

Разработка и пилотирование системы автоматизированного подбора банковских продуктов и маршрутизации кредитных заявок

Инициатор: цифровая платформа vsebanki.kz

Направление: информационно-коммуникационные технологии (ИКТ)

Результат: работоспособный MVP с тестовыми интеграциями

Назначение документа: размещение технологической задачи на площадке Astana Hub

Редакция 1.0 · 21 октября 2026 года

1. Основание и бизнес-проблема

Кредитному брокеру требуется единый технологический механизм для приема заявок, проверки соответствия заемщиков условиям кредитных продуктов и управляемой передачи заявок в банки-партнеры. При разрозненной обработке повышаются операционные затраты, риск повторной отправки, время ожидания и сложность контроля статусов. Текущее состояние инфраструктуры и количественные показатели должны быть подтверждены инициатором до начала проекта.

2. Цель и границы проекта

Создать MVP программного комплекса, который принимает кредитную заявку, проверяет данные, отбирает подходящие продукты по формализованным правилам, ранжирует варианты, направляет запросы в тестовые банковские API и сохраняет результаты обработки с полным журналом событий.

Входит в MVP: API заявок, справочник продуктов и правил, движок фильтрации, настраиваемое ранжирование, оркестрация отправок, три тестовых банковских адаптера, обработка ответов и webhook, базовая панель оператора, журнал аудита, документация и тесты.

Не входит в MVP: подключение реальных банков без согласованных спецификаций и договоров; получение кредитных отчетов; автоматическое принятие решений от имени банка; обучение ML-модели на персональных данных; промышленный запуск с реальными клиентскими данными без отдельной проверки требований безопасности и права.

3. Функциональные требования

3.1. Прием и регистрация заявок

- REST API принимает параметры: идентификатор клиента, возраст, доход, действующие платежи, сумму и срок кредита, тип занятости, регион, согласия и служебный идентификатор запроса. Набор полей уточняется при проектировании.

- Валидация обязательных полей, типов, диапазонов и согласованности значений; ошибки возвращаются в машиночитаемом виде.
- Повторный запрос с тем же ключом идемпотентности и тем же содержимым возвращает ранее созданную заявку; с измененным содержимым — конфликт.

3.2. Справочник банковских продуктов и правил

- Хранение версионизируемых правил: возраст, доход, сумма, срок, занятость, регион, допустимая расчетная долговая нагрузка и дополнительные ограничения.
- Для каждого исключенного продукта сохраняются коды причин и версия примененного набора правил.

3.3. Подбор и ранжирование

- Сначала выполняется обязательная фильтрация по правилам; далее подходящие продукты ранжируются по настраиваемой модели с учетом исторических агрегированных показателей, времени ответа, доступности API и бизнес-приоритетов.
- Результат включает рейтинг, объяснение основных факторов и версию модели. Статистические показатели при недостатке наблюдений не должны трактоваться как надежная вероятность одобрения.

3.4. Оркестрация и интеграции

- Первичная отправка — не более чем в два тестовых банка одновременно; последующие отправки запускаются по настраиваемым правилам после отказа, тайм-аута или неподходящего предложения.
- Каждый банковский адаптер поддерживает преобразование данных, авторизацию в тестовой среде, корреляционные идентификаторы, тайм-ауты и обработку ошибок.
- Неопределенный результат передачи не приводит к слепому повтору: при отсутствии идемпотентности у банка задача переводится на сверку статуса или ручную проверку.

3.5. Статусы, события и кабинет оператора

- Статусы: создана, проверяется, продукты подобраны, направлена, на рассмотрении, предварительно одобрена, отказ, завершена, требует проверки.
- Отдельно ведутся статусы отправок в каждый банк; переходы ограничены правилами жизненного цикла.
- Панель оператора отображает заявки, банки, результаты фильтрации, попытки интеграций, причины ошибок, временные отметки и историю изменений.

4. Системы и источники данных

- vsebanki.kz: источник введенных клиентом заявок; интеграция через внутренний REST API. Наличие и контракт действующего API необходимо подтвердить.
- Справочник банковских продуктов: внутреннее хранилище правил и условий; первоначальное наполнение тестовыми данными и управление через административный интерфейс.
- Три банковских mock-сервиса: тестовые REST API и webhook с сценариями «одобрено», «отказ», «тайм-аут», «ошибка 500», «повторный webhook». Это не утверждение о наличии действующих интеграций.

- Исторические показатели: обезличенные или агрегированные результаты предыдущих заявок, если такие данные законно доступны и пригодны по качеству; иначе в MVP используются синтетические наборы.

