Содержание

[Введение 2](#_Toc132848805)

[1 Сравнительный обзор аналогов и выбор технологий 3](#_Toc132848806)

[1.1 Обзор аналогов 3](#_Toc132848807)

[1.1.1 Приложение Tilt brush 3](#_Toc132848808)

[1.1.2 Приложение Adobe Medium 5](#_Toc132848809)

[1.1.3 Приложение eDrawings Viewer 6](#_Toc132848810)

[1.2 Анализ прототипов 8](#_Toc132848811)

[1.3 Обзор средств разработки 9](#_Toc132848812)

[1.3.1 Обоснование выбора языка программирования 9](#_Toc132848813)

[1.3.2 Обоснование выбора платформы разработки 9](#_Toc132848814)

[1.3.3 Обоснование выбора средства программирования 10](#_Toc132848815)

[1.4 Выбор технологий и библиотек 11](#_Toc132848816)

[1.4.1 Технология Unity UI Toolkit 11](#_Toc132848817)

[1.4.2 Unity AR Foundation 11](#_Toc132848818)

[1.4.3 Фреймворк Zenject 11](#_Toc132848819)

[1.4.4 Unity MVVM Toolkit 12](#_Toc132848820)

[1.5 Выводы по разделу 13](#_Toc132848821)

[2 Проектирование веб-приложения 14](#_Toc132848822)

[2.1 Проектирование структурной схемы приложения 14](#_Toc132848823)

[2.2 Проектирование диаграммы вариантов использования 16](#_Toc132848826)

[2.3 Проектирование базы данных 17](#_Toc132848827)

[2.4 Выводы по разделу 19](#_Toc132848828)

[Список используемых источников 20](#_Toc132848829)

Введение

Концепция дополненной реальности (AR) получила широкое распространение в последние годы, став частью повседневной жизни людей. Многие компании и проекты используют AR для улучшения пользовательских интерфейсов, повышения вовлеченности пользователя и создания новых возможностей для обучения, развлечения и творчества. В этой области появляются новые технологии и инструменты, которые позволяют разработчикам создавать уникальные приложения и продукты для пользователей.

Целью дипломного проекта является разработка веб-приложения для создания иллюстраций в дополненной реальности. Оно будет предоставлять пользователям возможность создавать свои собственные иллюстрации, используя различные инструменты и функции, доступные в приложении. Пользователи смогут использовать камеру своего устройства для визуализации виртуальных объектов, наложенных на реальный мир, и добавлять к ним собственные элементы.

Приложение будет иметь широкий спектр применений, включая использование в обучении, развлечении, дизайне и инженерии. Оно может быть полезно для дизайнеров, художников и творческих людей, которые хотят создавать уникальные иллюстрации в AR, а также для обычных пользователей, которые просто хотят попробовать что-то новое и интересное.

Кроме того, приложение будет полезным для инженеров и других профессионалов, которые работают с технической документацией и нуждаются в возможности создавать пометки в реальности. Приложение поможет им улучшить качество и точность работы, а также повысить производительность и эффективность.

Целью данного проекта является создание инновационного приложения, которое будет позволять пользователям создавать уникальные иллюстрации в дополненной реальности. В рамках работы была разработана платформа со спектром функций и инструментов для создания иллюстраций.

Для достижения цели проекта сформулированы следующие задачи:

выполнение обзора аналогичных решений;

* проектирование веб-приложения;
* разработка веб-приложения;
* проведение тестирования веб-приложения;
* составление руководства пользователя;
* расчет экономической составляющей проекта.

1. Сравнительный обзор аналогов и выбор технологий

Для разработки актуального веб приложения необходимо ознакомиться с уже существующими аналогами. Это необходимо для того, чтобы понять, в каком функционале нуждаются пользователи, как он устроен, а также выделить ка сильные, так и слабые стороны приложения.

* 1. Обзор аналогов

На сегодняшний день существует достаточно мало веб-приложений, позволяющих создавать иллюстрации в дополненной реальности. Рассмотрим некоторые из аналогов и выявим сильнее и слабые стороны. Для обзора аналогов выл выбран следующий список веб-приложений:

* Tilt Brush [1];
* Adobe Medium [2];
* eDrawings Viewer [3].
  + 1. Приложение Tilt brush

Tilt Brush является приложением для виртуальной реальности, разработанное Google, которое позволяет пользователям создавать 3D-иллюстрации в дополненной реальности. Приложение доступно на платформах Oculus Rift, HTC Vive, Valve Index, а также на Oculus Quest в режиме Link.

В этом приложении можно создавать трехмерные иллюстрации. Пользователи могут использовать контроллеры VR, чтобы рисовать объекты в 3D-пространстве. Приложение поддерживает различные типы кистей и цветов, а также позволяет пользователям изменять размер и форму объектов (рисунок 1.1).



Рисунок 1.1 – Приложение Tiltbrush

Так же, данное приложение позволяет пользователю использовать музыку и звуковые эффекты в свои 3D-картинки. Это может создавать уникальные эффекты и дополнительную атмосферу в виртуальном пространстве. Для этого пользователь должен выбрать специальный инструмент, который называется «Waveform». Этот инструмент позволяет создавать 3д иллюстрации, которые будут реагировать на ритм играющей музыки, что позволяет создавать уникальные эффекты и добавляет динамичности различным иллюстрациям.

