##### МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ярославский государственный университет им. П. Г. Демидова»

Кафедра вычислительных и программных систем

***Курсовая работа***

по направлению 09.03.03 Прикладная информатика в экономике

**Разработка desktop-приложения для контроля финансов**

###### Научный руководитель

###### Старший преподаватель, к.т.н.

##### К. В. Лагутина

« » 2023 г.

###### Студент группы ПИЭ—31БО

А. А. Скворцов

« » 2023 г.

Ярославль 2023 год

Содержание

[Введение 3](#_Toc136521466)

[1. Предметная область 5](#_Toc136521467)

[1.1 Постановка задачи 5](#_Toc136521468)

[1.2 Python 6](#_Toc136521469)

[1.3 PySide6 7](#_Toc136521470)

[1.3.1 Qt Core 8](#_Toc136521471)

[1.3.2 Qt Widgets 8](#_Toc136521472)

[1.3.3 Qt GUI 8](#_Toc136521473)

[1.3.4 Qt Charts 9](#_Toc136521474)

[1.3 SQLite3 9](#_Toc136521475)

[1.4 Qt Designer 9](#_Toc136521476)

[1.5 pyqtdeploy 10](#_Toc136521477)

[1.6 Pydroid 3 10](#_Toc136521478)

[2. Разработка desktop-приложения для контроля финансов 11](#_Toc136521479)

[2.1 Анализ требований 11](#_Toc136521480)

[2.2 Проектирование интерфейса 13](#_Toc136521481)

[2.3 Разработка функциональности 15](#_Toc136521482)

[2.3.1 Проектирование диаграмм 15](#_Toc136521483)

[2.3.2 Разработка desktop-приложения 19](#_Toc136521484)

[2.3.3 Описание функциональности desktop-приложения 21](#_Toc136521485)

[3. Портирование на Android 26](#_Toc136521486)

[Заключение 29](#_Toc136521487)

[Литература 30](#_Toc136521488)

# Введение

Современный образ жизни требует от нас осознанного управления финансами. Контроль над доходами и расходами становится неотъемлемой частью достижения финансовой стабильности и финансовых целей. Ранее люди вели учет своих транзакций вручную, записывая их в тетрадь или используя таблицы в электронных документах. Однако, этот процесс мог быть трудоемким и подвержен ошибкам.

Разработка desktop-приложения для учета доходов и расходов находит свое обоснование во многих преимуществах, которые оно предоставляет по сравнению с традиционными методами ведения финансового учета.

Во-первых, desktop-приложение обеспечивает высокую степень удобства и гибкости в управлении финансами. Пользователи могут легко и быстро вносить данные о своих доходах и расходах в удобной форме, используя интуитивно понятный интерфейс. Это позволяет сократить время и усилия, затрачиваемые на ведение учета, и сосредоточиться на более важных аспектах финансового планирования.

Во-вторых, desktop-приложение обладает более широкими возможностями для анализа и отображения данных. Оно позволяет пользователю создавать диаграммы, графики и отчеты, которые помогают визуализировать финансовую ситуацию, выявлять тренды и понимать, как распределяются доходы и расходы. Это даёт более полное представление о финансовом состоянии и помогает человеку принимать информированные решения в отношении своих финансовых целей.

В-третьих, desktop-приложение обеспечивает более высокий уровень безопасности и конфиденциальности данных. В отличие от онлайн-сервисов, где информация хранится на удаленных серверах и может быть подвержена риску взлома или утечки, desktop-приложение позволяет хранить данные локально на компьютере пользователя. Это создает дополнительный уровень защиты и обеспечивает большую степень конфиденциальности.

Таким образом, разработка desktop-приложения для учета доходов и расходов находит свое обоснование в удобстве использования, более широких возможностях анализа данных и повышенной безопасности по сравнению с традиционными методами ведения финансового учета.

# **1. Предметная область**

## 1.1 Постановка задачи

Задача, рассматриваемая в данной работе, заключается в разработке desktop-приложения, которое предоставит пользователю удобный и надежный инструмент для учета расходов и доходов. Приложение позволит пользователям записывать, категоризировать и анализировать свои финансовые операции. Дополнительной задачей в рамках этой работы является портирование приложения на платформу Android.

Задача разбита на несколько этапов:

1. Анализ требований:

* Изучение потребностей пользователей в учете расходов и доходов.
* Определение основных функций и возможностей приложения, таких как запись транзакций, категоризация, анализ данных и генерация отчетов.

1. Проектирование интерфейса:

* Разработка пользовательского интерфейса, обеспечивающего удобное и интуитивно понятное взаимодействие с приложением.
* Создание дизайна, который облегчит пользователю ввод и просмотр финансовых операций.

1. Разработка функциональности:

* Создание основной функциональности приложения, включая возможности регистрации, авторизации, ввода и хранения данных о расходах и доходах.
* Разработка системы категоризации, позволяющей пользователю классифицировать транзакции по различным параметрам (например, по типу расходов или источнику доходов).
* Реализация аналитических возможностей, таких как генерация диаграмм и отчетов для визуализации и анализа финансовых данных.

1. Портирование приложения на Android

Ожидаемым результатом работы является готовое desktop-приложение для учета расходов и доходов, которое поможет пользователям легко и эффективно контролировать свои финансы. Портирование приложения на Android расширит охват пользователей и предоставит им возможность использовать приложение на своих мобильных устройствах.

## 1.2 Python

Python — это язык программирования, который широко используется в интернет-приложениях, разработке программного обеспечения, науке о данных и машинном обучении (ML). Разработчики используют Python, потому что он эффективен, прост в изучении и работает на разных платформах.

Язык Python имеет следующие преимущества:

* Python имеет большую стандартную библиотеку, содержащую многократно используемые коды практически для любой задачи.
* Активное сообщество Python состоит из миллионов поддерживающих разработчиков со всего мира. При возникновении проблем сообщество поможет в их решении.
* Python можно переносить на различные операционные системы: Windows, macOS, Linux и Unix.

Язык Python имеет несколько стандартных примеров использования при разработке приложений, в числе которых:

Разработка программного обеспечения [1]

Разработчики программного обеспечения часто используют Python для различных задач разработки и программных приложений, среди которых:

* Разработка прототипов программного обеспечения
* Разработка настольных приложений с использованием библиотек графического пользовательского интерфейса (ГПИ)
* Разработка игр: от простых текстовых игр до сложных видеоигр

## 1.3 PySide6

PySide6 предоставляет удобный и простой в использовании интерфейс для создания GUI-приложений на Python. С помощью PySide6 можно создавать современные и красивые приложения, работающие на различных платформах, включая Windows, macOS и Linux.

Одним из ключевых преимуществ использования PySide6 является его кроссплатформенная совместимость. Используя PySide6, вы можете написать приложение один раз и развернуть его на нескольких платформах без необходимости вносить какие-либо изменения в код. Это экономит ваше время и усилия, а также помогает гарантировать, что ваше приложение выглядит и ведет себя одинаково на всех платформах.

Еще одним важным преимуществом PySide6 является богатый набор GUI компонентов и виджетов. PySide6 предоставляет большую коллекцию готовых компонентов, которые вы можете использовать для создания пользовательского интерфейса вашего приложения. А если вам нужно создать собственные виджеты или компоненты, PySide6 позволяет легко это сделать благодаря интуитивно понятному и гибкому API.

PySide6 также предоставляет мощные инструменты для кастомизации, что позволяет создавать визуально привлекательные и отзывчивые пользовательские интерфейсы, которые отлично смотрятся на любом устройстве. А благодаря поддержке PySide6 механизма сигналов и слотов Qt, вы можете легко связать ваш пользовательский интерфейс с логикой и данными вашего приложения.

В итоге можно сказать, что PySide6 — это мощный кроссплатформенный GUI фреймворк для Python, который позволяет легко создавать красивые и отзывчивые приложения, работающие на разных платформах. [2]

### 1.3.1 Qt Core

Qt Core — основной модуль, который содержит все базовые средства Qt. На его основе построены все другие модули.

