

Минобрнауки России
Юго-Западный государственный университет
Кафедра программной инженерии

ОТЧЕТ

о преддипломной (производственной) практике

наименование вида и типа практики

на (в) Юго-Западном государственном университете

наименование предприятия, организации, учреждения

Студента 5 курса, группы ПО-12з

курса, группы

Лобашова Ильи Олеговича

фамилия, имя, отчество

Руководитель практики от
предприятия, организации,
учреждения

Оценка _____

директор

должность, звание, степень

Куркина А. В.

фамилия и. о.

подпись, дата

Руководитель практики от
университета

Оценка _____

к.т.н. доцент

должность, звание, степень

Чаплыгин А. А.

фамилия и. о.

подпись, дата

Члены комиссии

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

Курск 2026 г.

СОДЕРЖАНИЕ

1	Анализ предметной области	4
1.1	Краткая история по подбору штор и обоев	4
1.2	Характеристики штор и обоев	5
1.3	Анализ существующего программного обеспечения по подборке штор и обоев	6
1.4	Перспективы и направления развития	10
2	Техническое задание	12
2.1	Основание для разработки	12
2.2	Цель и назначение разработки	12
2.3	Актуальность темы разработки	13
2.4	Требования к программной системе	13
2.4.1	Требования к данным	13
2.4.2	Функциональные требования	14
2.4.3	Требования к безопасности	16
2.4.4	Требования к программному обеспечению	16
2.4.5	Требования к программному обеспечению	17
2.4.6	Требования к оформлению документации	17
3	Технический проект	18
3.1	Выбор технологии проектирования	18
3.2	Выбор средств разработки	20
3.2.1	HTML5	20
3.2.2	Язык С#	22
3.3	Компоненты приложения	27
3.3.1	Результат проектирования	29
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	36

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД – база данных.

ИС – информационная система.

ИТ – информационные технологии.

ПО – программное обеспечение.

РП – рабочий проект.

СУБД – система управления базами данных.

ТЗ – техническое задание.

ТП – технический проект.

UML (Unified Modelling Language) – язык графического описания для объектного моделирования в области разработки программного обеспечения.

1 Анализ предметной области

1.1 Краткая история по подбору штор и обоев

Первые занавески появились в Древнем Египте, где использовались натуральные материалы — лён, шёлк и пальмовые листья. Они служили защитой от солнца, пыли и насекомых, а также играли социальную роль, являясь признаком статуса и богатства. Богатые египтяне украшали свои дома тканями с геометрическими узорами и изображениями божеств.

В Древнем Риме шторы приобрели дополнительную функциональность и эстетическую ценность. Были разработаны примитивные механизмы подъёма штор, а тяжёлые ткани, такие как бархат и шёлк, подчёркивали роскошь интерьеров.

В XVIII–XIX веках промышленная революция сделала шторы и обои доступными широким слоям населения. Машины для ткачества и химическая промышленность позволили производить ткани и обои в больших объёмах и разнообразных цветах. В викторианскую эпоху шторы и обои часто подбирались из одной коллекции, формируя единую композицию интерьера.

Появились специализированные каталоги и руководства по выбору и сочетанию штор и обоев, что свидетельствует о формировании осознанного подхода к дизайну интерьера.

В XX веке произошёл отход от пышности и роскоши прошлых столетий. Минимализм и функционализм предложили простые формы и светлые оттенки, вытеснив традиционные узоры и пышные драпировки. Синтетические материалы упростили уход за шторами и сделали их более долговечными.

В наши дни шторы и обои становятся высокотехнологичными элементами интерьера. Широко используются экологичные материалы, умные системы управления светом и климатом, а также цифровые принты и лазерная резка, позволяющие создавать уникальные дизайны.

Синергия штор и обоев

Исторически шторы и обои развивались параллельно, но со временем их стали воспринимать как единое целое. Начиная с XIX века, появилась устойчивая традиция сочетать эти элементы по стилю, цвету и фактуре. Такое единство создаёт гармоничный и выразительный интерьер, подчёркивая индивидуальность и вкус владельца.

Таким образом, история подбора штор и обоев демонстрирует путь от утилитарных предметов к ключевым элементам художественного оформления интерьера, отражающим культурные, социальные и технологические изменения человечества.

1.2 Характеристики штор и обоев

Шторы представляют собой важный декоративный элемент интерьера, изготавливаемый из различных материалов, таких как хлопок, лен, шелк, полиэстер или смешанных волокон. Их плотность варьируется от легких воздушных тканей, таких как органза или тюль, до тяжелых и плотных материалов типа сатина или бархата. Светопроницаемость штор может быть разной: от светонепроницаемых блэкаутов до полупрозрачных моделей, пропускающих естественный свет. Выбор цветовой гаммы и рисунка включает широкий спектр оттенков и орнаментов — от классических однотонных вариантов до современных абстрактных композиций. Важными эксплуатационными свойствами являются легкость ухода, устойчивость к выгоранию и антистатичность, препятствующая накоплению пыли. Эстетические особенности штор варьируются от традиционных массивных портьер до лаконичных конструкций в стиле минимализм, позволяя подчеркнуть индивидуальный стиль помещения.

Обои также играют значительную роль в оформлении интерьера, предлагая разнообразие основ и материалов. Среди популярных видов выделяются бумажные, флизелиновые, виниловые, текстильные и натуральные покрытия, такие как бамбук или пробка. Поверхностная отделка обоев может быть матовой, глянцевой или рельефной, создавая интересные оптические эффекты и придавая помещению особую атмосферу. Водостойкость и дол-

говечность определяют возможность влажной уборки и продолжительность сохранения первоначального внешнего вида. Экологическая чистота и безопасность обоев обеспечивается гипоаллергенностью, паропроницаемостью и наличием антибактериальных добавок. Цветовая палитра и декоративные элементы обоев варьируются от простых однотонных решений до сложных орнаментальных рисунков, идеально вписываясь в любой стиль интерьера.

Гармоничное сочетание штор и обоев достигается посредством цветовой гармонии, стилистического единства и продуманного контраста или нюансировки. Архитектурные особенности помещения учитываются при выборе плотности и длины штор, а также текстуры и размера рисунка обоев. Правильно подобранные комбинации помогают подчеркнуть достоинства комнаты и скрыть недостатки, обеспечивая комфортное проживание и эстетическое наслаждение.

1.3 Анализ существующего программного обеспечения по подборке штор и обоев

Существует значительное количество программных решений для подбора штор и обоев, каждое из которых обладает своими уникальными характеристиками. Чтобы лучше понять рынок и конкурентов, предлагаем детально проанализировать три наиболее распространённых сервиса, выделив их ключевые особенности и различия. В данном разделе будет представлен анализ существующих систем, их функциональных возможностей, преимуществ и недостатков, а также примеры конкретных программ.

Houzz

Описание: Houzz представляет собой международную платформу для поиска идей, товаров и профессионалов в сфере ремонта и дизайна интерьеров. Платформа была запущена в 2009 году и насчитывает миллионы зарегистрированных пользователей. Одним из главных преимуществ сайта является огромный каталог фотографий, созданных профессионалами и обычными пользователями. Пользователи могут сохранять понравившиеся проекты в личные коллекции, создавать собственные дизайн-проекты и находить под-

рядчиков для воплощения задуманного в реальность. Основная аудитория — профессионалы индустрии и обычные потребители, заинтересованные в ремонте жилья или коммерческой недвижимости.

Особенности:

- Огромная база фотографий и проектов.
- Инструменты для самостоятельного создания проектов («Planner»).
- Возможность консультации с профессиональными архитекторами и дизайнерами.
- Каталог товаров и услуг с возможностью покупки прямо на сайте.
- Доступ к профессиональным услугам (архитектура, ремонт, ландшафтный дизайн).

