

## Введение

Современные пользователи всё чаще нуждаются в удобных и простых инструментах для поиска информации о медицинских учреждениях. В условиях высокой загруженности и необходимости оперативного получения достоверных сведений особую ценность приобретает разработка веб-сайта, который агрегирует данные о больницах, поликлиниках, врачах и предоставляемых услугах, делая их доступными в одном месте.

Цель разработки заключается в создании информационно-поисковой системы «MedGid Grodno», которая позволит жителям города быстро находить медицинские учреждения по различным параметрам: названию, профилю врача, оказываемым услугам, стоимости, времени работы и даже по симптомам или жалобам. В отличие от традиционного поиска через разрозненные источники, данный ресурс обеспечивает прозрачность, удобство и скорость получения информации, а также предоставляет дополнительные возможности — например, оставление отзывов и ознакомление с медицинскими новостями.

Далее приведем краткое описание разделов пояснительной записки. Первый раздел носит название «Анализ задачи». В нём рассмотрена постановка задачи, исследована предметная область, определены входные и выходные данные, а также проведён обзор существующих аналогов.

В разделе «Проектирование задачи» представлены основные аспекты разработки программного продукта: выбор модели жизненного цикла, проектирование структуры сайта, моделирование бизнес-процессов и данных, а также описание пользовательского интерфейса.

«Реализация» – это третий раздел пояснительной записки, в котором описываются организация данных, функции пользователя и администратора, а также логика работы сайта.

Четвёртый раздел – «Тестирование». В нём будет описана проверка корректности работы поиска, фильтров, отображения информации об учреждениях, отзывов и новостей, а также адаптивности сайта для различных устройств.

В пятом разделе «Применение» раскрывается назначение и область использования системы. Среда функционирования — веб-браузеры и мобильные устройства.

«Заключение» содержит краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, а также степень автоматизации процессов на различных этапах разработки.

В «Списке использованных источников» приведены материалы, использованные при создании проекта.

					ТРПО 5-04 0612-02.43.06.25 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

В графической части представлены диаграммы, модели данных и схемы работы системы. В «Приложении» размещены скриншоты страниц сайта, демонстрирующие его структуру и функциональность.

					ТРПО 5-04 0612-02.43.06.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		5

# 1 Анализ задачи

## 1.1 Постановка задачи

Наименование задачи: информационно-поисковая система медицинских учреждений города Гродно (сайт) «MedGid Grodno».

Цель разработки: создание сайта для поиска информации о медицинских учреждениях города Гродно.

Назначение: информационно-поисковая система «MedGid Grodno» разрабатывается для людей любого возраста, желающим получить информацию о медицинских учреждениях города Гродно в различных целях.

Периодичность использования: по мере необходимости.

Источники и способы получения данных: сеть интернет.

Обзор существующих аналогичных ПП: 103.by

103.BY — сервис с максимально полным каталогом медицинских учреждений и врачей, сервисом поиска и бронирования лекарств в аптеках и полезным контентом о здоровье и красоте. Основная задача сервиса — сделать процесс выбора медицинского центра, врача, конкретной медицинской услуги или препарата максимально прозрачным, удобным и быстрым.

В разрабатываемом ПП будет похожий функционал, а также добавлена функция предложения наиболее подходящих медицинских учреждений по указанным жалобам на здоровье.

## 1.2 Функциональные требования.

Описание перечня функций и задач, которые должен выполнять будущий ПП:

Гость:

- просмотр страниц сайта и приложения;
- поиск медицинских учреждений по названию/по симптомам или жалобам/требованиям/профилю врача;
- местоположение медицинских учреждений;
- сортировка по стоимости услуг (от бесплатных к платным по возрастанию);
- время работы медицинских учреждений;
- просмотр медицинских учреждений от а до я.
- возможность оставлять отзывы без регистрации на сайте.

Администратор:

- добавление/удаление категории;
- добавление/удаление услуг/врачей;
- редактирование страниц сайта и приложения;
- доступ к базе данных.

					ТРПО 5-04 0612-02.43.06.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		6

### 1.3 Описание процессов с входной, выходной и условно-постоянной информацией.

Входная информация:

- требуемый профиль врача;
- требуемые услуги;
- требуемое местоположение;
- требуемое время работы;
- имеющиеся симптомы/жалобы.

Выходная информация:

- перечень отвечающих всем заданным требованиям медицинских учреждений/их отсутствие.

Условно-постоянная информация:

- контакты медицинских учреждений и врачей;
- адрес медицинских учреждений;
- оказываемые услуги;
- время работы врачей;
- фотографии медицинских учреждений и врачей;
- прочая информация о медицинских учреждениях и его врачах (достижения, годы работы, стаж).

### 1.4 Эксплуатационные требования.

1. Требования к применению:
  - помогает быстро найти и узнать о подходящих медицинских учреждениях/врачах в них;
2. Требования к реализации:
  - для реализации статических страниц и шаблонов должны использоваться языки HTML и CSS;
  - для реализации интерактивных элементов клиентской части должны использоваться языки JavaScript;
  - для реализации динамических страниц должен использоваться язык PHP.
3. Требования к надёжности:
  - система может быть недоступна не более чем 24 часа в год.
4. Требования к интерфейсу:
  - при разработке сайта должны быть использованы преимущественно белые, синие оттенки;
  - меню и ссылки должны быть логичными и доступными;

					ТРПО 5-04 0612-02.43.06.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		7

- интерфейс быстро загружается и реагирует на действия пользователя. использование удобных шрифтов и контраста;
- сохранение общего стиля на всех страницах;
- удобные инструменты для сортировки и уточнения запросов;
- сайт должен адаптироваться под компьютер, телефон и планшет.

## 1.5 Инструменты разработки

Figma AI Make можно рассматривать как полноценный набор инструментов разработки, который объединяет этапы проектирования и программирования в единую среду. В основе его работы лежит анализ макета, созданного в Figma: система автоматически интерпретирует визуальные элементы — блоки, текстовые поля, изображения, кнопки — и преобразует их в структуру кода. Таким образом дизайнерский прототип сразу получает техническую реализацию, где каждый компонент оформлен в виде React-компонента со связанными стилями и логикой. Пользователь получает доступ к исходному коду проекта, который можно скачать и открыть в любой IDE, а затем вручную изменять, дополнять или оптимизировать. Это превращает Figma AI Make в инструмент, который не только ускоряет старт разработки, но и оставляет разработчику полную свободу для дальнейшей кастомизации.

После загрузки проекта на GitHub Figma AI Make продолжает выполнять роль инструмента разработки: он обеспечивает базовую структуру, корректные зависимости и готовые скрипты для сборки. Чтобы сайт стал рабочим, необходимо установить зависимости, выполнить сборку (`npm run build` для Vite), а затем опубликовать содержимое папки `dist` через GitHub Pages. В результате дизайнерский макет, преобразованный в код с помощью Figma AI Make, становится полноценным веб-сайтом, который можно поддерживать и развивать как обычный проект, используя привычные инструменты фронтенд-разработки.

					ТРПО 5-04 0612-02.43.06.25 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

## 2 Проектирование

### 2.1 Выбор стратегии разработки и модели жизненного цикла

Для разработки веб-ресурса «Информационно-поисковая система медицинских учреждений г. Гродно». следует выбрать стратегию разработки и модель жизненного цикла. Осуществляем выбор посредством составления таблиц 1-4.

Таблица 1 – Выбор модели жизненного цикла на основе характеристик требований

Критерии категории требований	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
Являются ли требования к проекту легко определяемыми и реализуемыми?	<u>Да</u>	<u>Да</u>	<u>Да</u>	Нет	Нет	Нет
Могут ли требования быть сформулированы в начале ЖЦ?	<u>Да</u>	<u>Да</u>	<u>Да</u>	<u>Да</u>	Нет	Нет
Часто ли будут изменяться требования на протяжении ЖЦ?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да
Нужно ли демонстрировать требования с целью их определения?	Нет	Нет	<u>Да</u>	Нет	<u>Да</u>	<u>Да</u>
Требуется ли проверка концепции программного средства или системы?	Нет	Нет	<u>Да</u>	Нет	<u>Да</u>	<u>Да</u>
Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
Нужно ли реализовать основные требования на ранних этапах разработки?	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>	<u>Да</u>

Вычисления: 2 за каскадную, 2 за V-образную, 5 за RAD, 3 за инкрементную, 4 за быстрого прототипирования и 4 за эволюционную.

Итог: На основе результатов заполнения таблицы 1 подходящей является RAD модель.

Таблица 2 – Выбор модели жизненного цикла на основе характеристик команды разработчиков

Критерии категории команды разработчиков проекта	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
Являются ли проблемы предметной области проекта новыми для большинства разработчиков?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да
Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков?	Да	Да	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да
Изменяются ли роли участников проекта на протяжении ЖЦ?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да	Да
Является ли структура процесса разработки более значимой для разработчиков, чем гибкость?	Да	Да	Нет	Да	Нет	Нет
Важна ли легкость распределения человеческих ресурсов проекта?	Да	Да	Да	Да	Нет	Нет
Приемлет ли команда разработчиков оценки, проверки, стадии разработки?	<u>Да</u>	<u>Да</u>	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>

Вычисления: 1 за каскадную, 1 за V-образную, 0 за RAD, 1 за инкрементную, 1 за быстрого прототипирования и 1 за эволюционную.

