

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМТО»

Факультет СПО

Направление подготовки (специальность) 09.02.07 Информационные системы и
программирование

О Т Ч Е Т
о преддипломной практике

Тема задания: Разработка прототипа клиентской части автоматизированной
информационной системы для клиники «Стоматология Доктора Вознесенского»

Обучающийся Кунаев Н.Д. Группа У2432
(Фамилия И.О.) (номер группы)

Согласовано:

Руководитель практики от профильной
организации:

Костяев Александр Александрович,
заместитель генерального директора ООО
«АКАДЕМИЯ ЦИФРОВОЙ КУЛЬТУРЫ»

(Фамилия И.О., должность и место работы)

Руководитель практики от Университета
ИТМО:

Говорова Марина Михайловна,
преподаватель факультета СПО

(Фамилия И.О., должность и место работы)

Практика пройдена с оценкой «отлично»

«19» мая 2021г.

Санкт-Петербург
2021

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет СПО

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ
на преддипломную практику

Студент	<u>Кунаев Н.Д.</u> (Фамилия И. О.)	Группа №	<u>У2432</u>
Руководитель от профильной организации	<u>Костяев Александр Александрович, заместитель генерального директора ООО «АКАДЕМИЯ ЦИФРОВОЙ КУЛЬТУРЫ»</u> (Фамилия И. О., место работы, должность)		
Руководитель от Университета ИТМО	<u>Говорова Марина Михайловна, преподаватель факультета СПО</u> (Фамилия И. О., место работы, должность)		
Тема задания:	<u>Разработка клиентской части автоматизированной информационной системы для клиники «Стоматология Доктора Вознесенского»</u>		
Сроки прохождения практики:	<u>22.04.2021 – 19.05.2021</u>		
Место прохождения практики:	<u>ООО «АКАДЕМИЯ ЦИФРОВОЙ КУЛЬТУРЫ»</u>		
Должность практиканта:	<u>-</u>		

1. Виды работ и требования к их выполнению:

Производственная практика проводится непрерывно на базе ООО «АКАДЕМИЯ ЦИФРОВОЙ КУЛЬТУРЫ». В ходе прохождения практики выполняются следующие виды работ:

1. Вводный инструктаж по технике безопасности и общим целям, и задачам практики.
2. Анализ поставленной задачи: уточнение требований технического задания.
3. Детальное проектирование системы/приложения/программы.
4. Реализация, отладка и тестирование системы/приложения/программы.
5. Формирование отчетных материалов и представление программы.

2. Виды отчетных материалов и требования к их оформлению:

По результатам прохождения практики составляется отчет, в котором представляется анализ поставленной задачи, описание процесса проектирования системы/приложения/программы, модель системы/приложения/программы и техническое задание. Программа представляется руководителю(ям) практики. Оформление отчета должно соответствовать Рекомендациям по оформлению технических документов факультета СПО Университета ИТМО.

3. ПЛАН-ГРАФИК

№ эта па	Наименование этапа	Срок завершения этапа	Виды работ	Форма отчетности
1	Вводный инструктаж	22.04.2021 – 25.04.2021	Ознакомление с инструкцией по технике безопасности. Ознакомление с целями и задачами производственной практики	Журнал по технике безопасности
2	Анализ поставленной задачи	26.04.2021 – 02.05.2021	Анализ технического задания. Уточнение требований (при необходимости).	Отчет по практике. Дневник прохождения практики.
3	Детальное проектирование системы/приложения/программы.	03.05.2021 – 16.05.2021	Детальное проектирование модулей (составных частей программы).	Отчет по практике. Дневник прохождения практики.
4	Реализация, отладка и тестирование системы/приложения/программы.		Реализация, отладка и тестирование системы/приложения/программы (доработка прототипа).	Отчет по практике. Дневник прохождения практики.
5	Подготовка отчетных материалов и представление программы	17.05.2021 – 19.05.2021	Формирование отчета о прохождении практики, демонстрация работы системы/приложения/программы.	Отчет по практике. Дневник прохождения практики.

4. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТНОСТИ

Наименование формы отчетности	Требования к оформлению
Отчет о практике	Отчет о преддипломной практике оформляется в соответствии с Методическими указаниями по преддипломной практике по специальности 09.02.07 Информационные системы и программирование и Рекомендациям по оформлению технических документов факультета СПО Университета ИТМО. Титульный лист отчета оформляется в соответствии с Приложением В Методических указаний.
Индивидуальное задание	Индивидуальное задание оформляется в соответствии с Методическими указаниями по преддипломной практике по специальности 09.02.07 Информационные системы и программирование (Приложение А) и включается в отчет о преддипломной практике после титульного листа.
Дневник практики	Дневник практики оформляется в соответствии с Методическими указаниями по преддипломной практике по специальности 09.02.07 Информационные системы и программирование (Приложение Б).
Аттестационный лист	Аттестационный лист оформляется в соответствии с Методическими указаниями по преддипломной практике по специальности 09.02.07 Информационные системы и программирование (Приложение Г)

Задание выдано:

Руководитель практики от Университета ИТМО _____ / М. М. Говорова
(подпись руководителя) ФИО

«22» апреля 2021г.

Согласовано:

Руководитель практики от профильной организации _____ / А. А. Костяев
(подпись руководителя) ФИО

«22» апреля 2021г.

Задание принял к исполнению:

_____ / Н. Д. Кунаев
(подпись обучающегося) ФИО

ВВЕДЕНИЕ

Целью преддипломной практики является разработка веб-приложения для медицинской клиники «Стоматология Доктора Вознесенского» Фрунзенского района Санкт-Петербурга. Практика проходит на базе ООО «Академия цифровой культуры».

Для достижения цели, поставленной в рамках преддипломной практики, необходимо выполнить следующие задачи:

1. Анализ технического задания;
2. Уточнение функциональных требований (при необходимости);
3. Программная реализация системы (клиентской части);
4. Проведение отладки и тестирования системы;
5. Формирование технического описания разработанной программной системы;
6. Представление реализованной программной системы.

Система позволяет хранить информацию о пациентах, о предоставленных услугах и о расписании работы врачей.

Отчёт содержит следующие разделы:

1. Введение;
2. Раздел 1 Общие сведения о разработке;
3. Раздел 2 Функциональное назначение;
4. Раздел 3 Описание логической структуры;
5. Раздел 4 Сведения о входных и выходных данных;
6. Раздел 5 Используемые технические средства;
7. Раздел 6 Вызов и загрузка программы;
8. Заключение;
9. Список источников;
10. Приложение А Техническое задание;
11. Приложение Б Модель системы.

1 ОБЩИЕ СВЕДЕНИЯ О РАЗРАБОТКЕ

1.1 Наименование программы

АИС для медицинской клиники «Стоматология Доктора Вознесенского» (клиентская часть).

1.2 Программное обеспечение, необходимое для функционирования программы

Для функционирования программы необходимы следующие веб-браузеры:

- Google Chrome, версии 90.0.4430.212+;
- Mozilla Firefox Developer Edition, версии 89.0 x64+;
- Microsoft Edge, версии 90.0.818.56+;
- Safari, версии 14.1+;
- Opera, версии 76.0.4017.123+.

