Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»

Образовательная программа «Веб-технологии»

Отчет по курсовому проекту

по дисциплине «Основы инженерного проектирования»

Тема: «Электронная регистратура Рязанской области»

**Выполнил:**

Студент группы 191-321

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Огородников Н. А.

подпись, дата

**Принял:**

Старший преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Даньшина М.В.

подпись, дата

Москва 2020

Оглавление

[Оглавление 2](#_Toc44291138)

[Введение 3](#_Toc44291139)

[Анализ аналогов 4](#_Toc44291140)

[Проектирование 5](#_Toc44291141)

[Разработака и тестирование 6](#_Toc44291142)

[Разработка структуры базы данных 6](#_Toc44291143)

[Описание таблиц базы данных 8](#_Toc44291144)

[Типовые запросы в БД 11](#_Toc44291145)

[Программирование 14](#_Toc44291146)

[Дизайн 15](#_Toc44291147)

[Тестирование 16](#_Toc44291148)

[Заключение 17](#_Toc44291149)

[Список используемых источников 18](#_Toc44291150)

Введение

Проект «Электронная регистратура Рязанской области» направлен на демонстрацию и фиксацию навыков, полученных в течение первых двух семестров обучения.

Процесс работы над проектом представляет собой Full-Stack разработку типового решения в области электронных записей на получение услуг.

Результатом проекта является функционирующий сайт записи к врачу с интеграцией базы данных.

Анализ аналогов

Электронные регистратуры широко распространены в наше время, поэтому у моего решения есть множество аналогов. Вот некоторые из них:

* [моездоровье62.рф](https://xn--62-dlchecl2bsdax2n.xn--p1ai/add_appointment/) – электронная регистратура Рязанской области
* [портал пациента 52](https://portal52.is-mis.ru/#!/clinics/) – электронная регистратура Нижегородской области
* [региональный портал медицинских услуг](https://k-vrachu.ru/)
* [единый портал здравоохранения Ставропольского края](https://zdrav26.ru/registry)
* [портал записи на прием к врачу в Ленинградской области](https://zdrav.lenreg.ru/)

Данные сайты имеют схожую структуру, и процесс записи к врачу на них сводится к четырем основным шагам:

1. Выбор клиники
2. Выбор услуги/специальности
3. Выбор даты и времени
4. Ввод личных данных и подтверждение записи

Опыт использования сайтов смешанный. Расположение элементов удобное и интуитивно понятное, но, с другой стороны, практически на всех сайтах отсутствует адаптивность. Дизайн не смотрится древним, однако передовым и современным назвать его нельзя.

Подводя итог можно сказать, что сайты являются типовыми решениями, и ничем не выделяются на общем фоне, однако выполняют свою функцию исправно, как и полагается порталам получения электронных услуг.

Проектирование

Проанализировав аналоги, я выделил следующие случаи использования разрабатываемого портала:

* Запись на прием в период самоизоляции
* Облегчение записи для людей с ограниченными возможностями
* Снижение нагрузки на регистратуры в клиниках
* Оптимизация очередей (каждый знает, когда его время)
* Увеличение количества клиентов путем облегчения получения услуги (некоторые люди не ходят к врачу, потому как не желают стоять в очередях; использование электронной записи уменьшает количество очередей)

Функционал сайта будет схож с другими типовыми решениями:

* Добавление записи в четыре шага (с выбором доступного времени на доступную дату)
* Удаление записи по номеру талона

Разработака и тестирование

Разработка структуры базы данных

Разработка базы данных началась с создания ER-диаграммы, удовлетворяющей функционалу, сформулированному на предыдущем этапе. В результате получилась следующая диаграмма:

