Федеральное государственное бюджетное образовательное учреждение высшего образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

Кафедра информационных систем и программной инженерии

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовому проекту по дисциплине**

**«Распределенные программные системы»**

**на тему:**

Прототип программной системы «Единая система автоматизации поликлиник (ЕСАП)»

Выполнил: студент гр. ИСТ-120

Карабанов А.А.

Принял: доц. Тимофеев А.А.

Владимир, 2023

**Аннотация**

Курсовой проект посвящён проектированию и разработке прототипа программной системы - «Единая система автоматизации поликлиник (ЕСАП)», которая предназначена для автоматизации и оптимизации работы поликлиник, позволяя эффективно управлять пациентами, записями, расписаниями и медицинскими данными.

В рамках проекта был проведен анализ и изучение существующих проблем и недостатков в работе поликлиник, таких как длительное ожидание, неэффективное распределение ресурсов и проблемы с доступностью медицинской информации. Программная часть системы реализована на основе платформы Java EE c использованием фреймворка Spring Boot и библиотеки React TS для реализации клиентской части.

Подсистема хранения данных курсового проекта разработана на основе СУБД PostgreSQL.

Курсовой проект представлен на 150 страницах, рисунков – 69, таблиц – 6, использованных источников – 6, приложений – 5.

The course project is devoted to the design and development of a prototype software system – «Unified Polyclinic Automation System (UPAS)», which is designed to automate and optimize the work of polyclinics, allowing you to effectively manage patients, records, schedules and medical data.

As part of the project, an analysis and study of existing problems and shortcomings in the work of polyclinics, such as long waiting times, inefficient allocation of resources and problems with the availability of medical information, was carried out. The software part of the system is implemented on the basis of the Java EE platform using the Spring Boot framework and the React TS library to implement the client part.

The data storage subsystem of the course project was developed on the basis of the PostgreSQL DBMS.

The course project is presented on 150 pages, drawings - 69, tables - 6, sources used - 6, applications - 5.

**СОДЕРЖАНИЕ**

[**1 ВВЕДЕНИЕ** 5](#_Toc136349911)

[**2 ПОСТАНОВКА ЗАДАЧИ** 6](#_Toc136349912)

[2.1 Описание предметной области 6](#_Toc136349913)

[2.2 Основные понятия предметной области 8](#_Toc136349914)

[2.3 Функциональные требования к системе 9](#_Toc136349915)

[2.4 Нефункциональные требования к системе 10](#_Toc136349916)

[2.5 Сравнительный анализ аналогов 10](#_Toc136349917)

[2.6 Обзор актуальных технологий 17](#_Toc136349918)

[**3 АНАЛИЗ ЗАДАЧИ И ФУНКЦИОНАЛЬНАЯ ДЕКОМПОЗИЦИЯ СИСТЕМЫ** 22](#_Toc136349919)

[3.1 Анализ бизнес-процесса «» 22](#_Toc136349920)

[3.2 Сценарий взаимодействия пользователя с системой 27](#_Toc136349921)

[3.3 Аутентификация пользователя в системе 31](#_Toc136349922)

[**4 СТРУКТУРНАЯ ОРГАНИЗАЦИЯ СИСТЕМЫ** 32](#_Toc136349923)

[4.1 Общие принципы организации системы 32](#_Toc136349924)

[4.2 Взаимодействие с внешними сервисами/удалённым модулем для … 32](#_Toc136349925)

[4.3 Организация доступа к данным 32](#_Toc136349926)

[4.4 Организация бизнес-логики 32](#_Toc136349927)

[4.5 Организация веб-интерфейса 32](#_Toc136349928)

[4.6 Взаимодействие компонентов системы для … 32](#_Toc136349929)

[4.7 Организация обработки ошибок 32](#_Toc136349930)

[4.8 Организация управления доступом 32](#_Toc136349931)

[**5 РЕАЛИЗАЦИЯ КОМПОНЕНТОВ СИСТЕМЫ** 33](#_Toc136349932)

[5.1 EJB-компоненты/компоненты бизнес-логики 33](#_Toc136349933)

[5.1.1 EJB-модуль 1 33](#_Toc136349934)

[5.1.2 EJB-модуль 2 33](#_Toc136349935)

[5.2 Веб-компоненты 33](#_Toc136349936)

[5.2.1 Классы управляемых бинов 33](#_Toc136349937)

[5.2.2 JSP-страница login.jsp 33](#_Toc136349938)

[5.2.3 JSP-страница register.jsp 33](#_Toc136349939)

[5.2.4 Конфигурирование JSF-приложения 33](#_Toc136349940)

[5.3 Физическая структура баз данных 33](#_Toc136349941)

[5.3.1 База данных Customer 33](#_Toc136349942)

[5.3.2 База данных Indent 33](#_Toc136349943)

[5.4 Обеспечение целостности данных 33](#_Toc136349944)

[5.4.1 Ограничения целостности 33](#_Toc136349945)

[5.4.2 Организация распределенных транзакций 33](#_Toc136349946)

[5.5 Реализация взаимодействия с внешним сервисом 33](#_Toc136349947)

[5.5.1 Протокол и форматы обмена данными 33](#_Toc136349948)

[**6 ИНСТРУКЦИЯ ПО УСТАНОВКЕ ПРИЛОЖЕНИЯ** 34](#_Toc136349949)

[**7 ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ ПРОЕКТОМ** 35](#_Toc136349950)

[7.1 Общие принципы взаимодействия в команде 35](#_Toc136349951)

[7.2 Распределение ролей в команде/зоны ответственности 35](#_Toc136349952)

[7.3 Календарный план работ 35](#_Toc136349953)

[**8 НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ** 36](#_Toc136349954)

[**9 ЗАКЛЮЧЕНИЕ** 37](#_Toc136349955)

[**ПРИЛОЖЕНИЕ А ЛИСТИНГ КОМПОНЕНТОВ СИСТЕМЫ** 39](#_Toc136349956)

[**ПРИЛОЖЕНИЕ Б СКРИПТ СОЗДАНИЯ БАЗЫ ДАННЫХ** 40](#_Toc136349957)

# **1 ВВЕДЕНИЕ**

Актуальность автоматизации процессов здравоохранения обусловлена ростом объемов обрабатываемой информации, а также необходимостью повышать качество предоставляемых услуг в области здравоохранения. Кроме того, актуальность связана с ростом внедрений информационных систем в области здравоохранения, что обусловлено Постановлением Правительства РФ №140 от 09.02.2022 "О единой государственной информационной системе в сфере здравоохранения".

Единая система автоматизации поликлиник — это комплексное программное решение, разработанное для улучшения организации работы поликлиник и обеспечения более качественного и эффективного медицинского обслуживания.

В современных условиях, когда здравоохранение является одной из важнейших сфер государственной деятельности, создание единой системы автоматизации поликлиник становится все более актуальным. Она позволяет автоматизировать все процессы, связанные с работой поликлиник, в том числе учет пациентов, их медицинской истории, расписание врачей, выписку рецептов и другие задачи.

Подобная система может быть использована не только для улучшения качества медицинского обслуживания, но и для оптимизации работы персонала поликлиники, повышения эффективности управления и уменьшения затрат на бумажную документацию.

Кроме того, единая система автоматизации поликлиник позволяет сократить время на обработку данных, увеличить точность ведения медицинских записей, улучшить координацию работы медицинского персонала и предоставить пациентам возможность более удобно и оперативно получать медицинскую помощь.

# **2 ПОСТАНОВКА ЗАДАЧИ**

## 2.1 Описание предметной области

В качестве предметной области рассматривается медицинское учреждение, имеющее несколько отделений, которые возглавляют ведущие специалисты поликлиники. Каждый сотрудник прикреплен за каким-то одним отделением. На каждого впервые обратившегося за услугами в поликлинику гражданина заводится амбулаторная карта, которая создается в электронном виде и хранится на сервере. Также пациенту выдается талон, даже в случае поступления больного на «скорой помощи». В случае необходимости стационарного лечения с пациентом заключается договор. Договор — это бумажная книжка, в которой фиксируются все события, связанные с лечением больного. После выздоровления пациента, договор прикрепляется к амбулаторной карте. Каждое медицинское оборудование закреплено за каким-либо кабинетом или хранится на складе.

