модуля на сервер в реальном времени.

3. Анализ данных:

Сервер должен анализировать поступающие данные, выявлять аномалии и предупреждать пользователя.

4. Хранение данных:

Данные должны сохраняться в базе данных PostgreSQL для последующего анализа и отчетности.

5. Пользовательский интерфейс:

ПП должен иметь удобный пользовательский интерфейс для просмотра текущих и исторических данных ЭКГ, а также для получения заключений по показаниям ЭКГ.

6. Отображение данных в реальном времени:

Пользователям должна быть предоставлена возможность мониторинга показаний ЭКГ в реальном времени через веб-интерфейс.

Нефункциональные требования:

1. Отказоустойчивость:

Приложение должно быть устойчивым к сбоям и обеспечивать непрерывную работу в течение длительного времени.

2. Безопасность данных:

Все данные, передаваемые между компонентами системы, должны быть защищены с помощью соответствующих механизмов шифрования и аутентификации.

3. Производительность:

Приложение должно обеспечивать высокую производительность при передаче и обработке данных, чтобы минимизировать задержки и обеспечить оперативную реакцию на изменения состояния пациента.

4. Масштабируемость:

Система должна быть способна масштабироваться в зависимости от количества пользователей и объема данных, обрабатываемых ежедневно.

5. Простота использования:

Веб-интерфейс должен быть интуитивно понятным и легким в использовании даже для неопытных пользователей.

3.2.2. Разработка программной части для модуля сбора данных на базе ESP32

Выбор фреймворка

Для разработки программного обеспечения на ESP32 был выбран Arduino Framework в связке с PlatformIO благодаря простоте и удобству использования. Arduino Framework предоставляет все необходимые инструменты для написания, компиляции и загрузки кода на мик-

роконтроллер. Его поддержка широкого спектра библиотек и большая пользовательская база делают решение возникающих проблем более легким.

Архитектура

При проектировании программного обеспечения использовался модульный подход, что позволяет изолировать различные функциональные компоненты системы. Это упрощает разработку, отладку и сопровождение кода, а также способствует повторному использованию и легкости расширения.

В системе используется паттерн Singleton, что позволяет гарантировать, что каждый модуль имеет только одну глобальную точку доступа. Это особенно важно для модулей, которые управляют аппаратными ресурсами, такими как SD-карта или BLE-сервисы. Singleton паттерн обеспечивает уникальность экземпляра и глобальный доступ к нему, что упрощает управление состоянием и ресурсами.

Структура программного кода на ESP32

Работа с модулями организована путем разделения на заголовочные файлы (.h) и файлы реализации (.cpp). Заголовочные файлы содержат объявления функций и классов, обеспечивая интерфейс между модулями, тогда как файлы реализации содержат конкретную реализацию этих функций.

Модули и точка входа

1. BLE Service Handler Module

Обрабатывает взаимодействие через Bluetooth Low Energy (BLE). Включает функции для настройки и обработки BLE-соединений.

2. Configuration Helper Module

Содержит параметры конфигурации, такие как настройки сети и параметры подключения. Обеспечивает централизованное управление конфигурациями.

3. Data Package Module

Осуществляет упаковку и распаковку данных для передачи. Включает структуры данных и функции для их обработки.

4. SPI Flash Module

Управляет взаимодействием с SPI Flash памятью. Содержит функции для чтения и записи данных во флеш-память.

5. Main Module(Основная точка входа в программу)

Отвечает за инициализацию системы и подключение модулей. Также содержит основной цикл работы, в котором происходит вызов функций других модулей.

Основная логика работы программы

Основная логика работы, инициализация и управление модулями содержится в Main Module программы. Ключевыми элементами данного модуля являются две основные функции Arduino Framework, а именно:

- **setup():** Инициализация системы и подключение модулей.
- **loop():** Основной цикл работы, в котором происходит вызов функций других модулей.