Модуль Qt Core содержит ядро функциональности, не касающейся GUI. [7]

### 1.3.2 Qt Widgets

Qt Widgets предоставляет набор элементов пользовательского интерфейса для создания классических пользовательских интерфейсов в стиле настольных компьютеров.

Виджеты являются основными элементами для создания пользовательских интерфейсов в Qt. Виджеты могут отображать данные и информацию о состоянии, принимать пользовательский ввод и служить контейнером для других виджетов, которые должны быть сгруппированы вместе. [8]

### 1.3.3 Qt GUI

Модуль Qt GUI предоставляет классы для интеграции оконной системы, обработки событий, интеграции OpenGL и OpenGL ES, двухмерной графики, базовой визуализации, шрифтов и текста. Эти классы используются внутри технологий пользовательского интерфейса Qt, но также могут быть использованы напрямую, например, для написания приложений, использующих низкоуровневые графические API OpenGL ES. [9]

### 1.3.4 Qt Charts

Модуль Qt Charts предоставляет набор простых в использовании компонентов диаграмм. Он использует Qt Graphics View Framework, поэтому диаграммы можно легко интегрировать в современные пользовательские интерфейсы. Диаграммы Qt могут использоваться как типы QWidgets, QGraphicsWidget или QML. Пользователи могут легко создавать впечатляющие графики, выбирая одну из тем диаграмм. [11]

## 1.3 SQLite3

SQLite3 – это C библиотека, реализующая легковесную дисковую базу данных (БД), не требующую отдельного серверного процесса и позволяющую получить доступ к БД с использованием языка запросов SQL. Некоторые приложения могут использовать SQLite3 для внутреннего хранения данных. Также возможно создать прототип приложения с использованием SQLite3, а затем перенести код в более многофункциональную БД, такую как PostgreSQL или Oracle. [3]

## 1.4 Qt Designer

Qt Designer — это инструмент Qt для проектирования и создания графических пользовательских интерфейсов (GUI) с помощью виджетов Qt. Вы можете создавать и настраивать окна или диалоговые окна по принципу "что видишь, то и получаешь" (WYSIWYG) и тестировать их, используя различные стили и разрешения.

Виджеты и формы, созданные с помощью Qt Designer, легко интегрируются с программируемым кодом, используя механизм сигналов и слотов Qt, так что вы можете легко назначать поведение графических элементов. Все свойства, заданные в Qt Designer, могут быть динамически изменены в коде. Кроме того, такие возможности, как продвижение виджетов и пользовательские плагины, позволяют использовать собственные компоненты в Qt Designer. [10]

## 1.5 pyqtdeploy

pyqtdeploy — это инструмент, который в сочетании с другими инструментами, поставляемыми с Qt, позволяет развертывать приложения PyQt, написанные на Python v3.5 или более поздней версии. Он поддерживает развертывание на настольных платформах (Linux, Windows и macOS) и на мобильных платформах (iOS и Android). На его дизайн в значительной степени повлияла необходимость поддержки мобильных платформ и кросс-компиляции. [12]

## 1.6 Pydroid 3

Pydroid 3 — это приложение для мобильных устройств на операционной системе Android, которое предоставляет среду разработки и интерпретатор Python. Оно позволяет пользователям писать, отлаживать и выполнять программы на языке Python непосредственно на своих мобильных устройствах. [13]

# **2. Разработка desktop-приложения для контроля финансов**

## 2.1 Анализ требований

В рамках данной работы автору была поставлена задача разработать desktop-приложение для отслеживания финансовых операций.

Задача

Цель: разработать приложение для учета доходов и расходов человека с возможностью записи транзакций, категоризации, анализа данных и генерации отчетов.

Основные функции и возможности приложения:

1. Регистрация и авторизация:

* Пользователь может зарегистрироваться в программе.
* Зарегистрированные данные пользователя сохраняются между запусками приложения.
* Пользователь может авторизоваться при повторном входе в программу.

1. Учет доходов:

* Пользователь может добавлять доходы в следующие категории: зарплата, премия, инвестиционные доходы и подарок.
* При добавлении дохода пользователь указывает соответствующую категорию, сумму и дату транзакции.

1. Учет расходов:

* Пользователь может добавлять расходы в следующие категории: продукты, бытовые товары, оплата ЖКХ, связь и интернет, транспорт, развлечения.
* При добавлении расхода пользователь указывает соответствующую категорию, сумму и дату транзакции.

1. Управление записями:

* Пользователь может изменять и удалять существующие записи о доходах и расходах.
* Записи о доходах и расходах хранятся и отображаются упорядоченными по дате.

1. Анализ данных:

* Пользователь может просматривать списки доходов и расходов по дате.
* Пользователь может просматривать списки доходов и расходов по категориям.
* Пользователь может просматривать общую сумму доходов и расходов.
* Пользователь может просматривать суммы доходов и расходов по категориям за заданный период.

1. Генерация отчетов:

* Пользователь может генерировать отчеты о доходах и расходах за заданный период.
* Отчеты могут включать общую сумму доходов и расходов, суммы по категориям и другую статистическую информацию.

Дополнительные требования:

* Приложение должно быть разработано на языке программирования Python.
* Для разработки графического интерфейса пользователя следует использовать PySide6.
* Для хранения данных о доходах и расходах следует использовать SQLite3.

Задача содержит основные функции и возможности приложения для учета доходов и расходов. Разработчик должен руководствоваться этими требованиями при создании приложения, чтобы обеспечить его соответствие ожиданиям и функциональность по заданию.

## 2.2 Проектирование интерфейса

При разработке desktop-приложения для учета расходов и доходов, особое внимание было уделено созданию графического пользовательского интерфейса, который обеспечит удобное и интуитивно понятное взаимодействие с приложением. При этом автор руководствовался рядом принципов, которые помогли достичь этой цели: [4] – [5]

1. Интуитивная понятность интерфейса: человеку не требуется объяснения о том, как пользоваться приложением, так как его функциональность и взаимодействие с ним интуитивно понятны.
2. Ориентирование на задачи пользователей: разработка интерфейса началась с определения задач, для которых приложение предназначено, чтобы обеспечить наиболее удобное использование продукта.
3. Учет поведения и привычек пользователей: известные элементы интерфейса были сохранены, а новые элементы были созданы таким образом, чтобы они были интуитивно понятными и соответствовали привычкам пользователей.
4. Форма и пропорции: основная форма элементов интерфейса прямоугольная с скругленными углами или круглая. Также обеспечено соблюдение пропорций для создания гармоничного внешнего вида.
5. Использование цвета: для элементов интерфейса были выбраны теплые цвета, учитывая психологию восприятия, чтобы обеспечить приятное визуальное впечатление.
6. Выделение основных элементов интерфейса: часто используемые элементы были выделены размером или цветом, чтобы пользователи могли быстро и легко обращаться к ним.
7. Удобный размер элементов: элементы интерфейса сделаны не слишком маленькими, чтобы пользователю было легко попадать по ним.
8. Монотонность интерфейса: для обеспечения быстрой привыкаемости и автоматизации действий пользователей, интерфейс разработан в монотонном стиле, где каждое действие может быть выполнено только одним способом.
9. Обратная связь от элементов интерфейса: элементы интерфейса отвечают на действия пользователя, например, при клике, чтобы пользователь мог видеть, что его действие было замечено и обработано.

В результате проектирования интерфейса были достигнуты следующие важные цели:

1. Пользовательский интерфейс приложения был разработан с учетом принципов удобства и интуитивной понятности. Каждый элемент и функция приложения были размещены таким образом, чтобы пользователю не требовалось объяснять, как им пользоваться.
2. Дизайн приложения был создан с целью облегчить пользователю ввод и просмотр финансовых операций. Интуитивно понятные элементы интерфейса, такие как кнопки, формы ввода и таблицы, были разработаны с учетом привычек и ожиданий пользователей. Основные функции приложения были выделены и сделаны легко доступными, позволяя пользователям быстро и удобно выполнять необходимые операции.

