

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)

Бизнес-информатика  
(направленность (профиль))

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВАРСКАЯ РАБОТА)

на тему Разработка системы виртуализации интерьера на основе BabylonJS

Общающийся

Д.А. Добрынин  
(Инициалы Фамилия)

(личная подпись)

Руководитель

Н.Н. Казаченок

(ученая степень, учебное звание, Инициалы Фамилия)

Тольятти 2023

## Аннотация

Бакалаврская работа выполнена на тему «Разработка системы виртуализации интерьера на основе BabylonJS».

Актуальность исследования обуславливается разработкой и внедрением системы, позволяющей оптимизировать внешние и внутренние потоки информации, а также модернизировать существующие процессы внутри компании, что существенно ускорит работу организации в целом.

Цель работы заключается в разработке системы виртуализации интерьера на основе BabylonJS.

Во введении представлены: актуальность темы исследования, определены объект и предмет исследования, поставлена цель и определена ведущая идея, выдвигается гипотеза и формулируются задачи работы, характеризуется научная новизна, теоретическая и практическая значимость результатов исследования, дается краткое содержание глав работы.

В первой главе данной бакалаврской работы раскрыта характеристика предприятия, направления деятельности и её структурные подразделения. Построены модели бизнес-процессов AS-IS и TO-BE.

Во второй главе рассмотрена технология проектирования, представлены диаграмма вариантов использования и диаграммы классов системы виртуализации интерьера, а также построенная концептуальная и логическая модель данных, определены требования к программно-аппаратному обеспечению, приведены прототипы дизайна системы.

Третья глава содержит выбор технологии разработки информационной системы, СУБД, построение физической модель системы, реализацию модулей информационной системы и проведение нагрузочного тестирования.

Четвертая глава включает оценку экономической эффективности информационной системы.

Бакалаврская работа представлена на 79 страницах и включает в себя 24 рисунка, 14 таблиц, 29 источников литературы, 4 приложения.

## Оглавление

Введение.....	5
Глава 1 Анализ и моделирование деятельности ООО «1Т».....	8
1.1 Общая характеристика предприятия .....	8
1.2 Направления деятельности и структурные подразделения.....	9
1.3 Моделирование бизнес-процессов.....	13
1.3.1 Модель бизнес-процесса AS-IS.....	13
1.3.2 Модель бизнес-процесса TO-BE .....	15
1.4 Выявление недостатков.....	17
1.5 Определение бизнес-требований .....	18
1.6 Требования к разрабатываемой информационной системе виртуализации 3D интерьеров «Interior Constructor».....	20
Глава 2 Проектирование информационной системы «Interior Constructor» ...	22
2.1 Выбор технологии проектирования.....	22
2.2 Варианты использования .....	23
2.3 Диаграмма классов системы виртуализации интерьера .....	30
2.3.1 Прототип диаграммы классов .....	31
2.3.2 Детализированная диаграмма классов .....	32
2.4 Концептуальная модель данных .....	34
2.5 Логическая модель данных.....	36
2.6 Требования к программно-аппаратному обеспечению .....	38
2.7 Прототипы дизайна системы .....	39
Глава 3 Разработка информационной системы виртуализации интерьера «Interior Constructor» .....	43
3.1 Выбор технологии разработки .....	43
3.2 Выбор СУБД.....	45
3.3 Физическая модель данных .....	46
3.4 Модули информационной системы виртуализации интерьера «Interior Constructor».....	47

3.5	Бэкенд информационной системы «Interior Constructor» .....	48
3.5.1	Реализация моделей посредством библиотеки Mongoose.....	50
3.5.2	Интерфейс программирования приложения (API) для информационной системы «Interior Constructor» .....	50
3.6	Фронтенд информационной системы «Interior Constructor» .....	52
3.6.1	Модуль регистрации и авторизации .....	53
3.6.2	Модуль навигации и разграничения доступа .....	54
3.6.3	Модуль управления интерьерами .....	58
3.7	Нагрузочное тестирование информационной системы «Interior Constructor» при помощи Yandex.Tank.....	59
Глава 4	Расчет экономической эффективности внедрения информационной системы виртуализации интерьера «Interior Constructor».....	62
4.1	Выбор и обоснование метода расчета экономической эффективности .....	62
4.2	Расчет фактических затрат на реализацию информационной системы виртуализации интерьера «Interior Constructor» .....	63
4.3	Расчет ожидаемого экономического эффекта от использования результатов проекта .....	68
	Заключение .....	70
	Список используемой литературы и используемых источников.....	71
	Приложение А Код эндпоинта sign-in.....	76
	Приложение Б Отчет о проведенном нагрузочном тестировании.....	77
	Приложение В Список задач календарного плана .....	78
	Приложение Г Фрагмент сетевого графика календарного плана .....	79

## Введение

В настоящее время цифровые решения активно применяются на различных уровнях управления – от отраслей до отдельных предприятий. Использование цифровых систем позволяет компаниям сократить временные затраты на выполнение операций, своевременно иметь информацию для принятия оптимальных решений и повысить качество выпускаемой продукции.

Тема исследования «Разработка системы виртуализации интерьера на основе BabylonJS» позволит совместить современный подход и новейшие технологии на базе легковесного кроссбраузерного JavaScript-фреймворка, используя 2D и 3D-графику в браузере без использования каких-либо сторонних плагинов и дополнений.

Актуальность настоящего исследования обуславливается, во-первых, разработкой и внедрением информационной системы, позволяющей оптимизировать внешние и внутренние потоки информации, а во-вторых, модернизировать существующие процессы внутри компании, а также существенно ускорить работу организации в целом.

Объект исследования: процесс виртуализации интерьера в ООО «1Т».

Предмет исследования: автоматизация бизнес-процесса разработки пользовательских трехмерных сцен на основе оптимального ИТ-решения.

Деятельность компании ООО «1Т» сосредоточена на разработке программных продуктов в соответствии с вызовами современного цифрового общества. Программными решениями охвачены сферы торговли, образования, консалтинга и экспертизы.

Цель исследования: разработка системы виртуализации интерьера на основе BabylonJS.

Гипотеза исследования состоит в том, что продуктивная разработка пользовательских трехмерных интерьеров будет достигнута, если:

- заблаговременно выявить и проанализировать бизнес-процессы;

- описать желаемый программный продукт в соответствии с требованиями, предъявляемыми компанией;
- разработать информационную систему;
- провести необходимые мероприятия по тестированию и интеграции информационной системы в бизнес-процессы компании.

Для достижения поставленной цели необходимо решить следующие задачи, которые формулируются из гипотезы данного исследования:

- дать общую характеристику предприятия и выделить бизнес-процесс, требующий автоматизации;
- разработать модели бизнес-процесса AS-IS и TO-BE;
- определить бизнес-требования к разрабатываемой информационной системе;
- спроектировать информационную систему;
- выбрать технологию проектирования, определить требования к программно-аппаратному обеспечению и дизайну системы;
- построить диаграмму вариантов использования и диаграмму классов системы виртуализации интерьера;
- отобразить концептуальную и логическую модель данных;
- разработать информационную систему с помощью выбранной технологии разработки;
- провести нагрузочное тестирование;
- рассчитать экономическую эффективность внедрения системы.

Научная новизна исследования заключается в использовании современной JavaScript-библиотеки для отображения трехмерных объектов, а именно для реализации системы, позволяющей пользователю создавать собственные интерьеры и другие группы объектов.

Теоретическая значимость исследования заключается в:

- рассмотрении алгоритма разработки программных компонентов;
- анализа бизнес-процессов;

- исследования стратегических и функциональных требований к системе;
- изучении и применении библиотеки BabylonJS.

Практическая значимость исследования представлена разработанной системой виртуализации интерьера на основе BabylonJS.

Личное участие автора в организации и проведении исследования состоит в качестве бизнес-аналитика и разработчика.

В первой главе раскрывается характеристика предприятия, направления деятельности и её структурные подразделения. Построены модели бизнес-процесса AS-IS и TO-BE.

Во второй главе раскрываются технологии проектирования, представлена диаграмма вариантов использования, приведены разработанные диаграммы классов системы виртуализации интерьера, а также построена концептуальная и логическая модель данных, определены требования к программно-аппаратному обеспечению и продемонстрированы прототипы дизайна системы.

В третьей главе раскрывается выбор технологии разработки, обоснован выбор СУБД, построена физическая модель системы, реализованы модули информационной системы, проведено нагрузочное тестирование при помощи Yandex.Tank в контейнере Docker.

В четвертой главе раскрывается оценка экономической эффективности информационной системы.

В заключении представлены основные результаты поставленных задач исследования.

Результатом проделанной работы является разработанная система виртуализации интерьера на основе BabylonJS.

Основной текст работы изложен на 79 страницах, иллюстрирован 14 таблицами, 24 рисунками, использовано 29 источников литературы, 4 приложения.

## **Глава 1 Анализ и моделирование деятельности ООО «1Т»**

### **1.1 Общая характеристика предприятия**

Общество с ограниченной ответственностью «1Т», реализует свою деятельность на основании Устава и действует в соответствии с положениями Гражданского кодекса РФ от 30.11.1994 № 51-ФЗ [6], а также Федерального закона от 08.02.1998 № 14-ФЗ «Об обществах с ограниченной ответственностью» [19].

Сокращенное фирменное наименование организации на русском языке: ООО «1Т».

Юридический адрес: 115093, г. Москва, ул. Серпуховская Б., дом 44, пом. I, ком.20

Основной вид деятельности ООО «1Т»: Деятельность в области информации и связи. ОКВЭД – 62.01 Разработка компьютерного программного обеспечения [20].

Сайт: <https://1t.ru/>

Основной целью компании ООО «1Т» является разработка программных продуктов в соответствии с требованиями современного цифрового общества [16].

Основные задачи, которые ставит перед собой компания:

- разработка программного обеспечения;
- создание проектов для бизнес- и государственных организаций;
- исследование конъюнктуры рынка;
- оказание консультативных, бухгалтерских, информационных, посреднических и иных услуг;
- проведение рекламной деятельности;
- организация планирования, проектирования компьютерных систем и подготовка их к эксплуатации;



- проведение технического обслуживания и ремонт офисных машин и вычислительной техники;
- обучение пользователей.

Компания занимается совершенствованием базовых продуктов в сфере образования. Например, на постоянной основе осуществляется техническое, технологическое и контентное развитие сайта «Учись дома онлайн» – национальной платформы педагогического образования, созданной по заказу Минобрнауки России [16].

Особое внимание компания уделяет квалифицированной консультационной поддержке клиентов и бизнес-партнеров, качественному техническому сопровождению своих продуктов.

Компания осуществляет образовательную деятельность в сфере информационных технологий по образовательным программам «Web-разработчик», «Менеджер IT-проектов», «Системный аналитик», «Дизайнер UI/UX», «Digital-маркетолог», «Бизнес-акселератор IT-проектов «IT АКСЕЛЬ» в рамках федерального проекта «Кадры для цифровой экономики» и образовательной программе «Инженер данных» в рамках федерального проекта «Искусственный интеллект» национальной программы «Цифровая экономика Российской Федерации».

## **1.2 Направления деятельности и структурные подразделения**

Компания ООО «1Т» выполняет широкий перечень видов работ. Основные направления работы ООО «1Т» следующие:

- корпоративные системы, представляющие собой системы автоматизации бизнес-процессов, которые включают инструменты для управления взаимодействием с клиентами, управления персоналом, проектами, документооборотом и другое;
- системы психологической диагностики и развития, то есть исследование и тренинг когнитивных способностей и свойств

- интеллекта, внимания, памяти, точности, скорости реакций и работоспособности в разных условиях для решения задач в сферах образования, профессиональной ориентации и управления кадрами;
- системы дистанционного обучения, представляющие собой среду для создания курсов повышения квалификации и цифровых образовательных ресурсов с инструментами управления динамикой обучения, интерактивным контентом, проверкой знаний и взаимодействием с обучающимися на всех стадиях работы;
  - электронные торговые площадки, включающие в себя отраслевые площадки, позволяющие в онлайн режиме купить или продать товар оптом по оптимальной рыночной цене при помощи аукциона;
  - торговые алгоритмы, работающие на основе технологий машинного обучения и искусственного интеллекта (например: были разработаны более 200 торговых алгоритмов на фондовом рынке с математическим ожиданием не менее 60% прибыльных сделок на интервале 3-х лет).

Широкий перечень компетенций, позволяет компании выполнять разнообразные комплексные проекты:

- информационные, включающие взаимодействие со средствами массовой информации, лидерами мнений, проведение пресс-конференций и информационное продвижение и сопровождение событий онлайн и офлайн;
- информационно-технические, позволяющие управлять жизненным циклом цифровых ресурсов, а также включающие в себя разработку и сопровождение, обеспечение безопасности;
- организационные, включающие проведение мероприятий, в том числе, международных, а также конференций, форумов, выставок, мониторинговых исследований, фокус-групп;

- экспертно-аналитические, включают в себя анализ и оценку проектов на предмет реализуемости, соответствия требованиям, финансово-экономическим и другим показателям;
- рыночные, содержащие в себе сопровождение и автоматизацию деятельности организаций и проведение сделок на различных рынках, в том числе в сфере производства и торговли;
- образовательные, включая в себя разработку образовательных ресурсов, программ повышения квалификации и переподготовки;
- финансовые, состоящие из ведения деятельности сертифицированного финансового советника: проект, разрабатываемый на базе искусственного интеллекта, где основной упор делается на анализ рисков и безопасность инвестиций.

В организационной структуре ООО «1Т» Генеральный директор является единоличным исполнительным органом компании.

Организационная структура ООО «1Т» основана на функциональном подразделении на определенные отделы. Оно характеризуется тем, что на верхнем уровне сосредоточены стратегические и оперативные директора. Решения спускаются сверху вниз.

Суммарная численность персонала, следующая:

- эксперты и методисты – 110 человек;
- программисты – 130 человек;
- аналитики – 80 человек;
- контакт-центр – 40 человек;
- отдел документооборота – 20 человек;
- дирекция и менеджмент – 25 человек.

Общая схема организационной структуры ООО «1Т» представлена на рисунке 1.

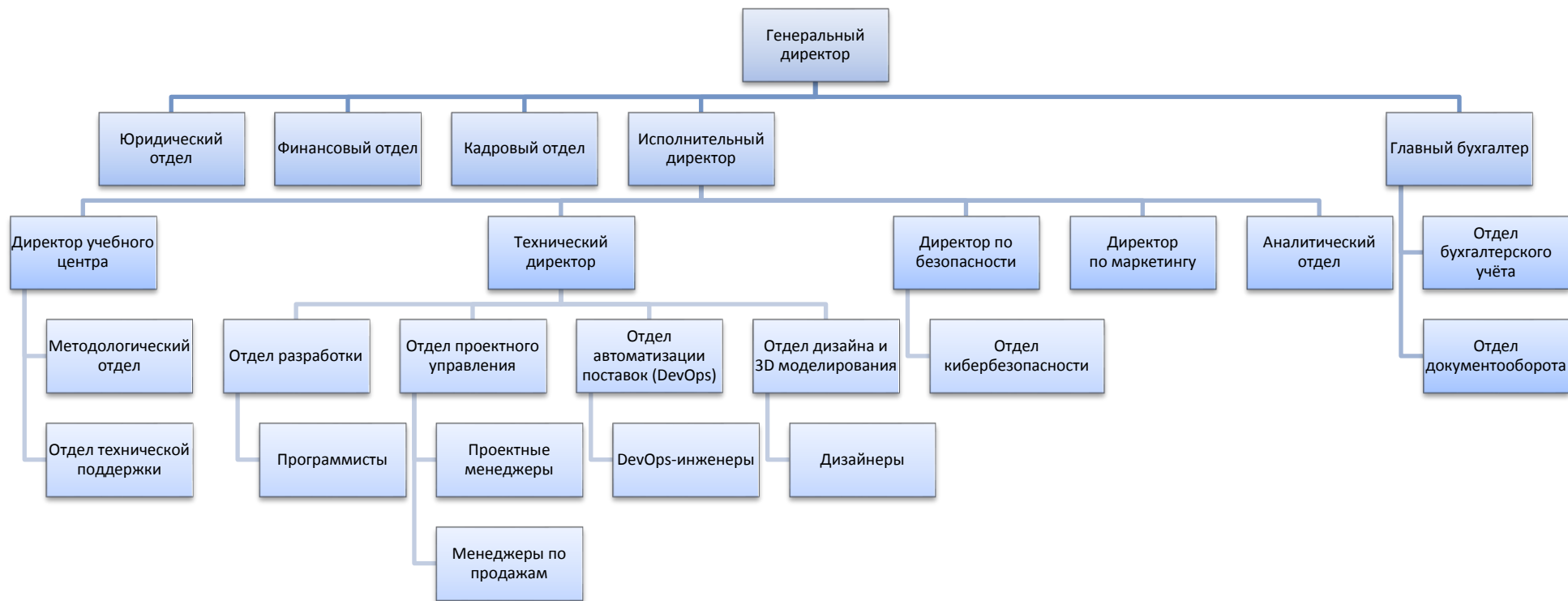


Рисунок 1 – Схема организационной структуры ООО «1Т»

Преимуществом данной линейно-функциональной организационной структуры является высокая степень специализации в определенной группе. Существует ясный порядок подчинения. Ответственность лежит на определенных людях, а также такой системе свойственна высокая эффективность и скорость работы, каждый отдел выполняет поставленные ему задачи.

### **1.3 Моделирование бизнес-процессов**

Моделирование бизнес-процессов (от англ. Business Process Modeling) является одним из методов повышения эффективности и прозрачности работы в организации [26]. Моделирование позволяет точно представить цели, исследуемые характеристики и конечные результаты каждого вида деятельности. Бизнес-процессы определяют прохождение потоков работ независимо от иерархии и границ подразделений, которые их выполняют [9].

В настоящий момент существует множество методологий и нотаций, которые применяются в целях моделирования, проектирования и анализа [12]. В качестве методологии моделирования бизнес-процессов в компании ООО «1Т» решено использовать методологию BPMN 2.0. BPMN является нотацией, которая моделирует шаги запланированного бизнес-процесса от начала до завершения. Наглядная блок-схема отображает полную последовательность операций и информационных потоков.

#### **1.3.1 Модель бизнес-процесса AS-IS**

Создание модели бизнес-процесса AS-IS позволяет выявить границы и основные компоненты процесса. Была разработана BPMN модель AS-IS бизнес-процесса «Процесс создания визуального представления трехмерного пространства», представленная, соответственно, на рисунке 2.

