Stock News - Norbert

1. Basic Elements of News Feed

1.1 News Article Display

The system should provide financial news stories related to stocks, market condition updates, potential events and user-submitted articles. Each news story must carry a title, publication source, date and time of release, and a summary. The system needs to create a visual distinction between read and unread articles.

1.2 News Feed Organization

The system shall organize news articles in reverse chronological order by default (newest first). The system shall support categorization of news by topics (e.g., Stock News, Company News, Market Analysis, Economic News, Functionality News). The system shall support text-based search functionality for finding specific news content.

1.3 News Feed Refresh

The system is set to update the news feed automatically after a set interval, as long as it is running. The system needs to provide a visual indication when loading new articles is being done. The system must allow the user to update the news feed manually.

2. Reading News Stories

2.1 Article View

The system is written to allow the user to choose and view the full content text of any news item. The system must display the whole article in proper format. The system shall provide a way to allow users to return from the article view to the news

feed with ease. The system needs to provide access to relevant stocks mentioned in articles via an integration with the Stock System.

3. User-Generated Content

3.1 Article Submission

The system shall allow authenticated users to submit articles for publication. The users will be presented with a submission form where they must provide a title, content, summary, and topic for submitting an article. The system shall allow users to specify related stocks for their submitted articles. If such stock doesn't exist, the system shall provide the user with a confirmation pop-up about the existing stocks. The user can continue without using an existing stock and the system will automatically create that stock. The users shall have an option to preview the article before submitting. Upon submission, the system shall provide a confirmation message.

3.2 Article Review Workflow

The system shall implement a reviewable form with status tracking (Pending, Approved, Rejected). Submitted articles shall remain in "Pending" status until reviewed by an administrator. The system shall only display approved articles in the main news feed.

3.3 Moderation and Admin Control

The system shall provide an administrative interface for managing user-generated content. Administrators shall be able to approve, reject, or delete user-submitted articles. The system shall display a preview mode for administrators to review articles before making decisions. The system shall provide filtering of user articles by status and topic categories. The system shall display a prompt before deleting for confirmation.

4. User Authentication

4.1 Login and Access Control

The system shall provide login functionality for users and administrators. The system shall restrict access to article submission features to authenticated users only. The system shall restrict access to administrative features to users with moderator privileges. The system shall provide logout functionality to clear user session data.

5. Performance and Accessibility

5.1 Accessibility

The system must allow text scaling to improve readability. The system shall adhere to the app's global accessibility standards.

5.2 Performance

The system shall load news article summaries quickly, with full article content loading on demand. The system shall efficiently cache news to minimize redundant requests. The system shall provide appropriate error handling and user feedback during loading failures.

6. State Management

6.1 Empty States

The system shall display appropriate empty state indicators when no articles are available. The system shall provide visual feedback during loading operations.

6.2 Error Handling

The system shall display appropriate error messages when operations fail. The system shall gracefully handle network and data access failures.

Profile - Bianca

- This application relies on a function of the TEAM 7UP to provide the account-related user data, mainly the CNP.

guest = unregistered user

When creating an account, it should be creating using the CNP(it is taken from the user's banking account and it is unique for each user), the user shall be provided by the system with a username which is initially a predefined name (random name from a list - the username is not necessarily unique for each user).
---- For every NEW user the profile image will be empty, the description box will be empty and their list of stocks will be empty.

The user's profile:
The system shall allow the user to update their profile information as follows:

- the username - it has to be between 8-24 characters long and it has to contain only letters and numbers
- the profile picture - it has to be a PNG file
- the description - a block of text that can have between 0 and 100 characters
- the option to have a hidden profile - if this option is enabled the profile page (profile picture, description and the user's stocks) will be hidden from other users. Only the username will be accessible and visible
- the option to become "admin" - once a user becomes admin he cannot undo it - In order to become admin the system will ask for a password (password: BombardinoCrocodilo). The system will compare the hash of this password with a saved hash and if they match the user should be promoted to an "admin".

