Техническое задание по конкурсу на внедрение Системы дистанционного банковского обслуживания юридических лиц (инсталляция и внедрение ПО, миграция, техническая поддержка).

1. Назначение и общие положения

1.1 Назначение проекта
Разработка, внедрение и поддержка новой системы Дистанционного банковского обслуживания (ДБО) для юридических лиц АКБ «Hamkorbank». Решение должно удовлетворять функциональным и техническим требованиям банка, обеспечивать масштабируемость, высокую производительность и отказоустойчивость.

1.2 Основные цели проекта
- Повышение качества и скорости обслуживания клиентов.
- Обеспечение полной автоматизации ключевых бизнес-процессов.
- Создание гибкой и масштабируемой инфраструктуры ДБО.

1.3 Условия реализации
- Удаленное подключение к серверам Банка для выполнения работ.
- Русскоязычная техническая поддержка 24/7.
- Согласие на подписание соглашения о конфиденциальности (NDA).

2. Общие требования к исполнителю

2.1 Требования к команде
Исполнитель должен иметь команду с опытом выполнения аналогичных проектов:
- Архитекторы, системные аналитики, разработчики (фронтенд и бэкенд), DevOps-инженеры.

2.2 Требования к опыту

- Наличие не менее 2 успешных кейсов по внедрению ДБО для юридических лиц непосредственно самой компанией, головной организацией или в рамках холдинга
- Предоставление отзывов от клиентов по форме, утвержденной банком.

3. Сроки реализации

3.1 План выполнения работ
Общий срок реализации проекта: до 16 месяцев (сроки согласовываются с учетом этапов и поставок).

3.2 Предполагаемые этапы проекта:
- Разработка и согласование архитектуры и ТЗ.
- Создание инфраструктуры и архитектуры.
- Разработка и тестирование модулей системы.
- Интеграция с АБС, ESB и внешними системами.
- Пилотный запуск на группе клиентов.
- Внедрение в промышленную эксплуатацию и поддержка.

4. Функциональные требования

4.2 Бизнес-функционал ДБО

1. Кор-часть ДБО:
  - Модуль расчетно-кассового обслуживания в национальной валюте (РКО).
  - Валютные операции (SWIFT, покупка/продажа валюты).
  - Зарплатные проекты.
  - Корпоративные карты.
  - Платежные требования и картотека.

2. Собственная бизнес логика:
- Функционал Кор-части ДБО должен быть реализован как независимый от АБС сервис, иметь собственную процессную и бизнес логику.
-Сервис должен позволять вносить в него изменения.
-Сервис должен самостоятельно осуществлять все необходимые валидации без задействования логики АБС.
-Сервис должен иметь свое открытое API и обеспечивать интеграцию с ним любых внутрибанковских систем и внешних систем.

3. Фронт-часть (UI/UX):
- Разработка интерфейсов на Web, iOS, Android.
- Локализация интерфейсов (русский, узбекский латиница, узбекский кириллица, английский).

4. Ролевая модель пользователей:
- Назначение ролей и прав доступа.
- Многоуровневая модель подписания платежей.
- Админ панель для сотрудников банка

5. Интеграции:
- Интеграция с АБС банка, шиной (ESB).
- Внешние системы (1С, Didox, процессинги, GPI-трекер).

6. Сервисы:
- Корпоративное хранилище данных.
- Генератор PDF-документов.
- Чат с техподдержкой.
- Сервис нотификаций(уведомления об операциях, сервисные уведомления и маркетинговые уведомления)

5. Нефункциональные требования

5.1 Техническая инфраструктура
Примечание - является обязательным требованием

- Собственная на сервере заказчика (on-premise): продуктивная среда, среда разработки, тестовая среда, препрод среда, среда нагрузочного тестирования
- Мониторинг и логирование всех модулей системы.
- Высокая отказоустойчивость за счет создания кластеризации из двух или нескольких серверов, расположенные на двух ЦОД-ах с моментальными репликами данных между серверами.
- Поддержка кластеризации серверов актив-актив / актив-пассив для обеспечения бесперебойной работы.
- Наличие механизмов автоматического/ручного переключения на резервные серверы в случае отказа.

- Кроссплатформенность – поддержка развертывания на различных ОС и использование open source СУБД.

5.2 Производительность
Примечание - является обязательным требованием

- Обработка до 50 тыс. операций в день.
- Поддержка 30–50 тыс. активных пользователей.

5.3 Масштабируемость
- Динамическое горизонтальное масштабирование.

5.4 Безопасность
Примечание - является обязательным требованием

5.4.1 Безопасность интеграции с другими системами
- Все интеграционные интерфейсы должны использовать зашифрованные каналы связи с использованием протоколов SSL/TLS.
- Для аутентификации запросов между системами необходимо применять токены или сертификаты безопасности.
- Предусмотреть систему логирование всех операций интеграции для последующего аудита.

5.4.2 Безопасность платежных и карточных данных
- Хранение и обработка карточных данных должны соответствовать требованиям PCI DSS.
- Для авторизации транзакций должна быть реализована двухфакторная аутентификация (2FA).
- Все операции с платежами и картами должны логироваться с детализацией данных для аудита.

5.4.3 Защита передачи данных
- Обязательное использование протокола HTTPS с поддержкой актуальных версий TLS для защиты передачи данных.
- Защита от MITM-атак путём внедрения механизма HSTS.
- Обеспечение целостности данных с помощью контрольных сумм и цифровых подписей.

5.4.4 Конфиденциальность, целостность и доступность (CIA-триада)

Конфиденциальность:
- Реализация политики минимальных привилегий и ролевого доступа.
- Шифрование всех данных, включая резервные копии.

Целостность:
- Применение хэширующих функций для проверки целостности данных.
- Включение механизма версионного контроля для критических данных.

Доступность:
- Использование отказоустойчивых кластеров и репликации данных.
- Регулярное выполнение тестов на отказоустойчивость и восстановление после сбоев.

5.4.5 Сетевая безопасность

Необходимо обеспечить односторонний доступ в DMZ:
- Мобильное приложение должно иметь возможность обращаться только к сервисам, размещенным в демилитаризованной зоне (DMZ), без возможности прямого доступа из DMZ во внутреннюю сеть.
- Доступ из внутренней сети в DMZ разрешен для обработки данных, но исходящий доступ из DMZ во внутреннюю сеть должен быть заблокирован.

5.4.6 Тестирование безопасности
Перед запуском в промышленную эксплуатацию необходимо провести пен-тест без уязвимостей

5.5 Требования к передаваемому коду
Примечание - является обязательным требованием
- Код должен быть написан с соблюдением общепринятых стандартов оформления и правил для используемого языка программирования.
- Понятные названия переменных, функций и классов, отражающие их назначение.
- Код должен быть структурирован в соответствии с принципами SOLID
- Каждый модуль, класс и функция должны содержать комментарии с описанием их назначения и логики работы.
- Код должен быть покрыт не менее чем на 85% юнит-тестами
- Код не должен содержать секретные данные (пароли, ключи API). Для этих целей следует использовать хранилища секретов.

5.6 Нагрузочное тестирование
Примечание - является обязательным требованием
Должны быть разработаны и переданы скрипты нагрузочного тестирования, а также результаты их проведения. Используемые виды нагрузочного тестирования:
- Тестирование производительности (Performance Testing)
- Стресс-тестирование (Stress Testing)
- Тестирование масштабируемости (Scalability Testing)
- Тестирование стабильности (Stability Testing)
- Тестирование на пиковой нагрузке (Peak Load Testing)

- Тестирование с учетом замедления ответов внешних систем

5.7 Тестирование
Примечание - является обязательным требованием
100% функционала должно быть покрыто тестами, 70% из которых автотесты, с учетом изолирования заглушками от внешних систем. Необходимо также выделить смоук тест, который должен запускаться перед каждым развертыванием на среде.

5.8 Требования к мониторингу
Примечание - является обязательным требованием
Централизованный сбор данных:
- Логи всех микросервисов должны собираться в централизованное хранилище (например, Elasticsearch, Loki).
- Метрики производительности и состояния системы должны агрегироваться с использованием инструментов мониторинга
Общие метрики:
- Время отклика каждого микросервиса (response time).
Уровень ошибок (error rate) по HTTP-кодам (например, 4xx, 5xx).
- Количество активных запросов (concurrent requests) и пропускная способность (throughput).
- Использование ресурсов: CPU, память, дисковое пространство, сеть.
Аварийные уведомления:
- Настроить оповещения при достижении критических значений метрик (например, превышение порога ошибок, высокая нагрузка на ресурсы, отказ сервиса).
- Уведомления должны направляться в систему управления инцидентами.
Детализированное логирование:
- Логи должны включать информацию о запросах (метод, URI, статус), времени выполнения операций и возникших ошибках.
- Поддержка уровня логирования (debug, info, warning, error) с возможностью гибкой настройки.
Слежение за зависимостями:
- Отслеживание состояния внешних интеграций (например, базы данных, очереди сообщений, сторонние API).
- Автоматическое оповещение при недоступности внешних сервисов.
Трассировка запросов:
- Включить поддержку распределенной трассировки (например, с использованием OpenTelemetry или Zipkin) для анализа пути запросов между микросервисами.
Каждый запрос должен иметь уникальный идентификатор (trace ID).
Визуализация данных:

- Предоставить доступ к панели мониторинга с визуализацией ключевых метрик.
- Панели должны включать дашборды для каждого микросервиса и общую сводку по системе.
- Доступность системы в данный момент и по итогам текущего месяца
Мониторинг отказоустойчивости:
- Проверка доступности сервисов с использованием health checks (readiness/liveness probes).

Подробные функциональные и нефункциональные требования описаны в приложении “Требования для ДБО юридических лиц Hamkorbank”

6. Обучение и поддержка

6.1 Обучение
- Проведение обучения для технических специалистов банка.
- Предоставление учебных материалов для сотрудников банка.

6.2 Техническая поддержка
- Гарантийный период: 2 месяца.
- Реализация SLA с реакцией на критические инциденты в течение 30 минут.

Подробное описание в приложении “SLA по технической поддержке ДБО юридических лиц”

7. Порядок поставки и внедрения

- Поставка ПО в формате on-premise/on-cloud.
- Этапы внедрения согласовываются с банком и сопровождаются поэтапным отчетом о выполнении работ.

8. Дополнительные требования к исполнителю

- Предоставить поэтапный план работы с детализацией задач и сроков.

- Обязательное проведение пилотного запуска на 50 клиентов банка.

9. Критерии приемки

- Полное соответствие требованиям ТЗ и Техспецификации.
- Успешное прохождение функциональных и нагрузочных тестов.
- Отсутствие критических ошибок на момент промышленного запуска.

10. Приложения

1. Форма конкурсного предложения.
2. SLA по технической поддержке ДБО юридических лиц
3. Справка об опыте внедрения аналогичных решений.
4. Требования для ДБО юридических лиц Hamkorbank

Требования для ДБО для юридических лиц

1. Кор-часть ДБО

1.1. Техническая часть. Создание и развертывание инфраструктуры для нового ДБО

1. Разработка и развертывание инфраструктуры:

- Разработка инфраструктуры для нового ДБО должна учитывать существующий технологический стэк банка, предоставленный ранее.
- Включает создание тестовых, разработческих, предпромышленных и промышленных сред, обеспечивающих стабильную и безопасную

разработку, тестирование, включая нагрузочное и эксплуатацию системы.
- Среды должны быть доступны для всех участников проекта, включая команду разработки, системных аналитиков, тестировщиков, DevOps и других заинтересованных сторон.
- Каждая из сред должна быть изолирована и иметь отдельные настройки безопасности и контроля доступа.

2. Контейнеризация и микросервисная архитектура:

- Микросервисная архитектура является ключевым аспектом нового ДБО, каждый сервис (например, сервис для платежей в сумах, валютные платежи, управление ролями и доступами, генерация PDF и т.д.) должен быть разработан и развернут как отдельный микросервис.
- Все микросервисы должны быть контейнеризированы с использованием Docker или аналогичной технологии, чтобы обеспечить гибкость и простоту развертывания.
- Микросервисы должны быть управляемы через оркестрационную платформу Kubernetes, что позволит эффективно управлять масштабируемостью и надежностью системы.

3. CI/CD и DevOps практики:

- CI/CD (Continuous Integration/Continuous Deployment) пайплайны должны быть настроены для каждого микросервиса, обеспечивая автоматическое развертывание изменений в тестовых и продакшн средах.
- В рамках CI/CD пайплайна должны быть настроены автоматические тесты (юнит-тесты, интеграционные тесты, end-to-end тесты), а также системы мониторинга и оповещения об ошибках и сбоях.
- DevOps практики для управления и автоматизации настройки инфраструктуры и для создания и управления инфраструктурными компонентами.
- Методы для минимизации рисков во время развертывания новых версий сервисов.

4. Обеспечение высоких требований к безопасности и соответствие регламентам:

- Все компоненты инфраструктуры должны соответствовать требованиям безопасности банка и регламентам Центрального Банка Республики Узбекистан, в том числе шифрование данных, защищенные каналы связи (SSL/TLS).
- Внедрить системы мониторинга и защиты от DDoS-атак, а также системы логирования и аудита всех операций внутри системы.

5. Интеграция с существующей архитектурой банка:

- Инфраструктура нового ДБО должна быть интегрирована с внутренней ESB (Enterprise Service Bus) для взаимодействия с существующей АБС и другими банковскими системами.
- Обеспечить бесшовную интеграцию с ключевыми системами, такими как 1С, Didox, и другими внешними сервисами, как описано ранее в требованиях.

6. Управление мониторингом и логированием:

- Внедрить системы мониторинга для отслеживания состояния микросервисов, баз данных и всех критических компонентов инфраструктуры.
- Включить логирование всех транзакций и действий пользователей с помощью лог-агрегаторов и системы оповещений.

7. Масштабируемость и отказоустойчивость:

- Все компоненты инфраструктуры должны быть масштабируемыми с возможностью динамически увеличивать мощности по мере роста числа пользователей и транзакций.
- Реализовать систему аварийного восстановления и резервного копирования, чтобы минимизировать потери данных и время простоя в случае сбоев.

8. Управление версиями и конфигурацией:

- Внедрить инструменты для управления версиями конфигураций, таких как GitOps, чтобы все изменения инфраструктуры могли быть воспроизводимыми и контролируемыми.

9. Интеграция с инструментами контроля и безопасности:

- Реализовать интеграцию с инструментами контроля качества кода и безопасности, для предотвращения ошибок в коде и выявления уязвимостей на ранних этапах.

10. Непрерывное улучшение и гибкость:

- Инфраструктура должна позволять быстрое и гибкое масштабирование, добавление новых микросервисов и их доработку без нарушений работы системы.

11. Планируемая нагрузка на ДБО

1. Численность пользователей и нагрузка

- На момент запуска:
  - Ежедневная активная аудитория (DAU): 5-10 тыс. пользователей.
  - Ежемесячная активная аудитория (MAU): 15–20 тыс. пользователей.

- После 2-3 месяцев:
  - Ежедневная активная аудитория (DAU): прирост 1500–2000 пользователей в месяц.
  - Прогнозируемый DAU через 6 месяцев: 30–35 тыс. пользователей.
  - Ежемесячная активная аудитория (MAU): 60–80 тыс. пользователей.

- Через 1 год:
  - Ежедневная активная аудитория (DAU): 35–50 тыс. пользователей.
  - Ежемесячная активная аудитория (MAU): 80–100 тыс. пользователей.

Критерии активного пользователя:

-Активным пользователем считается пользователь который 1 и более раз зашел на главную страницу ДБО в течении 1 дня.

2. Масштабируемость и миграция клиентов

- Миграция клиентов с текущего ДБО на новое ДБО будет осуществляться партиями по 3-5 тыс. пользователей.
- Нагрузка на сервисы авторизации в моменты миграции может значительно увеличиваться, особенно в часы пик, и должна быть учтена в расчетах для масштабируемости.
- Инфраструктура должна поддерживать обработку увеличенного потока авторизаций в момент миграции клиентов, особенно при одновременной миграции тысяч пользователей.

