

# CMPE 451

## Final Milestone Report

Fall 2019

Sercan Ersoy

Alper Çakan

Barış Zöngür

Ali Özden

İsmet Dağlı

Volkan Yılmaz

Furkan Aydar

Efe Önal

Ceren Tahtasız

# Table of contents

[Summary and Final Assessment](#)

[List and Status of Deliverables](#)

[Evaluation of Deliverables](#)

[Summary of Coding Work Done](#)

[Annotation Implementation & W3C standard Compliance](#)

[Project Requirements](#)

[1. Functional Requirements](#)

[1.1. User Requirements](#)

[1.2. System Requirements](#)

[2. Non-Functional Requirements](#)

[User Scenarios Used in Presentation](#)

[Diagrams](#)

[8.1. Use Case Diagram](#)

[8.2. Class Diagram](#)

[8.3. Sequence Diagrams](#)

[Mockups](#)

[System Manual](#)

[9.1. Backend](#)

[9.2. Frontend](#)

[9.3. Mobile](#)

[User Manual for Web Application:](#)

[API Documentation](#)

[Project Plan](#)

[User Manual for Mobile](#)

## 1. Summary and Final Assessment

This is the final milestone report that describes the work, the deliverables and the status of work accomplished by Group 8 so far. We have worked on a project called “Traders’ Platform” as a social platform to inspect, share, and perform actions for trading purposes.

We planned that we design articles, events, trading forexes, digital equipments, commodities and so on, comments on articles and trading equipments, follow and unfollow for user, search user, trading equipment and several minor functionalities. After Milestone 2 we planned to implement annotation, notification, recommendation and other forex items with assets and portfolio sections. Until now we accomplished what we planned and have made a good presentation to explain our application using real life scenarios.

We describe what we have done detailed in this report. We write overviewed project requirements. Backend team explain API documentation. Frontend and mobile teams explain scenarios that is created for presentation.

During this two semester course/project, we believe that we have gained a myriad of invaluable skills. First and foremost, we learned how to work on a complex software as a team. This skill does not only include technical aspects such as using code collaboration tools (e.g., Git and GitHub) but also soft skills such as planning and task sharing as a group. Furthermore, we learned a lot about both the practice and the theory of software engineering. Lastly, as a cherry on the top, we have also learned how to develop web applications with quite useful development tools and frameworks such as React and Django.

## 2. List and Status of Deliverables

Deliverable Name	Description	Status
Backend Project	Nova: backend application for Mercatus. Tech stack: Django	Completed.

Mobile project	Android client for Mercatus. Tech stack: Kotlin	Completed.
Frontend Project	Web client for Mercatus. Tech stack: React JS	Completed.
API Documentation	API is documented via both Swagger UI and Postman Documenter	Completed.
Deployment	Web and backend application's deployment in a publicly available server	Completed.
Group Meetings	Weekly meetings or meetings organized for urgent situations	Completed.
Project Requirements	Elicited requirements for Mercatus	Completed.
Diagrams	Use case, class and sequence diagrams for Mercatus	Completed.
Personas and User Stories	Created personas and the user stories (scenarios) for users corresponding to the personas	Completed.
Project Plan	Plan of the project that contains previous/future tasks and their statuses	Completed.
Issues	Issues opened for the subtasks, assigned proper labels, assignees	Completed.
Pull requests	Opened pull requests for the subtasks corresponding to these subtasks' issues	Completed.

### 3. Evaluation of Deliverables

So far, we have completed all of what we had planned to complete before the final milestone. After several discussions with our customer Alper, we had planned to accomplish most of the functionalities within the application. Our application is built on mostly feed screens of articles, events and trading equipment. Therefore, we planned to implement those and leave minor functionalities at the end.

We planned to finish, again just like we have discussed and concluded together with our customers, almost all of the features before the final milestone. Since we have succeeded in keeping up with our first and second milestone plan; our plan for the final milestone has not been delayed or adversely affected in anyway.

After the second milestone, we have implemented annotation, recommendation, notification and other trading equipment with asset to buy and sell functionalities and portfolio sections. We show our work in the demo presentation. We tried our best to accomplish this task and create a successful application even though our group was 8 persons work.

Finally, now, at the end of the semester, we see that we have indeed succeeded in implementing almost all of the requirements. Furthermore, our customer was quite satisfied with the way we implemented them.

### 4. Summary of Coding Work Done

Member	Work Done
Sercan Ersoy	<p><b>From 2nd milestone to final milestone:</b></p> <ol style="list-style-type: none"><li>1. Implemented semantic search for articles</li><li>2. Implemented portfolio recommendations</li><li>3. Implemented article recommendations</li><li>4. Implemented daily prices view</li><li>5. Implemented intradaily prices view</li><li>6. Implemented current price view</li><li>7. Implemented fetcher for daily prices of forexes and digital currencies from AlphaVantage</li></ol>

8. Implemented fetcher for intradaily prices of forexes and digital currencies from AlphaVantage
9. Implemented fetcher for current price of forexes and digital currencies from AlphaVantage
10. Wrote cron jobs for AlphaVantage fetchers in the server side
11. Fixed error for media variable (profile pictures)
12. Fixed error in *groups.json* file
13. Implemented fetcher for daily prices of indices from Nasdaq
14. Implemented fetcher for intradaily prices of indices from Nasdaq
15. Implemented fetcher for current prices of indices from Nasdaq
16. Implemented fetcher for daily prices of etfs from Nasdaq
17. Implemented fetcher for intradaily prices of etfs from Nasdaq
18. Implemented fetcher for current prices of etfs from Nasdaq
19. Implemented fetcher for daily prices of commodities from Nasdaq
20. Implemented fetcher for intradaily prices of commodities from Nasdaq
21. Implemented fetcher for current prices of commodities from Nasdaq
22. Implemented fetcher for daily prices of stocks from Nasdaq
23. Implemented fetcher for intradaily prices of stocks from Nasdaq
24. Implemented fetcher for current prices of stocks from Nasdaq
25. Wrote cron jobs for Nasdaq fetchers in the server side
26. Implemented cron job key for price fetcher internal views
27. Implemented Swagger API view
28. Added fixtures for commodities, forexes and indices
29. Fixed various import errors
30. Various code clean-ups and reformat

**From 1st milestone to 2nd milestone:**

1. Deleted unused framework Kronos from the backend project.
2. Cleaned up the code (organized imports, line breaks, indentations in order to stick with the Python coding conventions)
3. Added Kronos cron job decoration but then we decided not to use it. (I then added cron jobs manually in the server)
4. Fixed bug in the email verification part and cleaned up the code.
5. Added installation command for ‘requests’ library in backend project’s readme file.
6. Added basic serializer for user to prevent getting recursive responses (because followers are also user objects). This change was made in the server, I only committed it.
7. Other than that, I was controlling all the deployment (e.g. deployment when new commits are merged, server is down), database management (e.g. truncating database, checking whether some endpoints work safe and sound), and job scheduling (e.g. adding cron jobs for fetching trading equipment parities, checking whether it is working fine) parts of the project.

**Until 1st milestone:**

1. Added endpoint delete user endpoint to the backend.
2. Added endpoints follow/unfollow user, get followers/followings to

	<p>the backend.</p> <ol style="list-style-type: none"> <li>3. There were situations for example a user with token tries to modify some other user's profile using PUT method. Corrected these situations with checking if the user with token is the target user.</li> <li>4. When password is not sent to the PUT request, password was changing unintentionally. Corrected this situation.</li> <li>5. When creating only fields (like first_name and last_name) were not given to the PUT request, there was an error. Corrected this situation.</li> <li>6. Removed trailing slashes from url endpoints to stick with the REST conventions.</li> <li>7. Updated .gitignore file when needed, fixed a typo in the setup_db_creds.py.</li> <li>8. For testing, we were using SQLite. Changed this back to the Postgres.</li> </ol>
Alper Çakan	<p><b>From 2nd milestone to final milestone:</b></p> <ol style="list-style-type: none"> <li>1. Implemented a script to automatically create the supported stock types</li> <li>2. Implemented the new price architecture, model and serializer</li> <li>3. Refactored and reorganized the backend code (grouping all the view, utility methods etc.) to improve the organizational structure</li> <li>4. Implemented new portfolio list, create, update, query, delete endpoints</li> <li>5. Implemented new portfolio follow, unfollow</li> <li>6. Implemented endpoint for depositing cash to trader account</li> <li>7. Implemented the annotation model and serializer</li> <li>8. Implemented annotations create, list, get, update, delete endpoints</li> <li>9. Implemented utility method for cross exchange rate calculation</li> <li>10. Implemented utility method for price string to fixed point decimal conversion</li> <li>11. Implemented trader user asset query endpoint</li> <li>12. Implemented trading feature (buying and selling trading equipments) endpoint</li> <li>13. Updated the notification model and serializer to support notifications for new articles, comments and annotations</li> <li>14. Implemented endpoint for querying and clearing notifications</li> <li>15. Implemented endpoint for querying the number of unread notifications of the user</li> <li>16. Implemented model hooks for automatically create notification when a followed user creates an article, comment or annotation</li> <li>17. Implemented frontend feature for showing the user the portfolios he's following on his home page</li> <li>18. Implemented feature for showing live prices in portfolio cards</li> </ol>

19. Implemented frontend feature for showing various statistical graphs (such as, daily close prices, moving average prices etc.) about trading equipments
20. Implemented automatic profit loss calculation for trader users
21. Implemented endpoint for querying trader users' profit loss
22. Implemented frontend feature for showing the trader user his profit loss on his user page
23. Implemented frontend feature for computing the transaction amounts in both source and target trading equipment amounts

**From 1st milestone to 2nd milestone:**

1. Reformatted some of the backend code to comply with PEP rules
2. Fixed the bug that caused the passwords to be hashed twice
3. Implemented a feature for reading the email server config from the environment instead of hardcoding them
4. Implemented a feature for skipping the email activation (ie, activate by default) while in debug mode to speed up testing
5. Implemented a feature for saving configuring the axios to automatically send user token and headers to prevent code repetition
6. Removed the content length restriction imposed on the articles and the comments to make the models more flexible
7. Implemented the Events API for querying the financial events that happened/will happen on a particular day, along with their country of origin and importance
8. Created an event model for saving the events into the database
9. Implemented a serializer for the event model
10. Implemented a regex based parser that fetches the events from investing.com, parses them and saves them in the database
11. Implemented an internal endpoint that runs the events parser
12. Implemented a public authorization class (ie, all allowed) that was need for some of the new endpoints
13. Implemented Google Maps based location selection (with marker) in the frontend sign up form
14. Implemented a utility function that creates a meaningful, detailed human-friendly error message using the error object returned by the backend
15. Implemented some improvements for the sign up form (such as using a calendar based date of birth selection and showing meaningful error messages)

	<p>16. Co-implemented the feature that shows a selection of news articles on the home page in the frontend</p> <p>17. Fixed some of the styling problems in the frontend</p> <p>18. Reviewed pull requests</p> <p><b>Until 1st milestone:</b></p> <ol style="list-style-type: none"> <li>1. Initialized the Python Django backend app.</li> <li>2. Implemented a script that sets up the credentials for connecting the backend app to the database</li> <li>3. Implemented the extra Django settings (switching to postgres database, iso date format; CORS settings; email based auth)</li> <li>4. Wrote up a technical readme about the backend app that has instructions about setting up the app, the database, implementing a new endpoint, implementing a new model, a new serializer</li> <li>5. Implemented the data initializer (“fixture”) that sets up the authentication groups in the database</li> <li>6. Implemented the model “User”</li> <li>7. Implemented the “details” serializer for the model “User”, which also includes encrypted password storage in DB and defaulting the auth group to “basic user” when unspecified</li> <li>8. Implemented the “simple” user serializer (which I designed to eliminate the circular serialization problem with the user follow relationships)</li> <li>9. Implemented a new base serializer class that allows you to specify “create_only_fields” which are only written to DB while being created and never updated again (example usage: name surname)</li> <li>10. Implemented User listing, create, update, read (view profile)</li> <li>11. Implemented the authentication mechanism in the backend (auth tokens database, Django auth settings, fields etc.)</li> <li>12. Implemented the login (i.e., auth token create) endpoint</li> <li>13. Implemented the permissions mechanism and a couple of frequently needed permissions classes (is post request or is the user authenticated, is the user trader, is the user admin, ...)</li> <li>14. Implemented all-users listing functionality</li> <li>15. Implemented followers and followed lists feature for the user profile</li> <li>16. Fixed the “http unauthorized” bug of the user profile update feature</li> <li>17. Implemented follow user feature</li> <li>18. Implemented unfollow user feature</li> </ol>
Barış Zöngür	<p><b>From 2nd milestone to final milestone:</b></p> <p>1. Implemented majority of functionality in Trading Equipment page where you can select all trading equipments and compare them with a graph and in the same page buying and selling selected assets is implemented as well.</p> <p>1A) In the Trading Equipment page all equipments requested from the</p>

API. Equipments returned in a JSON format where each equipment's symbol (which will be used later) type and name is returned.

2A) These equipments grouped by type in the front-end in rendering to being able to choosing which type of equipment to buy, see the graph or compare with a graph.

3A) After the implementation of selecting the trading equipment to send appropriate request to API, which are buy/sell, rate, comments etc.. graph and other dropdown selections are rendered without page refresh to have a smoother user experience.

4A) In comparing and seeing the current rate of equipments, request of seeing the current rates are sent to API. Since API returned each equipments rate against US dollars, for example to compare TRY and EURO, these rates calculated in front-end and graph is rendered by those values calculated in front-end. So each rate is calculated and ordered in front-end and shown to user via graph.

5A) Daily and intradaily functionalities are implemented separately, since API requests are separate for each equipments. Calculations and rendering are done in the same way as part 4A.

6A) In the same page, with already grouped equipment, a new selection is added as buy and sell. With each equipment selection, your current amount of that asset is requested and shown to user in a dynamic manner. With which asset you can buy the current equipment is dynamically shown to user as well.

2. For each equipment, upvote and downvote visuals and functionalities implemented.

3. Even Trading Equipments page rendered in a single route, to being able to route from different pages as search results, portfolio etc. page is structured as getting the first equipment selection from route and rendering the page as first selection is already made.

4. Button and link routes are added to portfolio search results etc. to make user be able to access the page in different parts of the site.

5. A lot of bug fixed that took days...

6. Small visuals and styling.

#### **From 1st milestone to 2nd milestone:**

1. Done the restructuring of the front-end as first implementation was not suitable for bigger projects.

2. Implemented main big picture structure for routing and component placement and placed routing for all components accordingly.

3. Implemented front-end follow profile functionality and design

4. Implemented search functionality for users.

5. Added followings and followers pages reached from own profile page

6. Added own articles to the Navbar's dropdown menu and profile.

7. Implemented and designed like, dislike and number of likes for articles

8. Implemented and designed like dislike and number of likes for comments.

9. Restructured and designed the trading equipment page to final functionality as selecting parities and returning the corresponding

	<p>page.</p> <ol style="list-style-type: none"> <li>10. A lot of bug fixing of all pages for more than two full days.</li> <li>11. Implemented and designed other users profile pages as seen when a logged in user goes for another person's profile</li> <li>12. General functionality of profile pages.</li> <li>13. Ordered the folder structure for main code for readability.</li> <li>14. Some design choices as what to render when for example showing person's profile after log-in.</li> </ol> <p><b>Until 1st milestone :</b></p> <ol style="list-style-type: none"> <li>1. Implemented sign-up and log-in pages stated based primary functionalities using hooks and state registers.</li> <li>2. Implemented various state based render functions</li> <li>3. Implemented different types of rendering of navbar for different types of user types.</li> <li>4. Implemented sign-up pages connections with back-ends endpoints.</li> <li>5. Correlated work with other frontend members to help with various implementations.</li> <li>6. Main component design(app.jsx) and props passing to different components.</li> <li>7. General organization of the code and developing visuals for pages.</li> </ol>
Ali Özden	
İsmet Dağlı	<p><b>From 2nd milestone to final milestone:</b></p> <ol style="list-style-type: none"> <li>1. Created buy any kind of order like trading equipment.</li> <li>2. Created sell any kind of order like trading equipment.</li> <li>3. Created add cash.</li> <li>4. Connected all kinds of buy-sell activities with back-end.</li> <li>5. Design and implemented layout pages for buy-sell activities</li> <li>6. Fixed the bug with other phones with add cash and buy equipment problem.</li> <li>7. Created and mainly implemented most part of portfolio page.</li> <li>8. Implemented create a portfolio.</li> <li>9. Implemented get portfolios of a user.</li> <li>10. Implemented a specific portfolio.</li> <li>11. Implemented deletion the portfolio itself.</li> <li>12. Created all necessary pages and back-end connection with portfolio.</li> <li>13. Last day, I had a problem with push branch; I was pretty sure that I pushed but there was no code on github. So, With Volkan, we created adding equipment to portfolio and listing them in the portfolio page.</li> <li>14. Implemented events' date are static, change its format as it will be updated to the current date and set always the next 3 days including today.</li> </ol>

15. Events are randomly distributed. Order them according to their importance.
16. With Volkan, we created the mobile scenario for the demo day.

**From 1st milestone to 2nd milestone:**

1. Since the mobile team was 2 people and the increasing necessity for mobile, I transferred the mobile team and started to learn Kotlin
2. I added follow and unfollow functionality for users.
3. I added logout functionality.
4. I added the fix when you click the back button on Android, it logs out and exists the app. So, the app remembers the login credentials.
5. I enhanced register page screen visibility.
6. I added go events shortcuts in the navigation bar and connection activity
7. I created the page of the details of an event.
8. I changed the event fragment in the bar.
9. I created an adapter in order list events, and also created a connection with the back-end so that we can call all existing events.
10. I created floating action buttons and those actions through each trading equipment fragments'.
11. I created a search screen for trading equipments.
12. I connected these fragments with the same search screen via click event.
13. I created a trading item to print search results.
14. I connected equipment with the backend.
15. I created necessary back-up database info for mobile demo and helped for demos.
16. Since Volkan implemented lots of functionality and created lots of pull-request, it took quite a long time to test them in a piece by piece. I helped many of them.

**Until 1st milestone:**

1. Revised the requirement, mock-up, project-plan, design
2. Made research about testing methods.
3. Designed the database model with the back-end team
4. Documentation on the design database model
5. Made a research about how to implement sign-up and login with Google auth in Django
6. Testing Django implementation with SQLite, since I face unsolvable problems with PostgreSQL. (Due to PostgreSQL connection problems, I have not been able to make many contributions on implementation of the back-end.)
7. Due to inefficiency until milestone1 because of me, I moved the mobile team.

Volkan  
Yilmaz

### **From 2nd milestone to final milestone:**

1. I created mobile presentation and scenarios to be used in the demo presentation.
2. I added like and dislike article show page.
3. I created digital trading equipment page and make a connection with backend.
4. I created stocks and other trading equipment page and make a connection with backend.
5. I created asset page to load money and make transaction through the mercatus.
6. I change homepage to show recommended articles based on followers and followings of the user.
7. I created notifications page and create a system no notify users.
8. I help some bugs inside events and portfolios screens.