Благодаря эффективному проектированию интерфейса пользователи получат удобный и интуитивно понятный инструмент для учета доходов и расходов. Они смогут легко добавлять и просматривать свои финансовые операции, анализировать данные и принимать осознанные финансовые решения на основе понятного и функционального интерфейса приложения.

## 2.3 Разработка функциональности

### 2.3.1 Проектирование диаграмм

Перед началом разработки автором было принято решение построить диаграммы вариантов использования, диаграммы классов и диаграммы последовательности.

1. Диаграмма вариантов использования [6]

Диаграмма вариантов использования (англ. use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

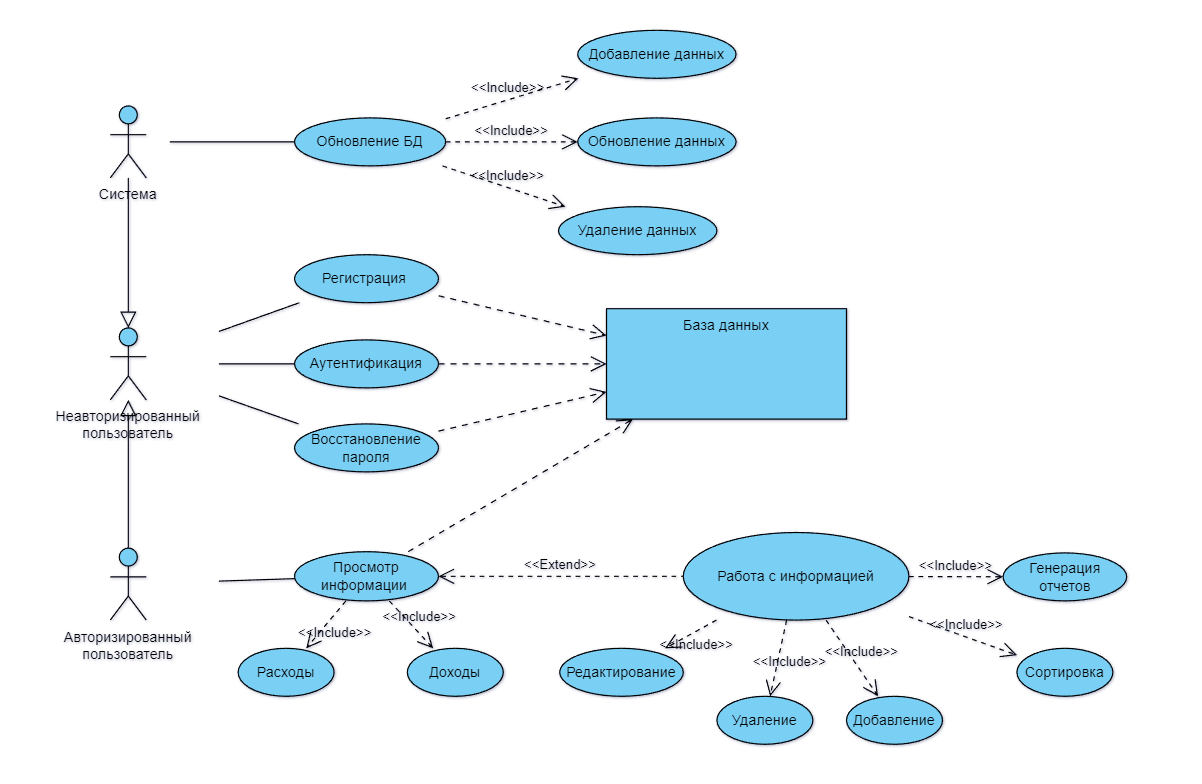


Рис. 1 Диаграмма вариантов использования

1. Диаграмма классов [6]

Диаграмма классов (от англ. "class diagram") предназначена для представления внутренней структуры программы в виде классов и связей между ними.

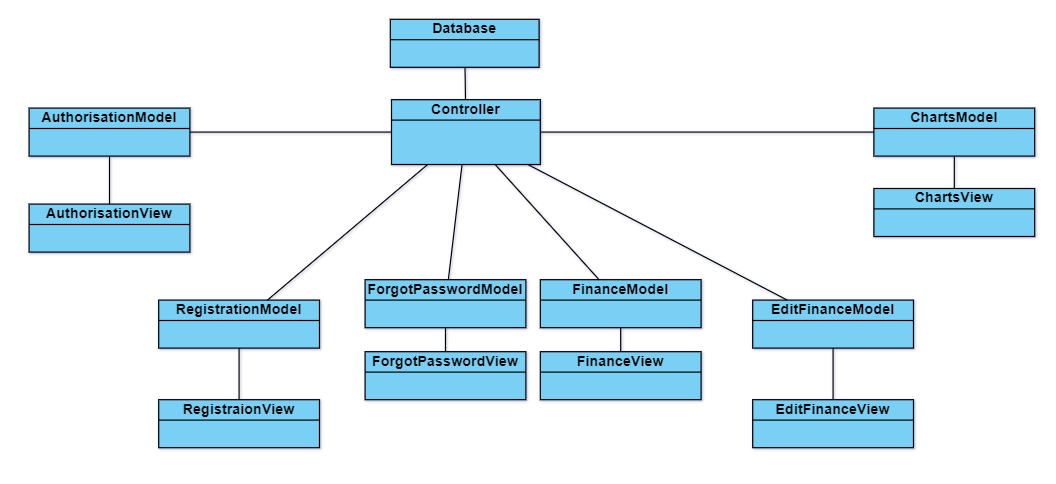


Рис. 2 Диаграмма классов

1. Диаграмма последовательности [6]

Диаграмма последовательности — это тип диаграммы, который позволяет описать взаимодействие между объектами в системе в виде последовательности сообщений, действий и операций, отображая порядок выполнения действий и обмена информацией между объектами во времени.

Диаграммы последовательности отображают взаимодействия между объектами, порядок выполнения операций и обмена информацией во времени. Они помогают улучшить понимание функционирования системы, выявить потенциальные ошибки и проблемы в процессе взаимодействия объектов.

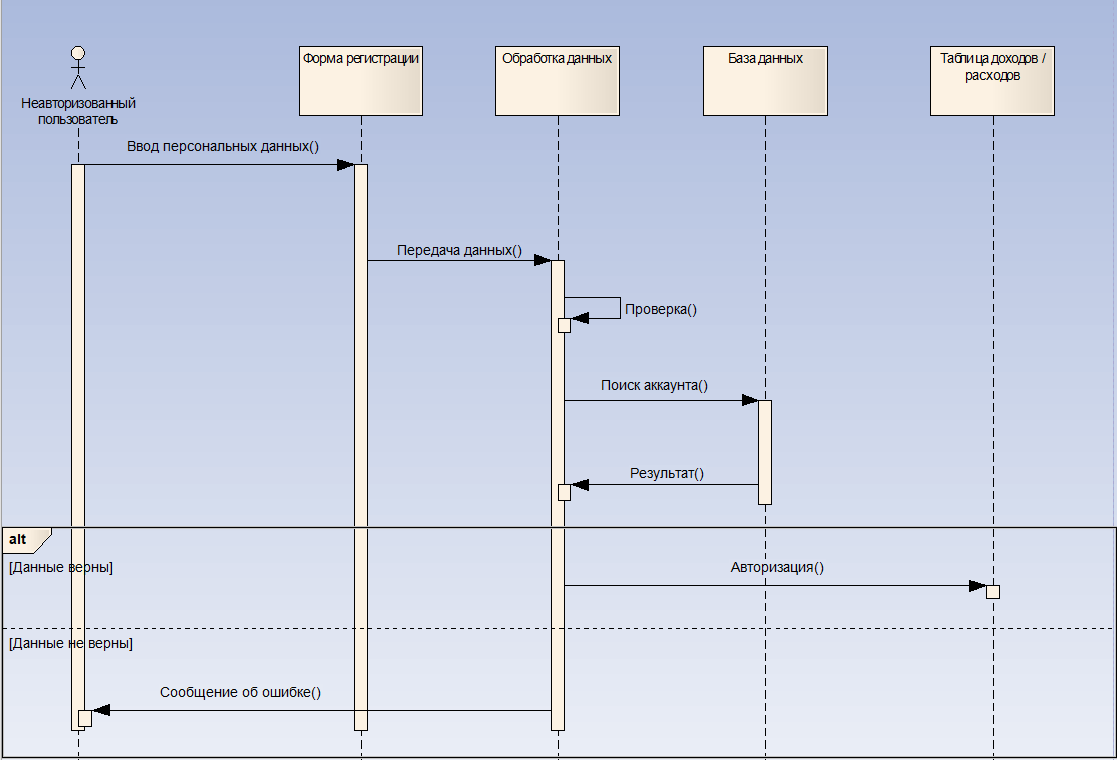


Рис. 3 Диаграмма последовательности «Авторизация»