3. Число операций и транзакций

- На момент запуска:
- Ожидается 2-3 тыс. платежных операций в день

- Первые 2-3 месяца от запуска:
  - Ожидается 5–6 тыс. платежных операций в день.

- Целевая нагрузка через год:
  - Ежедневное количество платежей: 40–50 тыс. операций.

- Инфраструктура должна обеспечивать стабильность обработки платежей в часы пиковой нагрузки.

4. Часы пиковой нагрузки

- Пиковая нагрузка на систему ожидается:
  - Ежедневно:
    - Утренние часы: с 9:00 до 11:00.
    - Вечерние часы: с 15:00 до 18:00.

  - В конце каждого месяца:
    - Периоды с 28 по 31 числа.

  - В периоды выплат зарплат:
    - Периоды с 1 по 5 числа каждого месяца.
    - Периоды с 20 по 25 числа каждого месяца.

Система должна штатно функционировать с учетом пиковой нагрузки в 180 операций в минуту.

5. Требования к масштабируемости системы

- Инфраструктура должна обеспечивать стабильную работу при нагрузке до 50 тыс. DAU с возможностью дальнейшего увеличения.
- Система должна быть готова к пиковым нагрузкам в дни миграции клиентов и в периоды высокой активности (ежедневные пики, конец месяца, выплаты зарплат).
- Требуется поддержка автоматического масштабирования системы при росте числа пользователей и операций.
- Время отклика и обработка запросов не должны ухудшаться при росте нагрузки до максимальных значений.

12. Отказоустойчивость

Система дистанционного банковского обслуживания (ДБО) должна быть спроектирована и развернута таким образом, чтобы обеспечить высокую степень отказоустойчивости и бесперебойной работы даже в случае полной потери одного из дата-центров (ЦОД). Для этого система должна удовлетворять следующим требованиям:

1. Распределенные ЦОДы:
   - ДБО должно быть развернуто как минимум в двух независимых дата-центрах, расположенных в разных местах для минимизации риска единой точки отказа.

- Каждый ЦОД должен иметь копию всех ключевых сервисов и баз данных для мгновенного переключения в случае отказа одного из ЦОДов.

2. Резервирование и балансировка нагрузки:
- Все критически важные компоненты системы (авторизация, обработка транзакций, взаимодействие с АБС, управление ЭЦП, база данных) должны иметь дублирующиеся экземпляры в каждом ЦОДе
- В случае падения одного из ЦОДов, нагрузка должна автоматически распределяться на оставшиеся компоненты в работающем ЦОДе без заметной задержки для пользователей.

3. Механизм автоматического переключения (failover):
- Вся система должна быть спроектирована для поддержания автоматического переключения между ЦОДами. При потере одного ЦОДа, рабочие сервисы должны немедленно и без участия оператора переключаться на резервный ЦОД, поддерживая работу пользователей.
- Время переключения должно составлять не более нескольких минут (идеально менее 1 минуты), чтобы минимизировать перерыв в работе ДБО.
- Данные пользователей, транзакции и логирование должны автоматически и моментально реплицироваться между дата-центрами.

4. Репликация данных в режиме реального времени:
- Базы данных должны быть настроены на асинхронную или синхронную репликацию между ЦОДами. Это позволит избежать потери данных при аварийных ситуациях и гарантировать непрерывность транзакций.
- Репликация должна осуществляться для всех данных: транзакций, авторизационных данных, данных клиентов и платежей.

5. Требования к устойчивости сервисов при увеличении нагрузки:
- Система должна выдерживать пиковые нагрузки при падении одного из ЦОДов. В случае потери одного ЦОД, второй должен обладать достаточной вычислительной мощностью, чтобы обрабатывать 100% запросов с увеличением числа пользователей.
- В случае миграции клиентов со старого ДБО на новое или во время критических периодов нагрузки (например, конец месяца или массовая загрузка платежей), система должна сохранять стабильную работу при увеличении нагрузки до 50% от среднедневного показателя.

6. Операции по мониторингу и восстановлению:
- Встроенные механизмы мониторинга и алертинга для раннего выявления проблем в одном из ЦОДов.
- Возможность быстрого восстановления данных и сервисов в случае отказа любого компонента системы.
- Наличие регулярного тестирования планов восстановления и аварийных ситуаций для минимизации времени простоя.

7. Минимизация влияния на пользователей:
   - В случае перехода на резервный ЦОД, пользовательская сессия и текущие действия (например, создание платежей или отправка транзакций) должны быть сохранены. Пользователь не должен заметить переключения между ЦОДами.

8. Оперативное восстановление и тестирование:
   - В случае потери одного ЦОД, должен быть запущен процесс восстановления, включая развертывание новых экземпляров сервисов и восстановление из резервных копий.

13. Интеграция с АБС

Интеграция с автоматизированной банковской системой (АБС)

Для успешного развертывания и функционирования нового дистанционного банковского обслуживания (ДБО) вендору необходимо обеспечить полную интеграцию с текущей автоматизированной банковской системой банка через внутренние интерфейсы, такие как шина данных или Enterprise Service Bus (ESB). Интеграция с АБС является одним из ключевых технических требований и должна быть реализована с учетом следующих аспектов:

1. Интеграция с ESB:
   - Вендор обязан интегрировать ДБО с существующей архитектурой банка через шину данных (ESB), которая уже используется для обмена данными между различными банковскими системами.
   - Все взаимодействие между модулями ДБО и АБС должно проходить через ESB для обеспечения стандартизированного и защищенного обмена данными.
   - ESB выступает в качестве промежуточного слоя для обработки всех запросов между ДБО и АБС, включая запросы на получение данных о счетах клиентов, балансе, транзакциях, платежах и другие критические операции.
  - Дополнительно вендору необходимо использовать уже существующие интерфейсы АБС-ФИДО для взаимодействия с ДБО.
  - ДБО должно обеспечивать полную синхронизацию с АБС по интерфейсам ФИДО для мгновенной обработки запросов и минимизации задержек в работе системы.

2. Аутентификация и безопасность:

- Для всех запросов, отправляемых из ДБО в АБС через ESB, необходимо использовать стандарты безопасности для обеспечения конфиденциальности и целостности данных.
- Все интерфейсы должны быть настроены на работу с механизмами аутентификации и авторизации в АБС для предотвращения несанкционированного доступа.

3. Обработка и синхронизация данных:
- Взаимодействие между ДБО и АБС должно поддерживать высокую скорость обработки данных и синхронизацию в режиме реального времени. Это особенно критично для операций с транзакциями и платежами.
- Любые изменения в данных клиента (остатки по счетам, проведение транзакций, выписки) должны мгновенно отражаться в интерфейсе ДБО и быть согласованы с АБС.

4. Мониторинг и управление запросами:
- Необходимо разработать механизмы мониторинга всех запросов, которые отправляются в АБС через ESB. Это позволит отслеживать статус выполнения запросов, фиксировать ошибки и своевременно реагировать на возникшие проблемы.
- Вендору также потребуется настроить логирование всех операций, связанных с обменом данными между ДБО и АБС, для дальнейшего анализа и устранения неполадок.

5. Тестирование и валидация интеграции:
- В процессе разработки и развертывания ДБО, вендору необходимо провести полное тестирование интеграции с АБС через ESB и интерфейсы ФИДО для проверки корректности работы всех запросов.
- Особое внимание следует уделить тестированию под нагрузкой, чтобы убедиться, что система выдерживает пиковую активность без потери производительности.

1.2 Авторизация пользователей

Примечание - является обязательным требованием

- Полная реализация первичной и вторичной авторизации пользователей с возможностью восстановления доступа через SMS или e-mail, поддержка бэкенда и фронта на Web, iOS и Android.
- Внедрение авторизации с использованием биометрических данных (Face ID, Touch ID) и двухфакторной аутентификации (2FA), реализованное на всех трех платформах.

1.2. Авторизация пользователей

Функциональные требования:

1. Первичная авторизация:

- Мобильная версия (iOS, Android):
  - Клиент авторизуется по логину и паролю, выданным сотрудником IT банка.
  - После ввода логина и пароля система отправляет SMS-код на номер телефона клиента, который был указан при выдаче учетной записи в банке.
  - Клиент вводит полученный SMS-код для подтверждения входа.
  - После успешного подтверждения клиенту предлагается установить код для входа в приложение (PIN).
  - Далее клиенту предлагается активировать биометрическую авторизацию:
    - iOS: Touch ID или Face ID.
    - Android: поддержка всех возможностей биометрической авторизации в рамках системы Android.

- Веб-версия:
  - Клиент также вводит логин и пароль на странице авторизации.
  - После ввода данных клиент может выбрать один из двух вариантов подтверждения:
    - Подтверждение с помощью SMS-кода (аналогично мобильной версии).
    - Подтверждение с помощью физической ЭЦП. Клиент вставляет в свой компьютер действующую ЭЦП с активным токеном, выданным банком.
  - Для веб-версии этапы с установкой кода для входа и биометрической авторизации отсутствуют.

2. Вторичная авторизация:

- Мобильная версия (iOS, Android):
  - Вторичная авторизация осуществляется с помощью ранее установленного кода (PIN) либо с помощью биометрической авторизации.
  - В настройках приложения в разделе «Настройки входа» клиент может:
    - Сменить код (PIN).
    - Включить или отключить биометрическую авторизацию.

- Веб-версия:
  - Вторичная авторизация как отдельный этап отсутствует. После первичной авторизации клиент остается авторизованным в рамках активной сессии, пока не будет полностью закрыта веб-страница с интернет-банком.
  - При обновлении страницы ДБО, если сессия еще активна, клиент автоматически попадает в систему без повторного ввода логина и пароля.

3. Безопасность и соответствие регламентам:

- Система авторизации должна соответствовать требованиям безопасности и регламентам Центрального банка Республики Узбекистан.
- Для хранения и обработки персональных данных клиентов должны быть использованы защищенные каналы связи (SSL/TLS) и методы шифрования.

4. Кроссплатформенная синхронизация:

- Все изменения, связанные с установкой и сменой пароля, кода входа или активацией биометрической авторизации, должны синхронизироваться между мобильной и веб-версиями системы.

1.3 Электронная цифровая подпись (ЭЦП)

Примечание - является обязательным требованием

- Разработка и внедрение виртуальной ЭЦП для подписания документов и транзакций с полной инфраструктурой для Web, iOS, Android.

Функциональные требования:

1. Выпуск ЭЦП для нового клиента через интерфейс ДБО

- После первой авторизации в ДБО, новый клиент должен видеть кнопку «Выпустить ЭЦП» на главной странице или в разделе «Настройки» -> «ЭЦП».
- При нажатии на кнопку «Выпустить ЭЦП», система должна автоматически инициировать процесс выпуска виртуальной ЭЦП без необходимости визита в банк.
- Выпуск ЭЦП должен быть интегрирован с центром сертификации банка для генерации и выдачи уникальной ЭЦП клиенту.
- После выпуска, ПИН-код от ЭЦП отправляется клиенту через SMS на номер телефона, привязанный к его аккаунту.
- Клиент должен быть проинформирован о успешном выпуске ЭЦП с помощью уведомления на экране и SMS-сообщения.

2. Перевыпуск ЭЦП по истечению срока действия

- В разделе «Настройки» -> «ЭЦП» должна быть предусмотрена кнопка «Перевыпустить ЭЦП».
- Когда срок действия текущей ЭЦП истекает (например, через год), клиент может инициировать перевыпуск ЭЦП самостоятельно через интерфейс ДБО без необходимости посещения банка.

- Процесс перевыпуска должен быть полностью автоматизирован, включая интеграцию с центром сертификации для генерации новой ЭЦП.
- Новый ПИН-код для обновленной ЭЦП отправляется клиенту по SMS, аналогично первичной выдаче ЭЦП.

3. Изменение ПИН-кода от ЭЦП

- В разделе «Настройки» -> «ЭЦП» должна быть реализована возможность изменения ПИН-кода от ЭЦП.
- Для изменения ПИН-кода клиент должен ввести старый ПИН-код, а затем указать новый.
- Система должна проверять корректность введенного старого ПИН-кода и применить новый ПИН-код после успешного ввода.
- После успешного изменения ПИН-кода, клиент должен получить подтверждение на экране и уведомление через SMS о том, что ПИН-код был успешно изменен.

4. Платформы

- Все указанные процессы должны быть реализованы на всех трех платформах: Web, iOS, Android.
- Необходимо обеспечить одинаковый пользовательский опыт и доступность функционала выпуска, перевыпуска и изменения ПИН-кода для ЭЦП на всех платформах.

5. Безопасность и уведомления

- Все операции, связанные с ЭЦП (выпуск, перевыпуск, изменение ПИН-кода), должны быть защищены двухфакторной аутентификацией (2FA) через SMS или другой подтвержденный метод.
- Любые действия с ЭЦП должны сопровождаться уведомлениями (SMS, push-уведомления) о статусе операции.

Инфраструктурные требования:
- Поддержка выпуска и перевыпуска ЭЦП через интеграцию с центром сертификации должна быть развернута и поддержана для всех платформ (Web, iOS, Android).
- Все сервисы, связанные с ЭЦП, должны иметь резервные копии для предотвращения потерь данных при сбоях.

1.4 Фронт-часть (UI/UX)

Примечание - является обязательным требованием

- Разработка и развертывание интерфейсов для основных разделов ДБО, таких как: главная, настройки, платежи, счета, история транзакций на Web, iOS, Android с полной поддержкой бэкенда.
- Локализация интерфейса на три языка: русский, узбекский, английский на всех платформах.

Функциональные требования:

1. Разработка и развертывание основных интерфейсов и разделов ДБО

- Разработка и реализация следующих ключевых разделов ДБО на трех платформах (Web, iOS, Android):
  - Главная страница
  - Настройки
  - Платежи
  - Счета
  - История транзакций
- Все разделы должны быть интуитивно понятными, функциональными и соответствовать требованиям UX/UI для банковских систем.

2. Брендирование под стиль банка

- Интерфейсы должны быть выполнены с учетом корпоративного стиля и брендбука банка, включая цветовую палитру, логотипы и шрифты.
- Визуальные элементы интерфейса должны полностью отражать фирменный стиль банка, создавая целостный и узнаваемый пользовательский опыт на всех платформах.

3. Интеграция с существующими операционными процессами

- Дизайн интерфейсов и пользовательского опыта (UX) должен учитывать текущие операционные процессы банка, такие как:
  - Процесс создания и отправки платежей.
  - Процесс подписания платежей с использованием ЭЦП.
  - Логика отображения и подтверждения статусов платежей.
- Все операционные процессы, связанные с взаимодействием пользователя с ДБО (создание, отправка, подписание платежей), должны быть отражены в интерфейсе с максимальной прозрачностью и удобством для клиента.

4. Адаптивность интерфейсов

- Веб-версия ДБО должна быть адаптивной для работы на различных разрешениях экранов.

- Мобильные версии (iOS, Android) должны быть оптимизированы под различные устройства с разными размерами экранов и операционными системами, обеспечивая одинаково качественный пользовательский опыт.

5. Дополнительные требования

1. Фронт-часть должна быть разработана по макетам нашего дизайнера:

- Весь UX/UI должен полностью соответствовать макетам, разработанным командой дизайна банка. Визуальная и функциональная составляющая интерфейсов должна быть реализована в строгом соответствии с этими макетами, чтобы обеспечивать единообразный и согласованный пользовательский опыт на всех платформах.
- Макеты будут включать:
  - Детализированные элементы интерфейса, в том числе кнопки, поля ввода, списки, выпадающие меню и другие UI-элементы.
  - Пользовательские пути, которые дизайнер заложит на основе исследований взаимодействий и опыта пользователей (user flows).
  - Адаптивный дизайн, поддерживающий корректное отображение на разных устройствах и разрешениях.

