Введение

CollectiblesVault - Это десктопное приложение, предназначенное для коллекционеров, которым важно эффективно управлять своей коллекцией предметов.

Актуальность проекта обусловлена тем, что коллекционирование требует системного подхода к учету экспонатов, их классификации и сохранению данных. Традиционные способы, такие как ведение записей вручную или использование стандартных таблиц, часто оказываются неудобными и не соответствуют потребностям пользователей. CollectiblesVault решает эти проблемы, предлагая удобное решение для хранения, поиска и анализа коллекции.

Цель разработки — создание удобного и интуитивно понятного приложения, позволяющего коллекционерам регистрироваться в системе, добавлять и редактировать записи о предметах коллекции, загружать изображения, управлять категориями и формировать отчёты.

Целевая аудитория включает как начинающих коллекционеров, так и опытных пользователей, которым важно иметь доступ к инструменту для ведения систематизированного учета своих предметов.

Основные функции приложения включают регистрацию и авторизацию пользователей, ввод данных о коллекционных предметах, настройку категорий, создание отчетов по периодам и категориям, а также экспорт данных в различных форматах.

Структура отчета включает несколько ключевых разделов, которые подробно раскрывают все этапы разработки. В разделе, посвященном методам разработки, рассматриваются выбор технологий, процесс проектирования интерфейса и реализация основных функций. Далее в отчете приводится анализ полученных результатов, включающий описание функциональных возможностей приложения, удобства его использования и оценки производительности. В заключительном разделе отчета проводится итоговая оценка эффективности приложения, рассматриваются возможные направления для его дальнейшего развития, включая возможные улучшения интерфейса, расширение функциональности и внедрение дополнительных технологий для повышения удобства работы пользователей.

1. Проектирование ПП

Проектирование программного продукта — это процесс создания архитектуры, структуры и детальных спецификаций будущей программы перед её непосредственной разработкой. Проектирование позволяет обеспечить качество, масштабируемость и поддержку программного продукта в будущем.

* 1. Диаграмма связей

Диаграмма связей (рисунок 1) является удобным инструментом визуализации, который отображает взаимосвязи между компонентами системы. С её помощью можно проследить логику взаимодействия отдельных элементов приложения, выявить ключевые зависимости и понять общие принципы функционирования системы.

Работа с приложением начинается с главного окна, где пользователь проходит регистрацию или авторизацию (если у него уже есть аккаунт). После успешного входа открывается главное меню, выступающее в роли центрального узла навигации. Отсюда пользователь получает доступ ко всем основным функциям программы.

Подобная структура делает систему интуитивно понятной: пользователь последовательно переходит от общего интерфейса к конкретным возможностям приложения. Диаграмма чётко отражает эту логику, демонстрируя взаимосвязь между аутентификацией, главным меню и остальными модулями системы.

A diagram of a diagram

AI-generated content may be incorrect.

Рисунок 1 – диаграмма структуры приложения

1.2 Диаграмма сценариев

Диаграмма сценариев, представленная на рисунке 2, является графическим отображением взаимодействия между пользователями и системой, демонстрирующим последовательность действий для выполнения различных задач.

На схеме отражены две основные категории пользователей. Гости, не прошедшие авторизацию, имеют ограниченный функционал и могут только приступить к процедуре регистрации. После успешного создания учетной записи пользователь получает доступ ко всем возможностям приложения.

Для авторизованных пользователей предусмотрен расширенный функционал. Через раздел «Ввод данных» осуществляется добавление новой информации в систему. Функция «Просмотр данных» позволяет работать с существующими записями. Дополнительно доступен раздел «Настройки», где можно персонализировать параметры системы в соответствии с индивидуальными предпочтениями.

