

Министерство образования Иркутской области

Государственное бюджетное профессиональное образовательное учреждение

Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

ДП.09.02.07-3.24.201.05. ПЗ

УТВЕРЖДАЮ

Зам. директора по УР, к.т.н.

_____ Е.А. Коробкова

**МОДУЛЬ МЕДИЦИНСКОЙ ИНФОРМАЦИОННО-
АНАЛИТИЧЕСКОЙ СИСТЕМЫ «ЛИЧНЫЙ КАБИНЕТ
ПАЦИЕНТА»**

Нормоконтролер:

(подпись, дата)

(Е.С. Кудрявцева)

Руководитель ДП:

(подпись, дата)

(П.Н. Чернигов)

Студент:

(подпись, дата)

(И.А. Григорьев)

Иркутск 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Предпроектное исследование	4
1.1 Описание предметной области.....	4
1.2 Анализ инструментальных средств разработки	5
2 Техническое задание на разработку программного продукта	10
3 Проектирование программного продукт	11
3.1 Архитектура программного продукта	11
3.2 Функциональное и структурное проектирование	12
3.3 Проектирование баз данных	15
3.4 Проектирование пользовательского интерфейса программного продукта	20
4 Реализация программного продукта	27
4.1 Разработка интерфейса пользователя	27
4.2 Разработка базы данных.....	31
4.3 Разработка модуля медицинской информационно-аналитической системы .	32
5 Тестирование программного продукта.....	37
6 Документирование программного продукта.....	40
ЗАКЛЮЧЕНИЕ	44
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	45
Приложение А – ER-модель.....	47
Приложение Б – Листинг кода страницы профиля.....	48
Приложение В – Листинг кода контроллера профиля	52

					ДП.09.02.07-3.24.201.05. ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Григорьев И.А.			Модуль медицинской информационно- аналитической системы «Личный кабинет пациента» пояснительная записка	Лит.	Лист	Листов
Провер.		Чернигов П.Н.					2	54
Реценз.						ГБПОУИО «ИАТ» ВЕБ-20-1		
Н. Контр.		Кудрявцева Е.С.						
Утверд.		Коробкова Е.А.						

ВВЕДЕНИЕ

Информационные технологии имеют огромное влияние на различные сферы нашей жизни, где медицина не является исключением, а даже наоборот наиболее востребована. В последние годы наблюдается значительный прогресс в разработке информационных систем, но доступность медицинского обслуживания не слишком изменилась за все время по причине того, что системы не соответствовали потребностям предприятия. Но с развитием технологий сейчас это стало доступным. Одной из таких систем является "Личный кабинет пациента".

“Личный кабинет пациента” – это модуль медицинской информационно-аналитической системы, предоставляющая пациентам возможность получать доступ к своим медицинским данным, записываться на приемы к врачам, получать результаты по прошедшим исследованиям и контролировать свое здоровье. Эта система эффективно объединяет пациентов и медицинские учреждения, упрощая процесс взаимодействия и повышая уровень качества медицинской помощи.

Целью дипломного проекта являлась разработка модуля информационной системы, которая позволила бы упростить наблюдение пользователей за своим состоянием здоровья и облегчить возможность записи на прием.

Основными задачами данной работы являются:

- провести исследование деятельности личных кабинетов мед учреждений;
- на основании полученных знаний спроектировать модель деятельности;
- разработать техническое задание;
- провести анализ программных средств разработки;
- обосновать выбор программной среды разработки;
- провести анализ инструментов интеграции;
- составить программную документацию.

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

1 Предпроектное исследование

1.1 Описание предметной области

Модуль медицинской информационно-аналитической системы «Кабинет пациента» предназначена для значительного повышения доступности и эффективности медицинского обслуживания.

Предметная область данного модуля информационной системы включает в себя процессы, связанные с просмотром личных медицинских данных, а также возможность записи на прием.

На предприятии, которое будет использовать данный модуль медицинской информационно-аналитической системы, основным процессом является процесс оказания медицинских услуг.

Основными объектами в предметной области являются:

1) Пациенты – внесенные в базу пользователи, которые могут просматривать свои личные и медицинские данные, а также имеющие возможность записываться на прием и просматривать результаты обследований;

2) Медицинские заключения - данные о состоянии здоровья пациента, включая анализы, диагнозы, назначения врачей и результаты обследований, которые хранятся в личном кабинете пациента.

3) Врачи - медицинские специалисты, чьи расписания работы, специализации и контактная информация доступны пациентам для записи на консультации и приемы.

4) Запись на прием - функция, позволяющая пациентам выбирать врача, указывать желаемое время и причину обращения для записи на прием через личный кабинет.

5) Результаты обследований - информация о проведенных исследованиях, их результатах и рекомендациях врачей, доступная пациентам для ознакомления и скачивания.

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

Таким образом модуль медицинской информационно-аналитической системы «Личный кабинет пациента» представляет собой инструмент для помощи в наблюдении за здоровьем и медицинскими данными пациентов, обеспечивая им удобный доступ к услугам медицинского учреждения и результатам этих услуг тем самым повышая качество медицинского обслуживания. Так же это поможет специалистам упростить работу с пациентами путем более простого обнародования результатов исследования, медицинского заключения, рецептов, а также поможет узнать нужную информацию о пациенте путем просмотра личного кабинета пользователя, где можно найти информацию о: фамилии, имени, отчестве, номере телефона, электронной почте, прошедших и будущих исследованиях пациента.

1.2 Анализ инструментальных средств разработки

Для разработки программного продукта с базой данных выбраны следующие инструментальные средства разработки программных продуктов:

- 1) PhpMyAdmin.
- 2) SQL.
- 3) Visual Studio Code.
- 4) PHP.
- 5) Draw.io.
- 6) HTML.
- 7) CSS.

PhpMyAdmin – программа, предназначенная для администрирования системы управления базы данных MySQL. PhpMyAdmin использует браузер для осуществления администрирования сервера MySQL, позволяет запускать команды SQL и просматривать данные таблиц и баз данных.

SQL – это язык программирования структурированных запросов, который используется в качестве эффективного способа сохранения данных, поиска их частей, обновления, извлечения из базы и удаления.

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

Основные преимущества:

- оптимальный для повседневных задач набор функционала;
- работает на стороне сервера (через браузер).

HTML (HyperText Markup Language) – язык разметки (маркировки) гипертекста. Язык разметки дает браузеру необходимые инструкции о том, как отображать тексты и другие элементы страницы на мониторе. Язык HTML интерпретируется браузерами и отображается в виде документа, в удобной для человека форме.

Основные 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.

Visual Studio Code (VS Code) – это бесплатный и открытый исходный код редактор кода, который может быть использован для анализа программных продуктов. Он разработан Microsoft и поддерживает множество языков программирования, включая JavaScript, Python, C++, Java и многие другие.

VS Code обладает множеством функций, которые делают его удобным инструментом для анализа программных продуктов. Он имеет встроенную поддержку Git, что позволяет легко работать с репозиториями и контролировать версии кода. Кроме того, VS Code поддерживает отладку кода, что позволяет быстро находить и исправлять ошибки.

PHP (Hypertext Preprocessor) – «Инструменты для создания персональных веб-страниц» - серверный 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.

Основные преимущества:

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

- 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.

Язык CSS (Cascading Style Sheets – каскадные таблицы стилей) описывает то, как HTML-элементы отображаются на странице: фон, отступы, цвета и так далее. Помимо дизайна элементов, CSS отвечает за отображение на устройствах с различной шириной экрана. CSS позволяет создавать классы, идентификаторы или специфицировать внешний вид для конкретного типа объектов (параграфы, таблицы и так далее).

JavaScript – это полноценный динамический язык программирования, который применяется к HTML документу, и может обеспечить динамическую интерактивность на веб-сайтах.

MySQL – свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle.

MySQL Workbench – инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL.

Draw.io – это онлайн-сервис для создания диаграмм и схем, который может быть использован для анализа программных продуктов. С помощью draw.io можно

создавать различные типы диаграмм, такие как блок-схемы, диаграммы классов, диаграммы последовательностей и другие.

Вывод, для создания программного обеспечения было решено использовать следующие технологии Visual Studio Code, draw.io, MySQL и MySQL Workbench.

Для выбора инструментов разработки были произведены сравнения, результат которых представлены в таблицах 1 и 2.

Таблица 1 – Сравнение IDE для разработки программного продукта

Название IDE	Visual studio Code	Notepad++,	JetBrains PhpStorm	JetBrains WebStorm
Бесплатность	- (есть Бесплатная студенческая Лицензия)	+	- (есть Бесплатная студенческая Лицензия)	- (есть Бесплатная студенческая Лицензия)
Автосохранение	+	-	+	+
Автодополнение	+	-	+	+
Интеграция с системами управления версиями(git)	+	-	+	+
Поиск по коду	+	-	+	+

Таким образом, после рассмотрения средств разработок, было принято решение использовать Visual Studio Code, так как оно просто в освоении, обладает большим количеством функций, так же, помимо этого, на выбор повлиял трехлетний опыт использования данного инструмента.

Таблица 2 – Сравнение языков программирования для разработки программного продукта

Название языка программирования	PHP	Python
Популярность	Язык, выбираемый для веб-разработки	Широко используется в искусственном интеллекте, науке о данных и научном сообществе
Подключение к базе данных	К более 20 различным БД	Oracle, SQL Server, MySQL и PostgreSQL

Продолжение таблицы 2

Название языка программирования	PHP	Python
Читабельность	Хорошо документирован	Использует очень строгие правила отступов
Фреймворки	Большее количество	Меньшее количество
Синтаксис	Понятный	Понятный

Таким образом, из представленных языков программирования можно выделить PHP, так как он имеет возможность интегрирования к более 20 различным базам данных, крупное сообщество которое может помочь почти с любыми возникшими вопросами, а также благодаря двухлетнему опыту разработки на данном языке программирования.