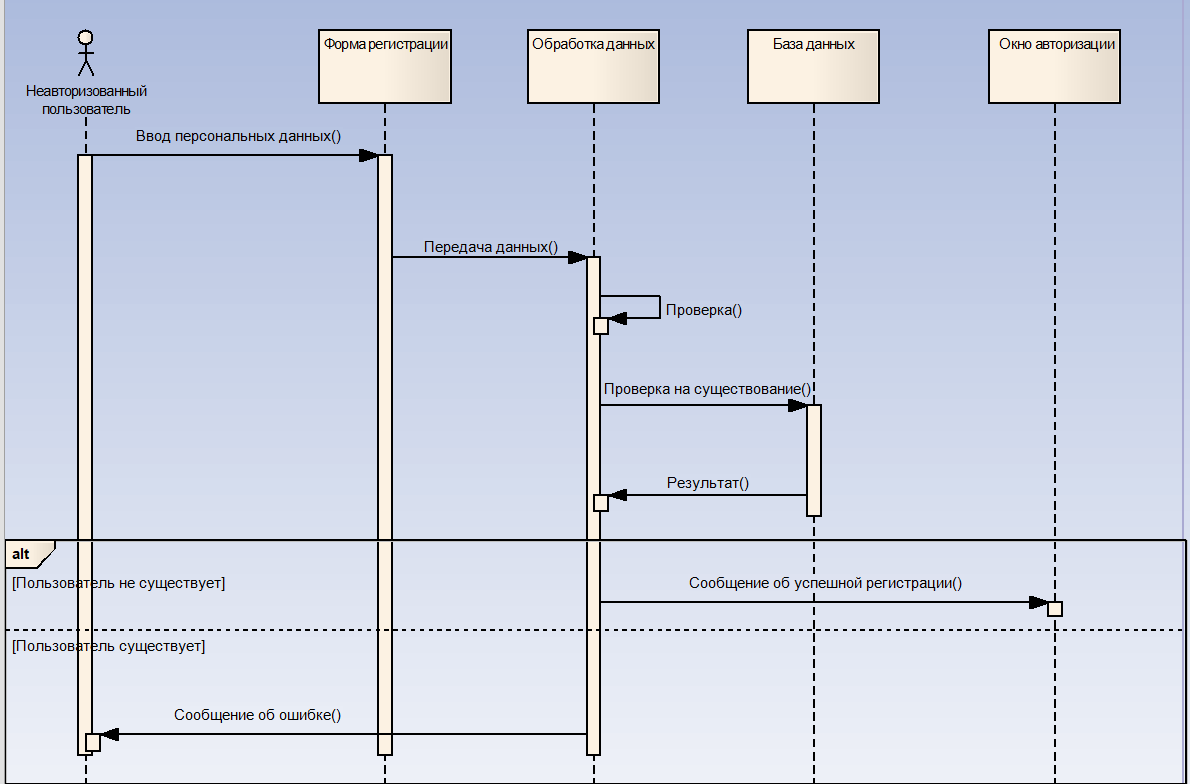


Рис. 4 Диаграмма последовательности «Регистрация»

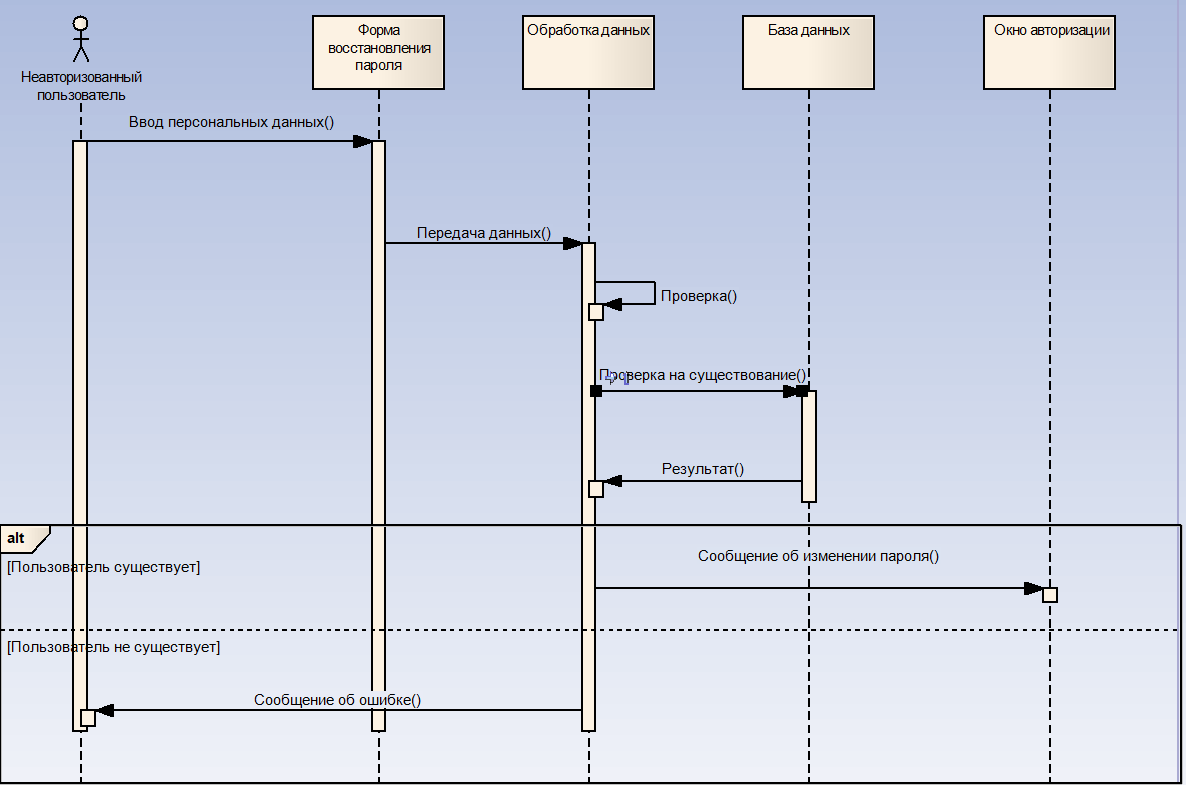
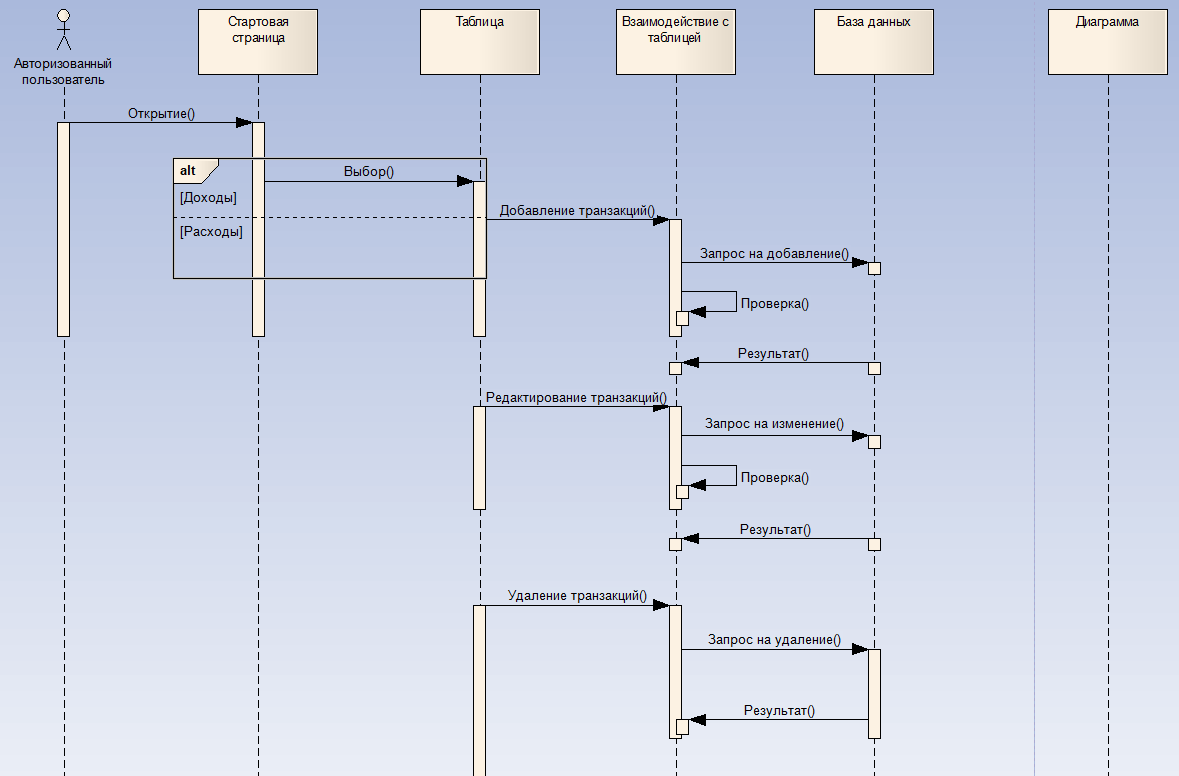