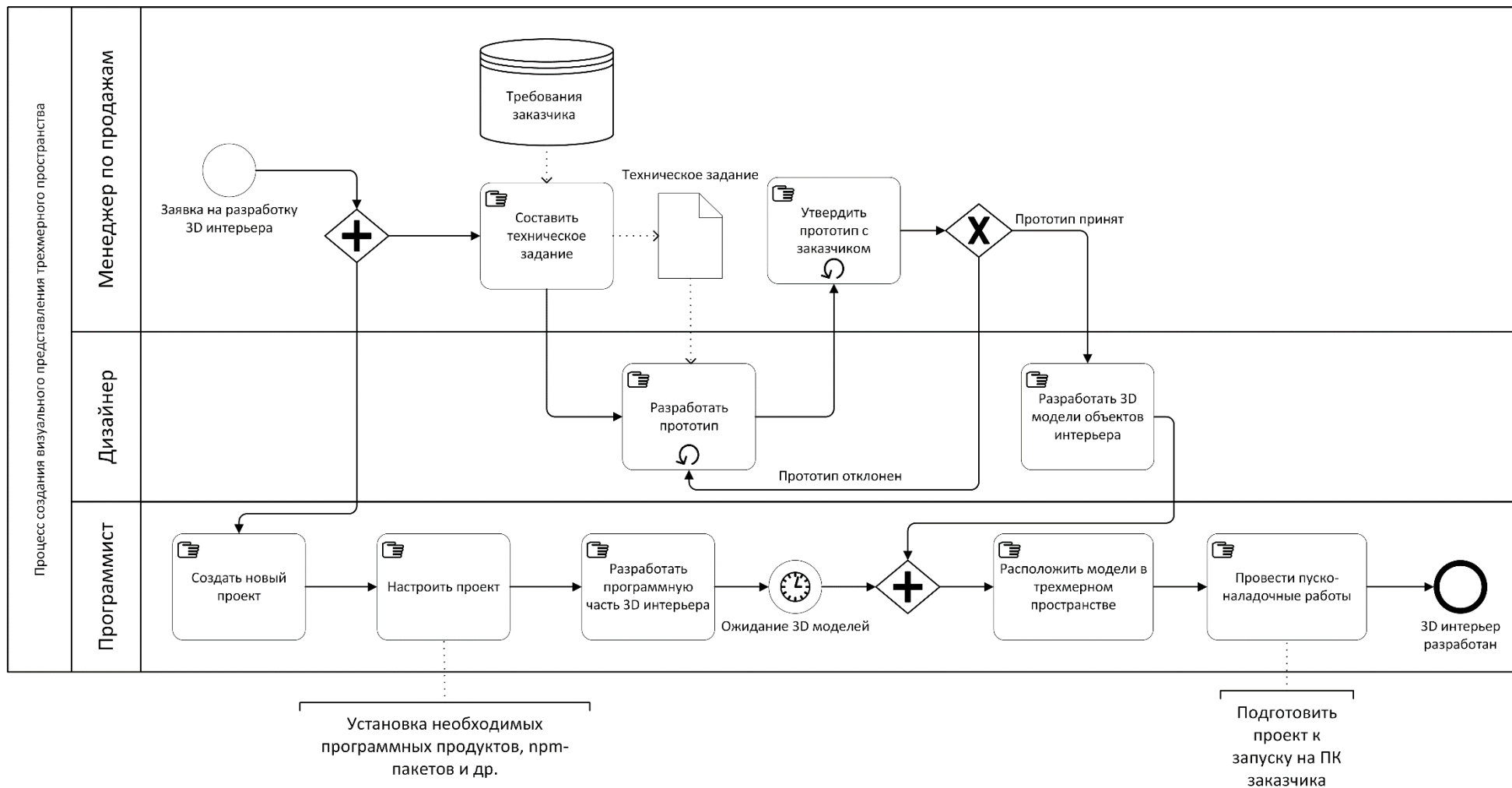


Рисунок 2 – Модель BPMN AS-IS бизнес-процесса «Процесс создания визуального представления трехмерного пространства»

На рисунке 2 представлена модель BPMN бизнес-процесса «Процесс создания визуального представления трехмерного пространства». Как видно на рисунке 2, большая часть бизнес-процесса выполняется без использования автоматизированных средств, а потому затрачиваемое время на исполнение данного бизнес-процесса остается высоким, в связи с его большой трудоемкостью и необходимостью повторять отдельные процедуры бизнес-процесса. Например, при появлении потребности разработки нового 3D интерьера, программисту необходимо вновь разворачивать и настраивать проект, а также разрабатывать базовую программную среду. Проблематика этого сглаживается возможностью скопировать настройки из предыдущего проекта, однако это не всегда представляется возможным, так как задача может быть поручена разным программистам, а код предыдущих проектов может быть утерян, удален, либо обладать несовместимой версией.

### **1.3.2 Модель бизнес-процесса ТО-ВЕ**

Построение модели бизнес-процесса ТО-ВЕ формирует состояние процесса, к которому необходимо стремиться [28]. На рисунке 3 представлена модель ТО-ВЕ бизнес-процесса «Процесс создания визуального представления трехмерного пространства».

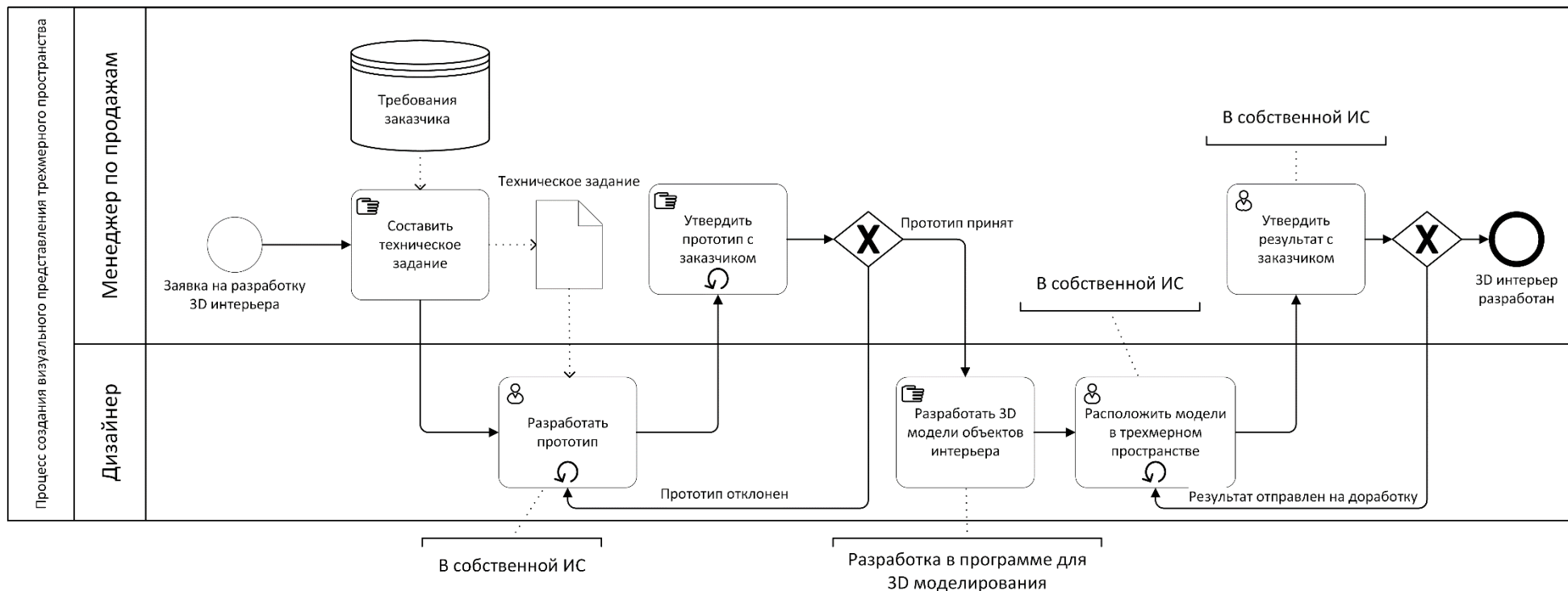


Рисунок 3 – Модель BPMN TO-VE бизнес-процесса «Процесс создания визуального представления трехмерного пространства»



Как видно из рисунка 3, в структуре бизнес-процесса «Процесс создания визуального представления трехмерного пространства» произошли значительные изменения. А именно добавлены новые процедуры для swimлейна (от англ. swimlane (прямоугольная область, отражающая участника бизнес-процесса)) «Дизайнер» и «Менеджер по продажам», а также полностью удален swimлейн «Программист». Теперь задачи «Разработать прототип» и «Расположить модели в трехмерном пространстве» будут выполняться дизайнером в специальной информационной системе, наравне с демонстрацией готового проекта заказчику, которую, стоит обратить внимание, будет выполнять менеджер по продажам.

Важными отличиями модели TO BE на рисунке 3 от модели AS-IS на рисунке 2 являются:

- уменьшение количества привлекаемых в бизнес-процесс сотрудников, так как программист более не участвует в бизнес-процессе;
- уменьшение общей трудоемкости выполнения бизнес-процесса, за счет внедрения информационной системы;
- рост качества выполнения бизнес-процесса, за счет дополнительного утверждения результата работ с заказчиком.

Задача выполнения описанных выше процедур бизнес-процесса «Процесс создания визуального представления трехмерного пространства» будет возложена на информационную систему, описание требований к которой приведены в последующих пунктах данной работы.

#### **1.4 Выявление недостатков**

Очевидно, что бизнес-процесс, выполняющийся часто, требует автоматизации. Учитывая еще и тот факт, что трудоемкость данного бизнес-процесса высокая, а компании требуется выполнять его чаще, необходимо провести его оптимизацию и реинжиниринг [21]. Данный реинжиниринг

решено проводить при помощи внедрения информационной системы, которая будет автоматизировано выполнять ручные задачи, представленные на рисунке 2, а именно задачи «Создать новый проект», «Настроить проект» и «Разработать программную часть 3D интерьера». Также информационная система позволит исключить задачу «Провести пуско-наладочные работы» и одновременно с этим повысит качество конечного продукта (интерьера), так как состояние работы по дизайну интерьера будет доступно через сеть Интернет.

Для решения обозначенной задачи прямого аналога найдено не было, но существуют системы, например, Unity и Unreal Engine, в задачи которых входит моделирование трехмерных интерьеров, экстерьеров и миров. Однако, использование вышеупомянутых программ влечет за собой потребность в компиляции результатов работы в них и, соответственно, установки определенного дистрибутива на персональном компьютере конечного потребителя [1].

В связи с тем, что для потребностей компании необходимо отображать 3D модели и интерьер прямо в браузере через сеть Интернет, без предустановки дополнительных сторонних программ и компонентов, необходимо реализовать собственную информационную систему. Перейдем к описанию функциональных требований проекта.

## **1.5 Определение бизнес-требований**

Бизнес-требования используются для определения рамок проекта.

В таблице 1 определена концепция проекта в соответствии с классом задач, к которым будет относиться реализуемый проект с точки зрения функций управления [7].

Таблица 1 – Концепция проекта

Концепция проекта	
Название проекта	Система виртуализации 3D интерьера «Interior Contractor».
Инициаторы проекта	Отдел разработки.
Содержание проекта	
Обоснование начала реализации проекта	Выявлена потребность в автоматизации действующих бизнес-процессов отдела.
Цель проекта	Добиться автоматизации бизнес-процессов отдела с помощью создания информационной системы, благодаря чему высвободить человеко-часы (сотрудников) на другие приоритетные задачи.
Результат проекта	Система автоматизации бизнес-процессов, соответствующая техническому заданию.
Продукт проекта	WEB-приложение для клиентского и административного взаимодействия, позволяющего реализовывать и просматривать 3D интерьеры.
Участники проекта	Отдел разработки.
Основные потребности (ожидания) участников	Реализованная система, соответствующая техническому заданию и позволяющая добиться автоматизации действующих бизнес-процессов отдела.

С точки зрения функций управления реализуемый проект будет относиться к классу задач связанным с планированием [18]. Функция планирования также включает в себя определение (уточнение) целей проекта в процессе его разработки. Основой для планирования является разделение общей деятельности по реализации проекта на фазы, этапы и стадии работ, которые должны быть выполнены в течение жизненного цикла проекта [2]. Спецификация отдельных видов деятельности основывается на конкретном характере проекта (тип, вид и цель).

IT-решение должно соответствовать бизнес-требованиям, в частности обеспечивать возможности разработки и сохранения 3D интерьера, а также просмотра сторонних 3D интерьеров.

IT-решение должно соответствовать требованию федерального закона №152-ФЗ «О персональных данных» при хранении и обработке персональных данных пользователей [22].

Система также должна соответствовать требованию масштабируемости. Масштабируемой называется система, которая может улучшать свою производительность пропорционально добавляемым ресурсам [1]. Данное требование является важным, так как в дальнейшем компания планирует расширение.

### **1.6 Требования к разрабатываемой информационной системе виртуализации 3D интерьеров «Interior Constructor»**

Составим требования к внедряемой информационной системе, призванной автоматизировать бизнес-процессы, требующие реинжиниринга [8]. Для этого воспользуемся технологией FURPS+, представляющую собой классификационную модель функциональных и нефункциональных требований к программному обеспечению [4].

Составим список соответствующих требований к информационной системе по технологии FURPS+ [3]:

- а) *functionality*, функциональность:
  - 1) инструмент взаимодействия с трехмерными объектами,
  - 2) возможность сохранения результатов работы;
- б) *usability*, удобство использования:
  - 1) интерфейс веб-приложения в минималистичном стиле,
  - 2) документация по основному функционалу;
- в) *reliability*, надежность:
  - 1) покрытие важных элементов системы автоматизированными тестами [5];
- г) *performance*, производительность:
  - 1) кеширование данных,

- 2) загрузка веб-приложения менее чем за 3 секунды;
- д) supportability, поддерживаемость:
  - 1) логирование внутренних процессов системы;
- е) проектные ограничения:
  - 1) реализация программного продукта должна выполняться на языке TypeScript версии 5.0,
  - 2) в качестве базы данных использовать MongoDB версии 6.0,
  - 3) в качестве инструмента 3D визуализации использовать JavaScript движок BabylonJS версии 5.0.

#### Выводы по первой главе

В первой главе была представлена общая характеристика компании ООО «IT», описаны основные направления её деятельности и составлена организационная структура. Было определено, что моделирование и анализ бизнес-процессов компании является действенным инструментом для оптимизации деятельности компании, а методология BPMN позволяет показать модели бизнес-процесса от начала до его завершения. Выполняемый часто бизнес-процесс «Процесс создания визуального представления трехмерного пространства» требует автоматизации, что наглядно показали блок-схемы моделей AS-IS и TO-BE, которые отражают полную последовательность операций и информационных потоков.

## **Глава 2 Проектирование информационной системы «Interior Constructor»**

### **2.1 Выбор технологии проектирования**

После этапа анализа и моделирования бизнес-процессов следует перейти к проектированию, а также рассмотреть особенности разработки информационной системы. Прежде уже отмечалось, что существует множество методологий и нотаций, которые применяются в целях проектирования. Поскольку в сформулированных требованиях к информационной системе присутствует указание на применение языков программирования, которые поддерживают ООП (объектно-ориентированное программирование), то целесообразно рассмотреть возможность применения комплементарной методологии [10].

Унифицированный язык моделирования UML является языком широкого профиля, подходит для описания объектного моделирования в области разработки программного обеспечения и используется для моделирования и системного проектирования [25]. UML позволяет осуществлять работы по проектированию и документированию на протяжении всего цикла разработки, обеспечивая возможность построения графиков и диаграмм. В отличие от подходов, которые основаны на функциональной декомпозиции анализируемого процесса, в объектно-ориентированном подходе система рассматривается как множество классов и объектов, которые в свою очередь обладают определенными свойствами и поведением. Все это без исключения повышает продуктивность разработки, сокращает затрачиваемое время, а также улучшает качество проектных решений. Данные принципы и особенности позволяют UML оставаться одним из наиболее перспективных подходов в проектировании и разработке информационной системы [25].

Опираясь на собранную информацию, можно сделать вывод о том, что применение языка моделирования UML в процессе проектирования информационной системы является целесообразным и обоснованным решением.

Диаграммы, используемые в UML, можно разделить на две категории: статические и динамические [29].

Статические диаграммы определяют константные сущности и их взаимосвязи. К статическим относятся диаграммы классов, объектов, компонентов и развертывания. Данные диаграммы подчеркивают статическую структуру систем, использующих объекты, атрибуты, операции и отношения [29].

Динамические диаграммы представляют протекающие процессы. К динамическим относятся диаграммы деятельности, последовательности, взаимодействия и состояний. Данные диаграммы показывают взаимодействие между объектами и изменения внутренних состояний объектов [29].

## **2.2 Варианты использования**

Диаграмма вариантов использования применяется для представления внешних, по отношению к системе, пользователей (акторов), а также доступный функционал, который реализует разрабатываемая система и их связи. Действующие лица иллюстрируются в виде силуэтов человека, а функционал системы рассматривается как множество прецедентов и изображается эллипсами [15].

Диаграмма вариантов использования показана на рисунке 4.

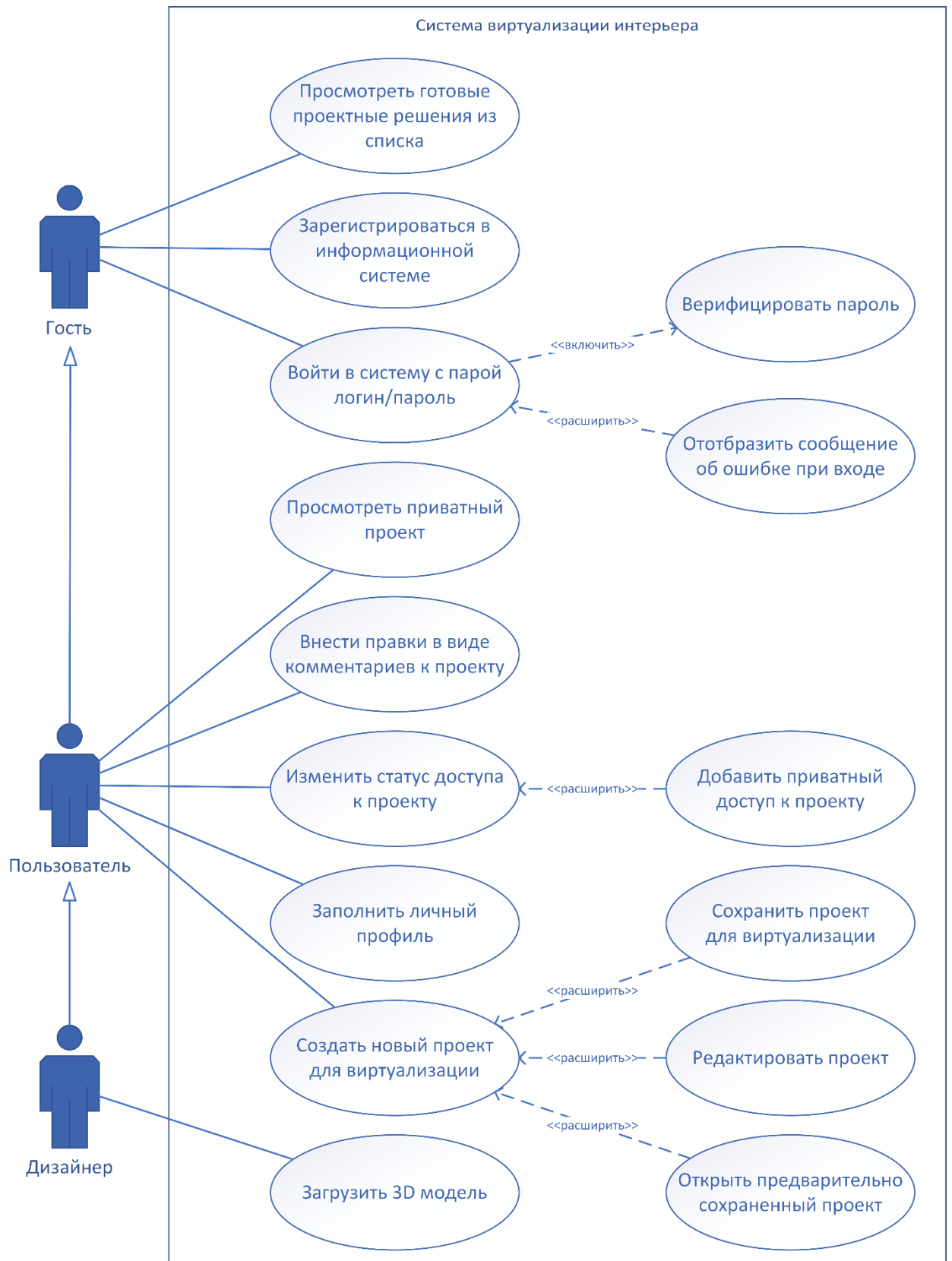


Рисунок 4 – UML диаграмма вариантов использования



В диаграмме на рисунке 4 следует обратить внимание на стрелки, указывающие обобщение акторов, взаимодействующих с системой. Например, «Гость», то есть незарегистрированный пользователь системы, имеет несколько сценариев взаимодействия, а именно «Посмотреть готовые проектные решения из списка», «Зарегистрироваться в информационной системе» и «Войти в систему с парой логин/пароль». Более обширными вариантами взаимодействия с системой обладает «Пользователь», то есть зарегистрированный участник системы, а наибольший доступ к системе у «Дизайнера», так как помимо всех прочих сценариев данная роль обладает возможностью использовать прецедент «Загрузить 3D модель».