Подводя итоги, все инструментальные средства для разработки и проектирования были выбраны согласно оптимальным показателям и удобству использования.

2 Техническое задание на разработку программного продукта

Техническое задание на разработку программного продукта – это документ, определяющий требования к ИТ-продукту, включая его назначение, функции, поведение, используемые компоненты, технологии, инструменты разработки, а также порядок выполнения работ. ТЗ служит руководством для бизнес- и технических групп, занимающихся созданием ИТ-решения.

В техническое задание входит:

- 1) Общие сведения.
- 2) Назначение и цели создания модуля системы.
- 3) Требование к системе в целом.
 - 3.1) Требование к структуре и функционированию модуля системы.
 - 3.2) Требование к надежности.
 - 3.3) Требования к безопасности.
 - 3.4) Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов модуля системы.
- 4) Требование к документированию.
- 5) Состав и содержание работ по созданию модуля системы.

Более подробно ознакомится с техническим заданием на разработку программного продукта можно в отдельном файле под названием «Техническое задание модуля информационной системы «Личный кабинет пациента»».

ПЗ оформляется в соответствии с ГОСТ 34.602-202.

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

3 Проектирование программного продукт

3.1 Архитектура программного продукта

Архитектура программного продукта – это базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы.

Для работы приложения было решено использовать клиент-серверную архитектуру (рисунок 1). Работа клиент-серверной архитектуры построена на взаимодействии между клиентом и сервером. Клиент отправляет запросы на получение данных с сервера, а сервер проводит проверку доступа к информации и формирует запрос в базу данных для получения, обработки и отправки необходимой информации клиенту.

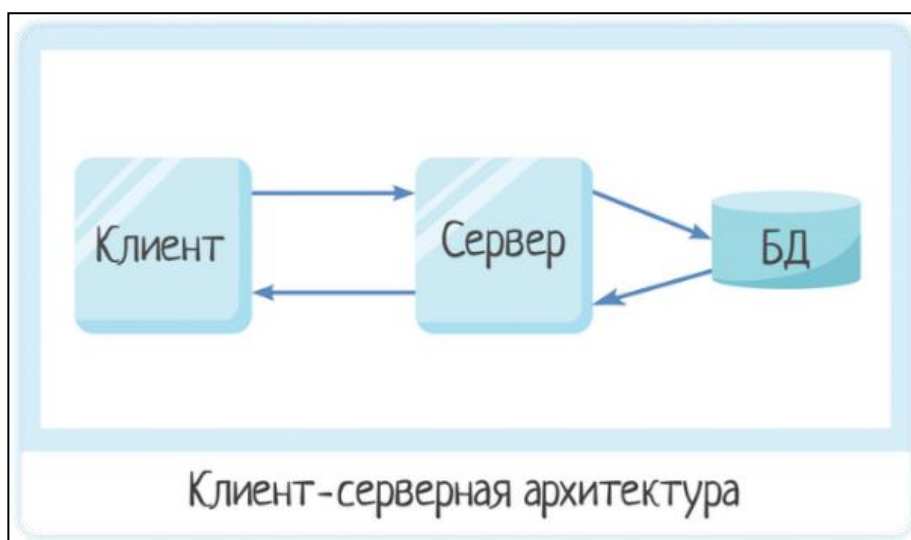


Рисунок 1 – Архитектура модуля медицинской информационно-аналитической системы «Личный кабинет пациента»

Сервер – специальное системное оборудование, которое предназначается для разрешения определенного круга задач по процессу выполнения программных кодов. Он выполняет работы сервисного обслуживания по клиентским запросам, предоставляет пользователям доступ к определенным системным ресурсам, сохраняет данные или БД.

Параметры, которые могут реализоваться на стороне сервера:

- 1) хранение и доступ к данным;
- 2) работа с поступающими клиентскими запросами;
- 3) процесс отправки ответа клиенту.

Клиент – локальный компьютер на стороне виртуального пользователя, который выполняет отправки запроса к серверу для возможности предоставления данных или выполнения определенной группы системных действий.

Параметры, которые могут реализоваться на стороне клиента:

- 1) формулировка запроса к серверу и его последующая отправка;
- 2) получение итогов запроса и отправка дополнительной группы команд (запросы на добавление, обновление информации, удаление группы данных).

3.2 Функциональное и структурное проектирование

Модуль медицинской информационно-аналитической системы «Кабинет пациента» предназначен для автоматизации работы медицинского учреждения и упрощения процесса взаимодействия между пациентами и медицинским персоналом. Функциональное проектирование данной системы включает в себя следующие шаги:

- 1) Определение целей продукта:
 - Обеспечение удобного доступа пациентов к информации о своем здоровье и истории болезни.
 - Уменьшение времени, затрачиваемого на запись к врачу и получение результатов анализов.
 - Снижение нагрузки на медицинский персонал за счет автоматизации процессов.
- 2) Идентификация функций:
 - Хранение информации о пациентах (анкетные данные, история болезни, результаты обследований и лечения).

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

- Возможность записи на прием к врачу онлайн.
- Просмотр расписания врачей и свободных ячеек времени для записи.
- Просмотр результатов анализов и исследований.

3) Анализ взаимосвязей:

– Данные пациентов должны быть доступны только для медицинского персонала, имеющего соответствующие права доступа.

– Возможность записи на прием к врачу должна зависеть от свободных ячеек времени в расписании врача.

– Результаты анализов и исследований должны быть доступны только пациенту и врачу, назначившему их.

4) Детализация функций:

– Возможность записи на прием к врачу должна включать в себя выбор врача и свободного времени для записи.

– Просмотр расписания врачей должен предоставляться с учетом специализации врача и его расписания.

– Результаты анализов и исследований должны быть доступны в виде отчетов с возможностью скачивания.

5) Создание функциональных спецификаций:

– Все функции продукта должны быть описаны в документации.

– Каждая функция должна содержать информацию о входных и выходных данных, условиях выполнения и прочие детали.

6) Проверка и корректировка:

– Провести тестирование всех функций системы на соответствие поставленным целям и требованиям.

– Внести необходимые корректировки в случае обнаружения недостатков.

Более подробно это показано в диаграммах прецедентов (рисунок 2) и диаграмме деятельности (рисунок 3).

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

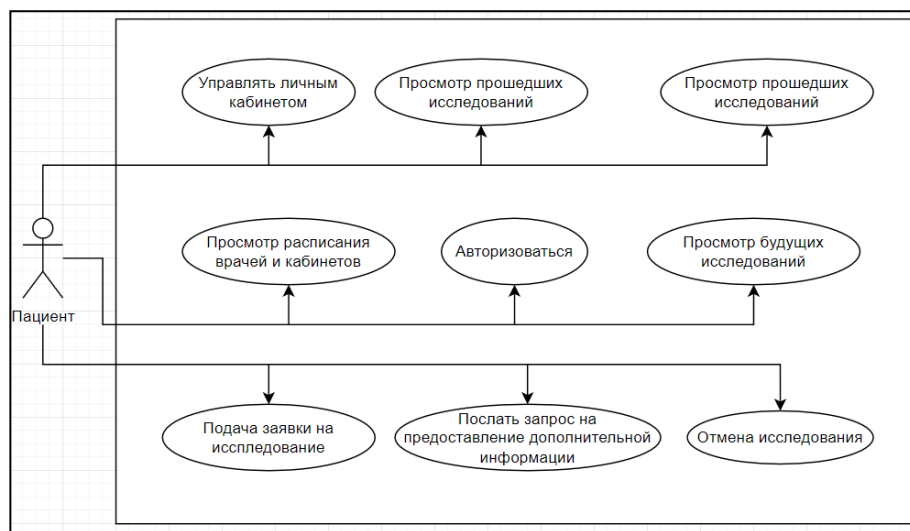


Рисунок 2 – Диаграмма прецедентов

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.

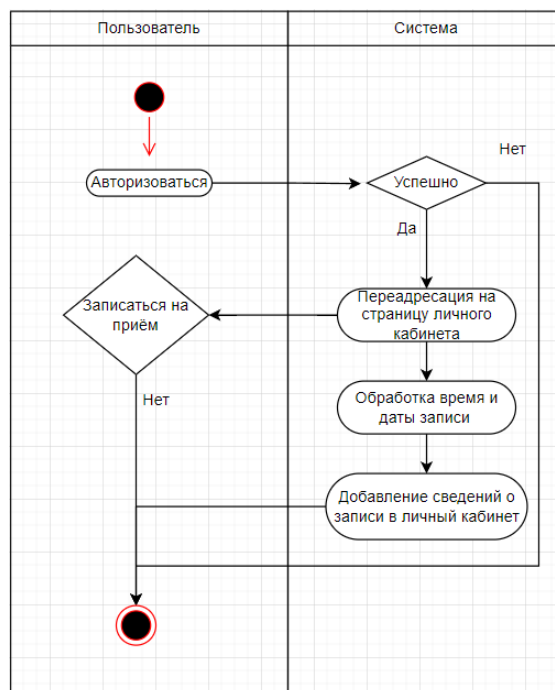


Рисунок 3 – Диаграмма деятельности

В итоге проектирования диаграммы деятельности были выделены основные возможные действия пользователя с программным продуктом.

В результате функционального проектирования должен быть создан модуль медицинской информационно-аналитической системы «Кабинет пациента», которая обеспечит удобный доступ к медицинским услугам для пациентов, а также упростит работу медицинского персонала.

