

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

Кафедра Прикладная математика и информатика  
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Разработка программного обеспечения

(направленность (профиль) / специализация)

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка мобильного приложения для визуализации архитектурных проектов с использованием технологии AR (на примере ООО «Кирамет»)»

Обучающийся

В.В. Долгалёв

(Инициалы Фамилия)

(личная подпись)

Руководитель

Н.Н. Казаченок

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

## Аннотация

В выпускной квалификационной работе рассматривается разработка AR-приложения для визуализации архитектурных проектов. Сервис представляет собой программное решение, созданное для повышения эффективности представления архитектурных идей и оптимизации взаимодействия между заказчиками и разработчиками. Основная цель работы заключается в создании современного инструмента, позволяющего осуществлять детальный виртуальный тур по 3D-моделям объектов, а также обеспечивающего удобное управление комментариями и скриншотами в реальном времени.

Во введении определены актуальность, цели, задачи, объект и предмет исследования.

В первой главе проведён анализ архитектурной отрасли, современных методов визуализации, бизнес-процессов ООО «Кирамет» и аналогичных программных решений, на основе которого сформулированы требования к системе.

Вторая глава описывает проектирование программного обеспечения, включающее выбор методологии и архитектуры, технологий, произведено моделирование системы и разработан интуитивный интерфейс AR-приложения.

Третья глава демонстрирует этапы реализации и тестирования ключевых функциональных компонентов, включая пользовательский интерфейс и AR-функциональность.

Выпускная квалификационная работа представлена на 61 страницах и включает 31 рисунок, 11 таблиц и список литературы из 23 наименований.

## Оглавление

Введение.....	4
Глава 1 Анализ предметной области и постановка задачи на разработку программного обеспечения.....	7
1.1 Характеристика архитектурной отрасли и современных методов визуализации проектов.....	7
1.2 Анализ бизнес-процессов предприятия ООО «Кирамет».....	9
1.3 Обзор и анализ аналогов программного обеспечения.....	13
1.4 Разработка модели бизнес-процесса «как должно быть».....	16
1.5 Разработка требований к программному обеспечению.....	18
Глава 2 Проектирование программного обеспечения.....	21
2.1 Выбор методологии проектирования программного обеспечения.....	21
2.2 Выбор технологий и инструментов разработки.....	22
2.3 Логическое моделирование программного обеспечения.....	25
2.4 Моделирование данных системы.....	32
2.5 Проектирование пользовательского интерфейса.....	34
Глава 3 Реализация и тестирование программного обеспечения.....	37
3.1 Реализация AR-приложения.....	37
3.2 Тестирование программного обеспечения.....	51
Заключение.....	57
Список используемой литературы.....	59

## Введение

В эпоху цифровой революции архитектурная отрасль переживает фундаментальную трансформацию своих методов и подходов. Внедрение передовых информационных технологий радикально меняет процессы визуализации и презентации архитектурных проектов, что не только повышает эффективность работы специалистов, но и выводит коммуникацию с клиентами на качественно новый уровень. Современные цифровые инструменты становятся не просто дополнением к традиционным методам проектирования, а необходимым условием для сохранения конкурентоспособности в быстро меняющемся рынке архитектурных услуг.

Среди современных технологий визуализации особенно перспективным является дополненная реальность (AR). В отличие от традиционных способов демонстрации проектов, AR позволяет увидеть 3D-модель архитектурного объекта в контексте реальной местности. Такой подход существенно упрощает процесс обсуждения проекта с заказчиком, помогая избежать потенциальных недопониманий на ранних этапах проектирования.

Учитывая значительные преимущества этой технологии, многие архитектурные компании начинают внедрять AR в свои рабочие процессы. Компания ООО «Кирамет», специализирующаяся на разработке архитектурных решений, стремится внедрить инновационные технологии в свою деятельность для повышения эффективности работы и улучшения качества обслуживания клиентов. В связи с этим возникла необходимость в создании специализированного мобильного AR-приложения, которое позволит оптимизировать процесс демонстрации и согласования проектов.

Целью данной выпускной квалификационной работы является разработка мобильного приложения с использованием технологий дополненной реальности для визуализации архитектурных проектов компании ООО «Кирамет». Объектом исследования выступает процесс визуализации

архитектурных решений в деятельности компании, а предметом исследования – применение AR-технологий для совершенствования этого процесса.

Для достижения цели необходимо выполнить следующие задачи:

- провести анализ и описание предметной области AR-визуализации в архитектурном проектировании;
- выполнить обзор и сравнительный анализ существующих AR-решений для визуализации архитектурных проектов;
- разработать функциональные и нефункциональные требования к программному обеспечению с учетом специфики AR-технологий;
- выбрать методологии проектирования и технического стека;
- выполнить логическое моделирование структуры AR-приложения;
- спроектировать схемы данных и взаимодействия компонентов системы;
- реализовать программное обеспечение с интеграцией AR-функционала и необходимых программных модулей;
- провести комплексное тестирование для проверки производительности, стабильности работы и удобства использования.

Практическая значимость работы заключается в создании инструмента, который позволит компании ООО «Киамет» повысить качество презентации проектов, оптимизировать процесс согласования решений с заказчиками и укрепить свои конкурентные позиции на рынке архитектурных услуг.

Работа состоит из трех глав, введения, заключения и списка литературы.

В первой главе проводится анализ предметной области, включающий характеристику архитектурной отрасли, современные методы визуализации, анализ бизнес-процессов предприятия, а также обзор аналогичных программных решений. На основании выявленных недостатков существующих процессов сформулированы требования к разрабатываемой системе.

Вторая глава посвящена проектированию программного обеспечения. Здесь описаны выбор методологии проектирования и архитектуры, проведен

выбор технологий и инструментов разработки, произведено логическое и физическое моделирование системы, а также разработан пользовательский интерфейс, обеспечивающий интуитивное взаимодействие пользователя с AR-приложением.

Третья глава демонстрирует этапы реализации и тестирования программного обеспечения. Описывается реализация ключевых функциональных компонентов приложения, а также результаты проведенного тестирования пользовательского интерфейса и функциональности AR-компонентов.

Выпускная квалификационная работа представлена на 61 странице и включает 31 рисунок, 11 таблиц и список литературы из 23 наименований.

## **Глава 1 Анализ предметной области и постановка задачи на разработку программного обеспечения**

### **1.1 Характеристика архитектурной отрасли и современных методов визуализации проектов**

Сфера архитектуры и инженерных изысканий представляет собой высококонкурентный и трудоемкий сегмент рынка. По данным открытых источников, на начало 2025 года в Российской Федерации зарегистрировано 7875 действующих организаций с кодом ОКВЭД 71.1, осуществляющих деятельность в области архитектуры и инженерных изысканий [17].

В контексте цифровой трансформации архитектурной отрасли особую значимость приобретают технологии, способные качественно улучшить взаимодействие с заказчиками и оптимизировать процесс проектирования. Клиенты ожидают не просто качественных архитектурных решений, но и инновационных способов их представления, которые позволяют наглядно оценить будущий проект еще на этапе концепции.

Среди современных методов визуализации особое место занимает дополненная реальность (AR), которая открывает принципиально новые возможности для архитектурного проектирования. Эксперты отмечают формирование новой экономической модели, основанной на впечатлениях, где ключевым фактором успеха становится внедрение технологий дополненной и виртуальной реальности как инновационных инструментов коммуникации в государственном и коммерческом секторах [4].

Технология дополненной реальности предоставляет ряд существенных преимуществ для архитектурной отрасли:

- создание интерактивных визуализаций проектов с возможностью взаимодействия в режиме реального времени, что значительно повышает качество презентационных материалов и способствует более эффективному принятию решений [5];

- обеспечение эффекта иммерсивного присутствия, позволяющего заказчикам получить реалистичное представление о пространственных характеристиках будущего объекта, включая оценку высоты помещений и общего масштаба проекта [8];
- сокращение временных и финансовых затрат на этапе проектирования за счет возможности проведения виртуальных испытаний и внесения оперативных изменений [5];
- формирование единой виртуальной среды для всех участников проекта, что способствует повышению эффективности командной работы и качества итогового результата [20].

Основу современных AR-платформ формирует комплекс взаимосвязанных технологий. Технология «Device tracking» обеспечивает точное отслеживание положения устройства в пространстве, в то время как «Plane tracking» позволяет системе распознавать горизонтальные поверхности для корректного размещения виртуальных объектов. Технология «Face tracking» реализует возможности отслеживания и взаимодействия с лицевыми движениями пользователя, а «Point clouds» осуществляет сегментацию изображения для получения данных о глубине сцены. Для работы с различными типами объектов применяются технологии «2D Image tracking», обеспечивающая распознавание двумерных изображений, и «3D Object tracking», позволяющая отслеживать трехмерные объекты в режиме реального времени. Именно эти технологии определяют функциональные возможности и эффективность современных AR-решений [6].

Таким образом, технологии дополненной реальности становятся важным инструментом в современном архитектурном проектировании, обеспечивая качественно новый уровень визуализации и взаимодействия на всех этапах реализации проектов.



## 1.2 Анализ бизнес-процессов предприятия ООО «Кирамет»

ООО «Кирамет», зарегистрированное 08.10.2014 в Симферополе, специализируется на предоставлении широкого спектра архитектурных и инженерных услуг. Основная деятельность компании относится к коду ОКВЭД 71.1 - архитектура, инженерные изыскания и предоставление технических консультаций в этих областях. Дополнительно компания осуществляет деятельность в сфере разработки компьютерного программного обеспечения и консультативной деятельности в области компьютерных технологий [10].

Основные направления деятельности компании включают: инжиниринг, градостроительство, реставрация, археология.

Организационная структура ООО «Кирамет» представлена на рисунке 1.

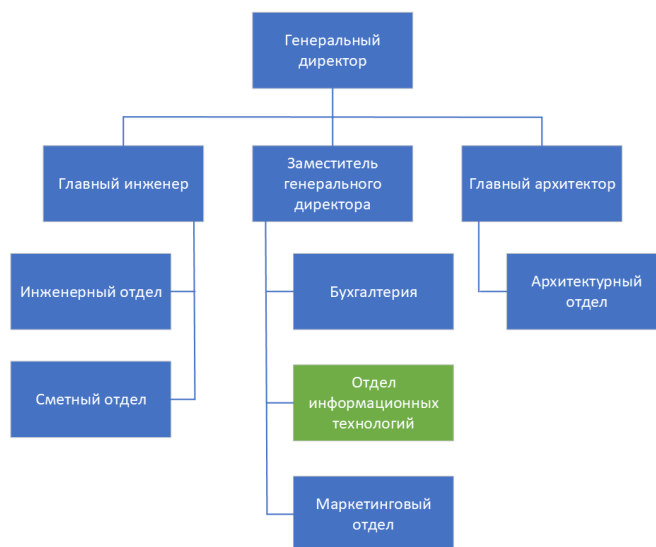


Рисунок 1 - Организационная диаграмма ООО «Кирамет»

Компания ООО «Кирамет» имеет иерархическую структуру управления с генеральным директором во главе, который отвечает за стратегическое планирование, финансовый контроль, развитие бизнеса и инновации.

Под руководством генерального директора находятся три ключевые должности:

- главный инженер, отвечает за техническую сторону проектов, включая работу инженерного и сметного отделов;
- заместитель генерального директора, курирует бухгалтерию, отдел информационных технологий и маркетинговый отдел;
- главный архитектор, руководит архитектурным отделом, где создаются концепции и дизайн-проекты.

Данная организационная структура позволяет эффективно координировать работу всех подразделений, обеспечивая высокое качество услуг в сфере архитектуры и строительства.

Для оптимизации работы компании и выявления потенциальных улучшений было проведено исследование действующих бизнес-процессов с использованием методологии IDEF0. Данная методология зарекомендовала себя как эффективный инструмент моделирования, применяемый системными аналитиками для решения разнообразных задач: от разработки программного обеспечения до управления производственными системами [18].

В результате анализа существующих бизнес-процессов была создана модель «как есть» (AS-IS), отражающая текущее состояние процесса работы над архитектурным проектом, представлена на рисунке 2.