После проведения анализа предметной области были выявлены следующие проблемы:

* с увеличением количества пациентов поликлиники, увеличивается количество информационных потоков, что приводит к снижению управляемости из-за несвоевременного получения информации, ошибочной информации или вовсе из-за её отсутствия. Это является недопустимым, так как может затянуть процесс лечения, привести к неверной или несвоевременной постановке диагноза;
* затраты большого количества временных и человеческих ресурсов на обработку документов и необходимость увеличивать эти ресурсы с ростом количества клиентов;
* тратится много времени на подготовку и поиск необходимых историй болезней, личных карточек, результатов обследований;
* в связи с большим бумажным документооборотом возрастает нагрузка на персонал, в связи с чем растет количество ошибок;
* при обращении пациента в различные поликлиники могут произойти ошибки при передаче сведений;
* из-за необходимости получать документы и данные из других подразделений возникают временные задержки, связанные с их доставкой.

Этот проект представляет собой современный взгляд на классическую систему здравоохранения, которая предоставляет медицинские услуги для удовлетворения потребностей населения в области здравоохранения.

Основная цель проекта – это разработать централизованную систему здравоохранения, позволяющую усовершенствовать информационную и программную поддержку основных процессов поликлиник.

Для достижения поставленной цели были сформулированы следующие задачи:

1. исследовать медицинские информационные системы;
2. разработать полную модель деятельности разрабатываемой ис;
3. выполнить анализ существующих процессов, построить модель idef0;
4. выработать требования и критерии к ис;
5. выполнить расчет экономической эффективности;
6. информационная поддержка принятия управленческих решений в медицинской организации;
7. мониторинг и управление потоками пациентов (электронная регистратура);
8. ведение электронной медицинской карты пациента (ЭМК);
9. оказание медицинской помощи с применением телемедицинских технологий;
10. организация профилактики заболеваний, включая проведение диспансеризации, профилактических медицинских осмотров;
11. организация иммунопрофилактики инфекционных болезней;
12. иные функциональные возможности по решению оператора информационной системы.

## 2.2 Основные понятия предметной области

*Поликлиника* — медицинское учреждение для оказания амбулаторной медицинской помощи больным на приёме и на дому.

*Медицинская карта* — медицинский документ, в котором лечащими врачами ведётся запись истории болезни пациента и назначаемого ему лечения.

*Врач* — человек, использующий свои навыки, знания и опыт в предупреждении и лечении заболеваний, поддержании нормальной жизнедеятельности организма человека.

*Главный врач —* врач, занимающий руководящую должность в поликлинике.

*Дежурный врач —* врач, который ведет прием пациентов непосредственно в день обращения в поликлинику.

*Регистратор —* сотрудник поликлиники, работающий в регистратуре. Он встречает пациентов, осуществляет распределение клиентов по специалистам и запись на прием, а также ведет картотеку организации.

Концептуальная диаграмма классов представлена на рисунке 1.

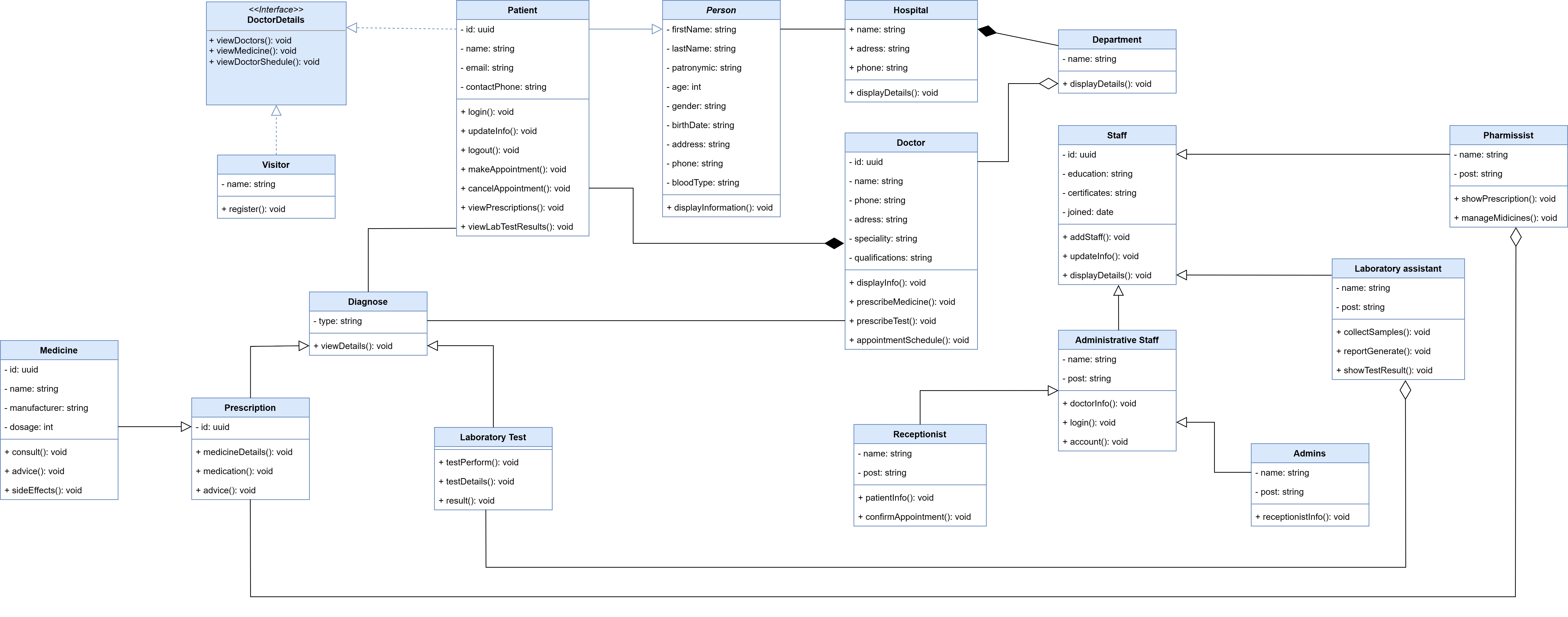


Рисунок 1 – Концептуальная диаграмма классов.

## 2.3 Функциональные требования к системе

Согласно приказу Министерства Здравоохранения РФ от 24.12.2018 № 911н «Об утверждении Требований к государственным информационным системам в сфере здравоохранения субъектов Российской Федерации, медицинским информационным системам медицинских организаций и информационным системам фармацевтических организаций» посредством медицинской информационной системы должны обеспечиваться:

* Система должна обеспечивать возможность регистрацию / удаление больниц в системе;
* Система должна обеспечивать возможность регистрации / авторизации / аутентификации пользователей;
* Система должна обеспечивать возможность установления ролей для медперсонала при регистрации (врач, лаборант, регистратор);
* Система должна обеспечивать возможность мониторинга и управления потоками пациентов (создание электронной очереди);
* Система должна обеспечивать возможность ведение электронной медицинской карты пациента (ЭМК);
* Система должна обеспечивать возможность создание версии для печати ЭМК;
* Система должна обеспечивать возможность печати ЭМК;
* Система должна обеспечивать возможность прикрепления пациента к больнице;
* Система должна обеспечивать возможность врачу получать данные о пациенте;
* Система должна обеспечивать возможность получение данных об назначенных обследованиях для лаборантов;

## 2.4 Нефункциональные требования к системе

Были выделены следующие нефункциональные требования к системе:

* Система должна иметь понятный русскоязычный интерфейс;
* Система должна открываться в течение не больше 15 секунд;
* Отклик системы для типовых действий не должен превышать 10 секунд для средней нагрузки (300000 - 500000 пользователей);
* Система должна обеспечивать безопасность хранения пользовательских данных, путём их кэширования;
* Система должна обеспечивать стабильную работу при одновременном использовании 1500000 пользователями;
* Клиент должен взаимодействовать с системой посредством протокола HTTP.
* Система должна работать на большинстве браузеров (Yandex, Mozilla, Chrome и д.р.).