### **From 1st milestone to 2nd milestone:**

9. I determined and then removed Android deprecated function calls and signatures.
10. I organized user profile page and created a more efficient layout. Moreover I added edit profile page and connected with backend to make changes in email and birthday. (Name and surname cannot be changed.)
11. I added animations in app opening activity and smooth transition between screens. Moreover, I designed an app logo in order to show more pleasant.
12. I reorganize choose user type menu to show users options in a more pleasant way.
13. I added if user login credentials are not match warning and international iban number in trader register screen .
14. I designed a navigator menu and added submenus in order to give user easy motions and to find functions in the app easily.
15. I created articles feed page, an article show page, and connected with backend to fetch results. I designed an article item to supply functionalities within an object for this particular article. I added make like-dislike or revoke like-dislike and comment features in articles. Also I implemented edit-delete options for articles. I also organize view options because only creator is able to edit or delete an article.
16. I designed a comment item to supply functionalities within an object for this particular comment. I implemented edit-delete a comment options and feed screen for comments. Moreover, I added like dislike comments below articles, forex and digital trading equipment. I also organize view options because only creator is able to edit or delete a comment.
17. I created forex feed page, a forex show page, and connected with backend to fetch results. I added line chart in show forex item screen

	<p>to view its values over time. I also added zoom screen horizontally. I designed a forex item to supply functionalities within an object for this particular forex and added increase or decrease situation with respect to last closing and opening data. I added make like-dislike or revoke like-dislike and comment features below forex items.</p> <ol style="list-style-type: none"> <li>18. I created digital feed page, a digital show page, and connected with backend to fetch results. I added line chart in show digital item screen to view its values over time. I also added zoom screen horizontally. I designed a digital item to supply functionalities within an object for this particular digital and added increase or decrease situation with respect to last closing and opening data. I added make like-dislike or revoke like-dislike and comment features below digital items.</li> <li>19. I added make prediction layouts in forex and digital items.</li> <li>20. I created a home page to show 10 forex items and 3 articles to be used more efficiently in the future</li> <li>21. I created an example profit screen.</li> <li>22. I helped to get Google maps api key.</li> <li>23. I helped to design user follow and unfollow operations.</li> <li>24. I made contributions in events feed screen with visual improvements.</li> <li>25. I created mobile presentation and scenarios to be used in the demo presentation.</li> </ol> <p><b>Until 1st milestone:</b></p> <ol style="list-style-type: none"> <li>1. I revise the requirement, mock-up, project-plan, design and request some changes.</li> <li>2. I initialize mobile branch and create first project structure.</li> <li>3. I implement sign in and sign up layouts. We determined 1 login activity for both users and 2 separate activities for trader and basic users since we request different credentials.</li> <li>4. I initialize and develop API connection with backend and mobile.</li> <li>5. I create models for register, login and user info with data.</li> <li>6. I implement sign in functionality and test it whether it works.</li> <li>7. I implement sign up functionality for both trader and basic user. Afterwards I test it whether it works. I added date picker to layout.</li> <li>8. I help to design and create edit profile screen. I generate shared preferences to transfer a user token when a user logs in.</li> </ol>
Furkan Aydar	<p><b>From 2nd milestone to final milestone:</b></p> <ol style="list-style-type: none"> <li>1. I implemented the initial version of the portfolio model, serializer and the create, read, delete, update views.</li> <li>2. I refactored the av.py(forex, digital currency fetcher) to comply with our new design choices.</li> </ol> <p>Frontend:</p>

3. I implemented the annotation creation by text selection, note adding modal and saving, restyling the annotated sections and displaying annotation info on user click.
4. Cleaned up / Refactored many frontend files.
5. Fixed the annotation bugs for a couple of days.
6. Added profile image adding function, updated the related jsx files to fetch imported profile image from the database.
7. Completely redesigned profile page. Changed layout, buttons, profile card design. Added additional info to the profile card.
8. Arranged different contents to be displayed on either on user's own profile and other user's profile.
9. Added recommendations to the profile page and handled hiding showing states of article and equipment recommendations.
10. Added my assets section to the profile page, listing all the equipments user has.
11. Updated the article page layout to make it stay vertical. Also styled write new article page.
12. Updated the comment box design of equipments and articles.
13. I designed the trading equipment graph page's graphic size, background, comments section. Added indicators of the currently selected equipment's indicators to the buy sell section.
14. Re-adjusted the layout of the portfolios tabs.
15. Designed the specific portfolio pages' layout, buttons, added new icons.
16. Styled the notifications page by repositioning the items, adding background and icons.

#### **From Milestone1 to Milestone2:**

1. I arranged the alphavantage keys in our settings.
2. I created the Article model.
3. I created the Trading Equipment and the Parity models.
4. I added base Comment, TradingEquipmentComment and Article Comment models.
5. I added base LikeDislike, ArticleLikeDislike and CommentLikeDislike models.
6. I added the Current Price model in relation with parities.
7. I added Prediction model for Trading Equipments.
8. I added article, trading equipment, parity and comment views

- including all of their functionalities (Create, Retrieve, Update, Delete)
9. I added like / dislike functionality to the comments and articles.
  10. I added prediction functionality to the trading equipments
  11. . I arranged endpoint urls for mentioned functionalities.
  12. I created some authentication classes in permissions.
  13. I created trading equipment, comment, article, like-dislike, parity, current price serializers for our models.
  14. I implemented av.py which includes functions for traversing all trading equipments. This file connects to alphavantage to retrieve Forex and Digital currency data and update our equipment objects.
  15. I implemented email authentication, and then fixed it to make it redirect users to just our front-end page.
  16. I redesigned our navbar, changed the buttons..
  17. I added icons to our dropdown menu in navbar
  18. I changed the look of the profile page image holder.
  19. I redesigned our ‘Write New Article’ and ‘View Article’ pages. I added new article summary cards. Tried to make these pages more responsive. I added comment box designs to the article pages, like/dislike button designs. These comment boxes are later used on Trading Equipment pages too.
  20. I redesigned our Signup and Login pages to give it more organized look.

#### **For the first milestone:**

1. Initialized the Spring Boot project.
2. Set up the MongoDb database
3. Connected database with the Spring Project.
4. Created the User model.
5. Created the User repository.
6. Created Security configuration for the app. (JWT Authentication)
7. Modified the security configuration to separate the basic user, trader,
8. guest and the admin users.
9. Created register, login and get users endpoints.
10. Created the CORS configuration.
11. Wrote the documentation for the initial version of the backend API.
12. Migrated the database to the PostGreSQL.
13. After that, team decided to change the backend framework and we did not use the commits I made. Yet they are still in our repository.

Efe Önal	<p><b>From 2nd milestone to final milestone:</b></p> <ul style="list-style-type: none"> <li>17. I adapted the code so that it would work for guest and basic users as well as trader users.</li> <li>18. I implemented all portfolio functionalities: OwnPortfolioPage that renders all portfolios of a user, singleOwnPortfolio page that renders a single portfolio with its content, otherPortfolioPage that renders all portfolios of another user, singleOthersPortfolioPage that renders the content of a speceific portfolio of another user, addTreqModal that enables users to add trading equipment to their portfolios, createPortfolio that enables a user to create a portfolio, remove portfolio functionality, and follow portfolio functionality.</li> <li>19. I implemented all notification functionalities: I added a notification button to the navbar on which, new notification number appears in red, I created a notifPage component which renders the notifListHandler component that renders notifications according to their reason of creation: article_create, comment_create, or annotation_create. One can go to the profile page of the user who caused the annotation by clicking on the user button.</li> <li>20. I implemented search functionality for trading equipment and users in frontend.</li> <li>21. I implemented semantic search functionality for articles, comments, annotations, and events in frontend.</li> <li>22. I implemented the search results which are rendered according to their types which were listed on 5. and 6.</li> <li>23. I made various bug fixes on various parts of the code.</li> <li>24. I made various UI changes on various parts of the code.</li> </ul> <p><b>From 1st milestone to 2nd milestone:</b></p> <ul style="list-style-type: none"> <li>1. I adapted the code so that it would make good use of the local storage and life cycles.</li> <li>2. I added trader sign-up and first Google Geocode api which was then changed by other group members. The previous maps functionality took the address of the user as an address and posted the coordinates to backend.</li> <li>3. I helped restructuring the code.</li> <li>4. I implemented the main article page which renders all articles, single article page on which one can read the full article, and write article page where one writes the article.</li> <li>5. I implemented dynamic trading equipment graphs for frontend which are fed historical data that was gotten from backend and single trading equipment page with the comments.</li> </ul>

	<p>6. I implemented the events page in which upcoming events for the next three days are rendered.</p> <p>7. I implemented commenting functionalities for trading equipment and articles.</p> <p>8. I implemented email verification. If the user has not verified her account, she is routed to “please verify” page, otherwise she logs in successfully.</p> <p>9. I implemented update credentials functionality however all fields have to be filled, which I am going to fix.</p> <p>10. Our first implementations were all based on trader users. I adapted the code so that it would work for basic users as well.</p> <p>11. I fixed small bugs at all parts of the frontend.</p> <p><b>Until 1st milestone:</b></p> <ol style="list-style-type: none"> <li>1. I created the first navbar and differentiated it based on the user types, a different navbar was rendered for each user. I customized it with a dropdown menu and a search box, as well as our logo.</li> <li>2. I created the profile page component which is the main structure of a profile page.</li> <li>3. I created the profile area component which is rendered in Profile Page component. I customized its design.</li> <li>4. I added update feature via an update button in the profile area.</li> <li>5. After UI design phase was complete, I made the successful login redirect the user into her profile page with her credentials gotten with a get method.</li> <li>6. I made research on Google Login and Google Maps features.</li> <li>7. We made three online meetings as the frontend team in order to be able to work in harmony.</li> </ol>
Ceren Tahtasız	<p><b>From 2nd milestone to final milestone:</b></p> <ol style="list-style-type: none"> <li>1. Checked all functionalities implemented by black box testing. Found bugs in Assets and reported them.</li> <li>2. Prepared project plan using ProjectLibre from the beginning of the term to the end of project.</li> <li>3. Prepared user scenarios to present in milestone demo.</li> <li>4. Made a research on W3C Annotation Standard for our own annotation functionality to be compliant with it.</li> <li>5. Reformatted and cleaned the new written code for aligning with our coding standards by peer programming.</li> </ol> <p><b>From 1st milestone to 2nd milestone:</b></p> <ol style="list-style-type: none"> <li>1. I fixed the search and edit buttons in profile page to prevent app crash.</li> </ol>

- |  |   |
|--|---|
|  | <ol style="list-style-type: none"> <li>2. I connected the register page with profile page so that when a user signs up, they don't have to go back and login: they're directly logged in and in their Home page.</li> <li>3. I implemented Google Maps functionality to choose location.</li> <li>4. I decided visual standards with mobile and web team together.</li> <li>5. I deleted unused layouts after Volkan revised the structure.</li> <li>6. I created user scenarios with mobile team to show in demo.</li> </ol> |
|--|---|

**Before 1st Milestone:**

- |  |   |
|--|---|
|  | <ol style="list-style-type: none"> <li>1. Updated requirements document and mock-up according to feedback from new members.</li> <li>2. Updated communication plan and ReadMe file.</li> <li>3. Made research on tools for mobile, git workflows and testing types for whole team.</li> <li>4. Planned the design of layouts and flow of activities for mobile.</li> <li>5. Implemented register layout: this page gives information to users on trader and basic user and when you click one, you go to corresponding registration form.</li> <li>6. Implemented profile page layout for users which shows profile image, birthday, user name and mail. There's a button to edit profile.</li> <li>7. Implemented profile page connection with back-end endpoints to retrieve data to show on user's profile. (layout and backend connection)</li> <li>8. Implemented edit profile page (layout and backend connections)</li> <li>9. Tested sign-in, sign-up, edit profile functionalities using black box testing.</li> <li>10. Created milestone presentation scenario and made the presentation.</li> </ol> |
|--|---|

## 5. Annotation Implementation & W3C standard Compliance

In Mercatus, article pages are annotable to some extent. Our team decided to make all annotations public and reachable by other users. Unfortunately, there are some restrictions on the annotable content inside the articles. An annotated slice of the text is not re-annotable again by another user, until the current occupance is removed. This rule goes for both the slice itself, and the smaller parts of it. Below the annotating process and the comparison with the W3C standard are discussed.

### **5.1: Annotating the Mercatus articles**

A part of the text is selected from the article. Annotate button appears at the end of the selection.

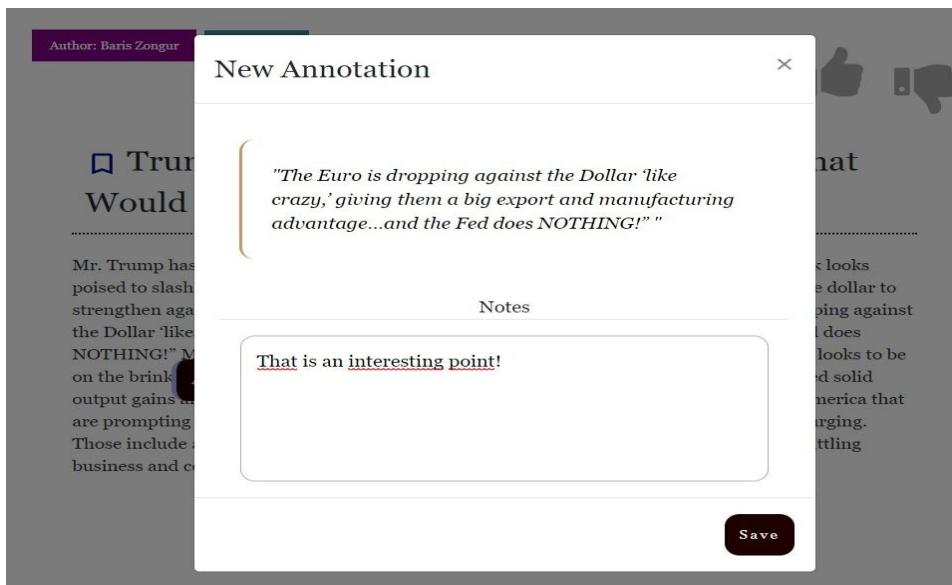
The screenshot shows a news article titled "Trump Wants Negative Rates. Here's How That Would Work" by Baris Zongur. The article discusses Mr. Trump's interest in negative interest rates and their potential impact on the US economy. A specific sentence is highlighted with a blue box: "The Euro is dropping against the Dollar 'like crazy,' giving them a big export and manufacturing advantage...and the Fed does NOTHING!" Below this sentence, there is an "Annotate" button. At the top of the article, there are author information ("Author: Baris Zongur") and rating controls ("Rating: 1" with a heart icon). Below the rating are two large, semi-transparent thumbs-up and thumbs-down icons.

Author: Baris Zongur    Rating: 1

Trump Wants Negative Rates. Here's How That Would Work

Mr. Trump has been attuned to negative rate policies abroad as the European Central Bank looks poised to slash a key rate further below zero. Mere anticipation of that move has caused the dollar to strengthen against the euro, making American exports less competitive. **The Euro is dropping against the Dollar 'like crazy,' giving them a big export and manufacturing advantage...and the Fed does NOTHING!"** Mr. Trun [REDACTED] Aug. 30. Germany is Europe's largest economy, and it looks to be on the brink of sinking into recession. The United States, on the other hand, has maintained solid output gains and strong consumer spending. That said, there are risks on the horizon in America that are prompting the Fed to lower rates — albeit much more slowly than the White House is urging. Those include a global manufacturing slowdown and Mr. Trump's trade wars, which are rattling business and consumer sentiment.

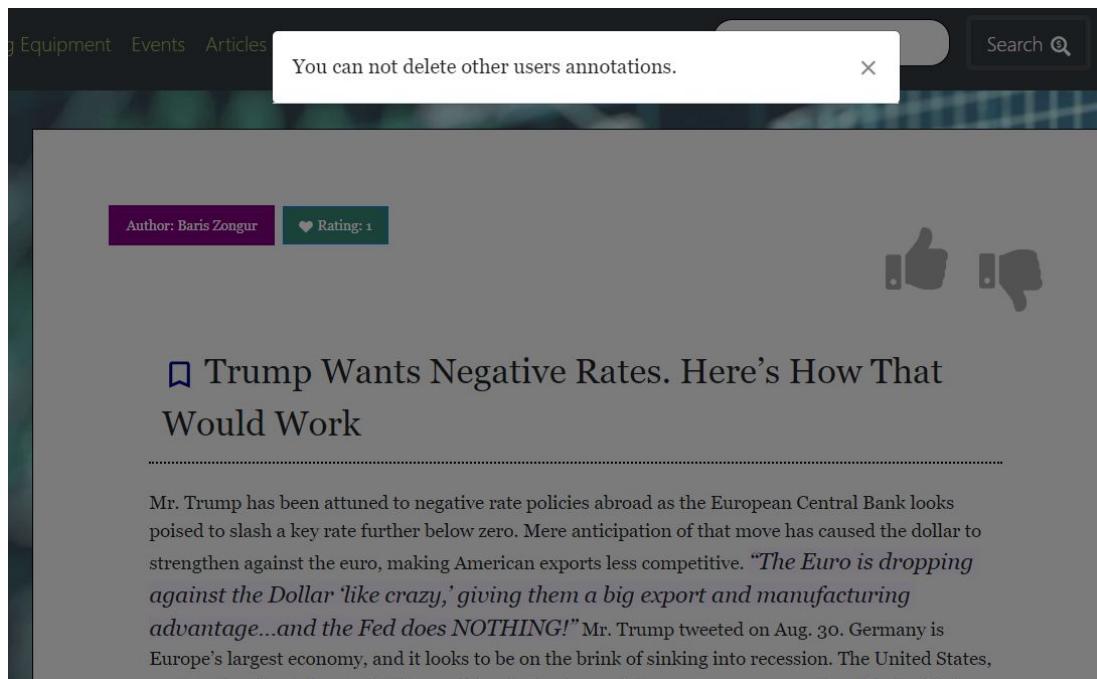
After the button press, a modal for new annotation appears.



After saving, annotation is displayed to the user with a new style. Clicking on the colored section will display the notes related to the selected text, and the note owner. Owner of the annotation is also able to remove the annotation.

A screenshot of a news article titled "Trump Wants Negative Rates. Here's How That Would Work". The article discusses Mr. Trump's interest in negative interest rates and their potential impact on the dollar and exports. A callout box highlights the quote "The Euro is dropping against the Dollar 'like crazy,' giving them a big export and manufacturing advantage..." with the note "That is an interesting point!" attributed to "Furkan Aydar".

If another user other than the owner of the annotation tries to remove the annotation, an error appears.



## 5.2: W3C Compliance

Our annotations support some selected requirements of W3C Annotation Protocol. Only supported Body type for the annotations is Text and it is applied for articles page only. Target values for annotations are also Text only. Our selector implementation for the Text body complies with W3C (it is based on the starting index and the ending index of the selected text, relatively to the main article)

A contradiction with the W3C standard is, our annotations are not editable; rather user needs to remove the current one and select the same fragment and annotate again.

Annotations related to the specific article, are also stylized as the W3C Model suggests. We used background colors for texts to indicate an annotation exists at the fragment.

