

**Приложение для
учёта финансов
“UWasting”**



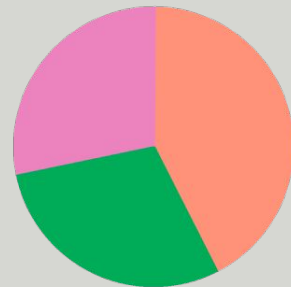
Состав команды и роли

- Менеджер проекта, технический писатель — Берчик А.С.
- Аналитик, тестировщик — Андрусенко Л.Д.
- Аналитик, дизайнер — Малкин Г.Д.
- Разработчики — Гришин А.В., Гоппен С.Д.



Описание проекта




- Проект UWasting представляет собой мобильное приложение для систематизации и контроля личных и семейных финансов. Оно позволяет пользователям эффективно учитывать доходы, расходы, кредиты и ссуды, обеспечивая прозрачность и наглядность финансовых потоков. Благодаря удобному интерфейсу и интегрированным инструментам аналитики приложение помогает принимать обоснованные решения и планировать бюджет на будущее.



15 340 \$

Расходы как обычно

Категории

	Продукты 3 операции	8 000 \$
	Ювелирия 1 операция	1 340 \$
	Прочее 6 операций	300 \$

Еще 4 категории на 5700 \$



Цели и требования

Цели:

Создать удобное приложение для управления личными финансами.

Повысить финансовую грамотность пользователей.

Обеспечить безопасность и конфиденциальность данных.

Предоставить персонализацию через категории.

Функциональные требования:

Регистрация и авторизация пользователей.

Управление доходами и расходами с категоризацией.

Управление кредитами и ссудами с учётом их влияния на бюджет.

Генерация отчётов и визуализация данных в виде графиков и диаграмм.

Прогнозирование будущих доходов и расходов на основе исторических данных.

Экспорт данных в CSV формате.

Нефункциональные требования:

Поддержка Android 12 и выше.

Быстрая и стабильная работа (время отклика ≤ 2 сек).

Высокая производительность и устойчивость к нагрузкам при росте базы данных и числа пользователей.

Безопасность данных (JWT, HTTPS).

Удобный интерфейс по стандартам Material Design.

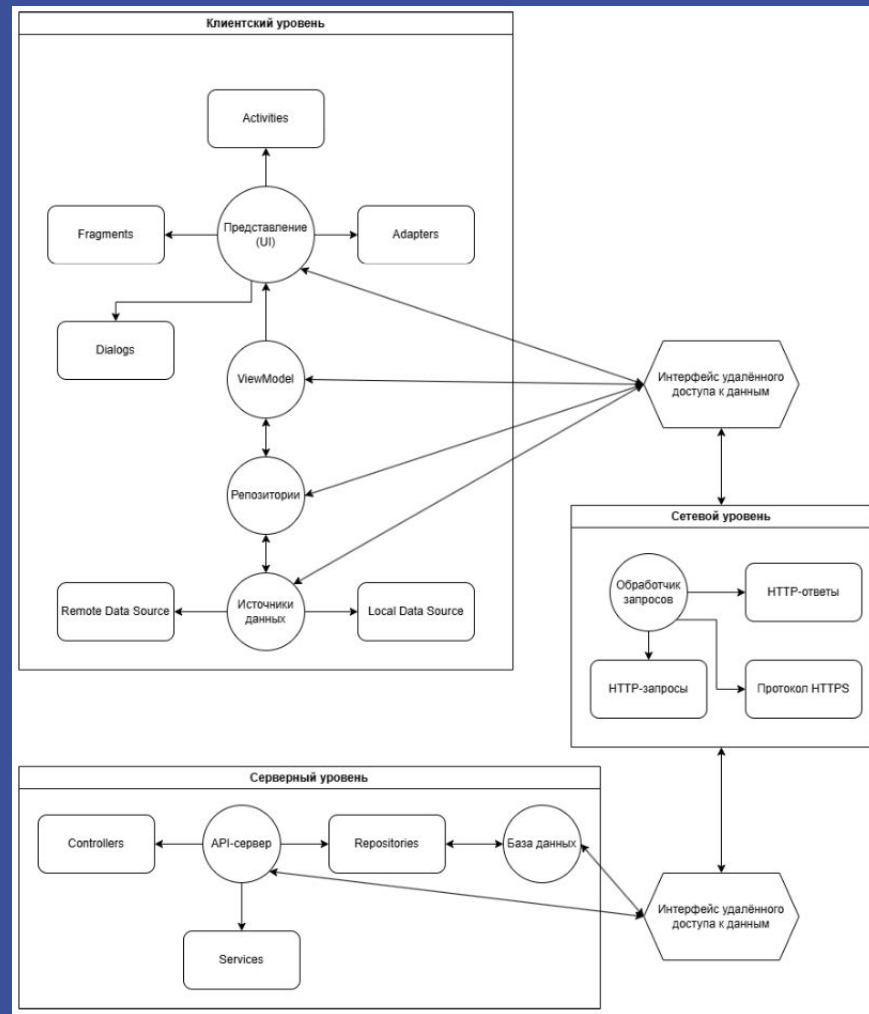
Архитектура программного продукта

Проект построен на многоуровневой архитектуре, что обеспечивает разделение ответственности и удобство масштабирования:

UI (Пользовательский интерфейс) — отвечает за взаимодействие с пользователем, отображение данных и получение ввода.