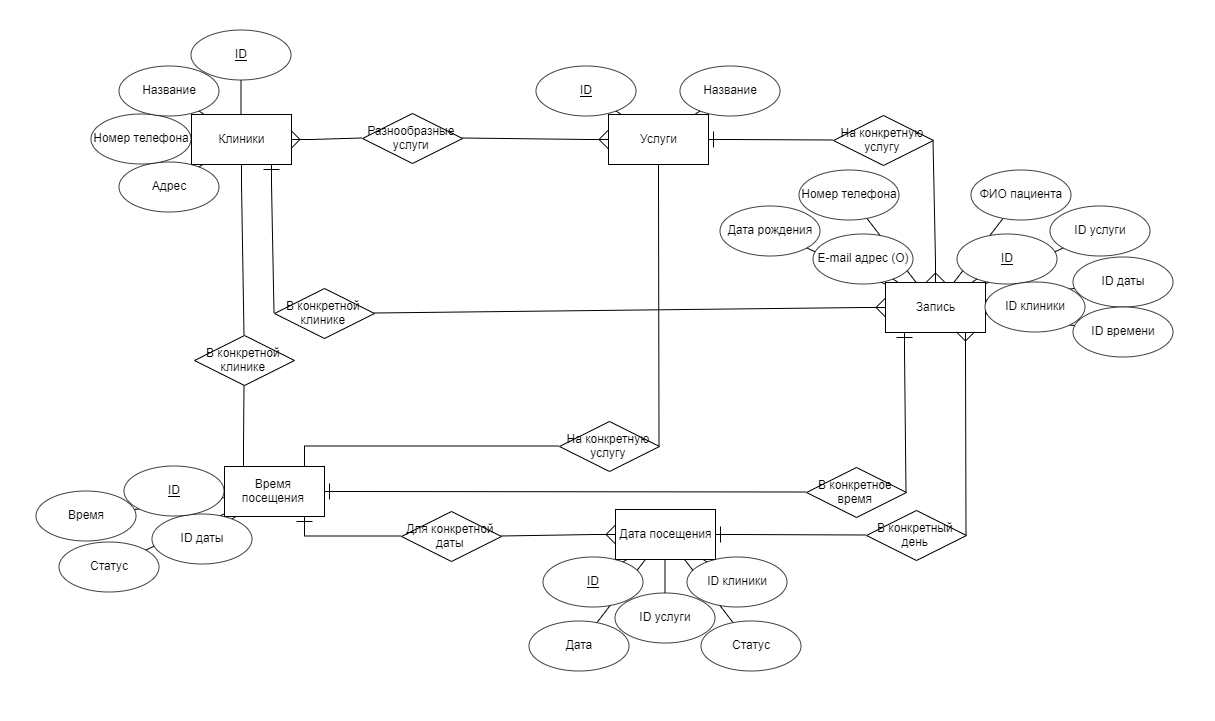


Рисунок 1, ER-диаграмма

Далее, опираясь на диаграмму, была создана схема будущей базы данных:

