* 1. **ОПИСАНИЕ ОСНОВНЫХ ФУНКЦИЙ ИНФОРМАЦИОННОГО WEB-ПРИЛОЖЕНИЯ “ЭЛЕКТРОННАЯ БОЛЬНИЦА”**

Информационное веб-приложение “Электронная больница” разделяет пользователей на следующие роли: неавторизованный пользователь, пациент, доктор, лаборант, администратор клиники, модератор клиники и администратор системы.

**5.1 Функциональная часть неавторизованного пользователя**

Возможности неавторизованного пользователя представляют собой функционал, доступ к которому имеют как гостевые пользователи, которче не были атворизованные в системе, так и пользователи, которые принадлежат к определенной роли. Главная страница, которая отображена на рисунке 5.1., открывается при образении по IP-адресу или DNS-имени сайта. По умолчанию, система маршрутизации вызывает метод Index контроллера Home. Меню навигации главной страницы позволяет получить доступ к клиникам, лабораториям, и также, к общей карте мед.учреждений.

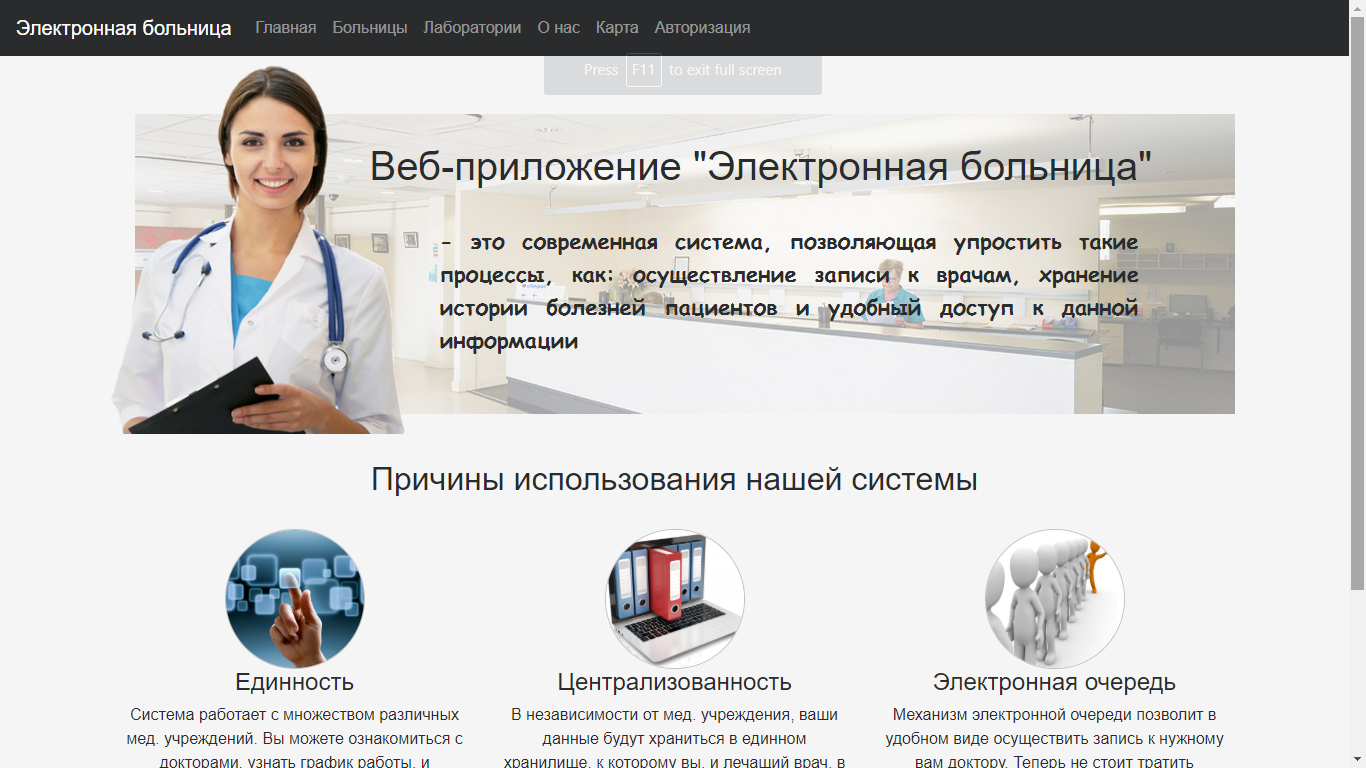


Рисунок 5.1 – Главное меню программы

При нажатии на меню "Больницы" происходит HTTP-запрос по адресу /Home/Clinics, где Clinics - метод класса контролера Home. Результатом запроса является страница со списком больниц, зарегистрированных в системе. Названия больниц представляет собой ссылку на страницу конкретной больницы. Страница со списком больниц изображена на рисунке 5.2.

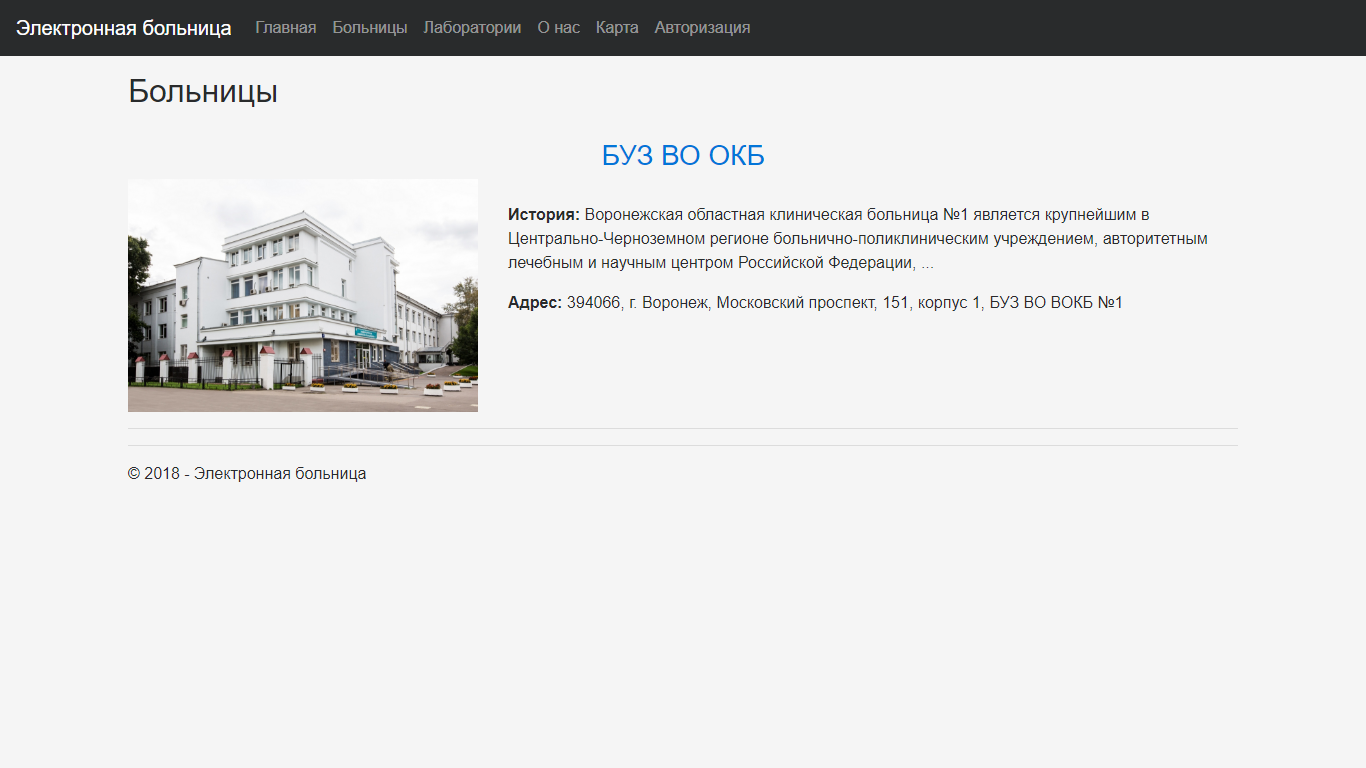


Рисунок 5.2 – Страница со списком больниц

Средствами LinQ to Entities, метод Clinics обращается к базе данных, и получает список больниц, который сортируется по названию по возрастанию. В представление возвращается этот список, и отображает информацию на экран. [HttpGet]

public async Task<ActionResult> Clinics()

{

List<Clinic> clinics = await db.Clinics.OrderBy(c => c.Name).ToListAsync();

return View(clinics);

}

После перехода по ссылке на страницу больницы, пользователю будет сформирована страница, которая отражена на рисунке 5.3. Страница состоит из четырех вкладок. Главная вкладка содержит название, фотографию больнице, и её историю.

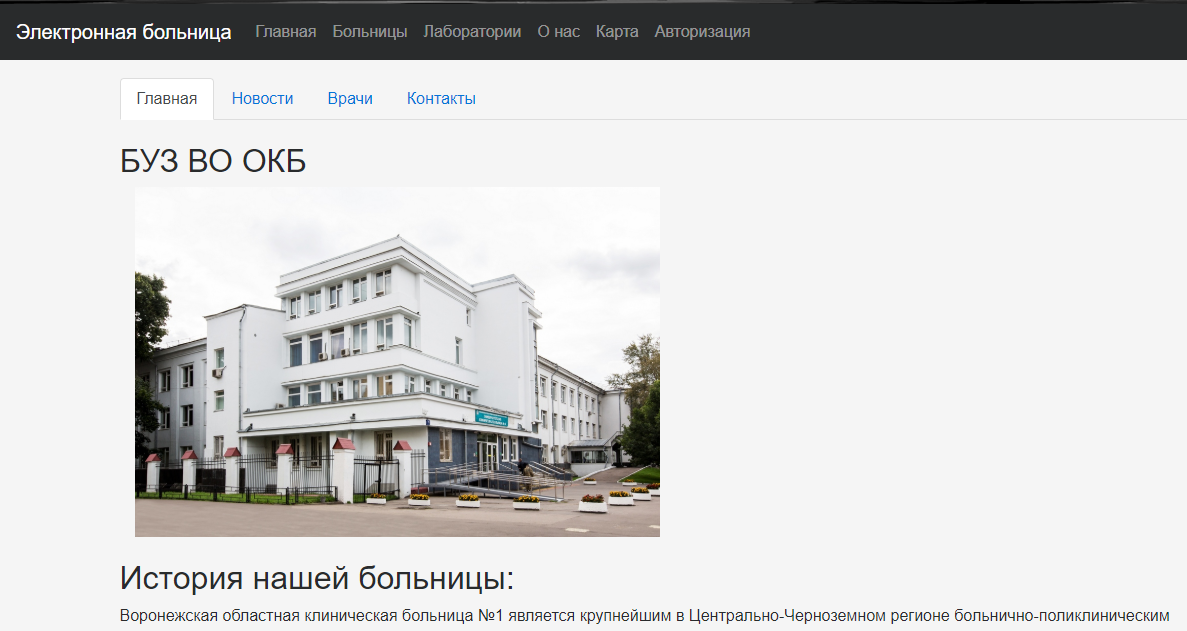


Рисунок 5.3 – Страница больницы

Ссылка вызывает метод Clinic контроллера Home, который принимает индекс больницы в качестве аргумента. С помощью метода Find происходит запрос к базе с поиском больницы по указанному индексу. Результат передается методу View, который передает клинику в строго типизированное представление.

public ActionResult Clinic(int id)

{

Clinic clinic = db.Clinics.Find(id);

return View(clinic);

}

Вкладка "Новости" позволяет просмотреть письмо новостей клиники с сокращенным описанием. Вместе с этим, пользователю будет выведена фотография новости и дата добавления. Вкладка с новостями больницы отражена на рисунке 5.4.

