Milestone Report 1

1. Executive Summary

Traders Platform Project has started on February 2019 and has finished its initial phases successfully. It has an up-to-date repository on Github where you can follow the improvements so far. Requirements analysis draft had prepared by the team and it has modified and verified during customer meetings. There are personas for possible user types and they have their user stories, scenarios and mockups which show functionalities of the system. In terms of design, our team designed use case, class and sequence diagrams which are consistent with each other and updated after customer reviews. Traders Platform needs no major change for moving forward because during the project development, communication between the customer and the team was healthy and effective. When a change is requested by the customer, it is considered by the team timely so that each step of the project is verified by the customer.

2. Deliverables

REPOSITORY	Created	Revised After Feedback	Done
--Project Documents			
----Group Meetings	13.02.2019	--	✓
----Customer Meetings	22.02.2019	--	✓
----PS's	26.02.2019	--	✓
----Github Research Assignment	29.02.2019	--	✓
----Member Pages	26.02.2019	--	✓
---- Communication Plan	24.02.2019	✓	✓
----Readme	13.02.2019	--	✓
--Requirements	24.02.2019	✓	✓
--Personas	27.02.2019	✓	✓
--User Stories and Mockups	27.02.2019	✓	✓
--Diagrams			
----Use Case Diagram	11.03.2019	✓	✓
----Class Diagram	14.03.2019	✓	✓
----Sequence Diagrams	17.03.2019	✓	✓
--Issue Management	--	--	--

3. Evaluation of Deliverables

3.1. Project Documents

3.1.1. Github Research Assignment

We started our project by exploring Github. In order to be able to use it for our own project, we observed other projects' repositories and we searched for interesting GitHub features. We documented this research on "GitHub and Git Research" document on our Wiki page. This document really helped us to get accustomed with GitHub and I believe it's useful for other readers too because it includes examination of other repo's and this provides user the criterias on how to evaluate a repo.

3.1.2. Readme

Readme file is one of the first pages we created in our repo using Markdown. At first, this file only consisted of names of group members. After feedback, we embedded our profile links to our names, we elaborated "About us" part, we added a group picture and indicated ourselves on the picture. It is useful so that now our customer can easily call group members by their names in real life.

3.1.3. Communication Plan

In our communication plan we tried to plan and detail possible communication mediums and how to use them. In the first draft, we only included communication mediums will be used and weekly meetup plan. After feedback, we included team members' and customers' names, communication mediums, some certain roles of team members and most importantly a tabular detailed plan of communication situations. It provides necessary information to readers on how to properly communicate with team members on certain cases.

3.1.3. Member Pages

Each team member have a Wiki page in which they share about themselves. Most of them consists of a brief professional description, skillset and contact information. Some are more detailed some are less but it's normal since what to share on web is up to each member. It provides the repo viewer an easy-to-reach brief personal information which is useful.

3.1.4. Group Meetings

We documented our weekly group meetings in which we generally discuss new assignment, feedbacks and where we are in certain tasks. At first meeting, we used a format we later found out that was wrong: agenda, attendees, where, when, results of the meeting. Then we passed to the right format: details, attendees, agenda, discussions, actions. We modified wrongly

formatted previous meetings notes to right format. We enumerated agenda, discussion and action elements. We started to write what's discussed in more detail and what is the result of that discussion. We added a tabular which shows details of each action (deadline, assignee etc.) and we linked actions with issues. These documents are really helpful to see the progress of project and to have more effective meetings. Linking actions with issues is a great way to see if the project is on track.

3.1.5. Customer Meetings and PS's

Even though the PS sessions and customer meetings are combined in real life we decided to have separate documents for each because course assistant's identity differs between. In customer meetings notes, we include project related questions and answers. To see all Q&A at once, we created a separate document in which we combined all customer meetings. In PS notes, we include our assistants' explanations and suggestions on assignments. When not all team members are not present in a PS or customer meeting, these notes are really crucial and helpful to inform everyone.

3.2. Requirements

In our Requirements document, first we decided our functional and non-functional requirements of our project based on our project document. After costumer meeting, we finalized our Requirement Analysis based on Project plan and costumer expectations.

3.3. Personas

There are 4 personas which represents possible users of Traders Platform. They differ from each other: their backgrounds, their expectations and functionalities they use are different. Each persona has a background, goal, user story and acceptance criteria of their story. Personal information they have are long and detailed enough to create a clear personality of a living person in viewer's mind. User stories and goals of personas are aligned. Acceptance criterias are linked to requirements in requirements document so that one can follow how many functionalities in the system are represented by personas. Personas really helped us to envision what users are looking for in the system.

3.4. User Stories and Mockups

We created user stories and mockups for our personas. Main purpose in our mockups was showing the functionalities of Traders Platform. Also, we cared the design of mockups. Every persona's mockup has distinct functionalities according to their story. One can easily see how the user reaches her/his goal that is in the story by looking the mockups.

We had some problems in our mockups such that so many things were done in each mockup. After the feedback that we received about this problem, we tried to split our mockups and show exactly one function in one mockup.

3.5. Design Documents

3.5.1. Use Case Diagram-

The diagram includes all use cases of Traders Platform for all user types in accordance with Requirements document. It was created on Lucidchart and a view link can be found at our Use Case Diagram page in Wiki.

There were a few difficulties due to lack of knowledge about the project which were of course dealt with by questions at the customer meeting and the feedback we received. While revising the diagram after the feedback, a few minor but necessary changes were made on Requirements document.

3.5.2. Class Diagram

Class diagram includes the classes that will appear in the coding part of the project. These classes are also used as the objects participating in the sequence diagrams. This diagram makes us see project in a wider perspective. We can easily see what kind of entities constitute the project, how many types of users there are, which properties of which classes are private etc.

Difficulties faced while making the class diagram were mainly about the lack of knowledge about making a proper class diagram and idea differences among the team members when trying to think how to design it all together. Although we conducted some research about how to design it, we had difficulties finding comprehensive resources to study. Thus, the first version of the diagram we made was not quite useful. We also experienced this when we were making the sequence diagrams.

After the feedback, we updated our class diagram and sequence diagram followingly. This improvement is conducted by a smaller number of team members in order to decide how to design faster. It was like making the class diagram whole again but we saw the magnificent difference when it comes to updating the sequence diagrams. We could easily see that a proper class diagram results in proper sequence diagrams.

3.5.3. Sequence Diagrams

Sequence diagrams show the interaction of the classes and the methods they call (or the messages they pass, in other words) during the “execution” of a particular use case. The most

important aspect that we paid attention during making the sequence diagrams was to create sequence diagrams for all of the key use cases.

We created about two dozens of sequence diagrams, so it is safe to say that we really covered most of the use cases. One of the problems we had about the sequence diagrams was coordination and consistency of the drawings, since we had not agreed upon a single design tool at the beginning. However, after a couple of the sequence diagrams, we noticed this problem immediately and solved it before it went too far. So, now, all of our diagrams have the same style.

Another problem, which we fixed after a feedback was provided, was that there were some differences between the messages/methods and classes we used in the sequence diagram and in the class diagram. This was mainly due to the fact that it is not easy to plan about the all methods and class we will need without actually thinking in detail about the implementations of the use cases. However, after completing the sequence diagrams, we then had thought about all the details; and once this problem was pointed out in the feedback, we went back and updated our class diagram accordingly.

Overall, we think that the final status of the sequence diagrams is great; they cover the most important use cases, they are detailed and consistent with the class diagram. As a result, we think that this part won't be changed much during the further weeks of the projects, hence it will only make our project plan easier.

3.6. Issue Management

We are using issues so that we can keep track of our tasks. We do so by making use of tags and deadlines. We create issues mostly on meetings but we also create issues to let group members check one's work, or, sometimes when tasks lead to other tasks, new issues may be created any time of the week.

We used to assign too many assignees to issues but after receiving a feedback about this matter, we are now assigning only who is actually responsible for the particular task.

In the end, issues are a really good way of keeping track our progress which we think we are using efficiently.

4. Work Done by Team Members

Ceren Tahtasız	● Took notes on weekly meetings ● Prepared questions and asked them during customer meetings ● Took part of all aspects of the project but especially: - LeetCode CLI (repo) and File Finder (feature) of GitHub research - System Requirements of Requirements assignment - Taylan's persona-user story-mockup of Scenario assignment - Use Case Diagram of Design assignment - Follow User Portfolio and Follow Another User on Sequence diagrams
Mehmet Can Ünal	● Created my personal WikiPage. ● Took part in arrangement of System Requirements. ● Took part in creation persona and mock-up. ● Took part in the making of the class diagram. ● Attended most of weekly meetings.
Alper Çakan	● Did research about Git & GitHub features ("commenting on a PR", "suggested changes", "cherry-pick") and found a couple of exemplary repos (TypeScript, Filament) and summarized these on the related Wiki page on our repo ● Did research about various issue labeling practices to find labels that we can use on our repo, added about 10 labels according to this ● Found some issue and PR templates that we can use on our repo and shared them with the group ● Created my personal Wiki page ● Wrote the "user requirements" part of the project requirements document ● Prepared questions to be asked to the customer to clarify the project requirements ● Fixed the formatting of some issues, wiki pages, meeting notes etc. ● Added some terms to the glossary part of the project requirements document ● Created the persona "Ahmet Kazan" (experienced, trader user) along with his story, background, acceptance criteria and drew all of his mockups ● Made some fixes to the requirements document according to the feedback given to our group ● Took part in the making of the class diagram ● Did research about MVC architecture to improve our class diagram ● Drew a couple of sequence diagrams (profit/loss calculation manual and by investment history use cases) ● According to the feedback given by the customer, fixed some problems about the personas, user stories and the mockups ● Attended some of the classes and the customer meetings ● Evaluated the sequence diagrams ● General issue maintenance, cleanup, tracking

Yusuf Yüksel	• Researched about good repo examples and github features(Spark as a repo,Project as a github feature) • Prepared glossary part of Requirements assignment with Furkan • Took part of Ilayda's persona-user-story-mockup Scenario assignment • Took part of Use Case Diagram of Design assignment • Prepared commenting on article's and trading equipment's sequence diagrams
Efe Önal	• Done research on exemplary GitHub repos. • Took part in preparation of User Requirements part of Requirements document. • Made necessary changes on User Requirements after feedbacks. • Took part in preparation of Ilayda Baser's persona, user story, and mockup. • Made necessary changes to Ilayda's persona, user story and mockup. • Took part in preparation of Use Case Diagram. • Made necessary changes on Use Case Diagram after the feedback. • Prepared Login and Following an Economic Event sequence diagrams and updated it after the feedback and updated Class Diagram. • Made minor changes on various diagrams and pages.
Sercan Ersoy	• Researched about useful GitHub repo examples • Prepared non-functional requirements of the project • Improved non-functional requirements according to the feedback • Took part in the creation of the Taylan Aksoy's persona and mockup scenarios of him. • Improved mockups of Taylan Aksoy according to the feedback • Created and edited some of the issues • Organized the directory system of the repository • Updated image links when they are changed because of the repo organization • Revised the writing organization of some of the wiki pages and the readme file • Took part in the creation of the first version of the class diagram • Made sequence diagrams for "Rating an article" and "Writing an article" • Improved class diagram according to the feedback • Improved sequence diagrams mentioned above according to the improvements in the class diagram • Attended to some of the customer meetings (or PS's) • Attended to some of the classes • Communicated with instructors when needed

Barış Zöngür	• Classic stuff like issues, sharing links etc. • Research of some GitHub repo examples • Updated my github and wiki page • Attended in most of the meetings • A part of user requirements of our project • Prepared android mockup based on story • Took part in Use Case diagram • Made certain Sequence Diagrams • In general updated and revised necessary parts based on revisions
Ahmet Yiğit Gedik	• Made researches about how to use github efficiently • Prepared agendas for our weekly meetings • Created my personal Wiki page • Created issues to communicate with my team members • Took part in creating System Requirements part of Requirements document • Took part in creating the persona Taylan Aksoy • Took part in creating mockup for the persona Taylan Aksoy • Improved mockups and scenarios according to the feedback • Took part in creating Class Diagram • Created Filter Economic Events by Country and Filter Economic Events by Significance Level sequence diagrams • Improved sequence diagrams which I created according to the feedback • Attended some of the classes and PS's • Attended every weekly meeting
Furkan Aydar	• Created my personal Wiki page. • Done a research about example repositories of Github (Create React App) • Prepared the Glossary part of our requirements page, with Yusuf. • Created the user persona and the use scenario of Salih Okutan. • Concerning the Class Diagram of the project, attended to several discussions with team, and took part in planning and drawing phase of the Class diagrams. Filled the required roles when needed, both in two versions of our Class diagram. • Created Economic Event Search and Semantic Search sequence diagrams and updated them according to the feedbacks. • I am also assigned to correct the spelling errors in all of team’s contents.
Ali Özden	• Made research about gitHub. • Created personal WikiPage. • Took part in the creation of mockups of the Taylan Aksoy’s story. • Took part of creation mock-up of Taylan Aksoy

5. Communication Plan

- **Communicator: Sercan Ersoy**

- **Note taker: Ceren Tahtasız**

Who	Why	When	Where	How
All team members	Weekly meetings	Tuesdays 17.30 - 18.30	BM, A5	Face-to-face
Available team members	Progress update	Sundays 19.00 - 20.00	Boğaziçi University or Online	Face-to-face or Skype
All team members	Urgent issues	When needed	Online	Whatsapp or Phone call
Team and Customer	Interaction with customer	When needed	Classroom or Online	Face-to-face or Piazza
All team members	Project related issues	All the time	Online	Github or Slack

Agenda will be prepared weekly on mondays by a selected group member who was decided on meeting.