Бизнес-логика — слой, реализующий правила обработки данных, вычисления и прогнозы.

Доступ к данным — слой, обеспечивающий работу с сервером и базой данных через REST API.



Наиболее значимые риски проекта

Недостаточная вовлечённость команды: снижает продуктивность и качество работы.

Недостаточное тестирование: увеличивает вероятность ошибок в финальной версии.

Проблемы безопасности данных: угрожают конфиденциальности и доверии пользователей.

Риски задержек и снижения качества: влияют на успешность сдачи и эксплуатацию проекта.



Описание стека технологий

Клиентская часть:

Kotlin — основной язык для разработки Android-приложений.

AndroidX — библиотека, обеспечивающая совместимость с современными компонентами Android.

Material Design — стандарт Google для разработки интуитивно понятных и красивых интерфейсов.

Retrofit — библиотека для удобной работы с REST API, поддерживающая типобезопасные запросы.

RxJava — для реактивного программирования и асинхронной обработки данных.

MPAndroidChart — библиотека для создания интерактивных графиков и диаграмм.

Серверная часть:

C# — язык программирования для серверной логики.

ASP.NET Core — фреймворк для создания высокопроизводительных и кроссплатформенных приложений и API.

Entity Framework Core — ORM для работы с базой данных через объектно-ориентированные модели.

База данных:

PostgreSQL — реляционная база данных с высокой надёжностью и масштабируемостью.

