УДК 615.477:004.94

Хайбуллин Камиль Ильдусович,

Инженер – электронщик в АйМастере УНПК
«Международный Университет Кыргызстана»
Email: www.haizis@gmail.com

Хайбуллин Камиль Ильдусович,
АйМастерде электроника инженери ОИӨК
«Кыргызстан эл аралык университети»
Email: www.haizis@gmail.com

Haibullin Kamil Ildusovich,
Electronics engineer at iMaster
ERPC "International University of Kyrgyzstan"
Email: www.haizis@gmail.com

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ПО КОНСТРУИРОВАНИЮ БИОНИЧЕСКИХ ПРОТЕЗОВ РУК

БИОНИКАЛЫК ПРОТЕЗДИК КОЛДОРДУ ДОЛБООРЛООНУН МААЛЫМАТТЫК СИСТЕМАСЫН ИШТЕП ЧЫГУУ

DESIGN AND DEVELOPMENT OF AN INFORMATION SYSTEM FOR THE CONSTRUCTION OF BIONIC HAND PROSTHESES

Аннотация: *В данной работе рассмотрена разработка информационной системы по конструированию бионических протезов рук, а также прототипа протеза руки. Разработан функциональный и информативный веб-сайт, предоставляющий пользователям полное представление о процессе конструирования бионических протезов рук. Веб-сайт включает разделы с блогом, личным кабинетом, каталогом протезов, а также с его удобным выбором и заказом (приобретением).*

Ключевые слова: *протез, API, .NET, MSSQL, база данных, рефакторинг*

Аннотациясы: *Бул макалада бионикалык кол протездерин, ошондой эле протез колунун прототибин конструкциялоо боюнча маалыматтык системаны иштеп чыгуу каралган. Колдонуучуларга бионикалык протездик колдорду куруу процесси жөнүндө толук түшүнүк берген функционалдык жана маалыматтык веб-сайт иштелип чыккан. Веб-сайт блог, жеке кеңсе, протездер каталогу, ошондой эле анын ыңгайлуу тандоосу жана буйрутмасы (сатып алуу) менен бөлүмдөрдү камтыйт.*

Негизги сөздөр: *протез, API, .NET, MSSQL, маалыматтар базасы, рефакторинг.*

Abstract: *This paper considers the development of an information system for the design of bionic prosthetic hands, as well as a prototype of a prosthetic arm. A functional and informative website has been developed that provides users with a complete understanding of the process of designing bionic prosthetic hands. The website includes sections with a blog, a personal account, a catalog of prosthetics, as well as its convenient selection and purchase.*

Key words: *prosthesis, API, .NET, MSSQL, database, refactoring.*

В современной медицине информационные системы стали незаменимым инструментом, особенно в сфере протезирования, где они позволяют создавать высокотехнологичные и индивидуально адаптированные решения. Специализированные веб-платформы объединяют базы данных о материалах и технологиях, форумы для обмена опытом и цифровые инструменты для проектирования, обеспечивая тем самым эффективное взаимодействие между врачами, биоинженерами и пациентами.

Этот проект стремится объединить передовые технологии в области биомедицины, инженерии и информационных технологий, предлагая инновационные решения для создания бионических протезов (Фримен А., 2022), которые не только максимально соответствуют индивидуальным особенностям каждого пользователя, но и обеспечивают высокий уровень функциональности.

Использование информационных систем значительно оптимизирует этот процесс: виртуальное моделирование сокращает время разработки, а облачные платформы дают возможность специалистам по всему миру совместно работать над проектами. Таким образом, интеграция передовых медицинских, инженерных и цифровых технологий открывает новые горизонты в протезировании, делая его более доступным и эффективным для пациентов.

Информационная система, предназначенная для конструирования протезов, должна предоставлять пользователям возможность заказывать и настраивать протезы в соответствии с индивидуальными потребностями. Она должна быть полезной как для пациентов, так и для инженеров и дизайнеров, задействованных в процессе разработки. Система обязана поддерживать простой и удобный процесс формирования заказов с учетом персональных параметров пациента — таких как размер, форма и функциональные характеристики. Это обеспечит выбор наиболее подходящей модели протеза (Фримен А., 2022).

Одной из ключевых функций системы должен стать модуль мониторинга состояния протезов, диагностики возможных неисправностей и автоматической генерации уведомлений о необходимости технического обслуживания.

Система должна быть интегрирована с мобильными приложениями, обеспечивая управление протезом в режиме реального времени. Это позволит пациентам гибко настраивать работу устройства в соответствии с текущими потребностями.

