

BOGAZICI UNIVERSITY
DEPARTMENT OF COMPUTER
ENGINEERING
Project Development in Software Engineering

**FINAL MILESTONE
REPORT
(Group 1)**

Emirhan Saraç
Ufuk Yılmaz
Abdullah Coşkun
Eray Kurtuluş
İlker Özkan

Mete Han Kurt
Yağmur Ceren Dardağan
Mehmet Altay İnce
İrem Üstünboyacıoğlu
Ömer F. Toptaş



Contents

1 Project Assessment	9
2 List and Status of Deliverables	11
3 Summary of Team Member's Work	13
3.1 Backend	13
3.2 Frontend	15
3.3 Android	17
4 Annotation Implementation & W3C Standard Compliance	19
4.1 Model	20
4.2 Annotation on Text	20
4.3 Annotation on Image	22
4.4 Annotation Implementation on Android	23
5 Design Documents	26
5.1 Mockups	26
5.1.1 Android	26
5.1.2 Web	31
5.2 Software Design	35
5.2.1 Use Case Diagram	35
5.2.2 Class Diagram	36
5.2.3 Sequence Diagram	37
6 Communication Plan	45
7 Requirements	46
8 Project Plan	54
9 Evaluation of tools and managing the project	55
9.1 Tracking Tools(GitHub)	55
9.1.1 Git Feature of Github	55
9.1.2 Issues	55
9.1.3 Pull Requests	55
9.1.4 Project section of the Github	55
9.1.5 Wiki	56

9.2 Backend	56
9.2.1 Python Django	56
9.2.2 Postman	56
9.2.3 Docker	56
9.2.4 PyCharm	57
9.3 Web	57
9.3.1 ReactJS	57
9.3.2 Material-ui library	57
9.3.3 Axios	58
9.3.4 History	58
9.3.5 React-location-picker	58
9.3.6 React-router-dom	58
9.3.7 ESLint	58
9.3.8 Webstorm IDE	58
9.3.9 Visual Studio Code	59
9.4 Android	59
9.4.1 Kotlin for Android	59
9.4.2 Retrofit	59
9.4.3 IDE and Simulation Preferences	60
10 User Manual	60
10.1 Android User Manual	61
10.2 Web User Manual	89
11 System Manual	100
11.1 Frontend	100
11.2 Android	100
11.2.1 Software Description	100
11.2.2 Benefits Value	101
11.2.3 Platform Requirements	101
11.2.4 Importing Project to Android Studio	102
11.2.5 Software Details	102
11.3 Backend	107
11.3.1 Set Up In Local	108
11.3.2 Set Up With Docker	109
11.4 Annotation Backend	109

12 API Documentation	111
12.1 POST trader register	111
12.1.1 Headers	111
12.1.2 Body	111
12.2 POST basic register	111
12.2.1 Headers	111
12.2.2 Body	111
12.3 POST login	112
12.3.1 Headers	112
12.3.2 Body	112
12.4 GET auto login/profile	112
12.4.1 Headers	112
12.5 GET User Retrieve	112
12.5.1 Headers	112
12.6 PUT User Update	113
12.6.1 Headers	113
12.6.2 Body	113
12.7 PUT User Password Change	113
12.7.1 Headers	113
12.7.2 Body	113
12.8 POST search	113
12.8.1 Headers	114
12.8.2 Body	114
12.9 POST Follow Someone With ID	114
12.9.1 Headers	114
12.9.2 Body	114
12.10 GET List Follower Of Current User	114
12.10.1 Headers	114
12.11 GET List Follower Of Given User	115
12.11.1 Headers	115
12.11.2 Body	115
12.12 GET List Following Of Current User	115
12.12.1 Headers	115
12.13 GET List Following Of Given User	115
12.13.1 Headers	115
12.13.2 Body	116
12.14 DELETE Unfollow Someone	116
12.14.1 Headers	116

12.14.2 Body	116
12.15GET Currency List	116
12.16GET Crypto Currency List	116
12.17GET Metal Currency List	116
12.18GET Stock List	116
12.19GET Etf List	117
12.20GET Trace Indices List	117
12.21GET Currency Last Month	117
12.22GET Currency	117
12.23GET currency convert	117
12.23.1 Headers	117
12.23.2 Body	117
12.24GET Metal Currency	117
12.25GET Crypto Currency	117
12.26GET Crypto Currency Last Month	118
12.27GET Stock Currency	118
12.28GET Trace Indices Currency	118
12.29GET Gainers in Trace Indices	118
12.30GET Last Month of Stock	118
12.30.1 Headers	118
12.30.2 Body	118
12.31POST Forgot Password	118
12.31.1 Headers	118
12.31.2 Body	119
12.32PUT User Password Change Forgot Password	119
12.32.1 Headers	119
12.32.2 Body	119
12.33PUT User Upgrade	119
12.33.1 Headers	119
12.34PUT User Downgrade	119
12.34.1 Headers	119
12.35POST Create Article	120
12.35.1 Headers	120
12.35.2 Body	120
12.36PUT Update Article	120
12.36.1 Headers	120
12.36.2 Body	120
12.37GET List Article	120

12.37.1 Headers	121
12.38 POST List Public Articles	121
12.38.1 Headers	121
12.39 POST List Article With User ID	121
12.39.1 Headers	121
12.40 DEL Delete Article	121
12.40.1 Headers	121
12.40.2 Body	121
12.41 POST Wallet Create	121
12.41.1 Headers	122
12.42 POST Wallet Retrieve	122
12.42.1 Headers	122
12.43 PUT Send USD to Wallet	122
12.43.1 Headers	122
12.43.2 Body	122
12.44 PUT Take Equipment	122
12.44.1 Headers	122
12.44.2 Body	123
12.45 PUT Sell Equipment	123
12.45.1 Headers	123
12.45.2 Body	123
12.46 POST Create Comment	123
12.46.1 Headers	123
12.46.2 Body	123
12.47 POST Buy Order	124
12.47.1 Headers	124
12.47.2 Body	124
12.48 POST Sell Order	124
12.48.1 Headers	124
12.48.2 Body	124
12.49 POST Follow Portfolio	125
12.49.1 Headers	125
12.49.2 Body	125
12.50 GET Get Portfolio Follow	125
12.50.1 Headers	125
12.51 GET List Followers With Portfolio Id	125
12.51.1 Headers	125
12.52 GET List All	125

12.52.1 Headers	126
12.53GET List Portfolios With Follower Id	126
12.53.1 Headers	126
12.54GET Is Following by User	126
12.54.1 Headers	126
12.55DEL Unfollow Portfolio	126
12.55.1 Headers	126
12.55.2 Body	126
12.56POST Portfolio Create	127
12.56.1 Headers	127
12.56.2 Body	127
12.57PUT Portfolio Update	127
12.57.1 Headers	127
12.57.2 Body	127
12.58DEL Portfolio Delete	127
12.58.1 Headers	128
12.59GET Portfolio Retrieve	128
12.59.1 Headers	128
12.60GET Portfolio List	128
12.60.1 Headers	128
12.61POST Notification Create	128
12.61.1 Headers	128
12.61.2 Body	128
12.62POST Set Notification	129
12.62.1 Headers	129
12.62.2 Body	129
12.63GET List Set Notification	129
12.63.1 Headers	129
12.64GET List Notification	129
12.64.1 Headers	129
12.64.2 Body	129
12.65POST Create Comment	130
12.65.1 Headers	130
12.65.2 Body	130
12.66DEL Delete Comment	130
12.66.1 Headers	130
12.66.2 Body	130
12.67PUT Update Comment	130

12.67.1 Headers	131
12.67.2 Body	131
12.68GET List Article Comments by Article Id	131
 12.68.1 Headers	131
12.69POST Like Article	131
 12.69.1 Headers	131
 12.69.2 Body	131
12.70POST Dislike Article	132
 12.70.1 Headers	132
 12.70.2 Body	132
12.71GET List Liked Articles by User Id	132
 12.71.1 Headers	132
12.72GET List Disliked Articles by User Id	132
 12.72.1 Headers	132
12.73GET List Article Likes by Article Id	132
 12.73.1 Headers	133
12.74GET List Article Dislikes by Article Id	133
 12.74.1 Headers	133
12.75GET Like Count by Article Id	133
 12.75.1 Headers	133
12.76GET Dislike Count by Article Id	133
 12.76.1 Headers	133
12.77GET isLikedByUser	133
 12.77.1 Headers	133
12.78GET isDislikedByUser	134
 12.78.1 Headers	134
12.79POST predict equipment	134
 12.79.1 Headers	134
 12.79.2 Body	134
12.80GET listevents	134
12.81GET searchevent	134
13 Annotation API Documentation	134
13.1 POST Create Annotation	134
 13.1.1 Headers	135
 13.1.2 Body	135
13.2 POST Create Creator	136
 13.2.1 Headers	136

13.2.2 Body	136
13.2.3 GET Creator List	136
13.2.4 GET Creator List Mobile	136
13.2.5 GET Annotation All List	136
13.2.6 GET is Creator Exist	136

1 Project Assessment

It is the time of the final delivery of the Khajiit project. It has been about 2 semesters to come to this point. Throughout the both semesters, our team has analyzed requirements, made project design documents such as mock-ups, class diagrams, brainstorms. Moreover, we have been in contact with our customers to understand and get feedbacks for current status of project. We have planned the project structure, prepared a project plan. Then we have come to the point of coding all three parts of project Android, Web, and Backend. We have divided into sub groups and tried to solve problems with other team members. We have also given feedbacks to each other in order to accomplish better results. At the end, we have documented what we have done until now, written unit tests, and prepared everything for customer.

Backend: We have chosen Django as main programming language for back-end developments since it is very simple and easy to learn. It is a high-level Python Web framework that encourages rapid development. We used PyCharm as our IDE, it is a developer-friendly platform and installation was very easy with our university accounts. We all have adapted to this new programming language easily and we immediately started development phase. We have used Postman for our endpoints and Docker as a container.

Lessons Learned: Creating a stable base part for a web application is very important for future developments. We have created "How To Run?" pages for answering possible questions, by this way we achieved fast development. We used Postman but since there are lots of backend developers, it was hard to maintain the last updated version. We could have considered using Swagger. Also we have learned that changing Database in a production environment must be done with a great attention.

Frontend: Since we decided to implement our front-end application with ReactJS, we need to an IDE that will make this ReactJS implementation process easier. We search for different IDEs and finally decided to use WebStorm which is one of the JetBrains tools which can be used as free with student license. The code completion of the WebStorm is ahead of any other IDE. It has blazingly fast debugger. It has built-in support for ESLint, we can fix ESLint problems with automatically with using it. A detailed explanation of the tools used can be found in section 9 of the report.

Lessons Learned: We have learned very well how important it is to manage a project as a group and to act synchronously, to comment on our comments and commit commitments even with small changes. In addition, the Frontend team has experienced that designing a beautiful interface for users is not as easy as it sounds. Although we followed the project plan, we had some difficulties in cultivating some features but we managed to overcome them by working with team spirit.

Android: We have used Android Studio as our IDE in our part since it has clear IDE and installed packages can be inferred from its gradle files so that other users should only need to build the gradles for requirements. We have also used Kotlin as our programming language. Best thing about Kotlin is that its readability is so simple and Java codes can be converted to Kotlin in the Android Studio automatically. Moreover, we have used Retrofit which is a library for Android to make HTTP requests as a REST Client. By this way compiler will validate all the request bodies while compiling and it will reduce errors caused by API queries. Overall, it was a good experience to learn Android and develop program.

Lessons Learned: From our point, we have learnt a lot of tools and frameworks in our project. More importantly, we have learnt advantages and disadvantages of working as a team. What should we do and What should not we do in a team cooperation task. Project management is not a write and run process, but something that requires deeper and detailed planning and formal and solid steps. At the end of the project, we are all happy to deliver Khajiit.

2 List and Status of Deliverables

Delivered	Name	Delivery Date
✓	Updated Communication Plan	Oct 1, 19
✓	Revised Requirements	Oct 1, 19
✓	Revised User Stories and Mockups	Oct 1, 19
✓	Updated Project Plan	Oct 18, 19
✓	User Manual	Jan 2, 20
✓	System Manual	Jan 2, 20
Backend		
✓	User Sign Up and Login Endpoint	Oct 15, 19
✓	User Profile Endpoint	Oct 22, 19
✓	User Profile Update Endpoint	Oct 22, 19
✓	User Follow Endpoint	Oct 22, 19
✓	User Search By Username Endpoint	Oct 22, 19
✓	User Password Update Endpoint	Oct 22, 19
✓	Trading Equipment Endpoints	Nov 15, 19
✓	Article Endpoints	Nov 15, 19
✓	Article Comment Endpoints	Nov 22, 19
✓	Wallet Endpoints	Nov 22, 19
✓	Event Endpoints	Nov 28, 19
✓	Article Like Endpoints	Nov 28, 19
✓	User Buy/Sell Operation Endpoints	Dec 24, 19
✓	User Prediction Endpoints	Dec 24, 19
✓	Notification For Followers of Portfolio Endpoints	Dec 24, 19
✓	Notification For Currency Endpoints	Dec 24, 19
✓	Portfolio Follow Endpoints	Dec 24, 19
✓	Search Event / Search Article Endpoints	Dec 24, 19
✓	Annotation Server and Annotation Endpoints	Dec 24, 19
✓	API Documentation	Jan 2, 20

Frontend		
✓	Homepage	Oct 15, 19
✓	User Sign Up and Login View	Oct 15, 19
✓	User Profile View	Oct 22, 19
✓	User Edit Profile View	Nov 25, 19
✓	Article Views (Create, List)	Nov 25, 19
✓	User Wallet View	Nov 25, 19
✓	User Portfolio View	Nov 25, 19
✓	Market Section View	Nov 25, 19
✓	Trading Equipment List View	Dec 24, 19
✓	Trading Equipment Buy/Sell View	Dec 24, 19
✓	User Wallet - Add Deposit View	Dec 24, 19
✓	Event List View	Dec 24, 19
✓	Text Annotation For Article	Dec 24, 19
✓	Image Annotation For Article	Dec 24, 19
Android		
✓	User Sign Up and Login View	Oct 15, 19
✓	User Profile View	Oct 22, 19
✓	User Profile Update View	Oct 22, 19
✓	User Follow Button	Oct 22, 19
✓	User Search By Username View	Oct 22, 19
✓	User Password Update View	Oct 22, 19
✓	Follow List View	Oct 22, 19
✓	Article Views (Create, List)	Nov 25, 19
✓	User Upgrade/Downgrade Views	Nov 25, 19
✓	User Private Profile View	Nov 25, 19
✓	Trading Equipment Buy/Sell View	Nov 25, 19
✓	User Wallet View	Nov 25, 19
✓	Event List View	Dec 24, 19
✓	Notification View	Dec 24, 19
✓	Search Event / Search Article Views	Dec 24, 19
✓	Approve / Reject Follow Request View	Dec 24, 19
✓	Text Annotation For Article	Dec 24, 19
✓	Article Like-Dislike View	Jan 2, 20
✓	Article Comment View	Jan 2, 20
✓	.apk File	Jan 2, 20

3 Summary of Team Member's Work

3.1 Backend

Team Members	Contributions
Ömer Faruk Toptas	<p>First of all I have joined every weekly meetings since the second milestone and also contributed in decision making part in each weekly meeting. I wrote several weekly meeting notes and added them to the wiki page. I also implemented the semantic search for an event and article features. I tried my best to help my group member's requests about the features. I regularly updated the features I have implemented according to feedbacks given by the other group members. Finally for the milestone report I have written some of the API documentation parts and annotation API documentation part.</p>
Abdullah Coşkun	<p>I attended all of the meetings both as whole group and as backend team. Since the second milestone I have completed all scheduled job with docker and with that create historical database. I have implemented notification system. Also I have developed buy order , sell order , set notification if some equipment goes higher some threshold or decreased to some threshold. I have make small changes all part of the code to ease front end and android teams. Also I have developed backend of annotation server. I have developed prediction system and portfolio operations. I also improved wallet operation. At last I wrote part of the API documentation System manual of both backend and backend of annotation. I also developed send followers a notification mechanism when a portfolio has updated or deleted.</p>
İlker Özkan	

Team Members	Contributions
İrem Üstünboyacıoğlu	<p>In general, I took part in the decision-making phase of our group meetings and I participated in weekly actions. I prepared Project Plan by using ProjectLibre tool according to our draft plan that we discussed in our meetings. During the semester, I have updated the Project Plan according to the feedback that we received from our customer regularly. As a member of backend team, I firstly learned Django and I installed PyCharm, Postman and Docker. Then, I implemented password security requirements which are 2.5.1 and 2.5.2. I also implemented the requirement that a password need to have at least 8 characters. I regularly updated issues, reviewed pull-requests. After Milestone 1, I have implemented Article Endpoints, Comment Endpoints and Like Endpoints. I have updated Postman link for endpoints and shared it with other teams. I have updated password security requirements according to our implementation in Milestone 1. After Milestone 2, I have implemented portfolio-follow requirement, I have attended the meeting where we discussed Annotation Structure. I also fixed requests and responses of endpoints according to meetings with Frontend and Android teams. Finally, I added the Updated Project Plan, List and Status of Deliverables to the Final Milestone Report.</p>

3.2 Frontend

Team Members	Contributions
Mete Han Kurt	I participated in all the group meetings and took notes and edited deadline times every week on the wiki page. In addition, as the Frontend team, I attended meetings between us to solve some problems. I added a lot of features to the Profile page in the application, I designed the Wallet page and provided money to add. On the Equipment page, I created a chart where users can see the last values of their currencies. I have created a page that will show the search results in detail. I showed prediction rates in each user's profile and added prediction sections on the equipment. In addition, in the final milestone report, I wrote the Evaluation of Tools and Managing the Project in Frontend, and revised the requirements.
Mehmet Altay İnce	I tried to attend most of the general meetings and front-end meetings. After the first milestone we had a job sharing. My parts were : implementing general and individual article page,comments on articles, market page in which the user can follow/unfollow equipments, user profiles page subsections and search bar's auto completion functionality. Events page added, sizes of components fixed for better QoE. Other than that I have revised merge requests and help other friends if needed.

Team Members	Contributions
Ufuk Yilmaz	<p>From the start of the semester, I have attended all of the front-end meetings, group meetings, and annotation meetings. I played active role on creating the project structure of front-end application. From the second milestone, I implemented the pages of Market Page, Events Page, Equipment Page, and Wallet Page with other teammates in the front-end team. I implemented the logic that enables the trader user make transaction on the web application. I have written service for taking trading equipment values before the second milestone and now I connect these values to the market-page and also equipment page. I also implemented text and image annotation features for articles that users can annotate on the text or image that they have selected. While implementing these logic, I sent feedback to back-end members. e.g. When they want to change the back-end, I inform them about these changes can damage to our front-end application so that we solved the problems before occur by staying in constant communication. I deployed the front-end application to the Amazon Web Server. In the final milestone report, I filled the contents of the sections that a part of the web application section of the user manual, and the part of the annotation model section and the front-end section of system manual.</p>

3.3 Android

Team Members	Contributions
Emirhan Sarac	I have attended all group and android meetings as a communicator. I have taken an active role in coding, design, and workload share in android part. After the Milestone 2, I have implemented Search Article and Search Event part. I have updated editing page so that users can update their profile picture. Then I have implemented rendezvous acceptance system for private user follows. I have created Wallet Page so that user can see his all equipment values and their current worth. I have implemented Portfolio Page and its functionalities including follow portfolio, edit-delete portfolio, and create portfolio. I have also completed design parts of the previously mentioned parts. I have also worked on bug and fix problems such as registering as trader direct instead of upgrading from basic crashes program since wallet is not created automatically. I have played active role while communicating with Web-Backend and in my team in order to complete required parts appropriately. I have helped Ceren on the annotation part and it was not much, only in the pop-up menu. I have also updated requirements according to current status of the proejct. I have completed Project Assessment, Design Documentation, Requirements, and User Manual parts of the Final Milestone.
Eray Kurtulus	I have attended the meetings and contributed to the planning of the project. After the second milestone, I have implemented the notifications functionality, apply/reject functionality for follow requests and setting buy/sell orders for equipments on the Android app. I have also created necessary data models and layouts for these functionalities. Also, I have updated some of the previous functionalities which had become broken after changes to the backend API.

Team Members	Contributions
Yağmur Ceren Dardağan	I have attended all of the general meetings, android meetings and annotation meetings. For the semester, I only missed one general meeting. After the Milestone 2, I implemented events fragment that lists all the events and it can be reached via bottom navigation bar. And also, I implemented a detailed event page to learn further information about that event. Furthermore, I added article image feature that can be selected from photos of the user on the phone and display feature if the articles on the list article pages and displaying an article. I also added like/dislike and comment feature for a particular article for users of the application. I also implemented displaying all the comments made by the other users. In addition to article page, I also added prediction feature of all trading equipment for logged in user. For guest users, I arranged all the fragments that guest user can see, I arrange all article, event and equipment page in a way that guest user only see title,content, like/dislike counts and comments of an article and it cannot like/dislike or comment. In a similar way, I rearranged trading equipment page to see all details but cannot buy/sell/predict or buy/sell order. Addition to these features, I implemented text annotation feature for articles in a way that all guest/logged in user can see the text annotation made by other users but only logged in users can tag/comment on the text that they have chosen. Finally, I set up all the image annotation end points and connections with backend however I couldn't handle the selecting part of image annotation and displaying annotations. For the milestone report, I help to write annotation section of the report and the system manual of the android part. I also created users, articles and portfolios for the milestone. I also arranged the usb that contains executables of backend,frontend and android.

4 Annotation Implementation & W3C Standard Compliance

Annotating, the act of creating connections between various pieces of information, is in many ways a prevalent online activity. W3C Data Annotation Model is a open annotation model that standardize the way of annotating of the web document. It provides an interoperable framework for annotation such that they can easily be shared between platforms, and it is easy to implement. We introduced our application's annotation framework with the respect to specifications of W3C Web Annotation Data Model.

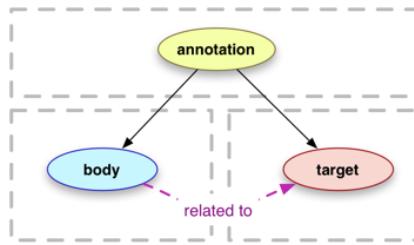


Figure 1: Intro Model

The Web Annotation Data Model is defined using the following basic principles:

- An Annotation is a rooted, directed graph that represents a relationship between resources.
- There are two primary types of resource that participate in this relationship, Bodies and Targets.
- Annotations have 0 or more Bodies.
- Annotations have 1 or more Targets.
- The content of the Body resources is related to, and typically "about", the content of the Target resources.
- Annotations, Bodies and Targets may have their own properties and relationships, typically including creation and descriptive information.

- The intent behind the creation of an Annotation or the inclusion of a particular Body or Target is an important property and represented by a Motivation resource.

4.1 Model

Model

Term	Type	Description
@context	Property	The context that determines the meaning of the JSON as an Annotation. The Annotation must have 1 or more @context values and http://www.w3.org/ns/anno.jsonld must be one of them. If there is only one value, then it must be provided as a string.
id	Property	The identity of the Annotation. An Annotation must have exactly 1 IRI that identifies it.
type	Relationship	The type of the Annotation. An Annotation must have 1 or more types, and the Annotation class must be one of them.
Annotation	Class	The class for Web Annotations. The Annotation class must be associated with an Annotation using type .
body	Relationship	The relationship between an Annotation and its Body. There should be 1 or more body relationships associated with an Annotation but there may be 0.
target	Relationship	The relationship between an Annotation and its Target. There must be 1 or more target relationships associated with an Annotation.

