Министерство образования и науки Республики Башкортостан

Государственное автономное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

УТВЕРЖДАЮ

Заместитель директора

по учебной работе

\_\_\_\_\_\_\_\_ З.З.Курмашева

«\_\_\_» \_\_\_\_\_\_\_\_\_\_2024 г.

ПРОЕКТИРОАНИЕ И РАЗРАБОТКА АИС «БОЛЬНИЦА»

Пояснительная записка к курсовому проекту

МДК.05.02 Разработка кода информационных систем

Руководитель проекта

\_\_\_\_\_\_\_\_\_\_\_ Е.К. Дмитриева

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Студент гр. 21ВЕБ-1

\_\_\_\_\_\_\_\_\_\_\_\_\_ Миронова Н.Е

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

2024

Министерство образования и науки Республики Башкортостан

Государственное бюджетное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заместитель директора  по учебной работе  \_\_\_\_\_\_\_\_\_\_\_\_З.З. Курмашева  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |

ЗАДАНИЕ

на курсовой проект студенту дневного отделения, группы 21ВЕБ-1, специальности 09.02.07«Информационные системы и программирование».

Фамилия, имя, отчество: Миронова Наталья Евгеньевна

Тема курсового проекта: Проектирование и разработка АИС «Больница»

Текст задания:

при выполнении курсового проекта должны быть решены следующие задачи:

1. разработана структура программы;
2. реализованы функции просмотра направлений, врачей, запись на прием, отправка письма о записи на почту.

В результате выполнения курсового проекта должны быть представлены:

1. пояснительная записка, состоящая из следующих разделов:

Введение

1 Проектирование информационной системы

2 Экспериментальный раздел

Заключение

Приложения

Список сокращений

Список источников

1. электронный носитель, содержащий разработанный программный продукт;
2. презентация курсового проекта в электронном виде.

Список рекомендуемой литературы:

1. Перлова, О.Н. Проектирование и  разработка информационных систем : учебник для студ. учреждений сред. проф. образования  / О.Н. Перлова, О.П. Ляпина, А.В. Гусева.  — 3-е изд.,испр. — М. : Издательский центр «Академия», 2020. — 256 с. ISBN 978-5-4468-9563-2. - Текст : электронный. - URL: <https://academia-moscow.ru/reader/?id=480245>  (дата обращения: 09.02.2022). – Режим доступа: по подписке.
2. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие / Г. Н. Федорова. — М.: КУРС : ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1138896>  (дата обращения: 06.08.2021). – Режим доступа: по подписке.

Задание к выполнению получил «\_\_» \_\_\_\_\_\_ 20\_\_ г.

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_Сидоров Сидор Сидорович

Срок окончания «\_\_» \_\_\_\_\_\_\_ 20\_\_ г.

Руководитель курсового проекта П.П. Петров

Задание рассмотрено на заседании цикловой комиссии информатики и программирования«\_\_» \_\_\_\_\_\_ 20\_\_ г.

Председатель цикловой комиссии информатики О.В. Фатхулова

Министерство образования и науки Республики Башкортостан

Государственное бюджетное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

ЗАКЛЮЧЕНИЕ

на курсовой проект

Студент

Группа

Специальность 09.02.07 Информационные системы и программирование

Тема

Объем курсового проекта:

количество листов пояснительной записки

количество листов графической части

Заключение о степени соответствия заданию на курсовое проектирование

Характеристика качеств, проявленных студентом при работе над проектом: самостоятельность, дисциплинированность, умение планировать работу и пользоваться литературным материалом и т.д.

Положительные стороны курсового проекта

Недостатки курсового проекта

Характеристика общетехнической и специальной подготовки студента

Заключение и предлагаемая оценка за курсовой проект

Руководитель курсового проекта

«» г.

Подпись

АННОТАЦИЯ

Пояснительная записка к курсовому проекту содержит постановку и программу решения задачи «Проектирование и разработка АИС «Больница» ». Программа написана на языке PHP, использовался фреймворк bootstrap, HTML и CSS. В качестве СУБД использовался MySQL,

Программа предназначена для работы в браузерах любой операционной системы.

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
| Введение | 3 |
| 1 Проектирование информационной системы | 10 |
| 1.1 Описание предметной области | 10 |
| 1.2 Описание входной информации | 14 |
| 1.3 Описание выходной информации | 15 |
| 1.4 UML диаграммы | 16 |
| 1.5 Концептуальное моделирование | 18 |
| 1.6 Логическое моделирование | 19 |
| 1.7 Описание структуры базы данных | 20 |
| 1.8 Контрольный пример | 23 |
| 1.9 Формирование требований к ИС | 23 |
| 2 Экспериментальный раздел | 28 |
| 2.1 Описание программы | 28 |
| 2.2 Руководство пользователя | 36 |
| Заключение | 49 |
| Список сокращений  Список литературы | 50  51 |
| Приложения | 52 |

ВВЕДЕНИЕ

Больница является сложной организацией, которая обрабатывает огромный объем информации. Эффективное управление этой информацией имеет решающее значение для обеспечения качественного медицинского обслуживания и удовлетворенности пациентов. В прошлом больницы использовали преимущественно бумажные системы для хранения и обработки медицинских данных. Однако эти ручные системы имели недостатки, включая трудности с организацией архивов, дублирование записей и медленный поиск информации. Для того чтобы записаться на прием нужно было прийти или позвонить в больницу, простоять в очереди. Чтобы преодолеть эти трудности, все больше больниц внедряют информационные системы (ИС), которые позволяют им хранить и управлять медицинскими данными. База данных больницы используется для хранения информации о врачах, о специальностях, пациентах, их диагнозах и о записях на прием.

Эти информационные системы предоставляют ряд преимуществ, включая:

* улучшенная организация и поиск информации: ИС позволяют больницам организовывать и хранить медицинские данные в структурированном и централизованном виде, что облегчает поиск и извлечение необходимой информации;
* уменьшение дублирования: ИС устраняют необходимость в ведении нескольких бумажных записей, что снижает риск дублирования и ошибок;
* повышение точности и согласованности: ИС помогают стандартизировать и автоматизировать процессы ввода данных, что улучшает точность и согласованность медицинской информации;
* повышение эффективности: ИС автоматизируют многие ручные задачи, что высвобождает время медицинского персонала для оказания медицинской помощи пациентам;
* улучшение обслуживания пациентов: ИС позволяют пациентам взаимодействовать с больницей в режиме онлайн, что повышает удобство и доступность медицинских услуг. Внедрение информационной системы в больнице является важным шагом на пути к модернизации и повышению эффективности работы. Это позволяет больницам предоставлять пациентам более качественные и своевременные медицинские услуги.

Цель работы: упростить ведение медицинской карточки и работу медицинского персонала путем разработки АИС «Больница».

Задачи:

* провести предпроектное исследование;
* выполнить проектирование системы;
* разработать базу данных;
* разработать клиентское программное обеспечение;
* провести тестирование программы;
* написать руководство пользователя.

1 Проектирование информационной системы

1.1 Описание предметной области

Больница — это стационарное медицинское учреждение, где проходит лечение, диагностический поиск или уточнение причины заболевания, которое стало причиной обращения в него.

Информационная система создается для упрощения записи посетителей к врачу. Клиент зайдет на сайт, где сможет ознакомиться с врачами и свободным временем, на которое он может записаться. Все данные о пользователе, история посещений врачей будут храниться в личном кабинете. После успешной записи на прием человек получит письмо на почту о записи.

Пользователями системы являются:

* пользователь;
* врач;
* администратор.

Пользователь должен иметь в системе следующие возможности:

* запись на прием;
* отмена записи;
* просмотр свободного времени для записи;
* просмотр врачей;
* просмотр направлений;
* редактирование персональных данных;
* доступ к информации о записях;
* прикрепление к больнице;
* доступ к электронной медицинской карте;
* доступ к личному кабинету.

Врачи должны иметь в системе следующие возможности:

* запись посетителей на прием;
* изменение статуса записи;
* доступ к личному кабинету;
* доступ к карточке пациентов.

Администратор в системе может иметь следующие возможности:

* добавление новых врачей;
* удаление из базы данных старых врачей;
* управление специальностями;
* управление направлениями.

Информация, хранимая в базе данных информационной системы больницы:

* сведения о врачах;
* информация о специальностях;
* информация о записях на прием;
* информация о электронной медицинской карте;
* информация о направлениях;
* сведенияо посетителях.

Основные сущности:

* посетители;
* карточка;
* врачи;
* специальность;
* направления;
* записи.

Пациент может прикрепиться к больнице пациент указав свои персональные данные. Данные о пользователе хранятся в системе и характеризуются следующими полями:

* код посетителя;
* почта;
* фамилия;
* имя;
* отчество;
* полис;
* паспорт;
* город;
* пароль;
* дата рождения.

После того как пациент прикрепился к больнице, у него открывается возможность записи на приём к врачам. Для этого нужно ознакомиться со списком направлений специалистов, где каждое направление характеризуется:

* код направления;
* описание направления;
* изображение;
* названиенаправления.

Так же пациент может посмотреть список врачей, который принимает в больнице, каждый врач характеризуется:

* код врача;
* специальность;
* фамилия врача;
* имя;
* отчество;
* почта;
* пароль;
* изображение.

У каждого врача есть специальность, которая характеризуется:

* код специальности;
* название специальности;
* код направления.

После того как пациент выбрал нужного врача он оформляет запись на определенную дату и время, к определенному врачу. О записях хранятся следующие сведения:

* номер записи;
* врач;
* медицинская карта пациента;
* дата записи;
* время записи;
* диагноз;
* записи доктора;
* статус (новая, отменена, пройдена, не пришел).

После первого приема у врача на пациента создается электронная медицинская карточка. Она характеризуется следующими данными:

* код карточки;
* код посетителя;
* полис;
* дата рождения;
* город;
* паспорт.

При каждом посещении врача карточка обновляется с новыми записями о приеме.

Ограничения в информационной системе:

* к одному врачу на одно и тоже дату и время не могут записаться два пациента;
* клиент не может записаться на прием к врачу с 20:00 до 08:00 утра;
* пациент, который не прикрепился не может записываться на прием.

1.2 Описание входной информации

Входная информация служит для осуществления деятельности системы, в которую входят данные пользователя при регистрации.

Информация о пользователе формируется на основе:

* фамилия пациента;
* имя;
* отчество;
* почта;
* пароль;
* полис;
* паспорт;
* город;
* дата рождения.

Так же входной информацией для системы являются данные о врачах, которые заносятся в базу в момент создания новой записи о враче администратором и характеризуются:

* фамилия врача;
* имя;
* отчество;
* почта;
* пароль;
* изображение;
* специальность.

Так же входной информацией для системы являются данные о специальности врачей, которые заносятся в базу в момент создания новой записи о специальности администратором и характеризуются:

* название специальности;
* направление.

Администратор вносит в базу новые направления, которые характеризуются:

* описание направления;
* изображение;
* название направления;
* специальности.

1.3 Описание выходной информации

Выходная информация — информация, которая возникает в результате обработки человеком или устройством входной информации.

