

Requirements Document — Financery

1. Введение

1.1 Название продукта

Financery — мобильное приложение для учёта личных финансов с возможностями сканирования чеков (OCR), автоматической категоризации расходов и доходов, аналитики, прогнозов и персональных рекомендаций на основе машинного обучения.

1.2 Краткое описание и контекст

Financery помогает пользователю фиксировать транзакции (вручную или автоматически через распознанные чеки), отслеживать динамику затрат по категориям и целям, получать визуальные отчёты (графики, диаграммы), сравнивать цены по товарам/магазинам и получать рекомендации для оптимизации расходов. Приложение предусматривает синхронизацию данных с облачным бэкендом и возможность резервного копирования.

1.3 Границы проекта (что будет и чего не будет)

Включено в проект:

- Мобильное приложение (Android / iOS) с UI, реализованным через FlutterFlow / Flutter.
- Регистрация и аутентификация пользователей (email + пароль; возможность OAuth-провайдеров как опция).
- Сохранение и синхронизация данных пользователя в облачном хранилище (MVP — Firebase Firestore; опция — собственный REST/GraphQL API).
- Захват изображения чека и распознавание (OCR) для извлечения позиций, суммы, даты, продавца.
- Постобработка распознанного текста и формирование транзакций.
- Автоматическая и ручная категоризация транзакций; обучение/улучшение модели по обратной связи.
- Графики и отчёты по периодам (день/неделя/месяц/год), итог по периоду (доходы, расходы, сальдо).
- Экспорт/импорт данных в формате JSON (локально).
- Базовые ML-фичи: рекомендации по целям, прогноз расходов, анализ тенденций, сравнение цен на позиции.
- Базовая локализация (русский), поддержка нескольких валют.
- Политика конфиденциальности и опция согласия на анонимную агрегацию данных.

Не входит / не будет реализовано в рамках начальной версии (MVP), если не оговорено отдельно:

- Интеграция с банковскими API для автоматического получения банковских транзакций (может быть запланирована как future work).

- Обработка и хранение номеров банковских карт, прямые платежи из приложения.
- Юридические/финансовые консультации и подбор сложных финансовых продуктов.
- Полноценный десктоп-клиент (в будущем возможна веб-версия).
- Гарантированная точность распознавания на всех типах чеков (будет улучшаться с помощью ML и правил).

2. Требования пользователя

2.1 Программные интерфейсы

Перечень предполагаемых внешних интерфейсов и интеграций:

1. Аутентификация и база данных

- Firebase Authentication (Email/Password, опционально OAuth).
- Firebase Firestore (или Postgres/Managed DB в production).
- Firebase Storage (или S3/GCS для хранения изображений чеков).

2. OCR / распознавание текста

- Google ML Kit (on-device) — для быстрого MVP.
- /или/ Google Cloud Vision / AWS Textract / Azure Form Recognizer — для более точного server-side распознавания.
- /или/ open-source решения (Tesseract / TrOCR + постобработка) при собственном ML-пайплайне.

3. ML/AI сервисы

- Внешний ML-сервис (REST API) для:
 - классификации позиций в категории;
 - рекомендаций/прогнозов (time-series моделей);
 - нормализации наименований товаров (fuzzy matching).
- ML модели могут быть размещены в виде Cloud Function / FastAPI + модельный сервер (TorchServe/TF-Serving) либо использованы managed ML.

4. Аналитика и мониторинг

- Firebase Analytics / Sentry / аналогичные для логирования ошибок и аналитики (анонимно, при согласии пользователя).

5. Вспомогательные библиотеки

- Библиотеки Flutter/FlutterFlow для работы с камерой, загрузки файлов, локального кэша, гистограмм/диаграмм (например, charts_flutter).
- Криптографические библиотеки для шифрования данных (как на клиенте, так и на сервере).

6. Опции интеграции (future)

- CSV/OFX импорт банковских выписок.

- Партнёрские API для получения прайс-листов (при развитии функции сравнения цен).

2.2 Интерфейс пользователя

Опиши основной пользовательский поток и поведение системы. Ниже — текстовое описание и таблица «действие — реакция».

Основные экраны/функции (краткий список):

- Экран регистрации / входа.
- Главный экран — сводка (баланс, быстрые отчёты, уведомления).
- Список транзакций (фильтры по дате/категории/поиску).
- Экран добавления транзакции (ручного ввода).
- Экран сканирования чека (камера / загрузка фото) → результат распознавания → проверка и подтверждение транзакции.
- Экран категории / управление категориями.
- Экран отчётов и графиков.
- Экран целей (создание целей, прогресс).
- Настройки (валюта, локализация, приватность, бэкапы).
- Экран профиля и управление аккаунтом (удаление данных).