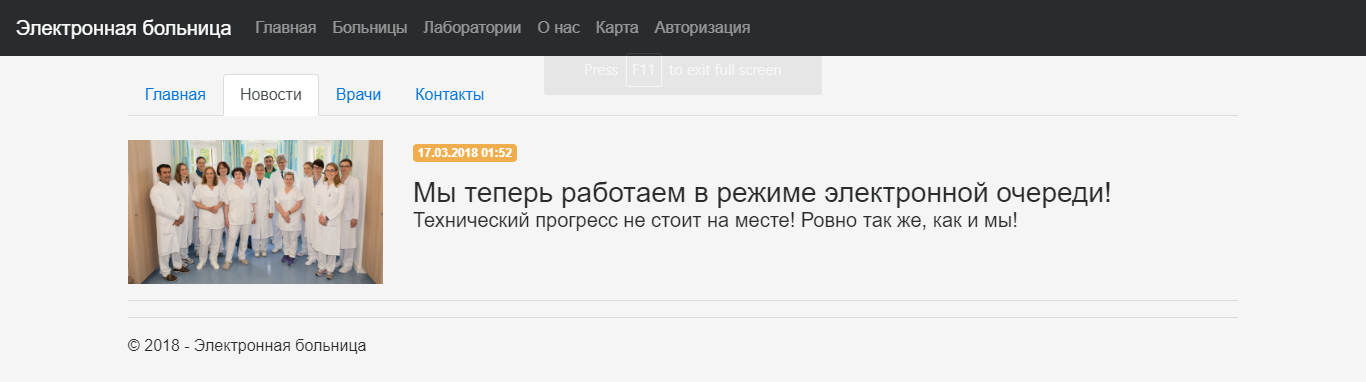


Рисунок 5.4 – Вкладка с новостями больницы

Название новости представляет собой ссылку на полную версию новости. После нажатии на название новости, будет отправлен HTTP-запрос на получение новости из базы данных по id. За обработку запроса отвечает метод News класса HomeController:

public ActionResult News(int id)

{

ClinicNews news = db.ClinicNews.Find(id);

return View(news);

}

После удачного поиска новости по идентификатору, пользователю будет выведено представление с полной новостью, которая содержит в себе название, полный текст новости, фотографии и комментарии зарегистрированных пользователей. Пример новости больницы отражен на рисунке 5.5.



Рисунок 5.5 –Новость больницы

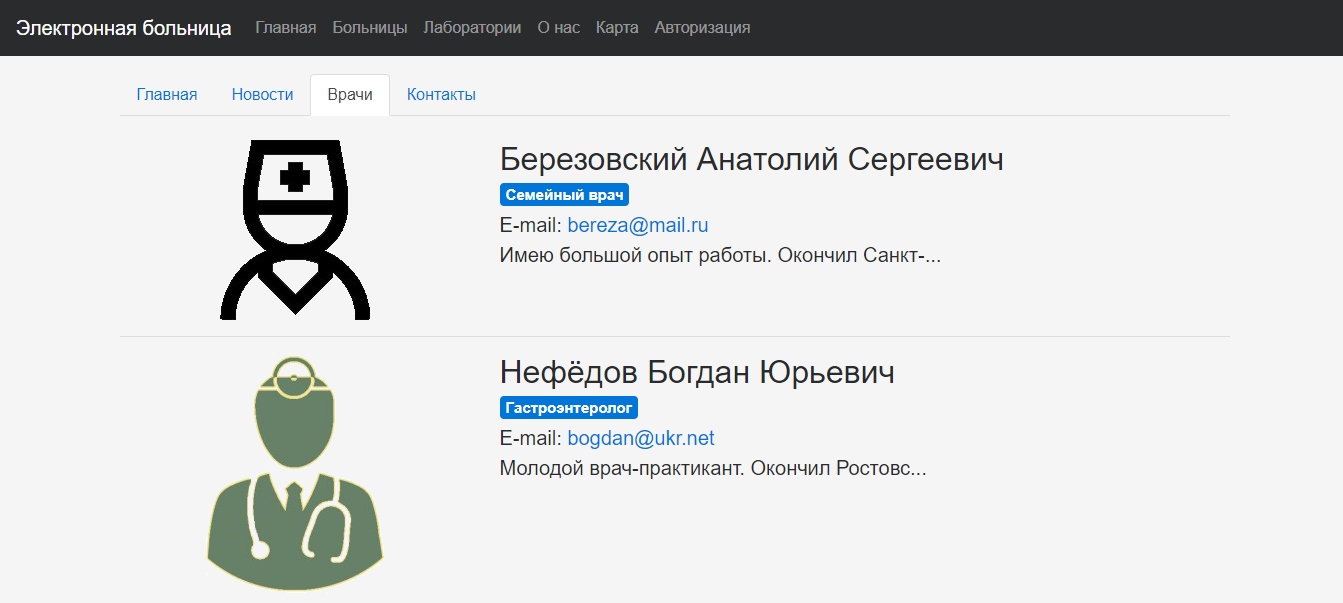
Следующая вкладка на странице больнице "Врачи". Данная вкладка содержит в себе ФИО докторов, которые были зарегистрированы в больнице, короткую строку с информацией о себе, фоторафию, электронный почтовый адрес, и свою специальность. ФИО врача является ссылкой на страницу конкретного врача. Вкладка врачей отражена на рисунке 5.6. 

Рисунок 5.6 – Вкладка врачей

За получение информации о враче по его идентификатору соответствует метод Doctor контроллеруа Home. Этот метод принимает целочисленный идентификатор врача, и отсылает запрос к базе данных.

public async Task<ActionResult> Doctor(int id)

{

Doctor doctor = await db.Doctors.FindAsync(id);

return View(doctor);

}

После удачного поиска, пользователю будет отправлена HTML-страница с полной информацией о враче. Страница содержит такую информацию, как: ФИО, електроный почтовый адрес, дату рождения, полную информацию о себе, и таблицу с графиком работы. Страница конкретного врача отражена на рисунке 5.7.

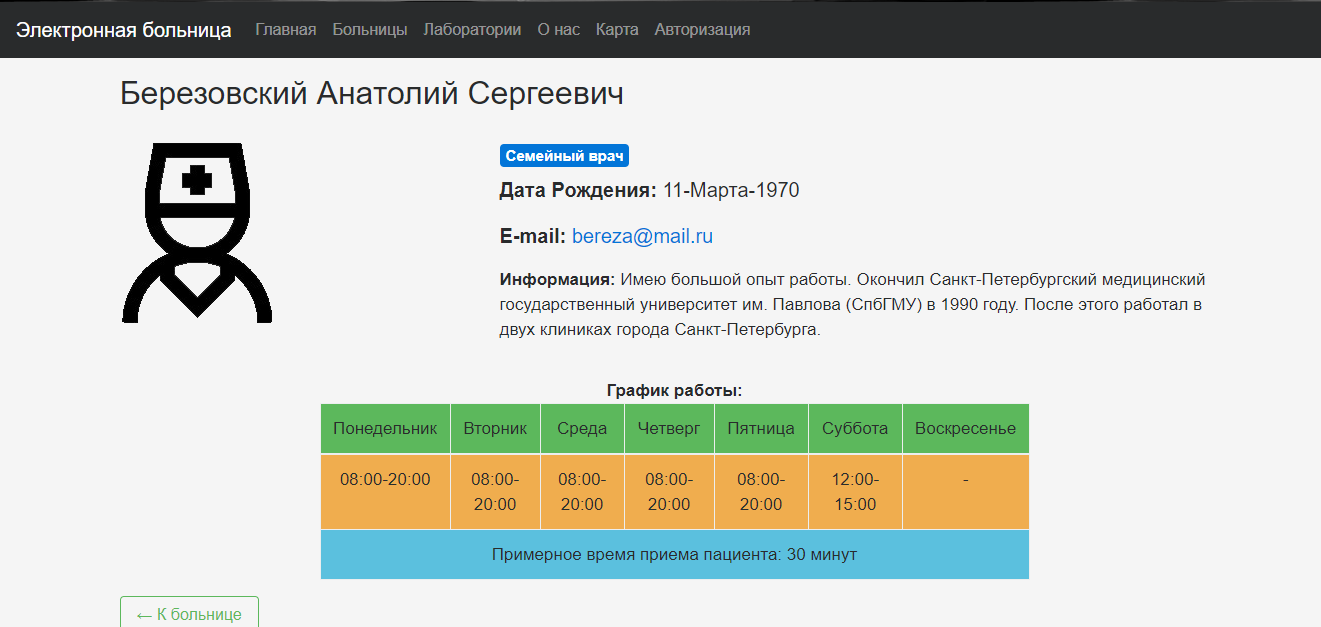


Рисунок 5.7 – Страница конкретного врача

Последняя вкладка "Контакты" содержит в себе Google-карту, на которой располагается маркер расположения заведения. Над маркером изображена панель с названием клиники, ее адресу, и номером телефона. Данная вклакда изображена на рисунке 5.8.

