|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Разработка интернет-приложений»

ОТЧЕТ

ПО ДОМАШНЕМУ ЗАДАНИЮ

Тема: «Сайт больницы с возможностью записи на приём к врачу»

Студент: ?

Группа ИУ5Ц-71Б

Преподаватель: Канев А.И.

2022г.

# Содержание

[Содержание 2](#_Toc122391645)

[Техническое задание 4](#_Toc122391646)

[Цель работы: 4](#_Toc122391647)

[Задачи: 4](#_Toc122391648)

[Требования к функциональным характеристикам: 4](#_Toc122391649)

[Требования к программному обеспечению: 5](#_Toc122391650)

[Введение 6](#_Toc122391651)

[Бизнес-процесс 7](#_Toc122391652)

[Диаграмма прецедентов 7](#_Toc122391653)

[Диаграмма состояний 8](#_Toc122391654)

[Диаграмма деятельности 9](#_Toc122391655)

[Архитектура 10](#_Toc122391656)

[Диаграмма ER 10](#_Toc122391657)

[Диаграмма компонентов 11](#_Toc122391658)

[Алгоритмы 12](#_Toc122391659)

[Диаграмма последовательности 12](#_Toc122391660)

[Описание интерфейса 13](#_Toc122391661)

[Главная страница 13](#_Toc122391662)

[Страница со списком врачей 13](#_Toc122391663)

[Страница просмотра информации о враче 14](#_Toc122391664)

[Страница регистрации 14](#_Toc122391665)

[Страница со списком пациентов 15](#_Toc122391666)

[Страница со списком палат 15](#_Toc122391667)

[Страница с историей болезней 15](#_Toc122391668)

[Страница с видом записей на приём для пациента или врача 15](#_Toc122391669)

[Страница добавления врача 16](#_Toc122391670)

[Страница с видом записей на приём для менеджера 16](#_Toc122391671)

[Панель администратора 17](#_Toc122391672)

[ЗАКЛЮЧЕНИЕ 18](#_Toc122391673)

# Техническое задание

Цель работы: реализовать веб-сервис для больницы, включая возможность записаться на приём к врачу, пользовательский интерфейс и мобильное приложение.

## Задачи:

1. Создать базу данных
2. Создать веб-сервис
3. Реализовать пользовательский интерфейс (фронтенд)
4. Подключить интерфейс к веб-сервису
5. Добавить бронирование услуг, авторизацию и аутентификацию
6. Создать мобильное приложение

## Требования к функциональным характеристикам:

Веб-сервис должен включать следующие роли пользователей:

1. Пациент
2. Врач
3. Менеджер
4. Администратор
5. Гость

Пациент может смотреть список всех врачей, записываться на приём к выбранному врачу, просматривать свои записи, отменять их и смотреть свою историю болезни.

Врач может смотреть просматривать список всех врачей, записи на приём к себе и истории болезней своих пациентов.

Менеджер может просматривать любые данные, а также редактировать информацию о врачах и менять статусы записей к врачам.

Администратор может просматривать и изменять любые данные через специальный интерфейс (панель администратора).

Гость может только смотреть список врачей.

## Требования к программному обеспечению:

На сервере должна быть установлена операционная система Windows 10 21H2.

Для запуска серверной части (бэкенда) на сервере должны быть установлены интерпретатор Python версии не ниже 3.9.7, пакеты Django 4.1.1, djangorestframework 3.14.0, django-filter 22.1 и drf-yasg 1.21.4.

Для сборки программы интерфейса (фронтенда) понадобится NodeJs версии не ниже 16.15.0, а также пакеты vue ^3.2.38, pinia ^2.0.27, vue-router ^4.1.5, vite ^3.0.9 и typescript ~4.7.4.

Клиенту для работы с данным веб-сервисом необходима любая операционная система, позволяющая запустить веб-браузер Firefox версии не ниже 107.0.1, либо любой другой веб-браузер, реализующий поддержку ECMAScript 2015.