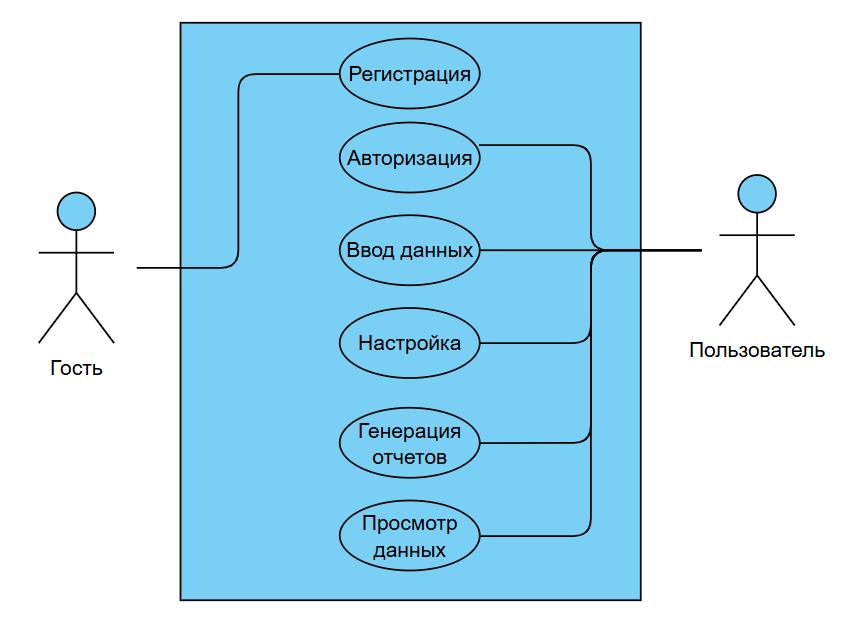


Рисунок 2 – Диаграмма сценариев приложения

1.3 Прототипирование и дизайн

Прототипирование помогает быстро проверить работоспособность идеи через создание упрощённых моделей, а дизайн превращает эти решения в удобные, функциональные и визуально привлекательные продукты.

На рисунке 3 представлен прототип, который демонстрирует логику работы приложения. Из начального экрана есть переход либо на страницу входа, либо на страницу регистрации. После входа/регистрации пользователь попадает на главную страницу, сразу попадая на страницу коллекций. С этой страницы сразу обеспечивается доступ ко всем остальным страницам приложения, таким образом переход по ним досупен благодаря всегда доступной вкладке со страницами сверху. На этой же вкладке расположена кнопка выхода из приложения.

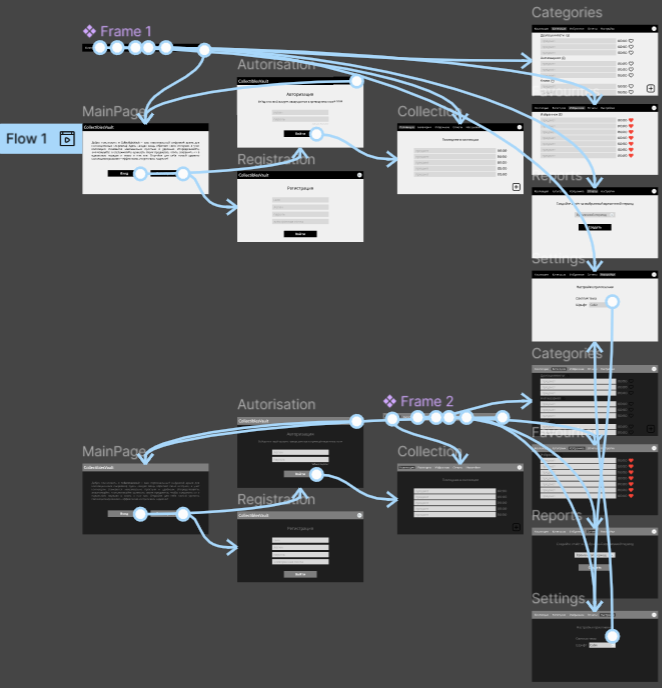


Рисунок 3 – прототип приложения

На рисунке 4 представлен дизайн начального экрана приложения с описанием приложения. В интерфейсе представлена кнопка, которая предусматривает переход на страницу входа.