Основные требования к выходной информации сводятся к обеспечению пользователей сайта данными в удобной для них форме.

После обработки всей входной информации на выходе получается документ, содержащий в себе данные о занятости записи на прием.

Описание выходных документов представлено в таблице 1.3.1.

Таблица 1.3.1 — Описание выходных документов

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование документа (шифр) | Периодичность выдачи | Откуда поступает документ | Количество |
| Письмо «Запись на прием» | При записи на прием | Система | 1 |
| Электронная медицинская карточка | После посещения врача | Система | 1 |

Шаблон выходного документа представлен на рисунке 1.3.1

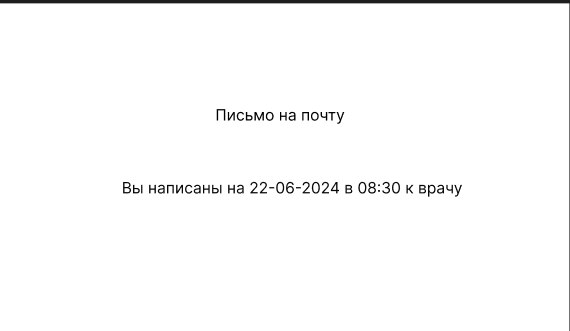


Рисунок 1.3.1- Шаблон выходного документа

1.4 UML диаграммы

UML (с английского аббревиатура расшифровывается как UnifiedModelingLanguage — унифицированный язык моделирования) — это способ наглядно описать архитектуру, проектирование и реализацию комплексных программных систем. Для разрабатываемой системы было решено спроектировать следующие диаграммы:

* диаграмма прецедентов;
* диаграмма классов.

Диаграмма прецедентов (диаграмма вариантов использования) -диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Диаграмма прецедентов представлена на рисунке 1.4.1.

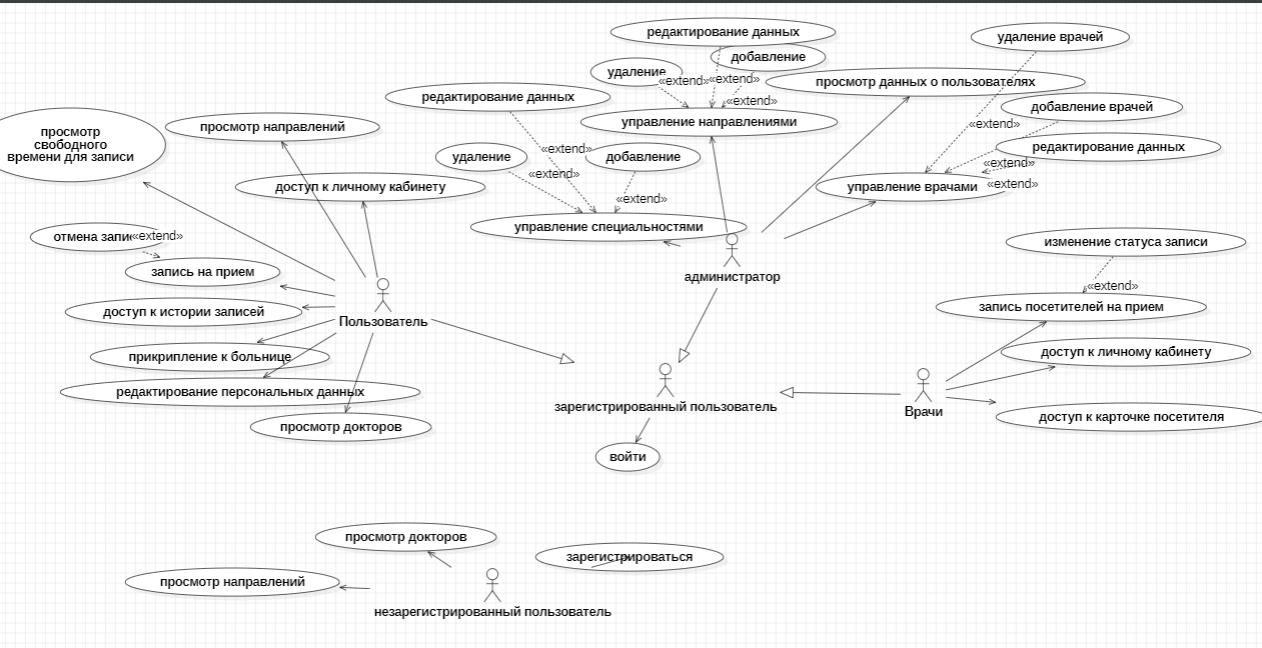
Рисунок 1.4.1 — Диаграмма прецедентов

Диаграмма классов (англ. classdiagram) — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними.

Диаграмма классов представлена на рисунке 1.4.2.

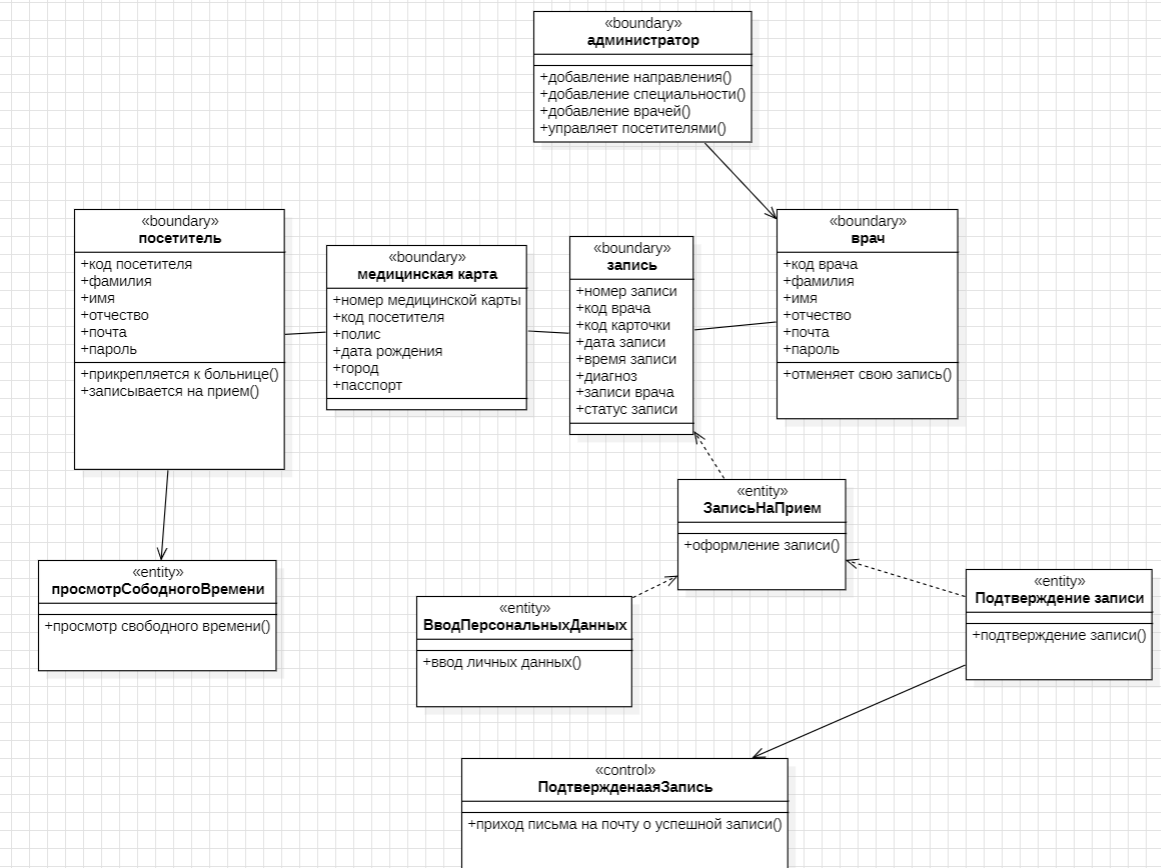


Рисунок 1.4.2— Диаграмма классов

1.5 Концептуальное моделирование

Концептуальное моделирование - это процесс создания абстрактной модели, которая представляет основные концепции, сущности и взаимосвязи в предметной области. Оно используется для понимания, описания и анализа систем, процессов или явлений.

Концептуальное моделирование представлено на рисунке 1.5.1.

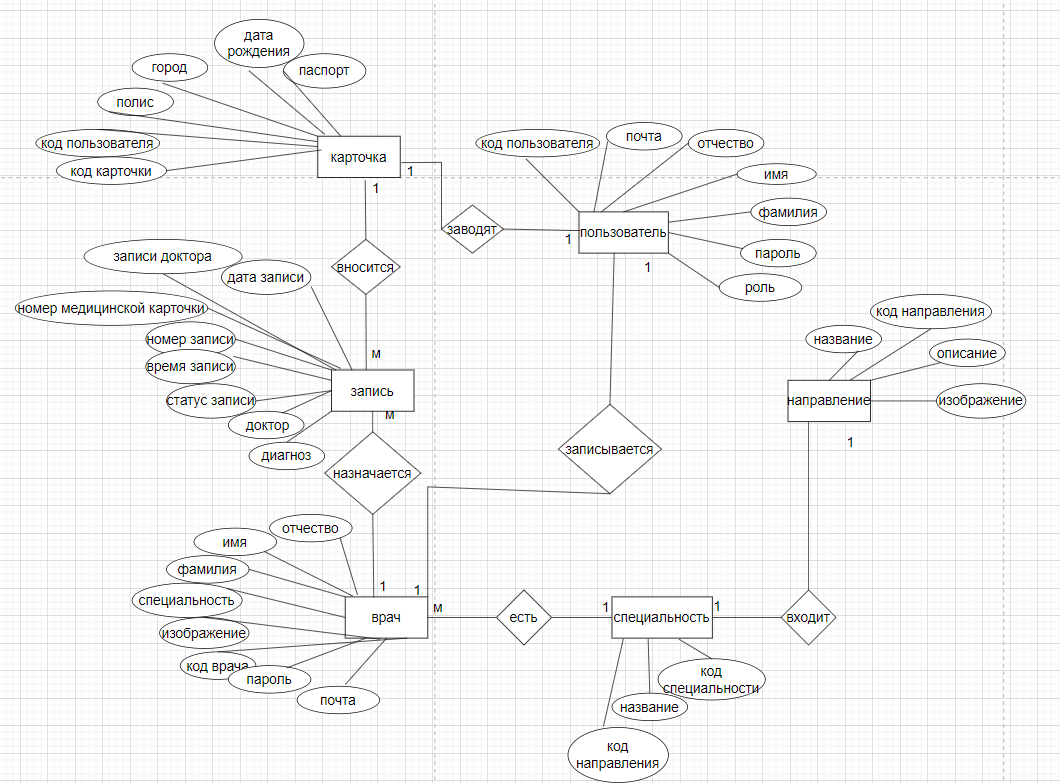


Рисунок 1.5.1 - ER- модель нотации Чена

1.6 Логическое моделирование

Логическое моделирование - это процесс создания структурированной модели, которая описывает данные, их отношения, ограничения и правила обработки данных на более детальном уровне, чем концептуальное моделирование.

