

# DB Coding Challenge Manual

## Helpful Index

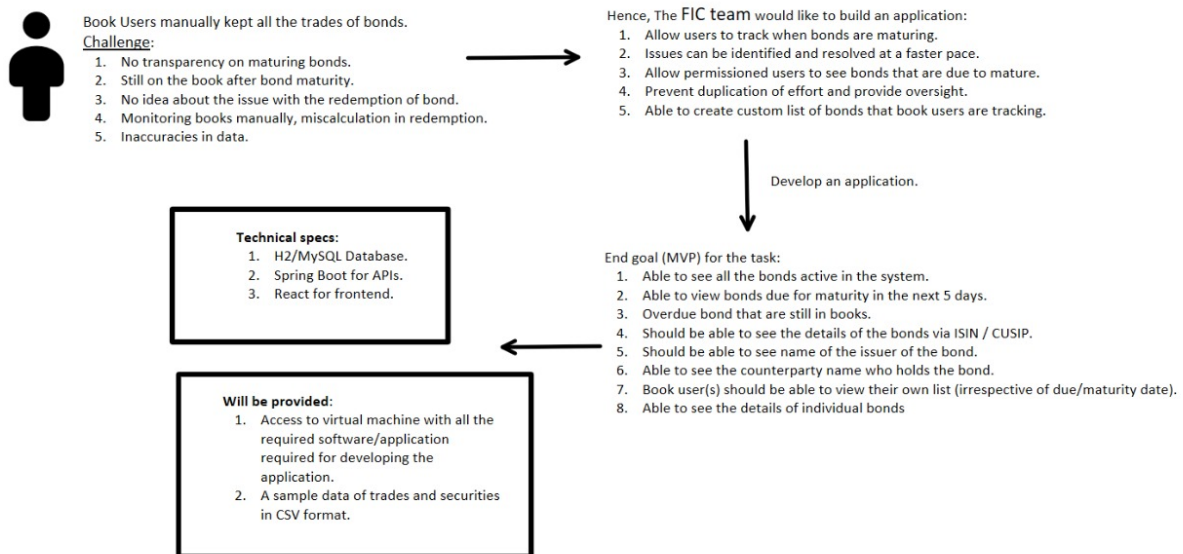
- For the idea of project flow, MVP, Architectural Diagram, Stories: [Project Brief](#)
- A guide for managing your four-day execution: [Suggested Approach](#)
- Details of resources each team will get to expediate application development: [What we will provide](#)
- For meaning of terms used in the course materials, refer this: [Glossary of Terms](#)
- For details of possible configuration, refer this: [Proposed Architecture](#)
- Description of the 4 -day program: [Coding Challenges](#)
- Get pre-defined responses for FAQs: [Note to trainers](#)

# Project Brief

## Project Arrakis Brief

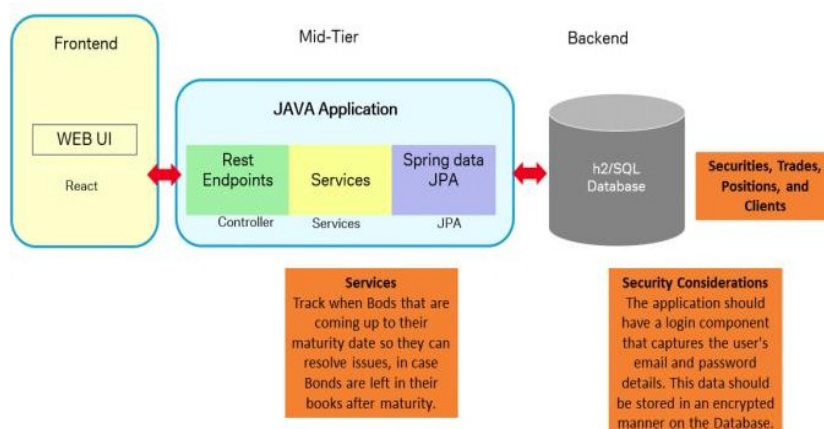
### Requirements

FIC Trading Operations require diagnostic tools to monitor data inaccuracies that could potentially prevent bond redemptions. The current redemption process heavily relies on the accuracy of instrument data and the timing of system events. Operations currently monitor portfolios manually, relying extensively on tools like email and spreadsheets.