Для уточнения подробностей взаимодействия по диаграмме вариантов использования, представленной на рисунке 4, применяются сценарии прецедентов. Описание прецедента «Посмотреть готовые проектные решения из списка» приведено в таблице 2.

Таблица 2 – Описание прецедента «Посмотреть готовые проектные решения из списка»

Вариант использования «Посмотреть готовые проектные решения из списка»	
Краткое описание	Позволяет пользователю системы просмотреть список всех имеющихся дизайн-проектов, которые находятся в открытом доступе.
Действующие лица	Любой пользователь системы.
Основной поток	<ul style="list-style-type: none"> <li>– начинается при нажатии пользователя на кнопку «Все интерьеры»;</li> <li>– система загружает и отображает на экране список дизайн-проектов, их название и описание;</li> <li>– пользователь системы может просматривать дизайн-проекты, используя постраничную навигацию (пагинацию);</li> <li>– вариант использования завершается.</li> </ul>

Описание прецедента «Зарегистрироваться в информационной системе» приведено в таблице 3.

Таблица 3 – Описание прецедента «Зарегистрироваться в информационной системе»

Вариант использования «Зарегистрироваться в информационной системе»	
Краткое описание	Позволяет незарегистрированному пользователю системы получить расширенный доступ к функционалу, доступному только после регистрации.
Действующие лица	Гость
Основной поток	<ul style="list-style-type: none"> <li>– начинается при переходе актора на страницу регистрации;</li> <li>– актор заполняет личные данные (логин и пароль);</li> <li>– актор нажимает кнопку «зарегистрироваться»;</li> <li>– система оповещает об успешной регистрации;</li> <li>– вариант использования завершается.</li> </ul>
Альтернативный поток А1: оповестить о существовании логина	<ul style="list-style-type: none"> <li>– если указанный актором логин уже существует в базе данных, то система оповещает об этом;</li> <li>– актор меняет логин;</li> <li>– актор нажимает кнопку «зарегистрироваться»;</li> <li>– система оповещает об успешной регистрации;</li> <li>– вариант использования завершается.</li> </ul>
Постусловия	Если поток выполнен успешно, то данные пользователя сохраняются в базе данных.

Описание прецедента «Войти в систему с парой логин/пароль» приведено в таблице 4.

Таблица 4 – Описание прецедента «Войти в систему с парой логин/пароль»

Вариант использования «Войти в систему с парой логин/пароль»	
Краткое описание	Позволяет зарегистрированному пользователю системы авторизоваться в информационной системе и получить доступ к расширенному функционалу.
Действующие лица	Гость
Основной поток	<ul style="list-style-type: none"> <li>– начинается при переходе актора на страницу авторизации;</li> <li>– актор заполняет личные данные (логин и пароль);</li> <li>– актор нажимает кнопку «войти»;</li> <li>– система оповещает об успешной входе;</li> <li>– вариант использования завершается.</li> </ul>

Продолжение таблицы 4

Вариант использования «Войти в систему с парой логин/пароль»	
Альтернативный поток А1: оповестить о неверной паре логин/пароль	<ul style="list-style-type: none"> <li>– если указанная актором пара логин/пароль неверна, то система оповещает об этом;</li> <li>– актор меняет пару логин/пароль;</li> <li>– вариант использования завершается.</li> </ul>
Постусловия	Если поток выполнен успешно, то для пользователя создается и присваивается уникальный идентификатор (токен).

Описание прецедента «Просмотреть приватный проект» приведено в таблице 5.

Таблица 5 – Описание прецедента «Просмотреть приватный проект»

Вариант использования «Просмотреть приватный проект»	
Краткое описание	Позволяет пользователю просмотреть приватный дизайн-проект, по предоставленной ему ссылке-доступу и коду доступа.
Действующие лица	Пользователь
Предусловия	Необходима ссылка-доступ и специальный код доступа.
Основной поток	<ul style="list-style-type: none"> <li>– начинается при переходе актора на страницу приватного дизайн-проекта по ссылке-доступу;</li> <li>– актор вводит предоставленный ему код доступа в поле ввода;</li> <li>– система оповещает об успешном входе в приватный дизайн-проект;</li> <li>– вариант использования завершается.</li> </ul>
Альтернативный поток А1: оповестить о неверном вводе кода доступа	<ul style="list-style-type: none"> <li>– если указанный актором код доступа является неверным, система выдает сообщение об ошибке;</li> <li>– вариант использования завершается.</li> </ul>

Описание прецедента «Внести правки в виде комментариев к проекту» приведено в таблице 6.

Таблица 6 – Описание прецедента «Внести правки в виде комментариев к проекту»

Вариант использования «Внести правки в виде комментариев к проекту»	
Краткое описание	Позволяет пользователю добавить правки, в частности комментарии, к дизайн-проекту.
Действующие лица	Пользователь
Предусловия	Предварительно подключиться к приватному проекту.
Основной поток	<ul style="list-style-type: none"> <li>– начинается при нажатии актором кнопки «добавить комментарий» на странице приватного дизайн-проекта;</li> <li>– актер вводит любой комментарий;</li> <li>– актер нажимает кнопку «добавить»;</li> <li>– система оповещает об успешном сохранении комментария;</li> <li>– вариант использования завершается.</li> </ul>
Альтернативный поток A1: отменить комментарий	<ul style="list-style-type: none"> <li>– если актер нажимает кнопку «отмена», комментарий не будет сохранен;</li> <li>– вариант использования завершается.</li> </ul>
Постусловия	Если поток выполнен успешно, то комментарии пользователя сохраняются в базе данных.

Описание прецедента «Заполнить личный профиль» приведено соответственно в таблице 7.

Таблица 7 – Описание прецедента «Заполнить личный профиль»

Вариант использования «Заполнить личный профиль»	
Краткое описание	Позволяет актору изменить описание своего личного профиля, указать данные о себе.
Действующие лица	Пользователь
Предусловия	Необходима регистрация в системе.
Основной поток	<ul style="list-style-type: none"> <li>– начинается при нажатии актором кнопки «личный кабинет» в меню информационной системы;</li> <li>– актер заполняет любые поля в предложенной ему форме на свой выбор;</li> <li>– актер нажимает кнопку «сохранить»;</li> <li>– система оповещает об успешном сохранении обновленных данных личного профиля;</li> <li>– вариант использования завершается.</li> </ul>
Постусловия	Если поток выполнен успешно, то данные пользователя сохраняются в базе данных.

Описание прецедента «Изменить статус доступа к проекту» приведено в таблице 8.

Таблица 8 – Описание прецедента «Изменить статус доступа к проекту»

Вариант использования «Изменить статус доступа к проекту»	
Краткое описание	Позволяет актору изменить статус доступа к проекту, перевести проект из статуса «черновик» в статус «опубликовано», а также сделать проект «приватным», то есть доступным только по специальной ссылке-доступу с кодом доступа.
Действующие лица	Пользователь
Предусловия	Предварительно создать проект, либо загрузить свой прежде сохраненный проект.
Основной поток	<ul style="list-style-type: none"> <li>– начинается при нажатии актором кнопки «Изменить статус» на странице редактирования дизайн-проекта;</li> <li>– актер выбирает один из пунктов списка: «черновик», «опубликовать», «сделать приватным»;</li> <li>– система оповещает об успешном сохранении статуса проекта;</li> <li>– вариант использования завершается.</li> </ul>
Альтернативный поток А1	<ul style="list-style-type: none"> <li>– если актер не выбирает один из предложенных пунктов в списке, то проект не изменяет статус своего доступа;</li> <li>– вариант использования завершается.</li> </ul>
Постусловия	Текущий статус проекта сохраняется в базе данных.

Описание прецедента «Создать новый проект для виртуализации» приведено в таблице 9.

Таблица 9 – Описание прецедента «Создать новый проект для виртуализации»

Вариант использования «Создать новый проект для виртуализации»	
Краткое описание	Позволяет актору создать новые дизайн-проект, который в дальнейшем можно сохранить, редактировать и загружать.
Действующие лица	Пользователь
Предусловия	Необходима регистрация в системе.
Основной поток	<ul style="list-style-type: none"> <li>– начинается при нажатии актором кнопки «создать новые дизайн» на главной странице информационной системы;</li> <li>– актер заполняет поле ввода «название проекта»;</li> <li>– система оповещает об успешном создании нового проекта;</li> <li>– вариант использования завершается.</li> </ul>
Постусловия	Новый проект сохраняется в базе данных

Описание прецедента «Загрузить 3D модель» приведено в таблице 10.

Таблица 10 – Описание прецедента «Загрузить 3D модель»

Вариант использования «Загрузить 3D модель»	
Краткое описание	Позволяет актору загрузить в систему 3D модель в формате .glb
Действующие лица	Дизайнер
Предусловия	Необходима регистрация в системе.
Основной поток	<ul style="list-style-type: none"><li>– начинается при нажатии актором кнопки «загрузить модель» на главной странице информационной системы;</li><li>– актер заполняет поле ввода «название модели»;</li><li>– актер нажимает кнопку «прикрепить модель» и в открывающемся окне выбирает путь к модели у себя на компьютере;</li><li>– актер нажимает кнопку «загрузить»;</li><li>– система оповещает об успешной загрузке модели;</li><li>– вариант использования завершается.</li></ul>
Альтернативный поток A1: отобразить ошибку расширения файла	<ul style="list-style-type: none"><li>– если актер указывает путь к файлу расширение которого не соответствует требуемым для загрузки расширениям, то система выдает ошибку;</li><li>– вариант использования завершается.</li></ul>
Постусловия	Новая модель сохраняется на файловом сервере, а информация о ней в базе данных.

Приведенные выше таблицы прецедентов подробно описывают различные варианты использования системы. На основании данных таблиц в дальнейшем будет происходить разработка информационной системы.

### 2.3 Диаграмма классов системы виртуализации интерьера

Беря за основу полученную концептуальную модель предметной области, можно построить диаграмму классов. Диаграмма классов является структурной диаграммой, которая демонстрирует общую структуру иерархии классов системы, их коопераций, атрибутов, методов, интерфейсов, а также включает информацию об их взаимосвязях.

Класс представляет собой ключевой элемент, и олицетворяет сущность, которая содержит данные и обладает поведением. На диаграмме классы

обозначаются рамкой, в которой задается имя класса, перечень его свойств и методов.

При построении диаграммы классов применялись следующие типы связей: ассоциация, генерализация, реализация и композиция. Связь по ассоциации демонстрирует взаимодействие двух классов. Связь по генерализации означает, что дочерний класс наследует свойства, состояние и поведение родительского класса. Связь по реализации показывает отношение между двумя элементами модели, в которой один элемент реализует поведение, заданное другим. Связь по композиции предполагает отношения вида целое-часть с тем условием, что часть не может существовать отдельно от целого.

### 2.3.1 Прототип диаграммы классов

С целью обобщенного понимания системы был составлен прототип диаграммы классов, который отражает общую структуру системы без излишней детализации. Данная диаграмма представлена на рисунке 5.

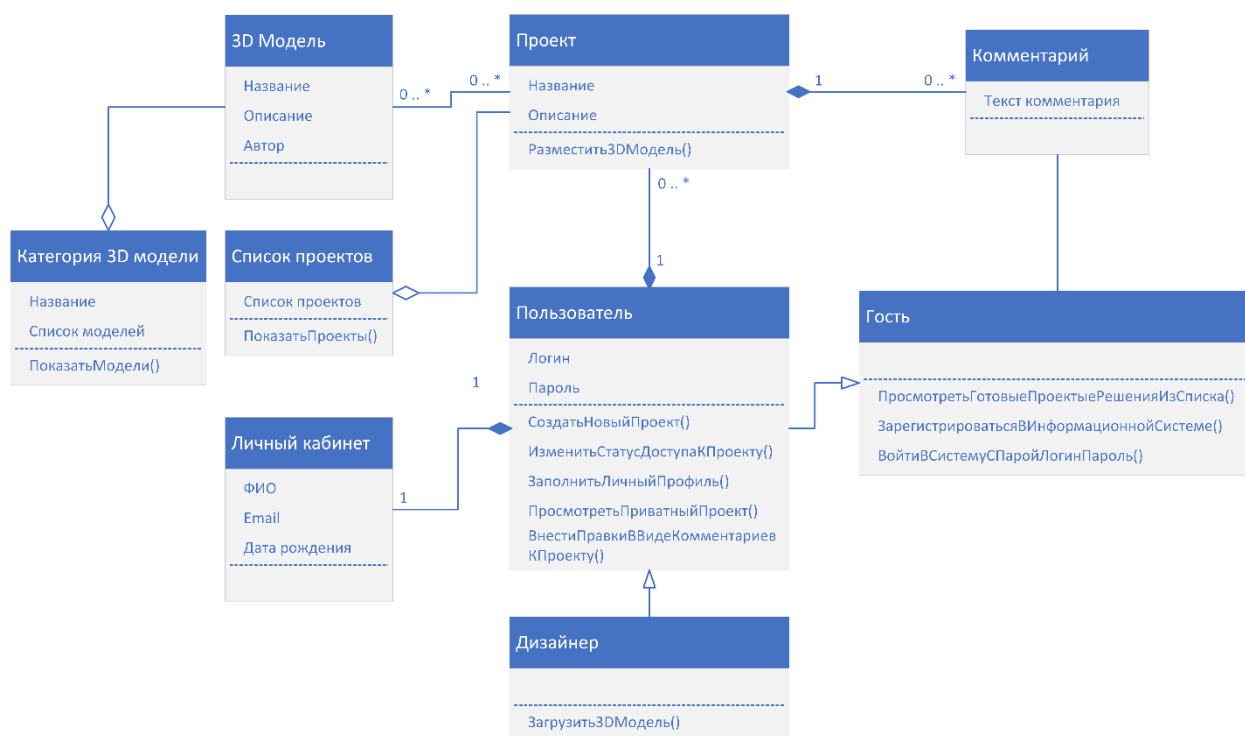


Рисунок 5 – Прототип диаграммы классов системы виртуализации интерьера

Модель на рисунке 5 характеризует систему в первичном представлении. На ней обозначены несколько агрегаций, а именно «Список проектов» и «Категория 3D модели», идущих от сущности «Проект» и «3D модель» соответственно. Также на данной модели показана генерализация сущностей «Гость», «Пользователь», и «Дизайнер», а именно разграничение их функциональных возможностей.

Связью типа композиция соединены сущности «Личный кабинет» и «Пользователь», а также «Комментарий» и «Проект» соответственно, так как данные сущности («Личный кабинет» и «Комментарий») не могут существовать без сущностей, на которые указывает композиционный тип связи. Например, при отсутствии сущности «Пользователь» будет отсутствовать и сущность «Личный кабинет».

### **2.3.2 Детализированная диаграмма классов**

Описание модели, представленной на рисунке 5, позволяет лучше оценить проектируемую систему и построить на её основании детализированную диаграмму классов системы.

В детализированной диаграмме классов будут описаны не только сущности системы, но и ее атрибуты, а также типы данных этих атрибутов, типы данных аргументов методов сущностей и типы данных возвращаемых значений этих методов. Детализированная диаграмма классов системы виртуализации интерьера представлена на рисунке 6.