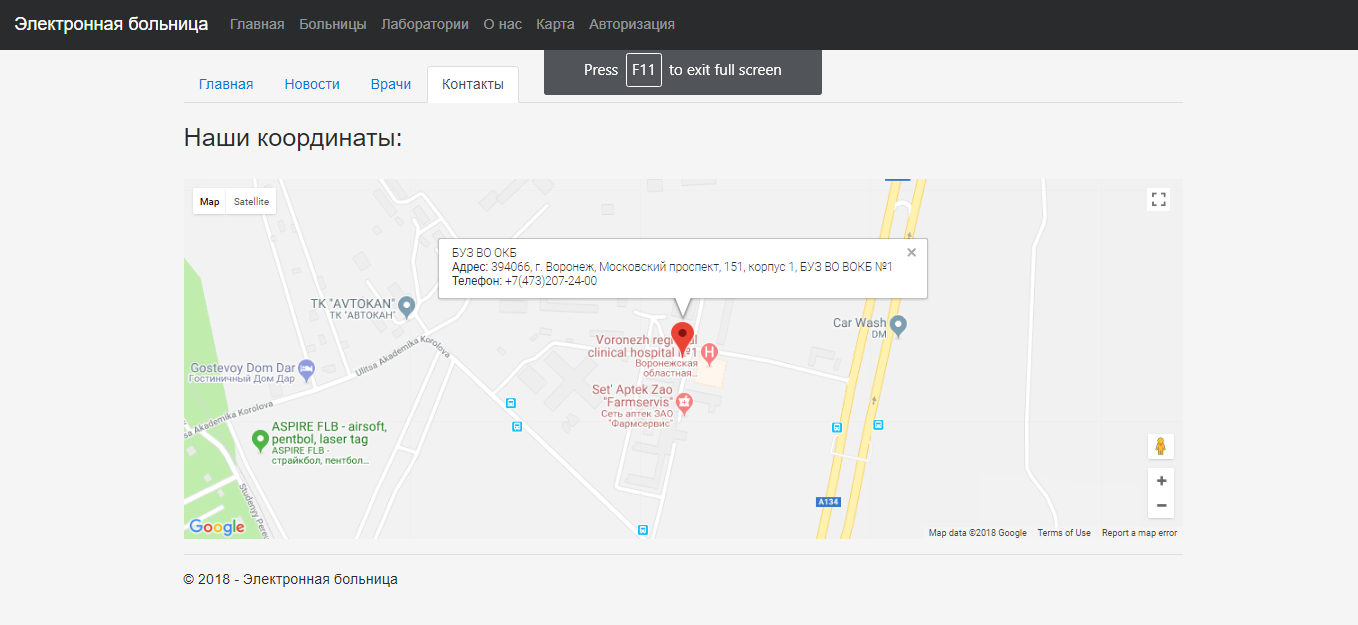


Рисунок 5.8 – Вкладка “Контакты”

Страница "Карта" содержит в себе Google-карту, на которой располагаются все медицинские учреждения и аптеки, которые были зарегистрированы в системе. Каждый вид заведения помечается маркером определенного цвета. Таким образом, больницы обозначается красным цветом, лаборатории - синим, а аптеки - желтым. Также, карта содержит легенду, которая подсказывает цвета пользователю. При нажатии конкретного маркера, сверху открывается панель с названием заведения, адресу, и номером телефона. Название содержит ссылку на страницу заведения. Страница с картой изображена на рисунке 5.10.

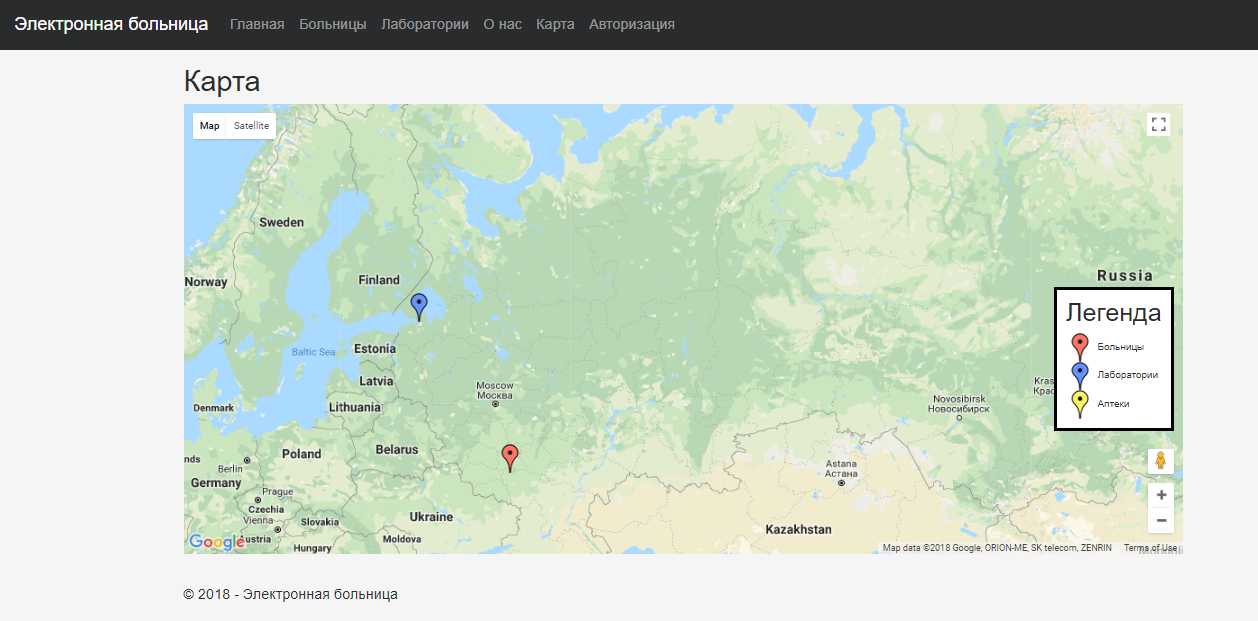


Рисунок 5.10 – Страница с картой

Следующая страница - страница авторизации пользователя в системе. Форма авторизации состоит из поля логина и пароля. При нажатии на кнопку "Войти", к серверу отправляется POST-запрос, который направляет запрос к базе данных. После получения результата, метод Authorization класса Home анализирует роль пользователя.

var currentUser = await db.Users.Include(u => u.Role).SingleAsync(u => u.Login == user.Login && u.Password == user.Password);

if (currentUser.Role.Name == "Модератор клиники")

Session["clinic"]=db.OtherWorkers.Find(currentUser.Id).Institution.Id;

else if (currentUser.Role.Name == "Администратор лаборатории")

Session["laboratory"]=db.OtherWorkers.Find(currentUser.Id).Institution.Id;

Session["currentUser"] = currentUser;

return RedirectToAction("Index", "Cabinet");

В случае, если роль пользователя администратор лаборатории или модератор больницы, в параметры сессии добавляется дополнительная переменная, которая указывает на идентификатор медицинского учреждения, где работает пользователь. После этого, для всех пользователей к сессии в параметр currentUser записывается объект пользователя. Страница авторизации изображена на рисунке 5.11.

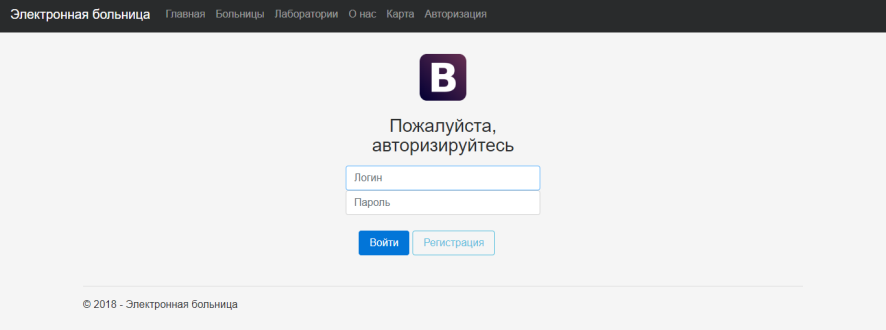


Рисунок 5.11 –Страница авторизации

Кнопка "Регистрация" позволяет пройти процесс регистрации пациента в системе. Форма регистрации состоит с полей ФИО, день рождения, электронного почтового адресса, поля для загрузки фотографии, логина и пароля. Форма регистрации изображена на рисунке 5.12.

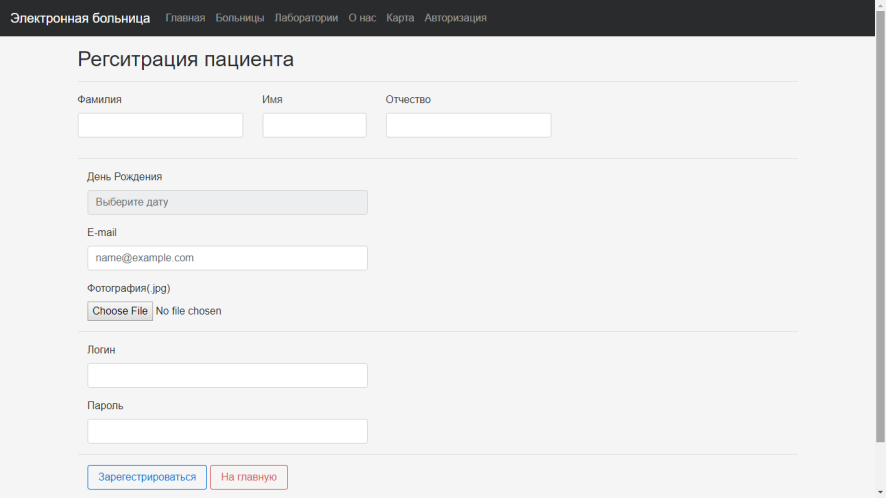


Рисунок 5.12 – Форма регистрации пациента

После отправки заполненной формы вызывается метод Registration контроллера Home. Ключевым действием метода является сохранение фотографий в базу в байтовом виде, и отправки письма подтверждения регистрации на почтовый адрес пользователя.

if (photo != null){

byte[] imageData = new byte[photo.ContentLength];

await photo.InputStream.ReadAsync(imageData,0, photo.ContentLength);

p.Photo = imageData;

p.cutPhoto();

}

MailSender send = new MailSender(p.Email);

send.SendMessageForConfirmAccount(Url.Action("ConfirmEmail", "Cabinet", new { id = p.Id, email = p.Email }, Request.Url.Scheme));

После успешного процесса авторизации происходит переход в кабинет пользователя. За формирование кабинета отвечает метод Index класса CabinetController. Личные кабинеты пользователей содержат фотографию, ФИО, дату рождения, почтовый адрес, и кнопки, по которым пользователь получит доступ к общему и специфическому функционалу. К общему функцоналу относятся такие функции, как: изменение электронного почтового адреса, пароль, и изображения.

**5.2 Функциональная часть администратора клиники**

Администратор системы является главным типом пользователя. Его функционал позволяет добавлять новые и удалять существующие медицинские учреждения, назначать администраторов больницы, запускать пересчет весов симптомов, и просматривать графики, которые отображается базу знаний системы. Аккаунт администратора может быть только один, и он регистрируется с самого начала функциюваня системы. Личный кабинет администратора системы изображен на рисунке 5.13.