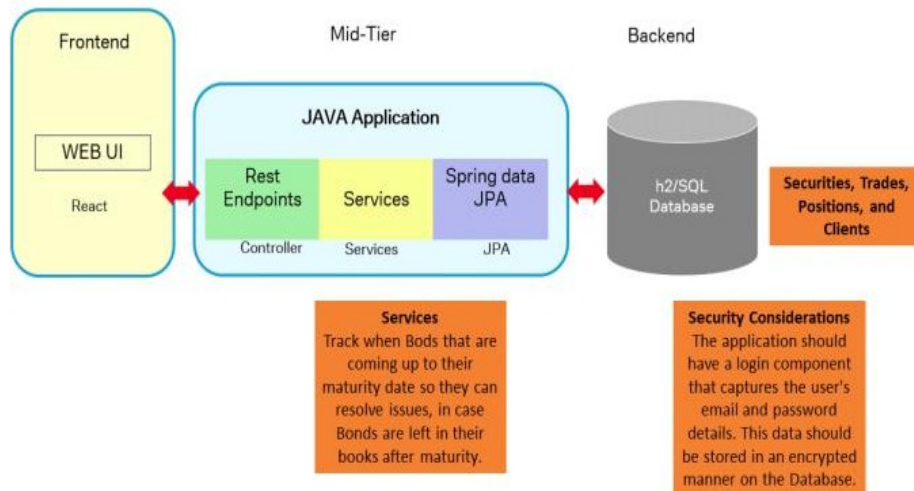


### Three possible configurations:

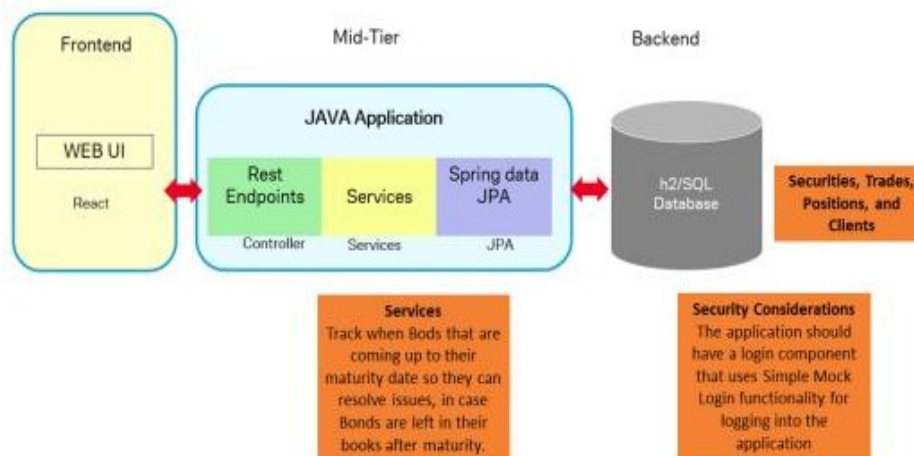
#### Option 1 - no security



### Option 2 - email/password (encrypted)



### Option 3 - Mock Simple Login (email/password)



### High Level Ask

The FIC team wants to develop an application for tracking bond maturity to promptly identify and address issues, with a specific emphasis on bonds that have already matured.

#### The application should at a high level:

- Enable authorized users to access information on bonds nearing maturity.

- Provide users with visibility into recently matured bonds.
- Display actions taken on almost mature bonds by others to avoid duplication and ensure oversight.
- (Optional) Allow users to create personalized lists of bonds they are monitoring.