# Введение

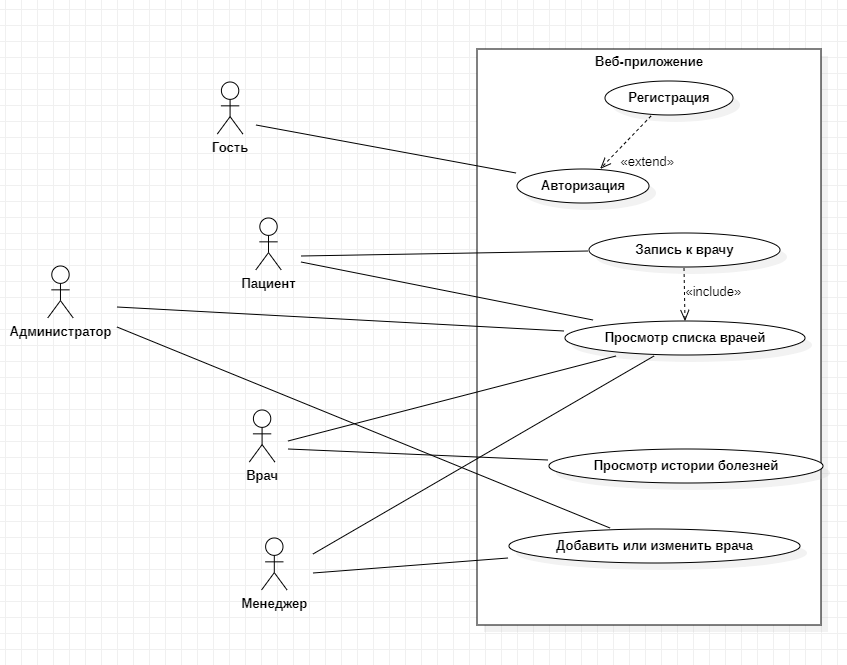
В настоящее время актуальны сервисы для просмотра услуг, совершения заказов, бронирования услуг онлайн. Для этого используются различные интернет-приложения, веб-сайты и мобильные приложения. В ходе курса «Разработка интернет-приложений» необходимо реализовать собственный сервис по выбранной предметной области и представить UML-диаграммы, показывающие его работу. Для выполнения работы была выбрана предметная область больница и запись к врачу.

Сайт больницы позволит пациентам записываться к нужному врачу на удобное для них время, а врачам – просматривать записи и принимать согласно ним пациентов, а также обращаться к истории болезней пациентов при необходимости.

Для создания веб-сервиса была выбрана СУБД SQLite и программная платформа Python + Django, а также фреймворк Vue Js для создания пользовательского интерфейса.

# Бизнес-процесс

## Диаграмма прецедентов



Есть следующие категории пользователей: гость, администратор, пациент, врач и менеджер.

Пациент может смотреть список всех врачей, записываться на приём к выбранному врачу, просматривать свои записи, отменять их и смотреть свою историю болезни.

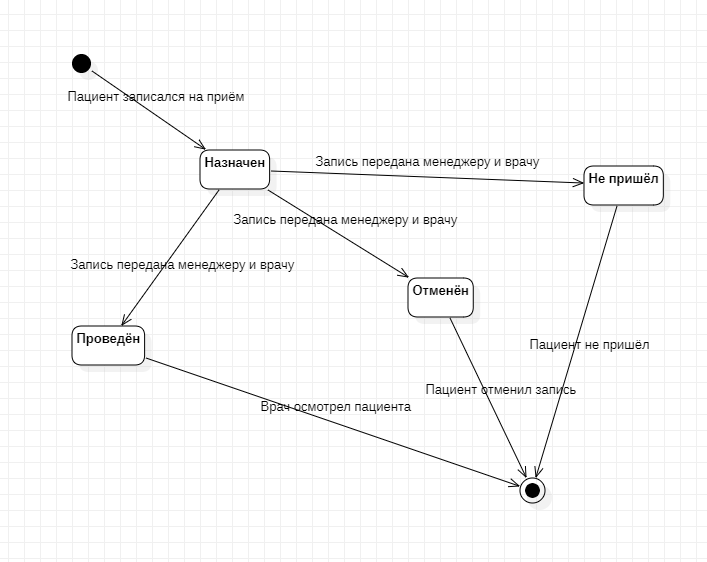
Врач может смотреть просматривать список всех врачей, записи на приём к себе и истории болезней своих пациентов.

Менеджер может просматривать любые данные, а также редактировать информацию о врачах и менять статусы записей к врачам.

Администратор может просматривать и изменять любые данные через специальный интерфейс (панель администратора).

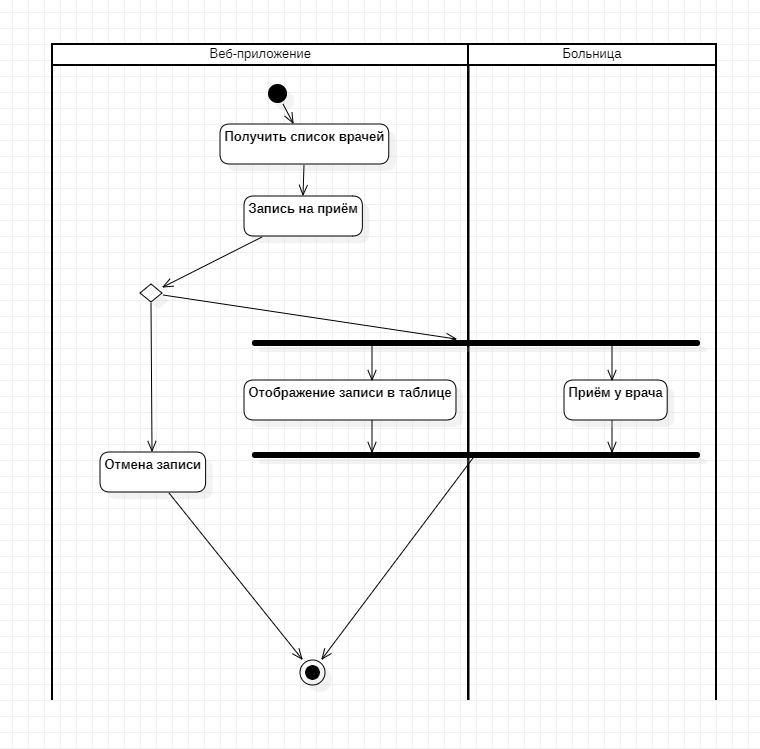
Гость может только смотреть список врачей.

