

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет Инфокоммуникационных технологий

Образовательная программа

О Т Ч Е Т

о практике Учебная, ознакомительная

Тема задания: Проектирование базы данных веб-приложения для составления графика лечения на основании рекомендаций врача.

Обучающийся: Никифоров Савелий Денисович K3220

Согласовано:

Руководитель практики от университета: Казанова Полина Петровна, тьютор,
Факультет инфокоммуникационных технологий

Практика пройдена с оценкой

Дата _____

Санкт-Петербург
2023

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ	3
1 Сравнение аналогов проектируемой системы	5
2 Анализ функциональных и нефункциональных требований ..	8
2.1 Функциональные требования	8
2.2 Нефункциональные требования	9
3 Диаграммы прецедентов	11
3.1 Диаграмма прецедентов для сущности "Пользователь"	11
3.2 Диаграмма прецедентов для сущности "Доктор"	12
3.3 Диаграмма прецедентов для сущности "Менеджер"	14
4 Проектирование базы данных приложения	16
4.1 Потенциальные объекты системы	16
4.2 Определение атрибутов и первичных ключей	16
4.3 Логическое проектирование базы данных	22
5 Алгоритмы работы с пользователем	24
5.1 Диаграммы активности.....	24
5.2 Контекстные диаграммы взаимодействия	27
6 Технологии программирования	31
6.1 Веб-приложение	31
6.2 Система уведомлений пользователей	31
6.3 Развертывание приложения.....	32
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	33

ВВЕДЕНИЕ

В современном мире все больше людей обращаются к веб-сервисам для автоматизации и упрощения собственной жизни. Они органично вписались во все направления жизни людей современного общества.

Одним из таких инструментов упрощения и автоматизации жизнедеятельности может стать веб-приложение для составления графика лечения на основании рекомендаций врача. Помимо удобства для пациентов, данное приложение может помочь докторам для более эффективного контроля процесса лечения.

Актуальность нашего приложения обоснована следующим: по данным исследования Ассоциации директоров по коммуникациям и корпоративным медиа России (АКМР) [1] от 2023 года показывает что 78 процентов Россиян пользуются интернетом на регулярной основе, что показывает большой охват аудитории для полноценной реализации и интеграции будущего веб-приложения.

В дополнение к этому, с учетом увеличивающейся численности старшего поколения и необходимости улучшения доступа к медицинским услугам, веб-приложение подобного рода может иметь значительный потенциал для улучшения уровня здравоохранения. Это также может помочь уменьшить нагрузку на медицинский персонал, освободив их время для более критически важных задач. Кроме того, с учетом текущей ситуации в мире, веб-приложения для здравоохранения открывают новые возможности для дистанционного взаимодействия и контроля за процессом лечения, что делает их еще более актуальными и востребованными.

Целью данной практической работы является проектирование и разработка программного обеспечения в виде веб-приложения для составления графика лечения на основании рекомендаций врача.

Из составленной цели были выдвинуты следующие задачи:

- Анализ конкурентов и аналогов приложения;
- Составление функциональных и не функциональных требований;
- Разработка базы данных приложения;
- Составление диаграм взаимодействия пользователей с системой;
- Выбор технологий программирования для создания конечного продукта;
- Анализ бизнес требований к приложению;

1 Сравнение аналогов проектируемой системы

В ходе тщательного анализа и изучения предметной области, мы провели обширный поиск и исследовали существующие аналоги на рынке. Данное исследование включало в себя изучение различных приложений и сервисов, их функционала, возможностей и особенностей. В результате, мы смогли выявить ряд ключевых аналогов, которые в настоящее время существуют и активно используются. Эти аналоги представляют собой различные продукты и решения, которые в той или иной степени могут совпадать с нашими целями и задачами. Изучение этих аналогов помогает нам лучше понять существующий рынок, определить ключевые тренды и возможности для инноваций. Изученные аналоги изображены в таблице 1.1.

Таблица 1.1 — Таблица сравнения аналогов

Название Аналога	Возможности	Тариф	Достоинства	Недостатки
1	2	3	4	5
Medisafe	Мобильное приложение, которое помогает отслеживать приём лекарств по дате и времени приёма с примечанием. Также существует интеграция с приложением хранящим и обновляющим данные о здоровье и есть возможность составлять отчёт о приёме.	Бесплатное доступны базовые функции, остальные как кастомизация уведомлений и внешнего вида и отсутствие рекламы доступны по подписке 4.99\$ в месяц и 39.99\$ в год.	Возможность следить за приёмом лекарств членов семьи и создание отчётности о лечении. Присутствие примечания о способе приёма лекарства и дозировки. Интеграция с приложением следящим за показателями здоровья.	Система уведомлений не гибкая и не настраиваемая.
Mytherapy	Мобильное приложение, которое напоминает о приёме лекарств, отслеживает показатели здоровья и имеет возможность создания отчётов приёме.	Бесплатное приложение	Существование журнала о пропущенных и подтверждённых приёмах лекарств. Отслеживание оставшегося количества лекарств. Поддержка схем дозировок. Выбор измерений показателей в зависимости от заболевания. Семейный доступ к приёму лекарств.	Обязует следить за количеством лекарств. Не удобный выбор времени приёма.

Продолжение таблицы 1.1

1	2	3	4	5
Мои таблетки	Мобильное приложение напоминающее о приёме лекарств. Так же имеющие возможность хранить историю приёмов. И создавать примечание о приёме.	Бесплатно доступно 10 курсов приёмов лекарств. По подписке 1.99\$ в месяц доступно неограниченное количество курсов приёмов.	Гибкая система уведомлений. Архив приёмов лекарств.	Отсутствие семейного доступа и учёта пропущенных приёмов.
Mr.Pillster	Мобильное приложение напоминающие о приёме лекарств и о измерении данных о здоровье. Существует возможность создавать график приёмов и измерений. Семейный доступ к приёмам лекарств. Быстрые кнопки “принять” и “отложить” в пуш-уведомлении	Бесплатно доступно добавление только приёмов лекарств. По подписке 329 рублей раз в три месяца, 749 рублей в год и 1790 рублей единоразово доступны все функции.	Возможность создать напоминание о изменении данных о здоровье. Семейный доступ к приёму лекарств. Возможность отложить приём лекарства при появлении уведомления.	В сравнение с другими приложениями не полный список типов дозировок. Также отсутствие возможности создать собственное примечание к приёму лекарства.
Pills time	Мобильное для напоминаний о приёме лекарств и приёмах к врачу. Семейный доступ к приёмам. Возможность добавить фото к добавленному лекарству. Гибкий выбор периодичности приёма.	Бесплатное добавление курса лекарств, без. По подписке 99 рублей в месяц в год будут доступны все функции.	Большой список примечаний связанных с едой. Семейный доступ. Возможность создать расписание приёмов у врача.	Неудобная система уведомлений. Короткий список типов дозировок.