В приложении так же имеется интеграция с другими приложениями дополненной реальности. Tilt Brush может использоваться в сочетании с другими приложениями дополненной реальности, такими как Google Blocks, что позволяет создавать более сложные и детализированные трехмерные модели (рисунок 1.2).



Рисунок 1.2 – Интеграция Tilt Brush и Google Blocks

Кроме того, Tilt Brush поддерживает экспорт и импорт трехмерных моделей в различных форматах, таких как OBJ и FBX, что позволяет пользователям использовать свои модели в других приложениях, где можно вносить дополнительные модификации, либо использовать созданные модели в проектах напрямую. Так же, приложение Tilt Brush позволяет импортировать существующие трехмерные модели.

Tilt Brush предоставляет возможность пользователям обмениваться своими 3D-картинками с другими пользователями, что позволяет создавать сообщества и делиться своими творениями.

В целом, Tilt Brush предоставляет широкий спектр инструментов и возможностей для создания 3D-картинок в VR. Это полезный инструмент для профессиональных художников и дизайнеров, а также для тех, кто интересуется VR и хочет попробовать свои силы в создании 3D-контента.

Из основных минусов можно выделить требование к пользователю иметь шлем для дополненной реальности, а также достаточно мощный ПК, что сильно снижает потенциальное количество пользователей.

* + 1. Приложение Adobe Medium

Следующим приложением для сравнения является Adobe Medium. Данное приложение используется для создания виртуальных объектов и сред. Оно было разработано для использования с устройствами виртуальной реальности, такими как Oculus Rift, Oculus Quest и HTC Vive. Приложение позволяет создавать 3D-модели, которые могут быть использованы в приложениях виртуальной реальности, играх, фильмах и других проектах (рисунок 1.3).



Рисунок 1.3 – Приложение Adobe Medium

Функционал приложения Adobe Medium включает в себя множество инструментов для создания и редактирования трехмерных моделей. Так же, приложение имеет инструменты для моделирования, которые позволяют создавать различные формы объектов, настраивать разнообразные поверхности, а также настраивать параметры текстур и освещения.

Кроме инструментов для трехмерных моделей, данное приложение имеет в себе инструменты для рисования, которые позволяют создавать трехмерные рисунки, аналогичные рисованию в двухмерном пространстве.

В приложение так же имеются инструменты для создания скульптур, которые позволяют создавать детализированные трехмерные модели, добавляя детали и выдавливая формы из различных материалов. Панель для выбора инструментов изображена на рисунке 1.4.



Рисунок 1.4 – Приложение Adobe Medium

Так же, в приложении имеются инструменты для работы с цветом, которые позволяют настраивать цвета и оттенки объектов, создавать текстуры и наносить рисунки, а также инструменты для работы с масками и слоями, которые позволяют создавать сложные многослойные объекты, настраивать прозрачность и смешивать цвета.

Кроме того, Adobe Medium имеет широкий спектр настроек, которые позволяют пользователю управлять качеством изображения и производительностью приложения. В приложении есть также функционал социального взаимодействия, позволяющий пользователям делиться своими проектами и работать вместе над созданием сцен виртуальной реальности.

В целом, Adobe Medium – это мощный инструмент для создания 3D-моделей и сцен в виртуальной реальности. Он предлагает широкий спектр инструментов и настроек, которые позволяют пользователю создавать качественные 3D-модели, а также работать вместе с другими пользователей и делиться своими проектами.

Как и в прошлом приложении, основной минусом Adobe Medium является использование шлема виртуальной реальности и доступ к мощному ПК.

* + 1. Приложение eDrawings Viewer

Первые два приложения используются только для творческих целей. Следующее же приложение было разработано в основном для инженеров и архитекторов. eDrawings Viewer является бесплатным приложением для просмотра и обмена 2D и 3D моделями. Оно позволяет пользователям открывать и просматривать файлы в различных форматах, включая DWG, DXF, SolidWorks, CATIA, Pro/ENGINEER и многие другие (рисунок 1.5).