2. Гибкость и возможность частых изменений:

- Гибкость фронт-части: Интерфейсы фронт-части ДБО должны быть спроектированы таким образом, чтобы они легко адаптировались к изменениям. Это значит, что при необходимости мы, как владельцы продукта, должны иметь возможность изменять дизайн, пользовательские пути и другие элементы интерфейса самостоятельно своей командой разработки.
- Редактирование без сложных доработок: Возможность оперативного изменения элементов UI (например, цвета кнопок, расположения блоков, последовательности действий пользователя) через административные панели или минимальные доработки в коде. Это требование связано с тем, что банк может периодически корректировать интерфейсы под новые задачи, требования законодательства, изменения в продуктах и клиентские пожелания.
- Частые обновления: Фронт-часть должна поддерживать возможность частых релизов с минимальным влиянием на текущую работу пользователей. Это подразумевает наличие механизмов «темизации» или «шаблонов», чтобы внесение изменений было быстрым и безопасным.

В идеале:
Фронт-часть должна обладать высокой модульностью, позволяющей не только гибко менять дизайн, но и изменять логику пользовательских

взаимодействий (например, последовательность шагов при создании платежа или изменение пользовательских сценариев авторизации) без существенных вмешательств в код. Это даст банку возможность подстраиваться под изменяющиеся требования рынка и оперативно реагировать на фидбек пользователей, не нарушая стабильность системы.

Инфраструктурные требования:
- Разработка фронт-части должна включать создание соответствующей инфраструктуры для поддержки всех трех платформ (Web, iOS, Android).
- Все интерфейсы должны быть интегрированы с основными бизнес-процессами банка, обеспечивая их бесперебойную работу и синхронизацию данных между платформами.

1.5 Настройки авторизации

- Реализовать настройки смены пароля, кодов для входа и управления биометрией с развертыванием на всех платформах: Web, iOS, Android.

Функциональные требования по настройкам авторизации должны быть разделены для двух типов платформ: Web и Мобильные приложения (iOS, Android).

Web-платформа

1. Смена пароля учетной записи:

- Клиент заходит в раздел «Настройки» и выбирает пункт «Настройки входа».
- Для смены пароля от учетной записи клиенту нужно выполнить следующие шаги:
  1. Ввести текущий пароль.
  2. Ввести новый пароль дважды.
  3. Подтвердить смену пароля одним из двух способов:
     - Через SMS-код: SMS-код отправляется на номер телефона, привязанный к учетной записи клиента.
     - Через физическую ЭЦП: Подтверждение смены пароля осуществляется путем подписания действия физической ЭЦП, которая хранится на USB-флешке клиента. Клиент вставляет флешку в компьютер и подписывает смену пароля.

2. Аутентификация с использованием физической ЭЦП:

- После смены пароля с использованием физической ЭЦП, система регистрирует изменение пароля, и клиент может использовать новый пароль для последующих входов.
- Вся информация по смене пароля должна логироваться для аудита.

Мобильные приложения (iOS, Android)

1. Смена пароля учетной записи:

- Процесс смены пароля аналогичен веб-версии:
  - Клиент должен ввести старый пароль и дважды новый пароль.
  - Подтверждение смены пароля на мобильной платформе выполняется через виртуальную ЭЦП (выпущенную для мобильного устройства) или через SMS-код.
  - Процесс не требует физической ЭЦП, так как на мобильной платформе используется виртуальная ЭЦП.

2. Смена кода для вторичной авторизации (PIN-код):

- В мобильных приложениях клиент может также сменить код для вторичной авторизации (PIN-код):
  1. Ввести текущий код.
  2. Ввести новый код дважды.
  - Подтверждение изменения кода осуществляется автоматически без необходимости SMS или ЭЦП.

3. Включение/отключение биометрической авторизации:

- Клиент может включить или отключить биометрическую авторизацию (FaceID, TouchID для iOS или аналогичные механизмы для Android) через раздел «Настройки входа».
- Если биометрия включена, клиент может использовать ее для вторичной авторизации в приложении (вместо использования кода).
- Возможность включения и отключения биометрии должна быть доступна только для устройств, поддерживающих эту технологию.

Общие требования

- Веб-версия и мобильные приложения должны синхронизировать изменения паролей и настроек авторизации в режиме реального времени.

- Все действия по смене пароля и кода должны логироваться с указанием времени, идентификационных данных и метода подтверждения (SMS-код, ЭЦП).
- В мобильных версиях ДБО клиент должен использовать виртуальную ЭЦП для подтверждения действий, а на веб-версии — физическую ЭЦП, если она выдана клиенту.

Кросс-платформенность:
- Любые изменения, сделанные на одной платформе (например, смена пароля или кода в мобильном приложении), должны синхронизироваться с другими платформами (веб и мобильные) в реальном времени.

1.6 Ролевая модель пользователей

Примечание - является обязательным требованием

- Полная реализация ролевой модели управления пользователями с возможностью назначения ролей и прав доступа, поддержка бэкенда и фронта на всех платформах.

Функциональные требования:

1. Основные роли:

- Владелец бизнеса/Генеральный директор:
  - По умолчанию имеет полный доступ ко всем модулям и функциональным возможностям ДБО.
  - Может просматривать, создавать, и подписывать все типы платежей и документов.
  - Имеет возможность настраивать права доступа для других пользователей своей компании.

- Бухгалтер с правом подписи:
  - Имеет аналогичные права с владельцем бизнеса. Может просматривать, создавать и подписывать все платежи и документы.

- Бухгалтер без права подписи:
  - Имеет доступ ко всем модулям, кроме подписания платежных поручений.
  - Может просматривать информацию и создавать платежи, но не имеет права их подписывать.

- Индивидуальная роль:
  - Настраивается в зависимости от требований клиента.

- Доступы можно настроить для каждого модуля и микросервиса отдельно.

2. Глобальные уровни прав доступа для ролевой модели:

Каждый модуль и микросервис в ДБО должен поддерживать три уровня прав доступа:
- Первый уровень: Просмотр.
- Второй уровень: Создание.
- Третий уровень: Подписание/Утверждение.

3. Настраиваемость индивидуальных ролей:

- В рамках индивидуальной роли клиент может настроить доступы к каждому модулю ДБО по следующим параметрам:
- Просмотр всех данных: возможность просматривать остатки по счетам, историю операций и другие данные без возможности создания или подписания.
- Создание документов и платежей: можно дать право создавать платежные поручения (сумовые, SWIFT и валютные), но без возможности подписания.
- Подписание/Утверждение: доступ к подписанию конкретных платежей, например, только сумовых платежей или SWIFT переводов, при этом без доступа к подписанию валютных операций.

4. Конструктор прав доступа:

- Система должна обеспечивать гибкость в настройке прав доступа. Владелец бизнеса или администратор сможет самостоятельно настраивать доступы пользователей, комбинируя уровни прав (просмотр, создание, подписание) для каждой функциональной области:
- Например, пользователь может иметь доступ к просмотру и созданию платежей в сумах, но не к их подписанию.
- Можно настроить, что пользователь имеет доступ к подписанию SWIFT-переводов, но не к созданию конвертации валюты.

5. Самостоятельное управление доступами владельцем бизнеса:

- Владелец бизнеса/Генеральный директор должен иметь возможность выдавать и настраивать доступы для других пользователей своей компании через интерфейс настроек в ДБО.
- Владелец бизнеса может создавать новые роли, настраивать их права и менять доступы в любое время.
- Ролевая модель должна учитывать возможность создания и удаления пользователей, а также настройки их прав в зависимости от потребностей компании.

6. Инфраструктурные требования:

- Все вышеуказанные роли и уровни прав доступа должны быть интегрированы во все модули и микросервисы ДБО на платформах Web, iOS, и Android.
- Система должна поддерживать централизованную админ-панель для управления ролями и правами доступа в рамках одной организации.

1.7 Админ-панель

- Разработка админ-панели для управления ролевыми моделями и выдачи доступов с интеграцией с центром сертификации ЭЦП для выпуска и синхронизации с центром сертификации, включая реализацию бэкенда и фронта на Web.

Функциональные требования:

1. Основное назначение:

Админпанель — это дополнительный фронт-интерфейс, предназначенный для сотрудников банка (в первую очередь IT-специалистов), которые занимаются выдачей доступов клиентам, выпуском, перевыпуском и обслуживанием электронной цифровой подписи (ЭЦП).

Админпанель должна предоставлять доступ к следующим ключевым функциям:

2. Основные функции:

2.1. Создание учетной записи клиента в ДБО:

- Форма для ввода персональных данных клиента: сотрудник банка должен иметь возможность вводить следующие данные:
  - Паспортные данные клиента.
  - Данные о компании клиента.
  - Номер телефона клиента.
  - Логин и временный пароль, который клиент сможет позже изменить в ДБО.

- Интеграция с АБС: после ввода данных клиента админпанель должна автоматически подтянуть информацию о компании клиента из АБС и привязать ее к учетной записи в ДБО.

- Система должна корректно отображать все счета и данные, связанные с компанией, чтобы обеспечить правильную привязку прав.

2.2. Выдача прав доступа клиенту:

- Присвоение ролей пользователю: после создания учетной записи сотрудник банка через админпанель должен иметь возможность назначить пользователю одну из ролей согласно ролевой модели:
  - Владелец бизнеса (генеральный директор).
  - Бухгалтер с правом подписи.
  - Бухгалтер без права подписи.
  - Индивидуальная роль (настраивается в зависимости от потребностей клиента).

- Ролевая модель: доступы должны назначаться на уровне всех модулей и микросервисов в ДБО (счета, платежи, отчеты и т.д.). Поддерживаются три уровня прав доступа:
  - Просмотр.
  - Создание.
  - Подписание/Утверждение.

2.3. Управление электронной цифровой подписью (ЭЦП):

- Мобильная ЭЦП: админпанель должна позволять сотруднику банка выпускать и перевыпускать мобильную ЭЦП для клиента. Клиент сможет выпускать ЭЦП самостоятельно на мобильном устройстве через ДБО, но админпанель должна поддерживать администрирование этого процесса.
  - Возможность заблокировать или перевыпустить мобильную ЭЦП.

- ЭЦП на флешке: админпанель должна быть интегрирована с сертификационным (удостоверяющим) центром для выпуска ЭЦП на физическом токене (флешке).
  - Сотрудник банка генерирует уникальный токен, который записывается на флешку клиента.
  - После выпуска ЭЦП админпанель передает информацию о ней в сервисы ДБО, обеспечивая возможность подписания платежей с использованием этой ЭЦП.
  - Возможность перевыпустить устаревшую ЭЦП на флешке.

2.4. Блокировка учетной записи:

- Админпанель должна позволять сотруднику банка заблокировать учетную запись клиента при необходимости, например, в случае подозрений в нарушении безопасности или при обращении клиента с запросом на блокировку.

2.5. Логи действий пользователя:

- В адмінпанели должна быть возможность выгрузки логов действий пользователя:
  - В логе отображаются все ключевые действия, совершенные клиентом в ДБО (создание платежей, отправка документов, изменение настроек).
  - Логи должны содержать информацию о времени действия, используемой учетной записи, устройстве и IP-адресе, с которого было выполнено действие.

3. Инфраструктурные требования:

- Адмінпанель должна быть развернута на Web
- Поддержка полноценной интеграции с АБС и сертификационными центрами.
- Обеспечение безопасности доступа и управление ролями сотрудников банка, использующих адмінпанель.

4. Модель доступов к Адмінпанели сотрудников банка

- Держателем и владельцем системы адмінпанель является департамент БИТ. Этот департамент управляет всеми учетными записями клиентов и их доступами в системе ДБО.

1. Роли в системе администрирования:

В системе администрирования для сотрудников банка будет предусмотрено пять основных ролей:

1. Супер Адмін(главный адмін):
   - Руководитель сотрудников it которые работают в отделениях
2. Адмін:
   - Сотрудник it работающий конкретном в отделений банка
3. Саппорт:
   - Сотрудник технической поддержки - телефон и чат
   - Support engineer - входит в состав команды разработки
4. Транзакционный менеджер:
   - Транзакционные менеджеры в отделениях
5. Просмотр
   - Сотрудники департаментов которые участвуют в обслуживании клиентов b2b

- Любой сотрудник банка которому по разным причинам понадобился доступ к адмнинпанели

5. Процессы, какие и кем из сотрудников банка будут выполняться

Детализация процессов для ролей в адмнинпанели

1. Роль SuperAdmin

- Общий доступ: Полный доступ ко всем функциям адмнинпанели на уровне банка, включая управление доступами сотрудников банка к самой адмнинпанели.
- Основные процессы:
  1. Создание учетной записи клиента: SuperAdmin может создавать учетные записи для всех клиентов банка, вне зависимости от отделения, в котором клиент открыл счет.
  2. Выдача и редактирование доступов к ДБО: Может назначать или редактировать доступы для всех клиентов, включая распределение ролей и прав.
  3. Выпуск и привязка ЭЦП: SuperAdmin имеет право выпускать и привязывать ЭЦП для всех клиентов банка.
  4. Перевыпуск ЭЦП: SuperAdmin может перевыпускать ЭЦП по запросу клиента или в случае необходимости.
  5. Редактирование данных учетной записи: Полный доступ к изменению данных всех клиентов банка.
  6. Выгрузка логов действий пользователя: SuperAdmin может выгружать логи действий всех клиентов для проведения аудитов.
  7. Блокировка учетной записи клиента: Может блокировать учетные записи клиентов в любой ситуации.
  8. Управление доступами сотрудников к адмнинпанели: SuperAdmin управляет доступами других сотрудников банка (например, назначает роли Admin, Support Engineer и Transaction Manager).

2. Роль Admin

- Общий доступ: Имеет доступ ко всем функциям адмнинпанели, но только для клиентов, у которых счета открыты в том отделении, где работает сотрудник it.
- Основные процессы:
  1. Создание учетной записи клиента: Admin создает учетные записи клиентов для своего отделения.

2. Выдача и редактирование доступов к ДБО: Admin управляет доступами для клиентов своего отделения, включая назначение ролей в ДБО.

3. Выпуск и привязка ЭЦП: Выпуск и привязка ЭЦП для клиентов отделения, где работает Admin.

4. Перевыпуск ЭЦП: Admin может перевыпускать ЭЦП для клиентов своего отделения.

5. Редактирование данных учетной записи: Администратор имеет возможность изменять данные учетных записей только по клиентам своего отделения.

6. Выгрузка логов действий пользователя: Admin может выгружать логи действий пользователей в ДБО для своего отделения.

7. Блокировка учетной записи клиента: Admin может блокировать учетные записи клиентов в рамках своего отделения.

3. Support

- Общий доступ: Ограниченный доступ к функциям, связанным с поддержкой пользователей, с правами на просмотр данных всех клиентов банка, вне зависимости от отделения. Инициация процессов доступна только после успешного проведения менеджером технической поддержки процесса идентификации клиента согласно регламенту банка.
- Основные процессы:

1. Редактирование данных учетной записи: Support Engineer может изменять данные учетных записей всех клиентов банка, если требуется вмешательство для решения технических вопросов.

2. Сброс пароля: Имеет право сброса пароля для всех клиентов банка.

3. Блокировка учетной записи клиента: Может инициировать блокировку учетной записи в случае обнаружения технических проблем, по запросу клиента или по запросу государственных органов.

4. Просмотр данных клиентов: Support Engineer имеет полный доступ на просмотр данных всех клиентов, вне зависимости от отделения, где открыт счет клиента.

4. Транзакционный менеджер

- Общий доступ: Доступ только к просмотру данных клиентов по счетам, открытым в его отделении.
- Основные процессы:

1. Просмотр данных клиентов: может просматривать данные учетных записей клиентов только в пределах своего отделения.

2. Просмотр доступов: Может видеть информацию о доступах клиента к системам, но не может их изменять или удалять.

5. Просмотр

- Общий доступ: Доступ только к просмотру данных клиентов

Основные процессы:

1. Просмотр данных клиентов: может просматривать данные учетных записей всех клиентов.
2. Просмотр доступов: Может видеть информацию о доступах клиента к системам, но не может их изменять или удалять.