3.3 Проектирование баз данных

Прежде чем приступить разработке программного обеспечения необходимо спроектировать базу данных, а именно, определить с какими данными будут работать участники системы, и чем данные связаны между собой. В этом заключается процесс проектирования.

Проектирование базы данных не ограничивается только этим этапом разработки, а проходит на протяжении всей разработки, до того момента пока в системе не появятся данные, которые нельзя потерять. Результатом проектирования базы данных является ER-модель которая представлена в приложении А. Она содержит множество таблиц возьмем несколько из них которые были задействованы в разработке модуля информационно-аналитической системы «Личный кабинет пациента»:

Таблица 3 – сущность «mskt_research_method»

Название	Тип поля	Описание
method_id	int unsigned	Идентификатор (PK)
short_name	varchar(45)	Укороченное название
name	varchar(100)	Название

Таблица 4 – сущность «mskt_employee»

Название	Тип поля	Описание
employee_id	int unsigned	Идентификатор (PK)
first_name	varchar(100)	Имя
last_name	varchar(100)	Фамилия
fathers_name	varchar(100)	Отчество

Таблица 5 – сущность «mskt_research_methodology»

Название	Тип поля	Описание
methodology_id	int unsigned	Идентификатор (PK)
method_id	int unsigned	Идентификатор метода (FK)
name	varchar(250)	Название
short	varchar(50)	Короткое название
contrast	tinyint(1)	Контраст

Таблица 6 – сущность «mskt_cabinet_method»

Название	Тип поля	Описание
cabinet_id	int unsigned	Идентификатор кабинета (FK)
method_id	int unsigned	Идентификатор метода (FK)

Таблица 7 – сущность «mskt_cabinets»

Название	Тип поля	Описание
cabinet_id	int unsigned	Идентификатор (PK)
number	smallint unsigned	Номер
name	varchar(150)	Название

Таблица 8 – сущность «mskt_cabinet_schedule»

Название	Тип поля	Описание
id	int unsigned	Идентификатор (PK)
cabinet_id	int unsigned	Идентификатор кабинета (FK)
date	date	Дата
time_start	time	Время начала
time_end	time	Время конца

Таблица 9 – сущность «mskt_employee_qualification»

Название	Тип поля	Описание
employee_id	int unsigned	Идентификатор работника (FK)
methodology_id	int unsigned	Идентификатор зоны (FK)
qualification_id	int unsigned	Идентификатор квалификации (FK)
type	int unsigned	Тип данных

Таблица 10 – сущность «mskt_employee_schedule»

Название	Тип поля	Описание
employee_id	int unsigned	Идентификатор работника (PK)
time_start	time	Время начала
time_end	time	Время конца

Таблица 11 – сущность «mskt_employee_post»

Название	Тип поля	Описание
employee_id	int unsigned	Идентификатор работника (FK)
post_id	int unsigned	Идентификатор должности (FK)
first_price	float	Первая цена
second_price	float	Вторая цена
time	time	Время

Таблица 12 – сущность «mskt_pacients»

Название	Тип поля	Описание
patient_id	int unsigned	Идентификатор (PK)
login	varchar(250)	Логин пользователя
password	varchar(250)	Пароль пользователя
last_name	varchar(50)	Фамилия
first_name	varchar(50)	Имя
father_name	varchar(50)	Отчество
sex	enum('m','f')	Пол
birthday	date	Дата рождения
phone	varchar(45)	Номер телефона
email	varchar(45)	Электронная почта

Таблица 13 – сущность «mskt_posts»

Название	Тип поля	Описание
method_id	int unsigned	Идентификатор метода (FK)
post_id	int unsigned	Идентификатор (PK)

Таблица 14 – сущность «mskt_research_schedule»

Название	Тип поля	Описание
employee_id	int unsigned	Идентификатор работника (FK)
method_id	int unsigned	Идентификатор метода (FK)
methodology_id	int unsigned	Идентификатор зоны (FK)
patient_id	int unsigned	Идентификатор пациента (FK)
date	date	Дата
time_start	time	Время начала
time_end	time	Время конца
cancel_time	time	Время отмены
cancel_status	int unsigned	Вторая цена
cancel_note	varchar(200)	Время

Таким образом, можно увидеть всю необходимую информацию для понимания системы хранения данных.

3.4 Проектирование пользовательского интерфейса программного продукта

Пользовательский интерфейс – это средства взаимодействия между человеком и компьютером. Иными словами, интерфейс – внешняя часть программы или устройства, с которыми работает пользователь.

Интерфейсы являются основой взаимодействия всех современных информационных систем. Если интерфейс какого-либо объекта не изменяется (стабилен, стандартизирован), это даёт возможность модификации, не перестраивая принципы его взаимодействия с другими объектами.

В качестве цветовой гаммы были выбраны следующие цвета и их оттенки: красный, желтый, синий и зеленый. Подобраны они были при помощи сайта Palletton.com который позволяет быстро и легко определить подходящую к друг другу палитру цветов. Следующим этапом было подобрать типографику которая соответствует тематике сайта. Для того чтобы соответствовать были выбран шрифт Peace Sans по причине того, что он лаконично смотрелся на странице и не выбивался из общего концепта проекта. На рисунке 5 можно наблюдать более подробную подборку шрифта и цветовой палитры.

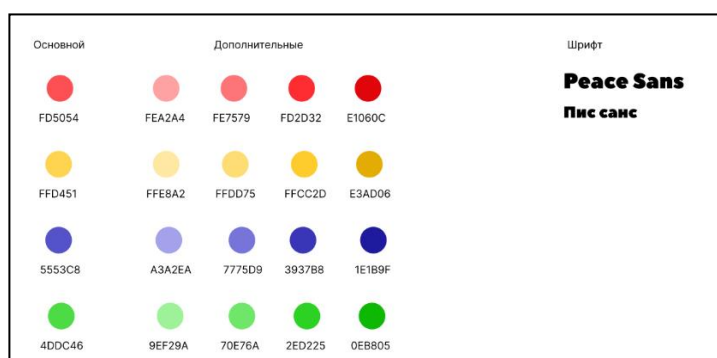


Рисунок 5 – Цветовая гамма и типографика

Так же не стоит забывать и про элементы интерфейса. В данном случае элементами интерфейса будут являться кнопки для отправки запросов и выглядеть они будут согласно примеру на рисунке 6.

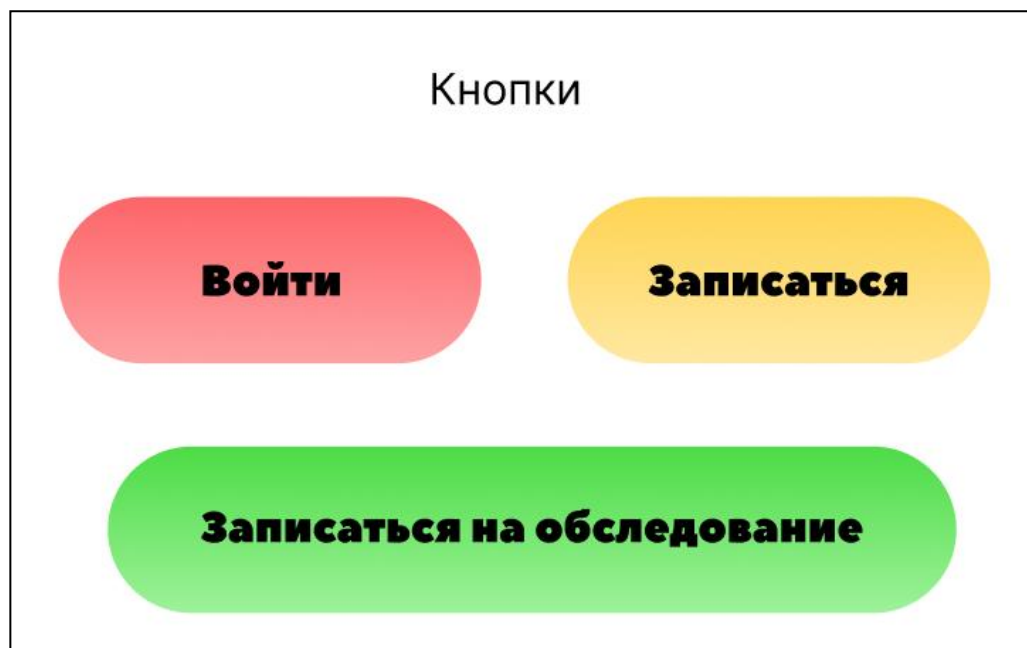


Рисунок 6 – Элемент интерфейса «Кнопка»

Так же для наглядного и корректного отображения статусов были добавлены иконки. Они позволят более корректно отображать статусы обследований. Часы означают что текущее обследование еще не завершено и ожидает пациента или решения врача, а галочка что обследование уже завершено и вы можете его скачать нажав на иконку pdf (рисунок 7).

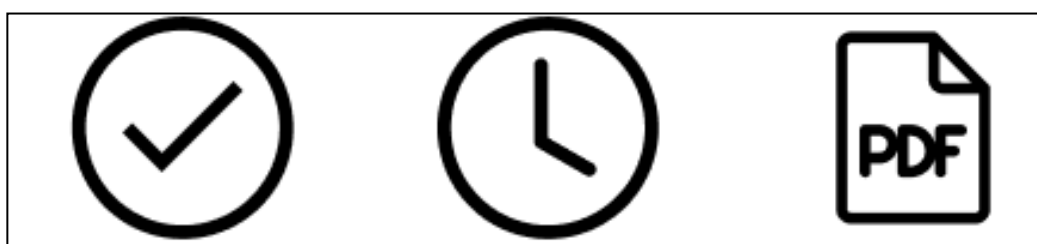


Рисунок 7 – Элементы интерфейсы «Иконки»

Следующим этапом были прототипы интерфейса для страниц авторизации пользователей, личного профиля, записи на прием. Построены они были в онлайн конструкторе Figma.