Admin:
The admin will have the ability to remove articles from the News Page.

The system will give the users the ability to see other users' profile (image, username, description and stocks). They can modify their account but not others'. They can only view other users' accounts without any modification - only if their profile is not set to "hidden".

Stocks:

Besides the other information about the user (username, image and description), the system will give the user the ability to see his stocks as a list (in no specific order). Each element (stock) will have the following information:

- the symbol of the stock. label
- the name of the stock. label.
- the quantity the user bought. label.
- the current value of the stock. label.
- a button with the text Enter" which if it is clicked, the system will let the user view the stock page for that specific stock

There will also be a button with the text “Back” that if it is clicked, the system will let the user go back to the previous page.

Gem store - Ana

- This requirement interacts with TEAM 7UP to retrieve the account list and request subtraction or addition of funds to an account.
- The Gem Store shall allow registered users to purchase and sell in-app currency (Gems) for real money. These Gems can be used later to acquire stocks.
- Guest users shall only be allowed to view Gem Deals without the ability to buy or sell Gems.

Buying gems:

- Users shall scroll through a list of Gem Deals, which can be permanent or special (available for a limited time).
- Each Gem Deal shall display:
  - A title.
  - The number of Gems included.
  - The price in real money.
- After selecting a deal, the user shall be prompted to choose a bank account from which the money will be deducted.
- A “BUY” button shall finalize the transaction once a bank account is selected,

the user's account being updated with the purchased Gems.

Selling gems:

- The system shall display the exchange rate: 100 Gems = 1 Euro.
- Users shall input the number of Gems they wish to sell.
- The system shall validate that:
  - The input is greater than 0.
  - The user has sufficient Gems to sell.
- If the input is invalid, the system shall display a specific error message.
- If the input is valid, the user shall be prompted to choose a bank account where the real money will be deposited.
- A "SELL" button shall finalize the transaction once a bank account is selected.

Gem Deals:

- The system shall display a list of Gem Deals in a scrollable container.
- Special Gem Deals shall be highlighted with a label "Special offer!" and shall expire after a set period.
- Expired special deals shall be replaced with new ones after a cooldown period.

Bank Account Selection:

- When buying or selling Gems, the user shall be prompted to select a bank account from a modal window.
- The modal shall list the user's bank accounts and require a selection to proceed.

Back Button:

- A back button shall be shown in the top-left corner of the Gem Store page.
- The button shall be an arrow pointing to the left.
- Pressing the button shall navigate the user back to the home page.

Home Page - Riky

- The homepage should also enable users to navigate to the Profile, Store,

History, Create Stock, News.

- The homepage shall display the following:
  - A button displaying the text “Profile”, which if clicked redirects to the profile page.
  - A button displaying the text “Store”, which if clicked redirects to the gem store page.
  - A button displaying the text “History”, which if clicked redirects to the transaction log page.
  - A button displaying the text “Create Stock”, which if clicked redirects to the create stock page. This button is only displayed for registered users.
  - A button displaying the text “News”, which if clicked redirects to the news page.
  - If the user is a Guest: A button displaying the text “Create Profile”, which if clicked, the user’s cnp will be added into the database, becoming a User.
  - The button described above will be shown ONLY if the user using the app is a Guest.
- The homepage of the Stocks App should allow all users to view stock details, by displaying a comprehensive list of all available stocks, each showing:
  - a stock symbol (a maximum of four uppercase letters),
  - the full stock name,
  - the latest available price,
  - and the percentage change relative to the last recorded value in the history if the history contains at least one recorded value.
  - All users should be able to search for stocks by:
    - symbol, a text input, containing alphabetical characters (A-Z), with a maximum of 4 characters.
    - name, a text input, containing alphabetical characters (A-Z), with a maximum of 50 characters.
  - Allow all users to have filtered stocks, and sort them by: (a drop-down menu containing label texts)
    - “Name” (A-Z or Z-A),
    - “Price” (ascending or descending),
    - “Percentage Changed” (ascending or descending)
  - A separate section of the page should display the registered user’s favorite stocks, allowing them to mark or unmark stocks as favorites.
    - The registered users will be able to mark / unmark stocks as “Favorite”