Сильные стороны:

- Масштабируемость и разнообразие контента.
- Активное сообщество профессионалов и энтузиастов.
- Удобный механизм сохранения избранных проектов.
- Постоянное обновление и расширение каталога товаров.

Слабые стороны:

- Некоторая сложность навигации среди большого объема информации.
- Ориентация на западный рынок, меньше внимания уделяется локальным особенностям некоторых стран.
- Некоторые функциональные ограничения бесплатной версии.

Стоимость: платформа доступна абсолютно бесплатно для большинства пользователей. Существуют премиальные услуги для бизнеса, включающие продвижение объявлений, рекламу и специальные коммерческие аккаунты. Стоимость зависит от региона и типа аккаунта, варьируется примерно от 50500 в месяц для коммерческих организаций.

Ключевые преимущества:

- Большой объем контента и постоянные обновления.
- Возможность непосредственного заказа товаров и услуг через сайт.
- Консультации профессионалов и сообщества участников.

Недостатки:

- Требуется оплата дополнительных услуг для бизнеса.
- Функционал ограничен бесплатными версиями.

DecorPad

Описание: DecorPad позиционируется как удобная виртуальная студия дизайна интерьера. Пользователи могут импортировать собственную фотографию своей комнаты или использовать готовый макет, чтобы экспериментировать с различными предметами интерьера, такими как шторы, ковры, картины и мебель. Сервис доступен бесплатно, однако существуют платные премиум-возможности. Особенностью является возможность создавать дизайн-проект на основе собственной фотографии помещения или стандартизированного макета.

Особенности:

- Простой и интуитивный интерфейс для начинающих пользователей.
- Широкий ассортимент предметов интерьера и отделочных материалов.

- Функция фотореалистичной визуализации интерьера.
- Готовые сценарии проектов и тематические подборки.
- Совместимость с мобильными устройствами и браузерами.

Сильные стороны:

- Легко освоить даже новичкам.
- Большое количество бесплатных ресурсов.
- Возможность быстрой демонстрации и оценки итогового результата.
- Регулярные обновления ассортимента продукции.

Слабые стороны:

- Недостаточная глубина деталей для сложных проектов.
- Ограниченнное число встроенных функций для профессиональной работы.
- Возможны проблемы с точностью воспроизведения оттенков и фактуры материалов.

Стоймость: сервис предоставляется бесплатно большинству пользователей. Однако существует платная подписка с дополнительными функциями,

такая как увеличенное хранилище, продвинутый функционал визуализации и отсутствие рекламы. Цена подписок начинается примерно от 1050 в месяц.

Ключевые преимущества:

- Простота использования и быстрое создание проекта.
- Встроенные инструменты визуализации.
- Бесплатный доступ к основному функционалу.

Недостатки:

- Платные подписки требуют регулярной оплаты.
- Ограниченные возможности для глубокой проработки проектов.

MyDeco

Описание: MyDeco – русскоязычный портал, созданный на основе зарубежного опыта, который стремится упростить процесс подбора элементов интерьера. Основное отличие от западных аналогов заключается в ориентации на российскую аудиторию, понимании местных особенностей рынка и адаптации к местным стандартам качества.

Особенности:

- Большая библиотека образцов обоев, штор и другого текстиля.
- Онлайн-консультации с местными специалистами-дизайнерами.
- Рекомендательные алгоритмы подбора изделий по стилевым характеристикам.
- Локальные фильтры и интеграция с магазинами российского сегмента.

Сильные стороны:

- Русскоязычная поддержка и адаптированный контент.
- Строгая нацеленность на отечественный рынок и местные тенденции.
- Простота и удобство для новичков.
- Точное соответствие российским стандартам и рынку.

Слабые стороны:

- Несколько устаревший интерфейс и ограниченный функционал по сравнению с иностранными конкурентами.
- Небольшая популярность за пределами России.

- Необходимость постоянного мониторинга соответствия новых тенденций и стандартов.

Стоимость: использование базовых функций сайта бесплатное. Дополнительные профессиональные услуги и консультация специалистов представляются на платной основе. Конкретные расценки зависят от сложности запросов и варьируются индивидуально. Средний диапазон стоимости консультаций составляет около 1000—5000 рублей за сессию.

Ключевые преимущества:

- Российское происхождение и адаптированность под местный рынок.
- Прямой контакт с российскими производителями и поставщиками.
- Понятный интерфейс и инструкции на русском языке.

Недостатки:

- Не самое современное оформление сайта и немного устаревшие методы представления информации.
- Оплата отдельных консультаций увеличивает общие расходы.

1.4 Перспективы и направления развития

Изучение перечисленных аналогов показало, что на рынке существует большой интерес к программам подбора элементов интерьера, особенно шторам и обоям. Каждый из рассмотренных сервисов имеет уникальные достоинства и недостатки, позволяющие обозначить направление дальнейших разработок.

Основные выводы анализа:

1. Важность удобной и доступной модели взаимодействия с пользователем.
2. Актуальность локализации и адаптация под региональные рынки.
3. Потребность в интеграции профессиональных консультационных услуг.
4. Потенциал роста благодаря расширению ассортимента и улучшенной персонализации рекомендаций.

Эти выводы подчеркивают важность баланса между простотой использования и мощностью функциональных возможностей, что открывает перспективу для будущих исследований и разработок.

Обзор представленных платформ выявил следующие важные моменты:

- Существует потребность в инструменте, совмещающем бесплатный доступ и возможность дополнительной монетизации путем продажи консультаций и премиального контента.

- Важно обеспечить хорошее качество контента и постоянную обратную связь с аудиторией.

- Адаптация продукта под региональные стандарты и культурные особенности повышает привлекательность сервиса.

Это исследование подчеркивает потенциал дальнейшего развития программного обеспечения для подбора штор и обоев, создавая основу для последующих глав и исследования требований пользователей.

2 Техническое задание

2.1 Основание для разработки

Полное наименование системы: «Программа для подбора штор и обоев в помещении».

Основанием для разработки программы является приказ ректора ЮЗГУ от «_____» 2025 г. №_____ «Об утверждении тем выпускных квалификационных работ».

2.2 Цель и назначение разработки

Цель проекта: разработать программное обеспечение, облегчающее выбор оптимального сочетания штор и обоев для пользователей на основе задаваемых ими критериев (цвет, материал, стиль, и т.д.) и обеспечивающее реалистичную визуализацию полученного результата.

С системой должны работать следующие группы пользователей:

- администратор программы;
- посетитель.

Администратор должен иметь возможность добавлять, удалять и редактировать свои категории. Например, «цвет» или «материал». Посетитель программы должен иметь возможность просматривать шторы и обои нужного цвета, добавлять их в отчет о подборе.

Целью разработки является выбор оптимального сочетания штор и обоев для пользователей.

Задачами данной разработки являются:

- Предоставлять пользователям удобный интерфейс для выбора штор и обоев;
- Автоматически предлагать комбинации материалов и цветов на основе введенных данных;
- Показывать результат выбора в режиме реального времени в виде предварительного эскиза или рендера;
- Сохранять и экспортить созданные проекты;

- Формировать отчет о подборе с указанием необходимой площади покрытия и ориентировочной стоимости материалов.

2.3 Актуальность темы разработки

Актуальность данной дипломной работы обусловлена необходимостью автоматизации процесса подбора декоративных элементов интерьера. Традиционно процесс выбора штор и обоев требует значительных временных и интеллектуальных усилий, поэтому появление эффективных инструментов значительно облегчает жизнь конечному пользователю. В условиях современного рынка автоматизация позволяет сократить время обработки заказов и минимизировать ошибки, что способствует улучшению качества обслуживания пользователей.