## 2.5 Сравнительный анализ аналогов

Существует множество систем автоматизации работы медицинских учреждений, каждая из которых имеет свои преимущества и недостатки. При выборе подходящей системы необходимо учитывать множество факторов, таких как размер медицинского учреждения, его специализация, требования к безопасности и т.д.

Рассмотрим некоторые из них:

1. **МИС «БАРС.Здравоохранение»** — это информационная система, разработанная для автоматизации работы больниц и поликлиник. Основной функционал системы включает в себя учет и анализ медицинских данных, планирование и учет услуг, управление персоналом, учет финансов и материальных запасов, анализ работы медицинского учреждения. Система позволяет организовать эффективную работу медицинского учреждения и повысить качество обслуживания пациентов.
2. **МИС «Инфоклиника»** — это система автоматизации работы медицинских учреждений. Основной функционал системы включает в себя учет медицинских карт, планирование и учет услуг, управление персоналом, учет финансов и материальных запасов, анализ работы медицинского учреждения. Одним из преимуществ системы является интеграция с электронным реестром и страховыми компаниями, что позволяет быстро и эффективно вести учет услуг и оплаты.
3. **МИС «Медиалог»** — это комплексная система автоматизации работы медицинских учреждений. Основной функционал системы включает в себя учет и анализ медицинских данных, планирование и учет услуг, управление персоналом, учет финансов и материальных запасов, анализ работы медицинского учреждения. Система позволяет эффективно управлять медицинским учреждением, повысить качество обслуживания пациентов и увеличить прибыльность.
4. **МИС «1С:Медицина»** — это программа для автоматизации работы медицинских учреждений. Основной функционал системы включает в себя учет и анализ медицинских данных, планирование и учет услуг, управление персоналом, учет финансов и материальных запасов, анализ работы медицинского учреждения. В программе реализованы модули для ведения учета электронных медицинских карт, реестра услуг и оплаты, статистического учета, управления документооборотом и другие.

Необходимо провести анализ на основе методики с использованием весовых коэффициентов, для определения наилучшей. Для получения наиболее обоснованного результата будет использована методика определения обобщённой степени соответствия альтернатив.

Для оценки альтернатив нужно выделить ряд критериев, по которым будет производится оценка аналогов:

1. **Стоимость** – оценивает стоимость приобретения и поддержки системы, включая начальные инвестиции и операционные расходы.

Значения:

* 8 – 10 низкая стоимость, система доступна для большинства клиник и может быть легко поддерживаема;
* 5 – 7 средняя стоимость, система доступна для большинства клиник, но ее поддержка может потребовать дополнительных ресурсов;
* 0 – 4 высокая стоимость, система может быть недоступна для большинства клиник.

1. **Масштабируемость** – определяет, насколько легко можно масштабировать систему для удовлетворения потребностей растущей клиники.

Значения:

* 8 – 10 система может быть легко масштабирована, чтобы удовлетворить потребности растущей клиники;
* 5 – 7 система может быть масштабирована, но могут потребоваться дополнительные ресурсы;
* 0 – 4 система не может быть легко масштабирована.

1. **Интерфейс пользователя** – оценивает, насколько интуитивно понятен и прост в использовании интерфейс системы для пользователей.

Значения:

* 8 – 10 интерфейс системы прост в использовании, интуитивно понятен и не вызывает затруднений у пользователей;
* 5 – 7 интерфейс системы может быть немного сложным или не интуитивным, что требует некоторого обучения пользователей;
* 0 – 4 интерфейс системы очень сложен и неудобен в использовании.

1. **Возможность экспорта/импорта данных** – определяет наличие возможности импорта и экспорта данных, что позволяет упростить процесс миграции данных и обмена информацией с другими системами.

Значения:

* 8 – 10 система поддерживает импорт и экспорт данных в различных форматах, что облегчает обмен информацией с другими системами;
* 5 – 7 система поддерживает импорт и экспорт данных в некоторых форматах, что может потребовать дополнительных ресурсов;
* 0 – 4 система не поддерживает импорт и экспорт данных.

1. **Функционал** – это набор возможностей, которые предоставляет система пользователю. В данном случае, оценивается наличие и разнообразие функций, которые включают в себя запись на прием, ведение медицинской карты, выписывание рецептов, проведение медицинских осмотров, формирование отчетности и другие.

Значения:

* 8 – 10 система имеет широкий функционал, позволяющий полностью автоматизировать работу поликлиники;
* 5 – 7 функционал системы охватывает основные процессы работы поликлиники, но может быть ограничен в некоторых моментах;
* 0 – 4 система имеет ограниченный функционал, не позволяющий полностью автоматизировать работу поликлиники.

1. **Удобность навигации** – это критерий, оценивающий удобство использования системы, ее логическую структуру, удобство доступа к различным функциям и информации.

Значения для данного критерия могут быть определены следующим образом:

* 8 – 10 навигация в системе легка и интуитивно понятна, все функции быстро доступны пользователю;
* 5 – 7 навигация в системе может быть несколько сложной, некоторые функции могут быть недоступны на первый взгляд;
* 0 – 4 навигация в системе крайне неудобна, функции плохо организованы и могут вызывать затруднения у пользователя.

Результат оценки аналогов по каждому критерию представлен в таблице 1.1.

Таблица 1.1 – Оценка систем по ряду критериев:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерии | Барс | Инфоклиника | Медиалог | 1C.Медицина |
| Стоимость | 6 | 8 | 7 | 6 |
| Масштабируемость | 9 | 9 | 8 | 9 |
| Интерфейс пользователя | 5 | 8 | 7 | 9 |
| Экспорт/импорт данных | 9 | 9 | 8 | 9 |
| Функционал | 9 | 9 | 9 | 9 |
| Удобность навигации | 7 | 8 | 7 | 9 |

Критерии и их веса представлены в таблице 1.2.

Вес критерия – определяет, на сколько сильное влияние оказывает тот или иной параметр на итоговое значение.

Взвесим критерии, т.е. определим, значение каких из них наиболее влияют на выбор системы.

Таблица 1.2 – Критерии и весовые коэффициенты.

|  |  |
| --- | --- |
| Критерии | Вес |
| Стоимость | 0,15 |
| Масштабируемость | 0,2 |
| Интерфейс пользователя | 0,15 |
| Экспорт/импорт данных | 0,2 |
| Функционал | 0,2 |
| Удобность навигации | 0,1 |

В данном случае были выбраны следующие весовые коэффициенты на основе предположения, что наиболее важными критериями являются масштабируемость, функционал и экспорт/импорт данных. Остальные критерии также являются важными, но их значимость ниже, поэтому им были присвоены меньшие веса.

Выполним оценку систем, подсчитав взвешенные оценки. Для этого для каждого аналога суммируем оценки по критериям, умноженным на веса этих критериев.

Таблица с результатами подсчёта взвешенных оценок представлена в таблице 1.3.

Таблица 1.3 – Результат подсчёта взвешенных оценок.

|  |  |
| --- | --- |
| Аналог | Взвешенные оценки |
| БАРС | 7,75 |
| Инфоклиника | 8,2 |
| Медиалог | 7,8 |
| 1C.Медицина | 8,55 |

На основе этой оценки, наиболее подходящей системой является Инфоклиника и 1C.Медицина, так как они набрали наибольшее количество баллов по всем критериям.

Необходимо провести сравнительный обзор по функциональности продуктов, с целью раскрыть основные функции систем.

Обзор функции систем представлен в таблице 1.4.

В качестве «+» подразумевается наличие данного функционала, «-»  
означает его отсутствие.

