

BOGAZICI UNIVERSITY  
DEPARTMENT OF COMPUTER  
ENGINEERING  
Project Development in Software Engineering

---

**MILESTONE REPORT 1  
(Group 1)**

---

Emirhan Sarac

Ufuk Yilmaz

Abdullah Coskun

Eray Kurtulus

Ilker Ozkan

Mete Han Kurt

Yagmur Ceren Dardagan

Mehmet Altay Ince

Irem Ustunboyacioglu

Omer F. Toptas

---



# Contents

<b>1 Executive Summary</b>	<b>4</b>
1.1 Description . . . . .	4
1.2 Project Status . . . . .	4
1.3 Future Goals . . . . .	4
<b>2 List and Status of Deliverables</b>	<b>5</b>
<b>3 Evaluation of Status of Deliverables and its impact on project</b>	<b>6</b>
3.1 Backend . . . . .	6
3.2 Frontend . . . . .	6
3.3 Android . . . . .	6
<b>4 Summary of Team Member's Work</b>	<b>7</b>
4.1 Backend . . . . .	7
4.2 Frontend . . . . .	8
4.3 Android . . . . .	10
<b>5 Communication Plan</b>	<b>12</b>
<b>6 Requirements</b>	<b>13</b>
<b>7 Project Plan</b>	<b>21</b>
<b>8 API Documentation</b>	<b>22</b>
8.1 POST trader register . . . . .	22
8.1.1 Headers . . . . .	22
8.1.2 Body . . . . .	22
8.2 POST basic register . . . . .	22
8.2.1 Headers . . . . .	22
8.2.2 Body . . . . .	22
8.3 POST login . . . . .	23
8.3.1 Headers . . . . .	23
8.3.2 Body . . . . .	23
8.4 GET auto login/profile . . . . .	23
8.4.1 Headers . . . . .	23
8.5 GET User Retrieve . . . . .	24
8.5.1 Headers . . . . .	24

8.6	PUT User Update . . . . .	24
8.6.1	Headers . . . . .	24
8.6.2	Body . . . . .	24
8.7	PUT User Password Change . . . . .	24
8.7.1	Headers . . . . .	24
8.7.2	Body . . . . .	25
8.8	POST search . . . . .	25
8.8.1	Headers . . . . .	25
8.8.2	Body . . . . .	25
8.9	POST Follow Someone With ID . . . . .	25
8.9.1	Headers . . . . .	25
8.9.2	Body . . . . .	26
8.10	GET List Follower Of Current User . . . . .	26
8.10.1	Headers . . . . .	26
8.11	GET List Follower Of Given User . . . . .	26
8.11.1	Headers . . . . .	26
8.11.2	Body . . . . .	26
8.12	GET List Following Of Current User . . . . .	27
8.12.1	Headers . . . . .	27
8.13	GET List Following Of Given User . . . . .	27
8.13.1	Headers . . . . .	27
8.13.2	Body . . . . .	27
8.14	DELETE Unfollow Someone . . . . .	27
8.14.1	Headers . . . . .	27
8.14.2	Body . . . . .	28
<b>9</b>	<b>Evaluation of tools and managing the project</b>	<b>28</b>
9.1	Tracking Tools(GitHub) . . . . .	28
9.1.1	Git Feature of Github . . . . .	28
9.1.2	Issues . . . . .	28
9.1.3	Pull Requests . . . . .	28
9.1.4	Project section of the Github . . . . .	29
9.1.5	Wiki . . . . .	29
9.2	Backend . . . . .	29
9.2.1	Python Django . . . . .	29
9.2.2	Postman . . . . .	29
9.2.3	Docker . . . . .	29
9.2.4	PyCharm . . . . .	30

9.3	Web . . . . .	30
9.3.1	ReactJS . . . . .	30
9.3.2	Material-ui library . . . . .	30
9.3.3	Axios . . . . .	31
9.3.4	history . . . . .	31
9.3.5	react-location-picker . . . . .	31
9.3.6	ESLint . . . . .	31
9.3.7	Webstorm IDE . . . . .	31
9.4	Android . . . . .	32
9.4.1	Kotlin for Android . . . . .	32
9.4.2	Retrofit . . . . .	32
9.4.3	IDE and Simulation Preferences . . . . .	32
<b>10</b>	<b>UI Design</b>	<b>33</b>
10.1	Web . . . . .	33
10.2	Android . . . . .	37
<b>11</b>	<b>User scenarios</b>	<b>46</b>
11.1	Web . . . . .	46
11.2	Android . . . . .	48

# **1 Executive Summary**

## **1.1 Description**

Our Project, TradersPlatform, is a social platform for people 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.

## **1.2 Project Status**

Most of the functionalities in the backend for this milestone have been completed ahead of the other parts. Register, session control, log out, login, profile page, user follow list retrieval, follow, editing credentials have been completed both in backend and Android part. Android part has completed almost all functionalities but it has fallen a bit behind in the UI design part. Currently in Android part, register as trader is not available. On contrary, web part has fallen behind in the functionality, but the design of web is progressing coherently with functionality part.

## **1.3 Future Goals**

For the upcoming milestone 2 and our goals, we are planning to complete more functionalities such as Home Page, completing register as trader at Android side, register through Gmail, location retrieval, upgrade from basic to trader, trading equipment pages, events, and notification functionalities. There were some synchronization problems for this milestone between frontend and Android that we have completed different part of functionalities. Also, request and response types from backend should be more generic, we will assure that in the upcoming milestone. Since this was the first milestone and we were doing most of the tasks first time, it will be easier and faster to complete further functionalities.

## 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
✓	Project Plan	Oct 18, 19
<b>Backend</b>		
✓	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
<b>Frontend</b>		
✓	Homepage	Oct 15, 19
✓	User Sign Up and Login View	Oct 15, 19
✓	User Profile View	Oct 22, 19
<b>Android</b>		
✓	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

### **3 Evaluation of Status of Deliverables and its impact on project**

#### **3.1 Backend**

Sign Up and Login View, Profile, Profile Update, Email Notification, Follow, Unfollow, List Followers and Followings (both user himself/herself and to another person), Search, Password Update functionalities has completed in the Backend. Sign Up has been completed except 1.1.1.1.2, 1.1.1.2.2. Password change mechanism in 1.1.1.3 developed with all features. 1.2.6 Search mechanism is implemented in a beginner level but open to development. The Follow mechanism has fully implemented in the backend which is in 1.1.1.5.7 in requirements. In general, in the backend part of the project, we believe we concluded more than sufficient part of this project.

