

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

(код и наименование направления подготовки, специальности)

Прикладная информатика в социальной сфере

(направленность(профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка информационной системы подбора гомеопатических
препаратов для ООО ГЦ «Биоравновесие»

Студент

А.В. Погодин

(И.О. Фамилия)

(личная подпись)

Руководитель

к.т.н., О.В. Аникина

(ученая степень, звание, И.О. Фамилия)

Консультант

А.В. Москалюк

(И.О. Фамилия)

Тольятти 2020

АННОТАЦИЯ

Тема: Разработка информационной системы подбора гомеопатических препаратов для ООО ГЦ «Биоравновесие».

Ключевые слова: информационная система, подбор гомеопатических препаратов.

Целью выпускной квалификационной работы является разработка информационной системы (ИС) подбора гомеопатических препаратов для ООО ГЦ «Биоравновесие».

Объект исследования - система подбора гомеопатических препаратов.

Предметом исследования является информационная система подбора гомеопатических препаратов.

Задачами ВКР являются:

- проанализировать предметную область подбора гомеопатических препаратов и организационную структуру организации ООО ГЦ «Биоравновесие»;

- выделить требования разрабатываемой информационной системы;

- выбрать средства проектирования и разработки;

- спроектировать информационную систему;

- разработать информационную систему;

В аналитической части была разработана BPMN модель процесса подбора гомеопатических препаратов.

На стадии концептуального проектирования сформулированы требования к разрабатываемой ИС. На стадии логического проектирования на основе объектно-ориентированного подхода разработана логическая модель ИС.

Выполнена реализация и тестирование ИС подбора гомеопатических препаратов.

Работа включает: 58 страниц с приложением, 29 рисунков, 8 таблиц, источников 30.

ABSTRACT

The title of the graduation work is *Development of an information system for selecting homeopathic medicines*.

The graduation work consists of an explanatory note on 56 pages, including 28 figures, 8 tables, the list of 30 references including 8 foreign sources.

The graduation work may be divided into several logically connected parts, which are dedicated to functional modeling, logical modeling and physical modeling of the information system for selecting homeopathic medicines.

The process of selecting homeopathic medicines, which is considered in this paper, is a time-consuming task. Therefore, the development and implementation of an information system will simplify the process of selecting homeopathic medicines and improve its effectiveness.

We examine how selection of homeopathic medicines works. We then analyze some ready-made solutions of the existing systems.

The issues of functional modeling, logical modeling and physical modeling of the information system are highlighted in the work along with details about development of the system.

The following tasks are completed: the analysis of the subject area and the structure of the organization where the system is subject to be implemented; the study of the requirements of the developed system; selecting design and development tools; design and development of the information system.

Testing has shown that the developed information system for selecting homeopathic medicines fully meets the customer's requirements.

In the future, it is planned to develop the project by adding new drugs and symptoms to the database and adding new functionality that would further make the information system a flexible and convenient tool for doctors.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
Глава 1 Функциональное моделирование информационной системы подбора гомеопатических препаратов	6
1.1 Характеристика организации ООО ГЦ «Биоравновесие».....	6
1.2 Концептуальное моделирование процесса подбора гомеопатических препаратов.....	7
1.2.1 Выбор технологии концептуального моделирования предметной области	7
1.2.2 Разработка модели бизнес-процесса подбора гомеопатических препаратов «КАК ЕСТЬ».....	9
1.2.3 Разработка модели бизнес-процесса подбора гомеопатических препаратов «КАК ДОЛЖНО БЫТЬ»	11
1.3 Постановка задачи на разработку информационной системы подбора гомеопатических препаратов	12
1.4 Анализ аналогов информационной системы подбора гомеопатических препаратов.....	13
Вывод по главе 1	16
Глава 2 Логическое проектирование информационной системы подбора гомеопатических препаратов	17
2.1 Логическая архитектура и варианты использования информационной системы подбора гомеопатических препаратов.....	17
2.2 Диаграмма классов информационной системы подбора гомеопатических препаратов.....	20
2.3 Диаграмма последовательности информационной системы подбора гомеопатических препаратов	23
2.4 Проектирование веб-интерфейса.....	25
Вывод по главе 2	27
Глава 3 Физическое проектирование информационной системы подбора гомеопатических препаратов	28

3.1 Выбор средств реализации информационной системы подбора гомеопатических препаратов	28
3.2 Разработка модели данных информационной системы подбора гомеопатических препаратов	29
3.3 Разработка серверного приложения	32
3.4 Разработка клиентского приложения	39
3.5 Описание функциональности информационной системы	45
3.6 Тестирование информационной системы	52
Вывод по главе 3	55
ЗАКЛЮЧЕНИЕ	56
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	58
ПРИЛОЖЕНИЕ А Программный код	61

ВВЕДЕНИЕ

В настоящее время информационные технологии бурно развиваются. Они помогают людям во всех сферах жизни. Уже невозможно представить работу предприятия без информационной системы.

Процесс подбора гомеопатических препаратов, который рассматривается в данной работе, очень кропотливая работа. Поэтому разработка информационной системы позволит упростить процесс подбора гомеопатических препаратов и улучшить его эффективность.

Руководством медицинского центра ООО ГЦ «Биоравновесие» было принято решение о разработке информационной системы подбора гомеопатических препаратов.

Объект исследования ВКР - система подбора гомеопатических препаратов.

Предмет исследования - автоматизация процесса подбора гомеопатических препаратов.

Целью работы является разработка информационной системы подбора гомеопатических препаратов.

Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать предметную область подбора гомеопатических препаратов и организационную структуру организации ООО ГЦ «Биоравновесие»;
- выделить требования разрабатываемой информационной системы;
- выбрать средства проектирования и разработки;
- спроектировать информационную систему;
- разработать информационную систему;

В первой главе произведено функциональное моделирование информационной системы подбора гомеопатических препаратов. Описана характеристика организации, разработана и проанализирована модель бизнес-

процесса подбора гомеопатических препаратов. Выработаны требования к разрабатываемой ИС. На основе этих требований проанализированы аналоги ИС подбора гомеопатических препаратов.

Во второй главе произведено логическое моделирование информационной системы подбора гомеопатических препаратов. Построена логическая архитектура ИС. Спроектирован веб-интерфейс информационной системы.

В третьей главе произведено физическое моделирование информационной системы подбора гомеопатических препаратов. Разработаны серверное и клиентское приложения. Описаны функциональные возможности разработанной информационной системы подбора гомеопатических препаратов. Также произведено тестирование разработанной ИС.

В заключении подведены итоги исследования. Описаны выполненные задачи. Сформулированы окончательные выводы по рассматриваемой цели.

Разработанная информационная система позволит автоматизировать процесс подбора гомеопатических препаратов.

Глава 1 Функциональное моделирование информационной системы подбора гомеопатических препаратов

1.1 Характеристика организации ООО ГЦ «Биоравновесие»

ООО Гомеопатический Центр «Биоравновесие», открылся в 2004 году в городе Тольятти. Данный медицинский центр предоставляет следующие услуги:

- врач-гомеопат;
- педиатр, терапевт, остеопат;
- определение причины заболевания;
- выявление аллергенов;
- подбор индивидуальной схемы лечения;
- оздоровительная программа для часто болеющих детей;
- биорезонансная терапия;
- гомеопатическое лечение;
- кислородотерапия, медицинский массаж;
- тестирование на переносимость и эффективность лекарственных

препаратов, косметических средств.

Юридический адрес организации ООО ГЦ «Биоравновесие»: г. Тольятти, ул. Свердлова, дом 24А, 2 этаж. Веб-сайт медицинского центра: <https://биоравновесие.рф>.

Организационная структура ООО ГЦ «Биоравновесие» представлена в виде блок-схемы, на рисунке 1.1.

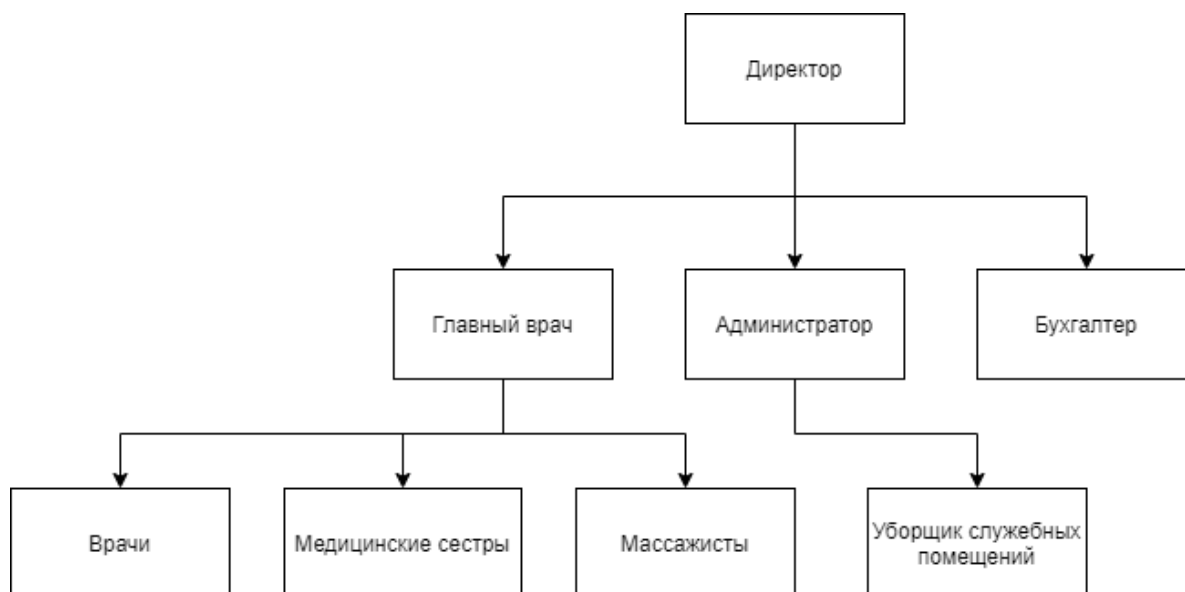


Рисунок 1.1 - Организационная структура организации
ООО ГЦ «Биоравновесие»

На основании рисунка 1.1 видно, что данная организация является небольшой. Во главе стоит директор, который руководит всем персоналом.

Главный врач непосредственно подчиняется директору. В его подчинении находятся врачи, медицинские сестры и массажисты. Деятельность врача и главного врача, а именно процесс подбора гомеопатических препаратов, подлежит автоматизации.

Бухгалтер занимается всеми финансовыми вопросами организации и непосредственно подчиняется директору.

1.2 Концептуальное моделирование процесса подбора гомеопатических препаратов

1.2.1 Выбор технологии концептуального моделирования предметной области

Организации используют моделирование бизнес-процессов (Business Process Modeling) для визуального документирования, понимания и совершенствования своих бизнес-процессов.

Как часть методологии реинжиниринга и управления бизнес-процессами (Business Process Management) моделирование бизнес-процессов (МБП) используется в качестве организационного инструмента для анализа

существующего состояния бизнес-процесса («как есть») и для определения его будущего состояния («как должно быть»), а контексте решения задач улучшения данного бизнес-процесса.

К достоинствам МБП также относятся:

- возможность создание визуальных моделей процессов. Документация, основанная на вербальной основе, недостаточно точно и наглядно отражает бизнес-процесс. Визуальные модели помогают обеспечить полную картину производства товара или услуги;

- упорядочивание операций. При любой новой бизнес-стратегии непрерывное отслеживание процессов после изменения требует определения того, как оставаться в рамках общей организационной стратегии. Анализы также выполняются для выявления узких мест и неэффективности, и обеспечения гибкости процесса;

- улучшение связи между процессами. Коммуникация является ключом ко всем из следующих задач: формализация существующих процессов, согласование процессов, устранение недоработок с бизнес-правилами, обработка исключений, обеспечение соблюдения нормативных требований и т.д.;

- повышение эффективности работы. Моделирование процессов способствует оптимизации, позволяя моделировать и иллюстрировать необходимые улучшения. Это сокращает время проектирования и способствует лучшему использованию ресурсов;

- получение конкурентных преимуществ. Процесс лучше в целом, когда он постоянно совершенствуется и согласовывается с бизнес-стратегиями менеджмента предприятия.

Для моделирования бизнес-процесса подбора гомеопатических препаратов выбираем нотация для моделирования бизнес-процессов BPMN, так как она наиболее современная и активно развивающаяся. Нотацию BPMN выбирает большинство профессионалов в управлении бизнес-процессами.

Спецификация BPMN описывает условные обозначения и их описание в XML для отображения бизнес-процессов в виде диаграмм бизнес-процессов. BPMN ориентирована как на технических специалистов, так и на бизнес-пользователей. Для этого язык использует базовый набор интуитивно понятных элементов, которые позволяют определять сложные семантические конструкции. Кроме того, спецификация BPMN определяет, как диаграммы, описывающие бизнес-процесс, могут быть трансформированы в исполняемые модели.

Событийный подход рассматривает модель как систему процессов. В его основе лежит концепция управления бизнес-процессами посредством дискретной передачи им информации об изменениях, происходящих внутри процесса или в других процессах. Модели в нотации BPMN позволяют увидеть и проанализировать все взаимодействия. Набор знаков в BPMN достаточен для описания любого процесса и обозначения любых типов событий.

1.2.2 Разработка модели бизнес-процесса подбора гомеопатических препаратов «КАК ЕСТЬ»

Модель деятельности «КАК ЕСТЬ» представляет собой «снимок» положения дел в организации на момент обследования и позволяет понять, что делает и как функционирует организация с позиций системного анализа, выявить ряд ошибок и узких мест и сформулировать предложения по улучшению ситуации.

Подбор гомеопатических препаратов очень кропотливый процесс, который требует от врача большого внимания и времени. Для того, чтобы подобрать гомеопатический препарат, врач, помимо рекомендаций медицинского ПО, очень часто обращается к реперторию на бумажном носителе, либо в электронном виде. Реперторий - это клинический гомеопатический справочник патологических симптомов и синдромов, их дифференциально-диагностических признаков и миазматических особенностей. Для того, чтобы подобрать препараты, нужно выполнить следующие действия:

- найти нужную главу;
- найти симптомы и препараты для них;
- найти общие препараты;

На основе этих действий была построена модель BPMN бизнес-процесса подбора гомеопатических препаратов «КАК ЕСТЬ». На рисунке 1.2 показана построенная модель BPMN бизнес-процесса подбора гомеопатических препаратов «КАК ЕСТЬ».