Особое внимание должно быть уделено защите персональных данных и конфиденциальной информации, касающейся как пациента, так и технических характеристик протезов (Картер П., 2023). Использование облачного хранилища позволит обеспечить как безопасность, так и доступность данных.

Наконец, архитектура системы должна быть модульной, что упростит внедрение новых функций, интеграцию с внешними сервисами и адаптацию к развитию технологий в области бионики и медицинского протезирования.

Разрабатываемая информационная система построена по клиент-серверной архитектуре, где клиентская часть реализована на технологии Blazor WebAssembly, а серверная часть — на ASP.NET Core Web API (Уайлдер Дж., 2022). В качестве базы данных используется Microsoft SQL Server, подключённый к серверной части через Entity Framework Core (EF Core) (Троелсен Э., Джапиксе Ф., 2022; Шарп Д., 2021). Такой подход обеспечивает модульность и масштабируемость приложения, а также позволяет легко разделить логику отображения, обработки данных и хранения информации.

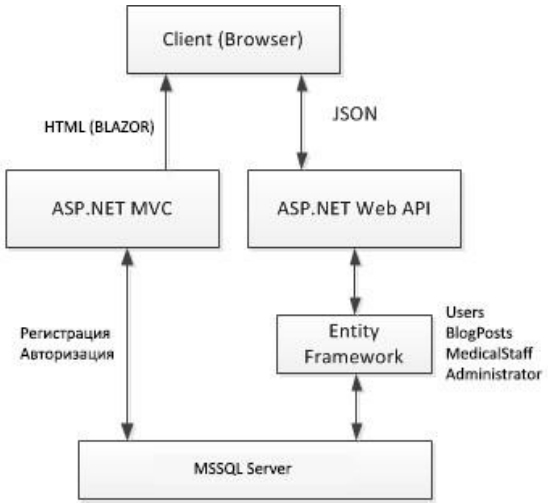

Рис 2.2.1. Клиент-серверная архитектура проекта

Проект реализован с использованием шаблона Blazor WebAssembly App (ASP.NET Core Hosted), что подразумевает наличие трёх основных частей: Client (Blazor WebAssembly) — фронтенд-приложение, исполняемое в браузере. Написано на языке C#, компилируется в WebAssembly и загружается на клиентскую сторону.

- Server (ASP.NET Core API) — серверное API, отвечающее за обработку запросов, авторизацию, взаимодействие с базой данных, а также служит хостом для клиента.
- Shared — общая библиотека классов и моделей, используемая как в клиентской, так и в серверной части. Например, модели данных (DTO, Entity) и контракты между клиентом и сервером.

Blazor WebAssembly — это технология, позволяющая создавать интерактивные вебприложения с использованием языка C# вместо JavaScript (Алиева Л., 2022). Благодаря компиляции C# в WebAssembly приложение исполняется прямо в браузере пользователя, без необходимости постоянного обращения к серверу (в отличие от Blazor Server).

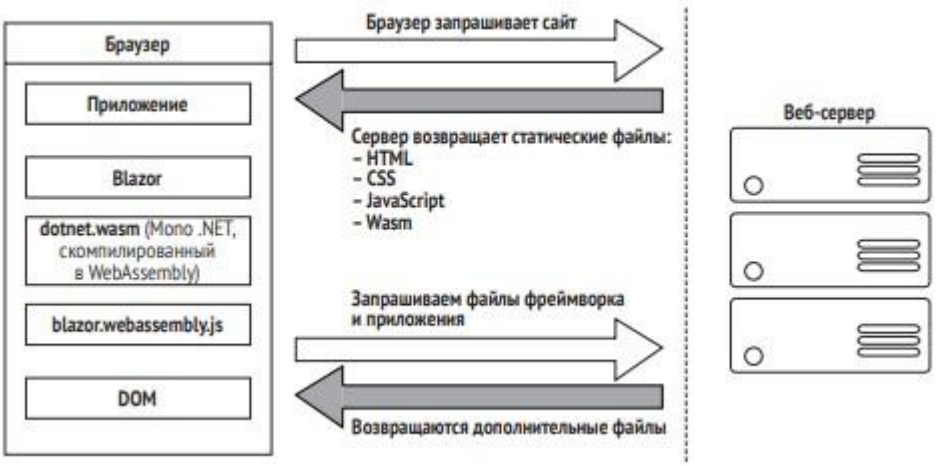

Рис 2.2.2. Принцип работы Blazor WebAssembly

Разработка клиентской части. Клиентская часть проекта была разработана с использованием технологии Isolated Blazor WebAssembly. Позволяющая легко и быстро проверять результаты верстки, по сравнению с обычными Blazor-проектами.