Резюме:

- SuperAdmin: Полный доступ ко всем функциям админпанели и управлению доступами сотрудников банка.
- Admin: Доступ ко всем функциям админпанели, но только для клиентов своего отделения.
- Support Engineer: Редактирование учетных записей, сброс паролей, блокировка учетных записей и полный доступ на просмотр для всех клиентов.
- Transaction Manager: Доступ только на просмотр данных клиентов в рамках своего отделения.

1.8 Чат с техподдержкой

- Встроить чат для поддержки пользователей с поддержкой на всех платформах: Web, iOS, Android. - интеграция с вендором чата который выбран

1.8. Чат с техподдержкой

Функциональные требования:

1. Основное назначение:

Чат с технической поддержкой должен предоставить пользователям возможность оперативного взаимодействия с техподдержкой банка по всем вопросам, связанным с использованием ДБО. Этот чат является важным элементом клиентского сервиса, позволяя пользователям быстро решать возникающие проблемы.

2. Основные функции:

2.1. Пользовательский интерфейс чата:

- Кастомизация интерфейса чата: интерфейс чата должен быть разработан и адаптирован в соответствии с визуальными требованиями банка. Чат должен полностью соответствовать фирменному стилю, включая цвета, шрифты, иконки и элементы навигации.
  - Пользователь должен легко находить и запускать чат из главного меню или раздела техподдержки в ДБО.

- Уведомления: реализовать пуш-уведомления и визуальные оповещения в интерфейсе, которые информируют пользователя о новых сообщениях в чате или о ответах от техподдержки.

2.2. Интеграция с текущим вендором чата:

- Интеграция с существующей системой чата: чат должен быть интегрирован с системой, которую уже использует банк для общения с клиентами. В случае использования стороннего вендора для чата, необходимо предусмотреть API для взаимодействия с этой системой.
  - Интеграция должна обеспечить возможность работы с существующими каналами поддержки банка, в том числе через внешних вендоров, если это требуется.

- Функциональные возможности интеграции:
  - Система должна поддерживать возможность передачи файлов, скриншотов и документов в рамках общения с поддержкой.
  - Поддержка автоматической маршрутизации запроса к нужному специалисту на основе темы вопроса (настройка платежей, вопросы безопасности и т.д.).
  - Чат должен сохранять историю переписки, доступную как пользователю, так и сотруднику техподдержки для быстрого доступа к ранее обсужденным вопросам.

2.3. Администрирование чата:

- Интеграция с CRM-системой банка: информация о чате и взаимодействиях пользователей с техподдержкой должна быть связана с учетными записями клиентов в CRM банка, чтобы сотрудники техподдержки могли иметь доступ к полным данным о клиенте и его предыдущих запросах.

- Системные сообщения и автоматические ответы: реализовать возможность отправки системных сообщений пользователю при поступлении его запроса, а также возможность использования шаблонных ответов.

3. Инфраструктурные требования:

- Чат должен быть доступен на всех платформах (Web, iOS, Android) с одинаковым функционалом.
- Поддержка защищенного обмена данными между пользователями и техподдержкой, включая шифрование сообщений.
- Возможность интеграции с существующими сервисами банка, включая базы знаний и базы частых вопросов для автоматического ответа на типовые запросы.

1.9 Клиентское хранилище данных

- Реализовать клиентское хранилище данных с доступом через интерфейсы на Web, iOS и Android. -

Функциональные требования:

1. Основное назначение:

Клиентское хранилище данных — это централизованная система хранения информации, связанной с действиями учетных записей в ДБО и документами, созданными или переданными в банк клиентом. Оно служит для хранения всех важных клиентских данных, которые могут потребоваться для аудита, выполнения регуляторных требований, а также для оперативного доступа клиента к своим документам и выпискам.

2. Основные функции:

2.1. Логирование действий учетных записей:

- В хранилище данных должна сохраняться информация о всех действиях учетных записей в ДБО, касающаяся использования каждого сервиса.
  - Логирование включает создание, изменение, отправку и подписание платежей, использование функций ДБО, смену настроек, и прочие важные действия.
  - Логи сохраняются для проведения аудита, возможных проверок со стороны государственных органов, прокуратуры и суда.
  - Логирование должно быть доступно для анализа и выгрузки при запросе государственных органов.

2.2. Хранение документов:

- Документы, созданные клиентом: все документы, выписки, справки и другие документы, сгенерированные по запросу клиента через ДБО, должны храниться в клиентском хранилище данных.
  - Например, когда клиент формирует выписку по счету, она создается в формате PDF, и этот PDF-документ должен быть автоматически сохранен в хранилище для дальнейшего использования.
  - Документы должны быть связаны с учетной записью клиента и иметь возможность быстрого поиска и выгрузки в интерфейсе.

- Документы, отправленные клиентом в банк: все документы, которые клиент передает в банк через ДБО, такие как платежные поручения, заявки, платежные требования и другие файлы, должны сохраняться в хранилище.
  - Клиент должен иметь возможность получить доступ к этим документам через интерфейс ДБО, а также просмотреть и скачать их.

2.3. Хранение финансовых данных:

- В хранилище данных должна храниться история всех транзакций клиента, включая детали транзакций (суммы, реквизиты, статусы) и доступы к ним через интерфейс ДБО.
  - Остатки по счетам клиента, включая все типы счетов (расчетные, валютные, карты), также должны быть доступны для хранения и просмотра через интерфейс.

3. Инфраструктурные и технические требования:

3.1. Безопасность данных:

- Должна быть обеспечена полная защита клиентских данных, хранящихся в хранилище, с использованием современных стандартов шифрования.
  - Доступ к данным в хранилище должен быть ограничен и предоставляться только аутентифицированным пользователям с соответствующими правами.

- Система должна поддерживать резервное копирование и восстановление данных на случай сбоев или потерь.

3.2. Логирование и аудиторская система:

- В хранилище должна быть предусмотрена система поиска и доступа к логам действий пользователей и транзакций для проведения внутренних и внешних аудитов.

- Администраторы системы должны иметь возможность выгружать логированные данные по запросу государственных органов или внутреннего контроля.

3.3. Доступ и интеграции:

- Хранилище должно быть доступно на внутренней Web платформе для доступа сотрудников банка к документам, логам и финансовым данным.
- Хранилище данных должно быть интегрировано с АБС банка и другими системами для обеспечения синхронизации данных в реальном времени.

1.10 Генератор PDF документов

- Разработка генератора документов PDF

Функциональные требования:

1. Основное назначение:

Генератор PDF-документов является важной частью бэкенд-архитектуры ДБО и обеспечивает создание и обработку различных документов в формате PDF, включая кастомные документы, выписки, справки, и заявления, с наложением электронной цифровой подписи (ЭЦП) банка и клиента.

2. Основные функции:

2.1. Создание кастомных PDF-документов:

- Генератор должен поддерживать создание кастомных документов по запросу клиента или банка. Примеры таких документов:
  - Заявления на открытие депозита.
  - Заявления на выдачу кредита.
  - Заявления на открытие нового счета и другие типы документов.

- Клиент должен иметь возможность сгенерировать такие документы через интерфейс ДБО, заполнить их, подписать своей ЭЦП и скачать итоговый документ.

2.2. Генерация документов банка с подписью банка:

- Генератор PDF-документов должен обеспечивать генерацию стандартных банковских документов, таких как выписки по счетам, справки, документы о состоянии счетов и другие виды отчетов.
  - На все такие документы должна быть наложена электронная цифровая подпись банка для обеспечения их юридической силы.
  - Клиенты должны иметь возможность скачивать эти документы с подписью через интерфейс ДБО (Web, iOS, Android).

- Для каждого сгенерированного документа должна быть возможность его сохранения в клиентском хранилище данных, о котором говорилось в пункте 1.9.

2.3. Генерация подписанных клиентом PDF-документов:

- После того, как клиент подписал документ (например, заявление на открытие депозита или кредита) своей ЭЦП, система должна:
  - Наложить подпись клиента на этот документ.
  - Предоставить клиенту возможность скачать документ с его подписью в формате PDF.

- Подпись должна быть юридически валидна и соответствовать стандартам цифровой подписи, признанным регуляторами.

3. Инфраструктурные и технические требования:

3.1. Гибкость настройки генератора документов:

- Функционал генератора должен быть гибким и настраиваемым, чтобы позволить разработчикам оперативно добавлять новые типы документов и изменять уже существующие шаблоны.
  - В идеале, для этого может быть реализован веб-интерфейс, через который администраторы или разработчики смогут управлять шаблонами документов.
  - Если веб-интерфейс не предусмотрен, настройка может производиться напрямую через базу данных, с возможностью вносить изменения в шаблоны документов.

3.2. Поддержка ЭЦП и интеграция с сертификационным центром:

- Генератор документов должен быть интегрирован с системой ЭЦП банка и клиентов для автоматического наложения цифровых подписей на PDF-документы.

- В случае генерации документов для подписания клиентом, система должна передавать документ на подписание, после чего возвращать подписанный PDF.

3.3. Мультиплатформенность:

- Генератор должен работать на всех платформах (Web, iOS, Android), обеспечивая пользователям одинаковые возможности по созданию, скачиванию и подписанию документов на любой платформе.

1.10 Взаимодействие ДБО и АБС

Примечание - является обязательным требованием

1. Общие принципы взаимодействия ДБО и АБС

- Требование: АБС не должна быть мастер-системой для ДБО. Все бизнес-логика и процессы должны быть реализованы на стороне ДБО. АБС выполняет функции бухгалтерской книги, ограничиваясь созданием и проведением проводок, а также хранением информации о клиенте.
- Цель: Сокращение зависимости от АБС и обеспечение большей гибкости в управлении процессами.

2. Модуль расчётно-кассового обслуживания (РКО)

- Требование:
  - Сущность платёжных поручений должна существовать в ДБО.
  - Валидация, создание, подписание платёжных поручений в национальной валюте (UZS) должны выполняться ДБО.
  - АБС получает команду на создание и проведение проводки только после успешной обработки платёжного поручения в ДБО.
- Цель:
  - Повышение автономности ДБО в процессе обработки национальных платежей.
  - Ускорение обработки транзакций и уменьшение нагрузки на АБС.

3. Тарифный модуль

- Требование:
  - Логика расчёта тарифов, клиринга и управления тарифами должна быть реализована в ДБО.
  - Тарифы хранятся и обрабатываются внутри ДБО.
  - АБС получает только команды на списание абонентской платы и комиссий, рассчитанных в ДБО.

- АБС не участвует в расчётах, вся логика расчётов выполняется модулем ДБО.
- Цель:
  - Централизация расчёта тарифов внутри ДБО.
  - Упрощение взаимодействия между системами и сокращение ошибок.

4. Модуль внешнеэкономической деятельности (ВЭД)

- Требование:
  - Сущности и процессы для всех видов платёжных поручений в иностранной валюте (SWIFT-платежи, покупка/продажа валюты, конвертация) должны быть реализованы в ДБО.
  - Валидация, подписание и интеграции с необходимыми внешними системами должны обрабатываться на стороне ДБО.
  - АБС получает только команды на создание и проведение проводок, прошедших проверку и обработку в ДБО.
- Цель:
  - Повышение эффективности и автономности обработки международных платежей.
  - Снижение операционных рисков и зависимости от АБС.

5. Модуль зарплатного проекта

- Требование:
  - Сущности зарплатных ведомостей и списков сотрудников должны храниться и обрабатываться в ДБО независимо от АБС.
  - Модуль должен включать собственную бизнес-логику и процессы расчёта зарплат.
  - Интеграция с процессингами (Uzcard, Humo, Visa) осуществляется напрямую из ДБО.
  - ДБО самостоятельно рассчитывает и отправляет команды в АБС для распределения денежных средств на карты сотрудников.
- Цель:
  - Централизация процессов зарплатных проектов в ДБО.
  - Упрощение интеграций с процессингами и сокращение времени обработки.

6. Модуль корпоративных карт

- Требование:
  - Сущность корпоративных карт и их логика должна быть реализована в ДБО.
  - Модуль должен содержать бизнес-процессы клиринга и интеграции с процессингами (Uzcard, Humo, Visa).

- Возможность подключения внешних систем (например, банкоматов) через ДБО.
- В АБС отправляются только проверенные и подписанные проводки для их проведения.
- Цель:
  - Упрощение процессов управления корпоративными картами.
  - Повышение гибкости и адаптивности интеграций.

Эти требования направлены на повышение автономности ДБО, минимизацию зависимости от АБС, ускорение бизнес-процессов и повышение клиентского опыта.

2. Транзакционный сервис для сумовых платежей

2.1 Подписание платежей ЭЦП

Примечание - является обязательным требованием

- Реализация сервиса подписания платежных поручений с использованием ЭЦП с поддержкой бэкенда и фронта на Web, iOS и Android.

Функциональные требования к подписанию платежей ЭЦП можно разделить на несколько направлений:

1. Создание и развертывание статусной модели платежных поручений:

- Платежные поручения должны иметь четкую статусную модель, соответствующую как внутренним регламентам банка, так и требованиям Центрального Банка Республики Узбекистан.
- Статусы платежей должны синхронизироваться с АБС и учитывать следующие состояния:
  - Создан: Платеж создан, но не подписан.
  - Ожидает подписи: Платеж находится в ожидании подписи.
  - Подписан: Платеж подписан клиентом и готов к отправке в банк.
  - Отправлен в банк: Платеж успешно отправлен в банк.
  - Ошибка подписания: Возникли ошибки при подписании (неверный пин-код, неверные данные).
  - Исполнен: Платеж успешно обработан банком.
  - Отклонен: Платеж отклонен банком или клиентом.

2. Создание сущности платежных поручений:

- Необходимо разработать сущность платежных поручений в системе, которая будет включать:
  - Платежные поручения, которые созданы, но не подписаны.
  - Платежные поручения, которые подписаны, но еще не отправлены в банк.
  - Поддержка всех типов платежных поручений (сумовые, валютные, SWIFT и т.д.).
- Эти сущности должны храниться на сервере до тех пор, пока платеж не пройдет все этапы подписания и отправки в банк.

3. Хранение информации о подписаниях и валидации:

- Для каждого подписанного платежного поручения должна храниться полная информация о подписании, включая:
  - Данные об ЭЦП клиента (включая информацию о сертификате).
  - Дата и время подписания.
  - Данные о проверке ЭЦП (валидность, соответствие сертификата и пр.).
  - Валидация правильности пин-кода и данных клиента перед подписанием.
- В случае ошибок подписания (например, неверный пин-код или неверные данные в платежном поручении), система должна:
  - Выводить соответствующие уведомления для клиента.
  - Логировать информацию об ошибке и предлагать клиенту повторить операцию.
  - Информировать клиента о необходимости повторной проверки данных.
  - Предоставить возможность исправить ошибки перед повторной попыткой подписания.

4. Реализация фронтенд-функционала:

- Фронтенд должен предоставлять клиенту следующие возможности:
  - Просмотр статуса платежного поручения на всех трех платформах (Web, iOS, Android).
  - После успешного подписания платежа клиент должен видеть саксэс-экран с подтверждением, что платеж подписан и отправлен в банк.
  - В случае ошибки подписания система должна отобразить клиенту подробную информацию об ошибке и предложить способы её устранения.
  - Возможность повторного подписания при возникновении ошибки.
  - Вся информация и статусы должны быть синхронизированы между платформами в режиме реального времени.

5. Кросс-платформенность:

- Данный функционал должен быть доступен и одинаково работать на всех платформах: Web, iOS, Android.

- В случае успешного или неуспешного подписания на одной платформе, соответствующие данные и статусы должны быть моментально синхронизированы на других платформах, чтобы клиент мог видеть актуальные данные вне зависимости от устройства.

6. Логирование и отчетность:

- Весь процесс подписания должен логироваться для последующего аудита. В логах должны храниться данные о:
  - Времени подписания.
  - Действиях клиента.
  - Ошибках, возникших при подписании.
  - Истории изменений платежного поручения до и после подписания.