#### **3.2 Frontend**

We were able to meet the specifications that we planned for the first milestone at the beginning of the semester. We implemented the sign-up page for basic and trader user and login page. We've improved the interface with some animations. We also designed and build profile page of the users. We add validations to sign-up and login form submission. We add location picking feature from google maps to the sign-up page. We also implemented profile page. Up to now, we successfully implemented all the features specified in the project plan for the first milestone and spent the most of the time to learn front-end programming since we had no earlier knowledge.

#### **3.3 Android**

As shown in the section 2; Sign Up and Login View, Profile, Profile Update, Follow, Search, Password Update functionalities has completed in the Android side. 1.1.1.1. Sign Up has been completed except 1.1.1.1.1.3, 1.1.1.1.1.2, 1.1.1.1.2, 1.1.1.2.2. However, they are all implemented, in both backend and android part, only integration problem exist right now. 1.1.1.3.1 Reset Password requirements have been completed. 1.1.1.5.7. Follow, 1.2.3. Interface, Most basic version of 1.2.6. Search Mechanism has been completed.

Up to this Milestone, we have designed and go over the requirements to match required parts. We will do that for next milestone. However, we have

missed some exhausting requirements and postponed them to next milestone. Overall, we have succeeded enough for this milestone.

## 4 Summary of Team Member's Work

### 4.1 Backend

Team Members	Contributions
Ömer Faruk Toptas	As a responsible group member I have participated in every group meeting and I have joined every decision making process also I have took notes of some group meetings.
Abdullah Coşkun	At first I commit initial django project with only user register mechanism. Then I extend the generic user in Django and create a register mechanism for this user type. Then I add group for every user to understand if they are trader or basic user and according to url, register mechanism categorize them. After that, I develop an email notification mechanism. After that I developed user update and password change mechanism and I arranged an authentication system that only user itself can change his/her password. Then I create a user retrieve url so users can view other people's profile page. Then I create a simple search mechanism which takes a string and returns users if their username contains that string. Then I create follow mechanism which allows users to follow or unfollow each other, see their followers and followings and finally see other people's followers and followings. Also in general I create an authentication mechanism that, allows to operate this features only if they have permission to operate. I create postman collection to provide mobile and front end, information about requests that I create. Lastly I wrote API documentation in Milestone Report 1.

<b>Team Members</b>	<b>Contributions</b>
İrem Üstünboyacıoğlu	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. 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 searched how tests are written in Django and I started to add the missing tests. I regularly updated issues, reviewed pull-requests. Finally, I added the Project Plan, List and Status of Deliverables and Evaluation of Tools for Backend to Milestone Report 1.
İlker Özkan	Contributions

## 4.2 Frontend

<b>Team Members</b>	<b>Contributions</b>
Mete Han Kurt	I attended all group meetings and recorded the meeting notes on a regular basis. At the beginning of the period, I wanted to be part of the frontend team in my team and it was decided that we would use react.js for the frontend. I first created a registration page and made arrangements according to the project requirements. I've made the user interface appeal to the eye by adding a stylish visual to the background and adding custom icons for each field. In addition, I have checked some redirects and error or confirmation messages. Finally, I created a profile page and edited it visually. I applied the information sent by Backend to related fields. In general, I did my best to contribute to the project. In addition, in the milestone report, I helped to write the Evaluation of the Status of Deliverables and its Impact on Project, updated the requirements again, and revised and re-uploaded the project plan.

<b>Team Members</b>	<b>Contributions</b>
Mehmet Altay İnce	Implementing home page and location picker with google maps on sign in page.
Ufuk Yilmaz	From the start of the semester, I participated in all of the group meetings and front-end meetings and presented my ideas about the progress of the project. I added issue template to our repository. In the beginning of the semester, as a front-end team none of us had enough knowledge about front-end. In the first weeks, I made research about front-end programming to decide which tools to use and find reference materials. I found a UI Kit with a fresh design that we used while implementing our pages. Our teammates and I accelerated the process of learning front-end programming by helping each other and also we tried to synchronize with other teams by giving feedback. I implemented the pages(Login page, Sign-up Page, Profile Page) with other teammates in the front-end team. I implemented the logics that taking the input from form, validating form, passing properties and states to the other pages, authentication and sending requests to back-end. While implementing these logics, I sent feedback to back-end. e.g. Since the CORS configuration is not setup correctly, the browser console was presented a CORS error. I deployed the front-end application to the Amazon Web Server. In the milestone report, I wrote the sections that the front-end part of the evaluation of status of deliverables, evaluation of the tools we used in web and the evaluation of the GitHub and added the design pages of web applications.

### 4.3 Android

Android Team Members	Contributions
Emirhan Saraç	<p>As a team communicator of the team, I have attended all group and Android meetings. I have tried to distribute tasks in Android equally among members and tried to organize progress phase of the project. I have worked on brute-force design of Login and Register Pages. I have also designed UI parts of Profile Page, Profile Editing Page, Search, and User follow-following list. I have used Retrofit library to achieve request and response operations in Android. I have used this tool in Login, Register, Profile Page, Search, Editing Profile Page, User follow-following list. After completing request-response, corresponding operations has been handled in the regarding page for all previously mentioned pages. For Milestone Report 1, I have completed all parts of Executive Summary and Evaluation of Status of Deliverables-Android part.</p>
Yağmur Ceren Dardağan	<p>As a member of this team and Android team, I have joined all the group meetings and Android meetings. I have updated requirements of the project and participated in forming the project plan for this semester. I have also created skill/interest chart to decide on the languages for frontend, backend and android. After starting to develop android, I have designed the user interfaces. In addition to that, I have implemented two-phased registration part, first phase for basic user registration and the second phase for trader user registration in case the user wants to register as trader. After completing registration part, I have designed the appearance of home page and specified all the necessary link names to navigate other pages. I have added top action bar to navigate profile, settings and logout options. Finally, I have attached revised requirements, user scenarios and the evaluation of tools part for android in Milestone Report 1.</p>

Team Members	Contributions
Eray Kurtulus	<p>I have attended the group meetings and did my best to contribute to the planning of the project. On the Android part, I have integrated the Google Authentication and Google Location services into the Android app. The Google Authentication service enables users to sign up and sign in using their Google accounts, not forcing them to remember another password for another application. I have also used the Retrofit library to send the necessary requests to the backend endpoints for Google Authentication. The Google Location service is implemented on the registration page and automatically fetches the street address of the user, by retrieving the geometric coordinates from the GPS of the device and applying reverse geocoding to them on a parallel thread (Intent Service). The use of threads improves the responsiveness of the app. The app also asks for location permission on runtime. If the GPS is turned off, the user is informed with a toast and a listener is activated so that the location can be retrieved if the user enables GPS. In this milestone report, I have added the screenshots of the Android app.</p>

## 5 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 1: Communication Plan

## 6 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.
- **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.
          - **1.1.1.1.1.2.** Users shall verify their accounts by following steps in the mail sent to the user e-mail address.