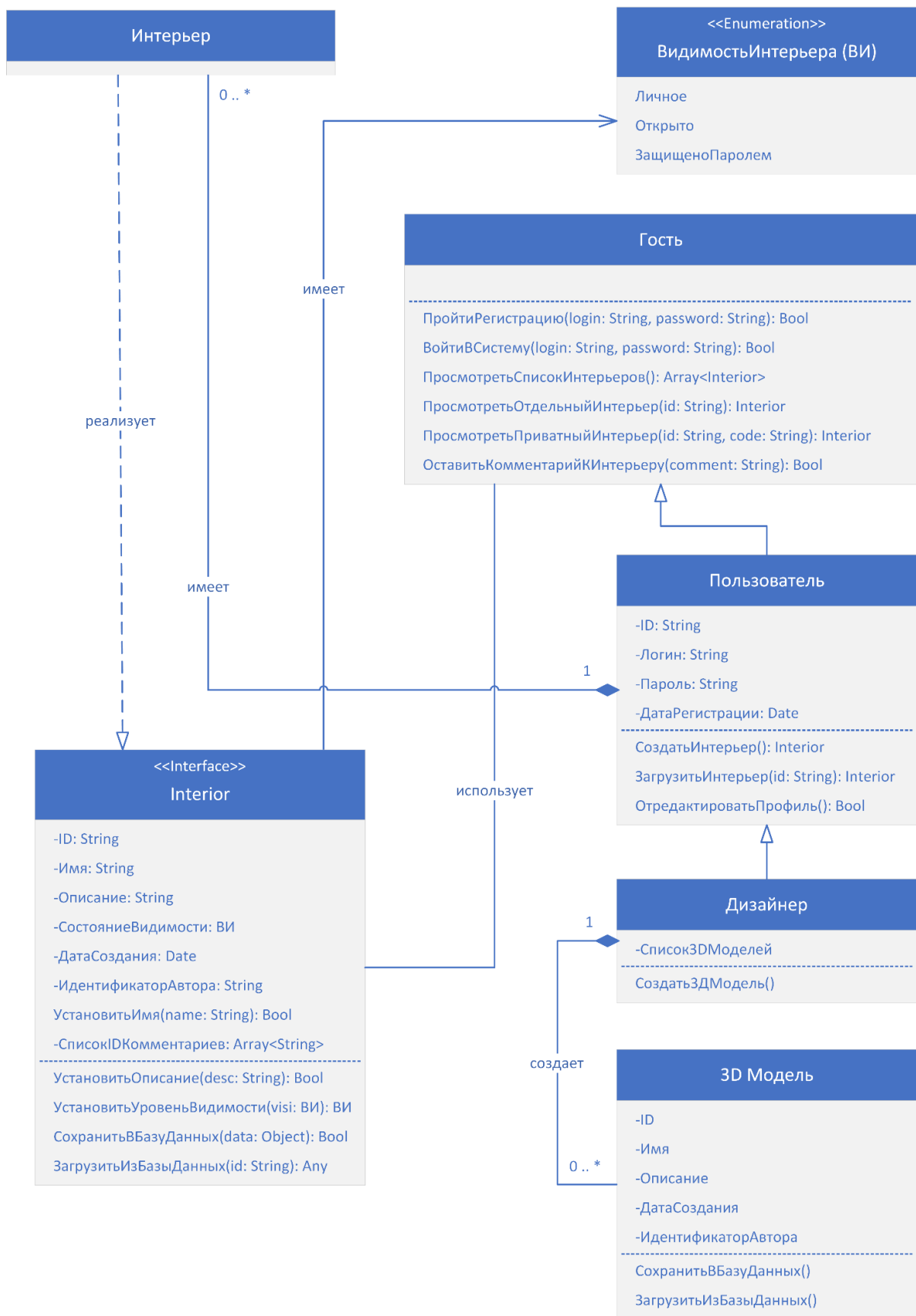


Рисунок 6 – Детализированная диаграмма классов

На диаграмме классов системы виртуализации интерьера, представленной на рисунке 6, класс «Дизайнер» наследуется от класса «Пользователь», а класс «Пользователь» наследуется от класса «Гость». Функционал системы, направленный на взаимодействие с частными интерьерами, должен быть реализован в классе «Гость», что и видно на рисунке 6. Также на диаграмме представлен интерфейс «Interior» и класс «Интерьер», который реализует данный интерфейс. Потребность разбить класс и интерфейс на два разных блока возникла при моделировании возвращаемых типов у функций классов «Гость» и «Пользователь». Интерфейс «Interior» также использует внутри себя перечисляемый тип данных (от англ. enumerated type), а именно «Видимость интерьера».

Класс «Пользователь» связан композиционной связью с классом «Интерьер», что означает невозможность существования класса «Интерьер» без связи с конкретным классом «Пользователь». Мультипликатор связи «ноль ко многим» показывает, что один экземпляр класса «Пользователь» (или его потомок) может иметь множество экземпляров класса «Интерьер». Все тоже самое верно и для класса «Дизайнер» и «3D Модель» соответственно.

Несмотря на достаточную полноту описания и именования функций, аргументов и их возвращаемых типов данных, диаграмма классов на рисунке 6 может быть изменена в дальнейшем.

## **2.4 Концептуальная модель данных**

Поскольку проектируемая система хранит, обрабатывает и обеспечивает доступ к данным, то одной из важных задач является разработка концептуальной модели данных. Концептуальная модель данных, включающая в себя все основные сущности и связи, представлена, соответственно, на рисунке 7.

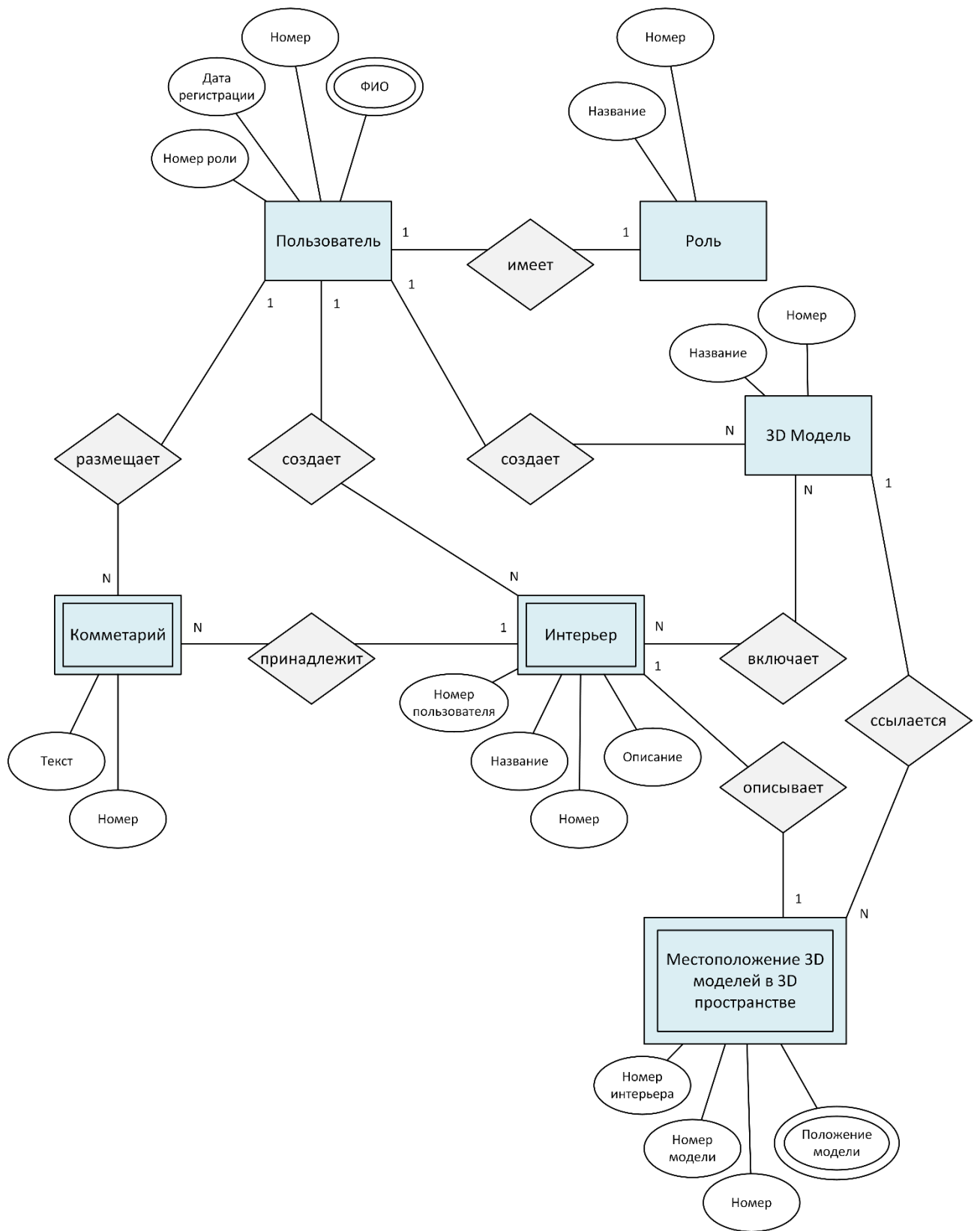


Рисунок 7 – Концептуальная модель данных системы виртуализации интерьера

Диаграмма на рисунке 7 составлена с применением нотации моделирования баз данных П. Чена. В прямоугольниках описаны сущности проектируемой базы данных, в овалах – атрибуты соответствующей сущности, в ромбах – отношения между сущностями. Отношения принято именовать глаголом [11].

В данной диаграмме также присутствуют слабые сущности, выделенные двойным прямоугольником, а именно сущности «Комментарий», «Интерьер» и «Местоположение 3D моделей в 3D пространстве». Слабые сущности называются таковыми, потому как не могут существовать без связанной с ними сильной сущности. Например, комментарий или интерьер не могут существовать без пользователя. Напротив, роль внутри базы данных, разграничивающая доступ к данным, может существовать независимо от наличия или отсутствия пользователя этой системы.

В концептуальной модели данных на рисунке 7 следует обратить внимание на сущность «Местоположение 3D моделей в 3D пространстве» и, в частности, на её атрибут «Положение модели». Данный атрибут является составным, то есть на самом деле имеет множество атрибутов внутри себя. На этапе проектирования базы данных использование составных атрибутов разрешено и оправдано, однако в дальнейшем данные атрибуты требуется исключить из SQL-ориентированных баз данных и разбить на атомарные атрибуты.

## **2.5 Логическая модель данных**

Исходя из проведенного исследования, а именно составления концептуальной модели данных, представленной на рисунке 7, можно описать логическую модель данных. Данная модель – это расширение концептуальной модели данных. Она включает в себя все сущности, атрибуты, ключи и взаимосвязи, которые представляют бизнес-информацию

и определяют бизнес-правила [27]. Логическая модель данных системы виртуализации интерьера представлена на рисунке 8.

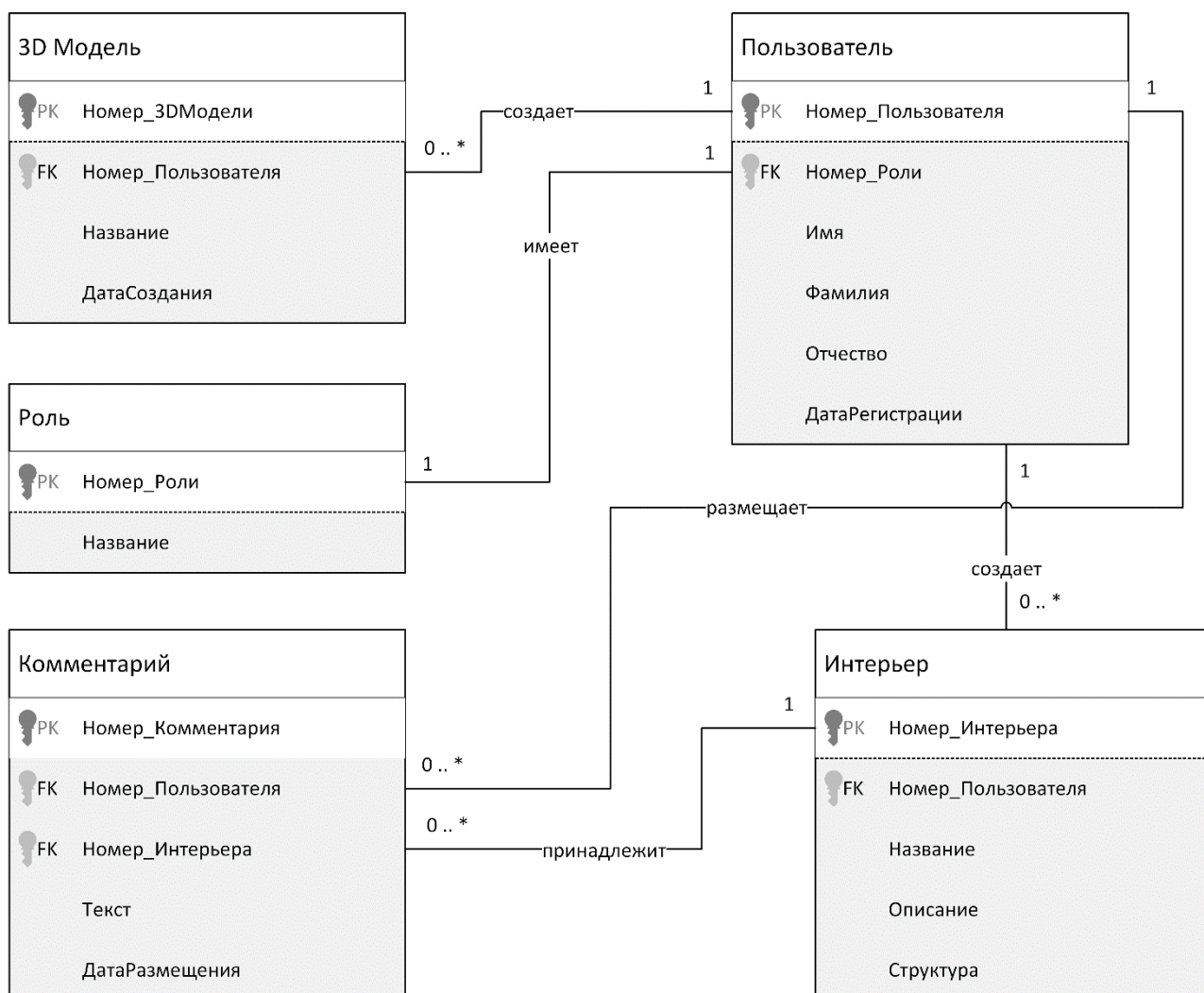


Рисунок 8 – Логическая модель данных системы виртуализации интерьера

На диаграмме, представленной на рисунке 8, можно увидеть все необходимые первичные и вторичные ключи у таблиц.

Проведя анализ предметной области, с целью того, чтобы построить наиболее объективную логическую модель базы данных, было выяснено, что данные интерьера пользователя не обладают строгой структурой. Это связано с тем, что нельзя заранее сказать какой именно интерьер пожелает создать пользователь, а равно нельзя предугадать все используемые им 3D модели и

материалы (текстурированное оформление 3D объектов). Более того, 3D объект всегда имеет большое количество собственных параметров, например, угол вращения вокруг одной из трех осей, положение в трехмерной системе координат, материалы, из которых состоит 3D модель и другое.

Однако, движок 3D графики BabylonJS обладает функционалом, позволяющим сериализовать контент сцены (в данном случае, интерьер пользователя), то есть перевести структуру данных в битовую последовательность, а затем выгрузить данные в формате JSON [24]. По этой причине, отдельного внимания заслуживает таблица «Интерьер», которая теперь обладает атрибутом «Структура». В данном атрибуте будет храниться информация о структуре всех объектов, расположенных на сцене. Данные этого атрибута будут представлены в формате JSON.

## 2.6 Требования к программно-аппаратному обеспечению

Согласно проектному решению, система размещается на трех узлах: клиентский узел, сервер приложений и сервер базы данных. В таблице 11 представлены требования к программно-аппаратному обеспечению.

Таблица 11 – Требования к программно-аппаратному обеспечению

Узел	Минимальные системные требования
Клиентская машина	Операционная система Windows 10 или более поздней версии либо Windows Server 2016 или более поздней версии, процессор Intel Pentium 4 или более поздней версии с поддержкой SSE3, WebGL-совместимое устройство, ОЗУ 4 Гб, HDD 60 Гб, браузер Firefox или Chrome или Edge, сетевая карта 10 Мбит.
Сервер приложений	Операционная система Windows Server 2012 либо Linux, процессор Intel Pentium 2.2 ГГц, ОЗУ 8 Гб, SSD: 80 Гб, сетевая карта 100 Мбит, фиксированный IPv4 адрес.
Сервер базы данных	Операционная система Windows Server 2012 или Linux, процессор Intel Pentium 2.2 ГГц, ОЗУ: 8 Гб, SDD: 150 Гб, сетевая карта 100 Мбит, фиксированный IPv4 адрес.

Как видно из таблицы 11, требования к системе на всех узлах не являются чрезмерно жесткими.

## 2.7 Прототипы дизайна системы

При проектировании системы полезным средством визуализации конечного продукта выступают прототипы пользовательского интерфейса. Прототипы системы разработаны с использованием средств графического онлайн-редактора Figma.

На рисунке 9 представлен прототип формы авторизации пользователя. На данном прототипе также предусмотрен случай, если пару логин и пароль пользователя не удалось найти в базе данных.

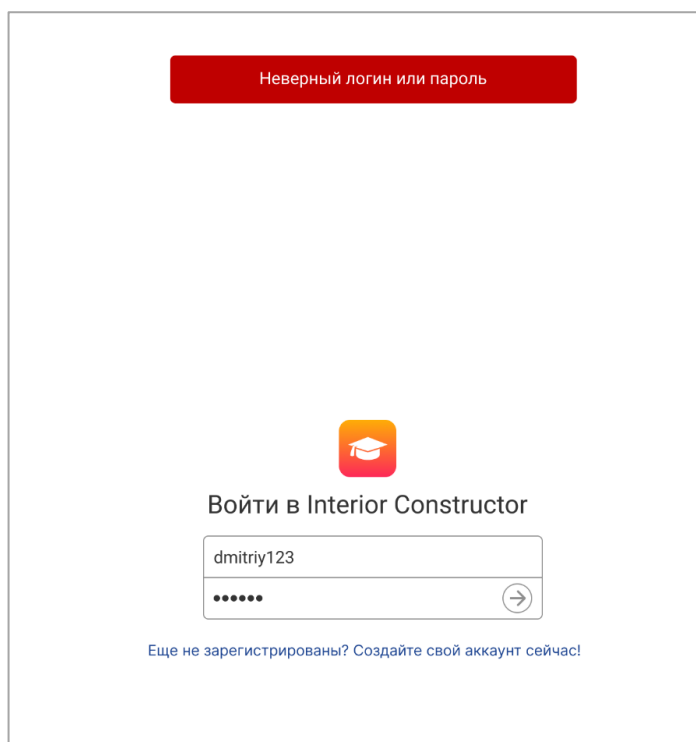


Рисунок 9 – Прототип экранной формы авторизации

На рисунке 10 представлен прототип главной страницы системы. Слева в навигационном меню представлены две ссылки, а именно «Интерьеры» (главная страница системы) и «Профиль» (личный кабинет пользователя).