Логическое моделирование сайта представлено на рисунке 1.6.1.

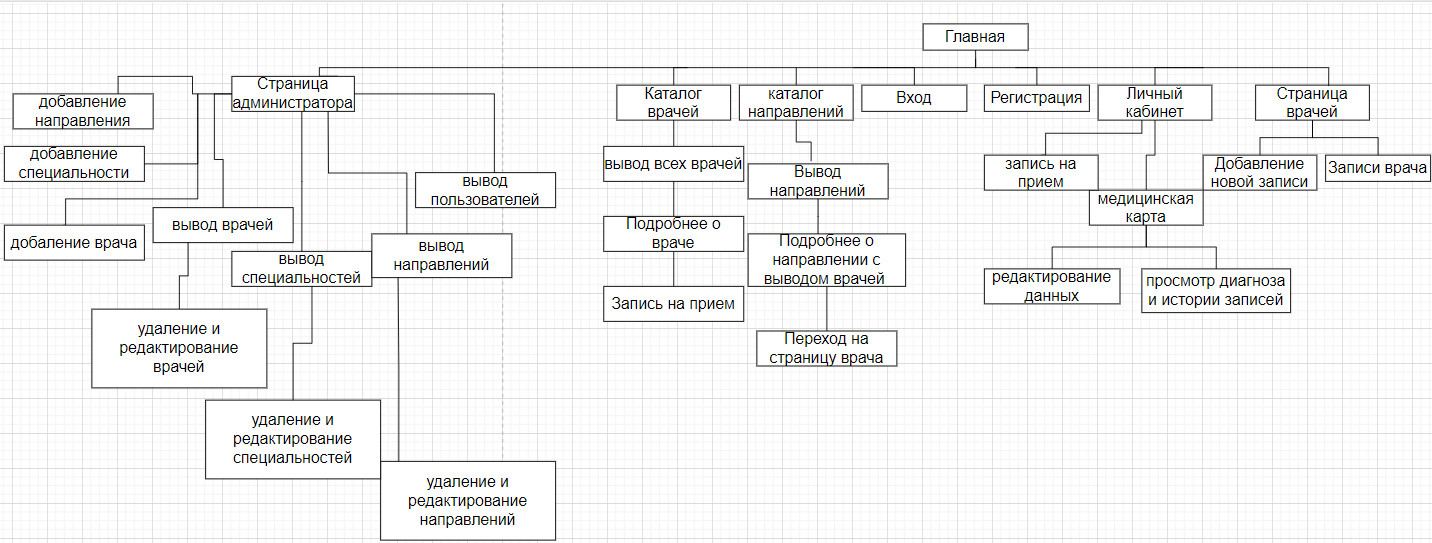


Рисунок 1.6.1 – Логическое моделирование

1.7 Описание структуры базы данных

Схема отношений базы данных представлена на рисунке 1.7.1.



Рисунок 1.7.1 – Схема отношений базы данных

Структура таблицы user представлена в таблице 1.7.1.

Таблица 1.7.1 - Users(Пользовтатели)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код пользователя | Id\_user | INTEGER(4) | Первичный ключ, уникальное значение |
| Почта пользователя | Email\_user | VARCHAR(30) | Обязательное поле, уникальное значение |
| Фамилия | surname | VARCHAR(30) | Обязательное поле |
| Имя | userName | VARCHAR(60) | Обязательное поле |
| Отчество | patronymic | VARCHAR(60) | Обязательное поле |
| Пароль | password | VARCHAR(30) | Обязательное поле |
| Роль | role | VARCHAR(7) | Обязательное поле |

Структура таблицы doctor представлена в таблице 1.7.2.

Таблица 1.7.2- Doctor(Врачи)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код доктора | Id\_doctor | INTEGER(4) | Первичный ключ,  уникальное значение |
| Код специальности | Id\_specialization | INTEGER(4) | Внешний ключ (к specialization ) |
| Код пользователя | Id\_user | INTEGER(4) | Внешний ключ (к users ) |
| Изображение | images | VARCHAR(100) | Обязательное поле |

Структура таблицы record представлена в таблице 1.7.3.

Таблица записей 1.7.3- Record(Записи)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код записи | Id\_record | INTEGER(4) | Первичный ключ, уникальное значение |
| Код доктора | Id\_doctor | INTEGER(4) | Внешний ключ (к  doctor ) |
| Код пользователя | Id\_user | INTEGER(4) | Внешний ключ (к Users ) |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Дата записи | date\_record | DATE | Обязательное поле |
| Время записи | time\_record | TIME | Обязательное поле |
| Диагноз | diagnosis | VARCHAR(100) |  |
| Записи доктора | doctor\_notes | VARCHAR(100) |  |

Структура таблицы card представлена в таблице 1.7.4.

Таблица 1.7.4 – Card(Карточка)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код карточки | Id\_card | INTEGER(4) | первичный ключ, уникальное значение |
| Код пользователя | Id\_user | INTEGER(4) | Внешний ключ (к Users) |
| Полис | policy | VARCHAR(16) | Обязательное поле , уникальное значение |
| Дата рождения | dateBirth | DATE | Обязательное поле |
| Город | city | VARCHAR(30) | Обязательное поле |
| Паспорт | passport | INTEGER(4) | Обязательное поле,уникальное значение |

Структура таблицы card представлена в таблице 1.7.5.

Таблица 1.7.5-Directions(Направления)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код направления | id\_directions | INTEGER(4) | Первичный ключ,уникальное значение |
| Название | name\_directions | VARCHAR(30) | Обязательное поле |
| Описание | description | VARCHAR(100) | Обязательное поле |
| Изображение | images | VARCHAR(100) | Обязательное поле |

Структура таблицы specialization представлена в таблице 1.7.6.

Таблица 1.7.6-Specialization(Специальности)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код специальности | id\_specialization | INTEGER(4) | Первичный ключ,уникальное значение |
| Название | name\_specialization | VARCHAR(30) | Обязательное поле |
| Код направления | id\_direction | VARCHAR(100) | Внешний ключ (к directions) |

1.8 Контрольный пример

Контрольный пример - это документ, который оформляет программное обеспечение и позволяет осуществить полную проверку функционирования разработанного программного модуля.

Контрольный пример представлен в приложении в таблицах А.1 — А.4 приложение А.

1.9 Формирование к требованиям ИС

1.9.1Введение

1. краткое описание проекта:

* Республиканская клиническая больница(РКБ) предназначен для поиска врача и записи к на прием.

1. цели и задачи разработки:

* РКБ создается с целью автоматизации записи на прием к врачу.

1. ссылки на связанные документы или проекты:

* отсутствуют.

1.9.2 Функциональные требования

1. Подробное описание функциональности, которую должен предоставлять программный продукт

Программный продукт должен иметь функционал:

* добавление, редактирование и удаление врачей;
* добавление, редактирование и удаление направлений;
* добавление, редактирование и удаление специальностей;
* вывод информации о врачах и направлениях;
* авторизация и регистрация;
* запись на прием.

1. Входы и выходы системы

Входы:

* данные пользователя;
* данные о врачах;
* данные о направлениях;
* данные о специальностях;
* данные о записях.

Выходы:

* письмо о записи на прием;
* электронная медицинская карта.

1. Взаимодействие с другими системами:

Отсутствует.

1.9.3 Требования к интерфейсу

1. Описание пользовательского интерфейса

Главная страница:

* направления с названием, описанием и изображением;
* врачи с именем, специальностью и фотографией врача;
* новости;
* контакты.

Регистрация:

* возможность ввода персональных данных: фамилия, имя, отчество, почта, город, паспорт, полис, день рождение, пароль;
* все поля являются обязательными для ввода.

Авторизация:

* возможность ввода персональных данных: почта, пароль;
* все поля являются обязательными для ввода.

Страница с врачами:

* фамилия, имя, отчество, специальность и изображение врача;
* при нажатии на фото врача должен происходить переход на страницу для записи на прием.

Страница с направлениями:

* название, описание и фото направления;
* при нажатии на фото направления должен происходитьпереход на страницу подробнее о направлении.

Страница подробно о направлении:

* название, описание и изображение направления;
* фамилия, имя, отчество, специальность и изображениеврача, относящегося к данному направлению;
* при нажатии на фото врача должен происходить переход на страницу для записи на прием.

Страница записи на прием:

* фамилия, имя, отчество, специальность и изображение врача;
* выбор даты с возможностью записи;
* кнопка «Записаться».

Профиль:

* личные данные;
* история записей;
* кнопка для перехода к электронной медицинской карте;
* кнопка для записи на прием.

Страница электронной медицинской карты:

* редактирование личных данных;
* личные данные;
* просмотр истории записей и диагнозов.

Панель администратора:

* управление пользователями;
* управление специальностями;
* управление направлениями;
* управление врачами.

Страница управления пользователями:

* вывод пользователей и персональных данных;

Страница управления направлениями:

* вывод направлений;
* кнопки для удаления и редактирования;
* кнопка для добавления направления.

Страница управления врачей:

* вывод врачей;
* кнопки для удаления и редактирования;
* кнопка для добавления врача.

Страница управления специальности:

* вывод специальностей;
* кнопки для удаления и редактирования;
* кнопка для добавления специальности.

Страница добавления врача/специальности/направления:

- ввод данных;

- кнопка «Добавить».

1. Требования к навигации и взаимодействию пользователя с системой

Навигационное меню пользователя: меню для быстрого доступа ко всем разделам (Профиль, Главная, Врачи, Направления, Контакты, Вход/Выход);

Навигационное меню администратора: меню для быстрого доступа ко всем разделам (Врачи, Направления, Пользователи, Специальности, Выход);

Навигационное меню врачей: меню для быстрого доступа ко всем разделам (мои записи, запись на прием);

Адаптивность: отзывчивый дизайн для различных разрешений экрана.

1.9.4 Требования к безопасности

1. Требования к управлению доступом и аутентификации

Разграничение прав доступа пользователей, врачей и администраторов.

1.9.5 Кроссбраузерность:

Сайт должен открывается в Opera, Edge,Chrome.

2 Экспериментальный раздел

2.1 Описание программы

Описание модулей(классов) представлено в таблице 2.1.1.