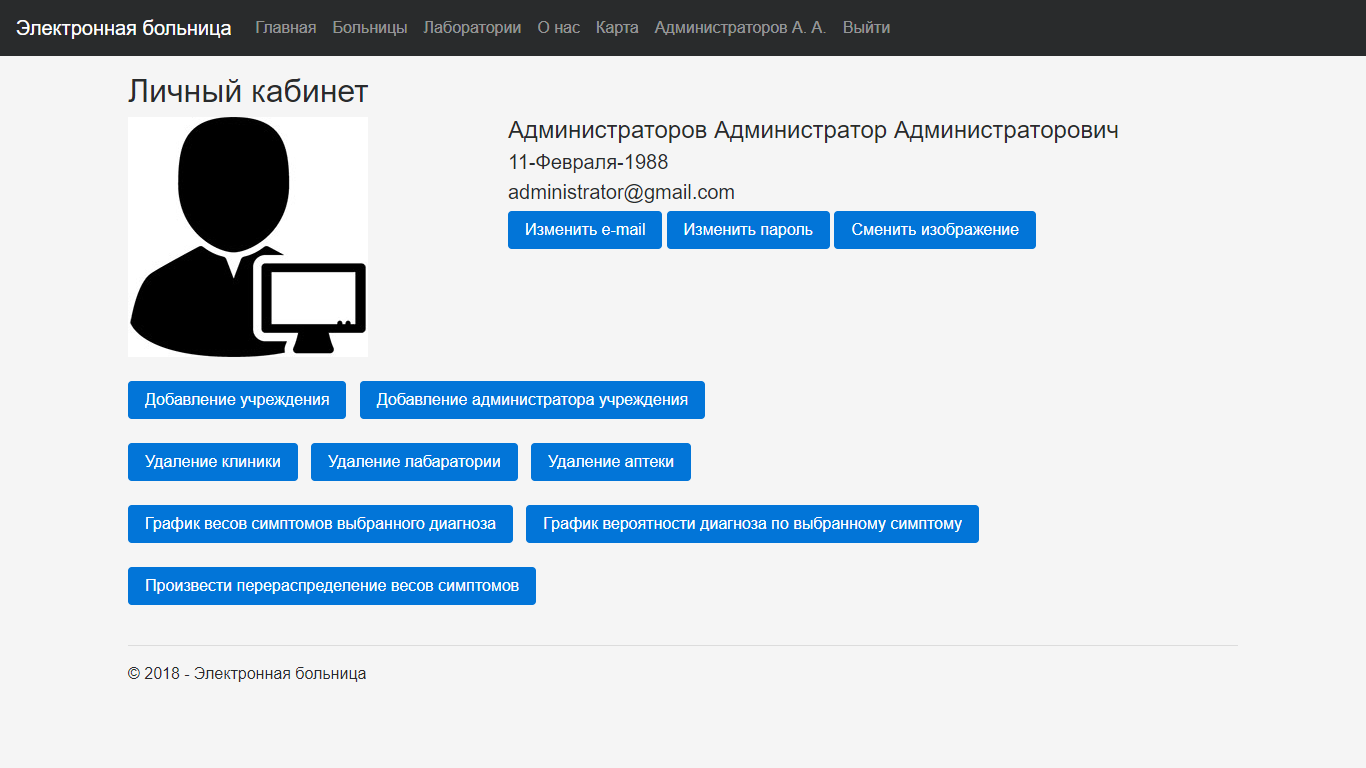


Рисунок 5.13 – Личный кабинет администратора системы

При нажатии кнопки для изменения почтового адреса, с помощью веб-фреймворка Bootstrap, будет открыто диалоговое окно с формой. Данная форма содержит поля для сторого адрес, и для нового. Значение из формы будут отправлены POST-запросом к методу ChangeEmail контроллеру Cabinet. Окно отражено на рисунке 5.14.

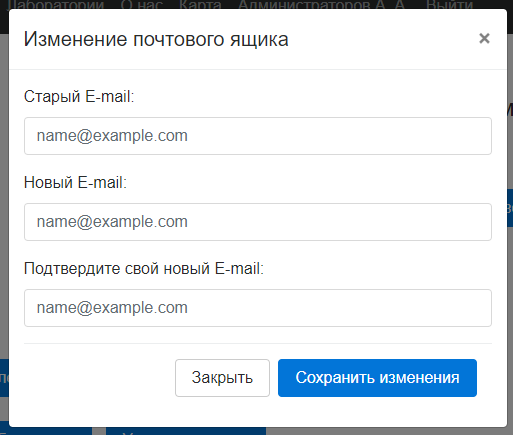


Рисунок 5.14 – диалогове окно для изменения E-mail

Первая функция администратора системы является добавлением нового медицинского учреждения в систему. Эта операция выполняется с помощью Google-карт. Страница добавления нового заведения изображена на рисунке 5.15.

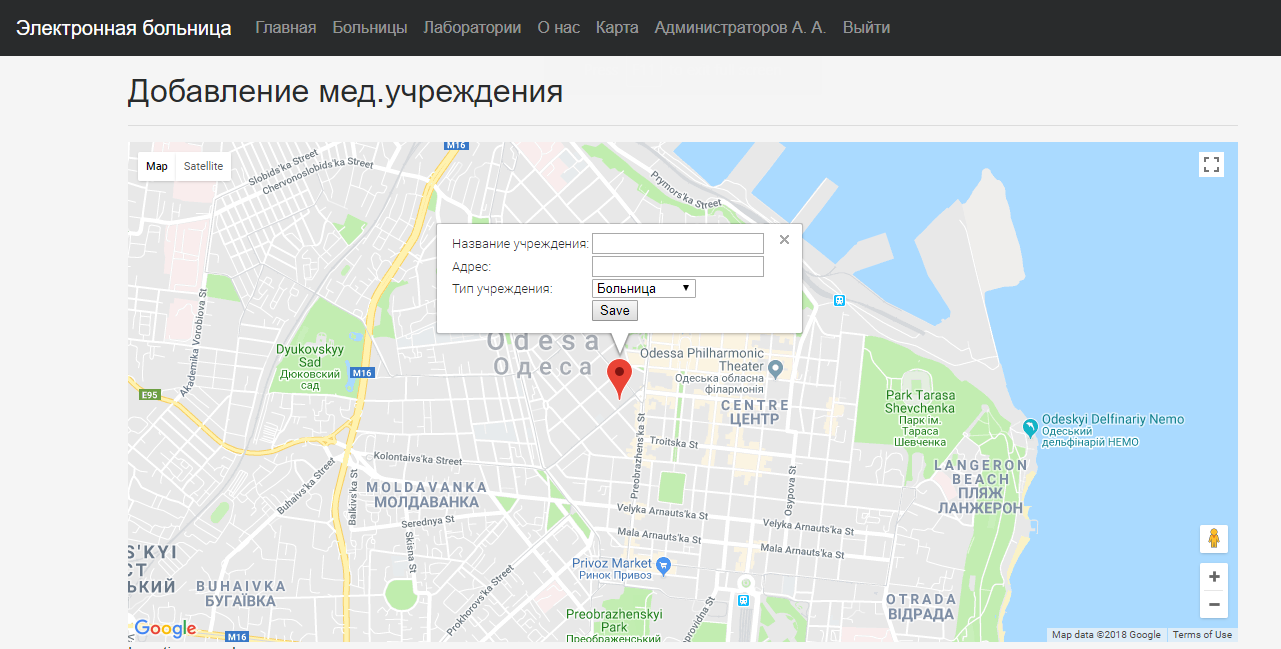


Рисунок 5.15 – Страница добавления нового учреждения

Пользователь нажимает кнопку мыши на нужном месте на карте. После этого появляется маркер с формой для заполнения. Форма содержит в себе название заведения, полный адрес, и тип заведения. Если администратор хочет переместить маркер, он сможет это сделать с помощью мыши. После отправки формы, сервер будет вызывать метод AddInstitution класса AdminController. Этот метод анализирует указанный тип, и в зависимости от этого добавляет данные в нужную таблицу.

switch (type){

case "clinic": db.Clinics.Add(ins.ConvertToClinic());

break;

case "laboratory": db.Labaratories.Add(ins.

ConvertToLabaratory());

break;

case "pharmacy": db.Pharmacies.Add(ins.

ConvertToPharmacy());break;

}

Блок функции с удалением медицинских учреждений работают схоже с собой. Разница есть данные, удаляется вместе с заведением. Таким образом, для удаления больнице есть за необходимости удаления модераторов и администраторов больницы, удаление всех новостей и комментариев под ними. В форме удаления, администратор выбирает необходимый заведение, и нажимает кнопку удаления. Форма удаления больниц отражена на рисунке 5.16.

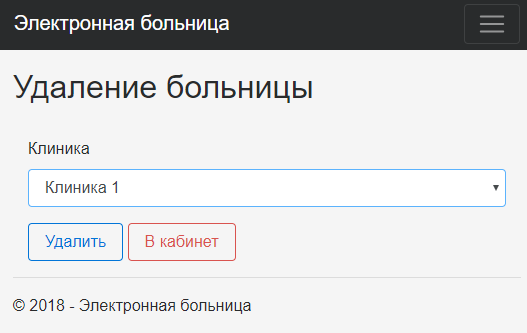


Рисунок 5.16 – Форма удаления клиники

Для просмотра состояния базы знаний есть две кнопки: просмотр графика весов симптомов для конкретного диагноза, и просмотр графика вероятности встречи дагноз по выбранному симптомов. При необходимости, диаграммы можно загрузить в .png формате нажав кнопку "Загрузить". Весы симптомов для инсульта изображенное на рисунке 5.17.

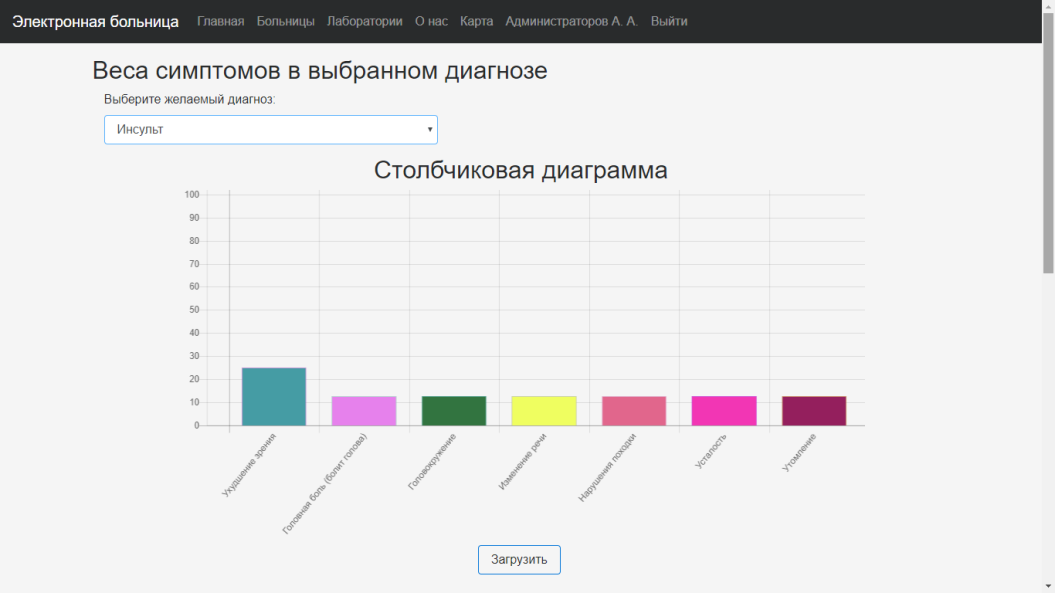


Рисунок 5.17 – Веса симптомов для иснульта

**5.3 Функціональна частина адміністратора клініки**

Администратор клиники имеет небольшое количество функции, направленные на регистрации працивникив клиники, и её удаления. Кабинет администратора больницы изображен на рисунке 5.18.