Таблица 1.4 – Функциональный обзор рассматриваемых систем.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Функции | БАРС | Инфоклиника | 1C.Медицина | Медиалог |
| Расписание врачей/кабинетов, запись на прием. | + | + | + | + |
| Учет пациентов, оформление  документов. | + | + | + | + |
| Ведение электронной медицинской  карты (ЭМК). | + | + | + | + |
| Планы лечения и стандарты лечения. | + | - | - | + |
| Работа с больничными листами | + | + | + | + |
| Учет направлений/рекомендаций от внешних и внутренних врачей (мед. учреждений). | + | - | - | + |
| Отчетность и аналитика (финансовая, медицинская, маркетинговая). | + | + | + | + |
| Учет финансовой информации –  регистрация оказания услуг, оплата услуг (нал./карта/безнал). | + | - | + | - |
| Интеграция с другими информационными системами. | - | - | + | - |

Вывод: По результатам данного анализа, можно сделать следующий вывод: система «1С:Медицина.Поликлиника» является одним из наиболее перспективных программных продуктов среди представленных. Она обладает преимуществами, которые делают ее привлекательным выбором для многих организаций в области здравоохранения. Главным преимуществом данной системы является ее широкий функционал, позволяющий автоматизировать и управлять различными аспектами работы поликлиники. Она предлагает интегрированные модули для учета пациентов, ведения медицинских карт, планирования приемов, финансового учета и других важных процессов. Поэтому разрабатываемая система также будет стремиться предоставить широкий функционал, сопоставимый с функционалом системы «1С:Медицина.Поликлиника».

## 2.6 Обзор актуальных технологий

* MVC

MVC (Model-View-Controller) – это популярная архитектурная модель, которая является универсальным шаблоном проектирования для создания веб-приложений, основанных на языках программирования.

MVC удобнее, так как он более простой и понятный для понимания любым программистом. Кроме того, паттерн используется во многих фреймворках и приложениях, что позволяет быстрее разрабатывать и поддерживать код. В MVC также проще реализовывать взаимодействие между сущностями и обрабатывать ошибки.

* JPA

JPA (Java Persistence API) — это высокоуровневый Java-интерфейс, который упрощает доступ к базам данных, обеспечивая абстракцию от конкретных реализаций и улучшая возможности для создания, чтения, обновления и удаления объектов Java в базе данных.

JPA можно сравнить с использованием слоя DAO. Реализовывая слой DAO, можно добиться большей производительности так как система не тратит время на генерирование SQL-кода, а JPA во много раз укоряет создание системы, поэтому он и был выбран.

* SPA

SPA (Single Page Application) — это веб-приложение, которое обновляет содержимое страницы без перезагрузки всей страницы. Вместо этого одна страница загружается при первом открытии и в дальнейшем модифицируется динамически при помощи AJAX-запросов к серверу. Существует несколько аналогов SPA, таких как Angular, React и Vue.js. Они являются фреймворками для разработки SPA, имеют свои особенности и выбор конкретного фреймворка зависит от конкретных задач и требований проекта.

* Быстрота загрузки: однажды загрузив весь необходимый код для приложения, пользователь может перемещаться между разными разделами, не ожидая загрузки каждой отдельной страницы;
* Удобство использования: приложение SPA позволяет работать более быстро и эффективно. Пользователь может получать более оперативный доступ к информации;
* Улучшенный пользовательский интерфейс: приложения SPA могут работать с более сложными и интерактивными пользовательскими интерфейсами, такими как анимации, расширенные формы;
* Удобное масштабирование: SPA можно легко масштабировать. Приложение можно заменять и обновлять поэтапно, не останавливая работу на сервере;
* Удобная поддержка: SPA обеспечивает более удобную и эффективную поддержку приложения. Загрузка всего приложения происходит в начале, поэтому обращения к серверу в различных зонах приложения становятся проще и эффективнее.
* React.js

React.js является библиотекой JavaScript, предназначенной для создания пользовательских интерфейсов. React был создан компанией Facebook и вышел в свет в 2013 году. Он используется для создания масштабируемых и быстрых веб-приложений.

Сравнивая React.js, Vue и Angular все три JavaScript-фреймворка являются мощными инструментами для создания современных веб-приложений. Однако, каждый из них имеет свои особенности:

React.js:

* Эффективный: React.js имеет множество возможностей и удобство, которые позволяют ему быть быстрым и масштабируемым для разработки больших приложений.
* Гибкий: React.js позволяет разработчикам использовать синтаксический сахар JSX, который делает написание кода чище, более понятным и производительным. Также фреймворк не является монолитным, и дает возможность разработчикам добавлять в него собственные библиотеки и инструменты.

Vue:

* Лёгкий: Vue.js использует меньше ресурсов, чем Angular и React, при этом имеет множество функций и удобство разработки.
* Простой: Vue.js имеет понятный синтаксис и удобный интерфейс, что делает его легче для освоения новичками.

Angular:

* Полный: Angular предоставляет множество возможностей в одном коробке, включая управление состоянием, валидацию форм, роутинг, тестирование и многое другое.
* Официальный: Angular разрабатывается и поддерживается компанией Google, интеграция с другими инструментами Google обеспечивает безопасность и стабильность работы.

Возвращаясь к вопросу, почему React.js лучше, чем Vue и Angular, следует отметить, что он предоставляет гибкость и эффективность в разработке, а также активно используется во многих крупных проектах, что подтверждает его надежность и эффективность.

* TypeScript

JavaScript и TypeScript являются двумя популярными языками программирования, широко используемыми для разработки веб-приложений. JavaScript является стандартным языком сценариев веб-страниц и обладает простым синтаксисом, что делает его доступным для начинающих разработчиков. TypeScript, с другой стороны, является надмножеством JavaScript и предоставляет статическую типизацию, расширяя возможности JavaScript и повышая безопасность и надежность кода. Оба языка являются мощными инструментами для создания интерактивных и масштабируемых веб-приложений.

JavaScript:

* Гибкий, динамичный и кроссплатформенный;
* Используется как на стороне клиента, так и на стороне сервера;
* Легкая интерпретация;
* Поддерживается всеми браузерами;
* Слабо типизированный;
* Jit компиляция;

TypeScript:

* Простота обслуживания и повышение продуктивности проекта;
* Возможна статическая типизация и аннотации;
* Поддерживает объектно-ориентированные функции, такие как интерфейс, наследование и классы;
* Легкая отладка и раннее обнаружение ошибок;

Когда код становится огромным, сложным в обработке и более подверженным ошибкам, лучше, если некоторые ошибки будут обнаружены во время самой компиляции. Вот здесь и помогает TypeScript. Поэтому он будет использоваться в проекте.

ДОПИСАТЬ ЧТО ИЗМЕНИМ В СИСТЕМАХ

# **3 АНАЛИЗ ЗАДАЧИ И ФУНКЦИОНАЛЬНАЯ ДЕКОМПОЗИЦИЯ СИСТЕМЫ**

## 3.1 Анализ бизнес-процесса «Управление медицинским обслуживанием пациентов».

**Участники процесса**

Пользователь — это работник медицинской организации, физическое лицо, которое имеет медицинское или иное образование, работает в медицинской организации и в трудовые обязанности которого входит осуществление медицинской деятельности.

Основная целевая аудитория — взрослые мужчины и женщины, проживающие в конкретном районе в возрасте от 22 до 65 лет, имеющие высшее медицинское образование и работающие по трудовому договору.

Второстепенная целевая аудитория — это занятые молодые пары, которые считают, что время — это ценный и ограниченный ресурс.

Пример портрета целевой аудитории:

«Соколова Наталья Филипповна работает в должности медсестры терапевтического отделения ОГБУЗ «Седановская городская больница №1» с мая 2011 года. Оформлена переводом с аналогичной должности из ГБ-2 города Братска. Общий стаж работы в лечебно-профилактических медицинских учреждениях 6 лет.

Окончила в 2008 году государственное медицинское училище г. Братска. В 2012 году прошла курсы повышения квалификации в Иркутском государственном медицинском университете по специальности «Организация сестринского дел».

В процессе работы Соколова Н.Ф. осуществляла уход за пациентами и наблюдение за их состоянием, обеспечивала соблюдение лечебно-охранительного и санитарно-эпидемиологического режимов в отделении, осуществляла получение лекарственных средств, обеспечивала их учёт и хранение. Обязанности выполнялись в полном объёме, врачебные назначения выполнялись своевременно и точно.