Таблица 2.1.1 – Описание модулей

|  |  |
| --- | --- |
| Методы | Назначение |
| 1 | 2 |
| Класс: Connect.php | |
| \_\_construct() | Метод для обращения к базе данных. |
| Класс User.php | |
| signup($surname, $name, $patronymic, $city, $policy, $passport, $dateBirth, $email, $password) | Регистация пользователя |
| selectDataUser($id\_user) | Вывод всей информации о пользователе по id |
| signin($email, $password) | Для авторизации пользователя |
| getUsers () | Для вывода всех пользователей в админ панели |
| Класс: Doctor.php | |
| checkEmpty($value, $field, $message) | Для проверки на пустоту |
| validateAddDoctor($surname, $name, $patronimyc, $specialization, $images, $email, $password) | Проверка на приход всех данных при добавлении врача |
| addDoctor($surname, $name, $patronimyc, $specialization, $images, $email, $password ) | Добавление врача |
| getDoctors () | Метод для вывода всех врачей |
| getDoctorsBy4() | Метод для вывода врачей и информации о них, на странице докторов по 4 в каждой строке |
| deleteDoctor($id\_doctor) | Метод для удаления врача по id |
| getDoctor ($id\_doctor) | Метод для вывода информации о враче по id |
| validateEditDoctor($id\_doctor, $name, $specialization, $images, $email, $password) | Проверка на приход данных при редактировании информации о врачах |

Продолжение таблицы 2.1.1

|  |  |
| --- | --- |
| 1 | 2 |
| editDoctor($id\_doctor, $name, $specialization, $images, $email) | Метод для добавления новых врачей |
| Класс: Direction.php | |
| getDirection ($id\_direction) | Вывод всей информации о направлении по id |
| getDirections () | Вывод всей информации о направлении |
| deleteDirection ($id\_direction) | Метод для удаления направления по id |
| checkEmpty($value, $field, $message) | Метод для проверки на пустоту |
| validateAddDirection($name, $images, $information) | Проверка на приход данных при добавлении направлени |
| addDirection($name, $images, $information) | Метод для добавления направления |
| validateEditDirection($id\_direction, $name, $information, $images) | Метод для проверки на приход данных при редактировании данных о направлении |
| editDirection($id\_direction, $name, $information, $images) | Метод для редактирования данных |
| getDirectionsBy4 () | Метод для вывода направлений и информации о них, на странице направлений по 4 в каждой строке |
| Класс: Specialization.php | |
| getSpecializations | Метод для вывода всех специальностей |
| getSpecialization ($id\_specialization) | Метод для вывода всей информации о специальности по id |
| deleteSpecil($id\_specialization) | Метод для удаления специальности по id |
| checkEmpty($value, $field, $message) | Метод для проверки на пустоту |
| validateEditSpecialization($id\_specil, $name, $direction) | Метод для проверки на приход данных при редактировании |
| editSpecialization($id\_specil, $name, $direction) | Метод для редактирования данных |
| validateAddSpecialization($name, $direction) | Метод для проверки на приход данных при добавлении специальности |
| addSpecialization($name, $direction) | Метод для добавления специальности |

Логическая структура представлена на рисунке 2.1.1.

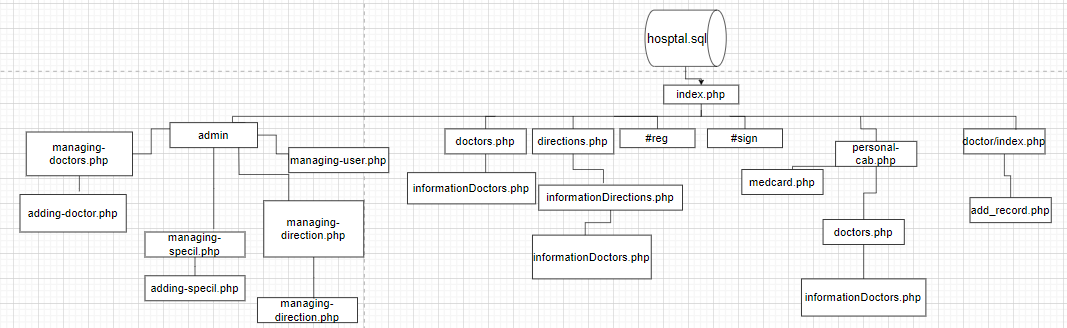


Рисунок 2.1.1- Логическая структура

Физическая структура представлена на рисунке 2.1.2.

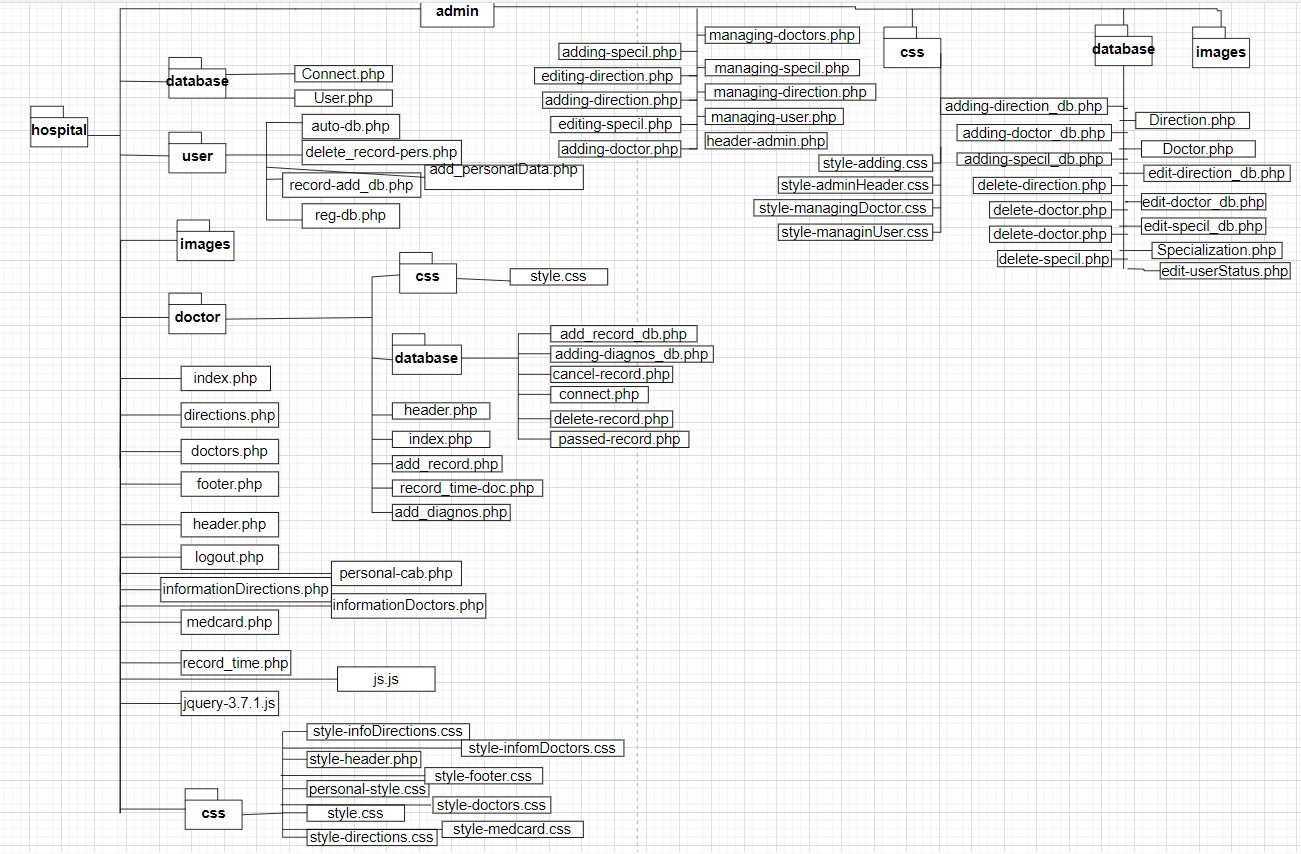


Рисунок 2.1.2- Физическая структура

2.2 Протокол тестирования программного продукта

Тестирование авторизации на корректных данных контрольного примера представлено в таблице 2.2.1.

Таблица 2.2.1 – Тестирование авторизации на корректных данных контрольного примера

|  |  |
| --- | --- |
| Наименование | Описание |
| Дата тестирования | 10.06.2024 |
| Приоритет тестирования *(Малый/Средний/высокий)* | Высокий. |
| Названиетестирования/ Имя | Проверка формы авторизации с правильной почтой и паролем. |
| Резюме испытания | Проверить программу при вводе корректных данных. |
| Этапы теста | 1. В форму ввели корректные данные; 2. Нажали на кнопку «Войти»; 3. На странице высветилось сообщение «Вы вошли как доктор»; 4. Перешли на страницу доктора. |
| Тестовые данные | Входные данные:  Почта: [13mir@mail.ru](mailto:13mir@mail.ru)  Пароль:123 |
| Ожидаемый результат | При выполнении тестирования, если данные корректны, то авторизированный пользователь переходит на страницу. На главной странице высвечивается уведомление об успехе. |
| Фактический результат | При выполнении тестирования пользователь успешно авторизовался.  На главной странице высвечивается уведомление «Вы вошли как доктор»(рисунок 2.2.1). |

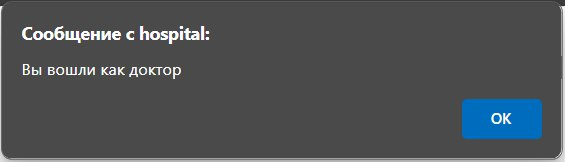


Рисунок 2.2.1- Уведомление об успешной авторизации

Тестирование авторизации на не корректные данные представлено в таблице 2.2.2.

Таблица 2.2.2 – Тестирование авторизации на не корректные данные

|  |  |
| --- | --- |
| Наименование | Описание |
| Дата тестирования | 10.06.2024 |
| Приоритет тестирования (Малый/Средний/высокий) | Высокий. |
| Название тестирования/ Имя | Проверка формы авторизации с неправильной почтой и паролем. |
| Резюме испытания | Проверить программу при вводе не корректных данных. |
| Этапы теста | 1. В форму ввели не корректные данные; 2. Нажали на кнопку «Войти»; 3. Пришло уведомление «Такой пользователь не найден»; 4. Перенаправление на главную страницу. |
| Тестовые данные | Входные данные:  Почта: [Masha06@mail.ru](mailto:13mir@mail.ru)  Пароль:Masha06 |
| Ожидаемый результат | При выполнении тестирования, если данные не корректны, то пользователь не может войти в личный кабинет .На главной странице высвечивается уведомление об ошибке. |
| Фактический результат | При выполнении тестирования пользователь увидел уведомление об ошибке «Такой пользователь не найден!»(рисунок 2.2.2), остался на главной странице |

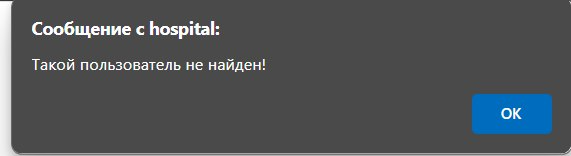


Рисунок 2.2.2 – Уведомление об ошибке авторизации

Тестирование регистрации на корректных данных контрольного примера представлено в таблице 2.2.3.