Рис. 5 Диаграмма последовательности «Восстановление пароля»



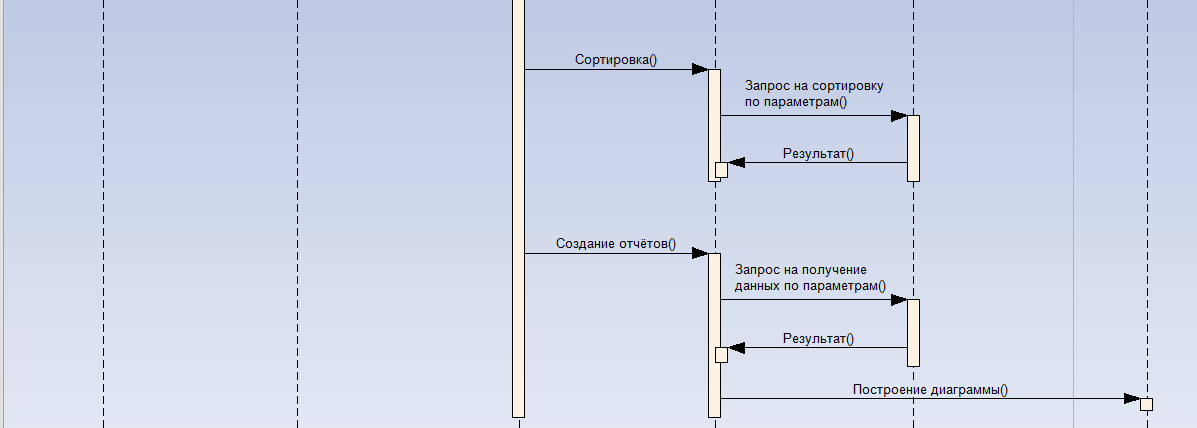


Рис. 6 Диаграмма последовательности «Основной функционал приложения»

В целом моделирование диаграмм позволяет автору яснее представить функциональность приложения, его структуру и взаимодействие между компонентами.

### 2.3.2 Разработка desktop-приложения

После завершения проектирования и определения необходимой функциональности, автор приступил к разработке desktop-приложения. Используя полученные результаты и диаграммы, автор начал создавать программный код и реализовывать требуемые компоненты и модули.

Автор начал с создания необходимых классов и компонентов, соответствующих модели, представлению и контроллеру в рамках принципа MVC. Каждый класс был разработан с учетом его функциональности и взаимодействия с другими компонентами.

Класс Controller наследует от базового класса QtWidgets - QApplication для инициализации и управления основными аспектами приложения, такими как окно, события и обработка пользовательского ввода. QtWidgets QApplication предоставляет функциональность для создания и запуска графического интерфейса, обработки событий, управления виджетами и взаимодействия с пользователем. Это позволяет классу Контроллеру эффективно управлять жизненным циклом приложения и обеспечивать его корректное функционирование.

Модели классов Authorisation, Registration и ForgotPassword наследуют от базового класса QtWidgets - QWidget для создания пользовательского интерфейса. Использование класса QLineEdit из QtWidgets позволяет отображать поле ввода пароля в виде звездочек для обеспечения безопасности.

Модель класса Edit наследует от базового класса QtWidgets QDialog для создания и отображения диалогового окна редактирования данных. QDialog предоставляет функциональность для создания диалоговых окон, в которых пользователь может вводить, редактировать или просматривать информацию.

Класс IncomesChart наследует от базового класса QtWidgets - QWidget для создания графического виджета, который будет отображать диаграмму доходов/расходов. Он также наследует QtCharts для использования функциональности генерации и отображения графиков и диаграмм. Из класса QtGui QPainter используется для рисования дополнительных элементов, таких как заголовки, метки осей и легенды или настройки внешнего вида диаграммы. Класс также использует QtCore Qt для создания цветов и класс QDate для заполнения dateEdit текущей датой, обеспечивая удобную работу с датами в приложении.

Главный класс Finance наследует от базового класса QtWidgets -QMainWindow для создания основного окна приложения и управления его виджетами и функциональностью. Из класса QtWidgets - QTableWidgetItem используется для реализации редактируемой таблицы в главном окне; QMenu используется для создания контекстного меню с дополнительными действиями; QMessageBox используется для вывода сообщений об ошибках или предупреждений. Из класса QtGui - QAction наследуется для добавления действий в контекстное меню. Из класса QtCore - Qt и QDate используются для инициализации кастомного меню и управления временными данными, такими как текущая дата для виджета QDateEdit.

При разработке приложения автор опирался на Qt6 API, доступ к которому предоставляет PySide6. PySide6 является привязкой Python к Qt6, что позволяет использовать все функции и возможности Qt Framework при разработке приложений на языке Python.

Параллельно с разработкой программного кода, автор активно тестировал функциональность и проверял соответствие приложения заявленным требованиям. Это позволило выявлять и устранять возможные ошибки и недочеты еще на ранних стадиях разработки.

В результате проделанной работы и разработки desktop-приложения автор получил функциональный продукт, готовый к последующим этапам тестирования, оптимизации и внедрения. Продукт предоставляет пользователю удобный способ управления финансами, ввода и просмотра финансовых операций, а также анализа данных для более осознанного финансового планирования и принятия решений.

### 2.3.3 Описание функциональности desktop-приложения

На рисунке 7 представлено окно авторизации, которое было разработано автором. Чистые линии, минималистичный стиль и понятные элементы позволяют пользователям быстро ориентироваться и взаимодействовать с приложением. Окно авторизации имеет все необходимые поля для ввода логина и пароля, а также кнопку входа. Этот дизайн подчеркивает простоту и функциональность приложения, делая его доступным и удобным для пользователей всех уровней навыков. Важно отметить, что весь интерфейс приложения разработан с учетом принципов использования современных шрифтов и расположения элементов. Это помогает пользователям быстро ориентироваться и эффективно взаимодействовать с функциональностью приложения.