Роль автоматизации в различных сферах деятельности неоценима. В производственном секторе автоматизация позволяет более точно подобрать элементы интерьера.

Таким образом, внедрение автоматизированных решений становится ключевым фактором для повышения качества работ, позволяя адаптироваться к быстро меняющимся условиям рынка и удовлетворять потребности клиентов на высоком уровне.

2.4 Требования к программной системе

2.4.1 Требования к данным

Выходными данными для системы являются:

- корректный подбор нескольких хороших комбинаций цветов и материалов;
- создание эскизов;
- создание списка рекомендаций.

2.4.2 Функциональные требования

Концептуальная модель Информационной Системы (ИС) для подбора штор и обоев

В системе должен быть представлен один действующий пользователь.

На основании анализа требований в программе должны быть реализованы следующие прецеденты:

1. Прецедент «Добавление пользователя». Прецедент позволяет добавить нового пользователя.
2. Прецедент «Изменения данных». Прецедент позволяет изменять данные находящиеся в программе и базе данных.
3. Прецедент «формирования отчетов». Прецедент позволяет создавать сохранять и печатать отчеты [8].

Сценарий для прецедента «Авторизация»

Основной исполнитель: Пользователь.

Заинтересованные лица и их требования. Пользователь хочет, чтобы приложение запустилась, и у него был полный доступ.

Предусловие Развёртывание базы данных на ПК.

Описание. Пользователь нажимает кнопку "Авторизоваться". Система в зависимости от введенного логина и пароля даёт зайти под разными ролями.

Результат. Пользователь авторизирован.

На рисунках 2.1 и 2.2 отображены диаграммы прецедентов для программы.