2 Анализ функциональных и нефункциональных требований

В данной главе будет представлен анализ будущего приложения с точки зрения выделения функциональных и нефункциональных требований.

2.1 Функциональные требования

В процессе анализа требований к веб-приложению для составления графика лечения на основе рекомендаций врача были выделены следующие функциональные требования, которые необходимы для полноценного функционирования приложения:

1. Веб-приложение должно позволять пациенту или его лечащему врачу создавать график лечения на основании рекомендаций и назначений. Для этого необходимо предоставить пользователю возможность добавлять информацию о лекарствах, дозах и промежутках времени приема, а также продолжительность лечения. График должен быть составлен на необходимый период времени и сохраняться для каждого пациента.
2. Приложение должно иметь функцию добавления рекомендаций и назначений, включая лекарства, дозы и промежутки времени приема.
3. Приложение должно позволять врачу указывать продолжительность лечения, чтобы график был составлен на необходимый период времени.
4. Приложение должно предоставлять возможность сохранять и редактировать график лечения для каждого пациента, чтобы врач мог обновлять его при необходимости.
5. Приложение должно иметь функцию проверки наличия противопоказаний и взаимодействия между различными лекарствами, чтобы врач мог быть уверен в безопасности назначенного лечения.

6. Приложение должно предоставлять возможность врачу проверить, были ли выполнены назначения пациентом, путем отметки о выполнении каждой рекомендации.
7. Приложение должно создавать напоминания для пациента о приеме лекарств и выполнении других назначений, которые были добавлены самим пациентом или его лечащим врачом.
8. Приложение должно обеспечивать безопасное хранение конфиденциальной информации о пациентах и доступ только для авторизованных врачей.

2.2 Нефункциональные требования

После формулирования функциональных требований к системе и анализа системы с точки зрения взаимодействия с конечным потребителем, были выделены следующие нефункциональные требования:

1. Веб-приложение должно иметь интуитивно понятный и привлекательный интерфейс для пользователей всех возрастов и уровней навыков.
2. Приложение должно поддерживать различные устройства и разрешения экранов, включая мобильные устройства, планшеты и компьютеры.
3. Гарантия конфиденциальности и безопасности персональных медицинских данных пациентов. Так как данный аспект строго регулируется государством.
4. Обеспечить совместимость с различными современными браузерами, при помощи которых будет исполняться взаимодействие с приложением для удобства пользователей
5. Обеспечить быструю загрузку и отзывчивость интерфейса при работе с графиком лечения и рекомендациями врача.

6. Поддерживать возможность локализации интерфейса на русском языке для удовлетворения потребностей пользователей на территории Российской Федерации.
7. Реализовать авторизацию и управление доступом к приложению и данным для обеспечения безопасности и ограничения доступа к конфиденциальным данным.
8. Обеспечить отказоустойчивость для минимизации времени простоя и потери данных в случае сбоев или неполадок. Приложение должно быть обеспечено необходимыми механизмами для минимизации времени простоя и потери данных в случае возникновения сбоев или неполадок. Это означает, что система должна быть способна восстанавливаться после сбоев без значительного воздействия на функциональность и производительность, обеспечивая непрерывность предоставления услуг пользователям.

3 Диаграммы прецедентов

В данном разделе будут представлены взаимодействия основных пользователей и ролей в системе в виде диаграмм прецедентов и схем активности. Для создания изображений диаграмм было использовано бесплатное программное обеспечения под названием "diagrams.net". [2]

3.1 Диаграмма прецедентов для сущности "Пользователь"

Сущность "Пациент" имеет следующие возможности:

1. Добавить прием лекарства: Эта возможность позволяет пациенту добавить информацию о приеме лекарства. В процессе добавления создается уведомление, которое отправляется пациенту телеграмм-ботом в определенное время, заданное пациентом;
2. Добавить визит врача: Эта возможность позволяет пациенту добавить информацию о визите к врачу. В процессе добавления создается уведомление, которое отправляется пациенту телеграмм-ботом в определенное время, заданное пациентом;
3. Регистрация пациента: Эта возможность позволяет пациенту зарегистрироваться в системе. В процессе регистрации создается экземпляр пациента в базе данных;
4. Редактирование профиля: Эта возможность позволяет пациенту редактировать свои данные в профиле. В процессе редактирования обновляются данные в базе данных;
5. Отправка обратной связи: Эта возможность позволяет пациенту отправить обратную связь о системе или услугах. В процессе отправки обновляются данные в базе данных;

Таким образом, сущность "Пациент" имеет пять возможностей, некоторые из которых включают в себя создание уведомления и их отправку телеграмм-ботом, а также обновление данных в базе данных. Диаграмма сущностей изображена на рисунке 3.1.