#### **Who are the users of this application?**

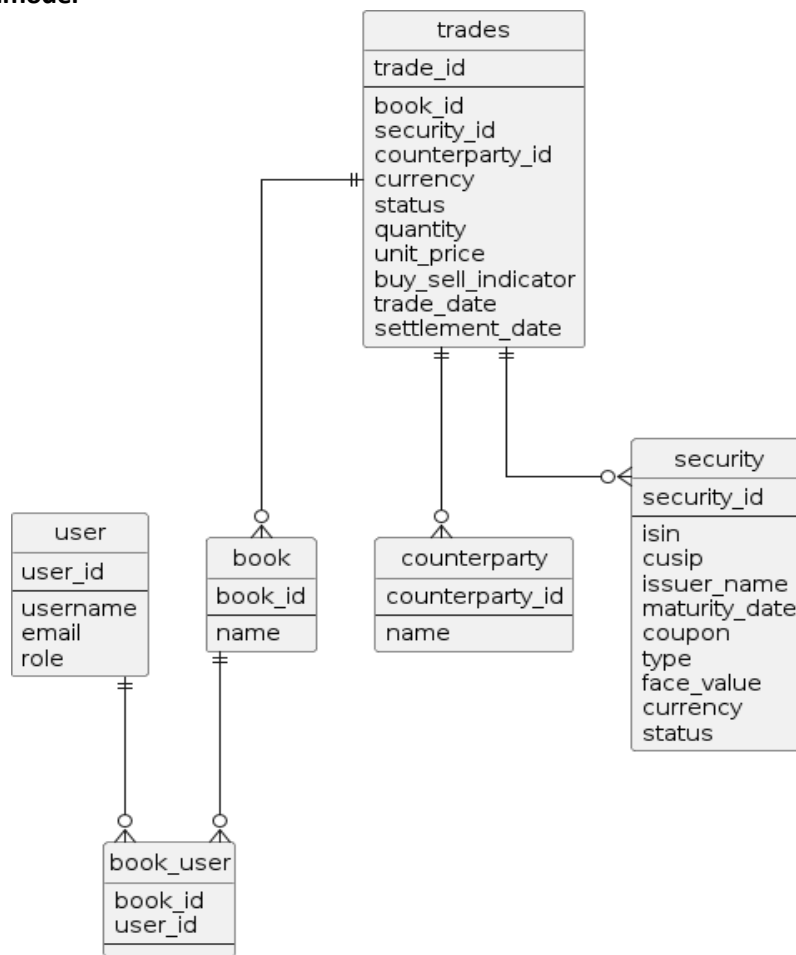
- Members of the FIC ops team are responsible for managing various books that include trades, ensuring accurate and correct trade bookings.
- Each user is assigned specific books, along with the trades associated with those books, to oversee.

#### **User Stories**

<b>Story</b>	<b>Comments</b>	<b>Acceptance Criteria</b>
As a user, I want to view all active bonds currently in the system to assess the portfolio.		Display all trades that have the status "OPEN"
To assist in investigating bonds that are still on the books post maturity, I want the ability to view bonds scheduled to mature within the next 5 days.	This implies within business working days	Display all trades that have a settlement date within the next 5 days. Do not display trades with settlement dates outside of this 5-day window.
To gain visibility into positions, I want the ability to see bonds scheduled to mature within the past and upcoming 5 days.		Display all trades that have a settlement date within the next or previous 5 days. Do not display trades with settlement dates outside of this 5-day range.
To uniquely identify a specific bond and its trades, I need to view the bond's ISIN or CUSIP.	ISIN - An International Securities Identification Number uniquely identifies a security. A CUSIP is a nine-digit numeric (e.g. 037833100 for Apple) or nine-character alphanumeric (e.g. 38259P508 for Google) code that identifies a North American financial security for the purposes of facilitating clearing and settlement of trades	The trade's ISIN and/or CUSIP should be displayed for user reference, allowing them to search for trades using the ISIN or CUSIP identifier.
To identify the bond issuer, I want to view the issuer's name associated with the bond.		The trade's issuer name should be visible to the user, who should also have the ability to search for trades based on the issuer's name.