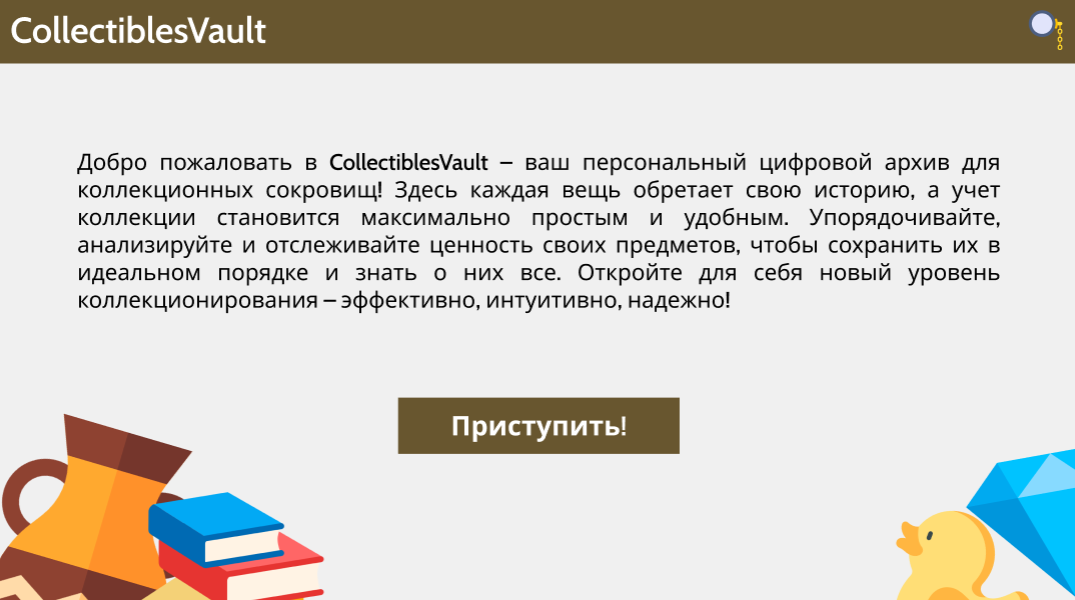


Рисунок 4 – дизайн начального экрана

На рисунке 5 представлен дизайн страницы категорий предметов с соответствующими разделами. В интерфейсе представлены верхнее меню перехода между страницами, кнопки добавления предметов по категориям, кнопки добавления предметов в избранное и кнопка добавления новой категории.

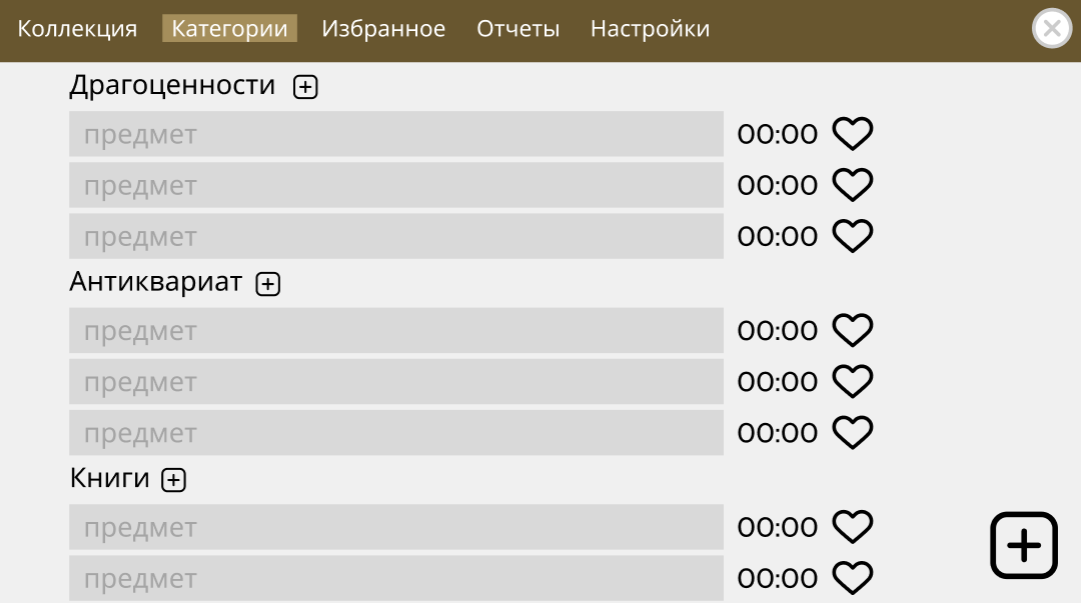


Рисунок 5 – дизайн страницы категорий

На рисунке 6 представлен дизайн страницы создания отчётов за выбранный период. В интерфейсе представлены верхнее меню перехода между страницами, выпадающий список выбора временного периода и кнопка создания отчёта.