Таблица 2.2.3 – Тестирование регистрации на корректных данных контрольного примера

|  |  |
| --- | --- |
| Наименование | Описание |
| Дата тестирования | 10.06.2024 |
| Приоритет тестирования (Малый/Средний/высокий) | Высокий. |
| Название тестирования/ Имя | Проверка формы регистрации при введении корректных данных |
| Резюме испытания | Проверить программу при вводе корректных данных. |
| Этапы теста | 1. Ввели в форму корректные данные; 2. Нажать на кнопку «Прикрепиться»; 3. Если в базе данных нет пользователя с такой почтой, то появляется уведомление «Вы успешно прикрепились»; 4. Переход на главную страницу для авторизации. |
| Тестовые данные | Входные данные:  Почта: [Marcina@mail.ru](mailto:13mir@mail.ru)  Фамилия:Маркина  Имя:Мария  Отчество:Игоревна  Полис:1234669056784536  Дата рождения:13.02.2003  Город:Уфа  Паспорт:8019708934  Пароль:marcina13 |
| Ожидаемый результат | При выполнении тестирования, если данные корректны, то пользователь прикрепляется к больнице .На главной странице высвечивается уведомление об успешном прикреплении «Вы успешно прикрепились». |
| Фактический результат | При выполнении тестирования пользователь увидел уведомление о прикреплении к больнице «Вы успешно прикрепились»(рисунок 2.2.3). |

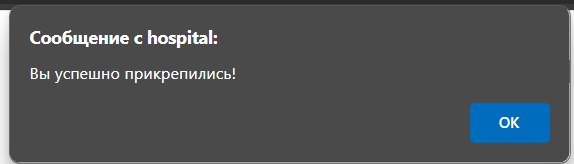


Рисунок 2.2.3 –Уведомление об успешном прикреплении к больнице

Тестирование регистрации на не корректные данные представлено в таблице 2.2.4.

Таблица 2.2.4 – Тестирование регистрации на не корректные данные

|  |  |
| --- | --- |
| Наименование | Описание |
| Дата тестирования | 10.06.2024 |
| Приоритет тестирования (Малый/Средний/высокий)*)* | Высокий. |
| Название тестирования/ Имя | Проверка формы регистрации при введении не корректных данных |
| Резюме испытания | Проверить программу при вводе не корректных данных. |
| Этапы теста | 1.Ввели в форму не корректные данные;  2.Нажать на кнопку «Прикрепиться»;  3.Если в базе данных есть пользователь с такой почтой, то появляется уведомление «Ошибка прикрепления!»;  4.Переход на главную страницу для повторного прикрепления. |
| Тестовые данные | Входные данные:  Почта: [Marcina@mail.ru](mailto:13mir@mail.ru)  Фамилия:Егорова  Имя:Елизавета  Отчество:Игоревна  Полис:5792346690845368  Дата рождения:31.08.2000  Город:Уфа  Паспорт:8087708934  Пароль:EgorovaLiza31 |
| Ожидаемый результат | При выполнении тестирования, если в базе данных уже существует пользователь с такой почтой то высвечивается уведомление «Ошибка прикрепления».Пользователя перенаправляет на главную страницу. |
| Фактический результат | При выполнении тестирования пользователь увидел уведомление об ошибке прикрепления.«Ошибка прикрепления»(рисунок 2.2.4). Пользователя перенаправляет на главную страницу. |

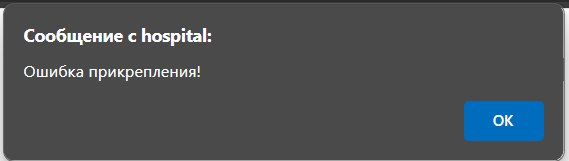


Рисунок 2.2.4 – Уведомление об ошибке прикрепления

Тестирование функции добавления врача на корректных данных контрольного примера представлено в таблице 2.2.5.

Таблица 2.2.5 – Тестирование функции добавления врача на корректных данных контрольного примера

|  |  |
| --- | --- |
| Наименование | Описание |
| Дата тестирования | 10.06.2024 |
| Приоритет тестирования *(Малый/Средний/высокий)* | Высокий. |
| Название тестирования/ Имя | Проверка добавления доктора с корректными данными |
| Резюме испытания | Проверить программу при добавлении доктора с корректными данными |
| Этапы теста | 1. Заполнить форму; 2. Нажать на кнопку «добавить»; 3. Вывод уведомления «Доктор добавлен»; 4. Перенаправление на страницу всех докторов. |
| Тестовые данные | Входные данные:  Почта: Mihail@mail.ru  Фамилия:Иванов  Имя:Михаил  Отчество:Дмитриевич  Специальность:Кардиолог  Пароль:Mihail123  Изображение: preview 2.png |
| Ожидаемый результат | При выполнении тестирования, если все поля заполнены и в базе данных нет врача с такой почтой то добавить в базу данных |
| Фактический результат | При выполнении тестирования пользователь увидел уведомление об успешном добавлении «Доктор добавлен»(рисунок 2.2.5) |

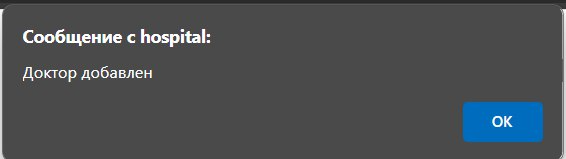


Рисунок 2.2.5 – Уведомление об успешном добавлении врача

Тестирование функции записи на прием на корректных данных представлено в таблице 2.2.6.

Таблица 2.2.6 – Тестирование функции записи на прием на корректных данных.

|  |  |
| --- | --- |
| Наименование | Описание |
| Дата тестирования | 12.06.2024 |
| Приоритет тестирования *(Малый/Средний/высокий)* | Высокий. |
| Названиетестирования/ Имя | Проверка на запись на прием |
| Резюме испытания | Проверить программу при записи на прием |
| Этапы теста | 1.Выбрать направление;  2. Выбрать врача;   1. Выбрать дату для записи; 2. Выбрать свободное время 3. Нажать на кнопку «Записаться»; 4. Если пользователь записался , появляется уведомление о записи и на почту приходит сообщение о времени и дате записи на прием. |
| Тестовые данные | Входные данные:  Дата:14-06-2024  Время: 8:00 |
| Ожидаемый результат | При выполнении тестирования, если время не занято то должна добавиться запись в базу данных и появится уведомление о записи, после записи приход сообщения на почту. |
| Фактический результат | При выполнении тестирования пользователь увидел уведомление об успешной записи «Пациент записан на прием, письмо о записи можно посмотреть на почте» (рисунок 2.2.6.1). и пришло письмо на почту(рисунок 2.2.6.2) |

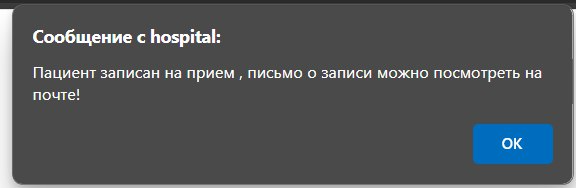


Рисунок 2.2.6.1 - Уведомление об успешной записи на прием

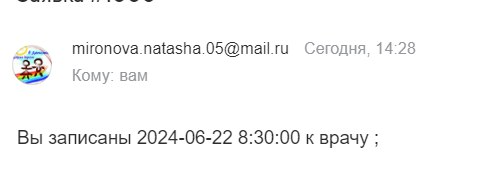


Рисунок 2.2.6.2- Сообщение о записи на прием

2.3 Руководство пользователя

Для того чтобы пользователю зайти на сайт нужно запустить Open Server Panel.php и в адресной строке ввести ссылку на сайт hospital. После открытия сайта пользователь попадает на главную страницу.

На главной странице пользователь может посмотреть новости, врачей, направления, изображение больницы и контактную информацию. Так же пользователь может воспользоваться навигационным меню: врачи, направления, главная, контакты, войти (рисунок 2.3.1).

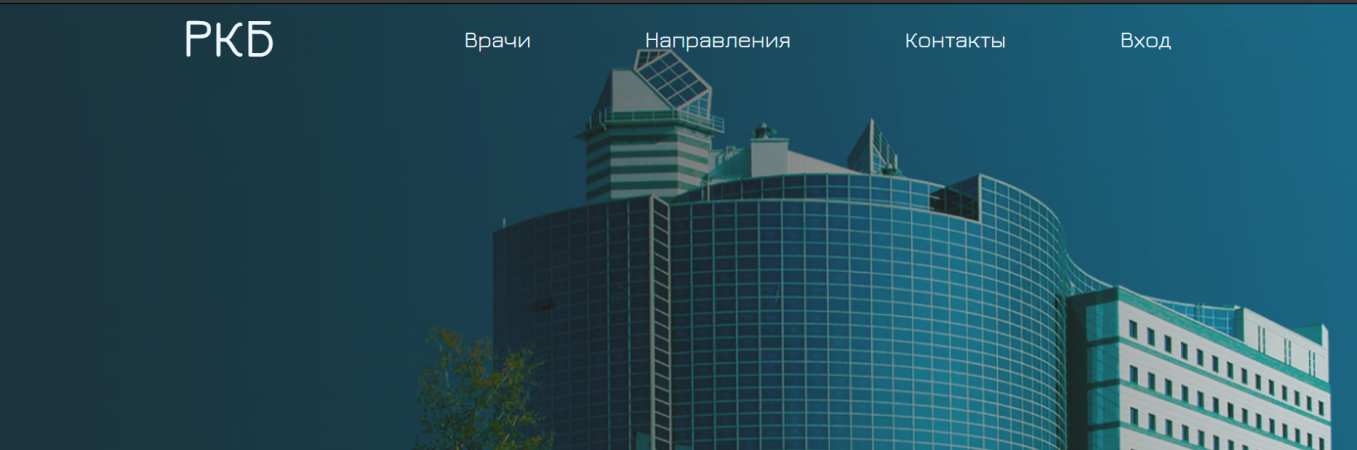


Рисунок 2.3.1 - Навигационное меню главной страницы

Пользователь может авторизоваться и прикрепиться к больнице. В навигационном меню нажав на кнопку «Вход» появляется модальное окно для авторизации (рисунок 2.3.2).

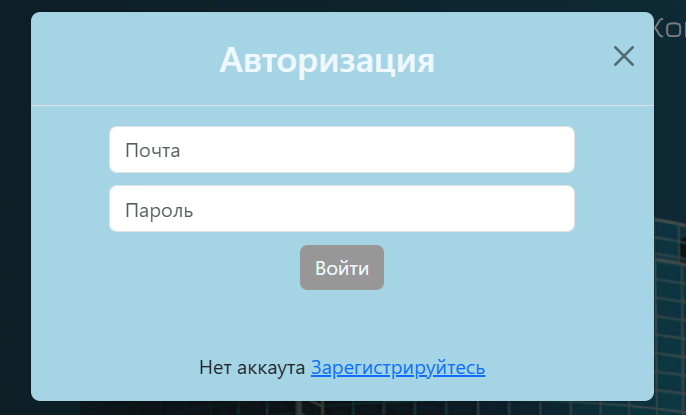


Рисунок 2.3.2 - Форма для авторизации

Если пользователя еще нет в системе он может зарегистрироваться, нажав на ссылку «Зарегистрироваться» откроется модальное окно для регистрации (рисунок 2.3.3).