To identify the owner of a bond, I want to view the name of the counterparty holding the bond.		The trade's counterparty should be visible to the user, who should also have the capability to search for trades based on the counterparty's name.
To manage my bond portfolio effectively, I want to view the bonds that I am accountable for.	This functionality necessitates the ability to assign a bond to a user and subsequently filter the data based on the assigned user.	Users should have the ability to assign bonds to themselves and view only those bonds assigned to them. They should also be able to search for unassigned bonds or bonds assigned to another user. Once a bond is assigned to a user, it cannot be reassigned by another user to themselves. An assigned user can choose to unassign themselves from a bond or assign it to another user as needed.
As a user seeking trading details, I want to access comprehensive information about individual bonds.		The interface should include fields such as book, maturity date, trade financials, trading currency, trade/settlement date, assigned user, and any other available database fields. Users should be able to search by these fields and customize which fields are displayed. Additionally, they should have the ability to rearrange the order of displayed fields and sort the dataset based on any field.

## Datamodel



## Definitions

### What Is A Bond?

A bond is a form of investment where an investor lends money to a company or government for a predetermined duration. In return, the investor receives regular interest payments. Upon maturity, the issuer of the bond repays the original investment amount to the bondholder. **Throughout this documentation, the term security is used synonymously and pertains to a tradable financial instrument.**

### What is a Position?

A position represents the quantity of a security owned by an entity or individual. The primary characteristics of a bond include:

- Face value/Par value: The sum returned to the holder upon bond maturity.
- Interest/Coupon: The interest amount paid to the bondholder per payment period, expressed as a percentage.
- Maturity: The date when the financial instrument's final payment is due, marking the return of the initial investment to the investor.

### Managing Bond Life Cycle Events

We need to monitor bonds in our inventory to ensure timely maturity and redemption of their face value to investors. Our objective is to provide transparency regarding maturing bonds, especially those that remain on our books after maturity. Any bond lingering past its maturity date indicates potential issues such as trade failures, mis-booking, or system malfunctions.

The FIC team aims to develop an application enabling users to track bond maturity dates effectively, with a specific emphasis on identifying and resolving issues associated with bonds that have passed their maturity date. This tool will empower operations personnel to investigate and address any underlying issues promptly.

### How to Approach the Project

This application will assist operations in identifying issues with bonds that remain on our books after maturity. The tool should offer visibility into the trades associated with each bond, aiding in the detection of trade failures, mis-bookings, timing discrepancies, and similar issues.

The basic solution will include a user interface (UI) for displaying upcoming redemptions and highlighting overdue redemptions. It will feature a dashboard and search functionality. Users will be able to query the system to view securities due for redemption within a specified timeframe.

### Agile Approach and Incremental Feature Delivery

The project is planned to be rolled out in incremental versions, progressively increasing in complexity over time. Your team should prioritize requirements that allow the solution to evolve through successive deliveries, beginning with fundamental features and advancing to more sophisticated functionalities. [This article] provides a comprehensive introduction to this approach.

### Working in Sprints



Your team should choose a limited number, possibly just one, feature to deliver within a Sprint. Aim to fully complete this item within the designated time frame, and then proceed to the next highest priority task. It's crucial to work in a manner that ensures you have functioning software to demonstrate at the end of each sprint. Avoid focusing exclusively on building one component, such as backend development, without also addressing frontend requirements to display the data.