- **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.
  - **1.1.1.3.2.** Forget Password
    - **1.1.1.3.2.1.** Users shall be able to request “Forget Password?” operation when users forget their password at sign in page.
    - **1.1.1.3.2.2.** Users should provide e-mail address.
    - **1.1.1.3.2.3.** Users shall get e-mail that contains randomly generated new password from system.
- **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.
  - **1.1.1.5.5.1.** Users shall be able to comment about predictions made by other users.
- **1.1.1.5.6.** Users shall be able to comment about any trading equipment.
- **1.1.1.5.7.** Users shall be able to follow other users.
- **1.1.1.5.8.** Users shall be able to send private messages to other users.
- **1.1.1.5.9.** Users shall be able to set alerts for certain trading equipment.
  - **1.1.1.5.9.1.** Users shall be able to set alerts for above or below of specific value on the trading equipment.
  - **1.1.1.5.9.2.** Users shall be able to set alerts for increase or decrease of specific change on the trading equipment.
- **1.1.1.5.10.** Users shall be able to see their search history.

\* **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** Nasdaq 100
      - **1.2.5.1.1.2** Dow 30
      - **1.2.5.1.1.3** SmallCap 2000
    - **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 Russell 200 (IWM)
      - **1.2.5.1.3.3** Invesco QQQ (QQQ)
    - **1.2.5.1.4.** Commodities
      - **1.2.5.1.4.1** Gold
      - **1.2.5.1.4.2** Silver
      - **1.2.5.1.4.3** Copper
    - **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.** System shall have currency converter tool.

- **1.2.5.1.5.4.** Examples :
  - **1.2.5.1.5.4.1** Pound
  - **1.2.5.1.5.4.2** Euro
  - **1.2.5.1.5.4.3** Dollar
- **1.2.5.1.6.** Bonds
  - **1.2.5.1.6.1** Municipal Bond Mutual Funds
  - **1.2.5.1.6.2** International Bond Mutual Funds
  - **1.2.5.1.6.3** Investment Grade Corporate Bond Mutual Funds
- **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 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.

## 7 Project Plan

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		Backend	24 days?	9/19/19 8:00 AM	10/22/19 5:00 PM		
2		Database Tables For Users	6 days?	9/19/19 8:00 AM	9/26/19 5:00 PM		Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
3		Sign Up	3.5 days?	9/27/19 8:00 AM	10/2/19 1:00 PM	2	Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
4		Sign In	3.5 days?	10/2/19 1:00 PM	10/7/19 5:00 PM	2;3	Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
5		Password Change	3.5 days?	10/8/19 8:00 AM	10/11/19 1:00 PM	4	Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
6		Sign Up Via Google Account	6 days?	10/8/19 8:00 AM	10/15/19 5:00 PM	3	Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
7		Privacy and Security Const.	11 days?	10/8/19 8:00 AM	10/22/19 5:00 PM		
8		User Profiles	11 days?	10/8/19 8:00 AM	10/22/19 5:00 PM	2;4	Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
9		User Actions - Follow	6 days?	10/15/19 8:00 AM	10/22/19 5:00 PM		Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
10		Frontend	24 days?	9/19/19 8:00 AM	10/22/19 5:00 PM		
11		Sign Up	10 days?	9/19/19 8:00 AM	10/2/19 5:00 PM		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
12		Sign In	10 days?	9/19/19 8:00 AM	10/2/19 5:00 PM		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
13		Sign Up Via Google Account	6 days?	10/8/19 8:00 AM	10/15/19 5:00 PM	11	Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
14		User Profiles	11 days?	10/8/19 8:00 AM	10/22/19 5:00 PM		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
15		User Actions - Follow	6 days?	10/15/19 8:00 AM	10/22/19 5:00 PM		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
16		Android	24 days?	9/19/19 8:00 AM	10/22/19 5:00 PM		
17		Sign Up	10 days?	9/19/19 8:00 AM	10/2/19 5:00 PM		Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
18		Sign In	10 days?	9/19/19 8:00 AM	10/2/19 5:00 PM		Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
19		Password Change	4.667 days?	10/3/19 8:00 AM	10/9/19 2:20 PM	18	Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
20		Sign Up Via Google Account	6 days?	10/8/19 8:00 AM	10/15/19 5:00 PM	17	Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
21		User Profiles	10 days?	10/9/19 8:00 AM	10/22/19 5:00 PM		Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
22		User Actions - Follow	6 days?	10/15/19 8:00 AM	10/22/19 5:00 PM		Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
23		Milestone 1	1 day?	10/22/19 8:00 AM	10/22/19 5:00 PM		All
24		Backend	25 days?	10/23/19 8:00 AM	11/26/19 5:00 PM		
25		User Actions	11 days?	10/23/19 8:00 AM	11/6/19 5:00 PM		Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
26		Transaction	11 days?	10/23/19 8:00 AM	11/6/19 5:00 PM		Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
27		Notification	6 days?	11/6/19 8:00 AM	11/13/19 5:00 PM		Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
28		Trading Equipments	6 days?	11/13/19 8:00 AM	11/20/19 5:00 PM		Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
29		Search Mechanism	5 days?	11/20/19 8:00 AM	11/26/19 5:00 PM		Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
30		Frontend	25 days?	10/23/19 8:00 AM	11/26/19 5:00 PM		
31		User Actions	11 days?	10/23/19 8:00 AM	11/6/19 5:00 PM		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
32		Database Tables	11 days?	10/23/19 8:00 AM	11/6/19 5:00 PM		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
33		Transaction	11 days?	10/23/19 8:00 AM	11/6/19 5:00 PM		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
34		Notification	6 days?	11/6/19 8:00 AM	11/13/19 5:00 PM		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
35		Trading Equipments	6 days?	11/13/19 8:00 AM	11/20/19 5:00 PM		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
36		Search Mechanism	5 days?	11/20/19 8:00 AM	11/26/19 5:00 PM		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
37		Android	25 days?	10/23/19 8:00 AM	11/26/19 5:00 PM		
38		User Actions	11 days?	10/23/19 8:00 AM	11/6/19 5:00 PM		Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
39		Database Tables	11 days?	10/23/19 8:00 AM	11/6/19 5:00 PM		Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
40		Transaction	11 days?	10/23/19 8:00 AM	11/6/19 5:00 PM		Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
41		Notification	6 days?	11/6/19 8:00 AM	11/13/19 5:00 PM		Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
42		Trading Equipments	6 days?	11/13/19 8:00 AM	11/20/19 5:00 PM		Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
43		Search Mechanism	5 days?	11/20/19 8:00 AM	11/26/19 5:00 PM		Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
44		Milestone 2	1 day?	10/22/19 8:00 AM	10/22/19 5:00 PM		All
45		Backend	15 days?	11/27/19 8:00 AM	12/17/19 5:00 PM		
46		Annotation	11 days?	11/27/19 8:00 AM	12/11/19 5:00 PM		Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
47		Performance Optimization	5 days?	12/11/19 8:00 AM	12/17/19 5:00 PM		Abdullah Coşkun;Ömer Faruk Toptaş;İlker Özkan;İrem Üstünboyacıoğlu
48		Frontend	15 days?	11/27/19 8:00 AM	12/17/19 5:00 PM		
49		Annotation	11 days?	11/27/19 8:00 AM	12/11/19 5:00 PM		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
50		Performance Optimization	5 days?	12/11/19 8:00 AM	12/17/19 5:00 PM		Mehmet Altay İnce;Mete Han Kurt;Ufuk Yılmaz
51		Android	15 days?	11/27/19 8:00 AM	12/17/19 5:00 PM		
52		Annotation	11 days?	11/27/19 8:00 AM	12/11/19 5:00 PM		Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
53		Performance Optimization	5 days?	12/11/19 8:00 AM	12/17/19 5:00 PM		Emirhan Saracı;Eray Kurtuluş;Yağmur Ceren Dardağan
54		End-to-End Tests	3 days?	12/14/19 8:00 AM	12/18/19 5:00 PM		All
55		Delivery	1 day?	10/22/19 8:00 AM	10/22/19 5:00 PM		All