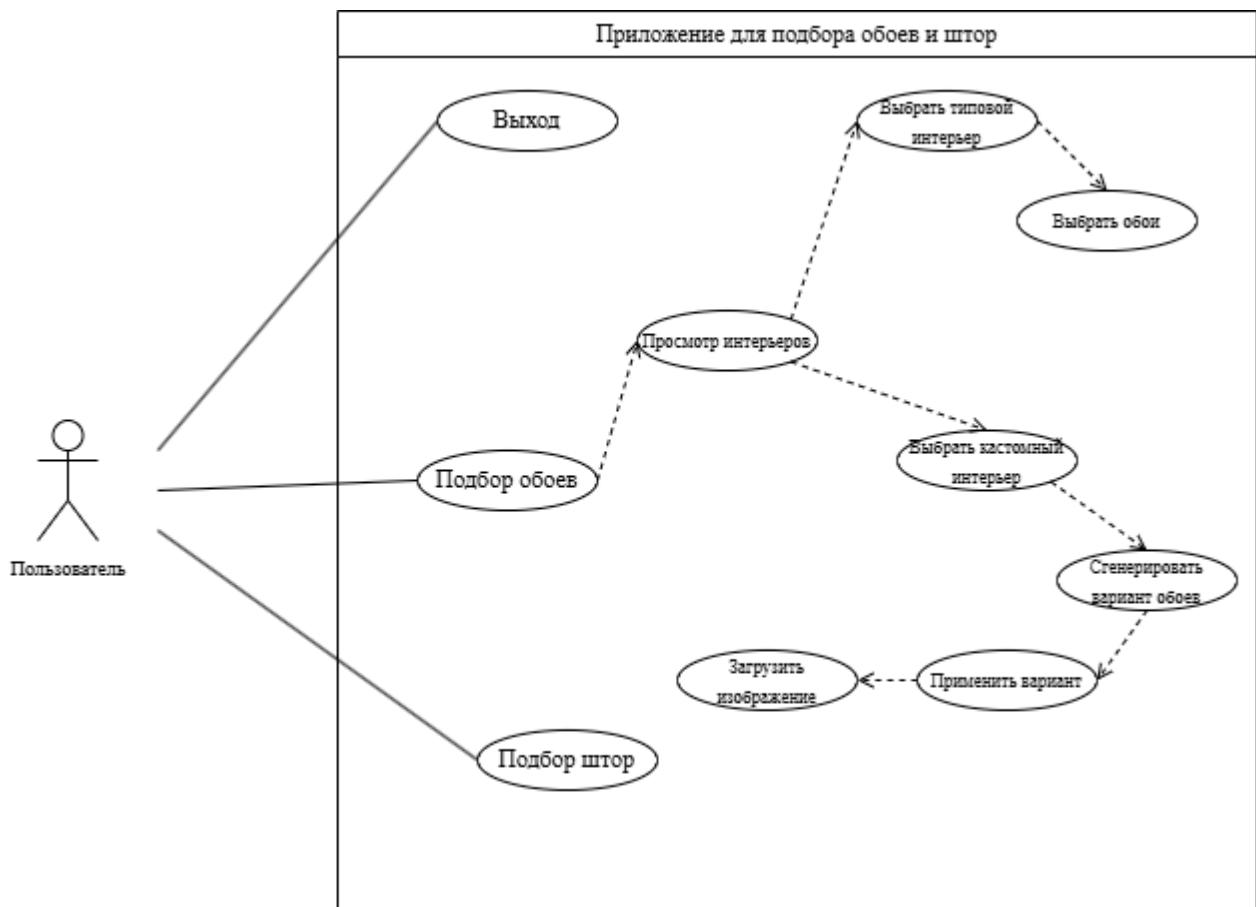


Рисунок 2.1 – Диаграмма прецедента пользователя для программы

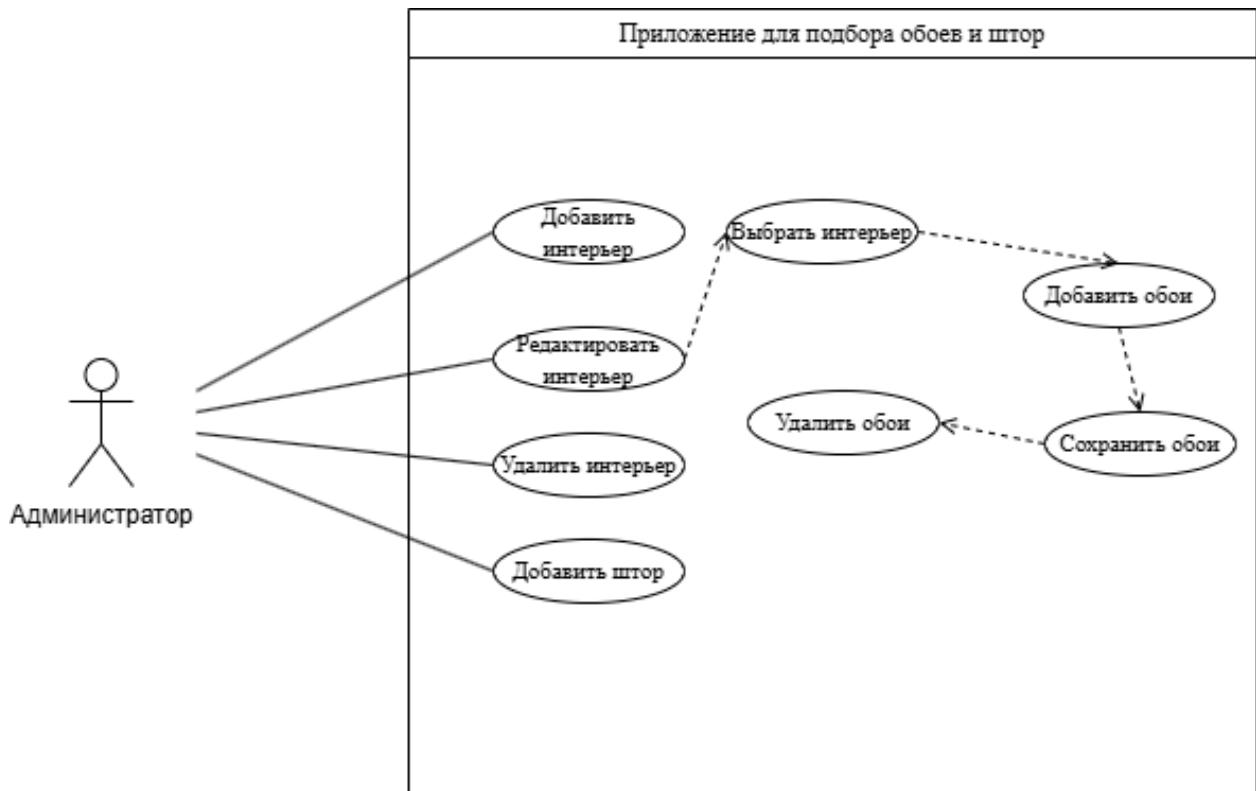


Рисунок 2.2 – Диаграмма прецедента администратора для программы

2.4.3 Требования к безопасности

Программа должна корректно работать при различных условиях эксплуатации и обрабатывать ошибки, возникающие при неправильном вводе данных или других непредвиденных ситуациях [23].

2.4.4 Требования к программному обеспечению

Для разработки программного продукта необходимы использовать:

1. Язык C# – для разработки клиентского приложения.
2. Microsoft Visual Studio не ниже версии 2017 – для компиляции кода программы.
3. Microsoft SQL Server 2018 — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft [24].

Программное изделие должно работать в операционных системах Windows с установленным .NET Framework версии не ниже 4.6.1[25].

Программная среда, в которой запускается программа, должна поддерживать русскую раскладку клавиатуры.

2.4.5 Требования к программному обеспечению

Для работы приложения требуется дисковое пространство не менее 2,5 Гб, минимум 2 Гб оперативной памяти и подключение к Интернету. Рекомендуется использовать процессор с 2 или более ядрами и частотой 2 ГГц или выше [26].

2.4.6 Требования к оформлению документации

Требования к стадиям разработки программ и программной документации для вычислительных машин, комплексов и систем независимо от их назначения и области применения, этапам и содержанию работ устанавливаются ГОСТ 19.102–77.

3 Технический проект

3.1 Выбор технологии проектирования

Выбор технологии проектирования для разработки программы автоматизации учета и предварительных заказов на предприятиях различных сфер деятельности является важным этапом, который определяет успешность реализации проекта. При выборе технологий следует учитывать несколько ключевых факторов.

Тип системы влияет на выбор архитектурного подхода — это может быть клиент-серверная архитектура, облачные решения или микросервисная архитектура. Облачные технологии обеспечивают доступность и гибкость, что особенно важно для предприятий с распределенной структурой.

Язык программирования и платформа выбираются исходя из опыта команды разработчиков, а также требований к производительности и безопасности системы. Например, можно рассмотреть Python, Java, C# и соответствующие фреймворки, такие как .NET, Java Spring или Node.js.[27,28].

Для хранения данных о заказах и учете важно подобрать подходящую систему управления базами данных. Реляционные базы (PostgreSQL, MySQL) подходят для структурированных данных, тогда как NoSQL решения (MongoDB и др.) используются при необходимости работы с неструктуризованными данными. В данной работе из-за небольшого ассортимента штор и обоев базу данных решено не создавать и обойтись наборами веб-страниц.

Интерфейс пользователя должен быть удобным и интуитивно понятным. Современные веб-фреймворки (React, Angular, Vue.js) или мобильные платформы (Flutter, React Native[29]) позволяют создавать качественные приложения для различных устройств.

Необходимо учитывать возможность интеграции с существующими системами учета и управления, что требует использования API и других технологий для обеспечения совместимости.

Безопасность системы предусматривает внедрение механизмов аутентификации и защиты от несанкционированного доступа.

Кроме того, технологии проектирования должны обеспечивать гибкость и масштабируемость решения, позволяя адаптироваться к изменяющимся требованиям бизнеса и расширять функциональность.

Критерии выбора технологий:

1. Популярность и поддержка сообществом: используемые технологии должны обладать широкой поддержкой и развитым сообществом разработчиков, что способствует легкости поиска решений возникших проблем и снижению риска долгосрочных отказов.

2. Производительность и масштабируемость: необходимо обеспечить высокую производительность и возможность масштабирования системы, так как предполагается обслуживание большого числа пользователей.

3. Простота интеграции и развёртывания: технология должна позволять быстро разворачивать систему на разных площадках и иметь минимальный барьер для интеграции с внешними сервисами.

4. Безопасность и защита данных: при выборе технологий особое внимание уделяется вопросам безопасности и защите персональных данных пользователей.

5. Открытый код и лицензирование: желательно использование открытых и свободно распространяемых технологий, чтобы избежать чрезмерных лицензионных платежей и снизить риск отказа поставщика.

Таким образом, выбор технологии проектирования должен базироваться на всестороннем анализе потребностей предприятия, ресурсов разработки и перспектив развития, что обеспечит создание надежной и эффективной системы автоматизации.

3.2 Выбор средств разработки

3.2.1 HTML5

HTML (HyperText Markup Language) - это стандартный язык разметки, который используется для создания веб-страниц. С помощью HTML можно определить структуру документа и разместить на нём текст, изображения, ссылки и другие элементы [11]. Этот язык состоит из тегов, каждый из которых определяет определенный элемент страницы. HTML - основной строительный блок веб-разработки и позволяет создавать интерактивные и креативные веб-страницы.

HTML5 (HyperText Markup Language) является основным языком разметки для создания веб-страниц. Он представляет собой набор тегов, которые определяют структуру документа: заголовки, абзацы, списки, таблицы, изображения и многое другое. HTML5 используется во всемирной паутине для создания контента, который отображается в веб-браузерах [12]. Для создания действительно красивых и функциональных веб-страниц, HTML5 часто комбинируется с CSS (Cascading Style Sheets) и JavaScript. HTML5 прост в освоении и является основой для изучения других технологий веб-разработки.

Преимущества HTML5:

1. Поддержка мультимедийного контента. HTML5 включает встроенные элементы `<audio>` и `<video>`, позволяющие легко интегрировать аудио и видеоконтент непосредственно в веб-страницу без необходимости использования сторонних плагинов, таких как Adobe Flash. Это делает веб-страницы легче и быстрее загружающимися.

2. Расширенные возможности форм. HTML5 вводит новые атрибуты и API для валидации форм, улучшает обработку ввода данных пользователями и добавляет новые типы полей ввода, такие как `email`, `date`, `color` и др., делая процесс заполнения форм проще и интуитивно понятнее.

3. Семантические элементы. Появились новые семантические элементы, такие как `<header>`, `<footer>`, `<article>`, `<section>`, `<aside>`, `<main>`, `<nav>`,

которые помогают улучшить структуру веб-страниц, повысить доступность и облегчить индексацию поисковиками.

4. Поддержка математических выражений (MathML). HTML5 интегрирует MathML, позволяющий удобно представлять сложные математические выражения и уравнения прямо в тексте веб-страницы.

5. Постоянное соединение с сервером (WebSocket). Технология WebSocket позволяет устанавливать двунаправленные соединения между клиентом и сервером, обеспечивая мгновенную передачу данных без необходимости постоянной перезагрузки страницы.

6. Реальное общение в браузере (WebRTC). Поддерживается технология WebRTC, позволяющая осуществлять голосовую и видеосвязь прямо в браузере без установки дополнительных плагинов или программного обеспечения.