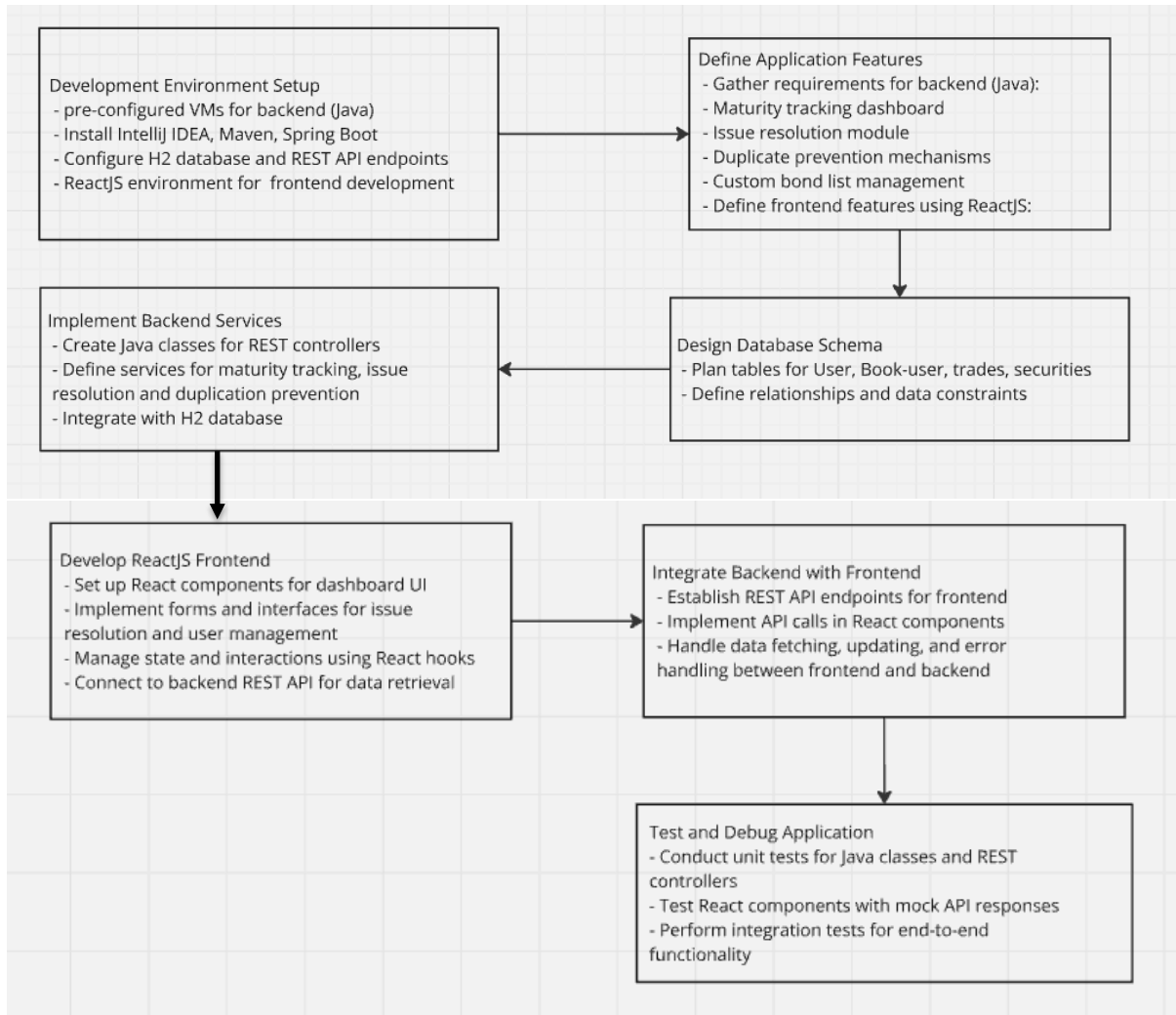
### **Working as a Team**

As a team, you will need to organize yourselves and develop different parts of the solution concurrently—such as the front end, back end, and database. However, these components are interdependent, requiring collaboration and coordinated effort. Utilize your version control system effectively to share code and prevent individual codebases from becoming isolated for extended periods, as merging and testing will become challenging. Pair programming can be beneficial to share knowledge and mitigate dependencies on specific team members.