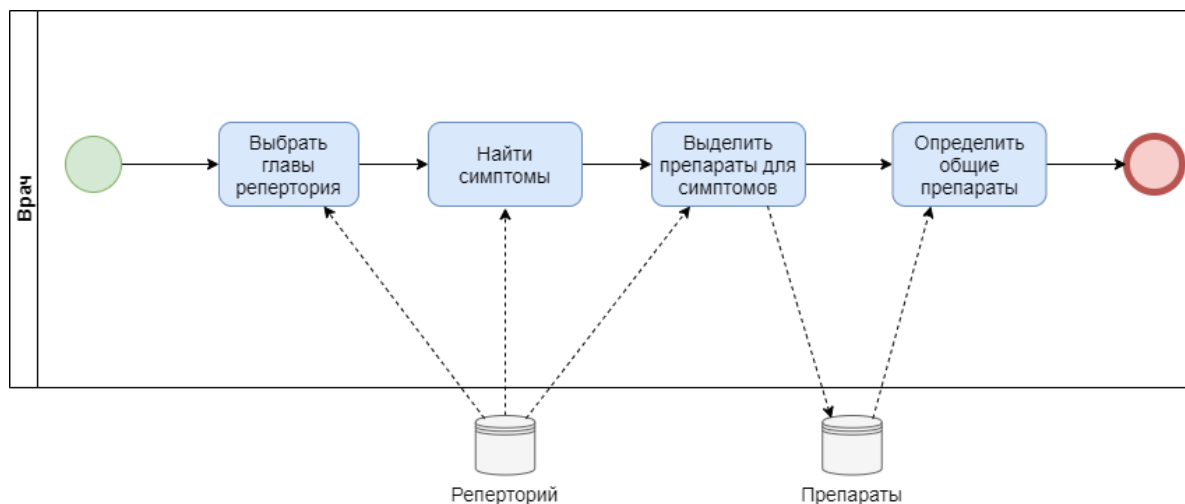


Рисунок 1.2 - Модель BPMN бизнес-процесса подбора гомеопатических препаратов «КАК ЕСТЬ»

При анализе модели бизнес-процесса «КАК ЕСТЬ» были выявлены следующие недостатки:

- 1) процесс подбора гомеопатический препаратов не автоматизирован, что снижает его надежность из-за влияния человеческого фактора, при выполнении рутинных операций вручную;
- 2) отсутствует собственная база данных препаратов;
- 3) отсутствует собственная база данных симптомов;
- 4) общие препараты определяются вручную.

Основные задачи автоматизации заключаются в:

- устранении рутинных операций;
- ускорении процессов обработки и преобразования информации;
- снижение человеческого фактора в процессе подбора гомеопатических препаратов;

- повышении эффективности подбора гомеопатических препаратов.

Поэтому исследуемый процесс будет автоматизирован в рамках данной работы.

С учетом вышеизложенного принято решение улучшить существующий бизнес-процесс подбора гомеопатических препаратов путем его автоматизации.

1.2.3 Разработка модели бизнес-процесса подбора гомеопатических препаратов «КАК ДОЛЖНО БЫТЬ»

С учетом анализа модели бизнес-процесса «КАК ЕСТЬ» разработана модель бизнес-процесса подбора гомеопатических препаратов «КАК ДОЛЖНО БЫТЬ», представленная на рисунке 1.3.

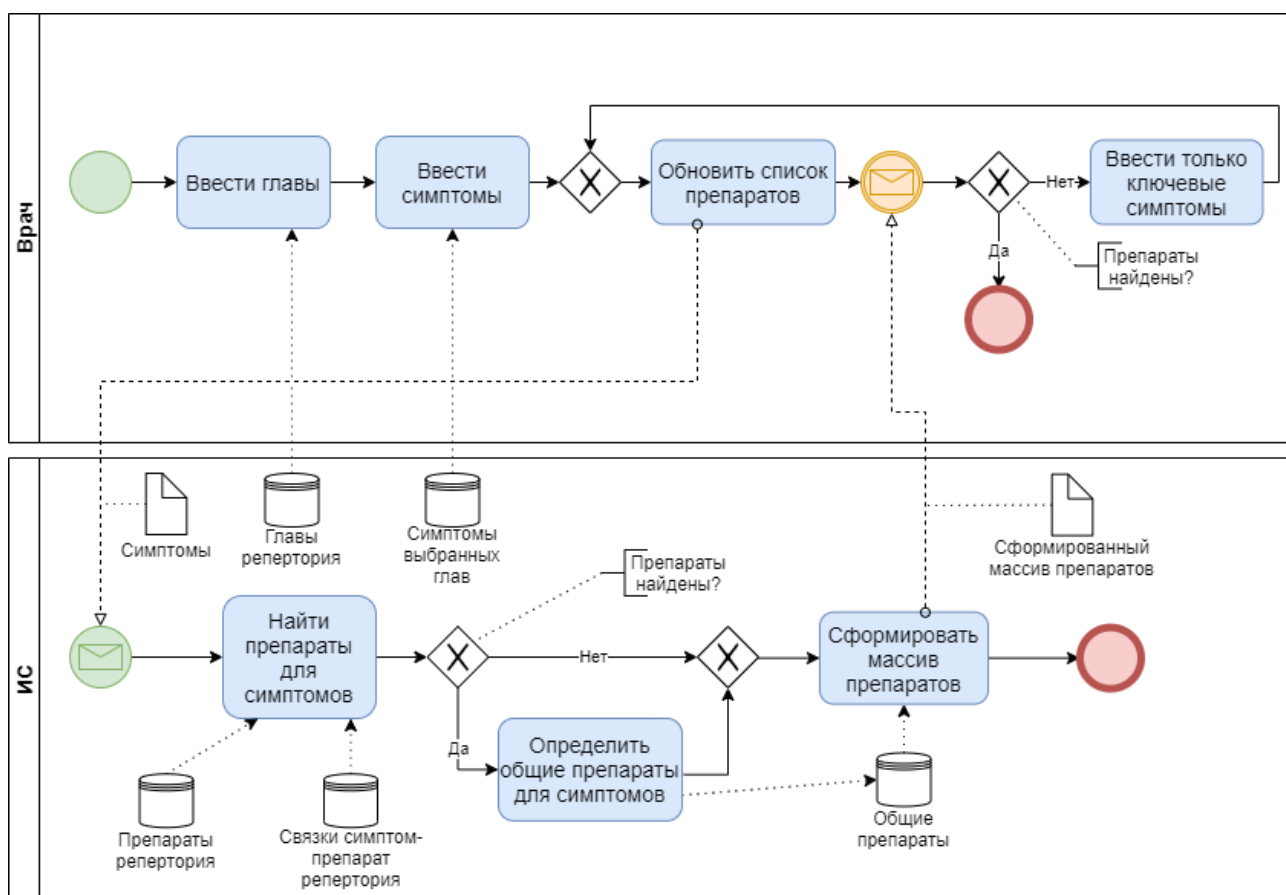


Рисунок 1.3 - Модель BPMN бизнес-процесса подбора гомеопатических препаратов «КАК ДОЛЖНО БЫТЬ»

Бизнес-процесс подбора гомеопатических препаратов «КАК ДОЛЖНО БЫТЬ» организован следующим образом:

- 1) врач выбирает главы репертория;

- 2) врач вводит симптомы пациента;
- 3) если препараты не найдены, то врач вводит только ключевые симптомы;
- 4) врач получает список подобранных препаратов для выбранных симптомов.

Процесс завершен.

Данная модель представляет собой концептуальную модель бизнес-процесса подбора гомеопатических препаратов и является основой для логического и физического моделирования ИС.

1.3 Постановка задачи на разработку информационной системы подбора гомеопатических препаратов

Требования к разрабатываемой информационной системе, предоставлены заказчиком. Задача была выработана и скорректирована в ходе дискуссии, в которой участвовали разработчик и директор.

Основные требования:

- минимальные затраты на разработку, внедрение и сопровождение ИС;
- простота использования ИС;
- не допускать избытка функциональности ИС;
- простота внедрения ИС;
- решение должно иметь адаптивный веб интерфейс;
- масштабируемость системы;
- ИС должна использовать собственную базу данных препаратов и симптомов;
- авторизация/регистрация пользователей;
- возможность добавления новых препаратов;
- возможность добавления симптомов;
- возможность создания новых связок симптом-препарат;

Процесс подбора гомеопатических препаратов в проектируемой ИС выполняется следующим образом:

- 1) врачу нужно подобрать гомеопатические препараты, для этого он заходит в приложение;
- 2) если он не зарегистрирован, то он регистрирует новый аккаунт;
- 3) проходит процесс авторизации, путем ввода логина и пароля;
- 4) на главной странице репертория вводит нужные главы;
- 5) вводит симптомы пациента;
- 6) получает список общих гомеопатических препаратов для выбранных симптомов.

Также стоит обратить внимание на то, что для врача важно иметь возможность добавлять в базу данных репертория новые препараты, симптомы и связки симптом-препарат, основываясь на своем опыте.

Будущая ИС должна упростить подбор гомеопатических препаратов по симптомам, позволить врачу добавлять новые препараты, симптомы и новые связки симптом-препарат. Снижение человеческого фактора позволит свести к минимуму количество ошибок, при подборе гомеопатических препаратов.

Разрабатываемую ИС планируется использовать для частых случаев, постепенно добавляя препараты, симптомы и связки симптом-препарат. Требования согласованы с директором организации ООО ГЦ «Биоравновесие».

Задача: автоматизировать процесс подбора гомеопатических препаратов, путем разработки информационной системы подбора гомеопатических препаратов.

1.4 Анализ аналогов информационной системы подбора гомеопатических препаратов

Рассмотрим и сравним известные аналоги ИС подбора гомеопатических препаратов.

В результате поисков аналогов разработки было выделено несколько продуктов, которые в большей или меньшей степени удовлетворяют требованиям:

- Киберис;

- Радуга-5.

Киберис - это гомеопатическая медицинская система. В таблице 1.1 отражена информация о соответствии требованиям данного решения.

Таблица 1.1 - Соответствие требованиям системы «Киберис»

Требование	Киберис
минимальные затраты на разработку, внедрение и сопровождение ИС	+
простота использования ИС	+
не допускать избытка функциональности АИС	-
простота внедрения ИС	-
решение должно иметь адаптивный веб интерфейс	-
масштабируемость системы	-
ИС должна использовать собственную базу данных препаратов и симптомов	-
авторизация/регистрация пользователей	+
возможность добавления новых препаратов	-
возможность добавления симптомов	-
возможность создания новых связей симптом-препарат	-

Радуга-5 - это российская гомеопатическая программа, предназначенная для работы как на приеме с пациентом, так и для заочного анализа случаев. В таблице 1.2 отражена информация о соответствии требованиям данного решения.

Таблица 1.2 - Соответствие требованиям программы «Радуга-5»

Требование	Радуга-5
минимальные затраты на разработку, внедрение и сопровождение ИС	-
простота использования ИС	-
не допускать избытка функциональности АИС	-
простота внедрения ИС	-
решение должно иметь адаптивный веб интерфейс	-
масштабируемость системы	-
ИС должна использовать собственную базу данных препаратов и симптомов	-
авторизация/регистрация пользователей	+
возможность добавления новых препаратов	-
возможность добавления симптомов	-

Продолжение таблицы 1.2

возможность создания новых связей симптом-препарат	-
--	---

На таблице 1.3 продемонстрирован сравнительный анализ аналогов ИС подбора гомеопатических препаратов

Таблица 1.3 - Сравнительный анализ аналогов ИС

Требование	Радуга-5	Киберис
минимальные затраты на разработку, внедрение и сопровождение ИС	-	+
простота использования ИС	-	+
не допускать избытка функциональности АИС	-	-
простота внедрения ИС	-	-
решение должно иметь адаптивный веб интерфейс	-	-
масштабируемость системы	-	-
ИС должна использовать собственную базу данных препаратов и симптомов	-	-
авторизация/регистрация пользователей	+	+
возможность добавления новых препаратов	-	-
возможность добавления симптомов	-	-
возможность создания новых связей симптом-препарат	-	-
Итого (+)	2	3

В результате анализа готовых решений, как следует из таблицы 1.3, выявлено, что ни одно приложение в полной мере не удовлетворяет требованиям, предъявляемых к ИС, а именно под требования «ИС должна использовать собственную базу данных препаратов и симптомов» и «не допускать избытка функциональности АИС» не попадает ни одно из вышеописанных решений.

Также стоит обратить внимание на то, что организации нужна ИС, которую планируется использовать только для наиболее частых случаев совместно с другим медицинским ПО, поэтому внедрять огромную

дорогостоящую ИС не имеет смысла, так как врач не будет использовать большую часть ее функций.

На основании анализа аналогов ИС подбора гомеопатических препаратов было принято решение о разработке информационной системы подбора гомеопатических препаратов.

Вывод по главе 1

Была проанализирована структура организации ООО ГЦ «Биоравновесие». Описана краткая характеристика организации.

В результате анализа был выявлен процесс, требующий автоматизации.

Проведен анализ существующего бизнес-процесса «КАК ЕСТЬ», в ходе которого были выявлены недостатки.

С учетом анализа модели бизнес-процесса «КАК ЕСТЬ», была разработана модель BPMN процесса подбора гомеопатических препаратов «КАК ДОЛЖНО БЫТЬ». На основе которой будет спроектирована информационная система.

Совместно с руководством организации были выработаны требования к разрабатываемой информационной системе. На основе этих требований были проанализированы готовые решения. Результат анализа показал, что ни одно из описанных решений не удовлетворяет в полной мере требованиям. Вследствие чего было принято решение о начале проектирования информационной системы подбора гомеопатических препаратов для организации ООО ГЦ «Биоравновесие».

Глава 2 Логическое проектирование информационной системы подбора гомеопатических препаратов

2.1 Логическая архитектура и варианты использования информационной системы подбора гомеопатических препаратов

В самом начале была построена логическая архитектура будущей ИС. На ее основе будет происходить дальнейшая разработка взаимодействия компонент системы. На рисунке 2.1 показана спроектированная архитектура ИС.

Логическая архитектура выделяет, прежде всего, вопросы взаимодействия компонент системы, интерфейсы и используемые протоколы.