Для визуального оформления использовались технологии HTML и CSS, а базовая структура страниц и интерфейсов создавалась на основе шаблона, подготовленного в сервисе Canva, с последующей интеграцией в Blazor. Компонентная модель Blazor включает в себя

множество встроенных элементов управления, таких как <NavLink>, <EditForm>, <InputText>, <InputSelect> и другие, которые позволяют удобно организовать пользовательский интерфейс.

Для реализации интерактивных функций и работы с DOM-элементами применялся механизм JavaScript Interop — встроенный способ взаимодействия между JavaScript и C# в Blazor. Это позволяет использовать существующие JavaScript-библиотеки или создавать собственные функции, которые могут быть вызваны из компонентов Blazor.

Разработка серверной логики. Серверная часть проекта реализована с использованием ASP.NET Core Web API и Entity Framework Core (Троелсен Э., Джапиксе Ф., 2022). Она отвечает за выполнение бизнес-логики, взаимодействие с базой данных (SQL Server), обработку HTTP-запросов от клиента и отправку ответов в формате JSON (Шарп Д., 2021; Коггинс Б., 2022).

Серверная часть размещена в отдельном проекте (находящемся в том же решении что и клиентская часть), который входит в состав Blazor WebAssembly Hosted-приложения. Она выступает в роли API-уровня, предоставляющего интерфейсы для операций с данными (получение, создание, изменение, удаление).

Используемые средства разработки информационной системы:

C# — основной язык разработки, применяемый как для серверной логики, так и для написания компонентов клиентской части на Blazor. Обеспечивает строгую типизацию, высокую производительность и хорошую поддержку асинхронного программирования.

JavaScript — применяется для добавления интерактивности (например, переключение форм логина/регистрации, анимации, взаимодействие с DOM). Также используется в рамках технологии Eye Control.

SQL (T-SQL) — язык запросов к базе данных, используемый для взаимодействия с Microsoft SQL Server. Применяется в Entity Framework Core при генерации миграций и выполнении SQL-команд на сервере.

Visual Studio 2022 — основная среда разработки, предоставляющая удобные инструменты для работы с Blazor, Entity Framework и ASP.NET Core. Обеспечивает мощную поддержку отладки, Git-интеграции и создания миграций базы данных.

NET 9 SDK — новейшая версия программной платформы от Microsoft, предоставляющая современную среду выполнения, поддержку Blazor WebAssembly и ASP.NET Core.

ASP.NET Core — применяется на серверной стороне для создания RESTful API, обработки HTTP-запросов и взаимодействия с базой данных.

Entity Framework Core — ORM-фреймворк, используемый для работы с базой данных. Позволяет управлять схемой БД через миграции, а также упрощает выполнение запросов и отображение данных на объектную модель.

Пример конфигурации сервиса и контекста в *Program.cs*, в серверной части проекта:

```
// Необходимые сервисы для работы
builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
// Подключение MSSQL через EF
builder.Services.AddDbContext<ApplicationDbContext>(options =>
options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection"))
```

Модуль Eye-Control. Управление взглядом (Eye-Control) — это инновационная технология, основанная на отслеживании направления взгляда пользователя с целью взаимодействия с пользовательским интерфейсом. Данный подход открывает новые

возможности для создания более доступных и интуитивно понятных интерфейсов, особенно для людей с ограниченными возможностями.

Суть технологии заключается в определении координат точки на экране, на которую направлен взгляд пользователя, с последующим выполнением различных действий (наведение, клик, прокрутка и т.п.). Это достигается за счёт использования камеры (вебкамеры или камеры смартфона) и программного обеспечения, способного обрабатывать данные с камеры в реальном времени.

Используемая библиотека: WebGazer.js

В рамках проекта была применена открытая JavaScript-библиотека WebGazer.js, позволяющая реализовать eye-tracking прямо в браузере, без необходимости установки дополнительных плагинов или драйверов. WebGazer использует обычную веб-камеру и алгоритмы машинного обучения для калибровки и точного определения направления взгляда.

Пример интеграции WebGazer.js в Blazor WebAssembly

Интеграция библиотеки WebGazer в проект на Blazor WebAssembly осуществляется через механизм JavaScript Interop, который позволяет C#-компонентам вызывать функции JavaScript.

