# Use Cases — Financery

Диаграмма прецедентов и детальные спецификации use-case для приложения Financery.

## Use Case Diagram (PlantUML)

Ниже находится визуальное приближение диаграммы прецедентов. Полный PlantUML-исходник сохранён в файле Financery\_UseCases.puml.



### PlantUML source

**Файл: Financery\_UseCases.puml**

@startuml Financery\_UseCases  
left to right direction  
skinparam packageStyle rectangle  
  
actor User  
actor "OCR Service" as OCR  
actor "ML Service" as ML  
actor "Cloud Backend" as Backend  
actor Admin  
  
package "Financery (Mobile)" {  
 usecase "UC-01\nРегистрация/Вход" as UC01  
 usecase "UC-02\nГлавная сводка" as UC02  
 usecase "UC-03\nДобавление транзакции\n(ручное)" as UC03  
 usecase "UC-04\nСканирование чека\n(OCR -> Receipt -> Transaction)" as UC04  
 usecase "UC-05\nРедактирование/Удаление\nтранзакции" as UC05  
 usecase "UC-06\nУправление категориями" as UC06  
 usecase "UC-07\nПоиск/Фильтрация" as UC07  
 usecase "UC-08\nОтчёты и графики" as UC08  
 usecase "UC-09\nЦели и рекомендации" as UC09  
 usecase "UC-10\nЭкспорт/Импорт JSON" as UC10  
 usecase "UC-11\nСинхронизация/Бэкап" as UC11  
 usecase "UC-12\nНастройки приватности\n(аналитика)" as UC12  
 usecase "UC-13\nFeedback для ML" as UC13  
 usecase "UC-14\nПросмотр рекомендаций\nи прогнозов" as UC14  
 usecase "UC-15\nУдаление аккаунта" as UC15  
 usecase "UC-16\nАдмин: лог/модели" as UC16  
}  
  
User --> UC01  
User --> UC02  
User --> UC03  
User --> UC04  
User --> UC05  
User --> UC06  
User --> UC07  
User --> UC08  
User --> UC09  
User --> UC10  
User --> UC11  
User --> UC12  
User --> UC13  
User --> UC14  
User --> UC15  
  
UC04 .> OCR : <<invoke>>  
UC04 .> Backend : <<store receipt>>  
UC13 .> ML : <<feedback>>  
UC03 .> ML : <<optional classify>>  
UC09 .> ML : <<recommendations>>  
UC08 .> Backend : <<query-data>>  
Admin --> UC16  
Backend --> ML : <<invoke model>>  
@enduml

## Actors (Акторы)

**User:** Конечный пользователь приложения — добавляет транзакции, сканирует чеки, управляет целями.

**OCR Service:** Сервис для распознавания текста в изображениях чека (on-device или cloud).

**ML Service:** Сервис, предоставляющий автокатегоризацию, нормализацию наименований и рекомендации.

**Cloud Backend:** Хранилище и API для синхронизации данных (Firebase/REST API).

**Admin:** Админ/разработчик системы с доступом к логам и управлению моделями.

## Use Cases (Список)

**UC-01 — Регистрация / Вход**

**UC-02 — Просмотр главной сводки**

**UC-03 — Добавление транзакции (ручное)**

**UC-04 — Сканирование чека и создание транзакций (OCR -> Receipt -> Transaction)**

**UC-05 — Редактирование / Удаление транзакции**

**UC-06 — Управление категориями и тегами**

**UC-07 — Поиск и фильтрация транзакций**

**UC-08 — Просмотр отчётов / графиков**

**UC-09 — Создание и отслеживание цели (Goal)**

**UC-10 — Экспорт / импорт данных (JSON)**

**UC-11 — Резервное копирование / синхронизация (sync)**

**UC-12 — Отключение/включение анонимной аналитики (consent)**

**UC-13 — Отправка обратной связи для ML (коррекция категории)**

**UC-14 — Просмотр рекомендаций и прогнозов (ML)**

**UC-15 — Удаление аккаунта**

**UC-16 — Административные операции (логирование, мониторинг, деплой моделей)**

## Детальные спецификации Use Cases

### UC-01 — Регистрация / Вход

Предусловия: Пользователь не обязан быть зарегистрирован; требуется сеть для регистрации.

