20232022 Coding Challenge - Problem Statement

A teams need to have diagnostic tools to be able to monitor inaccuracies in data then may lead to redemptions not taking place.

Currently the process of redeeming bonds is highly dependent on the quality of the instrument data and timing of system events, hence users are monitoring books manually with a large reliance on tools such as e-mail and spreadsheets.

High Level Ask

The TeamX would like to build an application that will allow users to track when bonds are maturing so any issues can be identified and resolved. A particular focus should be given to those bonds past maturity.

What Is A Bond?

A bond is a type of investment security where an investor lends money to a company or government for a set period of time, in exchange for regular interest payments. When the bond reaches maturity (the time when the bond issuer must repay the original bond value to the bond holder), the issuer of the bond returns the investor's money.

** The term security is used interchangeably throughout this documentation and refers to a tradable financial instrument. **

What is a Position?

A position is the amount of a security which is owned by an entity/person. The key features of a bond include the following: ·

- Face value/Par value – the amount of money the holder gets back when the bond matures
- Interest/ Coupon – the amount of interest the bondholder will receive per payment, expressed as a percentage
- Maturity – this date refers to the final payment date of the financial instrument at which point the original money is returned to the investor

Managing Bond Life Cycle Events

Background

We have to be able to track bonds in our inventory and understand when bonds mature, so that the face value can be returned to the investor. Objectives to provide transparency on maturing bonds, in particular those bonds that are still on our books post maturity - any bond remaining on our books post maturity would indicate there is an issue with the redemption of that bond e.g. trade fail, mis-booking, systems failure to name a few.

-
  -
    - The TeamX would like to build an application that will allow users to track when bonds are maturing so any issues can be identified and resolved. A particular focus should be given to those bonds past maturity
    - The tooling is intended to be used by users to identify and investigate issues that would cause a bond to remain on our books post maturity allowing those issues to be addressed

Conceptual Brief

This will be tooling that aids users in identifying issues with bonds post maturity that are still on our books. The tooling should provide visibility on the trades linked to a bond which might help identify failed trades, mis-bookings, timing issues etc.

The minimum viable product would entail a UI to display the redemptions due and flag redemptions past due. There should be a dashboard and search mechanism.

The user should be able to run a report to view which securities are due redemption over a specific period of time. The standard process enables redemptions to be triggered automatically, this tool supports the investigation of why the standard process failed.

It is intended for the project to be delivered across a series of MVP’s which expand over time in complexity.

Expected deliverables:

Bonds APP

Bonds is a tool for bond redemption management. It increases the efficiency and clarity with which users can manage their books and clients.

Bonds API is written in Java with a Spring Boot backend and a H2 database.

The application should at a high level

-
  - Allow permissioned users to see bonds that are due to mature
  - Allow users to view bonds that have matured recently

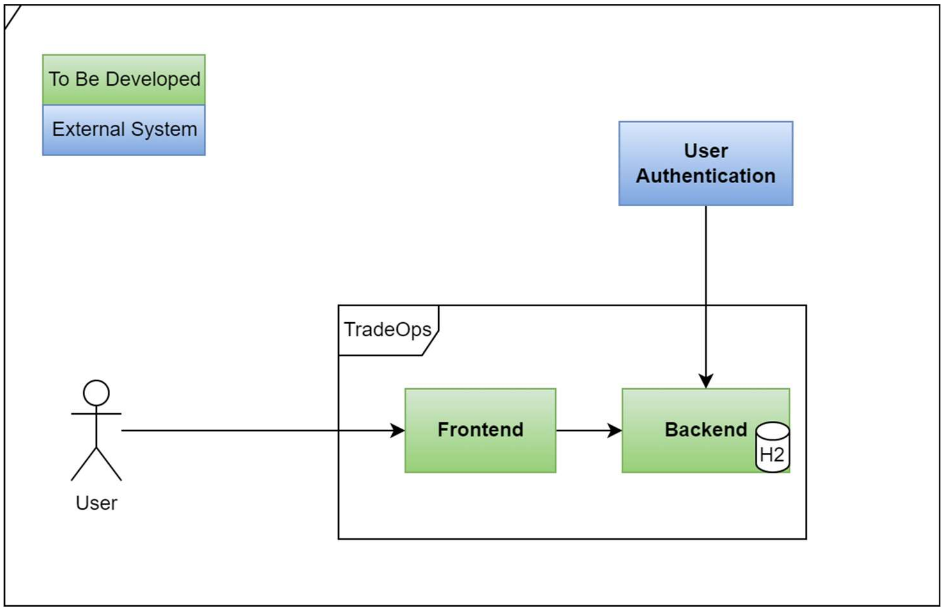
  - Show where almost mature bonds have been actioned by others to prevent duplication of effort and provide oversight
  - Optional allow uses to create custom list of bonds they are tracking

Who are the users of this application?