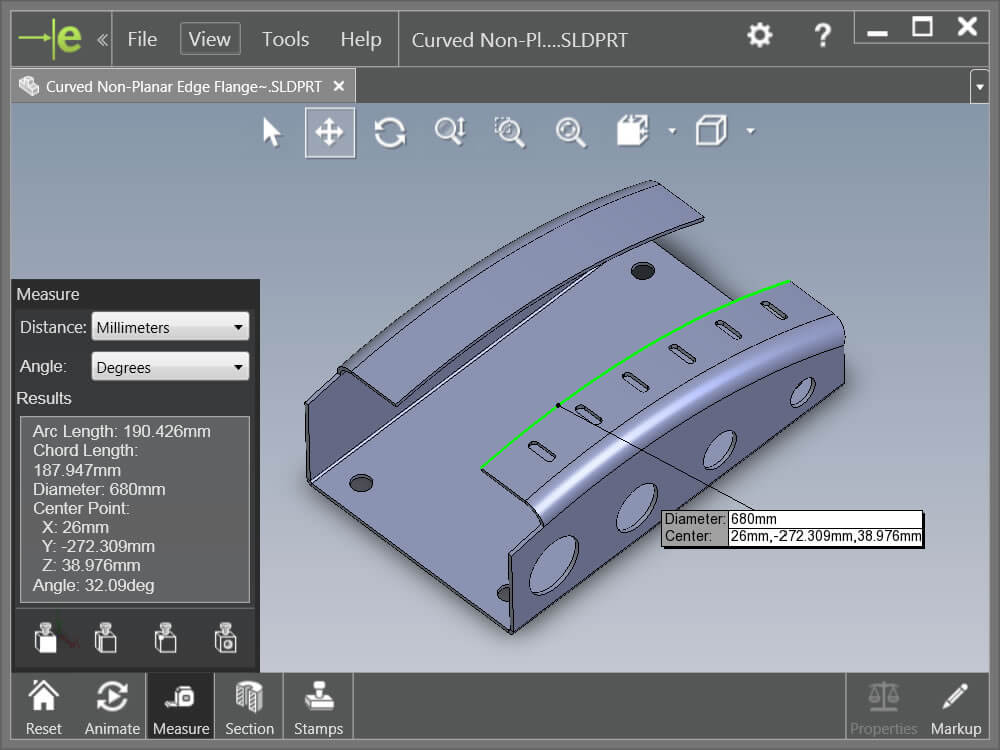


Рисунок 1.5 – Приложение eDrawings Viewer

Приложение позволяет пользователю просматривать и масштабировать двухмерные и трехмерные модели, приближать и отдалять, поворачивать их вокруг всех осей, а также устанавливать различные точки обзора и камеры, что позволяет лучше рассмотреть трехмерную модель, а также различные мелкие детали, которые пользователь мог упустить при просмотре с экрана монитора.

Так же, в приложении можно найти различные инструменты измерения: eDrawings Viewer имеет инструменты для измерения размеров и расстояний в двухмерных и трехмерных моделях, что позволяет пользователям более точно оценивать размеры и соотношения в модели.

Кроме того, eDrawings Viewer позволяет пользователям добавлять комментарии, маркеры, стрелки и другие элементы на модели, что упрощает коммуникацию и обмен информацией между участниками проекта.

В приложении так же пользователи могут экспортировать созданные модели в различных форматах, таких как BMP, JPEG, PNG, TIF, STL и другие, что позволяет использовать их в других приложениях и проектах. Кроме того, eDrawings Viewer также имеет функцию облачного хранилища, которая позволяет пользователям загружать и синхронизировать свои модели в облаке, а так же обмениваться ими с другими участниками проекта.

Пользователи могут использовать приложение на устройствах, поддерживающих AR, таких как смартфоны и планшеты с операционной системой iOS или Android. Для этого нужно скачать специальное приложение eDrawings AR и установить его на свое устройство (рисунок 1.6).



Рисунок 1.6 – Приложение eDrawings Viewer

В целом, eDrawings Viewer – это мощное и удобное приложение для просмотра и обмена 2D и 3D моделями, которое может быть использовано в различных областях, включая проектирование, машиностроение, архитектуру, промышленное производство и другие области.

В отличие от прошлых аналогов, приложение eDrawings Viewer может быть запущено как в виртуальной, так и в дополненной реальности. Из недостатков можно выделить то, что eDrawings Viewer не позволяет пользователю как-то модифицировать изображение, и имеет в себе только функционал просмотра моделей.

* 1. Анализ прототипов

Сравнив прототипы приложений, в случае первых двух приложений, можно отметить схожесть в их функциональности и интерфейсе. В каждом рассмотренном аналоге есть свои достоинства и недостатки. Наиболее важным является простота интерфейса и информативность. Из функциональных достоинств можно отметить возможность создания иллюстраций различными инструментами, экспортирование созданных иллюстраций, а также простые в управлении визуальные инструменты.

Из основных минусов всех вышеперечисленных приложений можно выделить требование к доступу к шлему дополненной реальности, а также к производительному персональному компьютеру, который сможет выполнять приложение и отображать изображение в VR шлем.

Оптимальным решением разработки собственного продукта является выбор отдельных возможностей каждого из конкурентов с последующими устранениями их недостатков, а также добавлением собственных возможностей и правил.

* 1. Обзор средств разработки

В данном подразделе будет обоснован выбор языка программирования, среды разработки и платформы, которые будут использованы в разработке приложения.

* + 1. Обоснование выбора языка программирования

Для разработки веб-приложения был выбран язык программирования C#.

C# – это объектно-ориентированный язык программирования, разработанный компанией Microsoft. Он является одним из основных языков программирования для создания Windows-приложений и игр, а также используется для разработки веб-приложений и мобильных приложений на платформе Xamarin [4].

C# является статически типизированным языком, то есть типы данных определяются на этапе компиляции. Он также поддерживает автоматическое управление памятью, что позволяет избежать проблем с утечками памяти.

C# поддерживает множество конструкций языка, таких как наследование, интерфейсы, делегаты, события, лямбда-выражения и т.д. Он также имеет широкий набор библиотек .NET Framework и .NET Core, что позволяет разработчикам создавать приложения быстро и эффективно.

C# является популярным языком программирования и используется многими крупными компаниями, такими как Microsoft, Amazon, Google и др. Он также имеет активное сообщество разработчиков и множество онлайн-ресурсов, которые помогают разработчикам улучшить свои навыки.

* + 1. Обоснование выбора платформы разработки

Unity – это кроссплатформенная среда разработки приложений и игр, которая позволяет создавать игры для мобильных устройств, компьютеров, игровых консолей, виртуальной и дополненной реальности и других платформ. В качестве языка для написания скриптов, Unity использует C# 9.0 на базе Mono [5].

Платформа Unity является популярным инструментом для разработки приложений, используемым многими крупными компаниями в игровой и развлекательной отраслях, а также для образовательных и научных целей. Unity имеет активное сообщество разработчиков и множество онлайн-ресурсов, которые помогают новичкам начать работу с платформой и продвинутым разработчикам улучшить свои навыки.

