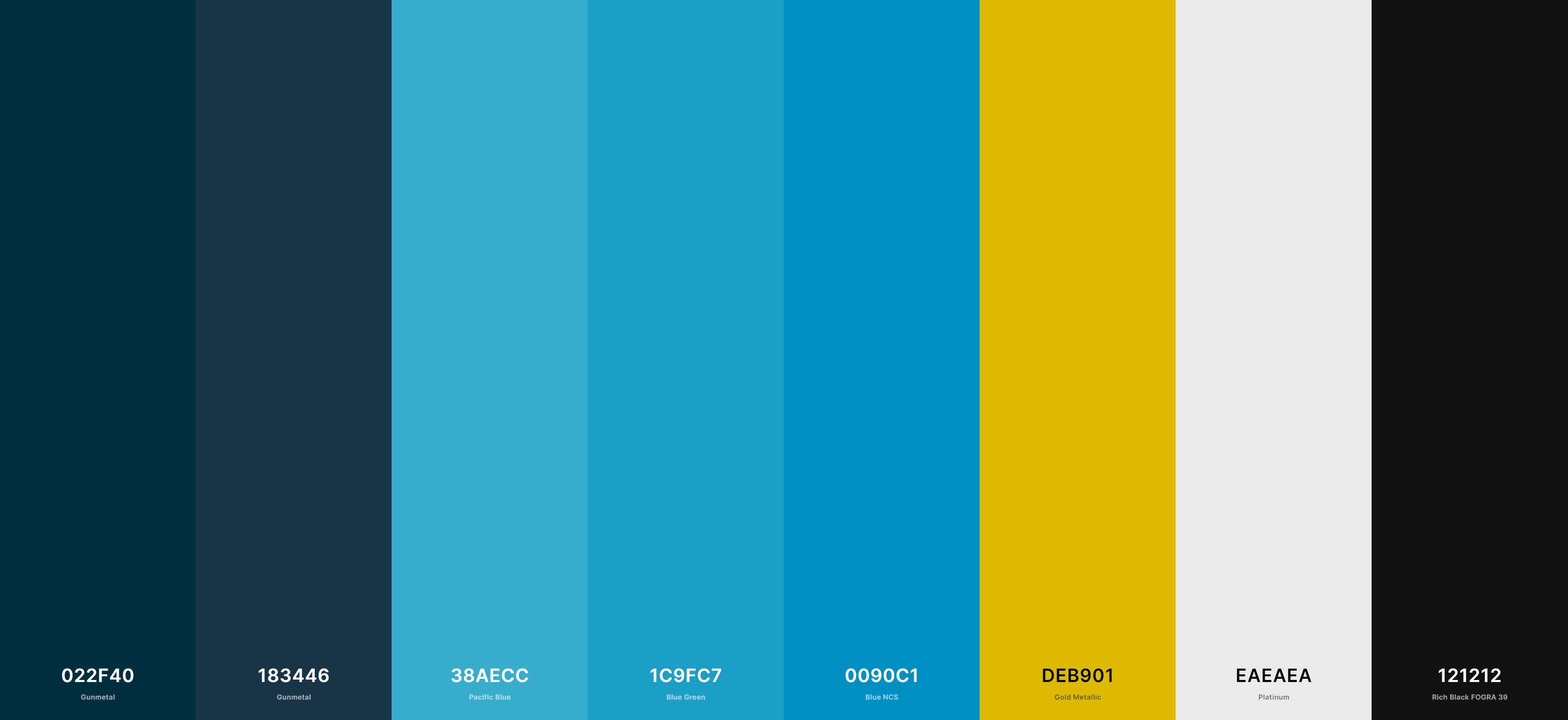
Stock Trading Simulator Project

Software Design Document

1. Model
   1. Game
      1. Game
         * Container holding a name, cash amount and positions.
      2. Name
         * String the player can put in for a name for their save game.
      3. Cash Amount
         * Amount in fake cash
      4. Positions
         * Array of position
   2. Position
      1. Symbol
         * String of purely lowercase letters. Displayed as upper case. A GetDisplay method exists.
      2. Buy Price per Share
         * Number. Cannot be negative
      3. Date
         * The date and time it was “purchased”
      4. Buy()
         * Fake buys more of a stock. Subtracts from cash amount in game. Gets current price for the calculation (Get Price)
      5. Sell()
         * Fake sells a certain amount of a stock. Adds to cash amount from game. Gets current price for the calculation. (Get Price)
   3. Cash Amount
      1. Debit
         * Increases the amount of cash by a certain amount. Throws if the value is negative.
      2. Credit
         * Decreases the amount of cash by a certain amount. Throws if the value is positive.
   4. API
      1. Get Price
         * Returns the current stock price of a given stock
      2. Get Chart
         * Returns the current chart of a given stock
      3. Get Module
         * Returns a requested module about the given stock.
2. User Interface
   1. Template
      1. Header
         * Displays at the top
      2. Footer
         * Displays at the bottom
      3. Navigation Bar
         * The navigation bar is a part of the header. It will provide buttons to the positions, search, and game screens.
      4. Content Container
         * Displays flexibly between the header and footer
         * Holds a screen.
         * Allows a screen to be replaced with another screen. Screen is passed in as an element.
   2. Screens
      1. **Welcome**
         * **Does not have a navigation button.**
         * **Default screen, but will switch if the user has one valid game in local storage.123**
         * Displays text input for a name
         * Displays radio buttons for difficulty, showing 3 different cash amounts.
         * Indicates to the user the app will use local storage.
         * Button for starting the first game.
      2. **Positions**
         * Displays the positions currently held in the game. Each displayed with a position view.
         * Allows a position to be sold.