Figure 2: Annotation Model

4.2 Annotation on Text

In our application, we add annotation mechanism on article texts. Since the, best way to refer is using annotations.

```

1   {
2     "1" : {

```

```

3     "id": "http://khajiittraders.tk/annotation_id76
4         ,
5     "creator": {
6         "id": "http://www.khajiittraders.tk/user/38
7             ,
8         "type": "Person",
9         "name": "Jessica Morgan",
10        "nickname": "jessicamorgan"
11    },
12    "body": [
13        {
14            "id": 133,
15            "type": "TextualBody",
16            "purpose": "tagging",
17            "value": "General Motors"
18        }
19    ],
20    "target": {
21        "id": 142,
22        "selector": {
23            "id": 142,
24            "refinedBy": {
25                "id": 142,
26                "type": "TextPositionSelector",
27                "start": 309,
28                "end": 311
29            },
30            "type": "FragmentSelector",
31            "value": "xpointer(/doc/body/section[2]/
32                para[1])"
33        },
34        "type": "Text",
35        "styleClass": "",
36        "source": "http://www.khajiittraders.tk/
            article/27/"
37    },
38    "type": "Annotation",
39    "motivation": "Commenting",

```

```
37     "created": "2020-01-01T19:54:20Z"
38 }
39 }
```

4.3 Annotation on Image

```
1 {
2   "0": {
3     "id": "http://khajittraders.tk/annotation_id72",
4     ,
5     "creator": {
6       "id": "http://www.khajittraders.tk/user/7/",
7       "type": "Person",
8       "name": "Ufuk Yilmaz",
9       "nickname": "ufuky"
10    },
11   "body": [
12     {
13       "id": 129,
14       "type": "TextualBody",
15       "purpose": "tagging",
16       "value": "Benjamin Franklin"
17     }
18   ],
19   "target": {
20     "id": 138,
21     "type": "Image",
22     "styleClass": "mystyle",
23     "source": "http://www.khajittraders.tk/
24       article/27/",
25     "image_id": "http://www.khajittraders.tk/
26       article/27/image1#xywh=52,65,27,33"
27   },
28   "type": "Annotation",
29   "motivation": "Referral",
30   "created": "2020-01-01T19:24:00Z"
31 }
```

4.4 Annotation Implementation on Android

To implement W3C Standard on android, we first need to handle Url mappings of article pages to handle displaying annotations from the web application of Khaji-it and create new text/image annotations compatible with the W3C Standard and web application respectively. To handle Url mappings, we add new URL mapping to the AndroidManifest.xml with URL mappings editor. The mapping settings can be seen in Figure below.

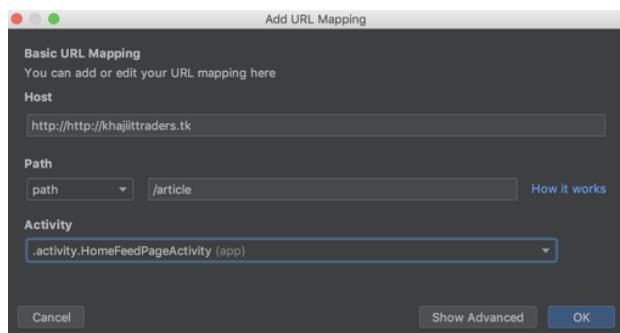


Figure 3: Android Url Mapping

After handling Url mappings, we have created new client for Annotation Server that Backend team provided for annotation via different port 8020.

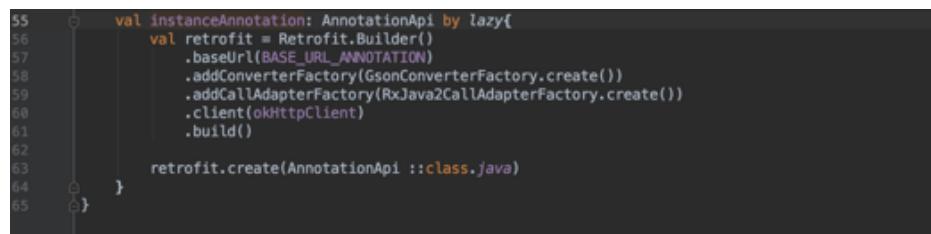
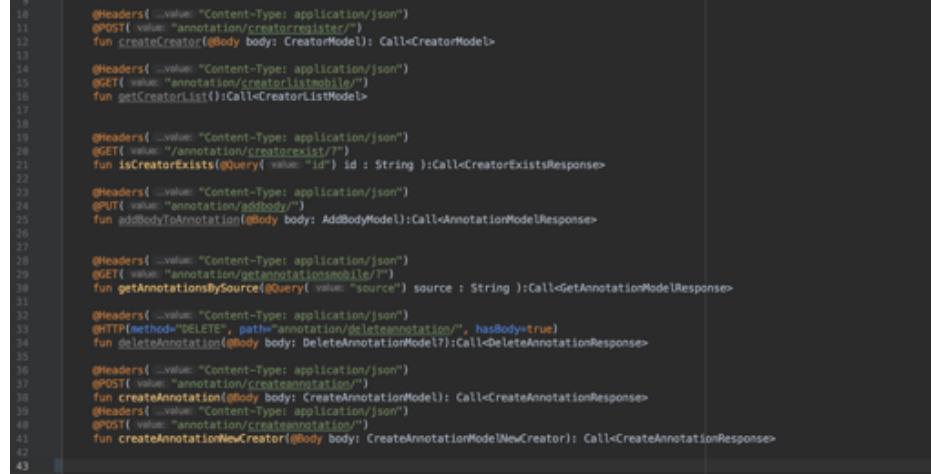


Figure 4: New Client For Annotation Server

To implement all the annotation features, we handled all the requests and

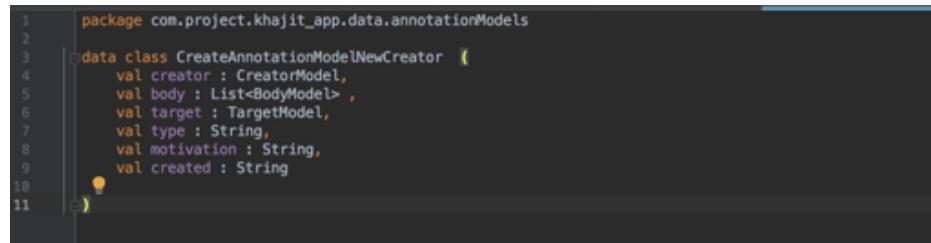
responses can be done via Annotation API. The annotation requests can be seen in figure below.



```
1 @Headers( ...value: "Content-Type: application/json")
2 @POST( value: "annotation/creatorregister/")
3 fun createCreator(@Body body: CreatorModel): Call<CreatorModel>
4
5 @Headers( ...value: "Content-Type: application/json")
6 @GET( value: "annotation/creatorlistmobile/")
7 fun getCreatorList():Call<CreatorListModel>
8
9 @Headers( ...value: "Content-Type: application/json")
10 @GET( value: "/annotation/creatorexist/")
11 fun isCreatorExists(@Query( value: "id") id : String ):Call<CreatorExistsResponse>
12
13 @Headers( ...value: "Content-Type: application/json")
14 @PUT( value: "annotation/addbody/")
15 fun addBodyToAnnotation(@Body body: AddBodyModel):Call<AnnotationModelResponse>
16
17 @Headers( ...value: "Content-Type: application/json")
18 @GET( value: "annotation/getannotationsmobile/")
19 fun getAnnotationsBySource(@Query( value: "source") source : String ):Call<GetAnnotationModelResponse>
20
21 @Headers( ...value: "Content-Type: application/json")
22 @HTTP(method="DELETE", path="annotation/deleteannotation/", hasBody=true)
23 fun deleteAnnotation(@Body body: DeleteAnnotationModel):Call<DeleteAnnotationResponse>
24
25 @Headers( ...value: "Content-Type: application/json")
26 @POST( value: "annotation/createannotation/")
27 fun createAnnotation(@Body body: CreateAnnotationModel): Call<CreateAnnotationResponse>
28 @Headers( ...value: "Content-Type: application/json")
29 @POST( value: "annotation/createannotation/")
30 fun createAnnotationNewCreator(@Body body: CreateAnnotationModelNewCreator): Call<CreateAnnotationResponse>
31
32
33
34
35
36
37
38
39
40
41
42
43
```

Figure 5: List Of Annotation Requests

To apply W3C standards we have created data models compatible with annotation bodies highlighted at the standart and handle all the annotation requests and responses with these data models. Sample data model for creating annotations can be seen in figures below.



```
1 package com.project.khajit_app.data.annotationModels
2
3 data class CreateAnnotationModelNewCreator {
4     val creator : CreatorModel,
5     val body : List<BodyModel>,
6     val target : TargetModel,
7     val type : String,
8     val motivation : String,
9     val created : String
10 }
```

Figure 6: Sample Annotation Request For Creating Text Annotation

```
1 package com.project.khajit_app.data.annotationModels
2
3 import android.provider.ContactsContract
4
5 data class CreatorModel(val id : String,
6                         val type : String?,
7                         val name : String?,
8                         val nickname: String?)
```

Figure 7: Data Model for Handling Creator

```
1 package com.project.khajit_app.data.annotationModels
2
3 data class BodyModel (val type : String?,
4                      val purpose : String?,
5                      var value : String?
6                     )
```

Figure 8: Data Model for Handling Bodies of Annotation

```
1 package com.project.khajit_app.data.annotationModels
2
3 data class TargetModel (val selector : SelectorModel,
4                        val type : String,
4                        val styleClass : String,
4                        val source : String,
4                        val image_id : String
4                     )
```

