Project Documentations:

1. **1. Project vision -** The stock market can be an intimidating environment for new investors. With hundreds of stocks to choose from spanning 11 sectors, knowing what to invest in and when can be a difficult task. Our goal through this project is to provide a user friendly toolset that allows our subscribers to track various money flows and provide guidance on which stocks to invest in, all while using easy to read models and statistics.
    a. **1.1. Backgrounds**
        i. Shawn Chapiewski - experienced in html, css, JavaScript, mongodb, node.js, some angular/typescript. Mainly web development languages. Since COVID I picked up Robinhood which is why I chose this project.
        ii. Nick: Just had CSI 4810 last semester so python is his strongest, then java. Two past projects used html, css, php, and a little sql from a database class, so he has experience in those.
        iii. Brandon: He is experienced in the expectations and usability aspects in the UI interaction of these trading platforms. He has used platforms that are archaic like Scottrade and modern stuff like robinhood. His dad is an experienced trader, and he can bring some great input for our group and would be a good person for user testing if we need it. Programming wise, he prefers front end work.
        iv. Alex: He is very experienced with Java, C#, Javascript (Angular and Jquery), Css and Html. He has the most experience in front end but currently working full time as SE and do a lot of backend
        v. Hark:
        vi. Kiratdeep:
    b. **1.2. Socio-economic Impact, Business Objectives, and Gap Analysis**
        i. Socio-economic Impact:
            1. Advantages include: Helping those new to the stock market choose a stock to invest in with a logical algorithm behind it.
            2. Disadvantages: False sense of security, as the market is never 100% predictable.
        ii. Business Objectives: To provide a user friendly toolset that allows our subscribers to track various money flows and provide guidance on which stocks to invest in, all while using easy to read models and statistics.
        iii. Gap Analysis (*Def*: A gap analysis is the process companies use to compare their current performance with their desired, expected performance.) **Can't be done until near the end of the semester**
    c. **1.3. Security and ethical concerns**
        i. Ensure that its clear the website is a prediction/suggestion and not fact
        ii. Probably some type of SEC required disclosure
    d. **1.4. Glossary of Key Terms**
        i. *Ticker Symbol* (def): A stock symbol is a unique series of letters assigned to a security for trading purposes. Stocks listed on the New York Stock Exchange (NYSE) can have four or fewer letters. Nasdaq-listed securities can have up to five characters. Symbols are just a shorthand way of

describing a company's stock, so there is no significant difference between those that have three letters and those that have four or five.

2. **2. Project Execution and Planning**
    a. **2.1. Team Information**
        i. Shawn Chapiewski
        ii. Nick Worthley
        iii. Brandon Cruz
        iv. Alex Kocab
        v. Harkirat Singh
        vi. Kiratdeep Kahlon
    b. **2.2. Tools and Technology**
        i. Languages: Python, Javascript, HTML, CSS, SQLite
        ii. Frameworks: Django, Angular
        iii. Config Management: GitHub, Trello, Discord
        iv. Server Setup: 000Webhost
        v. Development Environment: Pycharm, Visual Studio Code
    c. **2.3. Project Plan**
        i. Sprint 2 Presentation slides 6-7
    d. **2.4. Best standards and Practices**
    e. **2.5. Risk Management**
        i. Not much experience in machine learning
        ii. Limited knowledge of Django
        iii. Not everyone is fluent in stock market jargon or analysis
3. **3. System Requirement Analysis**
    a. **3.1. Function Requirements**
        i. When user enters information in create an account, system adds user to database and sends email for activation
        ii. When user updates information, system must change all requested information in user database
        iii. When user searches for a stock, the system provides most recent data related to said stock
        iv. When user adds stock to favorites, the system adds the information to user database
        v. System provides stock or sector recommendations for the most up to date data available
    b. **3.2. Non-functional Requirements**
        i. The system must send change password and account activation emails within 10 minutes of request
        ii. When requested stock information must be displayed within seconds of page access
        iii. Stock/Sector recommendations must be available instantly upon entering user dashboard
    c. **3.3. On-Screen Appearance of landing and other pages requirements.**
    d. **3.4. Wireframe designs**

4. **4. Functional Requirements Specification**
    a. **4.1. Stakeholders**
        i. People seeking help navigating investing in the stock market
    b. **4.2. Actors and Goals**
        i. User
            1. Uses website through an activated account
        ii. Account Database
            1. Stores user information
        iii. Stock Database API
            1. Provides stock information
    c. **4.3. User stories, scenarios and Use Cases**
        i. Use Cases found in first slides of sprints 1- presentations
        ii. User stories
            1. As a user, I want to be able to create an account
            2. As a user, I want to be able to login into my account
            3. As a user, I want to be able to search for stock symbols
            4. As a user, I want to be able to add stocks to my portfolio
            5. As a user, I want to be able to view stock information
            6. As a user, I want to be able to receive stock recommendations
            7. As a user, I want to be able to receive sector recommendations
        iii. Scenarios
            1. Short story about a user experience on website
    d. **4.4. System Sequence / Activity Diagrams**
5. **5. User Interface Specifications**
    a. **5.1. Preliminary Design**
    b. **5.2. User Effort Estimation**
6. **6. Static Design**
    a. **6.1. Class Model**
    b. **6.2. System Operation Contracts**
    c. **6.3. Mathematical Model**
    d. **6.4. Entity Relation**
7. **7. Dynamic Design**
    a. **7.1. Sequence Diagrams.**
    b. **7.2. Interface Specification**
    c. **7.3. State Diagrams**
8. **8. System Architecture and System Design**
    a. **8.1. Subsystems / Component / Design Pattern Identification**
    b. **8.2. Mapping Subsystems to Hardware (Deployment Diagram)**
    c. **8.3. Persistent Data Storage**
    d. ~~**8.4. Network Protocol**~~
    e. **8.5. Global Control Flow**
    f. **8.6. Hardware Requirement**
        i. Modern web browser
9. **9. Algorithms and Data Structures**