Customers

- Suzan Üsküdarlı
- Meriç Turan

Team Members

- Sercan Ersoy (communicator)
- Ceren Tahtasız
- Furkan Aydar
- Efe Önal
- Ahmet Yiğit Gedik
- Barış Zöngür
- Alper Çakan
- Yusuf Yüksel

- **Mehmet Can Ünal**
- **Ali Özden**

Communication Mediums

1. **Face-to-face**
2. **Github**
3. **Slack**
4. **Whatsapp**
5. **Phone call**
6. **Skype**

7. Project Requirements

7.1. Functional Requirements

7.1.1. User Requirements

7.1.1.1. Guests

- **7.1.1.1.1.** The system shall allow the guests to view the price of any trading equipment.
- **7.1.1.1.2.** The system shall allow the guests to read user comments about trading equipment.
- **7.1.1.1.3.** The system shall allow the guests to view articles.

7.1.1.2. User Authentication and Account Management

7.1.1.2.1 Sign Up

- **7.1.1.2.1.1.** The system shall allow the user accounts to be either "basic" type or "trader" type.
- **7.1.1.2.1.2.** The system shall require the user to provide their name, surname, e-mail address and location to be able to create a basic account.
- **7.1.1.2.1.3.** The system shall require the user to provide their name, surname, e-mail address, location, id number and IBAN to be able to create a trader account.
- **7.1.1.2.1.4.** The system, when it requires the user to provide their location, shall require the user to provide it using Google Maps.

- **7.1.1.2.1.5.** 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.
- **7.1.1.2.1.6.** The system shall obtain the necessary sign up information using text inputs, unless otherwise specified.

7.1.1.2.2 Sign In

- **7.1.1.2.2.1.** The system shall allow users to sign in only using their Google accounts or by providing the necessary sign in information, username and password.

7.1.1.3. Trading

- **7.1.1.3.1.** The system shall, for the trading users, have a "My Investments" section which shall have the functionalities specified here (1.1.3).
- **7.1.1.3.2.** The system shall allow the trading users to invest on any trading equipment.
- **7.1.1.3.3.** The system shall allow the trading users to make a buy order for a specified rate.
- **7.1.1.3.4.** The system shall allow the trading users to set stop/loss limits.

7.1.1.4. Tracking

- **7.1.1.4.1.** 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 (7.1.1.4).
- **7.1.1.4.2.** 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.
- **7.1.1.4.3.** 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.

7.1.1.5. Comments

- **7.1.1.5.1.** The system shall allow users to comment on articles.
- **7.1.1.5.2.** The system shall allow users to comment on trading equipment.

7.1.2. System Requirements

7.1.2.1. Tracking

- **7.1.2.1.1.** System shall support searching for users and trading equipments considering all information available and filtering the search results.
- **7.1.2.1.2.** System shall support semantic search.
- **7.1.2.1.3.** System shall support location based search.

7.1.2.2. Recommendation

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

7.1.2.3. Ranking

- **7.1.2.3.1.** System shall allow traders to rank others idea.

7.1.2.4. Trading Equipments

- **7.1.2.4.1.** System shall allow traders to trade indices, stocks, ETFs, commodities, currencies, funds, bonds, and cryptocurrencies.
- **7.1.2.4.2.** 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.

7.1.2.5. Communication

- **7.1.2.5.1.** System shall allow users to follow other users and trading equipments.
- **7.1.2.5.2.** System shall allow users to share their ideas as articles.
- **7.1.2.5.3.** System shall allow users to comment and rate ideas of other users.
- **7.1.2.5.4.** System shall allow users to comment about trading equipment.

7.1.2.6. Profit

- **7.1.2.6.1.** System shall have a Profit/Loss section for each user.
- **7.1.2.6.2.** This section should be private for each user.
- **7.1.2.6.3.** System shall allow basic users to see their profit/loss amount in terms of the currency they choose by manually entering their investments.
- **7.1.2.6.4.** 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.

7.1.2.7. Portfolio

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

7.1.2.8. Notification

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

7.1.2.9. Account

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

7.1.2.10. Event

- **7.1.2.10.1.** System shall have events section for users which contains economic events.
- **7.1.2.10.2.** Events shall have significance level and country base properties.
- **7.1.2.10.3.** Events shall be able to searched, filtered and chased.

7.1.2.11. Deployment

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

7.1.2.12. Standards

- **7.1.2.12.1.** System shall support W3C Web Annotation Data Model.
- **7.1.2.12.2.** System shall follow W3C Web Annotation Protocol.
- **7.1.2.12.3.** System shall follow the standards introduced by the World Wide Web Consortium (W3C).

7.1.2.13. Ethics and Legal

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

7.2. Nonfunctional Requirements

7.2.1. Accessibility

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

7.2.2. Availability

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

7.2.3. Reliability

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

7.2.4. Security

- **7.2.4.1.** The system should remind users to change their password after using it 1 year.
- **7.2.4.2.** 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.
- **7.2.4.3.** The system should not allow user to login for 10 minutes after 5 unsuccessful login attempts.
- **7.2.4.4.** The system should session expiration if a user is inactive for 5 minutes.
- **7.2.4.5.** The system should have human verification tool (reCAPTCHA) to prevent bots.

7.2.5. Privacy

- **7.2.5.1.** The system shall secure the private data of users according to information privacy law.

7.2.6. Database

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

Glossary

User: A person who is signed into the system.User's activities vary accordingly to its registration status.

Guest: A person who is not signed in to the system, allowed to view price of a trading equipment, read user comments about trading equipment and see articles written by users.

Account: A property which every user who signed-up to the system owns and which links users to the Trader, allowing them to use the following functionalities of the system : Signing in, giving buy orders, creating and editing their own portfolio, annotating(through comments), setting alerts, receiving notifications, following economic events, investing on trading equipments, searching users or trading equipments and filtering the search results, viewing articles. Given list of functions may vary due to the user's Account type.

Basic Account, Basic User: A registered user of Trader system, which can use fundamental functions of the application, namely: learning their profit/loss state according to their investments done manually, commenting on articles and trading equipments, filtering and following economic events, viewing articles. Alias, basic users can benefit from all the functionalities that Trader users have, except investing directly on trading equipments through our service.

Trading Account, Trading User: Users registered to the system, who are able to invest in any trading equipment and have all the rights given to the basic users, additionally.

Google Maps: A service used to retrieve location information. In order to register, users should specify their location on Google Maps API provided to the application.

Trading Equipment: Materials that can be interacted with various ways such as trading, commenting, rating, setting alerts. Indices, stocks, ETFs, commodities, currencies, funds, bonds and cryptocurrencies are examples.

Buy Order: Command given by the trading users, in order to invest on a specific trading equipment at a specific price.

Recommendation system: A system that recommends articles or trading equipment to the users based on their histories.

Portfolio: A user feature that enables tracking the changes in the trading equipment they are interested in.

Alert: A user feature that enables them to notify about trading equipments which are indices, stocks, ETFs, commodities, currencies,funds, bonds, and cryptocurrencies.

Notification: A system feature that gives information to users in accordance with their alerts and gives information about users they follow , if they write an article or make a comment.

Economic Events: A user feature in the "Events" section which have different significance levels (e.g., one star, two star, three star) and filterable by users considering their significance level and country base.

Annotation: A note added by way of comment or explanation. In Trader, commenting on ideas of users or on trading equipments serves as annotation. Graphs or figures included in the application should also be annotable.

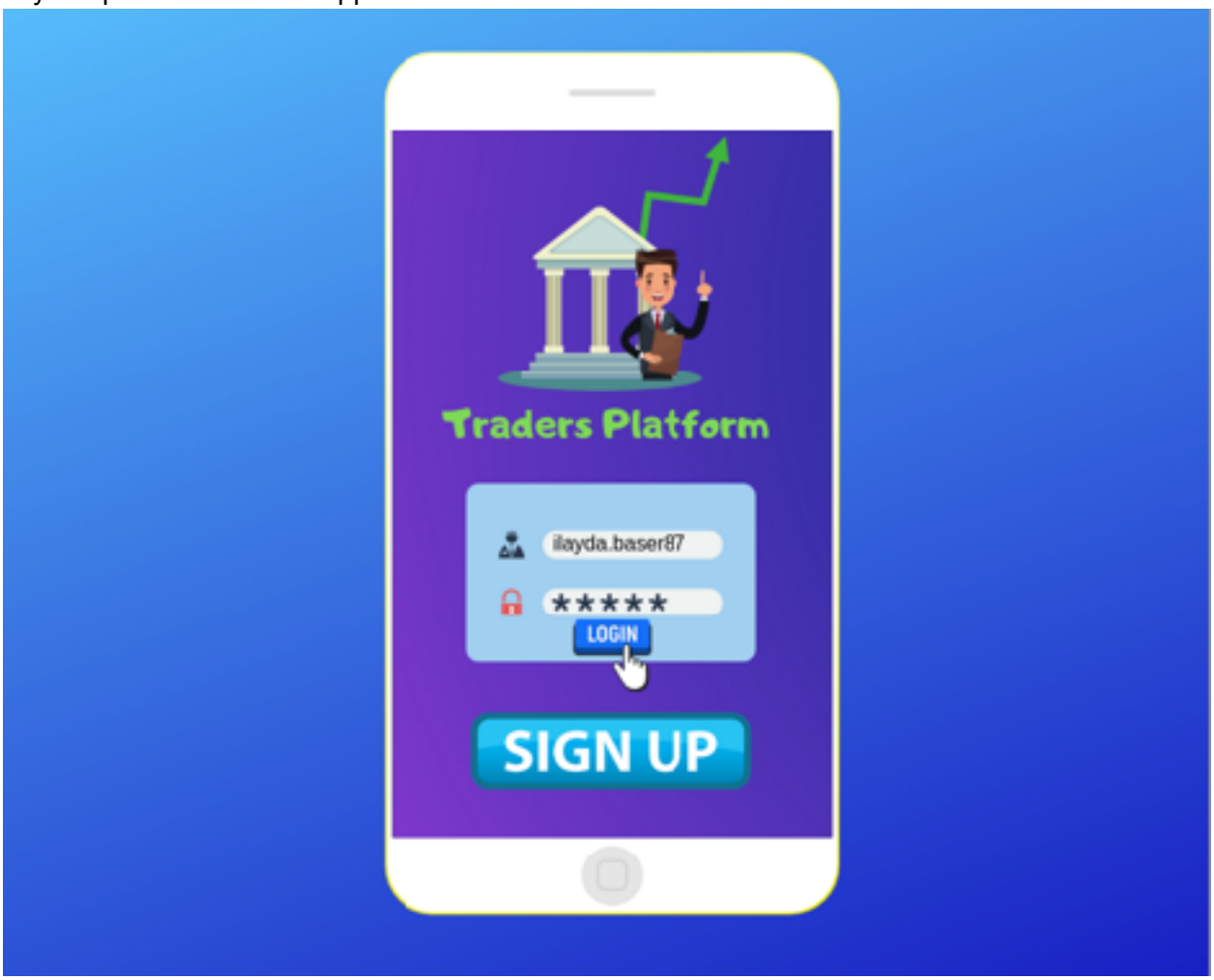
Sign-up: Process which should register the customers into the Trader system. Users are expected to provide information depending on their preference of user type. In order to sign-up, basic users should provide their name, surname, e-mail address and location whereas trader users should provide their name, surname, e-mail address, location, id number and IBAN.