Таблица «действие пользователя — реакция системы» (пример):

Действие пользователя: Открыть приложение

Ожидаемая реакция системы: Показать экран аутентификации (если не авторизован) или главную панель (если авторизован).

Действие пользователя: Войти с email/паролем

Ожидаемая реакция системы: Проверка учётных данных → успешный вход → загрузка пользовательских данных → переход на Главный экран.

Действие пользователя: Нажать «Сканировать чек»

Ожидаемая реакция системы: Открыть камеру → предложить сделать фото → после съёмки вызвать OCR → показать parsed receipt для проверки пользователем.

Действие пользователя: Подтвердить распознанный чек

Ожидаемая реакция системы: Создание одной или нескольких транзакций с привязкой к receiptId; автоматическое предложение категории; показать уведомление «Сохранено».

Действие пользователя: Редактировать транзакцию

Ожидаемая реакция системы: Изменить поля (сумму, категорию, описание) → сохранить → обновить базу и UI.

Действие пользователя: Фильтр по категории/дате

Ожидаемая реакция системы: Отфильтровать список и пересчитать итог по периоду → обновить графики и показатели.

Действие пользователя: Установить цель

Ожидаемая реакция системы: Сохранить goal → ML подсчитывает прогноз/рекомендации и выводит их в разделе «Цели».

Действие пользователя: Запросить экспорт JSON

Ожидаемая реакция системы: Сформировать JSON, предоставить возможность скачать/поделиться или сохранить в облако.

Действие пользователя: Отключить аналитику

Ожидаемая реакция системы: Сохранить настройку → перестать отправлять анонимные данные.

UI/UX примечания:

- В интерфейсе должны быть подсказки для съёмки чеков: освещение, фокус, рамка для чека.
- Возможность отмены автокатегоризации и быстрой корректировки (swipe → поменять категорию).
- Минималистичный дизайн, тёмная тема по умолчанию, элементы интерфейса схожи с нативными элементами платформы.

2.3 Характеристики пользователей

Определение пользовательских групп и их характеристик:

1. Обычные пользователи (основная целевая группа)

- Возраст: 18–65.
- Уровень образования: среднее и выше.
- Техническая грамотность: базовая — уверенное пользование смартфоном, камера, установки приложений.
- Цели: контроль расходов, формирование бюджета, поиск экономии.
- Потребности: простой и интуитивный интерфейс, быстрый ввод транзакций, понятные графики.

2. Продвинутые пользователи / аналитики личных финансов

- Возраст: 25–50.
- Техническая грамотность: выше среднего.
- Цели: глубокий анализ трат, экспорт данных, детальные фильтры.
- Потребности: расширенные фильтры, экспорт/импорт данных, возможность настройки категорий и целей.

3. Тестировщики / разработчики (внутреннее использование)

- Техническая грамотность: высокий уровень.

- Потребности: доступ к логам, настройкам бэкенда, ability to toggle ML features.

Особенности интерфейса для всех пользователей:

- Подсказки и onboarding при первом запуске.
- Возможность быстро исправлять автоматические решения (категории, суммы).
- В настройках — простой доступ к политике приватности и управлению данными.

2.4 Предположения и зависимости

Список факторов, которые влияют на требования и реализацию:

1. Сетевое соединение

- Предполагается, что у пользователя есть интернет для синхронизации и использования cloud-OCR / ML. Поддерживается режим офлайн для захвата чеков с последующей синхронизацией.

2. Качество камеры

- Точность OCR зависит от качества камеры и условий съёмки. Требуются UX-подсказки и предобработка (автокоррекция, выравнивание).

3. Доступность сторонних сервисов

- Использование облачных OCR/ML сервисов предполагает платную тарификацию и SLA провайдеров (Google, AWS и т.п.). Их недоступность окажет влияние на функциональность.

4. Юридические/регуляторные ограничения

- Политика обработки персональных данных (включая требования законодательства РФ и возможных других юрисдикций). Необходимо обеспечить прозрачный consent-flow и механизм удаления данных.

5. Хранение и стоимость

- Выбор backend (Firebase vs собственный сервер) влияет на стоимость хранения и масштабируемость. Для MVP предпочтительна managed-platforma.

6. Пользовательские соглашения

- Перед сбором анонимных/агрегированных данных требуется явное согласие пользователя (opt-in). Без согласия — данные не используются для агрегации.

