CMPE451 - Milestone 1 Report

Group Members

Sadullah Gültekin
Enes Turan Özcan
Burak Yüksel
Irmak Güzey
Baran Deniz Korkmaz
Barış Ege Sevgili
Burak İkan Yıldız
Fatih Mustafa Kurt
Mustafa Alparsan

Fall , 2019

Contents

1 Executive Summary 3

1.1 Introduction 3

1.2 Work done so far 3

1.3 Road ahead 3

2 List and Status of Deliverables 4

3 Evaluation of the Status of Deliverables 5

3.1 Project Plan 5

3.2 Authentication Endpoint 5

3.3 User Follow Endpoint 5

3.4 User Profile Endpoint 5

3.5 Authentication Interface in Android 6

3.6 Authentication Interface in Web Frontend 6

3.7 User Profile Page in Web Frontend 6

3.8 Home Page in Web Frontend 6

4 Summary of Work done 7

5 Requirements 8

5.1 Functional Requirements 8

5.1.1 User Requirements 8

5.1.2 System Requirements 9

5.2 Non-functional Requirements 10

5.2.1 Accessibility and Availability 10

5.2.2 Annotatitons 10

5.2.3 Performance 10

5.2.4 Privacy 10

5.2.5 Security 10

6 API Documentation 11

7 Project Plan 14

8 User scenarios 16

9 Code Structure 17

10 Evaluation Of Tools And Managing The Project 18

1 Executive Summary

1.1 Introduction

The Trader Platform aims to build a social platform in which the users can follow market of stocks, indices, currencies, bonds, funds, and cryptocurrencies and trade the equipments. The platform provides a functional interface with various type of interactions targetting to maximize the communication, since our main purpose is to present a useful platform for the benefits of the customers.

1.2 Work done so far

In the last semester we made all the designing process including determining the requirements, creating user scenarios, mock-ups and UML diagrams. Although we did lots of design, we didn't start to write the code itself. By all of these, every team member became very familiar with the project. In this semester we start by making a division of labod and dividing the team into three group which are backend, web and mobile teams. Backend team members are Enes Turan Özcan, Burak İkan Yıldız and Barış Ege Sevgili. Web team members are Sadullah Gültekin, Burak Yüksel and Irmak Güzey. Mobile team members are Fatih Mustafa Kurt, Mustafa Alparslan and Baran Deniz Korkmaz. Every team selected which technologies to use in their implementation. Until first milestone, we planned to write the authentication part of the project because its the most essential functionality. Because we were able to finish authentication part in the web earlier than we thought, we also started to implement user profile page. In the android we made only login and register functionalities and cannot implement the profile page yet. The reason is our team was inexperienced with mobile platform and it took considerable amount of time to learn how it is working. Yet, we are planning to implement profile page as soon as possible. In the backend, we implemented the base for our project. We created our database in its initial form. Also, we wrote some endpoints that are used by web and mobile teams. Detailed information can be found in our github page.

1.3 Road ahead

Next step is implementing the parts that are much more related with economic concepts. As a first step we will implement user system. Currently, we are able to see only our own profile page which is inconvenient because if we can not see other users page, we cannot follow them which prevents us to use the basics of our system. Then we will implement trading equipment parts which will make users able to buy, sell and invest to different goods.

For now our team is good at cooperation. We try to divide the labor between each member. Probably, no essential change will be done after the milestone.

2 List and Status of Deliverables

#	Deliverable	Due Date	Status	Explanations
1	Project Plan	29.09.2019	+	The road map of the practise app is drawn -including the example tutorial application-
2	Authentication Endpoint	10.10.2029	+	The endpoint for login and registration is implemented in the API.
3	User Follow Endpoint	25.10.2029	+	The endpoint for following users is implemented in the API.
4	User Profile Endpoint	25.10.2029	+	The endpoint for getting into the profil page is implemented in the API.
5	Authentication Interface in Android	15.10.2029	+	The interface for login and registration is implemented in Android.
6	Authentication Interface in Web Frontend	15.10.2029	+	The interface for login and registration is implemented in Web Frontend.
7	User Profile Page in Web Frontend	22.10.2029	+	The user profile page is implemented in Web Frontend.
8	Home Page in Web Frontend	22.10.2029	+	Home page is implemented in Web Frontend.

3 Evaluation of the Status of Deliverables

3.1 Project Plan

In order to have our project plan done we needed to split our team into three groups; Web Frontend team, Android team and Backend team. Our first plan was to have the Backend team start working as soon as possible so that our frontend teams would be able to implement the functionalities properly. So the approach was to have the endpoints done as soon as possible. And then for the first month our plan was to have the authentication and the user profile parts of the project done.
Now the next step is to have the trading equipments related features done, such as buying/selling trading equipments, making predictions and creating portfolios and etc... And while having these parts done backend endpoints should again be working in the future features which are the user functions such as creating events, articles, news and more features that makes the platform a social platform. After the frontend and android parts catching up with backend, our project will be done more or less.
After these basic and main features done, there will be a need for little modifications and then the Milestone 2 and the Final Delivery will arrive.

3.2 Authentication Endpoint

Authentication endpoints was the most urgent and the most important deliverable in our project until the first milestone. And the reason for that is that without this endpoint, implementations in both of the interfaces would disfunction.
The importance of this feature is also the reason why the deadline for this feature is specifically earlier than the others.
In this endpoints there are two basic links to ask for request; signup and login endpoints. Login link takes username, password and Google token as the data (Google Token is used if wanted but it is not necessary.). And the signup and signup/confirm endpoints accepts only a string message as the data. And does the proper checks on the backend. And there is also one logout endpoint which accepts also a string message and does the necessary job on the backend.

3.3 User Follow Endpoint

The user follow endpoints enable us to provide one of our main goals: To provide users an interactive platform. Since the modern economies are more dynamical structures, the knowledge should be spread so that many can follow and take actions. In our platform, as can be seen in most modern social media platforms, we want users to follow each other so that they can follow what is happening around and take their actions. The backend system provides the follow endpoints for this purpose with the following functionalities:

- A user can follow/unfollow the other users.
- The users can see the list of their followers, or the users they followed.
- The users can see the number of the users following themselves, or the users they followed.

The operations will be carried out succesfully after the verifications of the requests have been approved by our backend system.

3.4 User Profile Endpoint

The user profile endpoint is one of the most important endpoint since the one of the most basic functionalities of our platform is to present an interface such that the users can easily follow their actions on the economic market. The user profile endpoints present plenty of features. They contain the following functionalities:

- Providing the profile page of the user.
- Providing the profile pages of other users.
- Refreshing the page for a user.
- Setting the profile preferences, that is to set the profile as public or private.