Основной сценарий:

1. 1. Открыть приложение; выбрать регистрация/вход.
2. 2. Ввести email и пароль или использовать OAuth.
3. 3. Отправить данные на Backend и получить ответ.
4. 4. При успешной регистрации выполнить вход и загрузить начальные данные.

Альтернативные сценарии / ошибки:

* - Неправильные учётные данные — показать ошибку.
* - Нет сети — предложить офлайн режим и повторить позже.

Постусловия: Пользователь авторизован; создан/получен профиль.

Критерии приёмки:

* - Пользователь успешно зарегистрирован; может войти в систему.

Тест-кейсы (примерные):

* - Регистрация с валидным email и паролем (ожидается успех).
* - Попытка регистрации с уже существующим email (ожидается ошибка).
* - Вход с неверным паролем (ошибка).

### UC-02 — Просмотр главной сводки

Предусловия: Пользователь авторизован; есть сохранённые транзакции или пустой аккаунт.

Основной сценарий:

1. 1. Открыть приложение после входа.
2. 2. Система загружает и отображает баланс, графики краткой сводки и уведомления.

Альтернативные сценарии / ошибки:

* - Нет данных — показать подсказку для добавления первой транзакции.

Постусловия: Пользователь видит актуальную сводку по счёту.

Критерии приёмки:

* - Баланс и полуcуммы корректно отображаются.
* - Графики не ломаются при пустых данных.

Тест-кейсы (примерные):

* - Открыть приложение с аккаунтом имеющим транзакции.
* - Открыть приложение с новым аккаунтом (проверить подсказку).

### UC-03 — Добавление транзакции (ручное)

Предусловия: Пользователь авторизован.

Основной сценарий:

1. 1. Нажать 'Добавить транзакцию'.
2. 2. Заполнить поля и сохранить.
3. 3. Клиент валидирует и сохраняет локально, затем синхронизирует с Backend.

Альтернативные сценарии / ошибки:

* - Нет сети — транзакция ставится в очередь для синхронизации.

Постусловия: Транзакция сохранена и отображается в списке.

Критерии приёмки:

* - Запись отображается в UI; при синхронизации нет ошибок.

Тест-кейсы (примерные):

* - Добавление транзакции в онлайн-режиме.
* - Добавление транзакции без сети (проверить очередь sync).

### UC-04 — Сканирование чека и создание транзакций

Предусловия: Пользователь авторизован; доступ к камере или изображению чека.

Основной сценарий:

1. 1. Сделать фото чека или выбрать изображение.
2. 2. Клиент отправляет изображение OCR Service и получает raw\_text.
3. 3. Парсер формирует Receipt и items; ML предлагает категории.
4. 4. Пользователь подтверждает/редактирует и сохраняет транзакции.

Альтернативные сценарии / ошибки:

* - OCR некорректен — пользователь вручную редактирует raw\_text.
* - ML не уверен — пользователь выбирает категорию вручную.

Постусловия: Receipt сохранён; транзакции созданы и связаны с receiptId.

Критерии приёмки:

* - Receipt и транзакции видны; автокатегории предложены и изменяются пользователем.

Тест-кейсы (примерные):

* - Сканирование четкого чека (проверить распознавание total,date).
* - Сканирование плохого фото (проверить ручную правку).

### UC-05 — Редактирование / Удаление транзакции

Предусловия: Пользователь авторизован; транзакция существует.

Основной сценарий:

1. 1. Выбрать транзакцию из списка.
2. 2. Изменить поля и сохранить или выбрать удалить.
3. 3. Клиент обновляет локально и синхронизирует изменения.

Альтернативные сценарии / ошибки:

* - Удаление офлайн — пометка удаления и синхронизация при подключении.

Постусловия: Изменения применены или транзакция удалена.

Критерии приёмки:

* - Изменения корректно отражены в UI и на сервере.