        - The Guests will not be able to mark / unmark stocks as “Favorite”

Stock System (Page) - Iosua

- This requirement utilizes a function from TEAM CtrlAltDefeat, mainly the broadcast of an event (buying or selling a stock) to the friend list.

The system should provide users with information regarding a specified stock:

- Guests and Registered Users should be provided with:
    - The ability to go back to the previous page:
        - This will be achieved through a button with the text “Back”, which if clicked redirects the viewer back to the previous page (the system saves a history of the previous pages).
    - The symbol of the stock. A label.
    - The name of the stock. A label.
    - A button containing the username of the author of the stock, which if clicked redirects the viewer to the profile page of the author user.
    - The stock price history up to 30 values (including the current price) in the form of a graph.
        - If the stock does not have a history of prices, only the current price, the graph should be empty.
    - The current price of the stock, followed by the text " Gems". A label.
    - If the stock has a history of at least one value, display the change in the value of the stock (in percentage, whole numbers) [(current stock / last history stock value) * 100]. A label.
- Registered Users should additionally be provided with:
    - A button:
        - If the user does not have the stock saved as a favorite:
            - Displaying an image of an empty star with a yellow outline.
            - when the button is clicked, save the stock as a favorite for the user and update the button.
        - If the user does have the stock saved as a favorite:
            - Displaying an image of a filled star colored yellow.
            - when the button is clicked, save the stock as not a favorite for

the user and update the button.

- A button displaying an image of a bell, which if clicked redirects the viewer to the alarm manager page for this stock.
- The number of gems the user has. A label.
- A number input to select the amount of stocks to buy or sell. The default value is 0.
- A button displaying the text “Buy!”, which if clicked, validates if the user has the necessary Gems to buy (n * current stock price), where n is the value entered in the input above, should be equal or lower than user gems.
    - If requirements are met, the system will subtract the discussed amount (n * current stock price) from the user’s gems and add to their current stocks the stock bought by the amount selected. The system should add the current stock price to the stock price history and generate a new current stock price by the following formula Maximum between (current stock price - random number between 0 and 20) and 50. After the price has been changed, run the alerts system bounds check for this stock. The system will save a new transaction in the transacrion history composed of stock symbol, stock name, transaction type BUY, n, current stock price, n * current stock price, the date at which the transaction was executed, user.
        - Broadcast the action to the friend-list through TEAM CtrlAltDefeat
    - Otherwise, display a popup warning with the text “Invalid number input selected”.
- A button displaying the text “Sell!”, which if clicked, validates if the user has the necessary Stocks of this type (count of user stocks of this type > n).
    - If requirements are met, the system will subtract the discussed amount and type (n * current stock price) from the user’s stocks and add to the user’s gems (n * current stock price). The system should add the current stock price to the stock price history and generate a new current stock price by the following formula Maximum between (current stock price + random number between 0 and 20) and 50. After the price has been changed, run the alerts system bounds check for this stock. The system will save a new transaction in the transacrion history composed of stock symbol, stock name,

transaction type SELL, n, current stock price, n * current stock price, the date at which the transaction was executed, user.
        - Broadcast the action to the friend-list through TEAM CtrlAltDefeat
    - Otherwise, display a popup warning with the text “Invalid number input selected”.

Alert Management System - Rafa

All the following are only applicable to registered users.
The system should allow users to set alerts on stock prices and notify users when a stock’s price has gone outside a bound interval.