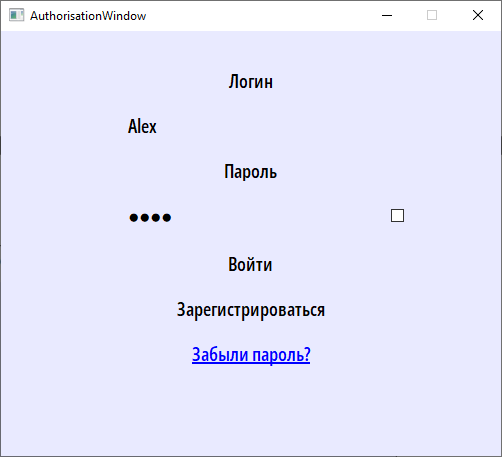


Рис. 7 Окно «Авторизации пользователя»

На рисунке 8 и 9 представлены окна регистрации и восстановления пароля соответственно. Оба окна, так же, как и окно авторизации, спроектированы с учетом требований к современным приложениям, обеспечивая понятный интерфейс.

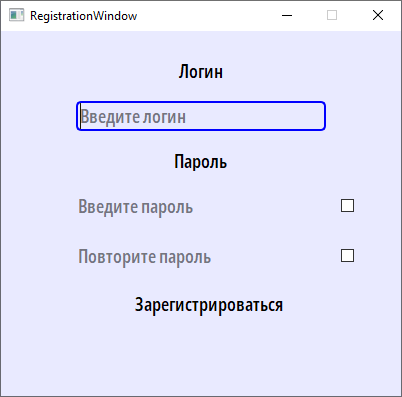


Рис. 8 Окно «Регистрации пользователя»

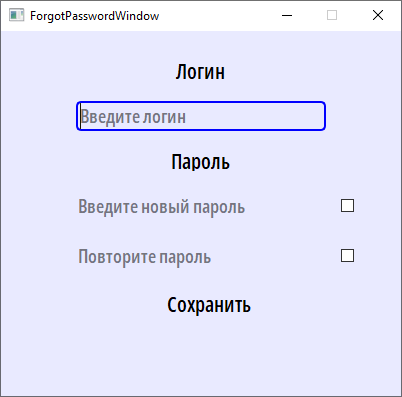


Рис. 9 Окно «Восстановление пароля пользователя»

На рисунке 10 представлено главное окно приложения, где пользователь может наблюдать таблицу с транзакциями. Согласно требованиям, предусмотрена функциональность сортировки этой таблицы, что позволяет упорядочить данные в соответствии с предпочтениями.

Удобной является возможность выбора временного интервала, что позволяет пользователю анализировать финансовые операции за определенный период.

Переключение между таблицей доходов и расходов также реализовано удобным образом, обеспечивая простоту и быстроту доступа к нужным данным.

Важно отметить, что функция генерации отчета по данным таблицы выделена цветом и расположена в удобном месте, что облегчает пользователю получение сводной информации о своих финансах.