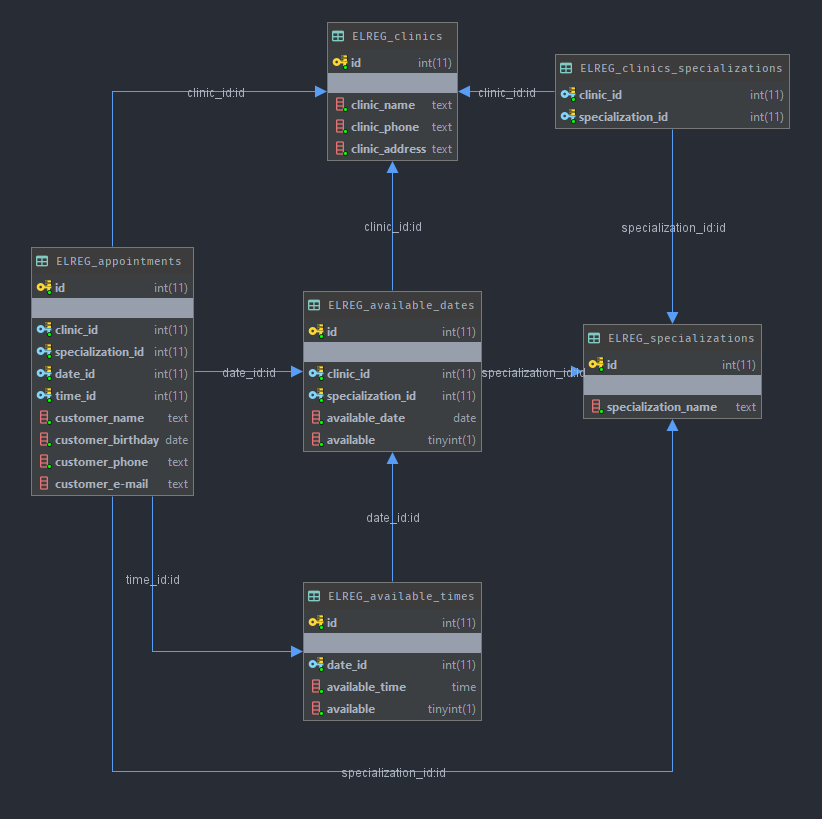


Рисунок 2, Схема БД

Описание таблиц базы данных

Список всех клиник с информацией о них содержит таблица ELREG\_clinics. Таблица представлена первичным ключом id и тремя столбцами с типом данных text, которые служат для вывода агрегирующей информации.

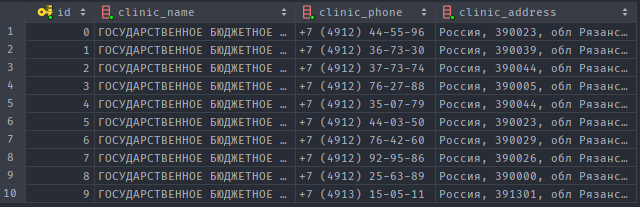


Рисунок 3, Таблица Клиники

Реестром всех услуг является таблица ELREG\_specializations. Имеет первичный ключ id и столбец specialization\_name, который содержит наименования услуг.

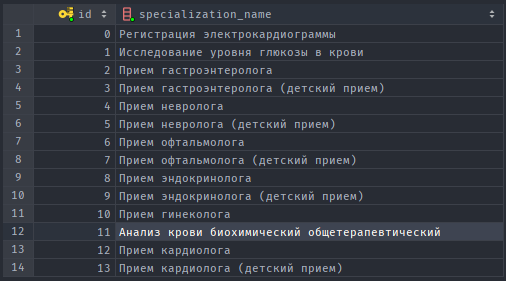


Рисунок 4, Таблица Услуги

У каждой клиники есть свой набор услуг. Разные клиники могу иметь одни и те же услуги. Следовательно, необходима таблица, в которой будет содержаться набор услуг для каждой клиники. Этой цели служит таблица ELREG\_clinics\_specializations. Имеет два внешних ключа, ссылающихся на ключи id из таблиц с клиниками и услугами. Таблица служит для вывода соответствующей информации на сайте.

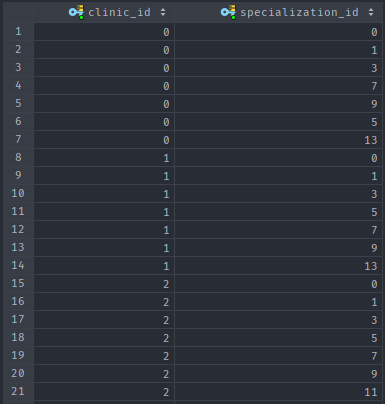


Рисунок 5, Таблица Услуги в клиниках

Далее идет таблица ELREG\_available\_dates. Она хранит в себе даты для каждой услуги в каждой клинике и их статус. Первичный ключ id, внешние ключи clinic\_id и specialization\_id ссылаются на соответствующие таблицы, available\_date хранит непосредственно дату, а ключ available хранит boolean величину, которая отражает статус даты (свободна/несвободна).