Figure 9: Data Model for Handling Targets of Annotation

5 Design Documents

5.1 Mockups

5.1.1 Android



Figure 10: Home Page

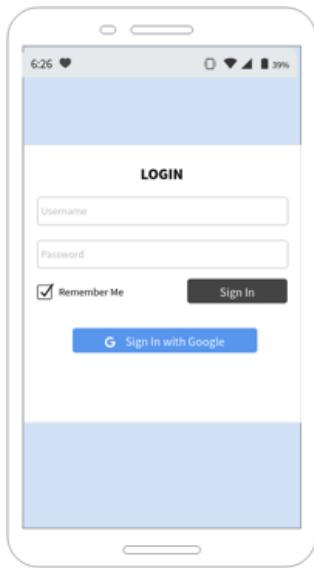


Figure 11: Login Page

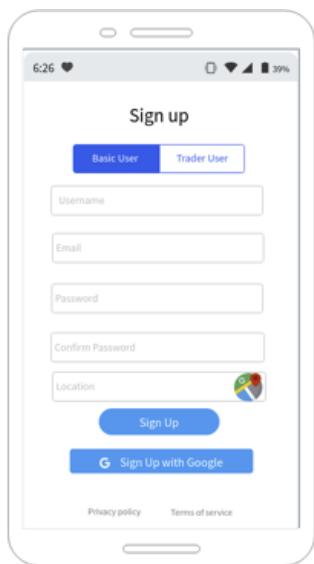


Figure 12: Sign Up Page

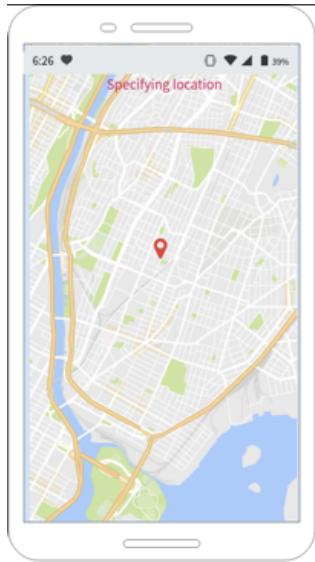


Figure 13: Google Map Page

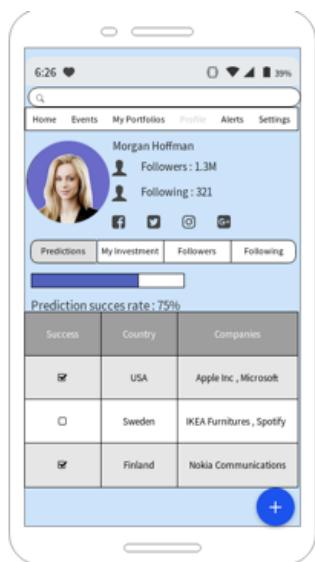


Figure 14: Prediction Page

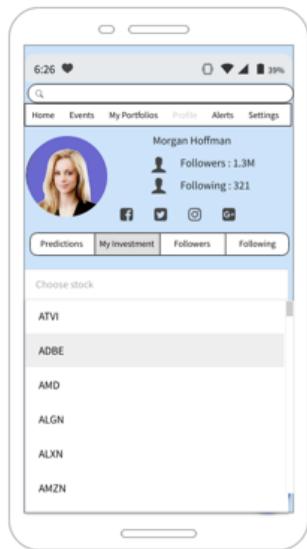


Figure 15: Investment Page



Figure 16: Event Page

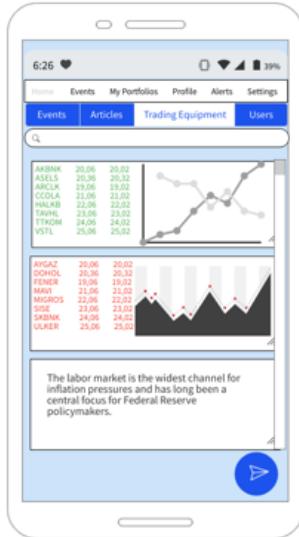


Figure 17: Equipment Page

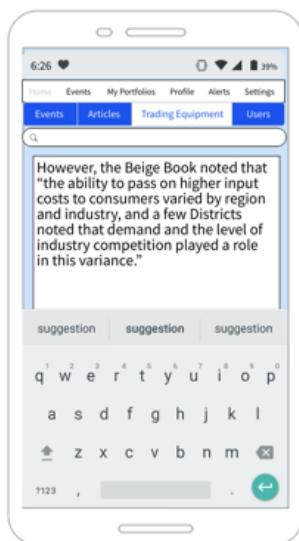


Figure 18: Comment Page

5.1.2 Web



Figure 19: Home Page

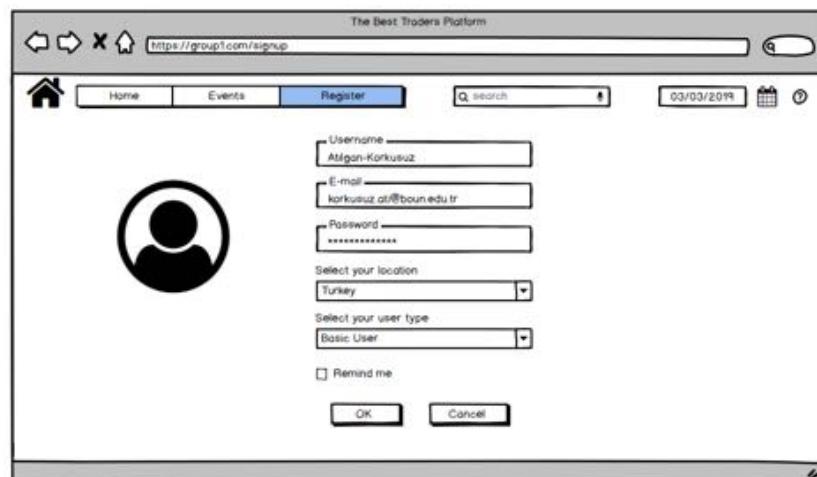


Figure 20: Sign Up Page



Figure 21: Equipment Page



Figure 22: Equipment Page

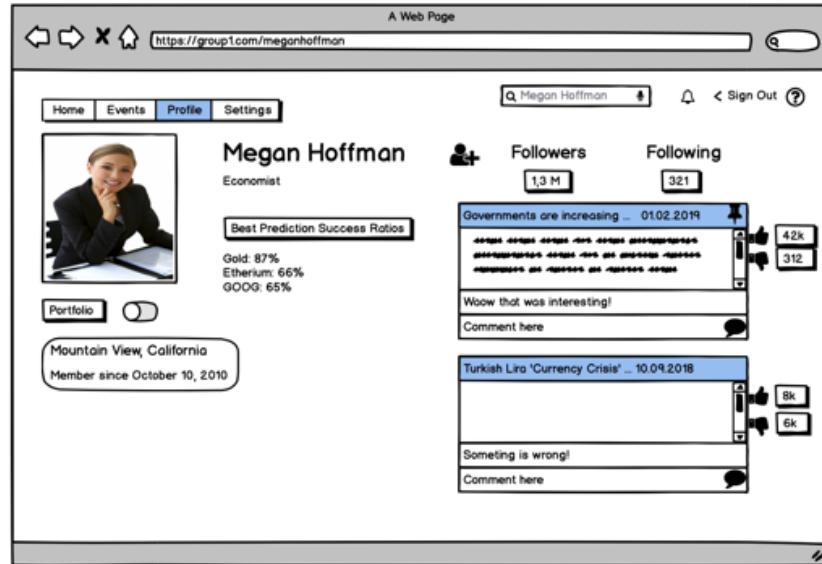


Figure 23: Article Page



Figure 24: Search Page

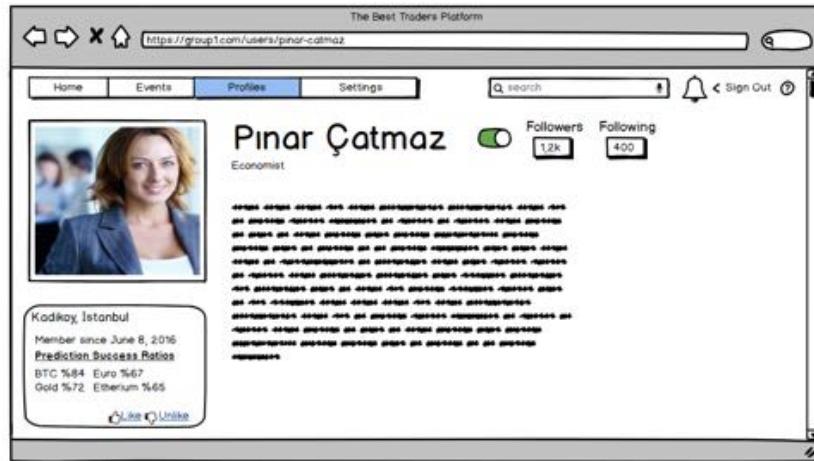


Figure 25: User Page

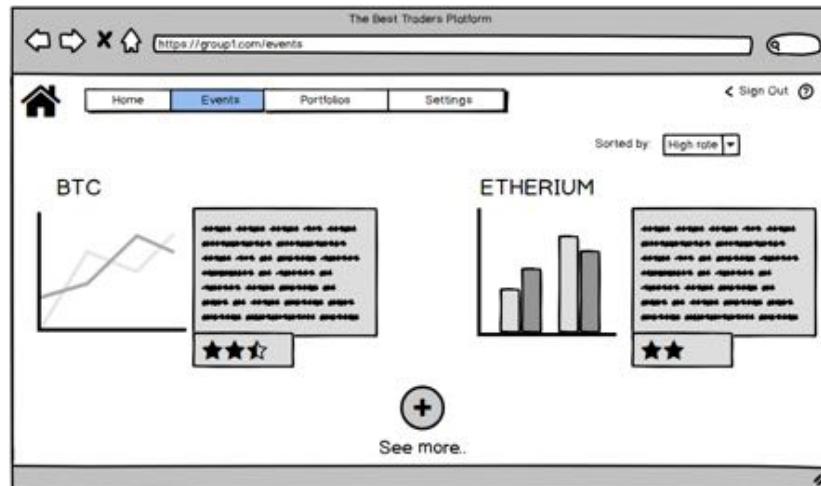


Figure 26: Event Page

5.2 Software Design

5.2.1 Use Case Diagram

<https://github.com/bounswe/bounswe2019group1/wiki/Use-Case-Diagram>

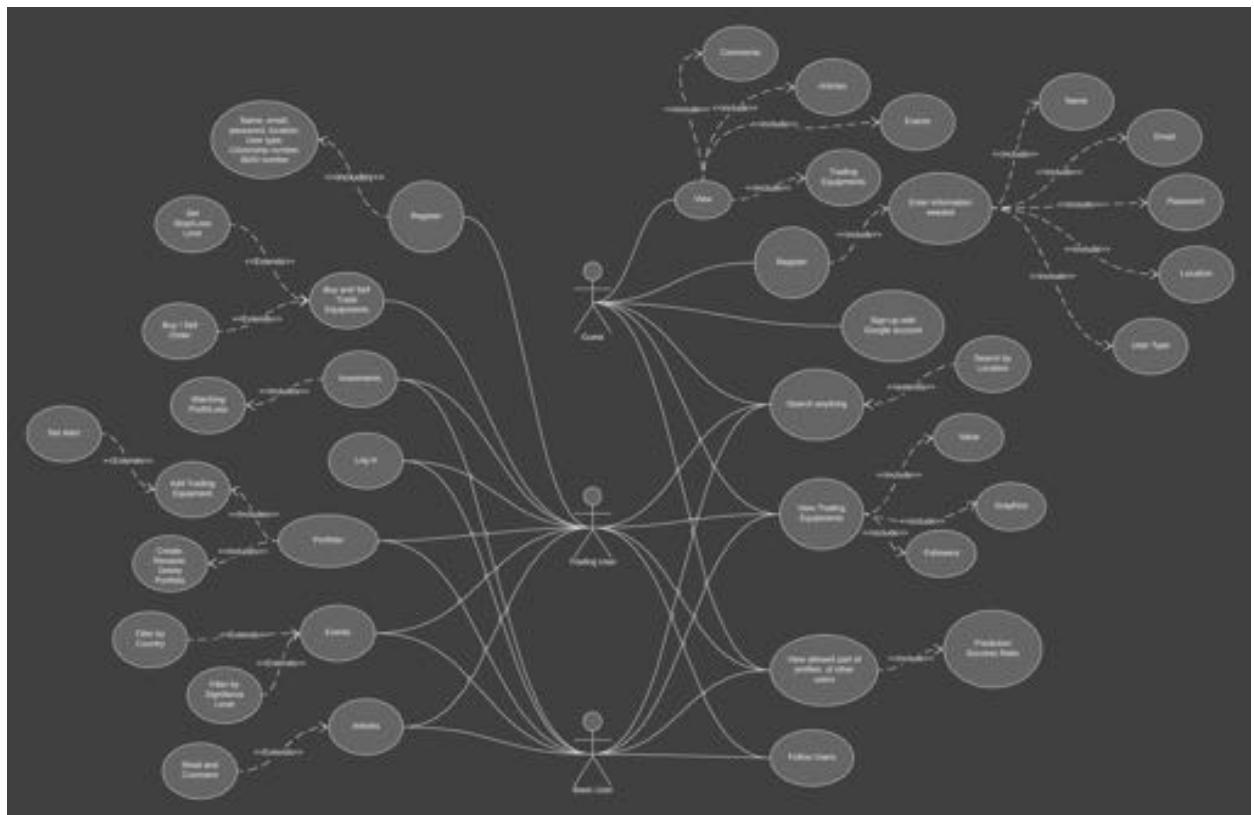


Figure 27: Use Case Diagram

5.2.2 Class Diagram

<https://github.com/bounswi/bounswi2019group1/wiki/Class-Diagram>

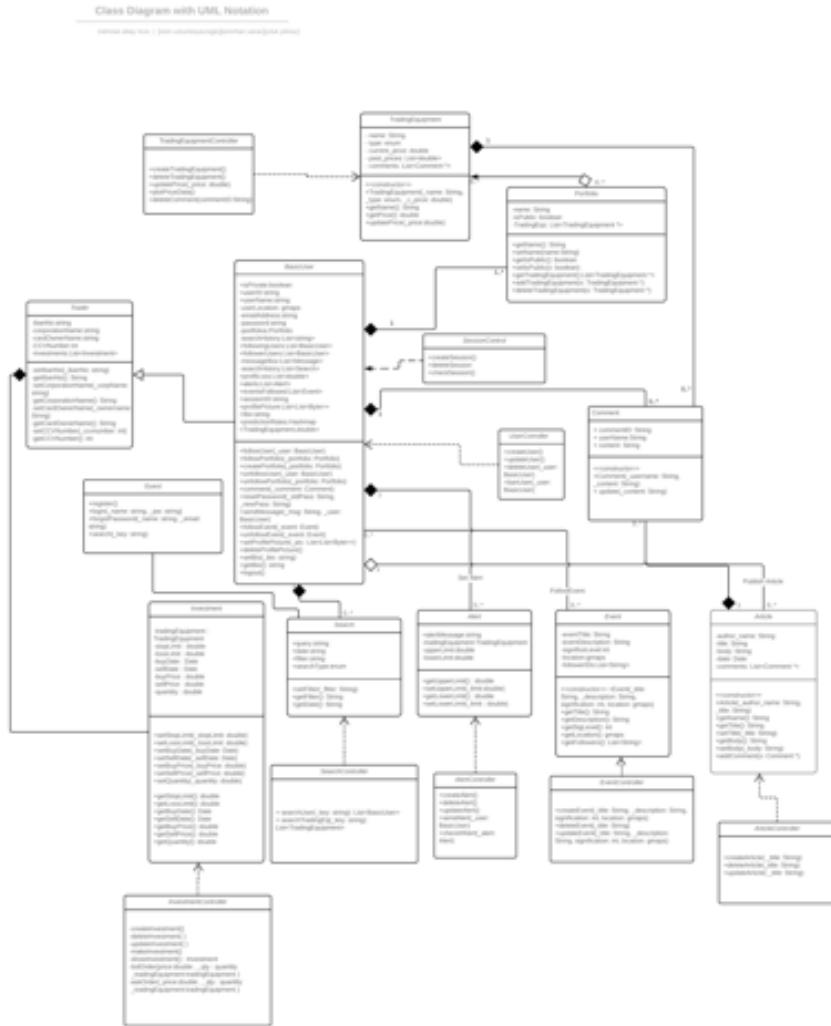


Figure 28: Class Diagram

5.2.3 Sequence Diagram

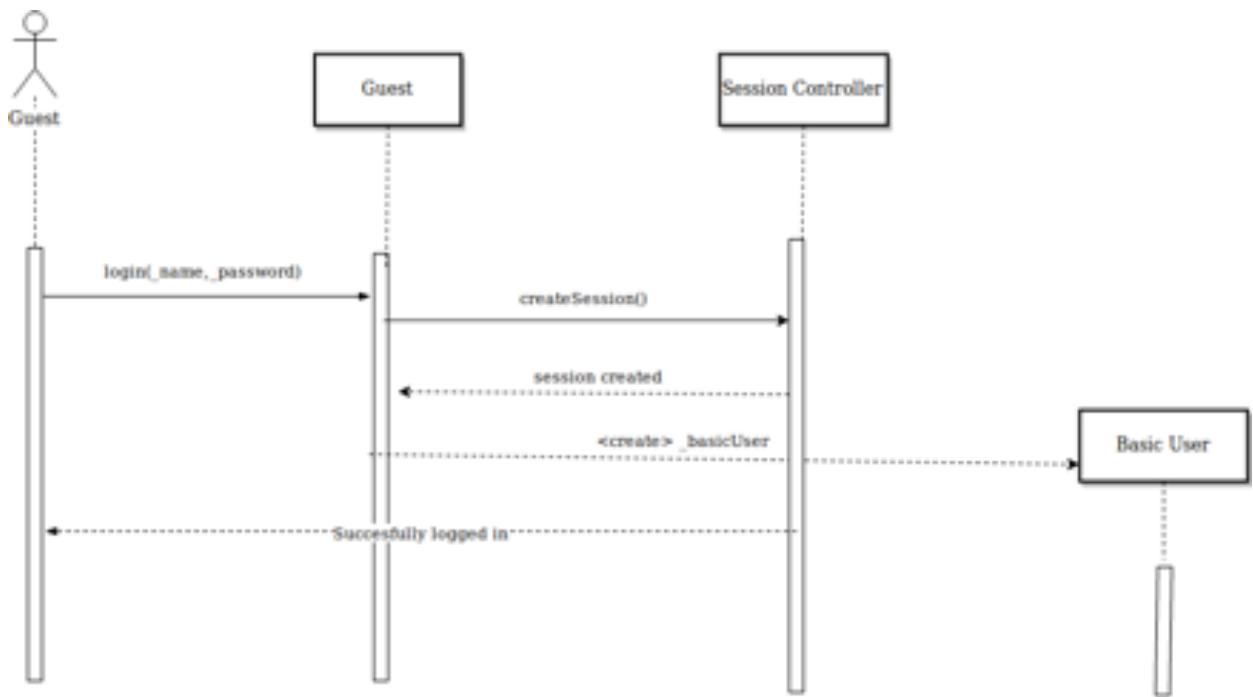


Figure 29: Login Sequence Diagram

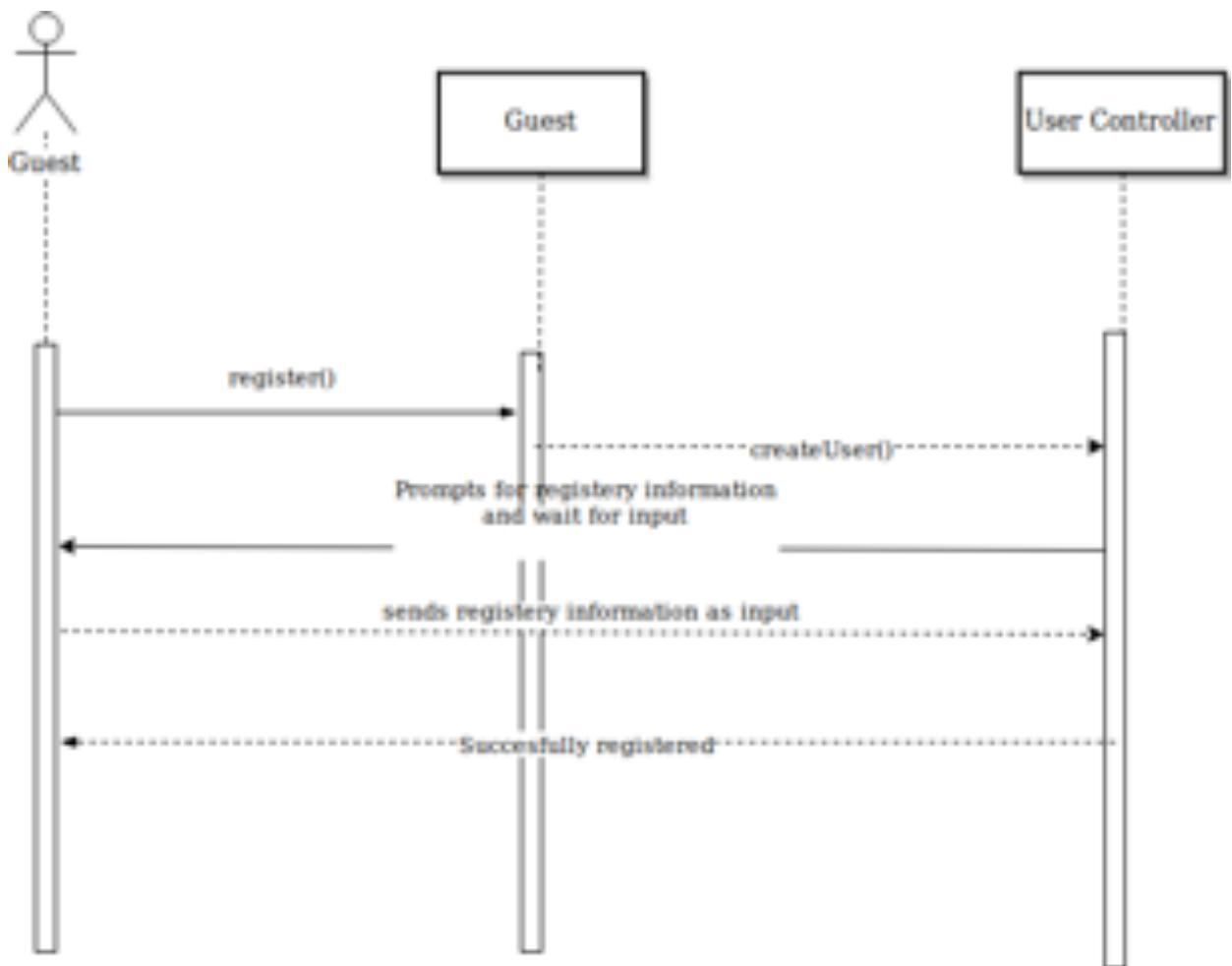


Figure 30: Register Sequence Diagram

User Search

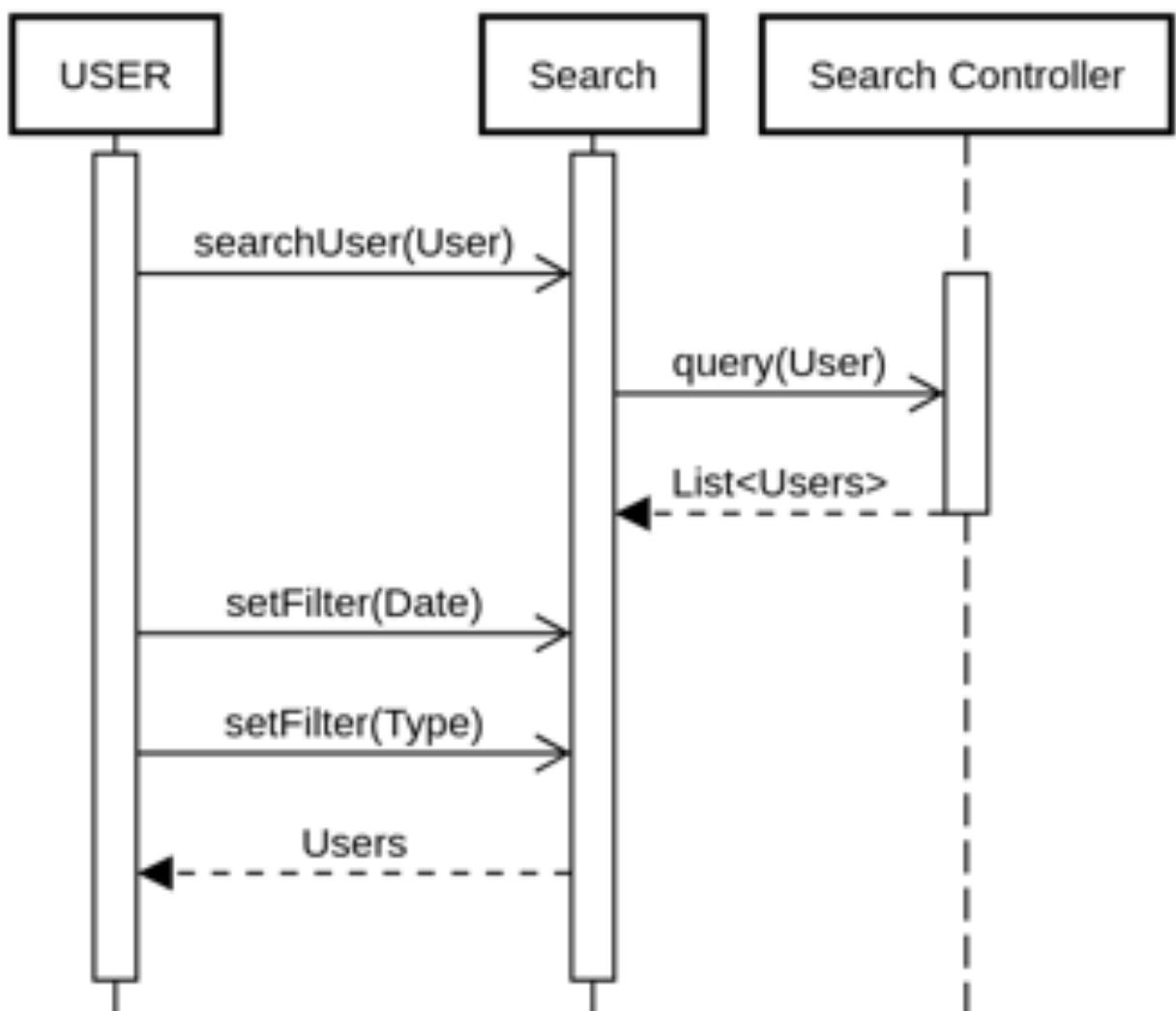


Figure 31: Search Sequence Diagram

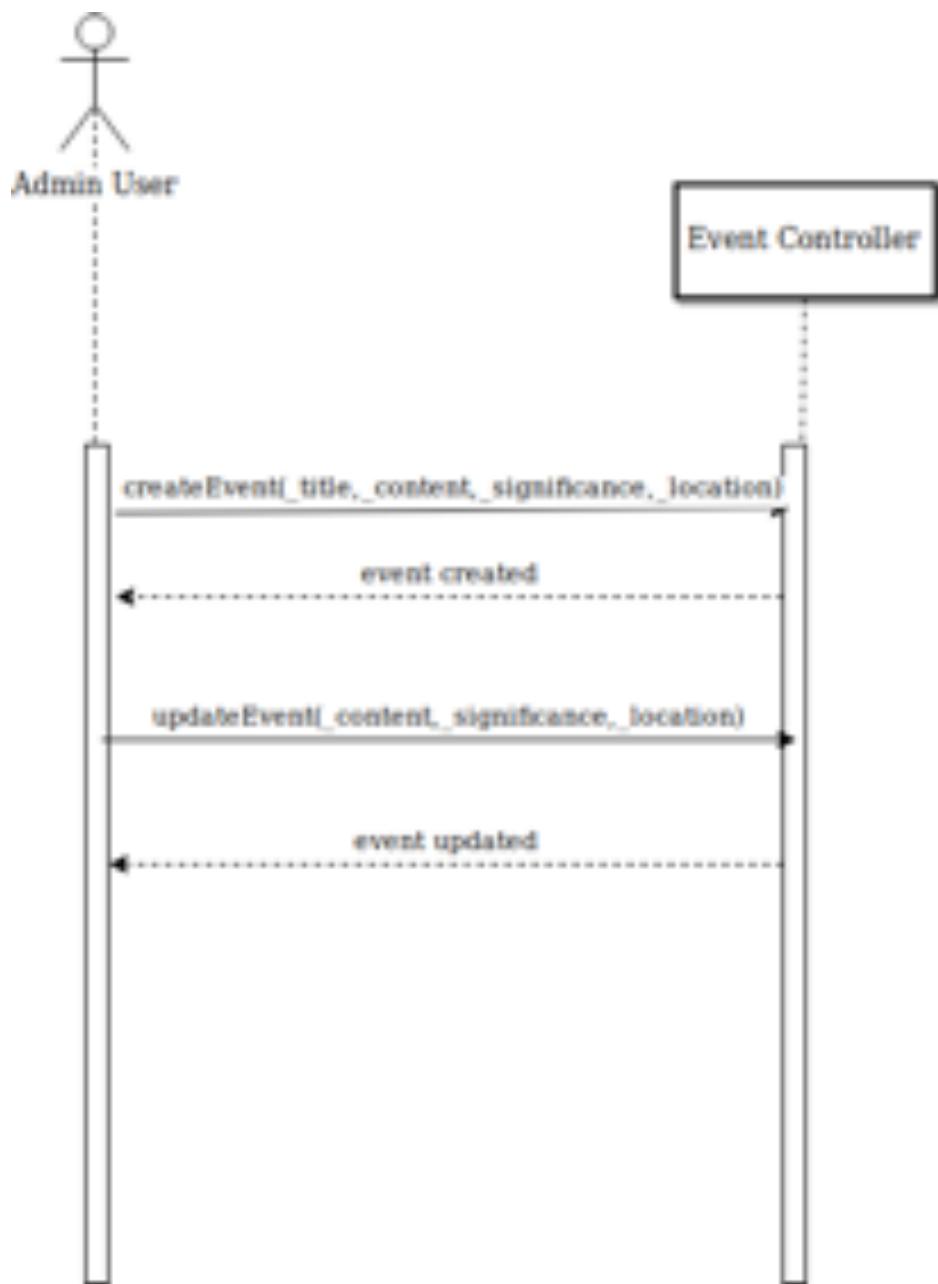


Figure 32: Event Sequence Diagram

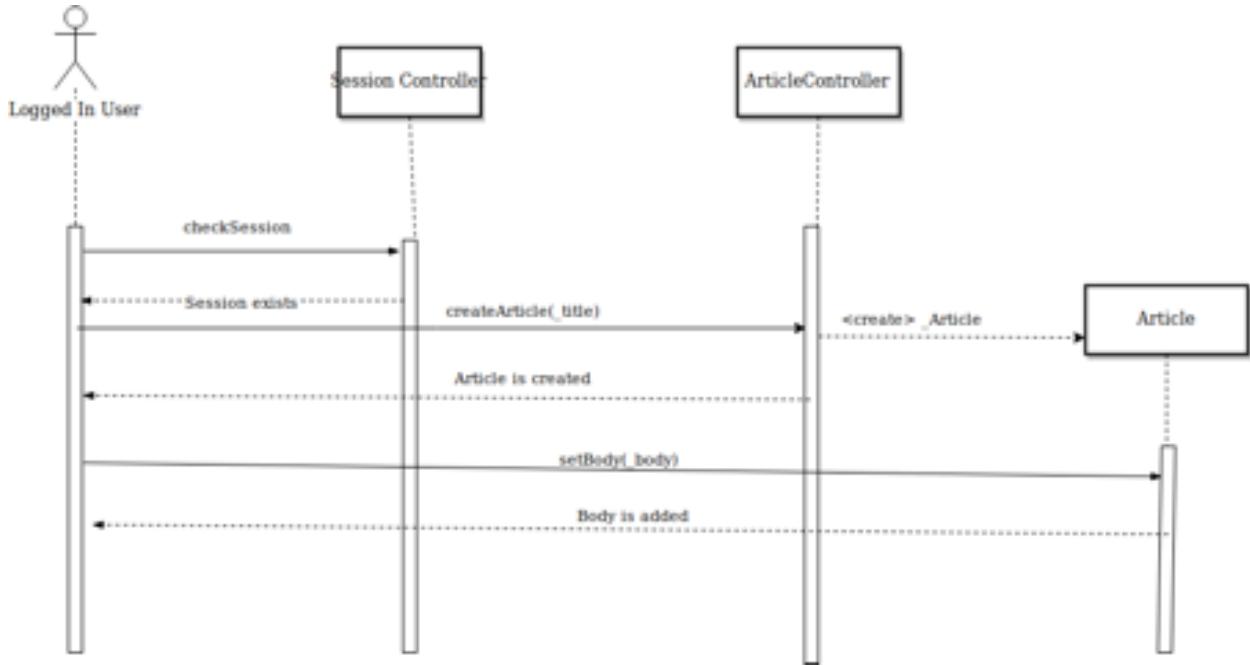


Figure 33: Publish Article Sequence Diagram

Make Investment

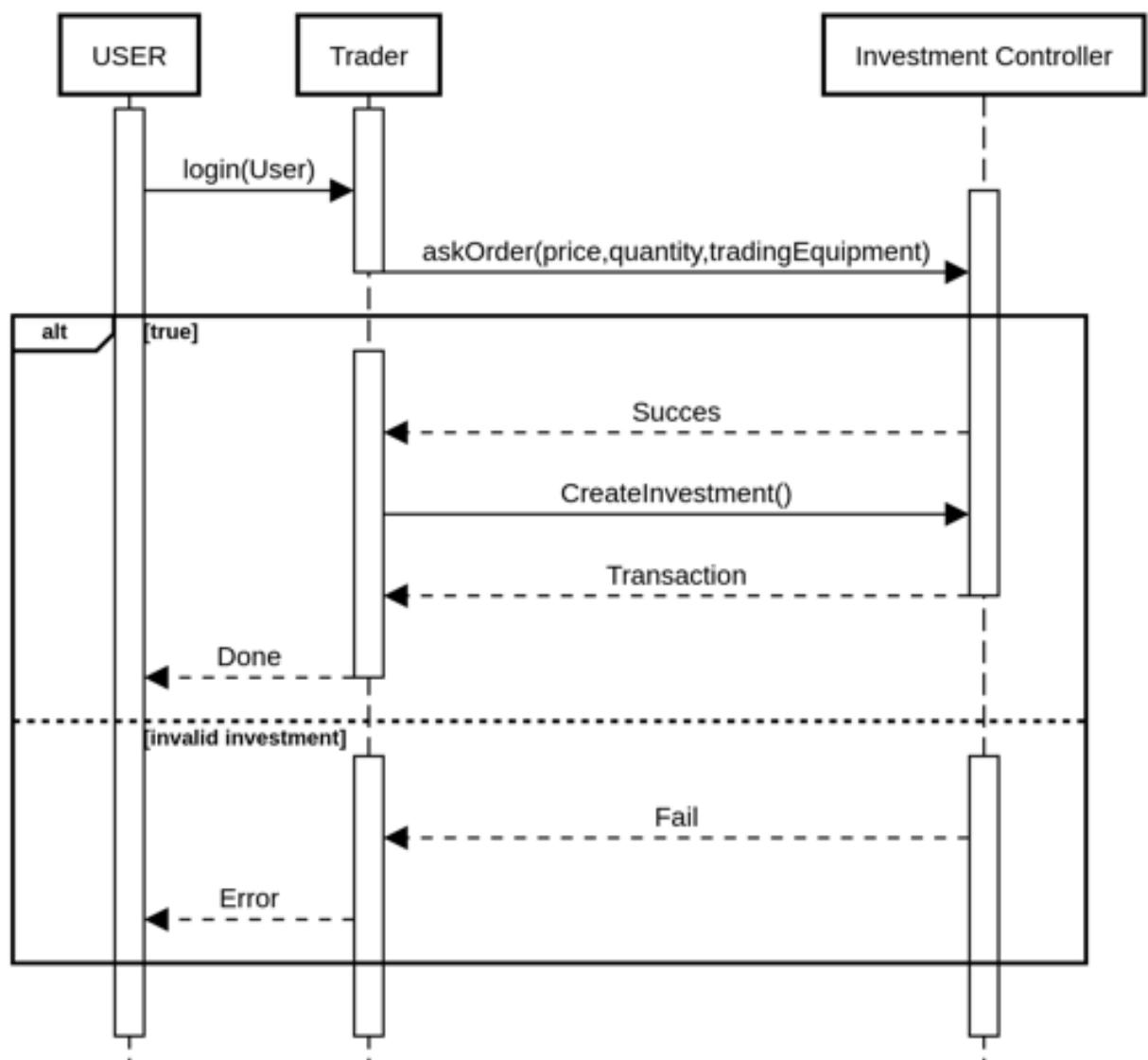


Figure 34: Investment Sequence Diagram

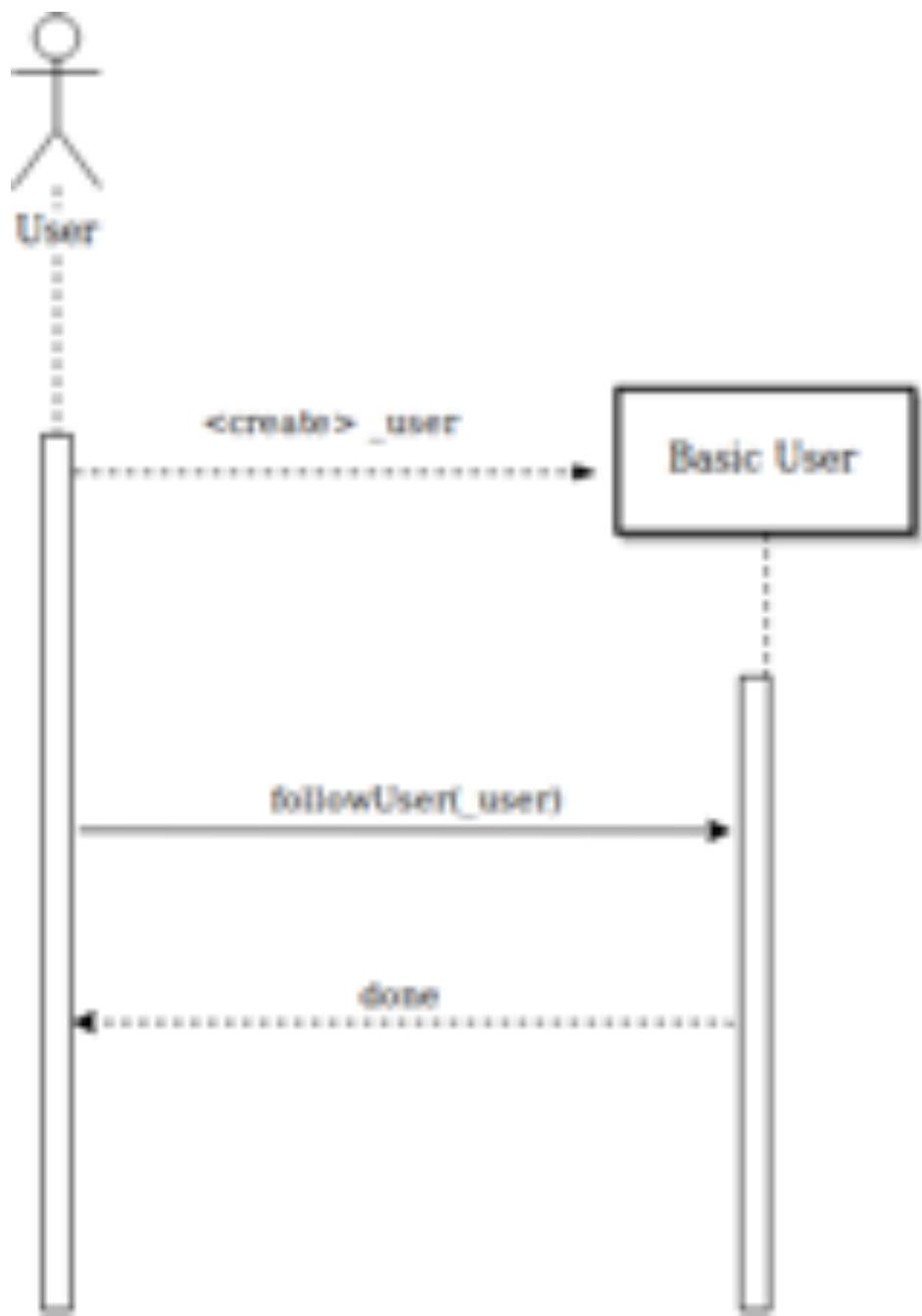


Figure 35: Follow User Sequence Diagram

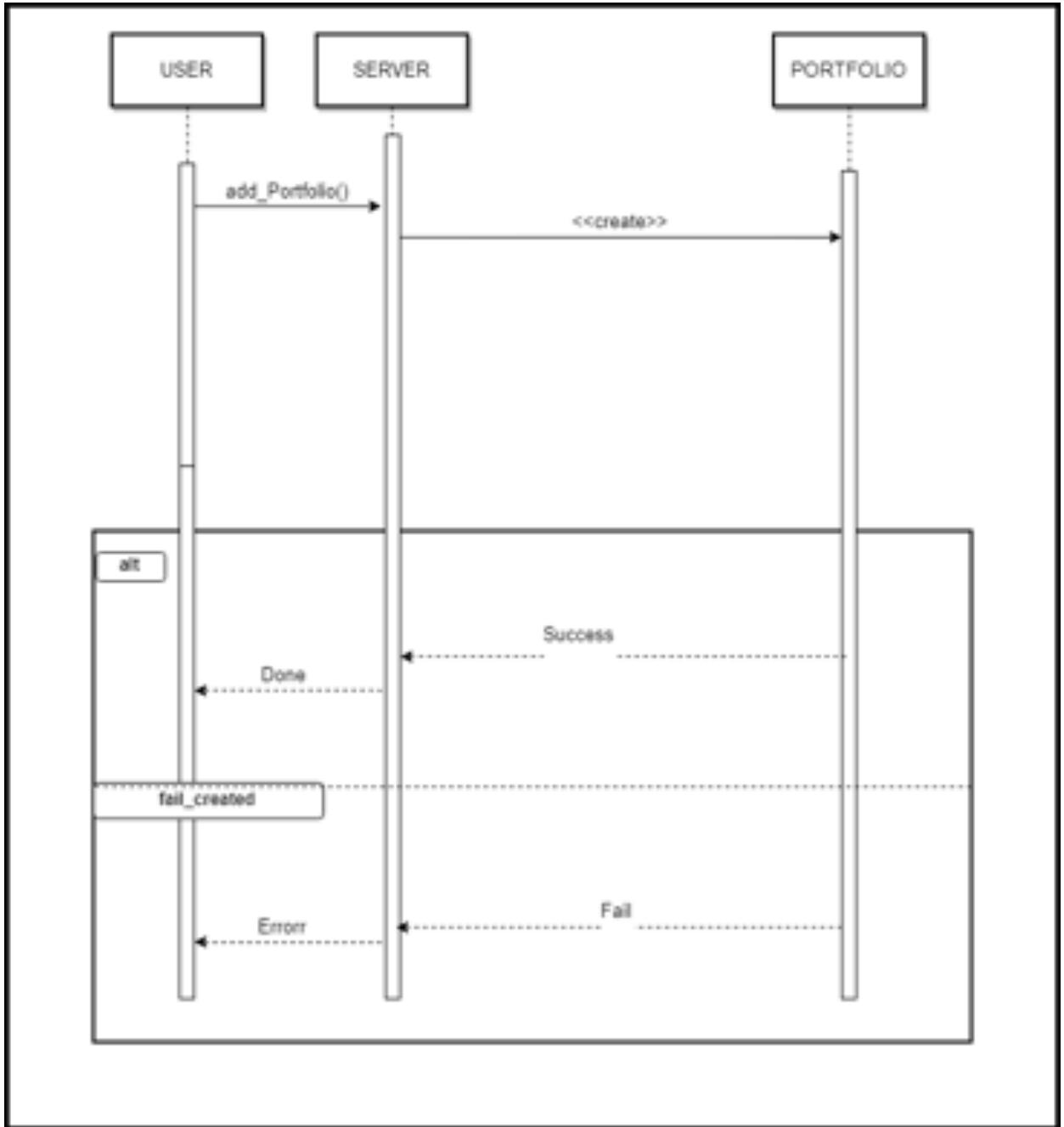


Figure 36: Portfolio Sequence Diagram

6 Communication Plan

Audience	Purpose	Delivery Frequency	Place	Delivery Method	Communicator
All team members	Discussion about instant issues	All the time	Online	Slack, GitHub, WhatsApp, Doodle	None
All team members	Evaluation and weekly plan	Every Tuesday 17.00 - 18.00	BM Building	Face-to-face	Emirhan Saraç
Available team members	Performing the planned job together	Every Tuesday 18.00 - 19.00	BM Building	Face-to-face	Emirhan Saraç
All team members	In case Tuesday meeting is cancelled	Every Wednesday 17.00 - 18.00	BM Building	Face-to-face	Emirhan Saraç
All team members	Feedback about progress of tasks individually	Every Friday night	Online	Through WhatsApp	Emirhan Saraç
Available members	Help and complete tasks that are not going to complete on-time	Every Saturday 12.00 - 18.00	BM Lounge	Face-to-face	None

Figure 37: Communication Plan

7 Requirements

Glossary

- **Annotation:** A note by way of explanation or comment added to the text and article.
- **Article:** Documents for users of traders platform about trading that helping users to reach knowledge and share with other users.
- **Basic user:** Registered user authorized to write comments and articles.
- **Comment:** Basic and traders users' ideas about relevant investment or the stock market.
- **Common User:** The user type which corresponds to both Basic and Trader.
- **Economic Event:** Important economic developments such as Fed Interest Rate, GDP of countries, ECB Interest Rate, and more.
- **Events:** News or happenings about stock, economy and trade.
- **Follow:** An act using for a user can reach others article, investments, comments and share their owns with others.
- **Guests:** Users who are only authorized to read reviews and view trading equipment.
- **Investment:** Buying and selling trading equipment(indices, stocks, ETFs, commodities, currencies, funds, bonds, and cryptocurrencies) in order to make a profit.
- **Log in:** When a user logs in to the system with the information they provided signing up.
- **Password:** String of characters used for user authentication to prove identity.
- **Prediction Success Ratio:** A scale that shows what percent of user's predictions were successful.

- **Profile Page:** The page providing information about prediction specified trading equipment of users on certain assets and success ratio on these predictions.
- **Portfolio:** A collection of trading equipments that are owned by the user.
- **Semantic search:** Semantic search seeks to improve search accuracy by understanding the users' intent and the contextual meaning of terms as they appear in the searchable dataspace, whether on the Web or within a closed system, to generate more relevant results.
- **Significance Level:** Marks for how much reliable and important events and news.
- **Tag:** A categorical designation that attaches to various items.
- **Trading user:** A person who can post, comment, like and invest in any trading equipment
- **Trading equipment:** Bills of exchange, commodities, crypto coins, stock market shares etc.
- **Username:** The name that identifies user account to the system.
- **1. Functional Requirements**
 - **1.1. User Requirements**
 - * **1.1.1. Common User Requirements**
 - **1.1.1.1. Sign Up**
 - **1.1.1.1.1.** Users shall sign up by following below steps.
 - **1.1.1.1.1.1.** Users shall sign up by providing their username, password, name, surname, e-mail address, and location.
 - **1.1.1.1.1.1.1.** Usernames and e-mail addresses shall be unique.
 - **1.1.1.1.1.1.2.** User passwords shall be in the specified form.
 - **1.1.1.1.1.1.3.** User location shall be specified via Google Maps.

- **1.1.1.1.2.** Users shall sign up via their Google account.
- **1.1.1.2. Sign In**
 - **1.1.1.2.1.** Users shall sign in by providing their e-mail and password.
 - **1.1.1.2.2.** Users shall sign in via their Google account.
- **1.1.1.3. Password Change**
 - **1.1.1.3.1.** Reset Password
 - **1.1.1.3.1.1.** Users shall be able to reset their password by requesting “Reset Password” operation at profile page.
 - **1.1.1.3.1.2.** Users should provide old password and new password they desire while logged in.
- **1.1.1.4. Profile**
 - **1.1.1.4.1.** Users shall have a profile either private or public.
 - **1.1.1.4.1.1.** Users with a private profile shall be followed in order to see the content of the private user.
 - **1.1.1.4.2.** Users shall have at least one **portfolio**.
 - **1.1.1.4.2.1.** Users shall be able to rename their portfolio.
 - **1.1.1.4.2.2.** Users shall be able to add any trading equipment to their portfolio.
 - **1.1.1.4.3.** Users each prediction success rate should be visible on their profile page.
- **1.1.1.5. User actions**
 - **1.1.1.5.1.** Users shall be able to share their portfolio on their profile page.
 - **1.1.1.5.2.** Users shall be able to follow shared portfolios.
 - **1.1.1.5.3.** Users shall be able to chase economic **events** and filter those events by considering the significance level and country base.
 - **1.1.1.5.4.** Users shall have a “Profit/Loss” section that can show profit/loss amount in terms of currency they choose.

- **1.1.1.5.5.** Users shall be able to share their ideas(predictions) on any trading equipment.
- **1.1.1.5.6.** Users shall be able to follow other users.
- **1.1.1.5.7.** Users shall be able to set alerts for certain trading equipment.
 - **1.1.1.5.7.1.** Users shall be able to set alerts for above or below of specific value on the trading equipment.
 - **1.1.1.5.7.2.** Users shall be able to set alerts for increase or decrease of specific change on the trading equipment.
- **1.1.1.5.8.** Users shall be able to publish article.
 - **1.1.1.5.8.1.** Users shall be able to write a comment to article.
 - **1.1.1.5.8.2.** Users shall be able to like and dislike the articles.
- **1.1.1.5.9.** Users shall be able to update their personal information.
- **1.1.1.5.10.** Users shall be able to update their profile picture.

* **1.1.2. Basic Users**

- **1.1.2.1.** Basic users shall be able to see their profit/loss amount in terms of the currency they choose by manually entering their investments.

* **1.1.3. Trading Users**

- **1.1.3.1.** Trading users should be able 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.
- **1.1.3.2.** Trading users shall have “My Investment” section where they can invest in any trading equipment, make a buy order for a specified rate, and set stop/loss limits.
- **1.1.3.3.** Trading users shall provide financial services corporation name and regarding IBAN number, name and surname of card owner, expiration date of the credit card and CCV number while signing up.

* **1.1.4. Guests**

- **1.1.4.1**. Guests shall be able to view the price of trading equipment and read user comments about trading equipment.
- **1.1.4.2.** Guests shall be able to search, which includes semantic searching, for users and trading equipment.

* **1.1.5. Administrators**

- **1.1.5.1.** Admins shall be able to ban users for inconvenient behaviors.

– **1.2. System Requirements**

* **1.2.1. Transaction**

- **1.2.1.1.** The system shall support transaction from users' banking account to its account balance.

* **1.2.2. Notification**

- **1.2.2.1.** The system shall be able to notice if there is a private message from other users.
- **1.2.2.2.** The system shall be able to notify about following trading equipment.
- **1.2.2.3.** The system shall be able to recommend articles and trading equipment to users based on their history.

* **1.2.3. Interface**

- **1.2.3.1.** The system shall support Turkish characters.
- **1.2.3.2.** System interface language shall be English.

* **1.2.4. Database**

- **1.2.4.1.** The system shall store and update data for each user in the database:
 - **1.2.4.1.1.** Users' search history
 - **1.2.4.1.2.** Articles published by users and its comments
 - **1.2.4.1.3.** Alerts set by users
 - **1.2.4.1.4.** Investments made by Traders
 - **1.2.4.1.5.** Portfolios
 - **1.2.4.1.6.** Trading equipment and its comments
 - **1.2.4.1.7.** Events

- **1.2.4.1.8.** Users' personal information
- * **1.2.5. Trading equipment**
 - **1.2.5.1.** The system shall support trading equipment as follow.
 - **1.2.5.1.1.** Trade Indices
 - **1.2.5.1.1.1** DJI
 - **1.2.5.1.1.2** IXIC
 - **1.2.5.1.1.3** INX
 - **1.2.5.1.2.** Stocks
 - **1.2.5.1.2.1** General Motors (GM)
 - **1.2.5.1.2.2** Google (GOOG)
 - **1.2.5.1.2.3** Apple (AAPL)
 - **1.2.5.1.3.** ETFs
 - **1.2.5.1.3.1** SPDR S&P 500 (SPY)
 - **1.2.5.1.3.2** iShares Core SP 500 (IVV)
 - **1.2.5.1.3.3** Vanguard Total (VTI)
 - **1.2.5.1.4.** Commodities
 - **1.2.5.1.4.1** Gold
 - **1.2.5.1.4.2** Silver
 - **1.2.5.1.5.** Currencies
 - **1.2.5.1.5.1.** System shall support most common currencies.
 - **1.2.5.1.5.2.** System shall have exchange rates for existing currencies.
 - **1.2.5.1.5.3.** Examples :
 - **1.2.5.1.5.3.1** Pound
 - **1.2.5.1.5.3.2** Euro
 - **1.2.5.1.5.3.3** Turkish Lira
 - **1.2.5.1.7.** Cryptocurrencies
 - **1.2.5.1.7.1** Bitcoin
 - **1.2.5.1.7.2** Litecoin
 - **1.2.5.1.7.3** Ethereum
 - **1.2.5.2.** Each trading equipment should include many functionalities as follow.

- **1.2.5.2.1.** The previous close
- **1.2.5.2.2.** Percentage change with the previous close
- **1.2.5.2.3.** Amount change with the previous close
- **1.2.5.2.4.** Day's range
- **1.2.5.2.5.** Moving averages

* **1.2.6. Search Mechanism**

- **1.2.6.1.** The system shall support search with keywords.
- **1.2.6.1.1.** The system shall support searching for users, trading equipment, and economic events.
- **1.2.6.1.2.** The system shall support some semantic search mechanism to find semantically similar users and trading equipment based on the context information provided in the semantic tags.
- **1.2.6.1.3.** The system shall enable the users to search for the other users in a specific location.

• **2. Non-functional Requirements**

– **2.1. Availability and Accessibility**

- * **2.1.1.** The application shall have a native web and native mobile(Android) client.
- * **2.1.2.** The application shall be deployable on a remote and manually configurable server.
- * **2.1.3.** The application shall be available in English.
- * **2.1.4.** The system should be able to continue operating properly in the event of the failure.
- * **2.1.5.** The number of system failures shall happen at most once in a year.

– **2.2. Annotations**

- * **2.2.1.** Annotations shall be congruent with the specifications of The W3C Web Annotation Data Model.