Тест-кейсы (примерные):

* - Редактирование суммы и категории.
* - Удаление транзакции и проверка отсутствия её в списке.

### UC-06 — Управление категориями и тегами

Предусловия: Пользователь авторизован.

Основной сценарий:

1. 1. Открыть управление категориями.
2. 2. Добавить/редактировать/удалить категорию.
3. 3. Система сохраняет и применяет изменения.

Альтернативные сценарии / ошибки:

* - Удаление категории с привязанными транзакциями — предложить пересортировать транзакции в другую категорию.

Постусловия: Список категорий обновлён.

Критерии приёмки:

* - Новые категории видимы при создании транзакций.

Тест-кейсы (примерные):

* - Создание пользовательской категории.
* - Удаление категории с транзакциями (проверка предложений).

### UC-07 — Поиск и фильтрация транзакций

Предусловия: Пользователь авторизован; есть транзакции.

Основной сценарий:

1. 1. Открыть список транзакций.
2. 2. Ввести поисковый запрос или выбрать фильтры (дата, категория).
3. 3. Система отображает результаты и обновляет подсчёты.

Альтернативные сценарии / ошибки:

* - Неверный формат даты — показать подсказку.

Постусловия: Отфильтрованный список отображён.

Критерии приёмки:

* - Фильтрация и поиск работают корректно и быстро.

Тест-кейсы (примерные):

* - Поиск по описанию.
* - Фильтрация по дате и категории.

### UC-08 — Просмотр отчётов / графиков

Предусловия: Пользователь авторизован; имеются данные для выбранного периода.

Основной сценарий:

1. 1. Открыть раздел 'Отчёты'.
2. 2. Выбрать период и фильтры.
3. 3. Система строит графики и вычисляет ключевые метрики.

Альтернативные сценарии / ошибки:

* - Мало данных — показать подсказки.

Постусловия: Отчёты отображены и можно экспортировать данные.

Критерии приёмки:

* - Графики корректно показывают агрегации.

Тест-кейсы (примерные):

* - Построение месячного отчёта.
* - Экспорт данных отчёта.

### UC-09 — Создание и отслеживание цели (Goal)

Предусловия: Пользователь авторизован; есть исторические данные желательно >1 месяц.

Основной сценарий:

1. 1. Перейти в раздел 'Цели' и создать цель.
2. 2. Задать параметры цели и период.
3. 3. Система оценивает baseline и показывает прогноз и рекомендации.

Альтернативные сценарии / ошибки:

* - Мало данных — предложить увеличить период наблюдения.

Постусловия: Цель создана и отслеживается.

Критерии приёмки:

* - Прогресс цели корректно рассчитывается.

Тест-кейсы (примерные):

* - Создать цель для категории и проверить расчёт прогресса.
* - Изменить цель и проверить пересчёт.

### UC-10 — Экспорт / импорт данных (JSON)

Предусловия: Пользователь авторизован.

Основной сценарий:

1. 1. Выбрать экспорт/импорт в настройках.
2. 2. При экспорте сформировать JSON и предложить скачать/сохранить.
3. 3. При импорте валидировать JSON и добавить транзакции.

Альтернативные сценарии / ошибки:

* - Неверный формат JSON — показать ошибку и пример структуры.

Постусловия: Данные экспортированы/импортированы.

Критерии приёмки:

* - Экспортируемый JSON валиден и содержит все транзакции.

Тест-кейсы (примерные):

* - Экспорт аккаунта с 10 транзакциями.
* - Импорт валидного JSON (ожидается успешный импорт).

### UC-11 — Резервное копирование / синхронизация (sync)

Предусловия: Пользователь авторизован.

Основной сценарий:

1. 1. Клиент синхронизирует локальные изменения с Backend при подключении.
2. 2. При отсутствии сети изменения ставятся в очередь.
3. 3. Система выполняет бэкапы на стороне сервера.

Альтернативные сценарии / ошибки:

* - Конфликты синхронизации — применить стратегию last-write-wins или prompt user.

Постусловия: Данные синхронизированы и сохранены в облаке.