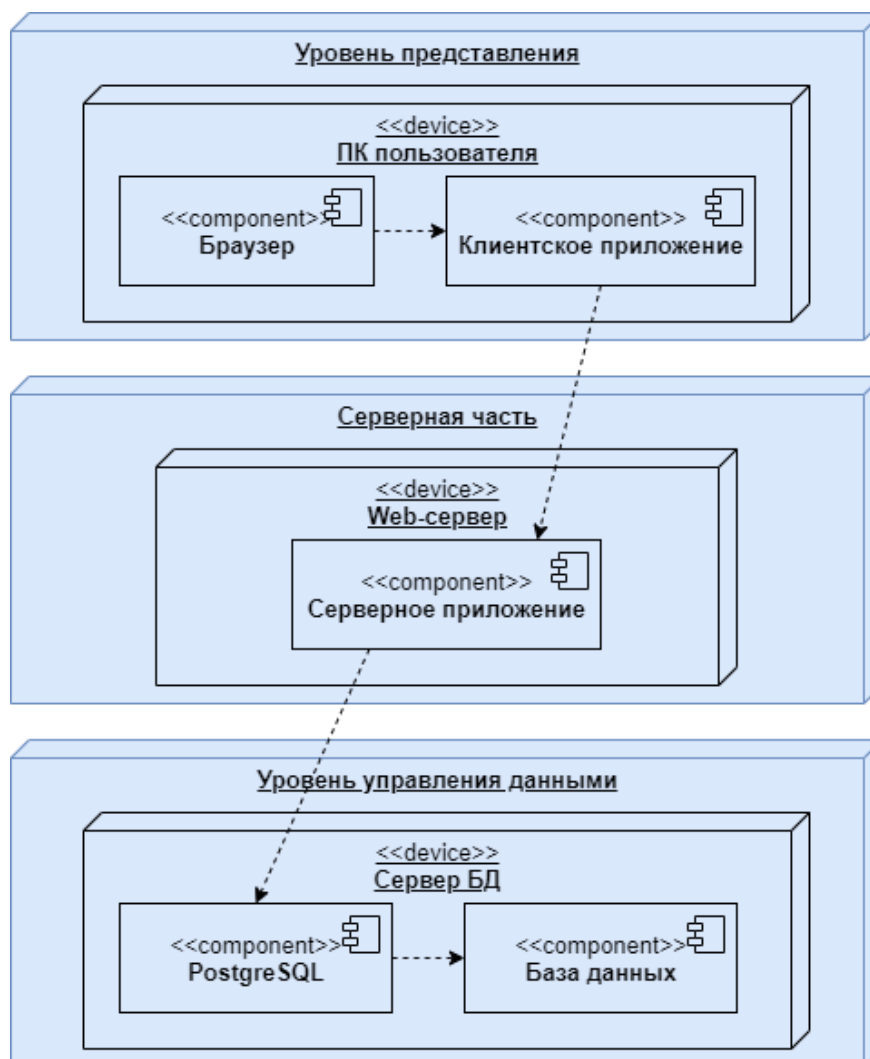


Рисунок 2.1 - Трехзвенная архитектура ИС

Унифицированный язык моделирования (Unified Modeling Language, UML) - это графический язык для визуализации, специфицирования, конструирования и документирования систем, в которых главная роль принадлежит программному обеспечению.

Конструкция или стандартный элемент языка UML вариант использования применяется для спецификации общих особенностей поведения системы или любой другой сущности предметной области без рассмотрения внутренней структуры этой сущности.

Диаграмма вариантов использования отражает общее описание возможных функций, которые доступны в системе для актеров, а именно для врача. На рисунке 2.2 продемонстрирована построенная диаграмма прецедентов процесса подбора гомеопатических препаратов.

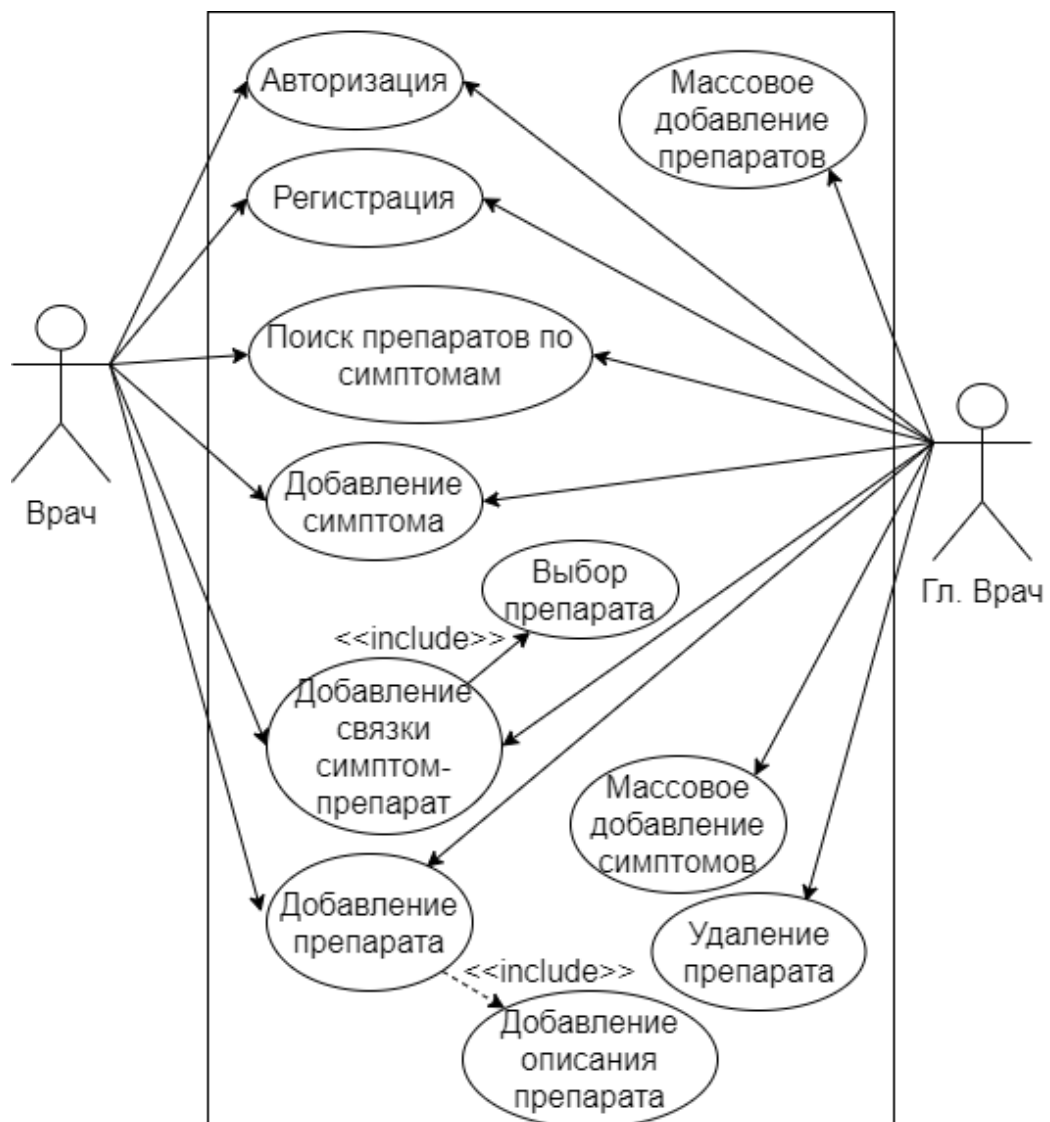


Рисунок 2.2 - Диаграмма прецедентов процесса подбора гомеопатических препаратов

В таблице 2.1 отражено краткое описание вариантов использования АИС.

Таблица 2.1 - Описание прецедентов

ID	Название прецедента	Описание
1	Авторизация	Авторизация в приложении, путем ввода логина и пароля
2	Регистрация	Регистрация в приложении
3	Поиск препаратов по симптомам	Врач осуществляет поиск препаратов для выбранных симптомов
4	Добавление симптома	Врач может добавить симптом в базу данных репертория
5	Добавления препарата	Врач может добавить препарат в базу данных репертория
6	Добавление связки симптом-препарат	Врач может создать новые связки симптом-препарат
7	Удаление препарата	Главный врач может удалить препарат из базы данных
8	Массовое добавление препаратов	Главный врач может использовать POST-запрос, передавая json массив препаратов в теле запроса, для массового создания препаратов
9	Массовое добавление симптомов	Главный врач может использовать POST-запрос, передавая json массив симптомов в теле запроса, для массового создания симптомов

Как показано на диаграмме, врач должен иметь возможность:

- регистрация/авторизация;
- поиск гомеопатических препаратов по симптомам;
- добавлять симптомы в базу данных ИС;
- добавлять препараты в базу данных ИС;
- создавать новые связки симптом-препарат.

Главный врач, помимо возможностей врача, должен иметь возможность:

- удалять препараты;
- массово добавлять препараты с помощью POST-запроса;
- массово добавлять симптомы с помощью POST-запроса.

Ниже описаны спецификации некоторых прецедентов в таблицах:

Таблица 2.2 - Прецедент: Поиск препаратов по симптомам

Прецедент: Поиск препаратов по симптомам
ID: 3

Краткое описание: Врач осуществляет поиск препаратов для выбранных симптомов
Главные актеры: Врач
Второстепенные актеры: нет
Продолжение таблицы 2.2
Предусловия: <ol style="list-style-type: none"> 1. Авторизация прошла успешно. 2. Актер выбрал главы репертория. 3. Актер выбрал симптомы.
Основной поток: <ol style="list-style-type: none"> 1. Система ищет препараты для выбранных симптомов. 2. Система находит общие препараты для симптомов, не имеющих дочерних симптомов. 3. Система отправляет список препаратов актеру.
Постусловия: Система сформировала список препаратов и отправила его актеру
Альтернативные потоки: нет

Таблица 2.3 - Прецедент: Добавление связки симптом-препарат

Прецедент: Добавление связки симптом-препарат
ID: 6
Краткое описание: Врач может создать новые связки симптом-препарат
Главные актеры: Врач
Второстепенные актеры: нет
Предусловия: <ol style="list-style-type: none"> 1. Авторизация прошла успешно. 2. Актер выбрал главы репертория. 3. Актер выбрал симптомы. 4. Актер выбрал препарат для добавления его к выбранным симптомам.
Основной поток: <ol style="list-style-type: none"> 1. Система находит препарат и симптомы в базе данных. 2. Система добавляет препарат для выбранных симптомов. 3. Система отправляет результат выполнения запроса актеру.
Постусловия: Система вернула результат выполнения запроса актеру
Альтернативные потоки: нет

Диаграмма вариантов использования будет использоваться как основная при разработке функциональности.

2.2 Диаграмма классов информационной системы подбора гомеопатических препаратов

Диаграмма классов - это тип диаграмм UML, который определяет и предоставляет обзор и структуру системы с точки зрения классов, атрибутов и методов, а также отношений между различными классами.

Она используется для иллюстрации и создания функциональной схемы классов системы и используется в течение всего жизненного цикла разработки ее программного обеспечения.

Иными словами, диаграмма классов в первую очередь предназначена для разработчиков, чтобы обеспечить концептуальную модель и архитектуру разрабатываемой системы

Диаграмма классов моделирует отношения между классами, объектами, атрибутами и операциями.

Классы представляют собой абстракцию объектов с общими характеристиками. Связи между ними представляют собой отношения между классами.

На рисунке 2.3 представлена диаграмма классов ИС подбора гомеопатических препаратов.

Класс «Симптомы» включает в себя атрибуты: «ID симптома», «название», «ID родительского симптом», «ID категории» и метод «создать симптом».

Класс «Категории» описывает главы репертория и включает в себя атрибуты: «ID категории», «название», «описание» и метод «создать категорию».

Класс «Пользователи» включает в себя атрибуты: «ID пользователя», «электронная почта», «имя», «пароль» и «ID роли».

Класс «Связки симптом-препарат» включает в себя атрибуты: «ID связки», «ID симптома», «ID препарата», «тип» и метод «создать связку».

Класс «Препараты» включает в себя атрибуты: «ID препарата», «название», «описание» и методы «создать препарат», «удалить препарат».

Класс «Роль» включает в себя атрибуты: «ID роли», «тип».

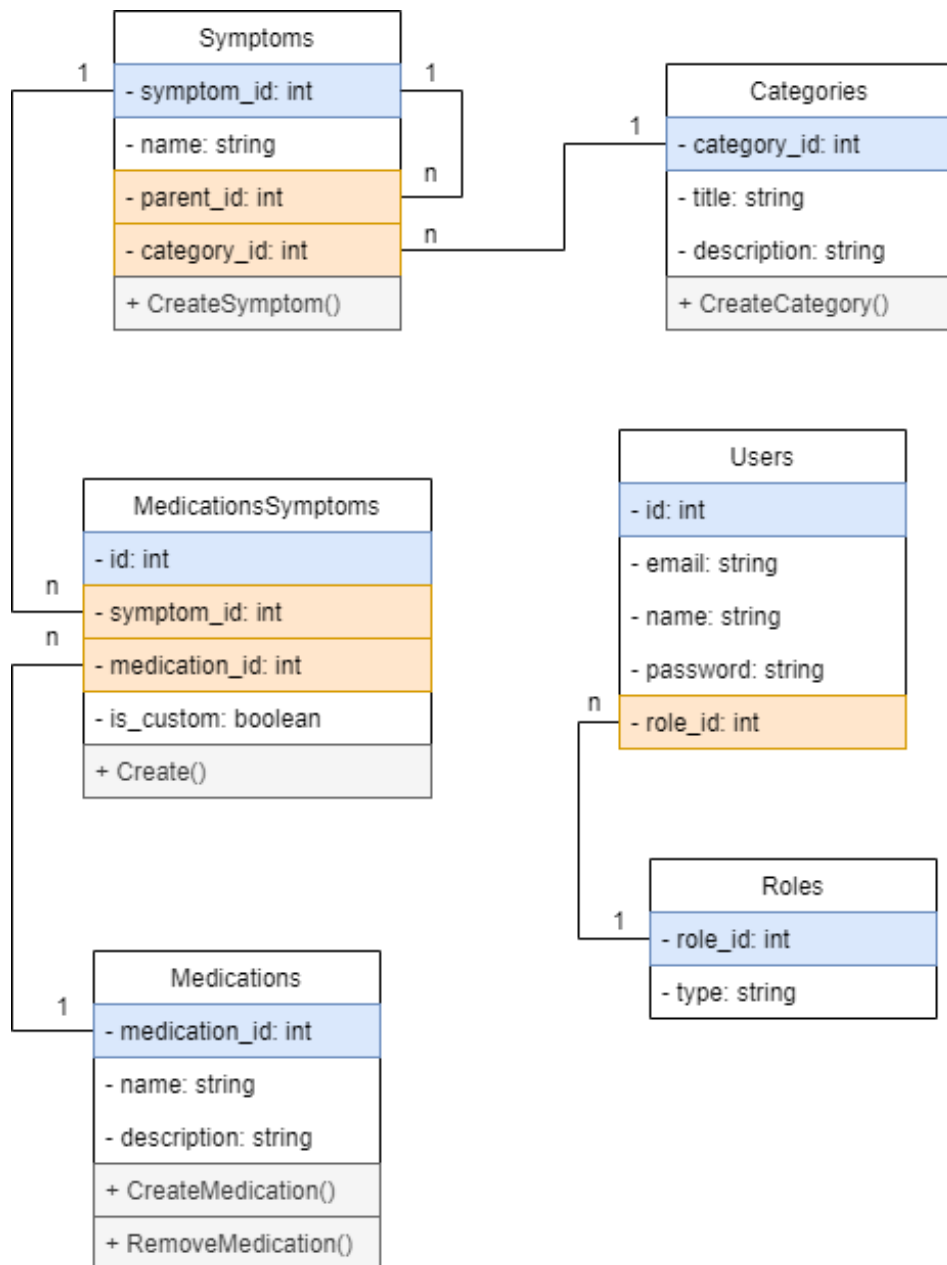


Рисунок 2.3 - Диаграмма классов ИС

Спецификация диаграммы классов:

«Users» - класс объектов - пользователей ИС.