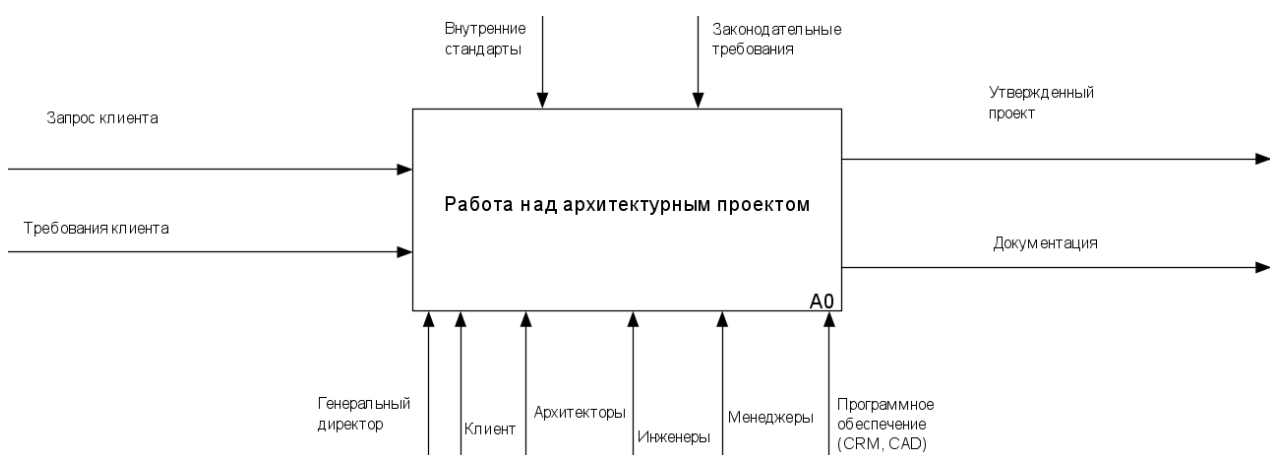


Рисунок 2 – Контекстная диаграмма AS-IS

На контекстной диаграмме представлен общий вид процесса «Работа над архитектурным проектом». Входными данными являются запрос клиента и требования клиента. Управляющими воздействиями выступают внутренние стандарты компании и законодательные требования. Механизмами реализации процесса служат генеральный директор, клиент, архитекторы, инженеры, менеджеры и программное обеспечение (CRM, CAD). Выходными данными являются утвержденный проект и документация.

Декомпозиция контекстной диаграммы представлена на рисунке 3.

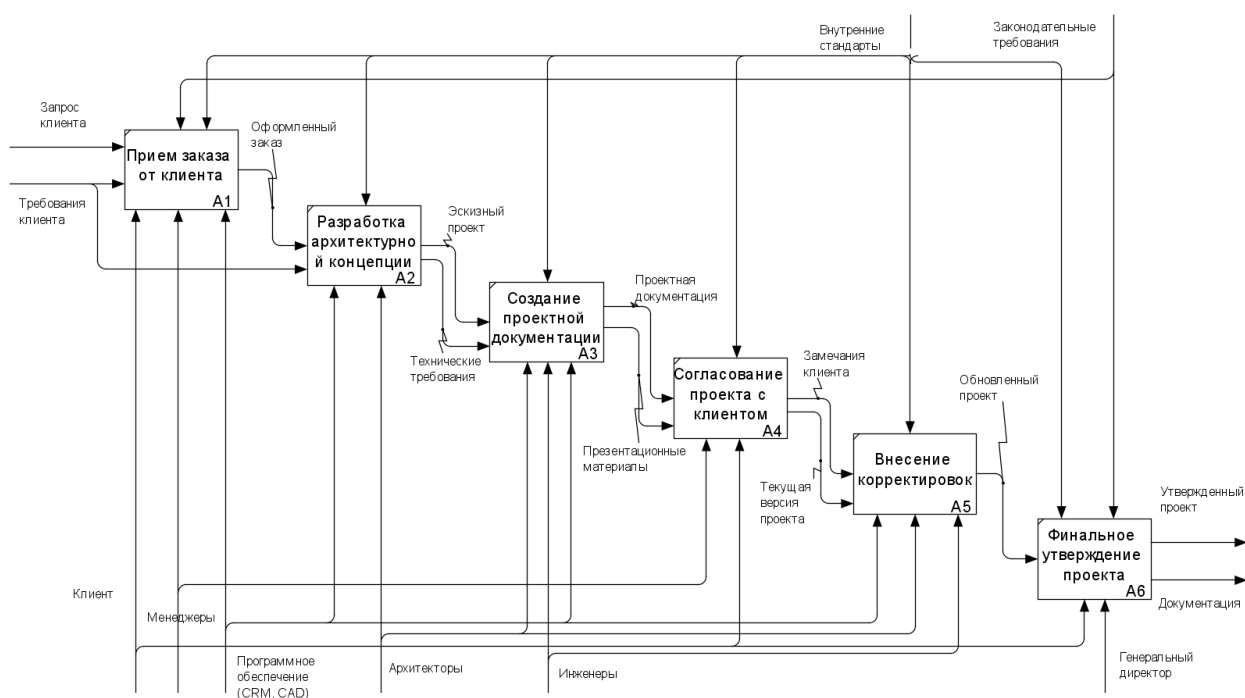


Рисунок 3 – Декомпозиция контекстной диаграммы AS-IS

Декомпозиция контекстной диаграммы детализирует процесс и демонстрирует последовательность взаимосвязанных подпроцессов:

- прием заказа от клиента (A1) – начальный этап, на котором происходит формирование оформленного заказа на основе запроса и требований клиента;

- разработка архитектурной концепции (A2) – процесс создания эскизного проекта на основе оформленного заказа с учетом требований клиента и внутренних стандартов;
- создание проектной документации (A3) – формирование полного комплекта проектной документации на основе эскизного проекта и технических требований;
- согласование проекта с клиентом (A4) – представление проектной документации клиенту для получения обратной связи, при этом учитываются предварительные материалы и текущая версия проекта;
- внесение корректировок (A5) – процесс изменения проекта на основе замечаний клиента и требований законодательства;
- финальное утверждение проекта (A6) – заключительный этап, на котором происходит официальное утверждение проекта и подготовка итоговой документации.

Анализ диаграммы AS-IS показал, что в текущем процессе «Работа над архитектурным проектом» есть ряд не эффективных процессов, а именно:

- сложности в визуализации проектов для клиентов на этапе согласования приводят к тому, что клиенту приходится принимать решение о согласовании проекта на основе презентации и предварительных чертежей;
- большие временные затраты на внесение корректировок и повторные согласования, так как после внесения клиентом замечаний тратится много времени на исправления предварительных чертежей и повторное согласование с клиентом;
- отсутствие инструментов для удаленной работы и коммуникации с клиентами, вызывает увеличивающееся количество взаимодействий, которые требуют очного присутствия клиента и архитекторов, что увеличивает временные и организационные издержки.

Выявленные в ходе анализа проблемы напрямую влияют на эффективность работы компании и удовлетворенность клиентов. Сложности визуализации приводят к тому, что клиенты не могут в полной мере оценить проект до его реализации, что увеличивает риск последующих корректировок. Большие временные затраты на внесение изменений и повторные согласования создают дополнительную нагрузку на проектную команду и увеличивают сроки выполнения заказов. Отсутствие инструментов удаленной работы ограничивает географию деятельности компании и создает неудобства для клиентов.

Внедрение AR-технологий способно эффективно решить данные проблемы за счет:

- создания интерактивных трехмерных визуализаций проектов, которые клиент может рассматривать со всех сторон и в реальном масштабе;
- возможности вносить изменения в проект и сразу же демонстрировать результат клиенту, что значительно сократит цикл согласований;
- обеспечения удаленного доступа к визуализации проекта через мобильные устройства, что позволит проводить презентации и согласования в дистанционном формате.

Таким образом, внедрение специализированного AR-приложения является обоснованным решением для оптимизации ключевых бизнес-процессов ООО «Кирамет» и повышения качества обслуживания клиентов.

### **1.3 Обзор и анализ аналогов программного обеспечения**

В настоящее время на рынке представлен ряд программных решений для AR-визуализации архитектурных проектов.

Сравнительный анализ существующих AR-решений, включая их функциональные возможности, стоимость и технические характеристики, представлен в таблице 1.

Таблица 1 - Сравнение существующих решений

Решение	Преимущества	Недостатки	Стоимость	Основное назначение
Planoplan	Простота использования, доступная цена	Ограничено интерьерами, низкий функционал	Низкая	Визуализация интерьеров
ARvis	Низкие требования к оборудованию	Ограниченный функционал базовой версии	Низкая	Универсальное приложение для создания AR изображений
Gamma AR	Визуализация BIM в реальном масштабе, совместная работа	Высокая стоимость лицензии, сложность освоения	Высокая	Крупные BIM-проекты
VisualLive AR	Интеграция с Revit, профессиональные инструменты	Высокие затраты, сложность развертывания	Очень высокая	Крупные проекты в Revit и Navisworks

Одним из популярных инструментов от российских разработчиков для 3D-визуализации интерьеров с элементами AR является Planoplan. Его преимущество - простота использования, низкий порог входа и доступная цена. Однако основное предназначение Planoplan - работа с интерьерами, что ограничивает его применение для крупных архитектурных проектов [14].

Еще одна российская платформа для дополненной реальности, предоставляющая гибкий инструмент для работы с AR в архитектуре - ARvis. Имеет низкие требования к оборудованию, что делает её доступной для малого и среднего бизнеса. Недостатком является ограниченный набор функций [11].

Среди зарубежных разработок существует платформа Gamma AR, которая выделяется визуализацией BIM-моделей в реальном масштабе, интеграцией с популярными САПР-системами и поддержкой совместной

работы [12]. Однако высокие лицензионные затраты, сложность освоения и требования к производительности оборудования ограничивают его применение для небольших компаний или проектов.

Также одним из аналогов выступает платформа VisualLive AR, которая ориентирована на крупные проекты благодаря интеграции с Revit и Navisworks, а также наличию профессиональных инструментов [13]. При этом высокие затраты на лицензии и обучение, а также сложность развертывания делают его менее подходящим для компаний с ограниченными ресурсами.

Анализ существующих решений показал, что доступные на рынке продукты Planoplan и ARvis не обеспечивают необходимый уровень функциональности, а Gamma AR и VisualLive AR требуют значительных затрат на внедрение и сопровождение. Кроме того, ни одно из решений не предоставляет достаточной гибкости для адаптации под специфические процессы ООО «Кирамет».

Таким образом, несмотря на разнообразие доступных решений, разработка собственного AR-приложения для ООО «Кирамет» представляется обоснованной. Среди основных причин выделяются:

- экономические факторы, так как разработка собственного решения позволяет избежать регулярных лицензионных платежей, снизить затраты на внедрение и обучение персонала, а также обеспечивать гибкость в масштабировании системы без значительных дополнительных вложений;
- технические требования, связанные с возможностью собственного приложения интегрироваться с существующей ИТ-инфраструктурой компании, удовлетворять специфические запросы, возникающие при работе с локальными проектами и оптимизировать работу под имеющееся оборудование;
- организационные аспекты, обусловленные тем, что создание продукта внутри компании позволяет оперативно адаптировать его

- под меняющиеся условия, обеспечивать полный контроль над разработкой и исключить зависимость от сторонних поставщиков;
- особенности работы компании, поскольку локальная разработка учитывает специфику местного рынка, поддерживает текущие бизнес-процессы и соответствует корпоративным стандартам.

Указанные причины предоставляют предприятию возможность эффективно адаптироваться к изменениям рынка и максимально использовать потенциал новых технологий.

#### 1.4 Разработка модели бизнес-процесса «как должно быть»

Для устранения выявленных недостатков, была разработана модель «как должно быть» (TO-BE), представленная на рисунке 4. Основным отличием новой модели стало внедрение программного обеспечения дополненной реальности (AR) в бизнес-процесс «Работа над архитектурным проектом».



Рисунок 4 – Контекстная диаграмма TO-BE

На рисунке 5 представлена декомпозиция контекстной диаграммы TO-BE оформленной в нотации IDEF0, которая отражает оптимизированные процессы с использованием AR-технологий.