## Suggested Approach

### Overview

Use this page as a guide for managing your four-day execution.



## **Approach**

- Teams are self-managing and taking breaks as needed.
- Maintain a Kanban board with columns for backlog, in progress, and done as a minimum requirement.
- Refer to this section for board access instructions.
- Conduct a standup each morning and a retrospective each evening before signing off; use the Kanban board in morning standups.
- Keep all code stored in a Git repository accessible to trainers.
- Regularly push code updates.
- Utilize pull requests for code review purposes.
- Decide on the number of Sprints (recommended minimum of three) per day, e.g., 1 per day or 2 per day.
- Use mob programming to enhance team understanding of code.
- Employ pair programming to expedite coding efforts.
- Develop and test the application locally on your LOD (Local Operating Device) before committing code to the feature branch.

## **Definition of Ready**

Typical criteria include:

- Understanding of the user story.
- Completion of acceptance criteria.
- Organized team (paired or mob programming).
- Testing approach defined, aligned with acceptance criteria.
- Understanding of business value.

## **Definition of Done**

Typical criteria include:

- Functional code that runs.
- Completion of tests.
- Achievement of acceptance criteria.
- Documentation, if required, is finished.

## What we will provide

To expedite application development, each team will receive the following resources:

**1. Pre-configured Development Environment:**

Each participant or team will receive access to a virtual machine that has been pre-configured with all the necessary software and applications required for the development of the bond tracking application. This setup will save participants time and effort in configuring their development environments and ensure consistency across all participants.

**2. Operating System:**

The VM will run a commonly used operating system, such as Windows.

**3. Integrated Development Environment (IDE):**

Popular IDEs such as IntelliJ IDEA or Visual Studio Code will be pre-installed to provide a seamless coding experience.

**4. Build Tools:**

Maven will be available for managing project dependencies and building the application.

**5. Database:**

A local instance of a database management system, such as H2, will be set up and configured for database interactions.

**6. Spring Boot Framework:**

The latest version of Spring Boot and its dependencies will be installed, allowing participants to leverage its features for rapid application development.

**7. Version Control System:**

Git will be installed and configured for version control, with integration to Bitbucket repositories as needed.

**8. Collaboration Tools:**

Access to collaboration tools such as Microsoft Teams or Trello/Jira will be provided to facilitate communication among team members.

**Our expectations include:**

1. Organizing the data you provide into a normalized structure within the database and prepare your own tables and insert data into those tables.
2. Incorporating security measures, such as implementing Email/Password, Simple Mock Login, or OAuth 2.0 credentials.

## Coding Challenge

### Problem Statement:

We aim to empower Operations with diagnostic tools to effectively monitor data inaccuracies that could potentially prevent redemptions from being processed. Currently, the redemption process for bonds heavily relies on the accuracy of Instrument data and the timing of system processes. As a result, Operations currently monitor books manually and rely on tools like email and spreadsheets.

### High Level Ask

The FIC team aims to develop an application enabling authorized users to track Bonds nearing maturity. This tool will facilitate Operations in identifying and resolving issues related to Bonds that remain unsettled past their maturity date. Key functionalities include:

- Providing visibility of Bonds approaching maturity.
- Displaying Bonds that have not been fully settled.
- Showing recently matured Bonds.
- Highlighting Bonds nearing maturity that have already been actioned to avoid duplication and ensure oversight.
- Serving as the initial step to initiate the redemption process.
- Optional feature:
- Allowing users to create customized lists of Bonds for tracking and monitoring purposes.