Первым в списке идет страница авторизации пользователей она представлена на рисунке 8.

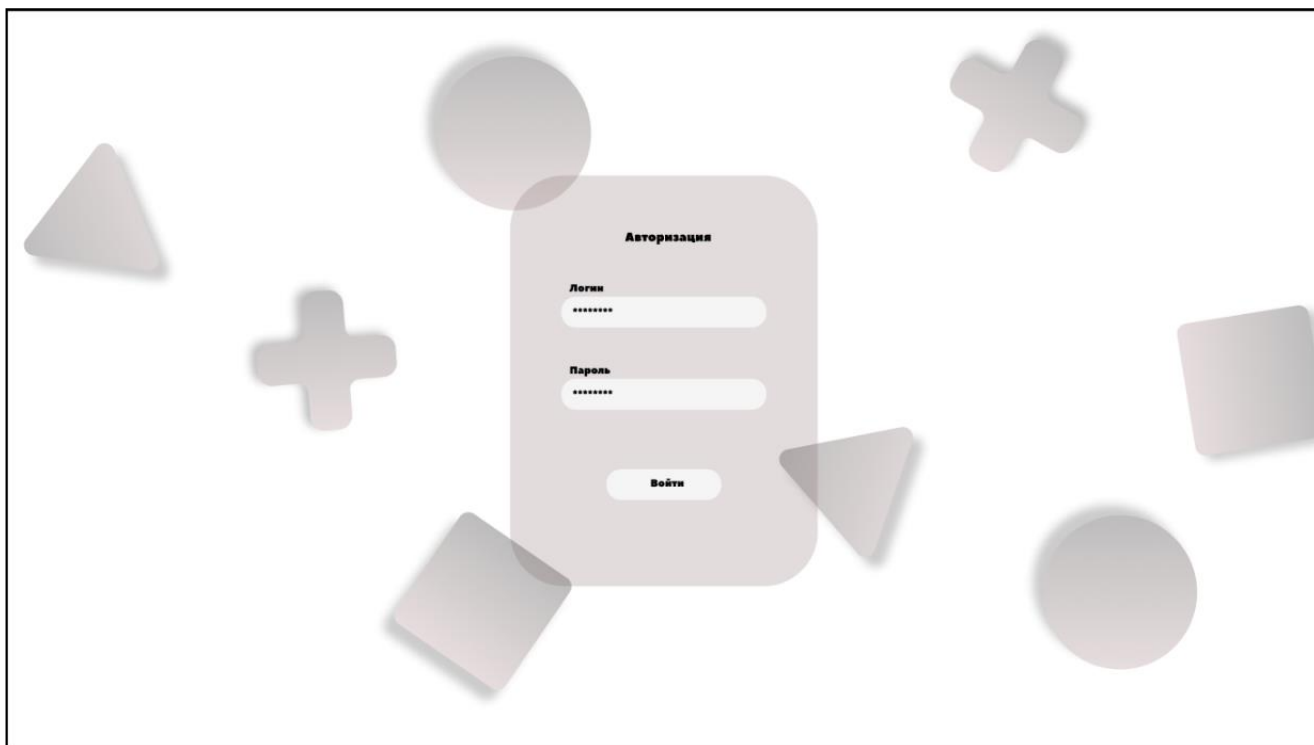


Рисунок 8 – Макет страницы «Авторизация»

Авторизация пользователей одна из ключевых функций необходимых для работы модуля медицинской информационно-аналитической системы «Личный кабинет пациента», позволяющая защитить свои личные данные, не предназначенные для просмотра другими лицами. На данном макете представлена форма для входа в аккаунт пользователя позволяющая сверять данные с имеющимися в базе и присваивать токен сессии для дальнейшего доступа к работе с модулем медицинской информационно-аналитической системы. Обозначенные на макете элементы включают в себя поля ввода для логина пользователя и пароля.

Второй макет будет предназначен для пользователя, который успешно прошел процесс авторизации и был переадресован на страницу с его обследованиями, она в свою очередь изображена на рисунке 9.

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						22
Изм.	Лист	№ докум.	Подпись	Дата		

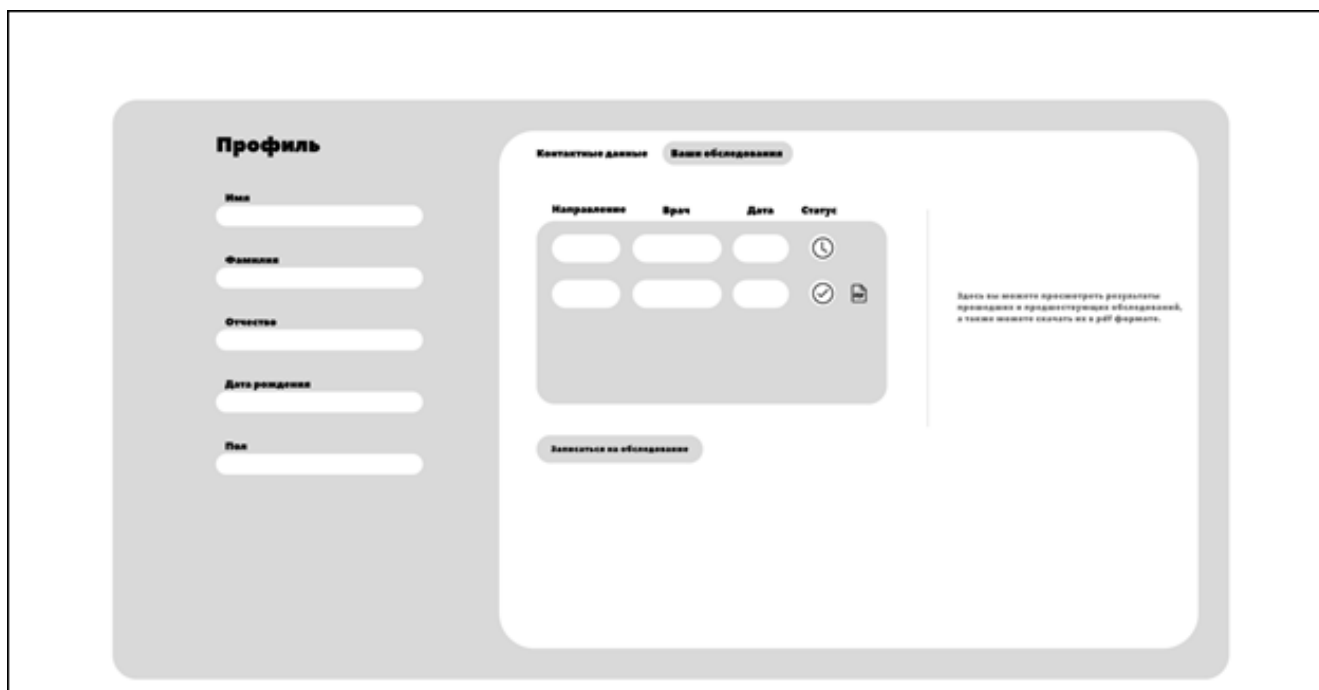


Рисунок 9 – Макет страницы «Профиль пользователя» вкладка «Обследования»

На данном макете пользователю представлены данные о его текущих, прошедших и будущих обследованиях. Так же на этой странице имеется возможность скачать уже завершившееся обследование или отменить еще не начатое. Эти функции доступны по нажатию на кнопку «Скачать заключение», если пользователю нужны результаты, или «Отменить», если пользователю нужно отменить заявку на обследования.

Помимо этого, на вкладке «Обследования» имеется кнопка, которая переадресовывает пользователя на страницу «Запись на прием» где он имеет возможность заполнить поля и оставить заявку на обследование (рисунок 10).

Рисунок 10 – Макет страницы «Запись на прием»

На данном макете отображены поля нужные для корректной записи на прием, а именно:

- Направленность обследования;
- Метод обследования;
- Врач доступный для данного обследования;
- Дата обследования;
- Время обследования;

После заполнения всех нужных полей, без каких либо ошибок, пользователь должен нажать на кнопку «Записаться» и тогда его переадресует на страницу с его обследованиями, где уже и отобразится запись.

Если же у пользователя была цель просмотреть свои данные или изменить их, то он может перейти во вкладку «Личные данные» (рисунок 11).



Рисунок 11 – Макет страницы «Профиль пользователя» вкладка «Контактные данные»

На этом макете будут отображаться личные данные авторизованного пользователя такие как:

- ФИО;
- дата рождения;
- пол;
- номер телефона;
- электронная почта.

После изменения нужных пользователю данных, он нажимает на кнопку «Сохранить» и нужные данные сохраняются для пользователя.

Если же пользователь хочет посмотреть время работы врачей, то он обращается ко вкладке «Расписание врачей» (рисунок 12). В ней он аналогично вкладке «Запись на прием» вводит нужную информацию о методе, направлении, и самом враче. И после выбора даты ему высвечивается время работы выбранного пользователем врача.

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		

Рисунок 12 – Код страницы «Расписание врачей»

Более подробно с возможностями перехода по страницам можно ознакомиться в диаграмме, навигации которая находится в отдельном файле под названием «Схема навигации модуля медицинской информационно-аналитической системы», который приложен в папке «docs».