- * **2.2.2.** The platform shall follow W3C Web Annotation Protocol so that the contents (e.g., graph, figure, comment) can be annotated by users.

– **2.3. Performance**

- * **2.3.1.** The system shall respond to requests in 3 seconds.
- * **2.3.2.** The system should be able to respond up to 1000 requests per second.
- * **2.3.3.** The system shall be able to support up to 100000 users.

- **2.4. Privacy**

- * **2.4.1.** Personal data of users cannot be used without explicit consent.
- * **2.4.2.** Processing personal data like location shall be asked for permission.
- * **2.4.3.** How Information About You is Shared
- * **2.4.4.** Your Choices

- **2.5. Security**

- * **2.5.1.** Users shall be forced to change their passwords 6 months after registration or the latest password change.
- * **2.5.2.** Users shall be prevented to construct his/her password with well-known public pieces of information such as his name or his birthday.

8 Project Plan

	Ad	Süre	Başlat	Bitti	Önceki	Kaynak Adları
1	Backend	20 günler ¹	19.09.2019 08:00	16.10.2019 17:00		
2	Database Tables For Users	6 günler ¹	19.09.2019 08:00	26.09.2019 17:00		Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
3	Sign Up	3,5 günler ¹	27.09.2019 08:00	02.10.2019 13:00	2	Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
4	Sign In	3,5 günler ¹	02.10.2019 13:00	07.10.2019 17:00	3;3	Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
5	Password Change	3,5 günler ¹	08.10.2019 08:00	11.10.2019 13:00	4	Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
6	Sign Up Via Google Account	6 günler ¹	08.10.2019 08:00	15.10.2019 17:00	3	Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
7	Privacy and Security Const.	7 günler ¹	08.10.2019 08:00	16.10.2019 17:00		
8	User Profiles	7 günler ¹	10.10.2019 08:00	16.10.2019 17:00	2;4	Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
9	User Actions - Follow	2 günler ¹	15.10.2019 08:00	16.10.2019 17:00		Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
10	Frontend	14,5 günler ¹	02.10.2019 13:00	22.10.2019 17:00		
11	Sign Up	3,5 günler ¹	02.10.2019 13:00	16.10.2019 13:00	3	Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
12	Sign In	3,5 günler ¹	04.10.2019 08:00	21.10.2019 17:00	4	Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
13	Sign Up Via Google Account	4,5 günler ¹	16.10.2019 13:00	22.10.2019 17:00	6;11	Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
14	User Profiles	4 günler ¹	17.10.2019 08:00	22.10.2019 17:00	8	Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
15	User Actions - Follow	4 günler ¹	17.10.2019 08:00	22.10.2019 17:00	9	Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
16	Android	14,5 günler ¹	02.10.2019 13:00	22.10.2019 17:00		
17	Sign Up	3,5 günler ¹	02.10.2019 13:00	16.10.2019 13:00	3	Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
18	Sign In	3,5 günler ¹	08.10.2019 08:00	26.10.2019 17:00	4	Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
19	Password Change	3 günler ¹	17.10.2019 08:00	21.10.2019 17:00	5;18	Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
20	Sign Up Via Google Account	2,5 günler ¹	18.10.2019 13:00	22.10.2019 17:00	6;17	Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
21	User Profiles	2 günler ¹	18.10.2019 08:00	21.10.2019 17:00	8	Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
22	User Actions - Follow	2 günler ¹	18.10.2019 08:00	21.10.2019 17:00	9	Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
23	Milestone 1	1 gün ¹	22.10.2019 08:00	22.10.2019 17:00	All	
24	Backend	25 günler ¹	23.10.2019 08:00	26.11.2019 17:00		
25	User Actions	11 günler ¹	23.10.2019 08:00	06.11.2019 17:00		Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
26	Database Tables	2,75 günler ¹	23.10.2019 08:00	25.10.2019 15:00		Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
27	Transaction	11 günler ¹	23.10.2019 08:00	06.11.2019 17:00		Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
28	Trading Equipments	6 günler ¹	13.11.2019 08:00	20.11.2019 17:00		Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
29	Search Mechanism	5 günler ¹	20.11.2019 08:00	26.11.2019 17:00	26	Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
30	Frontend	14 günler ¹	07.11.2019 08:00	26.11.2019 17:00		
31	User Actions	11 günler ¹	07.11.2019 08:00	21.11.2019 17:00	25	Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
32	Transaction	11 günler ¹	07.11.2019 08:00	21.11.2019 17:00	27	Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
33	Trading Equipments	4 günler ¹	21.11.2019 08:00	26.11.2019 17:00	28	Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
34	Search Mechanism	4 günler ¹	21.11.2019 08:00	26.11.2019 17:00		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
35	Android	15 günler ¹	07.11.2019 08:00	27.11.2019 17:00		
36	User Actions	11 günler ¹	07.11.2019 08:00	21.11.2019 17:00	25	Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
37	Transaction	11 günler ¹	07.11.2019 08:00	21.11.2019 17:00	27	Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
38	Trading Equipments	4 günler ¹	21.11.2019 08:00	26.11.2019 17:00	28	Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
39	Search Mechanism	5 günler ¹	21.11.2019 08:00	27.11.2019 17:00		Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
40	Milestone 2	1 gün ¹	26.11.2019 08:00	26.11.2019 17:00	All	
41	Backend	15 günler ¹	27.11.2019 08:00	17.12.2019 17:00		
42	Notification	6 günler ¹	27.11.2019 08:00	04.12.2019 17:00		Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
43	Annotation	11 günler ¹	27.11.2019 08:00	11.12.2019 17:00		Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
44	Performance Optimization	5 günler ¹	11.12.2019 08:00	17.12.2019 17:00		Abdullah Cojuk;Omer Faruk Toptaş;İker Özkan;İrem Üstünboyacıoğlu
45	Frontend	9 günler ¹	04.12.2019 17:00	17.12.2019 17:00		
46	Notification	5 günler ¹	04.12.2019 17:00	11.12.2019 17:00	42	Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
47	Annotation	4 günler ¹	12.12.2019 08:00	17.12.2019 17:00	43	Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
48	Performance Optimization	5 günler ¹	11.12.2019 08:00	17.12.2019 17:00		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
49	Android	9 günler ¹	04.12.2019 17:00	17.12.2019 17:00		
50	Notifiers	5 günler ¹	04.12.2019 17:00	11.12.2019 17:00	42	Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
51	Annotation	4 günler ¹	12.12.2019 08:00	17.12.2019 17:00	43	Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
52	Performance Optimization	5 günler ¹	11.12.2019 08:00	17.12.2019 17:00		Emrah Saracı;Eray Kurtuluş;Yağmur Ceren Dardan
53	End-to-End Tests	2 günler ¹	14.12.2019 08:00	17.12.2019 17:00	All	
54	Delivery	1 gün ¹	17.12.2019 08:00	17.12.2019 17:00	All	

9 Evaluation of tools and managing the project

9.1 Tracking Tools(GitHub)

9.1.1 Git Feature of Github

Git is a fast, low-weight distributed version control system with a rich command set which makes collaborative project development easier. GitHub provides us remote repository We can create easily a remote repository that we can push our local repository to it for other people to view, fetch, and update. We see the changes between commits with ease by using the user interface of the GitHub.

9.1.2 Issues

In addition to providing Git remote repository, GitHub has many other features like issues, pull requests, web hooks etc. Issue feature of the GitHub is one of the most useful feature along them. We assign the actions and duties of each teammate by using GitHub Issues. Issues provide us a way to document the tasks. By using it, we can track the status of tasks. It also give an environment to discuss about the new feature, report the bugs etc.

9.1.3 Pull Requests

By using Git, we work on independent branches to develop different features. When we complete the implementation of the feature on branch, we make sure this branch is what we want, and finally we create a pull request for merging the branch into your master branch which is deployed to production. In this flow, pull request give us a chance to review the code of each other. We can easily see differences, and we can add review comment line by line by using the user interface of GitHub.

9.1.4 Project section of the Github

We track the progress of the tasks(issues) by using GitHub. In a single view, we can see the status of tasks by using this project section of the GitHub.

9.1.5 Wiki

We usually document our project to wiki page on GitHub. The requirements, the meeting notes, the communication plan of our project can be seen in the wiki page of our repository on GitHub.

<https://github.com/bounswe/bounswe2019group1/wiki>

9.2 Backend

9.2.1 Python Django

Django is chosen as main programming language for backend developments. Main reason behind that is Django is very simple and easy to learn. It is a high-level Python Web framework that encourages rapid development. We used Django REST framework which is a powerful and flexible toolkit for building Web APIs. The Django authentication system that handles both authentication and authorization is used for user authentication part of the project. For further development we use external APIs to follow current value of trading equipment and use postgresql database to store them. Besides that we also found an API for events and provide users with ability to create read update article with Django.

9.2.2 Postman

Postman is a collaboration platform for API development. We used Postman's features that simplify each step of building an API and streamline collaboration so we can create better APIs. It is easy to use, we have created a collection for our endpoints and we update it regularly. We share the latest version of the collection with other teammates, so that we can work parallel with Frontend and Android teams. Also, Postman allows us to publish our API documentation quickly and easily.

9.2.3 Docker

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.[*]

We created a Dockerfile and a docker-compose.yml file to use Docker. Then, all we need to do to run the application was running docker-compose build and docker-compose-up commands.

By doing so, thanks to the container, we assured that the application will run on any other machine regardless of any customized settings that machine might have that could differ from the machine used for writing and testing the code. We also trying to use docker as organizer of our scheduled jobs but it is still in development stage.

[*] <https://opensource.com/resources/what-docker>

9.2.4 PyCharm

As IDE, PyCharm is chosen since it is used in specifically for Python language and developed by JetBrains. It is one of the most widely used IDEs for Python programming language and also provides first-class support for a robust Python web framework like Django. It lets us to enhance productivity while coding by providing some features like suggestions, Local VCS etc.

9.3 Web

9.3.1 ReactJS

ReactJS is the most popular JavaScript library in front-end development. It has a good reputation among front-end developers and it gives us a learning motivation, and makes the learning process easier(a good reference sources can be found after a brief research). Hence, we decided to use ReactJS library in the implementation of our front-end application. React is a JavaScript library that enables us to create interactive user interface. React efficiently update and render just the right components when the data changes occurs at the state variables. Also, we can use XML like syntax called JSX with ReactJS that provide us a method for embedding HTML into React code. React is very ease of use, and modular. We used other js libraries with React such as:

9.3.2 Material-ui library

We used the production-ready components of material ui such as button, grids, card, menu item, icons etc. in our front-end pages. They are easy to use, and well-designed components that implement Google's Material Design.

9.3.3 Axios

Axios is a promise based HTTP client for the JavaScript front-end applications. It is very easy to use. We make HTTP Requests to Back-end with using it.

9.3.4 History

We manage the session history(feature that enables us go back to previous page and forward to next page in browser) by using history library.

9.3.5 React-location-picker

We used to react-location-picker library to implement the feature that pick location on Google Maps. 46

9.3.6 React-router-dom

We used to react-router-dom library to allow our application to navigate between different components, changing the browser URL, modifying the browser history, and keeping the UI state in sync.