Итог:
Эти требования должны обеспечить полную прозрачность процесса подписания платежей с использованием ЭЦП, а также обеспечить надежность и синхронизацию данных между всеми платформами.

2.2 Платежные поручения в сумах

Примечание - является обязательным требованием

- Создание и отправка платежных поручений на счет, карту, в казначейство, в бюджет с поддержкой Анор 24/7 с бэкендом и фронтом на всех платформах.

Функциональные требования:

1. Основное назначение:

Платёжные поручения в сумах должны обеспечить возможность выполнения всех типов транзакций в национальной валюте Республики Узбекистан. Этот функционал должен быть разработан в полном соответствии с регламентами и требованиями Центрального Банка Республики Узбекистан, а также учитывать специфические бухгалтерские модели и требования банка.

2. Основные функции:

2.1. Соответствие регламентам Центрального Банка:

- Все платёжные поручения должны быть разработаны с учётом требований Центрального Банка Республики Узбекистан.
  - Учёт обязательных реквизитов, таких как наименование клиента, ИНН, МФО банка, расчётные счета, назначение платежа, суммы платежей и другие обязательные параметры.
  - Поддержка форматов данных, регламентируемых Центральным Банком.
  - Обработка дополнительных требований по оформлению платёжных поручений, таких как поддержка различных валютных операций и соблюдение законодательных ограничений.

2.2. Учёт бухгалтерских моделей АБС банка:

- Платёжные поручения должны быть интегрированы с автоматизированной банковской системой (АБС), чтобы учитывать бухгалтерские модели и особенности банка.
  - Система должна подтягивать необходимые данные из АБС, такие как реквизиты счетов и статус клиента.
  - Поля платёжного поручения должны корректно заполняться и проверяться с учётом требований АБС и её бухгалтерских моделей.

2.3. Обязательные и необязательные поля:

- В зависимости от типа платёжного поручения (перевод на счет, на карту, в казначейство, в бюджет), система должна:
  - Предоставлять возможность заполнения обязательных полей, предусмотренных для каждого типа платежа.
  - Отображать необязательные поля в зависимости от специфики платёжного поручения.
  - Обеспечить валидацию данных на этапе заполнения, чтобы предотвратить возможные ошибки, такие как неверные реквизиты или несуществующие счета.

2.4. Поддержка системы быстрых платежей АНОР 24/7:

- Платёжные поручения должны иметь возможность отправки как по стандартному процессу внутрибанка, так и через систему быстрых платежей Центрального Банка АНОР 24/7.
  - Стандартный процесс: передача платежей через обычный банковский маршрут с соблюдением графиков и регламентов по обработке платёжных поручений.

- АНОР 24/7: система должна поддерживать отправку платежей в режиме 24/7 через систему АНОР, обеспечивая моментальное проведение транзакций в рамках данной системы.
- Интеграция с системой АНОР должна включать проверку статусов платежей и их обработки, а также корректную обработку возвратов и ошибок, связанных с отправкой.

3. Мультиплатформенность:

- Платёжные поручения должны быть доступны на всех платформах (Web, iOS, Android), обеспечивая возможность создания, отправки и отслеживания всех типов платежей через любые доступные устройства.

4. Бизнес логика

- Функционал ПП в сумах должен быть реализован как независимый от АБС сервис и иметь свою бизнес логику.
-Сервис должен позволять вносить в него изменения.
-Сервис должен самостоятельно осуществлять все валидации и отправлять в АБС только платежи которые успешно прошли валидацию.
-Сервис должен иметь свое открытое API и обеспечивать интеграцию с ним любых внутрибанковских систем.

2.3 Массовая загрузка платежей

- Внедрение массовой загрузки и подписания платежей с полной поддержкой бэкенда и фронта для Web, iOS и Android.

Функциональные требования:

1. Основное назначение:

Массовая загрузка и подписание платежей предназначены для того, чтобы ускорить и упростить процесс создания и отправки большого количества платёжных поручений, минимизируя ручные действия. Это включает в себя возможность массовой загрузки платежей через шаблон Excel и массового подписания платежных поручений с помощью ЭЦП.

2. Массовая загрузка платежей:

Функциональность массовой загрузки можно разделить на несколько шагов, обеспечивающих удобство и гибкость для пользователей ДБО при работе с множественными платежами.

2.1. Шаблон для массовой загрузки:

- Для удобства клиентов должен быть предоставлен шаблон в формате Excel, который содержит все необходимые поля для платёжных поручений. Поля должны соответствовать стандартным обязательным и необязательным реквизитам платёжных поручений (таким как наименование плательщика и получателя, ИНН, МФО, расчётный счёт, сумма платежа, назначение платежа и т.д.).
- Пользователь должен иметь возможность скачать этот шаблон, заполнить его необходимыми данными и затем загрузить обратно в систему.

2.2. Парсинг данных и автоматическое создание платежных поручений:

- После загрузки файла с заполненными платёжными данными, система должна автоматически спарсить данные из шаблона Excel.
  - Система должна распознавать и обрабатывать все поля, как обязательные, так и необязательные.
  - На основе данных, которые ввёл пользователь, система должна автоматически создать платёжные поручения.
  - Поддерживаются все виды платежей в сумах: на счёт, на карту, в казначейство, в бюджет.
  - Система должна валидировать данные перед созданием платёжных поручений, проверяя корректность введённых данных, таких как номера расчётных счетов, суммы и прочие обязательные поля.
  - В случае ошибок или некорректных данных, система должна уведомлять пользователя и указывать на ошибочные поля для исправления.

3. Массовое подписание платежей:

Функциональность массового подписания предназначена для того, чтобы клиенты могли подписывать несколько платёжных поручений одновременно, что значительно ускоряет процесс обработки больших объёмов транзакций.

3.1. Подписание всех созданных, но неподписанных платежей:

- После создания платёжных поручений клиент должен иметь возможность выбрать несколько платёжных поручений или выбрать все платежи, которые находятся в статусе "Создан, но не подписан".
- Система должна предусматривать возможность массового подписания всех выбранных платежей одной операцией. Пользователь вводит ПИН-код от ЭЦП один раз для подтверждения всех платежей, после чего система подписывает и отправляет их в банк.
  - Должен быть предусмотрен механизм уведомления пользователя о статусе каждой транзакции после подписания и отправки.

4. Кроссплатформенность:

- Массовая загрузка и массовое подписание платежей должны работать кроссплатформенно, обеспечивая синхронизацию данных между Web, iOS и Android версиями ДБО.
  - Например, пользователь может загрузить платёжные поручения через веб-версию, а другой пользователь с соответствующими правами (например, директор) должен иметь возможность увидеть эти платёжные поручения на мобильной платформе и подписать их.
  - Подписанные платежи на одной платформе должны мгновенно обновляться на других платформах.

2.4 Платежи через Мунис

Примечание - является обязательным требованием

- Реализация возможности создания и обработки платежей через Мунис с развертыванием на всех платформах.

Функциональные требования:

1. Основное назначение:

Платежи через систему Munis должны быть реализованы с полной поддержкой всех типов платёжных поручений, предоставляемых системой Munis. Данная функциональность должна автоматизировать процесс создания платёжных поручений на основе инвойсов или счетов на оплату,

которые есть у клиента, и обеспечивать лёгкость и скорость обработки таких платежей.

2. Автоматическое создание и заполнение платёжного поручения через Munis:

- Интеграция с системой Munis должна позволить автоматическое создание и заполнение платёжных поручений на основе данных, доступных в системе Munis.
  - Пользователь вводит номер инвойса или счёта на оплату, который был выдан через систему Munis.
  - ДБО должно обратиться к системе Munis через API и получить все необходимые данные для формирования платежного поручения, включая реквизиты получателя, тип платежа, сумму и другие обязательные данные.
  - Платежное поручение должно быть автоматически создано и заполнено в ДБО на основе этих данных.

3. Возможность редактирования платёжного поручения:

- Клиент должен иметь возможность просмотреть и, при необходимости, редактировать платёжное поручение перед его отправкой(подписанием).
  - Клиент может изменить сумму платежа, детали платежа и добавить или изменить комментарии, или скорректировать другие необязательные поля если они присутствуют в типе платежного поручения которое заполняет клиент.
  - Однако основные реквизиты, такие как реквизиты получателя, тип платежа, должны подтягиваться из системы Munis и оставаться корректными в соответствии с системой Munis.

4. Подписание и отправка платёжного поручения:

- После автоматического заполнения платёжного поручения, клиент должен иметь возможность его сразу подписать с помощью ЭЦП, после чего платёж будет отправлен.
  - Для клиента, который не желает вносить изменения в платёжное поручение, процесс должен быть максимально упрощён: после создания платежа остаётся только его подписать и отправить.
  - Также клиент может сохранить созданное платёжное поручение и вернуться к его редактированию или подписанию позже.

5. Поддержка всех типов платёжных поручений через Munis:

- Должна быть реализована поддержка всех возможных типов платёжных поручений, которые предоставляет система Munis

6. Кроссплатформенность:

- Платежи, созданные через систему Munis, должны быть доступны на всех платформах: Web, iOS, и Android.
  - Клиент может создать платёжное поручение через Munis на одной платформе (например, веб), а подписать его на другой платформе (например, мобильной версии).
  - Все платежи, созданные через систему Munis, должны синхронизироваться между всеми платформами.

2.5. Синхронизация данных между платформами

Примечание - является обязательным требованием

Функциональные требования:

1. Основное назначение:

Синхронизация данных между платформами (Web, iOS, Android) должна происходить оперативно и, по возможности, моментально. Скорость синхронизации является ключевым аспектом, влияющим на удобство работы клиента с ДБО, независимо от используемого устройства.

2. Оперативная синхронизация:

- Моментальная синхронизация данных между всеми платформами (Web, iOS, Android) должна быть обязательной для всех изменений, внесённых клиентом. Время синхронизации должно составлять минимальное количество секунд, чтобы клиент мог seamlessly переходить между платформами без задержек.

3. Синхронизация всех клиентских данных:

- Помимо синхронизации платёжных поручений, синхронизации должны подлежать следующие данные:
  - Остатки по расчётным счетам — актуальные данные по балансу на счетах клиента должны моментально обновляться и отображаться на всех платформах.
  - Выписки по счетам — если клиент заказал выписку на одной платформе (например, на iOS), она должна быть автоматически доступна на других платформах (Android и Web).
  - Информация о перевыпуске ЭЦП — если клиент перевыпустил ЭЦП на мобильном устройстве, информация об этом перевыпуске должна быть доступна на всех платформах (включая уведомление о перевыпуске).
  - Выдача прав доступа к учетным записям — если владелец бизнеса выдал права доступа новому пользователю, эта информация должна моментально отразиться на всех платформах.

4. Полная синхронизация всех действий:

- Все действия клиента, которые влекут за собой изменение данных или создание отметки в системе (например, выпуск выписки, изменение настроек авторизации, обновление профиля), должны автоматически и моментально отображаться на всех платформах.
  - Например, если клиент заказал справку через Web, она должна быть доступна для скачивания и на мобильных устройствах (iOS и Android).
  - Если клиент изменил пин-код для ЭЦП на iOS, эта информация должна быть сразу доступна на других платформах.

5. Интерфейс и действия:

- Вся информация, которая влияет на интерфейс и отображение действий клиента (например, создание платёжного поручения, выдача доступа к учётной записи, запрос выписки или справки), должна моментально синхронизироваться между платформами.
  - Интерфейс должен автоматически обновляться, и клиент должен видеть изменения в реальном времени на всех устройствах, без необходимости вручную обновлять данные.

6. Технологии для синхронизации:

- Для реализации синхронизации необходимо использовать технологии и подходы, обеспечивающие минимальную задержку в обновлении данных на всех платформах.
  - Возможно использование push-уведомлений и веб-сокетов для мгновенного оповещения клиента о всех изменениях данных на других устройствах.

2.6 Зарплатная ведомость

Примечание - является обязательным требованием

- Внедрение функционала для формирования и отправки зарплатных ведомостей с поддержкой на всех платформах.

Функциональные требования:

1. Разделение на две сущности:

- Сущность 1: Зарплатная ведомость
- Сущность 2: Сотрудники

2. Раздел «Сотрудники»

Функционал раздела «Сотрудники»:
- Отображение списка всех сотрудников в рамках зарплатного проекта. Пользователь с правами на подпись (генеральный директор, владелец бизнеса, бухгалтер с правом подписи) имеет доступ к этому разделу.

Действия с сотрудниками:

1. Добавление нового сотрудника:

  - Возможность добавить сотрудника с действующей картой физического лица Хамкор Банка.
  - Поля для ввода при добавлении сотрудника: ФИО, паспортные данные, номер телефона и номер карты.
  - При успешном добавлении сотрудник автоматически добавляется в зарплатный проект.

2. Удаление сотрудника:

- Удаление сотрудника нажатием одной кнопки «Удалить». Сотрудник удаляется из списка активных сотрудников, но остается в истории зарплатных ведомостей, в которых фигурировал ранее.

3. Редактирование данных сотрудника:

- Возможность зайти в карточку сотрудника и изменить данные (имя, номер телефона, номер карты). Все изменения производятся онлайн, и они сразу вступают в силу.

Интеграция с 1С:

- Возможность выгружать список сотрудников из 1С напрямую в ДБО через API. Все сотрудники из 1С автоматически добавляются в зарплатный проект.

3. Раздел «Зарплатная ведомость»

Функционал раздела «Зарплатная ведомость»:

1. Загрузка зарплатной ведомости из 1С:

- Зарплатная ведомость создается на основе данных, выгруженных из 1С по API. Данные включают список сотрудников и суммы выплат. Клиенту остается выбрать счет для списания и подписать ведомость.

2. Массовая загрузка сотрудников в ведомость через Excel:

- Клиент может выгрузить шаблон в формате Excel, где обязательные поля включают:
  - ФИО сотрудника.
  - Номер счета или карты сотрудника.
  - Сумма выплаты.
- Клиент заполняет шаблон, загружает его в ДБО, и система автоматически формирует ведомость для подписания.

3. Ручное создание зарплатной ведомости:

- Клиент может вручную создать ведомость, выбрав счет списания. После выбора счета отображается список сотрудников с полями для ввода суммы заработной платы. Клиент вводит суммы вручную и подписывает ведомость. После подписания ведомость отправляется в банк на исполнение.

4. Общие требования:

- Интерфейс:
  - Разделы «Сотрудники» и «Зарплатная ведомость» должны быть доступны на всех платформах (Web, iOS, Android).
  - Интерфейс должен быть интуитивно понятным, поддерживать массовую загрузку и редактирование данных.

- Синхронизация:
  - Вся информация должна синхронизироваться между платформами Web, iOS и Android моментально.

- Подписание:
  - Все ведомости должны быть подписаны с использованием ЭЦП, с возможностью массового подписания ведомостей.

- Безопасность:
  - Все действия по добавлению, удалению и редактированию сотрудников, а также формированию и подписанию зарплатных ведомостей должны логироваться для аудита.

5. Бизнес логика

- Функционал должен быть реализован как независимый от АБС сервис и иметь свою бизнес логику.
-Сервис должен позволять вносить в него изменения.
-Сервис должен самостоятельно осуществлять все валидации и отправлять в АБС только ведомости которые успешно прошли валидацию.
-Сервис должен уметь самостоятельно ходить в процессинг для получения актуальных статусов по зарплатным картам.
-Сервис должен иметь свое открытое API и обеспечивать интеграцию с ним любых внутрибанковских систем.

2.7 Шаблоны платежей

Примечание - является обязательным требованием

- Возможность создания и использования шаблонов для повторяющихся платежей с поддержкой на всех платформах.

Функциональные требования:

1. Раздел «Шаблоны» на всех платформах:

- На всех платформах (Web, iOS, Android) в ДБО должен быть раздел «Шаблоны», где отображаются все шаблоны платежных поручений, созданные клиентом.