Итог: На основе результатов заполнения таблицы 2 подходящими являются все модели, кроме RAD.

Таблица 3 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей

Критерии категории коллектива пользователей	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
Будет ли присутствие пользователей ограничено в ЖЦ разработки?	Да	Да	<u>Нет</u>	Да	<u>Нет</u>	Да
Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
Будут ли пользователи вовлечены во все фазы ЖЦ разработки?	<u>Нет</u>	<u>Нет</u>	Да	<u>Нет</u>	Да	<u>Нет</u>
Будет ли заказчик отслеживать ход выполнения проекта?	Нет	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>

Вычисления: 0 за каскадную, 0 за V-образную, 0 за RAD, 1 за инкрементную, 2 за быстрого прототипирования и 2 за эволюционную.

Итог: На основе результатов заполнения таблицы 3 подходящей является модель быстрого проектирования и эволюционная.

Таблица 4 – Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

Критерии категории типов проекта и рисков	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
1	2	3	4	5	6	7
Разрабатывается ли в проекте продукт нового для организации направления?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да	Да
Будет ли проект являться расширением существующей системы?	Да	Да	Да	Да	<u>Нет</u>	<u>Нет</u>
Будет ли проект крупно- или среднемасштабным?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
Ожидается ли длительная эксплуатация продукта?	<u>Да</u>	<u>Да</u>	Нет	<u>Да</u>	Нет	<u>Да</u>



#### Продолжение таблицы 4

1	2	3	4	5	6	7
Необходим ли высокий уровень надежности продукта проекта?	Нет	<u>Да</u>	Нет	<u>Да</u>	Нет	<u>Да</u>
Предполагается ли эволюция продукта проекта в течение ЖЦ?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
Велика ли вероятность изменения системы (продукта) на этапе сопровождения?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
Является ли график сжатым?	<u>Нет</u>	<u>Нет</u>	Да	Да	Да	Да
Предполагается ли повторное использование компонентов?	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>	<u>Да</u>
Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)?	Нет	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>

Вычисления: 1 за каскадную, 2 за V-образную, 1 за RAD, 6 за инкрементную, 5 за быстрого прототипирования и 7 за эволюционную.

Итог: На основе результатов заполнения таблицы 4 подходящей является эволюционная модель.

Таблица 5 – Общий итог

Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
4	5	6	11	12	14

В итоге заполнения таблиц 1-4 наиболее подходящей является эволюционная модель.

## 2.2 Диаграмма вариантов использования

В ходе разработки веб-ресурса «Информационно-поисковая система медицинских учреждений г. Гродно» была составлена диаграмма вариантов использования, представленная на рисунке 1.

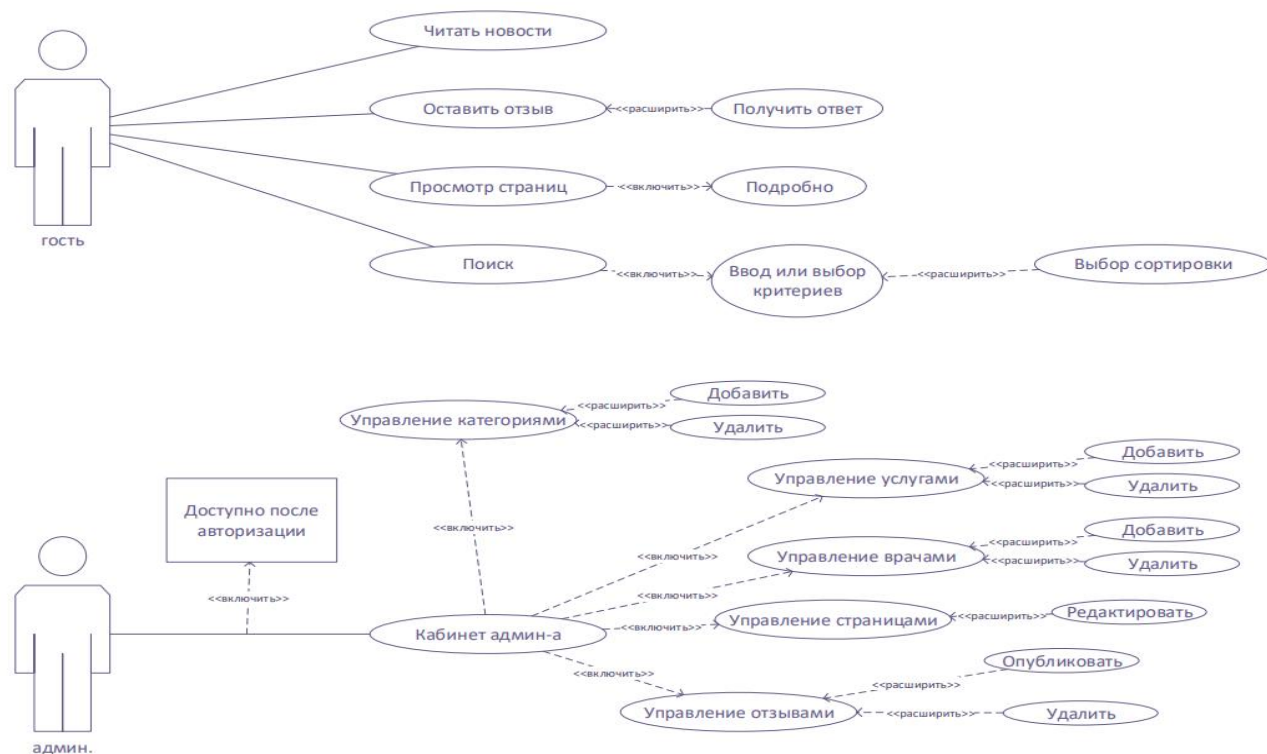
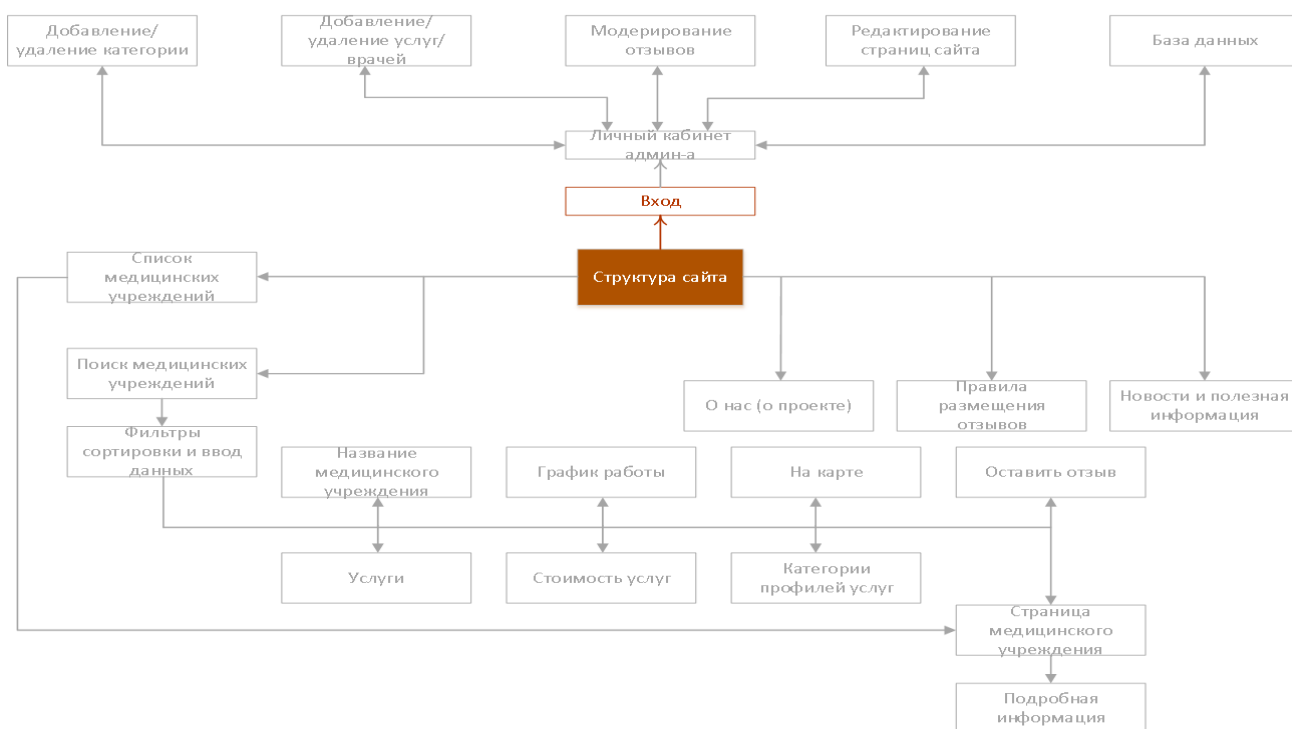


Рисунок 1 – Диаграмма вариантов использования

## 2.3 Проектирование структуры сайта

В ходе разработки веб-ресурса «Информационно-поисковая система медицинских учреждений г. Гродно» была составлена структура сайта, представленная на рисунке 2.