9.3.7 ESLint

ESLint analyzes JavaScript source code to flag programming error, bugs, stylistic errors, and suspicious constructs such as:

Strings should always use double quotes (never single quotes) and should always appear on a single line.

Never use a slash to create a new line in a string.

Numbers should be written as decimal integers, e-notation integers, hexadecimal integers or floating-point decimals with at least one digit before and one digit after the decimal point.

The usage of the special value "null" is restricted only to special conditions. By using ESLint, we have agreed on a particular coding style. Our code stays clean and easy to read as possible.

9.3.8 Webstorm IDE

Since we decided to implement our front-end application with ReactJS, we need to an IDE that will make this ReactJS implementation process easier.

We search for different IDEs and finally decided to use WebStorm which is one of the JetBrains tools which can be used as free with student license. The code completion of the WebStorm is ahead of any other IDE. It has blazingly fast debugger. It has built-in support for ESLint, we can fix ESLint problems automatically with using it.

9.3.9 Visual Studio Code

In addition to Webstorm, we produced solutions quickly and accurately with smart suggestions or completions by using VS Code, another popular application for writing javascript code. VS Code has syntax highlighting for the 47 various source code elements and, if you put the cursor on a parentheses, the matching bracket is also selected.

9.4 Android

9.4.1 Kotlin for Android

Kotlin, a new open source programming language, has become popular among Android developers since Google announced its preferred language as Kotlin for developing Android applications. Here is the list of properties that have made us choose Kotlin:

- Kotlin is compatible with Java and it operates under the Java Virtual Machine, so it is possible to switch language between Kotlin and Java without any side effect.
- Kotlin syntax offers to write less code and increase in readability compared to Java, results in saving in time.
- Kotlin handles NullPointerExceptions and runtime overhead problems in Java.

9.4.2 Retrofit

Retrofit is a library for Android to make HTTP requests as a REST Client. Retrofit basically creates model classes and uses these models to parse API responses and requests. The reason we have chosen this technology is that type-safe feature of Retrofit. By this way compiler will validate all the request bodies while compiling and it will reduce errors caused by API queries.

9.4.3 IDE and Simulation Preferences

We choose Android Studio IDE to develop our Android application considering integration with Kotlin and built-in emulators to simulate our Android application. In order to test our application and its compatibility, various devices have been used such as Pixel 3(API 27), Lenovo P2(API 24) and Lenovo 0560(API 25)

10 User Manual

This document is the version 1.0 of the Software User Manual for the Khaji-it Web & Mobile Application. The Software User Manual instructs how to use the Khaji-it software. This project is part of the Software Engineering course (CMPE451). For information regarding how to install and setup Khaji-it, please take look at our System Manual.

Introduction

Khaji-it is a social platform for users who can be considered as a trader. The platform enables the users to trade and follow a broad variety of financial vehicles including indices, stocks, ETFs, bonds, commodities, currencies, funds, bonds, and cryptocurrencies. The platform will support various interactions, such as sharing ideas as an article, commenting and rating ideas of other users, commenting about trading equipment. Users will be able to follow other users, and trading equipment and also set alerts for certain levels of trading equipment.

Another remarkable feature of Khaji-it is that you can add an annotation to articles as a text or an image part. This feature was created according to the W3C Standards.

10.1 Android User Manual



Figure 38: Home Page

When you open the Khaji-it app, you will be redirected to Home Page. From here you can login, register, or continue as guest.



Figure 39: Guest Article List Page

When guest clicks continue as guest in the Home Page, it is redirected to article list page. Here guest can see article list and can go into one of the articles. At the top bar, guest can click login or register to be redirected to regarding pages. User can switch to other pages from bottom bar.

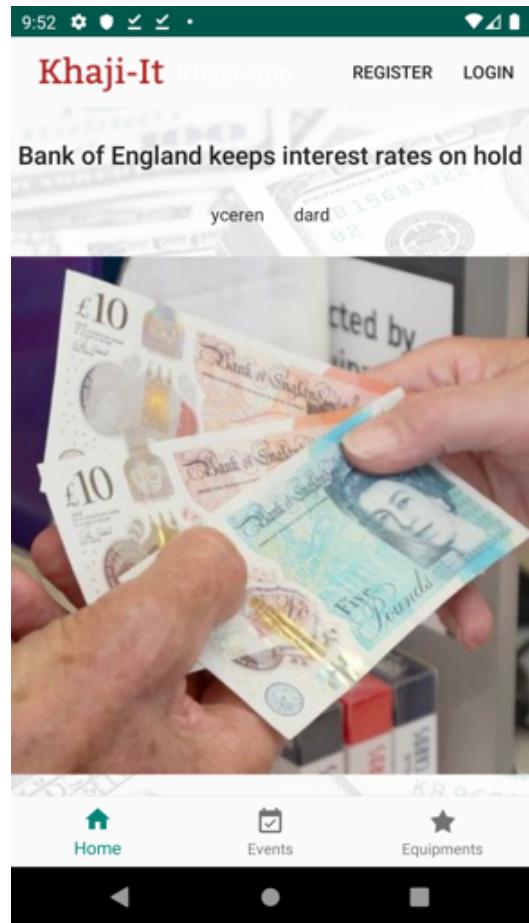


Figure 40: Guest Article Page

Here guest can see the information about article including writer and annotation.

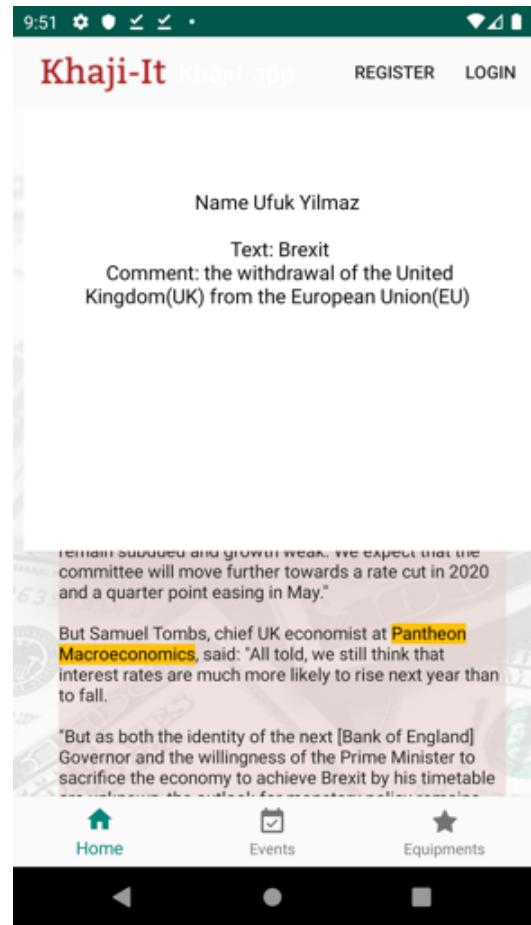


Figure 41: Guest Article Annotation Page

When guest clicks on the yellow background text; guest can see the annotated text, annotation, and who wrote this annotation.

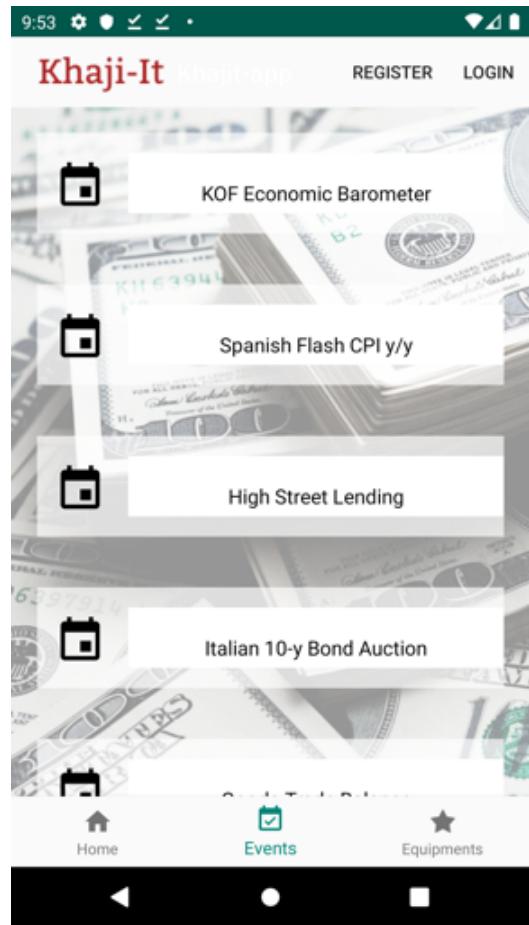


Figure 42: Event List Page

All users can see this page and can get the detail of the specific event by clicking one of them.



Figure 43: Event Page

Here users can see detailed version of the event.

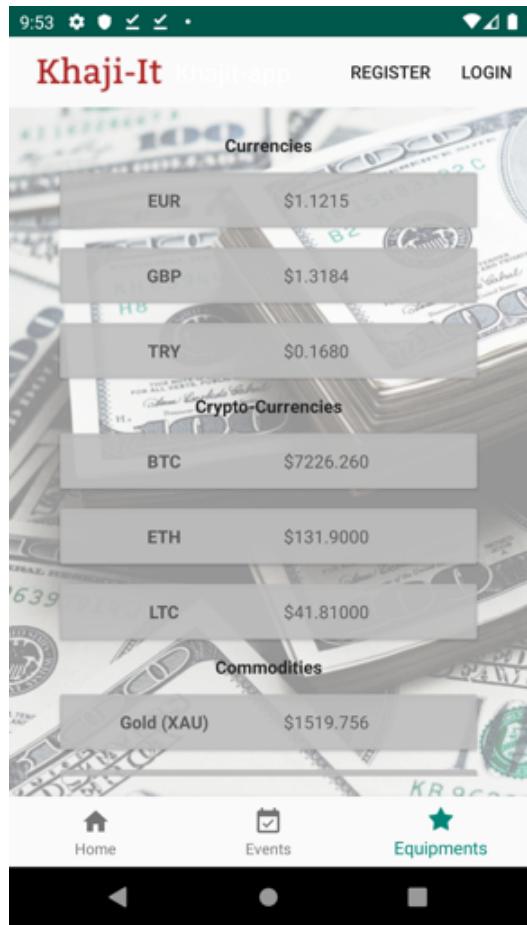


Figure 44: Equipment Page

Guest user can see the current prices of the equipments.

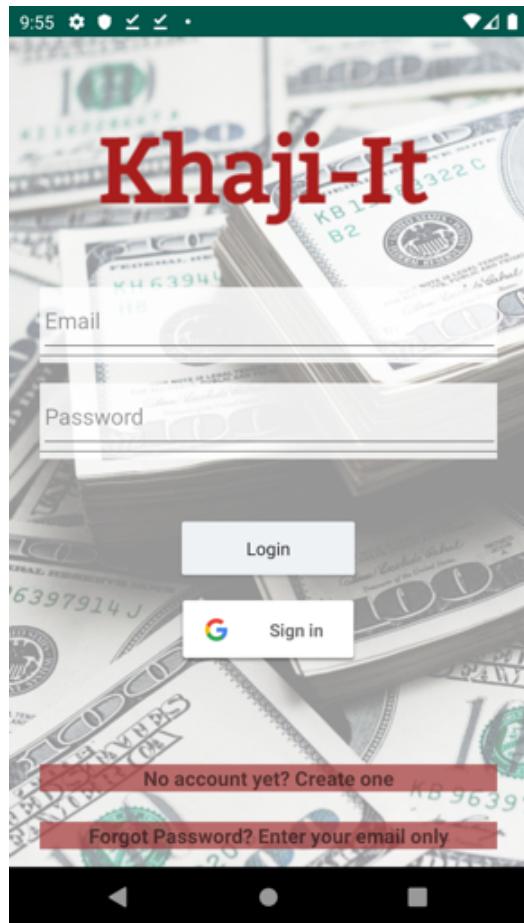


Figure 45: Login Page

If user clicks to Login from Home Page, user is redirected to this page. Here user can type his personal information to login or can simply sign in with Gail account. User can be directed to register back with clicking "No account yet?..." After login user is move to Article page as in Guest Article List Page.

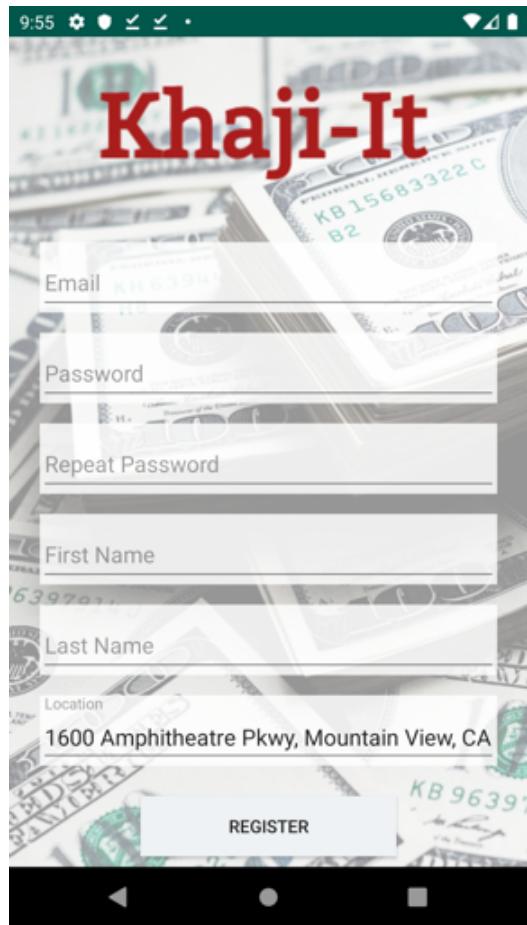


Figure 46: Register Page

If user clicks to Register from Home Page, user is redirected to this page. User fills the required boxes in order to register. Location is retrieved automatically from your GPS and application will request from you to enable GPS in this part.



Figure 47: Search Article Page

User can search in this page and can be directed to here by clicking search icon from bottom bar. Here user can search for users, articles, and events.

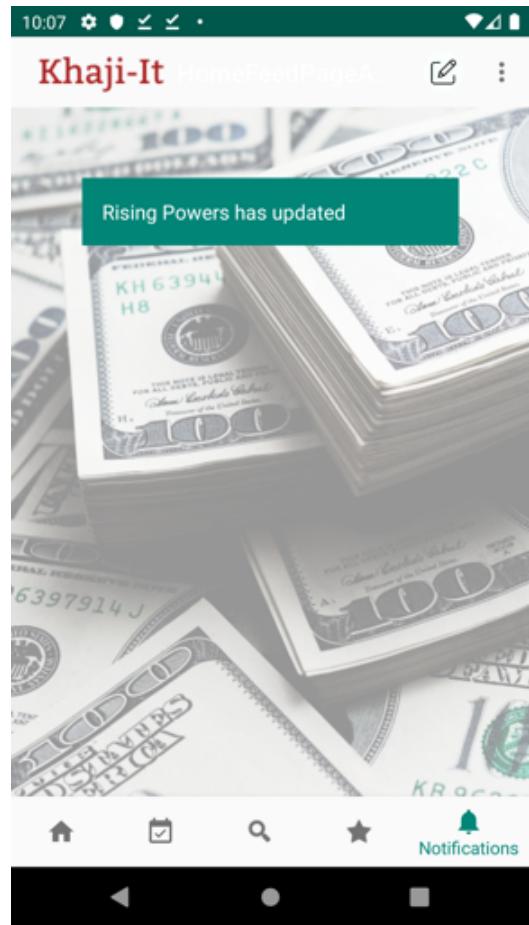


Figure 48: Notification Page

User can switch to notification page by clicking star symbol from bottom bar. Here user can see changes on the followed portfolios or friend requests to the respective user.(Example is a portfolio change)

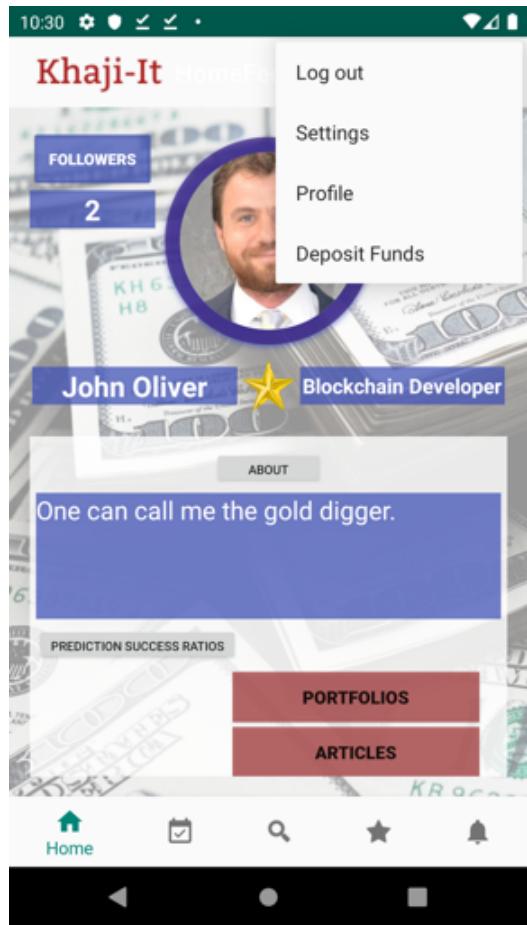


Figure 49: Top Menu Page

User can switch to other pages by clicking top-right corner and top menu will be seen. From there user can log out, go into settings page where user can edit his personal information, go into profile page(as in this page), or can go into deposit page.

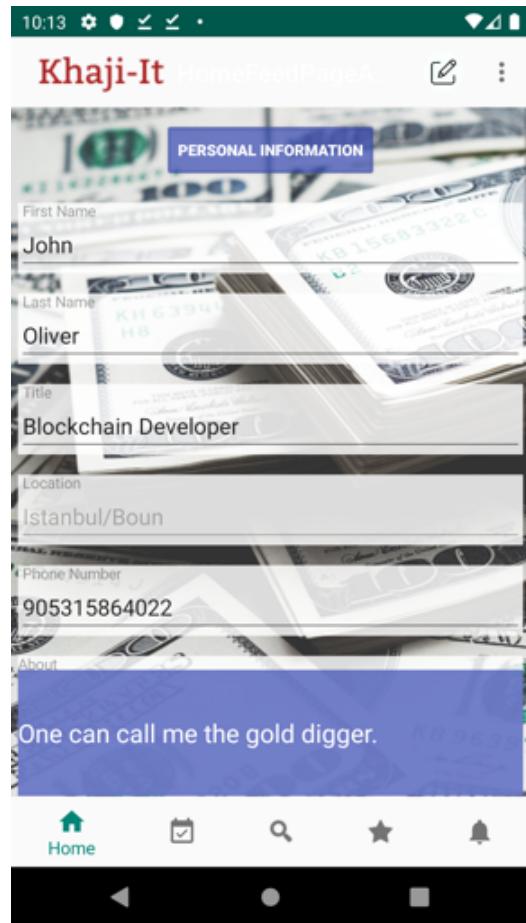


Figure 50: Top Setting Page

After clicking settings from top menu, user reaches here. User can edit its personal information in this page such as changing first name, last name, title, phone number, about, upgrade/downgrade to trader/basic, change privacy mode, change password, and update profile picture.

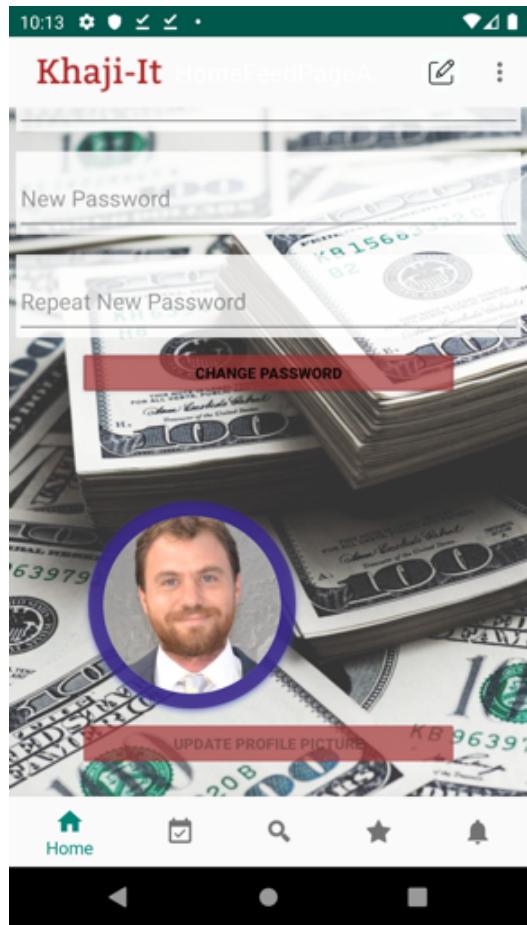


Figure 51: Bottom Settings Page

User can change his password from settings menu as shown in the top of the figure. User can change his account profile page by clicking current image from here and will be redirected to media and will be requested to access to files. After clicking an image user will see here how it looks and can apply those changes by clicking bottom of the picture bar.

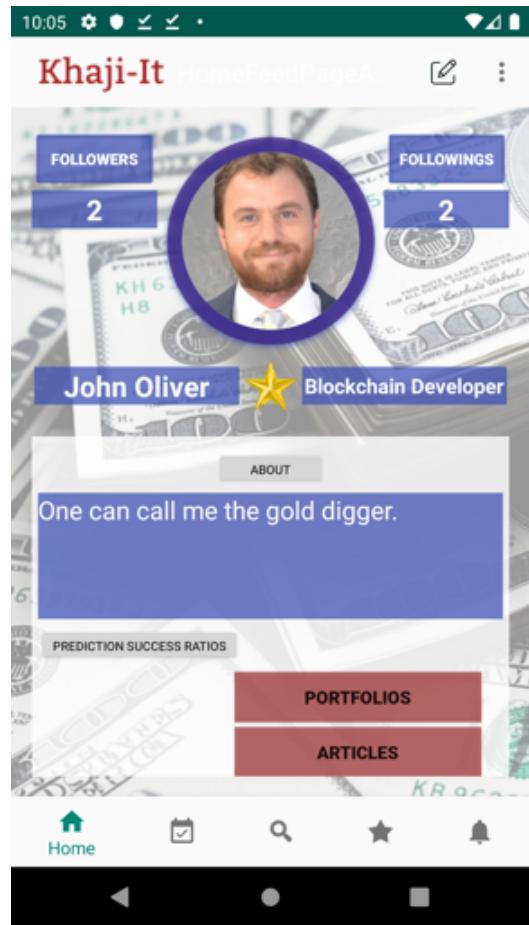


Figure 52: Profile Page

After clicking Profile from top menu, user reaches here. User can see his information and if middle star is shining that means user is trader. User can go into Portfolio page and Follower/Following List Page by clicking respective bars.

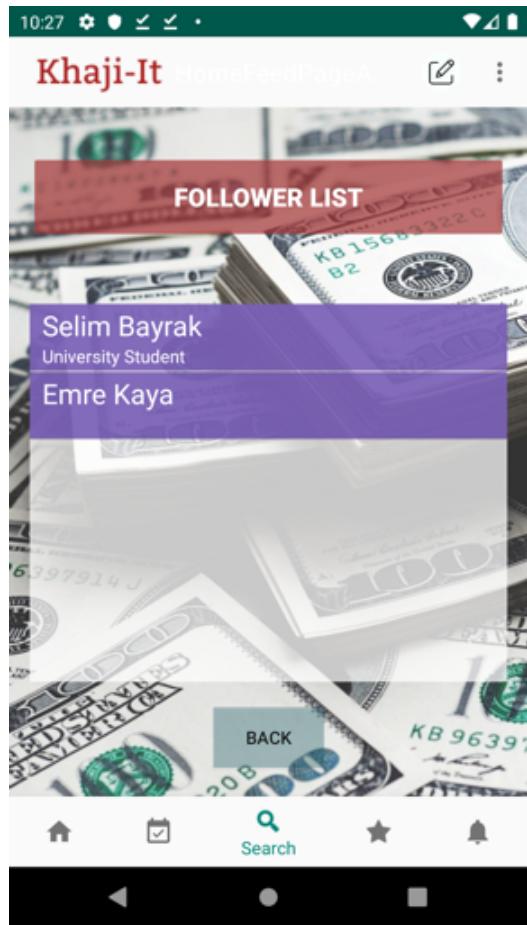


Figure 53: Follower List Page

If user clicks either follower or following in the Profile Page, he will see the list of the users.



Figure 54: My Portfolio List Page

Users can see their or others' portfolio pages as in here. Shown names are the name of the created portfolios. Since this is a My Portfolio page user can create portfolio or can see one of the portfolios.



Figure 55: Portfolio Page

This is how portfolio is shown. If shown portfolio belongs to other user, user will see follow/unfollow button. If belongs to this user, he can edit the portfolio.