### Your Task

Your task is to develop a tool that assists Operations in identifying issues with bonds that remain on the books after their maturity date. The tool should provide visibility into trades associated with these bonds, aiding in the detection of failing trades, mis-booked trades, timing discrepancies, and other related issues.

### Minimum Viable Product Description

The minimum viable product (MVP) will include a user interface capable of displaying bonds due for redemption and highlighting those that should have been redeemed. It will feature a dashboard and search functionality to provide comprehensive overviews of events. Operations users should be able to generate reports to view securities due for redemption within specified time frames. The application should support the automatic triggering of redemptions under normal circumstances, allowing users to quickly identify and address any issues that arise.

### Your definition of Done is that you (all of which are important learning points)

- Develop a web app for Fixed Income & Currencies Technology.
- Components include a React frontend, Java SpringBoot midtier, and SQL (H2) backend.
- Use TDD, specifically in the midtier with tools like Postman.
- Visualize relational database data through tabular formats, highlighting key data points.
- Implement security measures.
- Utilize specified tools exclusively.
- Adopt agile methodologies.
- Ensure comprehensive test code coverage throughout development.

### **We would also like to see**

- Understand and apply the application's architecture, ensuring components like React frontend, Java SpringBoot midtier, and SQL (H2) backend are integrated effectively.
- Implement SOLID design principles to ensure scalability, maintainability, and extensibility of the codebase.
- Prioritize security considerations throughout development, including data encryption, authentication mechanisms, and secure API endpoints.
- Emphasize the importance of testing with comprehensive unit tests, integration tests, and end-to-end tests to validate functionality and performance.
- Govern development processes by adhering to established architectural guidelines, ensuring consistency and coherence across the project.
- Foster teamwork by encouraging collaboration, allowing each team member to contribute based on their expertise and fostering a cohesive working environment.

### **Who are the users of this application?**

- Application users belong to the Operations team within DB.
- Users manage multiple Books containing trades, ensuring accuracy and correctness in trade bookings.
- User registration requires manual association with specific Books they can access, treated as a necessary DBA task for delivering the MVP (Minimum Viable Product).

### **Managing Bond Life Cycle Events**

The application's core function is to track bonds in inventory and monitor their maturity dates. It aims to provide transparency on bonds nearing maturity, particularly those that remain on the books post-maturity. The tool serves as an investigative resource for Operations to analyze issues and discrepancies related to bond management.

### **Possible Application Look and Feel**

The application UI design should prioritize usability and functionality tailored to the needs of Operations. It must effectively display bonds nearing maturity, those that have matured, or expired. While fixed data will be initially uploaded for testing, the application should support viewing data from different date periods. Detailed information on individual bonds should be easily accessible for deeper analysis and investigation.

## MVP Stories

This section contains predefined responses for questions engineers may ask about the stories. These responses are standardized to ensure consistency in answers provided by trainers.

**Advisable to not share with participants.**

### MVP 1.0 – 1<sup>st</sup> Iteration of Product Building

Story	Comments	Value	Acceptance Criteria	Q&A
As a user viewing the portfolio, I want to access all currently active bonds within the system.		High		<ul style="list-style-type: none"> <li>• Sorting the data would be desirable but not essential.</li> </ul>
To aid in investigating bonds still on the books post-maturity, I want the ability to view bonds scheduled to mature within the previous and upcoming 5 days.	Bank standards dictate adherence to business days, but this necessitates further discussion.	High		<p>Should weekends be included in the count?</p> <p>- No.</p> <p>How would you prefer the data to be presented?</p> <p>- As clearly as possible, possibly in a table format with some highlighting.</p> <p>The data is outdated; their application should include a date picker.</p>
To provide visibility into positions, I want the ability to see bonds scheduled to mature within the past and upcoming 5 days.	I want the capability to review positions five days into the future and the past to analyze emerging patterns and trends.	High		<p>Should weekends be included in the count?</p> <p>- No.</p> <p>How should the data be displayed?</p> <p>- Clearly, perhaps in a table format with highlighting.</p> <p>The data is outdated; their application should include a date picker.</p>
To uniquely identify bond trades, I want to view both the ISIN and CUSIP codes of a bond.	ISIN - International Securities Identification Number for Bonds, determined by an external organization CUSIP - A nine digit numeric or nine character alphanumeric code that identifies a North American Financial Security	Medium		<p>How do you prefer the data to be presented?</p> <ul style="list-style-type: none"> <li>• - As clearly as possible, possibly in a table format with optional highlighting.</li> </ul>
To identify the issuer of the bond, I want to view information about the bond's issuer.		Low		
To identify the current owner of the bond, I want to view the client name associated with the bond.		Low		