## Рисунок 2 – Структура сайта

### 2.4 Моделирование бизнес-процессов

В ходе разработки веб-ресурса «Информационно-поисковая система медицинских учреждений г. Гродно» была составлена модель бизнес-процессов, представленная на рисунках 3-5.

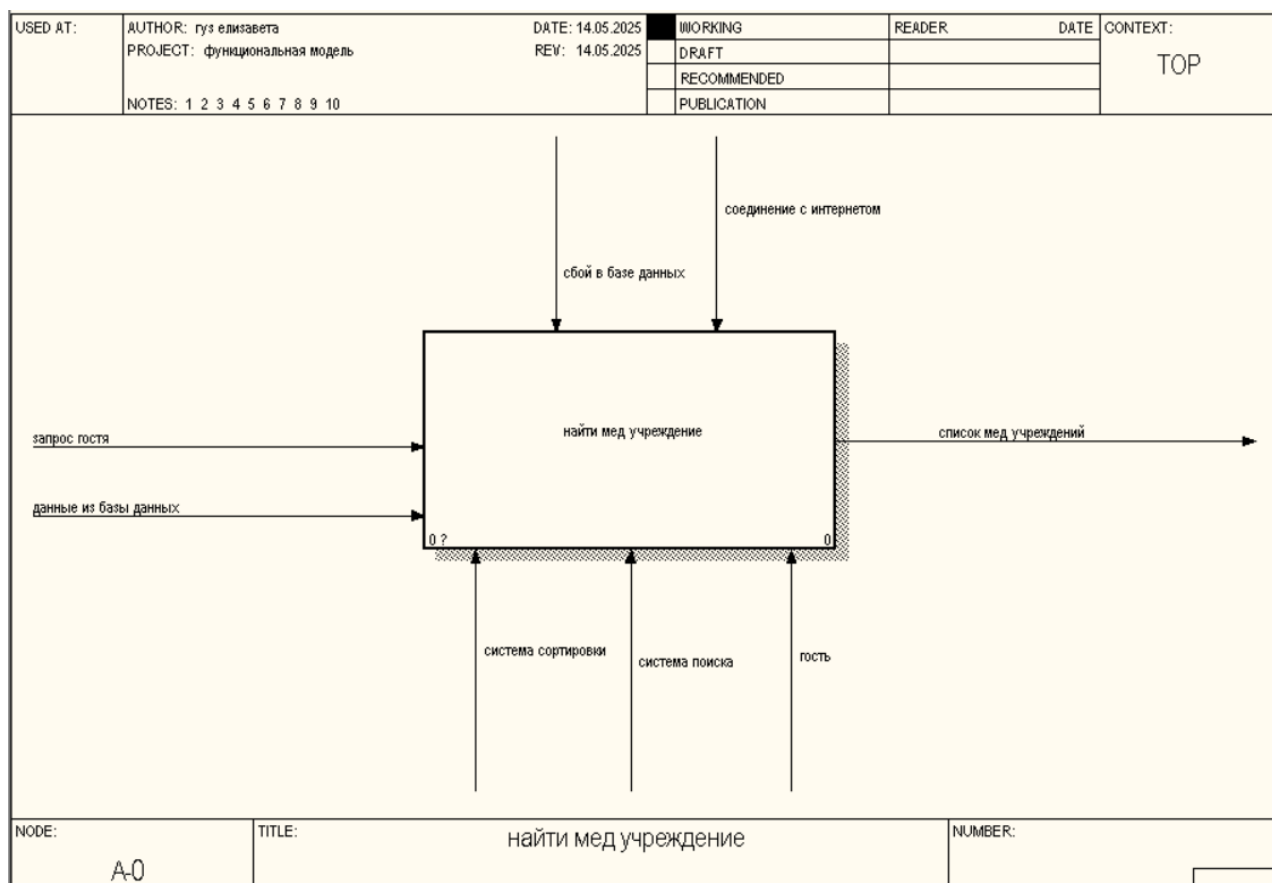


Рисунок 3 – Блок A0

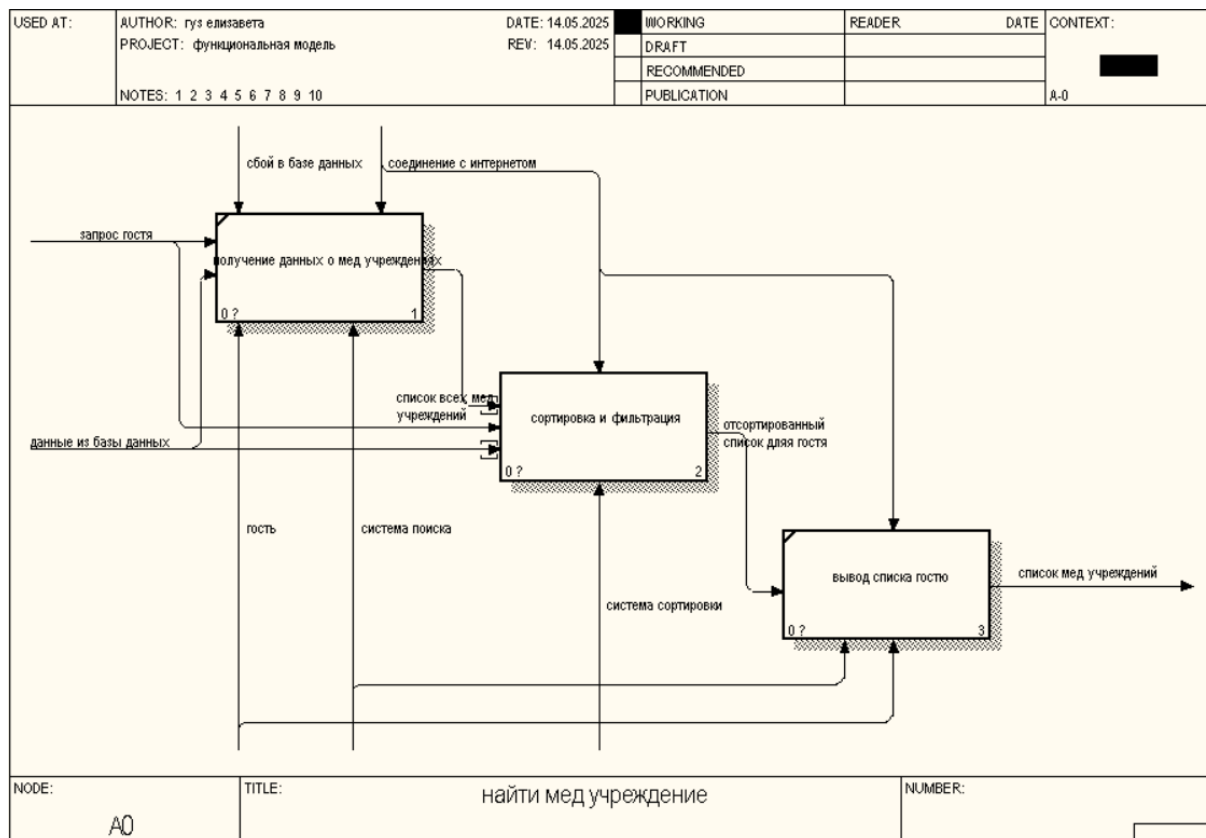


Рисунок 4 – Блоки А1, А2, А3

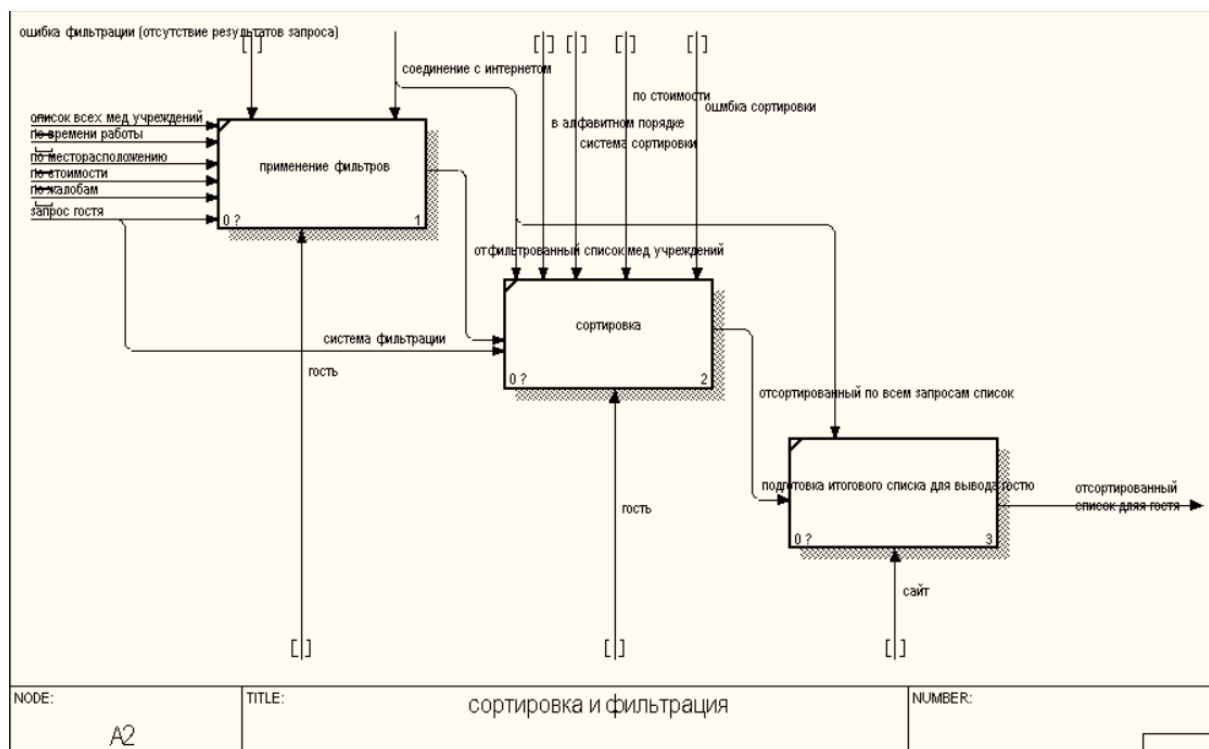


Рисунок 5 – Блоки А2.1, А2.2, А2.3

## 2.5 Моделирование данных

В ходе разработки веб-ресурса «Информационно-поисковая система медицинских учреждений г. Гродно» была составлена модель данных, представленная на рисунке 6.