Так же, Unity является отличным выбором для создания AR (дополненной реальности) приложений, так как имеет отличную Поддержку AR-технологий, таких как ARKit для iOS и ARCore для Android, а также других платформ AR. Это позволяет разработчикам использовать функциональность AR-технологий в своих приложениях без необходимости писать сложный код с нуля.

Unity использует 2 бэкенда для написания скриптов – Mono и IL2CPP. IL2CPP (Intermediate Language To C++) – это компилятор, который используется в Unity для преобразования байт-кода .NET Framework и C# скриптов в нативный код C++, который может быть выполнен на разных платформах. IL2CPP является альтернативой компиляции Just-In-Time (JIT), которую использует стандартный механизм компиляции в .NET Framework. Одним из главных преимуществ IL2CPP является то, что он позволяет улучшить производительность игр и приложений, уменьшив время загрузки и увеличив количество кадров в секунду (FPS). Также IL2CPP позволяет разработчикам экспортировать код из Unity в другие среды разработки, такие как Xcode для iOS или Android Studio для Android. В целом, использование IL2CPP в Unity может помочь улучшить производительность игр и приложений, особенно на мобильных устройствах и других платформах с ограниченными ресурсами.

Кроме того, Unity имеет широкие возможности работы с графикой, что позволяет создавать качественные и привлекательные AR-сцены. Unity также имеет широкий набор ресурсов, таких как текстуры, модели и эффекты, которые могут быть использованы в приложениях с дополненной реальностью.

В целом, Unity предоставляет мощный и гибкий инструментарий для создания высококачественных AR-приложений, и поэтому является одним из лучших выборов для разработки AR-приложений.

* + 1. Обоснование выбора средства программирования

Интегрированная среда разработки (Integrated Development Environment) – совокупность программных средств, с помощью которых программистами производится написание, корректировка, преобразование в машинные коды, наладка и запуск различного программного обеспечения [6].

JetBrains Rider – это интегрированная среда разработки (IDE) для работы с языками программирования .NET, включая C#, F#, Visual Basic .NET и т.д. Это продукт компании JetBrains, которая также создала другие популярные инструменты разработки, такие как IntelliJ IDEA, PhpStorm и PyCharm.

Rider предоставляет множество функций, которые делают его одним из лучших выборов для разработки .NET-приложений. Rider обладает широкой многоязыковой поддержкой, включая C#, F#, Visual Basic .NET и т.д. Он также поддерживает языки, такие как HTML, CSS, JavaScript и т.д., что делает его универсальным инструментом для разработки полноценных веб-приложений.

Главным плюсом Rider является его отличная интеграция с Unity. Rider позволяет разработчикам создавать скриптыдля Unity, предоставляя полный набор инструментов для написания, отладки и тестирования кода. Разработчики могут создавать скрипты и отлаживать их непосредственно из Rider, не используя другие IDE.

Так же, Rider имеет чистый и удобный интерфейс, что делает его легким в использовании. Он также обладает множеством инструментов для улучшения производительности разработчика, таких как автодополнение кода, рефакторинг и т.д.

Кроме того, Rider предоставляет широкие возможности для тестирования кода, включая интеграцию с NUnit, xUnit и другими фреймворками тестирования.

В целом, Rider является мощным и удобным инструментом для разработки приложений на платформе .NET. Он предоставляет широкий набор функций и инструментов, которые помогают улучшить производительность и качество кода, а также ускорить процесс разработки.

* 1. Выбор технологий и библиотек
     1. Технология Unity UI Toolkit

Unity UI Toolkit (ранее известный как UIElements) – это инструмент для создания пользовательских интерфейсов в Unity. Он был введен в Unity 2019.1 и разработан на основе технологии Web UI. Unity UI Toolkit является более быстрым, легковесным и гибким, чем старый системный Unity UI [7].

С помощью Unity UI Toolkit можно создавать и редактировать пользовательские элементы интерфейса с помощью языка разметки XML под названием USS (Unity Style Sheets), который очень похож на CSS. USS позволяет создавать и изменять стили элементов интерфейса, такие как цвета, шрифты, размеры и положение.

UI Toolkit также позволяет создавать пользовательские элементы интерфейса программно, используя скрипты C#. Внутри Unity UI Toolkit элементы интерфейса могут быть связаны с данными и использоваться для отображения и изменения информации в режиме реального времени.

Unity UI Toolkit предлагает множество возможностей для настройки и управления интерфейсом в Unity, таких как события, анимации, гибкая компоновка и другие функции. Благодаря этому инструменту можно создавать настраиваемые и динамичные пользовательские интерфейсы в играх и других проектах на Unity.

Наконец, Unity UI Toolkit предоставляет возможность легко переносить пользовательские интерфейсы между различными платформами, что делает его идеальным инструментом для создания игр и приложений, которые работают на различных устройствах и платформах.

* + 1. Unity AR Foundation

Unity AR Foundation – это набор инструментов для разработки приложений дополненной реальности (AR) на Unity. Он предоставляет универсальное API, которое позволяет создавать AR-приложения, которые могут работать на различных устройствах и платформах, таких как iOS, Android, ARCore и ARKit [8].

AR Foundation обеспечивает интеграцию различных AR-технологий, таких как механика определения местоположения и отслеживание поверхностей, и предоставляет единое API, которое позволяет разработчикам создавать AR-приложения для широкого спектра устройств.

