Project: Develop a Spring Boot backend application to facilitate offline payment processing.

Essential Features:

User Registration and Approval:

Users can register by providing a unique userId, the response will be a user secret which can be exchanged for tokens. All the apis must be secured using JWT tokens.
Users can enroll for offline payment functionality.
Admins review and approve/reject registrations.
Approved users must wait 15 minutes before using requested functionalities.

Wallet Management:

Users can add funds to their wallet via a REST API.
Users allocate a maximum of 5000 rupees for offline payments.

Offline Payment Codes:

Approved users with positive wallet balances can download 5 unique offline payment codes.

Offline Payment Toggle:

Users can switch offline payments on/off, providing GPS location when enabling.
A 15-minute cooldown period applies after disabling offline payments.

Payment Types:

System supports both online and offline payments.
Online payments require OTP validation for completion.
All Payments must be secured to detect concurrent modifications by @Version

Offline Payment Process:

Users initiate offline payments using a unique code(download before), GPS location, and vendor ID.

Flag transactions for potential fraud if:

GPS location changes exceed 500 kms within 5 hours.

Vendor ID exhibits unusual activity. (sudden burst of varying transaction prices)

Admin Oversight:

Admins review and approve/reject flagged transactions.

Amounts for rejected transactions return to the user's wallet.

Amounts for approved transactions transfer to the vendor's wallet.

Amounts for all flagged transactions temporarily move to the company's wallet.

Vendor Registration and Approval:

Vendors register through a dedicated API with a unique id and gps location codes.

Admins approve vendors before payment acceptance.

Vendors can transfer money to personal wallets from store wallets.

All the payments to the vendors must happen with 20km radius of the gps code used for registration.

Vendor can request offline payment from user by providing secret code (encrypt user gps + secret code) and vendor id.

Authentication:

Secure all the api endpoints using a spring Authentication, User details must be fetched strictly from SecurityContextHolder

Users can fetch an Authentication token valid for 5 mins using a secret key used while registration, if a token is expired user must renew the token

Auditing:

All the service calls should be saved into a separate database in an asynchronous way. Even if the main method fails audits should always be stored in the database. Using Spring Aspect oriented Programming to implement the above

Template generation Using freemarker/Velocity:

Generate notifications to the users in a friendly text using freemarker/velocity. This should be sent to the user in a batch process using Scheduling in Spring. Store all the user notifications in a separate table. Create a Separate Project (this should not be a spring boot web project) to process the notifications and store the results into a database table as a CLOB column.