7. Единый и простой DOCTYPE. Теперь используется единый универсальный DOCTYPE <!DOCTYPE html>, что значительно облегчает жизнь разработчикам и устраняет необходимость выбора между различными вариантами деклараций документа.

8. Работа в оффлайновом режиме Благодаря возможностям локального хранения данных и кэшированию приложений, веб-приложения могут функционировать даже тогда, когда отсутствует подключение к интернету.

9. Локальное хранилище данных HTML5 предоставляет механизмы для хранения данных на стороне клиента, такие как LocalStorage и SessionStorage, что повышает производительность и удобство работы пользователей².

10. Интеграция с устройствами Появилась возможность взаимодействия с аппаратными возможностями устройств, такими как камера, микрофон и геолокационные сервисы, открывая новые горизонты для мобильных и десктопных веб-приложений.

11. Графические возможности Интеграция WebGL и SVG позволяет создавать сложную графику и анимацию прямо в браузере, повышая уровень визуализации и интерактивности веб-приложений.

12. Повышение производительности Код HTML5 оптимизирован таким образом, что страницы грусятся быстрее и работают эффективнее, особенно на мобильных устройствах.

13. Увеличенная доступность Новые возможности HTML5 улучшают доступность веб-ресурсов для людей с ограниченными возможностями, благодаря поддержке ARIA-атрибутов и улучшению структуры страниц.

3.2.2 Язык С#

C# — это современный, объектно-ориентированный язык программирования, разработанный компанией Microsoft[31]. Он широко используется для создания различных приложений, включая веб-приложения, настольные программы и игры. Рассмотрим основные преимущества и недостатки C#.

Преимущества С#

1. C# поддерживает основные принципы объектно-ориентированного программирования, такие как инкапсуляция, наследование и полиморфизм, что позволяет создавать гибкие и масштабируемые приложения.

2. C# интегрирован с платформой .NET, предоставляя доступ к большому количеству библиотек и инструментов для разработки, упрощая процесс создания приложений.

3. C# является языком со строгой типизацией, помогая избежать многих ошибок на этапе компиляции и повышая надежность кода.

4. C# предлагает удобные конструкции для работы с асинхронными операциями, позволяя создавать высокопроизводительные приложения.

5. С помощью .NET Core и .NET 5/6 C# можно использовать для разработки приложений, работающих на различных операционных системах, включая Windows, macOS и Linux.

6. C# имеет большое и активное сообщество, обеспечивающее доступ к множеству ресурсов, библиотек и фреймворков, а также возможность получения помощи и обмена опытом.

Недостатки С#

1. Ограниченнная производительность. C# уступает в производительности низкоуровневым языкам, таким как C++ или С. Причина кроется в работе через промежуточный слой виртуальной машины, что ведет к дополнительным затратам ресурсов и снижению общей эффективности выполнения программ.

2. Высокое потребление ресурсов. Автоматическое управление памятью («сборка мусора») и работа через виртуальную машину требуют большего объема оперативной памяти и вычислительных ресурсов по сравнению с языками, использующими ручное управление ресурсами.

3. Относительная слабость в низкоуровневой разработке

C# не способен эффективно решать задачи, требующие непосредственного взаимодействия с оборудованием и операционной системой на низком уровне. Такие задачи традиционно реализуются на языках С/C++.

4. Историческая ориентация на Windows. Хотя современные версии C# стали кроссплатформенными благодаря технологиям .NET Core и .NET 5/6, ранее язык был жестко привязан к среде Windows. Эта особенность сохраняется в ряде аспектов и может осложнять миграцию существующих проектов на другие платформы.

5. Проблемы с безопасностью. Код, написанный на C#, легко подвергается декомпиляции, что потенциально открывает доступ к внутренним механизмам приложения и может использоваться злоумышленниками для анализа уязвимостей или кражи интеллектуальной собственности.

6. Стоимость коммерческих инструментов. Профессиональные инструменты разработки, такие как Visual Studio Enterprise, стоят недешево, что может стать серьезным барьером для малых компаний и индивидуальных разработчиков.

7. Сложность синтаксиса для новичков. Некоторые начинающие разработчики считают синтаксис C# излишне сложным и трудным для освоения по сравнению с простыми скриптовыми языками, такими как Python или JavaScript.

8. Менее развитая экосистема по сравнению с некоторыми другими языками. Экосистема инструментов и библиотек для C# считается менее разнообразной и развитой по сравнению с популярными языками общего назначения, такими как Java или Python

WPF (.NET Framework):

WPF (.NET Framework) – WPF (Windows Presentation Foundation) — это технология от компании Microsoft, входящая в состав .NET Framework, предназначеннная для создания настольных приложений с богатым пользовательским интерфейсом. WPF предлагает мощные средства для разработки интерфейсов, обеспечивая гибкость и возможности, которые трудно реализовать с использованием традиционных технологий создания пользовательских интерфейсов [2] [3].

Основные компоненты WPF:

1) XAML (Extensible Application Markup Language):

- Используется для определения интерфейсов пользователя.
- Позволяет описывать компоненты интерфейса в декларативном стиле.
- Поддерживает привязку данных, стили и шаблоны.

2) Visual Tree и Logical Tree:

- Logical Tree: отображает логическую структуру интерфейса.
- Visual Tree: отображает визуальную структуру, включающую все визуальные элементы, такие как кнопки, текстовые поля и другие контролы.

3) Bindings (Привязка данных):

- Позволяет синхронизировать данные между UI и бизнес-логикой.
- Поддерживает различные виды привязок, включая одно- и двустороннюю привязку.

4) Styles и Templates:

- Styles: определяют внешний вид и форматирование элементов интерфейса.
- Templates: позволяют изменять визуальное представление элементов без изменения их логики.

- Events (События):

- Поддерживает как стандартные события (например, нажатие кнопки), так и сложные маршрутизируемые события.

5) Animation и Media:

- Поддержка анимации для создания динамических интерфейсов.

- Возможность встраивания мультимедийного контента.

Преимущества WPF:

- Гибкость и мощь: позволяет создавать сложные пользовательские интерфейсы с продвинутыми графическими возможностями.

- Отделение логики от представления: Использование XAML для описания интерфейса позволяет отделить представление от бизнес-логики.

- Поддержка MVVM (Model-View-ViewModel): способствует созданию легко тестируемых и поддерживаемых приложений.

- Высокая производительность: благодаря использованию аппаратного ускорения для рендеринга графики.

- Унификация подходов: Единая платформа для работы с 2D и 3D графикой, мультимедиа, текстом и анимацией.

Недостатки и ограничения:

- Крутая кривая обучения: из-за своей мощи и гибкости WPF может быть сложен для изучения, особенно для новичков.

- Зависимость от Windows: Приложения, созданные с использованием WPF, работают только на платформе Windows.

- Производительность: при неосторожном использовании ресурсоемких компонентов может наблюдаться снижение производительности.

Примеры применения:

- Создание программ для работы с мультимедийным контентом.

- Разработка инструментов для моделирования и визуализации данных.

- Построение интерфейсов для научных и инженерных приложений.

- Создания визуального интерфейса

- Привязка данных, макетов, двухмерную и трёхмерную графику, анимации, стили, шаблоны, документы

WPF — это мощная и гибкая технология для разработки настольных приложений под Windows. Благодаря своей способности создавать сложные, богатые пользовательские интерфейсы и поддержке современных подходов к разработке, таких как MVVM, WPF остается популярным инструментом среди разработчиков. Однако для успешного использования этой технологии требуется хорошее понимание её возможностей и особенностей.