AR Foundation предоставляет разработчикам универсальный набор инструментов и API для создания AR-приложений на Unity. Он облегчает процесс разработки и позволяет разработчикам сосредоточиться на создании креативных и инновационных AR-приложений для различных устройств и платформ.

* + 1. Фреймворк Zenject

Zenject – это библиотека внедрения зависимостей (DI) для Unity. Она предоставляет мощный и гибкий механизм для управления зависимостями в проектах, что облегчает разработку масштабируемых и легко поддерживаемых приложений [9].

Dependency Injection (DI) – это паттерн проектирования, который позволяет управлять зависимостями между объектами в приложении. Он заключается в том, что объекты не создают свои зависимости самостоятельно, а получают их отвне, через внешние источники, такие как контейнеры DI или фабрики.

DI позволяет создавать более гибкую и масштабируемую архитектуру приложения. Он упрощает тестирование и отладку, так как каждая зависимость может быть заменена на имитацию или заглушку для удобного тестирования. Кроме того, DI позволяет лучше разделять ответственности между объектами, уменьшая связность и увеличивая модульность кода.

DI может быть реализован различными способами, например, через конструкторы, свойства или методы объектов, которым нужны зависимости. Важно, чтобы все зависимости были определены внешними источниками и передавались в объекты с помощью механизма DI.

Общим принципом DI является инверсия управления (Inversion of Control, IoC), где контроль за созданием объектов переносится из классов во внешние источники, которые управляют созданием объектов и их зависимостей.

Zenject предоставляет простой и интуитивно понятный API для внедрения зависимостей. Он позволяет легко настроить и управлять зависимостями в проекте.

Так же, Zenject обеспечивает гибкость и масштабируемость вашего проекта, что позволяет легко добавлять, изменять и удалять зависимости без необходимости изменения большого количества кода.

Фреймворк Zenject поддерживает не только Mono несколько платформ и языков, включая Mono и IL2CPP. Кроме того, Zenject предоставляет механизм для создания и тестирования классов с внедрением зависимостей, что делает тестирование вашего проекта более эффективным и удобным.

Unity Zenject также предоставляет множество других функций, таких как автоматическое создание объектов, жизненный цикл объектов, события жизненного цикла и другие, которые помогают упростить процесс разработки и обеспечивают более эффективную работу в вашем проекте Unity.

* + 1. Unity MVVM Toolkit

MVVM (Model-View-ViewModel) – это шаблон проектирования, который используется для создания пользовательского интерфейса (UI) в приложении. Он основывается на идеях разделения интерфейса пользователя и логики приложения на два отдельных компонента, а ViewModel обеспечивает связь между ними [10].

В отличие от простых приложений с пользовательским интерфейсом, которые обрабатывают события напрямую в коде, MVVM предлагает разделить пользовательский интерфейс и логику обработки событий.

Совместное использование Unity UI Toolkit и MVVM позволяет разделить логику приложения и пользовательский интерфейс, улучшить поддержку кода и упростить тестирование. Unity UI Toolkit может использоваться для создания пользовательского интерфейса, в то время как MVVM может быть использован для разделения логики и данных. Модель представления может обрабатывать взаимодействие пользователя с пользовательским интерфейсом.

Применение MVVM позволяет лучше организовать код проекта, повысить его читабельность и упростить поддержку проекта, поскольку логика обработки событий находится в ViewModel, а не в пользовательском интерфейсе. Это также позволяет легче тестировать код, поскольку ViewModel может быть легко тестирована без необходимости отображения пользовательского интерфейса.

Кроме того, привязки данных (Data Binding) в MVVM позволяют автоматически отображать изменения данных в пользовательском интерфейсе без необходимости вручную обновлять элементы пользовательского интерфейса. Это упрощает процесс отображения данных и уменьшает количество ошибок, связанных с ручным обновлением пользовательского интерфейса.

* 1. Выводы по разделу

Данный раздел содержит информацию о аналогах разрабатываемого программного средства, был приведен кратки обзор к каждому из них, рассмотрены их возможности и недостатки, выделены плюсы и минусы.

В итоге был сделан вывод, что в приложении важен выбор различных настроек кистей, фигур, и прочих визуальных инструментов, а также набор инструментов для пользователей, которые будут использовать приложение в инженерных целях.

Из основных минусов всех вышеперечисленных приложений можно выделить требование к доступу к шлему дополненной реальности, а также к производительному персональному компьютеру, который сможет выполнять приложение и отображать изображение в VR шлем.

В результате было принято решение на создание собственной разработки в рамках дипломного проекта, которая призвана добавить функционал создания иллюстраций в дополненной реальности. Системные требования для запуска приложений в дополненной реальности достаточно низкие, что позволяет куда большему количеству пользователей использовать приложение.

Обзор аналогов, анализ предметной области помогли в определении используемых технологий. Для разработки серверной части и базы данных приложения была выбраны технологии Firebase. Клиентская часть использует Unity.

1. Проектирование веб-приложения

Хорошая архитектура – один из важных факторов удачного проекта. Она делает процесс разработки и сопровождения программы более эффективным. Программу с хорошей архитектурой проще расширять и изменять, а также тестировать, отлаживать и снизить порог понимания для новых разработчиков.