Рассмотрим основную логику работы этих функций:

setup():

1. Инициализация серийного порта для вывода отладочной информации.
2. Настройка BLE-сервиса с помощью вызова соответствующих функций из BLE Service Handler.
3. Загрузка конфигурации из SPI Flash памяти.
4. Инициализация всех необходимых периферийных устройств и модулей, таких как сенсоры и Wi-Fi.

loop():

1. Циклический опрос состояния системы.
2. Сбор данных с сенсоров и их обработка.
3. Упаковка данных с помощью Data Package Module.

4. Передача данных через BLE или по Wi-Fi на сервер.
5. Управление состояниями системы и обработка событий, таких как потеря соединения или ошибки передачи данных. (В данном случае данные сохраняются на SD карту)

Тестирование

В ходе разработки проекта возникли ограничения в виде отсутствия собранного тестирующего стенда, что затруднило проведение автоматизированного традиционного тестирования.

На данный момент применяется тестирование модуля только в области форматирования и проверки стиля кода, который на него загружается, что в конечном итоге все равно способствует повышению его читаемости и качества.

3.2.3. Приложение для передачи данных (Application Layer)

Electron-приложение служит посредником для передачи данных ЭКГ с устройства ESP32 на сервер.

Сценарии использования

- **Аутентификация и авторизация:**

1. Пользователь открывает Electron приложение и вводит свои учетные данные (логин и пароль).
2. Приложение отправляет данные на сервер для аутентификации.
3. В случае успешной аутентификации пользователь получает доступ к функционалу приложения.
4. Пользователь может выйти из системы, закрыв Electron приложение или используя соответствующую функцию в интерфейсе приложения.
5. После выхода из системы пользователь теряет доступ к функционалу приложения до следующего входа.

- **Отправка данных ЭКГ на анализ с ESP32 на сервер:**

1. Пользователь может отправлять данные ЭКГ на сервер для анализа и получения дополнительной информации о своем состоянии здоровья с помощью технологии BLE.

- **Просмотр информации о количестве отправленных пакетов:**

1. Пользователь может видеть общее количество отправленных пакетов.
2. Пользователь может просматривать информацию о качестве подключения.

3.2.4. Сервер для сбора, анализа и хранения данных (Server Layer)

Выбор фреймворка и языка

Python вместе с Flask был выбран в качестве основного фреймворка для разработки серверной части веб-приложения по нескольким причинам:

- 1. Легковесность и Гибкость:**

Flask отличается простотой и легковесностью в использовании, что делает его идеальным выбором для разработки небольших и средних проектов. При этом, он обладает достаточной гибкостью, чтобы удовлетворить потребности различных задач.

- 2. Простота Создания RESTful API:**

Flask предоставляет все необходимые инструменты для создания RESTful API, что позволяет легко интегрировать сервер с различными компонентами системы. Это критически важно для проекта, где требуется эффективное взаимодействие с базой данных и клиентскими приложениями.

- 3. Широкая Поддержка Библиотек:**

Язык программирования Python обладает богатой экосистемой библиотек, что делает его идеальным выбором для разработки веб-приложений. Python предоставляет широкие возможности для анализа данных, что идеально подходит для обработки медицинских данных.

Архитектура

В разработке серверной части веб-приложения использован принцип архитектуры, основанный на модели MVC (Model-View-Controller), дополненный использованием REST API для взаимодействия с клиентской стороной, написанной на Vue.js.

На серверной стороне, основные компоненты архитектуры включают в себя:

- **Модель (Model):**

Представлена моделями данных и логикой взаимодействия с базой данных, реализованными с использованием ORM (объектно-реляционного отображения). Данные хранятся и обрабатываются на сервере, что обеспечивает централизованное хранение и управление информацией.

- **Представление (View):**

Фронтенд, написанный на Vue.js, действует как "представление" и отображает данные, полученные через REST API от сервера. Фронтенд обеспечивает пользовательский интерфейс для взаимодействия с приложением и представляет пользовательский опыт.

- **Контроллер (Controller):**

Контроллеры, представленные в виде маршрутов в API, обрабатывают HTTP-запросы от клиента и возвращают соответствующие данные. Они являются посредниками между моделью и представлением, обеспечивая передачу данных между ними и управление бизнес-логикой приложения.