4 Реализация программного продукта

4.1 Разработка интерфейса пользователя

В данном проекте используются четыре страницы, а именно: страница авторизации, страница профиля, страница записи на обследование, страница просмотра расписания врачей. Была разобрана одна из страниц, а именно страницу профиля (рисунок 13).

```
return (  
  <AuthenticatedLayout  
    user={auth.user}  
  >  
    <Head title="Профиль" />  
  
    <div className="py-12">  
      <div className="max-w-9xl mx-auto sm:px-6 lg:px-8 space-y-6">  
        <Box sx={{ width: '100%' }} className="">  
          <Box sx={{}}>  
            <Tabs value={value} onChange={handleChange} aria-label="basic tabs example" className="ml-[80px]">  
              <Tab label="Ваши обследования" {...allyProps(1)} />  
              <Tab label="Личные данные" {...allyProps(0)} />  
              <Tab label="Расписание врачей" {...allyProps(2)} />  
            </Tabs>  
          </Box>  
        </Box>  
        <CustomTabPanel value={value} index={0}>  
          <div className="p-4 sm:p-8 bg-white shadow sm:rounded-lg">  
            <MyExam className="max-w-xl"  
              guzle={guzle}  
              workTime={workTime} />  
          </div>  
        </CustomTabPanel>  
        <CustomTabPanel value={value} index={1}>  
          <div className="p-4 sm:p-8 bg-white shadow sm:rounded-lg">  
            <UpdateProfileName  
              className="max-w-xl"  
            />  
          </div>  
        </CustomTabPanel>  
        <CustomTabPanel value={value} index={2}>  
          <div className="p-4 sm:p-8 bg-white shadow sm:rounded-lg">  
            <ScheduleEmployee  
              className="max-w-xl"  
              workTime={workTime}  
              methods={methods}  
              methodology={methodology}  
              sheduleOkTime={sheduleOkTime}  
            />  
          </div>  
        </CustomTabPanel>  
      </div>  
    </div>  
  </AuthenticatedLayout>  
);
```

Рисунок 13 – Код страницы «Профиль»

Как и другие страницы, эта начинается с тега который подключает стартовую структуру HTML и позволяет не прописывать это в ручную в каждом файле – <AuthenticadeLayout>. Так же помимо этого он содержит в себе навигацию.

Следом из важных аспектов идут теги подключенные из библиотеки MUI или material UI под названием <Box> и <Tabs>, нужны они для отображения вкладок. Вкладки – это элемент графического интерфейса, который дает возможность переключения между открытыми окнами. На данный момент на странице открыты две вкладки при переходе на страницу: Личные данные и ваши обследования. Их код представлен на рисунке 14 и 15.

```
return (
  <section>
    <header className="flex w-auto text-clip">
      <h2 className="text-lg font-medium text-gray-900 w-1/2">Информация  профиле</h2>
    </header>

    <form onSubmit={submit} className="mt-6 space-y-6" >
      <div className="flex ">
        <div>
          <InputLabel htmlFor='first_name' value='Имя' />
          <TextInput
            id='first_name'
            className="mt-1 block w-full "
            value={data.first_name}
            autoComplete='first_name'
            disabled
          />
        </div>
        <div className="ml-10">
          <InputLabel htmlFor='last_name' value='Фамилия' />
          <TextInput
            id='last_name'
            className="mt-1 block w-full "
            value={data.last_name}
            autoComplete="last_name"
            disabled
          />
        </div>
      </div>
      <div>
        <InputLabel htmlFor='father_name' value='Отчество' />
        <TextInput
          id='father_name'
          className="mt-1 block "
          value={data.fathers_name}
          autoComplete="father_name"
          disabled
        />
      </div>
      <div className="flex ">
        <div>
          <InputLabel htmlFor='birthday' value='Дата рождения' />
          <TextInput
            id='birthday'
            className="mt-1 block w-full"
            value={data.birthday}
            autoComplete="birthday"
            disabled
          />
        </div>
      </div>
    </form>
  </section>
)
```

Рисунок 14 – Код страницы «Профиль» вкладка «Личные данные»

Здесь основным элементом является HTML-тег <form>, в совокупности с <input> и <button>, так как на странице есть данные которые пользователь может изменить, по тем или иным причинам, он может ввести их в поле и нажать на кнопку для сохранения данных. Также помимо этого в коде присутствует тег от библиотеки Breeze <InputLabel>, она выполняет функцию направляющей, чтобы указать какое поле и какие данные содержит.

```
return (
  <section>
    <header className="flex w-auto text-clip">
      <h2 className="text-lg font-medium text-gray-900 w-1/2">Информация о ваших обследованиях</h2>
    </header>

    <a href="/createExam" className="btn btn-primary">
      <PrimaryButton>
        Записаться на прием
      </PrimaryButton>
    </a>

    <div className="container space-x-10 mt-10 bg-gray-200 rounded-lg p-8 grid text-center">
      <table className="table-auto justify-center">
        <thead>
          <tr>
            <th>Дата</th>
            <th>Время</th>
            <th>Метод</th>
            <th>Зона</th>
            <th>Врач</th>
            <th>Контраст</th>
            <th>Статус</th>
          </tr>
        </thead>
        <tbody>
          {ExTime}
        </tbody>
      </table>
    </div>
  </section>
)
```

Рисунок 15 – код страницы «Профиль» вкладка «Мои обследования»

Следующая вкладка представляет из себя таблицу с данными об обследованиях созданную при помощи HTML-тега <table> и сопутствующие ему <thead>, <tr> и <th>.

а так же элемент <button> который выполняет функцию перехода на страницу с записью на обследование.

Заключительной частью стала вкладка с отображением времени работы врачей. Она схожа с кодом страницы «Запись на прием», но с некоторыми изменениями. На данной странице отсутствует форма, цель которой обозначить поля для заполнения, а также кнопка для отправки данных на сервер (рисунок 16).

```
return (
  <section>

    <div className="py-12">
      <div className="max-w-9xl mx-auto sm:px-6 lg:px-8 space-y-6">
        <header className="flex w-auto text-clip">
          <h2 className="text-lg font-medium text-gray-900 w-1/2">Расписание врачей</h2>
        </header>

        <Head title="CreateExam" />
        <div className="flex ">

          <div>
            <InputLabel htmlFor='method' value='Метод обследования' />
            <Autocomplete
              disablePortal
              id="method"
              name='method'
              options={methodList}
              onChange={(e, newValue) => {
                setData('methoData', newValue ? newValue.id : null);
              }}
              isOptionEqualToValue={(option, value) => option.id === value}
              className="mt-5 block w-full"
              sx={{ width: 300 }}
              renderInput={(params) => <TextField {...params} label="Выберите метод обследования" />}
              required
            />
          </div>

          <div className="ml-10">
            <InputLabel htmlFor='methodology' value='Направление' />
            <Autocomplete
              disablePortal
              id="methodology"
              name='methodology'
              options={methodologyList}
              className="mt-1 block w-full mt-5"
              onChange={(e, newValue) => {
                setData('methodologyData', newValue ? newValue.id : null);
              }}
              isOptionEqualToValue={(option, value) => option.id === value}
              sx={{ width: 300 }}
              renderInput={(params) => <TextField {...params} label="Выберите направление" />}
              required
            />
          </div>

          <div>
            <InputLabel htmlFor='employ' value='Врач' />
            <Autocomplete
              disablePortal
              id="employ"
              name='employ'
              options={EmployerList}
              className="mt-1 block w-full mt-5"
            />
          </div>
        </div>
      </div>
    </section>
  )
```

Рисунок 16 – код страницы «Профиль» вкладка «Мои обследования»

4.2 Разработка базы данных

Для разработки модуля медицинской информационно-аналитической системы была взята существующая база данных и импортирована при помощи инструмента для работы с СУБД PhpMyAdmin. Для реализации модуля требовалась модификация существующей базы данных. Для этого были реализованы миграции, которые дополнили имевшуюся базу данных, а именно:

- В сущность пациенты (рисунок 17), где были добавлены поля «Логин» и «Пароль».
- В сущность Расписание кабинетов (рисунок 18), где был добавлен связующий ключ.

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('mskt_pacients', function (Blueprint $table) {
            $table->string('login')->nullable()->unique();
            $table->string('password');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        //
    }
};
```

Рисунок 17 – Миграция для сущности «Пациенты»

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('mskt_cabinet_schedule', function (Blueprint $table) {
            $table->foreign('cabinet_id')
                ->references('cabinet_id')->on('mskt_cabinets');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        //
    }
};

```

Рисунок 18 – Миграция для сущности «Расписание»

4.3 Разработка модуля медицинской информационно-аналитической системы

Так как был разработан интерфейс пользователя, следующим этапом стала разработка функциональной части модуля медицинской информационно-аналитической системы, где в пример так же была приведена страница профиля рисунок 19.


```

import * as React from 'react';
import PropTypes from 'prop-types';
import Tabs from '@mui/material/Tabs';
import Tab from '@mui/material/Tab';
import Typography from '@mui/material/Typography';
import Box from '@mui/material/Box';

function CustomTabPanel(props) {
  const { children, value, index, ...other } = props;

  return (
    <div
      role="tabpanel"
      hidden={value !== index}
      id={`simple-tabpanel-${index}`}
      aria-labelledby={`simple-tab-${index}`}
      {...other}
    >
      {value === index && (
        <Box sx={{ p: 3 }}>
          <Typography component="span" variant="body2">{children}</Typography>
        </Box>
      )}
    </div>
  );
}

CustomTabPanel.propTypes = {
  children: PropTypes.node,
  index: PropTypes.number.isRequired,
  value: PropTypes.number.isRequired,
};

function allyProps(index) {
  return {
    id: `simple-tab-${index}`,
    'aria-controls': `simple-tabpanel-${index}`,
  };
}

export default function Edit({ auth, mustVerifyEmail, status, workTime }) {
  const [value, setValue] = React.useState(0);
  const handleChange = (event, newValue) => {
    setValue(newValue);
  };

```

Рисунок 19 – Серверная часть страницы «Профиль»

Как видно из подключенных библиотек здесь присутствует плагин «MUI» (material ui) и уже при помощи него реализована функция перехода на вкладки.