8. Mockups

8.1. Scenario 1 (İlayda Başer)

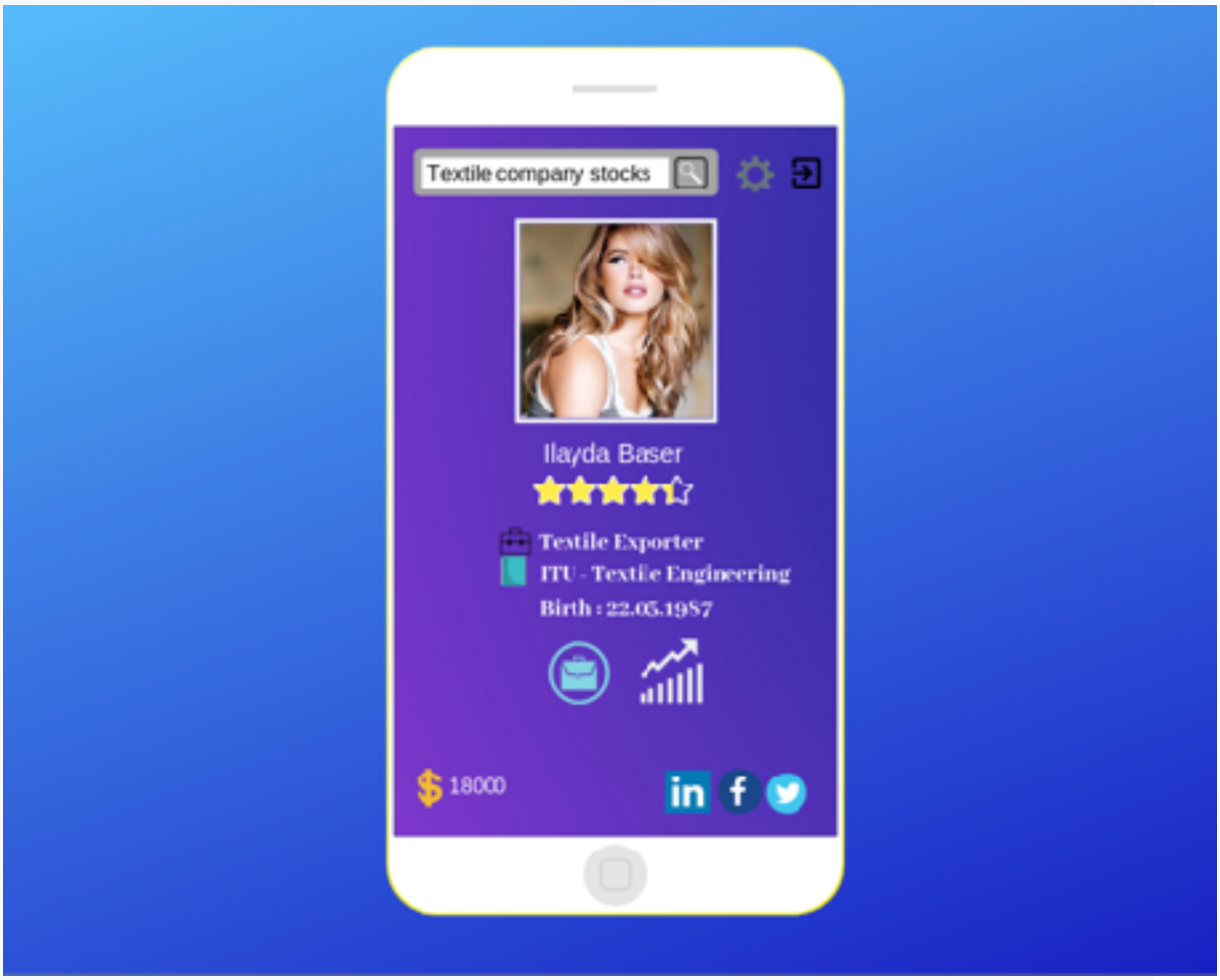
8.1.1. Flow

- 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.



8.2. Scenario 2 (İlayda Başer)

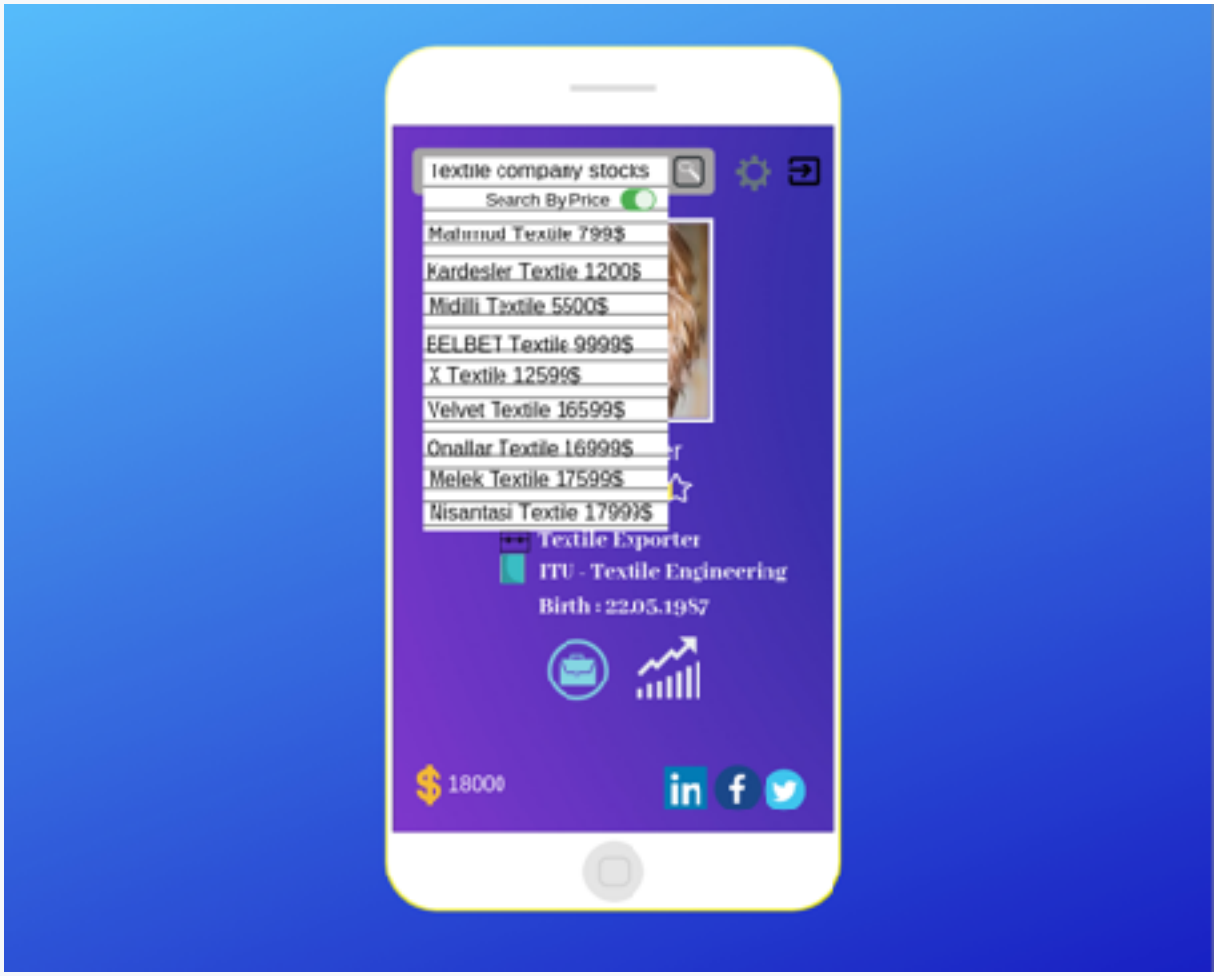
8.2.1. Preconditions

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

8.2.2. Flow

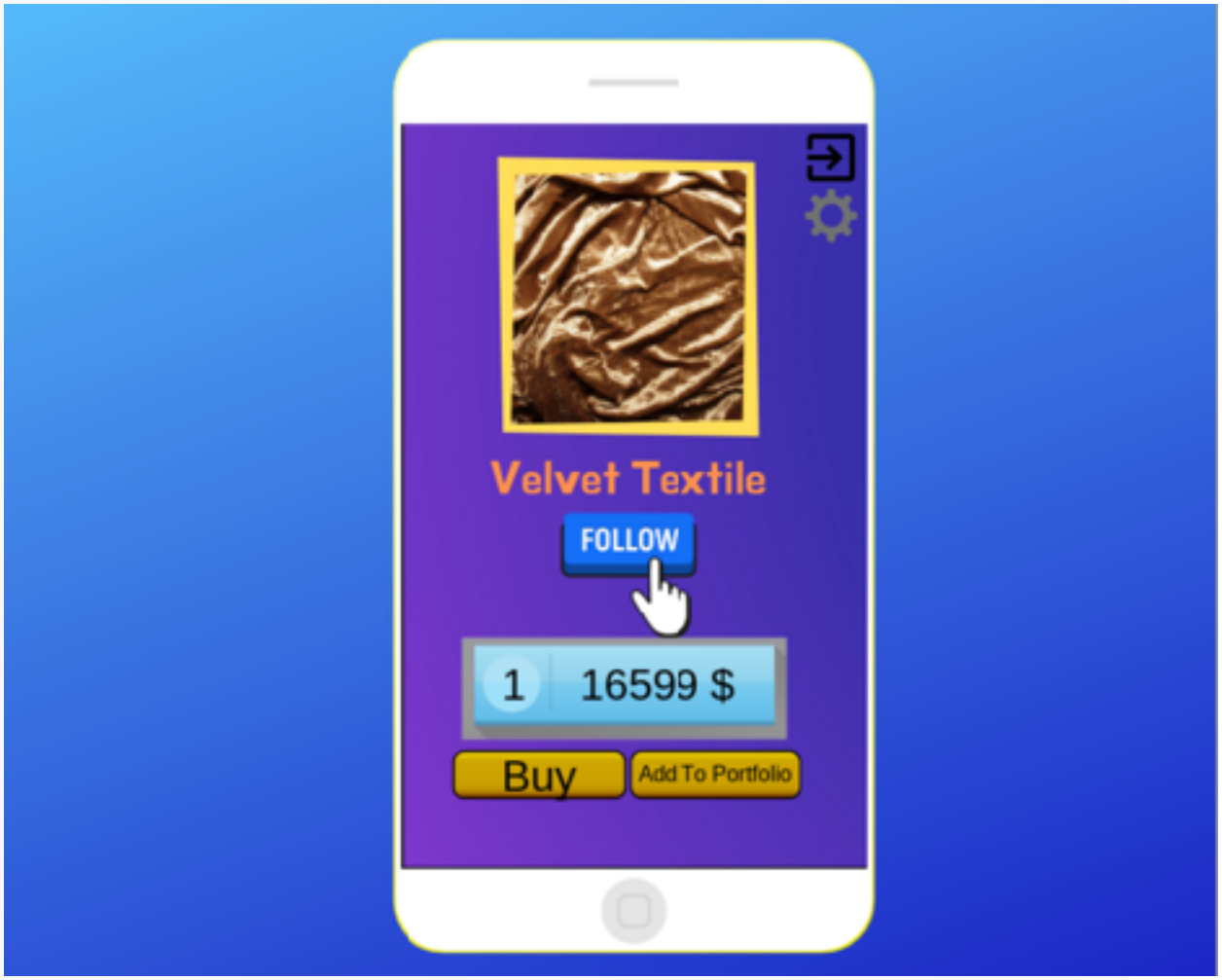
- Ilayda searches for textile company stock enabling the price filter so the offers which exceed her budget are not visible to her. She picks an offer of her choice (Velvet

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.



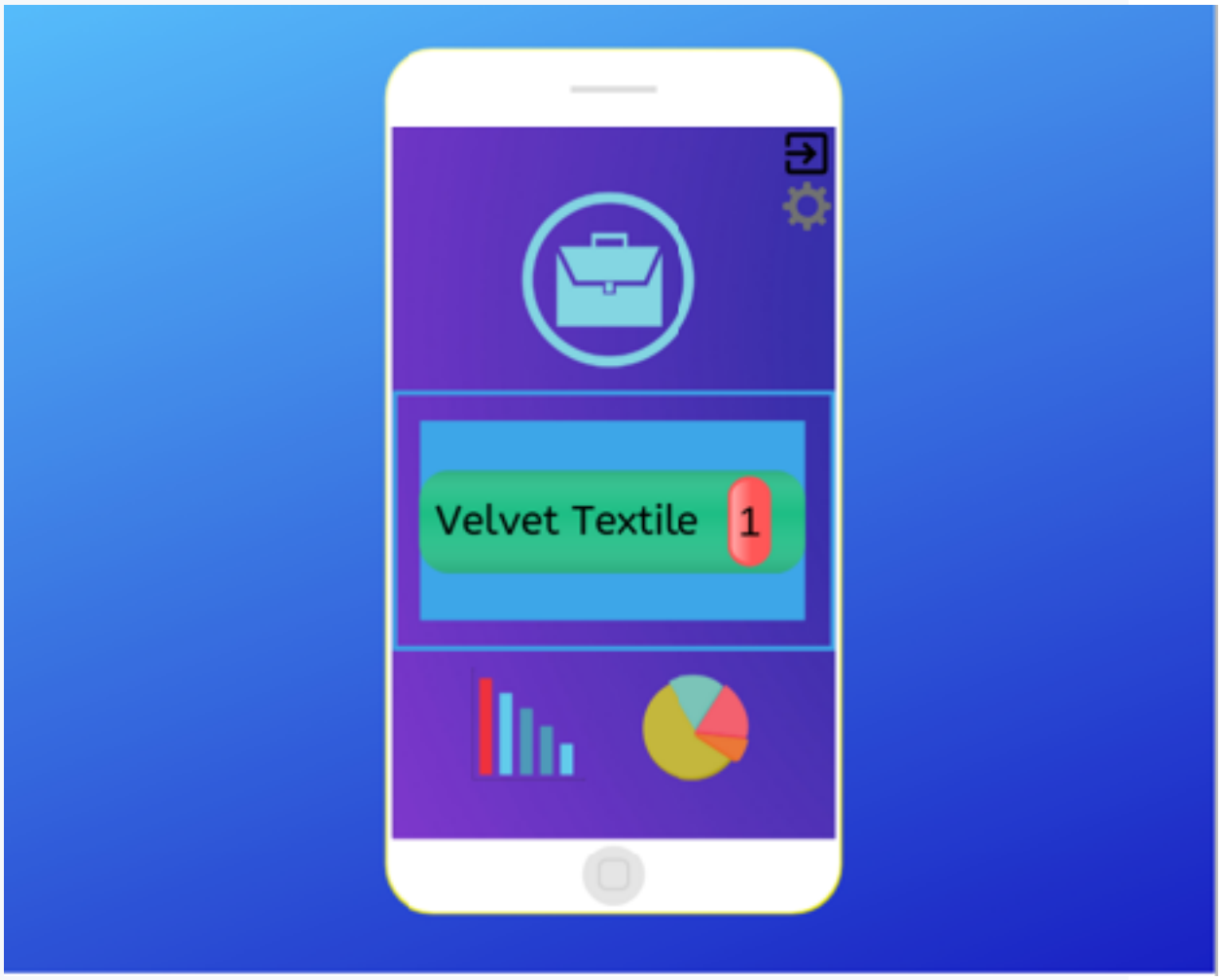
8.3. Scenario 3 (İlayda Başer)

8.3.1. Preconditions

- Ilayda 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).