– Диаграммы процессов проекта

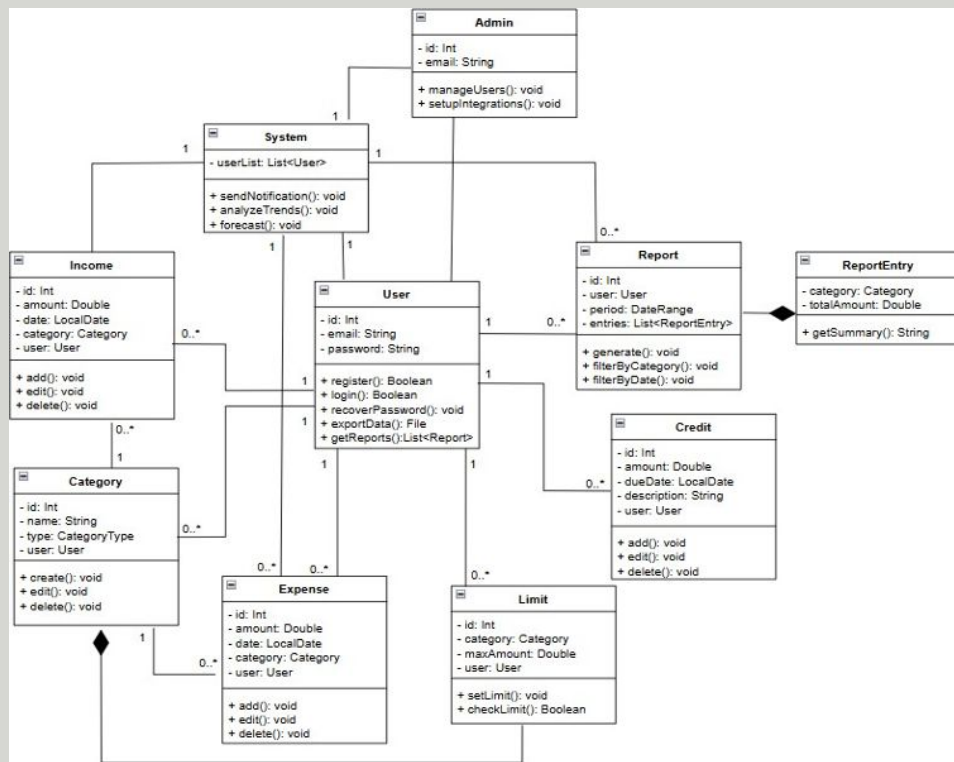


Диаграмма классов

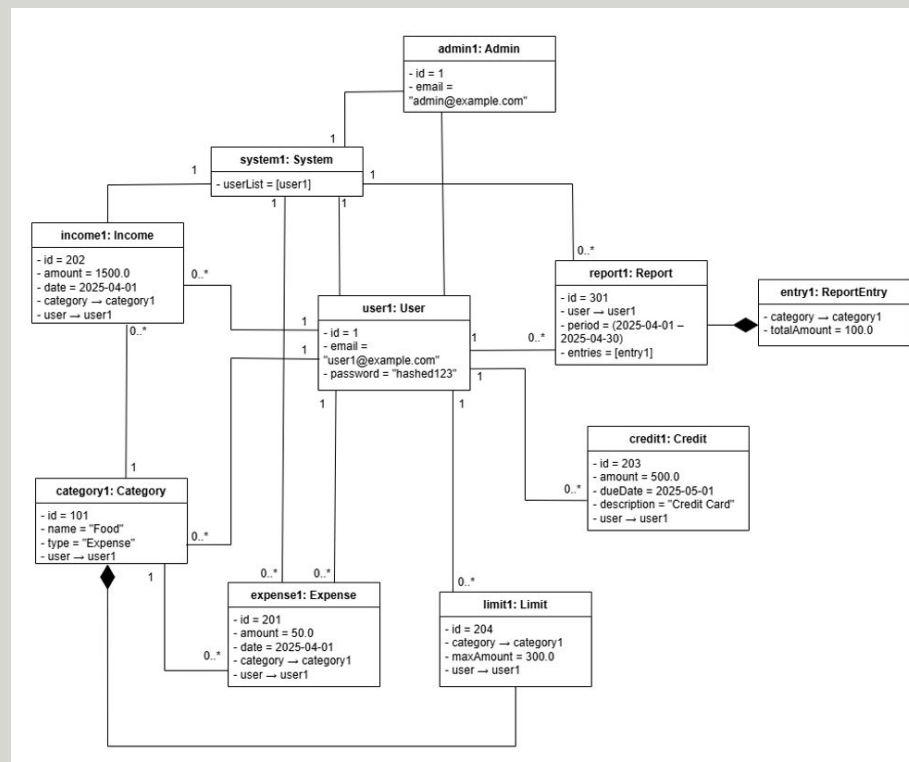
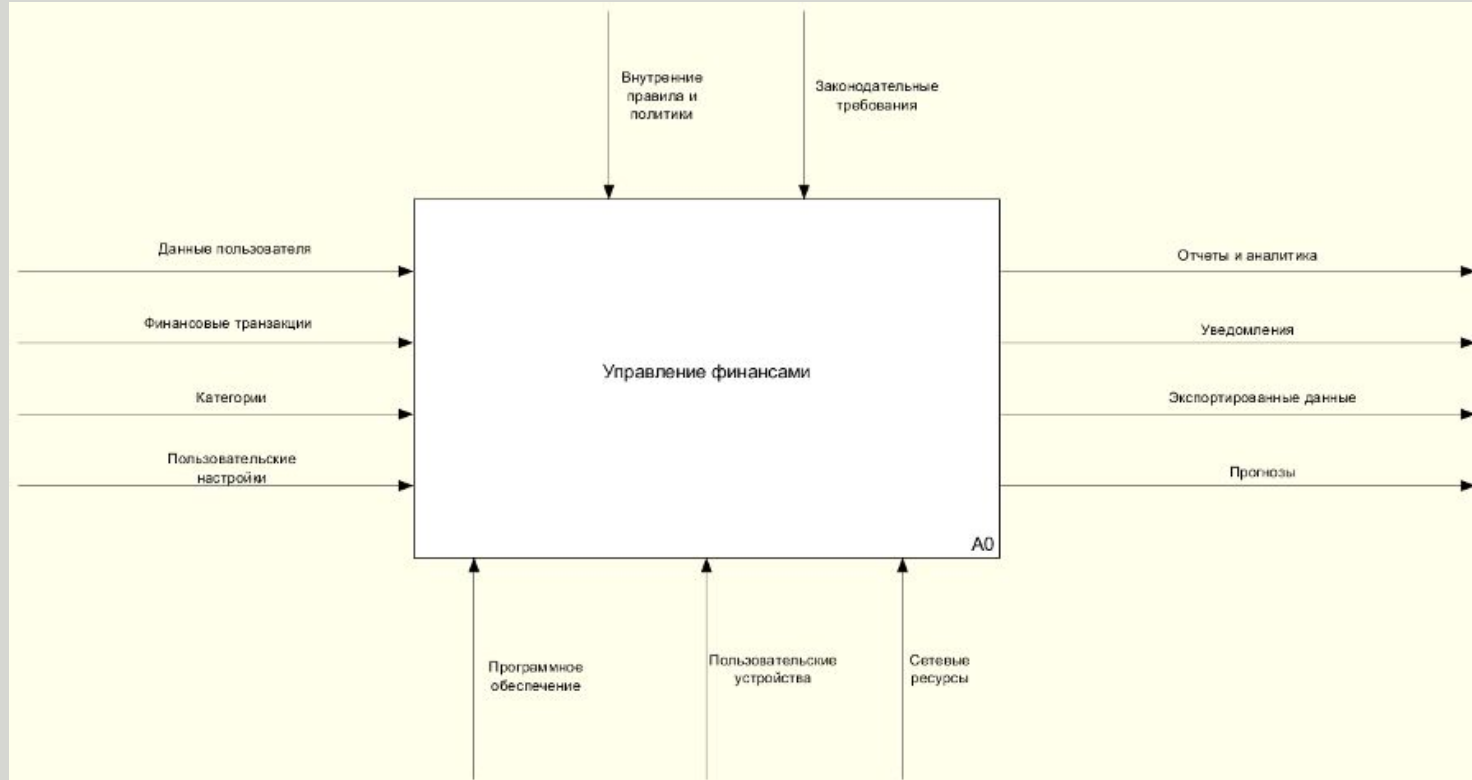