# 6. Project Requirements

## 1. Functional Requirements

### 1.1. User Requirements

#### 1.1.1. Guests

- 1.1.1.1. The system shall allow the guests to view the price of any trading equipment.
- 1.1.1.2. The system shall allow the guests to read user comments about trading equipment.
- 1.1.1.3. The system shall allow the guests to view articles and annotations in articles.

#### 1.1.2. User Authentication and Account Management

- 1.1.2.1. Sign Up
  - 1.1.2.1.1. The system shall allow the user accounts to be either "basic" type or "trader" type.
  - 1.1.2.1.2. The system shall require the user to provide their name, surname, email address and location to be able to create a **basic** account.
  - 1.1.2.1.3. The system shall require the user to provide their name, surname, email address, location, id number and IBAN to be able to create a **trader** account.
  - 1.1.2.1.4. The system, when it requires the user to provide their location, shall require the user to provide it using Google Maps.
  - 1.1.2.1.5. The system should allow the user to use their Google account to provide the necessary sign up information which are available in their Google account and require the rest of the necessary information to be input using text.
  - 1.1.2.1.6. The system shall obtain the necessary sign up information using text inputs, unless otherwise specified.
  - 1.1.2.1.7. The system shall allow users to change their account type when necessary information is provided. Trader user can downgrade to Basic user and Basic user can upgrade to Trader user.
- 1.1.2.2. Sign In
  - 1.1.2.2.1. The system shall allow users to sign in only using their Google accounts or by providing the necessary sign in information, username and password.

#### 1.1.3. Trading

- 1.1.3.1. The system shall, for the trading users, have a "My Investments" section which shall have the functionalities specified here (1.1.3).
- 1.1.3.2. The system shall allow the trading users to invest on any trading equipment.

- 1.1.3.3. The system shall allow the trading users to make a buy order for a specified rate.
- 1.1.3.4. The system shall allow the trading users to set stop/loss limits.

#### 1.1.4. Tracking

- 1.1.4.1. The system shall provide each user with a "Profit/Loss" section that is private to the respective users, and this section shall have the functionalities specified here (1.1.4).
- 1.1.4.2. The system shall show both basic and trading users their profit/loss amount, in terms of the currency they choose, by the user manually entering their investments.
- 1.1.4.3. The system shall show trading users their profit/loss amount, in terms of the currency they choose, by the user's trading history on the system.
- 1.1.5. Comments
- 1.1.5.1. The system shall allow users to comment and annotate on articles.
- 1.1.5.2. The system shall allow users to comment on trading equipment.

### 1.2. System Requirements

#### 1.2.1. Search

- 1.2.1.1. System shall support searching for users and trading equipments considering all the information available and filtering the search results.
- 1.2.1.2. System shall support semantic search.
- 1.2.1.3. System shall support location based search.

#### 1.2.2. Recommendation

- System shall have a recommendation system that recommends articles or trading equipment to users based on their history (what they already follow).

#### 1.2.3. Ranking

- 1.2.3.1. System shall allow traders to rank others idea.

#### 1.2.4. Trading Equipment

- 1.2.4.1. System shall allow traders to trade indices, stocks, ETFs, commodities, currencies, funds, bonds, and cryptocurrencies.
- 1.2.4.2. Each trading equipment shall include many functionalities, including but not limited to: the previous close, percentage change with the previous close, amount change with the previous close, day's range, and moving averages.

#### 1.2.5. Communication

- 1.2.5.1. System shall allow users to follow other users and trading equipment.
- 1.2.5.2. System shall allow users to share their ideas as articles.

- 1.2.5.3. System shall allow users to comment and rate ideas of other users.
- 1.2.5.4. System shall allow users to comment about trading equipment.

#### 1.2.6. Profit

- 1.2.6.1. System shall have a Profit/Loss section for each user.
- 1.2.6.2. This section should be private for each user.
- 1.2.6.3. System shall allow basic users to see their profit/loss amount in terms of the currency they choose by manually entering their investments.
- 1.2.6.4. System shall allow trading users to see their profit/loss amount in terms of the currency they choose by both manually entering their investments and using the investments they made in the Traders Platform.

#### 1.2.7. Portfolio

- 1.2.7.1. System shall ensure each user has at least one portfolio.
- 1.2.7.2. Portfolios shall be able to renamed, edited, shared and followed (if already shared).
- 1.2.7.3. System shall allow users to add any trading equipment to their portfolios.

#### 1.2.8. Notifications

- 1.2.8.1. System shall allow users to set alerts for certain levels of trading equipment.
- 1.2.8.2. System shall notify users in accordance with their alerts.

#### 1.2.9. Account

- 1.2.9.1. System shall be provided necessary information for signing up. (Basic users are expected to provide their name, surname, email address, and location.)
- 1.2.9.2. System shall have specific location of users on Google Maps.
- 1.2.9.3. System should use Google account to retrieve necessary information for signing up/in.
- 1.2.9.4. System shall offer basic functionality to users after signing in and validating their email address.

#### 1.2.10. Event

- 1.2.10.1. System shall have events section for users which contains economic events.
- 1.2.10.2. Events shall have significance level and country base properties.
- 1.2.10.3. Events shall be able to searched, filtered and chased.

### **1.2.11. Deployment**

- 1.2.11.1. System shall be able to handle native web and native mobile (Android) clients. (Hybrid applications are not allowed.)
- 1.2.11.2. System shall be deployable on a remote and manually configurable server. (Amazon EC2 or Digital Ocean are strongly recommended.)

### **1.2.12. Standards**

- 1.2.12.1. System shall support W3C Web Annotation Data Model.
- 1.2.12.2. System shall follow W3C Web Annotation Protocol.
- 1.2.12.3. System shall follow the standards introduced by the World Wide Web Consortium (W3C).

### **1.2.13. Ethics and Legal**

- 1.2.13.1. System shall handle personal information, contact information, copyrighted contents and license issues according to legal requirements.

## **2. Non-Functional Requirements**

### **2.1. Accessibility**

- 2.1.1. The system shall be accessible on Android and Web.
- 2.1.2. The system shall be compatible with at least Chrome 70, Firefox 64, IE 11 or higher version.
- 2.1.3. The system shall be compatible with Android 5.1 or higher version.
- 2.1.4. The system shall have responsive UI for different screen resolutions for both Android devices and web browsers.

### **2.2. Availability**

- 2.2.1. The system shall have at least 97% uptime. (Except during maintenance or unexpected errors)
- 2.2.2. The system should be under maintenance every Tuesday between 2.00 am and 2.15 am regularly.
- 2.2.3. If there exists an ongoing maintenance, the system should inform users about the estimated termination of the construction at least one day before maintenance.

### **2.3. Reliability**

- 2.3.1. The system should generate a proper log file which includes server reports, error logs and user activities when an error occurs.
- 2.3.2. The system should create backup file everyday.

## 2.4. Security

- 2.4.1. The system should remind users to change their password after using it for 1 year.
- 2.4.2. The system should require passwords to contain number of characters between 8 and 20; must contain at least one lowercase letter, one uppercase letter, one numeric digit, but cannot contain whitespace.
- 2.4.3. The system should not allow user to login for 10 minutes after 5 unsuccessful login attempts.
- 2.4.4. The system should session expiration if a user is inactive for 5 minutes.
- 2.4.5. The system should have human verification tool (reCAPTCHA) to prevent bots.

## 2.5. Privacy

- 2.5.1. The system shall secure the private data of users according to [Information Privacy Law](#).

## 2.6. Database

- 2.6.1. The system shall encrypt personal data of users with RSA before storing them in the database.
- 2.6.2. The database architecture should have trigger functionality in the case of unexpected network issues in the middle of value exchanges between users.
- 2.6.3. The database architecture should handle 97% of the queries less than 0.1 second.

# 7. User Scenarios Used in Presentation

### Mobile Scenario:

#### Mehmet Yerli

Mehmet Yerli has a company in the chemical industry. He wants to learn more about different commodities. He has already installed ***Mercatus***. He wants to read some articles in order to buy a commodity.

Preconditions: Mehmet Yerli is registered trader user.

1. Mehmet Yerli opens Mercatus and goes to **articles** section.
2. Mehmet Yerli reads an article about recent **crude oil** updates.
3. Mehmet Yerli **likes** this article , makes a comment about the same idea and **follows** the author.

4. At the same time, Sevim Sezgi gets an **notification** for the article because of comments.
5. Mehmet Yerli opens **events** section and checks the latest events about petroleum.
6. Mehmet Yerli opens **Values** section and finds crude oil. Then clicks on it to view past values from latest month. He makes prediction and writes a comment.
7. Mehmet Yerli goes **asset** section and add 100\$ cash for himself.
8. Mehmet Yerli buys commodities **crude oil** for 50\$.
9. Mehmet Yerli **creates a portfolio**.
10. Mehmet Yerli **adds** a trading equipment for his **portfolio**.
11. Mehmet Yerli **deletes** the portfolio.

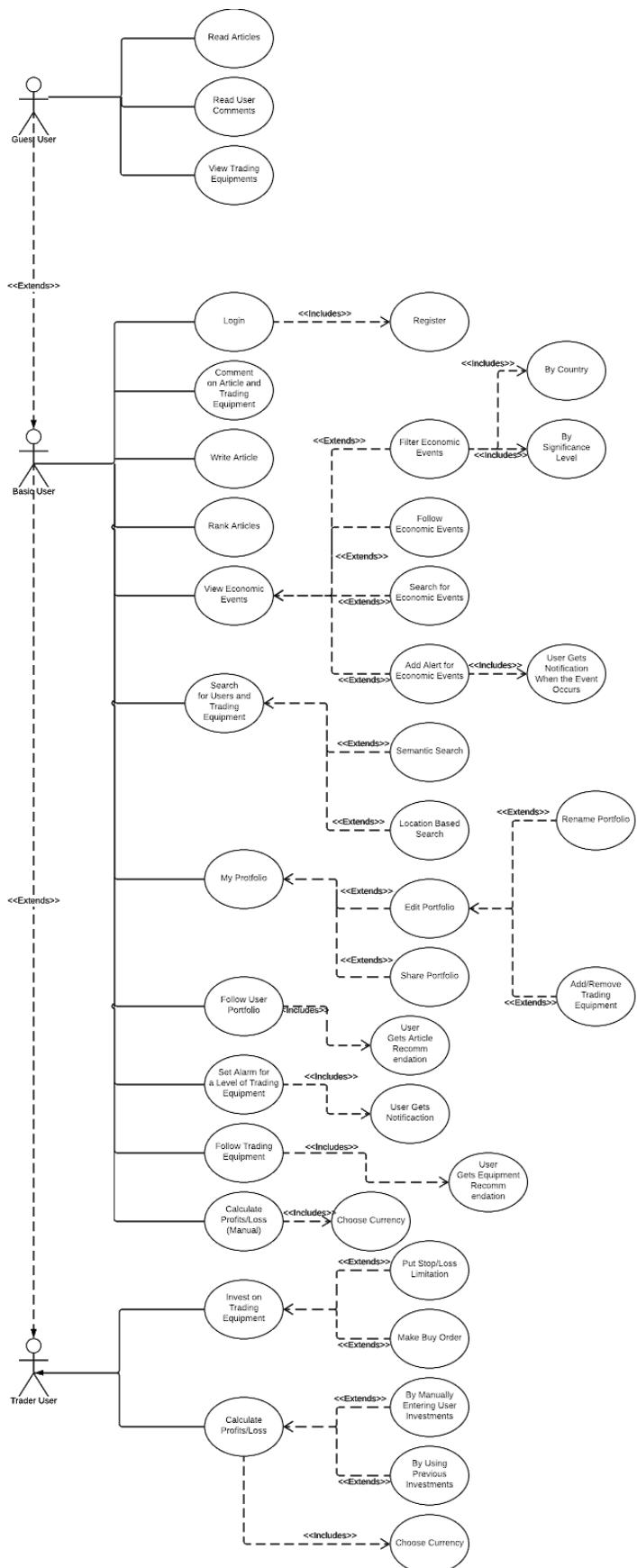
#### **Web Scenario:**

Ahmet Kazancı is an old successful businessman who has been investing for a long time. He uses Mercatus Trading Application for learning about stocks, and making the transaction. He loves the Mercatus App.

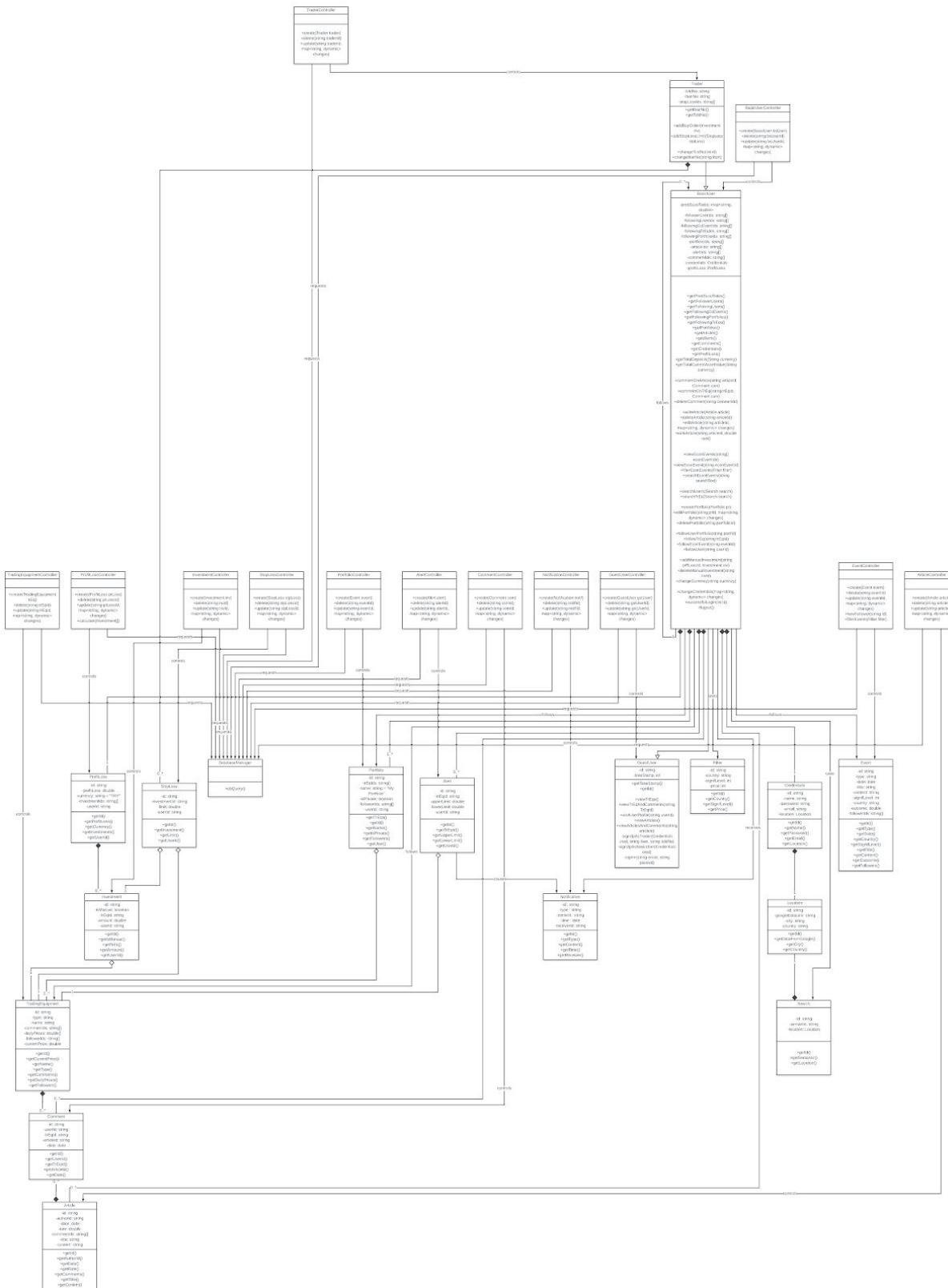
1. Ahmet makes a search for a word.
2. An article is found semantically.
3. He annotates to the article.
  
1. Can Uzun is new to trading.
2. He is also a follower of Ahmet Kazancı.
3. Can goes to the Portfolio section and creates a portfolio.
4. Then he adds GBP to his portfolio since he is interested in buying some.
5. He goes to the equipment page for GBP and makes an investment.
6. Ahmet then goes to his profile page to check if he can see the GBP he bought in his assets section.
7. He realized that there is a section for recommendation, so he checks what kind of recommendations does the App have for him.
8. Finally, he clicks on notifications and observes the annotation of the previous scenario.