## Диаграмма состояний



После выбора времени, если данное время свободно, запись на приём добавляется в таблицу. Пользователь может прийти на приём в больницу, отменить приём или просто не прийти в больницу. В первом случае проведение приёма подтверждается менеджером. Во втором случае пациенты смогут снова записываться на ранее занятое время. Если же пациент не пришёл в назначенное время, то менеджер отметит приём как не состоявшийся.

## Диаграмма деятельности

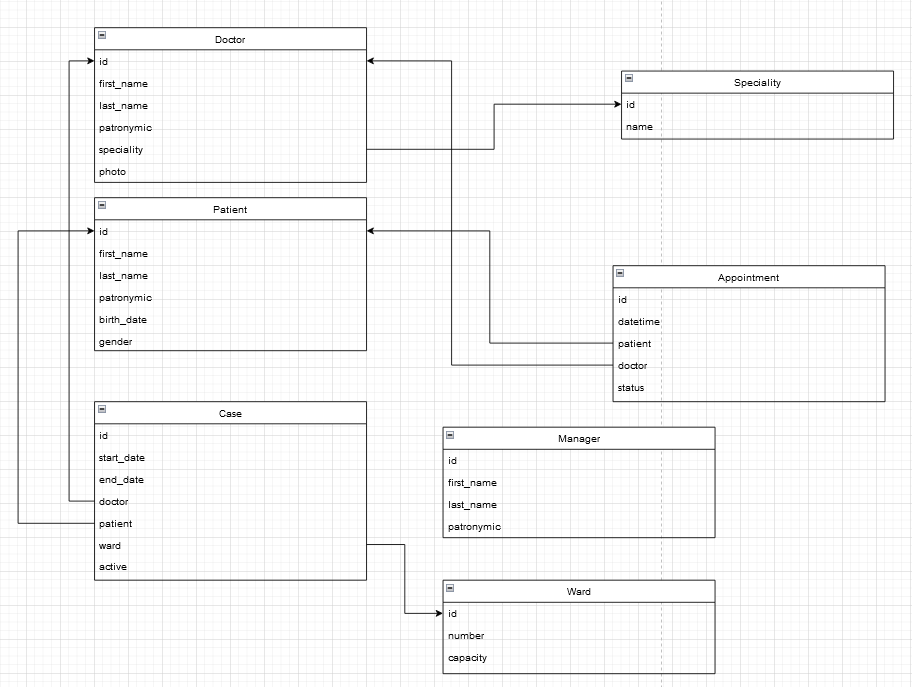


Пользователь получает список врачей и выбирает желаемого. После выбора он переходит на страницу данного врача и выбирает время. Если выбранное время свободно, то запись о приёме добавляется в таблицу. Пользователь может снова перейти к выбору врачей и записаться к другому врачу, либо отменить назначенный приём.

Менеджер подтверждает состоявшийся приём или отмечает его не состоявшимся, если пациент не пришел.