## 8 API Documentation

### 8.1 POST trader register

{host}/user/registertrader/

#### 8.1.1 Headers

Content-Type application/json

#### 8.1.2 Body

```
raw
{
  "username": "melike_",
  "password": "12qwas.1?",
  "email": "melike.sonmez@boun.edu.tr",
  "first_name": "melike",
  "last_name": "sonmez",
  "location": "İstanbul",
  "iban_number": "1111111111131211
}
```

### 8.2 POST basic register

{host}/user/registerbasic/

#### 8.2.1 Headers

Content-Type application/json

#### 8.2.2 Body

```
raw
{
```

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

### 8.3 POST login

{host}/user/login/

#### 8.3.1 Headers

Content-Type application/json

#### 8.3.2 Body

raw

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

### 8.4 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.

#### 8.4.1 Headers

Authorization JWT {token}

## **8.5 GET User Retrieve**

{host}/user/retrieve/3/

### **8.5.1 Headers**

Authorization JWT {token}

## **8.6 PUT User Update**

{host}/user/updateuser/

### **8.6.1 Headers**

Content-Type application/json  
Authorization JWT {token}

### **8.6.2 Body**

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

## **8.7 PUT User Password Change**

{host}/user/updatepass/

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

### **8.7.1 Headers**

Content-Type application/json  
Authorization JWT {token}

### **8.7.2 Body**

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

## **8.8 POST search**

{host}/user/search\_user/

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

### **8.8.1 Headers**

Content-Type application/json

Authorization JWT {token}

### **8.8.2 Body**

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

## **8.9 POST Follow Someone With ID**

{host}/follow/follow/

With this request current user can follow another person.

### **8.9.1 Headers**

Content-Type application/json

Authorization JWT {token}

### **8.9.2 Body**

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

## **8.10 GET List Follower Of Current User**

{host}/follow/listFollower/

### **8.10.1 Headers**

Content-Type application/json  
Authorization JWT {token}

## **8.11 GET List Follower Of Given User**

{host}/follow/listFollowerWithId/

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

### **8.11.1 Headers**

Content-Type application/json  
Authorization JWT {token}

### **8.11.2 Body**

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

## **8.12 GET List Following Of Current User**

{host}/follow/listFollowing/

### **8.12.1 Headers**

Content-Type application/json

Authorization JWT {token}

## **8.13 GET List Following Of Given User**

{host}/follow/listFollowingWithId/

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

### **8.13.1 Headers**

Content-Type application/json

Authorization JWT {token}

### **8.13.2 Body**

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

## **8.14 DELETE Unfollow Someone**

{host}/follow/delete/

### **8.14.1 Headers**

Content-Type application/json

Authorization JWT {token}

#### **8.14.2 Body**

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

For further information about our API here.

## **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.

### **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.

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

#### 9.2.4 PyCharm

As IDE, PyCharm is chosen since it is used 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.