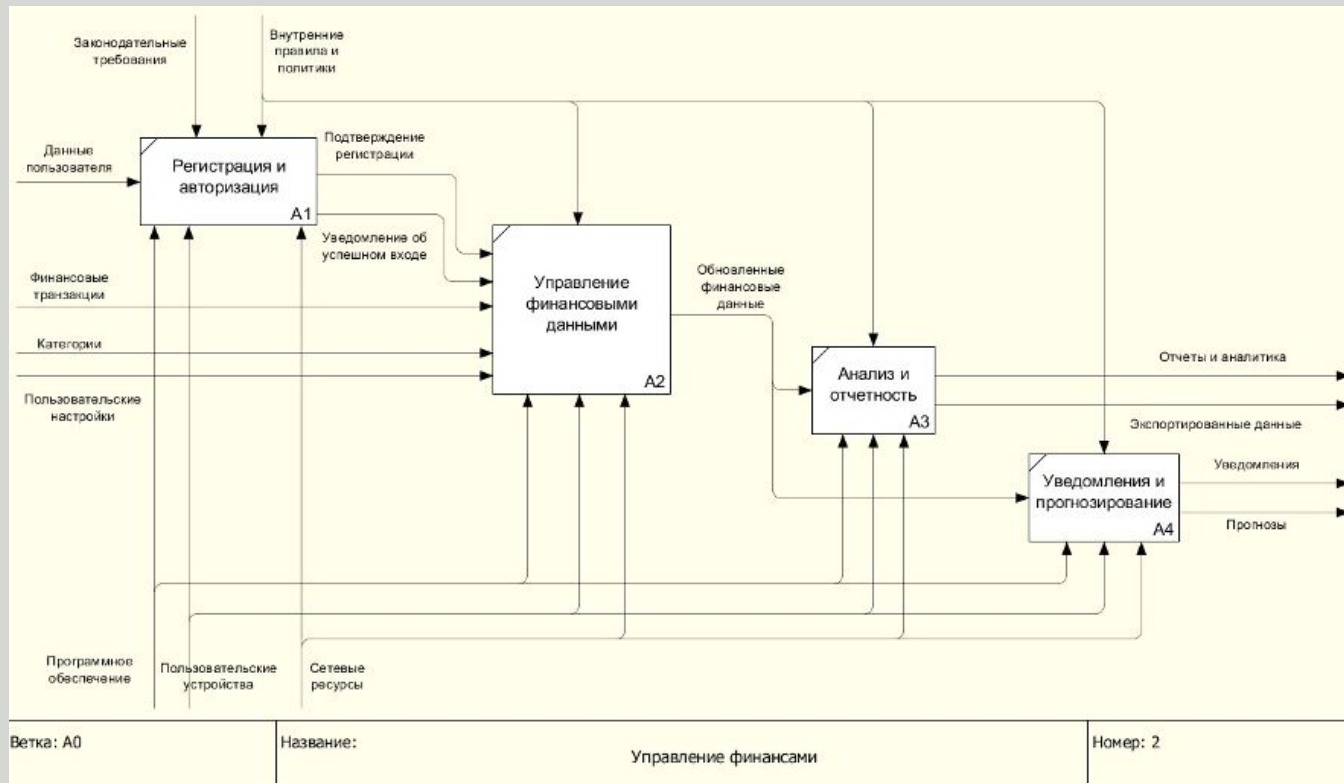
Диаграмма объектов

– Диаграммы процессов проекта



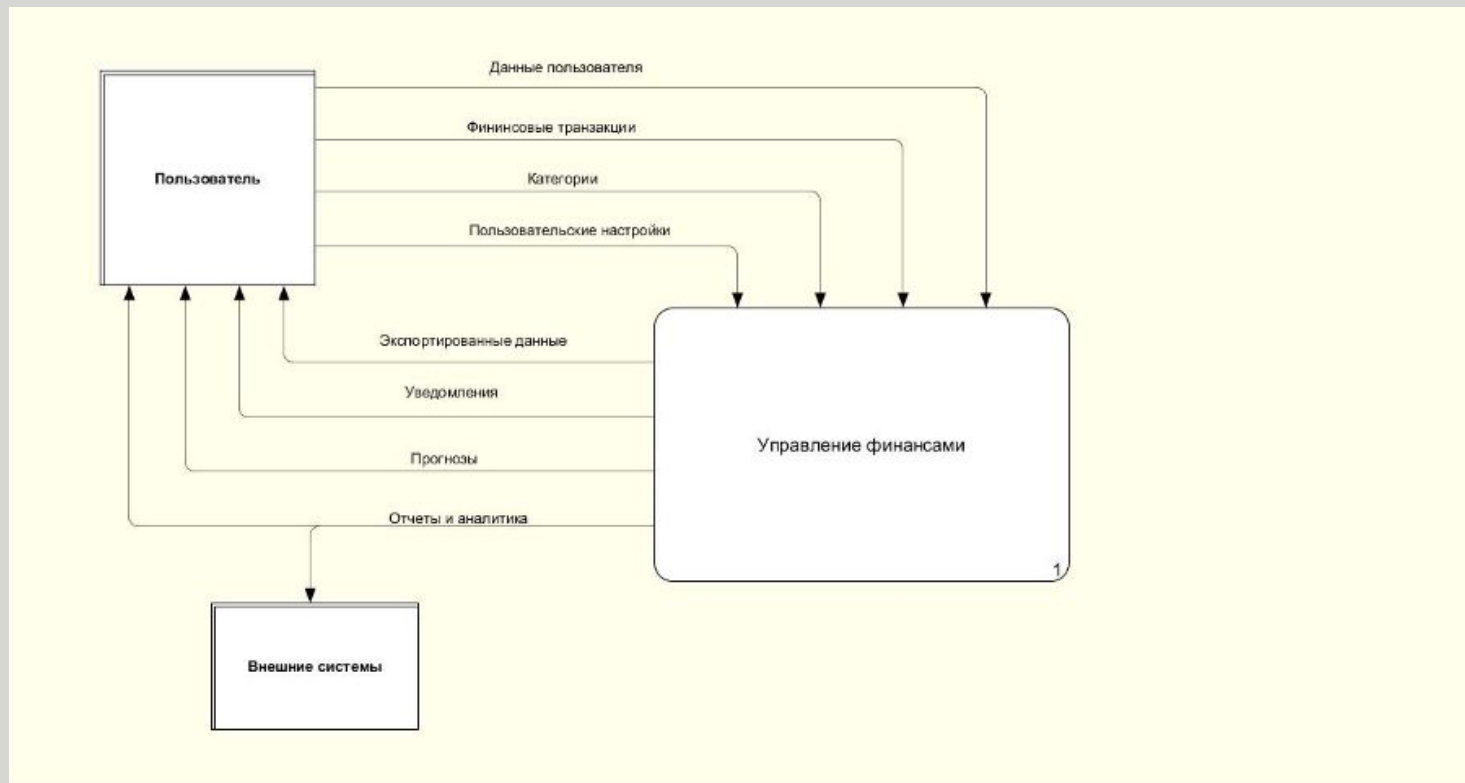