XAML (Extensible Application Markup Language) — это язык разметки, используемый для создания и инициализации объектов в .NET Framework [4], в частности, в таких технологиях, как Windows Presentation Foundation (WPF), Silverlight, Windows UI (WinUI) и Xamarin.Forms. XAML предоставляет декларативный подход к разработке пользовательских интерфейсов и объектов, упрощая создание сложных визуальных элементов и логики привязки данных.

Основные концепции и элементы XAML:

1) Декларативный синтаксис:

- Позволяет описывать элементы интерфейса в текстовом формате.
- Поддерживает вложенность элементов, что отражает иерархическую структуру UI.

2) Элементы и атрибуты:

- Элементы: представляют собой объекты, такие как кнопки, текстовые поля, контейнеры и т. д.
- Атрибуты: определяют свойства этих объектов, например, цвет, размеры, текст и т. д.

3) Пространства имен:

- Используются для организации и разделения элементов и свойств.
- Позволяют подключать пользовательские и сторонние библиотеки.

4) Привязка данных (Data Binding):

- Обеспечивает связь между элементами UI и данными приложения.
- Поддерживает одно- и двустороннюю привязку, а также привязку к коллекциям и объектам.

5) Ресурсы (Resources):

- Позволяют задавать общие стили, шаблоны и другие настройки, которые могут быть повторно использованы в разных частях приложения.

Преимущества XAML:

- Отделение логики от представления: Разметка интерфейса отделена от кода логики, что улучшает читаемость и упрощает поддержку.

- Гибкость и мощность: Возможность описывать сложные интерфейсы, анимации и привязку данных декларативным способом.

- Повторное использование: Ресурсы, стили и шаблоны могут быть использованы в разных частях приложения, способствуя уменьшению дублирования кода.

- Поддержка инструментов: Интеграция с инструментами разработки, такими как Visual Studio и Blend for Visual Studio, облегчает визуальное проектирование интерфейсов.

- Примеры использования XAML:

- WPF (Windows Presentation Foundation): Разработка настольных приложений с богатыми пользовательскими интерфейсами.

- Silverlight: Создание кроссплатформенных веб-приложений с богатым UI.

- Xamarin.Forms: Разработка мобильных приложений для Android, iOS и UWP.

- WinUI: Создание современных пользовательских интерфейсов для Windows.

XAML — это мощный и гибкий язык разметки, который значительно упрощает процесс создания и управления пользовательскими интерфейсами в различных технологиях .NET. Благодаря своей декларативной природе, XAML способствует созданию легко поддерживаемых, расширяемых и визуально привлекательных приложений.

3.3 Компоненты приложения

Для составления диаграммы компонентов приложения по теме "Подбор штор и обоев в помещении" нужно определить основные компоненты си-

стемы и их взаимодействия. В типичном приложении такого типа компоненты могут включать:

- компонент пользовательского интерфейса (UI) для работы с заказами и учетными данными;
- компонент бизнес-логики, реализующий правила выбора штор и обоев;
- компонент доступа к данным, который занимается сохранением и извлечением данных;
- компоненты интеграции с внешними системами (если есть);
- дополнительные компоненты, например, модуль авторизации и безопасности

Так как согласно правилам, итоговое приложение должно быть реализовано во frontend (html+css+js), архитектура будет соответствовать одноровневому приложению с модулями в рамках одной HTML-страницы, но диаграмма компонентов представляет концептуальное деление. Опишем диаграмму компонентов в текстовом виде с описанием компонентов и взаимодействий:

- так как инструмент визуализации не доступен, дам текстовое описание диаграммы компонентов;
- компоненты приложения для автоматизации учета и предварительных заказов;
- компонент пользовательского интерфейса (UI);
- обеспечивает взаимодействие пользователя с приложением;
- включает формы для ввода заказов, отображение списка заказов и отчетов;
- обрабатывает события и отправляет запросы к бизнес-логике;
- компонент бизнес-логики;
- реализует правила создания, изменения и удаления обоев и штор;
- компонент управления данными;
- при необходимости выполняется авторизация перед доступом к функциям;

- отчеты формируются на основе данных;
- такое разбиение упрощает развитие и поддержку приложения, обеспечивая разделение ответственности.

3.3.1 Результат проектирования

На основании анализа предметной области пункта 1 технического задания выпускной квалификационной работы мною был составлен словарь предметной области. В словаре представлены термины на русском языке.

В таблице 3.1 представлен словарь предметной области.

Таблица 3.1 – Сравнение протоколов SSE и WebSocket

Термин	Описание
Приложение	Программа, рассчитанная на непосредственное взаимодействие с пользователем.
Веб-сайт	Совокупность логически связанных между собой страниц.
Очистить выбор	Вернуть первоначальную картинку.
Кнопка	Элемент, при нажатии на который происходит перемещение между окнами.
Изображение	Графическое представление, содержащее визуальную информацию. В контексте данной программы это кадр, полученный с камеры.
Скрулла	Вернуть первоначальную картинку.
Очистить выбор	Это горизонтальная или вертикальная полоса прокрутки, элемент управления интерфейса, формы, веб-страницы и т д.

Моделирование вариантов использования

На основании анализа требований, в приложении должны реализовываться следующие прецеденты:

1. Прецедент «Подбор обоев». Прецедент позволяет пользователю перейти к окну интерьеров.
2. Прецедент «Просмотр интерьеров». Прецедент позволяет пользователю просматривать список из встроенных и пользовательских интерьеров, доступных для подбора обоев
3. Прецедент «Выбрать типовой интерьер». Прецедент позволяет пользователю выбрать один из просматриваемых типовых интерьеров.
4. Прецедент «Выбрать обои». Прецедент позволяет пользователю выбрать вариант обоев для текущего интерьера из предложенного набора.
5. Прецедент «Выбрать кастомный интерьер». Прецедент позволяет пользователю перейти к окну пользовательского интерьера, где можно загрузить собственное изображение помещения и выполнять подбор обоев на его основе.
6. Прецедент «Сгенерировать вариант обоев». Прецедент позволяет пользователю запустить генерацию варианта обоев с использованием сервиса искусственного интеллекта для текущего изображения интерьера.
7. Прецедент «Применить вариант». Прецедент позволяет пользователю применить выбранный или сгенерированный вариант обоев к текущему изображению интерьера.
8. Прецедент «Загрузить изображение». Прецедент позволяет пользователю загрузить собственное изображение интерьера из файла. Загруженное изображение отображается в окне кастомного интерьера и используется для дальнейшего подбора и генерации вариантов обоев.
9. Прецедент «Выбор штор». Прецедент позволяет пользователю перейти в окно подбора штор и выполнить выбор варианта цвета или оформления штор для интерьера.
10. Прецедент «Выход». Прецедент позволяет пользователю закрыть приложение.

На рисунке 3.1 представлена диаграмма прецедентов программной среды.