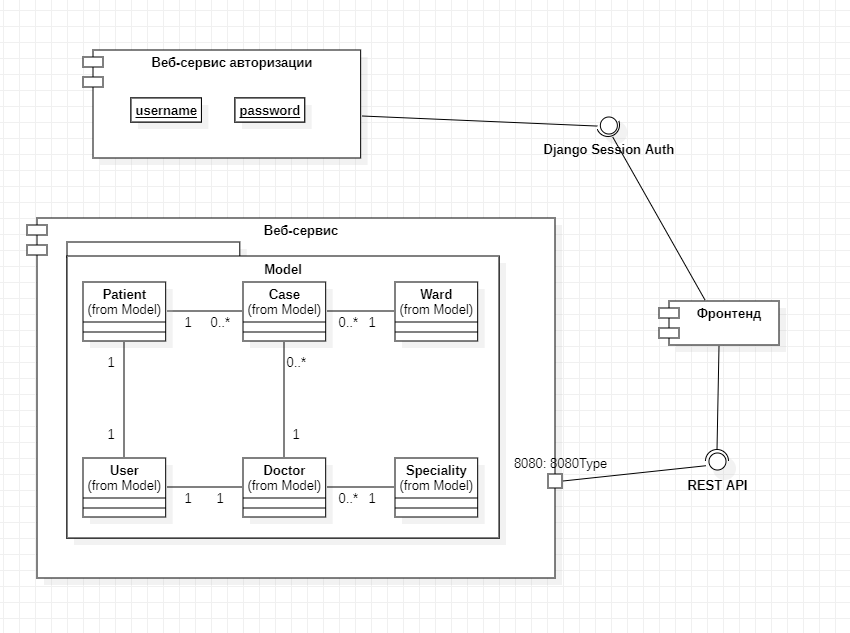
# Архитектура

## Диаграмма ER



На диаграмме представлена база данных, содержащая 6 таблиц: Врачи, Пациенты, Менеджеры, Записи в истории болезней, Записи о приёмах, Специальности врачей и их атрибуты.

## Диаграмма компонентов

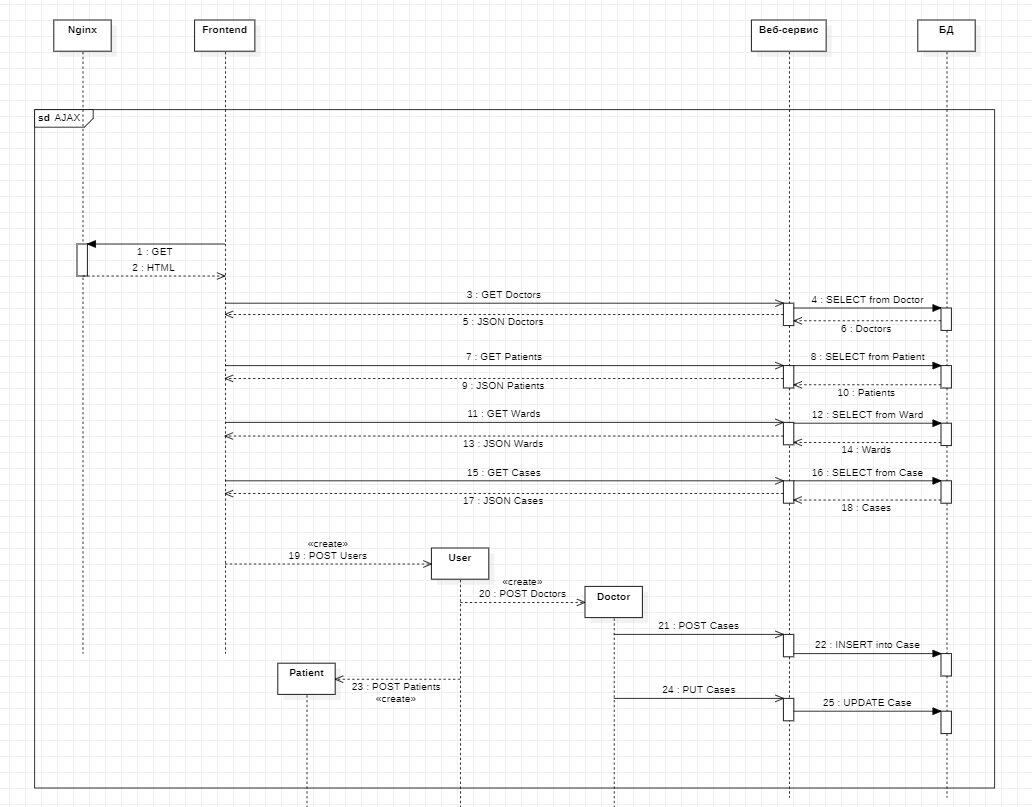


В пользовательском интерфейсе реализованы следующие страницы – с врачами, с палатами, история болезни, таблица записей на приём и также отдельная страница с регистрацией. На странице с врачами происходит взаимодействие с компонентом Doctors по REST API и таблицей Doctor из базы данных.

Аналогично для остальных компонентов.