IDEFo - Контекстная диаграмма

– Диаграммы процессов проекта



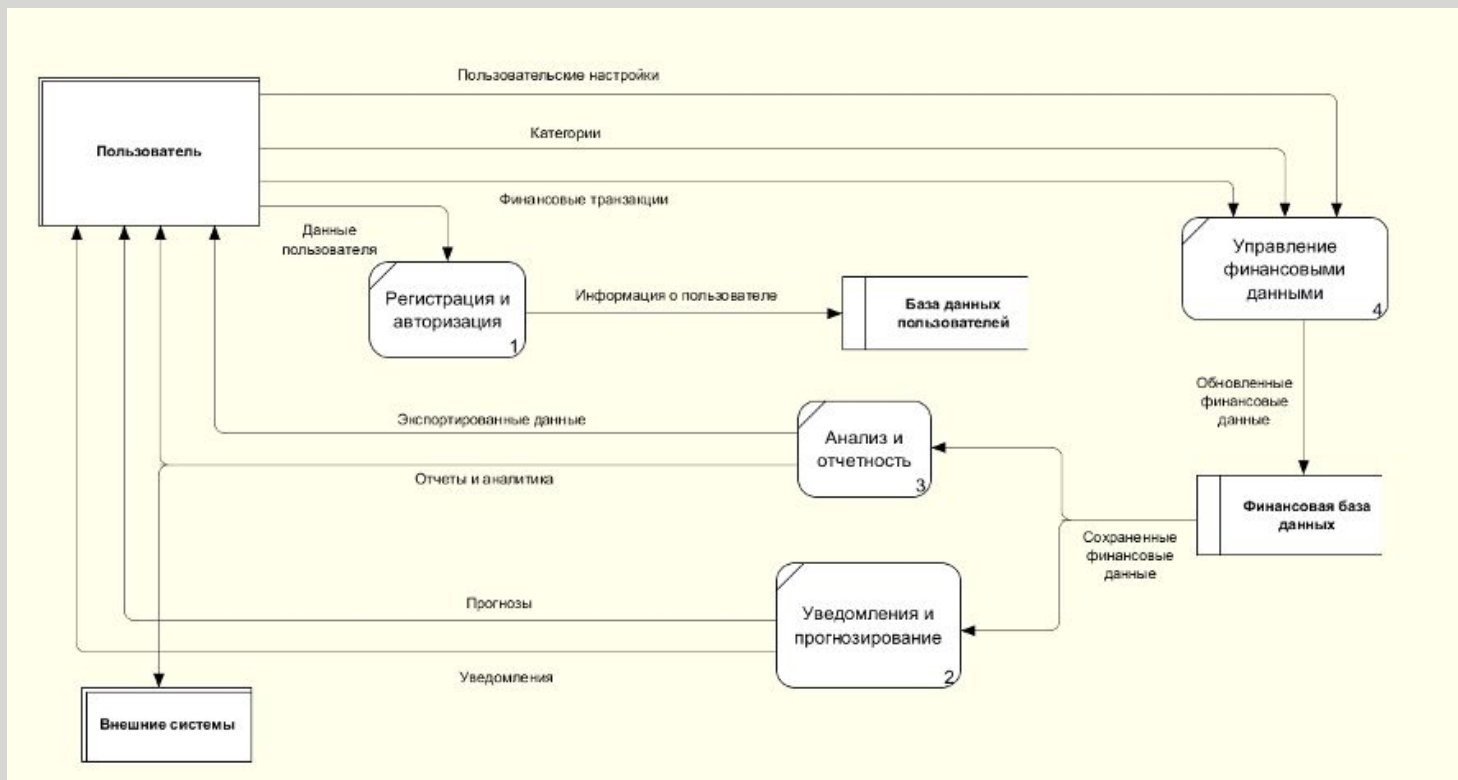
IDEFo - Декомпозиция контекстной диаграммы