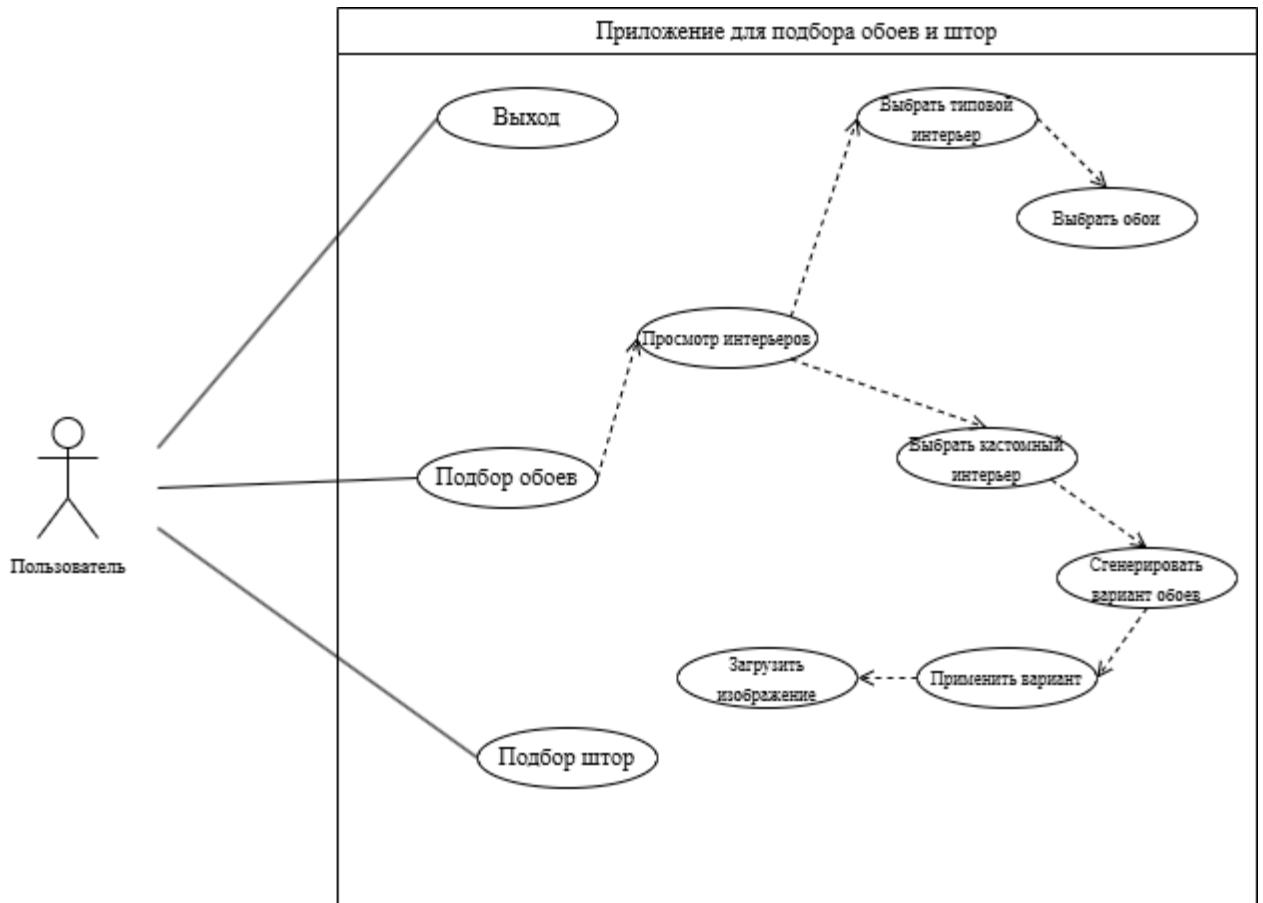


Рисунок 3.1 – Диаграмма прецедента пользователя для программы

Сценарий для прецедента «Подбор обоев»:

Основной исполнитель: пользователь.

Заинтересованные лица и их требования: пользователь хочет подобрать обои в комнату.

Предусловие: в программе нет ошибок.

Описание: пользователь запускает программу, нажимает кнопку «Подбор обоев».

Результат: открывается окно выбора интерьеров.

Сценарий для прецедента «Просмотр интерьеров»:

Основной исполнитель: пользователь.

Заинтересованные лица и их требования: пользователь хочет ознакомиться с доступными интерьерами, чтобы выбрать подходящий.

Предусловие: открыто окно подбора обоев с галереей интерьеров.

Описание: пользователь просматривает изображения интерьеров, последовательно выбирая миниатюры или переключаясь между вариантами.

Результат: пользователь визуально оценивает доступные интерьеры и может принять решение, какой интерьер выбрать для дальнейшего подбора обоев.

Сценарий для precedента «Выбрать типовой интерьер»:

Основной исполнитель: пользователь.

Заинтересованные лица и их требования: пользователь хочет работать с конкретным интерьером.

Предусловие: пользователь просматривает список типовых интерьеров в окне подбора обоев.

Описание: пользователь нажимает на один из интерьеров. Программа фиксирует выбранный интерьер и открывает для него окно выбора обоев.

Результат: выбранный интерьер становится активным, пользователь переходит к детальному подбору обоев для данного интерьера.

Сценарий для precedента «Выбрать обои»:

Основной исполнитель: пользователь.

Заинтересованные лица и их требования: пользователь хочет увидеть, как интерьер выглядит с определёнными обоями.

Предусловие: открыт интерьер в окне подбора обоев, доступен набор вариантов обоев.

Описание: пользователь выбирает один из вариантов обоев (по кнопке или миниатюре). Программа подставляет соответствующее изображение интерьера с выбранными обоями.

Результат: на экране отображается интерьер с выбранными обоями.

Сценарий для precedента «Выбрать кастомный интерьер»:

Основной исполнитель: пользователь.

Заинтересованные лица и их требования: пользователь хочет подобрать обои для собственного изображения комнаты.

Предусловие: приложение запущено, пользователь находится в главном окне или окне подбора обоев.

Описание: пользователь нажимает кнопку перехода к кастомному интерьеру. Программа открывает окно кастомного интерьера с возможностью загрузки изображения.

Результат: пользователь переходит в режим работы с кастомным интерьером.

Сценарий для прецедента «Сгенерировать вариант обоев»:

Основной исполнитель: пользователь.

Заинтересованные лица и их требования: пользователь хочет получить автоматически сгенерированный вариант обоев для своей комнаты.

Предусловие: в окне кастомного интерьера загружено исходное изображение, настроен доступ к сервису ИИ-генерации.

Описание: пользователь нажимает кнопку «Сгенерировать обои ИИ». Программа отправляет изображение и текстовый запрос в сервис генерации, периодически проверяет статус и после завершения загружает сгенерированное изображение.

Результат: в правой части окна отображается сгенерированный вариант интерьера с новыми обоями.

Сценарий для прецедента «Применить вариант»:

Основной исполнитель: пользователь.

Заинтересованные лица и их требования: пользователь хочет утвердить понравившийся вариант обоев.

Предусловие: в окне кастомного интерьера сгенерирован и отображён вариант обоев.

Описание: пользователь нажимает кнопку «Применить вариант». Программа заменяет основное изображение интерьера на сгенерированное и сохраняет путь к выбранному варианту.

Результат: выбранный вариант обоев считается применённым, основной вид интерьера обновлён.

Сценарий для прецедента «Загрузить изображение»:

Основной исполнитель: пользователь.

Заинтересованные лица и их требования: пользователь хочет использовать своё фото комнаты для подбора обоев.

Предусловие: открыто окно кастомного интерьера.

Описание: пользователь нажимает кнопку «Загрузить изображение», в диалоге выбирает файл с изображением комнаты. Программа считывает файл и отображает загруженное изображение в области просмотра.

Результат: пользователь видит свой интерьер в окне программы и может выполнять дальнейший подбор обоев.

Сценарий для прецедента «Выход»:

Основной исполнитель: пользователь.

Заинтересованные лица и их требования: пользователь хочет выйти из приложения.

Предусловие: в программе нет ошибок.

Описание: пользователь запускает программу, нажимает кнопку «выход».

Результат: Приложение закрывается

Сценарий для прецедента «Подбор штор»:

Основной исполнитель: пользователь.

Заинтересованные лица и их требования: пользователь хочет подобрать цвет для штор.

Предусловие: в программе нет ошибок.