Функционал вкладки «Личные данные» требует от себя только вывод информации о пользователе и изменение некоторых внутренних данных (рисунок 20).

```

import InputError from "@/Components/InputError";
import InputLabel from "@/Components/InputLabel";
import PrimaryButton from "@/Components/PrimaryButton";
import TextInput from "@/Components/TextInput";
import { useForm, usePage } from "@inertiajs/react";
import * as React from 'react';

export default function UpdateProfileName() {
  const user = usePage().props.auth.user;

  const { data, setData, errors, processing, patch } = useForm({
    first_name: user.first_name,
    last_name: user.last_name,
    fathers_name: user.fathers_name,
    sex: user.sex,
    birthday: user.birthday,
    phone: user.mobile_phone,
    email: user.email,
  });

  const submit = (e) => {
    e.preventDefault();

    patch(route('profile.update'), data); // Добавьте скобки для вызова функции patch
  };

```

Рисунок 20 – Серверная часть страницы «Профиль» вкладки «Личные данные»

Функционал же вкладки «Мои исследования» уже более обширен (рисунок 21). В главной функции указан массив, содержащий наименование для каждого статуса, в котором сейчас находится исследование.

При помощи элементов map, действие которого перечисляет все данные в массиве, и элемента filter, который находит записи подходящие по условию. В следующей функции уже реализован вывод данных о записях пациента.

```

export default function MyExam({ workTime, guzle }) {
  const user = usePage().props.auth.user;

  const { data, setData, processing, put, get } = useForm({
    descript: null,
    id: null,
  });

```

Рисунок 21 – Серверная часть страницы «Профиль» вкладки «Мои исследования»

```

const [isModalOpen, setIsModalOpen] = useState(false);

const openModal = (id) => {
  setData({ ...data, id: id })
  setIsModalOpen(true);
};
const closeModal = () => {
  setIsModalOpen(false);
};

const updateStatus = (e) => {
  e.preventDefault();
  put(route('status.update', data.id), {
    body: {
      id: data.id,
      descript: data.descript
    },
    onSuccess: () => {
      closeModal();
    },
    onError: (e) => { console.log(e) }
  });
}

const downloadPdf = (e, id) => {
  e.preventDefault();
  window.open(route('exLoad', data.id))
}

const status = {
  "-1": 'Отменено',
  5: 'Описание выдано',
  10: 'Описание готово',
  11: 'Описание готово - ожидание распечатки',
  12: 'Проверено',
  14: 'Требуется доработка',
  15: 'Описание на проверке',
  20: 'Описание отложено',
  24: 'Удаленная проверка',
  25: 'Исследование завершено',
  30: 'Исследование начато',
  35: 'Приход',
  40: 'Дозвонились',
  45: 'Не дозвонились',
  50: 'Записан на исследование',
  55: 'Предварительно записан',
};

const cancellist = {
  1: 'Отказ',
  2: 'Длительное опоздание',
  5: 'Не пришел',
  3: 'Технические условия',
  4: 'Другое',
  6: 'Отменена пациентом'
}

```

Рисунок 22 – Продолжение серверной части страницы «Профиль» вкладки «Мои исследования»

Так же имеется страница с выводом расписания врачей. Логика данной страницы схожа со страницей «Запись на прием», за исключением отсутствия функции отправки какого-либо метода запроса на сервер. Так же были добавлены некоторые функции от библиотеки `day.js` с целью осуществить фильтрацию по

доступным для работы врачей дням (рисунок 22). На странице также имеется локализация для календаря обозначение начала недели с понедельника при помощи библиотеки day.js, объявление переменных для хранения изменяемых во время работы данных, запрос данных на api страницы, функция которая не позволяет нажать пользователю на дату которая недоступна для записи.

```
export default function ScheduleEmployee({ workTime, methods, methodology, sheduleOkTime }) {
  dayjs.extend(updateLocale);
  dayjs.updateLocale('en', {
    // Sunday = 0, Monday = 1.
    weekStart: 1,
  });

  const { data, setData } = useForm({
    methoData: null,
    methodologyData: null,
    employeeData: null,
    dayData: null,
    date: dayjs(),
  });

  const today = dayjs();

  const isDateDisabled = (date) => {
    const dayjsDate = dayjs(date);
    // Проверяем, присутствует ли дата в списке доступных дат
    return dayjsDate.day() === 0;
  };

  const formatingData = data.date.format('YYYY-MM-DD');

  const methodList = methods.map((elem) => ({ label: elem.name, id: elem.method_id }));

  const methodologyList = methodology.filter((elem) => elem.method_id === data.methoData).map((elem) => ({ label: elem.name, id: elem.methodology_id }));

  const EmployerList = (methodology.find((elem) => elem.methodology_id === data.methodologyData) || { employees: [] }).employees
    .map((elem) => ({ label: ` ${elem.last_name} ${elem.first_name} ${elem.fathers_name}`, id: elem.employee_id }));

  const getDates = () => {
    axios.get('http://127.0.0.1:8001/api/employeeSchedule', {
      employee: data.employee_id,
      timeStart: data.time_start,
      timeEnd: data.time_end,
    }).then((response) => {
      setWorkTimes(response.data.data)
    }).catch((err) => { console.log(err) });
  };

  const [WorkTimes, setWorkTimes] = useState([])

  const parseWorkTime = () => {
    const elem = WorkTimes.find((elem) => elem.employee_id === data.employeeData);
    const timeSeans = methodology.find((elem) => elem.methodology_id === data.methodologyData)
    let rows = [];
    if (elem) {
      const timeStart = dayjs(formatingData + elem.time_start)
    }
  }
}
```

Рисунок 22 – Серверная часть страницы «Профиль» вкладки «Расписание врачей»

Более подробно рассмотреть код вы можете в приложении Б, в котором представлен код страницы профиля и в приложении В, в котором указан код контроллера профиля.

5 Тестирование программного продукта

Самым главным аспектом хорошей медицинской информационно-аналитической системы является его корректная работа. Функциональное тестирование проверяет правильность работы функций и возможностей модуля медицинской информационно-аналитической системы.

Laravel для проверки функциональности и возможностей предоставляет функцию тестов. Так как работа с данным модулем медицинской информационно-аналитической системы начинается с авторизации то и тесты стоит начать с неё (рисунок 23). Код тестирования авторизации содержит в себе:

- 1) Имя файла;
- 2) Необходимые зависимости для корректной работы теста;
- 3) Объявление класса для понимания TestCase элементов тестирования;
- 4) Элемент тестирования;
- 5) Внутренний функционал теста;

```
namespace Tests\Feature;

use Illuminate\Foundation\Testing\RefreshDatabase;
use Illuminate\Foundation\Testing\WithFaker;
use Tests\TestCase;

use App\Models\Patient;

0 references | 0 implementations
class AuthorizationTest extends TestCase
{
    /**
     * A basic feature test example.
     */
    0 references | 0 overrides
    public function testLogin(): void
    {
        $response = $this->get('/login');

        $response = $this->post('/login', [
            'login' => 'login',
            'password' => 'password',
        ]);
        $this->assertAuthenticated();
    }
}
```

Рисунок 23 – Код теста функции «Авторизация»

Следующий тест был реализован с целью проверить корректность записи на прием и его содержание показано на рисунке 24.

```

1  <?php
2
3  namespace Tests\Feature;
4
5  use Illuminate\Foundation\Testing\RefreshDatabase;
6  use Illuminate\Foundation\Testing\WithFaker;
7  use Tests\TestCase;
8
9  use App\Models\Patient;
10
11  0 references | 0 implementations
12  class CreateExamTest extends TestCase
13  {
14      /**
15       * A basic feature test example.
16       */
17
18      0 references | 0 overrides
19      public function testCreate(): void
20      {
21          $response = $this->get('/login');
22
23          $response = $this->post('/login', [
24              'login' => 'login',
25              'password' => 'password',
26          ]);
27          $this->assertAuthenticated();
28
29          $response = $this->get('/createExam');
30
31          $response = $this->post('/createExam', [
32              'method_id' => '1',
33              'methodology_id' => '307',
34              'employee_id' => '2',
35              'patient_id' => '1',
36              'date' => '2024-05-03',
37              'time_start' => '08:00:00',
38
39              'to_employee_check' => '0',
40              'vip_research' => '0',
41              'cabinet_id' => '0',
42              'methodology_group' => '0',
43              'methodology_group_bonus' => '0',
44              'methodology_side' => '0',
45              'diagnosis_id' => '0',
46              'is_first' => '0',
47              'is_first_place' => '0',
48              'is_research' => '0',
49          ]);
50      }
51  }

```

Рисунок 24 – Код теста функции «Запись на прием»

Код тестирования записи на прием также, как и код тестирования авторизации содержит в себе:

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						38
Изм.	Лист	№ докум.	Подпись	Дата		

- 1) Имя файла;
- 2) Необходимые зависимости для корректной работы теста;
- 3) Объявление класса для понимания TestCase элементов тестирования;
- 4) Элемент тестирования;
- 5) Внутренний функционал теста;

После написания внутреннего содержимого тестов вводится команда `php artisan test`, и после недолгой обработки по завершению работы он выдаст соответствующий результат (рисунок 25).

```
PS C:\Users\ProgramistVelikiy\Desktop\cabinet-pac
ienta\project> php artisan test

PASS Tests\Unit\AuthorizationTest
✓ that true is true                                0.59s

PASS Tests\Feature\AuthorizationTest
✓ login                                             1.26s

PASS Tests\Feature\CreateExamTest
✓ create                                           0.75s

Tests: 3 passed (3 assertions)
Duration: 7.00s
```

Рисунок 25 – Результат тестирования

В данном случае все тесты прошли успешно и функции модуля медицинской информационно-аналитической системы все работает корректно и без каких либо ошибок.