1.3 Программные средства

В приложении использованы парадигмы реактивного, декларативного, объектно(прототипно)-ориентированного и функционального программирования.

Для реализации были использованы:

1. Во время разработки клиентской части:

- a. NodeJs Server, версии 14.15.4;
- b. Пакетный менеджер Yarn, версии 1.22.10;
- c. Метаязык SASS, версии 1.32.5;
- d. ReactJS, версии 17.0.1;
- e. Bootstrap, версии 4.6.0;
- f. React-bootstrap, версии 1.5.0.

2. В продакшн версии веб-приложения:

- a. HTML, версии 5.3;

- b. CSS, версии 2.1;
- c. JS, версии ES9 (ECMAScript 2018).

2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

2.1 Описание решаемых задач

Система предоставляет различный функционал для разных групп пользователей (директор/администратор/врач).

Врач может просматривать своё расписание, просматривать и вносить изменения в медицинские карты пациентов.

Администратор может делать то, что может делать врач, а также вносить изменения в расписание, добавлять новых пациентов, вписывать пациентов в расписание врача (записывать на приём).

Директору доступен весь функционал других категорий пользователей, а также он может добавлять новые услуги, изменять их цену, добавлять новых пользователей, изменять любые данные пользователей, назначать рабочие дни для администраторов и врачей.

Просмотр расписания реализован посредством таблицы, отображающей календарь с метками врачей, работающих в определённые дни.

2.2 Описание функций

При входе в веб-приложение открывается форма авторизации (рисунок 1). После ввода корректных данных (логин и пароль) и нажатия на кнопку «Войти», исходя из введенных данных, определяется к какой группе прав относится пользователь, в зависимости от неё пользователь будет видеть разные варианты интерфейса (пока реализован только интерфейс директора). Форма входа показана на рисунке 1.



Стоматология доктора Вознесенского

Имя пользователя

Пароль

Войти

Рисунок 1 – Окно аутентификации пользователей

После удачной аутентификации происходит переадресация на главную страницу, на которой можно увидеть меню навигации, позволяющее перейти в другие разделы веб-приложения.

Главная страница представлена на рисунке 2.



Рисунок 2 – Страница «Главная»

Перейдя по ссылке «Пациенты», открывается окно со списком всех пациентов, которое показано на рисунке 3.

<div>Главная</div> <div>Пациенты</div> <div>Визиты</div> <div>Услуги</div> <div>Сотрудники</div> <div>Статистика</div> <div>Выйти</div>	Пациенты				<div>↺</div> <div>+</div>
	Бавилов Василий Викторович	0	+79991110022	2001-12-17	✓
	Писарев Александр Дмитриевич	19990	+79992220022	2001-05-17	🗑
	Вав Дав Дим	10000	+7999888776655	2000-01-01	🗑
	Вав Дав Дми	0	+79992223311	2000-01-01	🗑
	Ава Дак Фор	-100	+79992223311	1900-01-01	🗑
	Дмитриев Виктор Степанович	-1000	+7911911111111	2001-01-01	🗑
	Пак Пак Ман	0	Нет	2001-02-03	✓
	Вик Виктор Владимирович	0	3123123123	1990-01-01	🗑
	Дик Виктор Владимирович	0	3123123123	1990-01-01	✓
	Тестов Тест Тестовна	0	8999111002233	2001-02-03	🗑

Рисунок 3 – Страница «Пациенты»

При нажатии на строку с конкретным пациентом открывается модальное окно изменения, которое представлено на рисунке 4.

The screenshot displays a web application interface for managing patients. On the left is a dark sidebar with icons for 'Главная' (Home), 'Пациенты' (Patients), 'Визиты' (Visits), 'Услуги' (Services), 'Сотрудники' (Staff), 'Статистика' (Statistics), and 'Выйти' (Logout). The main area is titled 'Пациенты' and contains a table of patient records. A modal window titled 'Пациент' is open, showing the details for a patient named 'Писарев Александр Дмитриевич'. The form includes fields for various personal and identification details.

Идентификатор	2
Номер паспорта	4040112233
ФИО	Писарев Александр Дмитриевич
Дата Рождения	2001-05-17
Адрес Регистрации	Самара
Адрес проживания	Самара
Профессия	Слесарь
Телефоны	+79992220022
Пол	Мужчина
Место рождения	Казань
Статус архивации	да/нет
Причина архивации	
Откуда узнал	От знакомых
Адрес электронной почты	test@gmail.test
Комментарий	null
Комментарий	null
Базис	10000

In the background, a table of patients is visible, with the following data:

Пациент	Дата	Статус
Вавилов Василий Викторов	2001-12-17	✓
Писарев Александр Дмитриевич	2001-05-17	✗
Вав Дав Дим	2000-01-01	✗
Вав Дав Дми	2000-01-01	✗
Ава Дак Фор	1900-01-01	✗
Дмитриев Виктор Степанов	2001-01-01	✗
Пак Пак Ман	2001-02-03	✓
Вик Виктор Владимирович	1990-01-01	✗
Дик Виктор Владимирович	1990-01-01	✓
Тестов Тест Тестовна	2001-02-03	✗

Рисунок 4 – окно изменения пациент

При нажатии на кнопку «+» появляется форма добавления нового пациента, которая представлена на рисунке 5.