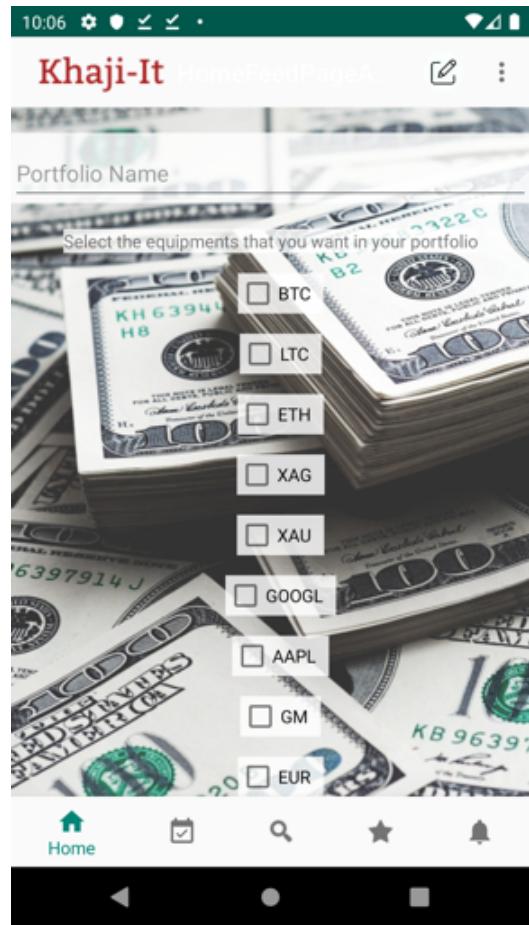


Figure 56: Create/Edit Portfolio Page

When user tries to create portfolio, he is expected to enter a name for portfolio and click at least one equipment.

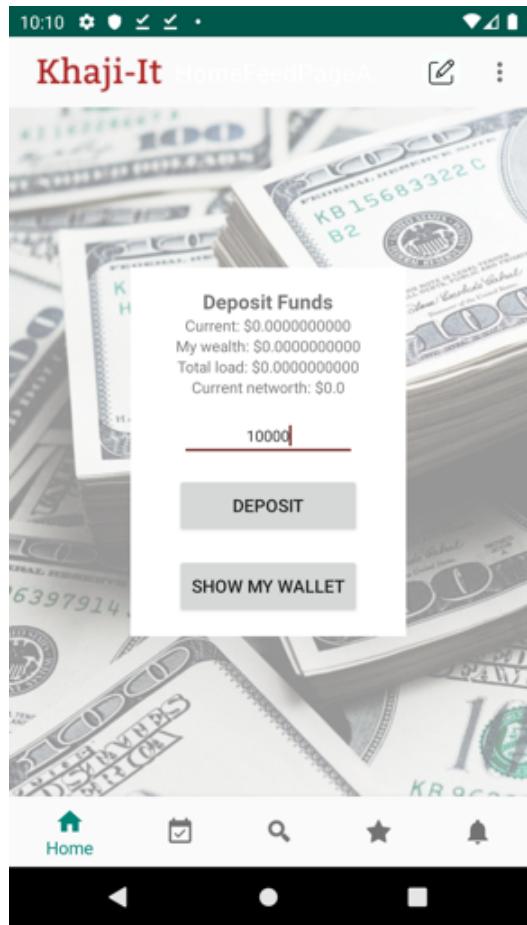


Figure 57: Deposit Page

After clicking deposit fund from top menu, user reaches here. Here user can deposit money to his Khaji-it application account. Here also shows current wallet values such as current money, wealth, total load, and current networth.



Figure 58: Wallet Page

If user clicks Wallet button from Deposit Page, user comes here. User can see all belongings to himself in the application. Both amounts and their current values.

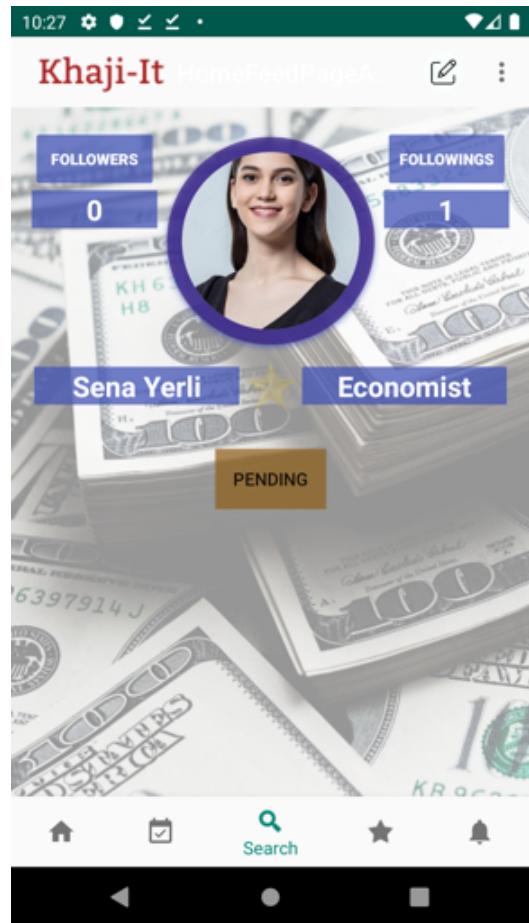


Figure 59: Pending Other User Page

This is page of other users. User can look at the other users by searching them. In this example, user has clicked to follow and in the pending status.

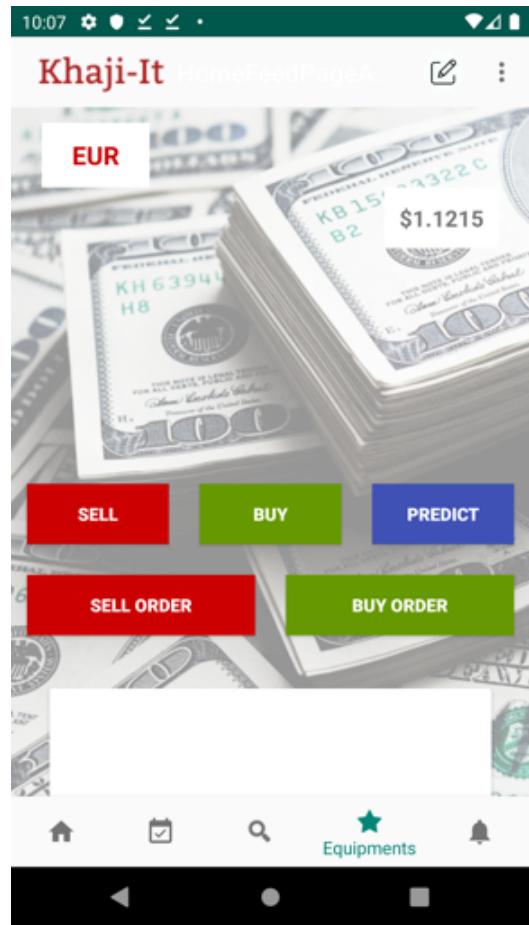


Figure 60: EUR Equipment Page

From Equipment Page by clicking one of the equipments, user can go into one specific equipment page. User can see the value of the trading equipment, buy/sell equipment, predicts its future value, make a buy or sell order for this equipment.

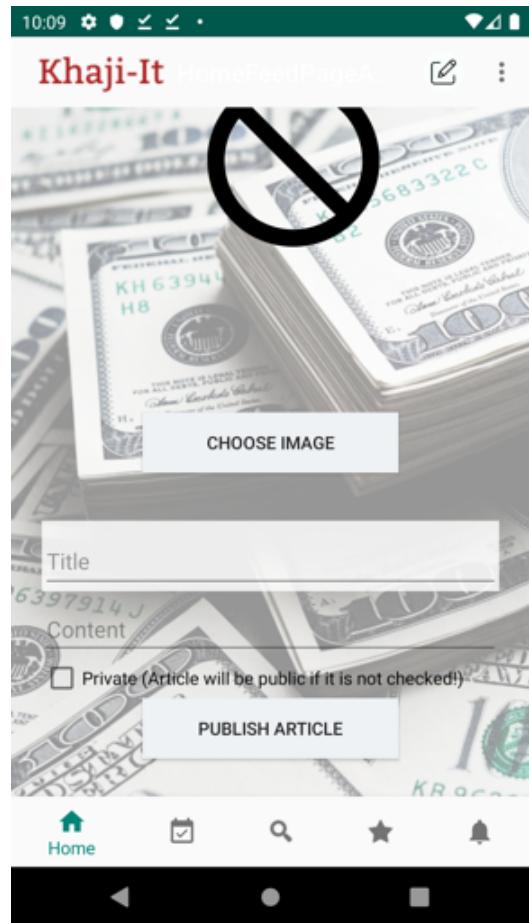


Figure 61: Create Article Page

User can write article by clicking top right "write icon". In here user can upload a image for article and write about this article. User can define article as public or private.

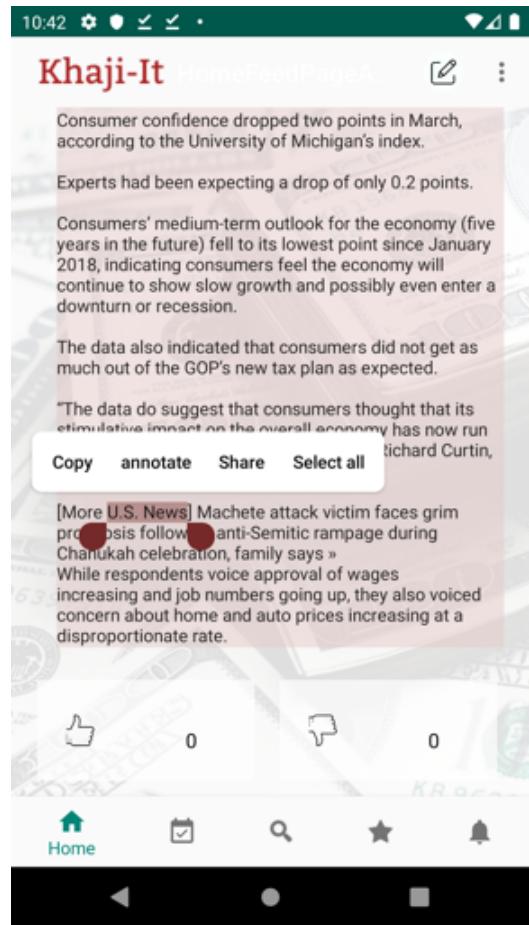


Figure 62: Example Annotation Page 1

Let's give an examples for 3 steps of annotation in the articles. Here logged in user selects "U.S. News" from article and clicks annotate pop-up button.

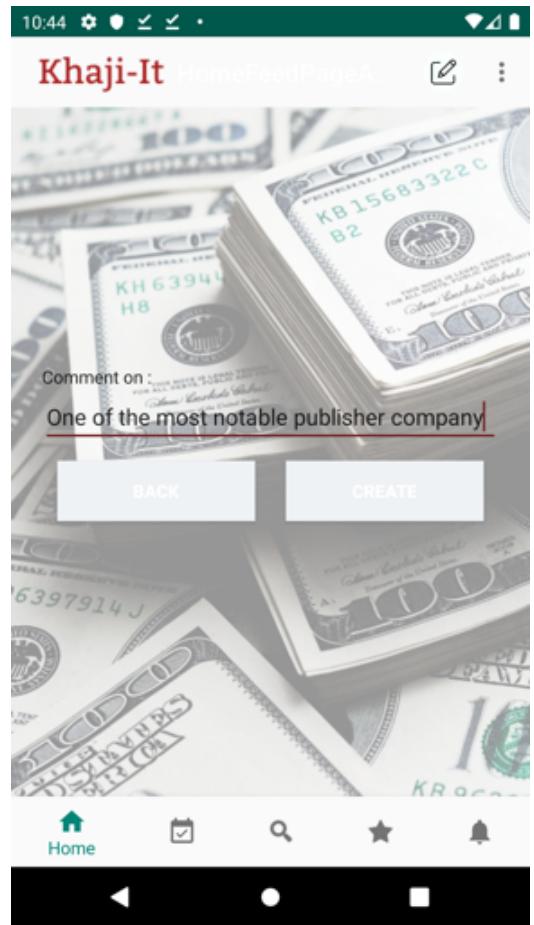


Figure 63: Example Annotation Page 2

Writes about annotated text and click create.



Figure 64: Example Annotation Page 3

Now annotation is complete and when any user clicks on the yellow text, he can see the information about annotation. More people can annotate same text and when clicked on the text, list of the annotation will be shown. An annotation can be wrapped by another annotation and clicking the sub annotation will also show wrapper annotation and list will indicate which annotation belongs to which text.

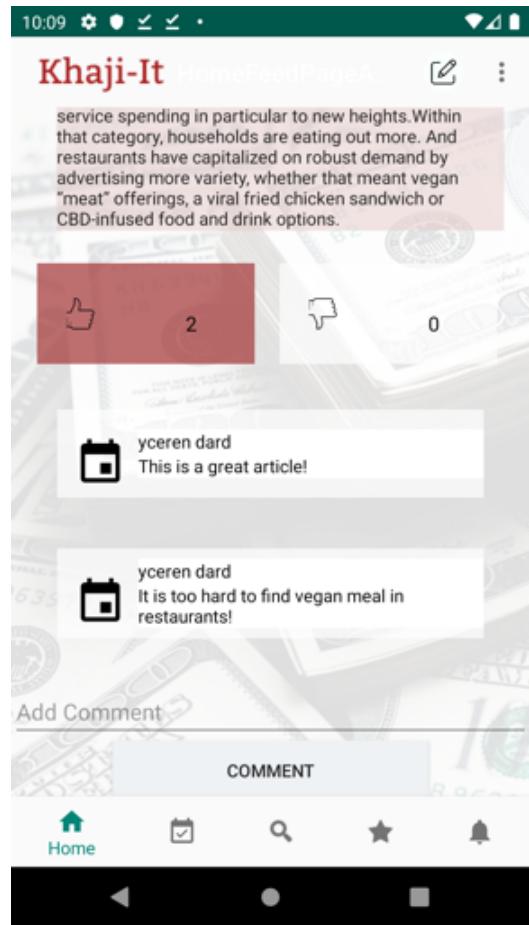


Figure 65: Article Bottom Page

This is page bottom part of the article page. Here user can like or dislike the article. Moreover, he can write a comment about the article. Only logged in users can do this.

10.2 Web User Manual

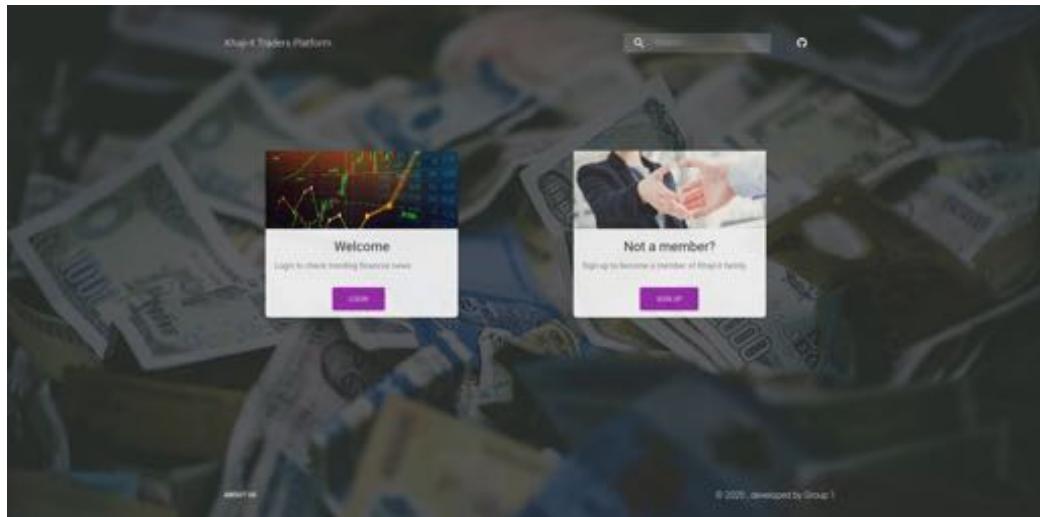


Figure 66: Home Page

Thanks to the handy home page, users can easily sign up when they enter the site or log in with the user information they have previously created.

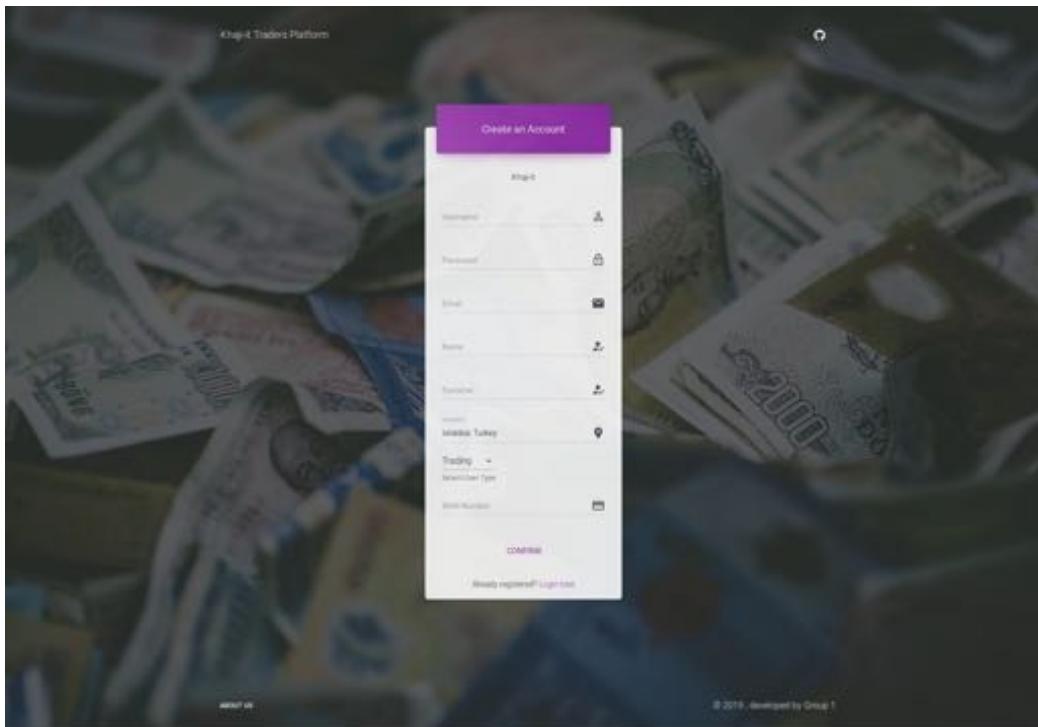


Figure 67: Register as Trader

On the registration screen you can fill in the required information and register as trader or basic user. You can also add your address easily thanks to the location finding feature on the map.

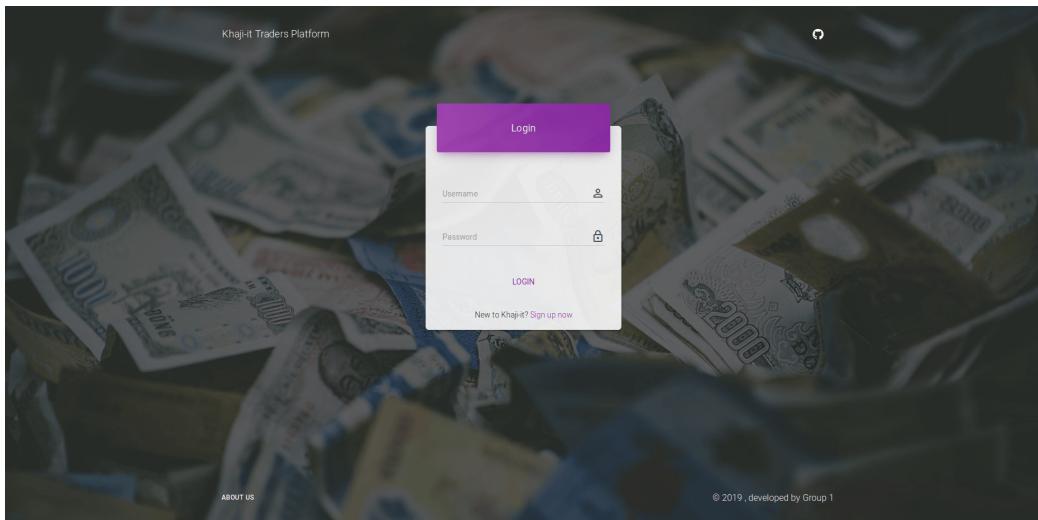


Figure 68: Login Page

Thanks to the simple design you can easily login to the site. If you forget your password, you can go to the reset password page.

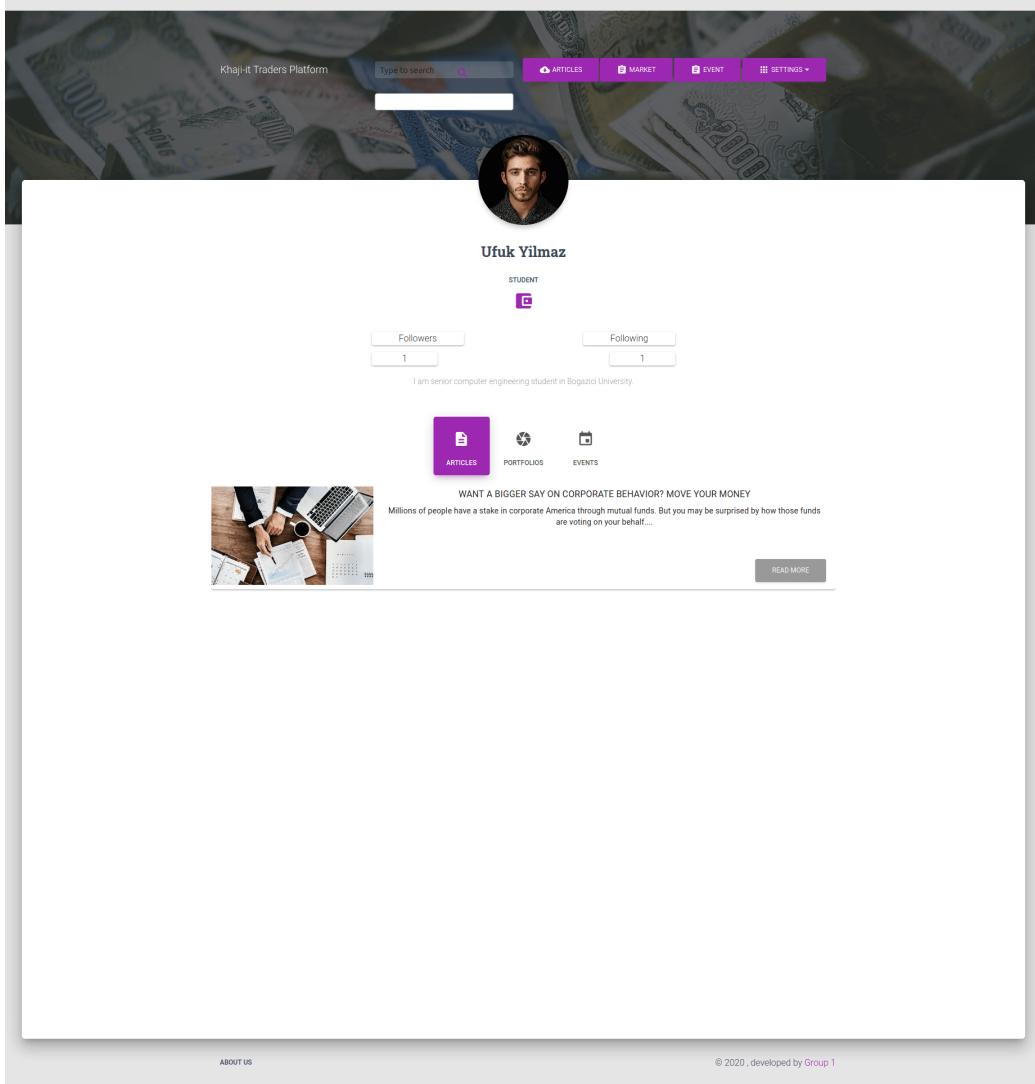


Figure 69: Profile Page

You can view the articles you have created on the profile page, view the events you like and find out the number of followers. You can also check how much money you have by going to the wallet page.