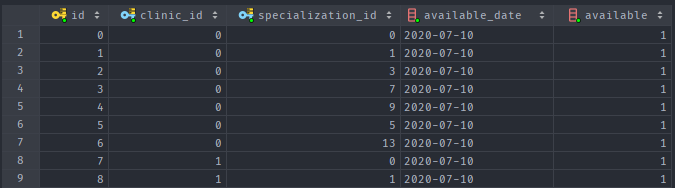


Рисунок 6, Таблица Даты

Поскольку для каждой даты существует свой набор времён приема, для хранения этой информации есть таблица ELREG\_available\_times. Хранит в себе первичный ключ id, вторичный ключ date\_id, отсылающий нас к таблице с датами. В available\_time хранится время, а ключ available работает так же, как и в таблице с датами.

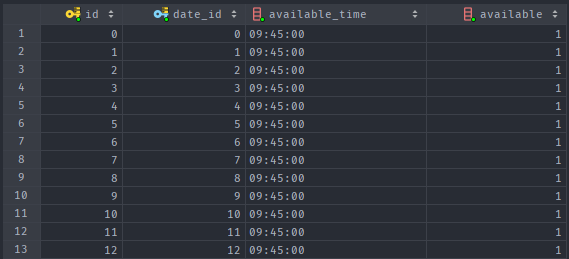


Рисунок 7, Таблица Время

В процессе записи на прием пользователь выбирает те или иные данные из предыдущих таблиц, а еще вводит свои. После подтверждения все эти данные уходят в таблицу ELREG\_appointments. Первичный ключ id, четыре внешних ключа: clinic\_id, specialization\_id, date\_id и time\_id из соответствующих таблиц представляют выбранную информацию, а ключи customer\_name, customer\_birthday, customer\_phone и customer\_e-mail (необязательный) представляют персональную информацию пользователя.

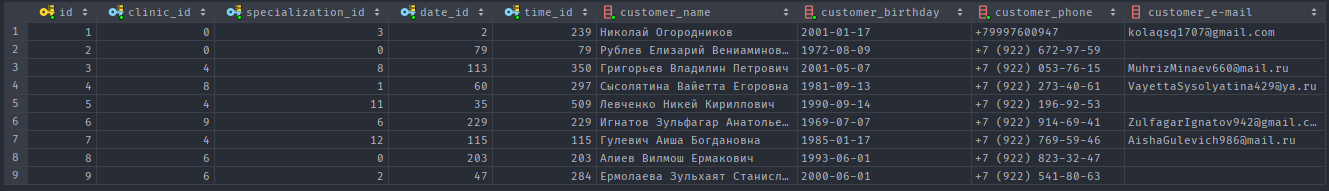


Рисунок 8, Таблица Талоны

Типовые запросы в БД

Запрос на вывод списка всех клиник:

*select \* from* ELREG\_clinics

Запрос на вывод всех услуг для выбранной клиники (здесь и далее «?» - величина, задаваемая php-кодом, зависящая от выбора пользователя):

*select* id, specialization\_name  
*from* ELREG\_specializations  
 *join* ELREG\_clinics\_specializations Ecs *on* ELREG\_specializations.id = Ecs.specialization\_id  
*where* clinic\_id = ?

Запрос на вывод выбранной услуги:

*select* specialization\_name  
*from* ELREG\_specializations  
*where* id = ?

Запрос на вывод свободных дат для выбранной услуги в выбранной клинике в формате «дд.мм.гггг»:

*select* id, *date\_format*(available\_date, '%d.%m.%Y') *as* available\_date  
*from* ELREG\_available\_dates  
*where* clinic\_id = ?  
 *and* specialization\_id = ?  
 *and* available = *true*

Запрос на вывод выбранной даты в формате «дд.мм.гггг»:

*select date\_format*(available\_date, '%d.%m.%Y') *as* available\_date  
*from* ELREG\_available\_dates  
*where* id = ?