6 Документирование программного продукта

Модуль медицинской информационно-аналитической системы «Личный кабинет пациента» имеет простой и интуитивно понятный интерфейс, что позволяет легко понять любому пользователю, как с ней работать.

При открытии проекта в Visual Studio Code и последующим запуском сервера с помощью инструкций в файле помощника ReadMe.txt перед пользователем открывается страница авторизации (рисунок 26). Чтобы приступить к взаимодействию с программой и получить доступ к функциям необходимо пройти авторизацию.

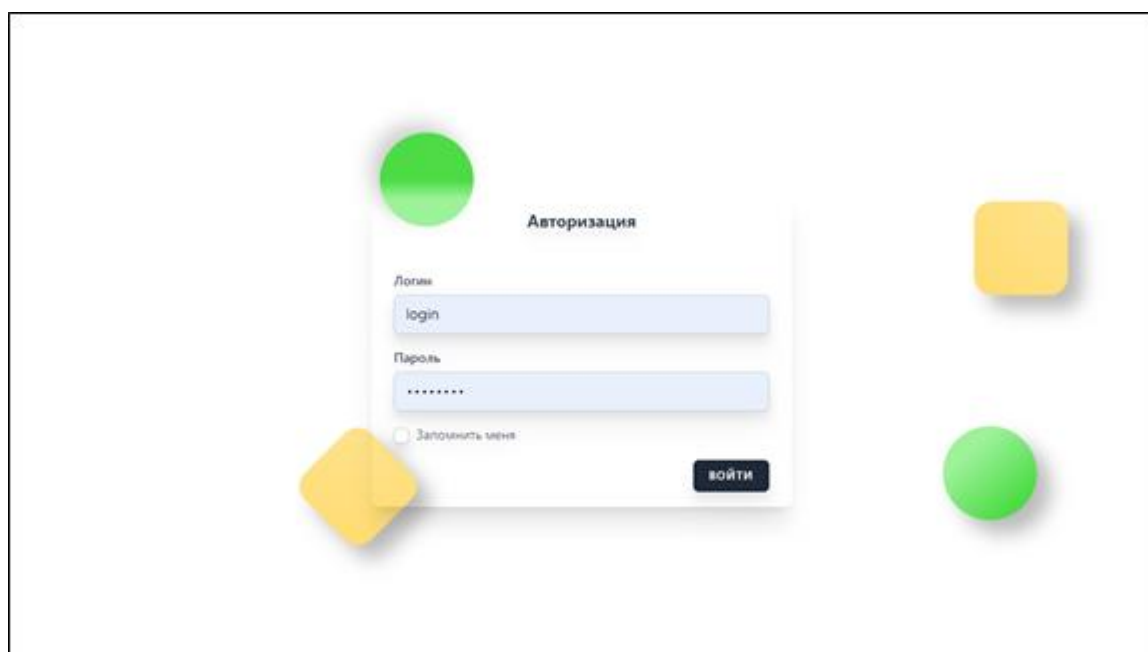


Рисунок 26 – Страница «Авторизация»

Если пользователь написал данные неверно то ему высветиться ошибка и дальнейший доступ будет невозможен, но после ввода верных данных пользователь попадает на страницу с его исследованиями, где указаны его прошедшие, нынешние и будущие исследования (рисунок 27).

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

ВАШИ ОБСЛЕДОВАНИЯ

ЛИЧНЫЕ ДАННЫЕ

Информация о ваших обследованиях

ЗАПИСАТЬСЯ НА ПРИЕМ

Дата	Время	Метод	Зона	Врач	Контраст	Статус	Скачать
2024-05-04	с пн	МРТ	Артерий головного мозга с ДКУ	jPeeq000 Александр xbPFUy0W0x	-	Предварительно записан	ОТМЕНИТЬ
2024-05-02	с пн	МСКТ	Экскреторная МСКТ урографии	jPeeq000 Александр xbPFUy0W0x	-	Записан на исследование	ОТМЕНИТЬ
2024-05-02	с пн	МСКТ	ЭКГ синхронизированное МСКТ исследование сердца и ангиография грудной и брюшной аорты (подготовка к чрезосудистому протезированию аортального клапана)	enY2xVj5w Павел FKCINkQZv	-	Записан на исследование	ОТМЕНИТЬ
2024-05-03	с пн	УЗИ	Эхокардиография - СВМ	enY2xVj5w Павел FKCINkQZv	-	Записан на исследование	ОТМЕНИТЬ
	с пн	МСКТ	ЭКГ синхронизированное МСКТ исследование сердца для определения кальцификации коронарных артерий	AMAzdmArnta Виктор vVnKbScENKJ	-	Записан на исследование	ОТМЕНИТЬ
	с пн	УЗИ	Эхокардиография - Фуск С.А.	q5uxC0fYau Stancislav TWQsREEZi	-	Записан на исследование	ОТМЕНИТЬ

Рисунок 27 – Страница «Профиль» вкладка «Мои обследования»

Если у пользователя есть цель записаться на прием, то после нажатия на кнопку записаться на прием его переадресует на страницу записи на исследования, где пользователь выбирает: метод, зону, врача, дату и время для обследования (рисунок 28).

Запись на прием

Метод обследования

Направление

Выберите метод обследования

Выберите направление

Врач

Выберите врача

Дата

Время

May 2024

<

>

M

T

W

T

F

S

S

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

ЗАПИСАТЬСЯ

Рисунок 28 – Страница «Запись на прием»

Если никаких проблем нет, то происходит отправка данных с последующей переадресацией на страницу «Мои обследования».

Если же пользователь захочет добавить электронную почту или просто просмотреть свои личные данные, то по нажатии на вкладку «Личные данные» его переадресует на одноименную страницу, где он может уже просмотреть доступную для него информацию (рисунок 29).

Профиль ivjKgdtBbsonOVGhANzb eIMELWXOQDeOVR

ВАШИ ОБСЛЕДОВАНИЯ ЛИЧНЫЕ ДАННЫЕ

Информация о профиле

Имя: eIMELWXOQDeOVRnJaKn Фамилия: ivjKgdtBbsonOVGhANzb

Отчество: pghaczOvuGrzaMekyitv

Дата рождения: 0000-00-00 Пол: Мужской

Номер телефона: 662 Электронная почта: eBpqQURkiDRaKiNVlMvpI

СОХРАНИТЬ

Рисунок 29 – страница «Профиль» вкладка «Личные данные»

Помимо основных функций пользователь также может перейти на страницу просмотра работы врачей которые ему необходимы. Для этого он аналогично странице с созданиями заявки на обследование, выбирает нужные пункты для более оперативного поиска врача. И по нажатии на дату пользователю высвечивается время работы данного врача (рисунок 30).

ВАШИ ОБСЛЕДОВАНИЯ

ЛИЧНЫЕ ДАННЫЕ

РАСПИСАНИЕ ВРАЧЕЙ

Расписание врачей

Метод обследования

Выберите метод обследования

Рентгеновская компьютерн...

Направление

Выберите направление

Ангиография грудной и бр...

Врач

Выберите врача

Березин Александр Викто...

Дата

May 2024

<

>

М

Т

В

Т

П

С

С

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

Время

Время работы: 08:00 до 16:00

Рисунок 30 – Страница «Расписание врачей»

ЗАКЛЮЧЕНИЕ

В заключение, модуль медицинской информационно-аналитической системы «Личный кабинет пациента» играет важную роль в помощи в работе с клиентами, предоставляя удобные и эффективные инструменты для взаимодействия пациентов с медицинским учреждением. Она обеспечивает пациентам доступ к своим медицинским данным, позволяет записываться на прием, просматривать результаты обследований, скачивать заключения, изменять свои личные данные и просматривать расписание врачей.

Дизайн модуля медицинской информационно-аналитической системы «Личный кабинет пациента» способствует повышению уровня удовлетворенности пациентов, они могут легко управлять своими личными данными и наглядно быть в курсе своего здоровья. Это также помогает оптимизировать работу медицинского персонала, сокращая время и затраты на поиски необходимой информации.

Использование данного модуля медицинской информационно-аналитической системы «Личный кабинет пациента» способствует повышению качества медицинского обслуживания, улучшению доступности и удобства для пациентов, а также повышению эффективности работы медицинского учреждения в целом. Эта система является важным шагом в развитии цифровизации здравоохранения и создает благоприятную основу для дальнейшего совершенствования медицинской практики и улучшения здоровья населения.

В ходе выполнения были выполнены задачи и функциональные требования по проектированию макета страниц и сопутствующих им кнопок и иконок, а также соответствующий тематике выбор цветовой гаммы и типографики. Так же была выполнена главная цель проектирования, которая заключалась в создании лаконичного и комфортным для использования интерфейса и соответствующий этому интерфейсу функционалу.

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						44
Изм.	Лист	№ докум.	Подпись	Дата		

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 creately– Простое руководство к диаграммам развертывания UML. – URL: <https://creately.com/blog/ru/uncategorized-ru/учебное-пособие-по-диаграмме-развёрт/> (дата обращения: 12.03.2024). – Текст: электронный.

2 depix – Система управления базами данных MySQL – URL: https://depix.ru/articles/sistema_upravleniya_bazami_dannyh_mysql?ysclid=lai22eq1ab602973556 (дата обращения: 25.02.2024). – Текст: электронный.