## 8. Diagrams

### 8.1. Use Case Diagram

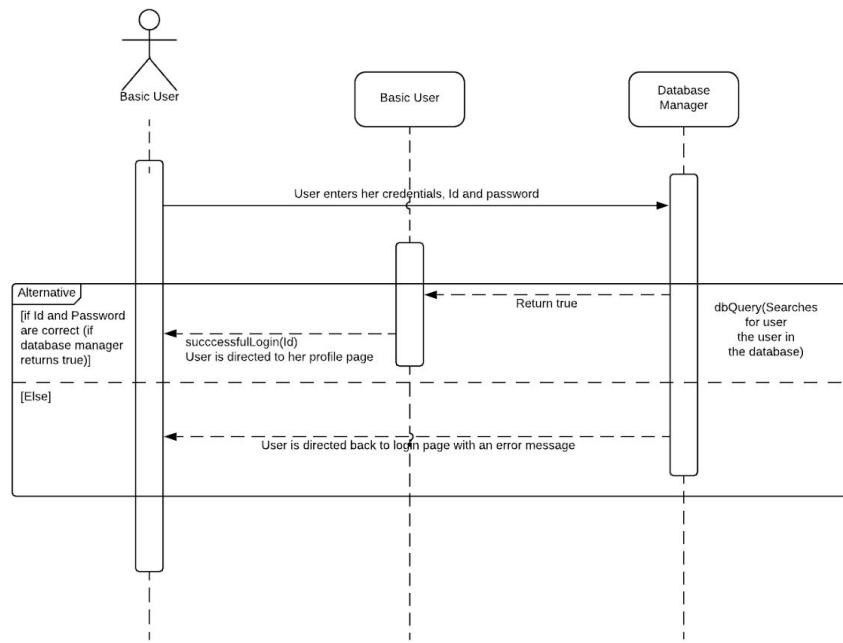


## 8.2. Class Diagram

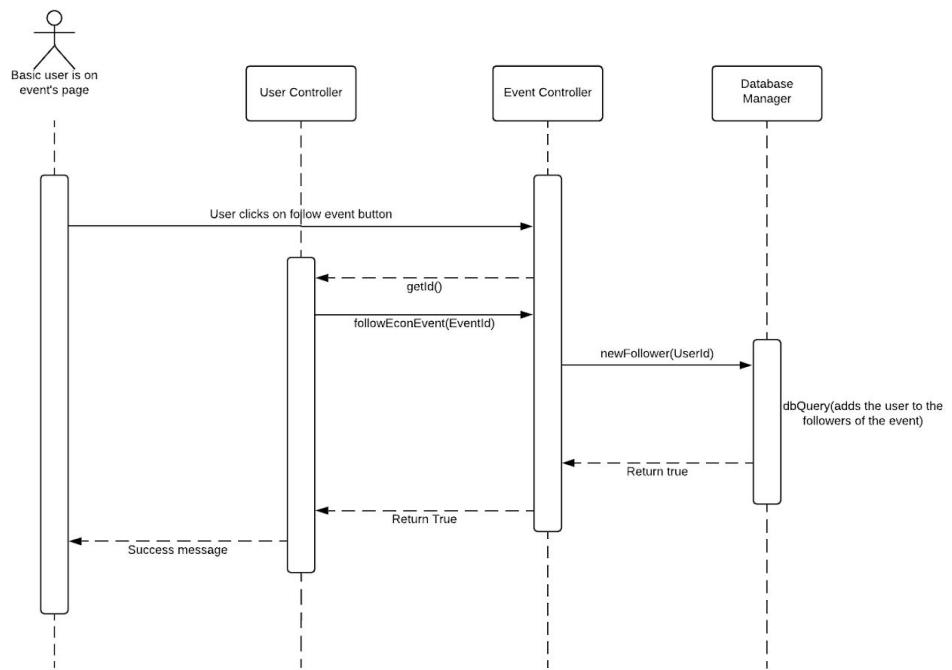


## 8.3. Sequence Diagrams

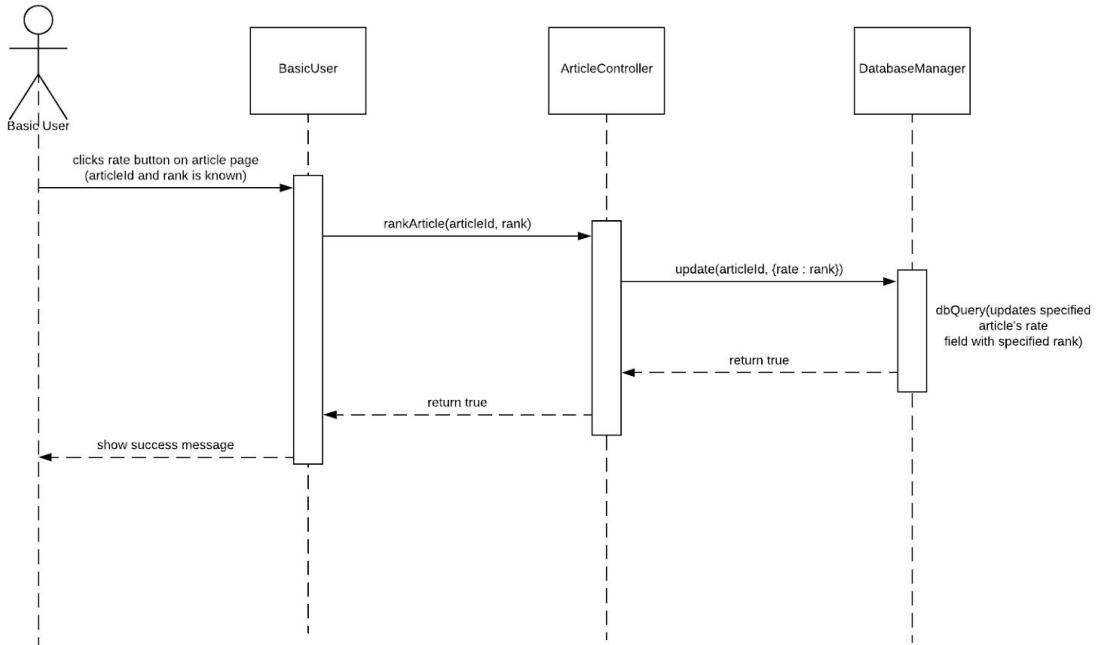
### 8.3.1. Login



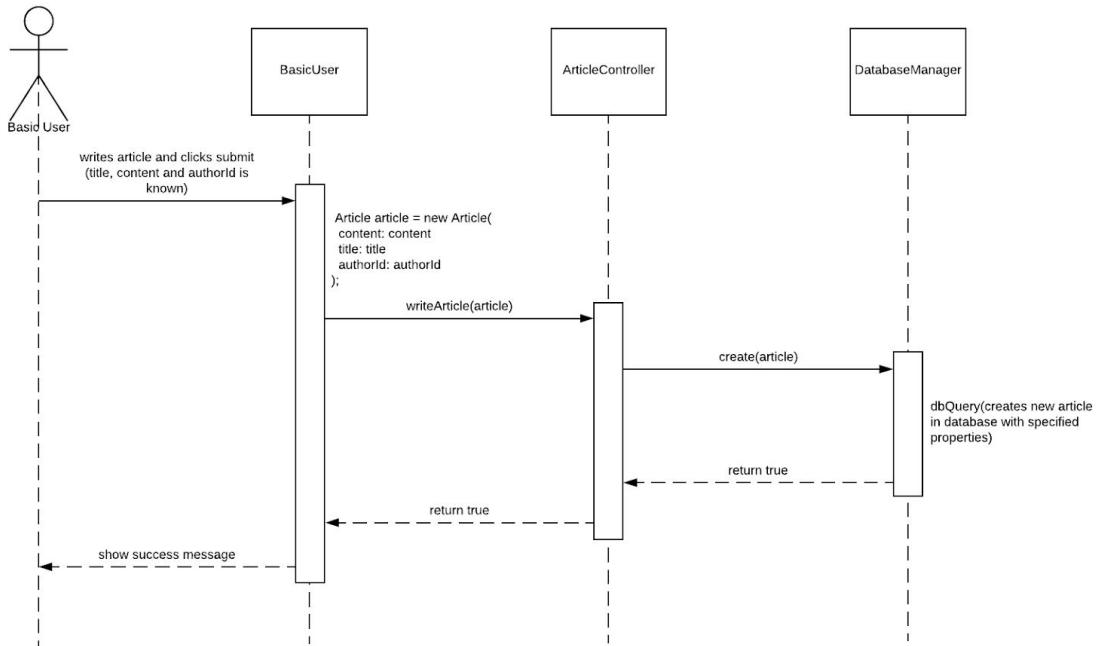
### 8.3.2. Following an Economic Event



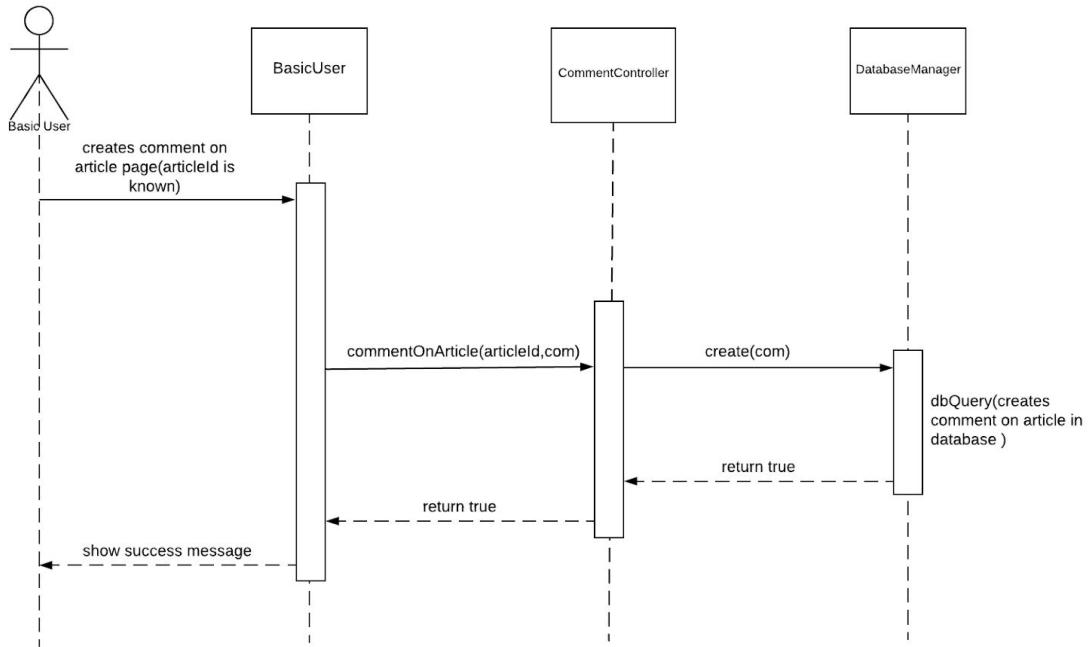
### 8.3.3. Rating an Article



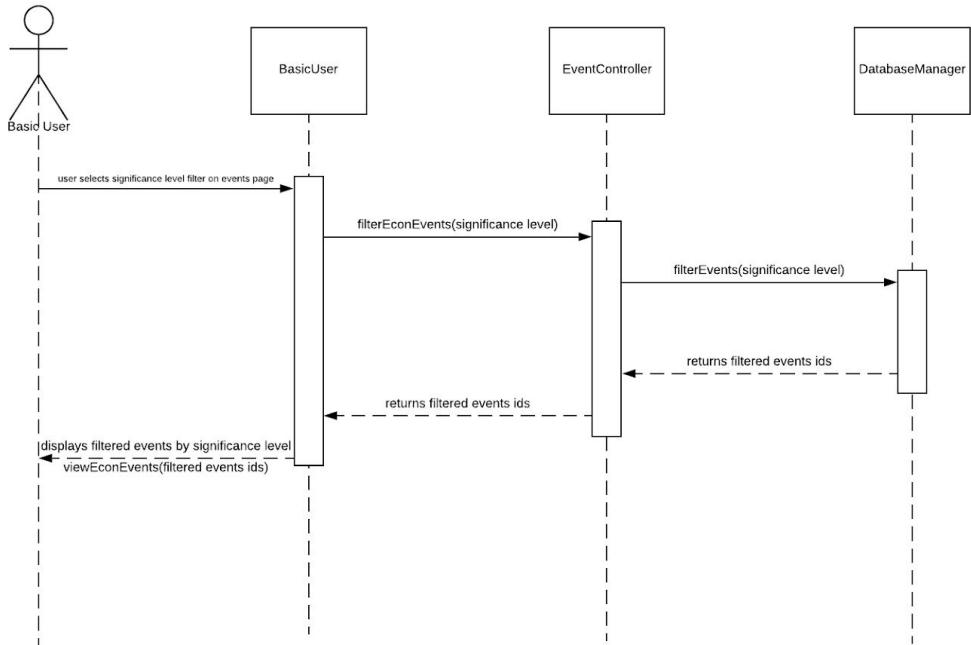
### 8.3.4. Writing an Article



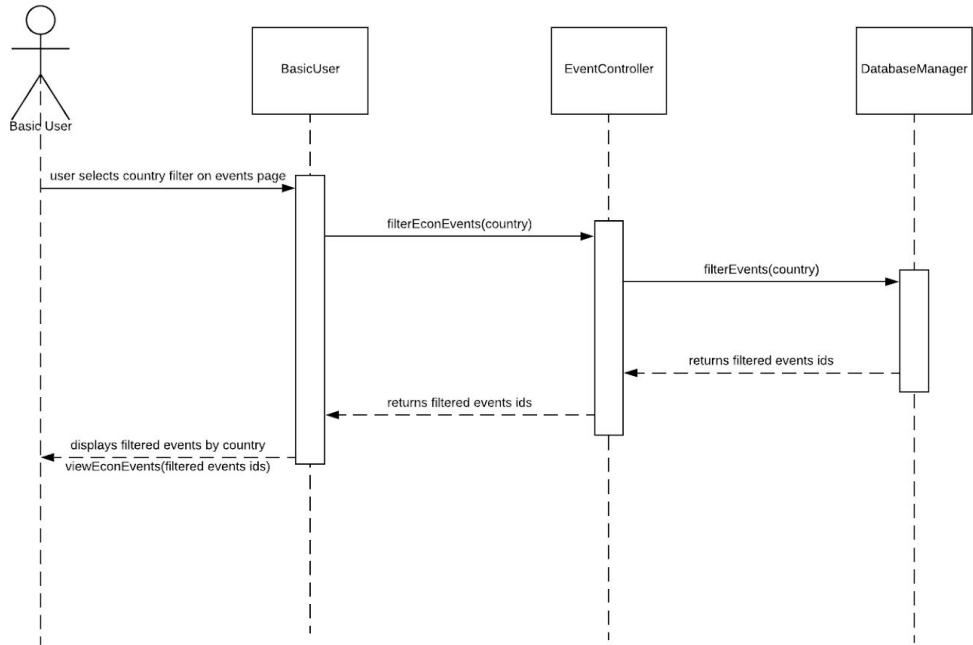
### 8.3.5. Commenting on Article



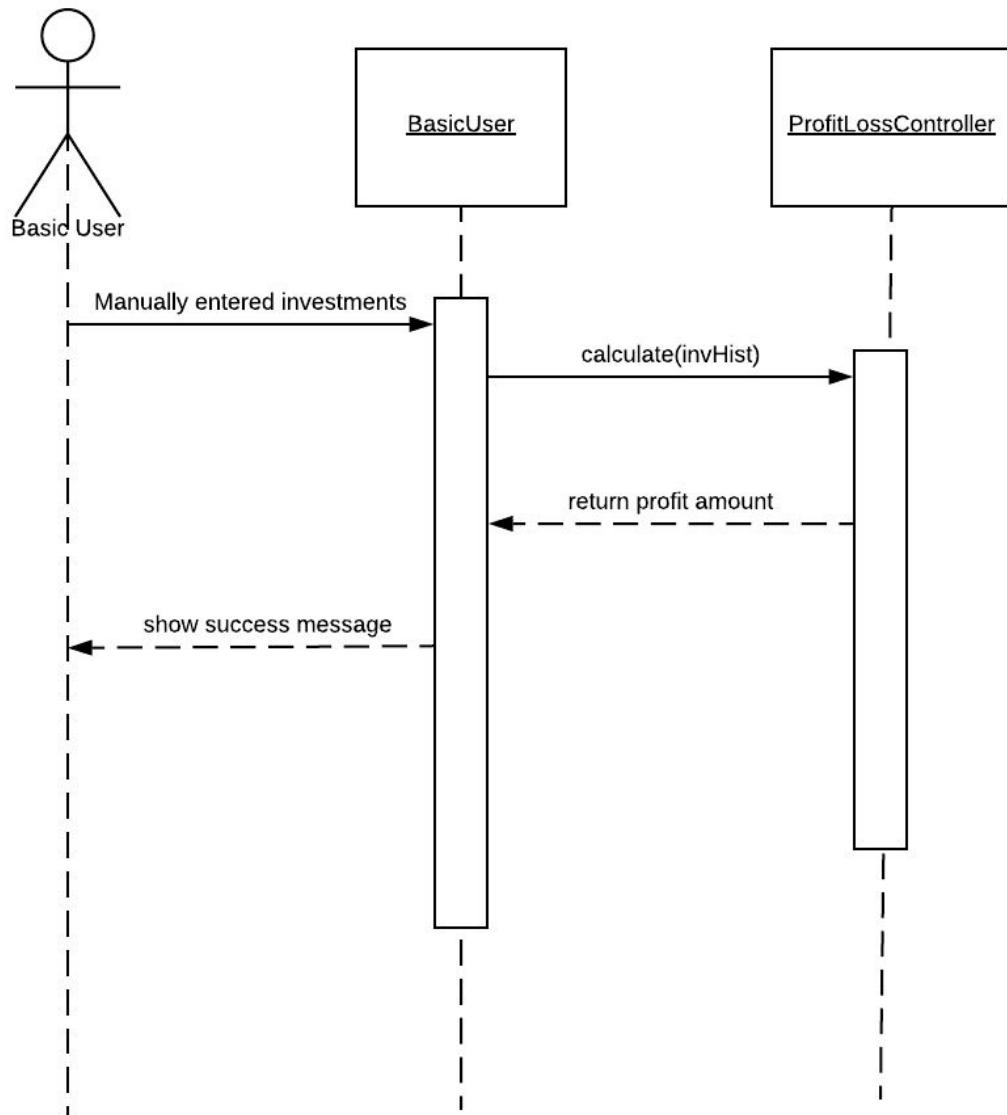
### 8.3.6. Filtering Events by Country



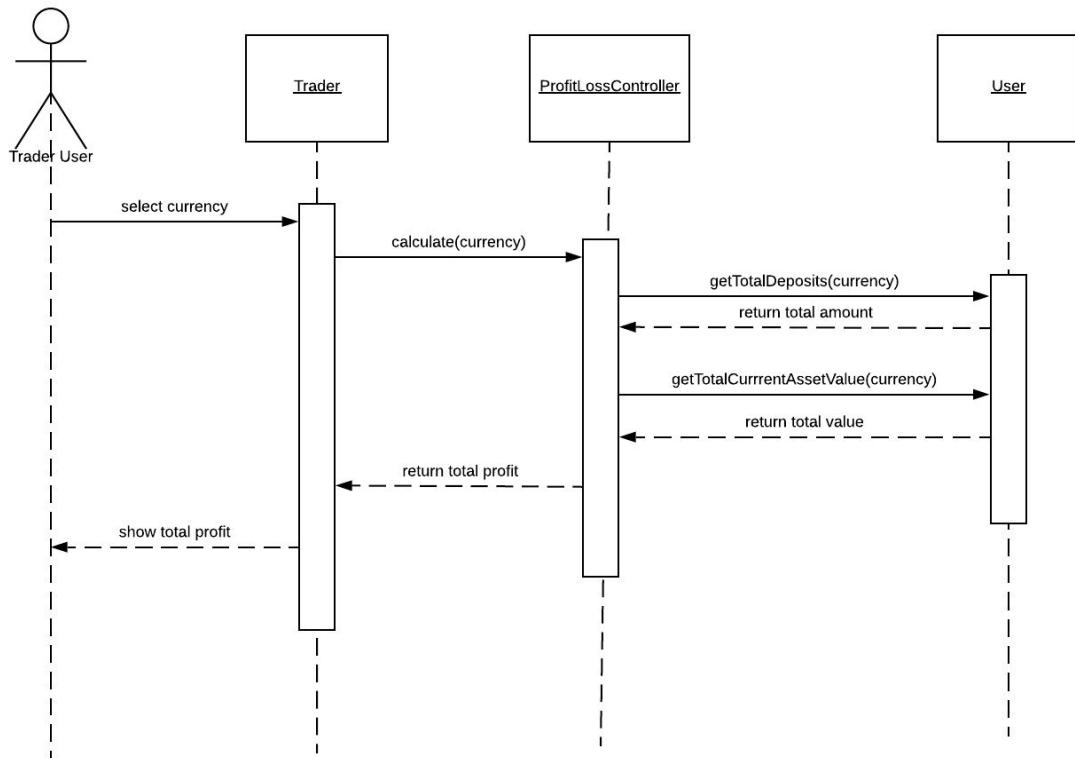
### 8.3.7. Filtering Events by Significance Level