8.3.2. Flow

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



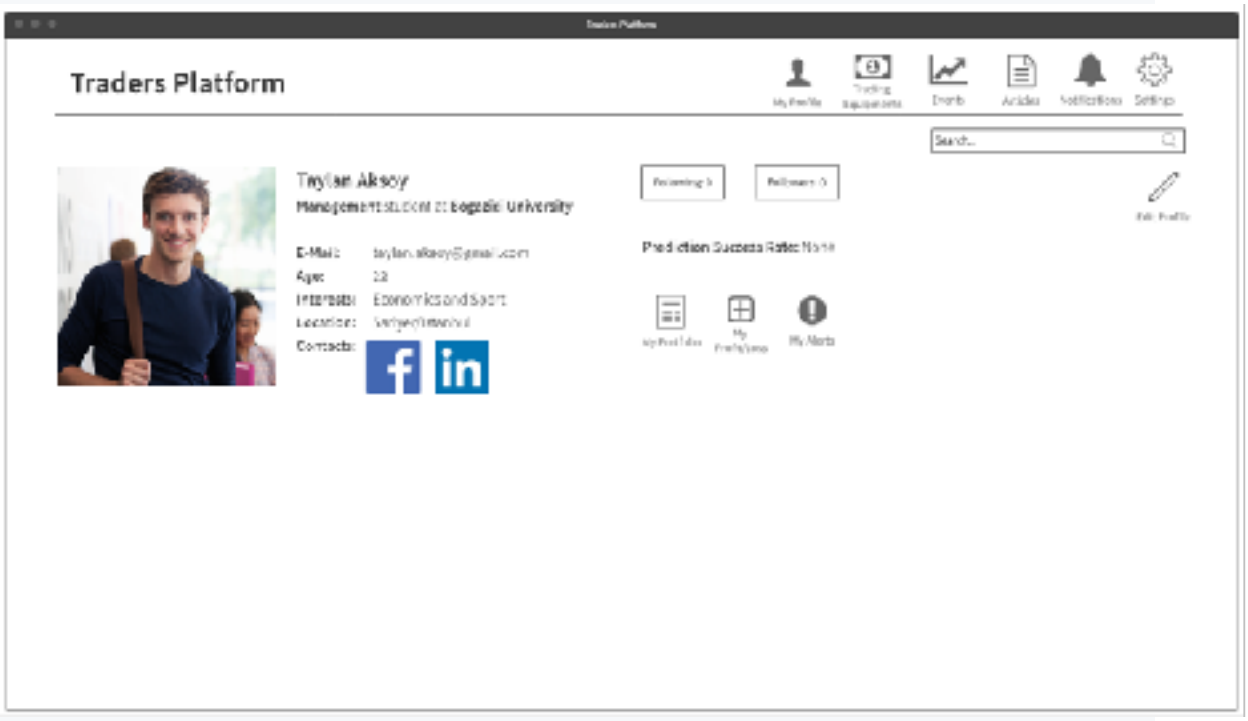
8.4. Scenario 4 (Taylan Aksoy)

8.4.1. Preconditions

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

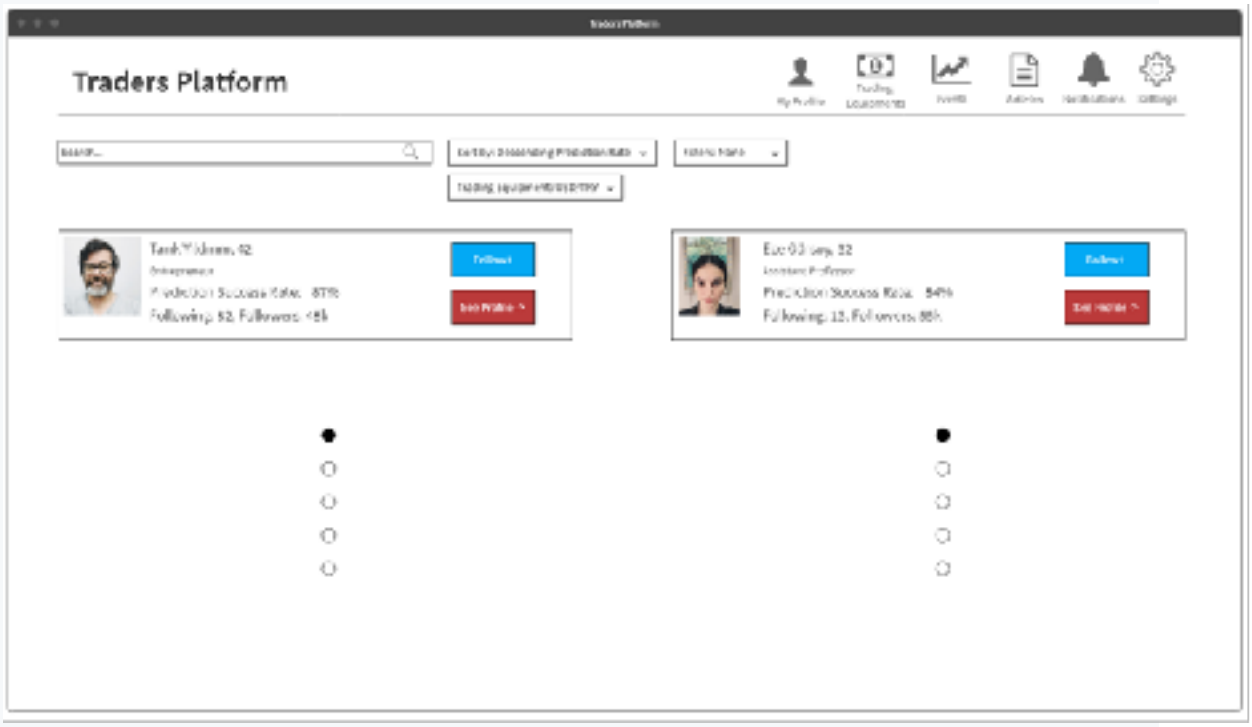
8.4.2. Flow

- 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.



- 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.



8.5. Scenario 5 (Taylan Aksoy)

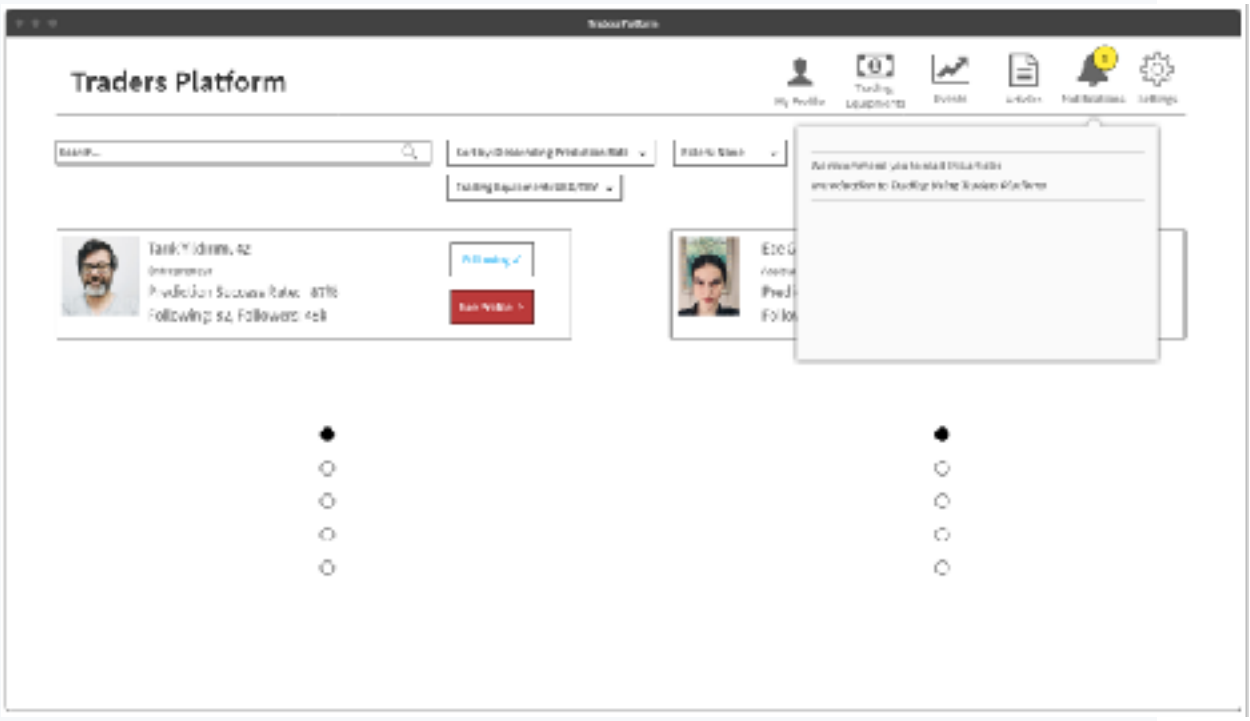
8.5.1. Preconditions

- 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.

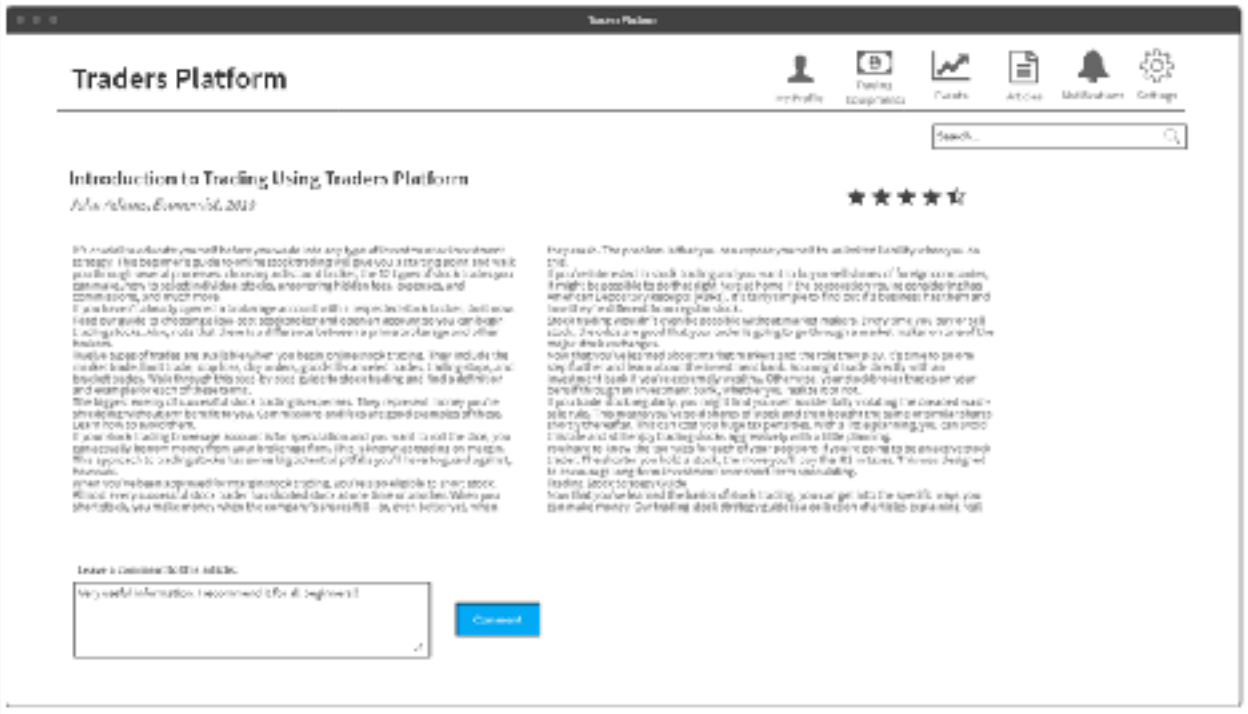
8.5.2. Flow

- 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".