The operations will be carried out successfully after the verifications of the requests have been approved by our backend system.

3.5 Authentication Interface in Android

There are three pages existing in the authentication interface in Android: Login, Sign-Up, and Forgot Password pages. A simple template first created for the Login page aiming at a simple and useful interface for the users. After a division of labor, the Android team members completed the design of the other pages accordingly.

After the design of the pages have been completed and the pages connected each other succesfully, the implementation of the functionalities have been tried to complete. The registration functionalities have been completed succesfully and the authentication verifications such as valid e-mail, password, iban checks have been implemented. However, due to some external conditions, some functionalities have remained incomplete. These implementations will later be implemented as soon as possible.

3.6 Authentication Interface in Web Frontend

There are two pages existing in the authentication interface: Login and Registration pages. A simple template was used in Web Frontend implementation with a major modification in all of the deliverables.

Login page was in the template. But the check was made with a hard coded mock user in the template and it wasn't functioning. The authentication needed to be connected with the API. And this was done by connecting the whole system to our server and modificating the checking system. And the design of the login page was modified according to the home page afterwards.

There was no registration page at the template. Registration page was created from scratch by using the components at the template. Username, email address and a password is taken from the user, validity of them is checked and then sent to the backend. Design and connection to the backend was made after html, javascript and css files are written.

3.7 User Profile Page in Web Frontend

The user profile page is slightly modified on top of the template. Article, events, profile editing and more tabs are added to the interface. Articles, events and portfolios can be made through profile page. Profile photo and more information about the user can be modified though profile page.

3.8 Home Page in Web Frontend

The page when user first sees in the website is the home page. When a user gets into the website there are two options for him/her, to register or to login. First, the design of this page was done and then patched into the website instead of the login page as the first page.

4 Summary of Work done

Group Member	Contribution
Sadullah Gültekin	Project plan is designed. With other frontend team members, example templates are searched and the most useful one is selected. The template that is found is read and its inner functionalities are understood. Profile page is implemented. Creating article page and editing profile pages are implemented also.
Enes Turan Özcan	Authentication system has been implemented. Admin, basic and trader roles has been defined. Creating new account, changing password, logging in & out, getting user profile information, private & public profile status, JWT authorization, data model for user and skeleton project has been implemented.
Burak Yüksel	Project plan is designed. With other frontend team members, example templates are searched and the most useful one is selected. The template that is found is read and its inner functionalities are understood. Signup page is designed. Communication with the server for registration is implemented.
Irmak Güzey	With other frontend team members, example templates are searched and the most useful one is selected. The template that is found is read and its inner functionalities are understood. Home page is implemented. Designs of login and registration page are changed accordingly. Presentation for the milestone is prepared.
Baran Deniz Korkmaz	The design of forgot password page of Android application has been carried out.
Barış Ege Sevgili	Java Spring Framework is analyzed and understood. Unfollow user, get followees, number of followers and followees endpoints and amIFollowing endpoint and related functionalities have been implemented. According to feedbacks, error response messages have been modified and corrected. Implemented functions related to following system are corrected.
Burak İkan Yıldız	Follow system tables has been implemented in the database. Java Spring Framework is analyzed and understood. Follow user, get followers, can follow endpoints and functionalities have been implemented. According to feedback, error response messages have been modified and corrected. User profile page modified to conform with follow system.
Fatih Mustafa Kurt	Android Signup functionality and data flow architecture has been implemented.

5 Requirements

5.1 Functional Requirements

5.1.1 User Requirements

1. Guests shall be able to only view the price of trading equipment.
2. Guests shall be able to browse economic articles, trading equipment and read comments about them.
3. Users shall be able to create an account by providing their username, e-mail, password and location using Google Maps. (implemented)
4. Registered users shall be able to login to the application by providing their username or email and password. (implemented)
5. Users should be able to sign up via Google account.
6. Already signed users should be able to sign in via Google account.
7. User shall validate account via e-mail. (implemented)
8. The users who want to be trader user shall provide their IBAN number. (implemented)
9. Users shall be able to reset their passwords if they forget their passwords by clicking ”Forgot your password?” button. (implemented)
10. User shall be able to follow other users directly if the user’s profile is public. (implemented)
11. User shall be able to send follow requests to users that have a private profile, and shall wait requests to be accepted.
12. User shall be able to accept or reject the following requests.
13. User shall be able to follow trading equipment.
14. User shall be able to set alerts for certain levels of trading equipment.
15. Users shall have at least one portfolio.
16. Users shall be able to create/delete their portfolios.
17. Users shall be able to change the name of their portfolios.
18. Users shall be able to add/delete new trading equipment to their portfolio.
19. Users shall be able to read/rate other users’ articles, comments about trading equipment.
20. Users shall be able to write new articles and comments, make a prediction about any trading equipment, and view daily prediction rates of trading equipments.
21. Users shall be able to edit/delete their article comments.
22. Trader users shall be able to trade trading equipment.
23. Trader users shall be able to set stop/loss limits.
24. Users shall be able to make their profile public or private.
25. Users shall be able to see their own portfolios in their own profile page.
26. Users shall be able to edit their own bio in profile page.
27. Users shall be able to reach their own followers and following list and their old articles/comments. (implemented but articles/comments)
28. Users shall be able to see their own prediction success rate.

29. Users shall be able to see other users' prediction success rate in their profile.
30. Users shall be able to see other users' followers, articles, following list if the profile is public or if he/she follows him/her.
31. Users shall be able to only see other user's prediction rate if their profile is private and if he/she is not his/her friend.
32. Trading users shall have my investments page.
33. Basic users shall not have my investments page.
34. Trading users shall be able to invest on trading equipment in my investments page.
35. Trading users shall be able to create a buy order for a trading equipment for a specified rate in my investments page.
36. Trading users shall be able to set stop/loss limits on trading equipment in my investments page.
37. Users shall have a profit/loss section which is private to each user.
38. Users shall be able to see profit/loss in terms of currency chosen by user.
39. Users shall be able to manually enter investments to see calculated profit/loss.
40. Users shall be able to see news fetched from third party website.
41. Users shall be able to be redirected to third party site when click the news they would like to read in detail.