В коллективе Соколова Н.Ф. зарекомендовала себя с положительной стороны: принимает активное участие в общественной жизни коллектива, пользуется уважением коллег, имеет положительные отзывы о пациентов. Наиболее заметными качествами в личном плане являются аккуратность, ответственность, доброжелательное отношение к людям, осознанное стремление всемерно повышать свой профессиональный уровень».

**Перечень ролей**

В рамках Единой системы автоматизации поликлиник (ЕСАП) предусмотрены следующие роли пользователей:

* Врач:
  + Может просматривать расписание приема пациентов;
  + Может создавать, изменять и удалять записи на прием;
  + Может просматривать медицинскую историю пациента;
  + Может добавлять записи;
  + Может вносить диагнозы и рекомендации для пациентов;
* Регистратор:
  + Может создавать новых пациентов в системе;
  + Может записывать пациентов на прием к врачам;
  + Может просматривать и редактировать информацию о пациентах;
  + Может проверять доступность и свободное время в расписании врачей;
* Главный врач:
  + Обладает всеми правами врача и регистратора;
  + Может управлять расписанием приема врачей;
  + Может назначать и удалять врачей;
  + Может просматривать общую статистику работы поликлиники;
  + Может просматривать и управлять медицинскими данными и историей пациентов;
* Администратор:
  + Обладает всеми правами в системе;
  + Может управлять пользователями и их ролями;
  + Может просматривать и управлять доступом к системе;

**Диаграммы бизнес-процессов**

AS-IS диаграмма автоматизируемого бизнес-процесса «Деятельность поликлиники» в нотации IDEF0 представлена на рис. 8-12.



Рис. 8. Диаграмма верхнего уровня.



Рис. 9. Диаграмма декомпозиции idef0.



Рис. 10. Диаграмма работы «Регистрирование больных».



Рис. 11. Диаграмма работы «Проведение анализов».



Рис. 12. Диаграмма работы «Прием больного».

Рассмотрим подробнее работы, включающие основные процессы клиники. На этот раз воспользуемся нотацией BPMN.

При звонке пациента с целью записи на прием к врачу, администратор просматривает журнал записи с целью поиска свободного времени для записи и при этом удобного пациенту. После согласования с пациентом времени приема регистратор формирует запись, фиксируя данные пациента в журнал.

Процесс оформления документов в регистратуре перед приемом представлен на рис. 14.

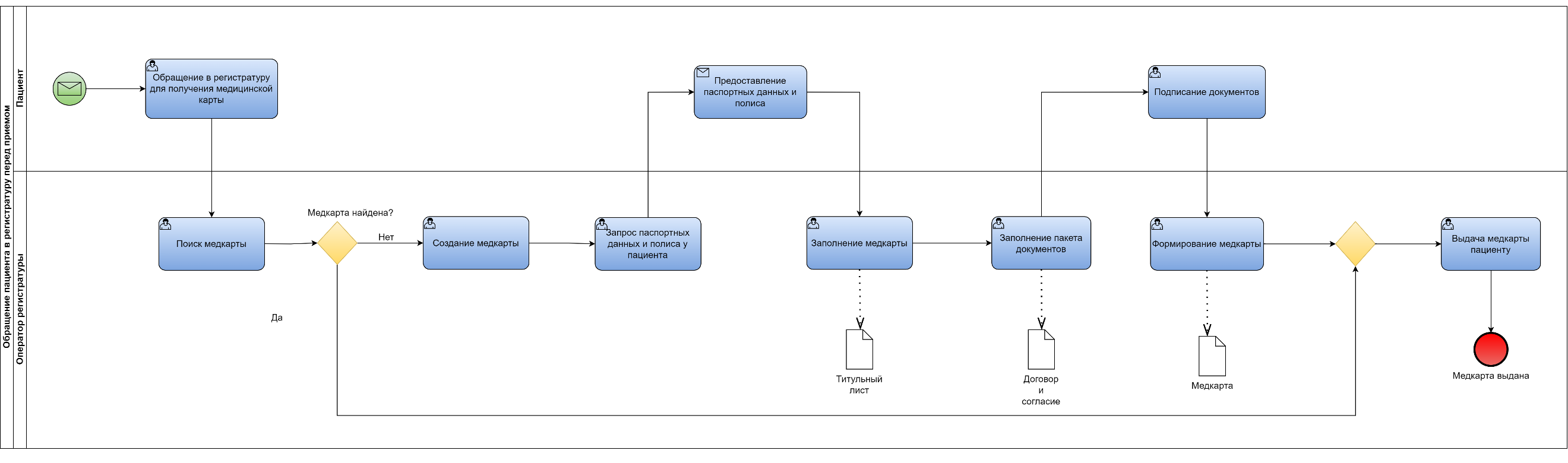


Рис. 14. Обращение пациента в регистратуру перед приемом.

Перед приемом пациент обращается в регистратуру для оформления документов. Все документы заполняются регистратором вручную.

Прием пациента врачом представлен на рис. 15.

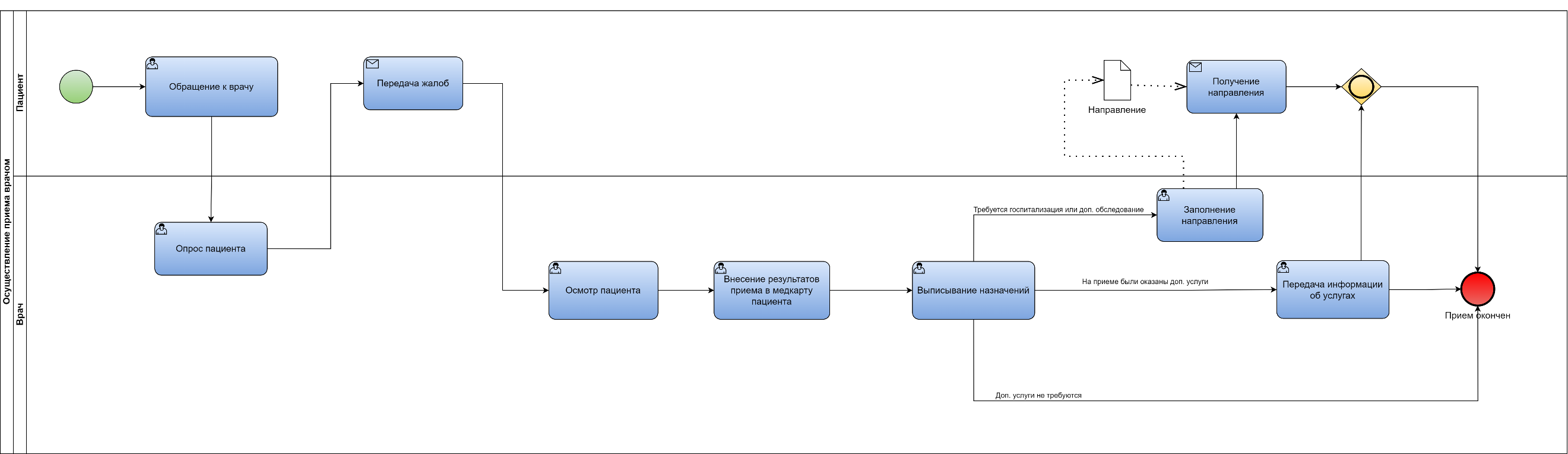


Рис. 15. Осуществление приема пациента врачом.

Врач на приеме после осмотра и консультации пациента вносит результаты приема в карту пациента, выписывает направления и рецепт вручную.

TO-BE диаграмма основных процессов клиники. Описание процессов также представлено в нотации BPMN. Представленная модель показывает то, как изменится процесс при внедрении МИС. Основное изменение в том, что появиться централизованная информационная база.

При записи регистратор проверяет, имеется ли пациент в МИС электронную медицинскую карту (далее ЭМК), после чего заводит карту пациента (если ее нет).

Процесс обращения пациента в регистратуру для получения медкарты представлен на рис. 17.