«Roles» - класс объектов - ролей пользователей ИС.

«Categories» - класс объектов - категорий/глав репертория.

«Symptoms» - класс объектов - СИМПТОМОМ.

«Medications» - класс объектов - препаратов.

«MedicationsSymptoms» - класс объектов - связей симптом-препарат.

Связи между классами - именованные ассоциации. Диаграмма классов отражает статистический аспект ИС подбора гомеопатических препаратов.

2.3 Диаграмма последовательности информационной системы подбора гомеопатических препаратов

Диаграмма последовательности UML визуально моделируют поток логики в проектируемой системе и обычно используются как для анализа, так и для целей проектирования.

Диаграммы последовательности являются наиболее популярным артефактом UML для динамического моделирования, который фокусируется на определении поведения в системе. Диаграммы последовательности также, как диаграммы классов и диаграммы вариантов использования являются наиболее важными моделями уровня проектирования для разработки современных бизнес-приложений.

Диаграмма последовательности используется главным образом для отображения взаимодействий между объектами в последовательном порядке, в котором эти взаимодействия происходят. Сотрудники организации могут использовать диаграммы последовательности для того, чтобы показать, как в данный момент работает бизнес, отражая взаимодействие различных бизнес-объектов. Помимо документирования текущих задач организации, диаграмма последовательности бизнес-уровня может использоваться в качестве документа требований для будущей реализации системы.

Диаграмма последовательности является отличным способом визуализации и проверки различных сценариев выполнения вариантов использования. Они могут помочь предсказать, как система будет себя вести и обнаружить задачи, которые класс, возможно, должен иметь в процессе моделирования новой системы.

Рассмотрим основные сценарии функционирования ИС подбора гомеопатических препаратов. На рисунке 2.4 представлена диаграмма последовательности сценария подбора гомеопатических препаратов.

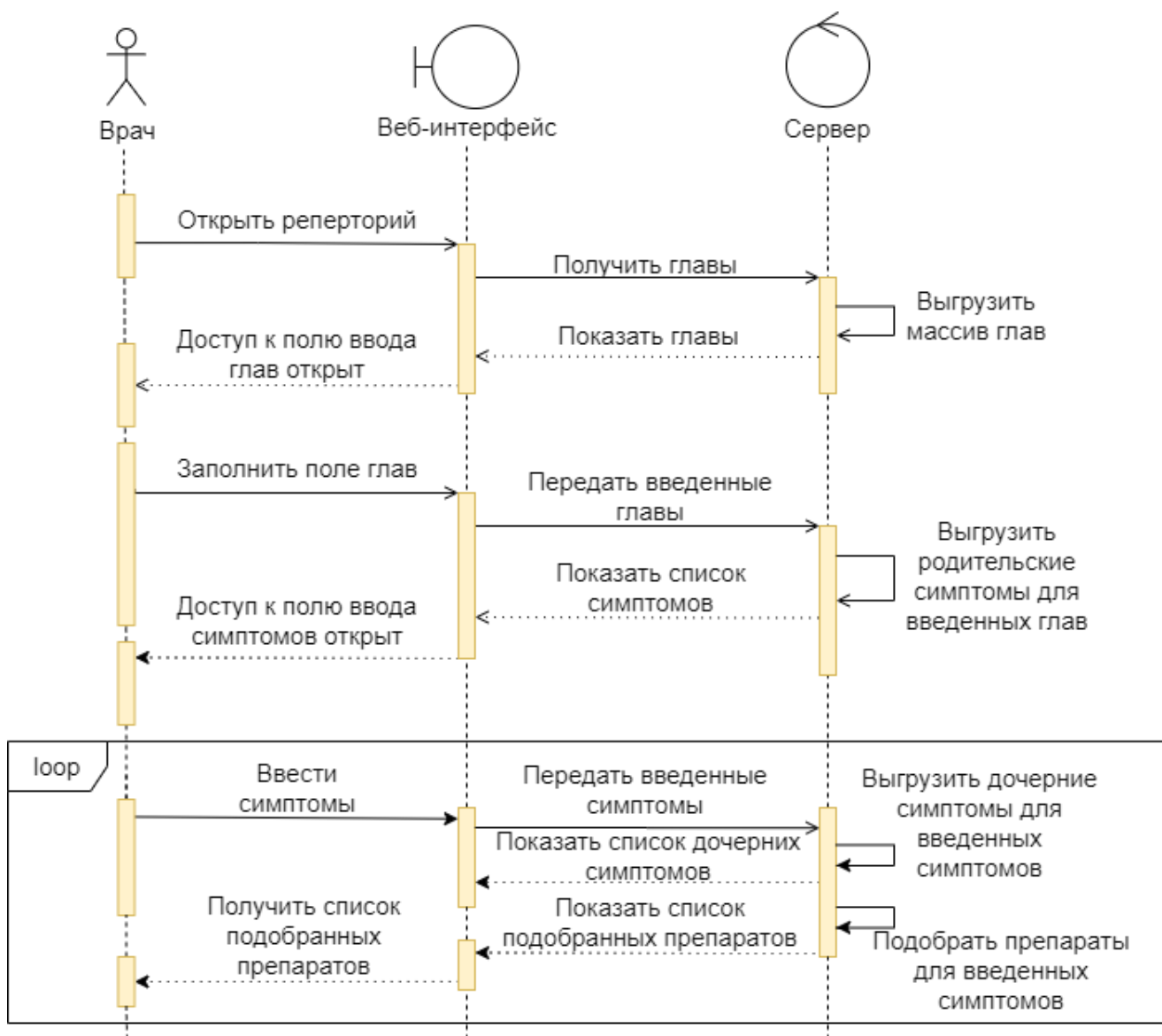


Рисунок 2.4 - Диаграмма последовательности сценария подбора гомеопатических препаратов

Врач открывает страницу репертория в веб-интерфейсе. С сервера приходит массив, который используется для функции автодополнения, глав репертория. После этого становится доступным поле для ввода глав. Когда врач начинает вводить название главы, ИС показывает список доступных глав. Врач вводит нужные главы. Введенные главы передаются на сервер, где для них выгружается массив симптомов, который так же используется для функции автодополнения. При вводе симптомов, ИС подсказывает пользователю о доступных симптомах. Врач вводит симптомы. Введенные симптомы передаются на сервер, где для них выгружается массив дочерних симптомов и

подбираются общие препараты. После каждого ввода симптома, с серверной части подгружаются его дочерние симптомы, которые отображаются в списке доступных симптомов, при заполнении поля симптомов, и подбираются общие препараты для уже введенных симптомов. Препараты подбираются только для симптомов, которые не имеют дочерних симптомов среди введенных. Врач повторяет процедуру ввода симптомов, пока не введет нужные симптомы. Если препараты не найдены, то врач вводит только ключевые симптомы. Подобранные препараты показываются в веб-интерфейсе. Процесс подбора гомеопатических препаратов завершается.

Диаграммы последовательности описывают взаимодействия между классами системы в терминах обмена сообщениями в динамике.

2.4 Проектирование веб-интерфейса

Доступ к ИС подбора гомеопатических препаратов будет осуществляться через веб-интерфейс. Для обеспечения функциональности, согласно требованиям к ИС, должны быть реализованы следующие страницы:

- 1) страница авторизации;
- 2) страница регистрации;
- 3) страница репертория с всплывающим окном для создания связки симптом-препарат;
- 4) страница препаратов с возможностью добавления новых;
- 5) страница симптомов с возможностью добавления новых.

Так как в требованиях есть требование о том, чтобы веб-интерфейс был адаптивным, в целях ускорения разработки было принято использовать Mobile-first верстку. То есть интерфейс изначально разрабатывается под мобильные устройства, а затем масштабируется для устройств с большим экраном. Ниже представлены эскизы веб-интерфейса, которые были нарисованы в сервисе DrawIO, и которые будут использоваться в дальнейшем для верстки.

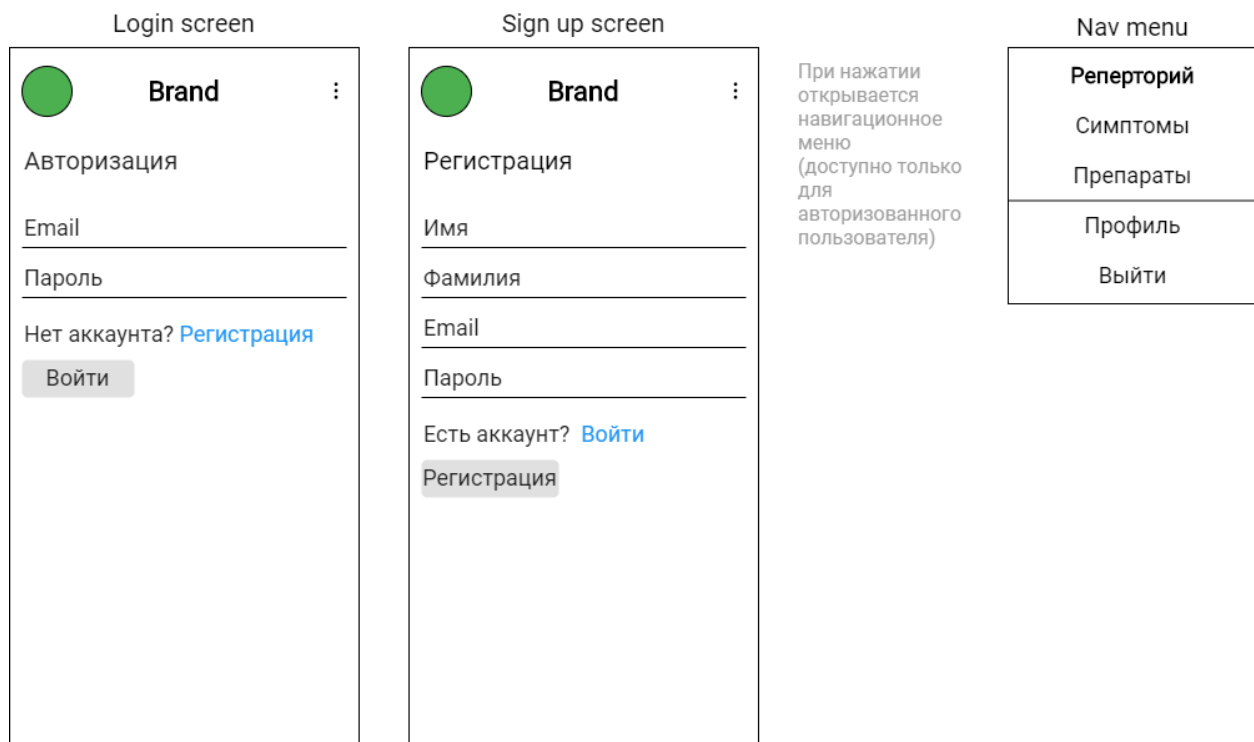


Рисунок 2.5 – Эскизы страниц авторизации и регистрации

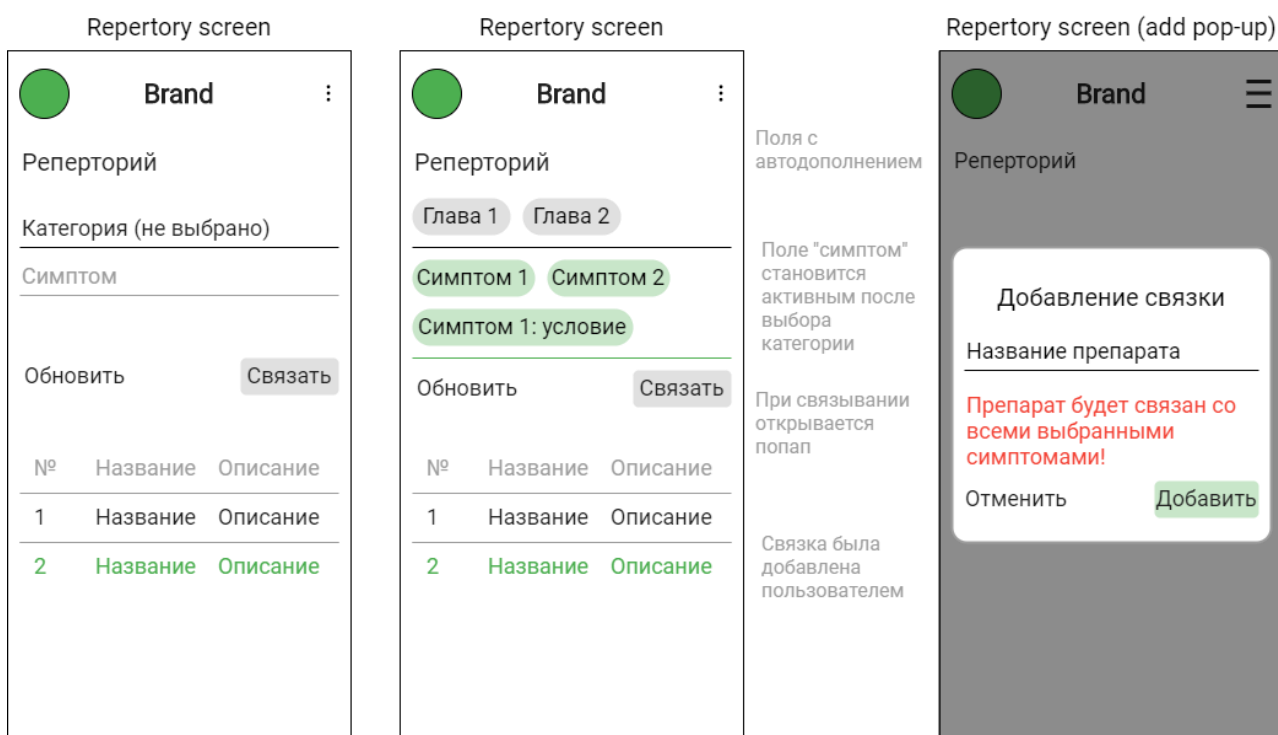


Рисунок 2.6 – Эскиз страницы репертория

В процессе проектирования веб-интерфейса были выделены следующие компоненты клиентского приложения:

- 1) компонент заголовка страницы;
- 2) компонент авторизации с формой;

- 3) компонент регистрации с формой;
- 4) компонент страницы препаратов с формой для добавления нового препарата и таблицей для отображения препаратов из базы данных ИС;
- 5) компонент навигационной панели с навигационным меню;
- 6) компонент страницы профиля пользователя;
- 7) компонент страницы репертория с формой для ввода симптомов и таблицей для вывода подобранных препаратов;
- 8) компонент страницы симптомов с полем для отображения симптомов из базы данных информационной системы и формой для добавления новых симптомов.