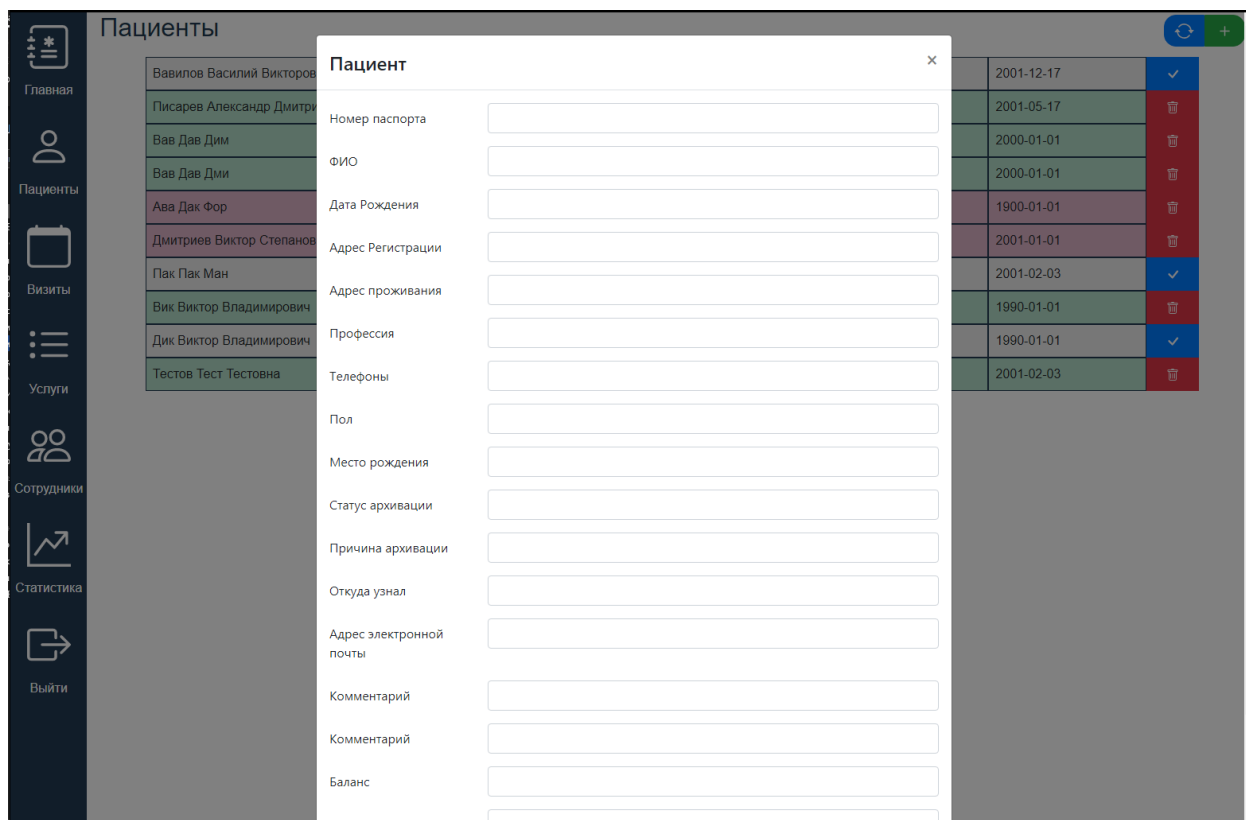


Рисунок 5 – окно добавления пациентов

При переходе по ссылкам «Визиты» и «Услуги» выводятся пустые страницы с ещё нереализованным функционалом, которые показаны на рисунках 6 и 7.

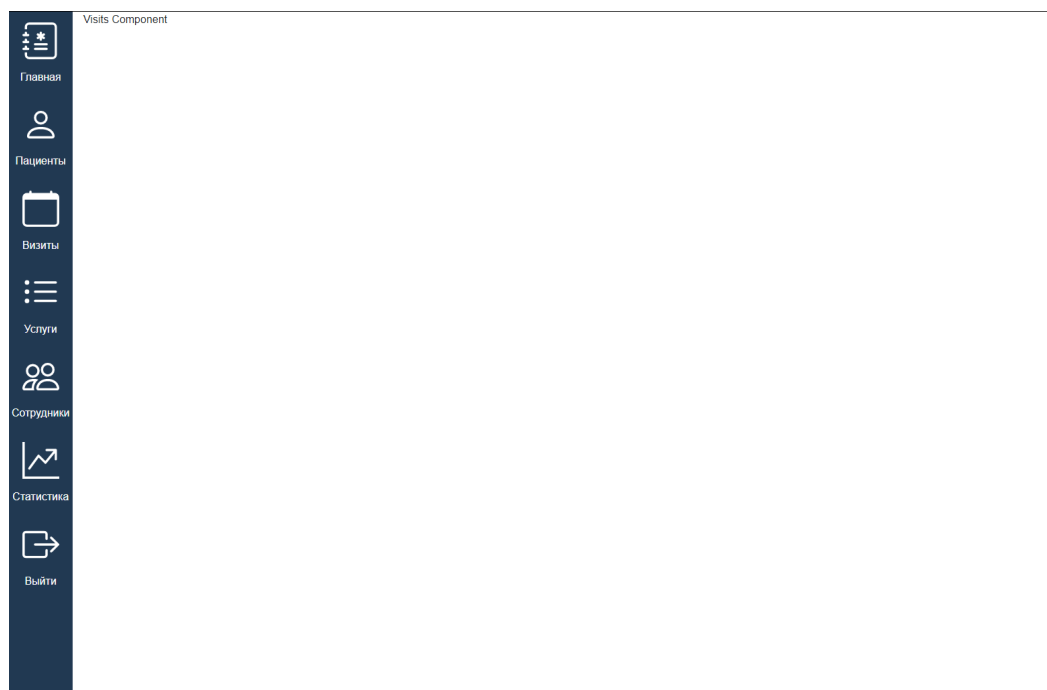


Рисунок 6 – Страница «Визиты»

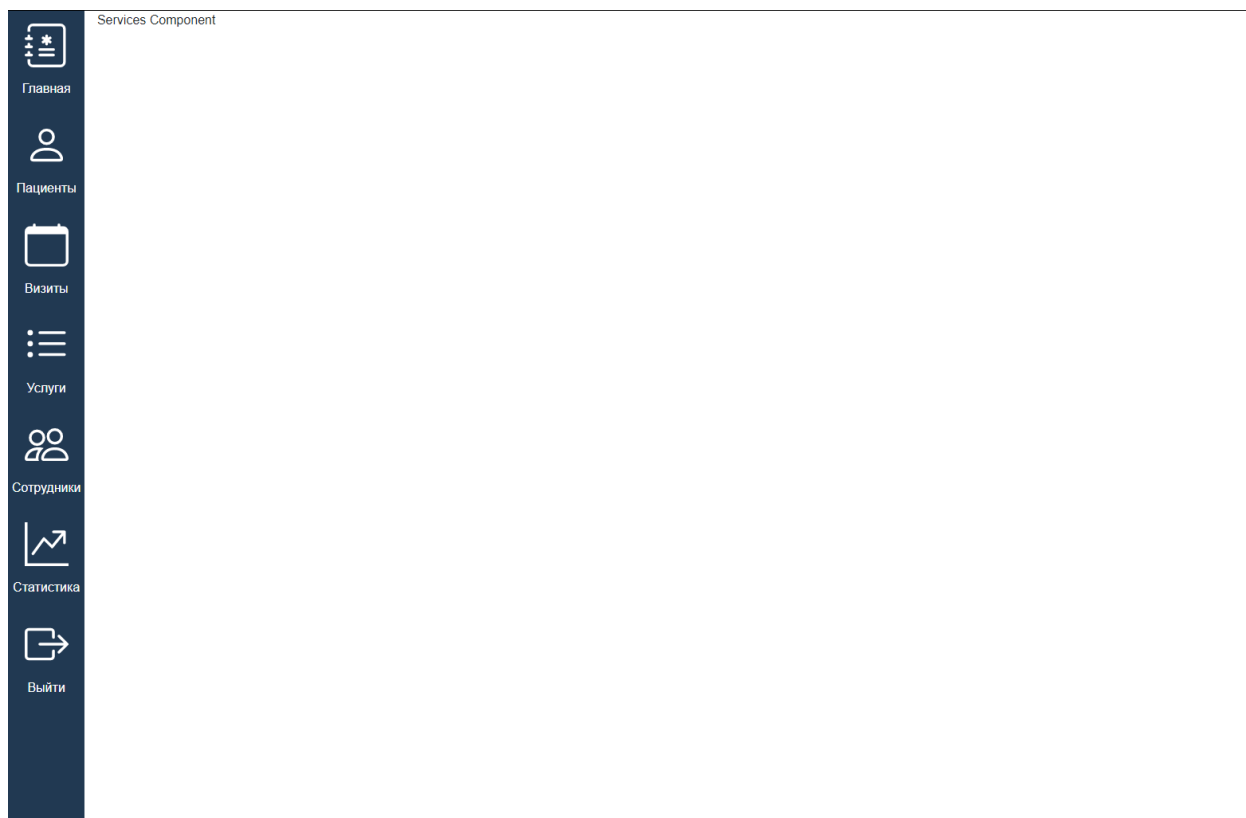


Рисунок 7 – Страница «Услуги»

При переходе по ссылке «Сотрудники» открывается список всех зарегистрированных сотрудников, представленный на рисунке 8.

