Введение

HomeBudget — десктопное приложение для операционной системы Windows, представляющий собой систему учета личных финансов.

Актуальность разработки обусловлена современными тенденциями повышения финансовой грамотности населения и растущей потребностью в инструментах для контроля расходов. В условиях экономической нестабильности и увеличения количества финансовых операций, совершаемых ежедневно, приложение становится незаменимым помощником для тех, кто стремится к осознанному распоряжению своими денежными средствами. Особую значимость проект приобретает благодаря своей простоте использования и доступности для широкого круга пользователей.

Основная цель проекта заключается в создании удобного и функционального инструмента для систематического учета личных финансов. Приложение должно предоставлять пользователям возможность не только фиксировать доходы и расходы, но и анализировать свою финансовую деятельность через систему отчетов и визуализации данных, что в конечном итоге способствует более грамотному распределению бюджета.

Целевая аудитория приложения включает в себя различные группы пользователей: от студентов и молодых специалистов, только начинающих вести учет своих расходов, до семейных пар и фрилансеров, нуждающихся в постоянном контроле денежных потоков. Приложение будет особенно полезно тем, кто стремится к финансовой дисциплине и планированию долгосрочных сбережений.

Ключевые функции приложения охватывают весь цикл работы с личными финансами: от базовых операций регистрации и авторизации пользователей до сложных функций анализа данных. Основной функционал включает создание и редактирование финансовых записей, категоризацию расходов и доходов, формирование детализированных отчетов за различные периоды времени, а также возможность настройки интерфейса под индивидуальные предпочтения пользователя. Особое внимание уделено системе экспорта отчетов в популярные форматы для дальнейшей обработки.

Структура отчета отражает последовательность разработки приложения и включает несколько основных разделов. Вначале рассматриваются общие требования к проекту и его технические характеристики, затем подробно описывается процесс проектирования базы данных и пользовательского интерфейса. Особый раздел посвящен реализации функциональных возможностей приложения и методам тестирования его работоспособности. Завершается отчет анализом полученных результатов и выводами о степени соответствия готового продукта первоначальным требованиям технического задания.

1. Проектирование Программного продукта

Проектирование программного продукта — это этап разработки, на котором создается архитектура, функциональность и пользовательский интерфейс будущего приложения. Оно определяет, **как** система будет работать, прежде чем начнется ее непосредственная реализация в коде.

1. Диаграмма связей

Представленная диаграмма связей (рисунок 1) отражает архитектуру приложения для учета личных финансов, демонстрируя взаимосвязи между его основными функциональными модулями. В центре системы находится главное меню, выступающее в качестве ключевого узла навигации. Из него пользователь получает доступ к пяти основным разделам: модулю внесения и редактирования доходов, разделу учета расходов, компоненту управления категориями операций, блоку формирования аналитических отчетов и разделу персонализации настроек приложения.

Вход в систему начинается с меню входа, где пользователю предлагается два варианта действий — авторизация существующих пользователей или регистрация нового аккаунта. После успешного прохождения процедуры аутентификации открывается доступ к главному меню, которое обеспечивает интуитивно понятную навигацию между всеми функциональными разделами. Каждый из основных модулей связан с главным меню двусторонней связью, что позволяет не только быстро переходить к нужному разделу, но и возвращаться обратно после выполнения необходимых операций.

Архитектура приложения построена таким образом, чтобы обеспечить логичную последовательность работы с финансовыми данными: от внесения информации о доходах и расходах через соответствующие модули до анализа накопленных данных с помощью системы отчетов. Особое внимание уделено унификации интерфейса - все функциональные блоки спроектированы в едином стиле и связаны через центральное меню, что значительно упрощает процесс освоения приложения новыми пользователями. Отдельный модуль настроек позволяет адаптировать интерфейс под индивидуальные предпочтения, повышая удобство ежедневного использования системы.

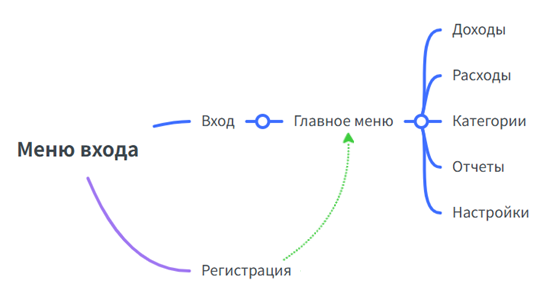


Рисунок 1 — Диаграмма связей приложения

1. Диаграмма сценариев

На представленной диаграмме сценариев (рисунок 2) представлено взаимодействие двух типов пользователей с системой — гостя и зарегистрированного пользователя. Гость может выполнять только базовые операции: регистрацию новой учетной записи с указанием необходимых персональных данных и последующую авторизацию в системе. После успешного входа в систему статус гостя изменяется на авторизованного пользователя, что открывает доступ к полному функционалу приложения.

Для авторизованных пользователей предусмотрены расширенные возможности работы с системой. Основной цикл взаимодействия включает ввод финансовых данных о доходах и расходах, настройку параметров учета, генерацию различных отчетов и просмотр накопленной информации в удобном формате визуализации. Все функциональные возможности взаимосвязаны и образуют логичную последовательность действий: от первичного ввода данных через их систематизацию к анализу и визуализации результатов.

Особенностью представленной схемы является четкое разделение прав доступа между гостями и зарегистрированными пользователями, а также последовательное расширение функциональных возможностей после прохождения процедуры авторизации. Архитектура взаимодействия построена по принципу постепенного усложнения — от простых операций регистрации к комплексной работе с финансовыми данными, что обеспечивает комфортное освоение системы новыми пользователями. Все элементы диаграммы связаны между собой, образуя целостную картину пользовательских сценариев работы с приложением.