7. Версии ОС и аппаратные ограничения

- Приложение целевое на Android 8.0+ и iOS 13+. Поддержка более старых версий потребует дополнительных затрат.

3. Системные требования

Этот раздел содержит требования к продукту описанные на уровне достаточном для однозначного понимания того, как система может быть спроектирована и протестирована на предмет удовлетворения данных требований.

3.1 Функциональные требования

FR-1 — Регистрация и аутентификация

Система должна предоставлять регистрацию пользователей по email/паролю и возможность входа. Для аутентификации использовать безопасный механизм (например, Firebase Auth или JWT на собственном бэкенде).

FR-2 — Управление аккаунтом

Пользователь должен иметь возможность изменить профиль (display name), изменить пароль, выйти и удалить аккаунт (с удалением всех данных).

FR-3 — CRUD транзакций

Пользователь должен иметь возможность создавать, просматривать, редактировать и удалять транзакции (Income / Expense).

FR-4 — Категории и теги

Система должна предоставлять набор глобальных категорий и позволять пользователю создавать/редактировать/удалять собственные категории и теги.

FR-5 — Сканирование чеков (OCR)

Приложение должно позволять сделать фото чека или загрузить изображение и преобразовать его в structured JSON (receipt) с полями: items, total, date, vendor, raw_text.

FR-6 — Автоматическое формирование транзакций из чеков

Система должна предлагать создание одной или нескольких транзакций на основе распознанного чека и предлагать категории для каждой позиции.

FR-7 — Автоматическая категоризация (ML)

Для новой транзакции система должна автоматически предложить категорию с указанием confidence. Пользователь должен иметь возможность принять или изменить предложенную категорию. Система должна сохранять обратную связь для обучения модели.

FR-8 — Поиск и фильтрация

Пользователь может искать транзакции по описанию/продавцу/тегам и фильтровать по дате, категории, сумме и источнику (ocr/manual/import).

FR-9 — Отчёты и графики

Система должна предоставлять отчёты и визуализации: расходы/доходы по периодам (день/неделя/месяц/год), распределение по категориям, тренды по времени.

FR-10 — Цели и рекомендации

Пользователь может задавать финансовые цели (например, «уменьшить траты на категории X на Y%»). Система должна вычислять текущий прогресс и выдавать рекомендации и прогнозы (на основе ML/статистики).

FR-11 — Экспорт / импорт данных

Приложение должно поддерживать экспорт и импорт данных пользователя в JSON-формате.

FR-12 — Резервное копирование и синхронизация

Данные должны синхронизироваться с облаком и поддерживаться резервное копирование; в случае отсутствия сети должна быть очередь для синхронизации.

FR-13 — Управление приватностью и согласиями

В настройках пользователь может включать/отключать отправку анонимных данных и просмотреть политику приватности; при включении сбор анонимных данных возможен.

FR-14 — Сравнение цен и индекс товаров

Система должна накапливать информацию о позициях и формировать усреднённые цены по товарам и регионам, а также выдавать рекомендации «где дешевле».

FR-15 — Локализация и валюты

Поддержка как минимум одной локали (ru) и возможность выбора валюты; суммы отображаются с двумя знаками после запятой; при необходимости поддержка конвертации валют (future).

FR-16 — Уведомления

Push-уведомления (опционально) для напоминаний по целям и важным событиям (например: «вы превысили бюджет по категории»).

FR-17 — Логи и мониторинг

Система должна вести логи ошибок и ключевых событий (с учётом приватности), доступные разработчикам для отладки.

FR-18 — Интерфейс обратной связи

Пользователь должен иметь возможность пометить автоматическую категоризацию как неверную и добавить корректную — это событие должно использоваться для дообучения модели.

3.2 Нефункциональные требования**3.2.1 Атрибуты качества**

Ниже перечислены ключевые качества системы, почему они важны и как их измерять/проверять.

Q-1 — Надёжность (Reliability)

Пользователь ожидает, что данные о транзакциях сохраняются корректно и не

теряются.

Требование и метрики: Достигать целевого SLA для хранения данных — минимум 99.5% успешных транзакций хранения; отсутствовать потерям данных при обычных условиях. Тестирование: стресс-тесты сохранения; сценарии отключения сети и повторной синхронизации; проверка целостности при восстановлении бэкапа.

Q-2 — Безопасность (Security)

Финансовые данные чувствительны; утечка данных наносит вред пользователю и репутации сервиса.

Требование и метрики: TLS (HTTPS) для всех соединений; токены доступа защищены (Android Keystore / iOS Keychain); данные at-rest шифруются (AES-256) на сервере; доступ по роли для внутренних инструментов. План тестирования: проверка шифрования, pentest, сканирование уязвимостей, ревью секретов. Обработка регламентирована политикой безопасности.