5. Техническая реализация и компоненты

Предлагается модульная серверная архитектура: API-шлюз/приложение, сервис правил и ранжирования, оркестратор заявок, очередь фоновых задач, интеграционные адаптеры, реляционная БД, административный веб-интерфейс и централизованное журналирование. Допускаются эквивалентные технологии при соблюдении требований.

- Реляционная БД (например, PostgreSQL): заявки, версии правил, отправки, ответы, события и аудит; уникальные ограничения для защиты от дублирования.
- Очередь (например, RabbitMQ): асинхронная обработка; транзакционный outbox или эквивалентный механизм для согласования сохранения заявки и публикации задания.
- Механизмы надежности: повторная доставка с контролем идемпотентности, ограничение параллелизма, backoff, circuit breaker, dead-letter queue и мониторинг зависших задач.
- Контроль доступа по ролям, TLS, защита секретов, маскирование персональных данных в логах и настройка сроков хранения. Реальные персональные данные в тестовом контуре не используются.

6. API MVP (минимальный контракт)

- POST /api/applications — создание заявки; заголовок Idempotency-Key; ответ 201 или существующий идентификатор при повторе.
- GET /api/applications/{id} — статус заявки и общая сводка.
- GET /api/applications/{id}/banks — проверенные продукты, причины исключения и результаты маршрутизации.
- GET /api/applications/{id}/history — хронология событий.
- POST /api/integrations/{bankId}/webhook — прием подписанных тестовых уведомлений с защитой от повторной обработки.

Форматы запросов и ответов, коды ошибок, права доступа и OpenAPI-спецификация утверждаются в ходе проектирования.

7. Пилотный сценарий и проверка MVP

Оператор или тестовый клиент создает синтетическую заявку. Сервис проверяет параметры, сопоставляет их с правилами трех тестовых банков, формирует ранжированный список, направляет запросы в первые два банка и обрабатывает разные сценарии ответов. Итоги, причины исключения, статусы и история видны в административной панели.

Критерии приемки

- Демонстрируется полный сквозной сценарий от POST заявки до отображения результата; доступны исходный код, OpenAPI, инструкция запуска и схема архитектуры.
- Поддержаны минимум три mock-адаптера; каждый воспроизводит успешный ответ, отказ, тайм-аут, HTTP 500 и дублирующий webhook.
- Повтор с одинаковым ключом идемпотентности не создает вторую заявку; конкурентные запросы и повторные webhook покрыты автоматизированными тестами.

- После рестарта фонового обработчика необработанные задания восстанавливаются без потери записей; неопределенные банковские отправки не дублируются автоматически без подтверждения безопасности повтора.
- В согласованной тестовой среде при 100 параллельных запросах регистрации р95 времени ответа API не более 1 секунды; внешние вызовы банков не включаются в этот показатель.
- Для каждого исключенного продукта доступны причины, для каждой отправки — статус, идентификатор корреляции и история событий.

8. Этапы и ожидаемые результаты

- Этап 1 — 2 недели: обследование, уточнение существующих систем, спецификация требований, модель данных, архитектура, API-контракты.
- Этап 2 — 4 недели: разработка API, хранилища, правил отбора и алгоритма ранжирования.
- Этап 3 — 3 недели: оркестрация, очередь, mock-адаптеры, обработка отказов и webhook.
- Этап 4 — 2 недели: административный интерфейс, аудит и метрики.
- Этап 5 — 2 недели: интеграционные и нагрузочные тесты, исправления, демонстрация и передача документации.

Предварительная длительность: 13 недель; сроки уточняются после обследования, оценки команды и согласования границ MVP.

9. Результаты, передаваемые инициатору

- Исходный код и конфигурация тестового развертывания; документация по сборке и запуску.
- Архитектурная схема, модель данных, описание алгоритма ранжирования и ограничений его применимости.
- OpenAPI-спецификация, тестовые данные, три mock-адаптера, протоколы испытаний и инструкция оператора.
- Демонстрация MVP и акт приемки по согласованным критериям.

10. Допущения и ограничения

Документ описывает запрашиваемую разработку, а не подтверждает наличие перечисленных компонентов или партнерских банковских API у vsebanki.kz. Банковские правила и статистика в MVP могут быть синтетическими. Подключение реальных банков, использование персональных данных, доступ к кредитным бюро, вопросы согласий и соблюдения законодательства Республики Казахстан оформляются отдельными требованиями и согласованиями. Квалификация проекта как ИКТ/НИОКР определяется правилами конкретной программы, а не одним названием задачи.

11. Ответственное лицо и контакты

Юридическое наименование инициатора: ТОО «do-it»

Проект / сайт: vsebanki.kz

Ф. И. О. ответственного: Коваленко И.В.

Должность: директор

Телефон: +77770880914

Email: info@vsebanki.kz

Перед публикацией необходимо подтвердить интеграционные возможности действующей платформы и согласовать критерии пилота.