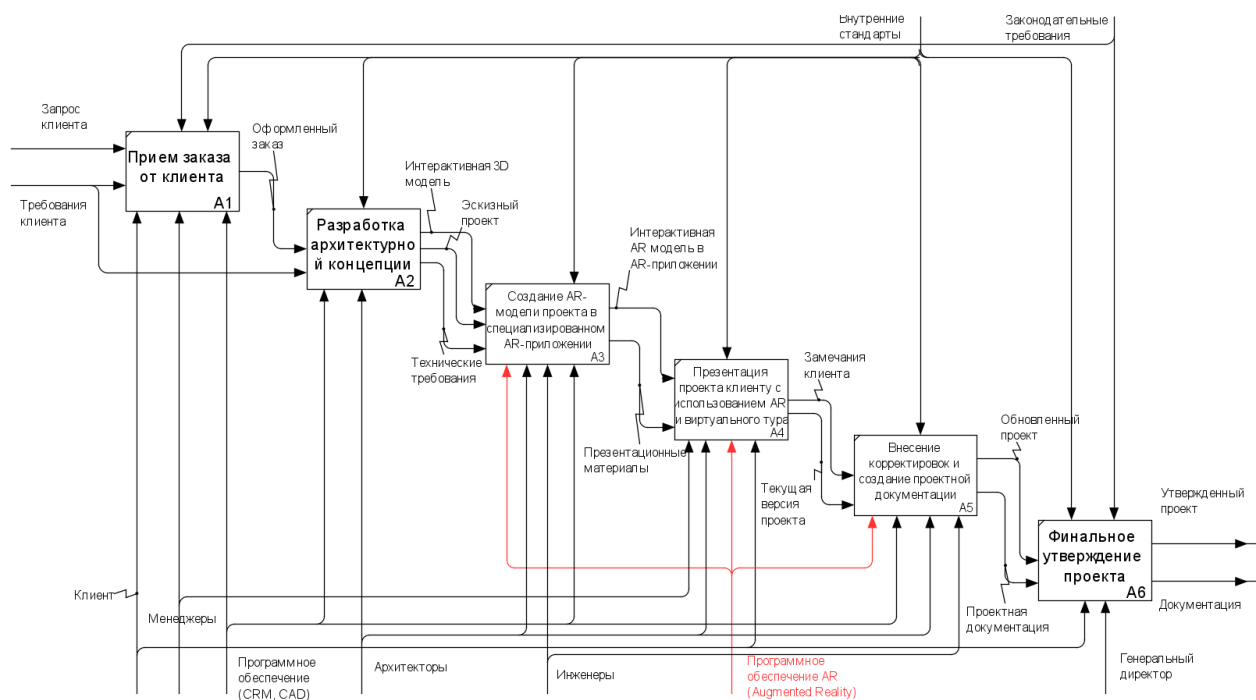


Рисунок 5 – Декомпозиция контекстной диаграммы TO-BE

Новая модель предусматривает следующие основные этапы работы над проектом:

- прием заказа от клиента;
- разработка архитектурной концепции;
- создание AR-модели проекта;
- презентация проекта с использованием дополненной реальности;
- внесение корректировок и создание проектной документации;
- финальное утверждение проекта.

Внедрение AR-технологий в бизнес-процессы компании обеспечивает комплексное решение выявленных проблем через следующие направления.

В области визуализации проектов технология обеспечивает:

- создание интерактивных AR-моделей;
- возможность просмотра проекта в реальном масштабе;
- наглядную демонстрацию архитектурных решений непосредственно в контексте окружающей среды.

Значительные улучшения достигаются в процессе согласования проектов за счет:

- сокращения времени на внесение изменений благодаря возможности корректировки проекта в реальном времени;
- уменьшения количества итераций согласования за счет более точного понимания проекта клиентом;
- возможности интерактивного обсуждения проекта с сравнительно быстрой визуализацией предлагаемых изменений.

В сфере коммуникации технология открывает новые возможности через:

- обеспечение удаленной работы с клиентами через мобильные устройства;
- повышение качества взаимодействия между всеми участниками проекта;
- возможность проведения виртуальных презентаций и согласований.

В результате процесс взаимодействия становится более прозрачным и эффективным, что положительно сказывается на сроках и качестве реализации проектов.

### **1.5 Разработка требований к программному обеспечению**

На основе проведенного анализа и разработанной модели TO-BE определены ключевые характеристики программного решения. Ниже в таблицах 2 и 3 представлена детальная спецификация технических и пользовательских требований к AR-приложению.

Функциональные требования описывают конкретные действия и функции, которые система должна выполнять, отвечая на вопрос «что» должна делать система. Нефункциональные (технические) требования, в свою очередь, определяют условия, при которых система должна функционировать. Процесс сбора требований охватывает извлечение, анализ, спецификацию и утверждение обоих типов требований, при этом важно не только учитывать

текущие потребности заинтересованных сторон, но и прогнозировать будущие изменения. [2]

Таблица 2 - Функциональные требования

Функциональная группа	Требование	Описание
Отображение 3D-моделей в режиме AR	Загрузка 3D-моделей	Возможность загружать и выводить 3D-модели в реальном окружении.
	Размещение моделей	Точное размещение объектов с учетом окружающего пространства.
	Реалистичное освещение	Освещение синхронизируется с реальными условиями.
	Привязка к точкам	Привязка модели к плоскостям или точкам реального мира.
	Детализация	Поддержка текстур и материалов высокого качества.
Масштабирование и вращение моделей	Управление масштабом	Масштабирование с помощью жестов (щипок).
	Вращение моделей	Вращение по осям через «свайп».
	Сброс параметров	Восстановление исходных настроек масштаба и ориентации.
	Навигация	Свободное перемещение внутри модели.
Переключение между режимами	Удобный интерфейс	Плавные переходы между режимами AR, 3D и виртуального тура.
	Форматы моделей	Совместимость с OBJ и FBX.
	Оптимизация моделей	Работа с облегченными версиями для слабых устройств.
Комментирование и пометки	Текстовые комментарии	Добавление заметок к элементам модели.

Таблица 3 - Нефункциональные требования

Категория	Требование	Описание
Производительность	Высокая производительность	Приложение работает без задержек на Android-устройствах.
Целевые платформы	Поддержка платформ	Приложение доступно для Android.
Безопасность	Защита данных	Конфиденциальные данные клиента и проектов защищены.
Масштабируемость	Расширяемость	Возможность добавления новых функций в будущем.

Данные требования напрямую решают выявленные проблемы в бизнес-процессах компании и создают техническую основу для внедрения новой модели работы.

#### Выводы по главе 1

В результате проведенного исследования была проанализирована специфика архитектурной отрасли и современных методов визуализации проектов. Анализ бизнес-процессов ООО «Киамет» выявил ряд существенных проблем, включая сложности в визуализации проектов, значительные временные затраты на внесение корректировок и отсутствие эффективных инструментов для удаленной работы с клиентами.

Изучение существующих программных решений на рынке показало, что доступные продукты либо не обеспечивают необходимый уровень функциональности, либо требуют значительных затрат на внедрение и сопровождение. Это обосновало целесообразность разработки собственного AR-приложения для компании. На основе проведенного анализа были сформированы функциональные и нефункциональные требования к разрабатываемому программному обеспечению, учитывающие специфику работы компании и современные тенденции в области AR-технологий.

### 2.1 Выбор методологии проектирования программного обеспечения

При разработке мобильного приложения для визуализации архитектурных проектов с использованием технологии AR ключевым этапом является выбор методологии проектирования программного обеспечения. Методология должна обеспечивать эффективное управление сложностью проекта, возможность детального моделирования взаимодействия с AR-технологиями и удобство документирования архитектурных решений приложения.

В качестве основной методологии проектирования был выбран универсальный язык моделирования UML (Unified Modeling Language).

Выбор UML обусловлен следующими факторами:

- диаграммы взаимодействия UML эффективно моделируют взаимодействие различных категорий пользователей с системой, помогая определить функциональные возможности и понять взаимосвязи между сценариями использования [7];
- UML обеспечивает детальное представление структуры классов и их взаимосвязей через диаграммы классов, демонстрирующие иерархические взаимоотношения в программе [7];
- диаграммы компонентов UML предоставляют широкие возможности для визуализации архитектуры системы, наглядно показывая структурные элементы и механизмы их взаимодействия [1];
- диаграммы последовательности UML позволяют наглядно отображать временную последовательность действий и сценарии поведения системы [7].

Для структурной организации приложения выбран архитектурный паттерн Model-View-Controller (MVC). Данный паттерн представляет собой эффективный подход к организации архитектуры приложения, основной

целью которого является разделение бизнес-логики и пользовательского интерфейса, что существенно упрощает процессы разработки, тестирования, масштабирования и поддержки приложения [19].

Сочетание методологии UML и паттерна MVC создает надежную основу для разработки, обеспечивая:

- четкую структуру проекта с явным разделением ответственности компонентов;
- наглядное представление архитектуры AR-приложения;
- эффективный процесс разработки и тестирования;
- возможность гибкого масштабирования системы;
- стандартизованную проектную документацию.

Выбранная методология позволила обеспечить необходимый уровень детализации всех аспектов системы и учесть специфику работы с дополненной реальностью.

## **2.2 Выбор технологий и инструментов разработки**

Разработка приложений дополненной реальности (AR) требует использования специализированных инструментов и платформ, которые обеспечивают функциональные возможности для создания интерактивного контента.

Для реализации архитектурных AR-проектов были выбраны два ключевых инструмента - платформа Vuforia Engine в сочетании с Unity. Этот выбор обусловлен их взаимодополняющими возможностями, так Unity предоставляет мощные инструменты для создания насыщенных и увлекательных эффектов дополненной реальности, которые интеллектуально взаимодействуют с реальным миром [15], а технология Vuforia Engine, интегрированная с Unity, значительно облегчает разработку AR-приложений [16].

Для обоснования данного выбора был проведен анализ существующих решений, в ходе которого прорабатывались популярные платформы для разработки AR-приложений такие как: Unity, Vuforia Engine, ARKit, Kudan, ARCore.

Unity - инструмент для разработки AR-приложений, который поддерживает работу множества AR-платформ и легко интегрируется с ними [6]. Unity позволяет разработчикам создавать сложные визуализации, используя мощные инструменты для трехмерного моделирования и анимации.

Vuforia Engine - одна из ведущих платформ для создания приложений дополненной реальности. Она поддерживает широкий набор устройств и предлагает инструменты для распознавания как 2D, так и 3D объектов, а также текста. Важной особенностью является высокая точность и стабильность отслеживания [9].

ARKit - технология, разработанная Apple для устройств iOS, предоставляющая мощные инструменты для создания высококачественных AR-приложений с высокой производительностью и стабильностью. Однако, она ограничена только экосистемой Apple [22].

Kudan - конкурент Vuforia Engine, специализирующийся на распознавании изображений и 3D-объектов. Эта платформа проста в использовании и интеграции с Unity, но имеет ограничения по функциональности в сравнении с более крупными платформами [23].

ARCore - это платформа дополненной реальности от Google, использующая технологию SLAM для точного размещения виртуальных объектов без маркеров. Она обеспечивает отслеживание движения, понимание геометрии окружающего пространства и оценку освещения. Однако ARCore имеет ограничения в поддержке маркерного трекинга и менее универсален для сложных сценариев промышленного применения [21].

Ключевые факторы, определившие выбор Vuforia Engine:

- функциональность, поскольку Vuforia Engine предоставляет все необходимые инструменты для архитектурной визуализации,

- включая поддержку 2D и 3D объектов. Её возможности распознавания текстур и детализированных объектов критически важны для точного отображения архитектурных элементов;
- кроссплатформенность, учитывая, что в отличие от ограниченных решений, таких как ARKit, Vuforia Engine поддерживает как iOS, так и Android. Это позволяет охватить широкий круг пользователей, что особенно важно для архитектурных проектов, где разные клиенты могут использовать различные устройства;
  - простота интеграции, поскольку Vuforia Engine легко интегрируется с Unity, что упрощает процесс разработки и открывает доступ к мощным возможностям игровой разработки. Это позволяет создавать высококачественные визуализации с минимальными затратами времени на интеграцию;
  - точность отслеживания, принимая во внимание что для архитектурных визуализаций важна высокая точность отслеживания объектов. Vuforia Engine предоставляет стабильные алгоритмы, что позволяет отображать виртуальные элементы с высокой степенью реализма;
  - облачное распознавание, благодаря которому в Vuforia Engine доступна возможность работы с большими объемами данных без увеличения размера приложения, что особенно полезно при работе с масштабными архитектурными проектами;
  - поддержка и документация, где обширная документация Vuforia Engine и активное сообщество разработчиков облегчают обучение и решение возникающих проблем, что важно для профессиональных архитектурных проектов.

Хотя Vuforia Engine не является бесплатной платформой, её возможности оправдывают инвестиции для профессиональных архитектурных проектов. Другие платформы, такие как ARKit и ARCore, предлагают высокую производительность, но их использование ограничено



определенными операционными системами. EasyAR и Kudan, несмотря на их простоту, не обладают всеми необходимыми функциями для сложных архитектурных визуализаций.

Таким образом, комбинация Vuforia Engine и Unity обеспечивает необходимую функциональность, кроссплатформенность и качество визуализации, удовлетворяя как требования разработчиков, так и конечных пользователей.

Как основной язык программирования выбран C# из-за его тесной интеграции с Unity и богатой экосистемы библиотек для работы с 3D-графикой. Git обеспечивает эффективное управление версиями и совместную работу над проектом.

В итоге технологический стек проекта включает:

- Unity 2022.3 LTS как основную среду разработки;
- Vuforia Engine для реализации функций дополненной реальности;
- C# в качестве языка программирования;
- систему контроля версий Git для управления кодовой базой.

Такой выбор технологий обеспечивает стабильность, масштабируемость и удобство разработки, позволяя эффективно реализовать ключевые функциональные возможности приложения.