– Диаграммы процессов проекта



DFD - Контекстная диаграмма

– Диаграммы процессов проекта



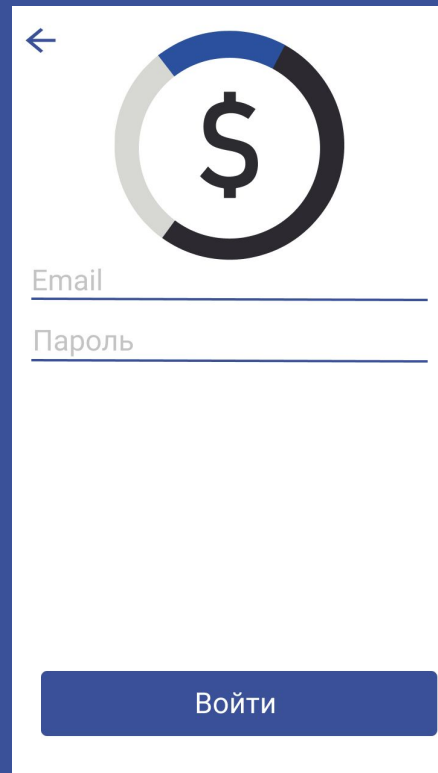
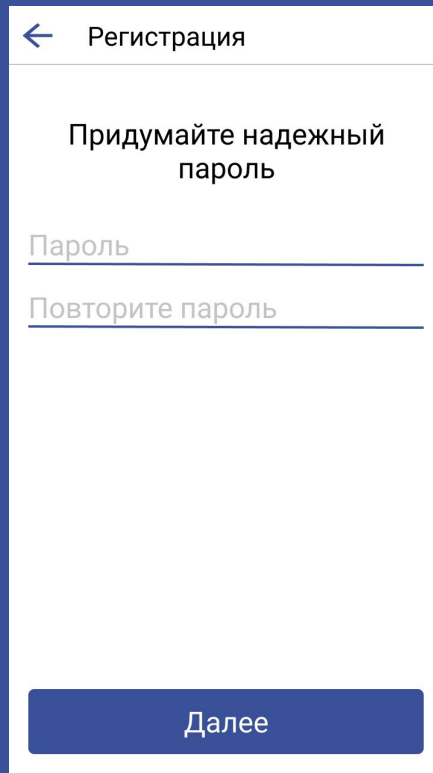
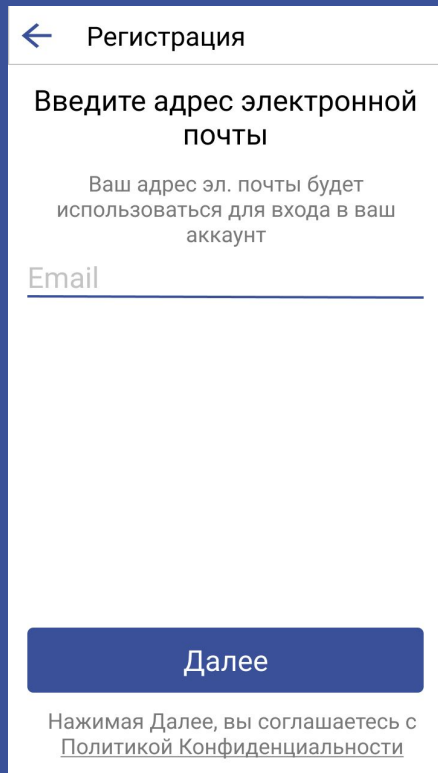
DFD - Декомпозиция контекстной диаграммы

