

BOGAZICI UNIVERSITY  
DEPARTMENT OF COMPUTER  
ENGINEERING  
**Fundamentals of Software Engineering**

---

**MILESTONE REPORT 2  
(Group 1)**

---

Emirhan Sarac  
Ufuk Yilmaz  
Emre Hosler  
Eray Kurtulus  
Ilker Ozkan

Mete Han Kurt  
Omer F. Dogru  
Altay Ince  
Irem Ustunboyacioglu  
Omer F. Toptas



# Contents

<b>1 Executive Summary</b>	<b>2</b>
1.1 Introduction . . . . .	2
1.2 Work Done So Far . . . . .	2
1.3 Road Ahead . . . . .	3
1.4 Challenges You Met As a Group . . . . .	3
<b>2 List and Status of Deliverables</b>	<b>3</b>
<b>3 Evaluation of Status of Deliverables</b>	<b>3</b>
3.1 Communication Plan . . . . .	4
3.2 Requirements . . . . .	4
3.3 User Stories and Mock-ups . . . . .	4
3.4 Design(UML Diagrams) . . . . .	4
3.5 Project Plan . . . . .	5
3.6 Practice App . . . . .	5
<b>4 Summary of Team Member's Work</b>	<b>6</b>
<b>5 Project Plan</b>	<b>9</b>
<b>6 Evaluation of Tools and Processes</b>	<b>12</b>
6.1 Node.js and its Libraries . . . . .	12
6.2 MongoDB(DB) . . . . .	12
6.3 ProjectLibre . . . . .	12
6.4 WebStorm . . . . .	12
6.5 Postman . . . . .	12
6.6 Commit Procedure . . . . .	13
6.6.1 Motivation . . . . .	13
6.6.2 Commit Message Template . . . . .	13
6.7 Pull Request Procedure . . . . .	13
<b>7 API URL Documentation</b>	<b>15</b>

# **1 Executive Summary**

## **1.1 Introduction**

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. Also, this platform will use various API's that will provide cryptocurrency ratios, currency ratios, latest economic news, and latest articles about economic events. Users will be able to follow other users, and trading equipment and also set alerts for certain levels of trading equipment.

## **1.2 Work Done So Far**

At the beginning of the software engineering course, we were not experienced about design issues, especially "what to do?" and "how to do?" parts. We have a team of 10 people and different kinds of skills in software engineering. For each task assigned to us, we have divided tasks into sub-tasks and assigned group members who are more appropriate for these tasks. Some tasks were challenging because none of the team members has an idea about it. However, we are students that are thirsty for knowledge and challenge so that we handle all tasks because of this.

Each Wednesday, we kept a regular meeting. We talk about current issues about the project, what have we done last week, and what should we have done this week. Team members were actively contributing to discussions and actions to do. Moreover, we have a regular check-up session on Fridays that we are talking about the progression of current tasks assigned to each member.

First, we have completed the requirements draft. Then, we have made mock-ups and corresponding scenarios to enlarge requirements. This is a way of thinking from the perspective of users. Up to this point, we have completed "What to do?" part. Then, we have moved to "How to do?" part, and created design parts as use case diagrams, class diagrams, and sequence diagrams. Even though, we are working on "How to do?" part, we have tried to enlarge our project requirements.

Then, we have moved to long-term goals and it was time to define our project plan. We have created our project plan for so far and future. It was a bit hard to define future goals since we don't know what to do. However as we progress on project, we were getting insight about future goals.

At last for this milestone, we have decided "How we will group up for sub-tasks" and "What are the sub-tasks" for practice app. We have divided team into two parts as front-end and back-end. Then, we have decided what we will use for implementation and which database we will choose. We have chosen NodeJS for backend implementation and MongoDB for database parts. We have created basic skeleton for

our project. Then we have implemented couple of pages and their back-operations. We have decided to use at least one API and use it in our project. Then we have created our own API and documented it. And create test cases for our project. Basically, we have tried to learn some fundamentals of implementation steps. At the end, we had signup, login, main, and APIs' pages in our practice app.

### 1.3 Road Ahead

Since last task, practice app, expects only runnable and basic project, we did not give full attention to front-end and back-end parts. We will make better front-end designs and more efficient backend operations. We will get used to asynchronous concept in NodeJS also. We will also create Android part of this project. More APIs' will be supported and functionalities that are described in requirements parts will be implemented in the project. We also need to make session control in our application, since we do not have enough time, we have skipped this part.