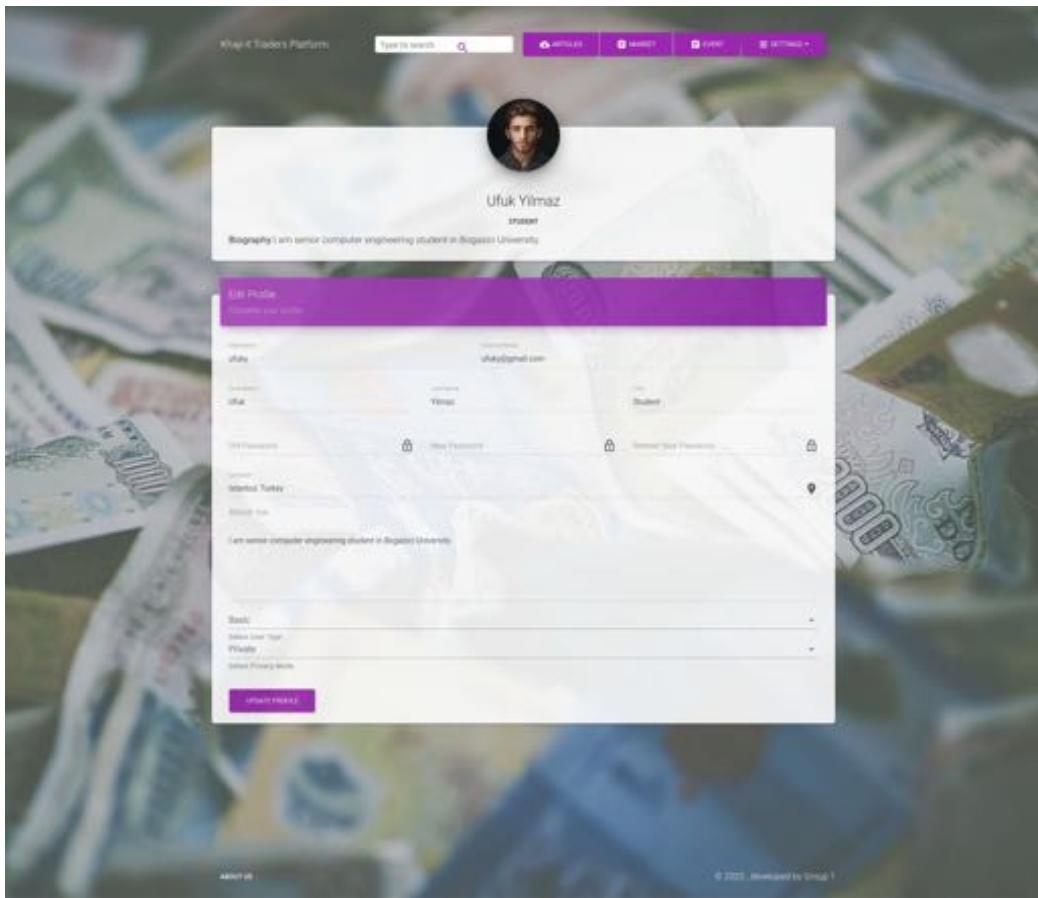


Figure 70: Edit Profile Page

From the profile edit page, you can edit your basic biography, change password, or edit basic configurations, such as changing your user type.

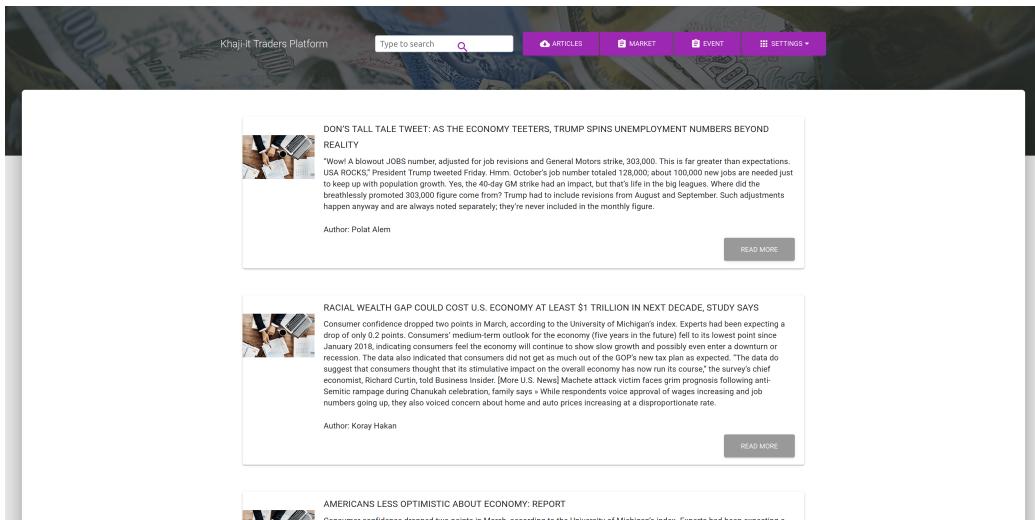


Figure 71: Article Listing Page

From the Articles page, you can view the articles of the users you follow and read and comment in detail by clicking read more.

The screenshot shows the Market page of the Khaji-it Traders Platform. At the top, there is a navigation bar with the platform's name, a search bar, and links for ARTICLES, MARKET, EVENT, and SETTINGS. Below the navigation bar, a purple header bar displays the word "Market" and the message "Equipment that is listed". The main content area is a table listing 19 different financial instruments, each with its symbol, long name, and current parity (exchange rate). The table has three columns: "Equipment Symbol", "Equipment Long Name", and "Parity". Each row also includes a small profile icon.

Equipment Symbol	Equipment Long Name	Parity
EUR/USD	Euro/US Dollar	1.1215515094961765
GBP/USD	British Pound Sterling/US Dollar	1.3184263265368843
TRY/USD	Turkish Lira/US Dollar	0.16806951489591707
XAG/USD	Silver/US Dollar	17.88204999821172300000
XAU/USD	Gold/US Dollar	1519.75683990578200000000
BTC/USD	Bitcoin/US Dollar	7226.260000000000
ETH/USD	Ethereum/US Dollar	1.2952
LTC/USD	Litecoin/US Dollar	46235.9183
GOOG/USD	Google/US Dollar	1398.800000000000
AAPL/USD	Apple/US Dollar	1296.4000
GM/USD	General Motors / US Dollar	52.4900
SPY/USD	Standard & Poor's Depository / US Dollar	\$321.86
IVV/USD	Ishares S&P 500 / US Dollar	123213
VTV/USD	Vanguard Total Stock Market Index Fund / US Dollar	123212

Figure 72: Market Page

On the Market page, you can get information about the current exchange of all currencies and view their pages privately for trading accordingly.

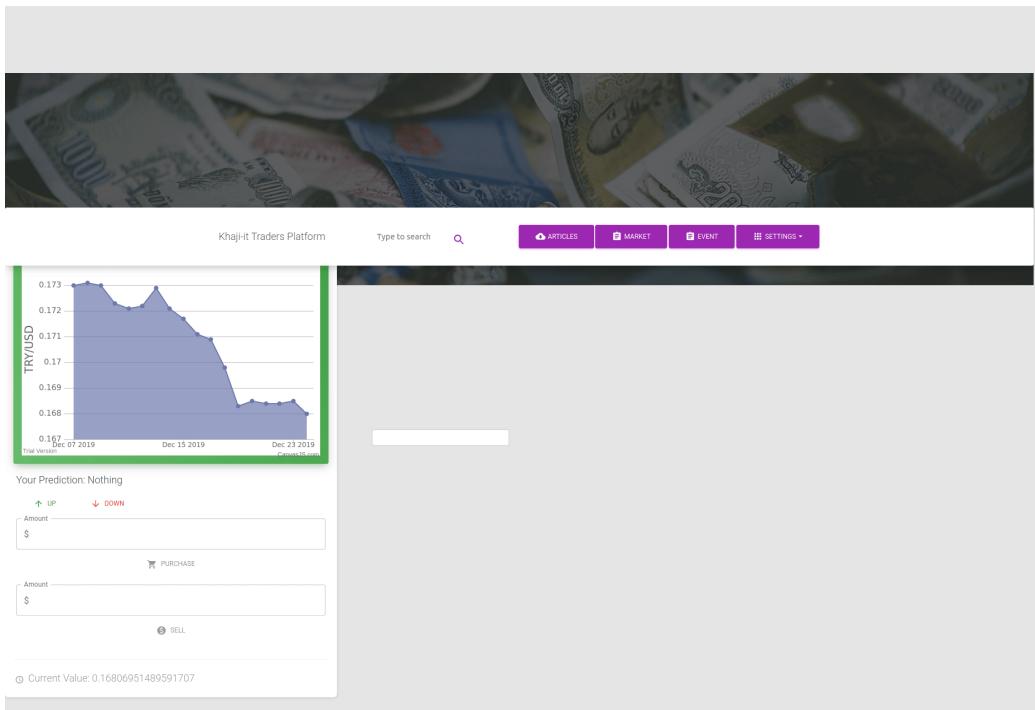


Figure 73: Equipment Page

On this page, you can view the exchange chart of the selected trading equipment, make a prediction about it, or make a purchase.

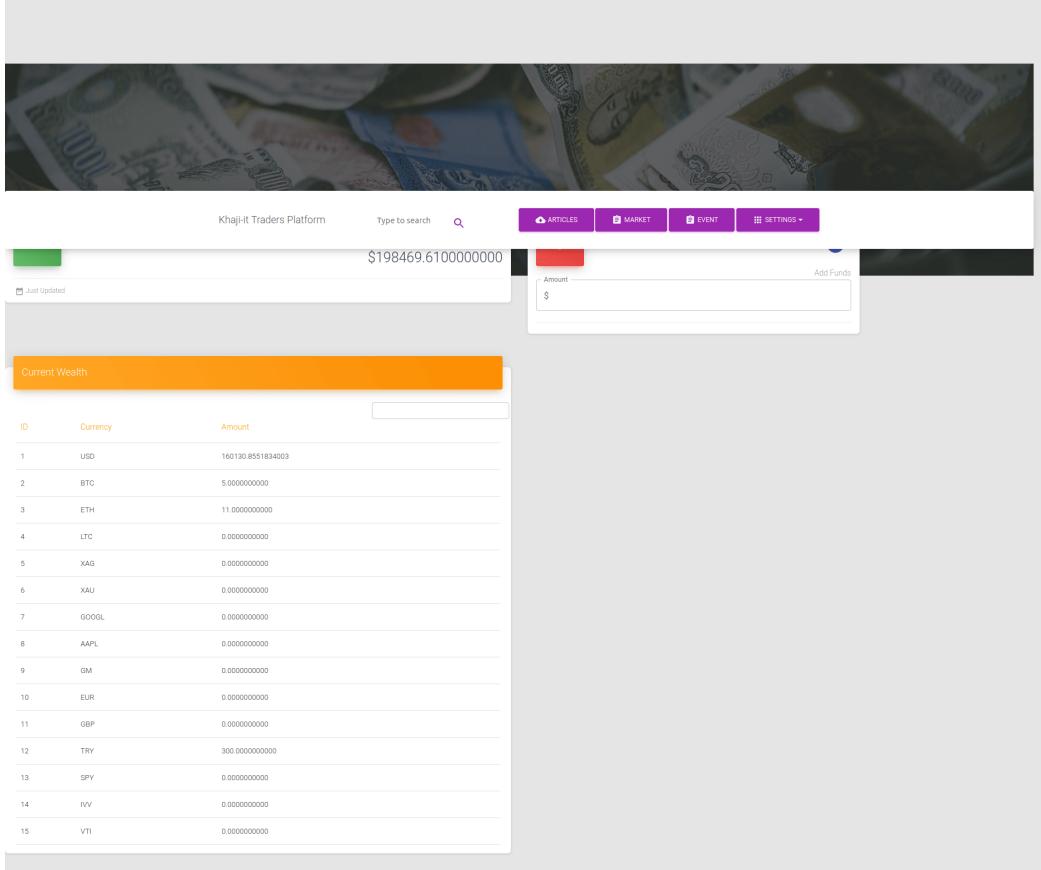


Figure 74: Wallet Page

On the wallet page, you can view how much money you have in total, and what amount of equipment you have. You can also add money from this page.

Title	Country	Date	Impact	Forecast	Previous
Private Sector Credit m/m	AUD	2019-12-22T19:30:00-05:00	Low	0.2%	0.1%
All Industries Activity m/m	JPY	2019-12-22T23:30:00-05:00	Low	-4.3%	1.5%
German Import Prices m/m	EUR	2019-12-23T01:48:00-05:00	Low	0.1%	-0.1%
Durable Goods Orders m/m	USD	2019-12-23T08:30:00-05:00	Medium	0.3%	0.6%
Core Durable Goods Orders m/m	USD	2019-12-23T08:30:00-05:00	Medium	1.5%	0.6%
GDP m/m	CAD	2019-12-23T08:30:00-05:00	High	0.1%	0.1%
New Home Sales	USD	2019-12-23T10:00:00-05:00	Low	730K	732K
Treasury Currency Report	USD	2019-12-23T17:03:00-05:00	Medium		
Monetary Policy Meeting Minutes	JPY	2019-12-23T18:00:00-05:00	Low		
BIZ Core CPI j/y	JPY	2019-12-24T00:00:00-05:00	Low	0.2%	0.2%
German Bank Holiday	EUR	2019-12-24T02:00:00-05:00	Holiday		
Richmond Manufacturing Index	USD	2019-12-24T10:00:00-05:00	Low	1	-1
Bank Holiday	NZD	2019-12-24T15:00:00-05:00	Holiday		
Bank Holiday	AUD	2019-12-24T16:00:00-05:00	Holiday		
GPII j/y	JPY	2019-12-24T18:00:00-05:00	Low	2.1%	2.1%
Bank Holiday	CHF	2019-12-25T01:00:00-05:00	Holiday		
French Bank Holiday	EUR	2019-12-25T02:00:00-05:00	Holiday		
German Bank Holiday	EUR	2019-12-25T02:00:00-05:00	Holiday		
Italian Bank Holiday	EUR	2019-12-25T02:03:00-05:00	Holiday		
Bank Holiday	GBP	2019-12-25T03:00:00-05:00	Holiday		
Bank Holiday	CAD	2019-12-25T07:50:00-05:00	Holiday		
Bank Holiday	USD	2019-12-25T08:30:00-05:00	Holiday		
Bank Holiday	NZD	2019-12-25T15:00:00-05:00	Holiday		
Bank Holiday	AUD	2019-12-25T16:00:00-05:00	Holiday		
Housing Starts j/y	JPY	2019-12-26T00:00:00-05:00	Low	-8.1%	-7.8%
BIZ Gov Funds Speaks	JPY	2019-12-26T00:30:00-05:00	Low		
Bank Holiday	CHF	2019-12-26T02:00:00-05:00	Holiday		
German Bank Holiday	EUR	2019-12-26T02:00:00-05:00	Holiday		
Italian Bank Holiday	EUR	2019-12-26T02:03:00-05:00	Holiday		
Bank Holiday	GBP	2019-12-26T03:00:00-05:00	Holiday		
Bank Holiday	CAD	2019-12-26T07:50:00-05:00	Holiday		
Unemployment Claims	USD	2019-12-26T08:30:00-05:00	Low	222K	234K
Unemployment Rate	JPY	2019-12-26T18:30:00-05:00	Low	2.4%	2.4%
Tokyo Core CPI j/y	JPY	2019-12-26T18:30:00-05:00	Low	0.6%	0.6%
BIZ Summary of Opinions	JPY	2019-12-26T18:50:00-05:00	Low		
Prelm Industrial Production m/m	JPY	2019-12-26T18:50:00-05:00	Low	-1.1%	-4.2%
Retail Sales j/y	JPY	2019-12-26T18:50:00-05:00	Low	-1.4%	-7.1%
ECB Economic Bulletin	EUR	2019-12-27T04:00:00-05:00	Low		
Credit Suisse Economic Expectations	CHF	2019-12-27T04:00:00-05:00	Low		-0.9
High Street Lending	GBP	2019-12-27T04:30:00-05:00	Low	41.5K	41.2K
Natural Gas Storage	USD	2019-12-27T10:00:00-05:00	Low		-107B
Crude Oil Inventories	USD	2019-12-27T11:00:00-05:00	Medium		-1.3M

Figure 75: Events Page

On the Event page, you can view the latest news about the equipment you are following and strengthen your prediction based on the information

here.

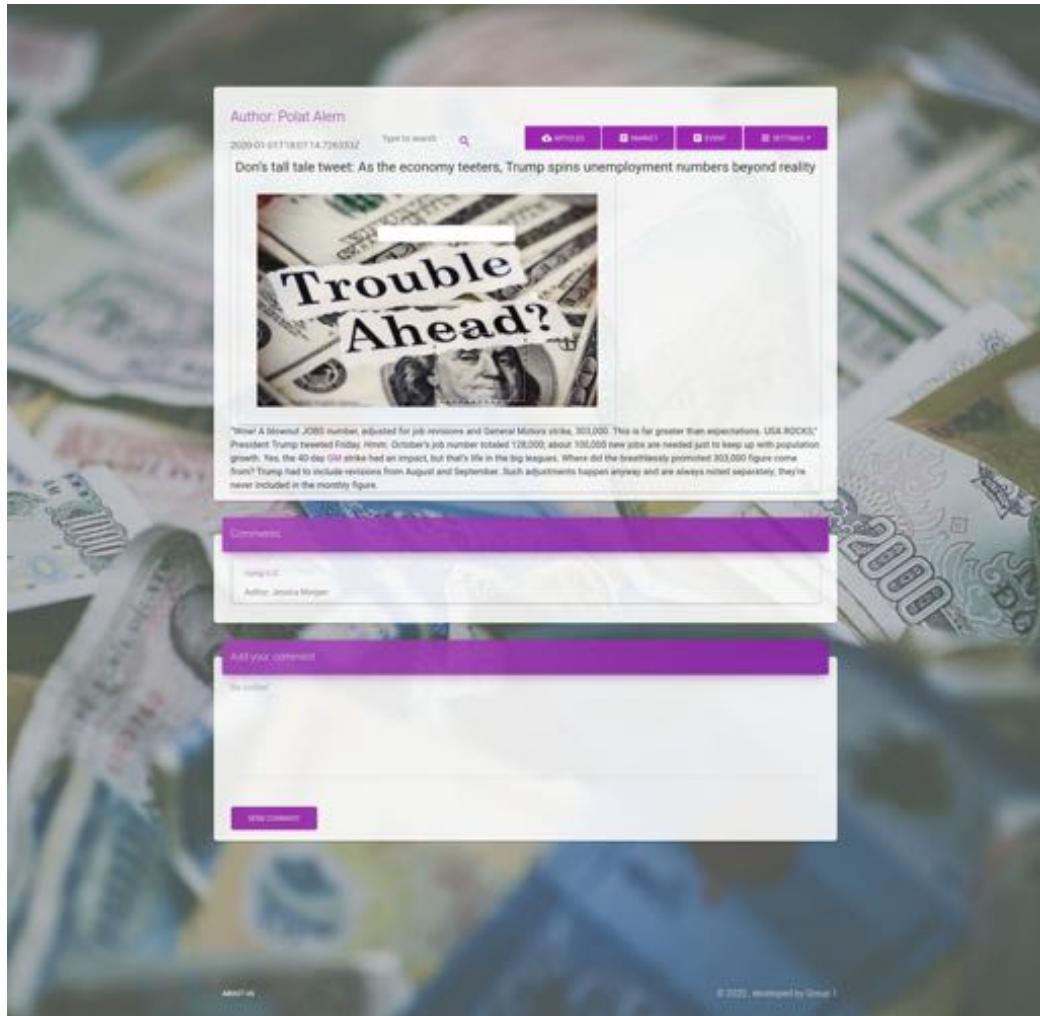


Figure 76: Article Page

At the bottom of the article page, user can can write a comment about the article. Only logged in users can do this.

11 System Manual

11.1 Frontend

The client-side of web application is implemented with using ReactJS library, on Javascript. In order to run client-side application, you need to install node first. Then you can run these commands below:

- ‘npm install‘ to install project dependencies.
- ‘npm start‘ to run the app in the development mode. Open [*http://localhost : 3000*] (*http://localhost : 3000*) to view it in the browser. The page will reload if you make edits. You will also see any lint errors in the console.
- ‘npm run build‘ Builds the app for production to the ‘build‘ folder. It correctly bundles React in production mode and optimizes the build for the best performance. The build is minified and the filenames include the hashes. The app is ready to be deployed. Then, serve the static contents using serve by running these commands `npm install -g serve` and then `serve -s build -l <PORT>`
- npm run lint to lint code
- npm run lint –fix to auto fix linter errors.

Browsers Support We test our application on

- Chrome 79.0.3945.88
- Firefox 71.0

11.2 Android

11.2.1 Software Description

Android software is simply an android application of Khaji-it project and provides simple user interfaces for clients and applies all the features of the Khaji-it API offers.

11.2.2 Benefits Value

The benefits of using with android software, developers may:

- Use API interface developed within Android environment
- Implement new use case scenarios based on provided layouts
- Implement new use case scenarios based on provided layouts

11.2.3 Platform Requirements

Requirements for android software can be listed as:

- Android SDK v29 (minSdkVersion: 24, targetSdkVersion: 29)
- Kotlin Android
- Dependencies: ,
 - Android Support Libraries (activity:1.0.0, annotation:1.1.0, appcompat-resources:1.1.0, appcompat:1.1.0, core-common:2.1.0, core-runtime:2.1.0, asynclayoutinflater:1.0.0, cardview:1.0.0, collection:1.1.0, constraintlayout-solver:1.1.3, constraintlayout:1.1.3, coordinatorlayout:1.1.0-beta01, core-ktx:1.1.0, core:1.1.0, cursoradapter:1.0.0, customview:1.0.0, databinding-adapters:3.5.1, junit:4.12, viewpager:1.0.0, versionedparcelable:1.1.0, espresso-core:3.2.0, recyclerview:1.1.0-beta04, databinding-common:3.5.1, fragment:1.1.0, lifecycle-common:2.1.0)
 - Square Libraries (okio:1.15.0, picasso:2.71828, retrofit:2.5.0, gson:2.8.2, viewpager2:1.0.0-beta04)
 - Other Libraries (kxml2:2.3.0, httpclient:4.0.1, httpcore:4.0.1, hamcrest-core:1.3, hamcrest-integration:1.3, hamcrest-library:1.3, kotlin-android-extensions-runtime:1.3.50, kotlin-stdlib-common:1.3.50, kotlin-stdlib-jdk7:1.3.50, kotlin-stdlib:1.3.50, kotlinx-coroutines-android:1.3.2, kotlinx-coroutines-core:1.3.2, glide:4.4.0, result:1.6.0, autoimages-lider:1.3.2, auth-api-impl:11.6.0, play-services-auth-api-phone:17.0.0, play-services-auth-base:17.0.0, play-services-auth:17.0.0, play-services-base:17.0.0, play-services-basement:17.0.0, play-services-location:17.0.0, play-services-places-placereport:17.0.0, play-services-tasks:17.0.0)

Khaji-it android software was developed in Kotlin language with Retrofit library which turns the project's HTTP API into a Java interface. In addition to Retrofit, Glide library is used to display images with their Uri provided from Khaji-it API. Besides, one may use Java as a language and take advantage of Java files which are generated from implemented Kotlin software. You may want to use Android Studio as development environment since project was developed within Android Studio and below descriptions include some usage of it.

11.2.4 Importing Project to Android Studio

File → New → Project from Version Control → Git

URL: <https://github.com/bounswe/bounswe2019group1> After importing project, you will see Android project files within Android scope in the Project panel. You can view and edit Gradle Scripts. There are two Gradle Scripts: one for app and one for Android. We suggest that editing Gradle Script for app is enough. Project file structure can be seen in app package. Implemented package name is "android.khajit".

If you want to see or edit implemented Kotlin software, the following path will take you to the required folder: app → src → main → java → com.project.khajit_app.activity. There are packages (folders) and MainActivity file.

11.2.5 Software Details

Interfaces

- interfaces.IOnBackPressed
- interfaces.fragmentOperationsInterface.kt

Activity

- com.project.khajit_app.activity.HomeFeedPageActivity:
- com.project.khajit_app.activity.HomeFeedPageActivity:
- com.project.khajit_app.activity.Globals:
Implements global variables during a session of a user.
- com.project.khajit_app.activity.LoginPageActivity:
Implements login action of a user.