Запрос на вывод свободного времени записи для выбранной даты в формате «чч:мм»:

*select* id, *date\_format*(available\_time, '%H:%i') *as* available\_time  
*from* ELREG\_available\_times  
*where* date\_id = ?  
 *and* available = *true*

Запрос на вывод выбранной клиники:

*select* clinic\_name  
*from* ELREG\_clinics  
*where* id = ?

Запрос на вывод выбранных даты и времени:

*select* available\_date, available\_time  
*from* ELREG\_available\_dates  
 *join* ELREG\_available\_times Eat *on* ELREG\_available\_dates.id = Eat.date\_id  
*where* Eat.id = ?  
 *and* ELREG\_available\_dates.id = ?

Запрос на вывод самого большого номера талона на данный момент:

*select max*(id)  
*from* ELREG\_appointments

Запрос на добавление талона с заданными параметрами:

*insert into* ELREG\_appointments  
(id, clinic\_id, specialization\_id, date\_id, time\_id,  
 customer\_name, customer\_birthday, customer\_phone, `customer\_e-mail`)  
*values* (?, ?, ?, ?, ?, ?, ?, ?, ?)

Запрос на обновление статуса выбранного времени (установить на «занято):

*update* ELREG\_available\_times  
*set* available = *false  
where* id = ?

Запрос на вывод оставшихся свободных дат в выбранный день:

*select \*  
from* ELREG\_available\_times  
*where* date\_id = ?  
 *and* available = *true*

Запрос на изменение статуса выбранного дня (установить на «занято»):

*update* ELREG\_available\_dates  
*set* available = *false  
where* id = ?

Запрос на вывод данных о талоне с указанными параметрами:

*select* ELREG\_appointments.id,  
 clinic\_name,  
 specialization\_name,  
 available\_date,  
 available\_time,  
 customer\_name,  
 customer\_birthday,  
 customer\_phone,  
 `customer\_e-mail`  
*from* ELREG\_appointments  
 *join* ELREG\_clinics Ec *on* ELREG\_appointments.clinic\_id = Ec.id  
 *join* ELREG\_specializations Es *on* ELREG\_appointments.specialization\_id = Es.id  
 *join* ELREG\_available\_dates Ead *on* Ec.id = Ead.clinic\_id  
 *join* ELREG\_available\_times Eat *on* Ead.id = Eat.date\_id  
*where* ELREG\_appointments.id = ?  
 *and* Ec.id = ?  
 *and* Es.id = ?  
 *and* Ead.id = ?  
 *and* Eat.id = ?

Запрос на вывод данных о выбранном талоне:

*select* ELREG\_appointments.id,  
 clinic\_name,  
 specialization\_name,  
 available\_date,  
 available\_time,  
 customer\_name,  
 customer\_birthday,  
 customer\_phone,  
 `customer\_e-mail`  
*from* ELREG\_appointments  
 *join* ELREG\_clinics Ec *on* ELREG\_appointments.clinic\_id = Ec.id  
 *join* ELREG\_specializations Es *on* ELREG\_appointments.specialization\_id = Es.id  
 *join* ELREG\_available\_dates Ead *on* Ec.id = Ead.clinic\_id  
 *join* ELREG\_available\_times Eat *on* Ead.id = Eat.date\_id  
*where* ELREG\_appointments.id = ?  
 *and* Ec.id = ELREG\_appointments.clinic\_id  
 *and* Es.id = ELREG\_appointments.specialization\_id  
 *and* Ead.id = ELREG\_appointments.date\_id  
 *and* Eat.id = time\_id

Запрос на удаление выбранного талона:

*delete  
from* ELREG\_appointments  
*where* id = ?

Запрос на изменение статуса выбранного дня (установить на «свободно»):

*update* ELREG\_available\_dates  
*set* available = *true  
where* id = ?

Запрос на обновление статуса выбранного времени (установить на «свободно):

*update* ELREG\_available\_times  
*set* available = *true  
where* id = ?