To effectively manage my bond portfolio, I want to view bonds that are within the books I oversee.		High		
To comprehend the security, I want the capability to view the specifics of individual bonds.		Medium		
As a user, I want the system to ensure security by restricting access to bond data only to registered users.		High		<p>Do we require OAuth 2.0 for authentication?</p> <ul style="list-style-type: none"> <li>• - No, you can utilize methods such as email/password stored in your database, but passwords must be encrypted when stored.</li> </ul>



## Other typical questions

Question	Answer(s)
Can we utilize graphical charting tools?	<ul style="list-style-type: none"><li>• Yes, but it's important to clarify what information you intend to present and whether it enhances value.</li></ul>
Do we need to implement security like a login feature?	<ul style="list-style-type: none"><li>• Yes, as it is necessary to distinguish between different users.</li></ul>

## MVP 2.0 – Event and Lifecycle Management

Story	Comments
In order to prioritise Bonds that are overdue for action, as a user I want to be able to alerted to Bonds that have passed maturity but do not have a status of redeemed	
In order to redeem a bond as a user I want to be able to trigger a redemption on a bond coming due for maturity	this should record an action that redemption has been triggered for a given bond which should be for maturity visible to other users it is not expected to trigger an external event.
In order to not duplicate effort I need to see bonds in my portfolio which have already been triggered for redemption	Question - would this not mean that the bond has been redeemed then?
In order to not duplicate effort I need to see bonds in my portfolio which have already been triggered for redemption	
In order to prioritise Bonds that are maturing soon, as a user I want to be alerted to Bonds that are due today	
In order to check I have actioned all the bonds I should as a user I want to see the Bonds that have matured in the last 5 days	
In order to ensure I am managing my clients, as a user I want to see the number of Bonds a client is holding in the books I manage	
In order to ensure clients remain within their and the banks trading limits, as a user I want to see the current position a client holds	<ul style="list-style-type: none"><li>• positions can be positive or negative</li><li>• assumption: position is for the books the user has access to</li></ul>

## Glossary of Terms

Bond	A bond is a type of investment security where an investor lends money to a company or government for a specified duration, receiving regular interest payments in return. Upon reaching maturity, the bond issuer repays the original value of the bond to the investor.
Bond Maturity	In this project, the term "fixed income" is crucial and commonly refers to bonds, as they provide investors with predetermined payments throughout the bond's lifespan.
CUSIP and ISIN	Bond identifiers are unique identifiers for financial products traded on exchanges. They serve as distinct labels for stocks, bonds, or other tradable items between parties. While both identifiers are linked to exchanges, CUSIPs are more generalized, often allowing for multiple ISINs associated with a single CUSIP. These identifiers typically follow a hierarchical structure.
IIRC	If I recall correctly
IMHO	In My Humble Opinion
Position	<p>A position represents the quantity of a security owned by an entity or individual. The principal characteristics of a bond include:</p> <ul style="list-style-type: none"><li>• Face value or par value: The amount received by the bondholder at maturity.</li><li>• Interest or coupon rate: The percentage rate at which interest payments are made to the bondholder.</li><li>• Maturity date: The final payment date of the financial instrument.</li></ul>
Quantity	The number of units