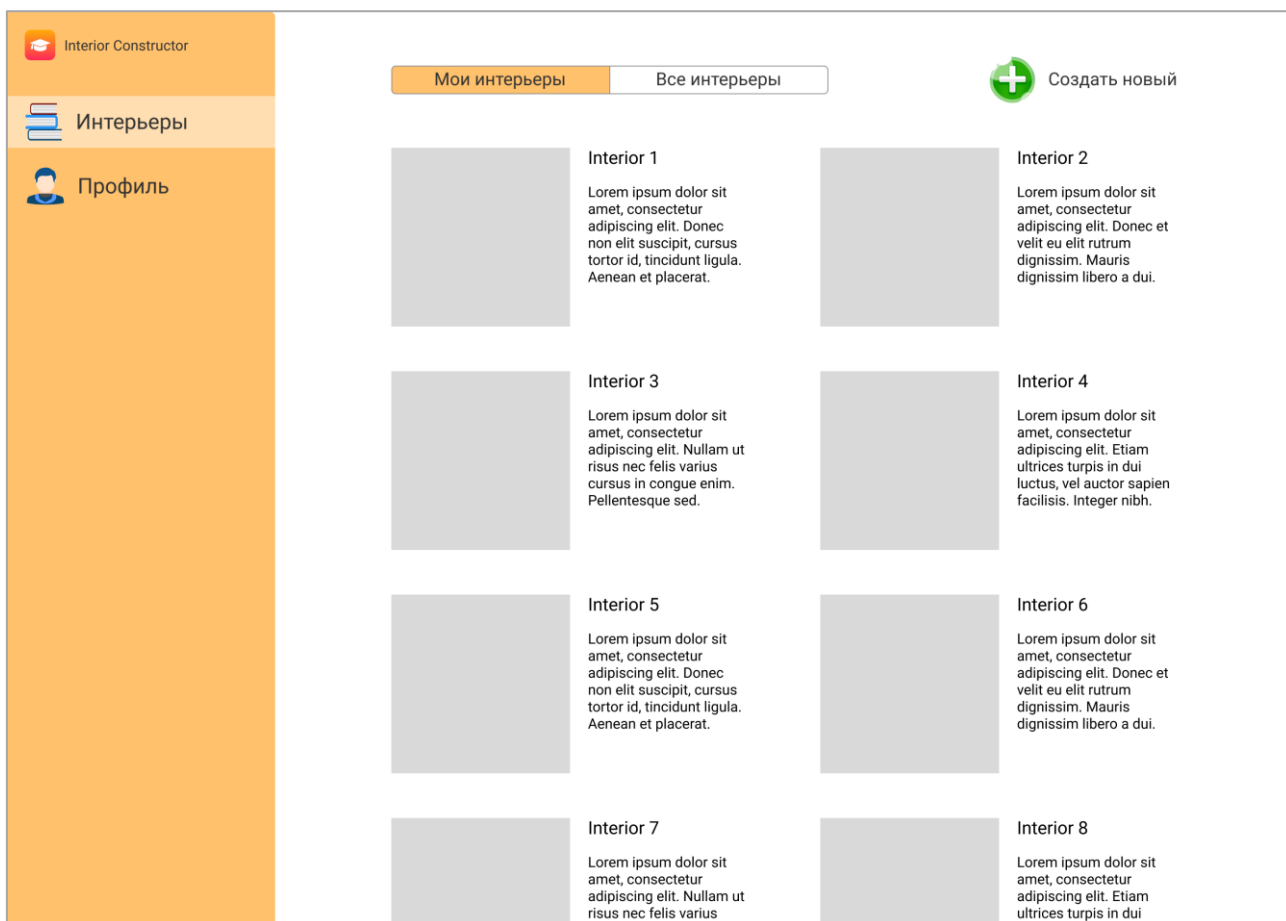


Рисунок 10 – Прототип страницы «Интерьеры»

Справа от навигационного меню в области основного контента представлены кнопки переключения между персональными интерьерами пользователя, а также интерьерами других участников системы, находящимися в опубликованном состоянии.

В правом верхнем углу рисунка 10 отображена кнопка «Создать новый» (интерьер), которая вызывает модальное окно, представленное на рисунке 11.



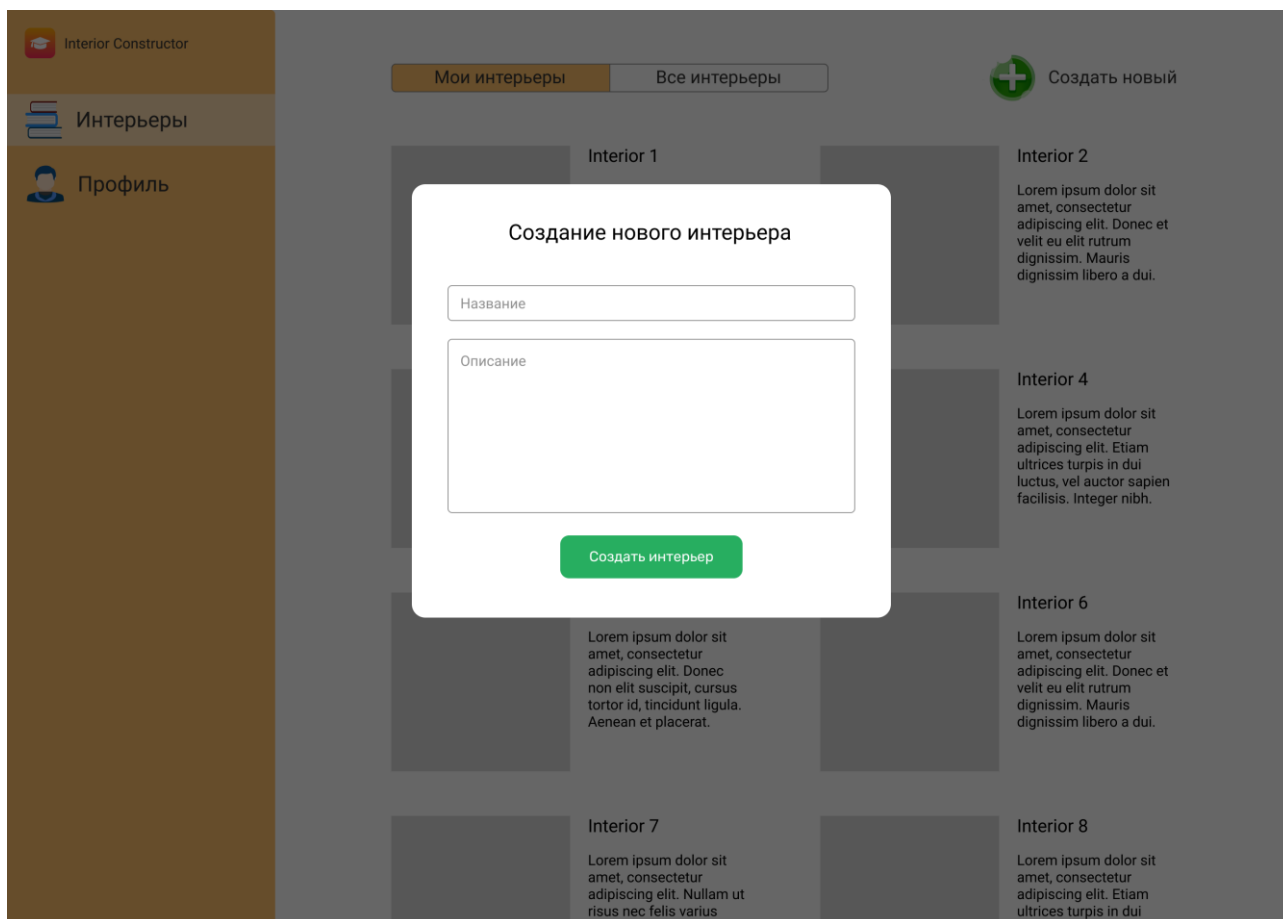


Рисунок 11 – Прототип модального окна «Создание нового интерьера»

Данное модальное окно позволяет создать новый интерьер пользователя. В нем присутствуют необходимые для этого процесса текстовые поля ввода данных, а именно «Название» и «Описание».

#### Выводы по второй главе

Использование UML позволило улучшить качество проектных решений, сократить затрачиваемое время и повысить продуктивность всей работы. Применение диаграммы вариантов использования, дополненной текстовым описанием прецедентов для представления функционала системы, позволило продемонстрировать ключевые аспекты работы проектируемой системы. С помощью диаграммы классов системы виртуализации интерьера были идентифицированы основные сущности предметной области и их взаимосвязи.

Концептуальная модель данных, выполненная в графической нотации П. Чена, включает в себя все основные сущности и связи. Логическая модель отразила внешние интерфейсы, а также обозначила границы функциональности компонентов системы. Для размещения компонентов системы применяется схема с тремя узлами, обеспечивающая надежность, отказоустойчивость и возможность дальнейшего масштабирования информационной системы, а учитывая приведенные требования к программно-аппаратному обеспечению система может быть успешно развернута в рабочей среде компании. При проектировании системы средством визуализации конечного продукта выступают прототипы пользовательского интерфейса. Прототипы системы разработаны с использованием средств графического онлайн-редактора Figma, который позволяет быстро разрабатывать дизайн в облаке.

## Глава 3 Разработка информационной системы виртуализации интерьера «Interior Constructor»

### 3.1 Выбор технологии разработки

Для выбора технологии разработки, а именно языка (языков) программирования, на котором будет осуществляться разработка системы виртуализации интерьера, требуется понять общую структуру размещения компонентов данной системы [17]. Для этих целей целесообразно построить структурную UML диаграмму развертывания, которая представлена соответственно на рисунке 12.

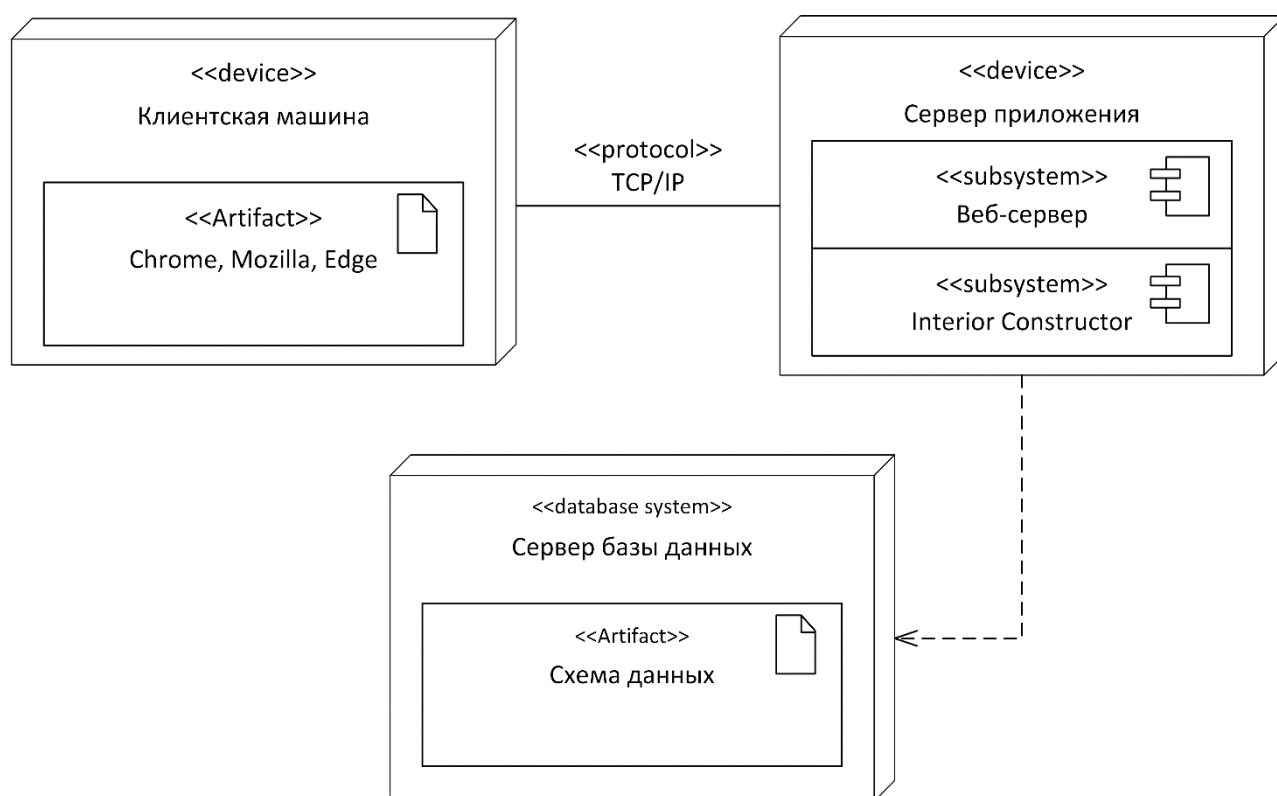


Рисунок 12 – Диаграмма развертывания

Система размещается в трех узлах, а именно в узле «Сервер приложения», состоящего из веб-сервера, который принимает запросы,

идущие от «Клиентской машины» используя протокол TCP/IP, и в узле «Сервер базы данных», в котором находится основная база данных приложения. Также внутри узла «Сервер приложения» находится и подсистема «Interior Constructor», то есть система виртуализации интерьера. Узел «Клиентская машина» отправляет запросы и получает ответы от веб-сервера, а также отображает подходящую информацию, включая HTML, CSS и JavaScript код.

Система виртуализации интерьера должна использовать 3D движок BabylonJS, который написан на TypeScript, что обязывает вести разработку клиентской части на JavaScript, либо TypeScript [24].

Заметим, что TypeScript является строго типизированным расширением языка JavaScript и предоставляет функционал, сравнимый с другими строго типизированными языками программирования, такими как C# или Java, а сам язык TypeScript, исходя из статистики за 2022 год, размещенной на GitHub.com, входит в список пяти самых популярных языков в мире. По этим причинам, рекомендуется выбрать язык TypeScript в качестве основного языка для разработки frontend-части приложения.

Для разработки серверной части приложения существует множество альтернатив, однако ссылаясь на популярность языков JavaScript и TypeScript, установленную необходимость вести разработку клиентской части приложения на языке TypeScript, а также стремление к консистентному ведению разработки, рекомендуется выбрать в качестве технологии реализации серверной части приложения стек Node.js.

Node.js – это программная платформа, в основе которой лежит движок V8 (позволяющий компилировать JavaScript в машинный код), превращающая JavaScript из языка узкой специализации в язык общего назначения.

### 3.2 Выбор СУБД

Выбор системы управления базой данных необходимо проводить с учетом выбранной ранее технологии разработки, принимая во внимание критерии масштабируемости, кроссплатформенности, поддержки средств проектирования и надежности [13]. Результаты сравнения по критериям, упомянутым выше, представлены в таблице 12.

Таблица 12 – Сравнение систем управления базами данных

Критерий	СУБД		
	MySQL	PostgreSQL	MongoDB
Кроссплатформенность	1	1	1
Бесплатно распространяемая	1	1	1
Поддержка и документация	1	1	1
Надежность	1	1	0.5
Масштабируемость	0.5	0.5	1
Быстродействие	1	0	1
Средства проектирования	1	1	1
Простота интеграции с Node.js	0.5	0.5	1
Итого	7	6	7.5

По результатам сравнения СУБД, приведенным в таблице 12, для разработки рекомендуется выбрать СУБД MongoDB.

СУБД MongoDB работает с документно-ориентированной моделью данных. В отличие от реляционных СУБД, MongoDB не требуются таблицы, схемы или отдельный язык запросов. Информация хранится в виде документов, либо коллекций.

Данная СУБД позиционируется как промежуточное звено между классическими СУБД и NoSQL решениями. MongoDB не использует схемы, как это происходит в реляционных базах данных, за счет чего повышается производительность данной СУБД.

### 3.3 Физическая модель данных

Физическая модель данных является модифицированной версией логической модели данных при построении которой учитываются особенности выбранной реализации СУБД. На рисунке 13 представлена диаграмма физической модели данных в нотации UML Database.

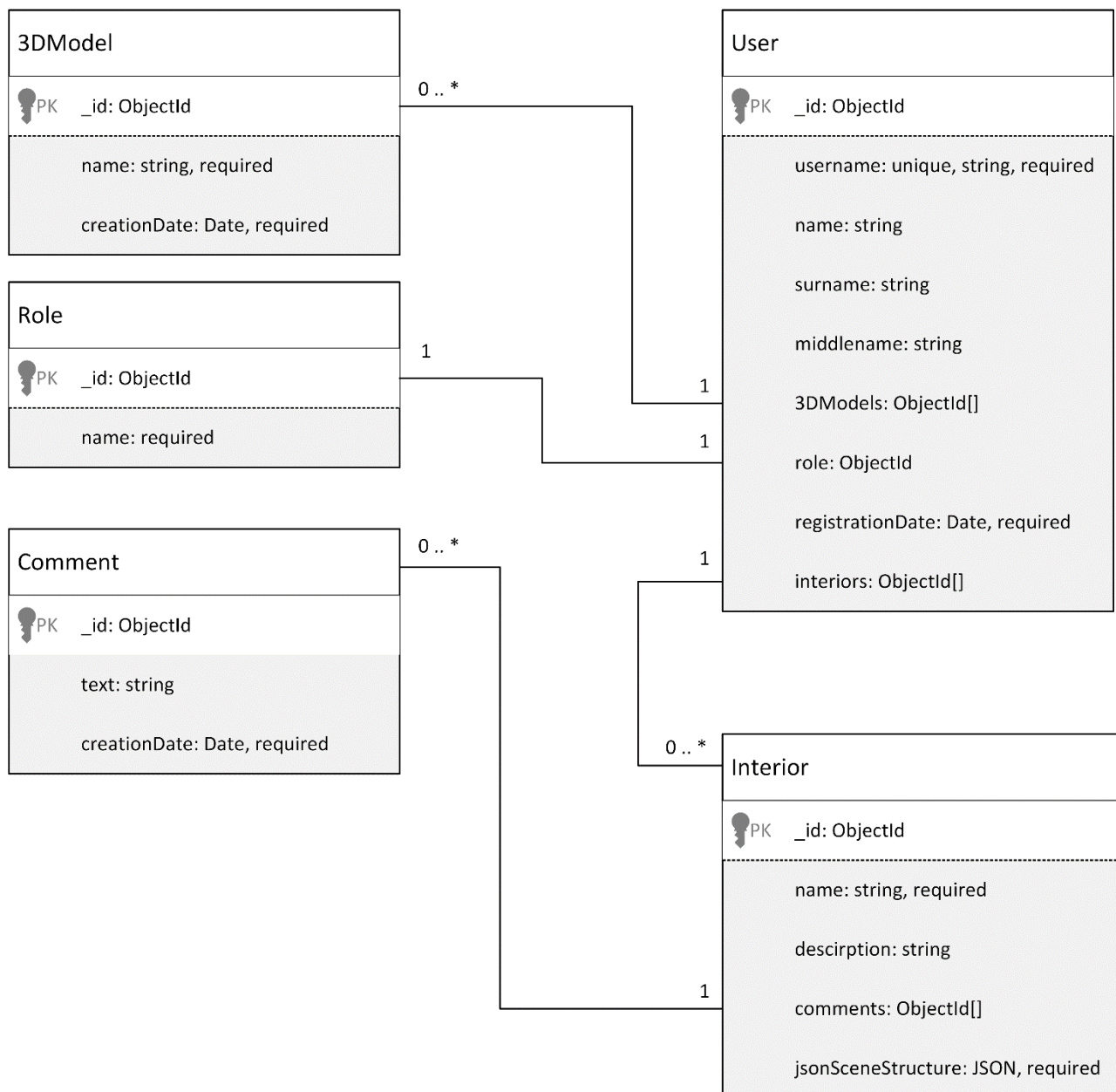


Рисунок 13 – Физическая модель данных MongoDB

MongoDB относится к классу NoSQL СУБД и работает с документами, а не с записями. По этой причине схема «User» изменилась по отношению к первоначально построенной логической модели данных (рисунок 8), а именно появились атрибуты «3DModels», «role», «interiors» с соответствующими типами хранимых данных, а именно специальным типом данных ObjectId, который предоставляет MongoDB. Атрибуты «3DModels» и «interiors» хранят массив ObjectId, что фактически означает, что они хранят уникальный идентификатор соответствующей записи таблицы (или в случае работы с MongoDB – соответствующего документа), то есть говоря языком реляционных баз данных – вторичный ключ. Тоже верно и для схемы «Interiors», а именно для атрибута «comments».