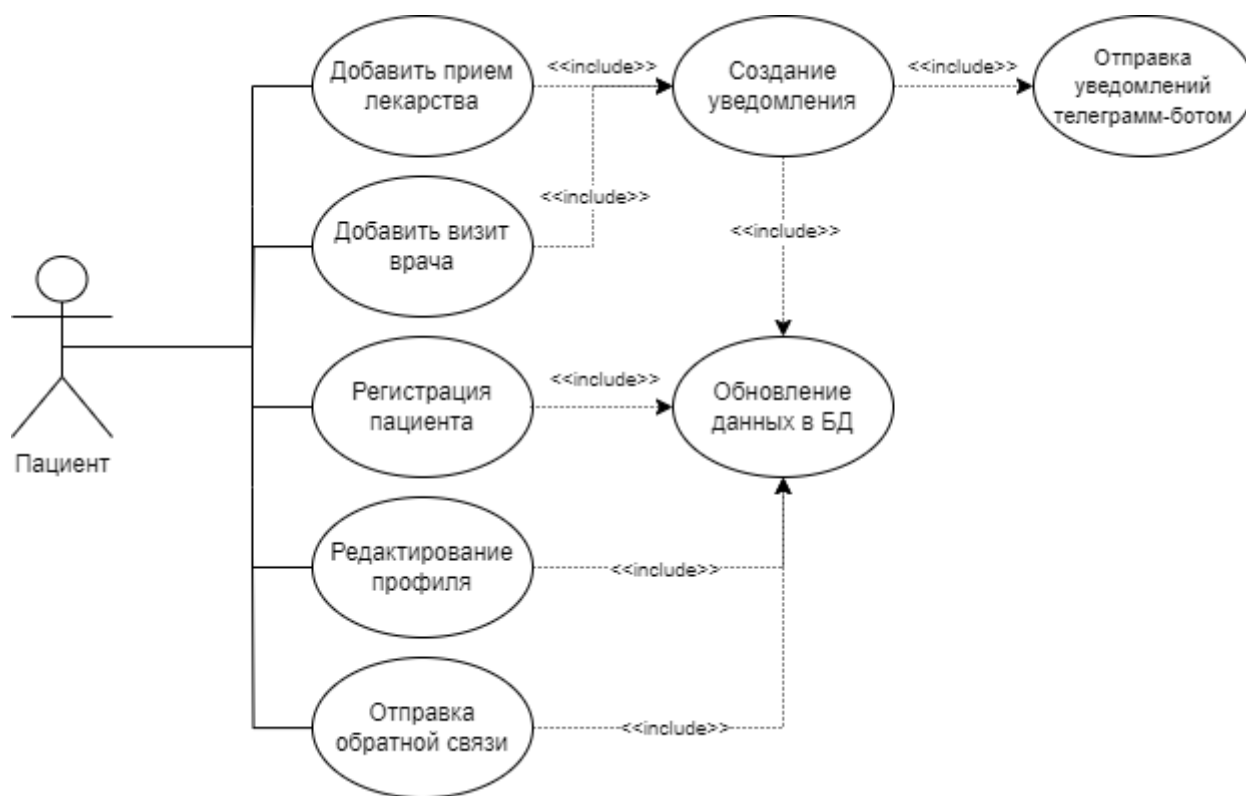


Рисунок 3.1 — Диаграмма прецедентов для роли "Пациент".

3.2 Диаграмма прецедентов для сущности "Доктор"

Сущность "Доктор" имеет следующие возможности:

1. Добавить график лечения пациентов: Эта возможность позволяет доктору добавлять информацию о приемах лекарств для пациента. В процессе добавления создается уведомление, которое отправляется пациенту телеграмм-ботом в определенное время, заданное доктором;
2. Назначить прием: Эта возможность позволяет доктору назначить прием с пациентом. В процессе добавления создается уведомление, которое

отправляется пациенту телеграмм-ботом в определенное время, заданное доктором;

3. Регистрация доктора: Эта возможность позволяет доктору отправить заявку на регистрацию модератору веб-приложения. В процессе регистрации создается экземпляр доктора с атрибутом `active=False` и заявка на подтверждение данных для модератора в базе данных;
4. Редактирование профиля: Эта возможность позволяет доктору редактировать свои данные в профиле. В процессе редактирования обновляются данные в базе данных;
5. Получить список пациентов: Эта возможность позволяет доктору получить список лечащихся и уже прошедших лечение пациентов;

Итоговая диаграмма прецедентов изображена на рисунке 3.2.

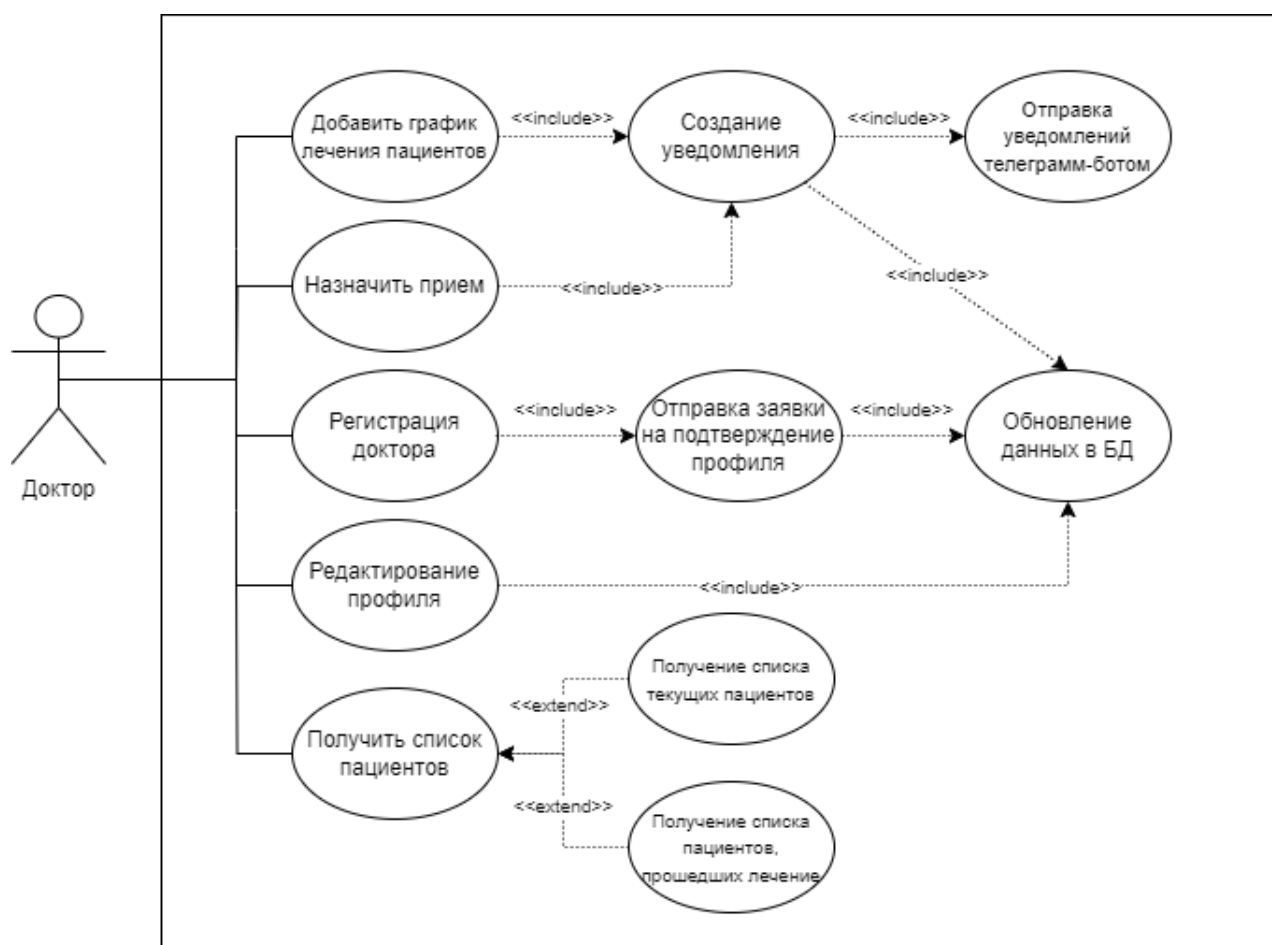


Рисунок 3.2 — Диаграмма прецедентов для роли "Доктор".