Сотрудники				
Пак Пак Ман	test@gmail.test	Доктор	+79991110022	2000-02-03
ПакМан Александр	test@gmail.test	Хирург	+79992220022	2000-02-03
Т Т Т		Т	0	1990-01-01

Рисунок 8 – Страница «Сотрудники»

При нажатии на строку с конкретным пациентом открывается модальное окно изменения, которое представлено на рисунке 9.

The screenshot shows a web application interface. On the left is a dark sidebar with icons and labels: Главная, Пациенты, Визиты, Услуги, Сотрудники, Статистика, and Выйти. The main area has a title 'Сотрудники' and a table with 5 columns. The table contains three rows of employee data. A modal window titled 'Пациент' is open in the center, containing a form with fields for various patient details and two buttons at the bottom: 'Сохранить' (green) and 'Отменить' (red).

Имя	Email	Профессия	Телефон	Дата рождения
Пак Пак Ман	test@gmail.test	Доктор	+79991110022	2000-02-03
ПакМан Александр	test@gmail.test	Хирург	+79992220022	2000-02-03
Т Т Т		Т	0	1990-01-01

Пациент		×	
Идентификатор	1		
Профессия	<input type="text" value="Доктор"/>		
ФИО	<input type="text" value="Пак Пак Ман"/>		
Дата Рождения	<input type="text" value="2000-02-03"/>		
Адрес проживания	<input type="text" value="Россия"/>		
Телефоны	<input type="text" value="+79991110022"/>		
Адрес электронной почты	<input type="text" value="test@gmail.test"/>		
<input type="button" value="Сохранить"/>		<input type="button" value="Отменить"/>	

Рисунок 9 – окно изменения сотрудников

При нажатии на кнопку «+» появляется форма добавления нового сотрудника, которая представлена на рисунке 10.

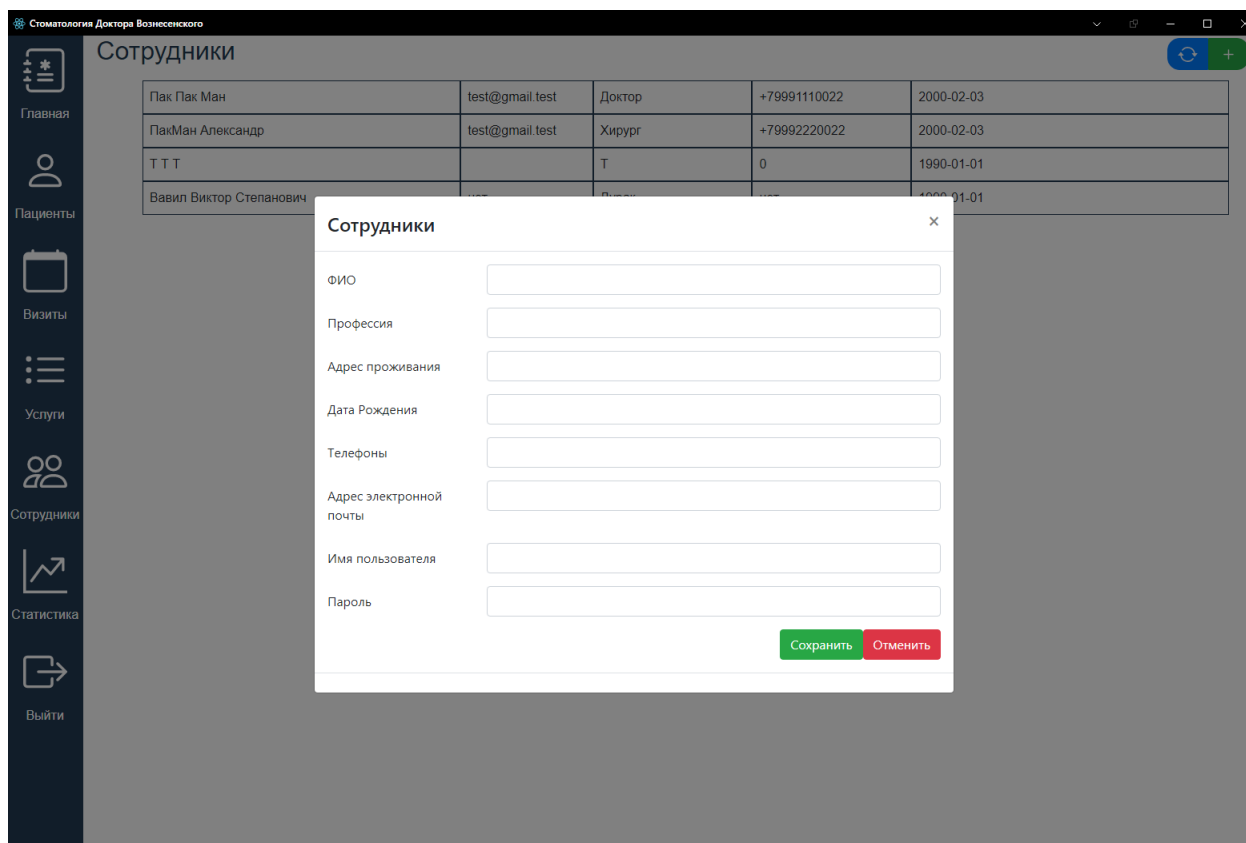
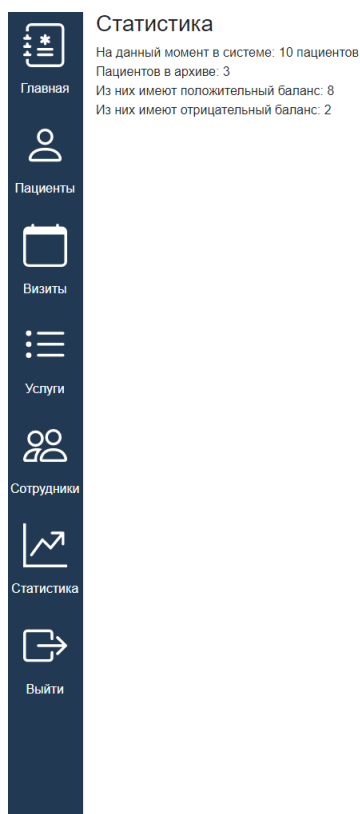


Рисунок 10 – окно добавления сотрудников

При переходе по ссылке «Статистика» открывается окно с базовой статистикой, которая представлена на рисунке 11.



2.3 Средства контроля правильности выполнения

Поскольку доступ к веб-приложению есть только посредством локальной сети, то снижается общий уровень уязвимости системы и хранящихся данных.