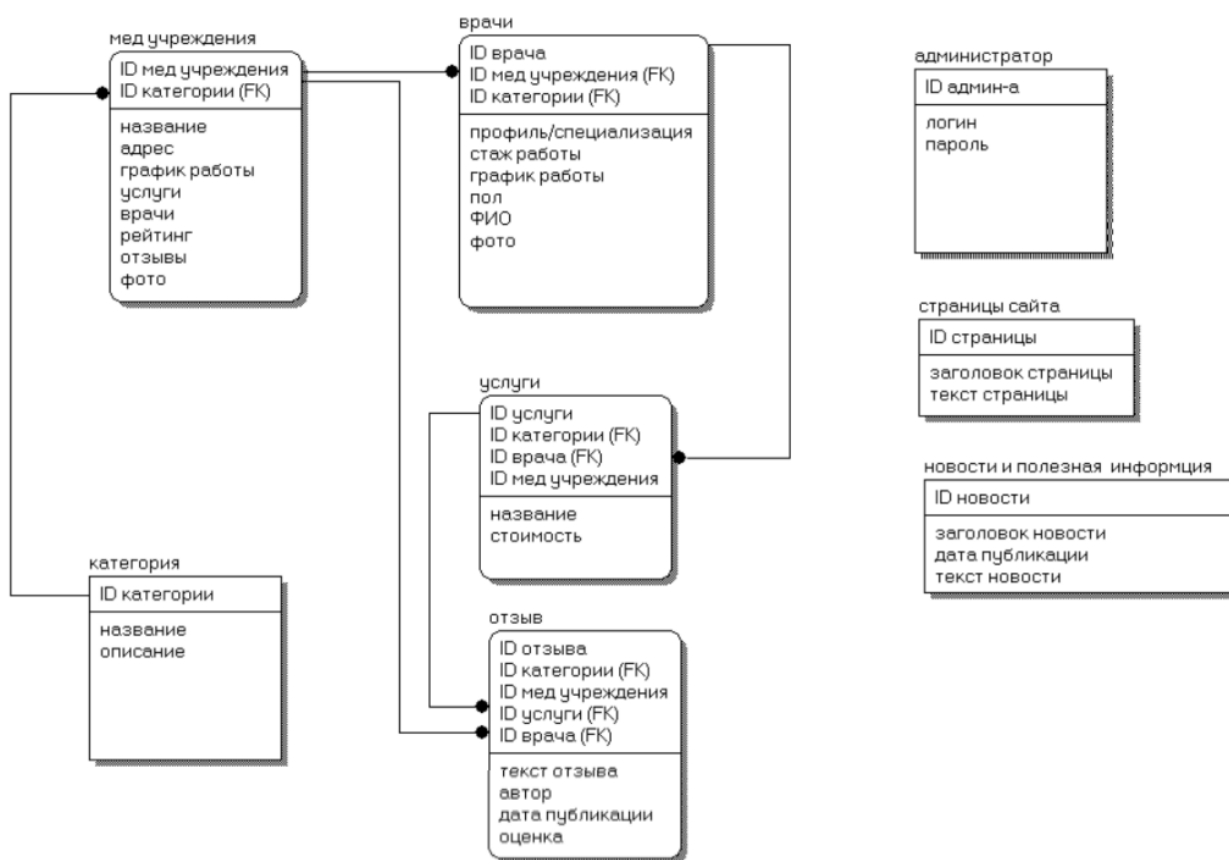


Рисунок 6 – Модель данных

## 2.6 Диаграмма последовательности

В ходе разработки веб-ресурса «Информационно-поисковая система медицинских учреждений г. Гродно» была составлена диаграмма последовательности, представленная на рисунке 7.

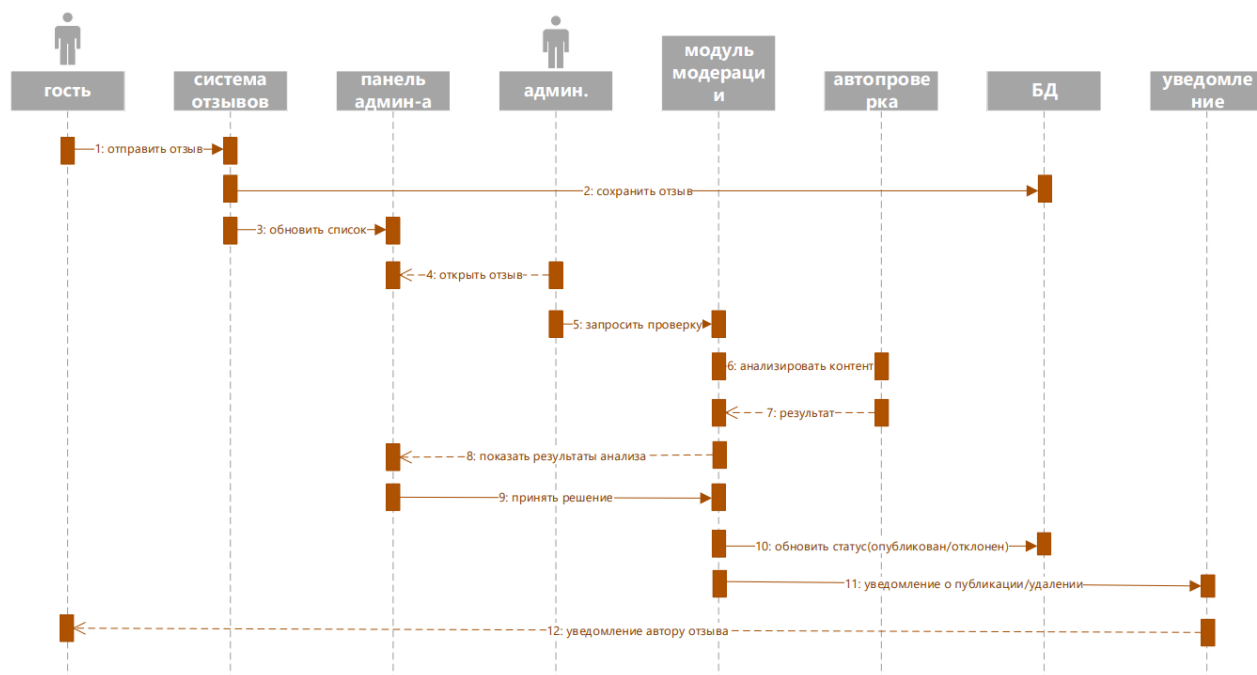


Рисунок 7 – Диаграмма последовательности

## 2.7 Диаграмма деятельности

Диаграмма деятельности отражена в приложении А.

## 2.8 Диаграмма классов

В ходе разработки веб-ресурса «Информационно-поисковая система медицинских учреждений г. Гродно» была составлена диаграмма классов, представленная на рисунке 8.