3.3 Диаграмма прецедентов для сущности "Менеджер"

Сущность "Менеджер" имеет следующие возможности:

1. Получение отзывов о врачах: Эта возможность позволяет менеджеру просмотреть статистику доктора, такую как количество принятых пациентов, проведенных консультаций, выписанных рецептов и других показателей, которые могут помочь оценить эффективность работы доктора. Менеджер может использовать эту информацию для анализа производительности доктора, определения его сильных и слабых сторон, а также для принятия решений о его дальнейшем развитии и обучении. Эта функция также может быть полезна для определения потребностей пациентов и улучшения качества медицинских услуг, предоставляемых клиникой;
2. Администрирование данных о врачах: Эта возможность позволяет менеджеру подтверждать, отклонять заявку о регистрации доктора, а также, при необходимости, отстранять доктора от врачебной деятельности. Все это влечет за собой обновление данных в БД;
3. Изменение данных пользователя. Это возможность позволяет менеджеру редактировать данные пациента, которые были указаны при регистрации. При изменении данных пользователя, система автоматически сохраняет новые значения в базе данных, чтобы они были доступны для системных процессов, которые могут нуждаться в этой информации;

Итоговая диаграмма прецедентов изображена на рисунке 3.3.

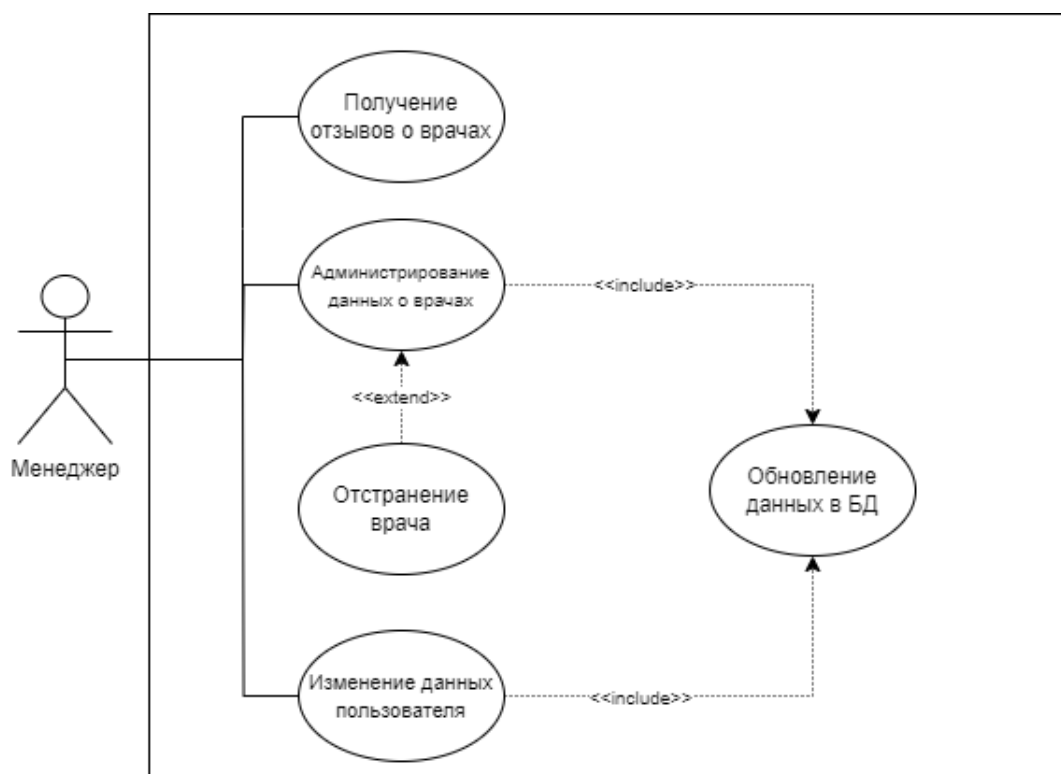


Рисунок 3.3 — Диаграмма прецедентов для роли "Менеджер".

4 Проектирование базы данных приложения

В данной главе будет представлен процесс проектирования базы данных веб-приложения для составления графика лечения на основании рекомендаций врача.

4.1 Потенциальные объекты системы

В процессе анализа предметной области были выделены следующие примерные бизнес-сущности, которые должны быть отображены в базе данных:

- Доктор;
- Пациент;
- Менеджер;
- Прием лекарств;
- Прием врача;
- Лекарственное средство;
- Рекомендации;
- Отзыв;

4.2 Определение атрибутов и первичных ключей

Для оптимизации работы веб-приложения, было принято решение о нормализации базы данных, объединив объекты системы в сущность "Пользователь". Для реализации контроля доступа внутри системы будет введен атрибут роли, который будет ограничивать доступ к изменению и получения данных из системы.

В таблице 4.1 продемонстрированы атрибуты сущности "Пользователь".

Таблица 4.1 — Описание атрибутов сущности "Пользователь".

Название атрибута	Обязательный/не обязательный (*/о)	Уникальный идентификатор (#)	Тип для логической модели
idUser	*	#	Числовой
email	*	(#)	Символьный
role	*		Символьный
firstName	*		Символьный
secondName	*		Символьный
lastName			Символьный
phoneNumber			Символьный
idTelegram			Числовой
idLicense			Символьный
isConfirmed			Логический

Поле "idTelegram" необходимо для авторации и рассылки уведомлений через телеграмм-бота, который будет напоминать врачам и пользователям о изменении их лечебного плана и напоминаний о приеме необходимых лекарственных препаратов.