### **2.3 Логическое моделирование программного обеспечения**

После выбора UML в качестве основной методологии проектирования было проведено логическое моделирование системы с использованием различных типов UML-диаграмм. Для полного описания архитектуры AR-приложения были разработаны диаграмма вариантов использования, диаграмма компонентов, диаграмма классов и диаграмма последовательности. Каждая из этих диаграмм описывает определенный аспект системы, что в совокупности дает полное представление о проектируемом программном обеспечении.

Первым этапом проектирования стала разработка диаграммы вариантов использования, представленная на рисунке 6, определяющая основную функциональность системы с точки зрения пользователя.

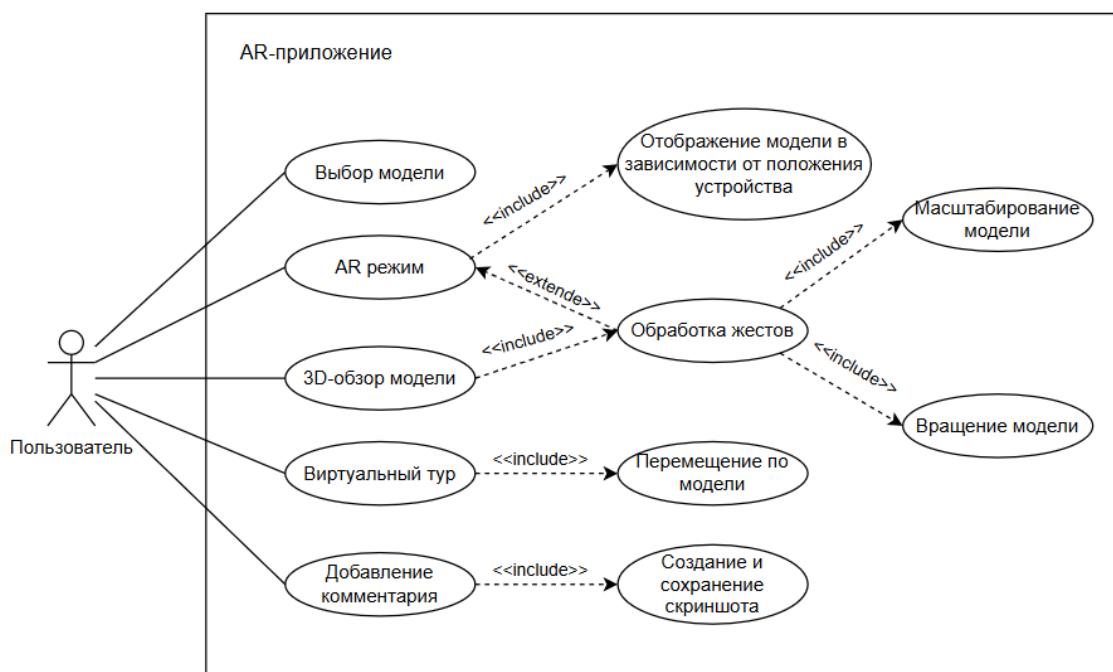


Рисунок 6 - Диаграмма вариантов использования AR-приложения

На представленной диаграмме вариантов использования центральным элементом выступает Пользователь, который имеет доступ к ключевым функциональным возможностям системы. Пользователь может начать работу с выбора модели - базовой операции, необходимой для дальнейшего взаимодействия с 3D-моделью. Система предлагает три режима работы: 3D-обзор модели, AR режим и Виртуальный тур. В 3D-обзоре модели пользователь может вращать и масштабировать объект с помощью жестов. AR режим обеспечивает отображение модели в зависимости от положения устройства относительно маркера, а при необходимости может быть дополнен обработкой жестов для манипуляции объектом. Виртуальный тур представляет собой отдельный режим, позволяющий пользователю перемещаться по модели

в виртуальном пространстве. Также пользователь может добавлять комментарии и фиксировать их скриншотами, которые автоматически сохраняются.

Для реализации описанной функциональности была определена архитектурная организация системы в виде диаграммы компонентов, представлено на рисунке 7, демонстрирующей разделение системы на три основных компонента согласно паттерну MVC.

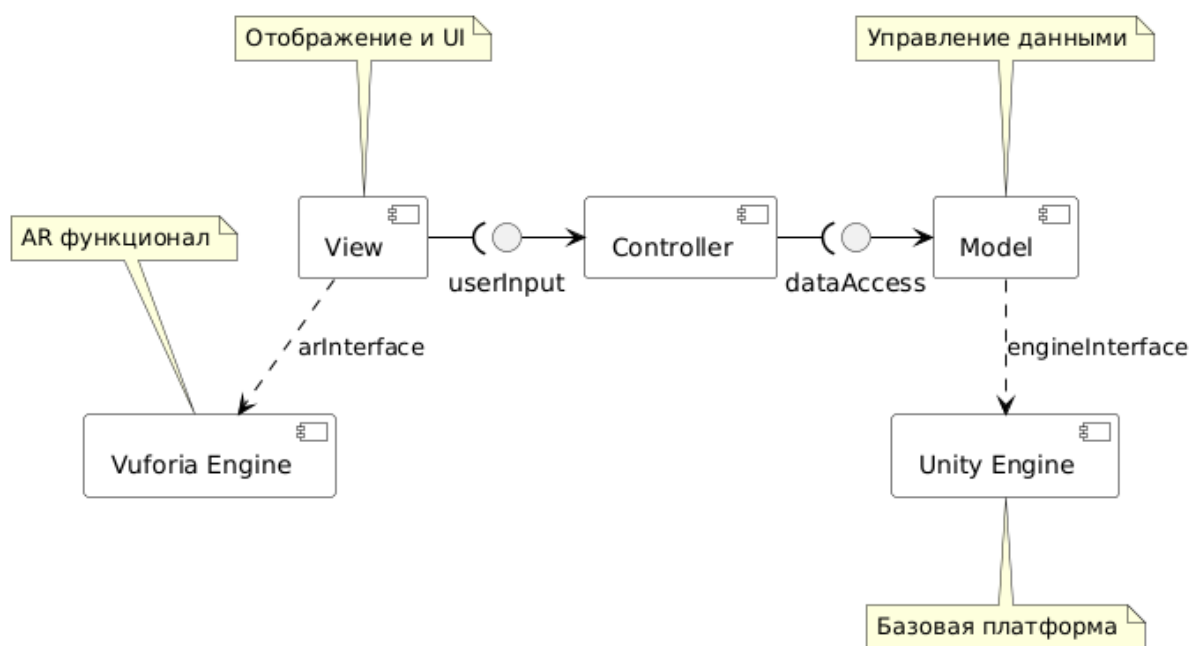


Рисунок 7 - Диаграмма компонентов AR-приложения

Основными компонентами разработанной архитектуры системы являются структурные элементы, реализующие ключевую функциональность приложения.

Model (Модель) - центральный компонент, отвечающий за данные и бизнес-логику системы. Основным классом является Model, содержащий структуру данных для хранения информации о моделях, их трансформациях и настройках режимов просмотра. Модель отвечает за хранение текущего состояния 3D-моделей, управление настройками различных режимов

просмотра и обновление данных при взаимодействии пользователя с моделями.

View (Представление) - компонент пользовательского интерфейса, включающий два ключевых класса. Класс View управляет основными UI элементами и обеспечивает базовое отображение моделей. Класс ARView специализированный класс для AR-визуализации, который взаимодействует с Vuforia Engine для обеспечения функциональности дополненной реальности.

Controller (Контроллер) - связующий компонент, обеспечивающий взаимодействие между Model и View. Включает три основных класса:

- класс Controller как основной контроллер приложения, координирующий работу всех компонентов;
- класс ModelInteractionController, обрабатывающий взаимодействие пользователя с моделями (масштабирование, вращение, перемещение);
- класс CommentScreenshotHandler, управляющий созданием и обработкой комментариев и скриншотов.

Система интегрируется с двумя ключевыми внешними компонентами - Unity и Vuforia Engine, обеспечивающими базовую функциональность для работы с 3D-графикой и дополненной реальностью соответственно.

Взаимодействие между компонентами организовано следующим образом:

- компонент Model хранит и обновляет данные о 3D-моделях и их состоянии;
- компонент View отвечает за отображение моделей и обработку пользовательского интерфейса, также он взаимодействует с Vuforia Engine для реализации AR-функциональности;
- компонент Controller обрабатывает действия пользователя, обновляет состояние модели и синхронизирует изменения с представлением;

- все компоненты используют функциональность Unity для базовых операций с 3D-графикой и обработки событий.

Такая организация компонентов обеспечивает четкое разделение ответственности согласно паттерну MVC, гибкость в реализации различных режимов отображения, эффективную интеграцию AR-функциональности, масштабируемость и поддерживаемость кодовой базы, а также возможность независимого тестирования компонентов.

Для детализации внутренней структуры системы была разработана диаграмма классов, представленная на рисунке 8, которая раскрывает взаимосвязи между классами внутри каждого компонента.

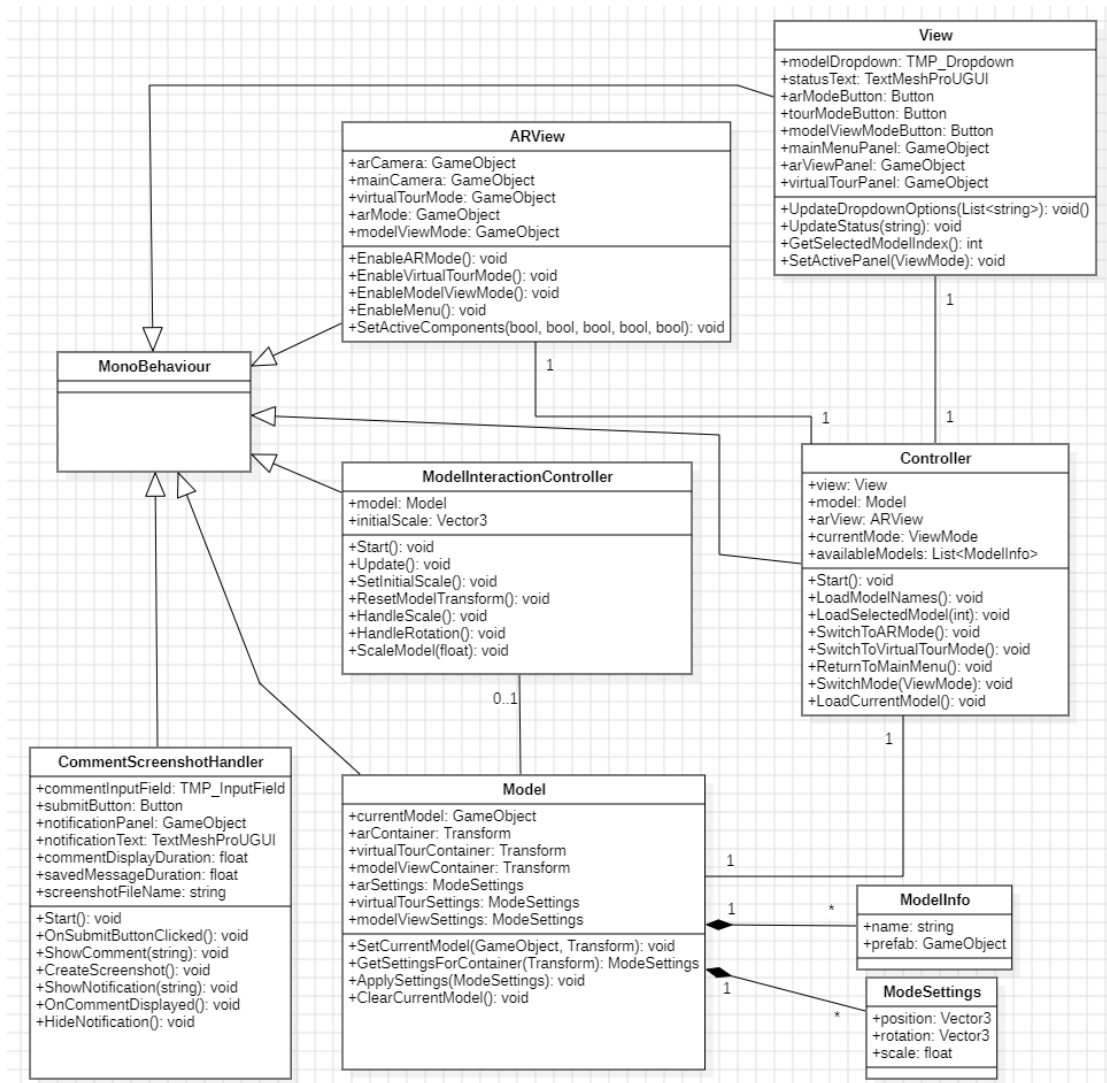


Рисунок 8 - Диаграмма классов AR-приложения

Диаграмма классов отражает архитектуру системы, построенную по паттерну MVC. Система состоит из шести основных и двух вложенных классов, связанных между собой. Все основные классы наследуют MonoBehaviour, что позволяет им функционировать как компоненты в Unity сцене и использовать его стандартные методы.