2.4 Ограничения области применения программы

Программа может быть использована только в медицинской клинике «Стоматология Доктора Вознесенского» (Санкт-Петербург, Фрунзенский район).

3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1 Алгоритм программы

Функциональная модель представлена в BPMN-диаграмме, на рисунке Б.1, включающей основной бизнес-процесс, который описывает процесс записи пациентов на приём, который включает в себя проведение лечения и оплаты. Описание поведения пользователей представлено на рисунке Б.2 в формате «Пользовательская история» (User story).

3.2 Используемые методы при разработке программы

В процессе разработки системы были использованы:

- git – для контроля версий файлов, версии 2.29.2.windows.3;
- https – протокол для обмена данными между клиентом и сервером;
- json – универсальный формат данных;
- yarn – пакетный менеджер, версии 1.22.10;
- sass – мета-язык для каскадных таблиц стилей (css), версии 1.32.5;
- html – стандартизированный язык разметки веб-документов, версии 5.3;
- ESLint – линтер для JavaScript, статический анализатор кода, версии 0.2.1.

При разработке были использованы следующие паттерны проектирования:

- Простые компоненты (stateless component);
- Условный рендеринг (conditional rendering);
- Типы потомков (children types);
- Массив как потомок (array as children);
- Функция как потомок (function as children);
- Функция в render (render callback);
- Перенаправление компонента (proxy component);
- Компоненты высшего порядка (higher-order component);

- Передача свойств вниз по дереву компонентов (passing down props);

После разработки производится сборка и компиляция dev-проекта в продакшн билд, который более оптимизирован для браузеров и работы в интернете: все js- и css-/sass-файлы собираются в набор минимизированных файлов, чтобы браузер загружал сразу весь функционал и не осталось не подгруженных файлов.

3.3 Структура программы

В таблице 1 приведена структура клиентской части веб-приложения до продакшн сборки.

Таблица 1 – Структура клиентской части приложения

Файл	Функции	Назначение
1	3	2
Auth.jsx	Auth	Функция, возвращающая визуальный компонент окна авторизации
	Auth.setField	Функция, для логирования изменений поля и записи данных в структуру
	Auth.setAuthorization	Функция, устанавливающая флаг авторизации
Auth.module.sass	-	Файл стилей
SideBar.jsx	SideBar	Возвращает визуальный компонент бокового меню
SideBar.module.sass	-	Файл стилей

Продолжение таблицы 1

1	2	3
Client.jsx	Client	Функция, возвращающая визуальный интерактивный компонент с информацией о пациенте
	Client.archiveClient	Функция архивации клиентов
	Client.dearchiveClient	Функция разархивации клиентов
ClientList.jsx	ClientList.componentDidMount	Функция, запускающаяся в момент загрузки компонента Client, которая вызывает функцию getClients
	ClientList.getClients	Функция, получающая данные пользователей из Базы Данных по средствам API
	ClientList.render	Функция, возвращающая разметку списка клиентов и передающая данные в класс Client
ClientList.module.sass	-	Файл стилей
ClientAddModalWrapper.jsx	ClientAddModalWrapper	Функция-обёртка для модальных окон, добавляющая возможность обновления и добавления клиентов
ClientEditModal.jsx	ClientEditModal	Функция, которая возвращает модальное окно с формой изменения существующих пациентов
	ClientEditModal.setField	Функция, для логирования изменений поля и записи данных в структуру
	ClientEditModal.handleSubmit	Функция, собирающая данные и проверяющая на наличие ошибок
	ClientEditModal.findFromErrors	Функция поиска ошибок во введённых данных

Продолжение таблицы 1

1	2	3
ClientAddModalClass.jsx	ClientAddModal	Функция, которая возвращает модальное окно с формой добавление пациентов
	ClientAddModal.setField	Функция для логирования изменений поля и записи данных в структуру
	ClientAddModal.handleSubmit	Функция, собирающая данные и проверяющая на наличие ошибок
	ClientAddModal.findFromErrors	Функция поиска ошибок во введённых данных
Employee.module.sass	-	Файл стилей
EmployeesList.jsx	Employee	Функция, возвращающая интерактивный визуальный компонент со списком сотрудников
EmployeesList.module.sass	-	Файл стилей
EmployeeAddModal.jsx	EmployeeAddModal	Функция, которая возвращает модальное окно с формой добавление сотрудников
	EmployeeAddModal.setField	Функция, для логирования изменений поля и записи данных в структуру
	ClientAddModal.handleSubmit	Функция, собирающая данные и проверяющая на наличие ошибок
	ClientAddModal.findFromErrors	Функция поиска ошибок во введённых данных
EmployeeAddModalWrapper.jsx	EmployeeAddModalWrapper	Функция-обёртка для модальных окон, добавляющая возможность обновления и добавления сотрудников

Продолжение таблицы 1

1	2	3
EmployeeEditModal.jsx	ClientEditModal	Функция, которая возвращает модальное окно с формой изменения существующих сотрудников
	ClientEditModal.setField	Функция, для логирования изменений поля и записи данных в структуру
	ClientEditModal.handleSubmit	Функция, собирающая данные и проверяющая на наличие ошибок
	ClientEditModal.findFromErrors	Функция поиска ошибок во введённых данных
Employee.jsx	Employee	Функция, возвращающая интерактивный визуальный компонент интерфейса, которые выводит таблицу всех сотрудников
SideBar.module.sass	-	Файл стилей
SideBar.jsx	SideBar	Функция, возвращающая разметку меню боковой навигации
Services.module.sass	-	Файл стилей
Stats.jsx		Функция, возвращающая статистические данные
Visits.jsx	Visits	Функция, возвращающая список визитов
Visits.module.sass	-	Файл стилей
WorkingSpace.jsx	WorkingSpace	Функция-маршрутизатор, определяет, что должно быть выведено, в зависимости от адреса
WorkingSpace.module.sass	-	Файл стилей
Services.jsx	Service	Функция, возвращающая список всех предоставляемых услуг
Stats.module.sass	-	Файл стилей
Api.js	axios.create	Обёртка функции из библиотеки axios
App.js	App	Функция-маршрутизатор, которая не позволяет неавторизованным пользователям получать доступ к системе

Продолжение таблицы 1

1	2	3
App.sass	-	Файл стилей
index.css	-	Файл стилей
index.js	ReactDOM.render	Функция, запускающая всё веб-приложение
package.json		Файл, в котором перечислены зависимости
yarn.lock		Файл, в котором перечислены зафиксированные версии зависимостей
.gitignore		Файл, в котором перечислены игнорируемые файлы
public/index.html	-	Файл разметки стартового веб-документа
public/index.css	-	Файл стилей

В таблице 2 представлена структура программы после продакшн сборки.

Таблица 2 – Структура клиентской части веб-приложения

Файл	Функции	Назначение
index.html	-	Файл разметки стартового веб-документа
index.css	-	Файл стилей
static/css/*.*.chunk.css	-	Минимизированные файлы стилей
static/js/*.*.chunk.js	Их слишком много, чтобы это было возможным описать	Минимизированные файлы js, содержащий основной функционал
static/js/*.*.chunk.js.LICENSE	-	Файлы лицензий
static/js/main.*.chunk.js	Их слишком много, чтобы это было возможным описать	Минимизированный файл webpack-скрипта, содержащий фундаментальные настройки сборки окружения
static/js/runtime-main.*.js	Их слишком много, чтобы это было возможным описать	Минимизированный файл js

3.4 Форматы обмена данными

Основной формат обмена данными – обращение к Rest API в виде JSON-запросов.