Поля "idLicense" и "isConfirmed" относятся к роли врача. "idLicense" относится к номеру идентифицирующему документ об образовании врача. Поле "isConfirmed" показывает был ли допущен к работе с пациентами после проверки документов менеджером.

Взаимодействие между пациентом и доктором реализуется при помощи сущности пересечения, которая показывает какие врачи производят наблюдения и рекомендации по лечебному плану для определенного пользователя.

В таблице 4.2 представлена сущность пересечения между ролями системы "Пациент" и "Доктор".

Таблица 4.2 — Описание атрибутов сущности пересечения ролей "Пациент" и "Доктор".

Название атрибута	Обязательный/не обязательный (*/о)	Уникальный идентификатор (#)	Тип для логической модели
idUser	*	#	Числовой
idDoctor	*	#	Числовой

Для реализации нотификаций, была выделена сущность "Прием лекарств". Данная сущность должна хранить в себе атрибуты, для следующих видов приема:

- Прием только один день;
- Ежедневный прием;
- Прием каждые X дней;
- Прием по конкретным датам;
- Цикл приема из X дней и Y дней пропуска;
- Прием через определенный промежуток времени;

Также необходимо хранить необходимое время приема в течении дня для повышения эффективности лечения.

В таблице 4.3 предоставлены атрибуты сущности "Прием лекарств".

Таблица 4.3 — Описание атрибутов сущности "Прием лекарств".

Название атрибута	Обязательный/не обязательный (*/о)	Уникальный идентификатор (#)	Тип для логической модели
idPill	*	#	Числовой
idUser	*		Числовой
type	*		Символьный
name	*		Символьный
form			Символьный
note			Символьный
startDate	*		Даты и времени
endDate			Даты и времени
time			Временной
days			Дата
deltaTakeDays			Числовой
deltaPassDays			Числовой
timeDelta			Временной
lastNotify			Даты и времени

Для отслеживание удовлетворенности пользователей в услугах выбранного лечащего врача была введена сущность "Отзыв". Данная сущность предназначена для администрирования площадки менеджером и отслеживании качества предоставляемых докторами услуг.

Атрибуты сущности "Отзыв" предоставлены в таблице 4.6

Таблица 4.4 — Описание атрибутов сущности "Отзыв".

Название атрибута	Обязательный/не обязательный (*/о)	Уникальный идентификатор (#)	Тип для логической модели
idComment	*	#	Числовой
idPatient	*		Числовой
idDoctor	*		Числовой
rate	*		Числовой
review			Символьный

Для эффективного процесса лечения также необходимы своевременное посещение лечащего врача. Для этого в системе необходимо хранить информацию о плановых посещениях, которые представлены в сущности "Прем врача".

В таблице 4.5 продемонстрированы атрибуты сущности "Прием врача".

Таблица 4.5 — Описание атрибутов сущности "Отзыв".

Название атрибута	Обязательный/не обязательный (*/о)	Уникальный идентификатор (#)	Тип для логической модели
idVisit	*	#	Числовой
idPatient	*		Числовой
idDoctor			Числовой
appointmentTime	*		Даты и времени
note			Символьный

Для предоставления комфортного пользования сервисом были выделены справочные сущности, которые позволяют пользователю ознакомиться с препаратами и предоставить поддержку советами о правильном лечении.

Сущность "Лекарственное средство" является справочной сущностью возможных лекарственных препаратов. Атрибуты данной сущности продемонстрированы в таблице ??

Таблица 4.6 — Описание атрибутов сущности "Отзыв".

Название атрибута	Обязательный/не обязательный (* / o)	Уникальный идентификатор (#)	Тип для логической модели
idPillType	*	#	Числовой
name	*		Символьный
dose	*		Числовой
form	*		Символьный
description			Символьный

Сущность "Совет" является справочной сущностью, которая должна помочь пользователю придерживаться эффективного плана лечения. Атрибуты сущности "Совет" предоставлены в таблице 4.7.

Таблица 4.7 — Описание атрибутов сущности "Отзыв".

Название атрибута	Обязательный/не обязательный (* / o)	Уникальный идентификатор (#)	Тип для логической модели
idTips	*	#	Числовой
title	*		Символьный
content	*		Символьный

4.3 Логическое проектирование базы данных

Логическое проектирование базы данных веб-приложения было произведено при помощи программы "PgAdmin" [?]. Данный комплекс программного обеспечения является мощным инструментом для логического проектирования и администрирования баз данных. Оно предоставляет возможность создания ERD (Entity-Relationship Diagram) диаграмм, которые представляют структуру базы данных и связи между ее элементами. "PgAdmin" также позволяет конвертировать ERD диаграммы в SQL-скрипты, которые можно использовать для создания таблиц в базе данных. Это значительно упрощает процесс разработки, так как позволяет автоматически генерировать SQL-скрипты на основе диаграммы. Результат преобразования сущностей в ERD диаграмму изображен на рисунке 4.1.