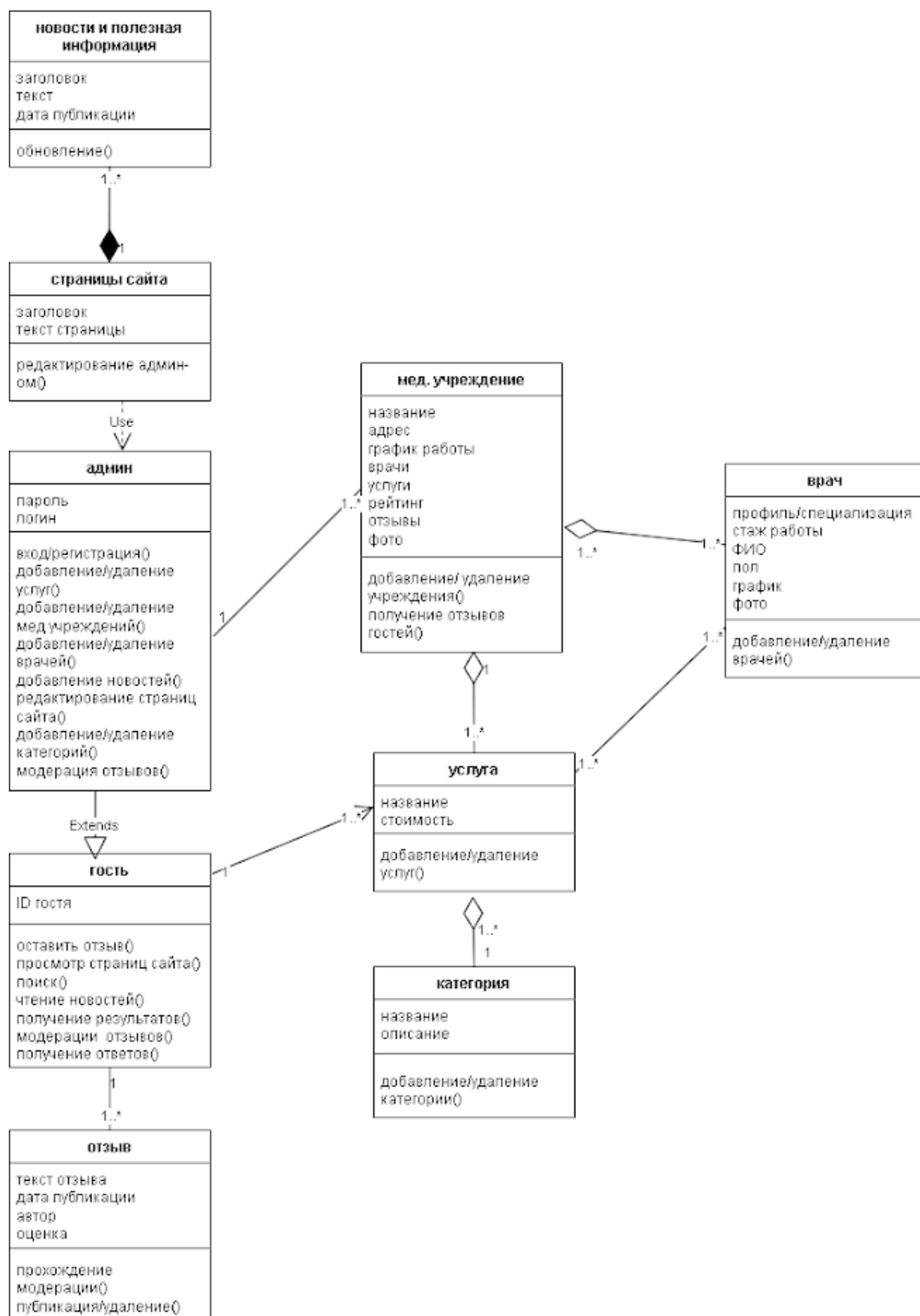


Рисунок 8 – Диаграмма классов

## 2.9 Диаграмма объектов

В ходе разработки веб-ресурса «Информационно-поисковая система медицинских учреждений г. Гродно» была составлена диаграмма объектов, представленная на рисунке 9.



Рисунок 9 – Диаграмма объектов

## 2.10 Разработка пользовательского интерфейса

В ходе разработки веб-ресурса «Информационно-поисковая система медицинских учреждений г. Гродно» были разработаны UX и UI-прототипы, представленные в Приложении Б.



## 3 Реализация

### 3.1 Руководство программиста

В Figma AI Make процесс создания сайта начинается с генерации интерфейса на основе макета, подготовленного в Figma. Сервис использует встроенные алгоритмы для анализа структуры дизайна — блоков, компонентов, текстов и изображений — и автоматически преобразует их в готовую верстку. В результате формируется проект на базе современных фронтенд-технологий (React, TypeScript, Vite), где каждая секция макета получает свой компонент и стили. После генерации пользователю предоставляется доступ к исходному коду: его можно скачать в виде архива или открыть в связанной среде разработки. Этот код полностью редактируемый, поэтому после загрузки я вручную вносила изменения — корректировала структуру компонентов, обновляла стили, добавляла собственные скрипты и оптимизировала сборку под нужды проекта. Таким образом, Figma AI Make выступает как инструмент автоматического старта, а дальнейшая доработка выполняется разработчиком напрямую в кодовой базе.

Чтобы сайт стал рабочим после загрузки на GitHub, необходимо выполнить несколько шагов. Сначала проект нужно разместить в репозитории, убедившись, что в корне присутствует корректный `package.json` и все зависимости установлены через `npm install` или `yarn`. Затем следует настроить сборку: в случае Vite это команда `npm run build`, которая формирует папку `dist` с готовыми статическими файлами. Для публикации на GitHub Pages нужно либо использовать встроенный GitHub Action, либо подключить пакет `gh-pages`, который автоматически загружает содержимое `dist` в ветку `gh-pages`. В настройках репозитория на GitHub необходимо указать, что сайт будет обслуживаться из этой ветки. После этого, при каждом обновлении кода и повторной сборке, достаточно заново загрузить содержимое `dist`, и сайт станет доступен по адресу вида `username.github.io/repository`. Таким образом, связка Figma AI Make и GitHub Pages позволяет быстро пройти путь от дизайна до работающего веб-сайта, сохраняя при этом возможность полной ручной доработки исходного кода.

### 3.2 Спецификация проекта

Организация данных включает в себя описание файлов, констант, переменных, баз данных, массивы и других данных, которые исполняют в проекте роль входной, выходной и постоянной информации и отображена в таблице 6.

					ТРПО 5-04 0612-02.43.06.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		20

Таблица 6 – Организация данных

Данные	Описание
1	2
ITEMS_PER_PAGE	Числовая константа (равная 6), определяющая количество карточек медицинских учреждений, отображаемых на одной странице при пагинации.
mockInstitutions	Массив объектов с начальными данными о медицинских учреждениях (больницы, поликлиники и т.д.) для демонстрации работы приложения.
mockReviews	Массив объектов с примерами отзывов пользователей для различных учреждений.
mockNews	Массив объектов с новостями медицины для наполнения раздела "Медицинские новости".
mockDoctors	Массив объектов с данными врачей, используемый для привязки специалистов к учреждениям.
defaultImages	Объект или массив со ссылками на изображения-заглушки (плейсхолдеры), которые отображаются, если у учреждения нет собственных фото (разделены по типам: hospital, clinic и т.д.).
categoryNames	Объект-словарь для маппинга технических названий категорий новостей (напр., 'health') в человекочитаемые русские названия (напр., 'Здоровье').
categoryColors	Объект, определяющий цветовые стили (CSS-классы Tailwind) для бейджей разных категорий новостей.
daysOfWeek	Массив объектов, содержащий ключи (напр., 'monday') и названия дней недели на русском языке для генерации формы графика работы.
institutions	Массив объектов, хранящий текущий список всех медицинских учреждений в приложении.
reviews	Массив объектов, хранящий список всех отзывов (как одобренных, так и ожидающих модерации).
news	Массив объектов, хранящий список медицинских новостей.
selectedInstitution	Объект или null, хранящий данные выбранного учреждения для отображения детальной страницы (карточки учреждения).
isAdminMode	Булево значение, определяющее текущий режим работы приложения (режим просмотра или режим администратора).
currentPage	Числовое значение, указывающее текущую активную страницу в пагинации списка учреждений.
activeTab	Строковое значение (например, 'institutions', 'news', 'reviews'), определяющее текущую активную вкладку в навигации или админ-панели.
filters	Объект состояния, содержащий все активные параметры фильтрации (query, priceType, specialization, district, workingNow, sortBy).

# Продолжение таблицы 6

1	2
isExpanded	Булево значение в компоненте фильтров, отвечающее за сворачивание/разворачивание панели расширенного поиска.
isMobileMenuOpen	Булево значение для управления видимостью выпадающего меню в мобильной версии хедера.
isAboutModalOpen	Булево значение, управляющее открытием/закрытием модального окна "О системе".
showAdminLogin	Булево значение, управляющее отображением модального окна входа в админ-панель.
isAddingInstitution / isAddingNews / isAddingDoctor	Булевы значения, управляющие открытием модальных окон создания новых записей.
editingInstitution / editingDoctor	Строка (ID) или объект, указывающий на сущность, которая в данный момент редактируется.
newReview	Объект, хранящий данные формы при написании нового отзыва (authorName, rating, comment).
showSuccessMessage	Булево значение для отображения уведомления об успешной отправке отзыва.
searchQuery	Строковая переменная, хранящая текст, введенный пользователем в строку поиска в хедере.
imageError	Булево значение, отслеживающее ошибку загрузки изображения (для переключения на запасное изображение).
filteredInstitutions	Массив учреждений, отфильтрованный и отсортированный на основе состояния filters и поискового запроса.
paginatedInstitutions	Подмножество массива filteredInstitutions, содержащее только элементы для текущей страницы (срез массива).
totalPages	Числовое значение, рассчитывающее общее количество страниц на основе отфильтрованных данных.
pendingReviews	Массив отзывов, ожидающих проверки администратором (isApproved: false).
approvedReviews	Массив отзывов, прошедших модерацию и доступных для просмотра (isApproved: true).
activeFiltersCount	Число, показывающее количество примененных пользователем фильтров для отображения бейджа

Структура и описание процедур и функций пользователя отображена в таблице 7.