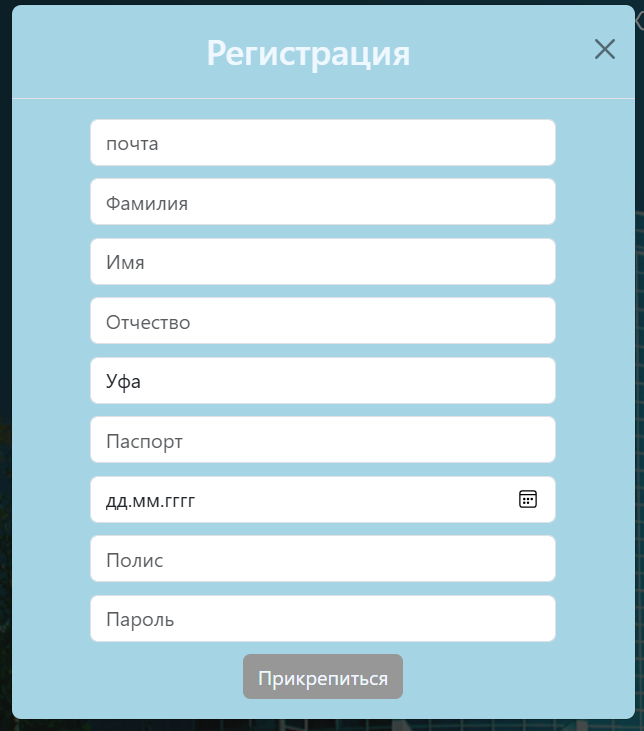


Рисунок 2.3.3 - Модальное окно для регистрации

После успешной авторизации пользователь попадает в личный кабинет в котором есть его данные, история записей, кнопка для перехода к электронной медицинской карте и кнопка для записи на прием(рисунок 2.3.4).

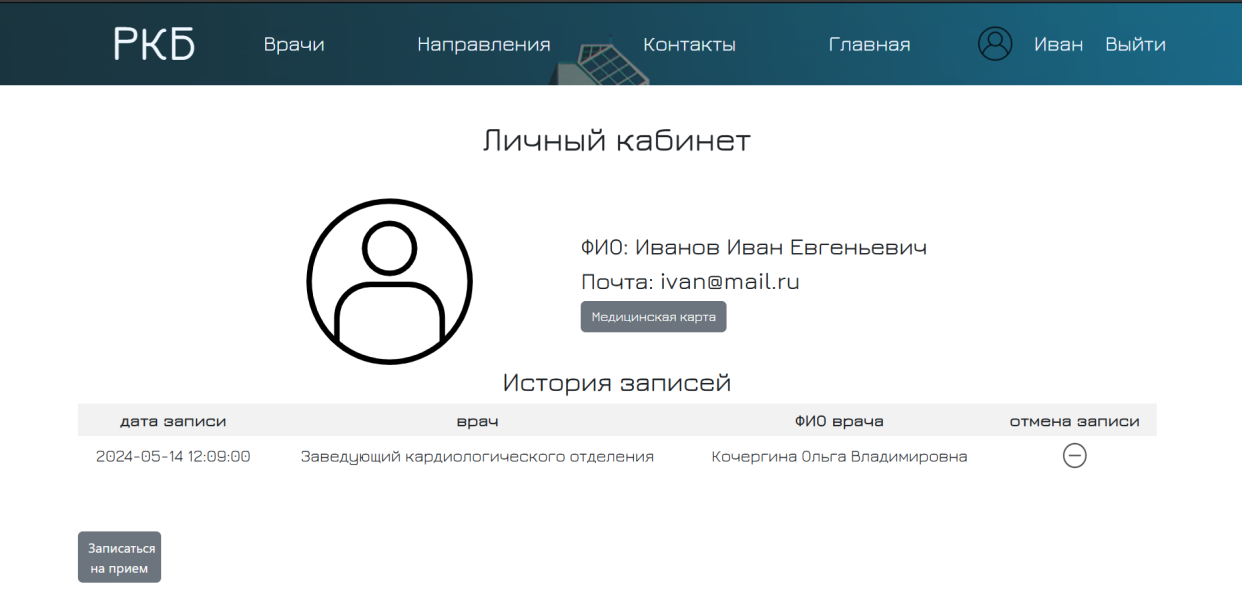


Рисунок 2.3.4 -Личный кабинет

Нажав на кнопку «Медицинская карта» пользователя перенаправляет на страницу электронной медицинской карты, где видна вся информация о пользователе с возможность редактирования данных, историю посещённых записей с записями врача и диагнозом (рисунок 2.3.5).

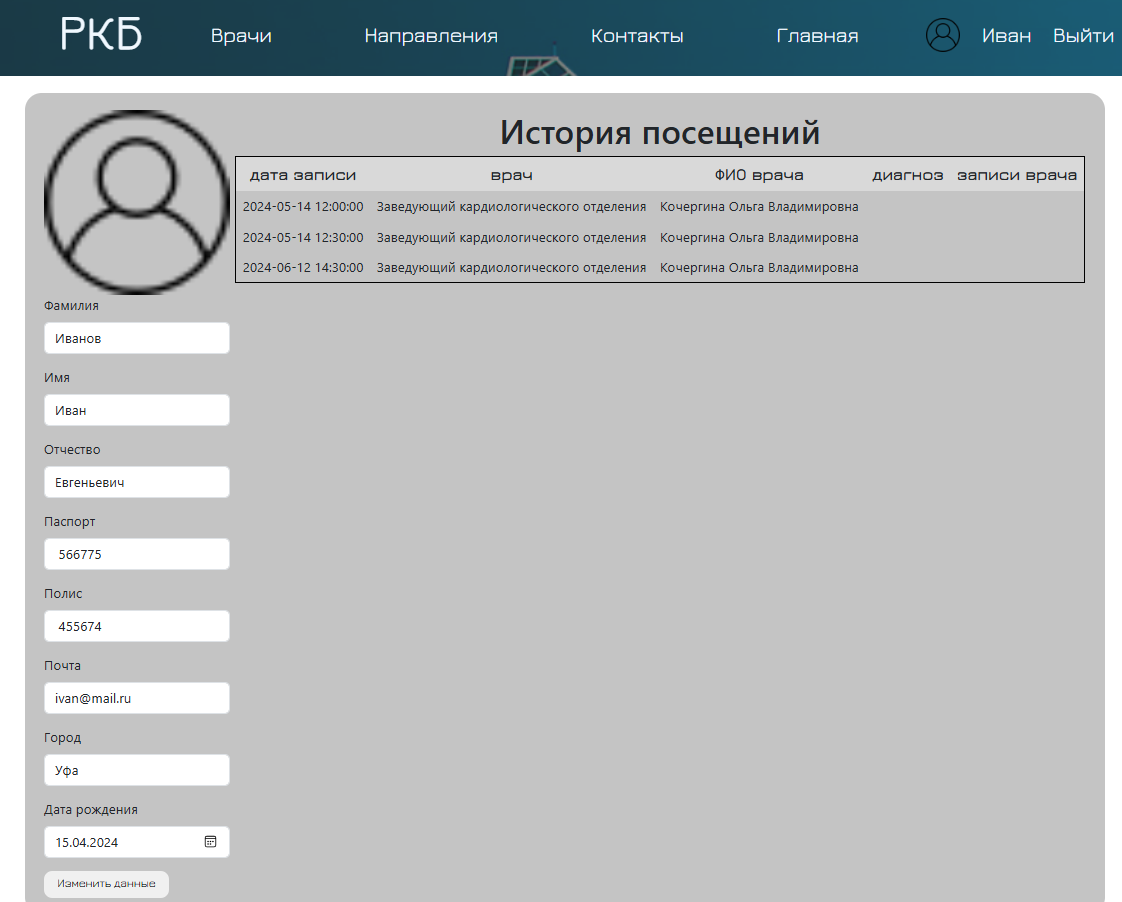


Рисунок 2.3.5 - Электронная медицинская карта.

Для того чтобы посмотреть доступные направления больницы необходимо нажать в навигационном меню на ссылку «Направления». После этого происходит переход на страницу направлений где пользователь может посмотреть название направления и фотографию(рисунок 2.3.6).

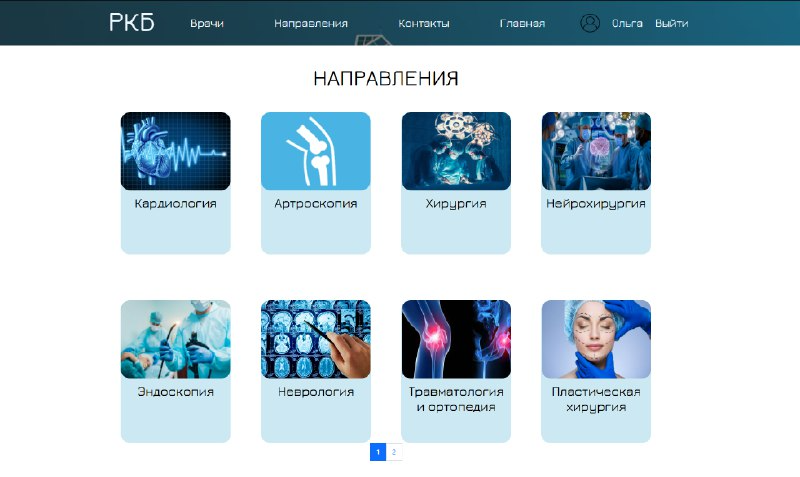


Рисунок 2.3.6 – Страница с направлениями больницы.

Выбрав нужное направление можно подробнее ознакомится с ним, нажав на картинку направления происходит переход на страницу где есть описание направления, фотография и врачи относящиеся к этому направлению(рисунок 2.3.7).

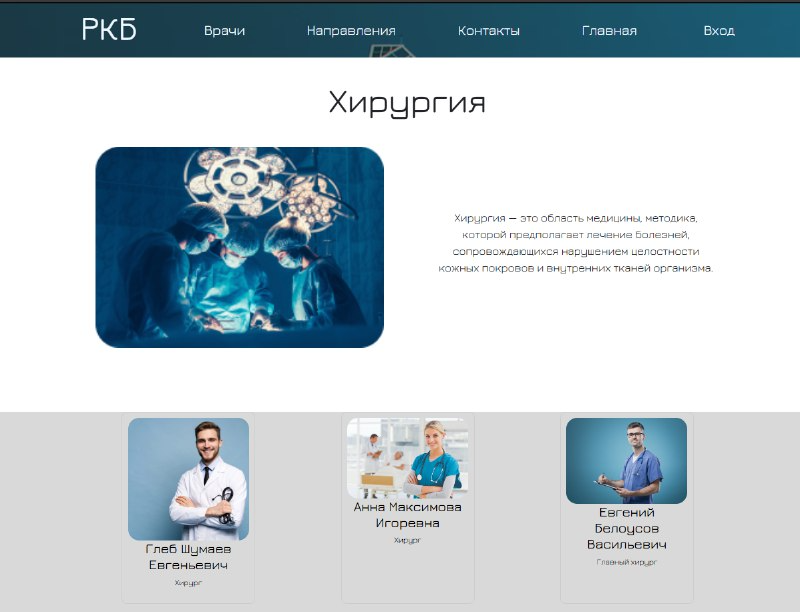


Рисунок 2.3.7 – Станица с подробным описанием направления.