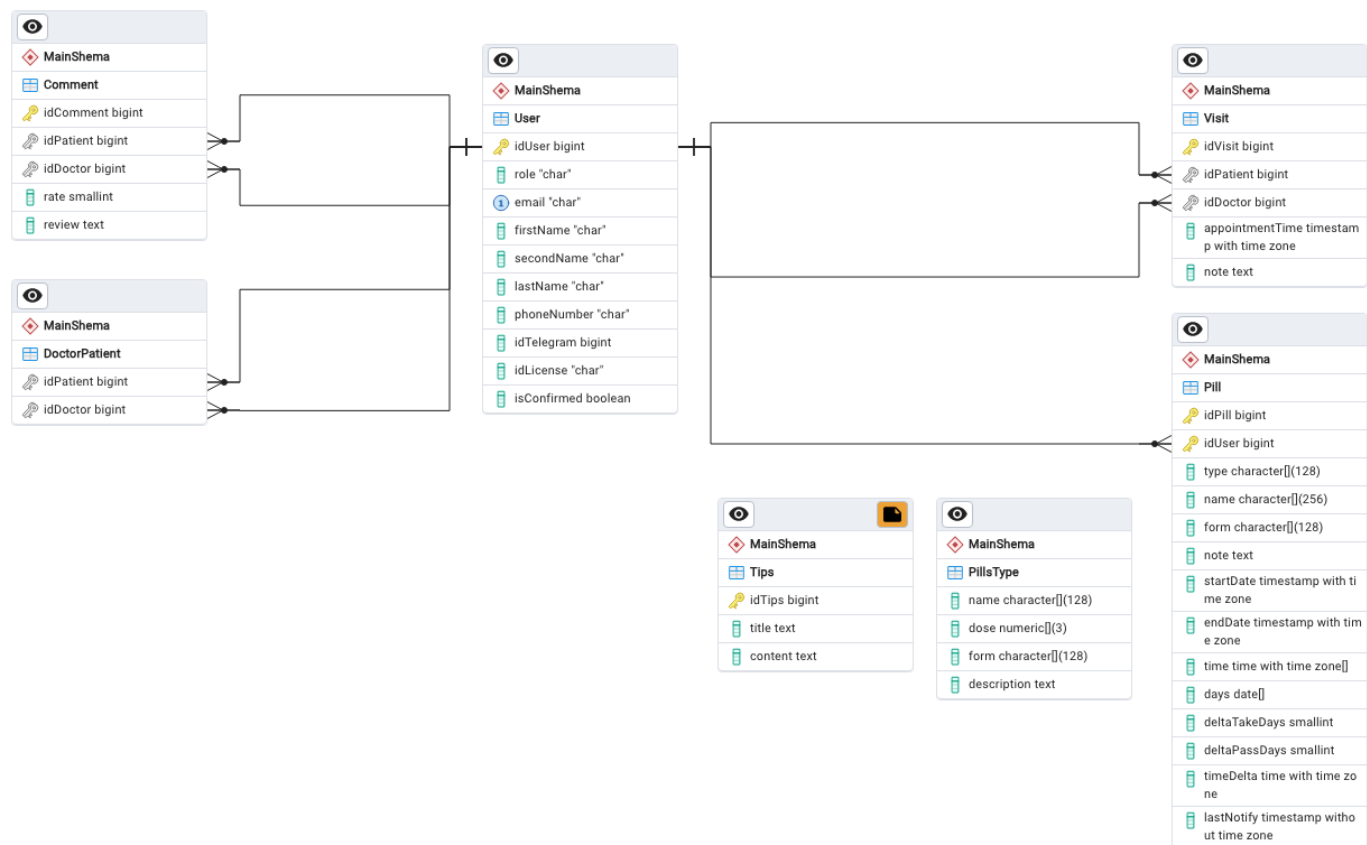


Рисунок 4.1 — ERD диаграмма базы данных.

5 Алгоритмы работы с пользователем

В данном разделе будут представлены алгоритмы работы с потенциальным пользователем веб-приложения.

5.1 Диаграммы активности

Диаграмма активности “Регистрация пациента” представляет собой последовательность следующих действий:

1. Пользователь отправляет данные для регистрации;
2. Происходит проверка данных на валидность, в случае не валидности данных пользователь вводит данные снова, а в случае валидности - происходит следующая проверка;
3. Происходит проверка на уникальность пациента, в случае не уникальности пациента - пользователь вводит данные снова, в ином случае - происходит регистрация пользователя;

Диаграмма активности “Регистрация пациента” изображена на рисунке 5.1.

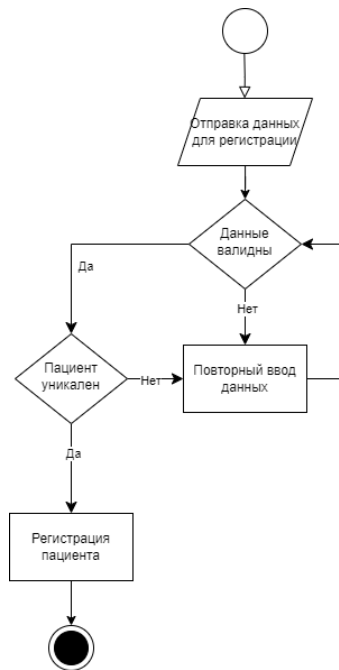


Рисунок 5.1 — Диаграмма активности “Регистрация пациента”.

Диаграмма активности “Получить список пациентов” представляет собой последовательность следующих действий:

- Пользователь отправляет запрос на получение списка пациентов;
- Происходит проверка на существование данных, в случае существования данных происходит следующий этап - получение списка пациентов, в ином случае - происходит извещение доктора об отсутствии пациентов;

Диаграмма активности “Получить список пациентов” изображена на рисунке 5.2.



Рисунок 5.2 — Диаграмма активности “Получить список пациентов”.

Диаграмма активности “Администрирование данных о врачах” представляет собой последовательность следующих действий:

1. Менеджер отправляет запрос на получение данных о врачах;
2. Происходит проверка корректности данных, если данные корректны, то происходит следующий этап, в ином случае происходит отказ в доступе к пациентам для врача;
3. Происходит проверка на приемлемость отзывов лечащего врача, оставленных пациентами, в случае, если отзывы приемлемы врач утверждается, в ином случае - происходит отказ в доступе к пациентам для врача;

Диаграмма активности “Администрирование данных о врачах” изображена на рисунке 5.3.

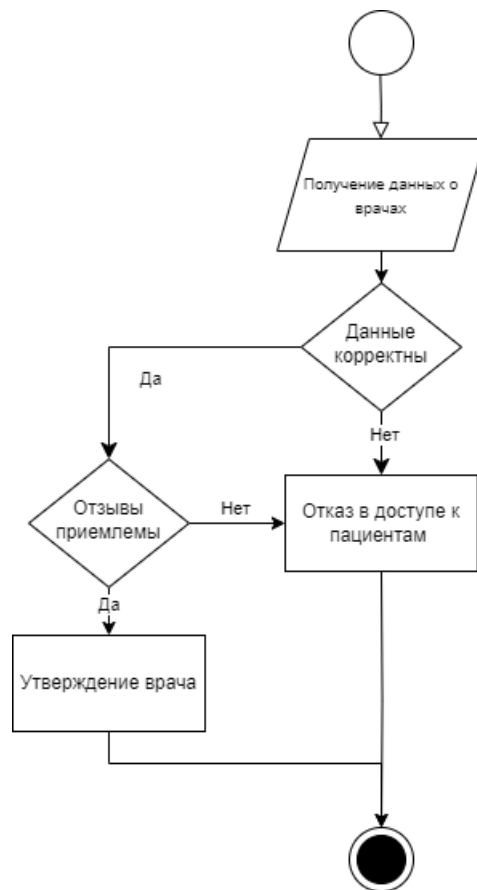


Рисунок 5.3 — Диаграмма активности “Администрирование данных о врачах”.