4 СВЕДЕНИЯ О ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ

Входные и выходные данные приложения описаны в приложения А «Техническое задание» в разделе 3.2.

5 ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Для использования разрабатываемого веб-приложения устройство пользователя должно обладать следующими техническими требованиями, соответствующим требованиям использования основных браузеров (Chrome, Firefox, Internet Explorer, Opera, Safari и т.п.) (URL: <https://support.google.com/chrome/a/answer/7100626>, дата обращения: 18.05.21):

- ОС Windows 7 (поддержка завершится не ранее 15.01.2022), Windows 8, Windows 8.1, Windows 10 или более поздней версии или OS X El Capitan 10.11 или более поздней версии.
- процессор Intel Pentium 4 или более поздней версии с поддержкой SSE3.

6 ВЫЗОВ И ЗАГРУЗКА СИСТЕМЫ

6.1 Конфигурирование и загрузка программы

Развертывание системы производится по следующему алгоритму:

1. Запуск сервера компонента логирования;
2. Запуск сервера компонента хранения;
3. Запуск компонента API;
4. Запуск веб-сервера.

Алгоритм также может быть описан следующей последовательностью команд (при использовании описанных в разделе 5 технических средств):

1. `systemctl influxd start;`
2. `systemctl mysqld start;`
3. `java -jar ./voz-backend.jar;`
4. `systemctl nginx start.`

6.2 Способ вызова и завершения работы программы

Разрабатываемая программа – веб-приложение, поэтому для начала работы требуется перейти на определённый URL, который будет назначен заказчиком на рабочем месте.

Для прекращения работы необходимо покинуть браузер или вкладку, в которой открыто веб-приложение.

6.3 Режимы работы программы

Пользователи взаимодействуют с веб-приложением в интерактивном режиме.

ЗАКЛЮЧЕНИЕ

В результате выполнения преддипломной практики было разработано веб-приложение для учёта пациентов в соответствии с техническим заданием и моделями разработки. Сформировано техническое описание разработанной системы.

Для достижения поставленной цели были выполнены следующее задачи:

1. Уточнение функциональных требований (при необходимости).
2. Программная реализация системы (клиентской части).
3. Проведение отладки и тестирования системы.
4. Формирование технического описания разработанной программной системы.
5. Представление реализованной программной системы.

Таким образом, все задачи решены в полном объёме. Цель практики достигнута.

В качестве перспективы дальнейшего развития приложения можно рассмотреть расширение системы для внедрения в другие частные медицинские клиники.

ПРИЛОЖЕНИЕ А

Техническое задание

ТЕРМИНЫ, ПОНЯТИЯ И СОКРАЩЕНИЯ

В данном разделе представлены основные термины, понятия и сокращения, используемые в настоящем документе – смотреть [Таблица 1].

Таблица 1 – Термины, понятия и сокращения

Термин, понятие или сокращение	Определение
БД	База данных
ОС	Операционная система
ПО	Программное обеспечение
Система	Разрабатываемый программный продукт
СУБД	Система управления базой данных
ТЗ	Техническое задание
GUI	Global User Interface – Глобальный пользовательский интерфейс

1 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Назначением разработки является создание веб-приложения для автоматизации учёта пациентов медицинской клиники «Стоматология Доктора Вознесенского».

Программа должна эксплуатироваться на объекте заказчика. Пользователями программы должны являться администраторы, директор и врачи. Пациенты не должны иметь доступ к системе.

2 ЦЕЛЬ И ЗАДАЧИ, РЕШАЕМЫЕ В ПРОЦЕССЕ ДОСТИЖЕНИЯ ЦЕЛИ

Целью является создание клиент-серверной автоматизированной информационной системы (АИС) для организации работы СК.

Задачи, решаемые в процессе достижения цели:

- уточнение требований к системе;
- детальное проектирование системы;
- программная реализация;
- тестирование.

3 ТРЕБОВАНИЯ К СИСТЕМЕ

3.1 Требования к функциональным характеристикам

Система должна обеспечить хранение информации о клиентах, предоставленных услугах и графике работы сотрудников.

В системе должны быть определены следующие категории пользователей:

- администратор;
- директор;
- врач.

В системе должны быть реализованы следующие функциональные возможности:

- работа с расписаниями сотрудников, изменение графиков работы;
- добавление, удаление и изменение информации о приемах пациентов;
- учет оказываемых клиникой услуг, их классификация по номенклатуре медицинских услуг;
- учет клиентов клиники, изменение и добавление информации о них, добавление клиента в архив и удаление из него;
- возможность ограничения прав.

3.2 Описание входных и выходных данных

Входными данными приложения являются следующие данные.

О пациентах:

- паспортные данные;
- ФИО;
- дата рождения;
- адрес регистрации;
- адрес проживания;
- профессия;

- телефон;
- пол;
- место рождения;
- статус архивации;
- откуда узнал;
- адрес электронной почты;
- комментарий;
- баланс.

О визитах:

- пациент;
- дата;
- причина;
- посещённые врачи;
- комментарий;
- статус.

О сотрудниках:

- ФИО;
- Дата рождения;
- должность;
- контактные данные.
- логин;
- пароль;
- права доступа.

Об услугах:

- цена;
- наименование;
- группа.

О предоставленных услугах:

- услуга;
- посещение, в которое услуга оказана.

3.3 Модель приложения

Функциональная модель поведения приложения должна соответствовать модели на рисунках Б.1.

Модель поведения пользователей должна соответствовать модели, представленной на рисунке Б.2.

Модель прикладного программного интерфейса должна соответствовать модели на рисунке Б.3.

3.4 Эргономические и технико-эстетические требования

При проектировании форм необходимо стремиться к использованию ограниченного набора цветов и уделять внимание их правильному сочетанию. По возможности должны использоваться системные цвета и свойственные системе элементы управления. Набор из трёх цветов продиктован основными цветами логотипа стоматологической клиники.

При неправильно введенных данных пользователю должно быть показано диалоговое окно с ошибкой (язык ошибки в данном случае может быть русским или английским).

Правила валидации полей определяются на стороне клиента.

3.5 Структура системы

Система должна включать в себя следующие компоненты клиентского интерфейса (подсистемы):

- пациенты – модуль, содержащий информацию о клиентах в системе;
- сотрудники – модуль, содержащий информацию о пользователях системы;
- визиты – модуль, содержащий информацию о посещениях пациентов клиники и посещённых врачах;
- услуги – модуль, содержащий информацию о возможных услугах, предоставляемых клиникой;

- статистика – модуль, содержащий статистическую информацию о работе клиники и о клиентах.

3.6 Требования к информационному обеспечению системы

Для обеспечения целостности данных должны использоваться системы валидации форм. Доступ к данным должен быть предоставлен только авторизованным пользователям с учетом их прав доступа.