При нажатии на картинку врача пользователь переходит на страницу где может выбрать дату и свободное время для записи(рисунок 2.3.8).

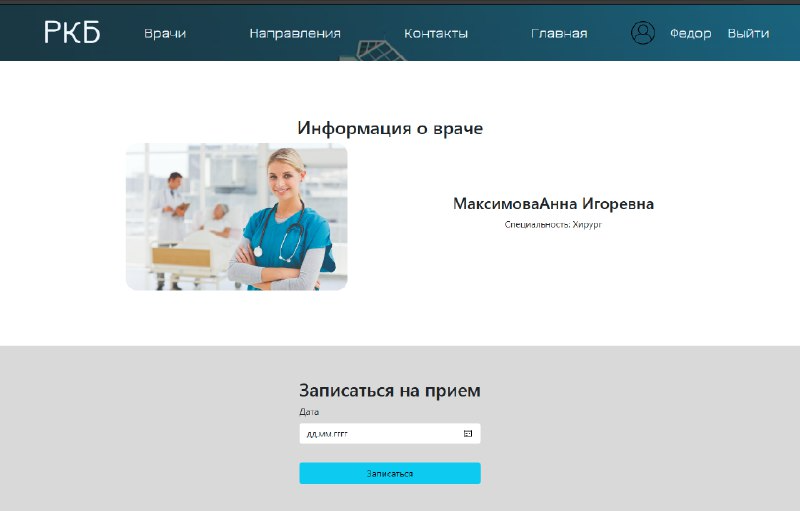


Рисунок 2.3.8 – Страница для записи на прием.

Для того чтобы посмотреть всех врачей в навигационном меню нужно нажать на ссылку врачи и пользователь перейдет на страницу с врачей(рисунок 2.3.9)

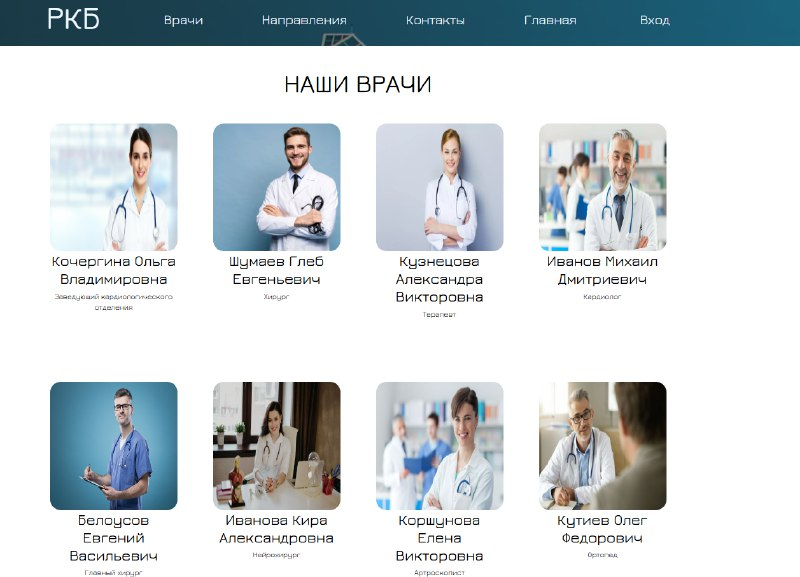


Рисунок 2.3.9 – Страница с врачами

Также на сайте есть страницы для администратора. Авторизовавшись администратор попадает на страницу с выводом всех направлений(рисунок 2.3.10).

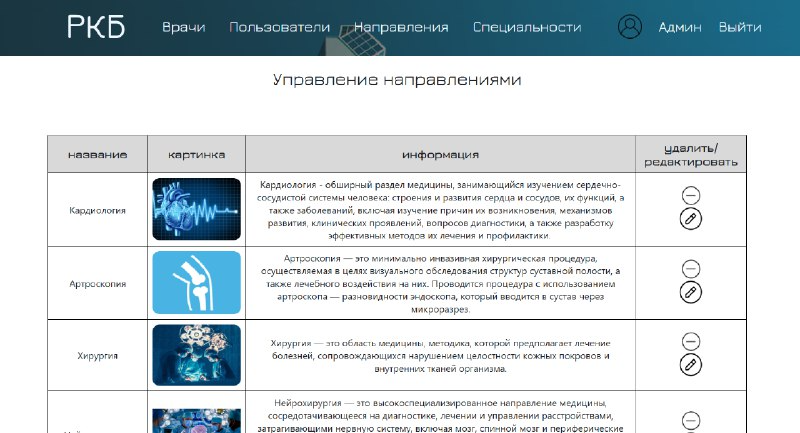


Рисунок 2.3.10 – Вывод всех направлений.

Каждое направление он может удалить либо редактировать, нажав на соответствующую иконку. Нажав на редактирование открывается страница для редактирования(рисунок 2.3.11 ).

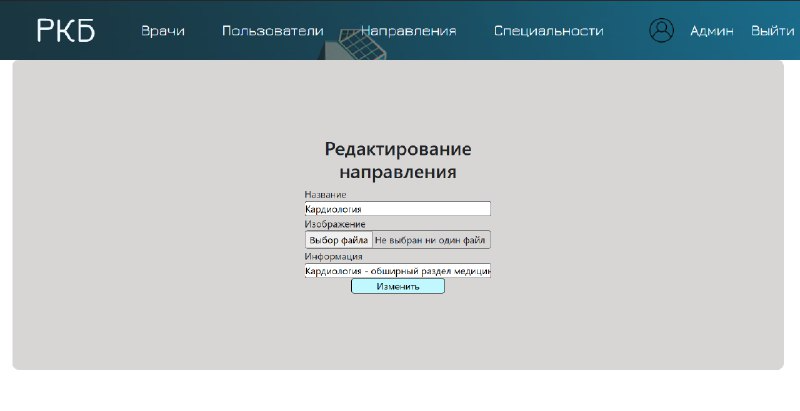


Рисунок 2.3.11 – Редактирование направления.

Также есть вывод всех докторов(рисунок 2.3.12), специальностей(рисунок 2.3.13) и пользователей(рисунок 2.3.14).

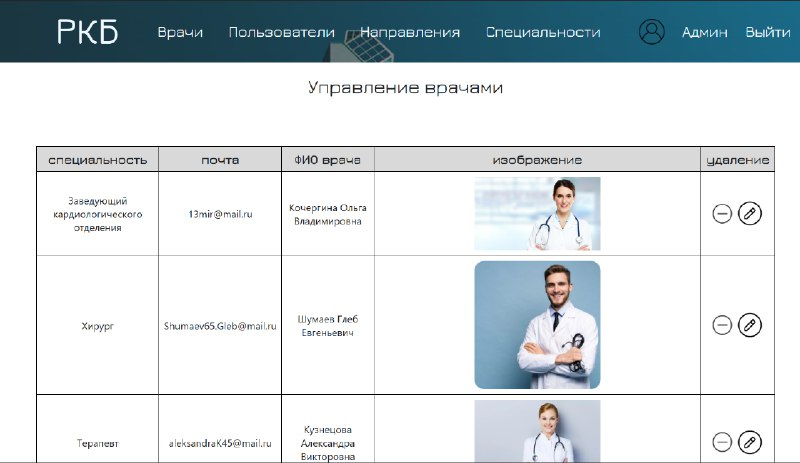


Рисунок 2.3.12 – Вывод врачей.



Рисунок 2.3.13 – Вывод всех специальностей.

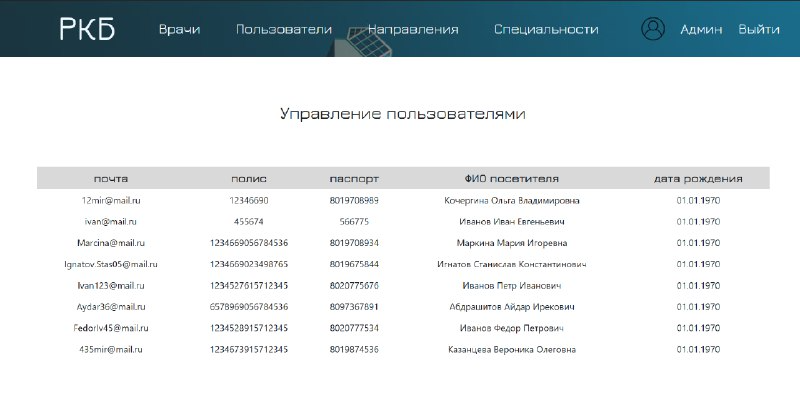


Рисунок 2.3.14 – Вывод всехпользователей.

Администратор может добавить направление(рисунок 2.3.15), а так же специальность, прикрепив ее к определенному направлению(рисунок 2.3.16).

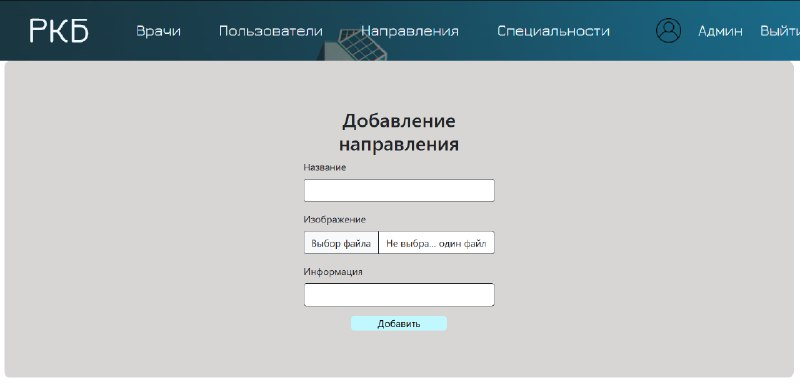


Рисунок 2.3.15- Добавление направления

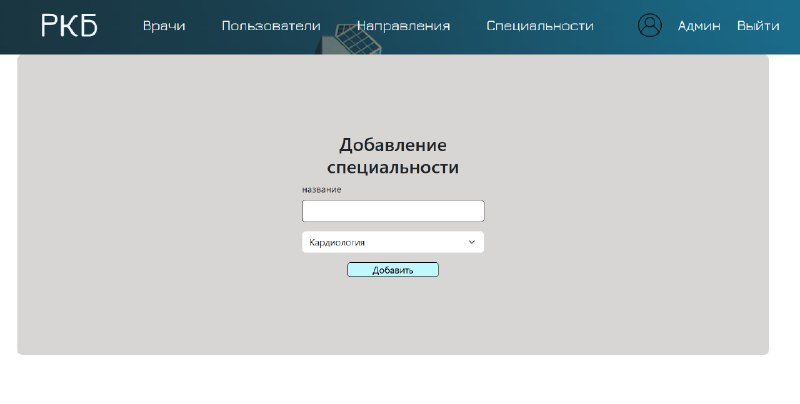


Рисунок 2.3.16 – Добавление специальности

После добавления специальности можно добавить врача на эту специальность (рисунок 2.3.17).