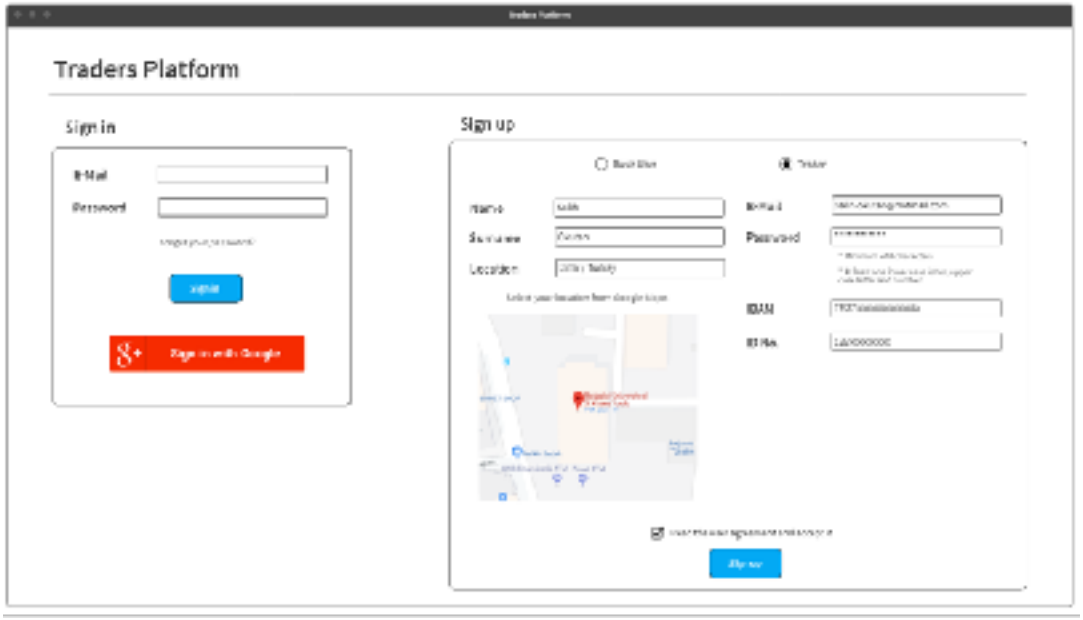
- After clicking the article name, he is directed to article page. He reads the article and leaves a comment.



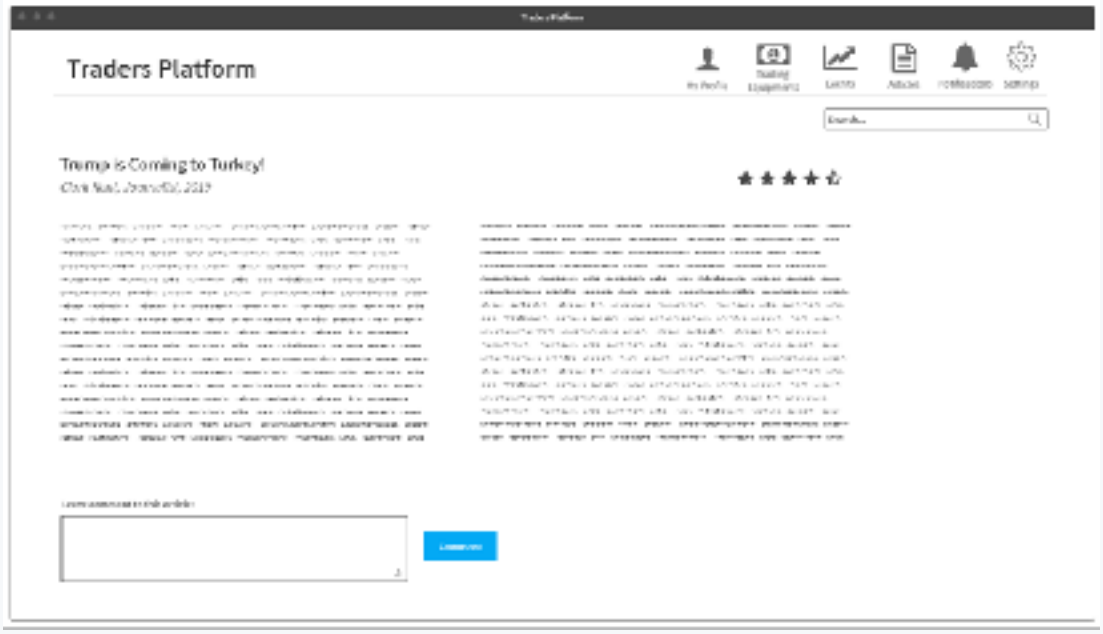
8.6. Scenario 6 (Salih Okutan)

8.6.1. Flow

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



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



- Then he sets an alert for a specific trader user's equipment, USD, in order to get a notification when USD decreases below 5 Turkish Liras.

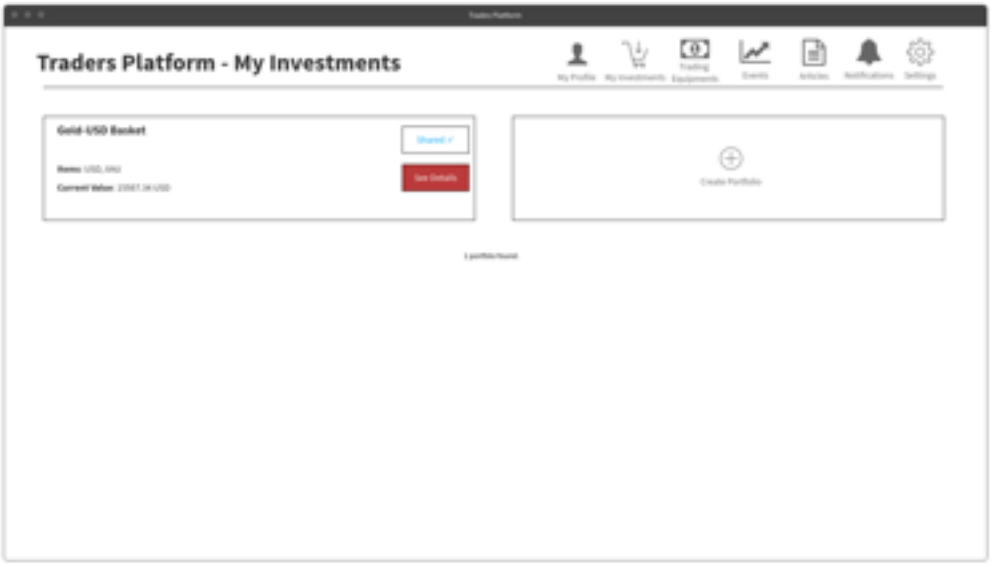
- After a week, he makes a buy order of 2000 USD when 1 USD = 4.93 TL.

8.7. Scenario 7 (Ahmet Kazan)

8.7.1. Flow

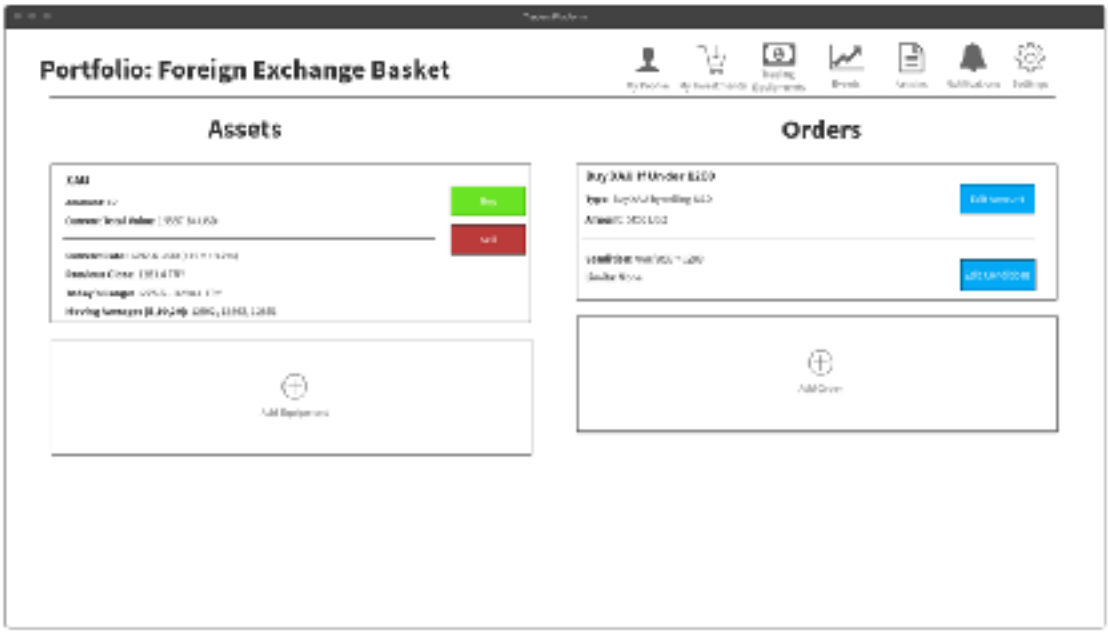
- Ahmet signs up to the platform on Web using his Google account.

- 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.



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

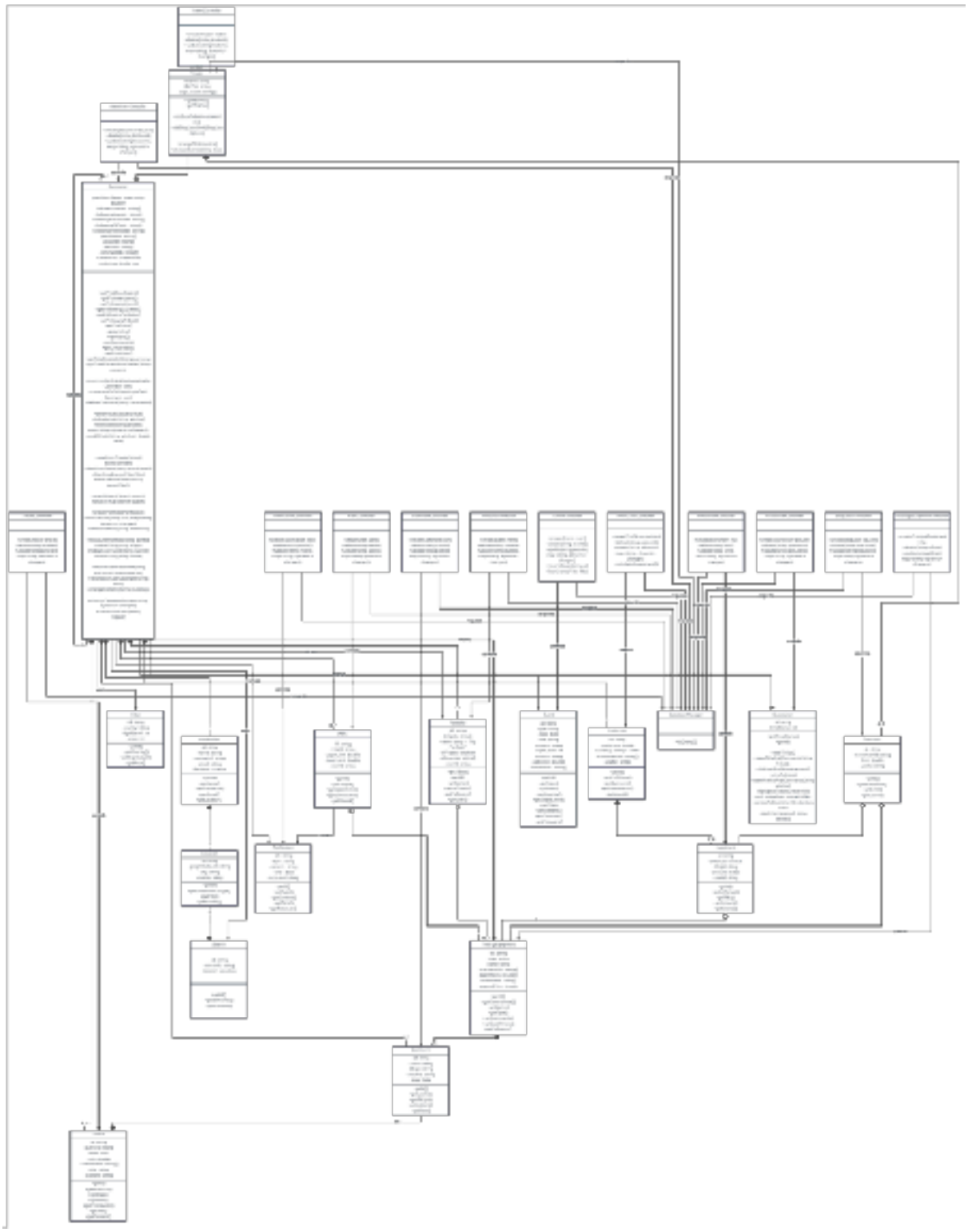
- Also, he creates an automatic buy order to buy some gold if the gold/USD ratio drop significantly.