Хорошая архитектура должна следовать следующим критериям:

* эффективность системы. В первую очередь программа должна решать поставленные задачи и выполнять их в различных условиях. Под этот критерий можно отнести такие характеристики, как надежность, безопасность, способность справляться с увеличением нагрузок, отказоустойчивость, производительность;
* гибкость системы. Любое приложение приходится менять со временем – изменяются поставленные задачи, добавляются новые, находятся проблемы или уязвимости в программном средстве. Чем быстрее и удобнее можно внести изменения в существующую функциональность, тем меньше затрат и ошибок это вызовет, следовательно, система является более гибкой и конкурентоспособной;
* расширяемость системы. Возможность добавления в систему новых сущностей и функций, не нарушая ее основной структуры – залог хорошей расширяемости. На начальном этапе разработки в систему имеет смысл закладывать только основной и самый необходимый набор функций. Но при этом архитектура должна позволять легко наращивать дополнительную функциональность по мере необходимости, причем так, чтобы внесение наиболее вероятных изменений требовало наименьших затрат;
* тестируемость. Код, который легче тестировать, будет содержать меньше ошибок и надежнее работать;
* возможность повторного использования кода. Систему желательно проек­тировать так, чтобы ее фрагменты можно было бы повторно использовать при написании других систем.
  1. Проектирование структурной схемы приложения

Архитектура приложения представляется собой системную структуру и элементы дизайна, из которых состоит приложение. Она также включает в себя методы, процессы и компоненты, используемые во время разработки приложения.

Для реализации приложения была выбрана SOLID архитектура. Этот принцип программирования необходим для создания масштабируемой архитектуры приложений. Масштабируемая архитектура приложений основана на гибкой, подвижной технологии. Это облегчает модернизацию, обновление и расширение при необходимости. Этот универсальный принцип программирования является установленной основой, на которой базируются руководящие принципы архитектуры приложений. Кроме того, этот принцип может быть применен в любом проекте, который использует объектно-ориентированный язык программирования, и их использование помогает создавать качественное ПО.

К преимуществам SOLID архитектуры можно отнести:

* принцип единственной ответственности, обозначает, что каждый объект должен иметь одну обязанность и эта обязанность должна быть полностью инкапсулирована в класс;
* принцип открытости и закрытости декларирует, что классы, модули,
* функции должны быть открыты для расширения и закрыты для изменений;
* принцип подстановки Барбары Лисков – функции, которые используют базовый тип, должны иметь возможность использовать подтипы базового типа, не зная об этом и не вызывая ошибок;
* принцип разделения интерфейсов говорит о том, что слишком «толстые» интерфейсы необходимо разделять на более маленькие и специфические, чтобы клиенты маленьких интерфейсов знали только о методах, которые необходимы им в работе;
* принцип инверсии зависимостей – модули верхних уровней не должны зависеть от модулей нижних уровней, а оба типа модулей зависят от абстракций.

Архитектура приложения представлена на рисунке 2.1.

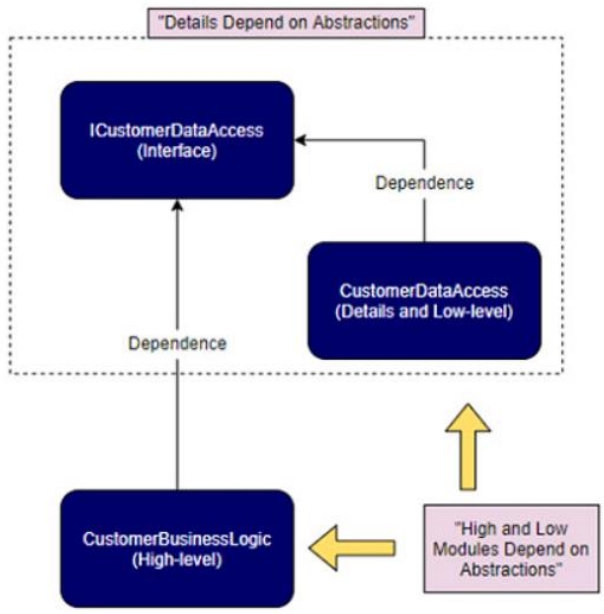


Рисунок 2.1 – Архитектура приложения

Использование данной архитектуры позволяет сократить временные затраты на добавление нового функционала. Сократить убытки от ошибок программного продукта, за счёт повышения его качества.

2. 1. Проектирование диаграммы вариантов использования

Чтобы начать разработку необходимо составить ряд требований, которым приложение должно соответствовать. Должны быть определены цели и задачи, продуманы все необходимые варианты использования, а также определены роли, используемые в приложении.

Приложение, разрабатываемое в данном дипломном проектировании, должно соответствовать требованиям безопасности, поэтому должна быть организована система защиты от несанкционированного доступа.

Каждый пользователь, прежде чем попасть в систему должен в ней аутентифицироваться. Для программного средства предусмотрены две роли: пользователь и админи­стратор. Для каждой из ролей определен свой ряд обязанностей и вариантов исполь­зования приложения.

Администратор имеет возможность просматривать аналитику.

Для роли пользователя определены следующие возможности:

* создание иллюстраций;
* публикация иллюстраций;
* экспорт иллюстраций;
* сохранение иллюстраций;
* выбор инструмента рисования;
* настройка параметров инструмента;
* настройка цвета инструмента;
* настройка материала инструмента;
* просмотр всех иллюстраций;
* фильтрация иллюстраций;
* просмотре превью иллюстраций;
* размещение иллюстраций в реальности.

Диаграмма вариантов использования (диаграмма прецедентов) в UML – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать разрабатываемую систему на концептуальном уровне.