Таблица 7 – структура и описание процедур

Структура	Описание
1	2
handleSearch(query)	Функция для быстрой фильтрации при клике на популярные теги (например, "Стоматология").
handleViewDetails(institution)	Функция, устанавливающая выбранное учреждение в состояние selectedInstitution для просмотра подробностей.

## Продолжение таблицы 7

1	2
handleBack()	Функция сброса selectedInstitution в null для возврата к списку.
handleAddReview(review)	Функция добавления нового объекта отзыва в общий массив reviews (с пометкой isApproved: false).
handleApproveReview(reviewId)	Функция администратора, меняющая статус отзыва на одобренный.
handleDeleteReview(reviewId)	Функция удаления отзыва из массива.
handleUpdateInstitution(id, updates)	Функция обновления данных существующего учреждения в массиве.
handleDeleteInstitution(id)	Функция удаления учреждения из массива по ID.
handleAddInstitution(institution)	Функция создания нового учреждения с генерацией уникального ID.
handleLogin(username, password)	Функция проверки учетных данных администратора (проверяет хардкод или localStorage).
isWorkingNow()	Логическая функция, проверяющая текущее время и день недели относительно графика работы учреждения, возвращает true или false.
getTypeLabel(type)	Функция-хелпер, преобразующая код типа учреждения (напр., 'hospital') в русское название ('Больница').
handleNavClick(section)	Функция навигации, выполняющая скролл к началу, фокус на поиске или открытие модальных окон.
onSearch	Событие, вызываемое при отправке формы поиска (submit), передает введенный запрос в родительский компонент.
onFiltersChange	Событие, вызываемое при изменении любого из фильтров (селекты, свитчи), передает обновленный объект фильтров.
onViewDetails	Событие клика по карточке учреждения, инициирующее переход к детальному просмотру.
onAdminToggle	Событие переключения режима администратора (вход/выход).
onSave	Событие в формах редактирования (учреждения, новости, врача), сохраняющее внесенные изменения.
onClose	Событие для закрытия модальных окон (диалогов) или отмены действия.
onError	Событие, срабатывающее, если основное изображение не удалось загрузить, переключает на запасное (fallback).

Структура и описание файлов отображена в таблице 8.

Таблица 8 – структура и описание файлов

Файл	Назначение
1	2
AboutModal.tsx	модальное окно
AdminLogin.tsx	вход администратора
AdminPanel.tsx	панель администратора

# Продолжение таблицы 8

1	2
Footer.tsx	нижний колонтитул
Header.tsx	верхний колонтитул
InstitutionDetails.tsx	детали учреждения
InstitutionEditor.tsx	редактор учреждения
MedicalInstitutionCard.tsx	карточка учреждения
MedicalNews.tsx	медицинские новости
NewsEditor.tsx	редактор новостей
ReviewSection.tsx	отзывы пользователей
SearchFilters.tsx	фильтры поиска
ImageWithFallback.tsx	изображение запасное
Ui.tsx	UI компоненты
mockData.ts	тестовые данные
globals.css	глобальные стили
App.tsx	корневой компонент
Attributions.md	авторские ссылки
index.css	базовые стили
main.tsx	точка входа
index.html	HTML шаблон
package.json	зависимости проекта
vite.config.ts	конфиг Vite

## 4 Тестирование

### 4.1 Тесты на использование

При разработке данного проекта многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Отчёт о результатах тестирования предоставлен в Приложении В.

### 4.2.Отчет о результатах тестирования

Команда тестировщиков описана в таблице 9.

Таблица 9 – Команда тестировщиков

ФИО	Должность	Роль
Гуз Елизавета Александровна	Разработчик, тестировщик	Ответственный тестирование приложения

Тестирование производилось под ОС Windows 10 x64. Все тест-кейсы выполнялись вручную. Тестирование показало высокую стабильность функциональности. Расписание проведения тестирования представлено в таблице 10.

Таблица 10 – Расписание проведения тестирования

ФИО	Дата	Деятельность	Продолжительность, ч
Гуз Елизавета Александровна	04.11.2025	Разработка тест-кейсов	2
Гуз Елизавета Александровна	06.11.2025	Тестирование приложения	3
Гуз Елизавета Александровна	10.11.2025	Анализ выполнения тест-кейсов	1
Гуз Елизавета Александровна	12.11.2025	Тестирование приложение. Проведение регрессионного тестирования	3
Гуз Елизавета Александровна	13.11.2025	Составление отчёта о результатах тестирования	2

Разработанные тест-кейсы и результаты тестирования представлены в приложении В. Статистика по разработанным тест-кейсам представлена в таблице 11.

					ТРПО 5-04 0612-02.43.06.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		25

Таблица 11 – Статистика по разработанным тест-кейсам

Статус	Количество тест-кейсов	Приоритет				
		Крайне низкий	Низкий	Средний	Высокий	Крайне высокий
Пройдено	52	16	9	11	9	7
Не пройдено	0	0	0	0	0	0
Заблокировано	0	0	0	0	0	0
Не протестировано	0	0	0	0	0	0

## 5 Руководство пользователя

### 5.1 Руководство пользователя для гостя

Так как сайт является веб-платформой, установка не требуется. Доступ осуществляется через браузер по адресу, предоставленному разработчиками. Все функции доступны сразу после загрузки страницы.

#### 1. Начало работы сайта.

Откройте сайт. Перед вами появится главный экран (рисунок 1).

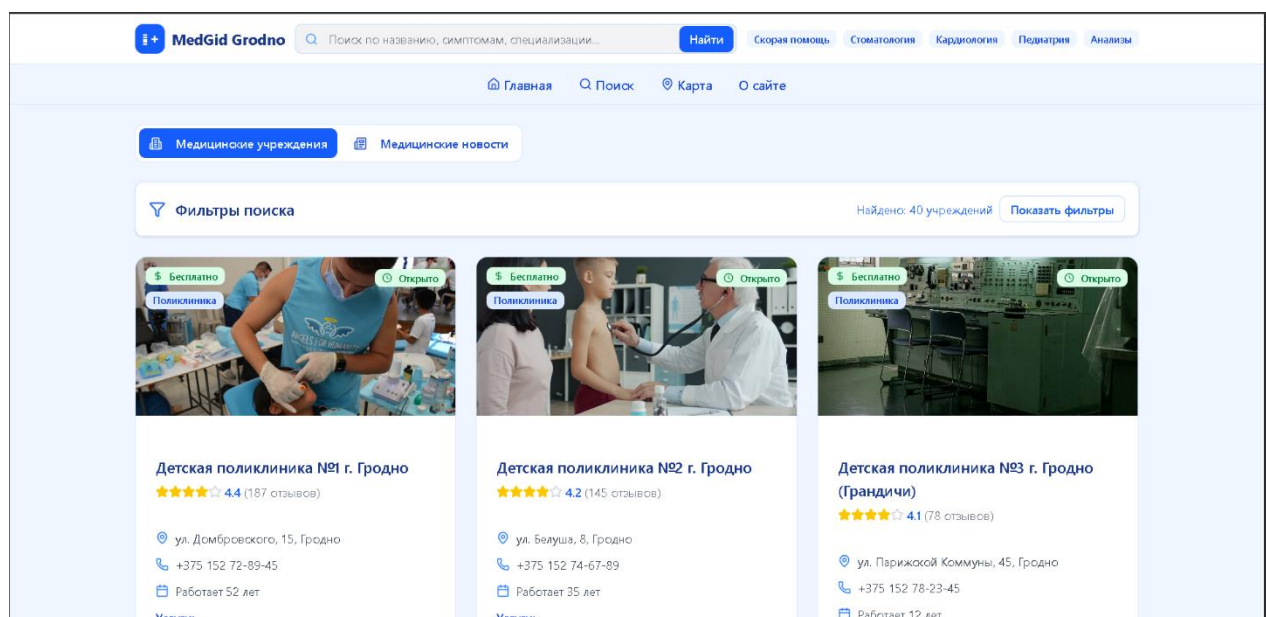


Рисунок 1 – Главный экран

#### 2. Просмотр ленты медицинских учреждений и новостей.

Для ознакомления на главной странице сайта показан список медицинских учреждений (рисунок 2) и новостей (рисунок 3).