-
  - The users are members of the TeamX
  - The users will manage a number of books which contain trades and have to ensure the trade bookings are correct and accurate

Architecture

High Level Design:

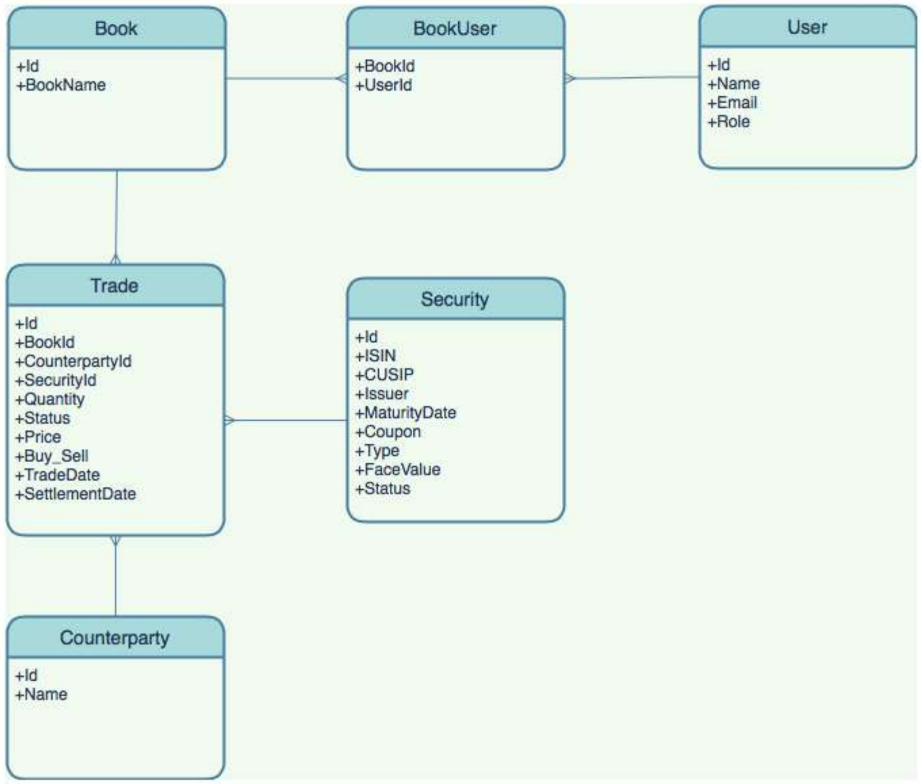


Data:

The main data entity the API handles are Book, Counterparty, Security, Trade and User. See the database diagram:

The database is currently populated based on pre-specified script. Moving forwards, we would like to integrate with another system to consume production bonds data (see Future features section below).

We will be using in-memory database which is H2 in this case for the case study.



Rest API:

Data is exposed to consumers via a REST API. There are two main controllers handling requests: SecurityController and TradeController.

SecurityController provides the following functionality:

-
    -
        - Get all securities
        - Get securities by ID, date range or corresponding user ID
        - Get all trades for a security
        - Update, create and delete a security

TradeController provides the following functionality:

-
    -
        - Get trade by ID
        - Retrieve security to which the trade belongs
        - Create and update a trade

Future Features:

-
  - Consume real-world data from the relevant bonds database/API
  - Implement more comprehensive error handling

Sample Repository

https://github.com/soumitra-soundankar/arrakis repo contains sample java-api and the react app. You can refer those while building **BONDS** application.

Points to be remembered

- Project needs to be built outside the DB network on your personal laptops
- Use the learning from the tech. curriculum (all modules)
- Assign roles within your team (Product Owner, Agile lead, UX Designer, UI Enggs., Fullstack Enggs., DevOps Enggs., more...)
- The projects will be evaluated based on the tech. curriculum learnings utilized and the MVP outcome