5.1.2 System Requirements

1. System shall provide a homepage for each user according to user interests.
2. System shall provide a homepage that enables registered users to see economic events, news and trading commodities that user is interested in.
3. System shall provide a navigation bar in order to enable users to switch to their profile section, articles section, news section etc.
4. System shall provide an "Events" section in which there are economic events, articles, news that is fetched from a third-party source.
5. System shall assign different significance levels for economic events.
6. Events shall be updated daily.
7. System shall support searching for users and trading equipment.
8. System shall support different type of searching criteria.
9. System shall support semantic search based on the context of information.
10. System shall support location-based search (be able to filter users based on city or district).
11. System shall support filtering events depending on their significance.
12. System shall notify users in accordance with their alerts.
13. System shall recommend articles, trading equipment, commodities for the users based on their history in the system.
14. System shall provide a "Trading" section where Trader users can buy/sell commodities, trading equipment.
15. System shall automatically detects when a trader make a transaction and update his/her portfolio.
16. System shall highlight the commodities that are trend or the user may be interested in based on his/her history.
17. System shall provide a simple, seamless and secure way for money transaction for trader users.
18. System should cooperate with 3rd party applications for money transaction and mobile payments.

5.2 Non-functional Requirements

5.2.1 Accessibility and Availability

1. The system shall have a native web and native mobile client.(implemented)
2. The system shall be deployable on a remote and manually configurable server.(implemented)
3. The website and the mobile application shall be available in English. (implemented)
4. The system shall support UTF-8(Unicode) charset. (implemented)
5. The web app shall provide support for Screen Reader applications.

5.2.2 Annotatitons

1. System shall support W3C Web Annotation Data Model and follow W3C Web Annotation Protocol so that the contents can be annotated by users.
2. System shall follow the standards introduced by the W3C.

5.2.3 Performance

1. System shall respond to request in at most 5 seconds. (implemented)
2. System shall use queue system to reduce response time. (implemented)
3. System should cache the commonly used contents to reduce response time.
4. Native Android app and web app should run smoothly and use low system resources. (implemented)

5.2.4 Privacy

1. The user shall be able to respond follow request before following is activated.
2. The user shall be able to choose whether he/she wants his/her profile to be public or private.

5.2.5 Security

1. Financial Transactions shall be reliable and secure.
2. System shall be invulnerable against potential SQL injection, XSS attacks and DDOS attacks. (implemented)
3. System shall force users to use strong passwords, which must consist of at least six characters (and the more characters, the stronger the password) that are a combination of letters, numbers and symbols. (implemented)
4. System shall protect users' information by denying any unauthorized accesses. (implemented)
5. System shall store hashed version of user sign in password and don't store the password itself in anyhow. (implemented)
6. System shall backup all data to AWS Storage after each week.
7. System shall encrypt every connection and data transfer by using latest version of TSL encryption.
8. The deployment server must be secured from any possible attack types, such as open port issues. (implemented)

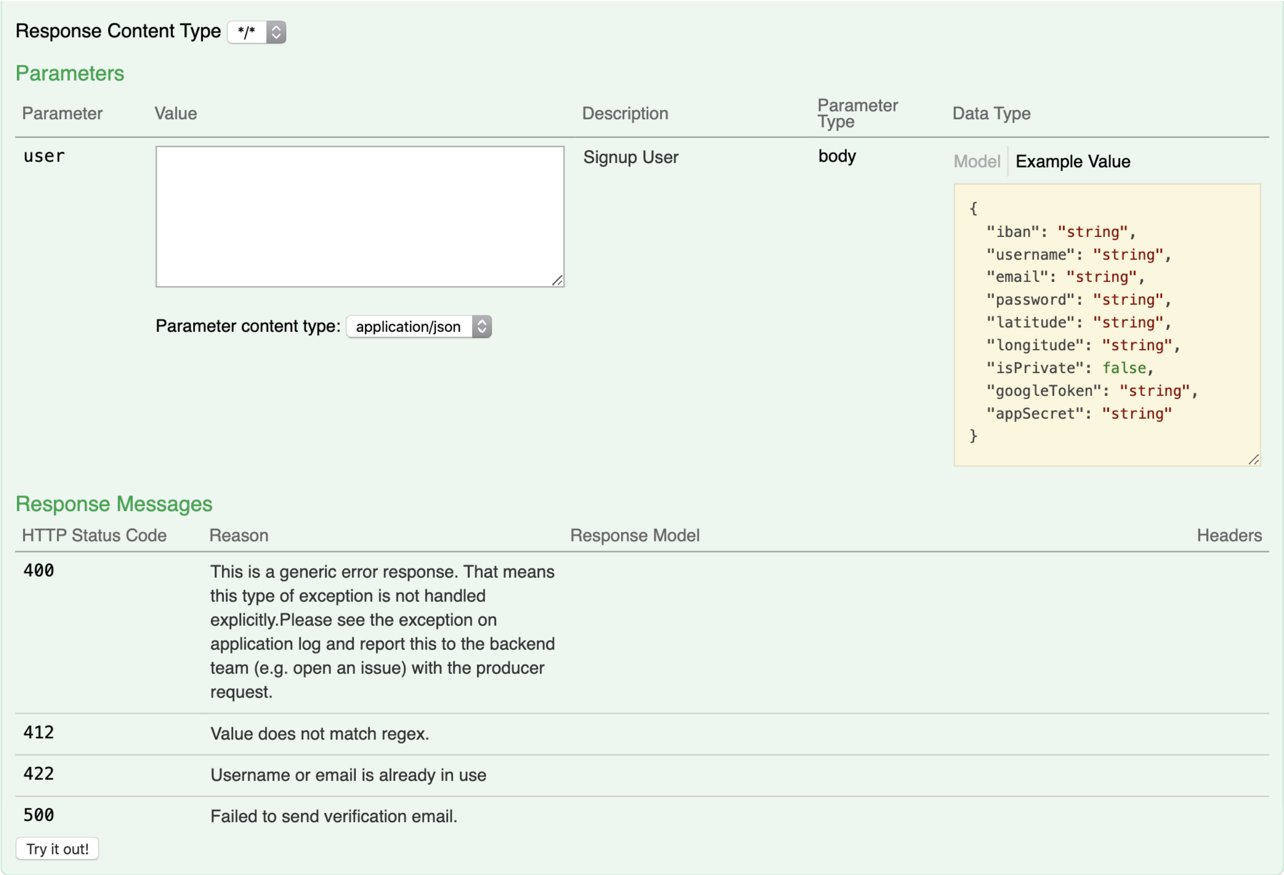
6 API Documentation

For the API Documentation of our project -rather than writing all the information for the endpoints manually, we use Springfox suite to create API Documentation automatically when our application is run. The Springfox suite of java libraries are all about automating the generation of machine and human readable specifications for JSON APIs written using the spring family of projects. Springfox works by examining an application, once, at runtime to infer API semantics based on spring configurations, class structure and various compile time java Annotations. For an endpoint, if you annotate and define the parameters and explanations for these parameters properly they show up on the documentation page with the corresponding data format for both request (i.e. Request parameter, path parameter or body parameter) and response. Once you configure the ”Docket” in the application, for each endpoint annotated with Swagger annotations there will be a manual page describing the endpoint on the document. Here is an example:

```

@PostMapping("")
@ApiOperation(value = "Registers a new user to the system.")
@ResponseStatus(HttpStatus.OK)
@ApiResponses(value = {
        @ApiResponse(code = 400, message = GlobalExceptionHandlerController.GENERIC_ERROR_RESPONSE),
        @ApiResponse(code = 422, message = "Username or email is already in use"),
        @ApiResponse(code = 412, message = "Value does not match regex."),
        @ApiResponse(code = 500, message = "Failed to send verification email.")})
public StringResponseWrapper signup(@ApiParam("Signup User") @RequestBody UserDataDTO user) {
    return new StringResponseWrapper(signupService.signup(modelMapper.map(user, User.class), user.getAppSecret()));
}
```