2. Действия с шаблонами:

1. Создание нового шаблона:

- Клиент может создать новый шаблон платежного поручения. Для этого отображается стандартная форма с типами платежных поручений (в суммах и в валюте):
  - На счет
  - На карту
  - В казначейство
  - В бюджет
  - SWIFT-перевод
  - Покупка валюты
  - Продажа валюты
  - Конвертация валюты
- Клиент заполняет выбранный тип платежного поручения по тому же сценарию, как и обычное платежное поручение, и нажимает кнопку «Создать шаблон».

2. Редактирование существующего шаблона:

- Клиент может открыть существующий шаблон, внести изменения и сохранить обновленный шаблон.

3. Удаление шаблона:

- Клиент может удалить любой шаблон платежа из списка.

3. Дополнительные возможности:

1. Создание шаблона на экране успеха (Success-экран):

- После подписания платежа клиенту должен отображаться Success-экран, где будет кнопка «Создать шаблон из этого платежа». Клиент может мгновенно создать шаблон на основе уже заполненного и подписанного платежного поручения.

2. Создание платежа из шаблона:

- После создания или выбора существующего шаблона у клиента должна быть возможность сразу создать новый платеж на основе этого шаблона, нажав кнопку «Создать платеж из этого шаблона».

4. Синхронизация между платформами:

- Все действия с шаблонами (создание, редактирование, удаление) должны синхронизироваться между Web, iOS и Android. Изменения, внесенные на одной платформе, должны моментально отображаться на остальных.

5. Безопасность и аудирование:

- Все действия, связанные с шаблонами, должны быть зафиксированы в логах для аудита и сохранены в клиентском хранилище данных.

2.8. Повтор проведенных платежей

- Реализовать возможность повтора платежей с поддержкой на Web, iOS, Android.

Функциональные требования:

1. Доступные точки для повтора платежей:

1. История транзакций:

- Клиент может повторить проведенный платеж, зайдя в историю транзакций и перейдя в детали конкретного платежа. В деталях платежа должна быть кнопка «Повторить платеж».

2. Success-экран после подписания платежа:

- После того, как клиент подписал платеж, на Success-экране должна отображаться кнопка «Повторить платеж» вместе с кнопкой «Создать шаблон из этого платежа».

2. Процесс повтора платежа:

- Когда клиент нажимает кнопку «Повторить платеж» в любой из доступных точек (история транзакций или Success-экран), ему открывается форма платежного поручения.

- Автоматическое заполнение формы: Все поля в форме платежного поручения заполняются автоматически на основе данных из платежа, который клиент хочет повторить.

3. Возможность редактирования:

- Клиент должен иметь возможность редактировать все поля, которые функционально доступны для редактирования в данном типе платежного поручения (например, сумма, дата, назначение платежа и т.д.).
- Поля, которые не подлежат редактированию, должны быть зафиксированы в исходном состоянии.

4. Подписание и отправка платежа:

- После внесения всех изменений клиент должен иметь возможность подписать платеж с помощью ЭЦП и отправить его в банк так же, как при создании нового платежного поручения.

5. Синхронизация и кроссплатформенность:

- Все действия по повтору платежа должны синхронизироваться между Web, iOS и Android платформами, и изменения должны отображаться на всех устройствах пользователя.

6. Безопасность и аудит:

- Процессы повтора платежей должны быть защищены, и все действия пользователя фиксируются в логах для аудита и отслеживания.

2.9 Счета и балансы

Примечание - является обязательным требованием

- Отображение всех доступных счетов и балансов в реальном времени с полной поддержкой бэкенда и фронта на всех платформах.

Функциональные требования:

1. Раздел «Счета»:

- В новом ДБО должен быть реализован отдельный раздел «Счета», в котором отображаются все счета клиента.
- Отображение счетов клиента должно происходить в соответствии с внутренними регламентами банка по отображению счетов и с учетом прав на просмотр счетов, установленных в ролевой модели пользователя.

2. Общая страница списка счетов:

- На общей странице должен быть представлен список всех доступных счетов клиента с отображением следующих параметров:
  - Баланс по каждому счету.
  - Валюта счета.
  - Ограниченный номер счета (например, последние пять цифр).

3. Детали счета:

- При переходе в детали конкретного счета, клиент должен видеть:
  - Историю транзакций по выбранному счету с фильтрацией по дате и типу транзакций.
  - Возможность просматривать детализированные данные по каждой транзакции.

4. Дополнительные функции:

- В деталях счета должны быть доступны следующие функциональные возможности:
  - Копировать реквизиты: при нажатии кнопки «Копировать реквизиты», полные реквизиты счета копируются в буфер обмена на устройстве клиента.
  - Создать выписку: возможность сгенерировать и скачать выписку по выбранному счету в формате PDF с печатью банка.
  - Создать платежное поручение: возможность инициировать создание платежного поручения с выбранного счета.

5. Балансы:

- Баланс должен отображаться отдельно по каждому счету.
- Должны быть предусмотрены два типа отображения общего баланса:
  - Общий баланс в национальной валюте (суммовой баланс).
  - Общий баланс в иностранных валютах.
- Клиент должен иметь возможность переключаться между суммовыми и валютными балансами, выбирая, какой именно баланс он хочет видеть.

6. Синхронизация и кроссплатформенность:

- Все данные по счетам и балансам должны быть синхронизированы и моментально обновляться на всех платформах (Web, iOS, Android), чтобы клиент имел доступ к актуальной информации с любого устройства.

7. Безопасность и аудит:

- Доступ к счетам, их просмотр и все действия пользователя должны быть фиксированы в логах для аудита и безопасности.

2.10 Выписки по счетам

Примечание - является обязательным требованием

- Генерация и скачивание выписок с печатью банка с полной поддержкой на Web, iOS и Android.

Функциональные требования:

1. Раздел «Выписки»:

- В ДБО должен быть реализован отдельный раздел «Выписки», в котором клиенты смогут создавать и загружать выписки по счетам.
- В этом разделе должны отображаться все типы выписок, которые поддерживает банк, а также выписки, соответствующие стандартам международной финансовой отчетности.
- В раздел «Выписки» также должны быть включены следующие типы справок:
  - Справка о наличии счета.
  - Справка о балансе по счету.

2. Генерация и форматы выписок:

- Клиент должен иметь возможность выбирать параметры для формирования выписки:
  - Период за который требуется выписка. Минимальный и максимальный период за который может быть заказана выписки определяется исходя из операционных процессов банка и технических возможностей текущей АБС.
  - Тип выписки (например, выписка о движении средств, выписка с деталями транзакций и т.д.).
  - Формат, в котором клиент хочет получить выписку.

3. Доступные форматы для скачивания выписок:

- Все выписки должны быть доступны для скачивания в следующих форматах:
  - PDF с печатью банка – официальный документ, содержащий цифровую подпись и печать банка.
  - Excel – таблица, удобная для обработки данных.
  - TXT – текстовый файл, пригодный для импорта в различные системы.
  - CSV – текстовый файл с разделением данных запятыми, удобный для использования в системах управления данными и финансовых системах.
  - Формат для BPMN-системы Коммунды – специальный формат, который позволяет интегрировать выписки и отчеты в системы процессного управления, такие как BPMN Коммунда.

4. Дополнительные функции:

- Все сгенерированные выписки должны храниться в клиентском хранилище данных для их последующего скачивания в любой момент.
- Клиент должен иметь возможность просматривать историю заказанных выписок, а также повторно скачивать уже сгенерированные документы.

2.11 История транзакций

Примечание - является обязательным требованием

- История транзакций с фильтрацией по типам, дате и другим параметрам, с поддержкой бэкенда и фронта на всех платформах.

Функциональные требования:

1. Раздел «История»:

- В ДБО должен быть реализован отдельный раздел «История», в котором будут находиться все платежные поручения клиента.
- В разделе должны отображаться как платежные поручения, которые уже отправлены на обработку в банк, так и те, которые только созданы, но не подписаны.

2. Работа с платежными поручениями:

- Отображение статусов платежей: В «Истории» должны отображаться все платежи с их статусами: «Создан, но не подписан», «На подписи», «Отправлен в банк», «Проведен», «Отклонен» и другие возможные статусы.
- Клиент должен иметь возможность работать с платежами прямо из раздела «История»:
  - Подписание созданных, но не подписанных платежей: Клиент может отметить платежи, которые находятся в статусе «Создан, но не подписан», и подписать их через ЭЦП.
  - Просмотр деталей платежа: Для каждого платежного поручения должна быть возможность просмотреть его детали. В деталях платежа отображаются:
    - Сумма платежа.
    - Тип платежа.
    - Реквизиты отправителя и получателя.
    - Дата создания и проведения платежа.
  - Функции в деталях платежа:
    - Повторить платеж. Клиент может создать новый платеж на основе уже проведенного, автоматически заполнив его данными из оригинального платежа.
    - Создать шаблон. Клиент может создать шаблон платежного поручения на основе проведенного платежа.
    - Скачать платеж в формате PDF. Если платеж уже проведен, клиент может скачать его копию в формате PDF с печатью банка.

3. Фильтры и поиск:

- Для удобства работы с большим количеством платежей должны быть реализованы фильтры и поиск:
  - Фильтрация по дате: Клиент может выбрать диапазон дат для поиска платежей.
  - Фильтрация по типу платежного поручения: Например, платеж на счет, на карту, в казначейство, в бюджет, свифт-платеж и другие типы.
  - Фильтрация по сумме: Клиент может задать диапазон суммы платежа.
  - Фильтрация по статусу платежа: Клиент может отфильтровать платежи по их статусу (например, все проведенные или все неподписанные).
  - Фильтрация по отправителю/получателю: Возможность фильтровать платежи по указанному получателю или отправителю.
  - Фильтрация по номеру платежного поручения: Клиент может искать платеж по уникальному номеру платежного поручения, присвоенному ему при создании.
  - Поиск по ключевым словам: Возможность поиска платежей по ключевым данным, например, имени контрагента или номеру счета.

4. Дополнительные функции:

- Вся информация по платежным поручениям должна синхронизироваться между всеми платформами (Web, iOS, Android) в реальном времени, обеспечивая доступ к истории и деталям платежей с любого устройства.

2.12 Тарифный план

Примечание - является обязательным требованием

- Отображение актуального тарифного плана клиента на всех платформах с полной поддержкой бэкенда.

Функциональные требования:

1. Раздел «Мой тариф» или «Тарифный план»:

- В ДБО должен быть создан отдельный раздел «Мой тариф» или «Тарифный план», в котором клиент может видеть все данные о своем текущем тарифном плане.

2. Информация о тарифе:

- Отображение текущего тарифа: Клиент должен видеть название своего тарифного плана.
- Стоимость тарифа: В разделе должна отображаться стоимость ежемесячной оплаты за текущий тариф.
- Условия тарифа: Клиент должен видеть все условия, связанные с тарифом:
  - Лимиты на операции.
  - Комиссии за платежи, переводы и другие услуги.
  - Лимиты на использование различных услуг (например, количество бесплатных платежей или доступных операций в рамках тарифа).
  - Дополнительно: Для клиентов с индивидуальными тарифными планами должно быть такое же отображение тарифа и его условий, как и у клиентов с стандартными тарифными планами.

3. Возможность просмотра альтернативных тарифных планов:

- В этом же разделе клиент должен иметь возможность увидеть список доступных для него альтернативных тарифных планов.
  - Отображение названий альтернативных тарифов.
  - Отображение условий и стоимости каждого альтернативного тарифа.

- Сравнение текущего плана с альтернативными планами по ключевым параметрам (стоимость, комиссии, лимиты и т.д.).

4. Заявка на смену тарифа:

- Клиент должен иметь возможность оставить заявку на смену текущего тарифного плана. Для этого в разделе должен быть предусмотрен специальный интерфейс:
- Кнопка «Сменить тариф».
- Отображение формы для заявки, где клиент выбирает новый тарифный план.
- После выбора нового тарифного плана клиент подтверждает запрос через SMS-код или ЭЦП.

5. История тарифных планов:

- В разделе «Мой тариф» должна быть возможность просмотра истории тарифных планов, которые использовал клиент, включая даты их действия и изменения.

6. Дополнительные функции:

- Все изменения в тарифных планах должны синхронизироваться на всех платформах (Web, iOS, Android), чтобы клиент мог видеть актуальные данные на любом устройстве.

7. Бизнес логика

- Функционал тарифов должен быть реализован как независимый от АБС сервис и иметь свою бизнес логику.
-Сервис должен позволять добавлять в него новые тарифные планы и корректировать действующие тарифные планы - конструктор тарифов.
-Сервис должен иметь свое открытое API и обеспечивать интеграцию с ним любых внутрибанковских систем.

2.13 Корпоративные карты

Примечание - является обязательным требованием

- Поддержка отображения балансов, операций и пополнения корпоративных карт на всех платформах.

Функциональные требования:

1. Раздел «Корпоративные карты»:

- В ДБО должен быть создан отдельный раздел «Корпоративные карты», находящийся рядом с разделом «Счета». В этом разделе клиент может управлять своими корпоративными картами.

2. Отображение списка карт:

- Клиент должен видеть список всех активных корпоративных карт, доступных в его компании.
  - Баланс карты: Отображение текущего баланса по каждой карте.
  - Ограниченный номер карты: Для безопасности в списке отображаются только последние 4 цифры карты.
  - Платежная система карты: Указание платежной системы (Visa, MasterCard, и т.д.).
  - Валюта карты: Валюта, в которой открыта карта (сум, доллар, евро и т.д.).

3. Функционал при отсутствии карт:

- Если у клиента нет активных корпоративных карт, ему должна предоставляться возможность подать заявку на открытие новой корпоративной карты.
  - Форма заявки на карту: Клиент заполняет информацию о типе карты, валюте, платежной системе, выбирает счет, к которому будет привязана карта, и отправляет заявку в банк.

4. Детали карты:

- При переходе в детали конкретной карты клиент видит:
  - Полные реквизиты карты.
  - Баланс и историю транзакций: Отображение всех операций по карте с возможностью фильтрации транзакций по дате, типу операции и сумме.
  - Платежная система и валюта карты.

5. Основные действия в разделе «Корпоративные карты»:

  - Копирование реквизитов карты: Возможность копирования реквизитов для дальнейшего использования.
  - Создание выписки по карте: Клиент может сформировать выписку по транзакциям за выбранный период в форматах PDF, Excel, TXT или CSV.
  - Создание шаблона платежа: Возможность создать шаблон платежа на основе выбранной карты для будущих транзакций.
  - Создание платежного поручения: Клиент может создавать платежи, списывающиеся с корпоративной карты.

- Пополнение карты: Возможность пополнить корпоративную карту через ДБО с одного из счетов компании.

6. История транзакций:

- Полная история транзакций по каждой карте должна отображаться в деталях карты.
- Клиент может фильтровать транзакции по дате, типу операции и сумме.

7. Синхронизация данных:

- Все действия и изменения с корпоративными картами должны моментально синхронизироваться между платформами (Web, iOS, Android), чтобы клиент мог видеть актуальные данные на любом устройстве.

8. Бизнес логика

- Функционал корп карт должен быть реализован как независимый от АБС сервис и иметь свою бизнес логику.
-Сервис должен позволять добавлять в него новые корпоративные и бизнес карты.
-Сервис должен иметь собственную интеграцию с процессингом - TIETO
-Сервис должен иметь свое открытое API и обеспечивать интеграцию с ним любых внутрибанковских систем.

2.14 Картотека 1 и 2, платежные требования

Примечание - является обязательным требованием

- Полная реализация поддержки Картотеки 1 и 2 для обработки платежных требований на Web, iOS и Android.

2.14.1. Платёжные требования

Функциональные требования:

1. Раздел «Платёжные требования»:

- В ДБО должен быть создан отдельный раздел «Платёжные требования», где клиент может управлять исходящими платёжными требованиями.

2. Создание платёжного требования:

- Клиент должен иметь возможность создавать исходящие платёжные требования, заполняя форму согласно регламентам Центрального Банка Республики Узбекистан и внутренним регламентам банка.
  - Обязательные и необязательные поля: Должны быть включены все поля, обязательные для заполнения по стандартам ЦБ Узбекистана, а также дополнительные поля, предусмотренные внутренними требованиями банка.