- com.project.khajit_app.activity.HomeFeedPageGuestActivity:
- com.project.khajit_app.activity.UserViewAdapters:
- com.project.khajit_app.activity.SignUpPageTraderActivity:
- com.project.khajit_app.activity.FollowListViewAdapter:
- com.project.khajit_app.activity.OtherPortfolioListAdapter:
- com.project.khajit_app.activity.PortfolioListAdapter:
- com.project.khajit_app.activity.HelperFunctions:
- com.project.khajit_app.activity.OnePortfolioViewAdapter:
- com.project.khajit_app.activity.ui:
 - com.project.khajit_app.activity.ui.annotation:
 - com.project.khajit_app.activity.ui.article:
 - com.project.khajit_app.activity.ui.editprofile:
 - com.project.khajit_app.activity.ui.event:
 - com.project.khajit_app.activity.ui.followlist:
 - com.project.khajit_app.activity.ui.home:
 - com.project.khajit_app.activity.ui.myportfolio:
 - com.project.khajit_app.activity.ui.mywallet:
 - com.project.khajit_app.activity.ui.notification:
 - com.project.khajit_app.activity.ui.otherprofile:
 - com.project.khajit_app.activity.ui.otherportfolio:
 - com.project.khajit_app.activity.ui.prediction:
 - com.project.khajit_app.activity.ui.search:
 - com.project.khajit_app.activity.ui.equipment:
 - com.project.khajit_app.activity.ui.profile:

API

- com.project.khajit_app.api.Api:

- com.project.khajit_app.api.AnnotationApi:
- com.project.khajit_app.api.RetrofitClient:

Data

- com.project.khajit_app.data.annotationModels:
 - com.project.khajit_app.data.annotationModels.AddBodyModel:
 - com.project.khajit_app.data.annotationModels.AnnotationModelResponse:
 - com.project.khajit_app.data.annotationModels.BodyModel:
 - com.project.khajit_app.data.annotationModels.CreateAnnotationModel:
 - com.project.khajit_app.data.annotationModels.CreateAnnotationModelNewCreator:
 - com.project.khajit_app.data.annotationModels.CreateAnnotationResponse:
 - com.project.khajit_app.data.annotationModels.CreatorExistsResponse:
 - com.project.khajit_app.data.annotationModels.CreatorListModel:
 - com.project.khajit_app.data.annotationModels.CreatorModel:
 - com.project.khajit_app.data.annotationModels.DeleteAnnotationModel:
 - com.project.khajit_app.data.annotationModels.DeleteAnnotationResponse:
 - com.project.khajit_app.data.annotationModels.GenericAnnotationModel:
 - com.project.khajit_app.data.annotationModels.GetAnnotationModelResponse:
 - com.project.khajit_app.data.annotationModels.RefinedByModel:
 - com.project.khajit_app.data.annotationModels.SelectorModel:
 - com.project.khajit_app.data.annotationModels.ShowImageAnnotationModel:
 - com.project.khajit_app.data.annotationModels.ShowTextAnnotationModel:
 - com.project.khajit_app.data.annotationModels.TargetModel:
 - com.project.khajit_app.data.annotationModels.sourceModel:
- com.project.khajit_app.data.Models:
 - com.project.khajit_app.data.modelsArticleCommentItem:
 - com.project.khajit_app.data.modelsArticleDislikeResponseModel:
 - com.project.khajit_app.data.modelsArticleLikeDisLikeResponseModel:

- com.project.khajit_app.data.modelsArticleLikeCountResponseModel:
- com.project.khajit_app.data.modelsArticleLikeDislikeModel:
- com.project.khajit_app.data.modelsArticleLikeResponseModel:
- com.project.khajit_app.data.modelsArticleSearchModelResponse:
- com.project.khajit_app.data.modelsArticleSearchResponse:
- com.project.khajit_app.data.modelsBasicRegisterResponse:
- com.project.khajit_app.data.modelsBasicUser:
- com.project.khajit_app.data.modelsChangePrivacy:
- com.project.khajit_app.data.modelsCommodityResponse:
- com.project.khajit_app.data.modelsCreateArticleModel:
- com.project.khajit_app.data.modelsCreateArticleResponseModel:
- com.project.khajit_app.data.modelsCreateCommentModel:
- com.project.khajit_app.data.modelsCreateCommentResponseModel:
- com.project.khajit_app.data.modelsCryptoResponse:
- com.project.khajit_app.data.modelsCurrencyResponse:
- com.project.khajit_app.data.modelsDepositFundsModel:
- com.project.khajit_app.data.modelsDepositFundsResponse:
- com.project.khajit_app.data.modelsETFResponse:
- com.project.khajit_app.data.modelsEquipmentBSModel:
- com.project.khajit_app.data.modelsEquipmentBSOrderModel:
- com.project.khajit_app.data.modelsEquipmentBSOrderResponse:
- com.project.khajit_app.data.modelsEventModel:
- com.project.khajit_app.data.modelsFollowIDModel:
- com.project.khajit_app.data.modelsFollowIDModelResponse:
- com.project.khajit_app.data.modelsFollowIdListRequestModel:
- com.project.khajit_app.data.modelsFollowModel:
- com.project.khajit_app.data.modelsFollowModel2:
- com.project.khajit_app.data.modelsFollowUnfollowModel:
- com.project.khajit_app.data.modelsFollowUnfollowResponseModel:

- com.project.khajit_app.data.modelsFollowingPendingListModel:
- com.project.khajit_app.data.modelsFollowingPendingResponseModel:
- com.project.khajit_app.data.modelsGeneralArticleModel:
- com.project.khajit_app.data.modelsGeneralArticleSpannableModel:
- com.project.khajit_app.data.modelsGeneralFollowModel:
- com.project.khajit_app.data.modelsGeneralFollowModel2:
- com.project.khajit_app.data.modelsGeneralNotificationModel:
- com.project.khajit_app.data.modelsGenericLoginModel:
- com.project.khajit_app.data.modelsGenericUserModel:
- com.project.khajit_app.data.modelsListArticleCommentModel:
- com.project.khajit_app.data.modelsListEventModel:
- com.project.khajit_app.data.modelsListEventResponse:
- com.project.khajit_app.data.modelsListNotificationsResponse:
- com.project.khajit_app.data.modelsListPortfolioResponse:
- com.project.khajit_app.data.modelsLoggedInUser:
- com.project.khajit_app.data.modelsLoginResponse:
- com.project.khajit_app.data.modelsOneEventResponse:
- com.project.khajit_app.data.modelsOnePortfolioResponse:
- com.project.khajit_app.data.modelsPasswordChange:
- com.project.khajit_app.data.modelsPendingFollowerResponse:
- com.project.khajit_app.data.modelsPortfolioDeleteResponseModel:
- com.project.khajit_app.data.modelsPortfolioEditRequestModel:
- com.project.khajit_app.data.modelsPortfolioEditResponseModel:
- com.project.khajit_app.data.modelsPortfolioModel:
- com.project.khajit_app.data.modelsPortfolioOwnerModel:
- com.project.khajit_app.data.modelsPredictionModel:
- com.project.khajit_app.data.modelsPredictionResponseModel:
- com.project.khajit_app.data.modelsPublicArticleListResponse:
- com.project.khajit_app.data.modelsSearchRequest:

- com.project.khajit_app.data.modelsSearchResponse:
- com.project.khajit_app.data.modelsShowArticleCommentModel:
- com.project.khajit_app.data.modelsStockResponse:
- com.project.khajit_app.data.modelsTradeIndiceResponse:
- com.project.khajit_app.data.modelsTraderUser:
- com.project.khajit_app.data.modelsUpdateUser:
- com.project.khajit_app.data.modelsUpdateUserResponse:
- com.project.khajit_app.data.modelsUpgradeDowngrade:
- com.project.khajit_app.data.modelsUserAllInfo:
- com.project.khajit_app.data.modelsUserInfoGet:
- com.project.khajit_app.data.modelsWalletResponse:
- com.project.khajit_app.data.modelscreateWalletResponse:
- com.project.khajit_app.data.modelsisFollowingPortfolioResponse:
- com.project.khajit_app.data.modelsisFollowingResponseModel:
- com.project.khajit_app.data.modelsuserToBeLogin:

Global

- com.project.khajit_app.global.User:

Service

- com.project.khajit_app.service.FetchAddressIntentService

11.3 Backend

We developed backend of this project with Python 3.7 and Django 2.2.6. To provide regularity and compatibility we create requirements.txt and use pip freeze to write libraries we used. In this point guide for backend divides two part: To run the code in local with sqlite and to run the code with docker over postgresql. Since we use docker to run in frontend and mobile team we try to maximize usability of docker. So we add every command to setup our application in Dockerfile and docker-compose.yml. Running code in local with sqlite mostly used in backend team because set up is more complicated but development with this procedure is easier to use.

11.3.1 Set Up In Local

Before running this project first you have to install every library we are using. If you want to user virtual environment:

To create virtual environment the command should be :

- `virtualenv "name_of_virtualenv"`

then to activate:

for windows:

- `source name_of_virtualenv/bin/activate`

for mac:

- `. name_of_virtualenv/bin/activate`

You should create virtual environment for only first time but activate every time.

Then you should install packages in requirements if any new package was added to project :

- `pip install -r requirements.txt`

Then the project is ready to go. To create or update database:

- `python manage.py makemigrations --settings=tradersplatform.settings.local`
- `python manage.py migrate --settings=tradersplatform.settings.local`

After that you can run the project:

- `python manage.py runserver --settings=tradersplatform.settings.local`

*Base settings were created according to the docker file so user who tries to run with `python manage.py` should also give the settings.

*Also if both python2 and python3 are installed in your machine or virtualenvironment you should specify the version of python in every python command like:

- `python3 manage.py runserver --settings=tradersplatform.settings.local`

11.3.2 Set Up With Docker

With docker setting up and running is much more easier because every database commands and run commands exists in dockerfile and docker-compose.yml. Only steps you should do will be :

- Installing docker
- Go the directory of docker-compose.yml file
- Run docker-compose build
- Run docker-compose up
- If any change in database was made docker-compose down then docker-compose up
- If any change in libraries was made docker-compose down then docker-compose build docker-compose up

11.4 Annotation Backend

Before running this project first you have to install every library we are using. If you want to user virtual environment:

To create virtual environment the command should be :

- `virtualenv "name_of_virtualenv"`

then to activate:

for windows:

- source name_of_virtualenv/bin/activate

for mac:

- . name_of_virtualenv/bin/activate

You should create virtual environment for only first time but activate every time.

Then you should install packages in requirements if any new package was added to project :

- pip install -r requirements.txt

Then the project is ready to go. To create or update database:

- python manage.py makemigrations
- python manage.py migrate

After that you can run the project:

- python manage.py runserver 0.0.0.0:8020

*Annotation should be served in 8020 port according to our design *Also if both python2 and python3 are installed in your machine or virtualenvironment you should specify the version of python in every python command like:

- python3 manage.py runserver 0.0.0.0:8020

12 API Documentation

12.1 POST trader register

{host}/user/registertrader/

12.1.1 Headers

Content-Type application/json

12.1.2 Body

```
raw
{
  "username": "melike_",
  "password": "12qwas.1?",
  "email": "melike.sonmez@boun.edu.tr",
  "first_name": "melike",
  "last_name": "sonmez",
  "location": "İstanbul",
  "iban_number": "111111111131211"
}
```

12.2 POST basic register

{host}/user/registerbasic/

12.2.1 Headers

Content-Type application/json

12.2.2 Body

```
raw
{
  "username": "melike_",
  "password": "12qwas.1?",
  "email": "melike.sonmez@boun.edu.tr",
  "first_name": "melike",
```

```
"last_name": "sonmez",  
"location": "İstanbul"  
}
```

12.3 POST login

{host}/user/login/

12.3.1 Headers

Content-Type application/json

12.3.2 Body

```
raw  
{  
"username": "melike_",  
"password": "12qwas.1?"  
}
```

12.4 GET auto login/profile

{host}/user/profile/

This request returns information about current user. It takes user id from token then accesses user information from it then returns that information. This request can be used in profile page.

12.4.1 Headers

Authorization JWT {{token}}

12.5 GET User Retrieve

{host}/user/retrieve/3/

12.5.1 Headers

Authorization JWT {{token}}

12.6 PUT User Update

{host}/user/updateuser/

12.6.1 Headers

Content-Type application/json
Authorization JWT {token}

12.6.2 Body

```
raw
{
  "title": "General Manager"
}
```

12.7 PUT User Password Change

{host}/user/updatepass/

This request updates password if user can give correct current password and a valid new password.

12.7.1 Headers

Content-Type application/json
Authorization JWT {token}

12.7.2 Body

```
raw
{
  "old_password": "qwerasdf",
  "new_password": "12345678"
}
```

12.8 POST search

{host}/user/search_user/

This request can returns a user list when given string is a substring of user-name.

12.8.1 Headers

Content-Type application/json
Authorization JWT {token}

12.8.2 Body

```
raw
{
  "username": "substring_of_username"
}
```

12.9 POST Follow Someone With ID

{host}/follow/follow/
With this request current user can follow another person.

12.9.1 Headers

Content-Type application/json
Authorization JWT {token}

12.9.2 Body

```
raw
{
  "following":2
}
```

12.10 GET List Follower Of Current User

{host}/follow/listFollower/

12.10.1 Headers

Content-Type application/json
Authorization JWT {token}

12.11 GET List Follower Of Given User

{host}/follow/listFollowerWithId/

When an id is given this request returns that user's followers.

12.11.1 Headers

Content-Type application/json

Authorization JWT {token}

12.11.2 Body

raw

```
{  
  "id":1  
}
```

12.12 GET List Following Of Current User

{host}/follow/listFollowing/

12.12.1 Headers

Content-Type application/json

Authorization JWT {token}

12.13 GET List Following Of Given User

{host}/follow/listFollowingWithId/

When an id is given this request returns that user's followings.

12.13.1 Headers

Content-Type application/json

Authorization JWT {token}

12.13.2 Body

```
raw
{
  "id":3
}
```

12.14 DELETE Unfollow Someone

{host}/follow/delete/

12.14.1 Headers

Content-Type application/json
Authorization JWT {token}

12.14.2 Body

```
raw
{
  "following":1
}
```

12.15 GET Currency List

{host}equipment/currencyList/

12.16 GET Crypto Currency List

{host}equipment/cryptocurrencyList/

12.17 GET Metal Currency List

{host}equipment/metalcurrencyList/

12.18 GET Stock List

{host}equipment/stockcurrencyList/

12.19 GET Etf List

{host}equipment/etfList/

12.20 GET Trace Indices List

{host}equipment/traceList/

12.21 GET Currency Last Month

{host}equipment/currencylastmonth/

12.22 GET Currency

{host}equipment/currency/

12.23 GET currency convert

{host}equipment/currencyconvert/

12.23.1 Headers

Content-Type application/json

12.23.2 Body

```
raw
{
  "from": "USD",
  "to": "TRY"
}
```

12.24 GET Metal Currency

{host}equipment/metalcurrency/

12.25 GET Crypto Currency

{host}equipment/cryptocurrency/

12.26 GET Crypto Currency Last Month

{host}equipment/cryptocurrencyhistorical/

12.27 GET Stock Currency

{host}equipment/stock/

12.28 GET Trace Indices Currency

{host}equipment/traceindices/

12.29 GET Gainers in Trace Indices

{host}equipment/traceindicesgainers/

12.30 GET Last Month of Stock

{host}equipment/lastmonth/

12.30.1 Headers

Content-Type application/json

12.30.2 Body

```
raw
{
  "company_symbol": "GM"
}
```

12.31 POST Forgot Password

{host}/user/forgotpassword/

12.31.1 Headers

Content-Type application/json

12.31.2 Body

```
raw
{
  "email": "abdullah.coskun1@boun.edu.tr"
}
```

12.32 PUT User Password Change Forgot Password

{host}/user/updatepassforgot/4/

12.32.1 Headers

Content-Type application/json

12.32.2 Body

```
raw
{
  "new_password": "12345678"
}
```

12.33 PUT User Upgrade

{host}/user/userupgrade/

12.33.1 Headers

Authorization JWT {token}

12.34 PUT User Downgrade

{host}/user/userdowngrade/

12.34.1 Headers

Authorization JWT {token}

12.35 POST Create Article

{host}/article/create/

12.35.1 Headers

Content-Type application/json
Authorization JWT {token}

12.35.2 Body

```
raw
{
  "title": "Article of Requirements",
  "content": "This is an article about traders platform",
  "is_public": true
}
```

12.36 PUT Update Article

{host}/article/update/

12.36.1 Headers

Content-Type application/json
Authorization JWT {token}

12.36.2 Body

```
raw
{
  "id": 2,
  "title": "Article of Requirements",
  "content": "EDIT! This is an article about traders platform",
  "is_public": true
}
```

12.37 GET List Article

{host}/article/list/

12.37.1 Headers

Authorization JWT {token}

12.38 POST List Public Articles

{host}/article/listPublicArticles/

12.38.1 Headers

Authorization JWT {token}

12.39 POST List Article With User ID

{host}/article/listArticleByUserId/1/

12.39.1 Headers

12.40 DEL Delete Article

{host}/article/delete/

12.40.1 Headers

Content-Type application/json

Authorization JWT {token}

12.40.2 Body

```
raw
{
  "id": 2
}
```

12.41 POST Wallet Create

{host}/wallet/createWallet/

12.41.1 Headers

Authorization JWT {token}

12.42 POST Wallet Retrieve

{host}/wallet/retrieve/

12.42.1 Headers

Authorization JWT {token}

12.43 PUT Send USD to Wallet

{host}/wallet/sendUSD/

12.43.1 Headers

Content-Type application/json

Authorization JWT {token}

12.43.2 Body

```
raw
{
  "USD": 12745
}
```

12.44 PUT Take Equipment

{host}/wallet/takeequipment/

12.44.1 Headers

Content-Type application/json

Authorization JWT {token}

12.44.2 Body

```
raw
{
  "name_of_eq": "VTI",
  "amount": 11.24
}
```

12.45 PUT Sell Equipment

{host}/wallet/sellequipment/

12.45.1 Headers

Content-Type application/json
Authorization JWT {token}

12.45.2 Body

```
raw
{
  "name_of_eq": "VTI",
  "amount": 1.24
}
```

12.46 POST Create Comment

{host}/article-comment/create/

12.46.1 Headers

Content-Type application/json
Authorization JWT {token}

12.46.2 Body

```
raw
{
```

```
"text": "This article is awesome!",  
"article_id": 5  
}
```

12.47 POST Buy Order

{host}/notification/createbuyorder/

12.47.1 Headers

Content-Typeapplication/json AuthorizationJWT usergeneraltoken

12.47.2 Body

```
raw  
{  
"currency":"AAPL",  
"amount":11.123,  
"buy_amount":15  
}
```

12.48 POST Sell Order

{host}/notification/createsellorder/

12.48.1 Headers

Content-Typeapplication/json AuthorizationJWT usergeneraltoken

12.48.2 Body

```
raw  
{  
"currency":"BTC",  
"amount":10.123,  
"sell_amount":0.001  
}
```

12.49 POST Follow Portfolio

{host}/portfolio-follow/follow/

12.49.1 Headers

AuthorizationJWT {{usergeneraltoken}}
Content-Typeapplication/json

12.49.2 Body

```
raw
{
  "portfolio_id": 6
}
```

12.50 GET Get Portfolio Follow

{host}/portfolio-follow/id/

12.50.1 Headers

AuthorizationJWT {{usergeneraltoken}}
Content-Typeapplication/json

12.51 GET List Followers With Portfolio Id

{host}/portfolio-follow/listFollowersWithPortfolioId/id/

12.51.1 Headers

AuthorizationJWT {{usergeneraltoken}}
Content-Typeapplication/json

12.52 GET List All

{host}/portfolio-follow/list/

12.52.1 Headers

AuthorizationJWT {{usergeneraltoken}}
Content-Typeapplication/json

12.53 GET List Portfolios With Follower Id

{host}/portfolio-follow/listPortfoliosWithFollowerId/id/

12.53.1 Headers

AuthorizationJWT {{usergeneraltoken}}
Content-Typeapplication/json

12.54 GET Is Following by User

{host}/portfolio-follow/isFollowing/portfolio-id/

12.54.1 Headers

AuthorizationJWT {{usergeneraltoken}}
Content-Typeapplication/json

12.55 DEL Unfollow Portfolio

{host}/portfolio-follow/unfollow/

12.55.1 Headers

AuthorizationJWT {{usergeneraltoken}}
Content-Typeapplication/json

12.55.2 Body

```
raw
{
  "portfolio_id": 2
}
```

12.56 POST Portfolio Create

{host}/portfolio/createportfolio/

12.56.1 Headers

AuthorizationJWT {{usergeneraltoken}}
Content-Typeapplication/json

12.56.2 Body

```
raw
{
  "name": "my portfolio new",
  "BTC": true,
  "AAPL": true
}
```

12.57 PUT Portfolio Update

{host}/portfolio/updateportfolio/6/

12.57.1 Headers

AuthorizationJWT {{usergeneraltoken}}
Content-Typeapplication/json

12.57.2 Body

```
raw
{
  "name": "my portfolio notification",
  "ETH": true,
  "AAPL": true
}
```

12.58 DEL Portfolio Delete

{host}/portfolio/deleteportfolio/6/

12.58.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.59 GET Portfolio Retrieve

{host}/portfolio/retrieveportfolio/4/

12.59.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.60 GET Portfolio List

{host}/portfolio/listportfolio/3/

12.60.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.61 POST Notification Create

{host}/notification/createnotification/

12.61.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.61.2 Body

```
raw
{
  "owner":3,
  "text":"trect"
}
```

12.62 POST Set Notification

{host}/notification/setnotification/

12.62.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.62.2 Body

```
raw
{
  "currency": "AAPL",
  "amount": 1211.123,
  "is_bigger": false
}
```

12.63 GET List Set Notification

{host}/notification/listsetnotification/

12.63.1 Headers

AuthorizationJWT {{usergeneraltoken}}

12.64 GET List Notification

{host}/notification/listnotification/

12.64.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.64.2 Body

```
raw
{
  "owner": 3,
```

```
    "text": "ttext"  
}
```

12.65 POST Create Comment

{host}/article-comment/create/

12.65.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.65.2 Body

```
raw  
{  
  "text": "This article is awesome3!",  
  "article_id": 9  
}
```

12.66 DEL Delete Comment

{host}/article-comment/delete/

12.66.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.66.2 Body

```
raw  
{  
  "id": 2  
}
```

12.67 PUT Update Comment

{host}/article-comment/update/

12.67.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.67.2 Body

raw
{
"id": 1,
"text": "—EDIT!! This article is great!",
"article_id": 7
}

12.68 GET List Article Comments by Article Id

{host}/article-comment/list/id

12.68.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.69 POST Like Article

{host}/article-like/like/

12.69.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.69.2 Body

raw
{
"article_id": 9
}

12.70 POST Dislike Article

{host}/article-like/dislike/

12.70.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.70.2 Body

raw

```
{  
  "article_id": 9  
}
```

12.71 GET List Liked Articles by User Id

{host}/article-like/likedArticlesByUserId/userId

12.71.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.72 GET List Disliked Articles by User Id

{host}/article-like/dislikedArticlesByUserId/userId

12.72.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.73 GET List Article Likes by Article Id

{host}/article-like/likesByArticleId/articleId

12.73.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.74 GET List Article Dislikes by Article Id

{host}/article-like/dislikesByArticleId/articleId

12.74.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.75 GET Like Count by Article Id

{host}/article-like/likeCountByArticleId/3

12.75.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.76 GET Dislike Count by Article Id

{host}/article-like/dislikeCountByArticleId/articleId

12.76.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.77 GET isLikedByUser

{host}/article-like/isLikedByUser/articleId

12.77.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.78 GET isDislikedByUser

{host}/article-like/isLikedByUser/articleId

12.78.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.79 POST predict equipment

{host}/prediction/predict/

12.79.1 Headers

Content-Typeapplication/json
AuthorizationJWT {{usergeneraltoken}}

12.79.2 Body

raw
{
 "tradingEquipment": "BTC",
 "is_Rising": false
}

12.80 GET listevents

{host}/event/list/

12.81 GET searchevent

{host}/event/search/

13 Annotation API Documentation

13.1 POST Create Annotation

{host}/annotation/createannotation/

13.1.1 Headers

Content-Typeapplication/json

13.1.2 Body

```
raw
{
  "@context": "http://www.w3.org/ns/annao.aaa",
  "creator": "http://example.org/abdullah21",
  "body": [
    {
      "type": "TextualBody",
      "purpose": "tagging",
      "value": "love"
    },
    {
      "type": "Choice",
      "purpose": """",
      "value": """
    }
  ],
  "target": {
    "selector": {
      "refinedBy": {
        "type": "TextPositionSelector",
        "start": 6,
        "end": 27
      },
      "type": "FragmentSelector",
      "value": "xpointer(/doc/body/section[2]/para[1])"
    },
    "type": "Image",
    "styleClass": "mystyle",
    "source": "http://example.com/image1",
    "image_id": "http://example.com/image1xywh=100,100,300,300"
  },
  "type": "Annotation",
```

```
"motivation": "commenting",
"created": "2015-10-13T13:00:00Z"
}
```

13.2 POST Create Creator

{host}/annotation/creatorregister/

13.2.1 Headers

Content-Typeapplication/json

13.2.2 Body

```
raw
{
  "id": "http://example.org/abdullah21",
  "type": "Person",
  "name": "A. Person",
  "nickname": "abdullah"
}
```

13.2.3 GET Creator List

{host}/annotation/creatorlist/

13.2.4 GET Creator List Mobile

{host}/annotation/creatorlistmobile/

13.2.5 GET Annotation All List

{host}/annotation/annotationsalllist/

13.2.6 GET is Creator Exist

{host}/annotation/creatorexist/?id=http://example.org/abdullah21