Stock Market Analytics and Recommender

Alex Kocab, Nick Worthley
Harkirat Singh, Kiratdeep Kahlon
Brandon Cruz, Shawn Chapiewski

CSI 4999 - Senior Capstone Project
January 6, 2021

Table of Contents

1. Project Vision:

1.1. Backgrounds:

1.2. Socio-economic Impact, Business Objectives, and Gap Analysis:

1.3. Security and ethical concerns:

1.4. Glossary of Key Terms:

2. Project Execution and Planning:

2.1. Team Information:

2.2. Tools and Technology:

2.3. Project Plan:

2.4. Best standards and Practices:

3. System Requirement Analysis:

3.1. Function Requirements:

3.2. Non-functional Requirements:

3.3. On-Screen Appearance of landing and other pages requirements:

3.4. Wireframe designs:

4. Functional Requirements Specification:

4.1. Stakeholders:

The main stakeholder for this site is the user that will be accessing the website to get more information on stocks.

4.2. Actors and Goals:

For actors we have the following: the User, accountDatabase, stockAPI.

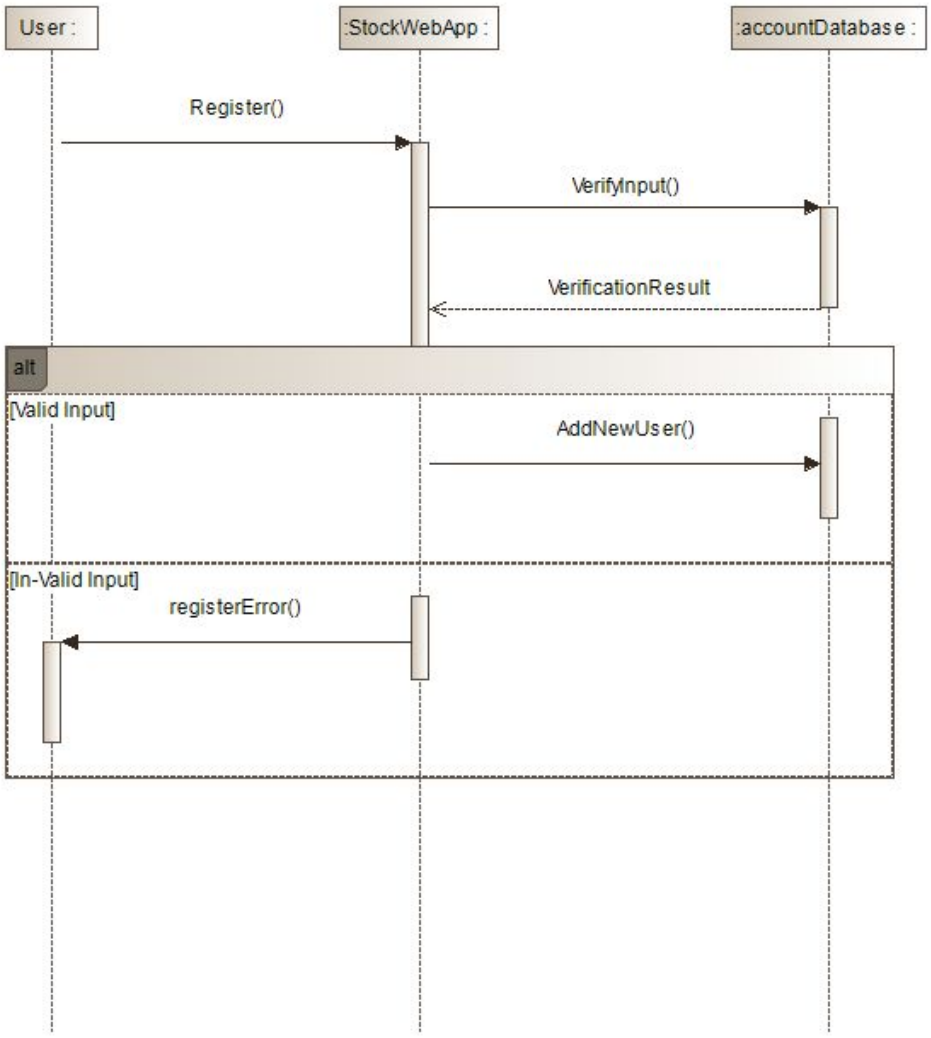
4.3. User stories, Scenarios and Use Cases:

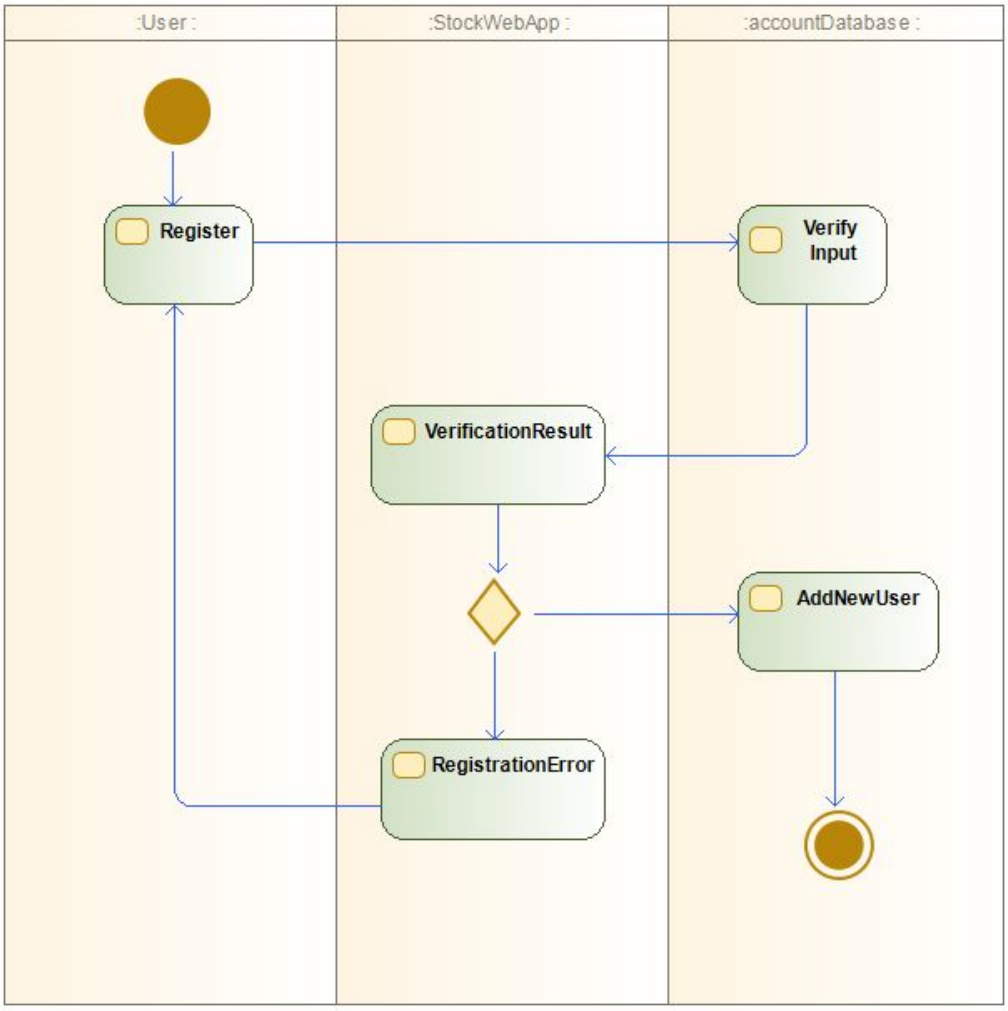
- Register User Account:
  - **Preconditions**: User is required an email
  - **Success Guarantee (Postconditions):** User can register to our website
  - **Main Success Scenario:**
    - 1. User Connects to our website
    - 2. User opens register tab
    - 3. User enters required information to sign up.
    - 4. User receives confirmation email
    - 5. User clicks on link and activates account
    - 6. User successfully registered.
  - **Extensions:**
    - 1. Page not loaded properly
    - 2. Email/ Password criteria does not meet.
  - **Special Requirements:** Strong password (Uppercase/ Lowercase/ Number/No Common Passwords)
- Verify Login:
  - **Preconditions**: User requires an activated account
  - **Success Guarantee (Postconditions):** User can login successfully
  - **Main Success Scenario:**
    - 1. User Connects to our website
    - 2. User Enters Login Credentials
    - 3. User Successfully Logged-In
  - **Extensions:**
    - 1. Page not loaded properly
    - 2. Email/ Password is invalid
- Update Profile:
  - Success Guarantee (Postconditions): User can update username, email, or password
  - Main Success Scenario:
    - 1. Users goes to update profile/email/update password
    - 2. User Enters updated username/password
    - 3. User information is updated
  - Extensions:
    - 1. Page not loaded properly
    - 2. Updated username/email/password does not meet requirements
- Add To Portfolio:
  - **Preconditions**: User requires an activated account