Q-3 — Конфиденциальность / Privacy

Сбор персональных данных регулируется и вызывает озабоченность пользователей.

Требование: Ясный consent screen; опция opt-in/opt-out для анонимной агрегации; механизм удаления аккаунта (включая удаление всех персональных данных) — action must be irreversible after user confirmation. Тестирование: проверка работоспособности удаления, проверка отсутствия PII в агрегированных данных.

Q-4 — Производительность (Performance)

Пользователи ожидают быстрый отклик интерфейса и минимальные задержки при синхронизации.

Требование и метрики: Время открытия главного экрана после авторизации ≤ 1.5 s; Первая загрузка списка транзакций (первые 50) ≤ 300 ms; OCR (on-device) — результат отображается ≤ 3 s; server-side OCR — время обработки $\leq 5-10$ s; Синхронизация изменений на фоне — latency < 10 s. Тестирование: нагрузочные тесты, профайлинг UI.

Q-5 — Масштабируемость (Scalability)

Система должна поддерживать вертикальное и горизонтальное масштабирование бэкенда. Для MVP предполагается использование управляемых сервисов (Firebase); для production — возможность переноса на многосерверную архитектуру с горизонтальным масштабированием ML- и OCR-сервисов. Тестирование: нагрузочные тесты, план capacity.

Q-6 — Доступность (Availability)

Для production доступность сервисов $\geq 99.5\%$ (SLA для managed-сервисов). Тестирование: мониторинг uptime, симуляция отказов.

Q-7 — Удобство использования (Usability)

Время на выполнение типичных задач (добавить транзакцию; просканировать чек)

— менее 30–60 секунд; присутствует onboarding и подсказки. Тестирование: юзабилити-тесты с реальными пользователями, оценки SUS.

Q-8 — Совместимость (Compatibility / Portability)

Поддержка Android 8.0+ и iOS 13+; приложение корректно работает на устройствах с разными экранами и разрешением. Тестирование: тесты на популярных моделях, автоматические UI-тесты.

Q-9 — Поддерживаемость (Maintainability)

Кодовая база должна следовать стандартам и иметь модульные тесты; CI/CD — настроен для автоматической сборки и тестирования. Метрики: покрытие unit-тестами $\geq 60\%$ у критичных модулей; время восстановления CI < 1 час.

Q-10 — Точность ML (Accuracy / Explainability)

Классификатор категорий — целевая точность $\geq 85\%$ на валидационном наборе; модель прогнозирования — MAPE $< 20\%$ для месячных прогнозов. Все рекомендации должны снабжаться объяснением (краткая подсказка). Тестирование: ML-валидация, A/B тестирование рекомендаций.

Q-11 — Безотказность данных / резервирование (Backup & Recovery)

Регулярные бэкапы (ежедневные) и тестируемый план восстановления; возможность экспортировать весь аккаунт (JSON). Проверка: имитация восстановления из бэкапа раз в месяц как тест.

Q-12 — Локализация (Localization)

Интерфейс на русском (минимум); структура UI и коды валют должны позволять добавление новых языков/региональных форматов. Тестирование: проверка отображения на рус/латинице, формат дат и денежных значений.

4. Примеры критериев приёмки

1. Пользователь зарегистрировался и входит — ОК.
2. Сканирование стандартного чека (четкое фото) — распознаются дата, итог и минимум 80% позиций (для MVP допустимо меньше; в дальнейшем улучшается).
3. Автокатегоризация: при распознавании транзакции предлагается категория; пользователь может её изменить; система сохраняет обратную связь.
4. Экспорт JSON для аккаунта — успешно скачивается и содержит корректную структуру.
5. Удаление аккаунта — все пользовательские данные удаляются (проверяем на бэкенде).
6. Отключение отправки аналитики — отправка действительно прекращается.

5. Примечания по реализации и рекомендации

- Для быстрого старта (MVP) рекомендую связку: FlutterFlow + Firebase (Auth, Firestore, Storage) + Google ML Kit (on-device OCR). Это минимизирует время разработки и эксплуатационные сложности.

- Для production: отдельный ML-пайплайн (сервер), более точный серверный OCR и нормализация наименований товаров.
- Реализуй механизм «feedback loop», чтобы пользователи могли корректировать автокатегоризацию — это ключ к улучшению качества ML.
- Обязательно подготовь и предоставь пользователю понятный consent-flow и возможность удаления данных.