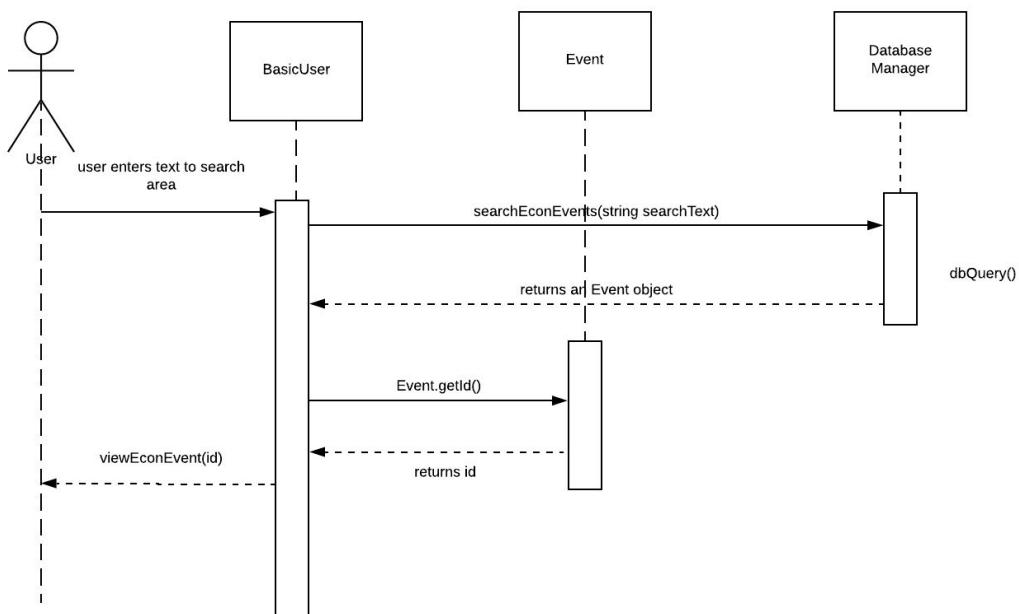
### 8.3.8. Calculate Profit/Loss Manually



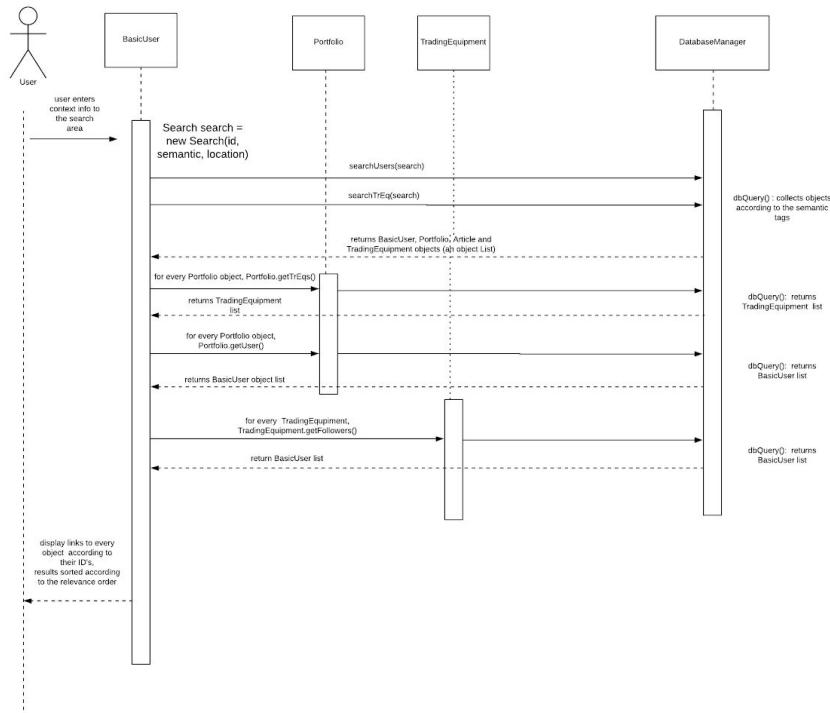
### 8.3.9. Calculate Profit/Loss



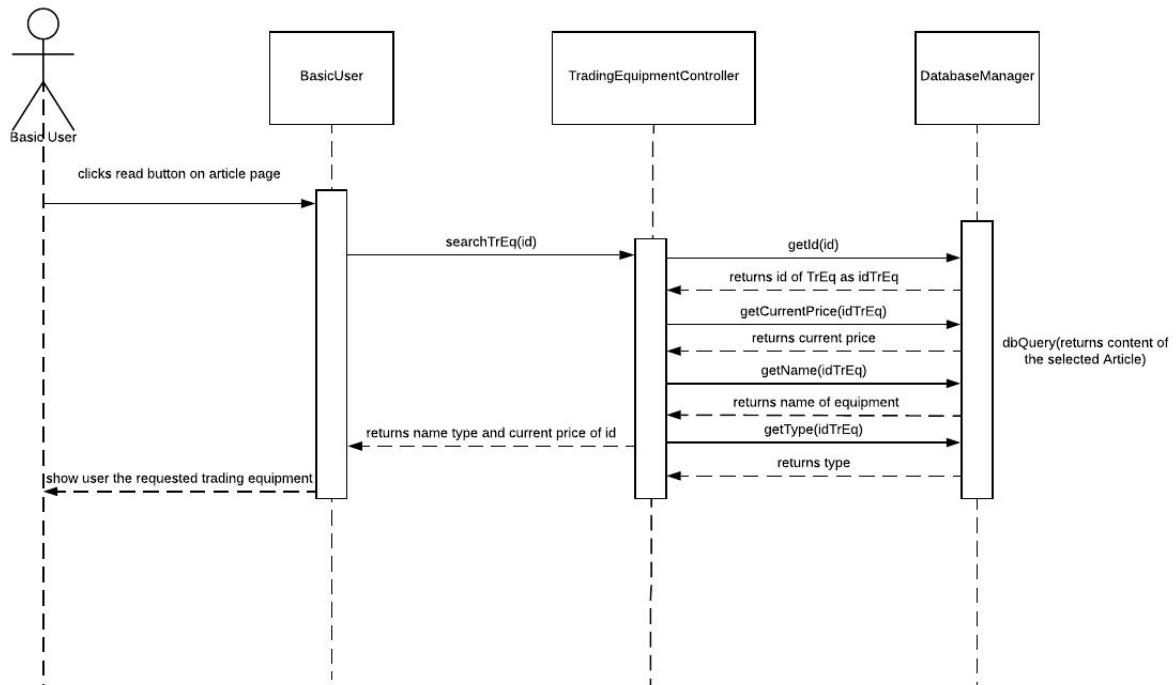
### 8.3.10. Search Economic Events



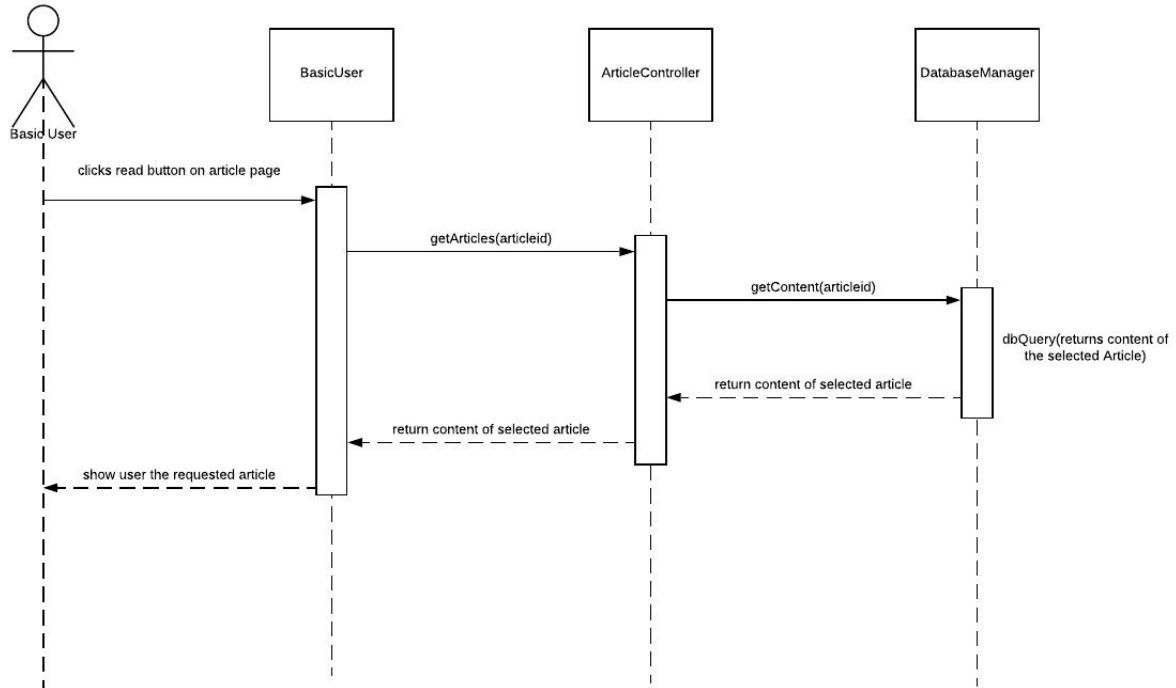
### 8.3.11. Semantic Search For Users/Trading Equipments



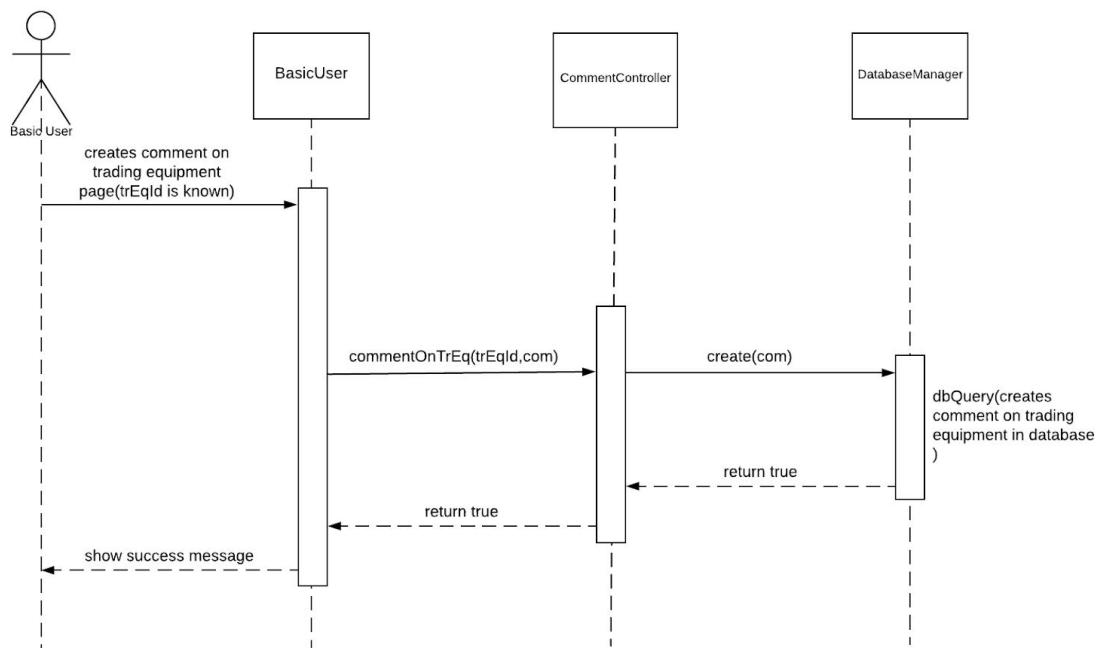
### 8.3.12. View Trading Equipment



### 8.3.13. Read an Article



### 8.3.14. Commenting On Trading Equipment

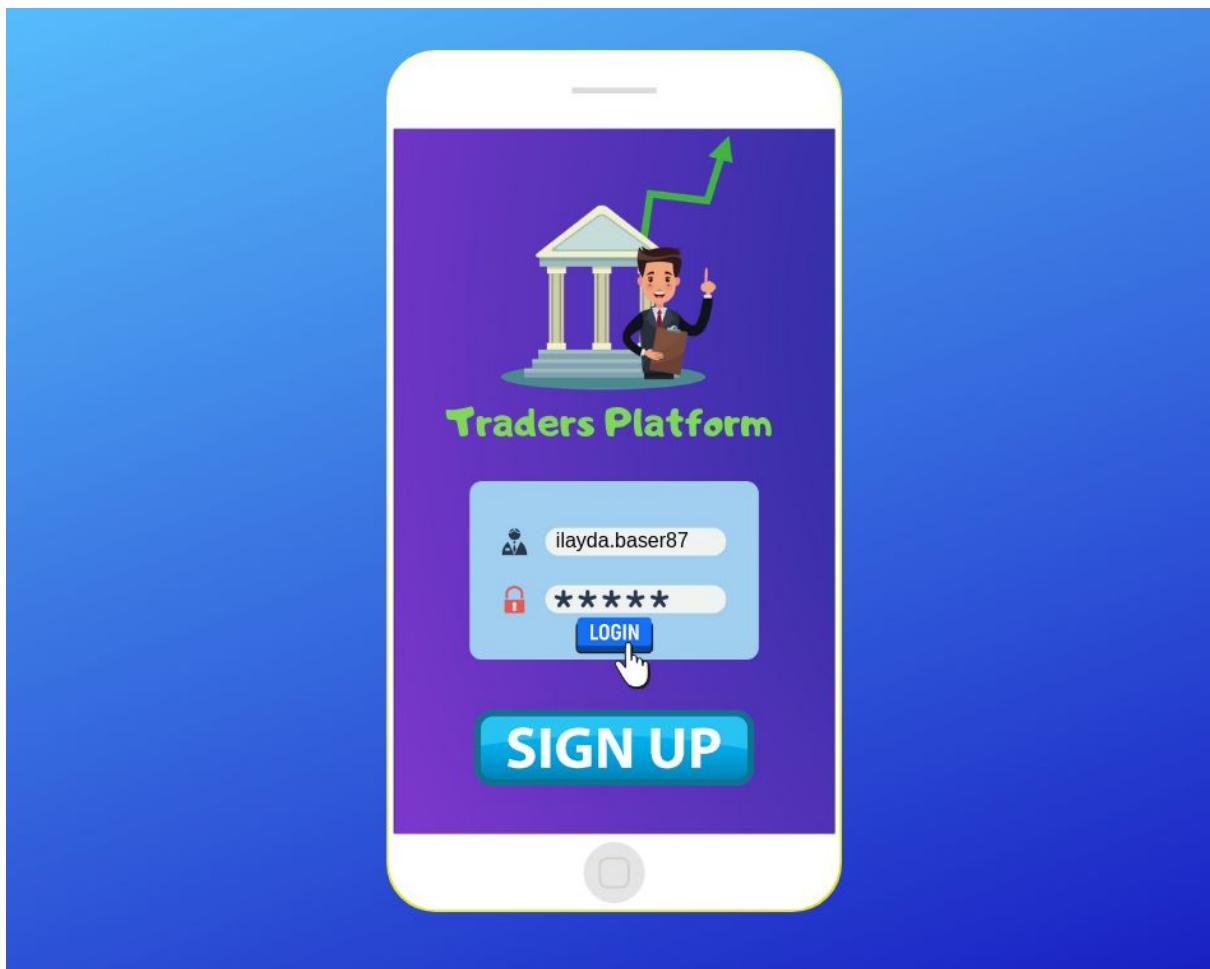


## 9. Mockups

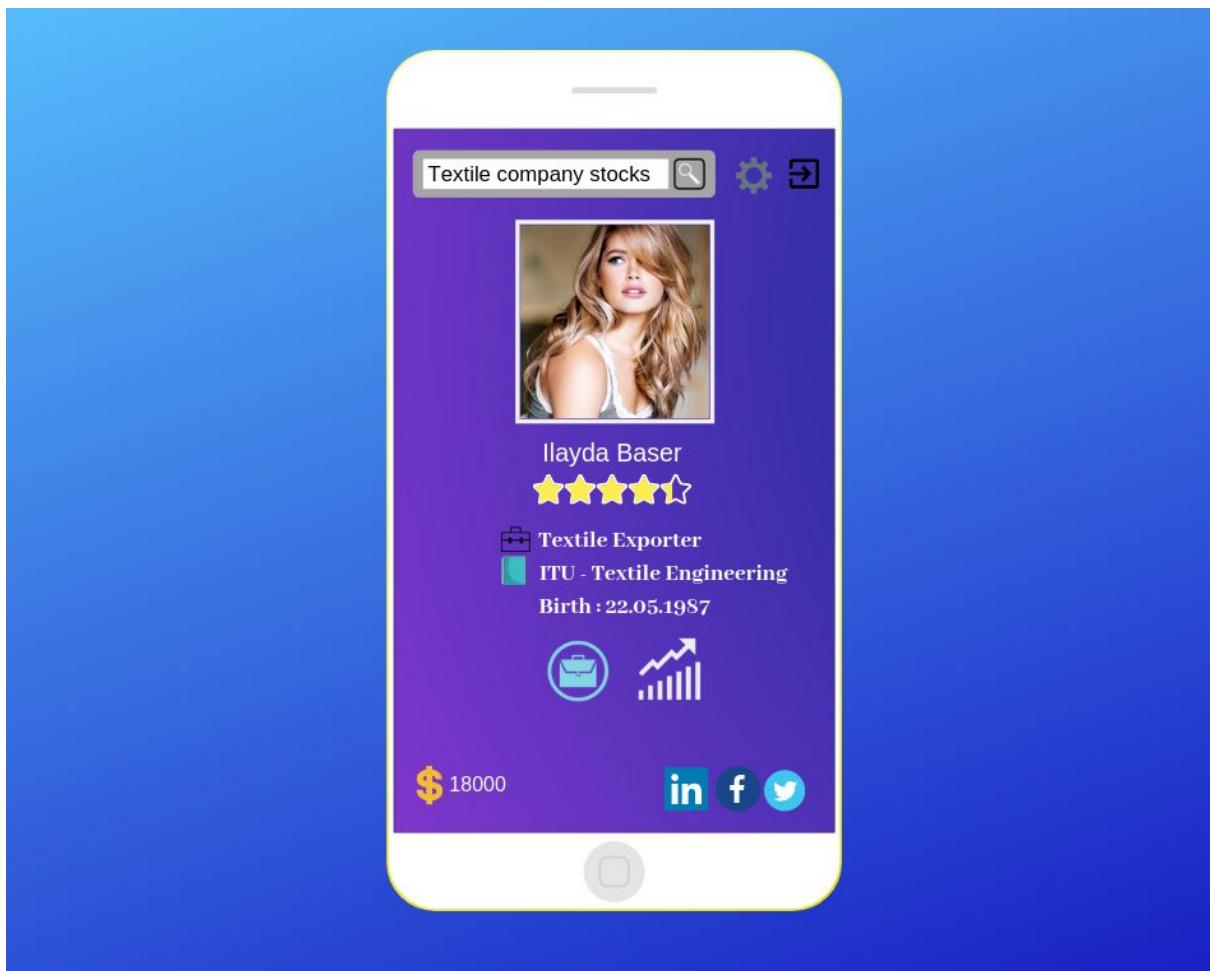
### 9.1. Scenario 1 (Ilayda Baser)

#### Flow

- Ilayda opens the android application and enters her credentials.



- After login, Ilayda is on the profile page and she searches for textile company stocks. She can go for her portfolio via the button below. Her weighted prediction success rate can be observed by the gold stars while she could use the graphics button below in order to see the details on her prediction success rate.