Основное назначение диаграммы – описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

Для данного дипломного проекта была разработана диаграмма вариантов использования, на которой изображены потребности пользователей системы и функциональные возможности разрабатываемого программного средства. С ее использованием будет проходить дальнейшая разработка веб-приложения.

Диаграмма представлена на рисунке 2.2.

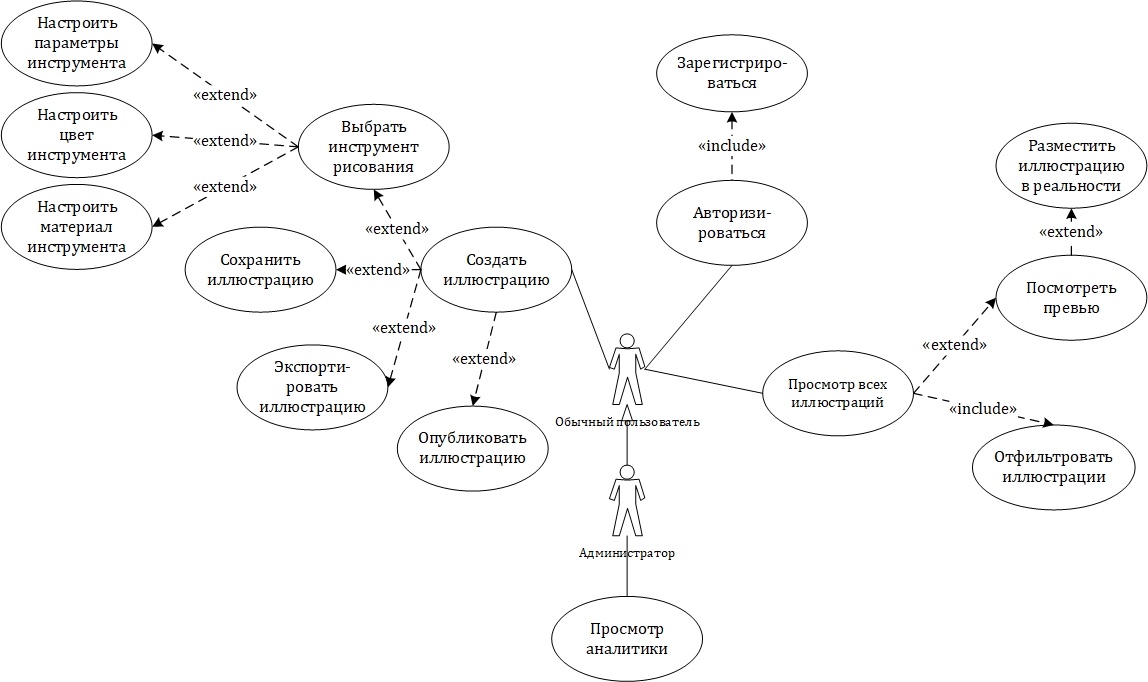


Рисунок 2.2 – Диаграмма вариантов использования

В результате была разработана диаграмма вариантов использования, на которой изображен функционал разрабатываемого программного средства. С использованием данной диаграммы будет проходить дальнейшая разработка приложения.

* 1. Проектирование базы данных

Прежде чем приступить к разработке программного средства необходимо спроектировать базу данных. На основе сформулированных требований к разрабатываемому приложению определяется состав информации, которая должна хранится в базе данных, а также выявляются взаимосвязи данных, размещаемых в БД. Состав и взаимосвязи данных отображаются моделью данных предметной области, которая является основой любой базы данных.

В качестве базы данных для приложения был выбран облачный сервис Firebase Firestore. Firebase Firestore – это гибкая, расширяемая облачная база данных от Google, которая используется в качестве серверного хранилища данных для веб-приложений, а также приложений для мобильных устройств. Firestore обеспечивает быстрый и масштабируемый доступ к данным, а также поддерживает синхронизацию данных в режиме реального времени [11].

Сервис Firestore использует NoSQL базу данных (Not Only SQL, не только SQL) – это семейство СУБД, которые отличаются от реляционных баз данных тем, что они не используют традиционные таблицы, столбцы и строки для хранения данных. Вместо этого они используют различные модели хранения данных, такие как ключ-значение, документоориентированные, графовые и другие.

Одной из причин появления NoSQL баз данных было желание обеспечить более высокую производительность и масштабируемость веб-приложений. Реляционные базы данных не всегда могут эффективно обрабатывать большие объемы данных или обеспечивать достаточную производительность при высокой нагрузке.

Firestore представляет собой документ-ориентированную базу данных, которая хранит данные в виде коллекций документов. Каждый документ может содержать несколько полей, каждое из которых представляет собой пару ключ-значение. Документы могут быть организованы в коллекции, которые в дальнейшем могут быть организованы в различные подколлекции.

Firestore имеет различные SDK для разных языков программирования и платформ, включая Java, Swift, Kotlin, JavaScript, Node.js, Unity и другие.