3.7 Требования к технологиям разработки

При разработке приложения должны использоваться парадигмы реактивного, декларативного, объектно(прототипно)-ориентированного и функционального программирования. Архитектура реализуемой системы – клиент-серверное веб-приложение.

3.8 Требования к программным средствам и технологиям разработки

Компонента клиентского интерфейса должно быть реализовано при помощи следующих средств:

- язык программирования JavaScript (версии ECMAScript 9 или выше);
- язык разметки HTML5;
- язык стилей CSS3;
- сервер для разработки Node.js версии не ниже 14.15.4;
- пакетный менеджер yarn версии не ниже 1.22.10;
- интегрированная среда разработки Web Storm версии 2020.3.3.

Также для реализации потребуются некоторые библиотеки и метаязыки:

- библиотека ReactJs;
- метаязык SASS;

При эксплуатации приложения должны быть установлены:

- любой современный браузер (Mozilla Firefox, Opera, Google Chrome, Safari, Yandex Browser).

3.9 Требования к защите от несанкционированного доступа

Контроль защиты информации от несанкционированного доступа осуществляется на стороне клиента по средствам использования токена авторизации.

Компонент прикладного программного интерфейса должен обеспечивать защиту от несанкционированного доступа:

- аутентификация пользователя;
- проверку полномочий пользователя при работе с системой.

Защищённая часть системы должна использовать пароли, символы должны маскироваться одним типом символов (*).

Директор имеет доступ ко всем функциям системы.

Администраторы не могут вносить изменения в прайс-листы, не может добавлять новых пользователей, не может изменять существующих пользователей.

Врачи могут получить доступ только к информации о пациентах, их визитах и выполненных услугах.

4 ТРЕБОВАНИЯ К МЕТОДАМ ТЕСТИРОВАНИЯ

Разрабатываемое веб-приложение будет тестироваться с помощью технологий белого и чёрного ящиков.

С помощью метода белого ящика будут проведены:

- модульное тестирование;
- интеграционное тестирование.

Тестирование веб-приложений проходит через несколько этапов [2]:

- на этапе функционального тестирования проверяются:
 - функциональные требования (с помощью метода чёрного ящика);
 - загрузка требуемых файлов (с помощью метода чёрного ящика);
- на этапе тестирования вёрстки проверяются:
 - соответствие вёрстки макету (чёрный ящик);
 - оптимизация графических изображений (чёрный ящик);
 - валидность кода [3] (белый ящик);
 - совместимость (чёрный ящик);
- на этапе тестирования удобства интерфейса:
 - собираются данные и отзывы пользователей (чёрный ящик);
 - анализ собранных данных и применение результатов анализа в последующей разработке;
- на этапе тестирования безопасности:
 - тестирование быстродействия сайта [4];
 - тестирование безопасности по OWASP testing guide [6];

