

Разработка Системы Управления Банковскими Картами

Описание задачи:

Необходимо разработать backend-часть системы управления банковскими картами с использованием Java и Spring Boot. Система должна обеспечивать создание, управление и просмотр данных о банковских картах, а также выполнение операций между картами пользователя.

Каждая карта должна содержать следующие атрибуты:

- Номер карты (masked, хранится в зашифрованном виде)
- Владелец карты
- Срок действия
- Статус карты (Активна, Заблокирована, Истек срок действия)
- Баланс

Система должна предоставлять REST API для взаимодействия с картами, включая фильтрацию и分页ную выдачу.

Требования:

✓ Аутентификация и авторизация:

- Реализовать аутентификацию по email и паролю с использованием Spring Security и JWT.
- Поддержка ролей: **Администратор** и **Пользователь**.

✓ Функциональные возможности:

Администратор может:

- Создавать, блокировать, активировать и удалять карты.
- Просматривать все карты.
- Управлять пользователями.

Пользователь может:

- Просматривать свои карты (с параметризованным поиском и постраничной выдачей).
- Запрашивать блокировку карты.
- Совершать переводы между своими картами, если у него есть несколько карт.
- Просматривать баланс по своим картам.

API:

- Эндпоинты для CRUD-операций над картами.
- Операции перевода средств между картами пользователя.
- Параметризованный поиск и постраничная выдача списка карт.
- Валидация входящих данных.
- Детализированные сообщения об ошибках.

Безопасность:

- Данные карт хранятся в зашифрованном виде.
- Доступ к API ограничен ролями пользователей.
- Маскирование номера карты при отображении (например: **** * 1234)

Работа с БД:

- Использование PostgreSQL или MySQL.
- Настроить миграции базы данных через Liquibase.







Документация:

- API должно быть описано с помощью OpenAPI (Swagger UI).
- README с инструкциями по локальному запуску.

Развертывание и тестирование:

- Использование Docker Compose для dev-среды.
 - Liquibase для управления миграциями.
 - Юнит-тесты с покрытием ключевых бизнес-функций.
-

Оценка работы:

-  Соответствие требованиям
 -  Чистота и качество кода
 -  Безопасность хранения данных
 -  Обработка ошибок
 -  Проектирование системы (ООП, уровни абстракции)
 -  Наличие тестов
-

Технологии:

Java 17+, Spring Boot, Spring Security, Spring Data JPA(Hibernate), PostgreSQL/MySQL, Liquibase, Docker, OpenAPI (Swagger)