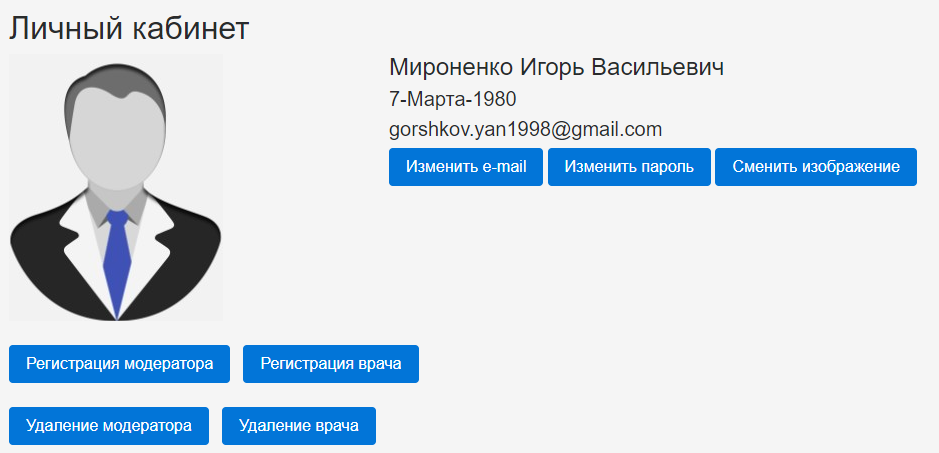


Рисунок 5.18 – Кабинет администратора клиники

За функционал данного типа пользователя соответствует класс ClinicAdminController. Регистрация доктора похожа с регистрацией пациента, но форма имеет дополнительные поля: информацию о враче, специалнисть, график работы, и время приема одного пациента и время дежурства. Пароль при регистрации задается по умолчанию - temppassword. Форма регистрации доктора отражена на рисунке 5.19.

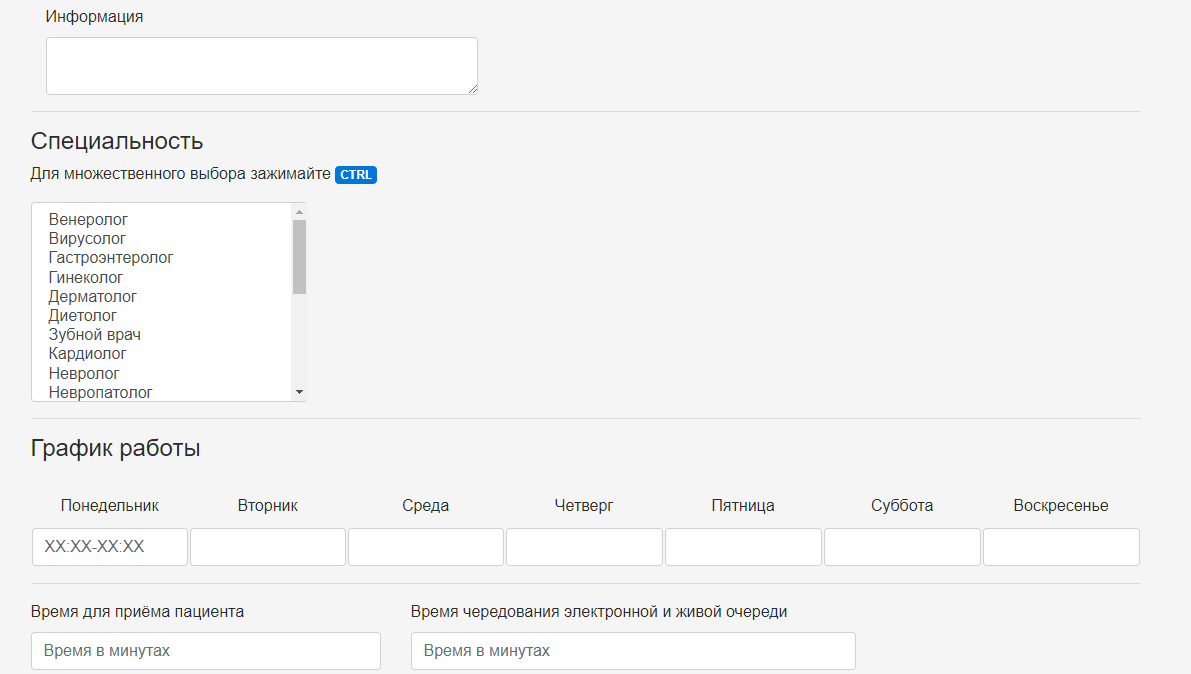


Рисунок 5.19 – Форма регистрации доктора

После отправки формы, веб вызовет метод AddDoctor контроллеру ClinicAdmin.

d.Password = “tempPassword”;

d.Specialities = new List<DoctorSpeciality>

(SpecialityList.Count);

foreach (var speciality in SpecialityList){

int id = int.Parse(speciality);

d.Specialities.Add(db.DoctorSpecialities.Find(id));

}

Удаление врача происходит с помощью формы со список врачей. Администратор выбирает нужного врача, и подтверждает выбор. Вместе с врачом будет удалены и текущей записи пациентов. Страница удаления врачей отражена на рисунке 5.20.

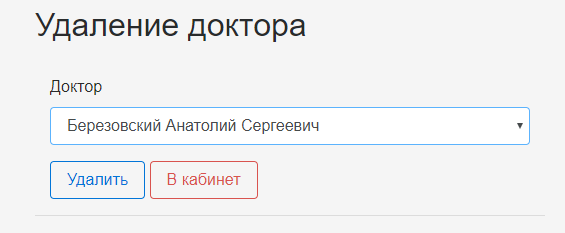


Рисунок 5.20 – Страница удаление врача

* 1. **Функціональна частина модератора клініки**

Модератор клиники отвечает за информационную составляющую клиники. К этому относятся новости, комментарии, информация о больнице. За функциональность модератора соответствует класс Moderator Controller.

Форма для добавления новости состоит из заголовка, сокращенной новости, полной новости, и изображения. После отправки формы, сервером будет вызван метод AddNews контроллеру Moderator. Форма добавления новости изображена на рисунке 5.21. Это единственная функция, которая доступна из кабинета модератора.

Доступ к инших функции модератор получает на странице больницы. Таким образом, для редактирования новостей, модератор переходит на страницу конкретной новости, и выбирает нужную функцию.

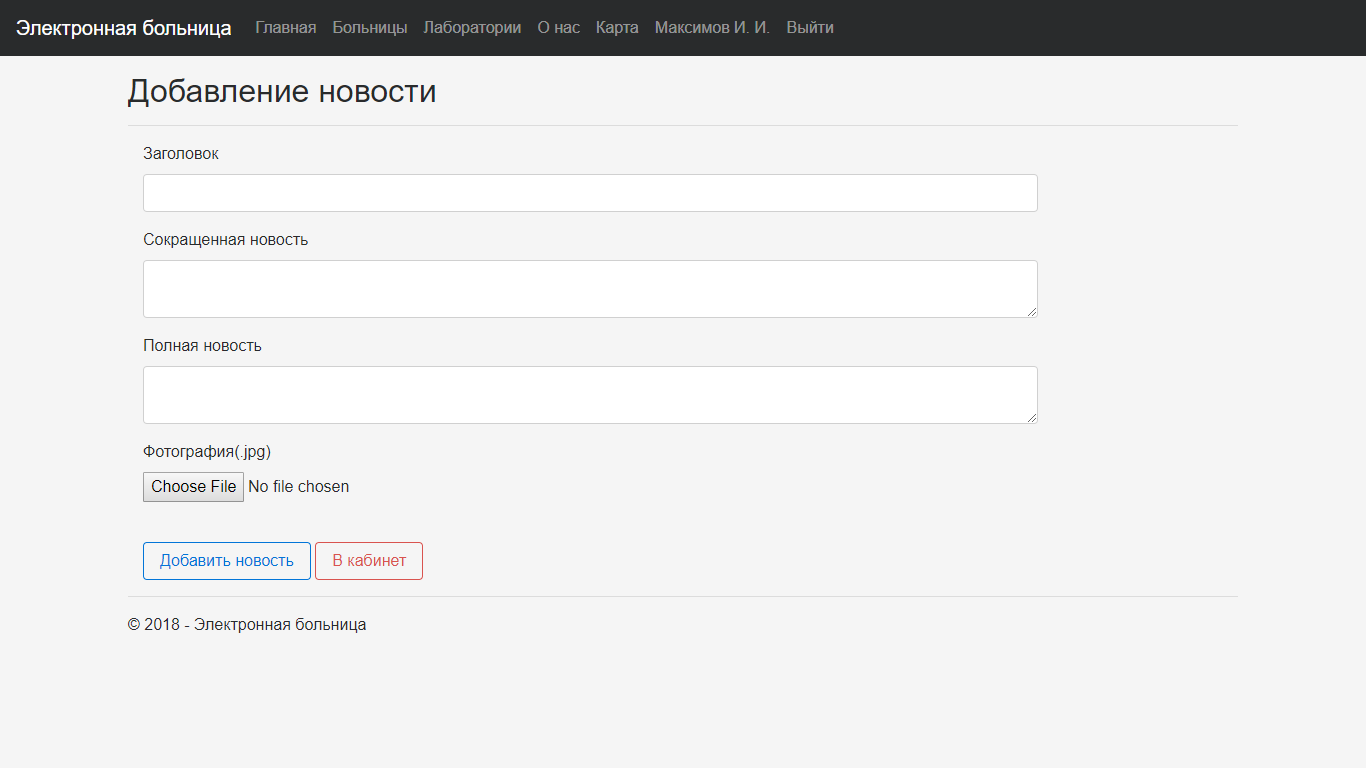


Рисунок 5.21 – Форма добавление новости

Страница новости для модератора изображена на рисунке 5.22. В этом месте содержится функции удаления комментариев, редактирования и удаления новостей. За это функционал соответствуют следующие методы контроллеру Moderator: EditNews, DeleteNews, DeleteComment.

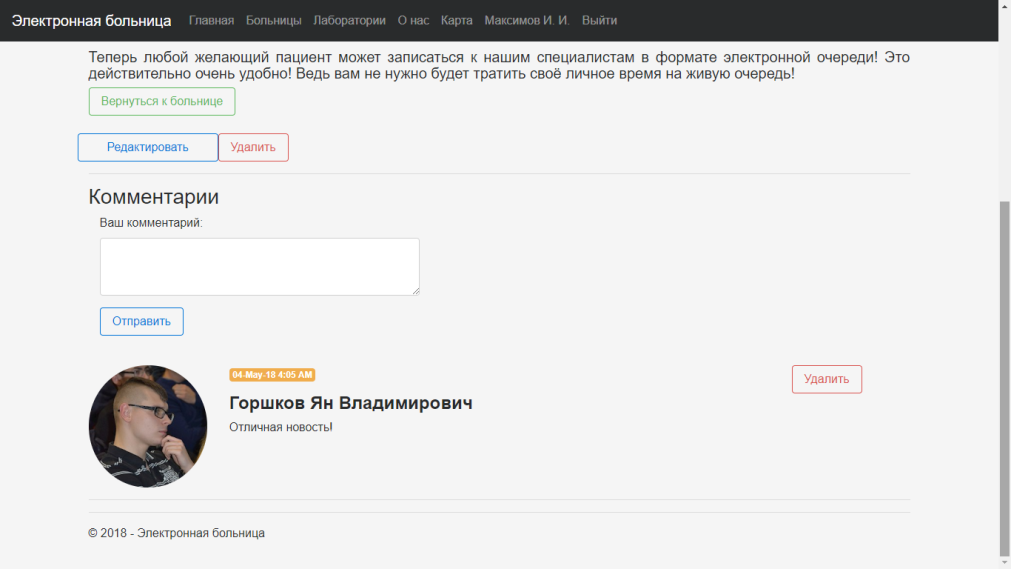


Рисунок 5.22 – Страница новости для модератора

При нажатии на кнопку "Редактировать", с помощью JavaScript и библиотеке jQuery, будет открыта форма с текущими данными новости.

