**Техническое задание для разработки страницы банковского приложения**

* **Введение**

1.1 **Цель документа**  
Этот документ представляет собой техническое задание для разработки бэкенда страницы банковского приложения с использованием Java Spring. ТЗ предназначено для команды студентов-бэкенд разработчиков, работающих над финальным проектом. Основной целью является создание безопасного и удобного интерфейса для пользователей банка с функционалом просмотра баланса, переводов средств, управления счетами и просмотра истории транзакций.

* **Общее описание**

2.1 **Обзор**  
Страница банковского приложения должна предоставлять клиентам возможность управлять своими банковскими счетами, выполнять переводы и проверять баланс. Администраторам необходимо предоставлять инструменты для управления пользователями, счетами и отчетностью.

Идеи названий :

*CardSphereMoneyLinkPayPulseCardFusionCardNexusPayFlow*

2.2 **Функции продукта**

* + Система авторизации и аутентификации клиентов.
  + Управление банковскими счетами (просмотр баланса, создание новых счетов, закрытие счетов, блокирование счета клиента).
  + Переводы средств между счетами внутри банка и на внешние счета.
  + Просмотр истории транзакций с фильтрацией.
  + Управление профилем пользователя (создание, внесение изменений).
  + Уведомления о транзакциях и активностях по счету

2.3 **Классы пользователей и характеристики**

* + **Клиенты**: физические лица, которые используют приложение для управления своими счетами и выполнения транзакций.
  + **Администраторы**: банковские сотрудники, управляющие счетами, транзакциями и отчетностью.
* **Требования к функциональности**

3.1 **Управление учетными записями пользователей**

* + Регистрация новых клиентов с вводом имени, фамилии, электронной почты, телефона, пароля и других необходимых данных, отправка подтверждающих документов.
  + Валидация данных пользователя при регистрации или внесении изменений (проверка администратором и подтверждение валидации)
  + Внесение изменений контактной информации (имени, фамилии, домашнего адреса, даты рождения, паспортных данных, адреса электронной почты и номера телефона).
  + Авторизация по email и паролю для безопасности.
  + Возможность восстановления доступа через email.
  + Возможность удалять учетную запись клиента.
  + Смена пароля и аутентификация
  + Возможность привязки карт для пополнения счета или оплаты услуг.

3.2 **Управление банковскими счетами**

* + Клиенты могут создавать новые счета (например, сберегательные, расчетные).
  + Возможность закрытия счета клиентом.
  + Просмотр баланса счета в реальном времени.

3.3 **Переводы средств**

* + Переводы между собственными счетами клиента.
  + Переводы на счета других клиентов банка.
  + Переводы на внешние счета (необходимо предусмотреть валидацию банковских реквизитов).
  + Ввод суммы перевода, выбор счета отправителя и получателя, подтверждение транзакции с использованием одноразового пароля (OTP).
  + Введение комиссий за переводы и расчет итоговой суммы.

3.4 **Получение оплаты от сторонних сервисов**

* + Публичный API для оплаты сервисов
  + Получение данных для проведения оплаты
    - номер карты клиента
    - дата действия карты
    - код CVC
    - сумма
    - комментарий для транцакции
    - номер счета получателя (интернет-магазина, зарегестрированного в банковской системе)

Ответ возвращает результат выполения операции перевода –

* + - результат операции – успешно/отклонено
    - номер транзакции (уникальный идентификатор перевода)
    - статус перевода (код ошибки, в случае неуспешной транзакции)

**код ошибки** в случае невозможности проведения оплаты:

* + - не действительная карта клиента
    - не совпадает CVC
    - не достаточно средств у клиента
    - не действителен счет получателя
    - техническая ошибка платежной системе
  + Проверка достатка средств для оплаты заказа при списывании со счета
  + Код проверки, который будет отправляться через емэйл (телеграм-бот) при введении оплаты

3.5 **История транзакций**

* + Пользователь должен иметь возможность просматривать историю транзакций по каждому счету.
  + Фильтрация по дате, сумме, типу транзакции (перевод, платеж, пополнение).
  + Возможность экспорта истории транзакций в формате CSV или PDF.

3.6 **Уведомления**

* + Уведомления в реальном времени при поступлении средств на счет или выполнении операции.
  + Возможность выбора типа уведомлений: по SMS, электронной почте или push-уведомлениям или через телеграм-бота

3.7 **Функционал для администраторов**

* + Возможность просматривать счет клиента
  + Просмотр и обработка пула от клиентов (список запросов, валидация, вопросы)
* **Требования к технологическому стеку**

4.1 **Язык программирования и фреймворки**

* + Язык программирования: Java 17 и выше.
  + Основной фреймворк: Spring Boot 3.x.
  + Безопасность: Spring Security.
  + Работа с данными: Spring Data JPA/Hibernate.
  + СУБД: MySQL.
  + Docker: для контейнеризации компонентов приложения.
* **Требования к интерфейсу**

5.1 **Внешние интерфейсы (API для фронтенда)**

* + **RESTful API**: приложение должно предоставлять REST API для взаимодействия с фронтендом. Поддержка основных HTTP методов (GET, POST, PUT, DELETE).
  + **Формат данных**: обмен данными осуществляется в формате JSON.
  + **Аутентификация и авторизация**: использование JWT для безопасности.
  + **Документация API**: подробное описание всех доступных эндпоинтов, параметров и ответов. Использование Swagger для автоматизации документации.

ОБСУДИМ ПОЗЖЕ

* **Предполагаемые спринты:**
  + **Спринт 1: Регистрация и аутентификация**
    - Создание структуры базы данных для пользователей и учетных записей.
    - Реализация API для регистрации и авторизации пользователей.
    - Имплементация работы с JWT токенами.
    - Создание документации API в Swagger.
  + **Спринт 2: Управление банковскими счетами**
    - Реализация API для создания, закрытия и просмотра банковских счетов.
    - Имплементация функционала просмотра баланса и истории транзакций.
  + **Спринт 3: Переводы и уведомления**
    - Реализация перевода средств между счетами.
    - Имплементация уведомлений о транзакциях.
    - Подключение системы двухфакторной аутентификации.
  + **Спринт 4: Отчетность и аналитика**
    - Реализация функционала отчетности для администраторов.
    - Генерация отчетов и аналитических данных.