5.2 Контекстные диаграммы взаимодействия

В данном подразделе представлена модель в стандарте IDEF3, которая описывает информационные потоки и взаимоотношения между процессами обработки информации и объектами, являющимися частью этих процессов. Модель представлена в виде контекстной диаграммы (рисунок 5.4), которая представляет собой графическое представление основных компонентов системы и их взаимосвязей и позволяет определить основные процессы и объекты, которые участвуют в обработке информации, а также информационные потоки между ними.

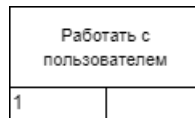


Рисунок 5.4 — Контекстная диаграмма (IDEF3)

Декомпозиции этой диаграммы изображена на рисунке 5.5. Декомпозиция представляет собой более детальное представление процессов и объектов, описанных в контекстной диаграмме, позволяет более подробно описать каждый процесс и объект, а также их взаимосвязи. Кроме того, она включает в себя описание входных и выходных данных, используемых ресурсов, исполнителей, необходимых для выполнения процессов.

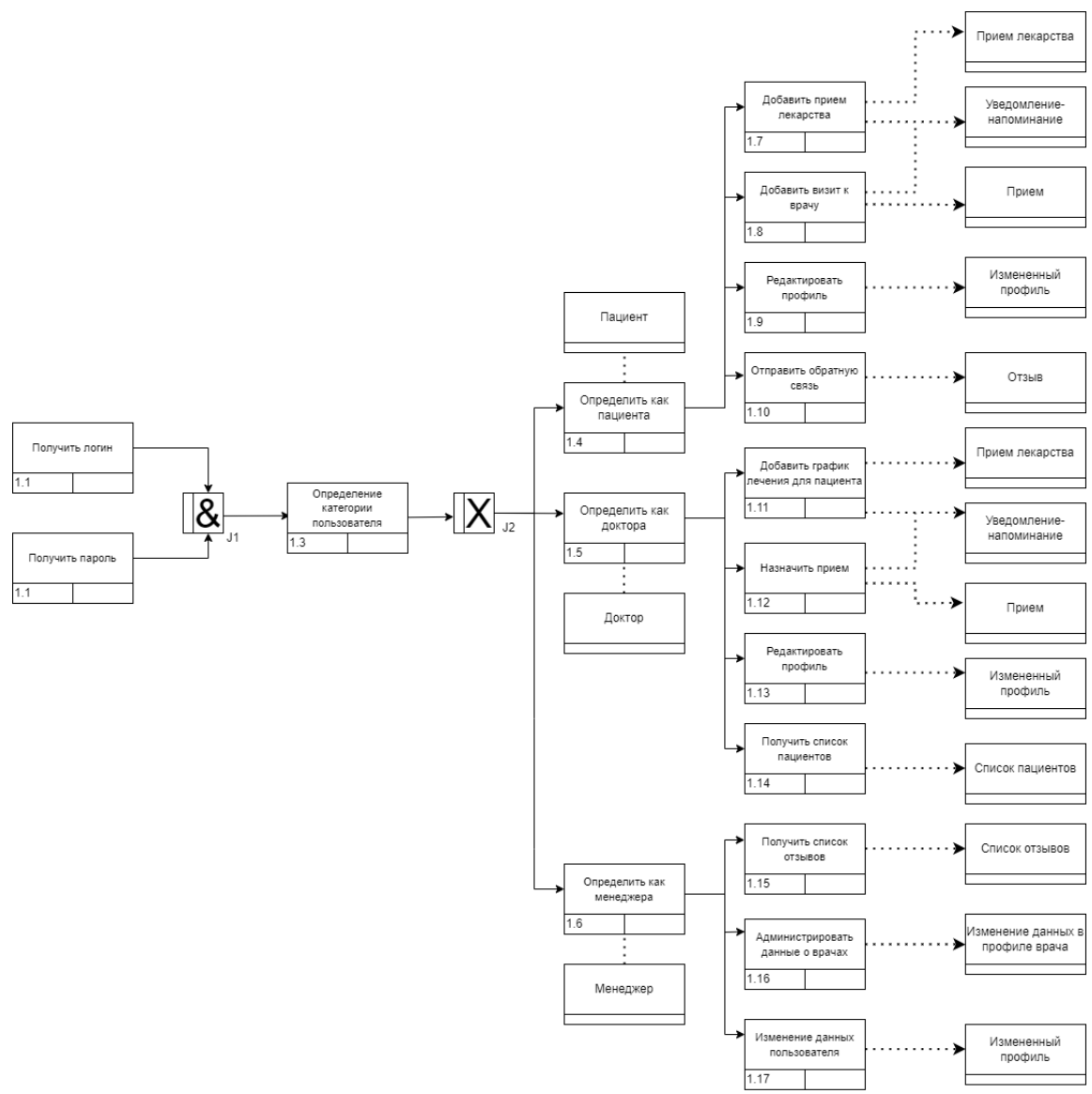


Рисунок 5.5 — Декомпозиция контекстной диаграммы

Декомпозиции блока “Администрировать данные о врачах” представлена на рисунке 5.6

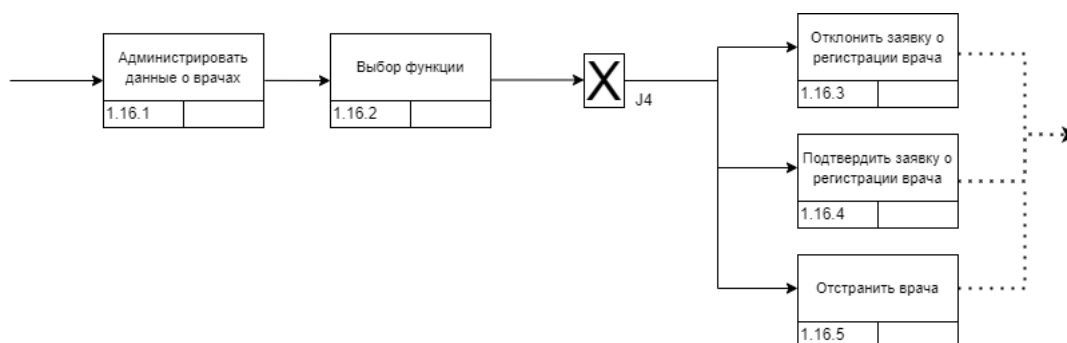


Рисунок 5.6 — Декомпозиции блока “Администрировать данные о врачах”