Программирование

Приложение работает по следующим принципам:

* Каждый шаг представляет из себя отдельный php файл
* В режиме добавления записи при переходе на следующий шаг данные, выбранные пользователем, передаются методом GET, включая данные с прошлых шагов
* При переходе на предыдущий шаг ему передаются данные, с которыми пользователь изначально пришел на него
* Данные, введенные пользователем, передаются методом POST

В приложении реализован CRUD. Сайт может:

* Создавать записи (талоны)
* Выводить информацию для чтения (клиники, услуги и тд.)
* Обновлять поля (изменение статуса времени и даты)
* Удалять записи (талоны)

В приложении реализована фильтрация, но пользователь не может на ею управлять. На шаге выбора даты и времени выводимая информация фильтруется по статусу. На экран будут выведены только свободные даты и время, а если таковых нет, то выводится сообщение об отсутствии талонов на данную услугу.

В приложении реализован поиск. В режиме удаления записи на первом шаге система предлагает пользователю ввести номер талона. После получения данных, система ищет талон с заданным номером и затем выводит информацию о данной записи.

Дизайн

Данное приложение имеет современный дизайн, выполненный в духе минимализма. При создании дизайна использовались самые современные решения в данной области:

* Использование Figma. Прототип дизайна приложения был разработан с помощью данного инструмента. И пусть итоговый вариант сайта выглядит иначе, он все равно продолжает и развивает идеи, заложенные в прототипе
* Использование иконок Material Design. Это иконки с мобильной платформы, обладают отличной узнаваемостью и помогают в создании интуитивно понятного пользовательского интерфейса
* Использование Microsoft Fluent Design. В данном приложении реализован элемент из новой идеологии дизайна компании Microsoft. Мне удалось скопировать эффект фона acrylic и удачно внедрить его. Данный эффект достигается за счет использования свойств background-color: rgba(0, 0, 0, .625) и backdrop-filter: blur(5px)
* Использование Animate.css и jQuery. С помощью данных инструментов удалось создать плавные анимации перехода между страницами
* Изменение ширины обводки элемента border\_width. В сочетании с hover и transition данная техника позволяет «вдохнуть жизнь» в статичные элементы
* Стилевое оформление checkbox. Прописать стили конкретно данному элементу не выйдет, но можно сделать свой checkbox. Это непросто, зато вместе со стилизованными полями ввода создает законченный внешний вид

Тестирование

Для данного приложения проводилось ручное тестирование с имитацией поведения обычного пользователя. Для этого было проведено добавление 10 записей на прием.

В ходе проведения тестирования были выявлены и исправлены следующие проблемы:

* Повторная отправка POST данных при перезагрузке финишной страницы
* Проблемы с навигацией
* Проблема с отображением даты без свободного времени
* Проблема, при которой на странице выбора даты и времени при отсутствии свободных дат ничего не выводилось
* Проблемы с некоторыми запросами в БД

По полученным результатам можно сделать вывод, что технология ручного тестирования в данном случае показала себя с наилучшей стороны и помогла выявить ряд фатальных проблем продукта.

Заключение

В результате процесса разработки получилось то, что и изначально планировалось: типовое решение в области электронной записи на получение услуг.

Также в процессе создания данного приложения я применил знания и навыки, полученные за первый курс обучения, освежил их и даже в некоторой степени развил.

Полноценная Full-Stack разработка была новым и интересным опытом в моей практике, и я намерен дальше совершенствовать свои навыки в данном направлении.

Список используемых источников

1. [Мануал по PHP](https://www.php.net/manual/en/index.php)
2. [Мануал по PHP Data Objects](https://www.php.net/manual/en/book.pdo.php)
3. [Мануал по MySQL](https://dev.mysql.com/doc/refman/8.0/en/)
4. [Документация jQuery](https://api.jquery.com/)
5. [Документация Animate.css](https://animate.style/#documentation)
6. [Stack Overflow](https://stackoverflow.com/)
7. [W3School](https://www.w3schools.com/)