### **3.4 Модули информационной системы виртуализации интерьера «Interior Constructor»**

Программа виртуализации интерьера позволяет создать трехмерный проект интерьера. В данную систему виртуализации интерьера будут входить следующие модули:

- «Авторизация». Включает поля Логин и Пароль, а также кнопку Вход. Позволяет зарегистрироваться в информационной системе и войти в нее с помощью логина и пароля.
- «Интерьеры». Главная страница системы, включающая модули «Мои интерьеры» и «Все интерьеры». А также модуль «Создать новый», в котором пользователь может создать новый интерьер, с возможностью дать ему название и описание.
- «Профиль». Личный кабинет пользователя.

Модуль «Мои интерьеры» позволяет просмотреть и отредактировать имеющиеся интерьеры.

Модуль «Все интерьеры» предлагает массу готовых решений для облегчения функций моделирования. Воспользовавшись функцией

«Добавить готовый интерьер» можно выбрать готовый макет из каталога и создать чертеж в один клик.

3D-программа включает десятки моделей предметов интерьера и техники. Все объекты имеют сортировку по назначению, например, для спальни, гостиной и кабинета, кухни и прихожей, ванной комнаты и так далее. Каждую добавленную модель можно отредактировать по категории – изменить размер, материал отделки и цвет. Чтобы приступить к расстановке мебели, откроем каталог и нажмем на кнопку «Добавить модель». Выбрав нужный предмет интерьера из имеющегося списка, нажмем на кнопку «Добавить модель на сцену», определим для неё место в помещении и развернем при необходимости. Во вкладке «Свойства» и «Материалы» при необходимости можно изменить параметры.

Для сцены также существует возможность настроить отделку пола, стен и потолка. Во вкладке «Свойства» и «Материалы» можно выбрать оформление для каждой из поверхностей.

Программа позволяет неоднократно возвращаться к визуализации проекта. Для этого предусмотрена возможность «Сохранить проект», а также вносить правки в виде комментариев к проекту. Существует возможность изменить статус доступа к проекту, то есть добавить проекту приватный доступ.

### **3.5 Бэкенд информационной системы «Interior Constructor»**

Бэкенд (от англ. back-end) – это часть компьютерной системы или приложения, к которой пользователь не имеет прямого доступа, обычно отвечающая за хранение данных и управление ими.

В нашем случае бэкенд будет реализован с использованием языка программирования Node.js. Node.js – это внутренняя среда выполнения JavaScript, работающая на движке V8 и выполняющая код JavaScript вне веб-браузера.



Для построения структуры бэкенд приложения будет использоваться веб-фреймворк Express.

```
1  {
2    "name": "interior-constructor-api",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "start": "nodemon src/index.js"
8    },
9    "keywords": [],
10   "author": "",
11   "license": "ISC",
12   "devDependencies": {
13     "nodemon": "^2.0.20"
14   },
15   "dependencies": {
16     "bcrypt": "^5.1.0",
17     "body-parser": "^1.20.1",
18     "cors": "^2.8.5",
19     "dotenv": "^16.0.3",
20     "express": "^4.18.2",
21     "jsonwebtoken": "^8.5.1",
22     "mongoose": "^6.7.0",
23     "multer": "^1.4.5-lts.1"
24   }
25 }
```

Рисунок 14 – Package.json файл back-end приложения

На рисунке 14 представлен package.json файл серверного приложения. В нем приведен список всех зависимостей (раздел dependencies), то есть сторонних библиотек, которые будут в той или иной степени задействованы в разработке.

### 3.5.1 Реализация моделей посредством библиотеки Mongoose

Mongoose представляет собой библиотеку объектно-ориентированного программирования JavaScript, которая создает соединение между MongoDB и средой выполнения JavaScript Node.js.

```
1  const mongoose = require("mongoose");
2
3  const UserSchema = mongoose.Schema({
4    username: {
5      type: String,
6      required: true,
7      index: {
8        unique: true,
9      },
10   },
11   password: {
12     type: String,
13     required: true,
14   },
15   name: String,
16   surname: String,
17   interiors: [{ type: mongoose.Schema.Types.ObjectId, ref: "Interior" }],
18 });
19
20 module.exports = mongoose.model("User", UserSchema);
21 |
```

Рисунок 15 – Модель User (схема с использованием mongoose)

На рисунке 15 продемонстрирован код, который создает схему для базы данных под управлением MongoDB.

### 3.5.2 Интерфейс программирования приложения (API) для информационной системы «Interior Constructor»

API (англ. application programming interface) – это программный интерфейс приложения, который представляет собой набор определений и протоколов для создания и интеграции прикладного программного обеспечения.

API для информационной системы «Interior Constructor» будет создан с использованием веб-фреймворка Express, значительно облегчающем работу с роутингом и эндпоинтами в целом.

```
1  const express = require("express");
2  const bcrypt = require("bcrypt");
3  const jwt = require("jsonwebtoken");
4  const User = require("../models/User");
5  const RefreshToken = require("../models/RefreshToken");
6  const router = express.Router();
7  const saltRounds = 10;
8
9  // Create a user      You, 3 months ago • Add basic structure (auth, middleware etc.)
10 router.post("/sign-up", async (req, res) => {
11   try {
12     const { username, password } = req.body;
13
14     if (username.length < 4) {
15       return res.status(400).json({
16         message: "Username must have atleast 4 symbols.",
17       });
18     }
19
20     if (password.length < 8) {
21       return res.status(400).json({
22         message: "Password must have atleast 8 symbols.",
23       });
24     }
25
26     const user = await User.find({ username });
27
28     if (user.length) {
29       return res.status(409).json({
30         message: `User with username '${username}' is already exist.`,
31       });
32     }
33
34     const hash = await bcrypt.hash(password, saltRounds);
35
36     const newUser = new User({
37       username,
38       password: hash,
39     });
40
41     await newUser.save();
42
43     res.json({
44       message: "User has been successfully created.",
45     });
46   } catch (error) {
47     res.status(500).json({
48       message: error,
49     });
50   }
51 });
```

Рисунок 16 – Код эндпоинта sign-up

На рисунке 16 представлен код эндпоинта `sign-up`, который обрабатывает данные и при удовлетворении всех необходимых условий создает новую запись (нового пользователя) в базе данных.

Необходимыми условиями для создания пользователя является:

- количество символов имени пользователя не меньше 4;
- количество символов пароля пользователя не меньше 8;
- пользователь с таким именем не существует в базе данных.

После соблюдения необходимых условий, для пользователя создается хешированный пароль (в качестве инструмента хеширования выступает библиотека `bcrypt`), после чего вызывается код, который отвечает за сохранение информации о пользователе в базе данных. В зависимости от успешности данной операции (база может быть недоступна) пользователю эндпоинта возвращается соответствующее сообщение о статусе операции.

Код, отвечающий за работу эндпоинта `sign-in`, а именно за вход в систему для зарегистрированного пользователя, приведен в Приложении А.

### **3.6 Фронтенд информационной системы «Interior Constructor»**

Фронтенд (англ. `frontend`) – презентационная часть информационной или программной системы, её пользовательский интерфейс и связанные с ним компоненты.

Для информационной системы «Interior Constructor» фронтенд разрабатывается с использованием библиотеки `React`, а также вспомогательных для него библиотек (`react-router-dom`, `formik` и другие), а также библиотеки `Bootstrap`, позволяющей достаточно легко и гибко настраивать `CSS`-стили веб-сайта, а также использовать некоторые готовые `React`-компоненты. Еще одним важным фреймворком для разработки фронтенд части приложения является непосредственно `Babylonjs` и все сопутствующие ему дополнительные пакеты (такие как `@babylonjs/materials`, `@babylonjs/loaders` и другие) [24].

### 3.6.1 Модуль регистрации и авторизации

Модуль регистрации и авторизации состоит из двух частей, выделенных в отдельный сервис внутри системы. Первая его часть, непосредственно, регистрация. Код, отвечающий за регистрацию пользователя показан на рисунке 17.

```
1 type UserData = {
2   username: string;
3   password: string;
4 };
5
6 type CreatedUser = {
7   message: string;
8   token: string;
9   status: number;
10 };
11
12 const apiUrl = process.env.API_URL || 'http://localhost:3000/v1';
13
14 const createUser = async (userData: UserData): Promise<CreatedUser> => {
15   try {
16     const response = await fetch(`${apiUrl}/users/sign-up`, {
17       method: "POST",
18       body: JSON.stringify(userData),
19       headers: {
20         "Content-Type": "application/json",
21       },
22     });
23
24     const responseBody = await response.json();
25
26     return {
27       ...responseBody,
28       status: response.status,
29     };
30   } catch (err) {
31     console.error(err);
32   }
33 };
```

Рисунок 17 – Код функции регистрации пользователя

На данном рисунке можно наблюдать запрос к серверному API по адресу *users/sign-up*. Код этого запроса представлен на рисунке 16.

На рисунке 18 показан код, отвечающий за взаимодействие с эндпоинтом *users/sign-in*, то есть код, который отвечает за авторизацию пользователя в информационной системе «Interior Constructor».

```
35 const signInUser = async (userData: UserData): Promise<CreatedUser> => {
36   try {
37     const response = await fetch(`${apiUrl}/users/sign-in`, {
38       method: "POST",
39       body: JSON.stringify(userData),
40       headers: {
41         "Content-Type": "application/json",
42       },
43     });
44
45     const responseBody = await response.json();
46
47     return {
48       ...responseBody,
49       status: response.status,
50     };
51   } catch (err) {
52     console.error(err);
53   }
54 };
55
56 export { UserData, createUser, signInUser };
```

Рисунок 18 – Код функции авторизации пользователя

Функция *signInUser* в строке 35 на рисунке 18 принимает в качестве аргумента объект *userData* типа *UserData*, а именно объект с полями *login* и *password*. Данный объект в дальнейшем отправляется на сервер.

### 3.6.2 Модуль навигации и разграничения доступа

В целях разграничения доступа к элементам информационной системы «Interior Constructor» построена система навигации (роутинга) с использованием средств библиотеки *react-router-dom*. Для этого был создан корневой файл со всеми необходимыми маршрутами внутри

информационной системы. Фрагмент кода из этого файла приведен на рисунке 19.

```
22 root.render(  
23   <React.StrictMode>  
24     <BrowserRouter>  
25       <AuthProvider>  
26         <Routes>  
27           <Route path="/sign-up" element={<Auth />} />  
28           <Route path="/sign-in" element={<SignIn />} />  
29           <Route  
30             path="/dashboard"  
31             element={  
32               <ProtectedRoute>  
33                 <LayoutDefault>  
34                   <Dashboard />  
35                 </LayoutDefault>  
36               </ProtectedRoute>  
37             }  
38           />  
39           <Route  
40             path="/profile"  
41             element={  
42               <ProtectedRoute>  
43                 <LayoutDefault>  
44                   <Profile />  
45                 </LayoutDefault>  
46               </ProtectedRoute>  
47             }  
48           />  
49           <Route path="*" element={<Auth />} />  
50         </Routes>  
51       </AuthProvider>  
52     </BrowserRouter>  
53   </React.StrictMode>  
54 );  
55
```

Рисунок 19 – Фрагмент кода файла index.tsx, отвечающего за роутинг внутри фронтенд-приложения

На рисунке 19 можно заметить, что маршруты на строках 29 и 39 соответственно являются защищенными, так как компоненты данных маршрутов обернуты в родительский компонент ProtectedRoute, код которого представлен на рисунке 20.

```
1  import { useAuth } from "@/contexts/AuthContext";
2  import { FC, Fragment, PropsWithChildren } from "react";
3  import { Navigate } from "react-router-dom";
4
5  const ProtectedRoute: FC<PropsWithChildren> = ({ children }) => {
6    const { token } = useAuth();
7
8    if (!token) {
9      return <Navigate to="/sign-in" />;
10   }
11
12   return <Fragment>{children}</Fragment>;
13 };
14
15 export default ProtectedRoute;
```

Рисунок 20 – Код компонента ProtectedRoute

Как видно на рисунке 20, компонент ProtectedRoute проверяет наличие токена у пользователя и в случае отсутствия такого перенаправляет пользователя системы на страницу авторизации. Токен пользователя получается с помощью деструктуризации из React-контекста useAuth(). Фрагмент кода компонента AuthContext.tsx представлен на рисунке 21.



```

6  const AuthContext = createContext({});
7
8  export const AuthProvider: FC<PropsWithChildren> = ({ children }) => {
9    const [token, setToken] = useToken();
10   const [errorMessage, setErrorMessage] = useState("");
11
12   const signUp = async (data: UserData) => {
13     try {
14       const { message, status } = await createUser(data);
15
16       if (message && status !== 200) {
17         setErrorMessage(message);
18       }
19     } catch (err) {
20       console.error(err);
21     }
22   };
23
24   const signIn = async (data: UserData) => {
25     try {
26       const { token: _token, message, status } = await signInUser(data);
27
28       if (_token) {
29         setToken(_token);
30         setErrorMessage("");
31       }
32
33       if (message && status !== 200) {
34         setErrorMessage(message);
35       }
36     } catch (err) {
37       console.error(err);
38     }
39   };
40
41   const value = useMemo(
42     () => ({ signUp, signIn, token, errorMessage }),
43     [signUp, token, errorMessage]
44   );
45
46   return <AuthContext.Provider value={value}>{children}</AuthContext.Provider>;
47 };

```

Рисунок 21 – Фрагмент кода компонента AuthContext.tsx

Компонент, фрагмент кода которого представлен на рисунке 21, отвечает за хранение токена пользователя. Данный компонент является контекстным. Контекст в React — это концепция, которая позволяет снабжать дочерние компоненты глобальными данными, независимо от того, насколько глубоко они находятся в дереве компонентов.

### 3.6.3 Модуль управления интерьерами

Данный модуль отвечает за отображение пользователю списка интерьеров, а также элементов управления интерьерами, в том числе их созданием. Фрагмент кода данного модуля представлен на рисунке 22.

```
1 import { FC, useEffect, useState } from "react";
2 import "holderjs";
3 import Button from "react-bootstrap/Button";
4 import ButtonGroup from "react-bootstrap/ButtonGroup";
5 import { getUserInteriors } from "@services/InteriorService";
6 import { useAuth } from "@contexts/AuthContext";
7
8 const Dashboard: FC = () => {
9   const { token } = useAuth();
10  const [interiors, setInteriors] = useState([]);
11
12  useEffect(() => {
13    const getInteriors = async () => {
14      const { interiors } = await getUserInteriors(token);
15      setInteriors(interiors);
16    };
17
18    getInteriors();
19  }, []);
20
21  return (
22    <div className="container">
23      <div className="row">
24        <div className="col-12">
25          <div className="d-flex align-items-center justify-content-between mb-5">
26            <ButtonGroup aria-label="Basic example">
27              <Button variant="primary">Мои интерьеры</Button>
28              <Button variant="secondary">Все интерьеры</Button>
29            </ButtonGroup>
30            <Button variant="primary" onClick={() => console.log(123)}>
31              Создать новый
32            </Button>
33          </div>
34        </div>
35      </div>
36    </div>
37  );
38}
```

Рисунок 22 – Фрагмент кода модуля управления интерьерами

На рисунке 22 на 14 строке вызывается функция `getUserInteriors`, которая принимает в качестве аргумента токен пользователя и возвращает массив всех связанных с пользователем интерьеров. В дальнейшем данный

массив отображается на странице информационной системы «Interior Constructor».

### **3.7 Нагрузочное тестирование информационной системы «Interior Constructor» при помощи Yandex.Tank**

Для того чтобы удостовериться в устойчивости разработанных модулей системы, целесообразно провести нагрузочное тестирование и оценить поведение продукта под ожидаемой нагрузкой в 2000 запросов в секунду.

В качестве инструмента для проведения нагрузочного тестирования выбрана программа Yandex.Tank. Yandex.Tank – это расширяемый инструмент нагрузочного тестирования с открытым исходным кодом для продвинутых пользователей Linux, который особенно хорош как часть набора автоматизированного нагрузочного тестирования.

В связи с тем, что данная работа выполняется в операционной системе Windows, а Yandex.Tank работает исключительно в операционной системе Linux, принято решение использовать Docker в качестве инструмента виртуализации. Docker – это открытая платформа для разработки, доставки и запуска приложений. С помощью Docker-контейнера можно установить образ Yandex.Tank в виртуальной среде Linux.

После создания контейнера с установленным в него образом Yandex.Tank можно запускать нагрузочное тестирование. Для этого необходимо сконфигурировать файл в формате .yaml со всеми требуемыми параметрами ключ-значение. Используемая конфигурация представлена на рисунке 23.

```
! load.yaml X
docker-desktop-data > data > docker > volumes > 73193384d2ad01d2d42d445566965571007fb95c9327c087a102c1419571371 > _data > ! load.yaml
1  overload:
2    enabled: true
3    package: yandextank.plugins.DataUploader
4    token_file: "token.txt"
5  phantom:
6    address: "192.168.0.106:9000"
7    load_profile:
8      load_type: rps
9      schedule: const(1,10s) line(1,40,1m) const(2000,1m)
10   header_http: "1.1"
11   uris:
12     - "/"
13   headers:
14     - "[Host: 192.168.0.106:9000]"
15  autostop:
16    autostop:
17      - http(5xx,10%,5s)
18  telegraf:
19    enabled: false
20
```

Рисунок 23 – Файл конфигурации для запуска нагрузочного тестирования

Результат проведения нагрузочного тестирования, а именно его отчет, представлен в Приложении Б.

На рисунке Б.1 видно, что нагрузочное тестирование в 2000 запросов в секунду, прошло успешно и все запросы вернулись со статусом ответа 200. Таким образом можно сделать вывод, что система устойчива и отвечает поставленным требованиям.

#### Выводы по третьей главе

Выбранные технологии для разработки системы виртуализации интерьера обеспечили необходимую функциональную и инструментальную поддержку, достаточное количество готовых библиотек, что в совокупности привело к улучшению качества системы и сокращению трудозатрат на ее разработку.

Выбранная система управления базой данных MongoDB принимает во внимание все необходимые критерии масштабируемости, кроссплатформенности, поддержки средств проектирования и надежности.

Физическая модель данных в нотации UML Database является модифицированной версией логической модели данных, при построении которой учитывались особенности выбранной реализации СУБД.

Модули системы виртуализации позволяют пользователю полностью погрузиться в создание 3D интерьера и без лишних усилий и временных затрат создать новый проект или подобрать из готовых решений необходимый интерьер в зависимости от своих потребностей.

Бэкенд информационной системы «Interior Constructor» был реализован с использованием языка программирования JavaScript в среде Node.js, а фронтенд разработан на языке TypeScript с использованием библиотеки React, включая вспомогательные для него библиотеки. Ещё одним важным фреймворком для разработки фронтенд части приложения является Babylonjs и все сопутствующие ему дополнительные пакеты (такие как @babylonjs/materials, @babylonjs/loaders и другие).