Описание функционала приложения

- **Регистрация и авторизация:** Пользователь создаёт аккаунт с email и паролем.
- **Управление доходами и расходами:** Пользователь добавляет, редактирует и удаляет записи о доходах и расходах с привязкой к категориям.
- **Категории:** Пользователь создаёт и управляет категориями.
- **Управление кредитами и ссудами:** В приложении ведётся учёт кредитов с информацией о сроках и платежах, которые учитываются в общем бюджете.
- **Отчёты и аналитика:** Система автоматически формирует подробные отчёты с графиками и фильтрами по категориям, датам и типам операций.
- **Прогнозирование финансов:** Приложение прогнозирует будущие доходы и расходы на основе истории и помогает планировать бюджет.
- **Экспорт данных:** Пользователь может экспортировать данные в CSV для резервного копирования и анализа.

Макеты приложения

Регистрация и авторизация



Макеты приложения

Управление доходами и расходами

←

Новый доход

Название дохода

Сумма

Дата совершения
3 февраля

Категория

Внести доход

←

Выберите категорию

Категории

\$

Зарплата

\$

Пенсия

\$

Стипендия

\$

Прочие доходы

\$

Президентские отчисления

Выбрать категорию

←

Новый расход

Название расхода

Сумма

Дата совершения
3 февраля

Категория

Внести расход

←

Выберите категорию

Категории

🛒

Продукты

💎

Ювелирия

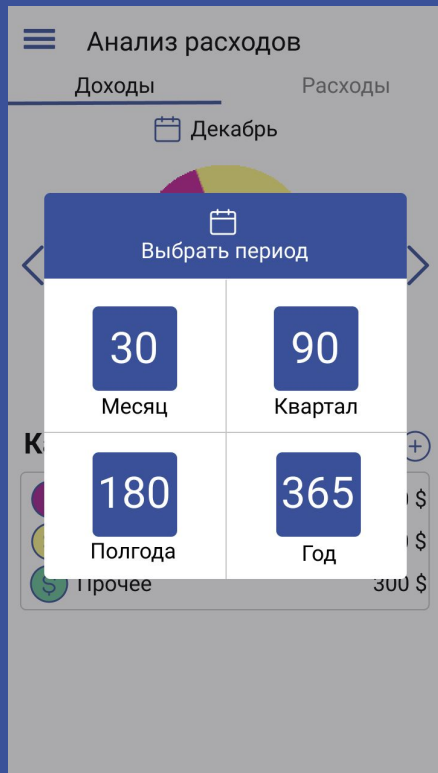
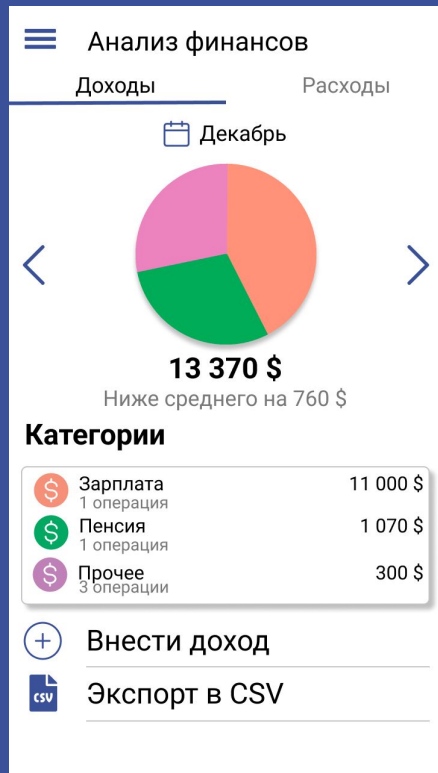
📁

Прочее

Выбрать категорию

Макеты приложения

Отчёты, аналитика и прогноз финансов



Макеты приложения

Кредиты и ссуды

Добавить кредит/ссуду

Сумма

Дата

Ссуда

☐

Кредит

Добавить

Тестирование

JUnit — модульное тестирование бизнес-логики,

Espresso — UI-тесты для Android,

Postman — ручное тестирование REST API,

Selenium — автоматизация тестов интерфейса,

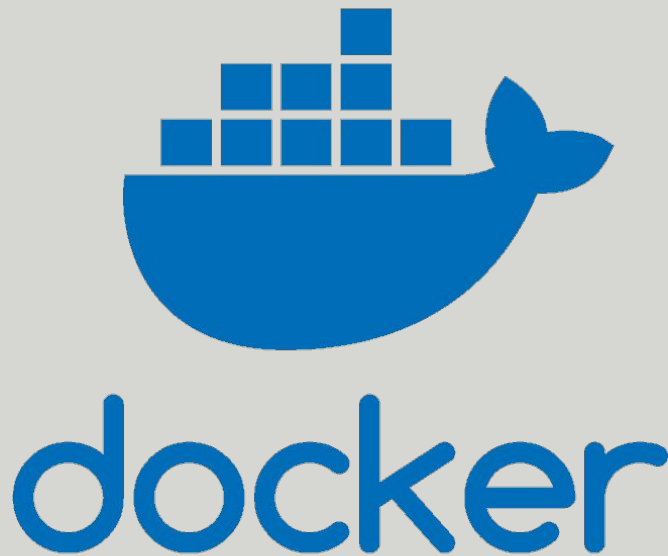
JaCoCo — анализ покрытия кода тестами.