Подключение WebGazer.js в основном файле *index.html,* в клиентской части проекта:

```
1. <script src="https://webgazer.cs.brown.edu/webgazer.js"></script>
2. <script>
3. window.startEyeTracking = function () {
    a. webgazer.setGazeListener((data, elapsedTime) => {
        i. if (data != null) {
        ii. console.log(`X: ${data.x}, Y: ${data.y}`);
        iii. }
    b. }).begin();
4. };
5. </script>
```

Blazor компонент для запуска eye-tracking, размещается на странице *.razor*:

```
1. @inject IJSRuntime JS
2. <h3>Eye-Tracking Активен</h3>
3. <button @onclick="StartTracking">Начать отслеживание взгляда</button>
4. @code {
5. private async Task StartTracking()
6. await JS.InvokeVoidAsync("startEyeTracking");
```

После запуска, библиотека начнёт отслеживать координаты взгляда и выводить их в консоль браузера. Эти координаты в дальнейшем могут быть использованы для взаимодействия с интерфейсом: автоматический фокус на элементах, подсветка, "взглядовый" клик и т.д.

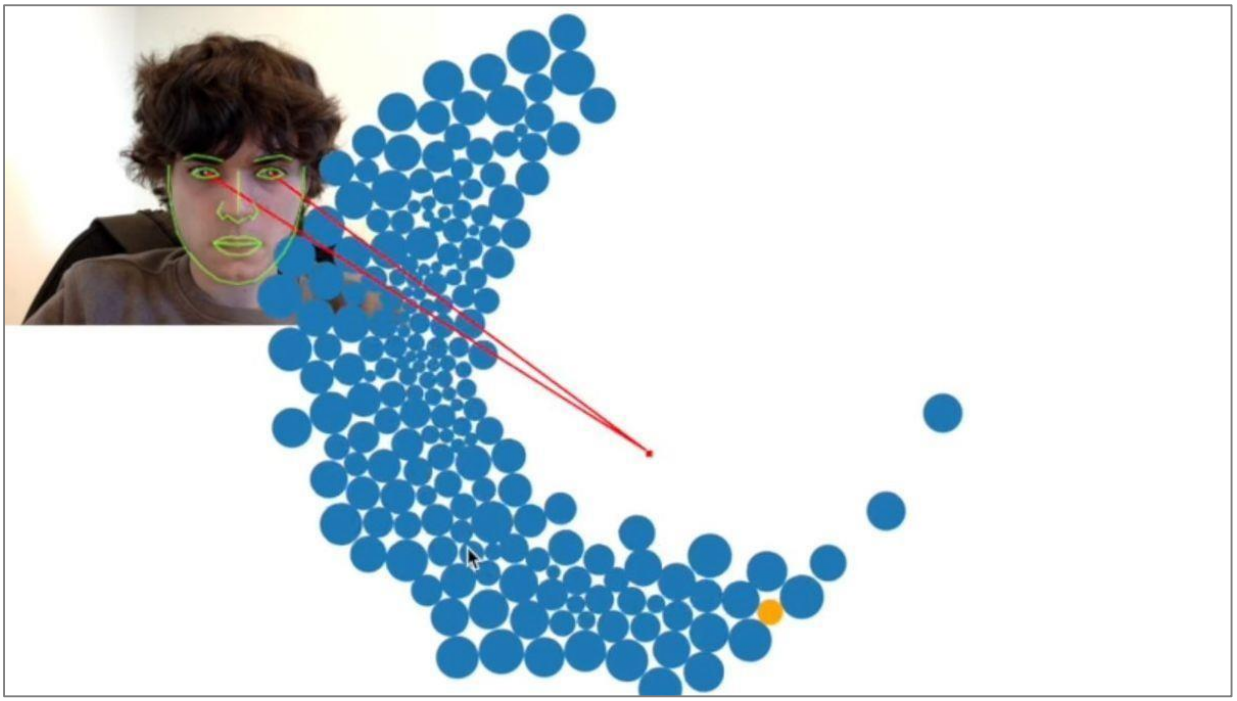

Рис 2.4.1 Пример калибровки WebGazer, для конкретного пользователя

Для базовой работы библиотеки достаточно встроенной камеры ноутбука или фронтальной камеры смартфона. Однако стоит учитывать, что точность eye-tracking при использовании обычной камеры будет ниже, чем при применении специализированных устройств (например, Tobii Eye Tracker).