    - **Success Guarantee (Postconditions):** User adds chosen stock to their favorites
    - **Main Success Scenario:**
        - User logs into an activated account
        - User searches for stock symbol
        - User clicks add to favorites
        - User gets notice that stock has been added to favorites
    - **Extensions:**
        - Page not loaded properly
        - Stock failed to add to favorites
- Get Stock Information:
    - **Preconditions**: User requires an activated account
    - **Success Guarantee (Postconditions):** User searches and views a stock's information
    - **Main Success Scenario:**
        - User logs into an activated account
        - User enters stock symbol into dashboard search bar
        - User selects symbol
        - User is brought to a page containing information on search symbol
    - **Extensions:**
        - Page not loaded properly
        - Stock not found
- Add Package:
    - **Preconditions**: User requires an activated account
    - **Success Guarantee (Postconditions):** User adds chosen package to their account
    - **Main Success Scenario:**
        - User logs into an activated account
        - User reviews and selects from one of our packages we offer: starter, deluxe, or ultimate.
        - User is redirected to the portfolio page
        - User gets notice that the package was added to their account and gives a description of what the package offers you.
        - Account is limited to however many tickers the package allows
    - **Extensions:**
        - Page not loaded properly
        - Package failed to add to account
- Historical Data:
    - **Preconditions**: User requires an activated account
    - **Success Guarantee (Postconditions):** User is able to see Historical Data of the Recommendation System for an inputted ticker
    - **Main Success Scenario:**
        - User logs into an activated account
        - User types in a valid stock ticker in the search bar
        - User is redirected to a page where they can view historical data of the ticker they inputted
    - **Extensions:**
        - Page not loaded properly
        - Ticker inputted is not a valid stock ticker

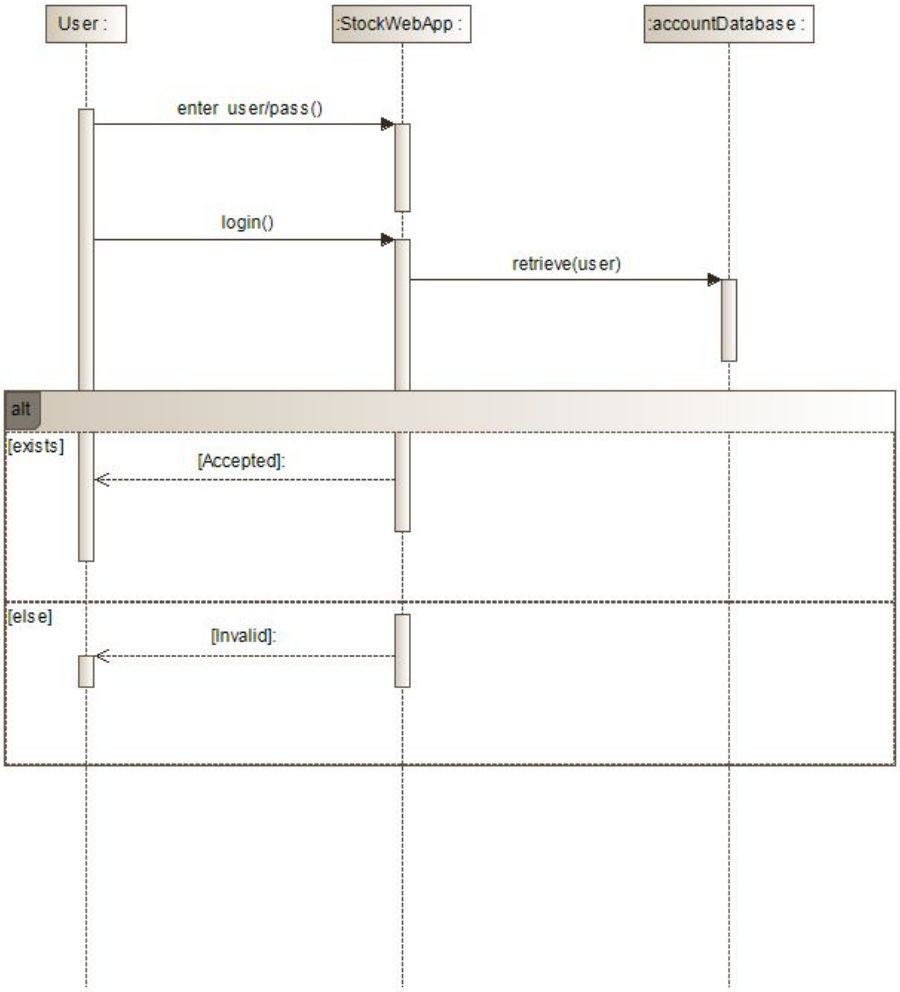
4.4. System Sequence / Activity Diagrams:

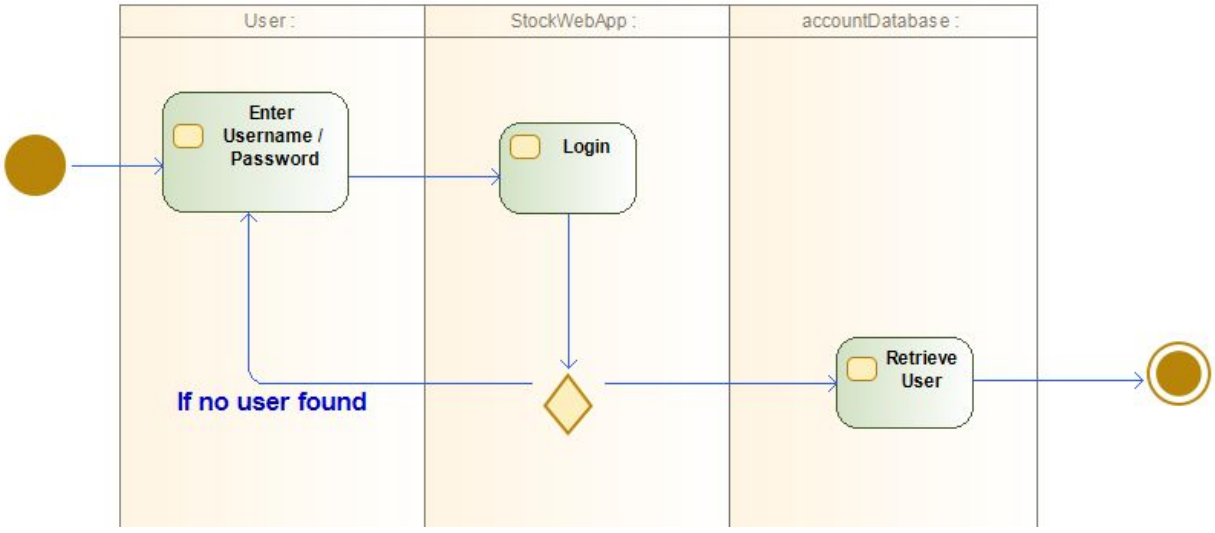
Register User Accounts:



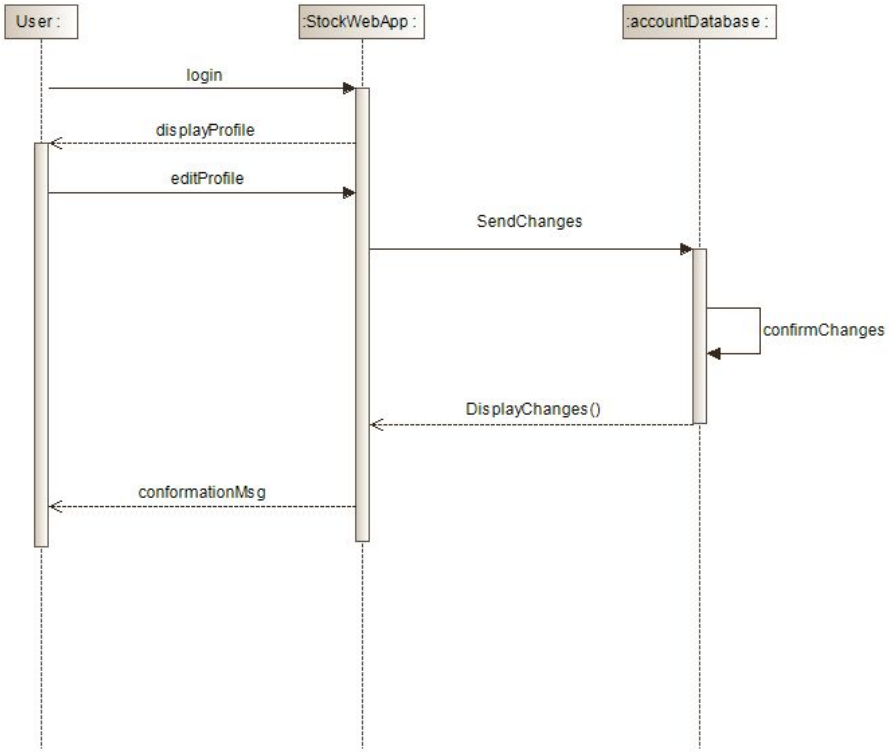


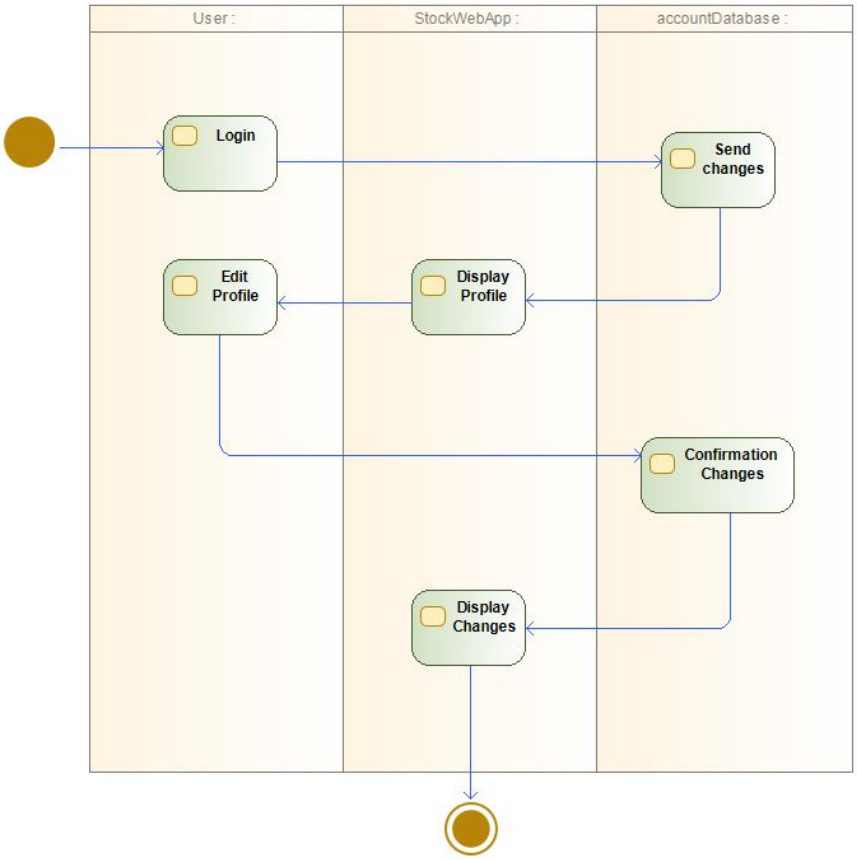
Verify Login:


User :
:StockWebApp :
:accountDatabase :
enter user/pass()
login()
retrieve(user)
alt
[exists]
[Accepted]:
[else]
[Invalid]:

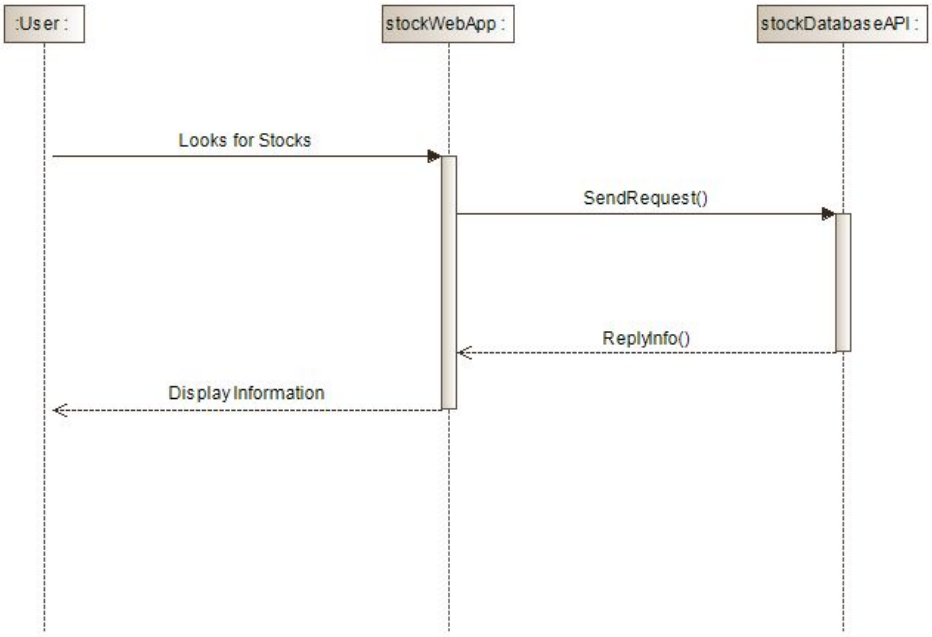

User :
StockWebApp :
accountDatabase :
Enter Username / Password
Login
Retrieve User
If no user found

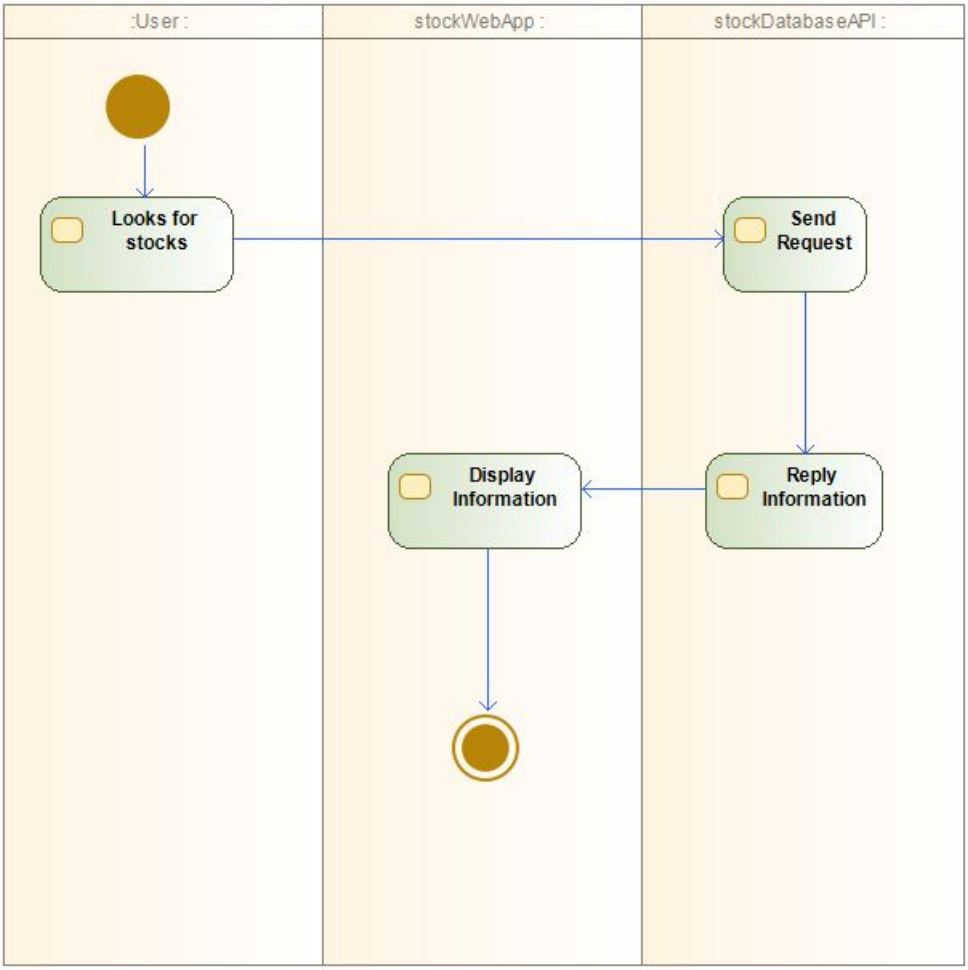
Update Profile:



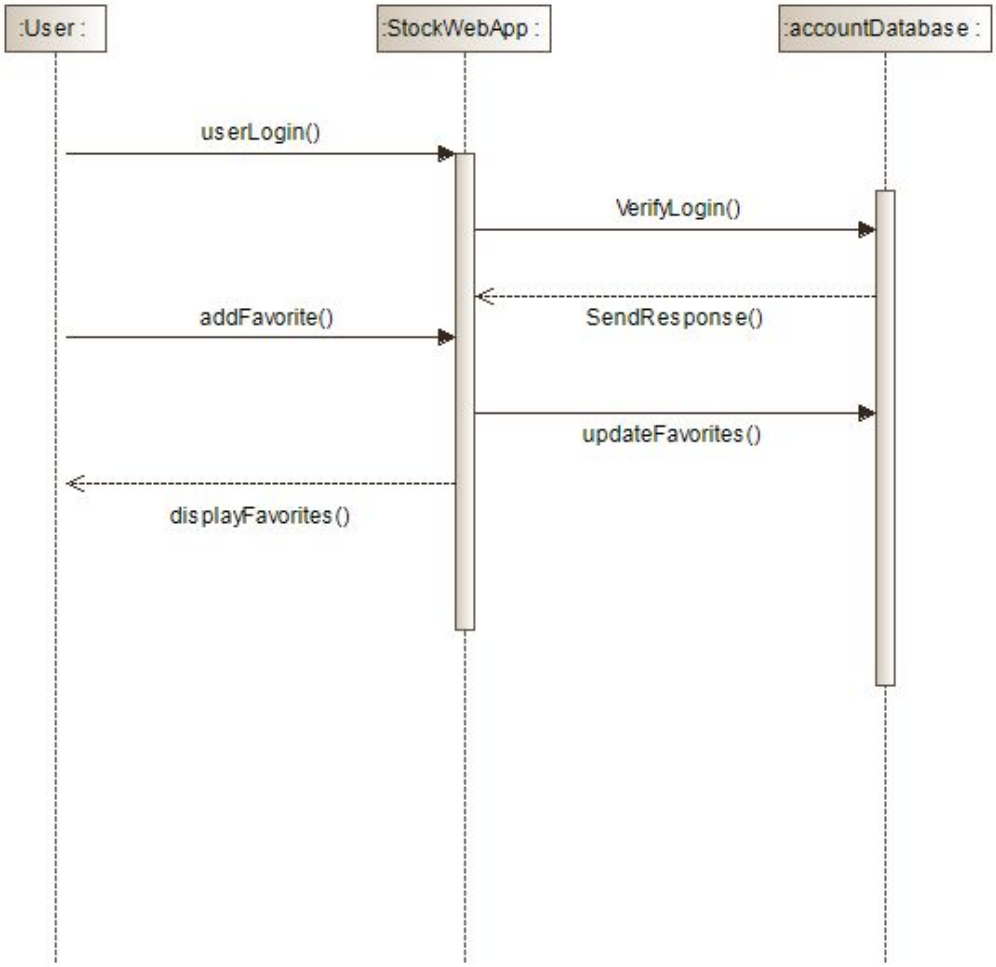


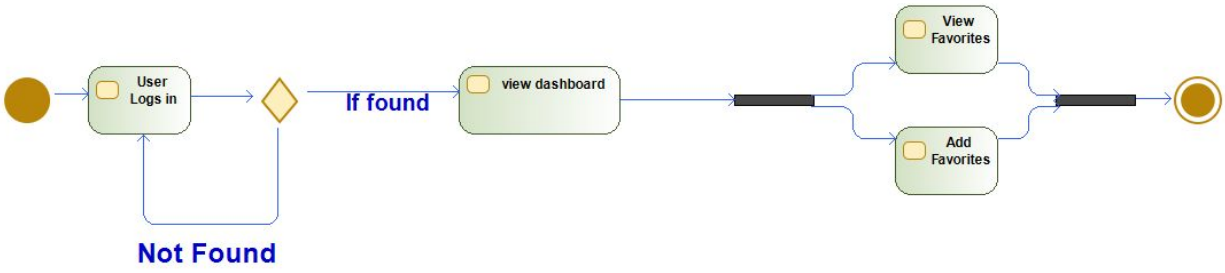
Get Stock Info:



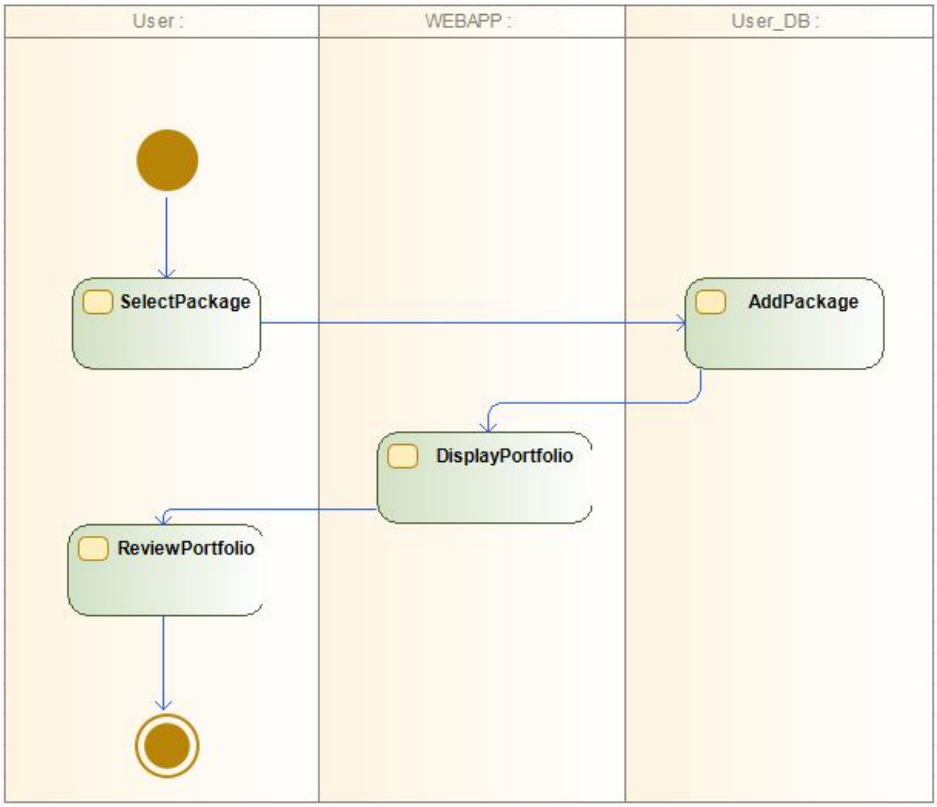


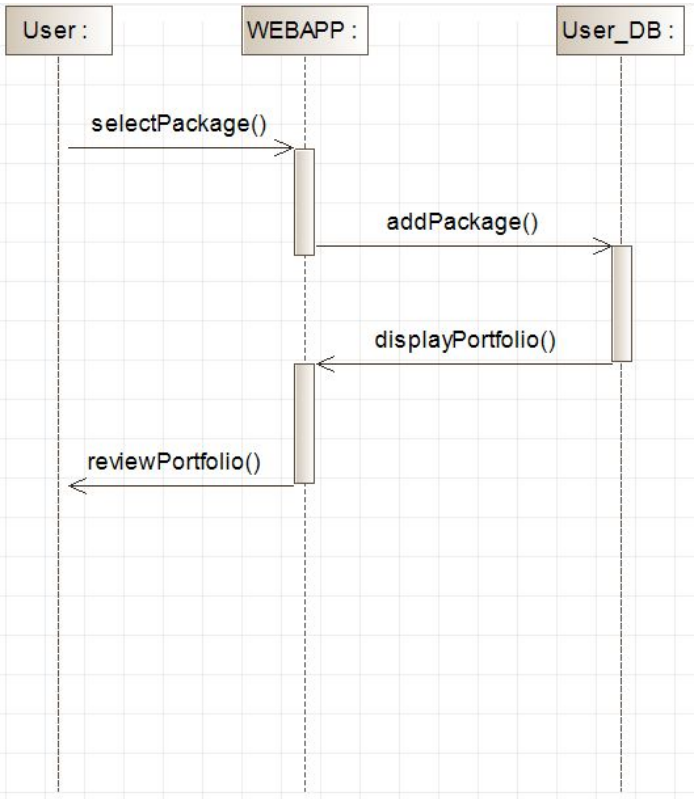
Add To Portfolio/Favorites:





Add Package:


User :
WEBAPP :
User_DB :
SelectPackage
AddPackage
DisplayPortfolio
ReviewPortfolio


User :
WEBAPP :
User_DB :
selectPackage()
addPackage()
displayPortfolio()
reviewPortfolio()

5. User Interface Specifications:

5.1. Preliminary Design:

5.2. User Effort Estimation:

6. Static Design:

6.1. Class Model:

6.2. System Operation Contracts:

6.3. Mathematical Model:

6.4. Entity Relation:

7. Dynamic Design:

7.1. Sequence Diagrams:

7.2. Interface Specification:

7.3. State Diagrams:

8. System Architecture and System Design:

8.1. Subsystems / Component / Design Pattern Identification:

8.2. Mapping Subsystems to Hardware (Deployment Diagram):

8.3. Persistent Data Storage:

8.4. Network Protocol:

8.5. Global Control Flow:

8.6. Hardware Requirement:

9. Algorithms and Data Structures:

9.1. Algorithms:

9.2. Data Structures:

10. User Interface Design and Implementation:

10.1. User Interface Design:

10.2. User Interface Implementation:

11. Testing:

11.1. Unit Test Architecture and Strategy/Framework:

11.2. Unit test definition, test data selection:

11.3. System Test Specification:

11.4. Test Reports per Spring:

12. Project Management:

12.1 Project Plan:

12.2. Risk management:

References