Модули системы успешно прошли нагрузочное тестирование в 2000 запросов в секунду в соответствии с заданием, продемонстрировав 100% устойчивость.

## **Глава 4 Расчет экономической эффективности внедрения информационной системы виртуализации интерьера «Interior Constructor»**

Данная часть бакалаврской работы направлена на анализ всех аспектов ИТ-проекта, которые могут значительно повлиять на его успех или неудачу, по итогам, описанным в предыдущих главах.

### **4.1 Выбор и обоснование метода расчета экономической эффективности**

Экономической категорией характеризующей финансовый результат предпринимательской деятельности организации является прибыль.

Прибыль – разница между доходами (выручки от продажи товаров и оказания услуг) и затратами на производство или приобретение и сбыт этих товаров и услуг в денежном выражении [14].

В нашем случае прибыль поступает за оказание услуг с помощью продукта, позволяющего решать задачи, связанные с моделированием интерьера в 3D пространстве.

Фактором, напрямую определяющим величину прибыли, являются применяемые цены. Свободные цены устанавливаются самой организацией в зависимости от конкурентоспособности, спроса и предложения. Величина прибыли напрямую зависит от объема реализованных работ и услуг, и от вклада сотрудника в увеличение дохода организации.

Для оценки эффективности ИТ-проекта будем использовать метод финансового анализа.

Рассчитать период окупаемости ИТ-проекта можно по формуле (1):

$$PP = \frac{K_0}{ПЧсг}, \quad (1)$$

где PP – срок окупаемости, выраженный в годах;

$K_0$  – сумма вложенных средств;

ПЧсг – чистая прибыль в среднем за год.

Планируется, что внедрение информационной системы будет способствовать сокращению временных затрат и увеличению производительности.

#### **4.2 Расчет фактических затрат на реализацию информационной системы виртуализации интерьера «Interior Constructor»**

Рассчитаем далее стоимость фактических затрат на реализацию проекта системы виртуализации интерьера «Interior Constructor». Для этого обратимся к календарному плану проекта и выведем его распределенный финансовый отчет расходов. Список задач календарного плана приведен в Приложении В, фрагмент сетевого графика данного календарного плана приведен в Приложении Г.

Программа MS Project, предоставляющая возможность управлять проектной деятельностью, ресурсами, задачами и бюджетами, позволяет составить отчет по задачам в их стоимостной оценке. Прежде чем обратиться к данному отчету, ознакомимся с таблицей 13, в которой представлены должности участников проекта и их заработная плата.

Таблица 13 – Стандартные ставки оплаты труда работников в рамках проекта

Название	Стандартная ставка
Lead-дизайнер	1 250,00 □/ч
Дизайнер	500,00 □/ч
Lead-разработчик	1 750,00 □/ч
Бизнес-аналитик	950,00 □/ч
Системный инженер	900,00 □/ч
Проджект-менеджер	900,00 □/ч
Разработчик	1 000,00 □/ч

Данные для колонки стандартной ставки для таблицы 13 были взяты на основе медианных заработных плат для отдельных должностей предприятий сферы информационных технологий [23].

Медианная зарплата – это уровень, выше и ниже которого получают доход одинаковое количество специалистов.

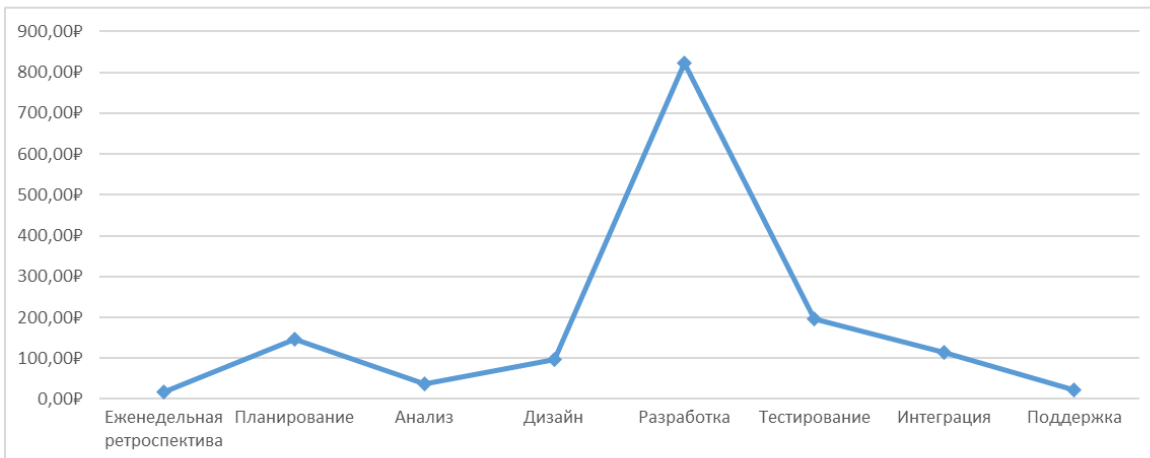
На основе данных, представленных в таблице 13, а также на основе календарного плана, составленного в программе MS Project, на рисунке 24 представлен отчет по суммарным затратам и распределениям стоимости затрат по задачам и ресурсам. Данный отчет позволяет оценить затраты по стадиям проекта, таким как «Планирование», «Анализ», «Дизайн», «Разработка», «Тестирование», «Интеграция», «Поддержка». В отчет также попала повторяющаяся задача «Еженедельная ретроспектива», которая не является стадией, а напротив повторяется еженедельно на протяжении всего жизненного цикла разработки приложения «Interior Constructor».



## Суммарные затраты 1 454 700,00 ₽

### РАСПРЕДЕЛЕНИЕ СТОИМОСТИ ЗАДАЧИ

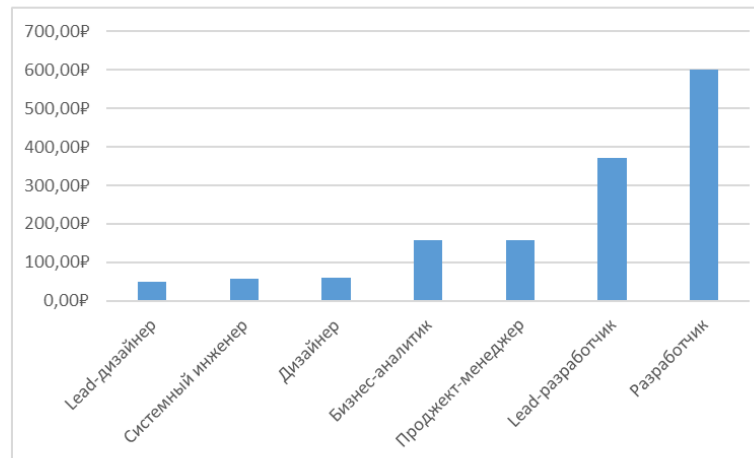
Распределение по стоимости для всех задач верхнего уровня в проекте.



Название	Затраты	Оставшиеся трудозатраты
Еженедельная ретроспектива	18 000,00 ₽	20 ч
Планирование	147 700,00 ₽	158 ч
Анализ	38 000,00 ₽	40 ч
Дизайн	97 200,00 ₽	152 ч
Разработка	821 600,00 ₽	700 ч
Тестирование	197 000,00 ₽	180 ч
Интеграция	113 600,00 ₽	104 ч
Поддержка	21 600,00 ₽	24 ч

### РАСПРЕДЕЛЕНИЕ СТОИМОСТИ РЕСУРСОВ

Распределение по стоимости для всех трудовых ресурсов.



Название	Затраты	Оставшиеся трудозатраты
Lead-дизайнер	50 000,00 ₽	40 ч
Системный инженер	57 600,00 ₽	64 ч
Дизайнер	60 000,00 ₽	120 ч
Бизнес-аналитик	157 700,00 ₽	166 ч
Проджект-менеджер	158 400,00 ₽	176 ч
Lead-разработчик	371 000,00 ₽	212 ч
Разработчик	600 000,00 ₽	600 ч

Рисунок 24 – Отчет по распределениям стоимости затрат по задачам и ресурсам

На рисунке 24 в правой его части также представлены затраты по ресурсами, как в виде графика, так и в виде таблицы. Здесь можно определить какая из перечисленных категорий ресурсов имеет наименьшую или наибольшую стоимость. Эти данные полезны для общего анализа стоимостной эффективности проекта.

Исходя из данных на рисунке 24 стоимость ИТ-решения равна 1 454 700 рублей.

В качестве основного показателя экономической эффективности программного продукта может быть использован годовой экономический эффект (экономическая прибыль):

$$\mathcal{E} = \Delta\mathcal{E}_{\text{год}} - C - E \times K = \Delta\mathcal{E}_{\text{год}} - \Pi, \quad (2)$$

где  $\Delta\mathcal{E}_{\text{год}}$  – годовая экономия (прибыль), вызванная программным продуктом без учета эксплуатационных затрат на него;

$C$  – эксплуатационные затраты на программный продукт;

$E$  – норма прибыли на капитал (нормативная прибыльность);

$K$  – единовременные затраты (капиталовложения);

$\Pi$  – годовые приведенные затраты на программный продукт,

$\Pi = C - E \times K$ .

В среднем компания выполняет 5 заказов по разработке 3D интерьера в месяц, а значительную часть работ выполняют разработчики (в соответствии с рисунком 2). Время выполнения одного заказа разработчиком составляет в среднем одну полную рабочую неделю, то есть 40 рабочих часов. Рассчитать общие трудозатраты в месяц можно используя формулу (3):

$$T = N \times t, \quad (3)$$

где  $T$  – общие трудозатраты в месяц, часы;

$N$  – количество проектов в месяц, штук;

$t$  – количество часов, требуемых на разработку проекта.

Подставляя необходимые значения в формулу (3), вычислим общие трудозатраты в месяц для Разработчика, участвующего в бизнес-процессе создания визуального представления трехмерного пространства:

$$T = 5 \times 40 = 200 \text{ ч.}$$

Из формулы (3) следует, что общие трудозатраты в месяц составляют 200 ч. Из таблицы 13 следует, что часовая тарифная ставка для разработчика равна 1 000 р., что делает месячные затраты в денежном эквиваленте равными 200 000 р.

Эксплуатационные расходы на проект, а именно заработная плата сотрудников, приведены в таблице 14.

Таблица 14 – Заработная плата сотрудников эксплуатационной фазы проекта

Должность	Заработная плата, руб./мес.
Сотрудник технической поддержки	21 600
Инженер поддержки	80 000

Норма прибыли на капитал в рыночных условиях должна быть больше годовой процентной банковской ставки. В качестве нормы прибыли на капитал возьмем коэффициент равный 0,25.

Рассчитаем годовую экономию проекта, используя формулу (2), подставив в неё все полученные выше данные:

$$\begin{aligned} \text{Э} &= 200\,000 \times 12 - (21\,600 \times 12 + 80\,000 \times 12) - 0,25 \times 1\,454\,700 = \\ &= 817\,125 \text{ р.} \end{aligned}$$

Следуя данной формуле, годовая экономия данного проектного решения составит 817 125 р.

Далее рассчитаем период окупаемости ИТ-проекта, подставив получившиеся данные в формулу (1):

$$PP = \frac{1\,454\,700}{817\,125} = 1,78.$$

Найдем срок окупаемости ИТ-проекта в месяцах. Для этого перемножим значение срока окупаемости, выраженного в годах на 12 месяцев:

$$PP_{\text{мес.}} = 1,78 \times 12 = 21,36.$$

Мы установили, что срок окупаемости ИТ-проекта после внедрения составит 21,36 месяца или 21 месяц и 11 дней.

#### **4.3 Расчет ожидаемого экономического эффекта от использования результатов проекта**

Разработанная система рассматривается как средство оптимизации бизнес-процесса «Процесс создания визуального представления трехмерного пространства». Внедрение системы приведёт к следующим положительным результатам:

- улучшение производительности процесса создания визуального трехмерного пространства;
- улучшение управляемости и ускорение бизнес-процесса, так как исключается потребность в пуско-наладочных работах и разработке программной части проекта;
- повышение работоспособности персонала за счет получения возможности выполнения качественно новой задачи.

Проектное решение, планируемое к внедрению в компанию, а именно система виртуализации интерьера «Interior Constructor», соответствует техническому заданию и является целесообразным. ИТ-проект позволит компании масштабироваться на внутреннем рынке благодаря возможности не расширяя штат сотрудников сократить временные затраты и увеличить производительность труда в компании.

#### Выводы по четвертой главе

Разработанная информационная система представляет практическую ценность для её использования, а проведенный расчет экономической эффективности разработанной системы говорит о том, что проектное решение, планируемое к внедрению в компанию, является целесообразным. Экономическая эффективность показала, что внедрение информационной системы позволит получать от неё годовую экономию в размере 817 125 р., а срок окупаемости составит 21 месяц и 11 дней.

## Заключение

Моделирование и анализ бизнес-процессов компании ООО «IT» выявило необходимость оптимизации часто выполняемого бизнес-процесса «Процесс создания визуального представления трехмерного пространства». По этой причине была выявлена необходимость разработки системы виртуализации интерьера, которая в свою очередь приведет к сокращению затрат, повышению скорости и качеству всей работы.

В настоящей бакалаврской работе была подробно проанализирована предметная область, раскрыта деятельность компании ООО «IT», определены направления её деятельности и структурные подразделения.

На основании цели работы, сформированных требований и задач исследования, выбрана технология проектирования, на основании которой выполнены последующие работы по проектированию функционала, ключевых сущностей, состояний, компонентов и моделей данных.

Применение BPMN 2.0 позволило визуализировать бизнес-процесс и улучшить его качество, а равно существенно сократить временные затраты и повысить продуктивность всей работы.

Использование различных UML-диаграмм позволило структурировать систему и определить ее технические особенности.

В процессе написания данной работы были выполнены все поставленные задачи.

Результат проделанной работы – разработанная система виртуализации интерьера на основе BabylonJS, а также опыт использования современной JavaScript-библиотеки для отображения трехмерных объектов Babylonjs, раскрывает научную и практическую значимость данного исследования.

Приведенная оценка экономической эффективности в свою очередь показала, что внедрение информационной системы позволит получать от неё годовую экономию в размере 817 125 р., а срок окупаемости составит 21 месяц и 11 дней.

## Список используемой литературы и используемых источников

1. Антипов, В.А. Введение в программную инженерию : учебник / В.А. Антипов, А.А. Бубнов, А.Н. Пылькин, В.К. Столчнев. – М. : КУРС : ИНФРА-М, 2019. – 336 с. – ISBN 978-5-906923-22-6. – Текст : электронный. – URL: <https://znanium.com/catalog/product/1035160> (дата обращения: 05.04.2023). – Режим доступа: по подписке.
2. Антонов, Г.Д. Управление проектами организации : учебник / Г.Д. Антонов, О.П. Иванова, В.М. Тумин. – М. : ИНФРА-М, 2018. – 244 с. – Текст : непосредственный.
3. ГОСТ 19.101-77. Единая система программной документации (ЕСПД). Виды программ и программных документов (с Изменением N 1) : дата введения 1980-01-01. – официальное издание : единая система программной документации : Сб. ГОСТов. – М.: Стандартиформ, 2010. – Текст: электронный. – URL: <https://docs.cntd.ru/document/1200007627?section=text> (дата обращения: 20.03.2023).
4. ГОСТ 19.201-78. Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению (с Изменением N 1) : дата введения 1980-01-01. – официальное издание : единая система программной документации : Сб. национальных стандартов. – М. : Стандартиформ, 2010. – Текст: электронный. – URL: <https://docs.cntd.ru/document/1200007648?section=text> (дата обращения: 20.03.2023).
5. ГОСТ Р 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов. Типовое руководство – Текст: электронный. – URL: <https://docs.cntd.ru/document/1200027444?section=text> (дата обращения: 10.03.2023).
6. Гражданский кодекс Российской Федерации (часть первая) от 30.11.1994 № 51-ФЗ (редакция от 16.04.2022). – Текст: электронный. – URL:

[https://www.consultant.ru/document/cons\\_doc\\_LAW\\_5142/](https://www.consultant.ru/document/cons_doc_LAW_5142/) (дата обращения: 03.03.2023).

7. Грекул, В.И. Методические основы управления ИТ-проектами : учебник / В.И. Грекул, Н.Л. Коровкина, Ю.В. Куприянов. – М. : ИНТУИТ, 2017; Саратов: Вузовское образование, 2017. – 392 с. – Текст : непосредственный.

8. Грекул, В.И. Управление внедрением информационных систем : учебное пособие / В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина. – М. : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. – 277 с. : ISBN 978-5-4497-0910-3. – Текст : электронный // IPR SMART : [сайт]. – URL: <https://www.iprbookshop.ru/102073.html> (дата обращения: 01.04.2023). – Режим доступа: для авторизир. пользователей

9. Джестон, Д. Управление бизнес-процессами: практическое руководство по успешной реализации проектов / Д. Джестон, Й. Нелис. – М.: Альпина Паблицер, 2019. – 648 с. – ISBN 978-5-9614-4350-9. – Текст : электронный // IPR SMART : [сайт]. – URL: <https://www.iprbookshop.ru/86792.html> (дата обращения: 22.03.2023). – Режим доступа: для авторизир. Пользователей

10. Заботина, Н.Н. Проектирование информационных систем: учеб. пособие. / Н.Н. Заботина. – М.: ИНФРА-М, 2022. – 331 с. + Доп. материалы – (Высшее образование: Бакалавриат). – DOI 10.12737/2519. – ISBN 978-5-16-004509-2. – Текст : электронный. – URL: <https://znanium.com/catalog/product/1840494> (дата обращения: 22.04.2023). – Режим доступа: по подписке.

11. Зараменских, Е. П. Основы бизнес-информатики : учебник и практикум для бакалавриата и магистратуры / Е. П. Зараменских. – М. : Издательство Юрайт, 2019. – 407 с. – (Высшее образование). – ISBN 978-5-9916-8210-7. – Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/433677> (дата обращения: 27.03.2023).



12. Иванова, В.В. Основы бизнес-информатики: учебник / В.В. Иванова, Т.А. Лезина, А.А. Салтан. – СПб : СПбГУ, 2014. – 244 с. – ISBN 978-5-288-05538-6. – Текст : электронный. – URL: <https://znanium.com/catalog/product/941009> (дата обращения: 28.02.2023). – Режим доступа: по подписке.

13. Ипатова, Э.Р. Методологии и технологии системного проектирования информационных систем : учебник / Э.Р. Ипатова, Ю.В. Ипатов. – 3-е изд., стер. – М.: ФЛИНТА, 2021. – 256 с. : табл., схем. – (Информационные технологии). – Режим доступа: по подписке. – Библиогр.: с. 95-96. – ISBN 978-5-89349-978-0. – Текст : электронный. – URL: <https://biblioclub.ru/index.php?page=book&id=79551> (дата обращения: 15.04.2023).