# Алгоритмы

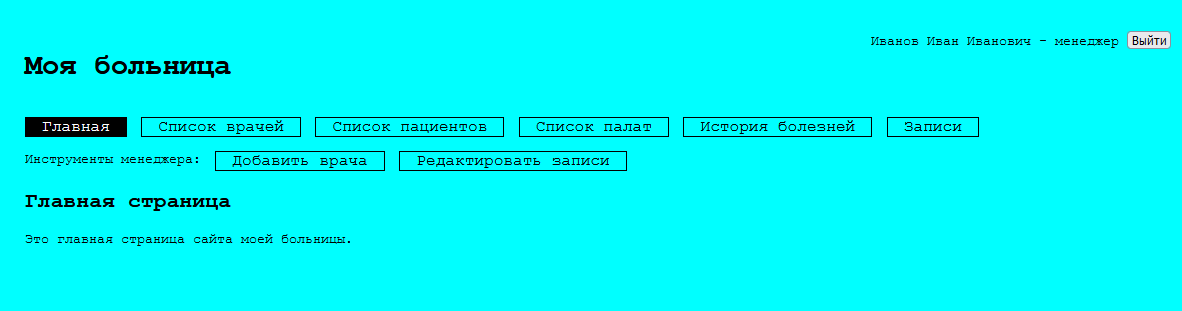
## Диаграмма последовательности



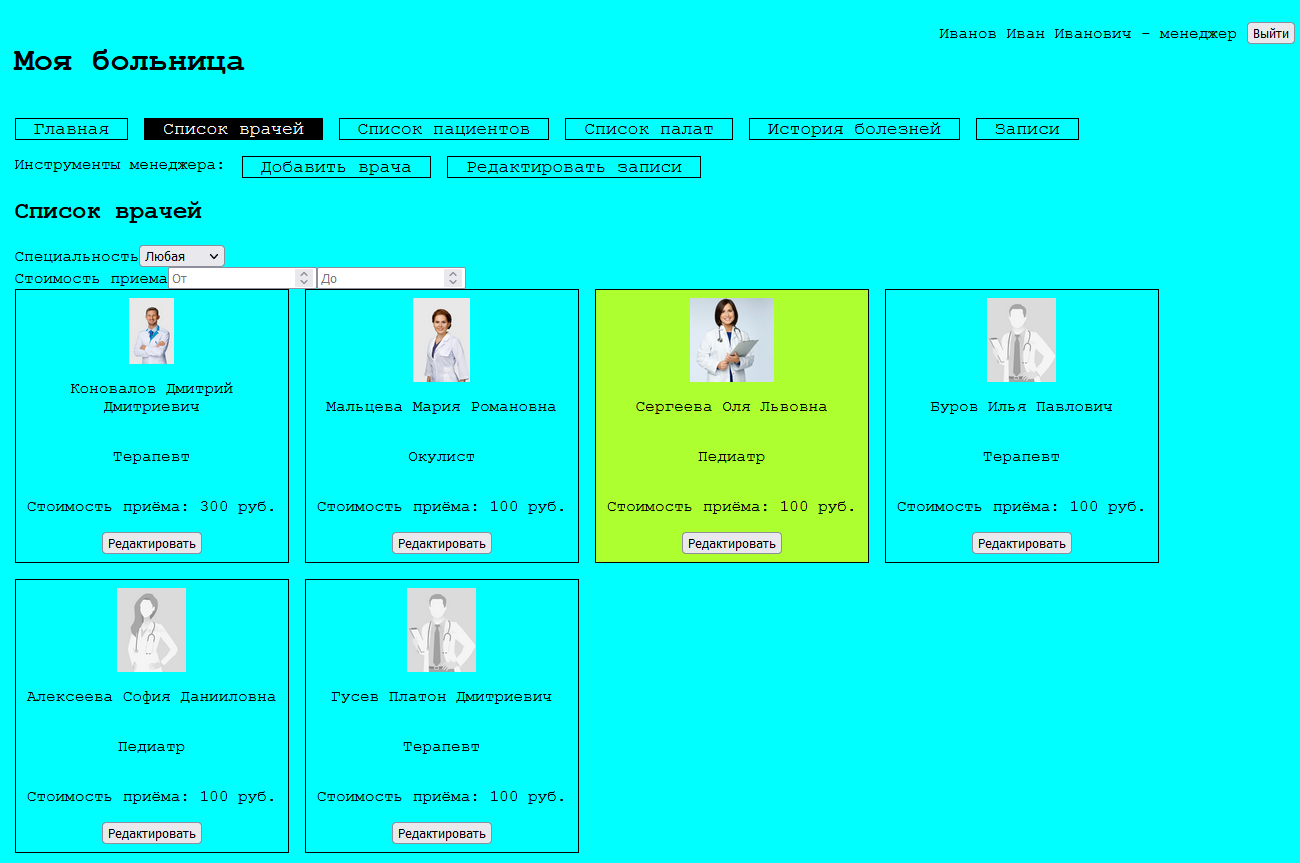
Фронтенд взаимодействует с веб-сервисом при помощи запросов, например, запрос GET запрашивает список врачей, после чего он возвращается фронтенду в JSON файле. Запросы POST и PUT используются для создания и обновления данных.