Перечисленные компоненты будут реализованы в клиентском приложении.

Вывод по главе 2

Во второй главе представлена логическая архитектура информационной системы подбора гомеопатических препаратов.

Построена диаграмма вариантов использования. Описаны прецеденты. Отражено краткое описание вариантов использования ИС.

Представлена построенная диаграмма классов ИС подбора гомеопатических препаратов. Описана спецификация диаграммы классов.

Построена диаграмма последовательности. Рассмотрены основные сценарии функционирования ИС.

Спроектирован веб-интерфейс информационной системы подбора гомеопатических препаратов, согласно требованиям заказчика. Выделены компоненты клиентского приложения.

Глава 3 Физическое проектирование информационной системы подбора гомеопатических препаратов

3.1 Выбор средств реализации информационной системы подбора гомеопатических препаратов

Для проектирования и разработки данной информационной системы были использованы следующие программные средства и сервисы:

- веб-сервис DrawIO;
- ERWin;
- язык программирования TypeScript (JavaScript);
- СУБД PostgreSQL 11.0.1;
- программная платформа NodeJS;
- NestJS;
- TypeORM;
- Angular 9;
- NX (NX dev);
- IntelliJ IDEA.

DrawIO - это веб-сервис, предназначенный для формирования диаграмм и схем.

ERWin – это программа для проектирования и документирования баз данных.

TypeScript - язык программирования, представленный Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript.

PostgreSQL - свободная объектно-реляционная система управления базами данных.

NodeJS - программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет

возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, которые обеспечивают вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения. В основе Node.js лежит событийно-ориентированное и асинхронное программирование с неблокирующим вводом/выводом.

NestJS - прогрессивный NodeJS фреймворк для создания эффективных, надежных и масштабируемых серверных приложений.

TypeORM - объектно-реляционный картограф для TypeScript и JavaScript, который поддерживает множество различных баз данных, таких как MySQL и PostgreSQL, и таких платформ, как NodeJS и Browser.

Angular - JavaScript фреймворк для создание клиентских приложений. Прежде всего нацелен на разработку SPA-решений (Single Page Application), то есть одностраничных приложений.

NX - расширяемые инструменты разработки для монорепозитория.

IntelliJ IDEA - интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JavaScript, Python, разработанная компанией JetBrains.

Для ускорения разработки клиентской и серверной частей был выбран один язык программирования – TypeScript (JavaScript), так как есть опыт клиентской разработки.

3.2 Разработка модели данных информационной системы подбора гомеопатических препаратов

На рисунке 3.1 изображена логическая модель базы данных ИС, построенная на основе ее диаграммы классов в CASE-средстве ERWin.

ERWin – это программа для проектирования и документирования баз данных.

Модель построена в усиленной 3-ей нормальной форме (модель Бойса-Кодда).

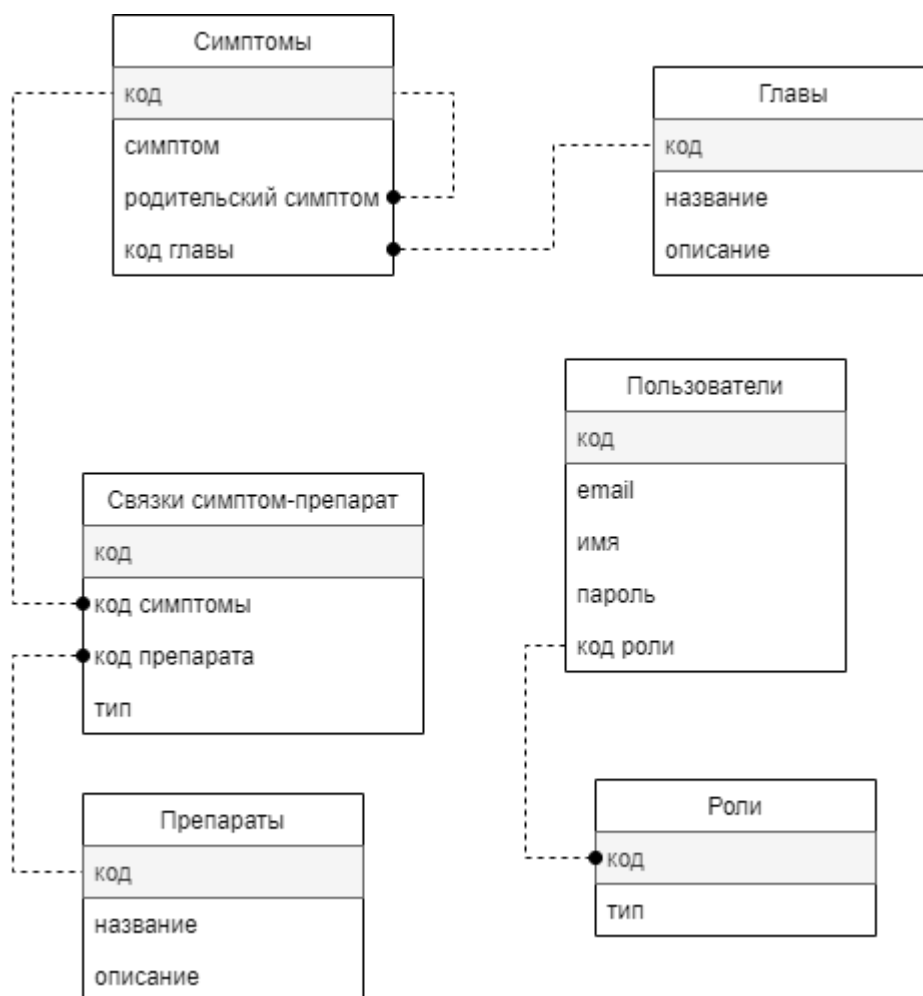


Рисунок 3.1 – Логическая модель данных ИС подбора гомеопатических препаратов

На рисунке 3.2 представлена физическая модель данных ИС, построенная для СУБД PostgreSQL также в программе ERWin.

В таблице Symptoms хранится список симптомов. У каждого симптома могут быть дочерние симптомы.

В таблице Categories хранится список глав репертория.

В таблице Medications хранится список гомеопатических препаратов.

Таблица MedicationsSymptoms связующая между таблицами симптомов и препаратов. В этой таблице хранятся связки симптом-препарат.

В таблице Users хранится список зарегистрированных пользователей информационной системы.

В таблице Roles хранится список ролей пользователей.

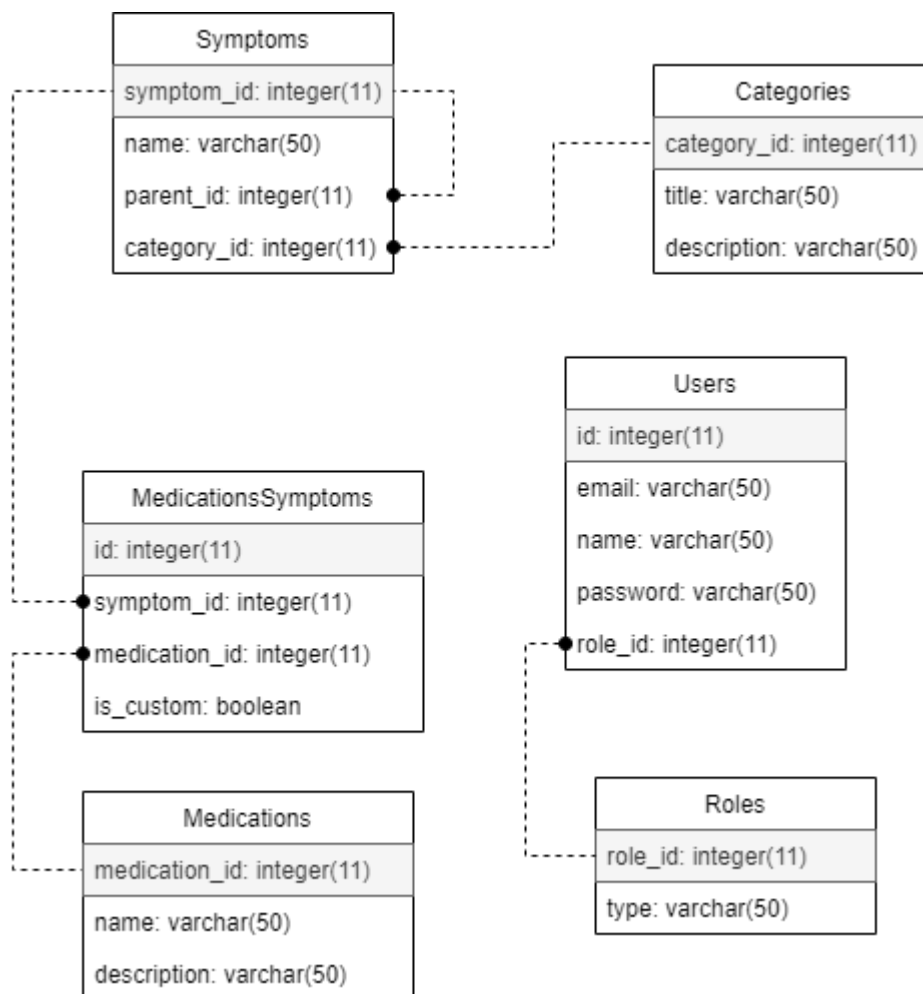


Рисунок 3.2 – Физическая модель данных ИС подбора гомеопатических препаратов

Для информационной системы была разработана база данных, которая будет в качестве СУБД использовать PostgreSQL. Данная база данных позволяет эффективно хранить данные и получать к ним доступ, обеспечивает надежность и гибкость использования.

На рисунке 3.3 изображена фреймовая структура «Связка симптом-препарат» (фрейм-прототип).

Фрейм - это структура, предназначенная для представления стереотипной информации. Каждый фрейм описывает один концептуальный объект, а конкретные свойства этого объекта и факты, относящиеся к нему, описываются в слотах – структурных элементах данного фрейма.

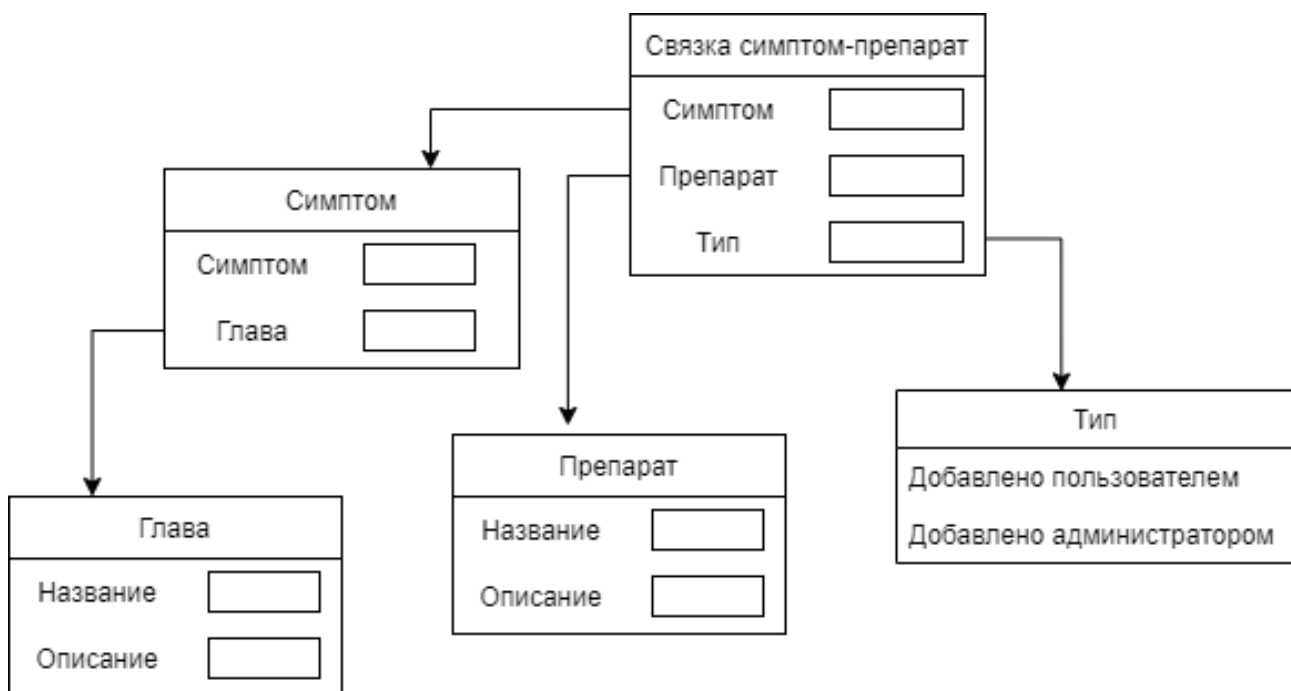


Рисунок 3.3 – Фреймовая структура «Связка симптом-препарат»

Все фреймы взаимосвязаны и образуют единую фреймовую систему, в которой объединены и процедурные знания. Концептуальному представлению свойственна иерархичность, целостный образ знаний строится в виде единой фреймовой системы, имеющей иерархическую структуру. В слот можно представить разные данные: числа или математические соотношения, тексты, программы, правила вывода или ссылки на другие слоты данного или других фреймов.