$("#editNews").click(function () {

var form = $("#editForm");

if (form.css("display") == "none")

form.show("quick");

else form.hide("quick");

});

После нажатия на кнопку "Сохранить", веб вызывает метод EditNews. Форма редактирования новости изображена на рисунке 5.23.

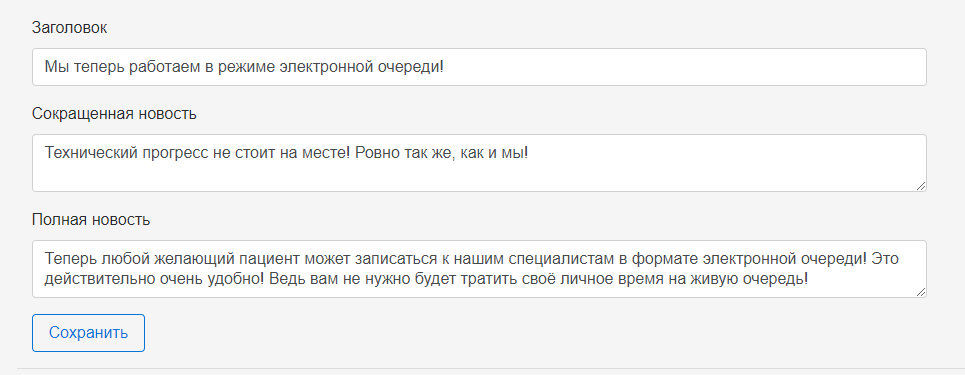


Рисунок 5.23 – Форма редактирования новости

На главной странице больницы модератору доступна кнопка, которая открывает форму редактирования информации больнице. Модератор редактирует такие поля, как: название клиники, описание и контактный телефон. Форма редактирования информации о клинике изображена на рисунке 5.24.

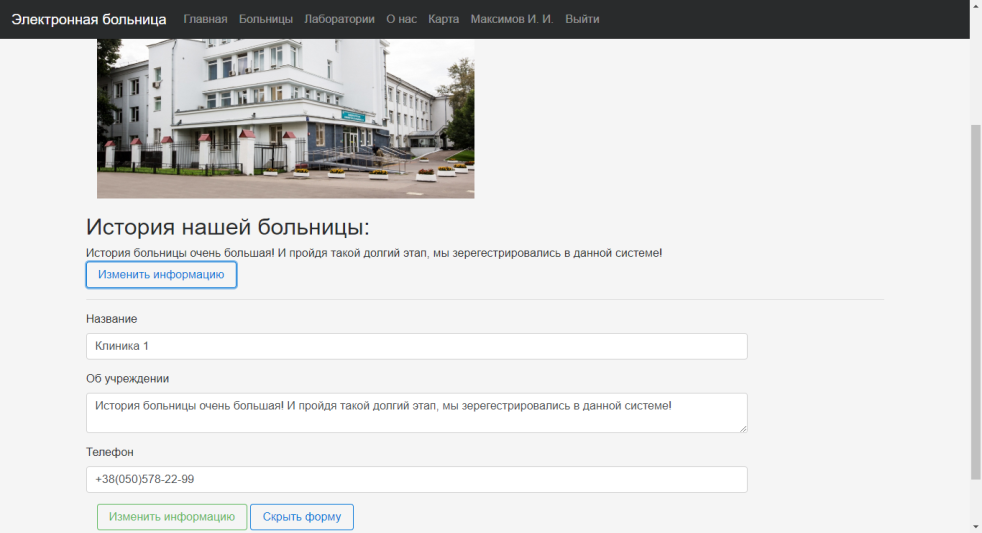


Рисунок 5.24 – Форма редактирования информации о клинике

**5.5. Функціональна частина лікаря**

Главная задача врача - обслуживание пациентов. Под обслуживанием понимается прием пациентов с электронным очереди и добавления выводов врача в медицинскую карту пациентв. За функциональность врача отвечает класс DoctorController. Личный кабинет врача изображен на риснке 5.25.

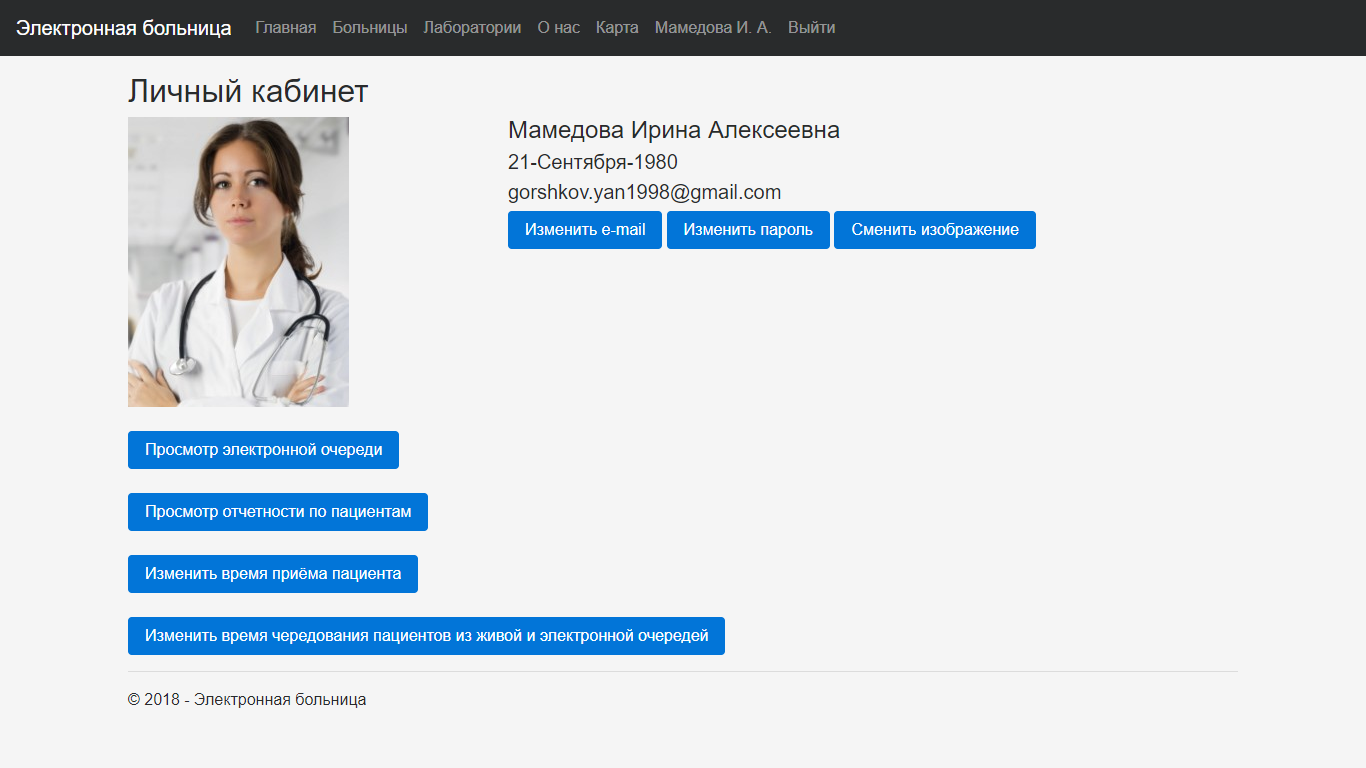


Рисунок 5.25 – Личный кабинет врача

Для просмотра электронной очереди пациентов нужно нажать соответствующую кнопку. Метод ShowAppointments контроллеру Doctor отправит запрос к базе данных и сгенерирует вид таблице пациентов, дня, и времени записи. При необходимости врач может отменить запись нажав красную кнопку. Страница просмотра качественные электронные очереди отражена на рисунке 5.26.

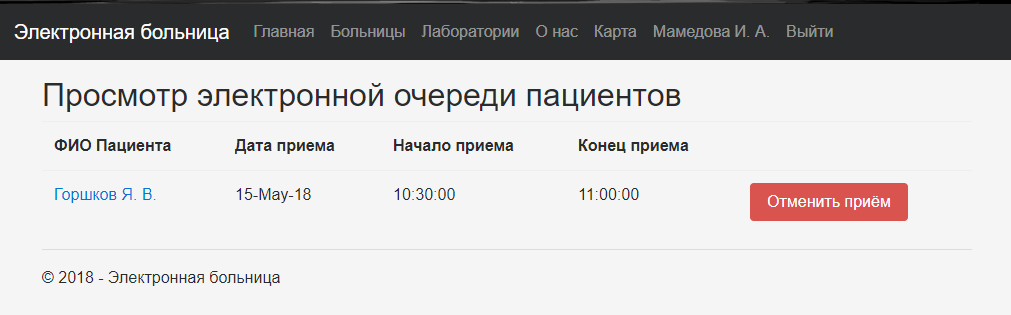


Рисунок 5.26 – Страница просмотра электронной очереди

ФИО пациента в каждом стоврчику представляет собой ссылку в анкету пациента. Анкета содержит в себе такие данные, как: ФИО пациента, электронный почтовый адрес, дату рождения, и список жалоб на момент записи. В зависимости от жалоб, система строит Столбиковая диаграмму с тремя вероятнее диагнозами. Далее, врач должен заполнить следующие поля: жалобы на момент приема, симптомы, по нужности температуру тела и давление, выбрать из списка или добавить новый диагноз, написать заключение врача, рекомендации и пирзначення. Страница с анкетой пациента показано на рисунке 5.27.