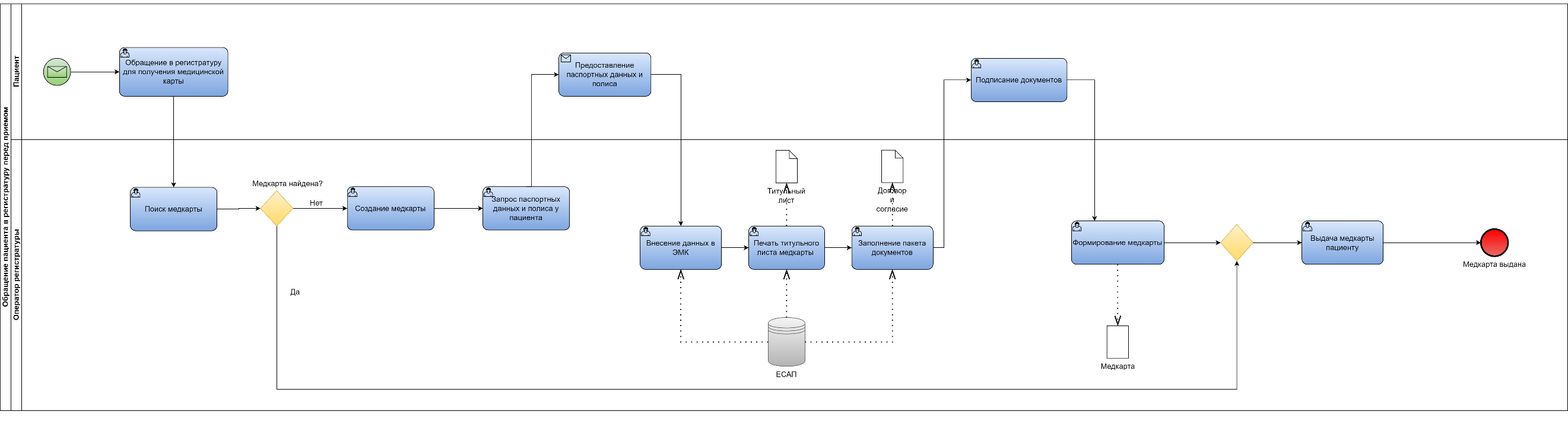


Рис. 17. Процесс обращения пациента в регистратуру для получения медкарты.

В данном бизнес-процессе показан процесс автоматизации формирования документов. Предполагается, что регистратор ничего не будет заполнять вручную, все должно печататься из МИС на основании информации, внесенной в ЭМК.

Врач на приеме также использует МИС. Протокол приема формируется на основании шаблона, куда врач вносит информацию по приему пациента. После чего все документы распечатываются. Теперь врач, ведущий осмотр, видит в одном окне, всю динамику пациента.

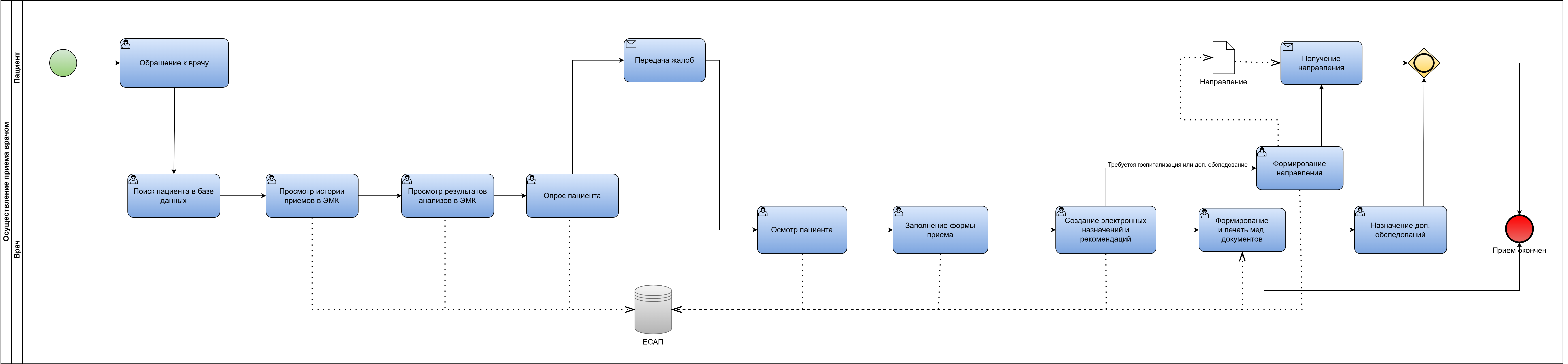


Рис. 19. Процесс приема пациента врачом.

## 3.2 Сценарий взаимодействия пользователя с системой

Диаграмма прецедентов разрабатываемой информационной системы представлена на рис. 10.

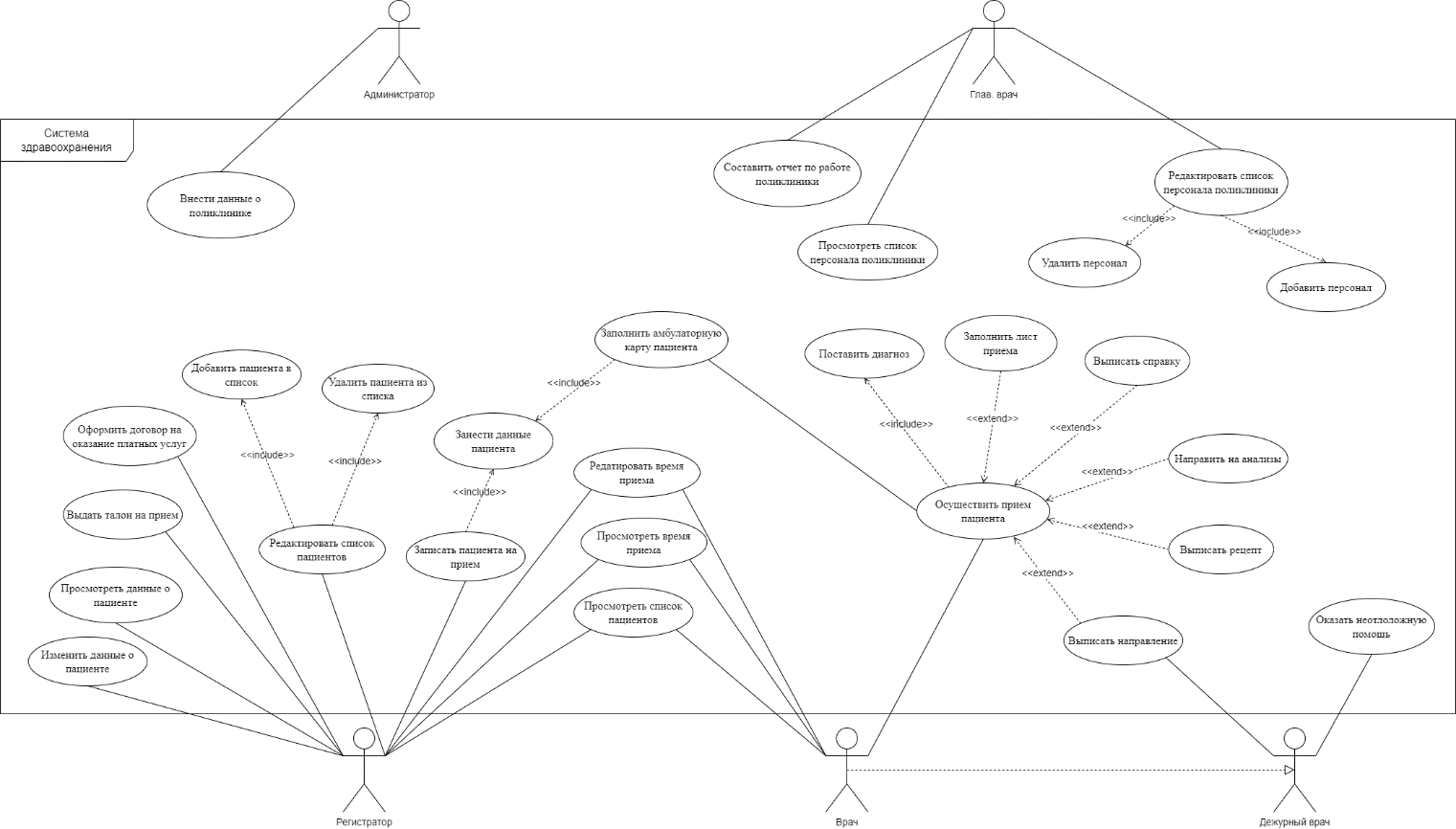


Рис. 10. Диаграмма прецедентов.