3.3 Разработка серверного приложения

Для написания серверной и клиентской части был выбран один язык программирования TypeScript, который расширяет возможности JavaScript.

С помощью NX, было создано рабочее окружение и серверное приложение Api, которое будет обрабатывать запросы с клиентской части. Серверное приложение было построено с использованием NestJS. Для работы с базой данных, использовался TypeORM, который поддерживает паттерн «Репозиторий».

TypeORM – это, как описывалось ранее, объектно-реляционный картограф для TypeScript и JavaScript, который поддерживает множество

различных баз данных, таких как MySQL и PostgreSQL, и таких платформ, как Node.js и Browser. В разрабатываемой информационной системе используется СУБД PostgreSQL.

Были реализованы следующие модули серверного приложения:

- 1) App - корневой модуль приложения;
- 2) Auth - модуль, который отвечает за авторизацию пользователей;
- 3) Db - модуль, предназначенный для работы с базой данных;
- 4) Repertory - модуль, который отвечает за работу с препаратами и симптомами;
- 5) Users - модуль, отвечающий за работу с пользователями.

Практически в каждом модуле есть контроллер, который обрабатывает запросы, относящиеся к этому модулю.

В приложении была реализована авторизация с помощью токена JWT. JSON Web Token (JWT) - это открытый стандарт для создания токенов доступа, основанный на формате JSON. Он содержит три блока, разделенных точками: заголовок (header), набор полей (payload) и сигнатура (signature).

В разрабатываемом приложении используется только access токен, так как отсутствует необходимость использования refresh токена.

Рассмотрим некоторые модули разрабатываемого серверного приложения. На рисунке 3.4 продемонстрирован фрагмент модуля App. В этом модуле конфигурируется приложение, настраивается подключение к базе данных и подключаются модули приложения. Для асинхронной передачи параметров конфигурирования TypeORM используется фабрика. Для конфигурирования TypeORM передаются следующие параметры:

- type;
- url;
- synchronize;
- autoLoadEntitie.

```
@Module({
  imports: [
    ConfigModule.forRoot({
      isGlobal: true,
      load: [mainConfig],
    }),
    TypeOrmModule.forRootAsync({
      inject: [ConfigService],
      useFactory: (configService: ConfigService) => ({
        type: "postgres",
        url: configService.get("dbUrl"),
        synchronize: true,
        autoLoadEntities: true,
      }),
    }),
    ServeStaticModule.forRoot({
      rootPath: join(__dirname, "../../../../../repertory-app"),
      exclude: ["/api/*.json"],
    }),
    AuthModule,
    UsersModule,
    RepertoryModule,
  ],
})
export class AppModule {}
```

Рисунок 3.4 - Фрагмент модуля App

В модуле Db импортируются сущности. Рассмотрим также некоторые из них. На рисунке 3.5 продемонстрирована сущность UserDAO, которая имеет следующие поля:

- уникальный идентификатор пользователя;
- имя;
- фамилия;
- электронная почта;
- пароль;
- роль;

```

@Entity()
export class UserDAO {
  @PrimaryGeneratedColumn()
  id: number;

  @Column({
    length: 20,
  })
  name: string;

  @Column({
    length: 20,
  })
  surname: string;

  @Column({
    length: 100,
  })
  email: string;

  @Column()
  password: string;

  @Column({
    type: "enum",
    enum: RoleEnum,
    default: RoleEnum.USER,
  })
  role: RoleEnum;
}

```

Рисунок 3.5 - Сущность UserDAO

В TypeORM, по умолчанию, тип string сопоставляется с типом varchar (255).

На рисунке 3.6 изображена сущность MedicationDAO, которая имеет следующие поля: medication_id, name, description. Эта сущность описывает гомеопатический препарат.

```
@Entity()
export class MedicationDAO {
  @PrimaryGeneratedColumn()
  medication_id: number;

  @Column({
    length: 25,
  })
  name: string;

  @Column({
    length: 50,
  })
  description: string;
}
```

Рисунок 3.6 - Сущность MedicationDAO

На рисунке 3.7 изображена сущность SymptomDAO, которая имеет следующие поля:

- уникальный идентификатор симптома;
- название;
- идентификатор родительского симптома;
- идентификатор главы.

Эта сущность описывает симптом. Поле `parent` ссылается на родительский симптом. Таким образом, в базе данных хранится иерархия симптомов. Поле `category` связано с сущностью `CategoryDAO` связью многие к одному, то есть в категории может быть несколько симптомов, но симптом может относиться только к одной категории. В разрабатываемой ИС главы репертория описаны как категории.

```

@Entity()
export class SymptomDAO {
  @PrimaryGeneratedColumn()
  symptom_id: number;

  @Column({
    length: 25,
  })
  name: string;

  @ManyToOne(() => SymptomDAO, {
    nullable: true,
    cascade: true,
    onDelete: "CASCADE",
  })
  @JoinColumn({ name: "parent_id" })
  parent: SymptomDAO;

  @ManyToOne(() => CategoryDAO, {
    cascade: true,
    onDelete: "CASCADE",
  })
  @JoinColumn({ name: "category_id" })
  category: CategoryDAO;
}

```

Рисунок 3.7 - Сущность SymptomDAO

Рассмотрим контроллер модуля Repertory, который отвечает за работу с препаратами и симптомами.

На рисунке 3.8 продемонстрирован фрагмент контроллера, а именно обработчик запроса GET: «api/repertory/medications» для получения списка всех препаратов. Контроллер доступен только для авторизованного пользователя, так как обернут в декоратор @UseGuards(JwtAuthGuard), то есть контроллер обработает запрос только тогда, когда в заголовке запроса присутствует валидный токен. Данный контроллер возвращает массив всех препаратов.

```
@UseGuards(JwtAuthGuard)
@Get("/medications")
getMedicationsAll(): Promise<MedicationDTO[]> {
    return this.repertoryService.getMedicationsAll();
}
```

Рисунок 3.8 - Контроллер getMedicationsAll

В NestJS используются декораторы для указания метода запроса. Также был реализован `JwtAuthGuard`, который проверяет наличие JWT токена в заголовке запроса. Если токен не валидный, то `JwtAuthGuard` возвращает клиенту статус «401 Unauthorized». В таком случае пользователь получает сообщение о том, что он не авторизован в системе.

При регистрации, пароль пользователя сохраняется в базе данных в виде хэша и соли. После успешной аутентификации, генерируется токен, который отправляется на клиентскую часть.

В разрабатываемой ИС валидация данных происходит как на стороне сервера, так и на стороне клиента. В серверном приложении это реализовано путем добавления правил проверки для полей в классах DTO, используя декораторы, представленные в пакете `class-validator`. При несоблюдении добавленных правил, если запрос достигает конечной точки с недействительным свойством в теле запроса, приложение автоматически ответит со статусом «400 Bad Request» и сообщением, содержащее массив ошибок.

Как видно на рисунке 3.8, в контроллере используется сервис `RepertoryService`, который в свою очередь работает с репозиториями. Паттерн «Репозиторий» посредничает между уровнями области определения и распределения данных (`domain and data mapping layers`), используя интерфейс, схожий с коллекциями для доступа к объектам области определения.

В контроллере `getMedicationsBySymptom`, который возвращает список подобранных препаратов для симптомов, общие препараты выбираются только среди тех симптомов, у которых нет дочерних симптомов среди выбранных.

Процесс подбора общих препаратов осуществляется следующим образом:

- 1) Среди введенных симптомов выбираются только те, которые не имеют дочерних симптомов;
- 2) для этих симптомов осуществляется поиск связок симптом-препарат;
- 3) для каждого из этих симптомов создается массив связанных с ним препаратов, путем перебора связок симптом-препарат;
- 4) среди созданных массивов препаратов, осуществляется поиск пересечений, для определения общих препаратов среди этих массивов.

Была реализована функция, которая выбирает общие препараты из нескольких массивов препаратов, путем поиска пересечений.

Подобранные препараты отправляются на клиентскую часть.

3.4 Разработка клиентского приложения

С помощью NX, было создано клиентское приложение `Repertory-app`, с использованием `Angular`. `Angular` предоставляет такую функциональность, как двустороннее связывание, позволяющее динамически изменять данные в одном месте интерфейса при изменении данных модели в другом, шаблоны, маршрутизация и так далее.

Компоненты и сервисы были сгенерированы, с помощью `Angular CLI`. Для описания стилей использовался `SCSS`.

В приложении были реализованы следующие компоненты:

- компонент заголовка страницы, который отображается на каждой странице и позволяет сохранять единый стиль приложения;
- компонент авторизации с формой;
- компонент регистрации с формой;

- компонент страницы препаратов с формой для добавления нового препарата и таблицей для отображения препаратов из базы данных ИС;
- компонент навигационной панели;
- компонент страницы профиля пользователя;
- компонент страницы репертория с формой для ввода симптомов и таблицей для вывода подобранных препаратов;
- компонент страницы симптомов с полем для отображения симптомов из базы данных ИС и формой для добавления новых симптомов.

В приложении была реализована авторизация. После успешной аутентификации, от серверной части приходит ответ, содержащий токен. `AuthInterceptor`, который имплементирует `Angular HTTP Interceptor`, записывает этот токен в заголовок каждого запроса клиентского приложения. Для обработки ошибок используется `ErrorInterceptor`, который в случае получения статуса 401 от серверного приложения, показывает пользователю ошибку, с сообщением о том, что он не авторизован, и перенаправляет его на страницу авторизации.

В корневом модуле клиентского приложения подключаются компоненты и модули, например, роутер, который добавляет маршрутизацию, то есть позволяет переходить между страницами.

Рассмотрим некоторые компоненты приложения. На рисунке 3.9 продемонстрирован шаблон компонента страницы авторизации.

```
<div>
  <brvns-reperty-heading title="Авторизация"></brvns-reperty-heading>
  <form (ngSubmit)="onSubmit()" [formGroup]="form" class="form">
    <mat-form-field class="form__input" hideRequiredMarker>
      <mat-label>Email</mat-label>
      <input matInput formControlName="email" type="email" required>
      <mat-error *ngIf="email.invalid">{{getEmailErrorMessage()}}</mat-error>
    </mat-form-field>
    <mat-form-field class="form__input" hideRequiredMarker>
      <mat-label>Пароль</mat-label>
      <input matInput formControlName="password" type="password" required>
      <mat-error *ngIf="password.invalid">{{getPasswordErrorMessage()}}</mat-error>
    </mat-form-field>
    <div class="form__progress">
      <mat-progress-bar *ngIf="loading" mode="indeterminate"></mat-progress-bar>
    </div>
    <span class="form__text">
      Нет аккаунта?
      <a class="form__text_link" routerLink="/register">Зарегистрироваться</a>
    </span>
    <button [disabled]="form.invalid || loading" class="form__btn" mat-button type="submit">Войти</button>
  </form>
</div>
```

Рисунок 3.9 – Шаблон компонента страницы авторизации

Шаблон компонента выглядит как обычный HTML, за исключением того, что он также содержит синтаксис шаблона Angular, который изменяет HTML на основе логики приложения и его состояния. Шаблон использует свойства и методы класса компонента. На рисунке 3.10 изображен фрагмент класса компонента страницы авторизации.

```

@Component({
  selector: "brvns-repository-login-form",
  templateUrl: "./login-form.component.html",
  styleUrls: ["./login-form.component.scss"],
})
export class LoginFormComponent {
  constructor(
    private FormBuilder: FormBuilder,
    private Router: Router,
    private AuthService: AuthService,
    private NotificationService: NotificationService
  ) {
    if (this.authService.currentUserValue) {
      this.router.navigate(["/"]);
    }
  }

  form: FormGroup = this.formBuilder.group({
    email: [
      "",
      Validators.compose([Validators.required, Validators.email]),
    ],
    password: [
      "",
      Validators.compose([
        Validators.required,
        Validators.minLength(5),
        Validators.maxLength(20),
      ]),
    ],
  });
  ...
}

```

Рисунок 3.10 – Фрагмент класса компонента страницы авторизации

Декоратор `@Component` позволяет идентифицировать класс, как компонент. Декоратор, в качестве параметра, принимает объект с конфигурацией, которая указывает фреймворку, как работать с компонентом и его представлением. В конструкторе класса инициализируются используемые модули и сервисы. На рисунке 3.11 продемонстрирован метод `onSubmit`, который вызывается после того, как пользователь нажал кнопку «войти».

```

onSubmit(): void {
  if (this.form.valid) {
    this.loading = true;
    const body = this.form.getRawValue();
    this.authService.login(body).subscribe(
      () => {
        this.notificationService.notification$.next(
          "Авторизация прошла успешно"
        );
        this.loading = false;
        this.router.navigate(["/"]);
      },
      (err) => {
        const msg = codes[err] || "Что-то пошло не так";
        this.notificationService.notification$.next(msg);
        this.loading = false;
      }
    );
  }
}

```

Рисунок 3.11 – Метод onSubmit класса компонента страницы авторизации

Если форма успешно прошла валидацию, то отправляется запрос с телом формы на серверную часть. При успешной аутентификации, пользователю показывается сообщение об успешной авторизации и роутер перенаправляет пользователя на главную страницу. В случае ошибки, пользователю показывается сообщение об ошибке.

Компонент страницы репертория реализован следующим образом: при монтировании компонента, отправляется запрос на сервер, чтобы получить главы репертория. Поле для ввода становится доступно, когда главы получены. Также поля для ввода глав и симптомов имеют функцию автодополнения. При вводе глав или симптомов, ИС подсказывает, какие главы или симптомы доступны. После того, как пользователь выбирает главы репертория, отправляется запрос, содержащий массив выбранных глав, на сервер, чтобы получить список симптомов верхнего уровня для этих глав. После ввода каждого симптома, отправляется запрос, содержащий список введенных симптомов, на сервер, чтобы получить его дочерние симптомы и подобрать

гомеопатические препараты для введенных симптомов. Чтобы обновить список подобранных препаратов, на сервер отправляется список выбранных симптомов, для которых на сервере происходит подбор общих препаратов. Как описывалось ранее, препараты подбираются только для тех симптомов, у которых нет дочерних симптомов среди выбранных.

В клиентском приложении применялась Mobile-first верстка, то есть в первую очередь идет верстка для мобильных устройств, затем она масштабируется для устройств с большим экраном. На рисунках 3.12 и 3.13 продемонстрировано, как выглядит главная страница репертория на мобильном устройстве и на устройстве с большим экраном.