### 1.4 Challenges You Met As a Group

Unbalanced work load was the most crucial problem in our group. Also we didn't get the idea of independent programming of sub-parts of the project. Such as backend and frontend implementation should be implemented independently, not totally. We have to obey the requirements of the project and all parts should implement their part according to it without interaction of other parts. Also, we did not use pull requests so well so that most of it merged without review.

## 2 List and Status of Deliverables

Name	Delivery Date	Delivered
Communication Plan	Feb 17, 2019	✓
Requirements	Feb 26, 2019	✓
User Stories & Mockups	March 3, 2019	✓
Design Diagrams	March 15, 2019	✓
Project Plan	April 8, 2019	✓
Practice App	May 6, 2019	✓

## 3 Evaluation of Status of Deliverables

All the deliverables that are planned to be finished by the time of this report are finished without any latency. Every member of the team concluded and submitted

his/hers work on time.

### **3.1 Communication Plan**

We made a realistic communication plan and complied with it almost all of the time. We tried to finish our missions before the deadline with regular control by WhatsApp and Slack. Balanced distribution of tasks and appropriate deadlines gave us a good outline to follow.

### **3.2 Requirements**

Requirements were elicited according to the Project Description. Up to that point, lots of revisions and edits were made on it. Some modifications and additions have been done after customer meetings and customer feedback. It is very important to understand what to do in a project. So, forming requirements satisfactory is a key point to proceed project well. Our requirements have reshaped many times according to our questions and customer's requests. But there can be some modifications in the future under new assumptions.

### **3.3 User Stories and Mock-ups**

Three user stories and their corresponding mock-ups have been created. Group members were divided into 3 subgroups, each with 3 members. In every subgroup, one member has created a persona and scenario while other ones have created mock-ups. Every group member was responsible for reviewing other group members. In that way, we managed to deliver User Stories and Mock-ups on time. Then, some modifications have been made on this deliverable based on feedback from the customer (such as more clarification and specification). To get into the mindset of the customer, created mock-ups for scenarios should be real-like and overall. The customer should see sufficiently many functionalities.

### **3.4 Design(UML Diagrams)**

As we learned the general characteristics of each diagram before sharing the task, we performed the task sharing in a good way(2, 3, 4 people to do use case, sequence, class diagrams). Because we needed use case diagram while creating other diagrams, we have paid attention to the use case diagrams to be completed as soon as possible. After use case diagram created, we created class and sequence diagrams by considering use case diagram and the project requirements. We communicated well during the process of designing class and sequence diagrams in order to avoid inconsistency. According to the feedback from the customer, we have seen some errors in diagrams and made necessary corrections in a short time.

The class diagram perfectly describes the relationships among the objects of our platform and also the constraints imposed on the system. Use case diagram gather system requirements and actors together. One can easily observe the possible events of our platform by looking at use-case diagram. Sequence diagram shows object interactions ordered in time. The different scenario flows can be seen in sequence diagrams. In conclusion, the diagrams is very useful for us in different aspects.

### **3.5 Project Plan**

For the project plan, two persons were assigned to each section. Because everyone cannot make changes at the same time, everyone uploads the part they have changed to Github and says that they have updated the current project plan from Whatsapp. After everyone completed the assignments, a group of our friends finally made all the checks and combined them into a file called "Project Plan". The file was uploaded to Github after everything was complete. The effect of this on the project can be said as follows; everyone can see at what date who is responsible for which parts and can correct it accordingly.

### **3.6 Practice App**

We were divided into two groups: backend and frontend. Everyone contributed to the common code through his branch on Github. When everyone uploaded the code in his branch to Github, he waited for approval to prevent a possible error on merging. Since the frontend and the backend are synchronized with each other, we can complete the assignment until the deadline.