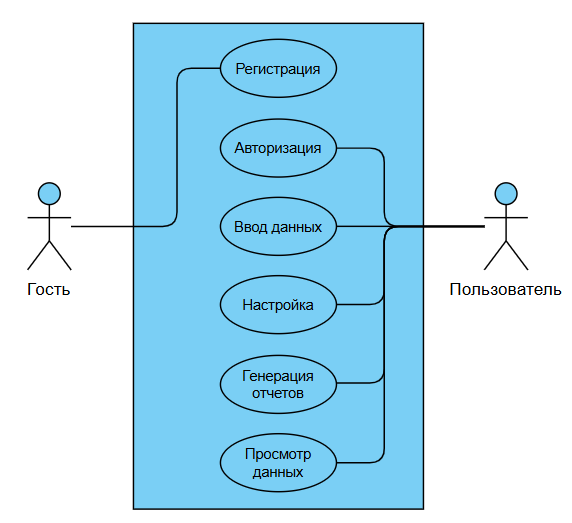


Рисунок 2 — Диаграмма сценариев приложения

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. С этой страницы сразу обеспечивается доступ ко всем остальным страницам приложения. 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 главной страницы. В интерфейсе представлено меню перехода между страницами, кнопка выхода из аккаунта и кнопка выхода из приложения.



Рисунок 5 – дизайн главной страницы

На рисунке 6 представлен дизайн страницы создания отчётов за выбранный период. В интерфейсе представлены кнопка возврата на предыдущую страницу, текстовое поле для временного периода, таблица с данными за соответствующий период и кнопка экспорта отчета.

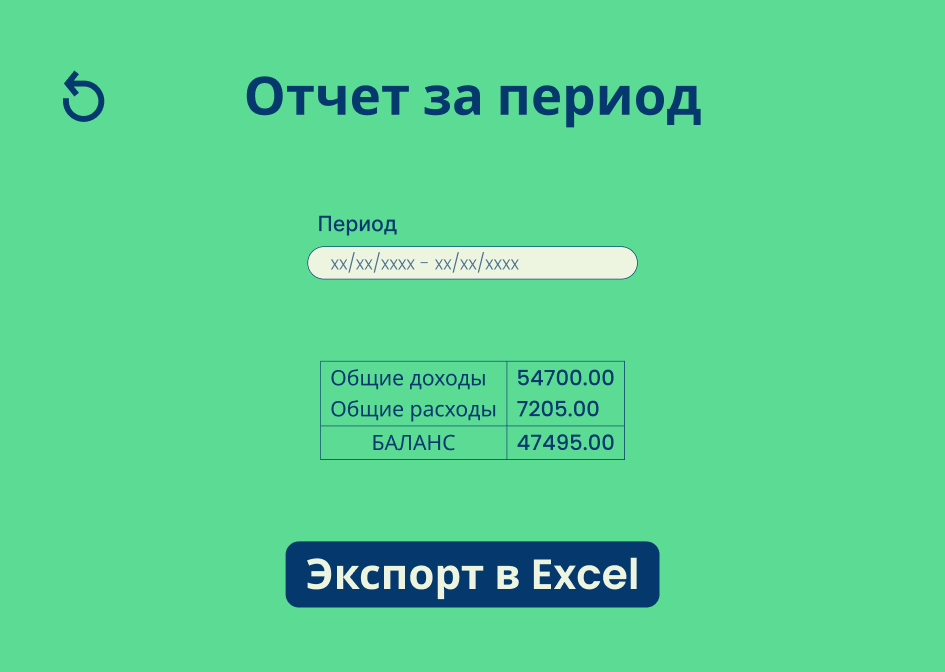


Рисунок 6 – дизайн страницы создания отчетов

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

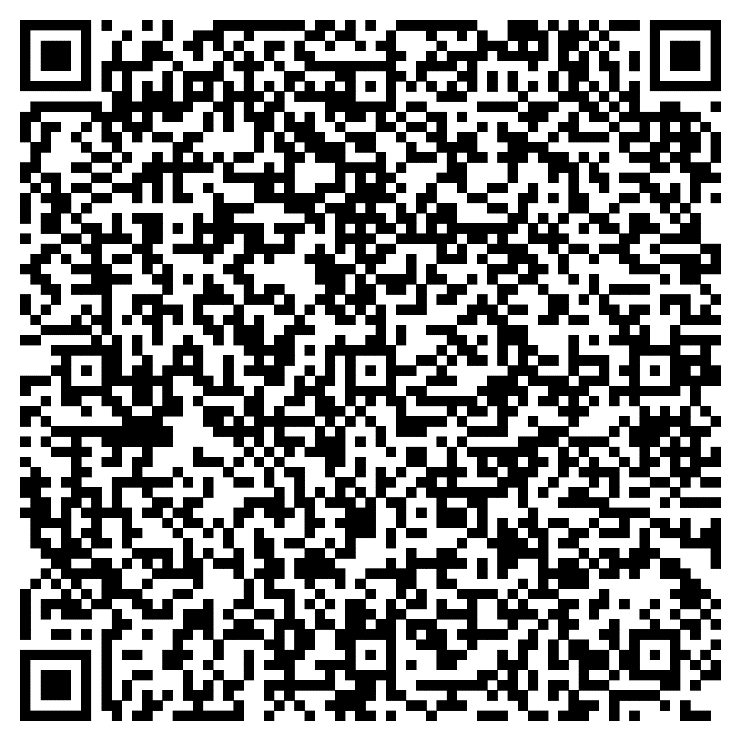


Рисунок 7 – QR-код к прототипу

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.