Then the corresponding API documentation for this endpoint will look like the following:



Response Content Type: */*

Parameters

Parameter	Value	Description	Parameter Type	Data Type
user	Parameter content type: application/json	Signup User	body	Example Value (below)

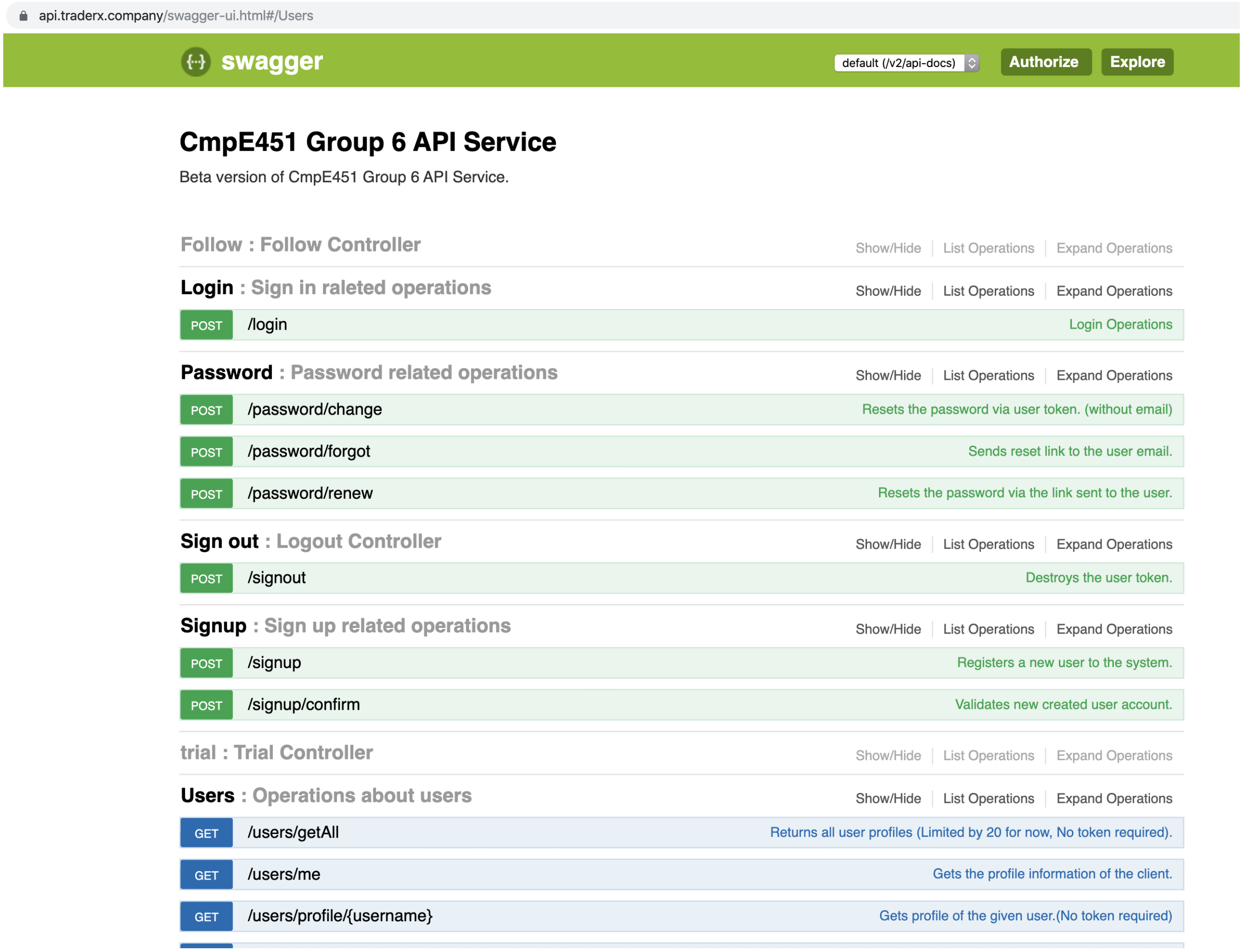
```
{
  "iban": "string",
  "username": "string",
  "email": "string",
  "password": "string",
  "latitude": "string",
  "longitude": "string",
  "isPrivate": false,
  "googleToken": "string",
  "appSecret": "string"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	This is a generic error response. That means this type of exception is not handled explicitly.Please see the exception on application log and report this to the backend team (e.g. open an issue) with the producer request.		
412	Value does not match regex.		
422	Username or email is already in use		
500	Failed to send verification email.		

As can be seen on the right, "UserDataDTO" object is converted to JSON format and field types are denoted. Also possible response messages and status codes are shown at the end. This rescues backend team from the burden of writing down all the document for all endpoints. Further, it helps us not forget to include an endpoint for a particular one since they are created automatically. Here is the API documentation screenshot of our project. It also can be seen on the website from the documentation link.

- API Documentation
  - https://api.traderx.company/swagger-ui.html
- JSON API Documentation
  - https://api.traderx.company/v2/api-docs
- Convert JSON to v2 Swagger API Documentation
  - https://editor.swagger.io



CmpE451 Group 6 API Service

Beta version of CmpE451 Group 6 API Service.

Follow : Follow Controller — Show/Hide | List Operations | Expand Operations

Login : Sign in raleted operations — Show/Hide | List Operations | Expand Operations

POST	/login	Login Operations

Password : Password related operations — Show/Hide | List Operations | Expand Operations

POST	/password/change	Resets the password via user token. (without email)
POST	/password/forgot	Sends reset link to the user email.
POST	/password/renew	Resets the password via the link sent to the user.

Sign out : Logout Controller — Show/Hide | List Operations | Expand Operations

POST	/signout	Destroys the user token.

Signup : Sign up related operations — Show/Hide | List Operations | Expand Operations

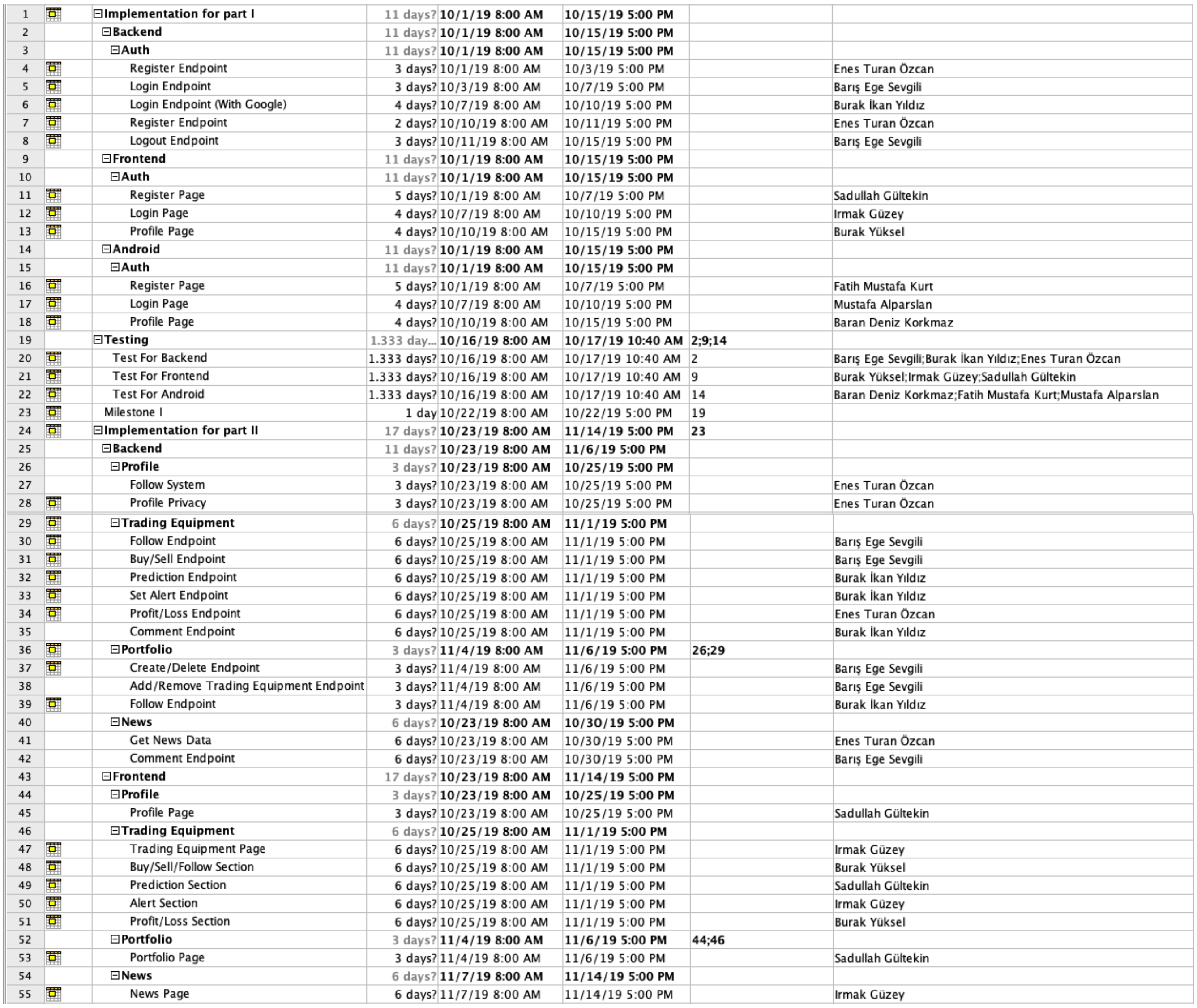
POST	/signup	Registers a new user to the system.
POST	/signup/confirm	Validates new created user account.

trial : Trial Controller — Show/Hide | List Operations | Expand Operations

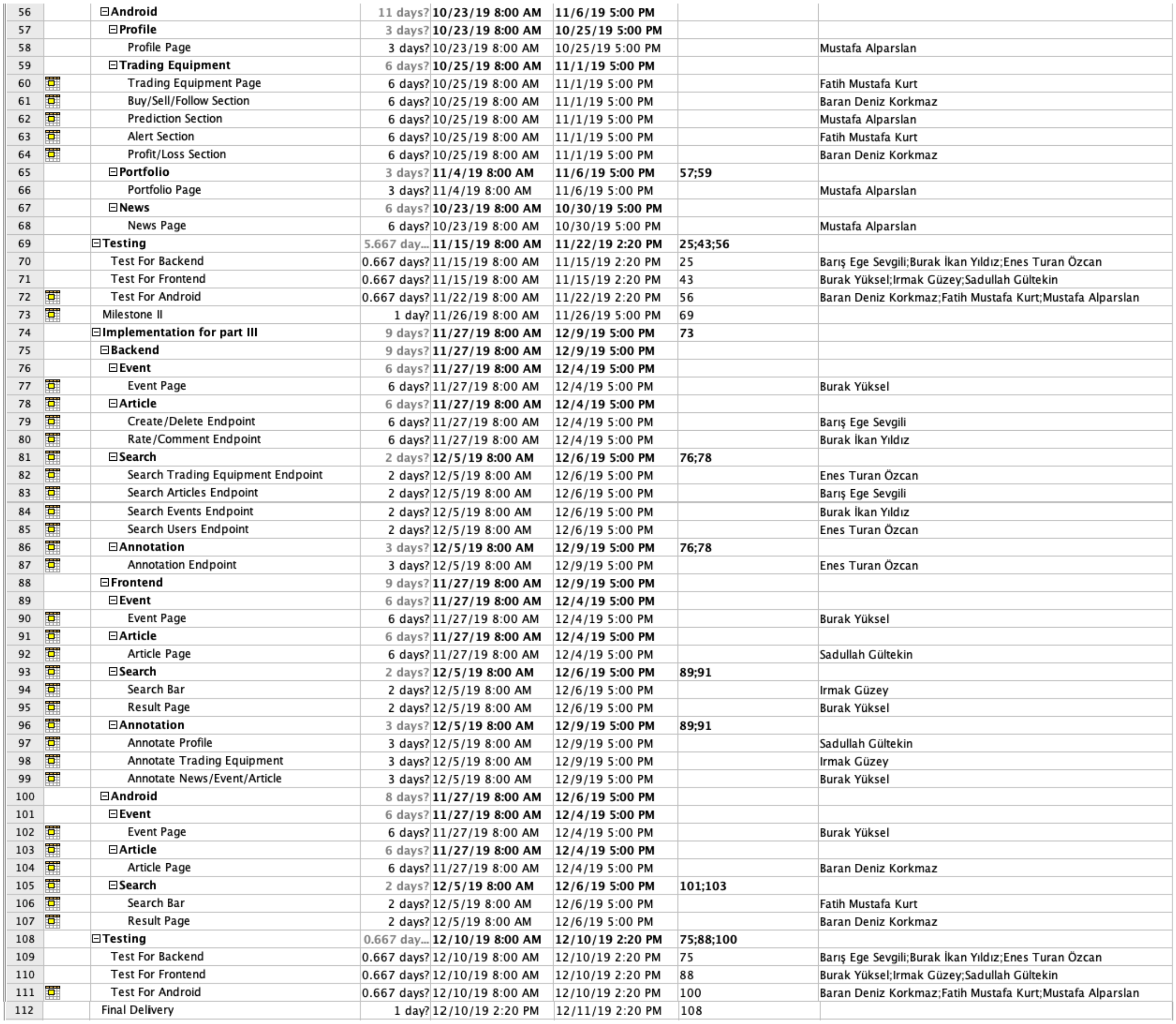
Users : Operations about users — Show/Hide | List Operations | Expand Operations

GET	/users/getAll	Returns all user profiles (Limited by 20 for now, No token required).
GET	/users/me	Gets the profile information of the client.
GET	/users/profile/{username}	Gets profile of the given user.(No token required)

7 Project Plan



1	▦	⊟**Implementation for part I**	11 days?	**10/1/19 8:00 AM**	**10/15/19 5:00 PM**		
2		⊟**Backend**	11 days?	**10/1/19 8:00 AM**	**10/15/19 5:00 PM**		
3		⊟**Auth**	11 days?	**10/1/19 8:00 AM**	**10/15/19 5:00 PM**		
4	▦	Register Endpoint	3 days?	10/1/19 8:00 AM	10/3/19 5:00 PM		Enes Turan Özcan
5	▦	Login Endpoint	3 days?	10/3/19 8:00 AM	10/7/19 5:00 PM		Barış Ege Sevgili
6	▦	Login Endpoint (With Google)	4 days?	10/7/19 8:00 AM	10/10/19 5:00 PM		Burak İkan Yıldız
7	▦	Register Endpoint	2 days?	10/10/19 8:00 AM	10/11/19 5:00 PM		Enes Turan Özcan