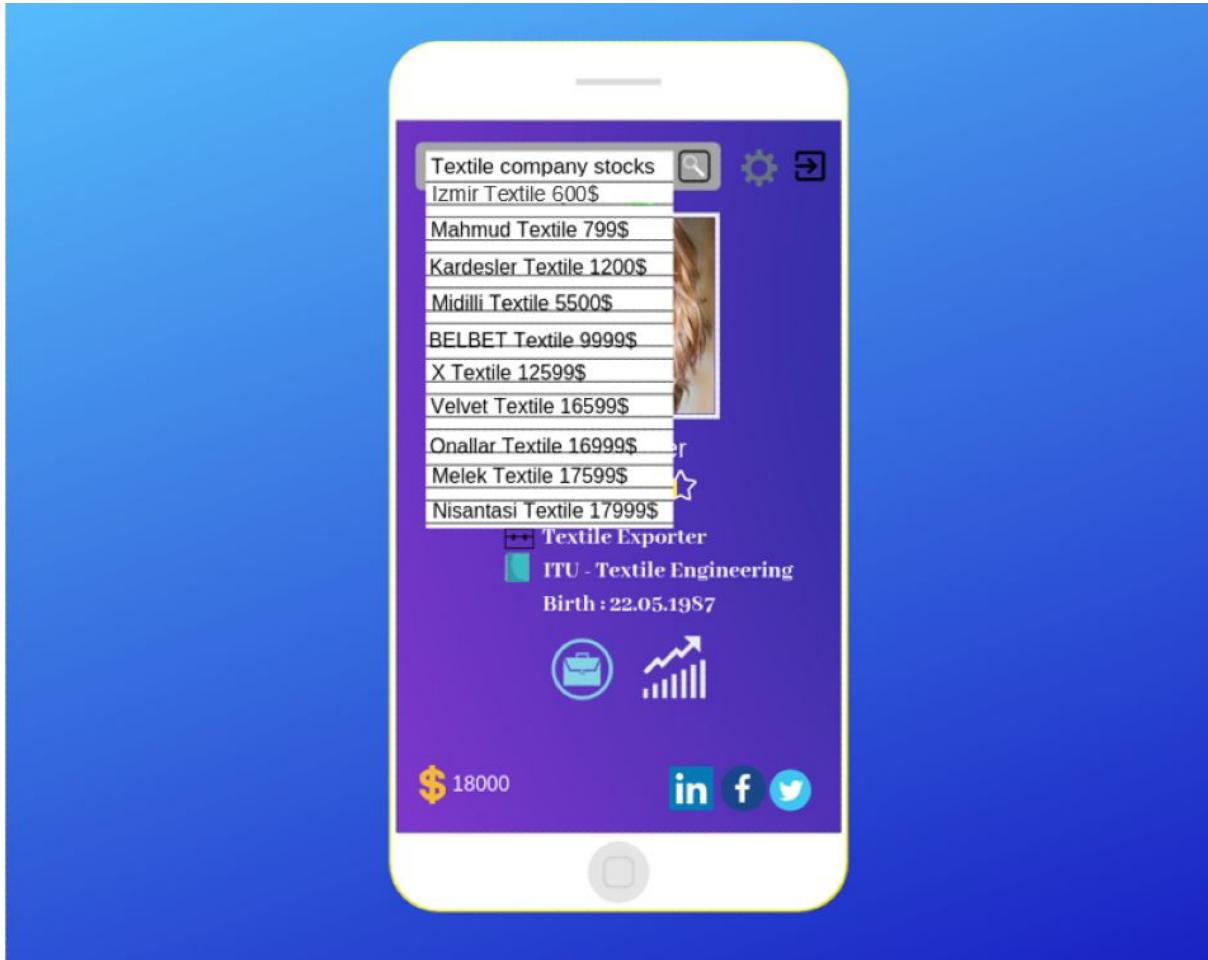
## 9.2. Scenario 2 (Ilayda Baser)

### Preconditions

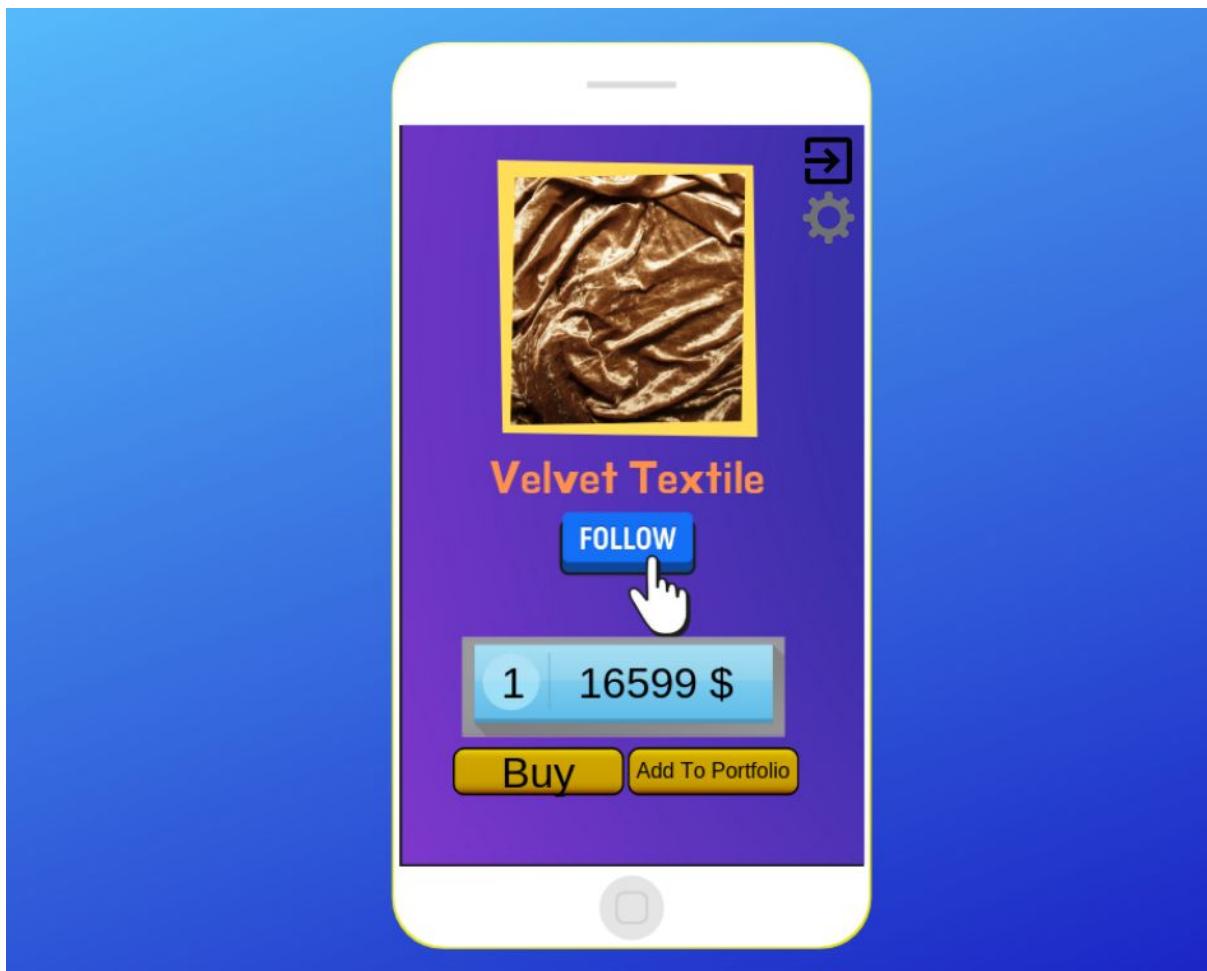
- Ilayda has already logged into the system and she is on her profile page.

### Flow

- Ilayda searches for textile company stock. She picks an offer of her choice (Velvet Textile) from the increasingly ordered list.



- After picking the offer, Ilayda is at the purchasing page. She chooses one stock of Velvet Company, she buys it. Then she follows Velvet Company and adds her stock to her portfolio.



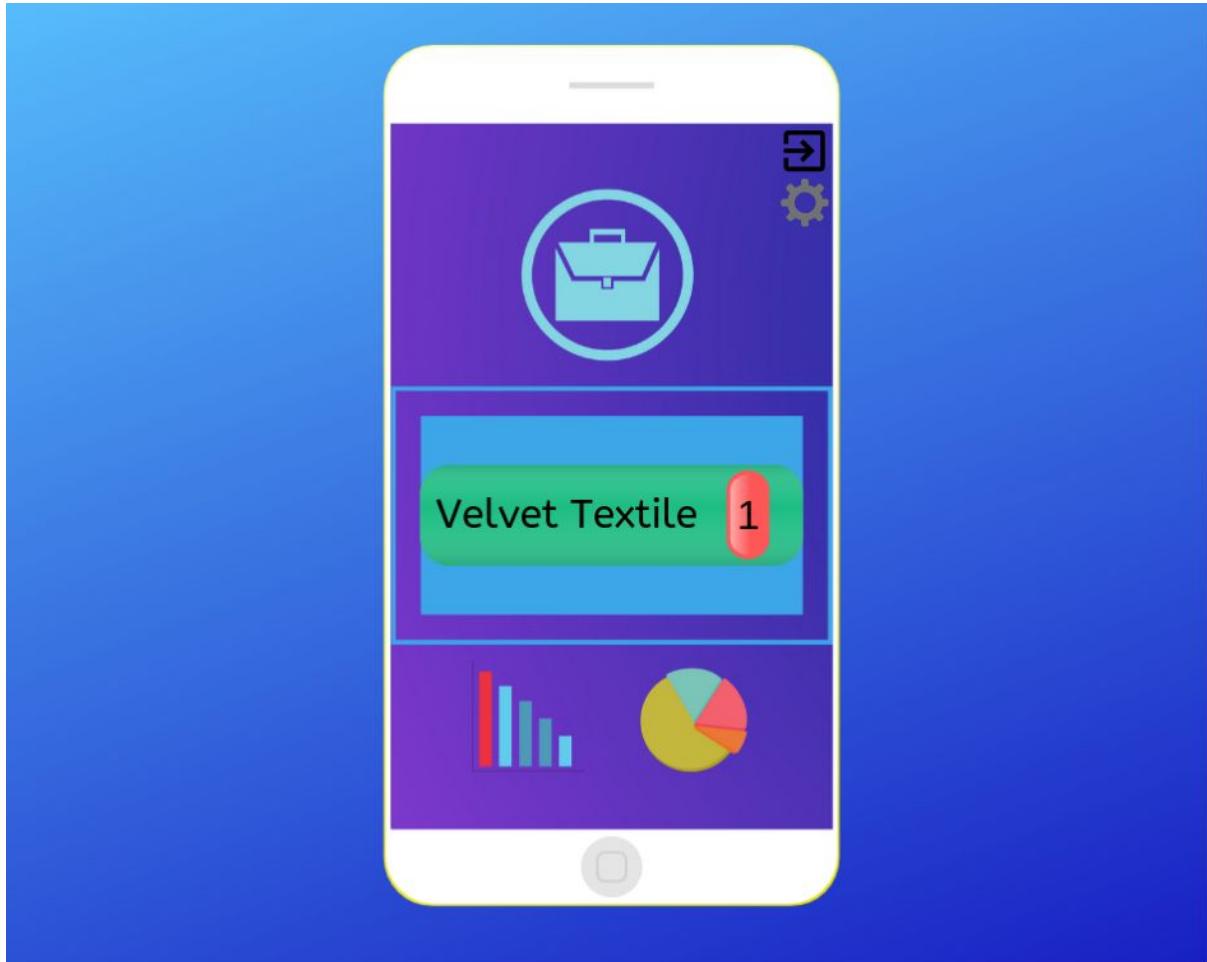
### 9.3. Scenario 3 (Illyda Baser)

#### Preconditions

- Illyda has already logged into the system and she is on her profile page.
- She opens her portfolio by clicking the portfolio button (the blue case button).

#### Flow

- Her stock can be seen as the only item of her portfolio.



## 9.4. Scenario 4 (Taylan Aksoy)

### Preconditions

- He has already signed in to the system and he sees his profile page now.

### Flow

- Taylan has already signed in to the system and the system displays his profile page. He clicks to the search button in order to make a sorted search.

The screenshot shows the Traders Platform user profile for Taylan Aksoy. At the top, there's a navigation bar with icons for My Profile, Trading Equipments, Events, Articles, Notifications, and Settings. Below the navigation is a search bar with placeholder text "Search...". The main content area features a profile picture of Taylan Aksoy, his name, and his title as a Management student at Bogazici University. It also displays his contact information: E-Mail (taylan.aksoy@gmail.com), Age (22), Interests (Economics and Sport), Location (Sariyer/Istanbul), and a list of contacts with Facebook and LinkedIn icons. Below this, social media links for Facebook and LinkedIn are shown. On the right side, there are buttons for "Following: 0", "Followers: 0", and a pencil icon labeled "Edit Profile". A note says "Prediction Success Rate: None". At the bottom, there are three buttons: "My Portfolio", "My Profit/Loss", and "My Alerts".

- He lists all users by sorting them according to their prediction success rate on USD/TRY and follows user with the highest prediction success rate by clicking the follow button.

The screenshot shows the Traders Platform search results page. At the top, there's a search bar, a dropdown for "Sort by: Descending Prediction Rate", a "Filters: None" dropdown, and a "Trading Equipment: USD/TRY" dropdown. Below the search bar are two user profiles: Tarik Yildirim (Entrepreneur, 42, 87% success rate, 82 following, 48k followers) and Ece Gürsoy (Assistant Professor, 32, 84% success rate, 13 following, 55k followers). Each profile has a "Follow+" button and a "See Profile" button. Below each profile, there are five circular icons, likely for pagination or filtering.

## 9.5. Scenario 5 (Taylan Aksoy)

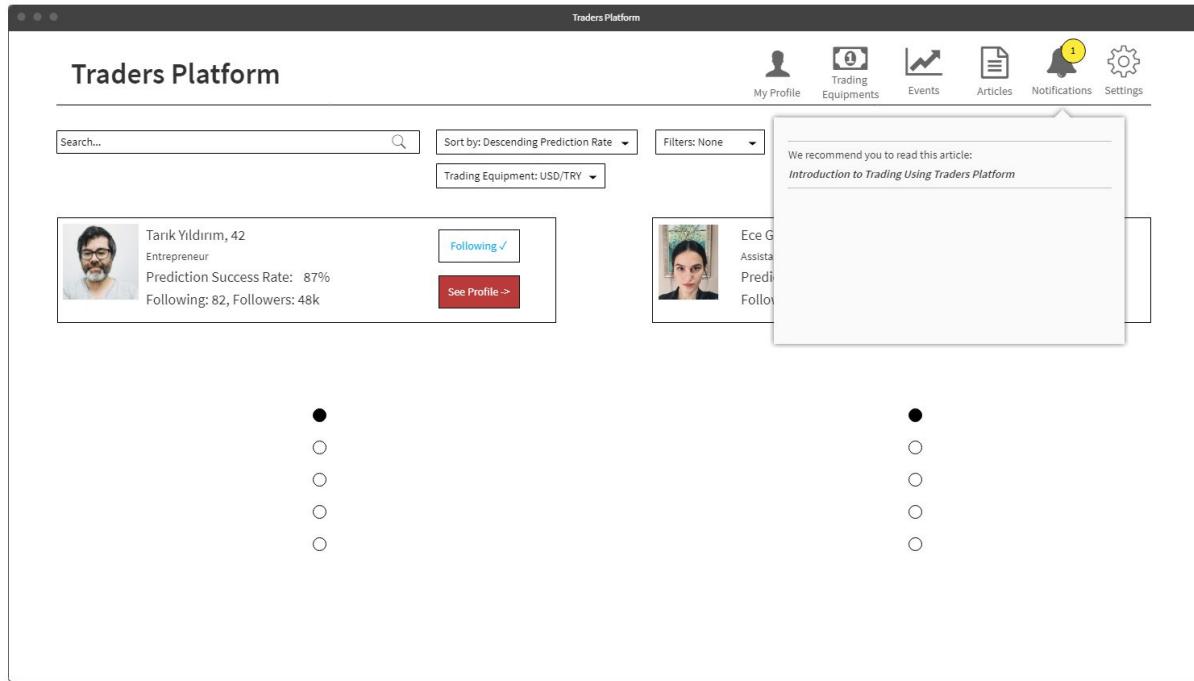
### Preconditions

- He has already signed in to the system and made a search in the system.
- He has followed the person with the highest prediction success rate on USD/TRY

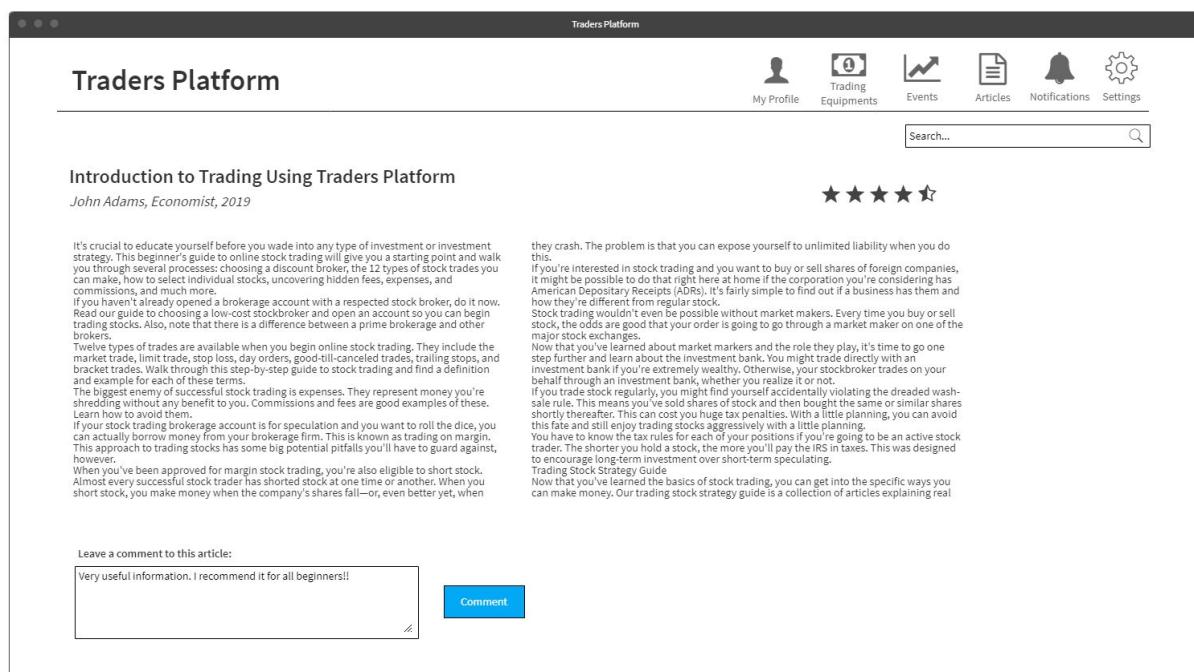
- He sees search page now.

## Flow

- When he is in the search page, the system sends him a notification of a recommended article. Then, he clicks the notification to read article "Introduction to Trading Using Traders Platform".