					ТРПО 5-04 0612-02.43.06.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		27



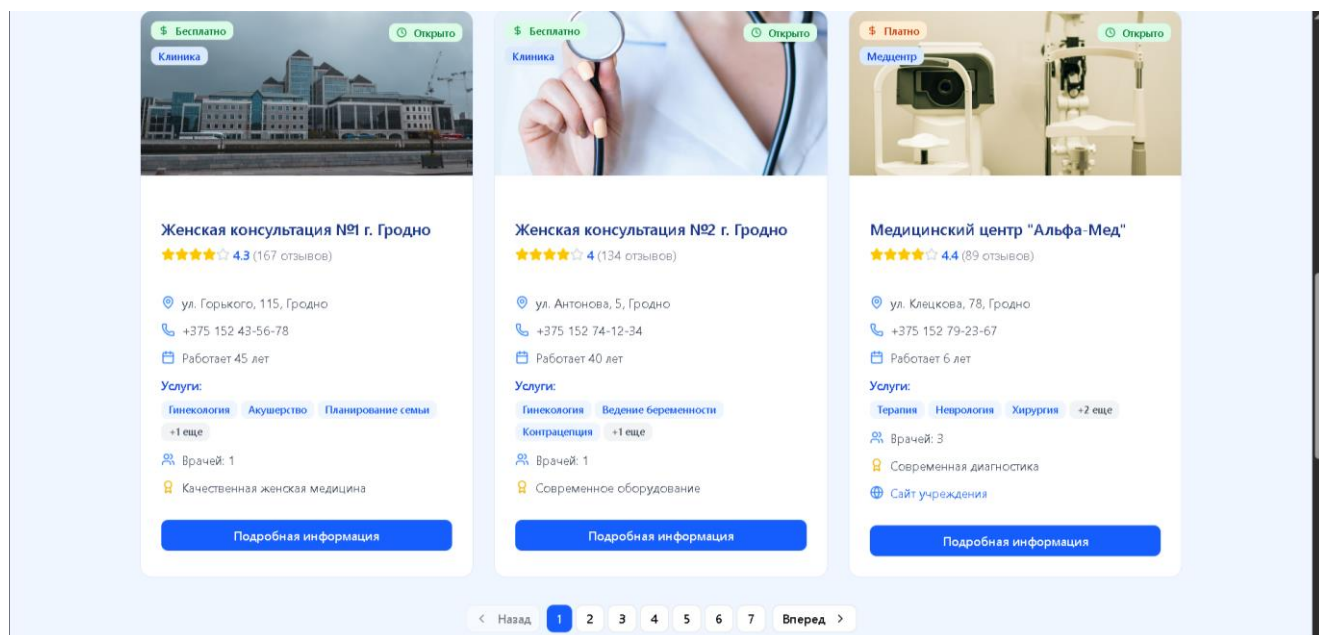


Рисунок 2 – Лента медицинских учреждений

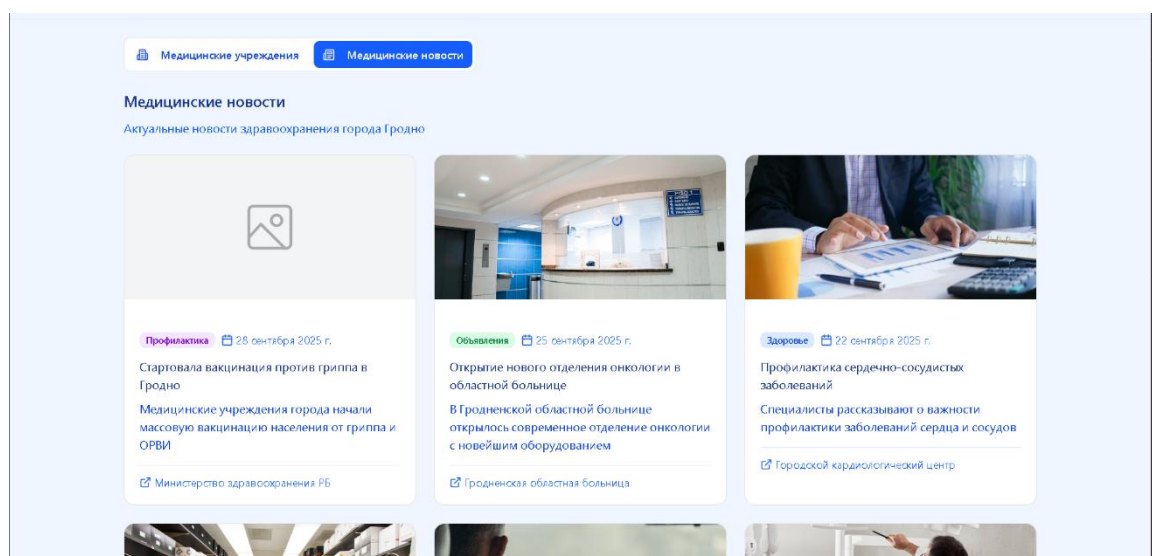


Рисунок 3 – Лента новостей

3. Все гости могут использовать фильтры для поиска определенных медицинских учреждений (рисунок 4), просматривать подробную информацию о медицинских учреждениях (рисунок 5) и оставлять отзывы (рисунок 6).

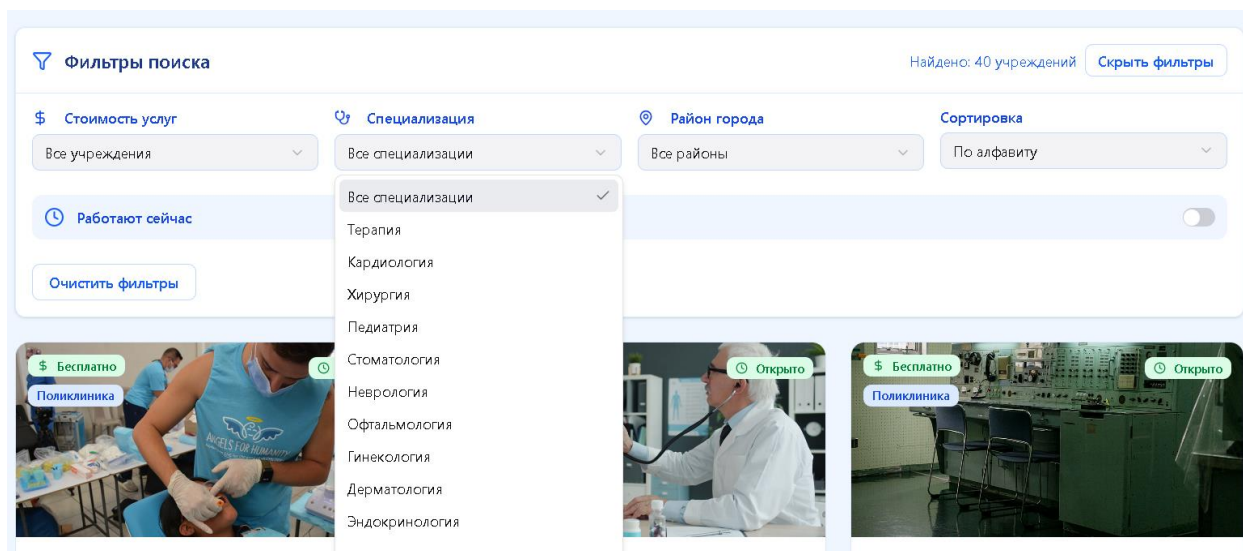


Рисунок 4 – Фильтры поиска

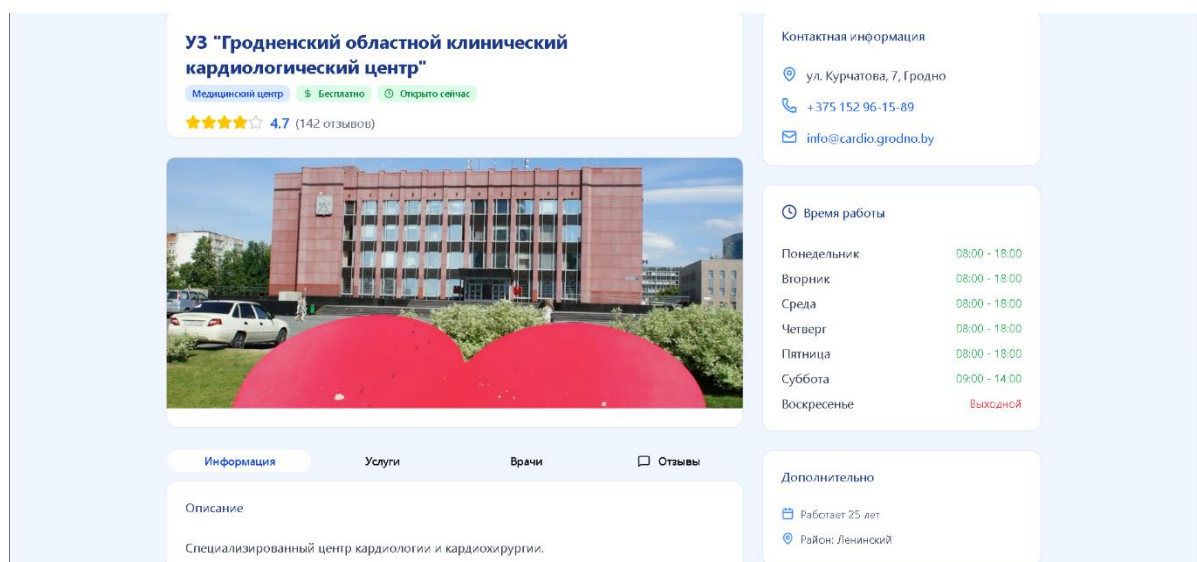


Рисунок 5 – Создание поста

Написать отзыв

Ваше имя

Введите ваше имя

Оценка

★

★

★

★

★

5 из 5

Комментарий

Поделитесь своим опытом посещения этого медицинского учреждения...