8	▦	Logout Endpoint	3 days?	10/11/19 8:00 AM	10/15/19 5:00 PM		Barış Ege Sevgili
9		⊟**Frontend**	11 days?	**10/1/19 8:00 AM**	**10/15/19 5:00 PM**		
10		⊟**Auth**	11 days?	**10/1/19 8:00 AM**	**10/15/19 5:00 PM**		
11	▦	Register Page	5 days?	10/1/19 8:00 AM	10/7/19 5:00 PM		Sadullah Gültekin
12	▦	Login Page	4 days?	10/7/19 8:00 AM	10/10/19 5:00 PM		Irmak Güzey
13	▦	Profile Page	4 days?	10/10/19 8:00 AM	10/15/19 5:00 PM		Burak Yüksel
14		⊟**Android**	11 days?	**10/1/19 8:00 AM**	**10/15/19 5:00 PM**		
15		⊟**Auth**	11 days?	**10/1/19 8:00 AM**	**10/15/19 5:00 PM**		
16	▦	Register Page	5 days?	10/1/19 8:00 AM	10/7/19 5:00 PM		Fatih Mustafa Kurt
17	▦	Login Page	4 days?	10/7/19 8:00 AM	10/10/19 5:00 PM		Mustafa Alparslan
18	▦	Profile Page	4 days?	10/10/19 8:00 AM	10/15/19 5:00 PM		Baran Deniz Korkmaz
19		⊟**Testing**	1.333 day...	**10/16/19 8:00 AM**	**10/17/19 10:40 AM**	**2;9;14**	
20	▦	Test For Backend	1.333 days?	10/16/19 8:00 AM	10/17/19 10:40 AM	2	Barış Ege Sevgili;Burak İkan Yıldız;Enes Turan Özcan
21	▦	Test For Frontend	1.333 days?	10/16/19 8:00 AM	10/17/19 10:40 AM	9	Burak Yüksel;Irmak Güzey;Sadullah Gültekin
22	▦	Test For Android	1.333 days?	10/16/19 8:00 AM	10/17/19 10:40 AM	14	Baran Deniz Korkmaz;Fatih Mustafa Kurt;Mustafa Alparslan
23	▦	Milestone I	1 day	10/22/19 8:00 AM	10/22/19 5:00 PM	19	
24	▦	⊟**Implementation for part II**	17 days?	**10/23/19 8:00 AM**	**11/14/19 5:00 PM**	**23**	
25		⊟**Backend**	11 days?	**10/23/19 8:00 AM**	**11/6/19 5:00 PM**		
26		⊟**Profile**	3 days?	**10/23/19 8:00 AM**	**10/25/19 5:00 PM**		
27		Follow System	3 days?	10/23/19 8:00 AM	10/25/19 5:00 PM		Enes Turan Özcan
28	▦	Profile Privacy	3 days?	10/23/19 8:00 AM	10/25/19 5:00 PM		Enes Turan Özcan
29	▦	⊟**Trading Equipment**	6 days?	**10/25/19 8:00 AM**	**11/1/19 5:00 PM**		
30	▦	Follow Endpoint	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Barış Ege Sevgili
31	▦	Buy/Sell Endpoint	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Barış Ege Sevgili
32	▦	Prediction Endpoint	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Burak İkan Yıldız
33	▦	Set Alert Endpoint	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Burak İkan Yıldız
34	▦	Profit/Loss Endpoint	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Enes Turan Özcan
35		Comment Endpoint	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Burak İkan Yıldız
36	▦	⊟**Portfolio**	3 days?	**11/4/19 8:00 AM**	**11/6/19 5:00 PM**	**26;29**	
37	▦	Create/Delete Endpoint	3 days?	11/4/19 8:00 AM	11/6/19 5:00 PM		Barış Ege Sevgili
38		Add/Remove Trading Equipment Endpoint	3 days?	11/4/19 8:00 AM	11/6/19 5:00 PM		Barış Ege Sevgili
39	▦	Follow Endpoint	3 days?	11/4/19 8:00 AM	11/6/19 5:00 PM		Burak İkan Yıldız
40		⊟**News**	6 days?	**10/23/19 8:00 AM**	**10/30/19 5:00 PM**		
41		Get News Data	6 days?	10/23/19 8:00 AM	10/30/19 5:00 PM		Enes Turan Özcan
42		Comment Endpoint	6 days?	10/23/19 8:00 AM	10/30/19 5:00 PM		Barış Ege Sevgili
43		⊟**Frontend**	17 days?	**10/23/19 8:00 AM**	**11/14/19 5:00 PM**		
44		⊟**Profile**	3 days?	**10/23/19 8:00 AM**	**10/25/19 5:00 PM**		
45		Profile Page	3 days?	10/23/19 8:00 AM	10/25/19 5:00 PM		Sadullah Gültekin
46		⊟**Trading Equipment**	6 days?	**10/25/19 8:00 AM**	**11/1/19 5:00 PM**		
47	▦	Trading Equipment Page	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Irmak Güzey
48	▦	Buy/Sell/Follow Section	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Burak Yüksel
49	▦	Prediction Section	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Sadullah Gültekin
50	▦	Alert Section	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Irmak Güzey