3. История платёжных требований:

- Клиент должен видеть список всех созданных исходящих платёжных требований, включая их статусы (на обработке, оплачено, отклонено, отложено).
  - Фильтрация: Должны быть реализованы фильтры по дате, сумме, статусу платежа.

4. Действия с платёжными требованиями:

- Клиент может просматривать детали любого созданного платёжного требования, а также иметь возможность:
  - Повторить платёжное требование.
  - Создать шаблон на основе платёжного требования.
  - Скачать платёжное требование в формате PDF.

2.14.2. Картотека 1

Функциональные требования:

1. Раздел «Картотека 1»:

- В этом разделе клиент видит входящие платёжные требования от других клиентов и может совершать следующие действия с каждым из них:
  - Оплатить полностью.
  - Оплатить частично.
  - Отклонить платёжное требование.
  - Отложить ответ на 5 дней.

2. Полные детали платёжного требования:

- Клиент может просматривать полные детали входящего платёжного требования, включая всю информацию, необходимую для принятия решения по нему.

3. Ограничение на использование ДБО:

- Пока клиент не примет решение по входящему платёжному требованию (оплатить, оплатить частично, отклонить или отложить), ему должна быть ограничена возможность пользоваться другими функциями ДБО. Это требование должно быть реализовано в соответствии с регламентом ЦБ Узбекистана.

2.14.3. Картотека 2

Функциональные требования:

1. Раздел «Картотека 2»:

- В разделе «Картотека 2» клиент видит все безакцептные платёжные требования, которые уже списались с его расчётного счёта.

2. Детали:

- Клиент может просматривать полные детали каждого безакцептного платёжного требования для понимания всех списаний.

3. История и фильтры:

- Клиент должен иметь возможность фильтровать безакцептные требования по дате, сумме, типу платежа, а также экспортировать список в удобные форматы (PDF, Excel, CSV).

2.15 Сервис уведомлений

Примечание - является обязательным требованием

- Внедрение сервиса уведомлений с поддержкой пуш-уведомлений и модальных окон на всех платформах.

Функциональные требования:

1. Разделение на бэкэнд и фронтэнд:

- Сервис уведомлений должен быть реализован как на уровне бэкэнда, так и на уровне фронтенда (web, iOS, Android), обеспечивая централизованное управление уведомлениями и их отображение на всех платформах.

2. Уведомления о транзакциях:

- Сервис должен автоматически уведомлять клиента о всех исходящих и входящих транзакциях. Уведомления должны приходить в виде:
  - Пуш-уведомлений: Настраиваются клиентом через раздел «Настройки».
  - Модальных окон: Отображаются внутри интерфейса ДБО при важной транзакции или сообщении.

3. Новости от банка:

- Сервис уведомлений должен поддерживать возможность отправки новостей и важных сообщений от банка. Эти сообщения отображаются через:
  - Модальные окна на всех платформах (web, iOS, Android).
  - Баннеры: Размещаются на главной странице ДБО и могут содержать информацию о новых продуктах или сервисах банка.

4. Маркетинговые уведомления:

- Сервис должен поддерживать маркетинговые коммуникации. Клиенту могут быть отправлены:
  - Маркетинговые баннеры.
  - Маркетинговые модальные окна.

5. Управление кастомными уведомлениями:

- Владелец сервиса должен иметь возможность загружать кастомные уведомления с разным текстом, изображениями и ссылками на внешние ресурсы.
  - Гибкая настройка: Возможность загрузки баннеров и уведомлений с различной структурой текста и изображений.
  - Вставка ссылок: В тексте уведомлений должна быть возможность вставлять ссылки на внешние ресурсы для взаимодействия с клиентом.
  - Прикрепление документов: Возможность загружать в уведоление документы в формате PDF, DOC, DOCX.

6. Кросс-платформенность уведомлений:

- Сервис должен учитывать кросс-платформенность:

- Механизм синхронизации уведомлений: Если клиент прочитал уведомление, баннер или маркетинговое сообщение на одной платформе, они не должны повторно отображаться на других платформах (web, iOS, Android).
- Отслеживание взаимодействий: Система должна отслеживать, какие уведомления были прочитаны или закрыты клиентом на любой из платформ, и синхронизировать это между всеми устройствами.

7. Управление частотой уведомлений:

- Клиент должен иметь возможность настроить частоту получения уведомлений через раздел «Настройки»:
- Выбор типов уведомлений: Настройка уведомлений о транзакциях, новостях, маркетинговых баннерах, модальных окнах и т.д.
- Включение/выключение уведомлений: Возможность полностью отключить определенные типы уведомлений, если это допустимо по политике банка.

Этот сервис позволит гибко управлять коммуникациями с клиентом, сохраняя удобство и актуальность уведомлений на всех платформах, а также предоставит маркетинговую гибкость для банка.

3. Транзакционный сервис для валютных платежей

3.1 SWIFT-переводы

Примечание - является обязательным требованием

- Реализация поддержки SWIFT-переводов на Web, iOS и Android.

Функциональные требования:

1. Реализация на всех платформах:

- SWIFT-переводы должны быть реализованы на всех трех платформах: iOS, Android, Web.
- На каждой платформе должна быть доступна одинаковая функциональность для создания и отправки SWIFT-платежей.

2. Создание отдельного типа платежного поручения:

- В ДБО должен появиться отдельный тип платежного поручения для SWIFT-переводов.
- Клиент должен иметь возможность выбрать данный тип платежа при создании платежного поручения.

3. Соответствие регламентам:

- SWIFT-перевод должен соответствовать всем внутренним регламентам банка, регламентам Центрального банка Республики Узбекистан и регламентам валютного законодательства Республики Узбекистан.
- Учитываются правила валютного контроля, валютного обмена, и требования по идентификации клиента при международных переводах.

4. Поля платежного поручения:

- Все обязательные поля платежного поручения для SWIFT-переводов должны быть настроены в соответствии с международными стандартами SWIFT.
- BIC код получателя.
- IBAN номер получателя.
- Наименование банка получателя.
- Наименование получателя и его адрес.
- Валюта и сумма перевода.
- Цель перевода (детализация перевода).
- Комиссии (за счет отправителя или за счет получателя).

5. Проверка данных перед отправкой:

- Перед отправкой SWIFT-платежа система должна проверять правильность всех введенных данных (например, IBAN и BIC) в соответствии с международными стандартами и внутренними требованиями банка.

6. Подпись и подтверждение:

- SWIFT-перевод должен подписываться клиентом с использованием ЭЦП.
- После успешного создания и подписания платежного поручения клиент должен получить уведомление о том, что платеж принят на обработку.

7. История и отслеживание:

- Клиент должен иметь возможность отслеживать статус своих SWIFT-платежей в разделе «История транзакций».

- После отправки платежа клиент может получить уведомление о статусе выполнения (например, «В процессе», «Завершено», «Отменено»).

8. Интеграция с валютными операциями:

- SWIFT-перевод должен учитывать текущие валютные курсы и условия банка при обработке валютных платежей.
- Если перевод осуществляется в иностранной валюте, клиенту должны быть предоставлены данные о текущем курсе валюты и комиссиях.

9. Поддержка массовой загрузки:

- Возможность массовой загрузки SWIFT-платежей с использованием шаблона, аналогично другим типам платежных поручений.

10. Локализация и кроссплатформенность:

- Вся информация о SWIFT-переводах должна синхронизироваться между платформами. Например, если платеж был создан на Web, его можно будет увидеть и подписать на iOS или Android.

Это дополнение позволит реализовать функционал SWIFT-переводов, соответствующий всем юридическим требованиям и стандартам, обеспечивая клиенту удобство работы с международными платежами.

11. Бизнес логика

- Функционал swift перевода должен быть реализован как независимый от АБС сервис и иметь свою бизнес логику.
-Сервис должен позволять вносить в него изменения.
-Сервис должен самостоятельно осуществлять все валидации и отправлять в АБС только платежи которые успешно прошли валидацию
-Сервис должен иметь свое открытое API и обеспечивать интеграцию с ним любых внутрибанковских систем.

3.2 Покупка и продажа валюты

Примечание - является обязательным требованием

- Поддержка покупки и продажи валюты с интеграцией на всех платформах.

Функциональные требования:

1. Реализация на всех платформах:

- Покупка и продажа валют должны быть реализованы на всех трех платформах: iOS, Android, Web.
- На каждой платформе должна быть доступна одинаковая функциональность для создания и отправки валютных платежей.

2. Создание отдельных типов платежных поручений:

- В ДБО должны появиться отдельные типы платежных поручений для:
  - Покупки валюты
  - Продажи валюты
- Клиент должен иметь возможность выбрать соответствующий тип платежа при создании валютного поручения.

3. Соответствие регламентам:

- Операции по покупке и продаже валюты должны соответствовать всем внутренним регламентам банка, а также регламентам Центрального банка Республики Узбекистан и требованиям валютного законодательства Республики Узбекистан.
  - Включает соблюдение валютного контроля, идентификацию клиентов, и установленные лимиты на валютные операции.

4. Поля платежного поручения:

- Все обязательные и необязательные поля для покупки и продажи валюты должны быть настроены в соответствии с требованиями банка и законодательством. Поля могут включать:
  - Наименование валюты.
  - Сумма в национальной или иностранной валюте.
  - Курс валюты.
  - Комиссии за покупку/продажу валюты.
  - Наименование и реквизиты контрагента.

5. Проверка данных перед отправкой:

- Перед отправкой платежа система должна проверять правильность всех введенных данных (например, сумму и курс), чтобы соответствовать внутренним стандартам и законодательным требованиям.

6. Подпись и подтверждение:

- Все операции по покупке и продаже валюты должны быть подписаны клиентом с использованием ЭЦП.

- После успешного создания и подписания валютного поручения клиент получает уведомление о принятии его на обработку.

7. История и отслеживание:

- Клиент должен иметь возможность отслеживать статус всех операций по покупке и продаже валюты в разделе «История транзакций».
- Статусы операций должны включать такие этапы, как «В процессе», «Завершено», «Отменено».

8. Интеграция с текущими валютными курсами:

- Для каждой операции покупки или продажи валюты система должна отображать актуальные валютные курсы, предоставляемые банком.
- Клиенту должны быть предоставлены подробные сведения о курсе, комиссии и итоговой сумме операции.

9. Поддержка массовой загрузки:

- Должна быть возможность массовой загрузки поручений на покупку и продажу валюты с использованием шаблона, аналогично другим типам платежных поручений.

10. Локализация и кроссплатформенность:

- Все данные о валютных операциях должны синхронизироваться между платформами, чтобы обеспечить кроссплатформенную работу. Например, если покупка валюты была инициирована на Web, клиент должен иметь возможность увидеть и завершить операцию на iOS или Android.

Данный функционал обеспечит возможность клиентам удобно и оперативно совершать операции по покупке и продаже валюты с учетом всех регламентов и требований, что позволит соответствовать стандартам безопасности и удобства для пользователей.

11. Бизнес логика

- Функционал покупки/продажи валюты должен быть реализован как независимый от АБС сервис и иметь свою бизнес логику.
-Сервис должен позволять вносить в него изменения.
-Сервис должен самостоятельно осуществлять все валидации и отправлять в АБС только платежи которые успешно прошли валидацию.
-Сервис должен иметь свое открытое API и обеспечивать интеграцию с ним любых внутрибанковских систем.

3.3. Конвертация валюты

Примечание - является обязательным требованием

- Полная поддержка конвертации валют между счетами клиента с реализацией бэкенда и фронта.

Функциональные требования:

1. Реализация на всех платформах:

- Конвертация валюты должна быть реализована на всех трех платформах: iOS, Android, Web.
- На каждой платформе должна быть доступна одинаковая функциональность для создания и отправки поручений на конвертацию валюты.

2. Создание отдельного типа платежного поручения:

- В ДБО должен быть создан отдельный тип платежного поручения для конвертации валюты.
- Конвертация осуществляется только между валютными счетами клиента, то есть это всегда валютные пары, такие как:
  - Евро-Доллар
  - Фунт-Юань
  - Йена-Казахский тенге и другие доступные валютные пары.

3. Соответствие регламентам:

- Операции по конвертации валюты должны соответствовать всем внутренним регламентам банка, а также регламентам Центрального банка Республики Узбекистан и требованиям валютного законодательства Республики Узбекистан.
  - Это включает соблюдение валютного контроля, лимитов на конвертацию, идентификацию клиента, а также контроль за валютными операциями.

4. Поля платежного поручения:

- Клиент должен иметь возможность выбрать два валютных счета для конвертации — счет списания и счет зачисления.
- Все обязательные и необязательные поля для конвертации должны быть настроены в соответствии с требованиями банка. Поля могут включать:
  - Валютная пара (выбор валют счетов).

- Сумма конвертации.
- Курс валюты (если применимо).
- Комиссии за конвертацию.
- Дата выполнения операции (если требуется отложенная конвертация).

5. Автоматический расчет суммы зачисления:

- После выбора валютной пары и ввода суммы для конвертации система автоматически рассчитывает сумму зачисления на основании актуального курса обмена.

6. Проверка данных перед отправкой:

- Перед отправкой платежного поручения система должна проверять корректность всех введенных данных и соответствие регламентам. Например, сумма, лимиты и курсы валют.

7. Подпись и подтверждение:

- Все операции по конвертации валюты должны быть подписаны клиентом с использованием ЭЦП.
- После успешного создания и подписания поручения на конвертацию клиент получает уведомление о принятии его на обработку.

8. История и отслеживание:

- Клиент должен иметь возможность отслеживать статус всех операций по конвертации валюты в разделе «История транзакций».
- Статусы операций должны включать такие этапы, как «В процессе», «Завершено», «Отменено».

9. Поддержка массовой загрузки:

- Должна быть возможность массовой загрузки поручений на конвертацию валюты через шаблон, аналогично другим типам платежных поручений.

10. Интеграция с валютными курсами:

- Система должна отображать актуальные курсы валют для выбранных валютных пар. Клиенту должны быть доступны детализированные сведения о курсе, комиссионных сборах и итоговой сумме конвертации.

11. Локализация и кроссплатформенность:

- Все данные о конвертационных операциях должны быть синхронизированы между платформами, чтобы обеспечить кроссплатформенную работу. Например, если конвертация валюты была инициирована на Web, клиент должен иметь возможность завершить операцию на iOS или Android.

12. Интеграция с АБС для онлайн конвертации:

- Система должна быть интегрирована с АБС для обновления данных по валютным счетам клиента после успешной конвертации валют, чтобы клиент мог совершить онлайн конвертацию

Этот функционал позволит клиентам осуществлять конвертацию валют между своими валютными счетами оперативно, с учетом всех регуляторных требований и обеспечения полной прозрачности и контроля операций.

13. Бизнес логика

- Функционал конвертации валюты должен быть реализован как независимый от АБС сервис и иметь свою бизнес логику.
-Сервис должен позволять вносить в него изменения.
-Сервис должен самостоятельно осуществлять все валидации и отправлять в АБС только платежи которые успешно прошли валидацию.
-Сервис должен иметь свое открытое API и обеспечивать интеграцию с ним любых внутрибанковских систем.

3.4 Валютные контракты

Примечание - является обязательным требованием

- Поддержка валютных контрактов с интеграцией с ЕИСВО, реализованная на Web, iOS и Android.

Функциональные требования:

1. Раздел "Валютные контракты":

- В ДБО должен быть создан отдельный раздел «Валютные контракты», где будут отображаться все валютные контракты клиента, зарегистрированные в системе ЕИСВО.
- Клиент должен иметь возможность получить полный доступ к информации о каждом валютном контракте, включая:

- Номер контракта.
- Дату заключения.
- Статус контракта (активный, завершенный).
- Валюту контракта.
- Страну контрагента.
- Вид деятельности (импорт/экспорт).
- Сумму контракта.
- Даты и суммы платежей, связанных с контрактом.