3 dev – Laravel Export/Import: Step-by-Step Guide with Repo Example – URL: <https://dev.to/fajarwz/laravel-exportimport-step-by-step-guide-with-repo-example-18p0> (Дата обращения: 29.02.2024). – Текст электронный.

4 dev-notes – Laravel: Все секреты миграции – URL: <https://www.dev-notes.ru/articles/laravel-migrations/?ysclid=loxzmih0ur909784887> (Дата обращения: 01.02.2024). – Текст электронный.

5 flowbite – Tailwind CSS Forms – Flowbite – URL: <https://flowbite.com/docs/components/forms/> (Дата обращения: 03.03.2024). – Текст электронный.

6 kinsta – Authentication in Laravel Using Breeze – URL: <https://kinsta.com/blog/laravel-breeze/> (Дата обращения: 07.03.2024). – Текст электронный.

7 laravel– Testing: Getting Started – URL: <https://laravel.com/docs/10.x/testing> (Дата обращения: 10.03.2024). – Текст электронный.

8 laravel.com – PHP-фреймворк для веб-мастеров – URL: <https://laravel.com/>. (дата обращения: 01.03.2024). – Текст электронный.

9 MDN Web Docs – репозиторий документации для веб-разработчиков – URL: <https://developer.mozilla.org/en-US/> (дата обращения: 26.04.2024). – Текст: электронный.

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						45
Изм.	Лист	№ докум.	Подпись	Дата		

10 mui – объектно-ориентированная система, для создания и работы в графическом интерфейсе пользователя – URL: <https://mui.com/material-ui/getting-started/> (дата обращения: 11.03.2024). – Текст: электронный.

11 react – это библиотека JavaScript – URL: <https://react.dev/learn> (дата обращения: 22.02.2024). – Текст: электронный.

12 Stack Overflow – система вопросов и ответов о программировании – URL: <https://stackoverflow.com> (дата обращения: 20.03.2024). – Текст: электронный.

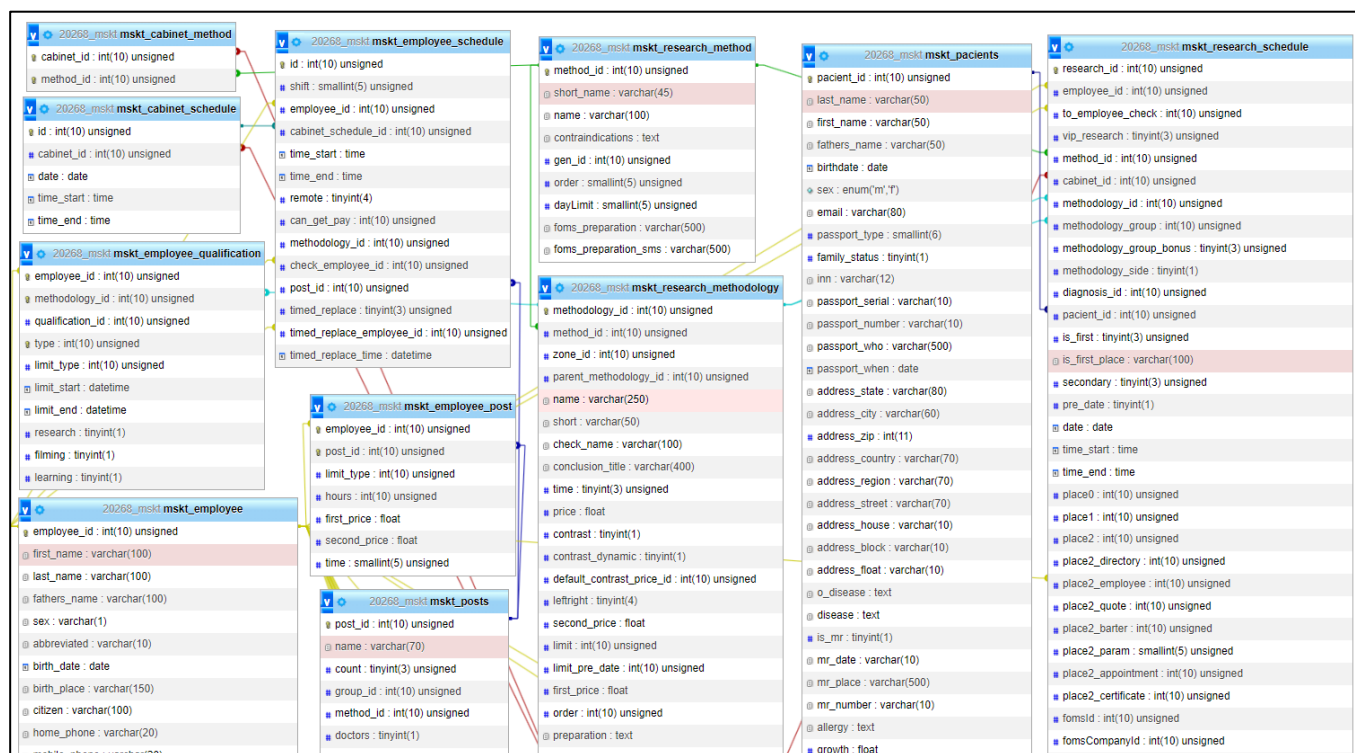
13 TailwindCSS – CSS-фреймворк – URL: <https://tailwindcss.com/docs/installation> (дата обращения: 04.03.2024). – Текст: электронный.

14 Колисниченко Д.Н. Современный сайт на PHP и JavaScript. - СПб.: Питер, 2009.

15 Уэнц, К. PHP и MySQL. Карманный справочник / К. Уэнц. М.: Вильямс, 2013. – 256 с.

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						46
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А – ER-модель



Приложение Б – Листинг кода страницы профиля

```
import AuthenticatedLayout from '@/Layouts/AuthenticatedLayout';

import MyExam from './Partials/MyExam';

import UpdateProfileName from './Partials/UpdateProfile';

import { Head } from '@inertiajs/react';


import * as React from 'react';

import PropTypes from 'prop-types';

import Tabs from '@mui/material/Tabs';

import Tab from '@mui/material/Tab';

import Typography from '@mui/material/Typography';

import Box from '@mui/material/Box';


function CustomTabPanel(props) {

  const { children, value, index, ...other } = props;


  return (

    <div

      role="tabpanel"

      hidden={value !== index}

      id={`simple-tabpanel-${index}`}

      aria-labelledby={`simple-tab-${index}`}

      {...other}

    >
```

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		


```

{ value === index && (
  <Box sx={{ p: 3 }}>
    <Typography component={'span'} variant={'body2'}>{children}</Typography>
  </Box>
)}
</div>
);
}

```

```

CustomTabPanel.propTypes = {
  children: PropTypes.node,
  index: PropTypes.number.isRequired,
  value: PropTypes.number.isRequired,

};

```

```

function allProps(index) {
  return {
    id: `simple-tab-${index}`,
    'aria-controls': `simple-tabpanel-${index}`,
  };
}

```

```

export default function Edit({ auth, mustVerifyEmail, status, workTime }) {

```

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						49
Изм.	Лист	№ докум.	Подпись	Дата		

```

const [value, setValue] = React.useState(0);

const handleChange = (event, newValue) => {

  setValue(newValue);

};

return (

  <AuthenticatedLayout

    user={auth.user}

  >

    <Head title="Профиль" />

    <div className="py-12">

      <div className="max-w-9xl mx-auto sm:px-6 lg:px-8 space-y-6">

        <Box sx={{ width: '100%' }}>

          <Box sx={{ }}>

            <Tabs value={value} onChange={handleChange} aria-label="basic tabs example"
              className="ml-[800px]">

              <Tab label="Личные данные" {...allYProps(0)} />

              <Tab label="Ваши обследования" {...allYProps(1)} />

            </Tabs>

          </Box>

          </Box>

          <CustomTabPanel value={value} index={0}>

```

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		

```
<div className="p-4 sm:p-8 bg-white shadow sm:rounded-lg">
```

```
<UpdateProfileName
```

```
className="max-w-xl"
```

```
/>
```

```
</div>
```

```
</CustomTabPanel>
```

```
<CustomTabPanel value={value} index={1}>
```

```
<div className="p-4 sm:p-8 bg-white shadow sm:rounded-lg">
```

```
<MyExam className="max-w-xl"
```

```
workTime={workTime} />
```

```
</div>
```

```
</CustomTabPanel>
```

```
</div>
```

```
</div>
```

```
</AuthenticatedLayout>
```

```
);
```

```
}
```

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						51
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение В – Листинг кода контроллера профиля

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\ProfileUpdateRequest;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;
use App\Models\EmployerPost;
use App\Models\Method;
use App\Models\Methodology;
use App\Models\ResearchSchedule;
use Inertia\Inertia;
use Inertia\Response;

class ProfileController extends Controller
{
    /**
     * Display the user's profile form.
     */
    public function edit(Request $request): Response
    {
        return Inertia::render('Profile/Edit', [

            'status' => session('status'),
            'methods' => Method::getActiveMethods()->get(),
            'methodology' => Methodology::getActiveMethodologies()->with('employees')->get(),
            'workTime' => ResearchSchedule::getActiveSchedule()-
            >with('employees','methods','methodologys','contrast','posts')->get(),

            'seansTime' => EmployerPost::getActiveTime()->get(),

        ]);
    }

    /**
     * Update the user's profile information.
     */
    public function update(ProfileUpdateRequest $request): RedirectResponse
    {
```

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

```
$request->user()->fill($request->validated());
```

```
$request->user()->save();
```

```
return Redirect::route('profile.edit');
```

```
}
```

```
}
```

					ДП.09.02.07-3.24.201.05 ПЗ	Лист
						53
Изм.	Лист	№ докум.	Подпись	Дата		