51	▦	Profit/Loss Section	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Burak Yüksel
52		⊟**Portfolio**	3 days?	**11/4/19 8:00 AM**	**11/6/19 5:00 PM**	**44;46**	
53	▦	Portfolio Page	3 days?	11/4/19 8:00 AM	11/6/19 5:00 PM		Sadullah Gültekin
54		⊟**News**	6 days?	**11/7/19 8:00 AM**	**11/14/19 5:00 PM**		
55	▦	News Page	6 days?	11/7/19 8:00 AM	11/14/19 5:00 PM		Irmak Güzey



56		⊟**Android**	11 days?	**10/23/19 8:00 AM**	**11/6/19 5:00 PM**		
57		⊟**Profile**	3 days?	**10/23/19 8:00 AM**	**10/25/19 5:00 PM**		
58		Profile Page	3 days?	10/23/19 8:00 AM	10/25/19 5:00 PM		Mustafa Alparslan
59		⊟**Trading Equipment**	6 days?	**10/25/19 8:00 AM**	**11/1/19 5:00 PM**		
60	📅	Trading Equipment Page	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Fatih Mustafa Kurt
61	📅	Buy/Sell/Follow Section	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Baran Deniz Korkmaz
62	📅	Prediction Section	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Mustafa Alparslan
63	📅	Alert Section	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Fatih Mustafa Kurt
64	📅	Profit/Loss Section	6 days?	10/25/19 8:00 AM	11/1/19 5:00 PM		Baran Deniz Korkmaz
65		⊟**Portfolio**	3 days?	**11/4/19 8:00 AM**	**11/6/19 5:00 PM**	**57;59**	
66		Portfolio Page	3 days?	11/4/19 8:00 AM	11/6/19 5:00 PM		Mustafa Alparslan
67		⊟**News**	6 days?	**10/23/19 8:00 AM**	**10/30/19 5:00 PM**		
68		News Page	6 days?	10/23/19 8:00 AM	10/30/19 5:00 PM		Mustafa Alparslan
69		⊟**Testing**	5.667 day...	**11/15/19 8:00 AM**	**11/22/19 2:20 PM**	**25;43;56**	
70		Test For Backend	0.667 days?	11/15/19 8:00 AM	11/15/19 2:20 PM	25	Barış Ege Sevgili;Burak İkan Yıldız;Enes Turan Özcan
71		Test For Frontend	0.667 days?	11/15/19 8:00 AM	11/15/19 2:20 PM	43	Burak Yüksel;Irmak Güzey;Sadullah Gültekin
72	📅	Test For Android	0.667 days?	11/22/19 8:00 AM	11/22/19 2:20 PM	56	Baran Deniz Korkmaz;Fatih Mustafa Kurt;Mustafa Alparslan
73	📅	Milestone II	1 day?	11/26/19 8:00 AM	11/26/19 5:00 PM	69	
74		⊟**Implementation for part III**	9 days?	**11/27/19 8:00 AM**	**12/9/19 5:00 PM**	**73**	
75		⊟**Backend**	9 days?	**11/27/19 8:00 AM**	**12/9/19 5:00 PM**		
76		⊟**Event**	6 days?	**11/27/19 8:00 AM**	**12/4/19 5:00 PM**		
77	📅	Event Page	6 days?	11/27/19 8:00 AM	12/4/19 5:00 PM		Burak Yüksel
78	📅	⊟**Article**	6 days?	**11/27/19 8:00 AM**	**12/4/19 5:00 PM**		
79	📅	Create/Delete Endpoint	6 days?	11/27/19 8:00 AM	12/4/19 5:00 PM		Barış Ege Sevgili
80	📅	Rate/Comment Endpoint	6 days?	11/27/19 8:00 AM	12/4/19 5:00 PM		Burak İkan Yıldız
81	📅	⊟**Search**	2 days?	**12/5/19 8:00 AM**	**12/6/19 5:00 PM**	**76;78**	
82	📅	Search Trading Equipment Endpoint	2 days?	12/5/19 8:00 AM	12/6/19 5:00 PM		Enes Turan Özcan
83	📅	Search Articles Endpoint	2 days?	12/5/19 8:00 AM	12/6/19 5:00 PM		Barış Ege Sevgili
84	📅	Search Events Endpoint	2 days?	12/5/19 8:00 AM	12/6/19 5:00 PM		Burak İkan Yıldız
85	📅	Search Users Endpoint	2 days?	12/5/19 8:00 AM	12/6/19 5:00 PM		Enes Turan Özcan
86	📅	⊟**Annotation**	3 days?	**12/5/19 8:00 AM**	**12/9/19 5:00 PM**	**76;78**	
87	📅	Annotation Endpoint	3 days?	12/5/19 8:00 AM	12/9/19 5:00 PM		Enes Turan Özcan
88		⊟**Frontend**	9 days?	**11/27/19 8:00 AM**	**12/9/19 5:00 PM**		
89		⊟**Event**	6 days?	**11/27/19 8:00 AM**	**12/4/19 5:00 PM**		
90	📅	Event Page	6 days?	11/27/19 8:00 AM	12/4/19 5:00 PM		Burak Yüksel
91	📅	⊟**Article**	6 days?	**11/27/19 8:00 AM**	**12/4/19 5:00 PM**		
92	📅	Article Page	6 days?	11/27/19 8:00 AM	12/4/19 5:00 PM		Sadullah Gültekin
93	📅	⊟**Search**	2 days?	**12/5/19 8:00 AM**	**12/6/19 5:00 PM**	**89;91**	
94	📅	Search Bar	2 days?	12/5/19 8:00 AM	12/6/19 5:00 PM		Irmak Güzey