2. Фильтрация и сортировка контрактов:

- Клиент должен иметь возможность фильтровать и сортировать свои валютные контракты по следующим параметрам:
  - Валюта контракта (например, доллар, евро, тенге и т.д.).
  - Страна контрагента (страна, с которой заключен контракт).
  - Вид деятельности (импорт, экспорт).
  - Статус контракта (активный, завершенный).
  - Дата заключения контракта.
  - Сумма контракта.
- Реализация удобной сортировки данных по любым доступным параметрам должна быть доступна на всех платформах (iOS, Android, Web).

3. Детализация контрактов:

- При переходе в детали конкретного контракта клиент должен видеть все связанные с ним операции и платежи. Это включает:
  - Историю совершенных платежей по контракту с указанием даты, суммы и статуса платежа.
  - Остаток по контракту (сумма, которая еще не была оплачена).
  - Сроки исполнения платежей по контракту (если применимо).

4. Форма регистрации новых контрактов:

- Клиент должен иметь возможность заполнить форму для регистрации нового валютного контракта через ДБО.
  - Форма должна включать такие поля, как:
    - Номер контракта.
    - Валюта контракта.
    - Страна контрагента.
    - Вид деятельности (импорт или экспорт).
    - Сумма контракта.
    - Дата заключения.
    - Дата исполнения обязательств.

- После заполнения формы клиент может отправить её на регистрацию в систему ЕИСВО. Интеграция с системой ЕИСВО должна быть реализована для автоматической отправки данных.

5. Завершенные контракты:

- В разделе «Валютные контракты» клиент должен иметь возможность просматривать завершенные контракты.
- В завершенных контрактах должна быть доступна вся информация по ранее совершенным операциям.
- Завершенные контракты должны иметь соответствующую метку «Завершено» и не должны смешиваться с активными контрактами.

6. Управление контрактами:

- В разделе также должны быть инструменты для управления контрактами:
- Возможность обновлять или редактировать информацию по контракту.
- Возможность закрывать контракт в случае выполнения всех обязательств.

7. Интеграция с платежной системой:

- Все платежи по валютным контрактам должны быть связаны с разделом «История транзакций» в ДБО. Это должно позволить клиенту:
- Видеть полную историю совершенных платежей по контрактам.
- Следить за выполнением обязательств и сроками оплаты.

8. Кроссплатформенность и синхронизация:

- Все данные по валютным контрактам должны быть синхронизированы между всеми платформами (iOS, Android, Web), чтобы клиент мог видеть актуальную информацию по своим контрактам на любом устройстве.

Этот функционал позволит клиентам управлять своими валютными контрактами, отслеживать статус их выполнения, контролировать платежи и своевременно выполнять обязательства по контрактам.

3.5 Шаблоны валютных платежей

Примечание - является обязательным требованием

- Внедрение шаблонов для валютных платежей с развертыванием на всех платформах.

Функциональные требования:

1. Раздел «Шаблоны валютных платежей»:

- В ДБО должен быть отдельный раздел «Шаблоны» на всех трех платформах (iOS, Android, Web), где будут храниться шаблоны валютных платежей.
- В этом разделе клиент должен видеть список всех существующих шаблонов валютных платежей, а также иметь следующие возможности:
  - Добавить новый шаблон.
  - Удалить существующий шаблон.
  - Редактировать существующий шаблон.

2. Создание нового шаблона:

- При создании нового шаблона клиенту должна отображаться стандартная форма с типами валютных платежных поручений, которые доступны в системе. Это могут быть следующие типы:
  - SWIFT-перевод.
  - Покупка валюты.
  - Продажа валюты.
  - Конвертация валюты.
- Клиент должен заполнить все обязательные и необязательные поля, как при создании обычного валютного платежного поручения, а затем нажать кнопку «Создать шаблон».
- После создания шаблона должна появиться опция «Создать платёж из этого шаблона», которая автоматически откроет заполненный платеж для подтверждения и отправки.

3. Саксесс-экран:

- После того как клиент подписал и отправил валютный платеж, на саксесс-экране должна отображаться кнопка «Создать шаблон из этого платежа», чтобы упростить создание шаблонов на основе уже выполненных операций.

4. Кроссплатформенность и синхронизация:

- Шаблоны валютных платежей должны быть синхронизированы между всеми платформами (iOS, Android, Web). Если клиент создал, отредактировал или удалил шаблон на одной платформе, изменения должны сразу же быть видны на других платформах.

5. Создание платежа из шаблона:

- Клиент должен иметь возможность создавать новый валютный платеж на основе любого существующего шаблона. Для этого при выборе шаблона

должна быть доступна функция «Создать платеж», которая откроет заполненную форму платежа, где клиент может отредактировать данные или просто подписать и отправить.

Эти требования обеспечат удобное создание, управление и использование шаблонов для всех видов валютных платежей в системе, аналогично тому, как это реализовано для суммовых платежей.

3.6 GPI трекер

- Интеграция GPI трекера для отслеживания международных платежей на всех платформах.

Функциональные требования:

1. Реализация GPI-трекера в ДБО:

- В ДБО должен быть реализован функционал GPI-трекера, который позволит клиентам отслеживать путь Swift-переводов.
- Данный функционал должен быть доступен на всех платформах (iOS, Android, Web).

2. Отображение деталей пути платежа:

- В разделе «История» и в разделе «Счета» (при переходе в валютный счет) должны быть доступны детализированные сведения о пути отправленного Swift-перевода.
- Клиент, провалившись в детали платежа, должен видеть статусы прохождения платежа по этапам и полную информацию о текущем статусе Swift-перевода.

3. Отдельный раздел для Swift-переводов:

- В ДБО должен быть предусмотрен отдельный раздел, где будут собраны все отправленные клиентом Swift-переводы.
- В этом разделе клиент видит список всех Swift-платежей, а также имеет возможность провалиться в детали каждого перевода, где отображается полный путь платежа и его статус на каждом этапе.

4. Функция отслеживания по идентификационному номеру:

- В разделе GPI-трекера должна быть доступна функция ввода идентификационного номера платежа. Клиент может ввести номер и нажать

кнопку «Отследить», после чего система загрузит и отобразит полный путь Swift-перевода, включая текущий статус.

5. Синхронизация между платформами:

- Все данные о статусе и пути Swift-переводов должны быть синхронизированы между всеми платформами (iOS, Android, Web), чтобы клиент мог отслеживать статус перевода с любого устройства.

6. Интеграция с АБС

На данный момент GPI-трекер интегрирован в АБС банка, чтобы GPI-трекер появился в ДБО, нужно будет реализовать интеграцию ДБО с АБС через внутрибанковскую ESB.

Этот функционал позволит клиентам банка легко отслеживать свои международные Swift-платежи, обеспечивая прозрачность и удобство использования на всех этапах.

4. Многоуровневая модель подписания

4.1 Подписание платежей несколькими уровнями

Примечание - является обязательным требованием

- Реализовать многоуровневую модель подписания с поддержкой нескольких уровней утверждения и подписания платежей для корпоративных клиентов на всех платформах.

Функциональные требования:

1. Определение многоуровневой модели подписания:

- Многоуровневая модель подписания — это расширение ролевой модели, которая позволяет клиентам организовывать процесс утверждения и подписания платежных поручений несколькими лицами перед тем, как платеж будет отправлен в банк.
- Данная модель является частью ролевой системы ДБО и позволяет настраивать индивидуальные этапы утверждения и подписания платежей.

2. Логика работы:

- Стандартная схема подразумевает создание и одно подписание платежа. В многоуровневой модели добавляются промежуточные этапы утверждения и подписания, которые могут выполняться несколькими лицами.
- Пример схемы: Платеж должен быть утвержден коммерческим директором, затем финансовым директором и подписан генеральным директором.
- У каждого из этих пользователей должна быть учетная запись в ДБО с заранее настроенными правами. Коммерческий и финансовый директора имеют право утверждать платежи, генеральный директор — подписывать платежи.
- Этапов утверждения может быть несколько, а этап подписания всегда один.

3. Настройка через админпанель:

- Сотрудники банка должны иметь возможность настраивать многоуровневую модель подписания через админпанель.
- При настройке прав через индивидуальную роль администратор определяет количество уровней утверждения и наделяет пользователей соответствующими правами на утверждение и подписание.

4. Прозрачность и синхронизация:

- Все этапы утверждения и подписания должны синхронизироваться между всеми платформами (iOS, Android, Web).
- Клиент, просматривая детали платежа в разделе «История транзакций», должен видеть текущее состояние платежа, включая информацию о том, кто и когда утвердил или подписал платеж.
- В истории должны отображаться промежуточные статусы: "на утверждении", "утвержден", "на подписании", "подписан".

5. Логирование действий:

- Все действия по утверждению и подписанию платежей должны логироваться. В деталях платежа должна отображаться информация о том, кто утвердил или подписал платеж, а также дата и время каждого действия.

6. Применение на все типы платежей:

- Многоуровневая модель подписания распространяется на все типы платежных поручений в ДБО, включая зарплатную ведомость.

7. Дополнительные функции:

- Клиент может отслеживать прогресс подписания в реальном времени, видеть, какие лица уже утвердили или подписали платеж, и сколько еще остается для завершения процесса.

Эта модель предоставляет гибкость и контроль над процессом проведения платежей, позволяя компаниям настроить сложную систему утверждений и подписаний.

5. Интеграции с внешними системами

5.1 Интеграция с DIDOX

- Реализовать интеграцию с DIDOX для работы с документами на всех платформах.(Опциональный пункт)

Функциональные требования:

1. Опциональность интеграции:

- Интеграция с DIDOX является опциональной и не является обязательной для всех клиентов. Она должна быть реализована по запросу и по желанию клиентов, использующих систему DIDOX

2. Двусторонний обмен данными:

- Загрузка платежей из ДБО в DIDOX:
  - Все проведенные платежные поручения, которые были отправлены через ДБО, должны автоматически загружаться в систему DIDOX.
  - Это позволит клиентам видеть все их транзакции и данные о платежах в DIDOX без необходимости дополнительного ввода информации.

- Загрузка счетов на оплату из DIDOX в ДБО:
  - Клиент может загружать в ДБО счета на оплату, которые ему выставлены через DIDOX.
  - ДБО должно автоматически формировать на основе этих данных платежное поручение с заполненными реквизитами, полученными из DIDOX.
  - Клиенту остается только подписать платежное поручение в ДБО.

3. Простой процесс интеграции:

- Процесс интеграции должен быть интуитивно понятным, как в случае с 1С, и предусматривать минимальные технические сложности для клиента.

4. Кроссплатформенность:

- Интеграция с DIDOX должна работать на всех платформах (Web, iOS, Android), предоставляя возможность клиентам управлять данными из любого устройства.

5. Гибкость настройки:

- Клиент должен иметь возможность активировать или деактивировать данную интеграцию в зависимости от их потребностей.

Эта интеграция позволит клиентам эффективно работать с двумя системами одновременно, минимизируя ручной ввод данных и обеспечивая автоматизацию процессов.

5.2 Интеграция с 1С

- Полная поддержка интеграции с 1С для бухгалтерского учета на Web, iOS и Android.

Функциональные требования:

1. Поддержка актуальной версии 1С:

- Интеграция должна быть настроена на дефолтную актуальную версию 1С и не требовать дополнительных доработок со стороны клиента или банка.
- Обеспечение двусторонней интеграции между ДБО и 1С для удобной работы клиентов с бухгалтерской документацией.

2. Двусторонний обмен данными:

- Загрузка документов из 1С в ДБО:
  - Клиент может загружать созданные платежные поручения и другие документы из 1С напрямую в ДБО.
  - В ДБО эти платежи будут создаваться автоматически на основе данных, полученных из 1С, как в функционале импорта документов.
  - После загрузки клиенту остается только подписать эти платежи.

- Автоматическая загрузка платежей в 1С:

- Все платежные поручения, проведенные в ДБО, должны автоматически загружаться в программу 1С для дальнейшей работы клиента с ними в бухгалтерском учете.
- Эта интеграция должна учитывать все проведенные транзакции для их корректного отражения в 1С.

3. Процесс интеграции:

- Интеграция должна происходить через генерацию уникального токена:
  - В ДБО клиент заходит в раздел «Интеграция с 1С», нажимает кнопку «Сформировать уникальный токен».
  - Получив токен, клиент вводит его в программе 1С в разделе «Интеграция с банком».
  - После ввода токена в 1С, интеграция между системой ДБО и 1С осуществляется автоматически без необходимости ручной настройки.

4. Прозрачность и простота интеграции:

- Процесс должен быть интуитивно понятным и не требовать технической поддержки или вмешательства специалистов для настройки.

5. Кроссплатформенность:

- Интеграция должна быть доступна и работать на всех платформах (Web, iOS, Android), предоставляя возможность клиентам использовать все функции на любом устройстве.

Интеграция с 1С позволит клиентам легко управлять платежами и бухгалтерской документацией, сокращая время на рутинные операции и обеспечивая актуальность данных как в ДБО, так и в бухгалтерской системе 1С.

6. Сервисы

Раздел «Сервисы»

В новом ДБО должен появиться отдельный раздел, который будет называться «Сервисы». Этот раздел должен быть доступен на всех платформах (Web, iOS, Android) и подразделяться на два основных подраздела:

1. Банковские продукты:

В этом подразделе клиент сможет увидеть все доступные для подключения онлайн банковские продукты. Среди них:

- Корпоративные карты: Возможность оставить заявку на выпуск корпоративной карты и управлять уже выпущенными картами.
- Зарплатный проект: Возможность подключения зарплатного проекта, включая добавление сотрудников и автоматизацию выплат.
- Депозиты: Возможность открыть депозит онлайн с выбором условий, сроков и сумм.
- Заявка на кредит: Подключение кредитных продуктов с возможностью подачи заявки и отслеживания ее статуса.
- Сопровождение валютных контрактов: Поддержка экспортно-импортных операций.
- Вэд-ассистент: Сервис для облегчения работы бизнеса во внешнеэкономической деятельности, полное сопровождение ВЭД клиента.
- Факторинг: Заявка на подключение факторинговых услуг для улучшения оборота средств компании.
- Заказ терминала для эквайринга: Заявка на подключение и доставку терминала.
- Подключение интернет - эквайринга: Возможность подключения интернет-эквайринга для приема платежей онлайн.
- Прием оплат через QR-коды: Подключение функционала для приема платежей через QR-коды.
- Банковские гарантии: Возможность подачи заявки на банковскую гарантию для выполнения обязательств перед контрагентами.

2. Нефинансовые сервисы:

Этот подраздел включает дополнительные бизнес-услуги, которые банк предоставляет своим клиентам:

- Проверка контрагента: Возможность онлайн проверки надежности контрагента.
- Онлайн-бухгалтерия: Сервисы для ведения бухгалтерии, учета и налоговой отчетности.
- Документооборот: Возможность оформления и отправки счетов-фактур, актов, накладных, а также заказ и управление документацией.
- Внутрибанковский marketplace: Сервис для заказа и покупки различных товаров и услуг, предлагаемых партнерами и клиентами банка.
- Услуги банка по размещению рекламы: Поддержка маркетинговых и рекламных услуг для клиентов банка.
- Банковская подписка: Платные подписные сервисы банка, которые облегчают доступ к дополнительным услугам и инструментам.

- Инфопространство банка: Раздел, где будут публиковаться информация о мероприятиях, курсах и лекциях для предпринимателей, а также записи вебинаров и тренингов.

Требования:
1. Раздел «Сервисы» должен быть доступен на всех трех платформах — Web, iOS, Android.
2. Интерфейсы разделов должны быть интуитивно понятными, с возможностью быстрого подключения продуктов и сервисов.
3. Возможность оставлять заявки и подключать продукты онлайн: Каждая услуга и продукт должны иметь возможность полного подключения через интерфейс ДБО без необходимости посещения банка.
4. Раздел должен поддерживать кроссплатформенную синхронизацию: Все продукты и заявки, созданные или поданные на одной платформе, должны быть синхронизированы на других платформах.
5. Нефинансовые сервисы должны быть реализованы в виде удобного каталога, с возможностью быстро найти нужный продукт или услугу и оформить заявку.