Декомпозиция блока “Получить список пациентов” представлена на рисунке 5.7

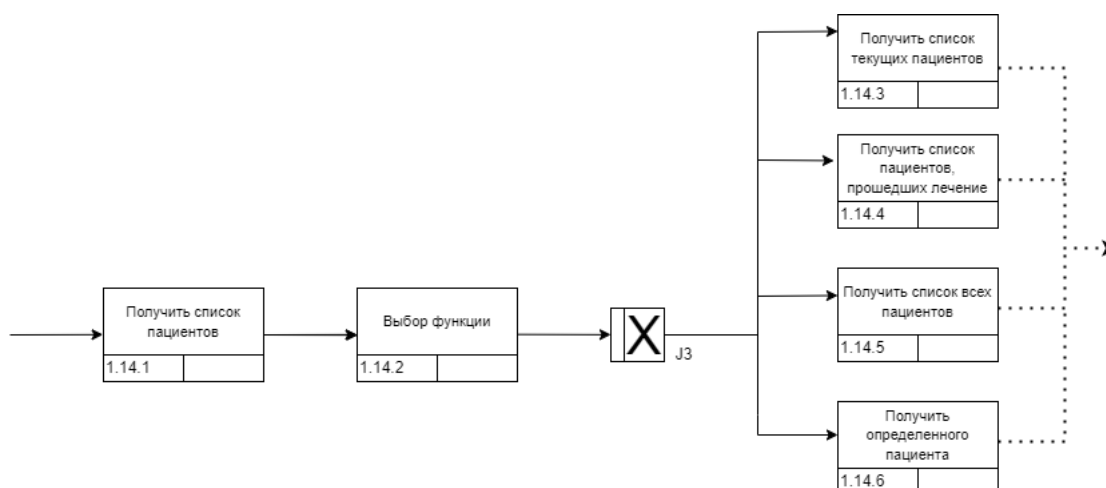


Рисунок 5.7 — Декомпозиции блока “Администрировать данные о врачах”

Декомпозиция блока “Редактировать профиль” представлена на рисунке 5.8

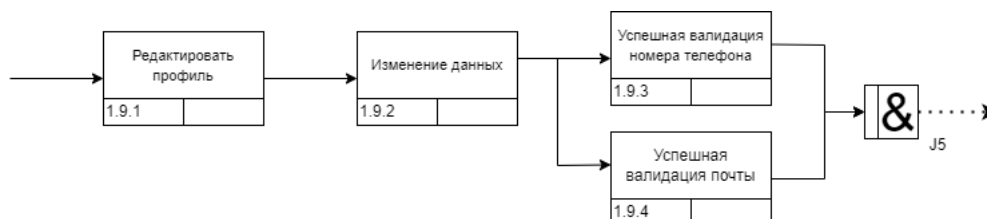


Рисунок 5.8 — Декомпозиции блока “Администрировать данные о врачах”

6 Технологии программирования

В данной главе будут представлены выбранные технологии программирования которые будут применены при разработке веб-приложения для составления графика лечения на основании рекомендаций врача.

6.1 Веб-приложение

Для реализации клиент-серверной архитектуры будут применяться следующие технологии программирования:

1. Django [3] - это обширная библиотека для языка Python, которая предоставляет множество инструментов для разработки веб-приложений. Она обеспечивает высокую производительность, безопасность и масштабируемость. Django также имеет встроенную поддержку для работы с базами данных, шаблонизации и аутентификации.
2. React [4] - это библиотека для создания пользовательских интерфейсов на языке JavaScript. Она предоставляет мощные инструменты для создания интерактивных и высокопроизводительных веб-приложений. React позволяет легко управлять состоянием приложения и обновлять его части, что делает его идеальным выбором для разработки сложных интерфейсов.

6.2 Система уведомлений пользователей

Для реализации системы уведомлений пользователей было принято решение воспользоваться телеграмм-ботом для рассылки уведомлений. Данный выбор связан с обширной базой пользователей телеграма [5]. Потому что, как правило, у потенциальных пользователей данный меснджер установлен на всех основных устройствах.

Для программной реализации данной концепции была выбрана библиотека Aiotelegram [6]. Данная библиотека предназначена для работы с интерфейсом программирования приложения (API) Telegram. Она позволяет разработчикам создавать ботов для Telegram и создавать интерактивные пути взаимодействия с пользователем. В данном случае будет создан телеграмм-бот для отправки уведомлений пациентам.

6.3 Развертывание приложения

Для успешной эксплуатации приложения, планируется использовать современные технологии развертывания приложений. В достижении гибкости использования предоставленной провайдером серверной архитектуры будет использоваться инструмент контейнеризации приложений под названием Docker [7]. Так как приложение является небольшим, то для оркестрации данных контейнеров будет использоваться поставляемый вместе с Docker инструмент Docker-compose [8]. Использование данного подхода технологий позволит создать надежное и масштабируемое веб-приложение для составления графика лечения на основании рекомендаций врача.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аудитория интернета в 2022 году.: MediaScope [Электронный ресурс]. - 2023. - URL: https://mediascope.net/upload/iblock/3d8/qrlhud7t7dxyzw1rhtzxcg3rwx8deg7uk/2022_P4P5P5P5P5P5P5P5.pdf (дата обращения 10.02.2023)
2. diagrams.net.: Официальный сайт. - URL: https://vk.com/wall-132685896_980 (дата обращения 10.02.2023)
3. Django.: Официальный сайт. - URL: <https://www.djangoproject.com> (дата обращения 10.02.2023)
4. React.: Официальный сайт. - URL: <https://ru.legacy.reactjs.org> (дата обращения 10.02.2023)
5. Telegram.: Официальный сайт. - URL: <https://telegram.org> (дата обращения 10.02.2023)
6. aiogram.: Официальный сайт. - URL: <https://aiogram.dev> (дата обращения 10.02.2023)
7. Docker.: Официальный сайт. - URL: <https://www.docker.com> (дата обращения 10.02.2023)
8. Docker.: Официальный сайт. - URL: <https://docs.docker.com/compose/> (дата обращения 10.02.2023)