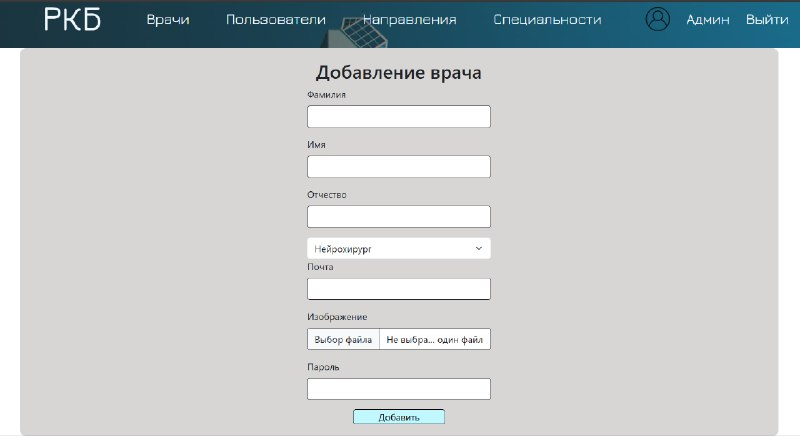


Рисунок 2.3.17 – Добавление врача

Так же есть страницы для врачей. Авторизовавшись врач переходит на страницу где видны все его записи , с возможностью просмотра и редактирования статуса записи(рисунок 2.3.18).

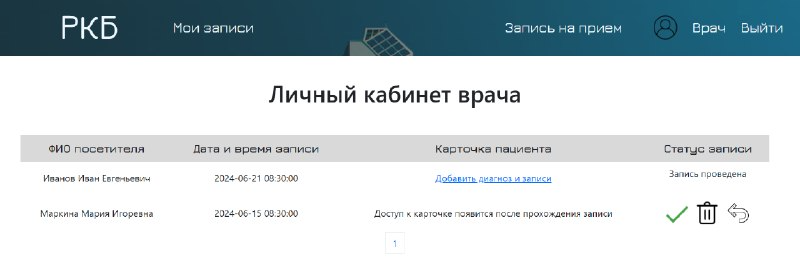


Рисунок 2.3.18 – Личный кабинет врача с записями

После того как пациент пришел на прием статус записи меняется на «Пройдено» и врач может внести в карточку диагноз и записи(рисунок 2.3.19).

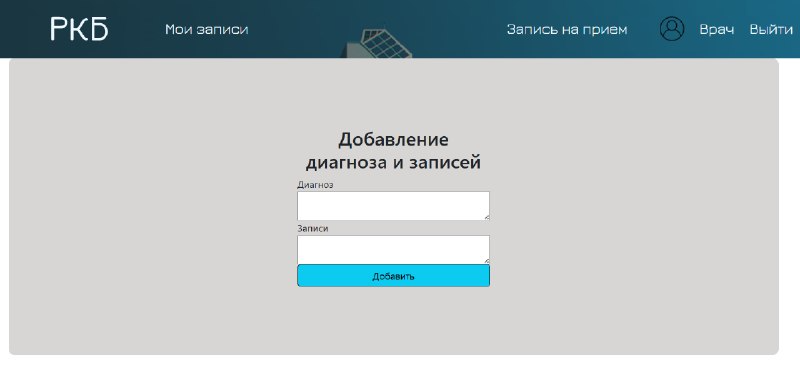


Рисунок 2.3.19 – Добавление диагноза

Для того чтобы записать пациента на прием, в навигационном меню нажав на ссылку «Запись на прием» доктор переходит на страницу где может выбрать пациента, дату и время записи(рисунок 2.3.20)

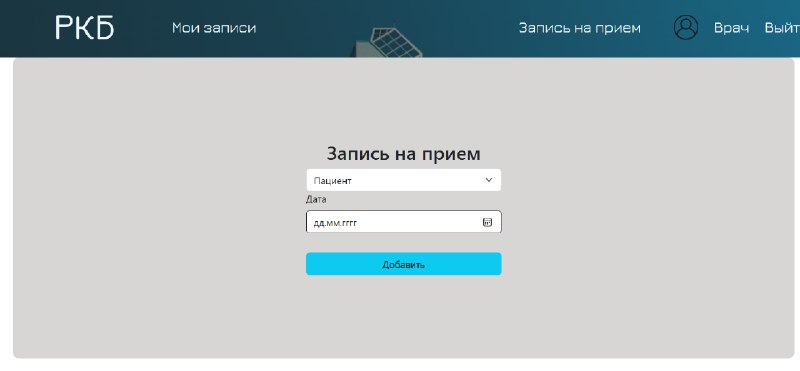


Рисунок 2.3.20 – Запись пациента на прием

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы была разработана АИС «Больница» для просмотра информации о больнице, врачах, направления, с функцией записи на прием. Результатом работы стало создание программного продукта написанного на PHP, используя фреймворк Bootstrap и с использованием СУБД MySQL.

Используя данные из контрольного примера, была проведена проверка программы на основе которой были написаны тест-кейсы. Так же написано подробное руководство пользователя для отображения всех функций сайта.

СПИСОК СОКРАЩЕНИЙ

РКБ –Республиканская Клиническая Больница

АИС – автоматизированная информационная система

СПИСОК ЛИТЕРАТУРЫ

1. metanit.com <https://metanit.com/php/tutorial/3.1.php>
2. Википедия [Электронный ресурс] // Свободная энциклопедия. – Режим доступа: http://ru.wikipedia.org/wiki/, свободный.
3. Гагарина, Л.Г., Разработка и эксплуатация автоматизированных информационных систем [Текст]: учебное пособие/ В.Д. Киселев, Е.Л. Федотова - М.: ИД «Форум»: ИНФРА-М, 2012. – 384 с.

Приложение А

Контрольный пример

Таблица А.1 –Контрольный пример входных данных «Пользователь»

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Фамилия | Имя | Отчество | Почта | Полис | Дата рождения | Паспорт | Город | Пароль |
| Иванов | Петр | Иванович | Ivan123@mail.ru | 1234527615712345 | 13.08.1987 | 8020775676 | Уфа | 123 |
| Игнатов | Станислав | Олегович | Ignatov.Stas05@mail.ru | 1234669023498765 | 16.07.2004 | 8019675844 | Уфа | stas16 |
| Маркина | Мария | Игоревна | [Marcina@mail.ru](mailto:13mir@mail.ru) | 1234669056784536 | 13.02.2003 | 8019708934 | Уфа | marcina13 |
| Абдрашитов | Айдар | Ирикович | Aydar36@mail.ru | 6578969056784536 | 04.09.2000 | 8097367891 | Уфа | Aydar36 |
| Иванов | Федор | Петрович | FedorIv45@mail.ru | 1234528915712345 | 13.12.1980 | 8020777534 | Уфа | 1234 |

Таблица А2 - Контрольный пример входных данных «Врачи»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Фамилия | Имя | Отчество | Почта | Пароль | Специальность | Изображение |
| Иванов | Михаил | Дмитриевич | Mihail@mail.ru | Mihail123 | Кардиолог | preview 2.png |
| Кочергина | Ольга | Владимировна | 13mir@mail.ru | Olga123 | Заведующий кардиологического отделения | 3-рисунок 1.png |
| Кузнецова | Александра | Викторовна | aleksandraK45@mail.ru | Sasha45 | Терапевт | 2-рисунок 1.png |
| Шумаев | Глеб | Евгеньевич | Shumaev65.Gleb@mail.ru | Gleb568// | Хирург | Image 3.png |
| Белоусов | Евгений | Васильевич | Beloysov.45@mail.ru | welcome | Главный хирург | 12.jpg |

Продолжение приложения А

Таблица А3 - Контрольный пример входных данных «Специальность»

|  |  |
| --- | --- |
| Название | Направление |
| Хирург | Хирургия |
| Кардиолог | Кардиология |
| Заведующий Кардиологическим отделением | Кардиология |
| Главный хирург | Хирургия |
| Артроскопист | Артроскопия |

Таблица А4 - Контрольный пример входных данных «Направление»

|  |  |  |
| --- | --- | --- |
| Название | Описание | Изображение |
| Хирургия | Хирургия - это область медицины, методика, которой предполагает лечение болезней, сопровождающихся нарушением целостности кожных покровов и внутренних тканей организма. | preview 1.png |
| Кардиология | Кардиология - обширный раздел [медицины](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B4%D0%B8%D1%86%D0%B8%D0%BD%D0%B0" \o "Медицина), занимающийся изучением [сердечно-сосудистой системы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B4%D0%B5%D1%87%D0%BD%D0%BE-%D1%81%D0%BE%D1%81%D1%83%D0%B4%D0%B8%D1%81%D1%82%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0" \o "Сердечно-сосудистая система) человека: строения и развития [сердца](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B4%D1%86%D0%B5_%D1%87%D0%B5%D0%BB%D0%BE%D0%B2%D0%B5%D0%BA%D0%B0" \o "Сердце человека) и [сосудов](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D0%BE%D1%81%D0%BD%D1%8B%D0%B5_%D1%81%D0%BE%D1%81%D1%83%D0%B4%D1%8B" \o "Кровеносные сосуды), их функций, а также заболеваний, включая изучение причин их возникновения, механизмов развития, клинических проявлений, вопросов диагностики, а также разработку эффективных методов их лечения и профилактики. | image 2.png |
| Артроскопия | Артроскопия -это минимально инвазивная хирургическая процедура, осуществляемая в целях визуального обследования структур суставной полости, а также лечебного воздействия на них. Проводится процедура с использованием артроскопа — разновидности эндоскопа, который вводится в сустав через микроразрез. | Artroskopiya 1.png |
| Офтальмология | Офтальмология -это область медицины, которая изучает глаз, его анатомию, физиологию и болезни, а также разрабатывает методы лечения и профилактики глазных болезней. | 1.jpg |
| Пластическая хирургия | Пластическая хирургия — это раздел хирургии, занимающийся оперативными вмешательствами, направленными на устранение деформаций и дефектов какого-либо органа, ткани или поверхности человеческого тела. | plastic.jpg |

Таблица А4 - Контрольный пример выходных данных «Электронная медицинская карта»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дата записи | Врач | ФИО врача | Диагноз | Записи врача |
| 20-06-2024 08:00 | Терапевт | Кузнецова Александра Викторовна | ОРВИ | Пропить курс витаминов |
| 21-06-2024 08:30 | Ортопед | Кутиев Олег Федорович | Плоскостопие | Носить стельки |
| 21-06-2024 08:30 | Травмотолог | Камалов Тагир Михайлович | Перелом со смещением | Наложение гипса |
| 21-06-2024 10:30 | Кардиолог | Иванов Михаил Дмитриевич | Аритмия | Пройти обследование |
| 22-06-2024 08:30 | Терапевт | Файзуллина Земфира Ильдаровна | Отит | Обратитесь к отоларингологу |