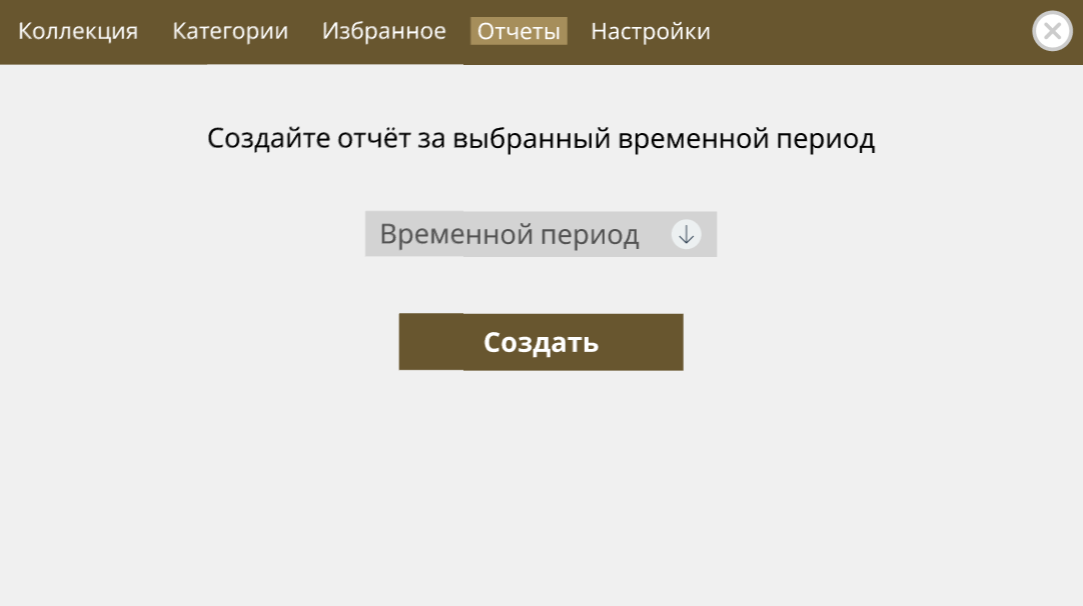


Рисунок 6 – дизайн страницы создания отчетов

На рисунке 7 представлен QR-код, ведущий на страницу с доступом к интерактивной версии прототипа приложения.



Рисунок 7 – QR-код к прототипу

2.1 Описание средств разработки

Visual Paradigm включен в процесс разработки как инструмент, обеспечивающий комплексное проектирование архитектуры через UML-диаграммы и формализацию требований к интерфейсу на начальных этапах.

Figma была выбрана в качестве основного инструмента проектирования интерфейсов благодаря своей облачной платформе, которая обеспечивает эффективную совместную работу команды в режиме реального времени. Возможность создания интерактивных прототипов позволяет проводить тестирование пользовательского опыта на ранних этапах разработки. Встроенные функции экспорта стилей и ресурсов оптимизируют процесс передачи дизайна, а кроссплатформенная доступность через веб-интерфейс устраняет проблемы совместимости с различными операционными системами.

Python с Tkinter был выбран в качестве основного инструмента для реализации графического интерфейса благодаря своей уникальной сочетаемости с языком Python. Будучи встроенным модулем стандартной библиотеки, он обеспечивает мгновенную готовность к работе без необходимости дополнительных установок или сложных конфигураций. Кроссплатформенная природа Tkinter гарантирует стабильную работу приложения на различных операционных системах, включая Windows, macOS и Linux, с единой кодной базой. Гибкость архитектуры Tkinter проявляется в поддержке множества готовых UI-компонентов, начиная от базовых кнопок и текстовых полей до более сложных элементов управления. Каждый компонент может быть тонко настроен через обширный набор параметров, позволяя создавать интерфейсы, соответствующие конкретным требованиям проекта. Система менеджеров компоновки предоставляет разработчику три различных подхода к расположению элементов, что особенно ценно при создании адаптивных интерфейсов. Глубокую интеграцию с Python можно считать ключевым преимуществом Tkinter. Эта связь позволяет напрямую соединять элементы интерфейса с бизнес-логикой приложения, написанной на Python.