### **9.3.6 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.

### **9.3.7 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.

## 9.4 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 UI Design

## 10.1 Web

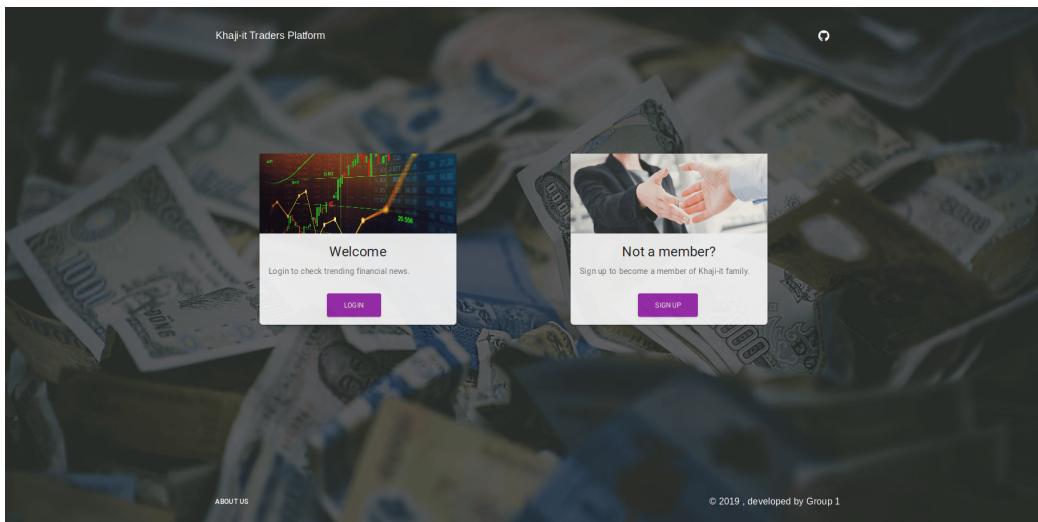


Figure 2: Homepage

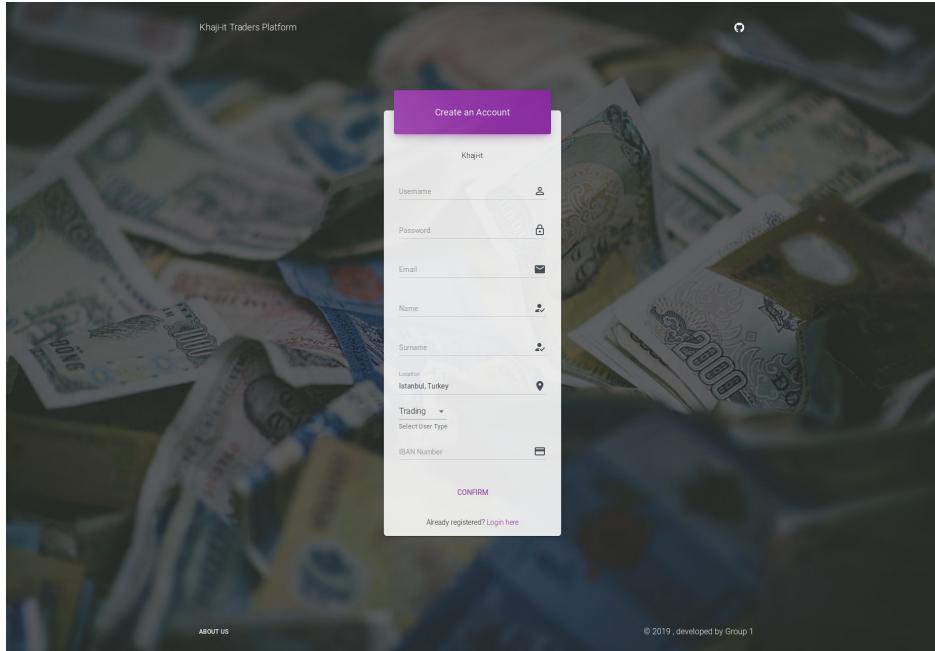


Figure 3: Register Page as a Trader User

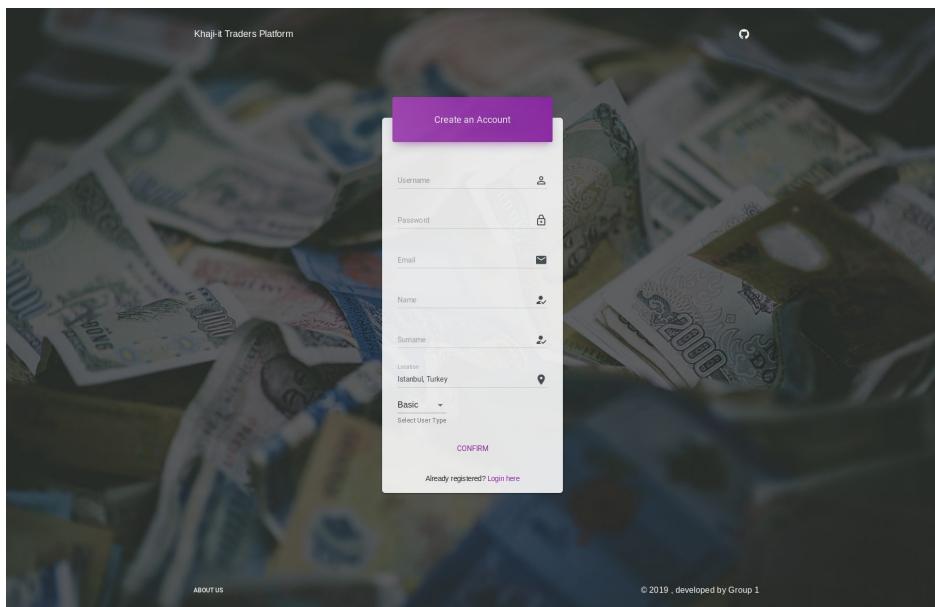


Figure 4: Register Page as a Basic User

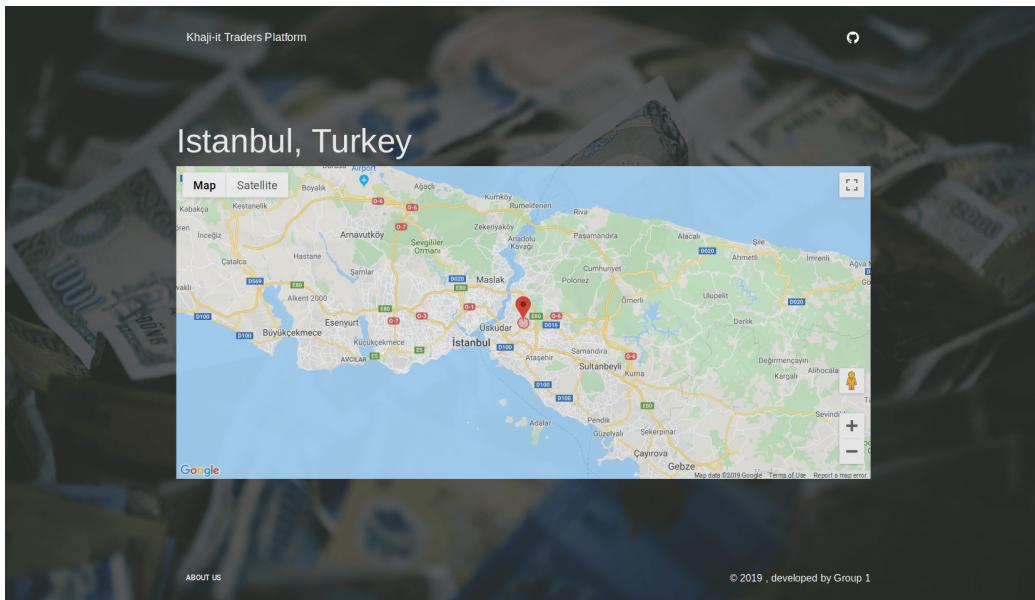


Figure 5: Location Selection Page with Google Maps

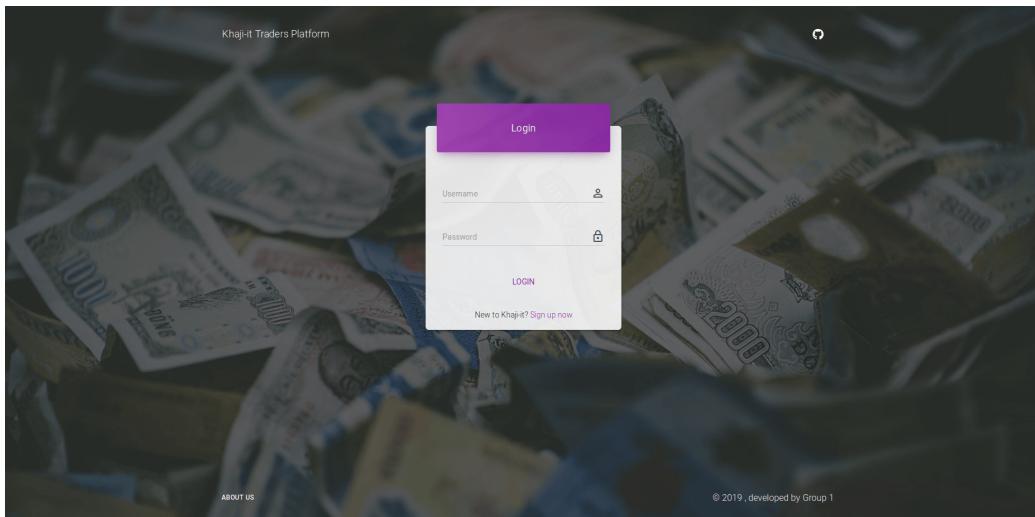


Figure 6: Login Page

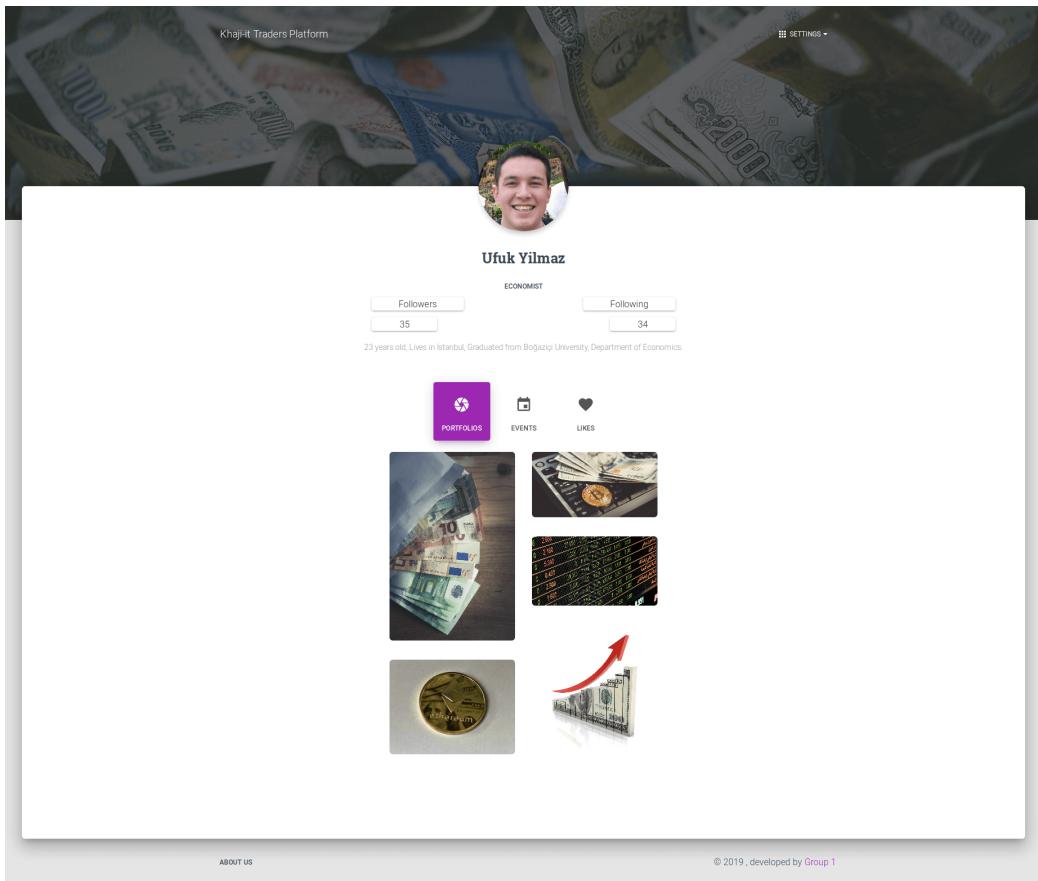


Figure 7: Profile Page

## 10.2 Android



Figure 8: Main Page

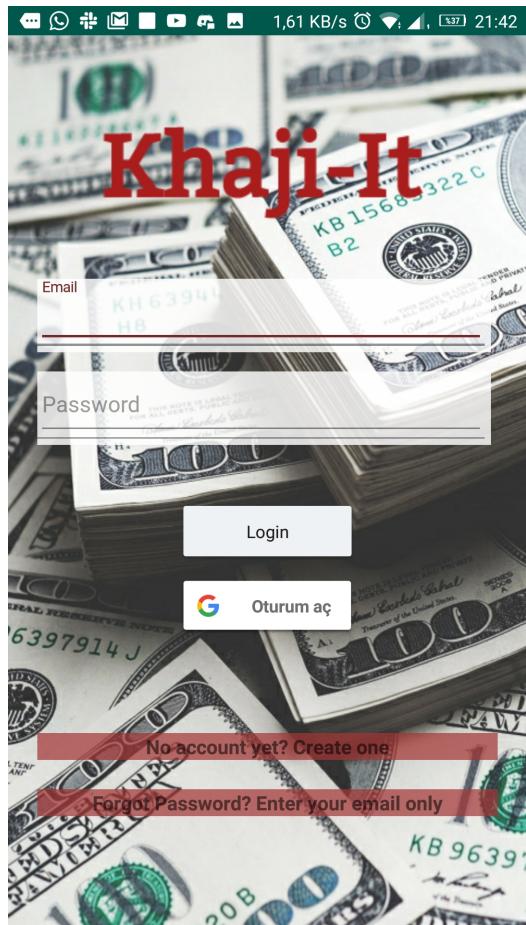


Figure 9: Login Page

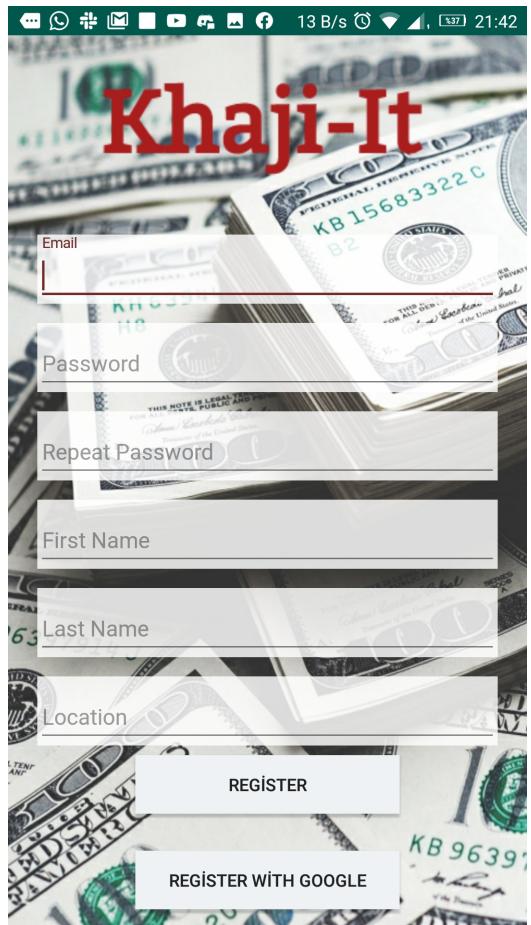


Figure 10: Register Page

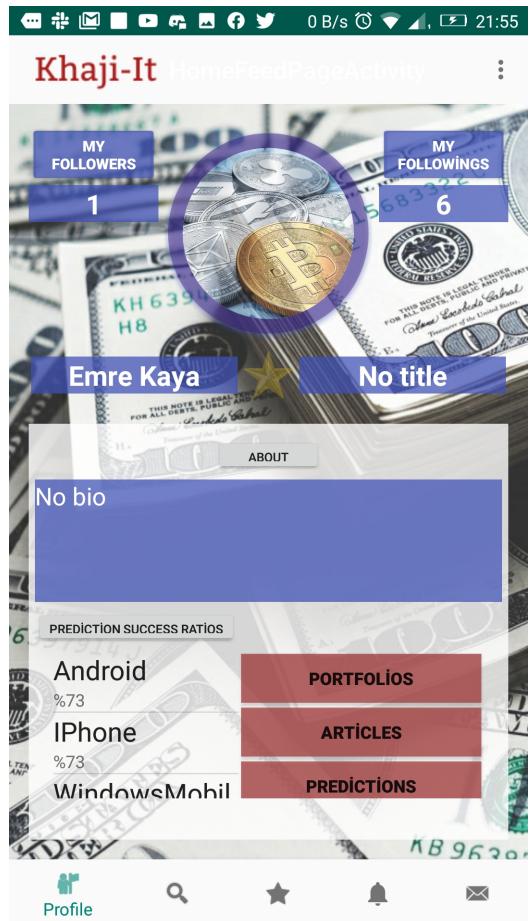


Figure 11: Profile Page



Figure 12: Search Page

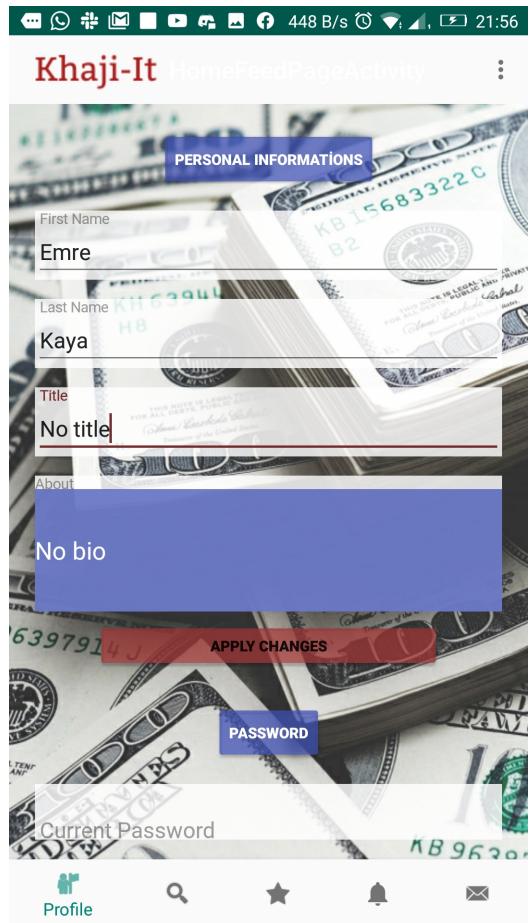


Figure 13: Edit Profile

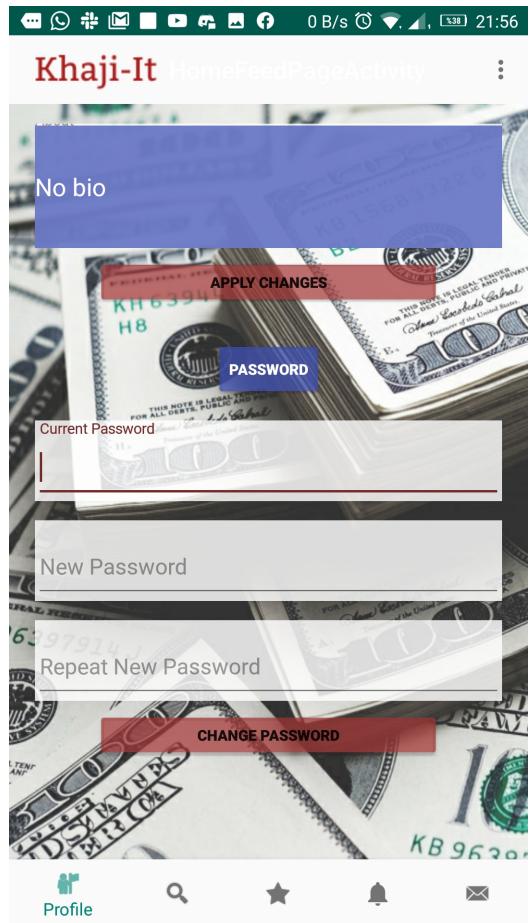


Figure 14: Change Password



Figure 15: Follower List



Figure 16: Following List

# 11 User scenarios

## 11.1 Web

### Persona

- Arda Sayar
- 23 years old
- Lives in İstanbul
- IBAN: TR560006200000012990022604
- TC: 82138333922
- Graduated from Boğaziçi University, Department of Economics.

### Goals

- To follow people known to be successful in the economy.
- To learn most recently used trading equipment.
- To chase economic events in a compact platform.
- To learn how to be a good trader before investing in.