8. Design

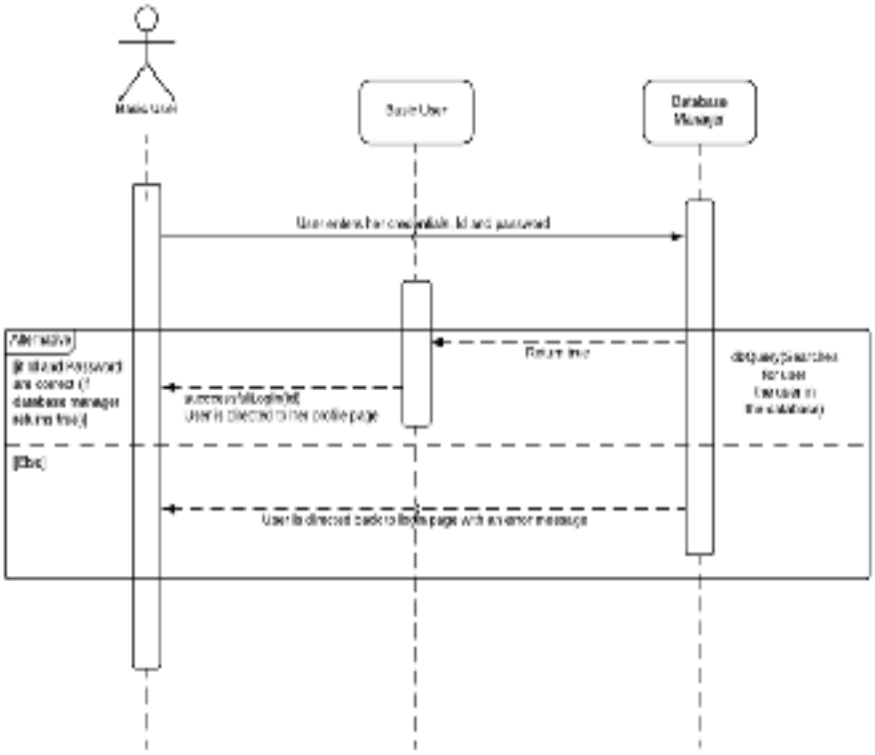
8.1. Use Case Diagrams

8.2. Class Diagram

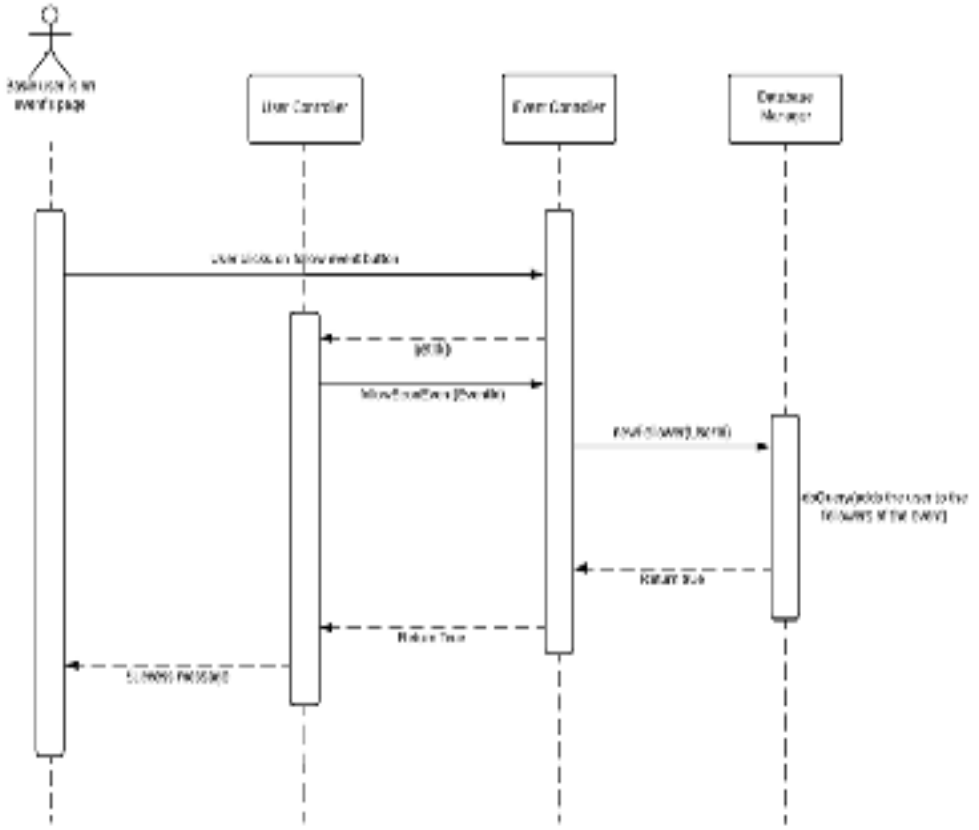


8.3. Sequence Diagrams

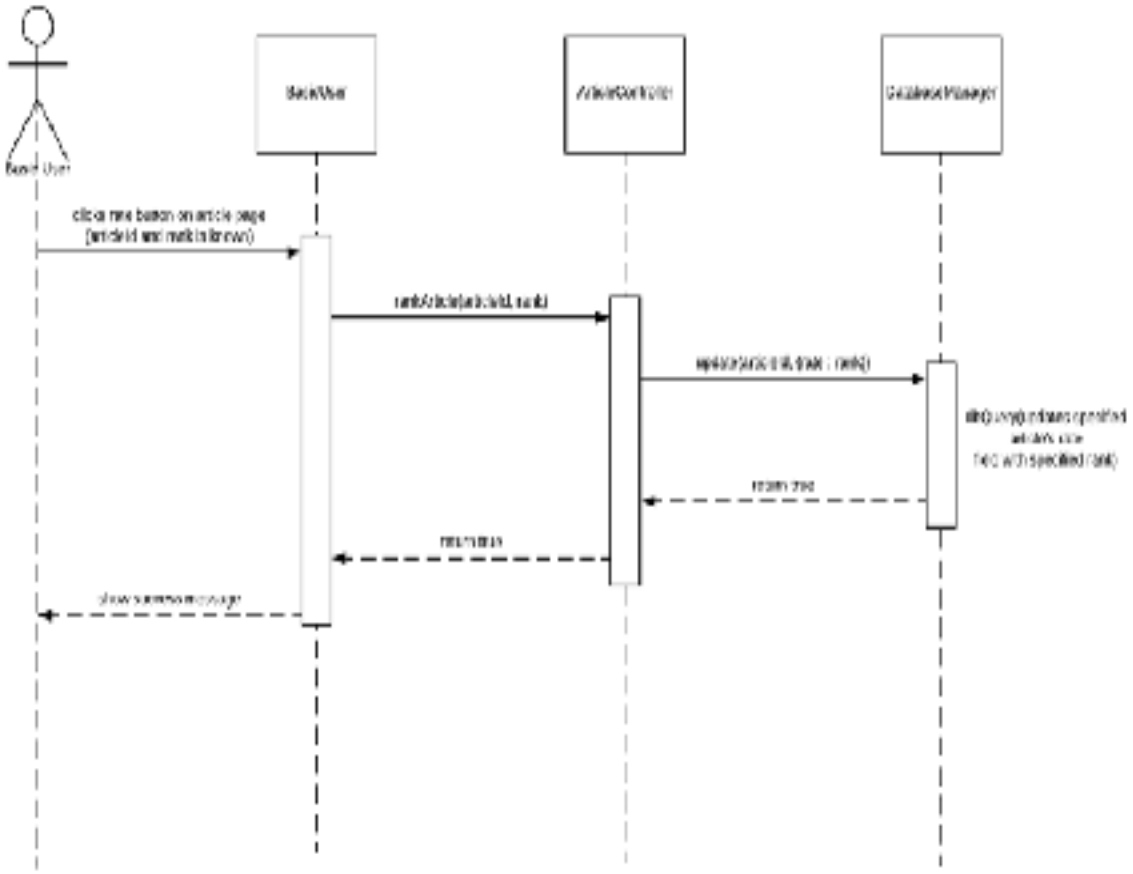
8.3.1. Login



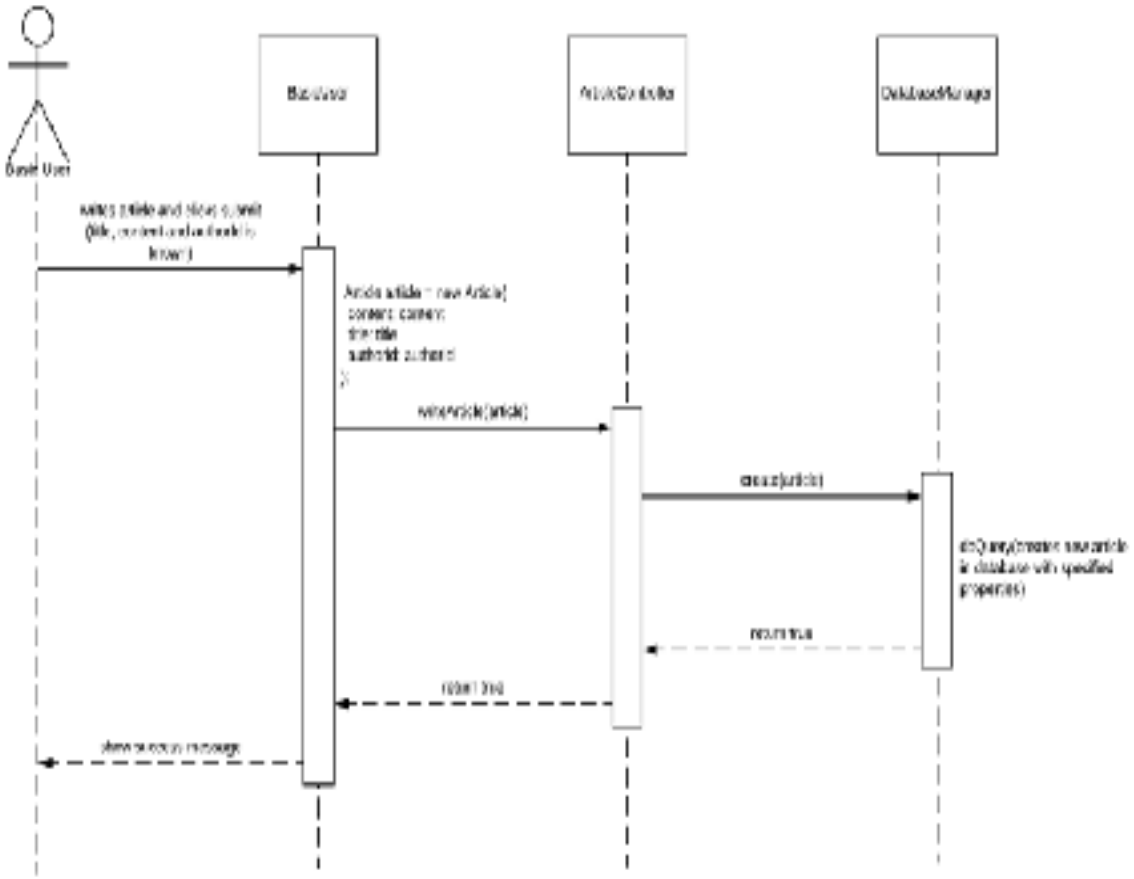
8.3.2. Following an Economic Event


User Controller
Event Controller

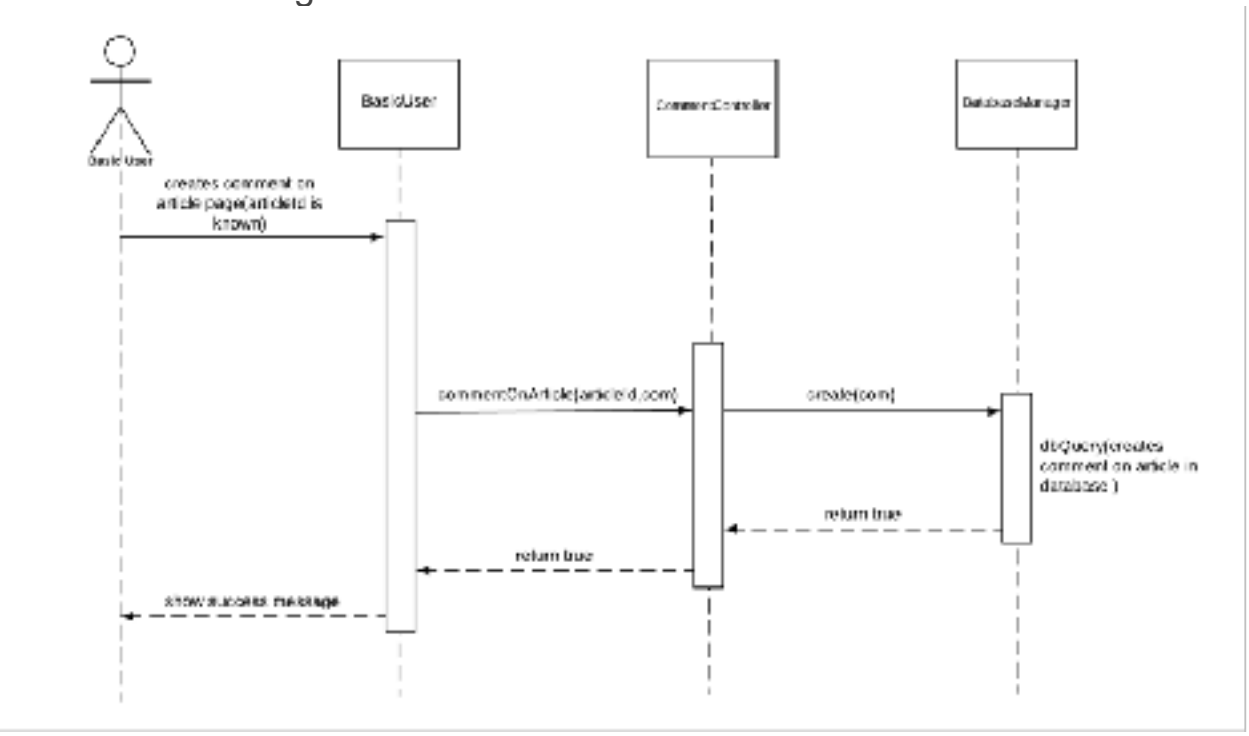
