# Software Requirements Document (SRD)

**Project name:** Stock Trading Application

**Client:** Trevor Banking Inc.

**Project manager:** Trevor Doto

**Date submitted:** June 2, 2025

**Document author:** Aditya Khajanchi

**Document status:** ☐ Draft ☒ Proposed☐Validated ☐Approved

**Document version:** 1.0

## TABLE OF CONTENTS



## REVISIONS

| Name | Date | Reason | Version |
|---|---|---|---|
| Aditya Khajanchi | June 2, 2025 | Proposed SRD | v1.0 |
| | | | |
| | | | |
| | | | |
| | | | |

# 1. Introduction

This document outlines the software requirements for a new mobile and web application designed to empower retail channel customers to independently manage their stock purchases and sales. The primary goal of this application is to enhance customer self-service capabilities within the financial institution. The application aims to provide a user-friendly and secure platform for basic stock trading, while offering a distinct and focused trading experience separate from existing bank account functionalities. The MVP focuses on essential trading features, scalability, and regulatory compliance for Canadian users.

# 2. Project Overview

## 2.1. Product Value

- **2.1.1. Enables Self-Service Trading:** Provides retail channel customers with a convenient platform to buy and sell stocks without direct broker intervention.
- **2.1.2. Optimized Broker Time:** Reduced the time brokers spend on low-profitability retail transactions, allowing them to focus on high-value clients.
- **2.1.3. Retains Customer Funds:** Offers an in-house trading solution to prevent customers from moving their investment funds to third-party trading applications.
- **2.1.4. Scalable:** Scalable architecture to accommodate future features and functionalities.

## 2.2. Success Metrics

The success of this project will be measured by the proportion of the bank's retail channel customer base (currently 150,000 customers) who frequently use the application, defined as logging in at least once or twice a day.

## 2.3. Product Scope

### 2.3.1. Inclusions

- **2.3.1.1.** Biometric login
- **2.3.1.2.** Integrated authentication with existing bank account credentials
- **2.3.1.3.** Check stock prices
- **2.3.1.4.** Place stock purchase/sell orders
- **2.3.1.5.** View portfolio and profits/losses
- **2.3.1.6.** View/export transaction history
- **2.3.1.7.** Important notifications (Purchase/sale updates)
- **2.3.1.8.** Help and FAQs
- **2.3.1.9.** Logout

#### 2.3.2. Exclusions

**2.3.2.1.** No access to brokers/admins for trading on behalf of customers

**2.3.2.2.** Core bank systems remain separate

### 2.4. Project Stakeholders

**2.4.1. Business Owner / Client:** Retail Channel of Trevor Banking Inc.

**2.4.2. End Users:** Retail Customers of Trevor Banking Inc.

**2.4.3. Support Teams:** Securities Branch, Technical Implementation Team

**2.4.4.** Analytics Team at Trevor Banking Inc.

## 3. System Overview

The application should provide retail customers with a user-friendly tool to check stock prices, execute buy/sell orders, view portfolio performance, and review transaction history. It should incorporate a simple, intuitive UI/UX that caters to a broad age range, from 18 to 90 years old and should follow the bank's color scheme of red and white. The simplistic design should include an industry-standard bottom-navigation bar for ease of use. Initially, data can be sourced from free, open-source providers like Yahoo Finance and Alpha Vantage, with support for integration with the bank's existing premium data partners.

## 4. Requirements

### 4.1. Functional Requirements

#### 4.1.1. Authentication and Authorization

**4.1.1.1. Service Opt-in:** Access to the application should require customers to have explicitly opted in for the service.

**4.1.1.2. Existing Bank Credentials Login:** The application should allow users to log in using their existing bank account credentials.

**4.1.1.3. Biometric Login:** The application should offer a biometric login option (e.g., fingerprint, face ID) that users can toggle on/off after their initial login.

#### 4.1.2. Stock Management

**4.1.2.1. Stocks List:** The application should display a list of stocks for users to monitor.

**4.1.2.2. Stock Detail View:** The application should provide a dedicated page for each stock, displaying general information and current trading options.

**4.1.2.3. Buy Stock:** The application should enable users to purchase stocks.

**4.1.2.4. Sell Stock:** The application should enable users to sell stocks.

#### 4.1.3. Portfolio Management

**4.1.3.1. Personal Portfolio View:** The application should display the user's personal investment portfolios (TFSA, CHQ, SAV) and show how much money is invested in each portfolio.