### User Story

Arda is a new graduated economist at Accenture. In addition to his business, he is looking for a useful platform where he can quickly trade, track fluctuations for cryptocurrencies in the market and follow experienced economists to learn how to earn extra income by investing. He also wants to use his economy information which he learned at the university. He hears about trader's platform from his friends and starts to look up cryptocurrencies. His main goal is following successful investors then making his own investments for extra income

### Scenario

Arda enters the website for the first time and admires the design. He wants

to have a quick look at the content, follow the experienced economists, but needs to create an account. Thanks to the easy-to-use homepage, he can go to the registration screen and be a member by entering the information one by one. When he enters the required information, he notices that the user type are basic and trading, but since he is new to the platform, he prefers to be the basic user first and registers in seconds. After the registration, click the login tab to go to the login screen. He tries to log in quickly by entering his email and password to see the content, but he notices that he entered the password incorrectly and tries again carefully. And he is now successfully logged in and redirected to the profile page. On the profile page he can see his avatar, portfolios and the articles he follows or likes.

## Acceptance Criteria

- 1.1.1.1. Sign Up
  - 1.1.1.1.1. Basic users shall sign up by providing their username, password, name, surname, e-mail, and location.
  - 1.1.1.1.1. Trading users shall sign up by providing their username, password, name, surname, e-mail, location, Iban and TC.
    - \* 1.1.1.1.1.1. Usernames and e-mail addresses shall be unique.
    - \* 1.1.1.1.1.2. User passwords shall be in the specified form.
- 1.1.1.2. Sign In