Отправить отзыв

Отмена

гость

15 сентября 2024 г.

★

★

★

★

★

5/5

Лучшая детская поликлиника в городе! Врачи очень любят детей, всегда найдут подход к ребенку.

Рисунок 6 – Оставление отзывов

5.2 Руководство пользователя для администратора

4 Регистрация.  
Для дальнейшей полноценной работы в панели администратора необходимо зарегистрироваться через форму (рисунок 7).

Рисунок 7 – Форма регистрации

4. Администратор может добавлять и удалять медицинские учреждения (рисунок 8), новости (рисунок 9) и модерировать отзывы (рисунок 10).

Рисунок 8 – Добавление медицинского учреждения

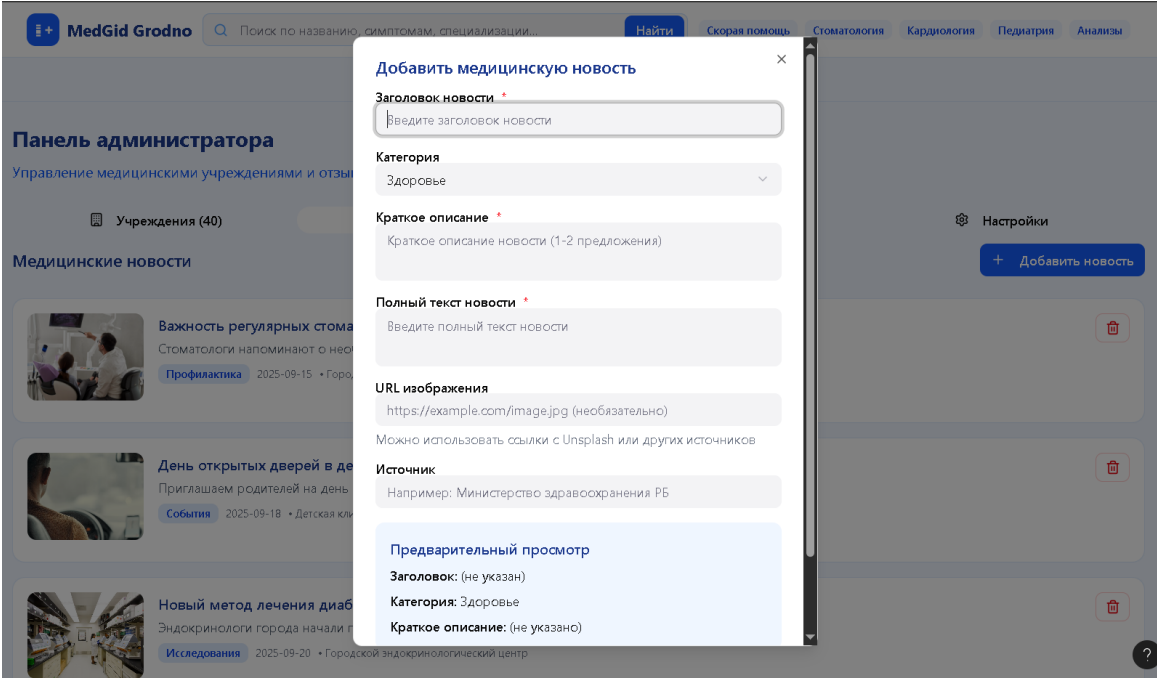


Рисунок 9 – Добавление новости

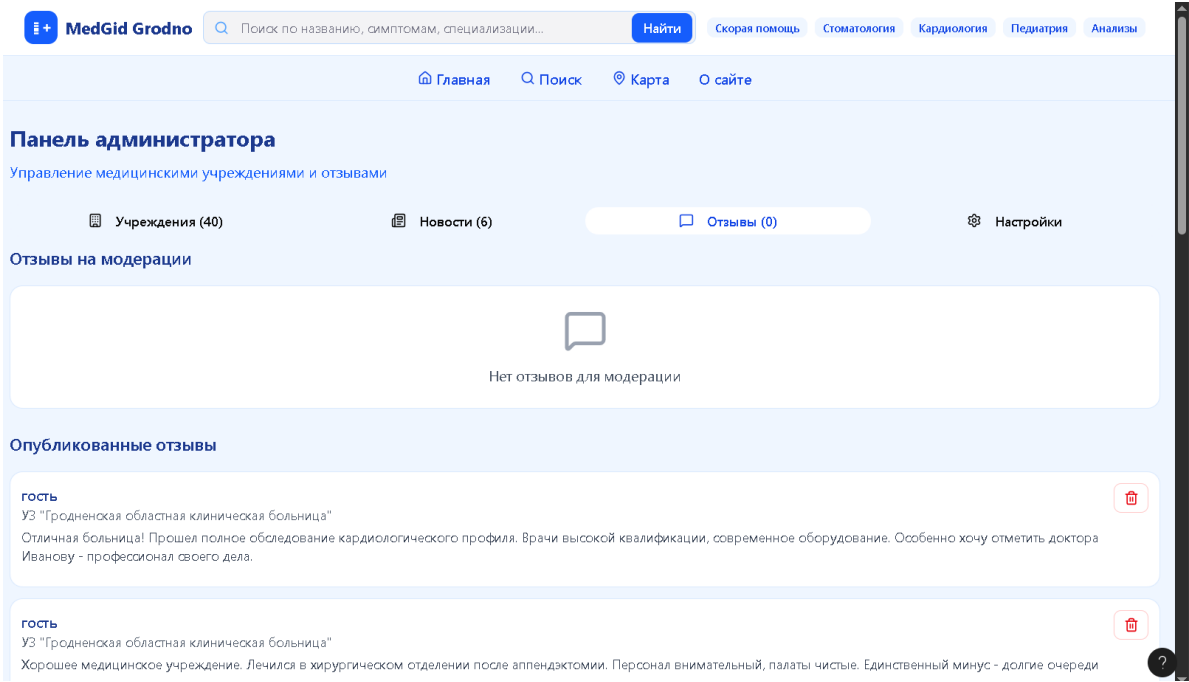


Рисунок 10 – Модерация отзывов

## Заключение

В рамках данной работы был разработан веб-сайт «MedGid Grodno», предназначенный для поиска и удобного представления информации о медицинских учреждениях города Гродно. Основная идея проекта заключалась в создании доступного, простого в использовании ресурса, который объединяет сведения о больницах, поликлиниках, врачах и предоставляемых услугах, помогая жителям и гостям города быстро находить подходящие варианты по своим запросам. Особенностью решения стало внедрение поиска не только по названию учреждения, но и по симптомам или жалобам, что делает систему более ориентированной на реальные потребности пользователей.

Поставленная задача была выполнена в полном объеме. Реализована последовательная структура взаимодействия: от поиска учреждения по фильтрам и ключевым параметрам до просмотра детальной информации, отзывов и новостей. Разработка охватила визуальное оформление, логику интерфейса, обработку пользовательских действий, адаптивную верстку и отображение актуальных данных о медицинских организациях. Пользователь может не только просматривать учреждения, но и оставлять отзывы, уточнять время работы и стоимость услуг, что обеспечивает удобное и понятное взаимодействие.

Для реализации проекта использовались языки HTML и CSS для статических страниц, JavaScript для интерактивных элементов и PHP для динамических страниц.

Проектные решения полностью соответствуют изначальным требованиям: сайт работает в онлайн-режиме, поддерживает удобную навигацию, предоставляет структурированную информацию о медицинских учреждениях и врачах, а также адаптируется под различные устройства.

В целом созданный веб-сайт является практичным, наглядным и полезным инструментом для жителей города Гродно, позволяющим быстро находить актуальные сведения о медицинских услугах и учреждениях. Он сочетает простоту, доступность и функциональность — всё, что необходимо для эффективного и востребованного городского сервиса.

					ТРПО 5-04 0612-02.43.06.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		33

### Список использованных источников

1 Главное управление здравоохранения Гродненского облисполкома [Электронный ресурс]. – Режим доступа: <https://grodnouzo.gov.by/> – Дата доступа: 06.12.2025.

2 Talon.by [Электронный ресурс]. – Режим доступа: <https://talon.by/policlinics/grodno>– Дата доступа: 04.12.2025.

3 103.by [Электронный ресурс]. – Режим доступа: <https://www.103.by/cat/med/medicinskie-centry/grodno/>– Дата доступа: 04.12.2025.

					ТРПО 5-04 0612-02.43.06.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		34