Описание: пользователь запускает программу, нажимает кнопку «Подбор штор».

Результат: открывается окно в котором пользователь может изменять цвет штор выбрав любой цвет из 16 цветовых оттенков.

Основной успешный сценарий:

1. Пользователь запускает готовый проект.
2. Пользователь попадает на главное окно.
3. Пользователь выбирает какой подбор его интересует.
4. Система без ошибок запускает окно выбора интерьера

5. Пользователь выбирает интерьер
6. Без ошибок запускается окно выбора обоев для данного интерьера
7. Пользователь выбирает интересующие его обои
8. Система без ошибок меняет картинку, на пользовательскую
9. Пользователь хочет очистить выбор, нажав на кнопку
10. Система без ошибок меняет картинку на стандартную без обоев
11. Пользователь нажимает кнопку вернуться назад
12. Система без ошибок возвращает его на предыдущее окно
13. Пользователь выбирает подбор штор
14. Система без ошибок запускает окно выбора штор
15. Пользователь выбирает цвет
16. Система без ошибок меняет картинку на ту которую выбрал пользователь

17. Пользователь хочет выйти из приложения, нажав на кнопку выход
18. Система без ошибок закрывает приложение

Бизнес-правила:

1. Качественный контент: предоставление достоверной информации об обоях и шторах.
2. Понятный и удобный интерфейс: обеспечение удобства навигации по сайту, легкости доступа к выбору штор и обоев, интуитивно понятного дизайна.
3. Контент-маркетинг: создание уникального контента для привлечения и удержания аудитории.
4. Все данные о лицах должны быть защищены от несанкционированного доступа.

Описание структур и форматов данных

Изображения: хранятся в форматах jpeg, jpg, png. В отдельно созданной папке resources

Содержимое окон:

Классы кнопок button, изображения различных форматов, 13 различных окон формата .xaml связанные пространством имен vkr

Используемые технологии:

Язык программирования C#.

Технология многостраничности и полиморфизма библиотеки WPF.

XAML-разметка с использованием тегов для структурирования контента.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бишоп, Дж. C# в кратком изложении / Дж. Бишоп, Н. Хорспул. - М.: Бином. Лаборатория знаний, 2019. - 472 с. Текст: непосредственный.
2. Бойко В.В., Савинков В.М. Проектирование баз данных информационных систем. - М.: Финансы и статистика, 2020. - 351 с.
3. Вагнер, Билл C# Эффективное программирование / Билл Вагнер. - М.: ЛОРИ, 2019. - 320 с. Текст: непосредственный.
4. Грабер М. Введение в SQL. - М.: ЛОРИ, 2021. - 380 с.
5. Зиборов, В.В. Visual C# 2017 на примерах / В.В. Зиборов. - М.: БХВ-Петербург, 2021. - 480 с. Текст: непосредственный.
6. Зиборов, Виктор Visual C# 2010 на примерах / Виктор Зиборов. - М.: "БХВ-Петербург 2019. - 432 с. Текст: непосредственный.
7. Бретт, М. PHP и MySQL. Исчерпывающее руководство / М. Бретт. – Санкт-Петербург : Питер, 2016 г. – 544 с. – ISBN 978-5-496-01049-8. – Текст : непосредственный.
8. Веру, Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру. – Санкт-Петербург : Питер, 2016 г. – 336 с. – ISBN 978-5-496-02082- Текст : непосредственный.
9. Гизберт, Д. PHP и MySQL / Д. Гизберт. – Москва : НТ Пресс, 2013 г. – 320 с. – ISBN 978-5-477-01174-2. – Текст : непосредственный.
10. Дэкетт, Д. HTML и CSS. Разработка и создание веб-сайтов / Д. Дэкетт. – Москва : Эксмо, 2014 г. – 480 с. – ISBN 978-5-699-64193-2. – Текст : непосредственный.
11. Грофф, Дж. P. SQL: полное руководство / Грофф Дж. Р., Вайнберг П. Н., Оппель Э. Дж. — 3-е изд. — Санкт-Петербург : Диалектика, 2020 г. — 960 с. - ISBN 978-5-907114-26-5. – Текст : непосредственный.
12. Шварц, Б. MySQL. Оптимизация производительности / Шварц Б., Зайцев П., Ткаченко В., Заводны Дж., Ленц А., Бэллинг Д. — 2-е изд. — Санкт-Петербург : Символ-Плюс, 2010 г. — 832 с. - ISBN 978-5-93286-153-0. – Текст : непосредственный.

13. Кириченко, А. В. Laravel для WEB-разработчиков. Практическое руководство по созданию профессиональных сайтов / А. В. Кириченко, Е. В. Дубовик. — Санкт-Петербург :Наука и Техника, 2021 г. - 432 с. - ISBN: 978-5-94387-726-1. – Текст : непосредственный.
14. Стоун, Б. The Everything Store. Джейф Безос и эра Amazon / Б. Стоун. – Москва : Азбука, 2021 г. – 448 с. – ISBN: 978-5-389-15490-2. – Текст : непосредственный.
15. Вирт, Н. Алгоритмы и структуры данных / Н. Вирт. – Москва : ДМК Пресс, 2010 г. – 274 с. – ISBN: 978-5-04-034809-1. – Текст : непосредственный.
16. Мандел, Т. Разработка пользовательского интерфейса / Т. Мандел. – Москва : ДМК Пресс, 2019 г. – 420 с. – ISBN: 978-5-04-195060-6. – Текст : непосредственный.
17. Игорь, Зорин Как трудного клиента сделать счастливым. Правила, приемы и техники работы / Зорин Игорь. // М.: Феникс, 2017. — 404 с.
18. Илюшечкин В.М. Основы использования и проектирования баз данных:учеб. Пособие - М.: Юрайт: ИД Юрайт, 2011. - 213 с.
19. Лапина, Т.И. Информационные системы. Проектный практикум по выполнению ВКР : учебное пособие / Т. И. Лапина. – М.: ЮЗГУ, ЗАО «Университетская книга», 2016. – 99с.
20. Макнейл, П. Веб-дизайн. Книга идей веб-разработчика / П. Макнейл. – СПб.: Питер, 2017. – 480 с.
21. Начало работы с Вебом . . . – Режим доступа: <https://developer.mozilla.org/ru/docs/Learn/Gettingstarted> (дата обращения: 10. 03.2025).
22. Недякин, Максим Искренний сервис. Как мотивировать сотрудников сделать для клиента больше, чем достаточно. Даже когда шеф не смотрит / Максим Недякин. // М.: Манн, Иванов и Фербер, 2017. — 811 с.
23. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS / Р. Никсон. – СПб.: Питер, 2013. – 560 с.

24. Официальный сайт uCoz . – Режим доступа: <https://www.ucoz.ru/> (дата обращения: 12. 03.2025).
25. Портер, Майкл Конкурентная стратегия. Методика анализа отраслей и конкурентов / Майкл Портнер. // М.: Альпина Паблишер, 2015. — 456 с.
26. Прокин Александр Александрович, Богатырская Валентина Алексеевна, Сергушина Елена Сергеевна, Клиников Владислав Юрьевич Основы разработки эффективного веб-сайта // E-Scio. 2018. №3 (18). URL: <https://cyberleninka.ru/article/n/osnovy-razrabortki-effektivnogo-veb-sayta> (дата обращения: 18. 03.2025).
27. Сырых, Ю. Современный веб-дизайн. Настольный и мобильный/ Ю. Сырых. – М.: Диалектика, 2019. – 384 с.
28. Уроки в Фотошопе по созданию графики для сайта и приложений . – Режим доступа: <https://photoshop-master.ru/lessons/web-and-app/> (дата обращения: 20. 03.2025).