Тестовые сценарии охватывали ключевые функции приложения, включая регистрацию, управление финансами и отчёты.

Особое внимание уделялось проверке лимитов расходов и работы с кредитами, что важно для корректности финансового контроля.

Развёртывание

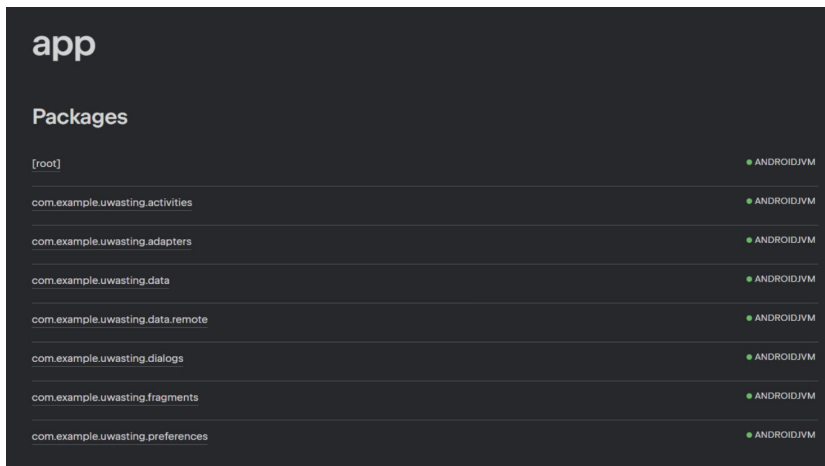
- Серверная часть приложения и база данных размещаются в Docker-контейнерах, что обеспечивает изоляцию среды выполнения и упрощает управление зависимостями.
- Для удобства и автоматизации запуска используется Docker Compose — инструмент, позволяющий одновременно поднимать и настраивать несколько связанных контейнеров (сервер, база данных) в едином окружении.



Документация разработчика

Для генерации документации использовался Dokka 1.9.10, что позволяет автоматически генерировать документацию из исходного кода проекта.

Документация включает описание архитектуры, API, используемых библиотек и фреймворков, а также инструкции по настройке и развертыванию приложения.



Документация пользователя

Документация для пользователя создана вручную и размещена на GitHub Wiki.

Включает подробные инструкции по установке приложения, регистрации и использованию основных функций (учёт расходов, создание отчетов и т.д.)

Основные функции

Andrew edited this page 3 minutes ago · [1 revision](#)

1. Синхронизация с сервером

- Все расходы сохраняются на сервере, указанном в настройках.
- Используется API (описано в UWasting-Server).
- Данные синхронизируются при запуске приложения или вручную через кнопку "обновить".

2. Учёт трат

- Создание новых записей с категорией, датой, описанием и суммой.
- Поддержка категорий: еда, транспорт, развлечения, коммунальные, прочее.
- Локальное и серверное хранение данных.

3. Статистика и анализ

- Визуальные отчёты: столбчатые и круговые диаграммы.
- Распределение по категориям.
- Выбор периода анализа: текущий день, неделя, месяц.

4. Настройки

- Возможность вручную задать адрес API сервера.
- Тест соединения с сервером.
- Настройка автоматической синхронизации и уведомлений.
- Очистка локальных данных.

Достоинства

Современный стек технологий: Использование Kotlin, .NET Core и PostgreSQL обеспечивает высокую производительность и стабильность.

Модульная архитектура: Упрощает поддержку и масштабирование системы, улучшает тестируемость и гибкость.

Интуитивно понятный интерфейс: Использование Material Design и MVVM обеспечивает удобство и простоту для конечных пользователей.

Безопасность данных: Защита пользовательских данных через современные методы аутентификации (JWT) и безопасную передачу данных через HTTPS.

Гибкость и масштабируемость: Приложение легко адаптируется под различные требования, благодаря контейнеризации с Docker.

Недостатки

Поддержка только Android: Приложение доступно только для Android, что ограничивает пользователей iOS.

Нет интеграции с банковскими API: Отсутствует автоматический импорт транзакций с банковских аккаунтов.

Отсутствие совместного бюджета: Невозможно делиться бюджетом с другими пользователями или членами семьи.

Ограниченные уведомления: Уведомления ограничены и не позволяют настраивать различные типы уведомлений по расходам.

Нет поддержки инвестиционных платформ: Приложение не интегрируется с популярными инвестиционными сервисами.