Расширенное описание прецедента «Осуществить прием пациента» представлено на таблице 1.

Таблица 1. Описание прецедента.

|  |  |
| --- | --- |
| **Название** | «Осуществление прием пациента» |
| **Предусловие** | Пациент записан на прием. |
| **Действующее лицо** | Врач |
| **Основной поток** | *Осуществление прием пациента*  Врач проводит осмотр пациента. После окончание осмотра врач вносит данные в электронную форму, которая отправляется на сервер. |
| **Альтернативный поток** | Пациент не явился на прием. |

Расширенное описание прецедента «Внести данные о больнице» представлено на таблице 2.

Таблица 2. Описание прецедента.

|  |  |
| --- | --- |
| **Название** | «Внести данные о больнице» |
| **Предусловие** | Главврач отправляет запрос на регистрацию медучреждения. |
| **Действующее лицо** | Администратор |
| **Основной поток** | *Внести данные о больнице*  Администратор заполняет форму данными, полученными от медучреждения, и сохраняет их в БД. |
| **Альтернативный поток** | Обязательные поля пустые. Ошибка валидации. |
| **Постусловие** | Если регистрация прошла успешно, системы сгенерирует имя пользователя и пароль для авторизации, которые администратор предоставит Главврачу. |

Расширенное описание прецедента «Добавить персонал» представлено на таблице 3.

Таблица 3. Описание прецедента.

|  |  |
| --- | --- |
| **Название** | «Добавить персонал» |
| **Предусловие** | Работник устроился на работу в медучреждение. |
| **Действующее лицо** | Главврач. |
| **Основной поток** | *Добавить персонал*  Главврач заполняет форму регистрации сотрудника в медучреждении. |
| **Альтернативный поток** | Обязательные поля пустые. Ошибка валидации. |
| **Постусловие** | Работник зарегистрированы в медучреждении. |

Диаграмма состояний документа «Пациент» представлена на рис. 11.

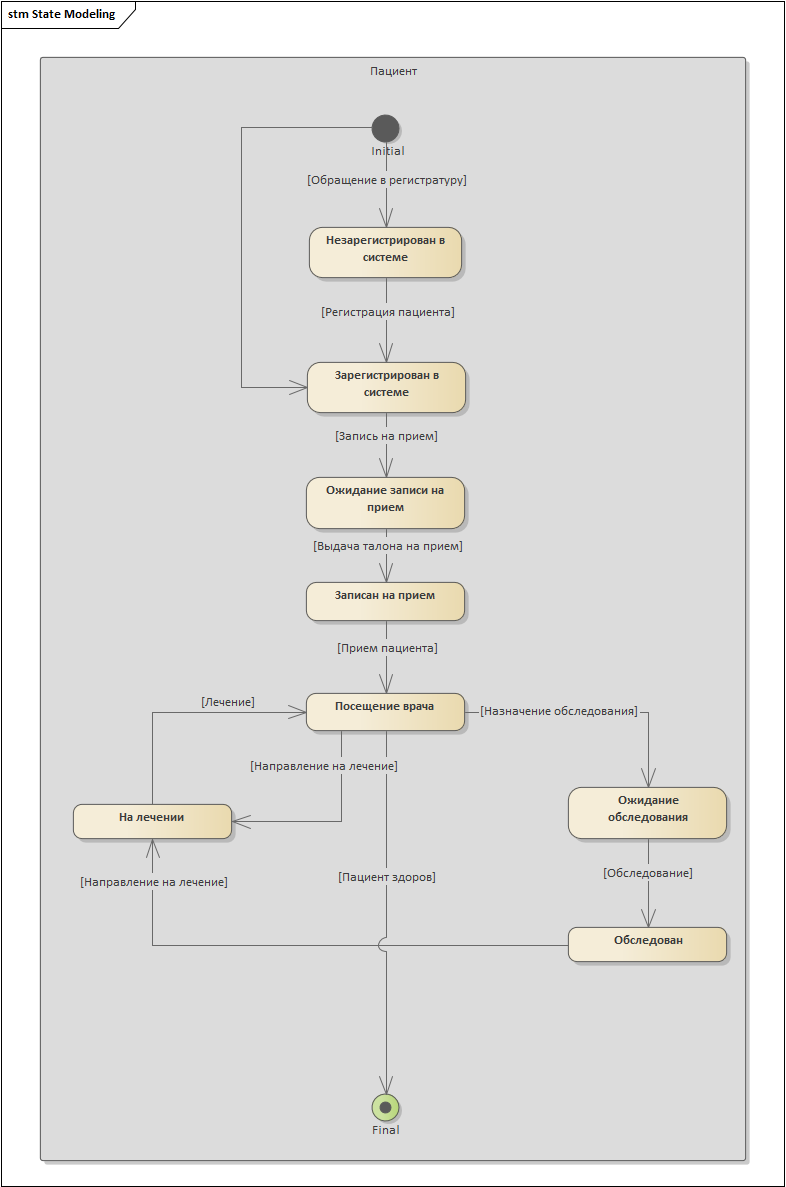


Рис. 11. Диаграмма состояний документа «Пациент».

Диаграмма состояний документа «Врач» представлена на рис. 12.

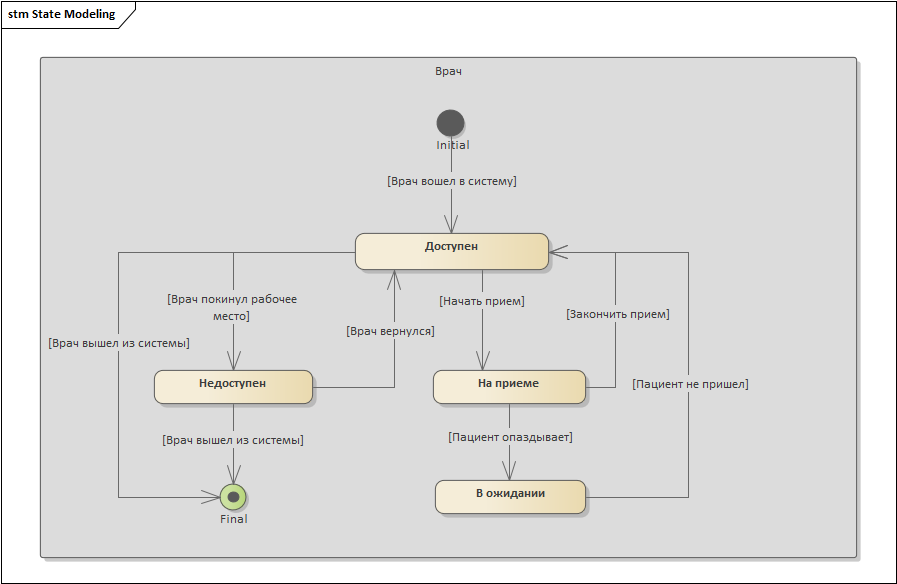


Рис. 12. Диаграмма состояний документа «Врач».

## 3.3 Аутентификация пользователя в системе

ЧТО?

# **4 СТРУКТУРНАЯ ОРГАНИЗАЦИЯ СИСТЕМЫ**

## 4.1 Общие принципы организации системы

В основу структуры системы заложен принцип REST API (Representational State Transfer Application Programming Interface) — это архитектурный подход, который используется для создания и предоставления веб-сервисов. Он основан на протоколе HTTP и включает в себя набор методов запросов, таких как GET, POST, PUT и DELETE. Каждый метод имеет свое предназначение и позволяет взаимодействовать с ресурсами.

Альтернативой может быть организация веб-сервера, который отдает статические HTML-страницы в ответ на HTTP-запросы. Однако, в случае, когда проект разрабатывается командой, где одна часть занимается разработкой пользовательского интерфейса, а другая - веб-сервера, более удобным решением будет использование REST API. Это позволяет разделить систему на два независимых проекта. Такой подход повышает эффективность создания приложения, позволяет решить проблему конфликта общего кода и упрощает разделение задач между разработчиками.