Класс Model является центральным элементом системы. Он содержит ссылку на текущую модель (currentModel), различные контейнеры трансформаций (arContainer, virtualTourContainer, modelViewContainer) и настройки режимов работы (arSettings, virtualTourSettings, modelViewSettings). Класс Model имеет вложенные классы ModelInfo и ModeSettings, которые хранят базовую информацию о 3D-моделях.

Класс Controller управляет взаимодействием между View, ARView и Model. Содержит ссылки на объекты View, ARView и Model, список доступных моделей (availableModels). Предоставляет методы для загрузки выбранной модели (LoadModel), переключения между режимами просмотра (SwitchToARMode, SwitchToVirtualTourMode, SwitchToModelViewMode) и возврата в главное меню (ReturnToMenu).

Класс ModelInteractionController отвечает за обработку взаимодействия пользователя с моделью. Реализует функции масштабирования (Scale), вращения (Rotate) и сброса трансформаций модели (ResetTransformations).

Класс CommentScreenshotHandler управляет функциональностью комментариев и скриншотов. Содержит UI элементы для ввода комментариев (commentInput), кнопку подтверждения (confirmButton) и область уведомлений (notificationArea).

Класс View отвечает за визуальное представление. Включает элементы пользовательского интерфейса: выпадающий список моделей (modelDropdown), кнопки переключения режимов (modeButtons) и текстовые поля статуса (statusText).

Класс ARView специализируется на работе с дополненной реальностью. Содержит необходимые игровые объекты (arCamera, viewModes) и методы для управления различными режимами работы с дополненной реальностью.

Классы взаимодействуют через систему связей:

- класс Controller напрямую связан с Model и View, координируя их работу (загружает модели, обновляет интерфейс);
- класс ModelInteractionController и CommentScreenshotHandler используют Model для обновления и сохранения данных (например, изменение размеров модели, добавление комментариев);
- класс ARView расширяет логику, заложенную в View, добавляя функциональность дополненной реальности (управляет камерами, режимами AR);
- все классы при необходимости получают доступ к ModelInfo, содержащему информацию о доступных 3D-моделях.

Для понимания динамического взаимодействия описанных компонентов была разработана диаграмма последовательности, показанная на рисунке 9.

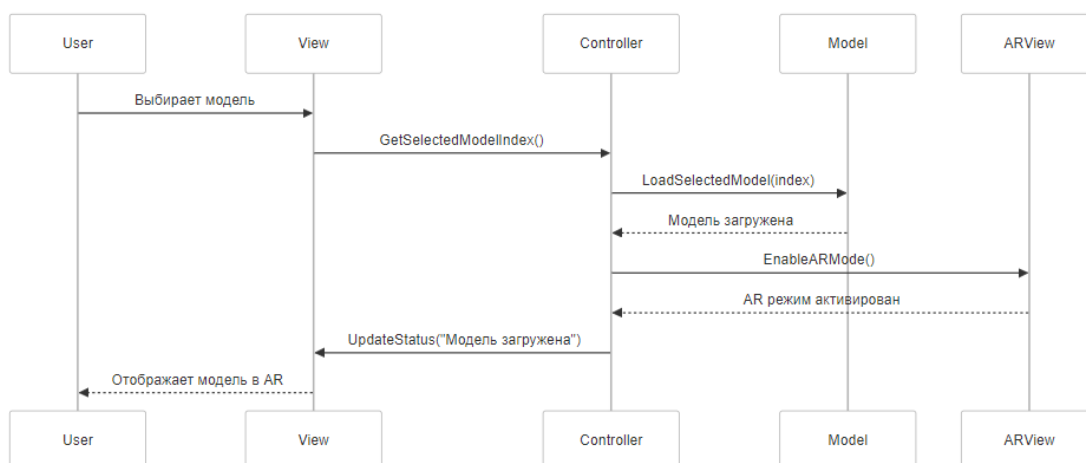


Рисунок 9 - Диаграмма последовательности взаимодействия компонентов AR-приложения

Специфика взаимодействия в AR-контексте основывается на двух ключевых компонентах. Класс ARView осуществляет тесное взаимодействие с Vuforia Engine для выполнения таких критических функций, как распознавание маркеров, отслеживание текущего положения камеры и корректное размещение 3D-моделей в пространстве. В свою очередь, ModelInteractionController берет на себя обработку различных пользовательских жестов, включая масштабирование с помощью pinch-жестов, вращение посредством swipe-движений, а также точное позиционирование моделей в пространстве.

Обработка событий в системе охватывает несколько ключевых аспектов: обработку пользовательского ввода через элементы интерфейса, управление AR-событиями, поступающими от Vuforia Engine, обработку системных событий Unity, а также отслеживание и реагирование на события, связанные с изменением состояния модели.

Интеграция с внешними системами реализуется через два основных компонента. Vuforia Engine интегрируется через класс ARView и отвечает за обработку функциональности дополненной реальности, трекинг и распознавание объектов. Unity Engine выступает в качестве базовой платформы разработки, обеспечивая рендеринг 3D-графики, физические расчеты и эффективное управление ресурсами системы. Такая архитектурная организация позволяет в полной мере реализовать все преимущества паттерна MVC.

## **2.4 Моделирование данных системы**

Разработанные в предыдущих разделах UML-диаграммы определили ключевые требования к организации данных в системе. Диаграмма классов выявила необходимость в создании четкой структуры для хранения информации о 3D-моделях и их метаданных. Особенно важным аспектом



стало проектирование класса `ModelInfo`, который должен обеспечивать эффективную загрузку моделей для AR-визуализации.

Основные структуры данных включают:

- модель (`Model.ModelInfo`), содержит название (`string name`) и префаб модели (`GameObject prefab`);
- настройки режима (`Model.ModeSettings`), содержит позиция (`Vector3 position`), вращение (`Vector3 rotation`), масштаб (`float scale`);
- комментарий (`CommentScreenshotHandler`), содержит текст комментария (`string`), время создания (`DateTime`);
- скриншот, содержит имя файла (`string`, `format: «Screenshot_yyyymmdd_HHmms.png»`).

Система хранения данных организована следующим образом:

- 3d-модели хранятся в виде префабов в проекте Unity;
- настройки режимов содержатся в памяти в экземпляре класса `Model`;
- скриншоты сохраняются как файлы изображений в файловой системе устройства;
- комментарии отображаются временно без долгосрочного хранения.

Базовая валидация данных осуществляется при загрузке моделей через встроенные механизмы Unity, проверяющие корректность загружаемых данных. Пользовательский ввод проверяется на наличие недопустимых символов и соответствие ограничениям по длине.

Процесс обработки данных организован как последовательность операций:

- загрузка и инициализация модели;
- обработка пользовательских действий (масштабирование, вращение);
- создание и сохранение пользовательского контента (комментарии, скриншоты).

Такой подход позволяет минимизировать ошибки на этапе ввода и обработки данных, обеспечивая стабильную работу приложения. Дополнительные механизмы логирования и обработки исключений помогают быстро выявлять и исправлять возможные проблемы.

## 2.5 Проектирование пользовательского интерфейса

На основе анализа требований, определенных в главе 1, были разработаны принципы организации пользовательского интерфейса, реализованные с использованием архитектуры MVC и разработанной модели данных.

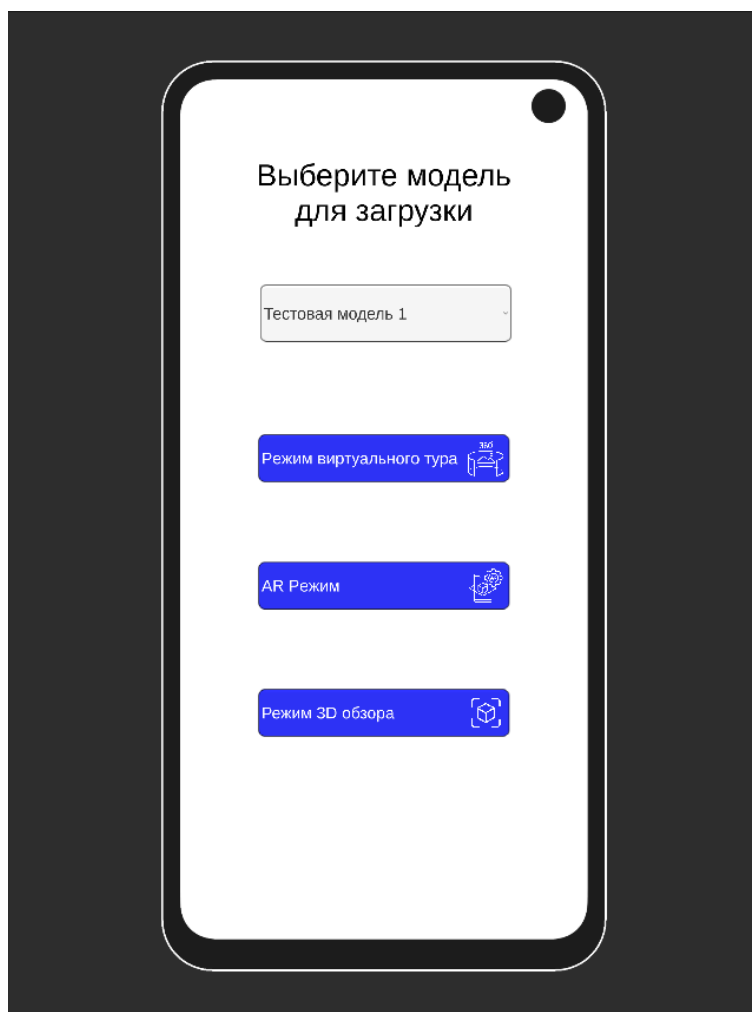


Рисунок 10 - Интерфейс главного меню AR-приложения

Структура интерфейса включает два основных экрана. Первый экран представляет собой главное меню, показанное на рисунке 10. В состав главного меню AR-приложения входит выпадающий список для выбора модели, набор кнопок для переключения между режимами работы (AR (обнаружения маркеров), виртуальный тур и 3D-обзор модели), а также система всплывающих сообщений, информирующих пользователя о выполненных действиях.

Интерфейс режимов отображения, представленный на рисунке 11, является вторым ключевым экраном приложения.

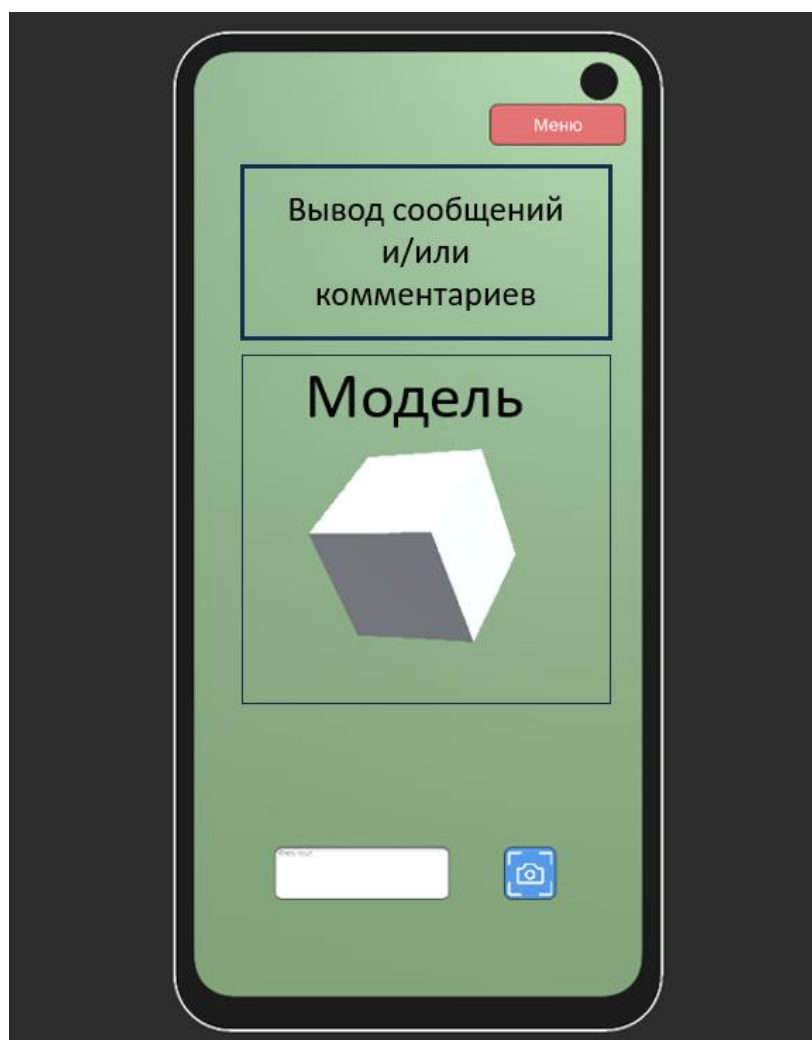


Рисунок 11 - Интерфейс режимов отображения AR-приложения

Интерфейс режимов отображения AR-приложения содержит:

- всплывающие сообщения о выполненных действиях/комментариях;
- кнопка создания комментария/скриншота;
- поле ввода текста;
- кнопка возврата в меню.