14. Каменская, С.И. Некоторые особенности современных подходов к регулированию заработной платы (на примере ранговых систем оплаты труда)/Социальное и пенсионное право № 4, 2013. – Текст : непосредственный.

15. Коваленко, В.В. Проектирование информационных систем: учеб. пособие / В.В. Коваленко. – 2-е изд. – Москва: ИНФРА-М, 2021. – 357 с. – Текст : непосредственный.

16. Корпорация 1Т – Текст : электронный. – URL: <https://1t.ru/> (дата обращения: 20.03.2023).

17. Кугаевских, А. В. Проектирование информационных систем. Системная и бизнес-аналитика : учебное пособие / А. В. Кугаевских. - Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2018. – 256 с. : табл., схем., ил. – Библиогр.: с. 247-251. – ISBN 978-5-7782-3608-0. – Текст : электронный.– URL: <https://biblioclub.ru/index.php?page=book&id=573827> (дата обращения: 22.04.2023). – Режим доступа: по подписке.

18. Лукманова, И.Г. Управление проектами : учеб. пособие / И. Г. Лукманова, А. Г. Королев, Е. В. Нежникова. – Москва : МГСУ, 2013. – 171 с.:

ил. – ISBN 978-5-7264-0752-4. – Текст : электронный. – URL: <http://www.iprbookshop.ru/20044.html> (дата обращения: 22.04.2023).

19. Об обществах с ограниченной ответственностью : Федеральный закон от 08.02.1998 № 14-ФЗ (ред. от 16.04.2022). – Текст: электронный. – URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_17819/](https://www.consultant.ru/document/cons_doc_LAW_17819/) (дата обращения: 01.03.2023).

20. ОК 029-2014 (КДЕС Ред. 2) : Общероссийский классификатор видов экономической деятельности (утвержден Приказом Росстандарта от 31.01.2014 № 14-ст) (редакция от 26.07.2022). – Текст: электронный. – URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_163320/](https://www.consultant.ru/document/cons_doc_LAW_163320/) (дата обращения: 02.02.2023).

21. Тельнов, Ю. Ф. Реинжиниринг бизнес-процессов : учебное пособие / Ю. Ф. Тельнов. – М. : Евразийский открытый институт, Московский государственный университет экономики, статистики и информатики, 2004. – 77 с. – ISBN 5-7764-0333-2. // Цифровой образовательный ресурс IPR SMART : [сайт]. – Текст : электронный. – URL: <https://www.iprbookshop.ru/10812.html> (дата обращения: 20.03.2023).

22. Федеральный закон №152 «О персональных данных» (с изменениями на 14 июля 2022 года). – Текст: электронный. – URL: <https://docs.cntd.ru/document/901990046?section=text> (дата обращения: 01.03.2023).

23. Хабр: Зарплаты айтишников в первом полугодии 2022: впервые за пять лет средняя зарплата не изменилась. – Текст: электронный. – URL: <https://habr.com/ru/specials/679698/> (дата обращения: 11.06.2023).

24. Babylon.js – Текст : электронный. – URL: <https://www.babylonjs.com/> (дата обращения: 20.03.2023).

25. Cheol-Han, K., Weston, R. H., Hodgson, A. & Kyung-Huy L. (2003). The complementary use of IDEF and UML modelling approaches. Computers in Industry, 20(1), 35-56. – Текст : электронный. – URL: [https://doi.org/10.1016/S0166-3615\(02\)00145-8](https://doi.org/10.1016/S0166-3615(02)00145-8) (дата обращения: 20.03.2023).

26. Cheol-Han, K., Weston, R. H., Hodgson, A. & Kyung-Huy L. (2003). The complementary use of IDEF and UML modelling approaches. *Computers in Industry*, 20(1), 35-56. – Текст : электронный. – URL: [https://doi.org/10.1016/S0166-3615\(02\)00145-8](https://doi.org/10.1016/S0166-3615(02)00145-8) (дата обращения: 15.04.2023).

27. Harmon, P. (2015). The Scope and Evolution of Business Process Management. In J. vom Brocke & M. Rosemann (Eds.), *Handbook on Business Process Management 1. International Handbooks on Information Systems* (pp. 37-80). Berlin, Heidelberg: Springer. – Текст : электронный. – URL: [https://doi.org/10.1007/978-3-642-45100-3\\_3](https://doi.org/10.1007/978-3-642-45100-3_3) (дата обращения: 20.03.2023).

28. Havey, M. *Essential Business Process Modeling* / M. Havey/ – Текст : непосредственный // O'Reilly Media, Inc. – Released August 2005. – P. 352

29. Rumpe, B. *Modeling with UML: language, concepts, methods* / B. Rumpe/ – Текст : непосредственный // 1st ed. Springer, 2016. – P. 295

## Приложение А

### Код эндпоинта sign-in

```
53 router.post("/sign-in", async (req, res) => {
54   const { username, password } = req.body;
55
56   if (username.length < 4) {
57     return res.status(400).json({
58       message: "Username must have atleast 4 symbols.",
59     });
60   }
61
62   if (password.length < 8) {
63     return res.status(400).json({
64       message: "Password must have atleast 8 symbols.",
65     });
66   }
67
68   const user = await User.findOne({ username });
69
70   if (!user) {
71     return res.status(400).json({
72       message: "The username or password is incorrect.",
73     });
74   }
75
76   bcrypt.compare(password, user.password, async (err, result) => {
77     if (!result) {
78       return res.status(400).json({
79         message: "The username or password is incorrect.",
80       });
81     }
82
83     const token = generateAccessToken({ username, id: user._id });
84     const refreshToken = jwt.sign({ username }, process.env.REFRESH_TOKEN_SECRET);
85
86     const _refreshToken = new RefreshToken({
87       value: refreshToken
88     });
89
90     await _refreshToken.save();
91
92     return res.json({
93       token,
94       refreshToken,
95       message: "You have successfully signed in.",
96     });
97   });
98 });
99
100 function generateAccessToken(user) {
101   return jwt.sign(user, process.env.ACCESS_TOKEN_SECRET, { expiresIn: '24h' });
102 }
```

Рисунок А.1 – Код эндпоинта sign-in

# Приложение Б

## Отчет о проведенном нагрузочном тестировании

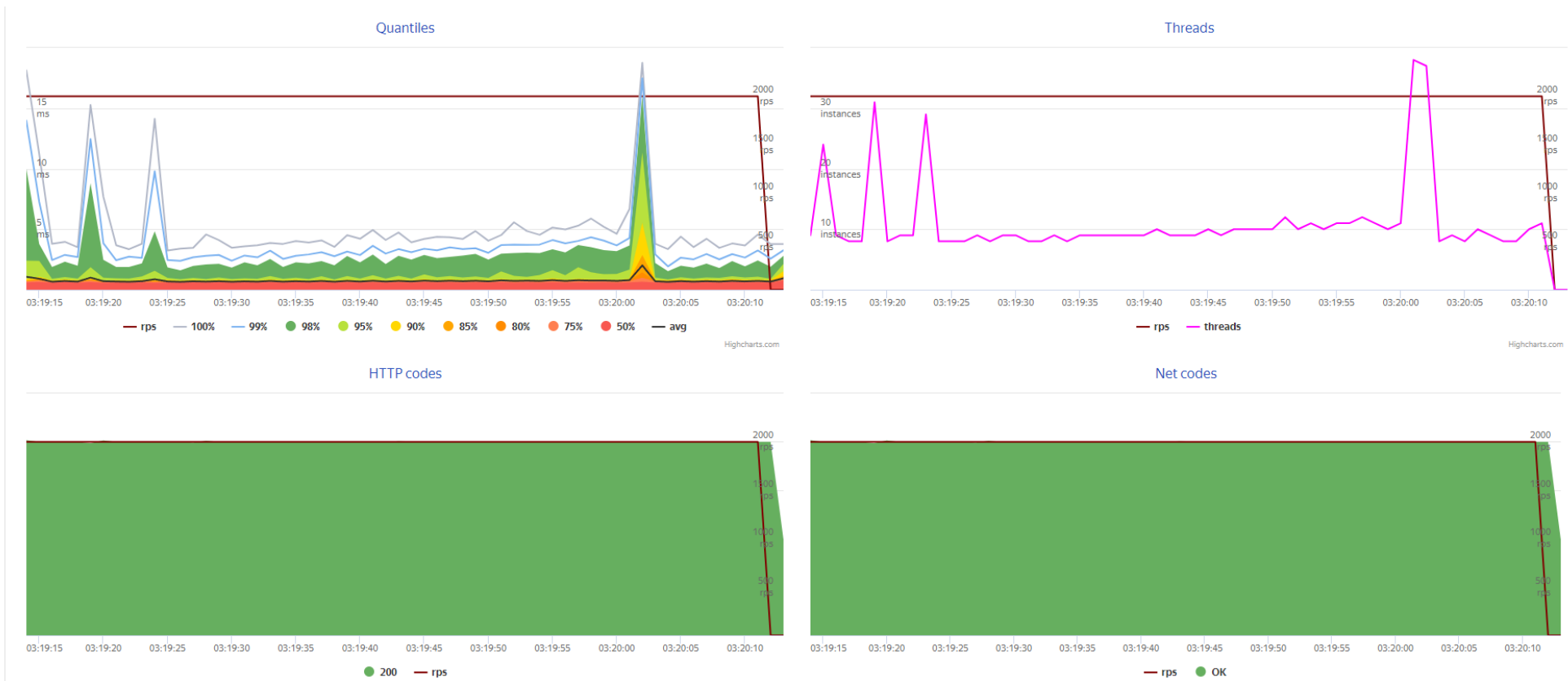


Рисунок Б.1 – Отчет о проведенном нагрузочном тестировании

## Приложение В

### Список задач календарного плана

	Режим задачи	СДР	Название задачи	Длительность	Начало	Окончание	Предшественники	Названия ресурсов
0		0	➤ <b>Разработка информационной системы 3D визуализации интерьера</b>	<b>122,25 дней</b>	<b>Пн 15.05.23</b>	<b>Ср 01.11.23</b>		
1		1	➤ Еженедельная ретроспектива	110,13 дней	Пн 22.05.23	Пн 23.10.23		
25		2	➤ <b>Планирование</b>	<b>19 дней</b>	<b>Пн 15.05.23</b>	<b>Чт 08.06.23</b>		
26		2.1	Определить функциональные требования к разрабатываемой системе	5 дней	Пн 15.05.23	Пт 19.05.23		Бизнес-аналитик
27		2.2	Провести анализ рисков	2 дней	Пн 22.05.23	Вт 23.05.23	26	Бизнес-аналитик
28		2.3	Определить потребности в ресурсах и персонале	4 ч	Ср 24.05.23	Ср 24.05.23	27	Бизнес-аналитик;Проджект-менеджер
29		2.4	Создать детальный декомпозированный план задач по проекту	5 дней	Ср 24.05.23	Ср 31.05.23	28	Проджект-менеджер
30		2.5	Провести стартовое совещание по проекту	2 ч	Ср 31.05.23	Ср 31.05.23	29	Проджект-менеджер
31		2.6	Составить план MVP	5 дней	Ср 31.05.23	Ср 07.06.23	30	Бизнес-аналитик
32		2.7	Провести совещание по согласованию MVP	2 ч	Ср 07.06.23	Ср 07.06.23	31	Бизнес-аналитик;Проджект-менеджер
33		2.8	Завершить стадию планирования	1 день	Чт 08.06.23	Чт 08.06.23	32	Бизнес-аналитик
34		3	➤ <b>Анализ</b>	<b>5 дней</b>	<b>Пт 09.06.23</b>	<b>Чт 15.06.23</b>		
35		3.1	Провести общий анализ трудоемкости проекта	2 дней	Пт 09.06.23	Пн 12.06.23	33	Бизнес-аналитик
36		3.2	Провести анализ конкурентных продуктов и систем	2 дней	Вт 13.06.23	Ср 14.06.23	35	Бизнес-аналитик
37		3.3	Завершить стадию анализа	1 день	Чт 15.06.23	Чт 15.06.23	36	Бизнес-аналитик
38		4	➤ <b>Дизайн</b>	<b>18 дней</b>	<b>Пт 16.06.23</b>	<b>Вт 11.07.23</b>		
39		4.1	Провести стартовое совещание по дизайну проекта	1 день	Пт 16.06.23	Пт 16.06.23	37	Lead-дизайнер;Проджект-менеджер;Дизайнер
40		4.2	Составить гайд-лайн дизайна проекта	2 дней	Пн 19.06.23	Вт 20.06.23	39	Lead-дизайнер
41		4.3	Реализовать дизайн системы в рамках MVP	14 дней	Чт 22.06.23	Вт 11.07.23	40ОН+1 день	Дизайнер
42		5	➤ <b>Разработка</b>	<b>69 дней</b>	<b>Ср 12.07.23</b>	<b>Вт 17.10.23</b>		
43		5.1	Провести стартовое совещание по разработке проекта	1 день	Ср 12.07.23	Чт 13.07.23	41	Lead-разработчик;Проджект-менеджер;Разработчик;Системный инженер
44		5.2	Собрать команду, назначить тим-лидера	1 день	Чт 13.07.23	Пт 14.07.23	43	Проджект-менеджер
45		5.3	Провести глубокую декомпозицию задач по разработке проекта	5 дней	Пт 14.07.23	Пт 21.07.23	44	Lead-разработчик;Проджект-менеджер[50%]
46		5.4	Реализовать MVP	60 дней	Вт 25.07.23	Вт 17.10.23	45ОН+2 дней	Lead-разработчик[25%];Разработчик
47		6	➤ <b>Тестирование</b>	<b>13 дней</b>	<b>Вт 03.10.23</b>	<b>Пт 20.10.23</b>		
48		6.1	Провести А/В тестирование	2 дней	Вт 17.10.23	Чт 19.10.23	46	Системный инженер
49		6.2	Провести глобальный smoke-test	3 дней	Вт 17.10.23	Пт 20.10.23	46	Проджект-менеджер
50		6.3	Добиться покрытия кода тестами в размере не менее 90%	10 дней	Вт 03.10.23	Вт 17.10.23	46ОН-10 дней	Lead-разработчик[25%];Разработчик
51		6.4	Провести функциональное тестирование	2 дней	Вт 17.10.23	Чт 19.10.23	50	Разработчик
52		6.5	Провести интеграционное тестирование	2 дней	Вт 17.10.23	Чт 19.10.23	50	Разработчик
53		6.6	Завершить стадию тестирования	1 день	Чт 19.10.23	Пт 20.10.23	52;50	Lead-разработчик
54		7	➤ <b>Интеграция</b>	<b>5 дней</b>	<b>Пт 20.10.23</b>	<b>Пт 27.10.23</b>		
55		7.1	Настроить продакшн-окружение и произвести его тестирование	3 дней	Пт 20.10.23	Ср 25.10.23	53	Системный инженер
56		7.2	Запустить MVP	2 дней	Ср 25.10.23	Пт 27.10.23	55	Lead-дизайнер;Lead-разработчик;Бизнес-аналитик;Проджект-менеджер;Системный инженер
57		8	➤ <b>Поддержка</b>	<b>3 дней</b>	<b>Пт 27.10.23</b>	<b>Ср 01.11.23</b>		
58		8.1	Сформировать команду поддержки	1 день	Пт 27.10.23	Пн 30.10.23	56	Проджект-менеджер
59		8.2	Провести ретроспективу проекта	2 дней	Пн 30.10.23	Ср 01.11.23	58	Проджект-менеджер

Рисунок В.1 – Список задач календарного плана

# Приложение Г

## Фрагмент сетевого графика календарного плана

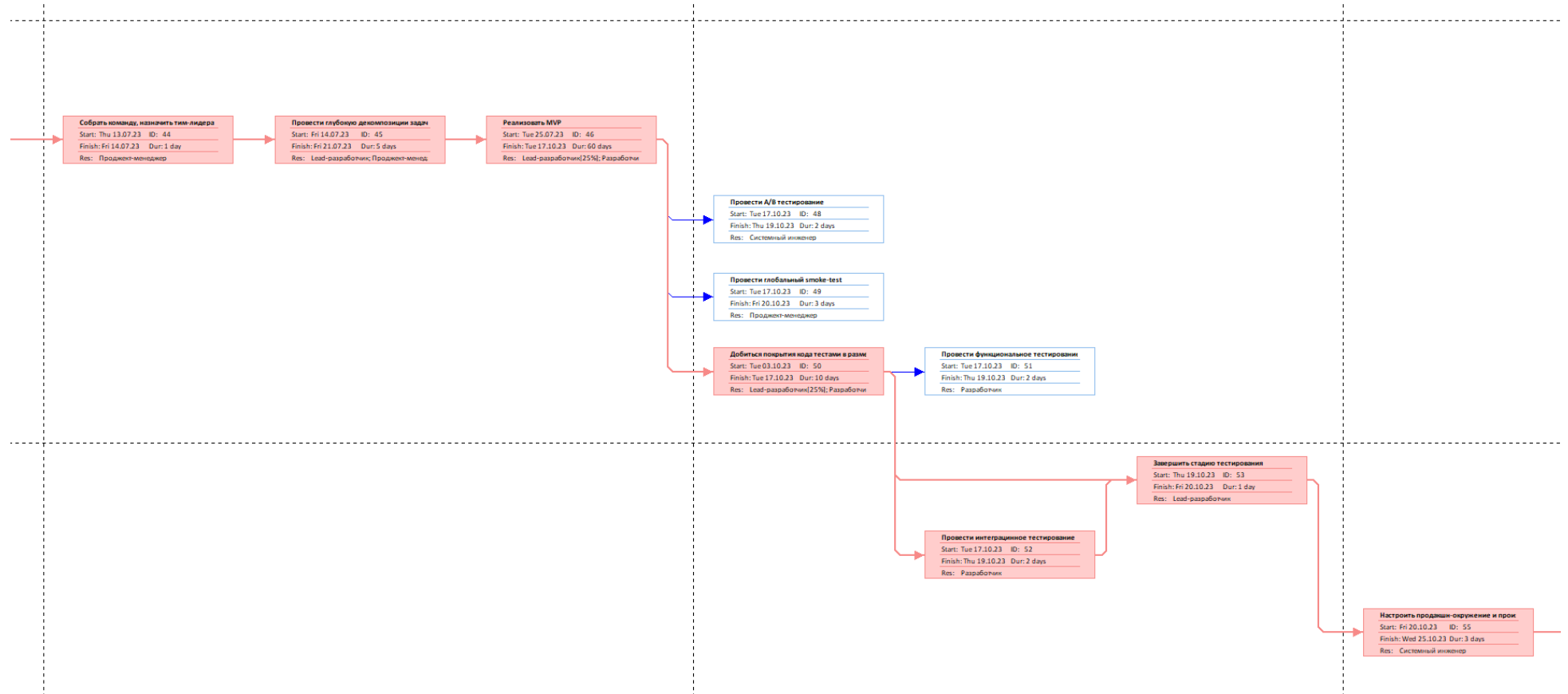


Рисунок Г.1 – Фрагмент сетевого графика календарного плана