ПРИЛОЖЕНИЕ Б

Модель системы

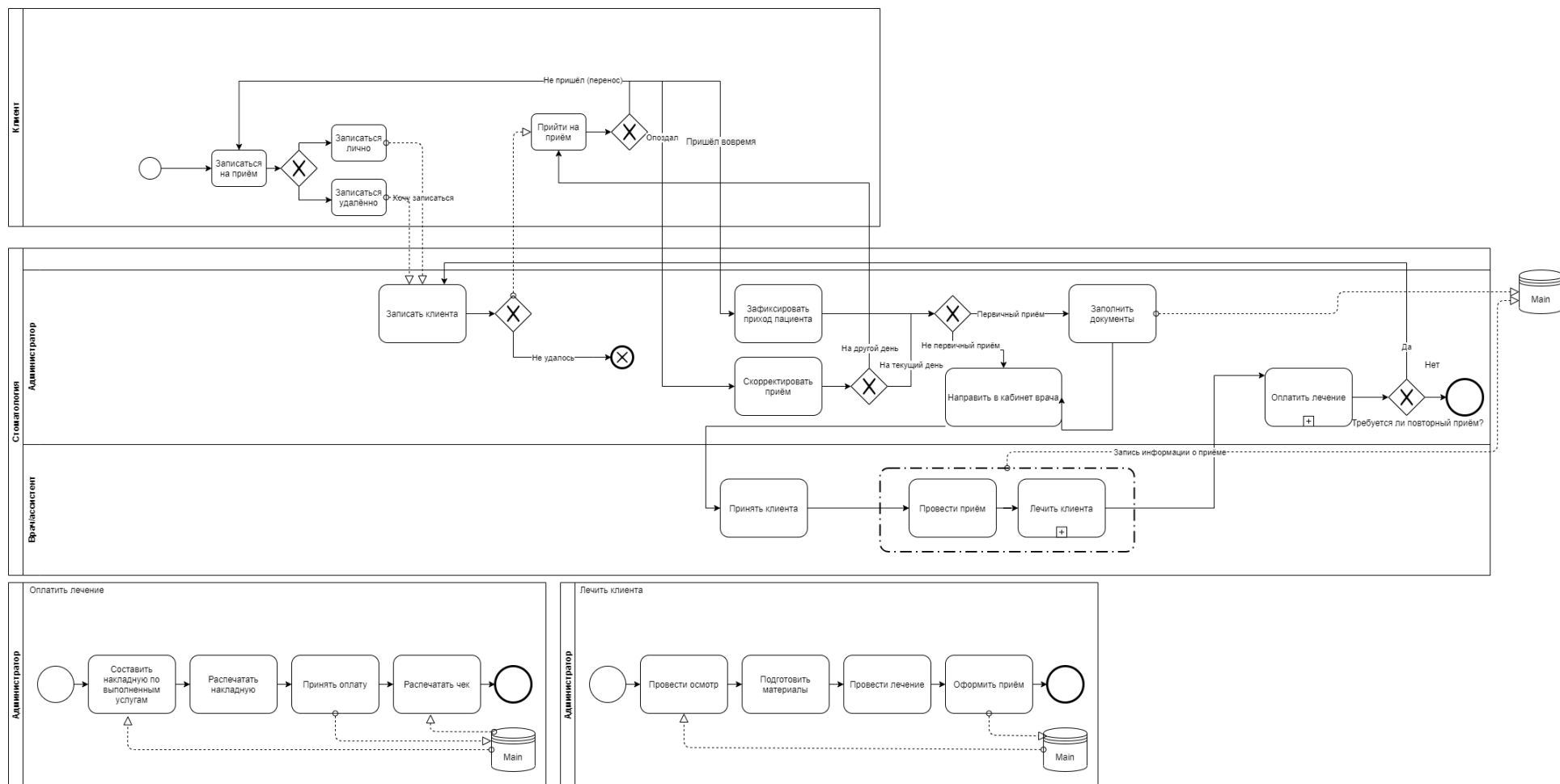


Рисунок Б.1 – Функциональная модель системы в нотации BPMN

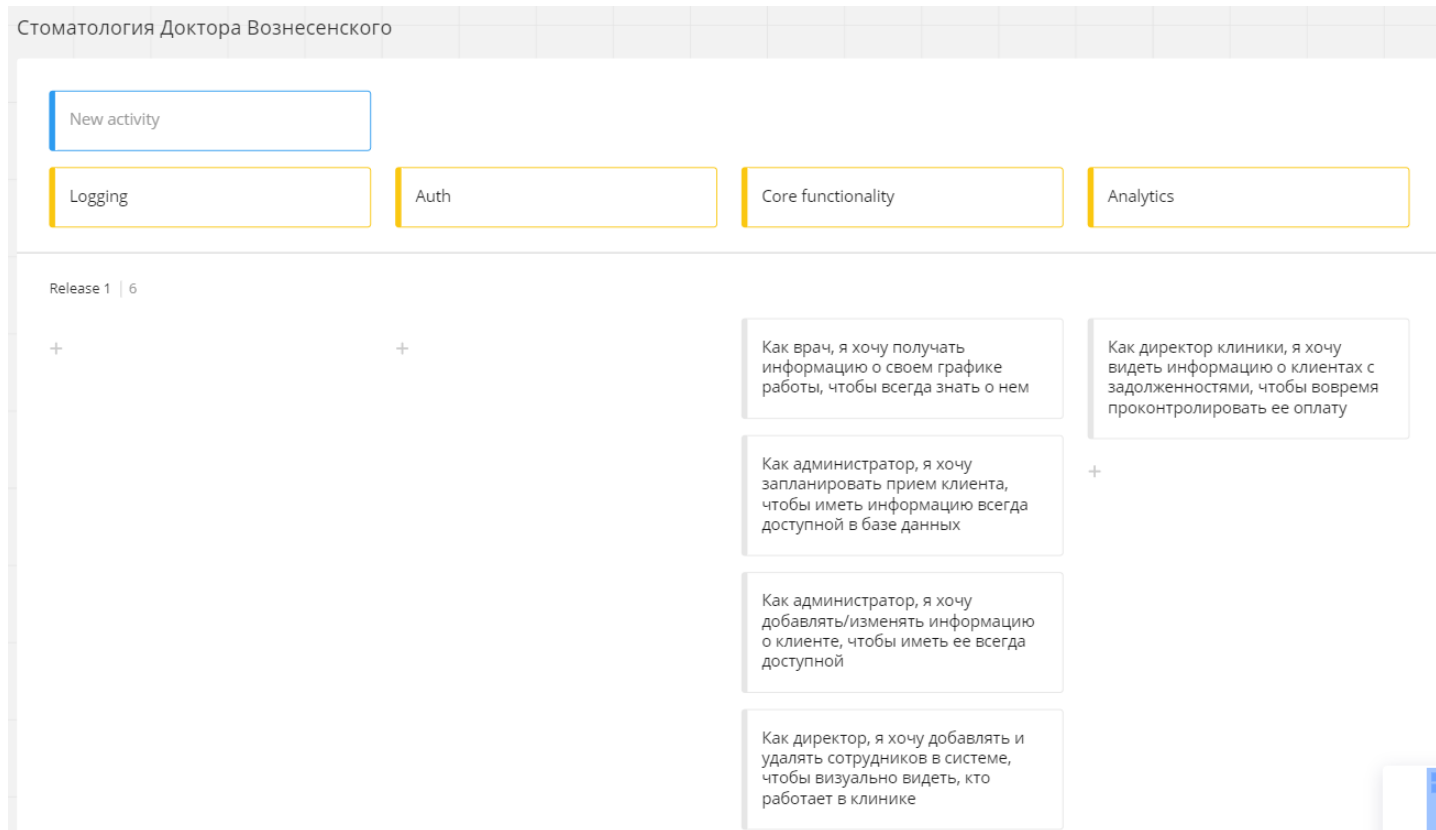


Рисунок Б.2 – Концептуальная модель: диаграмма User Story

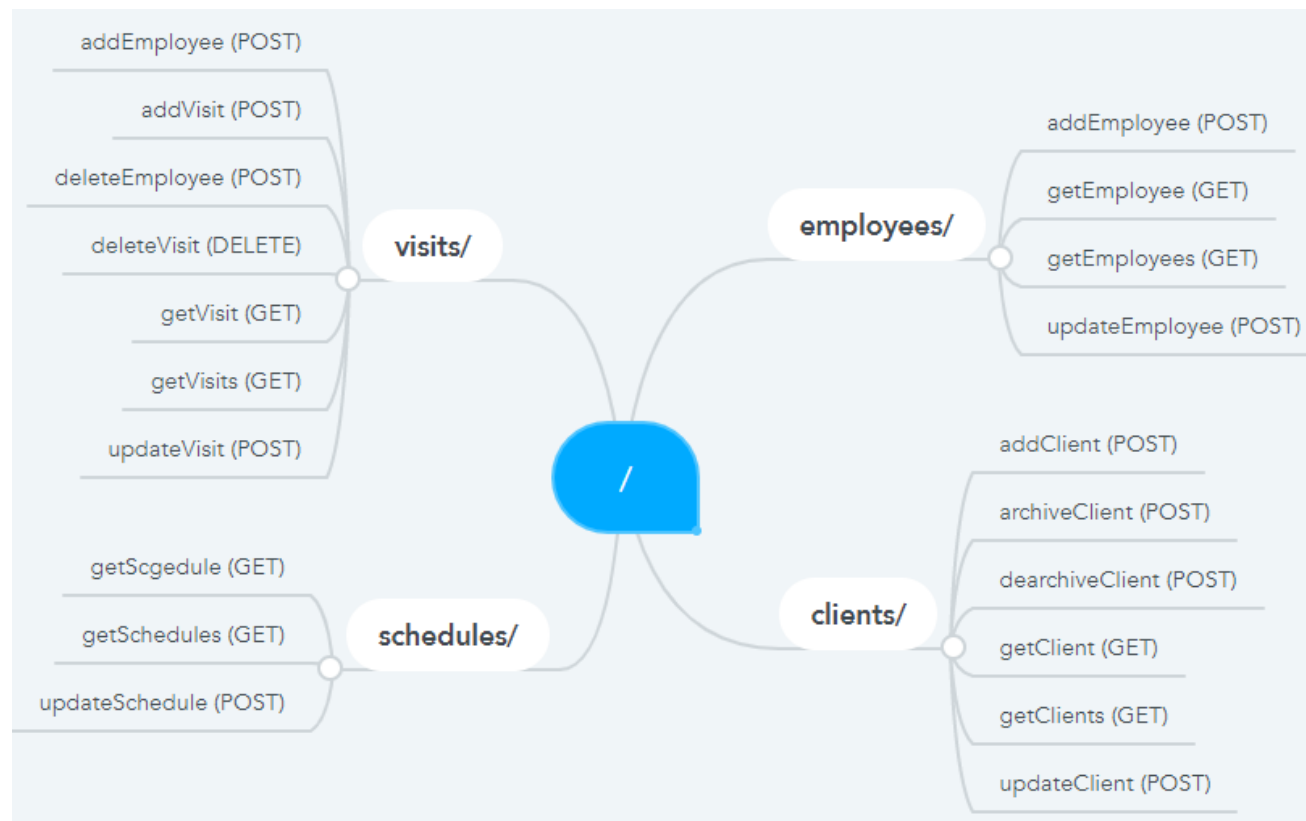


Рисунок Б.3 – Модель прикладного программного интерфейса