95	📅	Result Page	2 days?	12/5/19 8:00 AM	12/6/19 5:00 PM		Burak Yüksel
96	📅	⊟**Annotation**	3 days?	**12/5/19 8:00 AM**	**12/9/19 5:00 PM**	**89;91**	
97	📅	Annotate Profile	3 days?	12/5/19 8:00 AM	12/9/19 5:00 PM		Sadullah Gültekin
98	📅	Annotate Trading Equipment	3 days?	12/5/19 8:00 AM	12/9/19 5:00 PM		Irmak Güzey
99	📅	Annotate News/Event/Article	3 days?	12/5/19 8:00 AM	12/9/19 5:00 PM		Burak Yüksel
100		⊟**Android**	8 days?	**11/27/19 8:00 AM**	**12/6/19 5:00 PM**		
101		⊟**Event**	6 days?	**11/27/19 8:00 AM**	**12/4/19 5:00 PM**		
102	📅	Event Page	6 days?	11/27/19 8:00 AM	12/4/19 5:00 PM		Burak Yüksel
103	📅	⊟**Article**	6 days?	**11/27/19 8:00 AM**	**12/4/19 5:00 PM**		
104	📅	Article Page	6 days?	11/27/19 8:00 AM	12/4/19 5:00 PM		Baran Deniz Korkmaz
105	📅	⊟**Search**	2 days?	**12/5/19 8:00 AM**	**12/6/19 5:00 PM**	**101;103**	
106	📅	Search Bar	2 days?	12/5/19 8:00 AM	12/6/19 5:00 PM		Fatih Mustafa Kurt
107	📅	Result Page	2 days?	12/5/19 8:00 AM	12/6/19 5:00 PM		Baran Deniz Korkmaz
108		⊟**Testing**	0.667 day...	**12/10/19 8:00 AM**	**12/10/19 2:20 PM**	**75;88;100**	
109		Test For Backend	0.667 days?	12/10/19 8:00 AM	12/10/19 2:20 PM	75	Barış Ege Sevgili;Burak İkan Yıldız;Enes Turan Özcan
110		Test For Frontend	0.667 days?	12/10/19 8:00 AM	12/10/19 2:20 PM	88	Burak Yüksel;Irmak Güzey;Sadullah Gültekin
111	📅	Test For Android	0.667 days?	12/10/19 8:00 AM	12/10/19 2:20 PM	100	Baran Deniz Korkmaz;Fatih Mustafa Kurt;Mustafa Alparslan
112		Final Delivery	1 day?	12/10/19 2:20 PM	12/11/19 2:20 PM	108	

8 User scenarios

Ali Durdurucu is a high school student who is interested in Economics and whose dream job is computer engineering. He is not confident enough to cope with wise-guys of trading ecosystem. He just needs a place to share his ideas, criticize the ideas of others and track whats going on out there. Finally, when he was surfing on the web, he came across the trading platform, TraderX, that is written by Boğaziçi University students. He gave it a try.

- He enters the system for the first time. Clicks Register and face with the SignUp page
  - He fills the e-mail and username areas.
  - When he picks a password, the system warns him to correct his password compatible with the ”strong password standards”, and he does.
  - He shares his location information.
  - He decides to be a Basic User, since he doesn’t want to make any exchange until he got some experience in the area, he picks the related option.
  - As he presses the ”register” button, he faces with a pop-up that informs him about a verification mail that is sent to him.
  - He activates his account using the link from the mail that is sent to him. As he click the link, the system directs him to the login page.
- He logs in to the system with his username and password and gets to the his profile page directly.
  - He changes his profile settings as to be a private user by using the button at his profile page
  - He decides to write an article about High inflation rates in Turkey, accordingly he goes to the article section at his profile page.
  - Using the user interface he sets the title and the summary of the article. He writes all the ideas to the body of the article and shares his article succesfully
- He logs out from the system using logout button at his profile page

9 Code Structure

We have three main projects to develop, Web app, Android app and Backend app. Each projects has its own directory under the github repository, which are located as frontend/Traderx, android/Traderx, backend/Traderx. We have dedicated group members to each projects. Every member is responsible for his/her dedicated project's issues, bugs, and implementing new features.