# Описание интерфейса

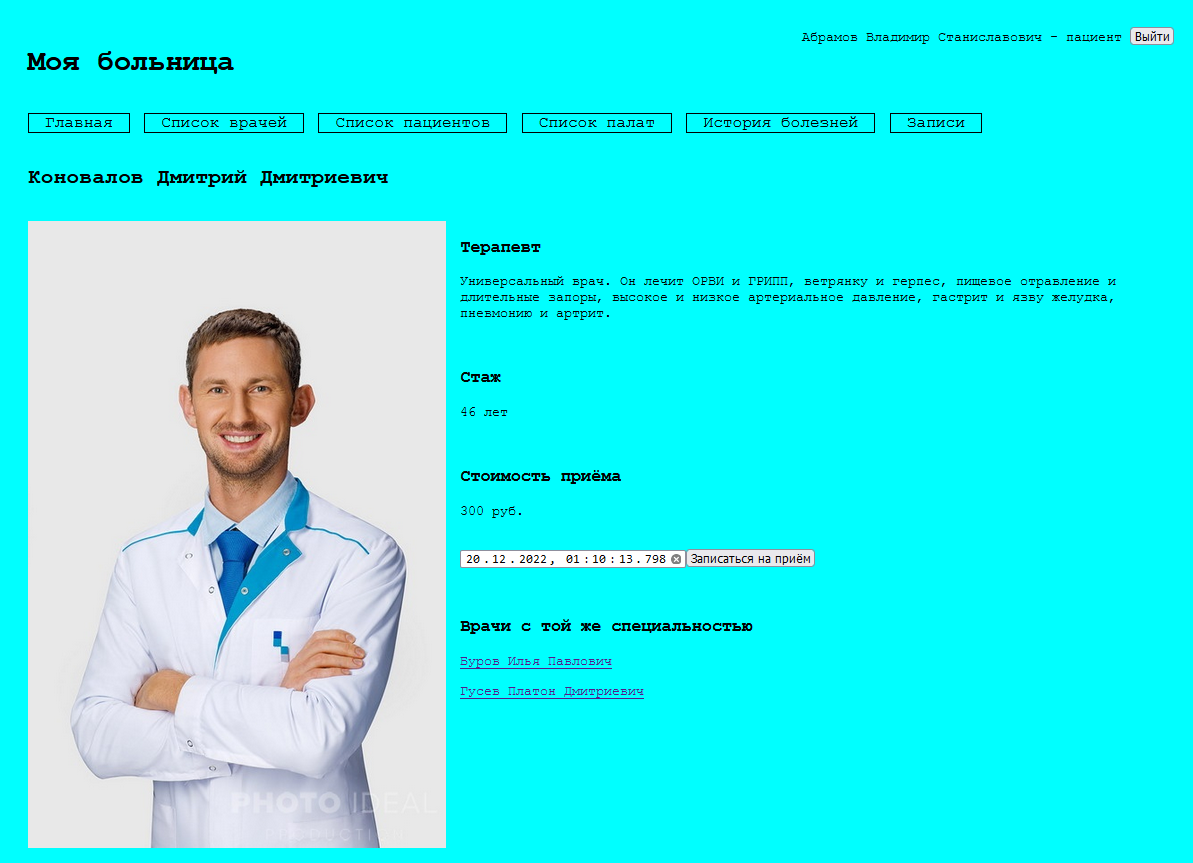
## Главная страница



## Страница со списком врачей



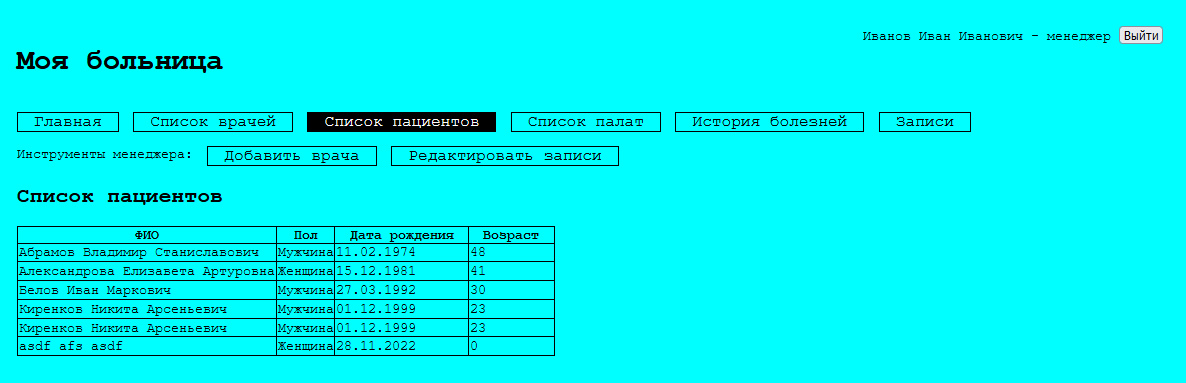
## Страница просмотра информации о враче