Помимо этого, в архитектуре также будет интегрирован модуль взаимодействия с приложением на Electron. Этот модуль будет отвечать за получение данных ЭКГ для дальнейшей обработки на сервере, а также за отображение ЭКГ и заключений на клиентской стороне приложения. Таким образом, функциональность приложения расширяется, обеспечивая его более широкий спектр использования.

Данный подход обеспечивает четкое разделение бизнес-логики, представления данных и управления запросами на отдельные компоненты, что делает приложение более модульным, легко сопровождаемым и масштабируемым.

Структура сервера

Модуль ORM и Работа с Базой Данных (ORM and Database Handling)

- **Описание:** Модуль отвечает за взаимодействие с базой данных с помощью объектно-реляционного отображения и выполнения SQL-запросов.
- **Назначение:**
 - Определение моделей данных.
 - Управление транзакциями базы данных.
 - Обеспечение абстракции для работы с реляционными данными.
 - Подключение к базе данных.
 - Управление схемой базы данных и инициализация данных.
- **Компоненты:**
 - Модели данных.
 - Конфигурация базы данных.
 - Скрипты инициализации базы данных.
 - Сессии для взаимодействия с базой данных.

Модуль Роутинга (Routing)

- **Описание:** Модуль управляет маршрутами и эндпоинтами API.
- **Назначение:**
 - Определение URL маршрутов.
 - Обработка входящих HTTP-запросов.
 - Связывание запросов с соответствующими функциями контроллеров.
- **Компоненты:**
 - Маршруты для аутентификации.
 - Маршруты для работы с веб-приложением.
 - Маршруты для поддержания взаимодействия с модулем Electron.

Модуль ИИ для Анализа Данных (AI Data Analysis)

- **Описание:** Модуль использует алгоритмы машинного обучения и другие методы искусственного интеллекта для анализа данных ЭКГ.
- **Назначение:**
 - Предобработка данных ЭКГ.
 - Анализ и классификация данных ЭКГ.
 - Генерация заключений на основе анализа.
- **Компоненты:**
 - Алгоритмы машинного обучения.
 - Модели для анализа данных.
 - Методы предсказания и оценки.

Модуль Взаимодействия с Electron (Electron Integration)

- **Описание:** Модуль обеспечивает интеграцию с десктопным приложением, написанным на Electron.
- **Назначение:**
 - Получение данных ЭКГ от десктопного приложения.
- **Компоненты:**
 - API для взаимодействия с Electron.
 - Логика обработки входящих данных.

Модуль Аутентификации и Авторизации (Authentication and Authorization)

- **Описание:** Модуль управляет регистрацией, входом и выходом пользователей, а также контролем доступа.
- **Назначение:**
 - Обработка регистрации и аутентификации пользователей.
 - Управление сессиями и токенами.

- Обеспечение безопасности доступа к API.

- **Компоненты:**

- Методы регистрации и входа пользователей.
- Управление JWT токенами.
- Политики доступа и проверки прав пользователей.

Модуль Логирования и Мониторинга (Logging and Monitoring)

- **Описание:** Модуль отвечает за запись логов и мониторинг состояния системы.

- **Назначение:**

- Логирование ошибок и событий.
- Мониторинг производительности и состояния системы.

- **Компоненты:**

- Конфигурация логирования.
- Методы мониторинга.

Модуль Тестирования (Testing)

- **Описание:** Модуль обеспечивает тестирование всех компонентов сервера.

- **Назначение:**

- Написание и выполнение тестов для проверки функциональности.
- Обеспечение качества кода и предотвращение ошибок.

- **Компоненты:**

- Тесты для маршрутов и API.
- Тесты для взаимодействия с базой данных.
- Тесты для модулей ИИ и анализа данных.

Основная логика работы программы

Инициализация и настройка приложения

1. При запуске приложения инициализируется объект Flask.
2. Устанавливается соединение с базой данных.
3. Настраивается JWT для безопасной аутентификации пользователей.