We are using branch system to differ applications development from each other. Therefore, we have three main branches in addition to master branch, these are frontend-dev, android-dev, backend-dev. This branching system offers great flexibility to track the development of apps independently from other apps developments. If a new feature will be added to a application, a new branch will be opened which is branched from related application's development branch. Since our applications are started to development yet, we do not have a main development branch which will be the parent of three application development branches. As the main functional requirements are met by apps, we will create a main development branch which differs from master branch. Master branch always will be containing a bug free application.

Hot-fixes, which must be merged into master immediately, are branched from master. After merging hot-fixes into master branch, other development branches pulls this fixes into their branch. With this method, we can offer fixes quickly to production application. All these opened branches merged into its parent branch by opening a pull request. If these pull requests are merging into development branch, reviewers will be set application's dedicated group members.

We are using Github issue system to state the app's bugs, discuss the topics to decide the best. This issue system helps a lot to our application developments. For example, if a group member forgets something which was decided so long ago, he/she can find the related issue and remember the reasonings behind the decision.

10 Evaluation Of Tools And Managing The Project

In this semester we are having difficulties with organizing a meeting because we don't have much to talk as a whole team. Although it is important to know others progress with the project, arranging a meeting for this purpose seems meaningless for most team members. From now, probably we will arrange meetings as small groups. For example we will organize a meeting that only frontend team members attend. Nevertheless, we will keep in touch with other team's.

We had some problems in the selection of framework that will be use. For example in the backend, we decided to write our program in Java. Other alternative was using Python. In our backend team only one member had an experience with writing Java. Other members thought that learning python would be easier than learning Java. However, choosing Python was much more risky, because if everyone fails to learn Python before milestone, it would be impossible to turn back from that point. Hence, we think that it was right choice to select the Java as a backend framework.

In the android team, one of our team members won't be available for the next month which will decrease our production power. Up until now, we didn't need anyone to help to android team. However we can make a migration from other teams if it is necessary.

Frontend team had some difficulties to understand the template that they have chosen. Because it took considerable amount of time to understand how its working, they couldn't proceed as they wanted. However, now they became very familiar with the template and we think that from now it will be much more easy to implement new functionalities.

As a whole, our team is working properly without any major deficiency. We think that until next milestone we will implement most parts of the project.