Интерфейс реализован с учетом следующих принципов:

- единообразие элементов управления во всех режимах;
- интуитивно понятное размещение элементов;
- мгновенная обратная связь на действия пользователя;
- корректное отображение на мобильных устройствах.

Таким образом, разработанный интерфейс режимов отображения AR-приложения гарантирует максимально позитивный пользовательский опыт.

## Выводы по главе 2

Во второй главе было выполнено проектирование программного обеспечения мобильного AR-приложения для визуализации архитектурных проектов. В качестве методологии проектирования был выбран язык UML, а в качестве архитектурного паттерна - MVC. В процессе работы был обоснован и осуществлен выбор технологического стека, включающий Unity как среду разработки, Vuforia Engine для реализации функций дополненной реальности и C# в качестве языка программирования. Представлена логическая модель системы в виде UML-диаграмм: вариантов использования, компонентов, классов и последовательности. Разработана модель данных и спроектирован пользовательский интерфейс с учетом специфики работы с AR-технологиями. Все разработанные в данной главе модели и диаграммы являются основой для последующей реализации мобильного приложения.

## Глава 3 Реализация и тестирование программного обеспечения

### 3.1 Реализация AR-приложения

#### 3.1.1 Реализация режима виртуального тура

Режим виртуального тура является ключевой функциональностью приложения, предназначенной для исследования архитектурных объектов в реальном масштабе. В отличие от статичного просмотра, этот режим позволяет пользователю физически перемещаться вокруг и внутри виртуальной модели с помощью мобильного устройства, что создает эффект присутствия и помогает лучше понять пространственные характеристики проекта.

Реализация режима виртуального тура основана на взаимодействии нескольких ключевых компонентов системы, организованных согласно паттерну MVC. Как показано на рисунке 12, основная логика переключения режимов реализована в классе ARView, который управляет активацией и деактивацией необходимых компонентов камеры и режимов отображения.

```
Ссылка: 0
public void EnableARMode() ...
Ссылка: 0
public void EnableVirtualTourMode()
{
    SetActiveComponents(true, false, true, false, false);
}
Ссылка: 0
public void EnableModelViewMode() ...
Ссылка: 0
public void EnableMenu() ...
Ссылка: 4
private void SetActiveComponents(bool arCameraActive, bool mainCameraActive,
bool virtualTourModeActive, bool modelViewModeActive, bool arModeActive)
{
    arCamera.SetActive(arCameraActive);
    mainCamera.SetActive(mainCameraActive);
    virtualTourMode.SetActive(virtualTourModeActive);
    modelViewMode.SetActive(modelViewModeActive);
    arMode.SetActive(arModeActive);
}
```

Рисунок 12 - Реализация методов управления режимами просмотра

При активации режима виртуального тура система автоматически включает камеру устройства в режиме дополненной реальности и специализированные компоненты, необходимые для корректного отслеживания перемещений пользователя в пространстве. Одновременно с этим происходит деактивация компонентов других режимов, что обеспечивает стабильную работу приложения и оптимальное использование ресурсов устройства.

Важную роль в работе виртуального тура играет класс Model, отвечающий за управление трехмерными моделями. Этот класс обеспечивает корректное позиционирование и масштабирование объектов в пространстве, что критически важно для создания реалистичного опыта взаимодействия.

```
Ссылка: 1
public ModeSettings virtualTourSettings = new ModeSettings
{
    position = new Vector3(0, 0, 0),
    rotation = new Vector3(0, 0, 0),
    scale = 1f
};

Ссылка: 1
> public ModeSettings modelViewSettings = new ModeSettings ...

Ссылка: 0
public void SetCurrentModel(GameObject model, Transform container)
{
    ClearCurrentModel();
    currentModel = model;
    currentModel.transform.SetParent(container, false);

    ModeSettings settings = GetSettingsForContainer(container);
    ApplySettings(settings);
}
```

Рисунок 13 - Структура настроек виртуального тура

На рисунке 13 представлена реализация настроек виртуального тура через вложенный класс ModeSettings, определяющий начальные параметры

отображения модели. Особое внимание уделено масштабу, который по умолчанию установлен в значение 1, что обеспечивает отображение объектов в их реальных размерах. Это решение имеет принципиальное значение для создания достоверного представления о пространственных характеристиках проекта.

Процесс переключения в режим виртуального тура, представленный на рисунке 14, координируется контроллером, который обеспечивает плавный переход между различными состояниями приложения.

```
public void SwitchToVirtualTourMode()
{
    SwitchMode(ViewMode.VirtualTour);
    arView.EnableVirtualTourMode();
    LoadCurrentModel();
}
```

Рисунок 14 - Реализация переключения режимов просмотра в режим виртуального тура

При активации режима виртуального тура происходит последовательное выполнение нескольких важных операций. Сначала устанавливается соответствующий режим отображения через перечисление ViewMode, затем активируются необходимые компоненты AR-системы, и наконец, производится загрузка текущей модели с применением специфических настроек виртуального тура. Такая последовательность действий обеспечивает корректную инициализацию всех компонентов и стабильную работу режима.

Результат работы режима виртуального тура на мобильном устройстве продемонстрирован на рисунке 15.

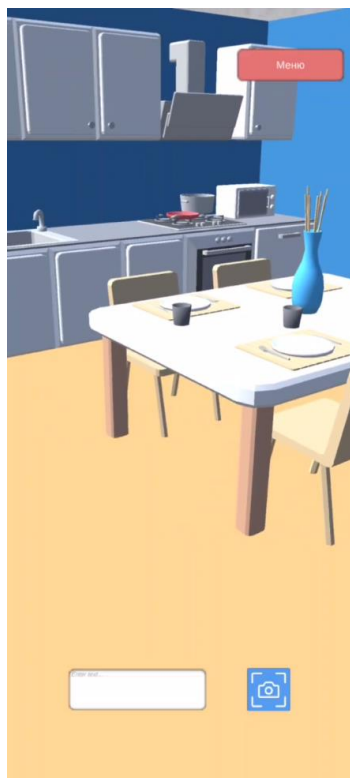


Рисунок 15 - Демонстрация работы приложения в режиме виртуального тура на мобильном устройстве Redmi Note 11

Выбранная архитектура приложения, основанная на паттерне MVC, обеспечивает не только четкое разделение ответственности между компонентами, но и создает основу для дальнейшего развития функциональности. Модульная структура позволяет легко добавлять новые возможности и модифицировать существующие без необходимости существенного изменения базового кода.

### **3.1.2 Реализация режима AR**

Режим AR в приложении реализует подход к визуализации архитектурных проектов через привязку 3D-моделей к физическим маркерам с использованием технологии Vuforia Image Target. Процесс начинается с создания базы маркеров в Vuforia Target Manager, где каждому изображению присваиваются уникальные параметры распознавания, что позволяет системе идентифицировать маркеры в реальном времени. При обнаружении маркера устройством 3D-модель автоматически устанавливается в заданном масштабе



и положении, а при выходе маркера из поля зрения модель скрывается, обеспечивая интуитивное и динамичное взаимодействие с архитектурным проектом [16].

Управление режимом AR осуществляется через класс ARView, который координирует работу камеры устройства в режиме дополненной реальности и соответствующих компонентов визуализации.

Переключение в режим AR инициируется через контроллер, представленный на рисунке 16, который обеспечивает корректную загрузку и позиционирование модели.

```
public void SwitchToARMode()
{
    SwitchMode(ViewMode.AR);
    arView.EnableARMode();
    LoadCurrentModel();
}
```

Рисунок 16 - Реализация переключения режимов просмотра в режим AR

При загрузке модели применяются специальные настройки позиционирования и масштабирования, представленные на рисунке 17, определенные в классе Model.

```
public ModeSettings arSettings = new ModeSettings
{
    position = new Vector3(0, 0, 0),
    rotation = new Vector3(0, 0, 0),
    scale = 1f
};
```

Рисунок 17 - Структура настроек режима AR

Взаимодействие с моделью в режиме AR контролируется через класс `ModelInteractionController`, который обрабатывает жесты пользователя для масштабирования и вращения объекта. Масштабирование модели осуществляется через обработку мультитач жестов, представленных на рисунке 18, а также посредством приближения и отдаления мобильного устройства к маркеру.

```
Ссылка: 1
private void HandleScale()
{
    if (Input.touchCount == 2)
    {
        Touch touch0 = Input.GetTouch(0);
        Touch touch1 = Input.GetTouch(1);
        Vector2 touch0PrevPos = touch0.position - touch0.deltaPosition;
        Vector2 touch1PrevPos = touch1.position - touch1.deltaPosition;

        float prevMagnitude = (touch0PrevPos - touch1PrevPos).magnitude;
        float currentMagnitude = (touch0.position - touch1.position).magnitude;

        float difference = currentMagnitude - prevMagnitude;
        ScaleModel(difference * 0.01f);
    }
}

Ссылка: 1
private void HandleRotation() ...

Ссылка: 1
private void ScaleModel(float scaleFactor)
{
    Vector3 newScale = model.currentModel.transform.localScale + Vector3.one * scaleFactor;
    float minScale = 0.1f;

    newScale.x = Mathf.Max(newScale.x, minScale);
    newScale.y = Mathf.Max(newScale.y, minScale);
    newScale.z = Mathf.Max(newScale.z, minScale);

    model.currentModel.transform.localScale = newScale;
}
```

Рисунок 18 - Реализация масштабирования объектов

Для обеспечения плавного вращения модели реализована обработка одиночных касаний, представленных на рисунке 19.

```
private void HandleRotation()
{
    if (Input.touchCount == 1)
    {
        Touch touch = Input.GetTouch(0);
        if (touch.phase == TouchPhase.Moved)
        {
            Vector2 deltaPosition = touch.deltaPosition;
            model.currentModel.transform.Rotate(Vector3.up, -deltaPosition.x * 0.081f, Space.World);
            model.currentModel.transform.Rotate(Vector3.right, deltaPosition.y * 0.081f, Space.World);
        }
    }
}
```

Рисунок 19 - Реализация вращения объектов

Режим AR также интегрирован с системой управления состоянием приложения через enum ViewMode и соответствующие методы переключения в контроллере. Это обеспечивает плавный переход между различными режимами просмотра и сохранение состояния модели при смене режимов.

Практическая реализация режима AR показана на рисунке 20.

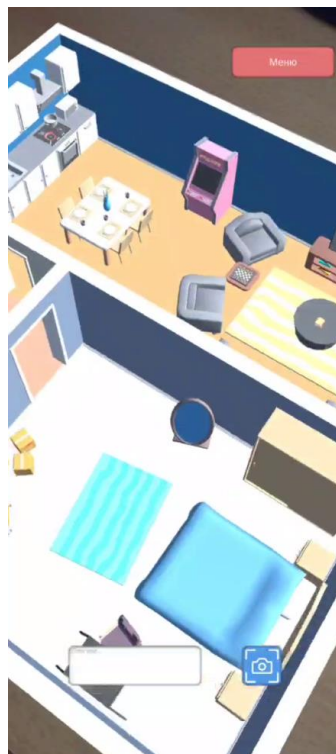


Рисунок 20 - Визуализация архитектурного объекта в режиме AR на мобильном устройстве Samsung Galaxy A35

Режим AR позволяет легко привязывать 3D-модели к физическим изображениям, что делает процесс визуализации интуитивно понятным и удобным. Гибкость этого режима позволяет использовать его как в офисе, так и на строительной площадке, обеспечивая быстрый доступ к цифровым макетам без необходимости сложной настройки.

### 3.1.3 Реализация режима 3D-обзора моделей

Режим 3D-обзора моделей предоставляет пользователям возможность детального изучения архитектурных объектов в контролируемой среде. В отличие от режимов AR и виртуального тура, этот режим фокусируется на предоставлении удобного интерфейса для манипуляции с 3D-моделью без привязки к реальному пространству.

Настройки позиционирования и масштабирования модели для всех режимов в среде разработки Unity представлены на рисунке 21.