- After clicking the article name, he is directed to article page. He reads the article and leaves a comment.



## 9.6. Scenario 6 (Salih Okutan)

### Flow

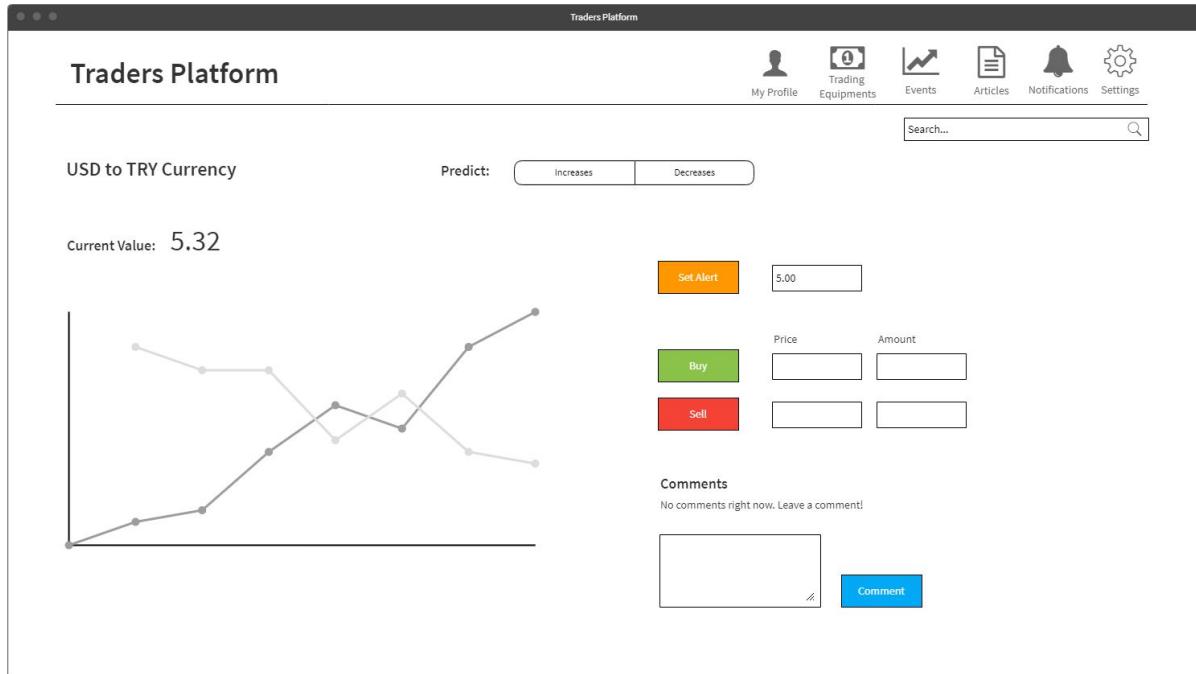
- Salih signs up to the Traders platform on Web because he is not used to using smartphones.

The screenshot shows the 'Traders Platform' sign-up interface. On the left, there's a 'Sign in' section with fields for E-Mail and Password, a 'Forgot your password?' link, and a 'Sign in' button. Below it is a 'Sign in with Google' button with a 'g+' icon. On the right, the 'Sign up' section is visible, featuring a radio button for 'Basic User' (unchecked) and one for 'Trader' (checked). The 'Trader' section requires Name (Salih), Surname (Okutan), Location (Izmir / Turkey), E-Mail (salih.okutan@hotmail.com), Password (\*\*\*\*\*), IBAN (TR27 xoooooooooooooxsx), and ID No. (123xxxxxx0000). A note below the location field specifies: \* Minimum of 8 characters and \* At least one lower case letter, upper case letter and number. There's also a 'Select your location from Google Maps' map showing a red pin at Bogazici University. A checkbox for 'I read the user agreement and accept it' is checked, and a 'Sign up' button is at the bottom.

- Browsing in the articles page, he finds an article about an upcoming political event which may affect the fluctuation of the USD/TL.

The screenshot shows the 'Traders Platform' articles page. At the top, there's a navigation bar with icons for My Profile, Trading Equipments, Events, Articles, Notifications, and Settings, and a search bar. Below the navigation is a news article titled 'Trump is Coming to Turkey!' by Clark Kent, Journalist, 2019. The article text is mostly placeholder text. To the right of the article is a five-star rating. At the bottom, there's a comment input field with the placeholder 'Leave comment to this article:' and a 'Comment' button.

- Then he sets an alert for a specific trader user's equipment, USD buying price, in order to get a notification when USD buying rate for TL decreases below 5. This alert will notify Salih whenever the Platform reads a buying rate below 5TL, for the first time.



- After a week, he makes a buy order of 2000 USD when 1 USD = 4.93 TL.



## 9.7. Scenario 7 (Ahmet Kazan)

### Flow

- Ahmet signs up to the platform on Web using his Google account.

The screenshot shows the 'Traders Platform' sign-up interface. On the left, there's a 'Sign in' form with fields for E-Mail and Password, and a 'Sign in' button. On the right, there's a 'Sign up' form. At the top of the 'Sign up' form, there are radio buttons for 'Basic User' and 'Trader', with 'Trader' selected. Below this is a 'Sign in with Google' button. The main 'Sign up' form has fields for Name (Ahmet), Surname (Kazan), E-Mail (ahmetkazarin@gmail.com), TC ID (82345678902), Location (Istanbul / Turkey), and IBAN (TR 77 0001 0022 4702 0935 1450 15). There's also a 'Select your location from Google Maps' section with a map showing a location in Istanbul. At the bottom of the 'Sign up' form is a checkbox for 'I read the user agreement and accept it' and a 'Sign up' button.

- After some (short) time getting used to the platform, now he knows very well about the features of the platform and he has some significant investments on this platform.

The screenshot shows the 'Traders Platform - My Investments' page. At the top, there's a navigation bar with icons for My Profile, My Investments, Equipments, Events, Articles, Notifications, and Settings. The main content area displays a portfolio named 'Gold-USD Basket'. It shows items: USD, XAU, and a Current Value of 15567.34 USD. There's a 'Shared' button with a checkmark and a 'See Details' button. To the right, there's a large empty space with a 'Create Portfolio' button. At the bottom of the page, it says '1 portfolio found.'

- Hearing that there will be an important Fed meeting soon, he sets up an alert for this event.

The screenshot shows the 'Events' section of the Traders Platform. It displays two upcoming events:

- Fed - the Federal Open Markets Committee**
  - Type: U.S. Federal Funds Rate
  - Time: 19 March 2019, 15:00 EST
  - Country: U.S.A
  - Significance: ★★
  - Outcome: N/A
  - Buttons: Chasing (blue), See Details (red)
- T.C.M.B - Monetary Policy Meeting**
  - Type: Policy Interest Rate
  - Time: 6 March 2018, 11:00 GMT+3
  - Country: Turkey
  - Significance: ★★
  - Outcome: 24%
  - Buttons: Chase (blue), See Details (red)

2 results found.

- Also, he creates an automatic buy order to buy some gold if the gold/USD ratio drop significantly.

The screenshot shows the 'Portfolio: Foreign Exchange Basket' section of the Traders Platform. It displays the following information:

- Assets**
  - XAU**
  - Amount: 12
  - Current Total Value: 15567.34 USD
  - Current Rate: 1292.6 USD (+19 = +8.7%)
  - Previous Close: 1283.4 TRY
  - Today's Range: 1275.5 - 1298.1 TRY
  - Moving Averages (5,10,20): 12862, 12863, 12851
  - Buttons: Buy (green), Sell (red)
- Orders**
  - Buy XAU If Under 1200**
  - Type: Buy XAU by selling USD
  - Amount: 5450 USD
  - Condition: XAU/USD < 1200
  - Limits: None
  - Buttons: Edit Amount (blue), Edit Conditions (blue)

Add Order

# 10. System Manual

## 10.1. Backend

First of all, cd into the backend app folder.

To be able to run the backend app, you have to install Python3 along with the Django REST framework and a couple of more supporting libraries.

To avoid version and compatibility problems, we suggest that you use conda. Once you create an environment and activate it, run the following commands:

```
conda install -c conda-forge djangorestframework
conda install -c conda-forge django-filter
conda install -c conda-forge markdown
conda install pillow
conda install psycopg2
conda install -c conda-forge django-cors-headers
conda install requests
conda install -c conda-forge django-rest-swagger
```

Then, create a PostgreSQL database named `mercatus`. The app fetches the DB credentials from specific environment variables. To automatically create these variables, run the following and supply the DB credentials when asked:

```
python setup_db_CREDS.py
```

Finally, setup and initialize the database by running:

```
python manage.py makemigrations nova
python manage.py migrate
python manage.py loaddata nova/fixtures/*.json
```

When you want to start the server, just run

```
python manage.py runserver
```

## 10.2. Frontend

To run the frontend app, you need to install the “serve” app, whose instructions can be found at <https://github.com/zeit/serve>. Then, cd into the `web_app` folder and run:

```
serve
```

This will run the frontend application. See serve's documentation to learn about how to configure the port and other options.

### 10.3. Mobile

Copy the apk file in the mobile app folder to your Android mobile phone and then double tap the file to install it.

## 11. User Manual for Web Application:

The web application has login and signup features. Basic user and trader user signups are different in accordance with the requirements.

On the main page of the web application, articles and most followed trading equipment are observed.

By clicking on “Trading Equipment” link on the navbar, a user is directed to the trading equipment page. From the menu on the top of the page, a user can choose the interval, type, main and compare parities of the trading equipment graph of the Trading Equipment page. In order to make an investment, the user must click on “buy/sell” button on the same menu. The button extends the menu and the user can now choose the trading equipment to be operated on and currency that is going to be used for the transaction. By clicking on “buy” or “sell” buttons, the user can make the transaction. The user can leave comments on trading equipment as well as liking the comments of others.

By clicking on the “Events” link, a user can observe the upcoming event with their names, dates, times, importances, and countries.

By clicking on the “Articles” link on the navbar, a user is directed to “Articles” page which essentially contains two tabs: “Article List” and “Write New Article”. All articles of all users are rendered under the “Article List” tab with their names, author, and a preview of the article. If the user clicks on the author button, she is directed to the profile page of the author of the article. If she clicks on “Read the article” button, the user is directed to the single article page on which she can read the whole article, comment on the article, and read the article. The user can observe all comments at the bottom of the page and she can rate the comments.

In order to make a search, the user has to fill the search box on the navbar and then click on the search button. The search results are rendered with icons indicating the type of the search result: article, user, trading equipment, comment, annotation, or event.

By clicking on the notifications button on the navbar, the user can go to the notifications page. All notifications of the user can be observed on this page with their reasons and times of creation along with the user that caused the notification. If the

user clicks on the cause user of the notification, she is directed to the profile page of that user.

By clicking on “Profile” button of the navbar dropdown, the user is directed to her profile page. On the profile page, the user can upload a profile picture by clicking on the camera icon under the profile picture, or delete the uploaded profile picture by clicking on the bin button next to the camera icon. The user can see her followers by clicking on “My Followers” button and the user that she follows by clicking on “My Followings” button. By clicking on “Articles” button the user can see her own articles. By clicking on “Update Info” button, the user is directed to the update credentials page, where the user can change her email, date of birth, and password. Under the main profile area that was explained above, there is the “Recommended For You” section. By clicking on the dropdown, the user can select recommended articles or portfolios and see the items, or hide the dropdown. Under your assets section, the user can see her assets. By clicking on the plus icon, the user can add US dollars to her account. At the bottom of the page the user can see her profits/loss.

By clicking on the “Articles” button on the navbar dropdown, the user can see her own articles.

By clicking on the “Portfolio” button on navbar dropdown, the user can go “Portfolio Page” which is essentially three tabs: “Your Portfolios”, “Create Portfolio”, and “Followed Portfolios”. Under “Your Portfolios” tab, all portfolios of the user are rendered. By clicking on a portfolio, the user can go to single portfolio page. On the single portfolio page, the user can open the add trading modal by clicking on “New Equipment” button. She can choose the equipment to be added in the modal and add it to the portfolio by clicking on “Add” button. The user can delete the portfolio by clicking on “Delete Portfolio” button. Equipment of the portfolio are rendered at the bottom of the single portfolio page. By clicking on the bin icon the user can remove the equipment from her portfolio and by clicking on the arrow button next to the bin button she can go to trading equipment page of that equipment. Under create portfolio tab, the user can create a portfolio by specifying its name and private status. By clicking on the “Create portfolio” button, the portfolio is created. Under followed portfolios tab, the user can see the portfolios that she is following. By clicking on the portfolio she can see the portfolio where the trading equipment are rendered. By clicking on an equipment she can go the trading equipment page of that equipment.

By clicking on the “Logout” button on the navbar dropdown, the user can logout.

On another user’s profile page, one can follow the user by clicking on the heart icon and unfollow the user by clicking on the heart icon again. The icon turns red if the user is followed.

## 12. API Documentation

Below is a list of endpoints. The detailed explanations of the endpoints take more than 100 pages when printed, so they are omitted here but can be accessed at  
<https://documenter.getpostman.com/view/2365580/SWEB1amT?version=latest>