8.3.3. Rating an Article



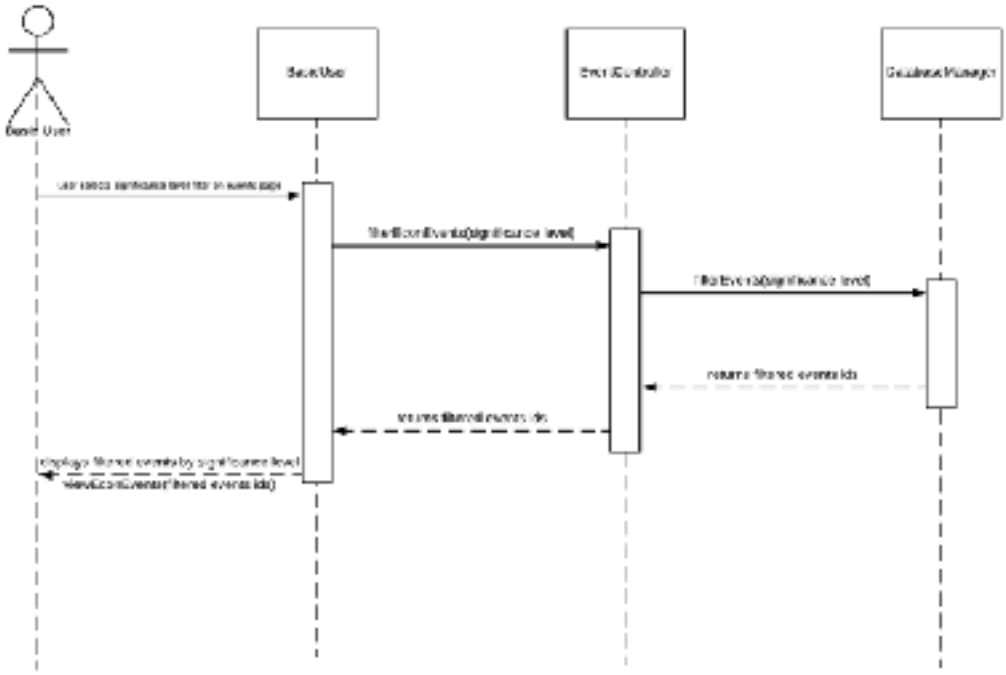
8.3.4. Writing an Article



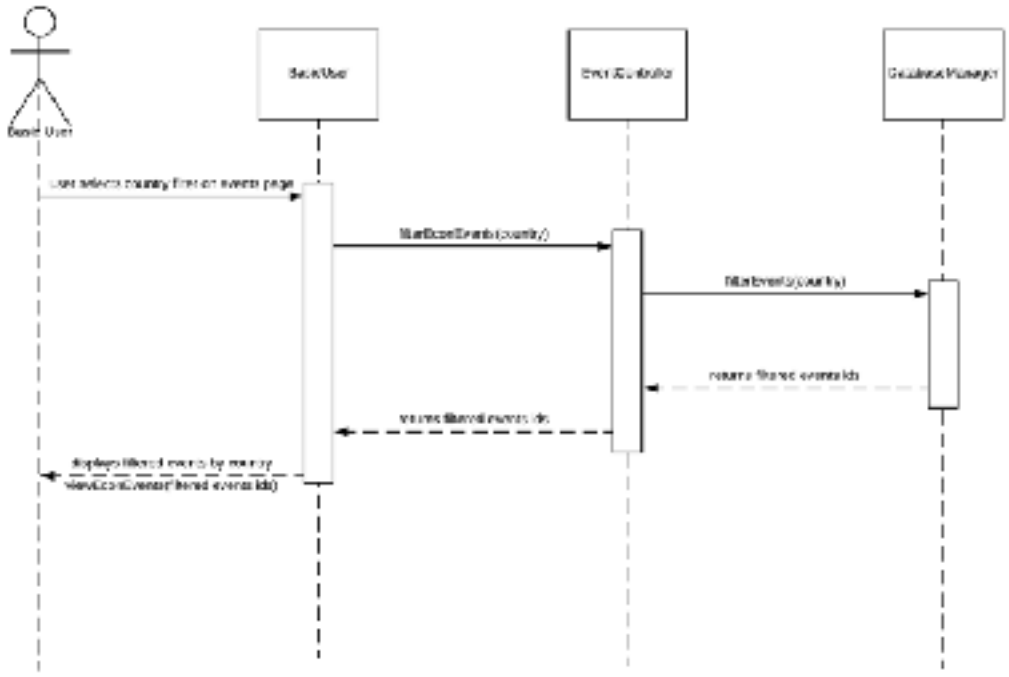
8.3.5. Commenting On Article



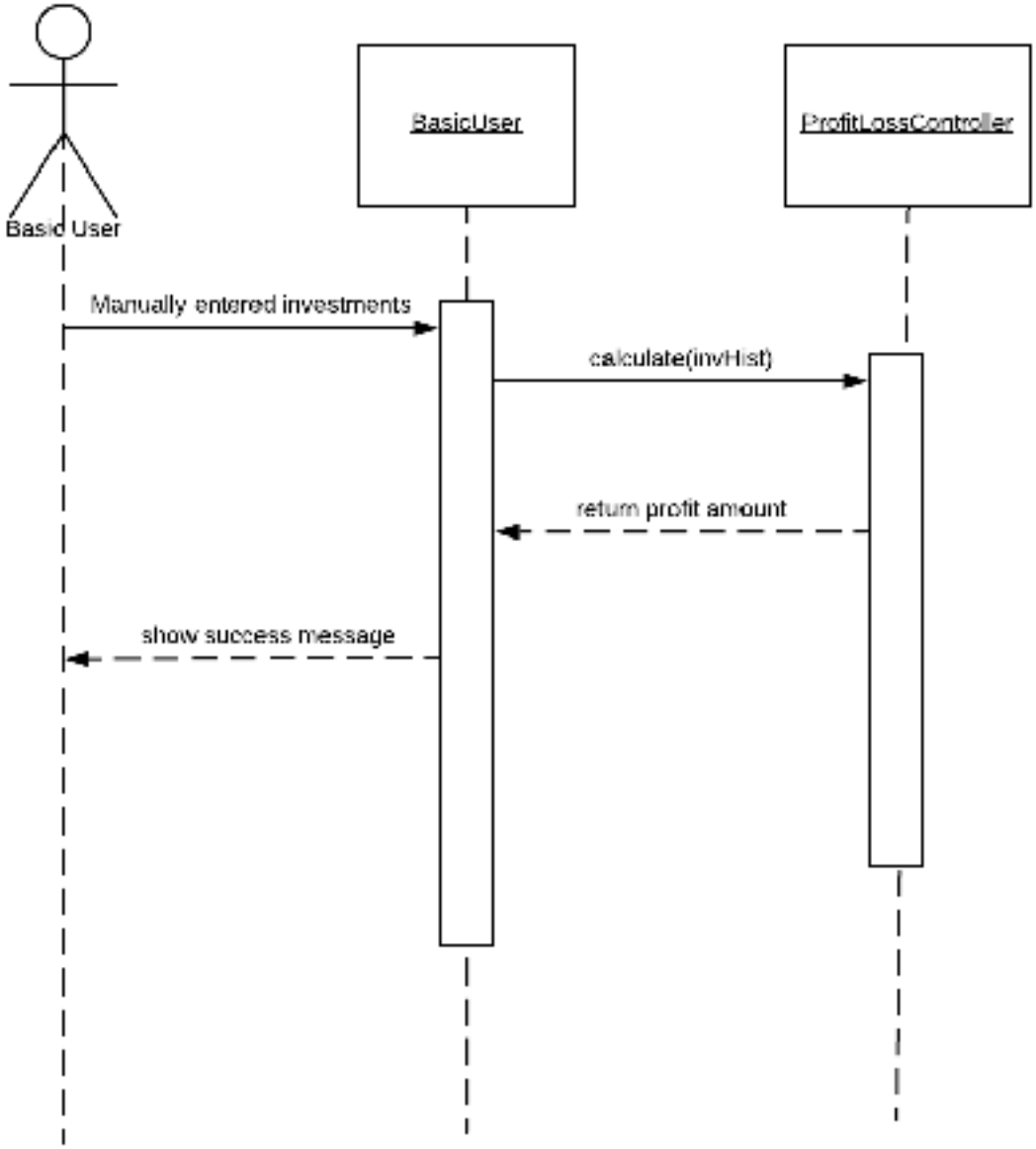
8.3.6. Filtering Events by Country



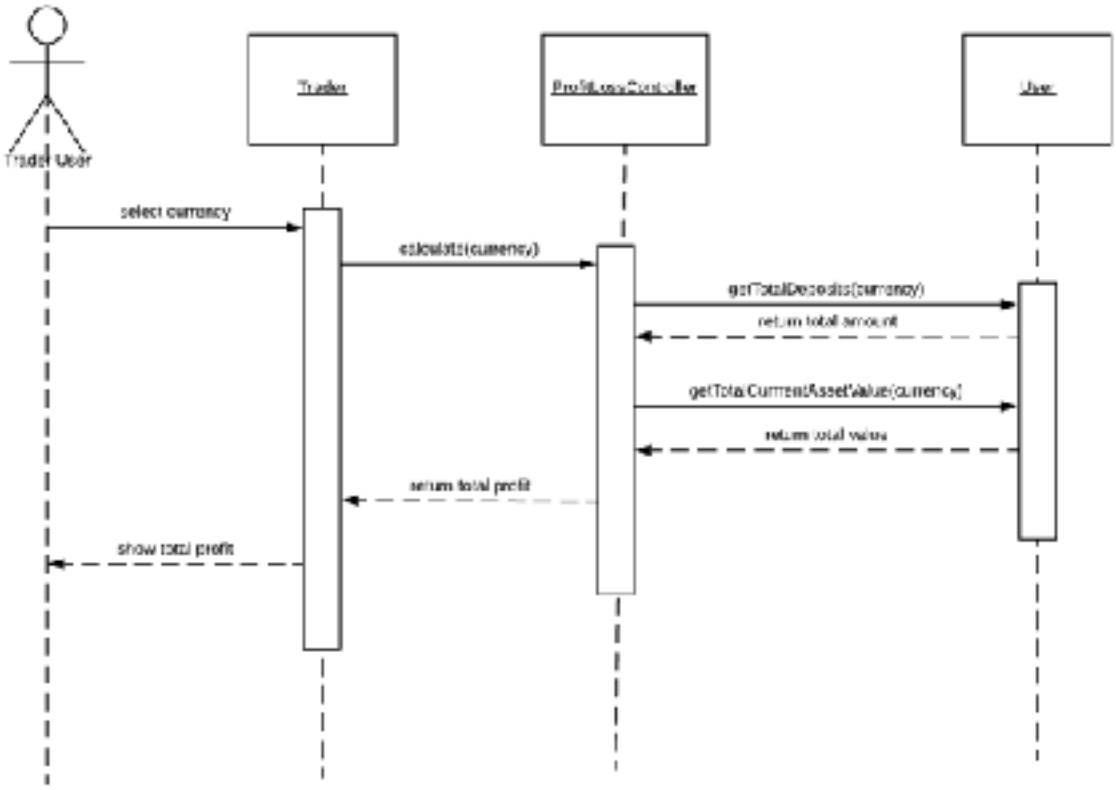
8.3.7. Filtering Events by Significance Level