Разработка прототипа протеза руки. Одним из перспективных направлений в области биомедицинской инженерии является разработка интеллектуальных протезов, управляемых с помощью электромиографических (EMG) сигналов. Эти протезы способны интерпретировать биоэлектрические сигналы, генерируемые мышцами, и преобразовывать их в команды для управления движением искусственной конечности (Бен-Ган И., 2021; Руссо М., Феррари А., 2022).

Принцип работы протеза

Работа протеза руки с ЭМГ-управлением строится по следующему алгоритму:

1. Сбор сигнала: EMG-датчики (например, MyoWare, Delsys) прикрепляются к поверхности кожи и регистрируют электрические импульсы мышц.
2. Фильтрация и усиление: полученные сигналы усиливаются, фильтруются от шумов и нормализуются.
3. Анализ и классификация: на основе полученного сигнала определяется намерение пользователя (сжать, разжать, повернуть и т.д.) с помощью алгоритмов машинного обучения или пороговых значений.
4. Передача команды на исполнительный механизм: в зависимости от результата анализа, микроконтроллер посылает команду сервоприводам протеза.
5. Обратная связь: возможно внедрение обратной тактильной или визуальной связи для пользователя.

Перспективы развития.

Развитие системы возможно по следующим направлениям: 3D-моделирования протезов и автоматизации расчётов.

- Инструменты совместной работы врачей через SignalR (чат, аннотации 3D-моделей).
- Автоматическая генерация моделей протезов на основе расчетов системы.
- Внедрение ИИ для прогнозирования оптимальных параметров протезов.
- Анализ медицинских данных (КТ/МРТ) с помощью компьютерного зрения.

Заключение.

Люди с инвалидностью часто сталкиваются с многочисленными барьерами, ограничивающими их участие в общественной жизни. К таким барьерам относятся физические препятствия, такие как недоступная инфраструктура, а также социальные и экономические факторы, включая дискриминацию, стигматизацию и ограниченный доступ к образованию и трудоустройству.

По сравнению с остальной массой населения люди с инвалидностью раньше умирают, имеют худшие показатели здоровья и сталкиваются с большими ограничениями в повседневной деятельности.

В результате проделанной работы была разработана полнофункциональная информационная система, построенная на базе современных веб-технологий. В качестве основы клиентской части была выбрана Blazor WebAssembly, а для серверной логики — ASP.NET Core API (Уайлдер Дж., 2022; Троелсен Э., Джапиксе Ф., 2022), что обеспечило высокую производительность, безопасное взаимодействие с базой данных и широкие возможности для масштабирования.

Одной из ключевых особенностей разработанной системы является наличие обширного информационного контента, направленного на ознакомление пользователей и пациентов с последними разработками в области протезирования. Контент структурирован в виде статей, новостей и справочных материалов, что облегчает доступ к нужной информации.

Административная панель является важным элементом системы. Она предоставляет административному персоналу доступ к расширенному функционалу управления контентом и пользователями: редактирование каталога протезов, управление заказами, публикация новостей, обработка заявок от клиентов и пользователей.

Еще одной важной особенностью является реализация универсального и удобного каталога. Каталог позволяет структурированно отображать элементы (например, модели протезов, материалы, комплектующие), используя фильтрацию и сортировку по различным критериям.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ:

1. Алиева Л. C# для начинающих: от основ к профессиональному программированию. — 360 с.
2. Шарп Д. MSSQL Server: запросы, хранимые процедуры и оптимизация. — 512 с.
3. Фримен А. Разработка облачных веб-приложений на ASP.NET Core 6 с использованием MVC, Blazor и Razor Pages. — 1080 с.
4. Уайлдер Дж. Blazor в действии. — 350 с.
5. Бен-Ган И. Оконные функции T-SQL для анализа данных и не только. — 400 с.
6. Руссо М., Феррари А. Полное руководство по DAX для Microsoft Power BI, SQL Server Analysis Services. — 600 с.
7. Бен-Ган И. Запросы T-SQL. — 800 с.
8. Коггинс Б. Тонкости работы с MSSQL Server: Администрирование и оптимизация. —

560 с.

9. Троелсен Э., Джапиксе Ф. Профессиональное программирование на C# 10 с использованием .NET 6: Основные принципы и практика. — 1705 с.

10. Картер П. Профессиональное администрирование SQL Server 2022: Руководство современного администратора баз данных. — 800 с.