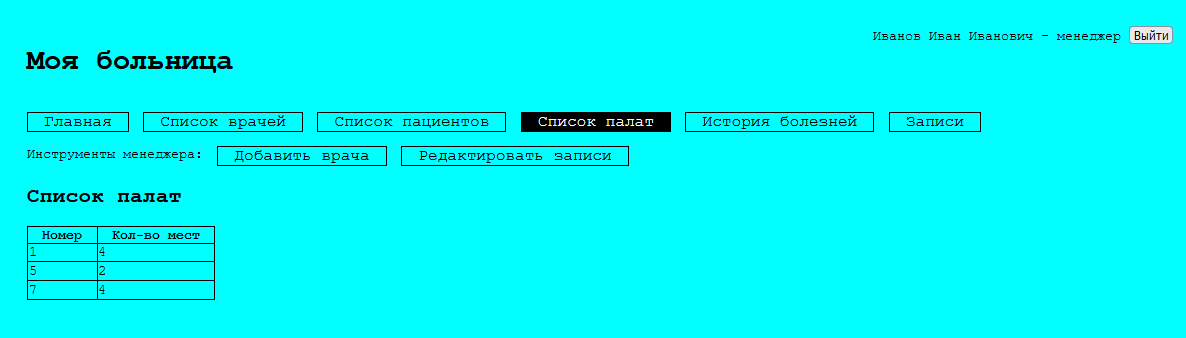
## Страница регистрации



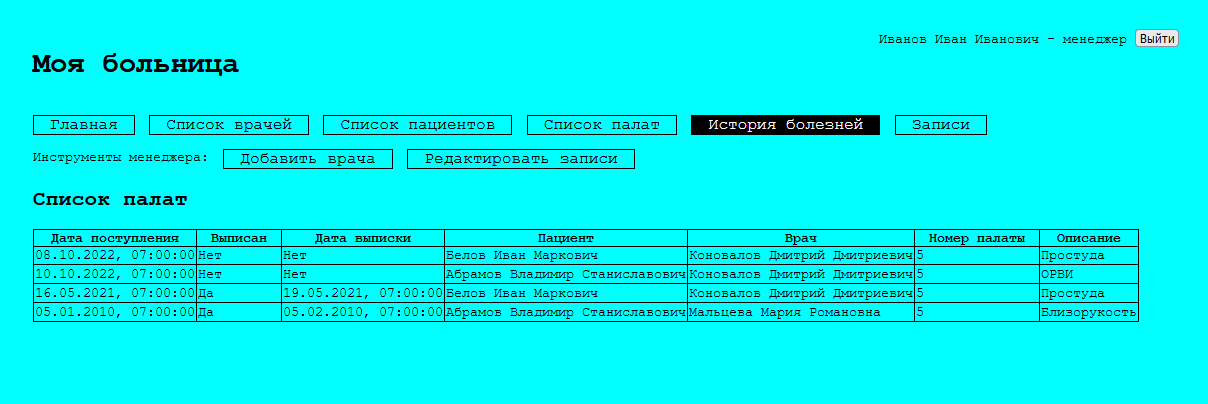
## Страница со списком пациентов



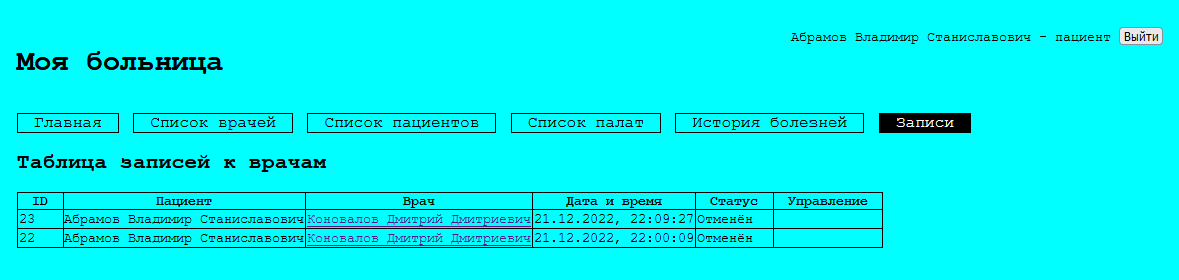
## Страница со списком палат



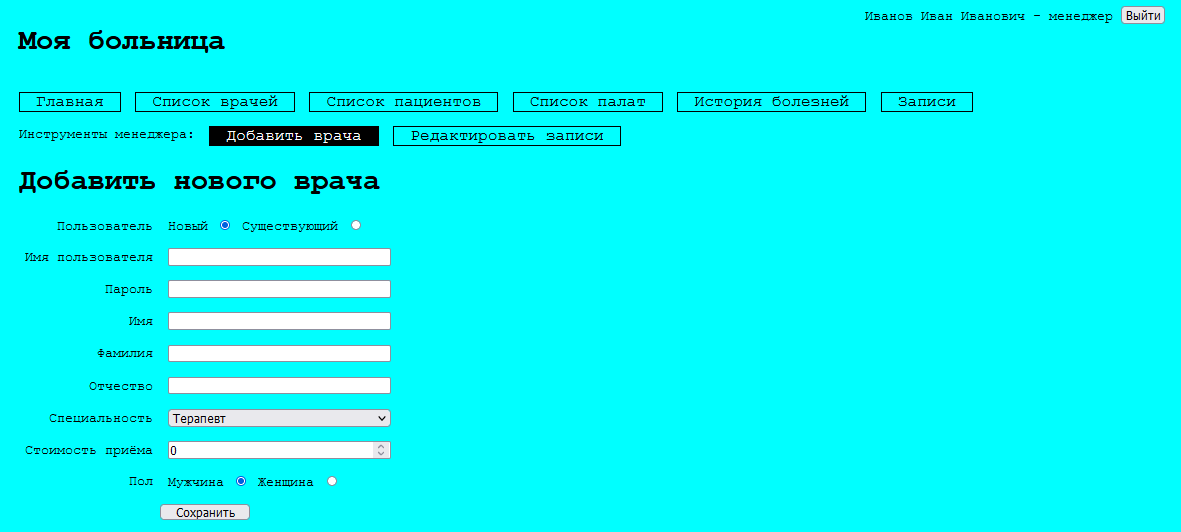
## Страница с историей болезней



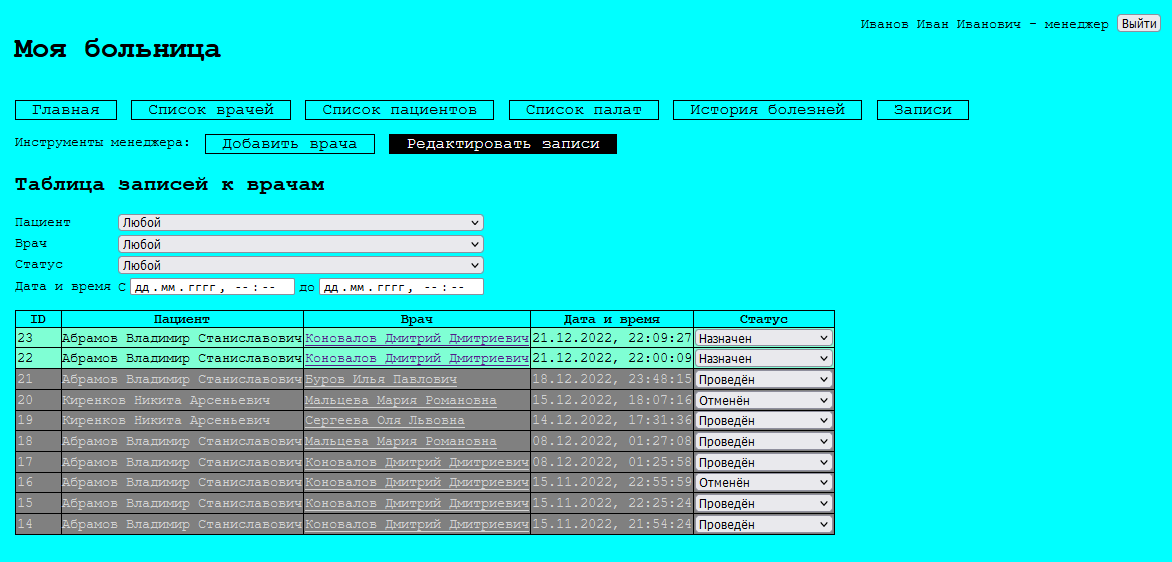
## Страница с видом записей на приём для пациента или врача



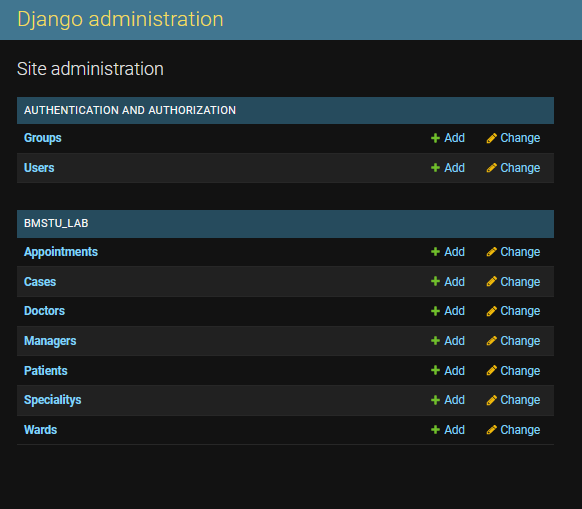
## Страница добавления врача



## Страница с видом записей на приём для менеджера



## Панель администратора



# ЗАКЛЮЧЕНИЕ

В ходе курса были изучены средства для создания интернет-приложений, таких как средства программирования на языках Python и JavaScript, фреймворки Django и Vue Js. Для разработки использовались среды PyCharm и Visual Studio Code.

По итогу курса были реализованы веб-сервис и пользовательский интерфейс для сайта больницы с возможностью записи к врачу, а также кроссплатформенное мобильное приложение. Помимо этого, оформлен единый отчет с представлением всех UML-диаграмм, показывающих работу приложения.