- On startup, the system should provide the user with a list of triggered alerts. An alert is triggered when the price of the stock on which the alert is set goes outside a bound interval set by the user.
- The system will display:
    - A list of all triggered alerts, containing the following:
        - The alert label.
        - The stock name on which the alert is ser.
        - The current price of the stock.
        - A button with the text “Enter” that redirects to the stock page for the respective stock.
    - A button displaying the text “Close” that redirects to the Main Page.

The user can access the alert management system via the stock page system.
The system should provide a user with the following for a specific stock:

- A list of all saved alerts composed of the following:
    - The alert label.
    - The stock name on which the alert is set.
    - The lower bound limit on the price of the stock.
    - The upper bound limit on the price of the stock.
- The system should allow users to remove alerts via a button next to each alert with the text “Remove”.

- The system should allow users to create alerts in the following manner:
    - The system should display the following for alert creation:
        - The alert label. A text input is limited to 32 characters.
        - The lower bound limit on the price of the stock. A number input with default value 0.
        - The upper bound limit on the price of the stock. A number input with default value 0.
        - A button displaying the text “Create”, which if clicked validates the input of both text inputs to be above 0, otherwise displays a popup alert with the text “Number input value must be above 0”, and validates if the lower bound input value is lower than the upper bound input value otherwise displays a popup alert with the text “The lower bound value must be lower than the upper bound value”. The stock is automatically assigned with the stock of the stock page that initialized the alert page. The system should save the new alert with the data above and update the alert list.

Create Stock - Ionut

This page allows registered users and moderators to define and create a new stock entry by specifying details such as name, category, price, and initial quantity.

Requirements:

User Access Control:

- Guest users cannot create stock entries.
- If a guest accesses this page, display a message restricting stock creation to registered users.
- Hide or disable form inputs for guests to prevent submissions.

Stock Creation Form:

Fields:

- Stock Name (Max 20 characters, only letters & spaces)
- Stock Symbol (Max 5 characters)
- Author CNP (Max 13 characters, only numbers)

Buttons:

- Create Stock: Validates and stores data in the database via a wrapper.
- Back : Allow the user to go back to the home page.

Validations:

- Prevent invalid input (negative values, missing fields, or exceeding limits).
- Real-time validation with error messages.

Feedback:

- Display success/error notifications upon submission.

UI Considerations:

- Ensure a user-friendly and responsive design following the application’s theme.

Transaction Log System - Denis

- The application shall allow users to view their stock trading transaction history, displaying details for each transaction including:
    - stock symbol, label.
    - stock name, label.
    - transaction type (BUY/SELL), label.
    - amount, label.

    - price per stock, label.
    - total value, label.
    - transaction date. label.
    - transaction author.
- Users shall be able to filter transactions based on:
    - stock name,
        - the system will display a text input and a button displaying “Search”, which if clicked will filter the list of transactions based on the value of the input, meaning that the name of the stocks in the list should contain the value found in the text input.
    - transaction type,
        - the system shall display a drop-down menu with the following entries: BUY, SELL. The default value is all. On click, the list will be list will be filtered such that all the elements of the list have the transaction type the saved value as the value selected in the dropdown.
    - total value range,
    - and date range,
- The system should validate that:
    - the transaction type is either BUY or SELL,
    - start date is prior to the end date,
    - that numeric values are entered for the amount, price per stock and total value.
- Additionally, the system shall allow users to sort the transaction history by:
    - date (oldest to newest or vice versa),
    - stock name (alphabetical ascending or descending),
    - total value (ascending or descending),
- Updates made instantly when the sorting option is selected.
- Users shall have the option to export transaction data in CSV and PDF formats, with the ability to specify a date range;
    - if no transactions are found within the specified range, the export will be prevented, and a message will notify the user.
- The application must perform with high efficiency, loading the transaction history page within 2 seconds on a 4G connection for 95% of requests, and completing filtering or sorting operations within 1 second for up to 10,000 transactions. - - - Export operations shall complete within 5 seconds for datasets containing up to 50,000 transactions.