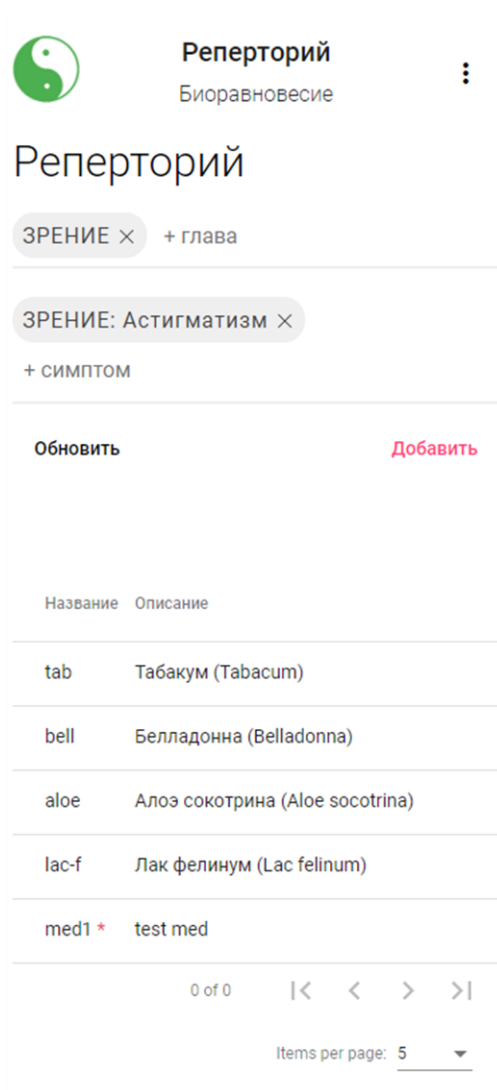


Рисунок 3.12 – Главная страница репертория на мобильном устройстве



Реперторий

ЗРЕНИЕ × + глава

ЗРЕНИЕ: Астигматизм × + симптом

Обновить

Добавить

Название	Описание
tab	Табакум (Tabacum)
bell	Белладонна (Belladonna)
aloe	Алоэ сокотрина (Aloe socotrina)

Рисунок 3.13 – Главная страница репертория на устройстве с большим экраном

Как видно из этих рисунков, ИС имеет адаптивный веб-интерфейс, который отображается на мобильных устройствах и на устройствах с большим экраном без дефектов.

3.5 Описание функциональности информационной системы

В соответствии со сформированными требованиями была реализована информационная система подбора гомеопатических препаратов.

ИС предоставляет следующие функции:

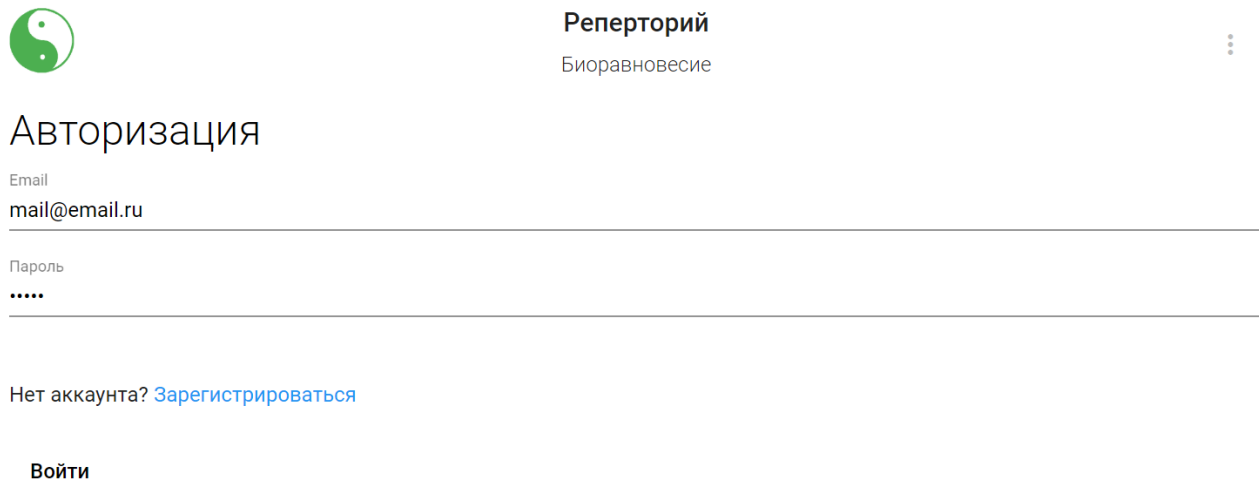
- 1) авторизация – авторизация в ИС;
- 2) регистрация – регистрация в ИС;
- 3) список препаратов – список препаратов из собственной базы препаратов;
- 4) поиск по списку препаратов;
- 5) добавление нового препарата;
- 6) удаление существующего препарата;
- 7) просмотр списка симптомов с вложенными симптомами;

- 8) добавление новых симптомов в базу данных репертория;
- 9) подбор гомеопатических препаратов по симптомам;
- 10) добавление новых связок симптом-препарат.

Веб-интерфейс ИС состоит из навигационной панели, которая содержит меню для навигации по страницам ИС, и контента страницы.

Клиентское приложение ИС представляет собой одностраничное веб-приложение, то есть оно использует всего один HTML-документ, который организует взаимодействие с пользователем через динамически подгружаемые HTML, JavaScript и CSS. Это обеспечивает более высокую производительность, в сравнении с традиционными веб-приложениями.

Работа с ИС начинается со страницы авторизации, на которой присутствуют поля для ввода электронной почты и пароля пользователя. Если у пользователя нет аккаунта, то он может зарегистрировать новый аккаунт на странице регистрации. На рисунке 3.14 продемонстрирована страница авторизации.



Реперторий
Биоравновесие

Авторизация

Email
mail@email.ru

Пароль
.....

Нет аккаунта? [Зарегистрироваться](#)

Войти

Рисунок 3.14 – Страница авторизации

На странице авторизации присутствуют следующие поля:

- электронная почта;
- пароль.

Эти поля имеют валидацию. Кнопка «Войти» доступна только тогда, когда все поля успешно прошли валидацию.

Если у пользователя нет аккаунта, то он может зарегистрировать новый. Для этого нужно перейти на страницу регистрации, которая представлена на рисунке 3.15.

Реперторий
Биоравновесие

Регистрация

Имя
Александр

Фамилия
Погодин

Email
awpogodin@mail.ru

Пароль
.....

Повторите пароль
.....

Есть аккаунт? [Войти](#)

[Зарегистрироваться](#)

Рисунок 3.15 – Страница регистрации

На странице регистрации присутствуют следующие поля:

- Имя;
- фамилия;
- электронная почта;
- пароль;
- подтверждение пароля.

Эти поля так же имеют валидацию. Кнопка «Зарегистрироваться» доступна только тогда, когда все поля успешно прошли валидацию.

После успешной авторизации, пользователю показывается уведомление об успешной авторизации, и он попадает на главную страницу репертория, которая продемонстрирована на рисунке 3.16.



Реперторий

Введите главу

Обновить

Добавить

Название

Описание

Items per page: 5

0 of 0

⏪ ⏩ ⏴ ⏵

* - Связка добавлена пользователем

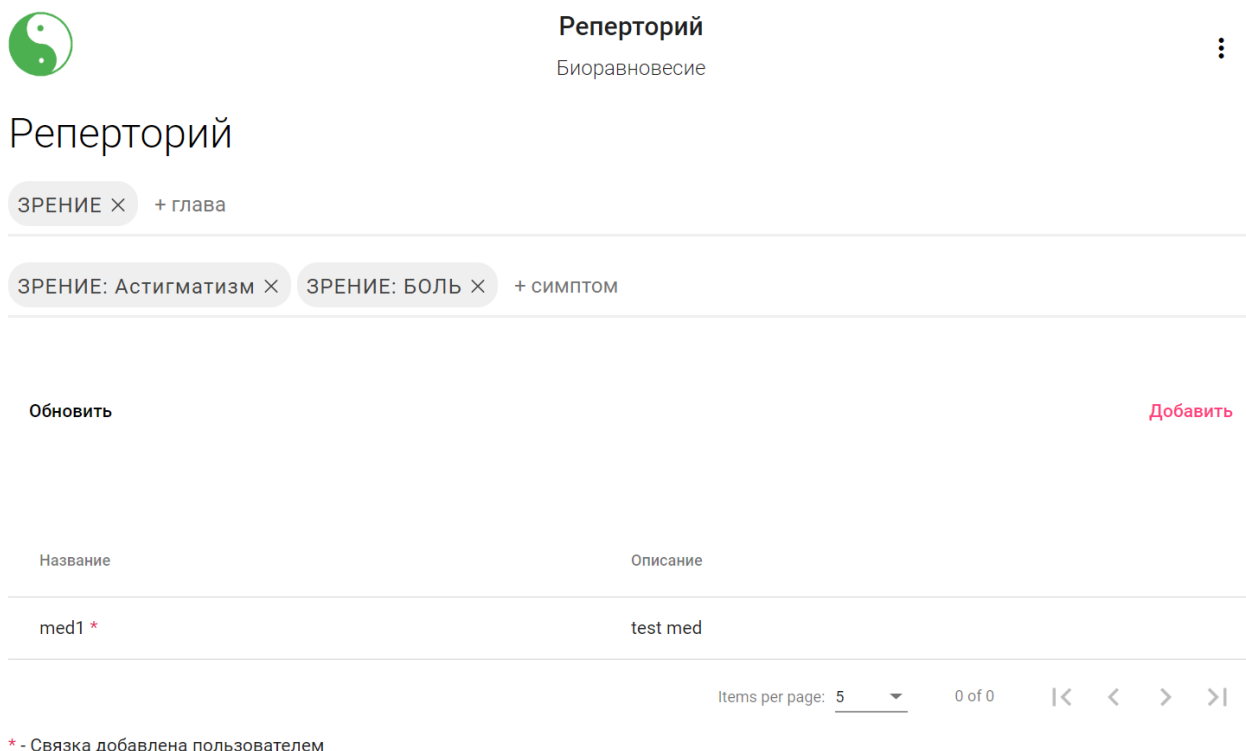
Авторизация прошла успешно

Закреть

Рисунок 3.16 – Главная страница репертория

На главной странице репертория пользователь может осуществить подбор гомеопатических препаратов по симптомам. Для этого он вводит главы и симптомы. Поля для ввода глав и симптомов имеют функцию автодополнения. Как только пользователь начинает вводить главу или симптом, ИС подсказывает пользователю, путем отображения выпадающего списка существующих глав или симптомов, в зависимости от того, какое поле заполняет пользователь. Каждый раз, когда пользователь вводит симптом, с серверной части подгружается список его дочерних симптомов, который добавляется в список доступных симптомов для функции автодополнения. Также после ввода симптома, осуществляется подбор препаратов для введенных симптомов. При необходимости, пользователь может вручную инициализировать процесс подбора препаратов, путем обновления списка подобранных препаратов. Список доступных глав подгружается с серверной части ИС, при загрузке страницы. Во время загрузки данных с серверной части, поля для ввода недоступны.

На рисунке 3.17 показан пример подбора тестовых препаратов по СИМПТОМАМ.



Реперторий
Биоравновесие

Реперторий

ЗРЕНИЕ × + глава

ЗРЕНИЕ: Астигматизм × ЗРЕНИЕ: БОЛЬ × + симптом

Обновить Добавить

Название	Описание
med1 *	test med

Items per page: 5 0 of 0 |< < > >|

* - Связка добавлена пользователем

Рисунок 3.17 – Пример подбора тестовых препаратов по симптомам

Для добавления связки, пользователю нужно нажать на кнопку «добавить» и ввести название препарата, с которым пользователь хочет связать симптом. На рисунке 3.18 продемонстрировано всплывающее окно, для добавления связки.

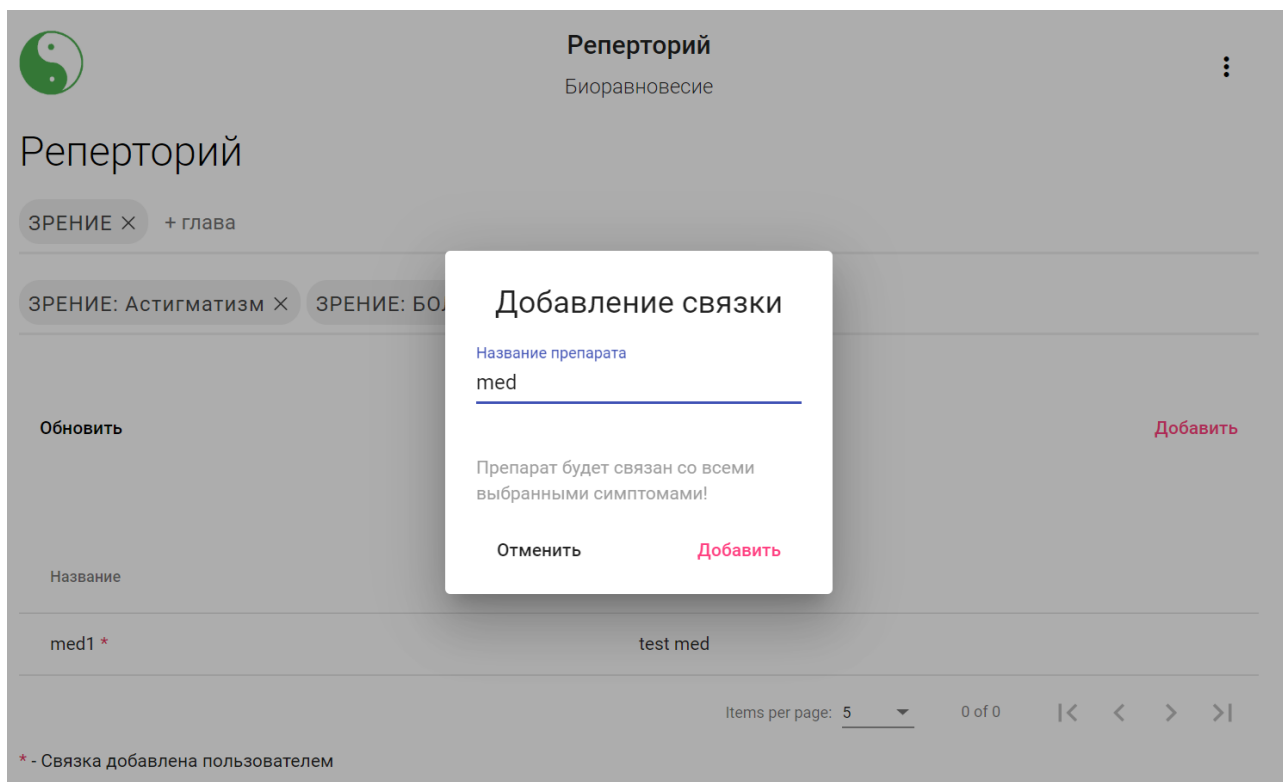


Рисунок 3.18 – Всплывающее окно для добавления связки

Также поле для ввода названия препарата имеет функцию автодополнения, то есть, при вводе названия препарата, ИС подсказывает пользователю, какие препараты существуют в базе данных, путем отображения выпадающего списка препаратов.