### activations

**GET**

/activations/{uidb64}/{token}

### articles

**DELETE**

/articles/{article\_pk}/annotations/{annotation\_pk}

**DELETE**

/articles/{id}

**DELETE**

/articles/{id}/dislikes

**DELETE**

/articles/{id}/likes

**GET**

/articles

**GET**

/articles/recommendations

**GET**

/articles/{article\_pk}/annotations

**GET**

/articles/{article\_pk}/annotations/{annotation\_pk}

**GET**

/articles/{id}

**GET**

/articles/{id}/comments

**GET**

/articles/{id}/dislikes

**GET**

/articles/{id}/likes

**POST**

/articles

**POST**

/articles/{article\_pk}/annotations

**POST**

/articles/{id}/comments

**POST**

/articles/{id}/dislikes

**POST**

/articles/{id}/likes

**PUT**

/articles/{article\_pk}/annotations/{annotation\_pk}

**PUT**

/articles/{id}

## auth\_tokens

**POST**

/auth\_tokens

## comments

**DELETE**

/comments/{id}

**DELETE**

/comments/{id}/dislikes

**DELETE**

/comments/{id}/likes

**GET**

/comments/{id}/dislikes

**GET**

/comments/{id}/likes

**POST**

/comments/{id}/dislikes

**POST**

/comments/{id}/likes

**PUT**

/comments/{id}

## cron\_jobs

**POST**

/cron\_jobs/av\_daily

**POST**

/cron\_jobs/av\_intradaily

**POST**

/cron\_jobs/nasdaq\_daily

**POST**

/cron\_jobs/nasdaq\_intradaily

## events

**GET**

/events/{date}

**POST**

/events/{date}

## portfolios

**GET**

/portfolios/recommendations

## profit\_loss

**GET**

/profit\_loss

## semantic\_search

**GET**

/semantic\_search

## trading\_equipment\_searches

**POST**

/trading\_equipment\_searches

## trading\_equipments

**DELETE**

/trading\_equipments/{id}/predictions/downvotes

**DELETE**

/trading\_equipments/{id}/predictions/upvotes

**GET**

/trading\_equipments

**GET**

/trading\_equipments/digital

**GET**

/trading\_equipments/forex

**GET**

/trading\_equipments/{id}

**GET**

/trading\_equipments/{id}/comments

**GET**

/trading\_equipments/{id}/predictions/downvotes

**GET**

/trading\_equipments/{id}/predictions/upvotes

**GET**

/trading\_equipments/{tr\_eq\_sym}/prices/current

**GET**

/trading\_equipments/{tr\_eq\_sym}/prices/daily

**GET**

/trading\_equipments/{tr\_eq\_sym}/prices/intradaily

**POST**

/trading\_equipments/{id}/comments

**POST**

/trading\_equipments/{id}/predictions/downvotes

**POST**

/trading\_equipments/{id}/predictions/upvotes

## **user\_searches**

**POST**

/user\_searches

## **users**

**DELETE**

/users/{id}

**DELETE**

/users/{user\_pk}/following\_portfolios/{portfolio\_pk}

**DELETE**

/users/{user\_pk}/followings/{following\_pk}

**DELETE**

/users/{user\_pk}/portfolios/{portfolio\_pk}

**GET**

/users

**GET**

/users/{id}

**GET**

/users/{id}/articles

**GET**

/users/{id}/followers

**GET**

/users/{id}/followings

**GET**

/users/{user\_pk}/assets

**GET**

/users/{user\_pk}/following\_portfolios

**GET**

/users/{user\_pk}/notifications

**GET**

/users/{user\_pk}/notifications/count

**GET**

/users/{user\_pk}/portfolios

**GET**

/users/{user\_pk}/portfolios/{portfolio\_pk}

**POST**

/users

**POST**

/users/{id}/followings

**POST**

/users/{user\_pk}/assets

**POST**

/users/{user\_pk}/cash

**POST**

/users/{user\_pk}/following\_portfolios

**POST**

/users/{user\_pk}/portfolios

**PUT**

/users/{id}

**PUT**

/users/{user\_pk}/portfolios/{portfolio\_pk}

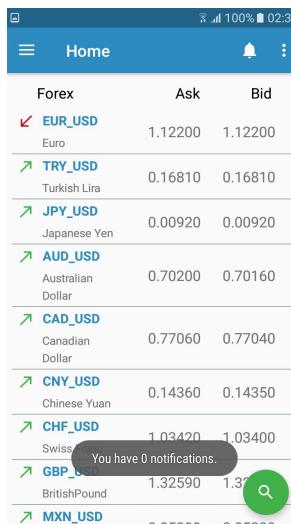
## 13. Project Plan

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		Meeting with new team	1 day	9/26/19 8:00 AM	9/26/19 5:00 PM		All
2		Revising project	17 days?	9/26/19 8:00 AM	10/18/19 5:00 PM		
3		Onboarding new members	0.167 days?	9/26/19 8:00 AM	9/26/19 9:20 AM		Ali Özden;Alper Çakan;... Ceren Tahtasız Volkan Yılmaz;İsmet D...
4		Updating communication plan and README	2 days?	10/17/19 8:00 AM	10/18/19 5:00 PM		
5		Revising requirements and mockups	3.5 days?	9/28/19 8:00 AM	10/3/19 1:00 PM		
6		Structuring the project and specifications	5 days?	10/4/19 8:00 AM	10/10/19 5:00 PM		
7		Deciding coding standards and git workflow	4 days?	10/5/19 8:00 AM	10/10/19 5:00 PM		All
8		Deciding programming languages	2 days?	10/9/19 8:00 AM	10/10/19 5:00 PM		All
9		Designing the database model	2.333 days?	10/4/19 8:00 AM	10/8/19 10:40 AM		Alper Çakan;Sercan Ers...
10		Design the initial API	1 day?	10/5/19 8:00 AM	10/7/19 5:00 PM		Alper Çakan;Sercan Ers...
11		Choosing application name	1 day?	10/9/19 8:00 AM	10/9/19 5:00 PM		All
12		Deciding visual standards	1.6 days?	10/8/19 8:00 AM	10/9/19 1:48 PM		Ali Özden;Barış Zöngür...
13	🏃	Front End	62 days?	9/30/19 8:00 AM	12/24/19 5:00 PM		Ali Özden;Barış Zöngür...
14		General React Study	5 days?	9/30/19 8:00 AM	10/4/19 5:00 PM		
15		Initializing Front End	6 days?	10/7/19 8:00 AM	10/14/19 5:00 PM		
16		Creating main application layout for different	6 days?	10/7/19 8:00 AM	10/14/19 5:00 PM		
17		Develop Login and SignUp Pages	7 days?	10/14/19 8:00 AM	10/22/19 5:00 PM		
18		Develop Profile Page	3 days?	10/17/19 8:00 AM	10/21/19 5:00 PM		
19		Establish communication with the API	2 days?	10/21/19 8:00 AM	10/22/19 5:00 PM		
20		Preparing presentation scenarios for MS1	2 days?	10/21/19 8:00 AM	10/22/19 5:00 PM		
21		Develop Following Functionality	6 days?	10/21/19 8:00 AM	10/28/19 5:00 PM		
22		Restructuring the code using routing library	5 days?	11/4/19 8:00 AM	11/8/19 5:00 PM		
23		Implement Article Functionalities	5 days?	11/11/19 8:00 AM	11/15/19 5:00 PM		
24		Implement Trading Equipment Functionalities	5 days?	11/18/19 8:00 AM	11/22/19 5:00 PM		
25		Implement Event Functionalities	3 days?	11/20/19 8:00 AM	11/22/19 5:00 PM		
26		Implement Commenting Functionality	5 days?	11/18/19 8:00 AM	11/22/19 5:00 PM		
27		Implement Like/Dislike Functionality	5 days?	11/19/19 8:00 AM	11/25/19 5:00 PM		
28		Implement email verification	4 days?	11/18/19 8:00 AM	11/21/19 5:00 PM		
29		Develop rate functionality for trading equipment	6 days?	11/18/19 8:00 AM	11/25/19 5:00 PM		
30		CSS improvements	4 days?	11/20/19 8:00 AM	11/25/19 5:00 PM		
31		Implement recommendation	5 days?	12/4/19 8:00 AM	12/10/19 5:00 PM		
32		Implement portfolio and assets	6 days?	12/9/19 8:00 AM	12/16/19 5:00 PM		
33		Implement notifications	5 days?	12/16/19 8:00 AM	12/20/19 5:00 PM		
34		Implement trade feature	15 days?	12/2/19 8:00 AM	12/20/19 5:00 PM		
35		Implement annotations	11 days?	12/9/19 8:00 AM	12/23/19 5:00 PM		
36		Implement semantic search	2 days?	12/20/19 8:00 AM	12/23/19 5:00 PM		
37		Preparing presentation scenarios for MS2	2 days?	12/23/19 8:00 AM	12/24/19 5:00 PM		
38	🏃	Back End	58 days?	9/30/19 8:00 AM	12/18/19 5:00 PM		Alper Çakan;Furkan A...
39		Initializing the app	6 days?	9/30/19 8:00 AM	10/7/19 5:00 PM		
40		User model and serializer	5 days?	10/7/19 8:00 AM	10/11/19 5:00 PM		
41		User CRUD endpoints	4 days?	10/9/19 8:00 AM	10/14/19 5:00 PM		
42		Authentication	6 days?	10/14/19 8:00 AM	10/21/19 5:00 PM		
43		Permissions classes	4 days?	10/16/19 8:00 AM	10/21/19 5:00 PM		
44		User follow-up follow features 27.10dan sonr	5 days?	10/21/19 8:00 AM	10/25/19 5:00 PM		
45		Implement Article endpoints	7 days?	10/28/19 8:00 AM	11/5/19 5:00 PM		
46		Implement Comment endpoints	7 days?	11/1/19 8:00 AM	11/11/19 5:00 PM		
47		Implement Forex and Digital Functionality	9 days?	11/5/19 8:00 AM	11/15/19 5:00 PM		
48		Implement Event endpoints	4 days?	11/15/19 8:00 AM	11/20/19 5:00 PM		
49		Implement recommendation	5 days?	11/29/19 8:00 AM	12/5/19 5:00 PM		
50		Implement portfolio and assets	6 days?	12/5/19 8:00 AM	12/12/19 5:00 PM		

51		Implement notifications	5 days?	12/12/19 8:00 AM	12/18/19 5:00 PM		
52		Implement annotations	10 days?	12/2/19 8:00 AM	12/13/19 5:00 PM		
53		Implement semantic search	2 days?	12/17/19 8:00 AM	12/18/19 5:00 PM		
54		Mobile	62 days?	9/30/19 8:00 AM	12/24/19 5:00 PM		Ceren Tahtasız;Volka...
55		General Kotlin and Android Study	5 days?	9/30/19 8:00 AM	10/4/19 5:00 PM		
56		Initialize Mobile	4 days?	10/2/19 8:00 AM	10/7/19 5:00 PM		
57		Develop SignIn and SignUp Layouts	5 days?	10/7/19 8:00 AM	10/11/19 5:00 PM		
58		Implement user page and main activities	6 days?	10/7/19 8:00 AM	10/14/19 5:00 PM		
59		Establish communication with the API	4 days?	10/14/19 8:00 AM	10/17/19 5:00 PM		
60		Sign in and sign up functionality	5 days?	10/15/19 8:00 AM	10/21/19 5:00 PM		
61		Preparing presentation scenarios for MS1	2 days?	10/21/19 8:00 AM	10/22/19 5:00 PM		
62		Implement show/edit profile	6 days?	10/21/19 8:00 AM	10/28/19 5:00 PM		
63		Determine and Remove deprecated functions	5 days?	10/28/19 8:00 AM	11/1/19 5:00 PM		
64		Implement Article Functionalities	10 days?	11/4/19 8:00 AM	11/15/19 5:00 PM		
65		Implement Commenting Functionality	13 days?	11/6/19 8:00 AM	11/22/19 5:00 PM		
66		Implement Forex and Digital Functionality	10 days?	11/11/19 8:00 AM	11/22/19 5:00 PM		
67		Implement Like/Dislike Functionality	5 days?	11/11/19 8:00 AM	11/15/19 5:00 PM		
68		Implement Home Feed Screen	5 days?	11/18/19 8:00 AM	11/22/19 5:00 PM		
69		Implement Event Functionalities	4 days?	11/20/19 8:00 AM	11/25/19 5:00 PM		
70		Mobile visual improvements	2 days?	11/22/19 8:00 AM	11/25/19 5:00 PM		
71		Implement recommendation	5 days?	12/4/19 8:00 AM	12/10/19 5:00 PM		
72		Implement portfolio and assets	6 days?	12/9/19 8:00 AM	12/16/19 5:00 PM		
73		Implement notifications	5 days?	12/16/19 8:00 AM	12/20/19 5:00 PM		
74		Implement trade feature	15 days?	12/2/19 8:00 AM	12/20/19 5:00 PM		
75		Implement annotations	11 days?	12/9/19 8:00 AM	12/23/19 5:00 PM		
76		Implement semantic search	2 days?	12/20/19 8:00 AM	12/23/19 5:00 PM		
77		Preparing presentation scenarios for MS2	2 days?	12/23/19 8:00 AM	12/24/19 5:00 PM		

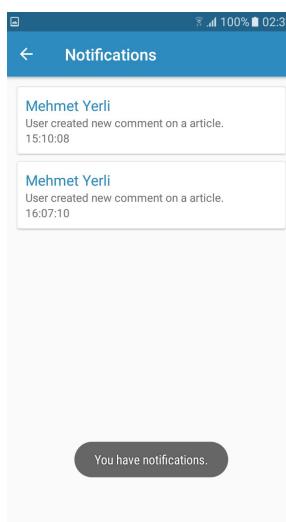
## 14. User Manual for Mobile

If someone wants to sign in and sign up, she just needs to give credentials to do it. After register is completed, she does not need to enter these credentials to sign in again.



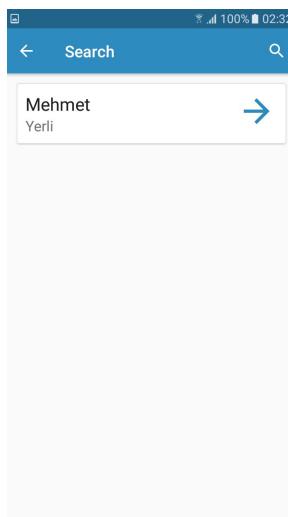
1) When user clicks on Mercatus Application icon in her mobile phone, home screen is loading. In the first opening she can see whether there are any new notifications or not.

2) Here Mercatus shows some forex items and some suggested articles based on her followers and followings.

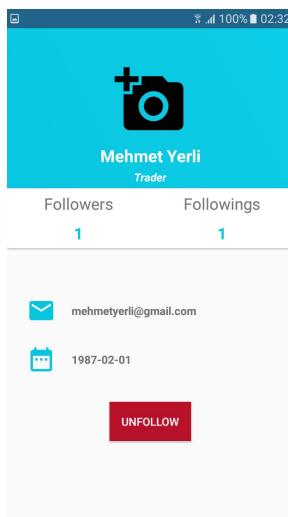


3) After she clicks on notification button on the top right corner with the ring image, she can now see past notifications as well.

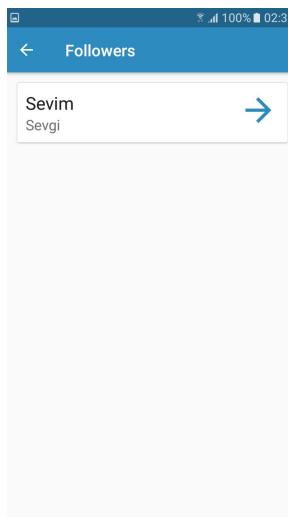
4) Here she can navigate through the user profile to whom makes a comment, annotation or article anywhere by clicking the blue name text on the screen.



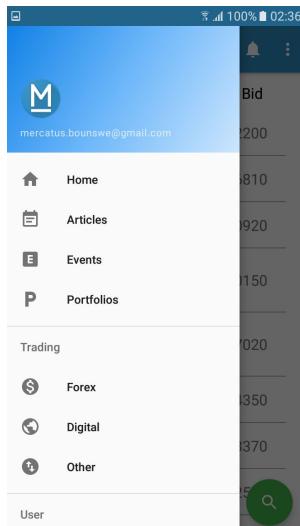
- 5) After clicking the search button on the home page in the first image, and writing down “mehmet”; she can now view the possible users who has name or surname that contains “mehmet” word.
- 6) Here she can go to his profile page by just clicking the name.



- 7) Here is the a user profile from any other user.
- 8) A user can see email address, birthday date, followers and followings list of the user.
- 9) She can click to see followers and following list of the user by just clicking on the corresponding text on the screen.
- 10) She can follow or unfollow by clicking the button in the middle.



- 11) Here is the followers list of Mehmet Yerli. Since Sevim Sevgi is the current user, it is natural to see herself here.



12) Here is the navigation bar to view all the submenus inside the mercatus. User can navigate through anywhere from here.



13) Here is the Forex page. Digital and other trading equipments page are also similar to this one. User can see several equipments from here and she can search a particular equipment by clicking the green search icon below.

14) From here she can click to see a particular forex item, arrows on the left states this forex item value is currently increasing or decreasing.

15) To log out she just needs to click 3 dots on top right corner and then click log out.



16) When forex item clicked this page opens. Here is the last 30 days of change euro against usd dollar. A particular point show the exact value on the chart.

17) She can give a vote to predict whether it will increase or decrease. She can like/dislike a comment, write one, change or delete her comments also from here.

18) To write a comment she can click on the bottom right corner action button.

19) If she wants to see more detailed view, she just needs to click zoom icon with blue background.



20) Here is the enlarged, fullscreen chart of the forex item.

**Istanbul event marks anniversary of D-8 economic group**

The D-8 Organization for Economic Cooperation held a reception in Istanbul to celebrate its 22nd anniversary of foundation. Addressing the opening speech of the ceremony, Shaari said the D-8 has become an ever-growing unity since its establishment.

A payment system peculiar to the D-8, the international D-8 university, the D-8 aviation network, as well as health and social protection programs prove the success of the organization, he added.

1 0 Ahmet Kazancı

**Trump Wants Negative Rates. Here's How That Would Work**

Mr. Trump has been attuned to negative rate policies abroad as the European Central Bank looks poised to slash a key rate further below zero. Mere anticipation of that move has caused the dollar to strengthen the euro, making American exports less competitive.

21) When user clicks articles button through navigation bar, articles screen opens.

22) From here the user can write an article by clicking green action button. or like/dislike a particular article, navigate to author's profile page, make a comment or enlarge the article by clicking on it.

23) If she want to navigate through the article page, she just needs to click on title with blue background.

**Istanbul event marks anniversary of D-8 economic group**

Ahmet Kazancı

The D-8 Organization for Economic Cooperation held a reception in Istanbul to celebrate its 22nd anniversary of foundation. Addressing the opening speech of the ceremony, Shaari said the D-8 has become an ever-growing unity since its establishment.

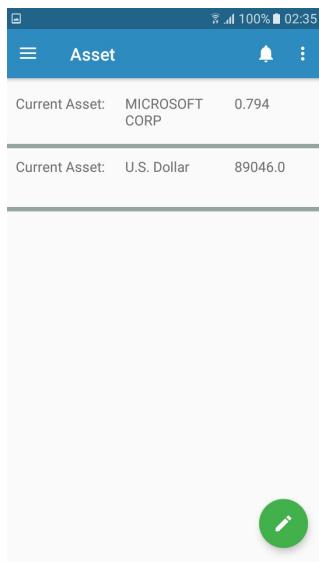
Comments: 1 0

**Sevim Sevgi**

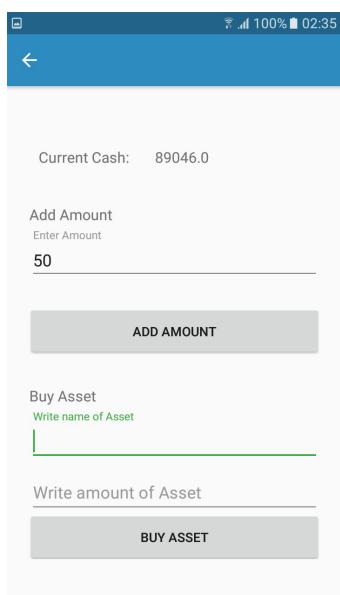
It is important in emea stocks.

Like 0 Dislike 1

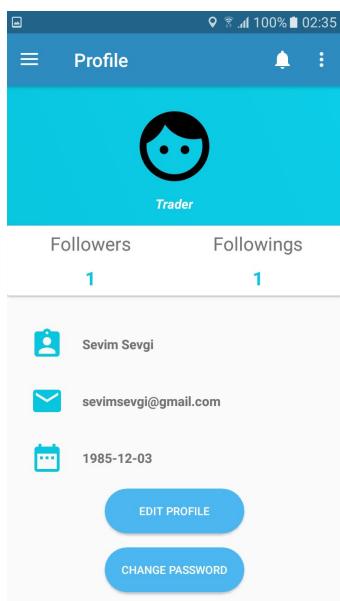
24) Here is the particular article page. She can make comments, delete or edit her comments, like and dislike comments or the article.



25) Here is the asset page of user assets. Here she can create new asset list by clicking bottom right action button.



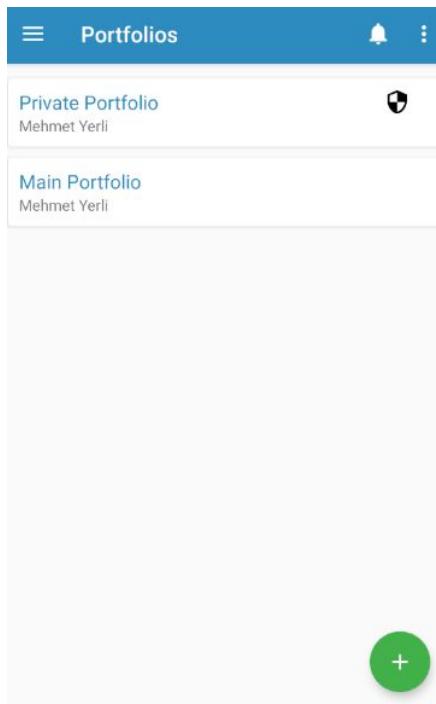
26) Here she can load money to her account, and buy asset in change of the loaded asset.



27) When she clicks on her profile page, she can edit profile or change password.



28) Here is a events page, they are listed with first the most importance ordered, if its importance is equal, then shows the closest one

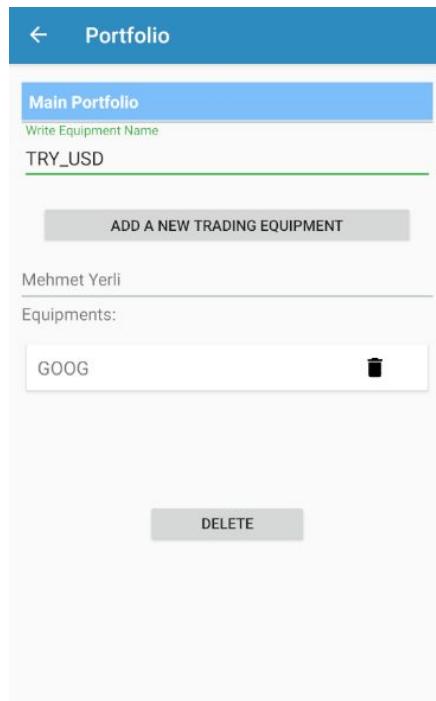


29) Here is a portfolio page, current portfolios of a user is listed in this page.

30) Security sign in the near of the portfolio name shows that the portfolio is private.

31) When the user clicks the name of the portfolio, the can will be directed to the that specific portfolio page.

32)When the user clicks the PLUS sign at the bottom of the screen, the user can create a portfolio

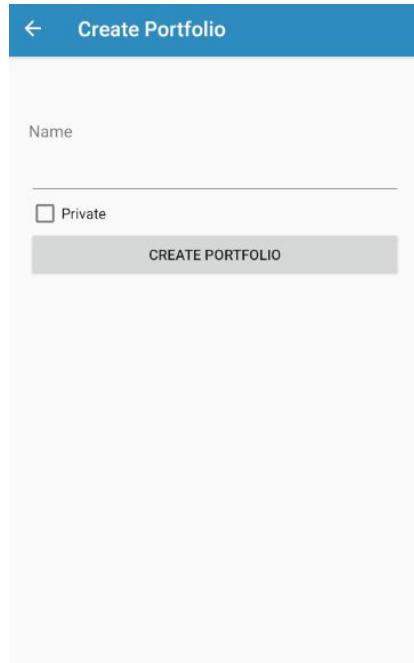


33) Here is a showing portfolio page, firstly writes the name of portfolio and then the owner of the portfolio and then the equipments of that specific portfolio

34) When the user writes the equipment name, the user can add the equipment to his/her portfolio

35) When the user clicks the trash button near of the equipment name, the equipment will be deleted

36) When he clicks the delete button, the portfolio will be deleted.



37) Here is a creating portfolio page

38) The user must give the name of the portfolio and choose the privacy, then the user can click the create portfolio button.