Обработка запросов

1. При поступлении запроса на сервер, Flask маршрутизирует его к соответствующему обработчику.
2. Обработчики маршрутов выполняют необходимые операции, включая обработку запросов аутентификации, доступа к данным, выполнение бизнес-логики и т.д.

Аутентификация и авторизация

1. При получении запроса, требующего аутентификации, сервер проверяет наличие и валидность JWT в заголовке запроса.
2. Если токен действителен, пользователь получает доступ к запрашиваемым ресурсам или операциям.

Взаимодействие с данными

1. При выполнении запросов на получение, создание, обновление или удаление данных, сервер взаимодействует с базой данных через модели и ORM SQLAlchemy.
2. Данные могут быть переданы обратно клиенту в виде JSON или другого формата в зависимости от запроса.

Работа с ИИ

1. При получении данных ЭКГ от клиента или других источников, сервер обрабатывает их с использованием алгоритмов машинного обучения и анализа данных.

2. Результаты анализа, включая диагностику и заключения, могут быть отправлены обратно клиенту для отображения и дальнейшего использования.

Тестирование

1. При разработке нового функционала, сервер проходит тестирование вручную с помощью разработанных модульных тестов.
2. Тесты включают в себя проверку основных сценариев использования, а также тестирование работы с данными и взаимодействия с клиентами.

Использование Docker для контейнеризации приложения

Поскольку сервер будет запущен на специально выделенной временной удаленной машине, контейнеризация необходима, что обеспечить независимость от окружения, что упрощает развертывание и масштабирование системы. Docker гарантирует, что приложение будет работать одинаково на всех платформах, предоставляя единое окружение для разработки, тестирования и производства.

3.2.5. База данных (Data Storage Layer)

Выбор СУБД

Для структурирования и хранения данных 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.

Проектирование структуры БД

1. Создание схемы базы данных:

- В ходе проектирования было определено три основных сущности: пользователи данные ЭКГ и аутентификационные данные.
- Для каждой сущности была разработана соответствующая таблица в базе данных, представляющая собой логическую структуру для хранения данных.

2. Определение полей и их типов:

- Каждая таблица содержит набор полей, определяющих характеристики сущности.
- Типы данных для полей выбирались с учетом требований к хранению и обработке информации.

3. Определение отношений между таблицами:

- Для связывания данных между таблицами были определены внешние ключи и отношения.
- Например, таблица аутентификационных данных имеет внешний ключ, связывающий ее с таблицей пользователей.

Использование Docker для контейнеризации приложения

Для базы данных PostgreSQL также был создан собственный сервис и контейнер для гарантирования работоспособности приложения на всех платформах.

3.2.6. Web Layer (Веб-приложение)

Web-приложение является основным источником для изучения и рассмотрения пользователем своих данных ЭКГ. Оно реализовано как одностраничное приложение (SPA), предоставляющее пользователю удобный интерфейс для взаимодействия с данными.

Сценарии использования

1. Авторизация и аутентификация:

- Пользователи могут войти в систему, предоставив свои учетные данные.
- После успешной аутентификации пользователи получают доступ к функционалу приложения.

2. Заполнение профиля:

- Пользователи имеют возможность добавлять и изменять свои личные данные в своем профиле.
- Обновленные данные сохраняются на сервере для последующего использования.

3. Отображение графиков данных ЭКГ:

- Пользователи могут запрашивать просмотр графиков данных ЭКГ, полученных от Electron приложения.
- Графики отображаются в интерфейсе приложения для удобного изучения пользователем своих данных.

4. Взаимодействие с сервером на Python с использованием Flask:

- Приложение взаимодействует с сервером, который построен на языке Python с использованием фреймворка Flask, посредством HTTP-запросов.
- Это позволяет приложению обновлять данные пользователя и получать рекомендации на основе информации о нем.

5. Отображение показаний здоровья и рекомендаций:

- Пользователям предоставляется информация о их текущем состоянии здоровья, основанная на данных ЭКГ и других медицинских параметрах.
- Кроме того, пользователи могут получать рекомендации по улучшению своего здоровья на основе анализа предоставленных данных.

4. Практическая апробация и внедрение

5. Список литературы

6. Приложение