Список доступных препаратов подгружается с серверной части, при открытии окна добавления связки.

На рисунке 3.19 продемонстрирована страница препаратов. На странице присутствует форма для добавления нового препарата и таблица со списком препаратов. Также на этой странице пользователь может удалить препарат из базы данных.



Препараты

Название

med

Описание

Тестовый препарат

[Добавить](#)

Поиск

Название	Описание	
анас	Анакардиум ориентале (Anacardium orientale)	⊖
abr	Абрус прекаториус (Abrus precatorius)	⊖
мис-р	Мукоза пульмонис суис (Mucosa pulmonis suis)	⊖

Рисунок 3.19 – Страница препаратов

На рисунке 3.20 продемонстрирована страница для просмотра и добавления симптомов. Пользователь выбирает главу репертория и ИС показывает список симптомов этой главы. Если пользователь выбирает какой-нибудь симптом, то подгружается список его дочерних симптомов. Также внизу страницы присутствует поле для ввода, которое позволяет добавить **новый симптом**.



СИМПТОМЫ

Глава

МЕНТАЛЬНОСТЬ

← активность, деятельность

умственная, психическая

беспокойная, неугомонная

улучшает

Симптом

Добавить

Рисунок 3.20 – Страница симптомов

Таким образом, был спроектирован и разработан современный и интуитивно понятный веб-интерфейс, который полностью соответствует требованиям заказчика.

3.6 Тестирование информационной системы

Тестирование, как заключительный этап разработки информационной системы, играет жизненно важную роль в процессе создания высококачественного программного обеспечения. Непосредственно тестирование проводится в несколько этапов. Тестирование будет производиться в ручном режиме. Также в данном тестировании будут участвовать три браузера: Google Chrome, Mozilla Firefox, Yandex Browser. Рассмотрим основные этапы тестирования:

1) дымовое тестирование, которое предполагает минимальный набор тестов на явные ошибки. На этом этапе осуществляются самые основные, базовые проверки;

2) функциональное тестирование. На этом этапе идет тестирование основных сценариев использования ИС и предназначен для оценки

соответствия разработанного программного обеспечения исходным функциональным требованиям заказчика. То есть проведение функционального тестирования позволяет проверить способность информационной системы в определенных условиях решать задачи, которые требуются пользователям;

3) usability тестирование. Оно предназначено для оценки информационной системы с точки зрения конечного пользователя. Это помогает определить соответствие продукта ожиданиям пользователей и выявляет проблемные места в интерфейсе;

4) тестирование веб-интерфейса (UI). Данный тест выполняется для проверки соответствия графического пользовательского интерфейса описанным ранее спецификациям.

Тестирование проводилось путем заполнения базы данных небольшим массивом тестовых данных. В базу данных ИС также было добавлено более 1500 препаратов из существующей базы данных. Применялись следующие виды тестирования: модульное тестирование и системное тестирование.

Компонентное (модульное) тестирование проверяет функциональность и ищет дефекты в частях приложения, которые доступны и могут быть протестированы по-отдельности.

Основной задачей системного тестирования является проверка как функциональных, так и не функциональных требований в системе в целом.

Были разработаны тест-кейсы. Некоторые из них представлены в таблице 3.1.

Таблица 3.1 – Тест-кейсы

№	Название	Инструкции	Ожидаемый результат
1.1	Регистрация	1) Открыть приложение; 2) выйти из аккаунта, если пользователь уже авторизован; 3) перейти на страницу регистрации; 4) ввести данные для регистрации; 5) зарегистрироваться.	Появилось сообщение об успешной регистрации

Продолжение таблицы 3.1

1.2	Авторизация	<ol style="list-style-type: none"> 1) Открыть приложение; 2) выйти из аккаунта, если пользователь уже авторизован; 3) перейти на страницу авторизации; 4) ввести данные от аккаунта; 5) войти. 	<p>Появилось сообщение об успешной авторизации.</p> <p>Пользователь переадресован на главную страницу</p>
1.3	Подбор тестового препарата	<ol style="list-style-type: none"> 1) Открыть приложение; 2) авторизоваться; 3) перейти на страницу репертория; 4) выбрать главу ЗРЕНИЕ; 5) выбрать в поле симптомов следующие симптомы: БОЛЬ, БОЛЬ: справа, БОЛЬ: слева. 	<p>В списке подобранных препаратов должен быть тестовый препарат med1</p>
1.4	Адаптивный веб интерфейс	<ol style="list-style-type: none"> 1) Открыть приложение; 2) посмотреть, как выглядит приложение на разных размерах экрана. Для этого можно использовать инструменты для разработчиков в браузере Google Chrome. 	<p>Интерфейс адаптируется под размеры экрана</p>
1.5	Возможность добавления новых препаратов	<ol style="list-style-type: none"> 1) Открыть приложение; 2) авторизоваться; 3) зайти на страницу препаратов; 4) заполнить поля и отправить форму. 	<p>Новый препарат был добавлен в список всех препаратов.</p>
1.6	Возможность добавления новых симптомов	<ol style="list-style-type: none"> 1) Открыть приложение; 2) авторизоваться; 3) зайти на страницу симптомов; 4) добавить новый симптом. 	<p>Новый симптом был добавлен в список симптомов.</p>

В процессе прохождения тест-кейсов, ошибки не были найдены. Результаты тестирования информационной системы подбора гомеопатических препаратов представлены в таблице 3.2.

Таблица 3.2 – Результаты тестирования ИС

	Google Chrome	Mozilla Firefox	Yandex Browser
дымовое тестирование	+	+	+
функциональное тестирование	+	+	+

Продолжение таблицы 3.2

usability тестирование	+	+	+
тестирование веб- интерфейса (UI)	+	+	+

Результаты тестирования показали, что разработанная информационная система подбора гомеопатических препаратов полностью соответствует основным требованиям заказчика.

Вывод по главе 3

В третьей главе осуществлялась разработка информационной системы подбора гомеопатических препаратов для ООО ГЦ «Биоравновесие». За основу были взяты спроектированные во второй главе диаграммы.

Были разработаны серверное приложение, с использованием фреймворка NestJS, и клиентское приложение, с использованием фреймворка Angular.

Были описаны функциональные возможности разработанной информационной системы подбора гомеопатических препаратов.

Также было произведено тестирование информационной системы. Разработаны тест-кейсы для тестирования ИС. Результаты тестирования показали, что разработанная информационная система подбора гомеопатических препаратов полностью соответствует требованиям заказчика.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы был проанализирован бизнес-процесс подбора гомеопатических препаратов, были выявлены недостатки существующего процесса. Был произведен анализ аналогов ИС подбора гомеопатических препаратов. По результатам анализа было принято решение о разработке информационной системы подбора гомеопатических препаратов.

В первой главе было произведено функциональное моделирование информационной системы подбора гомеопатических препаратов. Была описана характеристика организации, разработана и проанализирована модель бизнес-процесса подбора гомеопатических препаратов «КАК ЕСТЬ», на основе выявленных недостатков которой была разработана модель бизнес-процесса «КАК ДОЛЖНО БЫТЬ». Совместно с руководством организации были выработаны требования к разрабатываемой ИС. На основе этих требований были проанализированы аналоги ИС подбора гомеопатических препаратов. Результат анализа показал, что ни одно из описанных решений не удовлетворяет в полной мере требованиям. Вследствие чего было принято решение о разработке информационной системы подбора гомеопатических препаратов, которая позволила бы устранить выявленные недостатки существующего бизнес-процесса.

Во второй главе было произведено логическое моделирование информационной системы подбора гомеопатических препаратов. Была построена логическая архитектура ИС. Были разработаны диаграмма вариантов использования, диаграмма классов ИС и диаграмма последовательности. Была описана спецификация диаграммы классов. Рассмотрены основные сценарии функционирования ИС. Также был спроектирован веб-интерфейс информационной системы.

В третьей главе было произведено физическое моделирование информационной системы подбора гомеопатических препаратов. Были выбраны средства реализации ИС. Разработаны серверное и клиентское

приложения. Были описаны функциональные возможности разработанной информационной системы подбора гомеопатических препаратов. Также было произведено тестирование разработанной ИС. Тестирование показало, что разработанная информационная система подбора гомеопатических препаратов полностью соответствует требованиям заказчика.

Были выполнены следующие задачи:

- проанализирована предметная область подбора гомеопатических препаратов и организационная структура организации ООО ГЦ «Биоравновесие»;
- выделены требования разрабатываемой информационной системы;
- выбраны средства проектирования и разработки;
- спроектирована информационная система;
- разработана информационная система.

Все это позволило сделать выводы, что цели и задачи данной работы успешно достигнуты, что также подтверждает практическую значимость проекта.

В дальнейшем планируется развитие проекта, а именно добавление в базу данных новых препаратов и симптомов, добавление новых функциональных возможностей, которые еще больше позволят сделать из информационной системы гибкий и удобный инструмент для врача.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Леоненков, А.В. Самоучитель UML 2-е издание / А.В. Леоненков – СПб.: Питер, 2004г.
2. Карпова, И.П. Базы данных: курс лекций и материалы для практ. занятий: учеб. пособие для студентов техн. фак. / И. П. Карпова. – СПб.: Питер, 2013.
3. Трутнев Д. Р. Архитектуры информационных систем. Основы проектирования: Учебное пособие. / Д. Р. Трутнев – СПб.: НИУ ИТМО 2012г.
4. Файн Я., Моисеев А. Angular и TypeScript. Сайтостроение для профессионалов. – СПб.: Питер, 2018.
5. Хавербеке Марейн Выразительный JavaScript. Современное веб-программирование. 3-е изд. - СПб.: Питер, 2019.
6. Резиг, Бибо, Марас Секреты JavaScript ниндзя. 2-е изд.: Пер. с англ. - СПб.: ООО «Альфа-книга», 2017.
7. Новиков Д.А. Теория управления организационными системами / Д.А. Новиков. - М.: МПСИ, 2005. – 584 с.
8. Чистов, Д.В. Проектирование информационных систем. Учебник и практикум / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. – М.: Юрайт, 2016. – 260 с.
9. Заботина, Н.Н. Проектирование информационных систем – М.: ДРОФА, 2013. – 336 с.
10. Емельянова, Н.З. Проектирование информационных систем: учебное пособие / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. – М.: Форум, 2014. – 432 с.
11. Абрамова, Л.В. Инструментальные средства информационных систем: учебное пособие / Л. В. Абрамова ; Министерство образования и науки Российской Федерации, Северный (Арктический) федеральный университет

имени М.В. Ломоносова. -Архангельск : САФУ, 2013. – 118 с. : ил. – Библиогр. в кн. – ISBN 978-5-261-00851-4

12. Грекул В. И. Проектирование информационных систем : учеб. пособие / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. – 2-е изд., испр. – Москва : Интернет-Ун-т Информ. Технологий : БИНОМ. Лаб. знаний, 2008. – 299 с.

13. Золотов С. Ю. Проектирование информационных систем : учеб. пособие / С. Ю. Золотов ; Томский гос. ун-т систем управления и радиоэлектроники. - Томск : Эль Контент, 2015. - 86 с

14. Коннолли, Т. В. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли. - М.: Вильямс И.Д., 2017. - 1440 с.

15. Стружкин Н. П. Базы данных: проектирование : учебник для академического бакалавриата / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2016. — 476 с.

16. Фуфаев Э. В. Базы данных. Учебное пособие / Э.В. Фуфаев, Д.Э. Фуфаев. - М.: Академия, 2014. - 320 с.

Электронные ресурсы

17. Документация NestJS [Электронный ресурс]. - Режим доступа: <https://docs.nestjs.com/> (дата обращения 01.03.2020 г.)

18. Документация Angular [Электронный ресурс]. - Режим доступа: <https://angular.io/docs/> (дата обращения 01.03.2020 г.)

19. JavaScript MDN [Электронный ресурс]. - Режим доступа: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата обращения 01.03.2020 г.)

20. RxJS API List [Электронный ресурс]. - Режим доступа: <https://rxjs-dev.firebaseapp.com/api> (дата обращения 01.03.2020 г.)

21. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс] / Б. Мейер. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019.

71 — 285 с. — Режим доступа: <http://www.iprbookshop.ru/79706.html> (дата обращения 01.03.2020 г.).

22. Киселева Т. В. Программная инженерия. Часть 1 [Электронный ресурс] : учебное пособие / Т. В. Киселева. — Ставрополь : Северо-Кавказский федеральный университет, 2017. — 137 с. — Режим доступа: <http://www.iprbookshop.ru/69425.html> (дата обращения 01.03.2020 г.).

Литература на иностранном языке

23. Howard Gould Systems Analysis and Design: 1st edition / Howard Gould. – 2016

24. Williams L. Agile Software Development Methodologies and Practices. *Advances in Computers*, vol. 80, 2010, pp. 1-44.

25. Georgescu M., Jeflea V. The Particularity of the Banking Information System. *Procedia Economics and Finance*, vol. 20, 2015, pp. 268-276.

26. Tangkawarow I., Waworuntu J. A Comparative of business process modelling techniques. *IOP Conference Series: Materials Science and Engineering*, 2016.

27. Shen H., Wall B., Zaremba M., Chen Y., Brownie J.. Integration of business modelling methods for enterprise information system analysis and user requirement gathering. *Computers in Industry*, vol. 54, 2004, pp. 307-323.

28. Wang J.X. and Wang D. “Application of mathematical modeling in management accounting”, *Italian Journal of pure and applied mathematics*, 2017, n. 3.

29. Mkrtychev S.V., Ochepovsky A.V. and Enik O.A. “Configuration of management accounting information system for multi-stage manufacturing”, *J. Phys.: Conf. Ser.* 2018, 1015, 042039.

30. Joshua Marinacci, Chris Adamson, Swing Hacks. *Tips and Tools for Killer GUIs*, 2

ПРИЛОЖЕНИЕ А

Программный код

Исходный код разработанной ИС доступен на ресурсе:
<https://github.com/awpogodin/brvns-reperatory>