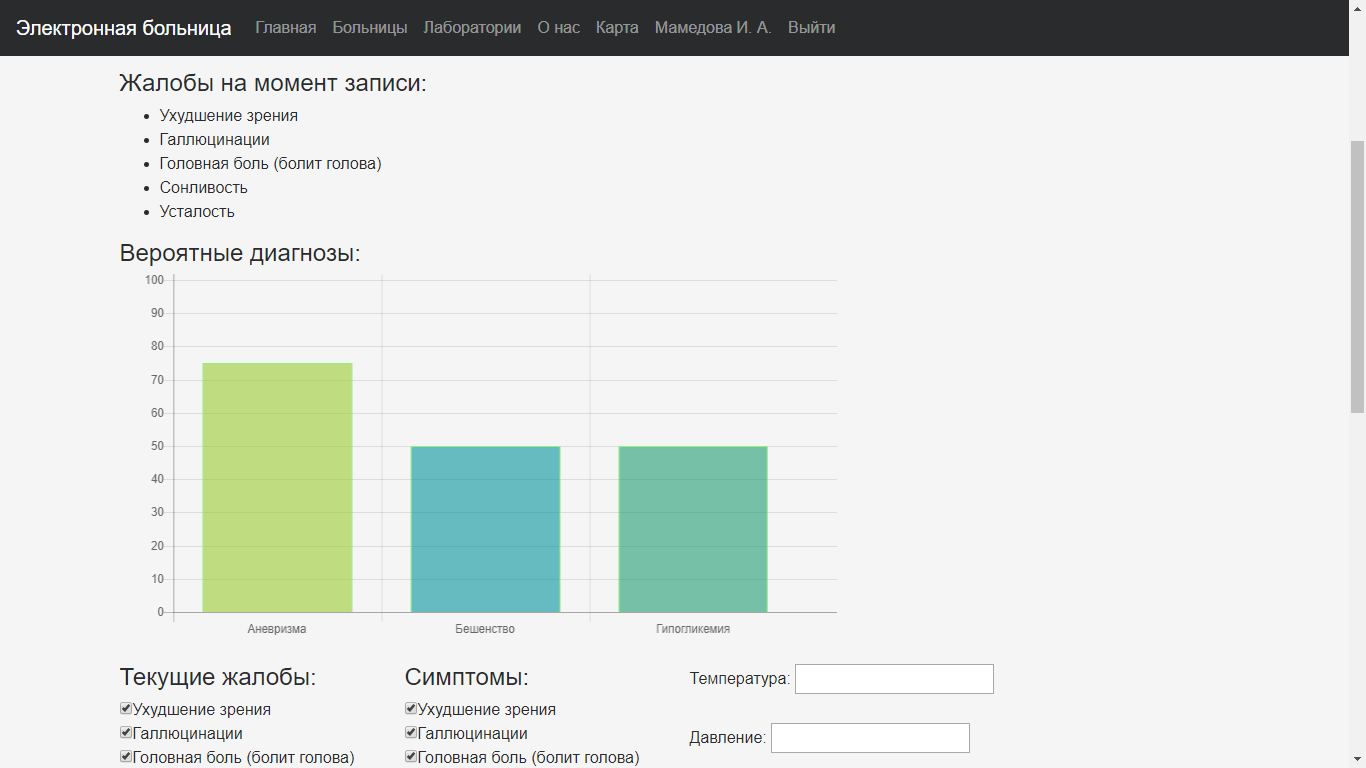


Рисунок 5.27 - Страница просмотра анкеты пациента

При необходимости, врач, к которому был записан пациент может посмотреть историю болезни пациента для выяснения определенных деталей. Медицинская карта пациентов состоит из выводов врачей, и выводов лаборатории. С помощью метода GetPatientReports контроллера Doctor происходит запрос к базе данных для получения этих выводов.

ViewBag.DoctorReports=db.DoctorReports

.Where(dr => dr.PatientId == idPatient)

.OrderByDescending(dr=>dr.Date)

.ToList<DoctorReport>();

ViewBag.LabaratoryReports = db.LabaratoryReports

.Where(dr => dr.PatientId == idPatient)

.OrderByDescending(dr => dr.Date)

.ToList<LaboratoryReport>();

После выполнения запроса, веб-сервером будет сгенерирован вид с двумя вкладками. Каждая вкладка будет содержит в себе все выводы отсортированы по дате. Страница с историей болезней отражена на рисунке 5.28.

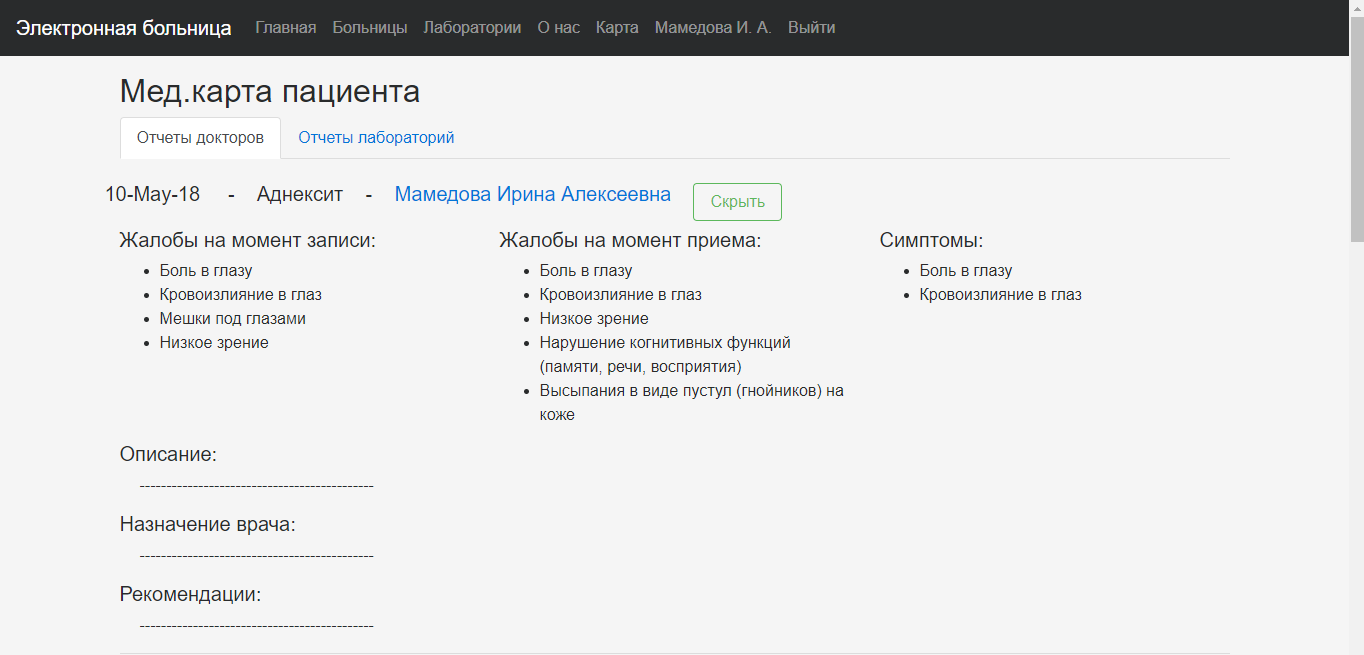


Рисунок 5.28 – Страница с историей болезни пациента

Следующая функция врача - перезапись пациента к другим врачам определенной специальности. Для этого врач должен перейти к странице нужного врача, и нажать на кнопку «Записать пациента на прием". После этого все данные записи пациента будут перенесены новом врачу. Кнопка перезапись пациента к другому врачу изображена на рисунке 5.29.

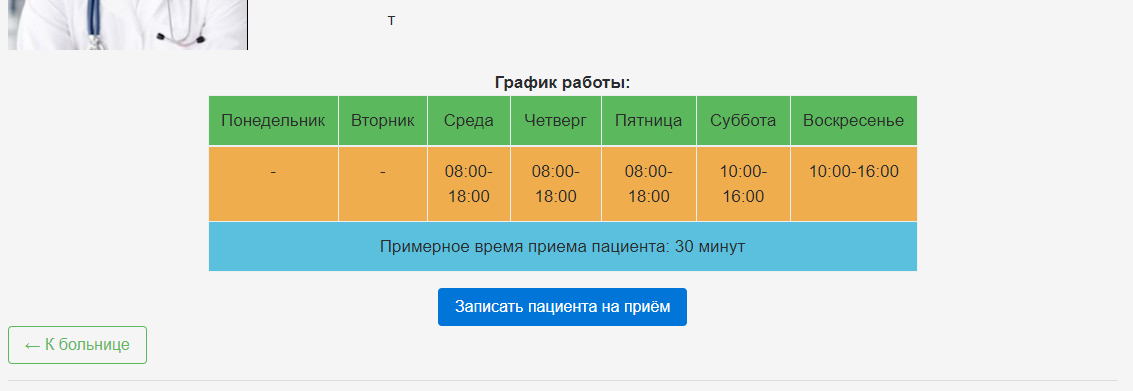


Рисунок 5.29 – Кнопка перезапись пациента к другому врачу

**5.6. Функціональна частина пацієнта**

Пациент является главным типом пользователей, для которого и разробляеться данная система. Пациента имеет возможность электронного записи к врачу, хранение истории болезней, и возможность комментировать Новон медицинских учреждений. За функциональность пациента соответствует класс PatientController.

Личный кабинет пациента предоставляет доступ к просмотру мадичнои карточки пациента, и пергляду записей к врачу и к анализа лаборатории. В случае, если был заключен договор с семейным врачом, ссылку на его страницу будет расположен у фотографии пользователя. Личный кабинет отражены на рисунке 5.30.

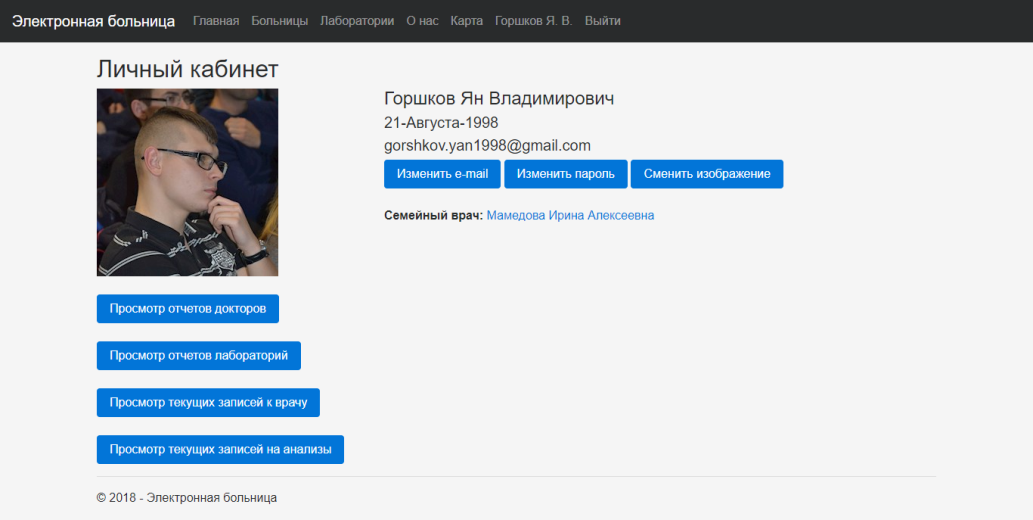


Рисунок 5.30 - Личный кабинет пациента

Если специальность врача позволяет пациенту осуществлять запись, то под графиком работы будет расположена кнопка. Если врач является семейным, то перед записью в него необходимо скалсты договор. В случае, когда договор был составленным с другим семейным врачом, возможность скалсты договор с новым появится только после расторжения прошлого. После этого к врачу можно осуществлять запись. Страница семейного врача изображена на рисунке 5.31.