Критерии приёмки:

* - Нет потерь данных при нормальной синхронизации.

Тест-кейсы (примерные):

* - Отключить сеть, добавить транзакцию, подключить сеть и проверить синхронизацию.
* - Симулировать конфликт изменения одной записи на клиенте и сервере.

### UC-12 — Отключение/включение анонимной аналитики (consent)

Предусловия: Пользователь авторизован.

Основной сценарий:

1. 1. Перейти в настройки приватности.
2. 2. Включить или отключить опцию отправки анонимных данных.
3. 3. Система сохраняет выбор и прекращает/возобновляет отправку аналитики.

Альтернативные сценарии / ошибки:

* - Нет сети — настройка сохраняется локально и синхронизируется позже.

Постусловия: Настройка сохранена; аналитика отправляется в соответствии с выбором.

Критерии приёмки:

* - После отключения аналитика действительно не отправляется.

Тест-кейсы (примерные):

* - Включить аналитику (проверить отправку).
* - Отключить аналитику (проверить отсутствие отправки).

### UC-13 — Отправка обратной связи для ML (коррекция категории)

Предусловия: Пользователь авторизован; есть транзакции с автокатегорией.

Основной сценарий:

1. 1. Пользователь исправляет категорию транзакции.
2. 2. Клиент формирует event feedback и отправляет в Backend/ML pipeline.
3. 3. ML команда/сервис использует данные для дообучения модели.

Альтернативные сценарии / ошибки:

* - Нет сети — фидбек кешируется и отправляется позже.

Постусловия: Фидбек сохранён и доступен для обучения моделей.

Критерии приёмки:

* - Фидбек попадёт в ML dataset; подтверждение в UI.

Тест-кейсы (примерные):

* - Исправление категории и проверка наличия event в логе.
* - Кеширование фидбека офлайн и отправка при синхронизации.

### UC-14 — Просмотр рекомендаций и прогнозов (ML)

Предусловия: Пользователь авторизован; есть достаточные исторические данные.

Основной сценарий:

1. 1. Перейти в раздел рекомендаций.
2. 2. Система запрашивает у ML сервиса прогнозы и рекомендации.
3. 3. Показать рекомендации с коротким объяснением и ссылками на действия.

Альтернативные сценарии / ошибки:

* - Недостаточно данных — предложить базовые рекомендации.

Постусловия: Рекомендации показаны пользователю.

Критерии приёмки:

* - Рекомендации отображаются и сопровождаются объяснением (why).

Тест-кейсы (примерные):

* - Проверить генерацию рекомендаций для активного аккаунта.
* - Проверить поведение при недостатке данных.

### UC-15 — Удаление аккаунта

Предусловия: Пользователь авторизован; подтверждён запрос на удаление.

Основной сценарий:

1. 1. Пользователь инициирует удаление аккаунта в настройках.
2. 2. Система предупреждает и просит подтверждение.
3. 3. Backend удаляет данные и отзываeт доступы.

Альтернативные сценарии / ошибки:

* - Запрос на восстановление невозможен без бэкапа.

Постусловия: Данные пользователя удалены; доступ запрещён.

Критерии приёмки:

* - После удаления пользователь не может войти; данные удалены с сервера.

Тест-кейсы (примерные):

* - Инициировать удаление и проверить отсутствие данных на сервере.
* - Проверка невозможности входа после удаления.

### UC-16 — Административные операции (логирование, мониторинг, деплой моделей)

Предусловия: Admin имеет доступ и привилегии.

Основной сценарий:

1. 1. Admin просматривает логи и метрики в панели администратора.
2. 2. Admin деплоит новую версию ML модели или проверяет статусы сервисов.

Альтернативные сценарии / ошибки:

* - Ограничение доступа — аудит и логирование действий админа.

Постусловия: Админские действия выполнены и зафиксированы в AuditLog.

Критерии приёмки:

* - Admin может просматривать логи и безопасно деплоить модели.

Тест-кейсы (примерные):

* - Проверка просмотра логов.
* - Тест деплоя ML модели в staging и проверка метрик.