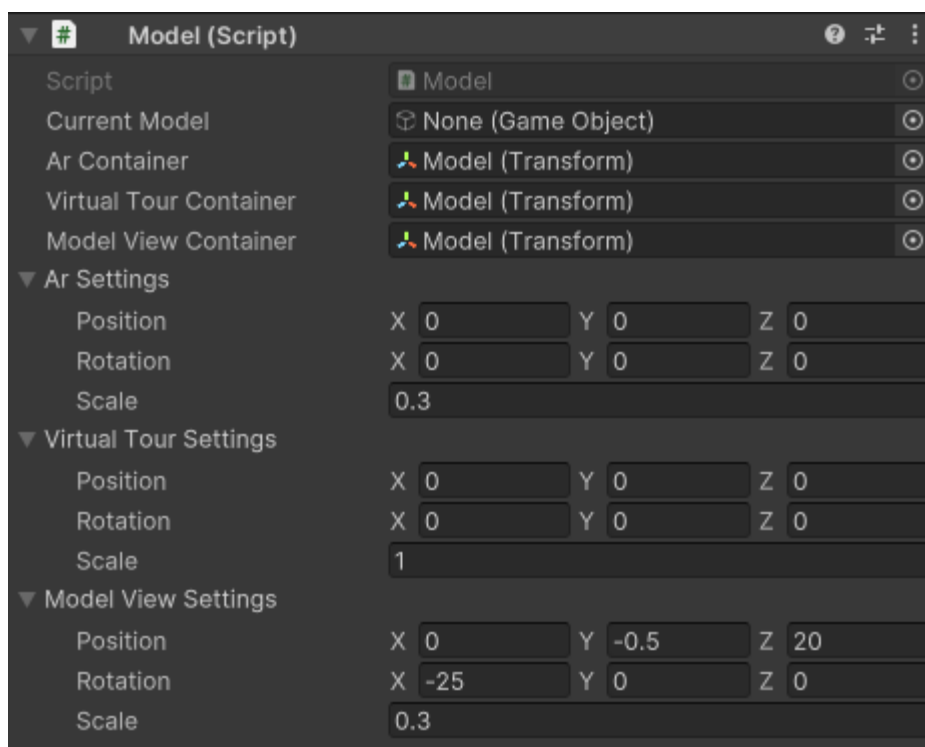


Рисунок 21 - Реализация настроек режимов в Unity

Реализация режима основана на взаимодействии нескольких компонентов: `ModelInteractionController` для обработки пользовательского ввода, `Model` для управления состоянием модели и `Controller` для координации работы всей системы. При переключении в режим 3D-обзора система активирует соответствующие компоненты через класс `ARView` при помощи метода `EnableModelViewMode`, представленного на рисунке 22.

```
Ссылка: 0
public void EnableModelViewMode()
{
    SetActiveComponents(false, true, false, true, false);
}
```

Рисунок 22 - Реализация активации режима 3D-обзора моделей

Как видно из рисунка 23, инициализация режима происходит через контроллер, который обеспечивает настройку окружения и загрузку модели.

```
Ссылка: 0
public void SwitchToModelViewMode()
{
    SwitchMode(ViewMode.ModelView);
    arView.EnableModelViewMode();
    LoadCurrentModel();
}
```

Рисунок 23 - Реализация переключения в режим 3D-обзора

При загрузке модели применяются специальные базовые настройки, оптимизированные для режима 3D-обзора, представленные на рисунке 24.

```

public ModeSettings modelViewSettings = new ModeSettings
{
    position = new Vector3(0, -0.5f, 0),
    rotation = new Vector3(-25f, 0f, 0f),
    scale = 0.3f
};

```

Рисунок 24 - Настройки режима 3D-обзора

Эти настройки обеспечивают оптимальное начальное положение модели для удобного обзора. Модель размещается немного ниже центра экрана и под углом, что позволяет лучше воспринимать её объемные характеристики. При необходимости пользователь может изменить эти настройки в редакторе Unity, показанном на рисунке 21.

Взаимодействие с моделью реализовано через класс `ModelInteractionController`, который обрабатывает различные типы пользовательского ввода. Масштабирование и вращение модели осуществляется через методы `HandleScale` и `HandleRotation`, представленные на рисунках 18 и 19.

Важной особенностью режима является возможность сброса трансформации модели к исходному состоянию, как показано на рисунке 25.

```

public void ResetModelTransform()
{
    SetInitialScale();
    if (model.currentModel != null)
    {
        model.currentModel.transform.localScale = initialScale;
        model.currentModel.transform.rotation = Quaternion.identity;
    }
}

```

Рисунок 25 - Реализация сброса трансформации модели

Загрузка моделей осуществляется через систему управления ресурсами, реализованную в контроллере. При выборе новой модели происходит корректная очистка текущей модели и загрузка новой с применением соответствующих настроек, показанных на рисунке 26.

```
public void SetCurrentModel(GameObject model, Transform container)
{
    ClearCurrentModel();
    currentModel = model;
    currentModel.transform.SetParent(container, false);

    ModeSettings settings = GetSettingsForContainer(container);
    ApplySettings(settings);
}
```

Рисунок 26 - Реализация загрузки моделей

Интерфейс режима 3D-обзора с элементами управления на мобильном устройстве представлен на рисунке 27.



Рисунок 27 - Интерфейс режима 3D-обзора на мобильном устройстве Redmi 10

Режим 3D-обзора моделей тесно интегрирован с системой пользовательского интерфейса через класс View, который управляет отображением соответствующих панелей и элементов управления. Это обеспечивает удобный доступ к функциям масштабирования, вращения и сброса трансформации модели.

Режим 3D-обзора даёт пользователям полный контроль над моделью, позволяя свободно вращать, масштабировать и изучать её с любого ракурса. Это делает его удобным инструментом для детального анализа архитектурных решений и обсуждения проектных деталей. В отличие от режима дополненной реальности, 3D-обзор не привязан к окружающему пространству, что позволяет сосредоточиться на модели вне зависимости от условий использования. Такой режим особенно полезен для предварительного изучения концепции, проработки деталей и подготовки к презентации.

#### **3.1.4 Реализация обработки пользовательских комментариев и создания скриншотов**

В процессе работы над архитектурными проектами часто возникает необходимость фиксации определенных ракурсов модели и добавления к ним комментариев для последующего обсуждения или внесения изменений. Для решения этой задачи в приложении реализован функционал создания скриншотов и добавления к ним текстовых комментариев через класс CommentScreenshotHandler.

Как показано на рисунке 28, основная логика обработки комментариев и создания скриншотов реализована в виде отдельного компонента, который инициализируется при запуске приложения.



```

public class CommentScreenshotHandler : MonoBehaviour
{
    Ссылка: 2
    public TMP_InputField commentInputField;
    Ссылка: 1
    public Button submitButton;
    Ссылка: 4
    public GameObject notificationPanel;
    Ссылка: 3
    public TextMeshProUGUI notificationText;
    Ссылка: 1
    public float commentDisplayDuration = 2f;
    Ссылка: 1
    public float savedMessageDuration = 1f;

    Ссылка: 3
    private string screenshotFileName;

    Ссылка: 0 | @Unity Message
    void Start()
    {
        submitButton.onClick.AddListener(OnSubmitButtonClicked);
        notificationPanel.SetActive(false);
    }
}

```

Рисунок 28 - Реализация обработчика комментариев и скриншотов

Система обработки комментариев, представленная на рисунке 29, реализует проверку валидности вводимых данных и обеспечивает обратную связь с пользователем через панель уведомлений. При добавлении комментария происходит его валидация и отображение в интерфейсе:

```

public void OnSubmitButtonClicked()
{
    string commentText = commentInputField.text;

    if (!string.IsNullOrEmpty(commentText))
    {
        Debug.Log("Комментарий добавлен: " + commentText);
        commentInputField.text = "";
        ShowComment(commentText);
    }
    else
    {
        ShowNotification("Комментарий не может быть пустым.");
    }
}

// Метод для показа комментария
Ссылка: 1
private void ShowComment(string commentText)
{
    notificationText.text = "Комментарий: " + commentText;
    notificationPanel.SetActive(true);
    Invoke(nameof(OnCommentDisplayed), commentDisplayDuration);
}
}

```

Рисунок 29 - Реализация валидации комментариев

После успешного добавления комментария система автоматически создает скриншот текущего состояния сцены. Это позволяет сохранить визуальный контекст комментария для последующего анализа. На рисунке 30 показана реализация функционала создания и сохранения скриншотов.

```
private void CreateScreenshot()
{
    screenshotFileName = "Screenshot_" + System.DateTime.Now.ToString("yyyyMMdd_HHmss") + ".png";
    ScreenCapture.CaptureScreenshot(screenshotFileName);
    Debug.Log("Скриншот сохранён: " + screenshotFileName);
}
Ссылка: 1
private void OnCommentDisplayed()
{
    CreateScreenshot();
    ShowNotification("Скриншот создан и комментарий сохранён.");
}
```

Рисунок 30 - Реализация создания скриншотов

Для обеспечения комфортной работы пользователя реализована система уведомлений, представленная на рисунке 31, которая информирует о результатах выполнения операций. Уведомления автоматически скрываются через заданный промежуток времени:

```
// Метод для показа сообщения о сохранении
Ссылка: 2
private void ShowNotification(string message)
{
    notificationText.text = message;
    notificationPanel.SetActive(true);

    // Скрываем уведомление через секунду
    Invoke(nameof(HideNotification), savedMessageDuration);
}

// Метод для скрытия уведомления
Ссылка: 1
private void HideNotification()
{
    notificationPanel.SetActive(false);
    notificationText.text = "";
}
```

Рисунок 31 - Реализация системы уведомлений

Система хранения комментариев и скриншотов организована таким образом, чтобы обеспечить удобный доступ к сохраненным данным. Каждый скриншот именуется с использованием текущей даты и времени, что упрощает их последующую идентификацию и сортировку. Файлы сохраняются в локальном хранилище устройства, что позволяет использовать их даже при отсутствии подключения к сети.

Интеграция системы комментариев и скриншотов с основным функционалом приложения осуществляется через контроллер, который обеспечивает доступ к этим возможностям из любого режима просмотра: AR, виртуального тура или 3D-обзора. Это позволяет пользователям фиксировать важные аспекты проекта независимо от текущего режима работы.

Важной особенностью реализации является асинхронная обработка операций создания скриншотов и сохранения комментариев, что обеспечивает плавную работу интерфейса даже при выполнении ресурсоемких операций. Система также предусматривает обработку возможных ошибок и информирование пользователя о проблемах в процессе сохранения данных.

## **3.2 Тестирование программного обеспечения**

### **3.2.1 Тестирование пользовательского интерфейса**

Тестирование пользовательского интерфейса является критически важным этапом в разработке AR-приложения для визуализации архитектурных проектов, поскольку от удобства и надежности интерфейса напрямую зависит эффективность работы пользователей с виртуальными моделями.

Для проверки корректности работы программного обеспечения был выбран метод функционального тестирования по принципу «черного ящика». Данный метод позволяет проверить соответствие программы заявленным требованиям путем проверки правильности выходных данных при различных входных параметрах без анализа внутренней структуры программы [3]. Этот

подход особенно эффективен для тестирования пользовательского интерфейса и взаимодействия компонентов системы, так как позволяет оценить работу приложения с точки зрения конечного пользователя. В рамках тестирования были проверены все основные функциональные возможности приложения, включая корректность работы различных режимов просмотра, обработку пользовательского ввода и стабильность работы AR-компонентов.

Особое внимание было уделено тестированию системы переключения между различными режимами просмотра. В таблице 4 представлены результаты тестирования основных методов переключения режимов.

Таблица 4 - Тестирование методов переключения режимов просмотра

Метод	Тестируемая функциональность	Результат проверки
SwitchToARMode()	Активация режима AR и соответствующей панели	Успешно
SwitchToVirtualTourMode()	Переключение в режим виртуального тура	Успешно
SwitchToModelViewMode()	Активация режима просмотра 3D-модели	Успешно
ReturnToMainMenu()	Возврат в главное меню и очистка текущей модели	Успешно

В ходе тестирования выпадающего списка моделей (modelDropdown) были проверены различные сценарии работы, результаты которых представлены в таблице 5.

Таблица 5 - Тестирование компонента выбора моделей

Сценарий тестирования	Ожидаемый результат	Фактический результат
Инициализация списка моделей	Корректная загрузка всех доступных моделей	Соответствует
Выбор модели из списка	Загрузка выбранной модели и обновление statusText	Соответствует

При тестировании компонента CommentScreenshotHandler были проверены основные функции работы с комментариями и скриншотами. Результаты представлены в таблице 6.

Таблица 6 - Тестирование системы комментариев и скриншотов

Функция	Параметры проверки	Результат тестирования
Валидация комментариев	Проверка пустых комментариев и спецсимволов	Успешная блокировка недопустимого ввода
Система уведомлений	Корректность отображения и автоматического скрывания	Работает согласно заданным таймингам
Создание скриншотов	Формирование имен файлов и сохранение изображений	Корректное сохранение в хранилище
Обработка частых запросов	Устойчивость при многократном создании скриншотов	Стабильная работа без утечек памяти

В рамках тестирования ModelInteractionController была проведена проверка всех реализованных жестов взаимодействия с моделью. Результаты представлены в таблице 7.