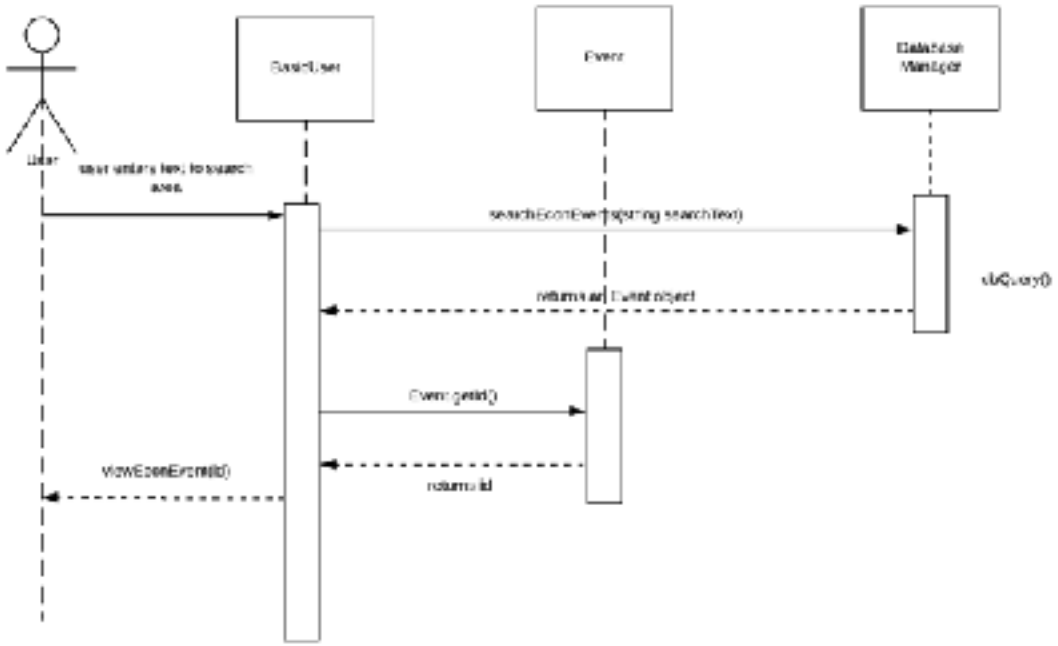
8.3.8. Calculate Profit/Loss Manually



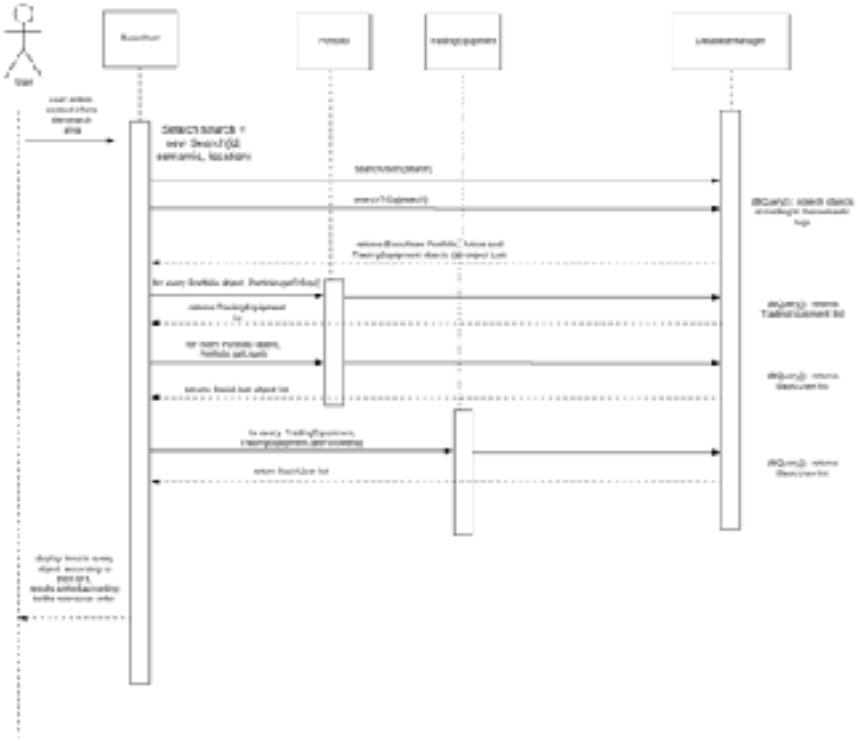
8.3.9. Calculate Profit/Loss



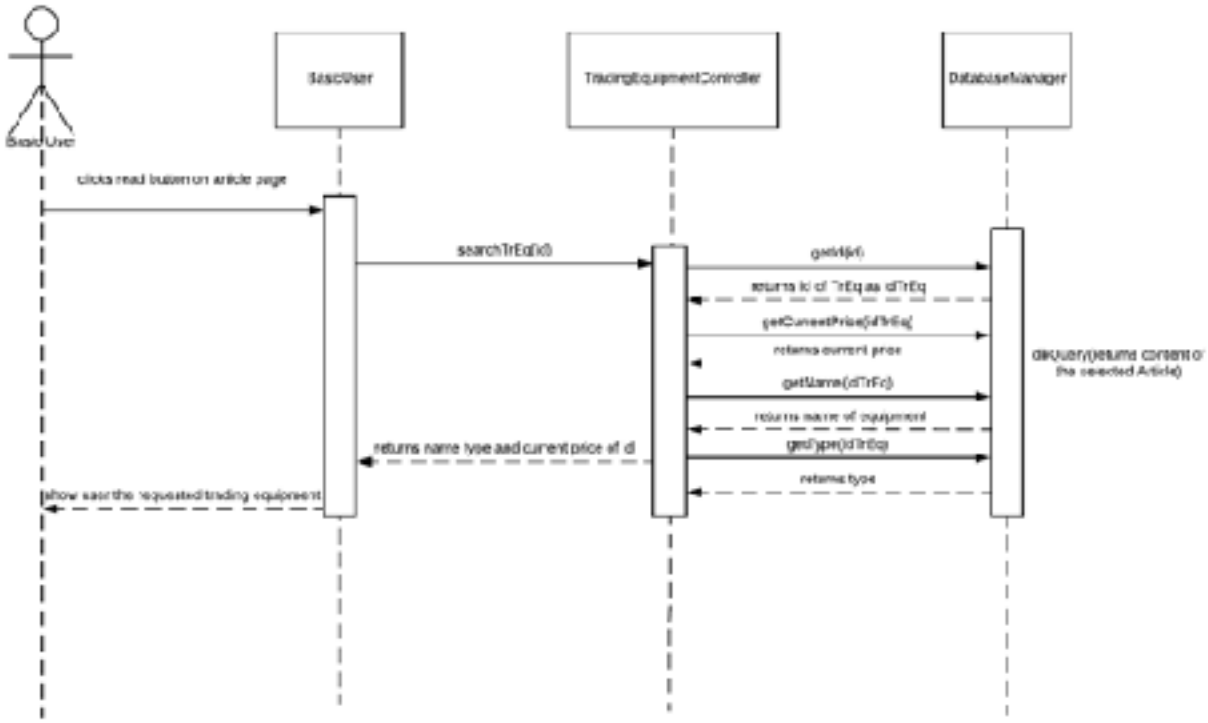
8.3.10. Search Economic Events



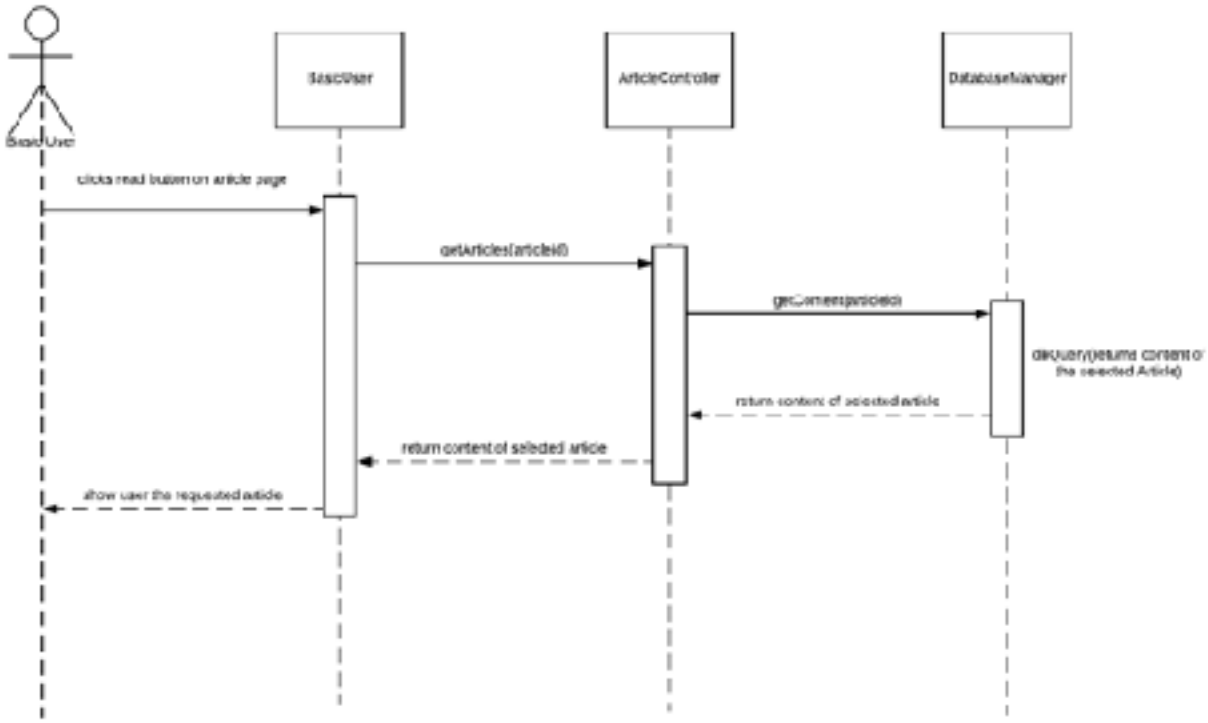
8.3.11. Semantic Search For Users/Trading Equipments



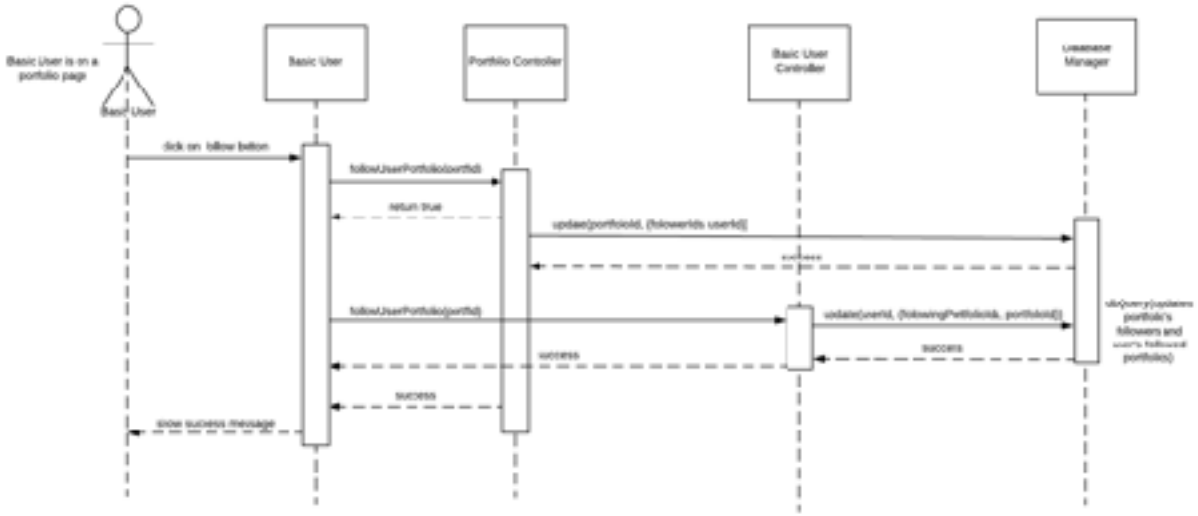
8.3.12. View Trading Equipment



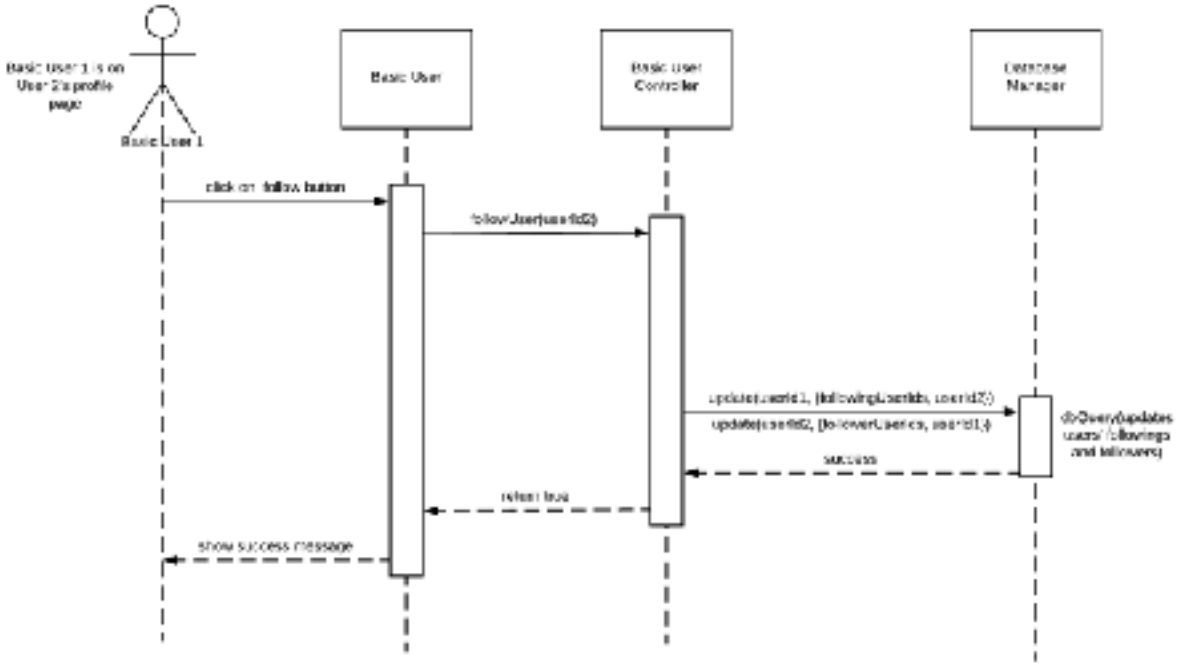
8.3.13. Read an Article



8.3.14. Follow User Portfolio



8.3.15. Follow Another User



8.3.14. Commenting On Trading Equipment