**4.1.3.2. Portfolio Performance Tracking:** The application should allow users to view the performance of their portfolios.

#### 4.1.4. Transaction History

**4.1.4.1. Detailed Transaction History:** The application should display a detailed transaction history with the following fields:

**4.1.4.1.1.** Ticker
**4.1.4.1.2.** Timestamp
**4.1.4.1.3.** Buy/Sell type
**4.1.4.1.4.** Associated Account/Portfolio
**4.1.4.1.5.** Purchase Price
**4.1.4.1.6.** Volume
**4.1.4.1.7.** Total Holding for the ticker.

**4.1.4.2. Transaction Search/Filter:** Users should be able to search and filter transaction history for up to the last 18 months.

**4.1.4.3. Transaction History Export:** Users should be able to download transaction data for the last 18 months in standard formats (CSV, JSON, XLSX, PDF). Data older than 18 months, up to 7 years, can be downloaded only as PDF statements.

#### 4.1.5. Fund Management

**4.1.5.1. Transfers:** Users should be able to move money to and from their existing bank chequing and savings accounts within the same bank.

#### 4.1.6. Notifications

**4.1.6.1. Real-time alerts:** The application should send important alerts when a purchase or sale transaction is completed, with an option to toggle specific alert types such as push, email, SMS.

#### 4.1.7. Data Export for Analytics Team

**4.1.7.1. Transactions Endpoint:** The application should have its own API to allow the bank's analytics team to fetch transaction history data of all customers for internal portal dashboards.

#### 4.1.8. General Settings

**4.1.8.1. App Preferences:** The application should provide the user with controls to manage notifications and toggle biometrics.

**4.1.8.2. Help Link:** The application should provide a link to a Help section.

**4.1.8.3. FAQs:** The application should provide access to Frequently Asked Questions.

**4.1.8.4. Logout Functionality:** The application should allow users to securely log out.

### 4.2. Non-Functional Requirements

#### 4.2.1. Performance and Availability

**4.2.1.1. Data Refresh Rate:** Stock price data should refresh every 30 seconds.

**4.2.1.2. Uptime:** The application should maintain an uptime of approximately 98%.

**4.2.1.3. Maintenance Window:** Maintenance activities should be scheduled during off-market hours.

#### 4.2.2. Usability

**4.2.2.1. UI/UX:** Simple enough for ages 18–90; bottom navigation bar; follow brand palette (Red, White).

**4.2.2.2. Responsive Design:** Native mobile and web support.

**4.2.2.3. Navigation:** The primary navigation bar should be located at the bottom of the page with the following tabs to switch between pages:

**4.2.2.3.1.** Home

**4.2.2.3.2.** Stocks List

**4.2.2.3.3.** Buy/Sell

**4.2.2.3.4.** Portfolio

**4.2.2.3.5.** Settings

#### 4.2.3. Security & Compliance

**4.2.3.1. Security Standards:** The application should follow standard industry practices and rules for privacy, authentication, and data security.

**4.2.3.2. Canadian Regulations:** The application should comply with standard regulatory rules and guidelines applicable to financial services in Canada, as it is designed for customers with Canadian accounts.

**4.2.3.3. Geo-Independence:** No geo-locking should be implemented, allowing customers to use the app while traveling.

**4.2.3.4. Access Restrictions:** There should be no admin components, and brokers should not be able to act on behalf of customers under any circumstances.

**4.2.4. Data Sources**

**4.2.4.1. MVP Data Source:** For the Minimum Viable Product (MVP), the application should use free open-source data providers (e.g., Yahoo Finance, Alpha Vantage) for stock data.

**4.2.4.2. Final Release Data Source:** For the final release, the application should have support for integration with the bank's existing premium data partners.

**4.2.5. Data Validation**

**4.2.5.1. Market Hours Validation:** The application should enforce strict validation for activity only during Canadian market hours.

**4.2.5.2. Funds Validation:** The application should prevent users from buying a stock without sufficient funds.

**4.2.5.3. Holdings Validation:** The application should prevent users from selling a stock they do not hold.

**4.2.6. Capacity**

**4.2.6.1. User Load:** The application should be able to support 150,000 customers logging in daily (once or twice) and handle associated transaction volumes.