Таблица 7 - Тестирование жестов взаимодействия с моделью

Тип жеста	Проверяемые параметры	Результат
Масштабирование	Плавность, соблюдение ограничений минимального масштаба	Корректная работа
Вращение модели	Отзывчивость, точность позиционирования	Соответствует требованиям
Сброс трансформации	Возврат к исходным параметрам масштаба и поворота	Полное восстановление
Комбинированные жесты	Корректность обработки одновременных касаний	Стабильная работа

Особое внимание уделялось обработке исключительных ситуаций и предоставлению пользователю понятной обратной связи в случае возникновения ошибок.

Результаты тестирования подтвердили, что реализованный пользовательский интерфейс полностью соответствует требованиям к надежности и удобству использования, обеспечивая комфортную работу с архитектурными моделями во всех режимах просмотра.

### 3.2.2 Проверка функциональности AR-компонентов

Тестирование AR-компонентов приложения производилось в соответствии с требованиями к стабильности и точности работы системы дополненной реальности. Основной фокус был направлен на проверку корректности работы всех реализованных AR-функций в различных условиях использования. Результаты тестирования базовых AR-функций Vuforia Engine приведены в таблице 8.

Таблица 8 - Тестирование базовых AR-функций Vuforia Engine

Функция AR	Параметры тестирования	Результаты проверки
Device Tracking	Точность отслеживания положения устройства в пространстве	Стабильное отслеживание с погрешностью не более 1-2 см
Plane Detection	Распознавание горизонтальных поверхностей различных типов	Успешное определение поверхностей с различными текстурами
Image Target	Распознавание маркеров при различном освещении	Надежная работа при различной освещенности

В процессе тестирования режима виртуального тура были проверены различные аспекты взаимодействия с виртуальной моделью, результаты которых представлены в таблице 9.

Таблица 9 - Тестирование компонентов виртуального тура

Компонент	Критерий тестирования	Результат тестирования
Масштабирование в пространстве	Сохранение реальных пропорций объекта	Точное соответствие реальным размерам
Позиционирование модели	Стабильность положения при перемещении пользователя	Отсутствие дрейфа позиции
Обработка коллизий	Корректность обнаружения столкновений с виртуальными объектами	Точное определение границ объектов
Синхронизация движения	Соответствие перемещения камеры движению пользователя	Отсутствие задержек при перемещении

При тестировании режима обнаружения маркеров были проведены комплексные проверки различных условий работы, что отражено в таблице 10.

Таблица 10 - Тестирование системы распознавания маркеров

Условие тестирования	Параметры проверки	Полученные результаты
Различное освещение	Тестирование при естественном и искусственном освещении	Стабильная работа при освещенности >100 люкс
Угол обзора	Распознавание маркеров под различными углами	Успешное распознавание при углах до 75°
Расстояние до маркера	Проверка дальности распознавания	Эффективная работа на расстоянии 0.3-3 м
Частично перекрытый маркер	Минимальная видимая область для распознавания	Распознавание при видимости >50% маркера

Особое внимание было уделено тестированию производительности AR-компонентов, результаты которого представлены в таблице 11.

Таблица 11 - Тестирование производительности AR-функций

Параметр	Измеряемые показатели	Полученные значения
Частота кадров	Стабильность FPS при различной сложности сцены	Поддержание >30 FPS на целевых устройствах
Задержка отклика	Время реакции на изменение положения устройства	Менее 50 мс
Потребление памяти	Использование оперативной памяти при длительной работе	Не более 300 МБ в пиковых значениях
Нагрузка на процессор	Загрузка CPU при активном использовании AR	Не более 40% на целевых устройствах

Проведенное тестирование подтвердило надежность и эффективность реализованных AR-компонентов, обеспечивающих стабильную работу приложения в различных условиях использования.

### Выводы по главе 3

В третьей главе была представлена практическая реализация мобильного приложения дополненной реальности для визуализации архитектурных проектов. Реализованы три основных режима работы приложения: виртуальный тур для исследования объектов в реальном масштабе, режим AR (обнаружения маркеров) для привязки 3D-моделей к физическим изображениям и режим 3D-обзора для детального изучения архитектурных объектов. Разработана система обработки пользовательских комментариев и создания скриншотов, позволяющая фиксировать важные аспекты проекта. Проведено комплексное тестирование программного обеспечения, включающее проверку пользовательского интерфейса и функциональности AR-компонентов, что подтвердило стабильность работы приложения на различных мобильных устройствах.



## Заключение

В рамках данной выпускной квалификационной работы была успешно разработана и реализована система дополненной реальности, предназначенная для визуализации архитектурных проектов. Основной целью работы являлось создание современного инструмента, способствующего улучшению процесса представления и оценки архитектурных решений, а также оптимизации взаимодействия между заказчиком и разработчиками проектов.

На этапе анализа были изучены современные методы визуализации в архитектурной отрасли, проведён детальный анализ бизнес-процессов предприятия ООО «Киамет» и проведён обзор аналогичных программных продуктов. Выявленные недостатки существующих решений позволили сформировать чёткие требования к разрабатываемому AR-приложению, что обеспечило его максимальное соответствие реальным потребностям пользователей.

При проектировании системы был выбран комплексный подход, включающий использование методологии UML для моделирования программного обеспечения и архитектуры MVC для структурирования кода. Детальное логическое и физическое моделирование данных, а также разработка интуитивно понятного пользовательского интерфейса позволили создать надёжную и масштабируемую систему, обеспечивающую комфортное взаимодействие с 3D-моделями архитектурных проектов.

В процессе реализации приложения были использованы современные технологии и инструменты разработки, такие как Unity и Vuforia, что позволило добиться высокой производительности и стабильности работы системы. Разработанный продукт включает в себя функциональные возможности для виртуального тура, обнаружения маркеров, 3D-обзора моделей, а также обработки пользовательских комментариев и создания скриншотов. Проведённое тестирование подтвердило соответствие созданного

решения всем поставленным требованиям, таким как удобство использования, точность визуализации и возможность дальнейшего расширения функционала.

Таким образом, выполненная работа демонстрирует высокую степень реализации поставленных целей и задач, а разработанное AR-приложение может быть успешно внедрено в практику для улучшения процессов представления архитектурных проектов. В дальнейшем предусмотрено развитие системы с целью адаптации её к специфическим требованиям различных сегментов архитектурной отрасли, что позволит значительно расширить спектр её применения.

## Список используемой литературы

1. Аникеев, Д. В. Архитектура информационных систем : учебное пособие / Д. В. Аникеев. — Рязань : РГРТУ, 2022. — 72 с. [Электронный ресурс]. URL: <https://e.lanbook.com/book/380360> (дата обращения: 25.10.2024).
2. Аншина, М. Л. Управление жизненным циклом информационных систем : учебное пособие / М. Л. Аншина. — Москва : РТУ МИРЭА, 2024. [Электронный ресурс]. URL: <https://e.lanbook.com/book/448937> (дата обращения: 09.03.2025).
3. Баланов, А. Н. Комплексное руководство по разработке: от мобильных приложений до веб-технологий : учебное пособие для вузов / А. Н. Баланов. — Санкт-Петербург : Лань, 2024. [Электронный ресурс]. URL: <https://e.lanbook.com/book/394577> (дата обращения: 16.02.2025).
4. Головин, С.Ю. АНАЛИЗ РЫНКА ДОПОЛНЕННОЙ И ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ В СОВРЕМЕННОЙ РОССИИ / С.Ю. Головин // Бизнес-образование в экономике знаний. — 2019. — № 3. — С. 26-29. [Электронный ресурс]. URL: <https://e.lanbook.com/journal/issue/311574> (дата обращения: 25.09.2024).
5. Кольева, Н. С. Применение AR-технологий в архитектурном проектировании / Н. С. Кольева, М. В. Панова, В. В. Шемакин // International Journal of Professional Science. – 2023. – № 11-2. – С. 6-12. [Электронный ресурс]. URL: <http://scipro.ru/article/13-11-2023> (дата обращения: 25.09.2024).
6. Максимов Е.В. Разработка приложения дополненной реальности для визуализации чертежей на платформе Unity // StudNet. 2022. №6. [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/razrabotka-prilozheniya-dopolnennoy-realnosti-dlya-vizualizatsii-chertezhey-na-platforme-unity> (дата обращения: 26.09.2024).
7. Мохов, В. А. Разработка проектов в области компьютерных наук. Курс лекций : учебное пособие / В. А. Мохов. — Новочеркасск : ЮРГПУ

(НПИ), 2023. [Электронный ресурс]. URL: <https://e.lanbook.com/book/391922> (дата обращения: 23.10.2024).

8. Мухтасимов, А. Д. Перспективы применения технологии дополненной реальности в индустрии архитектуры и строительства / А. Д. Мухтасимов, Д. М. Коростелева // Инновации в науке и практике : Сборник статей по материалам XVI международной научно-практической конференции. В 3-х частях, Барнаул, 17 апреля 2019 года. Том Часть 1. – Барнаул: Общество с ограниченной ответственностью Дендра, 2019. – С. 82-87. [Электронный ресурс]. URL: [https://elibrary.ru/download/elibrary\\_37599212\\_76318642.pdf](https://elibrary.ru/download/elibrary_37599212_76318642.pdf) (дата обращения: 25.09.2024).

9. Обзор лучших платформ для AR-разработки [Электронный ресурс] <https://infoshell.ru/blog/dopolnennaya-realnost-augmented-reality-ar/> (дата обращения: 25.09.2024).

10. Общероссийский классификатор видов экономической деятельности: Приказ Росстандарта от 31.01.2014 N 14-ст) (ред. от 26.07.2022). [Электронный ресурс]. URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_163320/](https://www.consultant.ru/document/cons_doc_LAW_163320/) (дата обращения: 25.09.2024).

11. Программный продукт «Arvis» [Электронный ресурс]. URL: <https://arvis.top/> (дата обращения: 24.02.2025).

12. Программный продукт «Gamma AR» [Электронный ресурс]. URL: <https://gamma-ar.com/> (дата обращения: 24.02.2025).

13. Программный продукт «VisualLive» [Электронный ресурс]. URL: <https://visuallive.ru/> (дата обращения: 24.02.2025).

14. Программный продукт «Планоплан» [Электронный ресурс]. URL: <https://planoplan.com/ru/> (дата обращения: 24.02.2025).

15. Программный продукт Unity [Электронный ресурс]. URL: <https://unity.com/ru/solutions/xr/ar> (дата обращения: 25.10.2024).

16. Программный продукт Vuforia [Электронный ресурс]. URL: <https://vuforia.mont.com/about.html> (дата обращения: 25.10.2024).
17. РБК Компании: ОКВЭД Код 71.1: Деятельность в области архитектуры, инженерных изысканий и предоставление технических консультаций в этих областях. [Электронный ресурс]. URL: <https://companies.rbc.ru/okved/71.1/> (дата обращения: 24.02.2025).
18. Силич, В. А. Моделирование и анализ бизнес-процессов : учебное пособие / В. А. Силич, М. П. Силич. — Томск : Томский государственный университет систем управления и радиоэлектроники, 2011. — 212 с. [Электронный ресурс]. URL: <https://www.iprbookshop.ru/13890.html> (дата обращения: 25.09.2024).
19. Туманова, М. Б. Проектирование программных систем : учебное пособие / М. Б. Туманова, Е. К. Михайлова, Е. А. Муравьева. — Москва : РТУ МИРЭА, 2023. [Электронный ресурс]. URL: <https://e.lanbook.com/book/398273> (дата обращения: 23.10.2024).
20. Тыллануров, И. Применение дополненной реальности в архитектуре: новые возможности и вызовы / И. Тыллануров, М. Аллагулиев // Научный взгляд 2024 : Сборник материалов V-ой международной очно-заочной научно-практической конференции, Москва, 08 апреля 2024 года. — Москва: Научный центр "Издание", 2024. — С. 166-168. [Электронный ресурс]. URL: <https://elibrary.ru/item.asp?id=68567228> (дата обращения: 25.09.2024).
21. Documentation ARCore SDK [Электронный ресурс]. URL: <https://developers.google.com/ar?hl=ru> (дата обращения: 24.02.2025).
22. Documentation ARKit [Электронный ресурс]. URL: <https://developer.apple.com/documentation/arkit/> (дата обращения: 24.02.2025).
23. Documentation Kudan Ar SDK [Электронный ресурс]. URL: <https://www.xlsoft.com/en/products/kudan/ar-sdk.html> (дата обращения: 24.02.2025).