         * Allows one to buy more.
         * Shows total value.
      3. **Search**
         * Displays a search bar
         * Shows search result below
      4. **Game**
         * Displays information about the game, like the name
         * Has a settings container with some basic settings. – *Low priority as it requires saving.*
         * New Game Button – Starts a new game
         * Shows a **Load Game Container**
   3. Views
      1. Container (CSS Only)
         * Class for how containers generally appear against the background
      2. Search Result
         * Displays the symbol and current price
         * Displays a chart, adjustable by time. Separate GET request.
         * Displays basic information in asset
         * Displays Balance Sheet
         * Displays Income Statement
         * Displays Cashflow Statement
      3. Position
         * Displays how much is owned.
         * Displays symbol.
         * Displays current price.
         * Displays price bought at
         * Displays number of shares
         * Displays buy button. Brings up transact for buy.
         * Displays sell button. Brings up transact for sell.
      4. Transact
         * Displays number of shares input
         * Displays transact button (which displays buy or sell, and saves the game)
         * Displays cancel button, which dismisses the view.
      5. Settings
      6. Load Game Container
         * Displays a list of load games from the local storage
      7. Load Game View
         * Displays a single load game
         * Displays its name
         * Displays last log in.
      8. Financial Statement
         * Renders a financial statement.
   4. Elements
      1. Button
      2. Text Input
      3. Element Builder
      4. Are You Sure – *Low Priority*
3. Theme
   1. Overall
      1. Clean, focused, calm, strong theme
   2. Colors



* + 1. Background – Gunmetal
    2. Secondary – Pacific Blue, Blue Green
    3. Accent – Blue NCS
    4. Text – Platinum
    5. Strong – Gold Metallic
  1. Shape
     1. Clean
     2. Slight bevel
     3. Very small rounded corners
     4. Convey stability with squares.
  2. Texture
     1. No texturing. Clean solid colors.

1. Web
   1. The application will be hosted on Heroku with a free account.
   2. The application will be hosted by a Node JS server.
   3. The application will be built with web-pack – *low priority*
2. Design Concerns
   1. How do we implement LIFO cost?