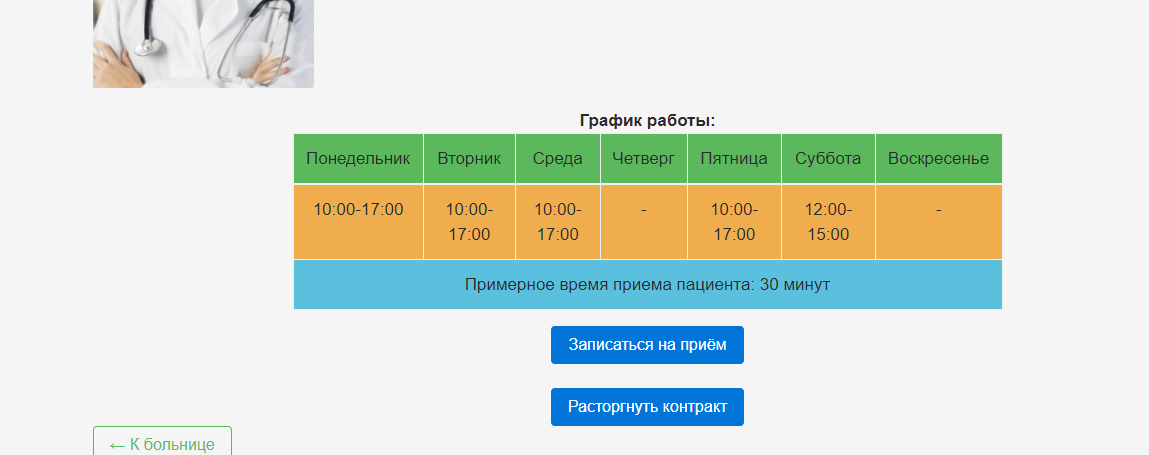


Рисунок 5.31 – Страница семейного врача

При здийснюванни записи к врачу необходимо выбрать из списка жалобы на здоровье. После выбора, пациент указывает желаемый для записи день. В зависимости от графика работы врача и его занятости, будет скалден список доступных для записи времен. За функциональность электронной очереди соответствует класс AppointmentController. Страница записи к врачу показано на рисунке 5.32.

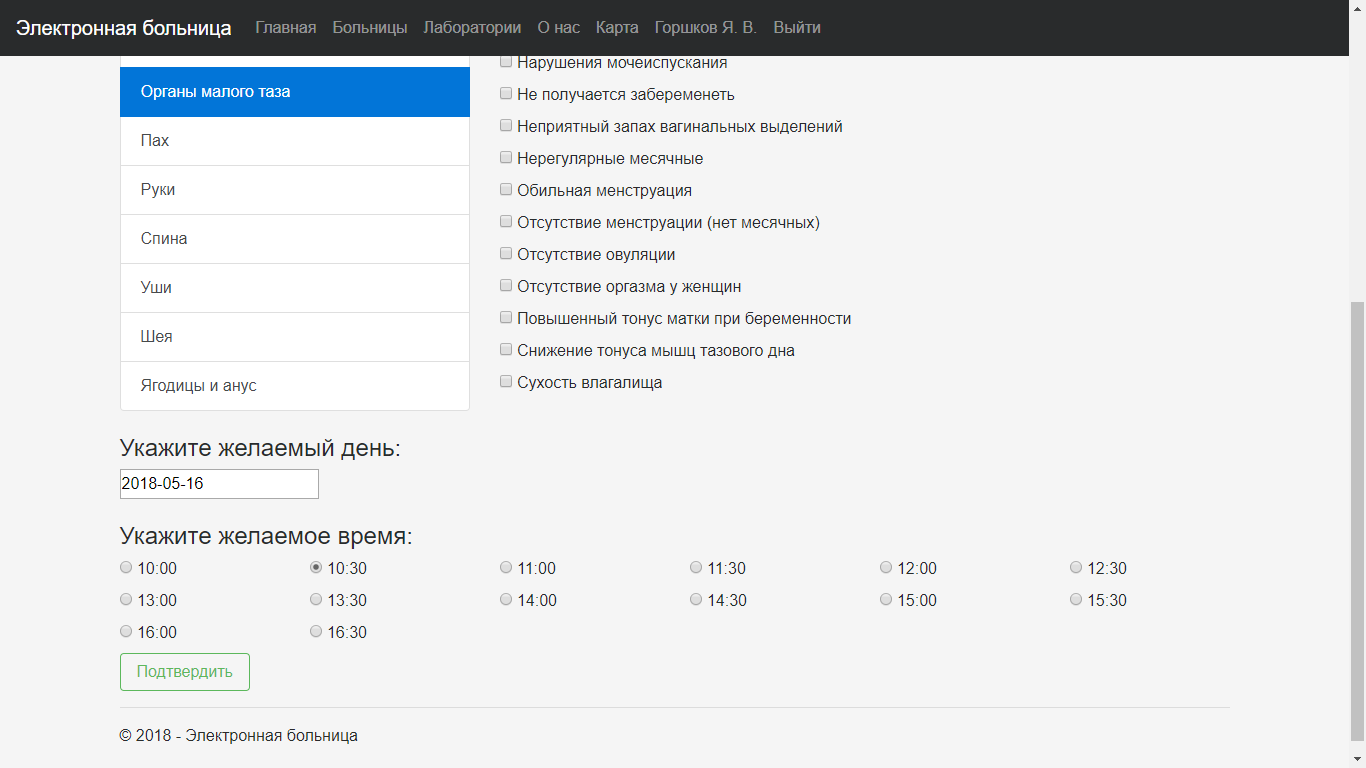


Рисунок 5.32 – Страница записи к врачу

После записи к врачу через личный кабинет пациент способен посмотреть все текущие записи к врачу, избранные жалобы, день и время. Если пациент не сможет прийти к определению времени, он сможет отменить запись к врачу. Страница перегяду текущих записей показано на рисунке 5.33.

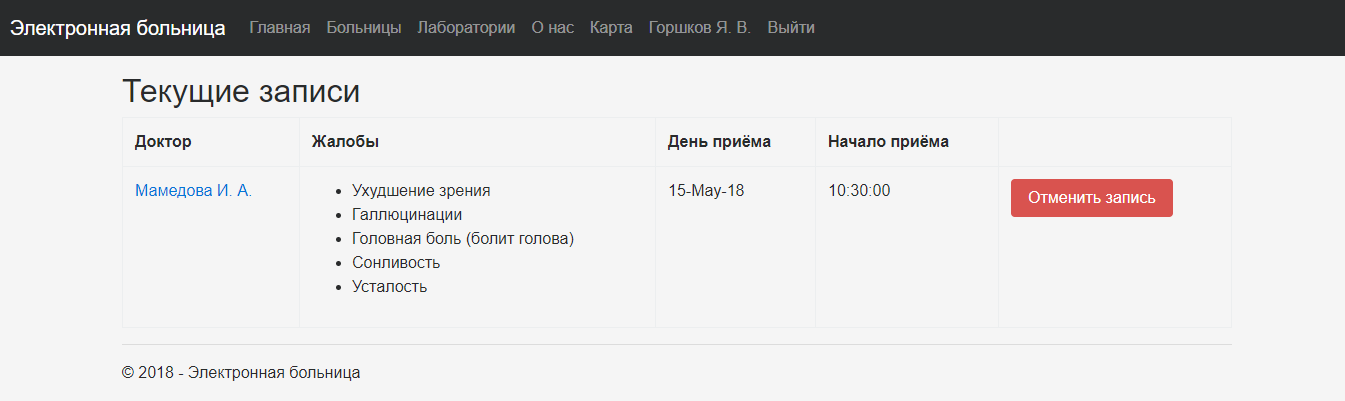


Рисунок 5.33 – Страница просмотра текущих записей к врачу

Система позволяет хранить и просматривать пациенту выводы приема к врачу. Этот вывод состоит из информации о враче, диагноз, избранные жалобы и определенные симптомы. Если во время приема был измерен давление и тмпература, они также отображаются пользователю. Страница с заключением врача показано на рисунке 5.34.

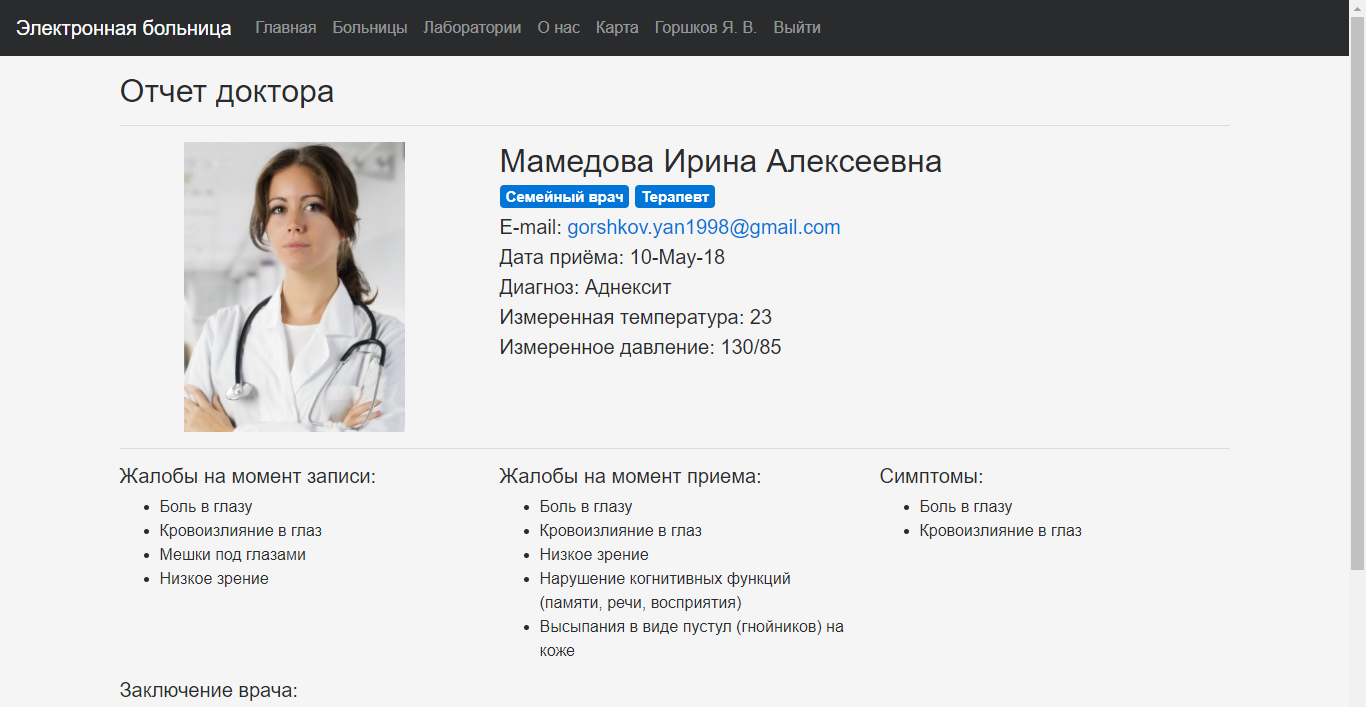


Рисунок 5.34 – Страница просмотра заключения врача