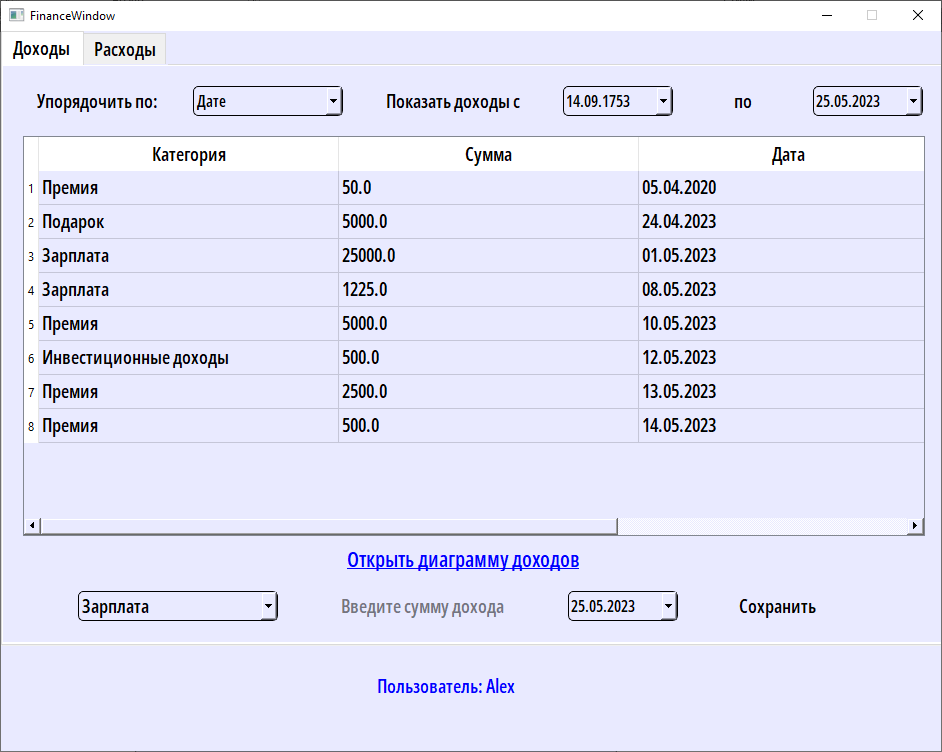


Рис. 10 Главное окно

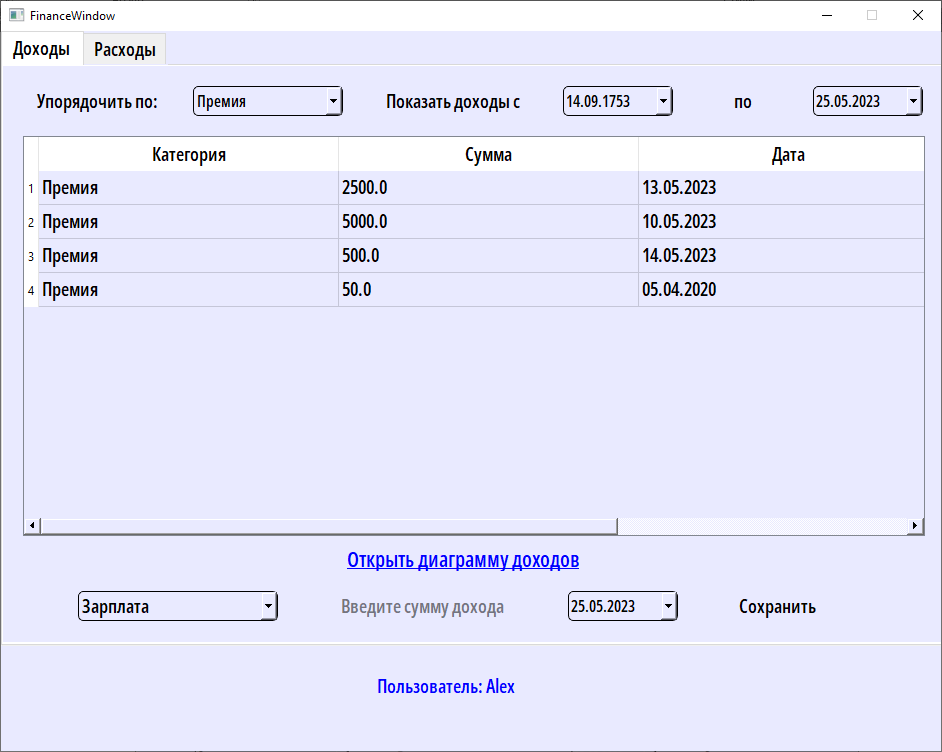


Рис. 11 Пример функционирования приложения

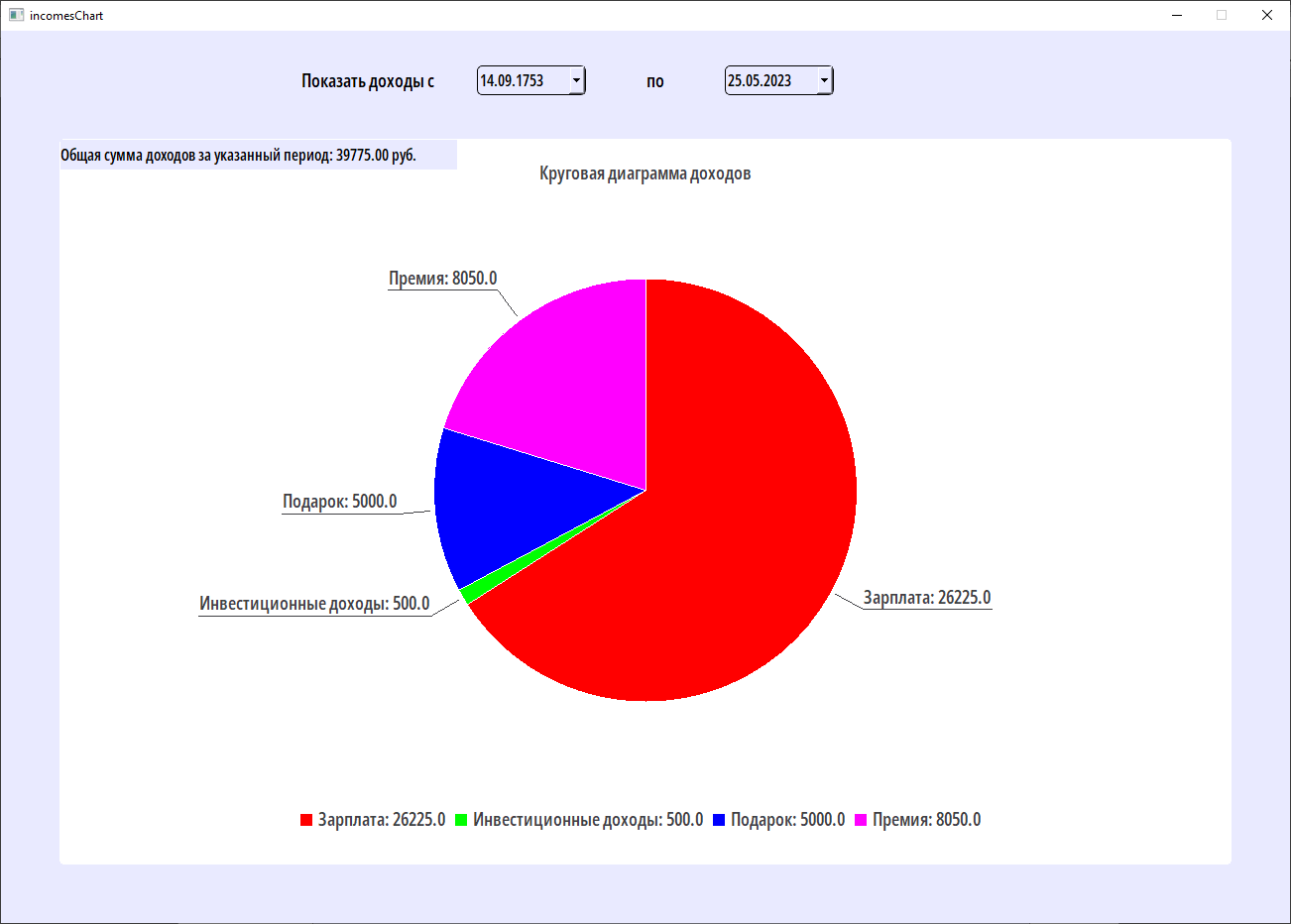


Рис. 12 Пример создания отчёта в виде круговой диаграммы

# 3. Портирование на Android

Портирование приложения на мобильное устройство является важным шагом, позволяющим расширить охват пользователей и обеспечить доступность приложения в любое время и в любом месте. Мобильные устройства стали неотъемлемой частью нашей повседневной жизни, и многие пользователи предпочитают выполнять задачи и получать информацию непосредственно на своих смартфонах или планшетах.

При попытке портирования приложения на мобильную платформу был выбран инструмент pyqtdeploy. Однако, в процессе работы стало очевидным, что этот инструмент имеет некоторые ограничения и неполноту документации, что затруднило успешное портирование приложения. Одной из основных проблем было ограниченное количество реальных "кейсов" и примеров, которые можно было использовать в процессе портирования. Это создало трудности в правильной настройке и адаптации приложения для работы на мобильных устройствах. В результате, несмотря на все усилия, не удалось достичь полноценного портирования приложения с использованием pyqtdeploy.

Учитывая эти обстоятельства, было принято решение обратиться к эмулятору Android и среде разработки PyDroid3. Сначала был загружен и настроен эмулятор Android, включая получение ROOT-прав. Затем было установлено приложение PyDroid3, предоставляющее возможность разработки и запуска приложений на языке Python на устройствах Android.

С помощью PyDroid3 удалось успешно портировать проект с платформы desktop на мобильное устройство Android. В результате была сохранена основная функциональность приложения, однако возникло исключение - Qt Charts. При использовании PyDroid3 нативные библиотеки, необходимые для работы Qt Charts, ограничены, и поэтому графический компонент Qt Charts не будет работать в портированном приложении.

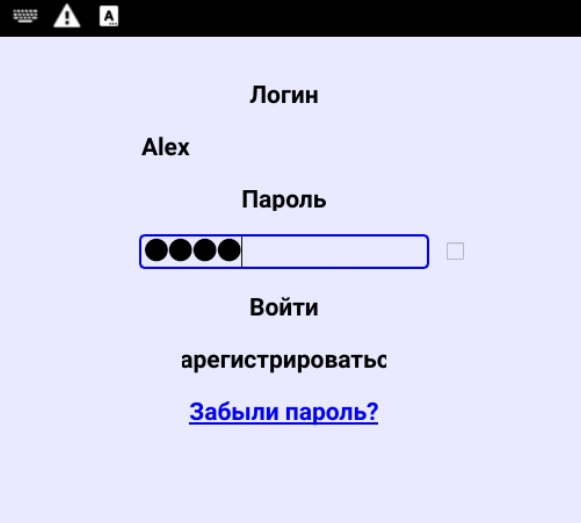
На рисунках 13, 14, 15 представлен пример работы приложения на Android.