## 4.3 Организация доступа к данным

## 4.4 Организация бизнес-логики

## 4.5 Организация веб-интерфейса

## 4.6 Взаимодействие компонентов системы для …

## 4.7 Организация обработки ошибок

## 4.8 Организация управления доступом

# **5 РЕАЛИЗАЦИЯ КОМПОНЕНТОВ СИСТЕМЫ**

## 5.1 EJB-компоненты/компоненты бизнес-логики

## 5.1.1 EJB-модуль 1

## 5.1.2 EJB-модуль 2

## 5.2 Веб-компоненты

## 5.2.1 Классы управляемых бинов

## 5.2.2 JSP-страница login.jsp

## 5.2.3 JSP-страница register.jsp

## 5.2.4 Конфигурирование JSF-приложения

## 5.3 Физическая структура баз данных

## 5.3.1 База данных Customer

## 5.3.2 База данных Indent

## 5.4 Обеспечение целостности данных

## 5.4.1 Ограничения целостности

## 5.4.2 Организация распределенных транзакций

## 5.5 Реализация взаимодействия с внешним сервисом

## 5.5.1 Протокол и форматы обмена данными

# **6 ИНСТРУКЦИЯ ПО УСТАНОВКЕ ПРИЛОЖЕНИЯ**

Проект размешен на ресурсе GitHub и имеет возможность редактирования и просмотра исходного кода. Далее будет описана инструкция для возможности запуска и компиляции исходного проекта.

Для запуска проекта необходимы следующие зависимости:

Для backend:

1. Java 17
2. PostgreSQL

Для frontend:

1. Node.js
2. npm

Далее необходимо выполнить следующие действия:

1) Клонировать или скачать исходный код с репозитория. https://github.com/javavlsu/course-work-team-1917-karabanov-bulanov

2) Перед настройкой подключения к БД необходимо создать базу данных в используемой СУБД. В каталоге «/backend/esap/src/main/resources/» в файле «application.properties» указать созданную базу данных и логин с паролем для PostgreSQL. Дефолтные значение указаны в файле. Пример свойств подключения БД:

* spring.datasource.url=jdbc:postgresql://localhost:5432/esap
* spring.datasource.username=postgres
* spring.datasource.password=postgres

3) Собрать проект с помощью Maven. В каталоге «/backend/esap/» открыть консоль и запустить скрипт maven c командой package.

4) В создавшемся каталоге «\backend\esap\target» будет находиться файл «esap-0.0.1-SNAPSHOT.jar» его необходимо запустить с помощью java с флагом -jar.

На этом запуск backend части закончен. Теперь необходимо запустить frontend.

5) В каталоге «frontend/esap-admin» запустить консоль и выполнить команды «npm install».

6) Затем в этом же каталоге выполнить команду «npm run start»

После этого можно заходить на url «http://localhost:3000/».

# **7 ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ ПРОЕКТОМ**

## 7.1 Общие принципы взаимодействия в команде

## 7.2 Распределение ролей в команде/зоны ответственности

## 7.3 Календарный план работ

Календарный план работ представлен на диаграмме Ганта.

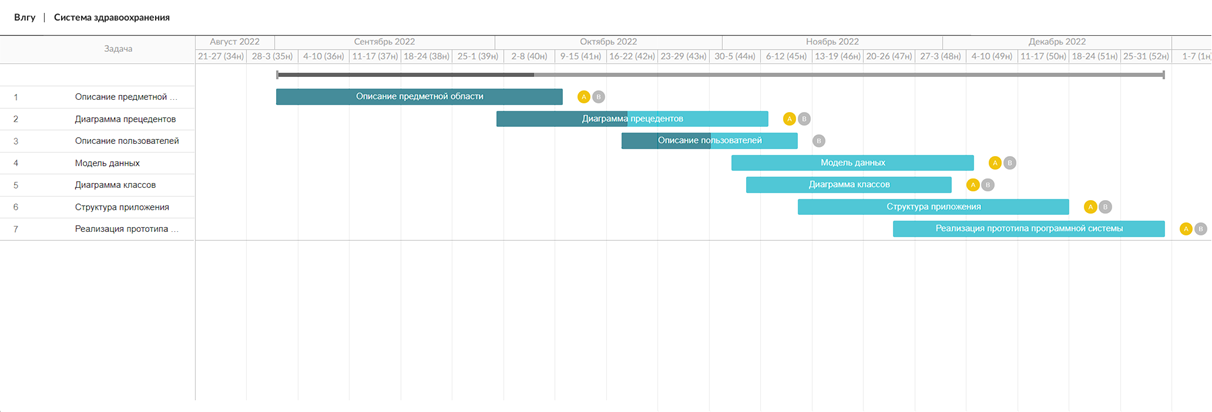


Рис. 1. Диаграмма Ганта.

# **8 НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ**

# **9 ЗАКЛЮЧЕНИЕ**

В рамках данного проекта был разработан прототип программной системы «Единая система автоматизации поликлиник (ЕСАП)» на платформе Java EE c использованием фреймворка Spring Boot и библиотеки React TS для реализации клиентской части. Целью данного проекта было создание прототипа системы, которая позволяет эффективно управлять пациентами, записями, расписаниями и медицинскими данными.

В процессе разработки были определены требования к системе, проведен анализ существующих аналогов и выбраны технологии, наиболее подходящие для реализации поставленных задач. Были применены современные подходы и принципы разработки, такие как модульность, масштабируемость и безопасность, что позволило создать гибкую и надежную систему.

В ходе разработки были реализованы основные функциональные возможности системы.

Полученный прототип системы был протестирован, и в ходе тестирования выявлены и исправлены некоторые ошибки и недочеты. Были проведены тесты на производительность, которые показали хорошую отзывчивость системы даже при большом количестве одновременных запросов.

В заключение можно сказать, что разработанная система «Единая система автоматизации поликлиник» полностью соответствует поставленным целям и требованиям. Прототип системы готов к дальнейшей разработке и расширению функциональности, и может служить основой для создания полноценного коммерческого приложения.

**8 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Постановление Правительства РФ «О единой государственной информационной системе в сфере здравоохранения» от 09.02.2022 № 140 // Docs.cntd.ru — электронный фонд нормативно-технической и нормативно-правовой информации Консорциума «Кодекс» - 11.02.202 URL: <https://docs.cntd.ru/document/728122057> (дата обращения: 25.02.2023).
2. Приказ Министерства Здравоохранения РФ «Об утверждении Требований к государственным информационным системам в сфере здравоохранения субъектов Российской Федерации, медицинским информационным системам медицинских организаций и информационным системам фармацевтических организаций» от 24.12.2018 № 911н // Docs.cntd.ru — электронный фонд нормативно-технической и нормативно-правовой информации Консорциума «Кодекс» - 19.06.2019 URL: <https://docs.cntd.ru/document/552196746> (дата обращения: 25.02.2023).
3. БАРС.Здравоохранение - МИС // Барс Груп URL: <https://bars.group/health/meditsinskaya-informatsionnaya-sistema> (дата обращения: 25.02.2023).
4. ИНФОКЛИНИКА // «Смарт Дельта Системс» URL: <https://sdsys.ru/products/infoclinica> (дата обращения: 25.02.2023).
5. МИС МЕДИАЛОГ // Компания «Пост Модерн Текнолоджи» (ПМТ) URL: <https://medialog.ru> (дата обращения: 25.02.2023).
6. 1С:Медицина. Поликлиника // Solutions.1c.ru — это ресурс фирмы "1С", на котором собрана информация обо всех отраслевых и специализированных решениях "1С:Предприятие 8". URL: <https://solutions.1c.ru/catalog/clinic/features> (дата обращения: 25.02.2023).

# **ПРИЛОЖЕНИЕ А ЛИСТИНГ КОМПОНЕНТОВ СИСТЕМЫ**

# **ПРИЛОЖЕНИЕ Б СКРИПТ СОЗДАНИЯ БАЗЫ ДАННЫХ**