**4.2.7. Compatibility**

**4.2.7.1. Mobile Platforms:** The application should be available on iOS and Android platforms.

**4.2.7.2. Web Browser Support:** The web application should support all common web browsers (Chrome, Edge, Firefox, Safari).

**4.2.8. Scalability**

**4.2.8.1. Scalable Architecture:** The application architecture should be designed to be scalable, allowing for the absorption of more features and functionalities in future releases.

**5. Application Pages**

**5.1. Home Page (Landing Page):** The initial entry point for users.

**5.1.1.** Overview of portfolio value, recent activity.

**5.2. Stocks List Page:** Browse and search stocks.

**5.2.1.** Clicking on a stock navigates to the individual Stock Detail Page.

- **5.3. Stock Detail Page:** Presents general information about a selected stock.
  - **5.3.1.** Includes options to initiate a "Buy" or "Sell" transaction, leading to the Buy/Sell Page.
- **5.4. Buy/Sell Page:** Transactional interface to facilitate stock purchase or sale orders.
  - **5.4.1.** User input required:
    - **5.4.1.1.** To choose a ticker from a pre-populated list (if not redirected from Stock detail page)
    - **5.4.1.2.** To specify number of stocks to buy/sell.
    - **5.4.1.3.** Biometric authentication to execute transaction.
- **5.5. Personal Portfolio Page:** Provides an overview of the user's investments.
  - **5.5.1.** Displays how much is invested in different portfolios (e.g., TFSA, CHQ).
- **5.6. Transaction History Page:** Shows a detailed record of past transactions.
  - **5.6.1.** Includes Ticker, Date, Buy/Sell type, associated Account/Portfolio, Purchase Price, Volume, and Total Holding for each ticker.
  - **5.6.2.** Users should be able to download transaction data for the last 18 months in standard formats (CSV, JSON, XLSX, PDF).
  - **5.6.3.** Data older than 18 months and up to 7 years, can be downloaded as PDF statements.
- **5.7. General Settings Page:** Allows users to log-out, manage app preferences, access FAQs, and help section.

6. **High-Level User Flow**
   - **6.1.** Login -> Browse Stocks -> View Stock Details -> Buy/Sell Stocks -> Logout
   - **6.2.** Login -> Browse Stocks -> View Stock Details -> Logout
   - **6.3.** Login -> Buy/Sell Stocks -> Logout
   - **6.4.** Login -> Monitor Portfolio -> Logout
   - **6.5.** Login -> Manage Funds -> Logout
   - **6.6.** Login -> Review Transactions -> Logout
   - **6.7.** Login -> Adjust Settings -> Logout

## 7. Use Cases

| Use Case | Basic Flow |
|---|---|
| **As a user, I want to buy stocks from my portfolio.** | 1. User logs into the application using existing bank credentials or biometric login.<br>2. User navigates to the "Buy/Sell" tab and selects a stock from the list or search.<br>3. User enters the amount and confirms the purchase, authenticates with biometrics, after which a confirmation message is shown and the transaction is recorded. |
| **As a user, I want to view my transaction history.** | 1. User logs into the application and navigates to the *Portfolio* tab.<br>2. User selects the *Transaction History* option to view all previous buy/sell activities.<br>3. User filters results by date or account type and optionally exports the data in CSV, XLSX, JSON, or PDF format. |
| **As a user, I want to check the current price and performance of stocks so that I can make informed investment decisions.** | 1. The user opens the app and navigates to the *Stocks List* section.<br>2. The user searches for a specific stock by entering its name or ticker symbol.<br>3. The app retrieves real-time data from a market data API.<br>4. The app displays key metrics associated with the ticker.<br>5. The user views data related to the ticker and makes their decision. |

## 8. Timeline

| Milestone | Date |
|---|---|
| **MVP Demo** | September 2025 |
| **Final Release** | Q2 2026 |

9. **Future Enhancements**
   - 9.1. **Stock Watchlist/Favorites:** The application shall support the creation of personalized stock watchlists or favorite lists.
   - 9.2. **Advanced Trade Types:** The application shall support additional trade types (e.g., stop-loss orders).
   - 9.3. **External Fund Addition:** The application shall allow users to add money from other bank accounts, wallets, or cards.
10. **Appendices**
    - 10.1. **Glossary**
      - 10.1.1. **MVP (Minimum Viable Product):** Initial version of the app with essential functionalities.
      - 10.1.2. **API (Application Programming Interface):** Interface for data exchange between the app and the bank's analytics systems.

***End of Document***