Рис. 13 Окно авторизации

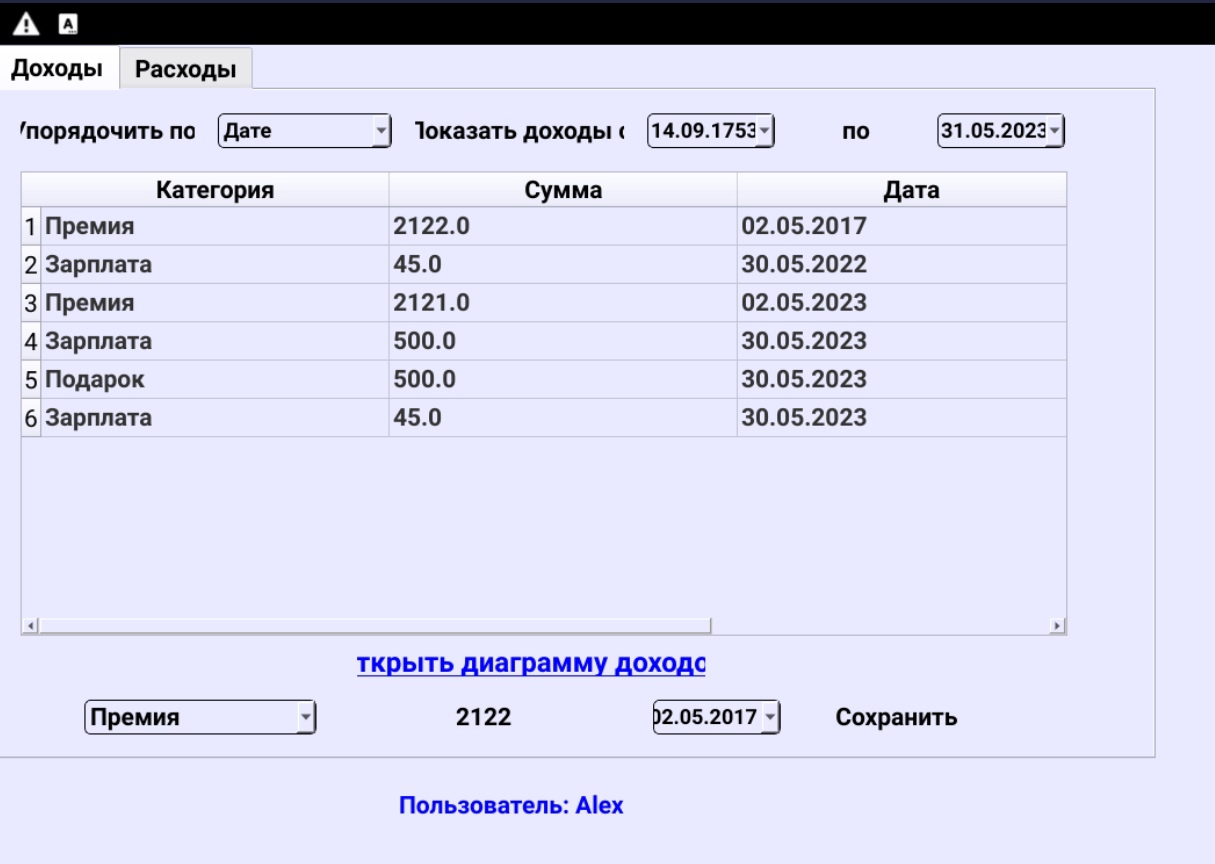


Рис. 14 Главное окно

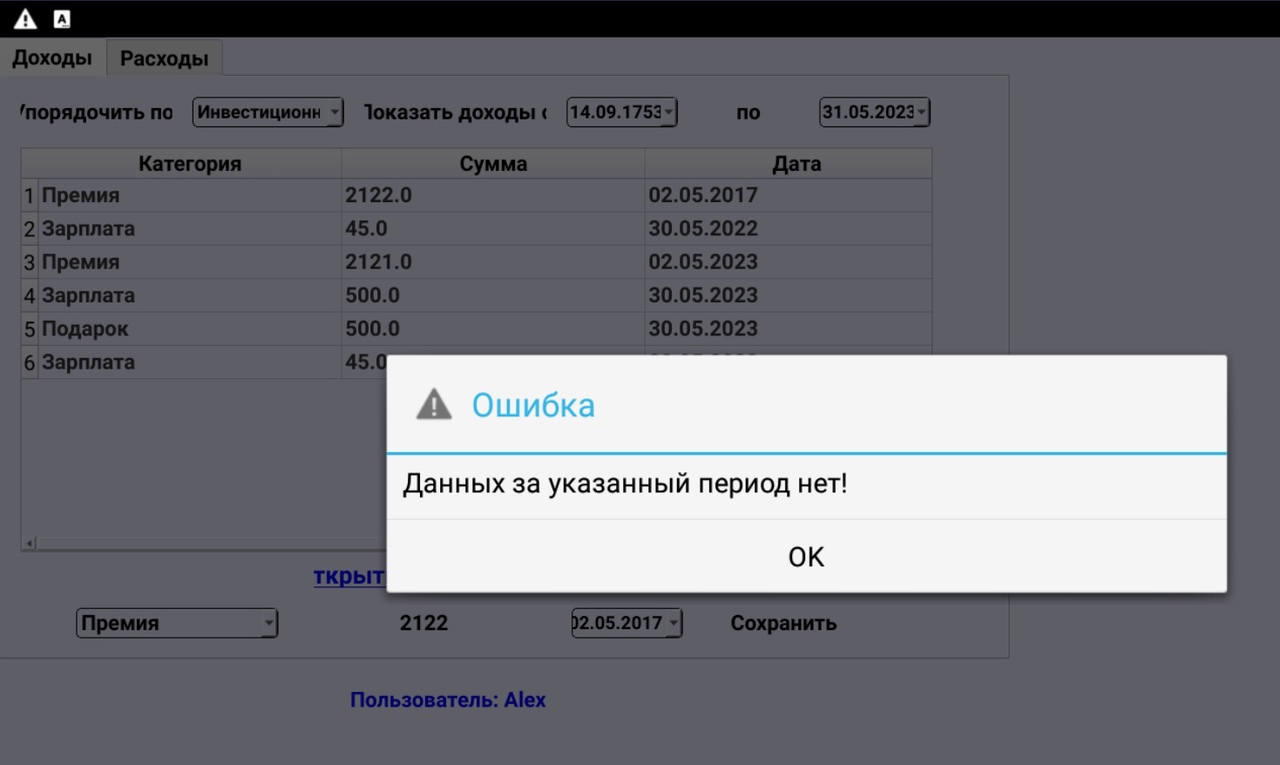


Рис. 15 Пример ошибки

# **Заключение**

В результате данной работы было успешно разработано desktop-приложение, предназначенное для хранения и контроля финансов. В ходе работы были созданы диаграммы вариантов использования, классов и последовательностей, которые служили основой для проектирования и реализации функциональности приложения. Приложение предоставляет пользователю удобный и интуитивно понятный интерфейс, позволяющий вести учет доходов и расходов, а также анализировать финансовые данные. Разработанное приложение представляет ценный инструмент для эффективного управления финансами и принятия осознанных финансовых решений.

Кроме того, была предпринята попытка портирования приложения на мобильное устройство Android. Приложение удалось успешно запустить на Android-устройстве, однако из-за недостаточности реальных примеров и неполноты документации инструмента pyqtdeploy не удалось создать исполняемый файл .apk для установки приложения на другие устройства. В дальнейшем, для портирования приложения на Android, был использован эмулятор Android и среда разработки PyDroid3.

# **Литература**

1. Python. URL [электронный ресурс]: <https://www.python.org/?hl=RU> (29.05.2023)
2. PySide6. URL [электронный ресурс]: <https://doc.qt.io/qtforpython-6/quickstart.html> (29.05.2023)
3. Sqlite3. URL [электронный ресурс]: <https://docs.python.org/3/library/sqlite3.html> (29.05.2023)
4. Проектирование GUI. Джеф Раскин // Интерфейс: новые направления в проектировании компьютерных систем – 2007
5. Проектирование GUI. Алан Купер // Об интерфейсе. Основы проектирования взаимодействия – 2021
6. Проектирование диаграмм. Мартин Фаулер // UML Основы – 2005 (29.05.2023)
7. Qt Core. URL [электронный ресурс]: <https://doc.qt.io/qt-6/qtcore-index.html> (29.05.2023)
8. Qt Widgets. URL [электронный ресурс]: <https://doc.qt.io/qt-6/qtwidgets-index.html> (29.05.2023)
9. Qt GUI. URL [электронный ресурс]: <https://doc.qt.io/qt-6/qtgui-index.html> (29.05.2023)
10. Qt Designer. URL [электронный ресурс]: <https://doc.qt.io/qt-6/qtdesigner-manual.html> (29.05.2023)
11. Qt Charts. URL [электронный ресурс]: <https://doc.qt.io/qt-6/qtcharts-overview.html> (29.05.2023)
12. Pyqtdeploy. URL [электронный ресурс]: <https://pypi.org/project/pyqtdeploy/> (1.06.2023)
13. Pydroid 3. URL [электронный ресурс]: <https://github.com/pyamsoft/pydroid> (1.06.2023)