  - 1.1.1.2.1. Users shall sign in by providing their email and password.
  - 1.1.1.2.2. Users shall sign in via their Google account.
- 1.1.1.4. Profile
  - 1.1.1.4.2. Users shall have at least one portfolio as default.
    - \* 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.

## 11.2 Android

### Persona

- Melike Sönmez
- 37 years old
- Lives in İstanbul
- IBAN: TR4958649586495864
- Recruiter at Deloitte.

### Goals

- To follow the most recent economic events.
- To look for job seekers and potential people for her company.
- To track trading equipments daily.

### User Story

Melike is a recruiter at Deloitte for 2 years and she was asked to find a new employee for the finance department of the company. In order to search for potential people for this position, she needs a platform so that she can find people who are interested in finance. She also wants to find a platform to track economic events and situations of trading equipments daily. She hears about trader's platform Khaj-it from a friend who suggests Emre for the position so that she can check this person's profile from the platform.

### Scenario

Melike enters the mobile application for the first time and she understands that she can register or continue as guest thanks to simple design of the application. However she wants to register to platform and she clicks "Register" button. Then, she easily fills email, password, repeat password, first name and last name fields without being hesitate, since these information are asked during registration all the time. After clicking register button, she is

surprised to see an additional IBAN number field. However, after reading two buttons below the IBAN field, "register as trader" and "skip this step and continue as basic user" respectively, she understands that IBAN information is obligatory for trader users and she chooses to continue as basic user. After clicking the button, application redirects her to main page and she clicks to login button to enter the application. After typing her e-mail and password, the app navigates her to profile page. To customize her profile, she clicks settings button and changes her "About" field and types that she seeks for potential employee's for the finance department of Deloitte. After editing her profile, she recognizes search button on the bottom navigation page and remembers the person that was recommended the position by her friend. She clicks the search page and she types the Emre's e-mail address that her friend gave and profile of Emre appears on the screen and she clicks to the Emre's name to visit his profile page. On the profile page, she can see the followers and followings of Emre, his picture, information about him and portfolios, articles and predictions that he has created.

## Acceptance Criteria

- 1.1.1.1. Sign Up
  - 1.1.1.1.1. Basic users shall sign up by providing their username, password, name, surname, e-mail, and location.
  - 1.1.1.1.1. Trading users shall sign up by providing their user-name, password, name, surname, e-mail, location, Iban and TC.
    - \* 1.1.1.1.1.1. Usernames and e-mail addresses shall be unique.
    - \* 1.1.1.1.1.2. User passwords shall be in the specified form.
- 1.1.1.2. Sign In
  - 1.1.1.2.1. Users shall sign in by providing their email and pass-word.
- 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.
  - 1.1.1.4.2.1. Users shall be able to rename their portfolio.
- 1.1.1.5. User actions
  - 1.1.1.5.7. Users shall be able to follow other users.
- 1.2.6.1. The system shall support search with keywords.