Для лучшего понимания структуры разрабатываемой базы данных была создана логическая схема, представленная на рисунке 2.3

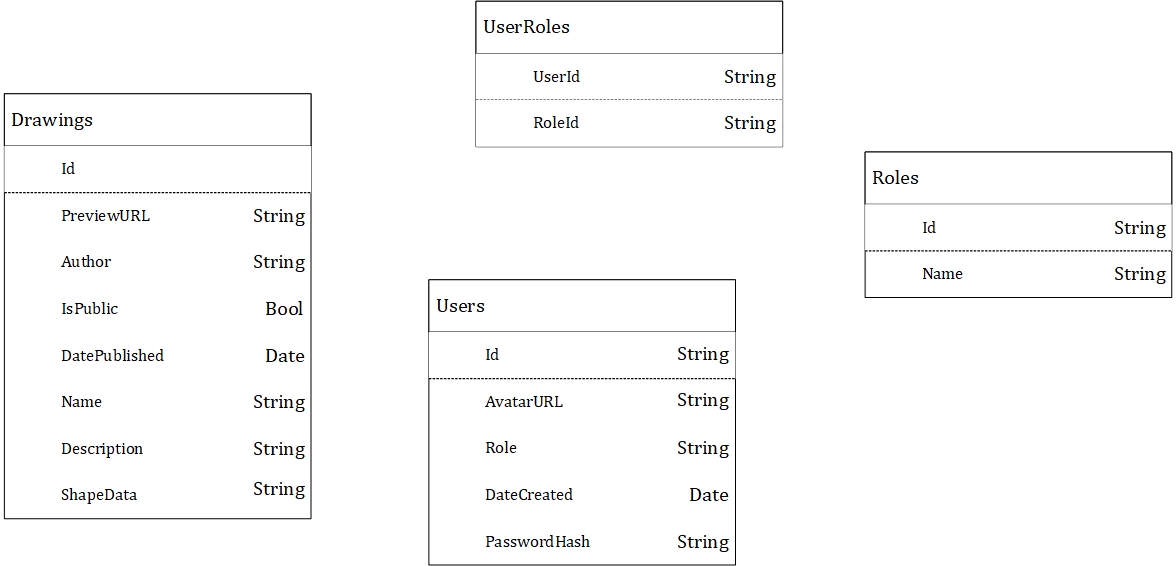


Рисунок 2.3 – Логическая схема базы данных

Для реализации функционала приложения будет создано 4 коллекции. Далее приведена структура и описание всех коллекций.

Коллекция «Users» содержит в себе информацию о пользователях. Структура коллекции представлена в таблице 2.1

Таблица 2.1 – Описание коллекции «Users»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id | String | Идентификатор пользователя |
| AvatarURL | String | Ссылка на аватар пользователя |
| Role | String | Роль пользователя |
| DateCreated | Date | Дата создания |
| PasswordHash | String | Хеш пароля пользователя |

Коллекция «Drawings» содержит в себе информацию об иллюстрациях пользователей. Структура коллекции представлена в таблице 2.2

Таблица 2.2 – Описание коллекции «Drawings»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id | String | Идентификатор иллюстрации |
| Author | String | Автор иллюстрации |
| IsPublic | Bool | Публична ли иллюстрация |
| DatePublished | Date | Дата создания |
| Name | String | Название иллюстрации |
| Description | String | Описание иллюстрации |
| ShapeData | String | Сериализованная иллюстрация |

Коллекция «UserRoles» содержит в себе информацию о ролях, выданных пользователям. Структура коллекции представлена в таблице 2.3

Таблица 2.3 – Описание коллекции «UserRoles»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| UserId | String | Идентификатор пользователя |
| RoleId | String | Идентификатор роли |

Коллекция «Roles» содержит в себе информацию о ролях. Структура коллекции представлена в таблице 2.4

Таблица 2.4 – Описание коллекции «Roles»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id | String | Идентификатор роли |
| Name | String | Название роли |

* 1. Выводы по разделу

В данном разделе описан процесс разработки диаграммы вариантов использования, в которой был представлен основной функционал разрабатываемого приложения, а также описан процесс выбора архитектуры для разрабатываемого программного модуля. Помимо этого, дано краткое описание технологии базы данных, которая используется в приложении, а также разработана модель базы данных. Представлены и описаны коллекции документов БД, а также их особенности.

Список используемых источников

1. Tilt Brush [Электронный ресурс]. – Режим доступа: https://www.tiltbrush.com/. – Дата доступа: 15.04.2023.
2. Adobe Medium [Электронный ресурс]. – Режим доступа: https://www.adobe.com/products/medium.html. – Дата доступа: 15.04.2023.
3. eDrawings Viewer [Электронный ресурс]. – Режим доступа: https://www.edrawingsviewer.com/. – Дата доступа: 15.04.2023.
4. Документация по C# [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/csharp/. – Дата доступа: 17.04.2023.
5. Unity User Manual [Электронный ресурс]. – Режим доступа: https://docs.unity3d.com/Manual/index.html. – Дата доступа: 18.04.2023.
6. Rider IDE [Электронный ресурс]. – Режим доступа: https://www.jetbrains.com/ru-ru/rider/. – Дата доступа: 19.04.2023.
7. UI Toolkit User Manual [Электронный ресурс]. – Режим доступа: https://docs.unity3d.com/Manual/UIElements.html. – Дата доступа: 20.04.2023.
8. AR Foundation [Электронный ресурс]. – Режим доступа: https://unity.com/unity/features/arfoundation. – Дата доступа: 22.04.2023.
9. Zenject [Электронный ресурс]. – Режим доступа: https://github.com/modesttree/Zenject. – Дата доступа: 24.04.2023.
10. Unity MVVM Toolkit [Электронный ресурс]. – Режим доступа: https://github.com/LibraStack/UnityMvvmToolkit. – Дата доступа: 25.04.2023.
11. Firebase Documentation [Электронный ресурс]. – Режим доступа: https://firebase.google.com/docs. – Дата доступа: 27.04.2023.