## 4 Summary of Team Member's Work

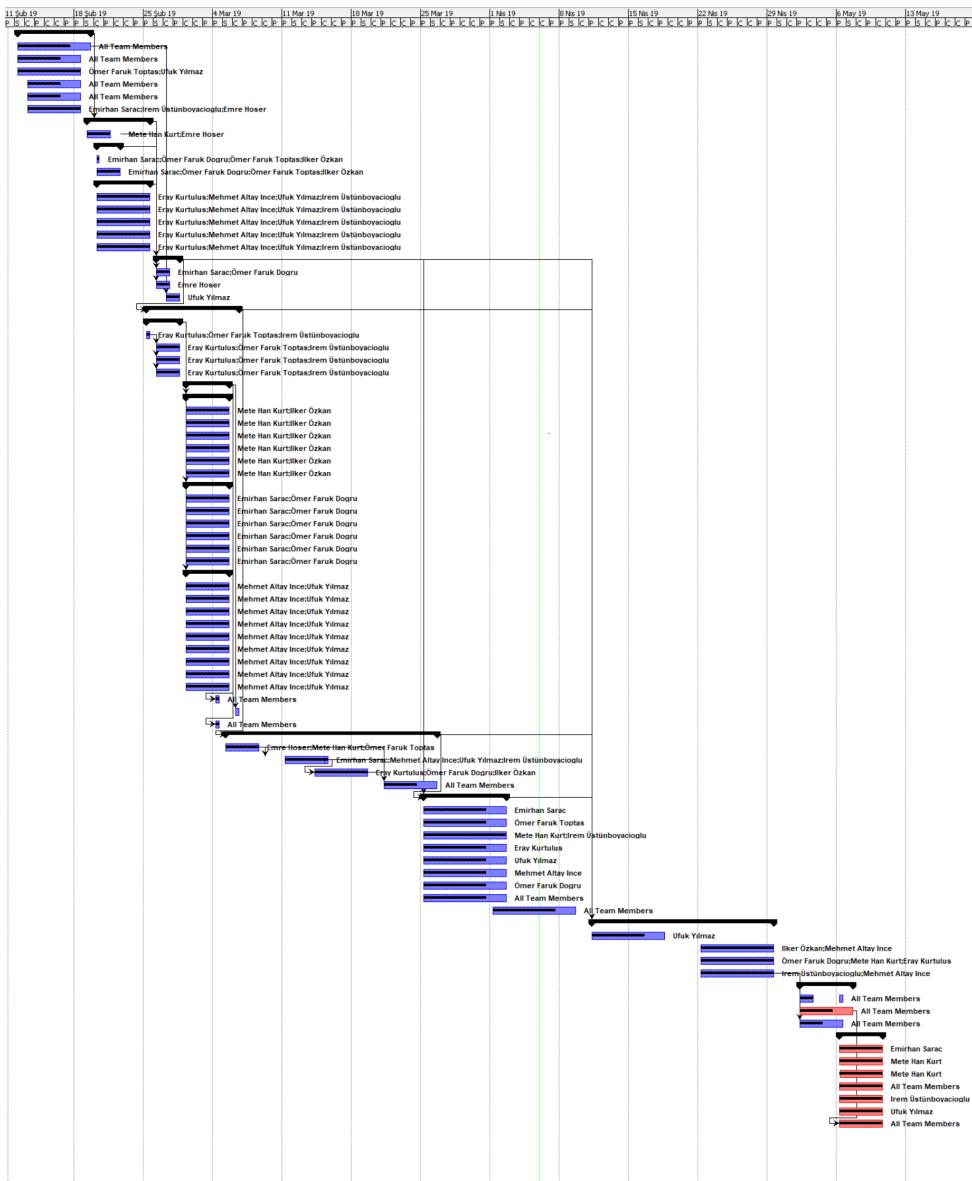
Team Member	Contributions
Ömer Faruk Doğru	<p>As a group member, I mostly joined group meetings and participated in discussions about assignments and feedbacks. I took part in some part of assignments and completed them in time. I have researched how to implement an API and tried to create it with javascript code. I have chosen Google Maps API for location information and combined it with the code. I contributed some part of backend section.</p>
Mete Han Kurt	<p>I attended all group meetings and recorded the meeting notes on a regular basis. I completed the List Status and Evaluation of Deliverables for Milestone Report2. I have done several API research for the project and have learned how to implement them. I have used TradingEconomics API and used the functions to get information about a lot of currencies I tried to contribute to the project in the backend section of the practice-app. I tried to fetch the current currency values and their daily, weekly, monthly, yearly change information from the API. And I prepared a testcase to see if the currency.js has the right values. In general, I did my best to contribute to the project and added the documentation to the get currency API.</p>
Irem Üstünboyacıoğlu	<p>In general, I tried to take part in the decision-making phase of our group meetings and I participated in weekly actions. As a member of this team, every week I have added missing links of our project repository to the home page. After first milestone, I have edited Project Plan by adding tasks that have been done between 05.03.2019 - 01.04.2019. Then, for practice-app assignment, I have studied APIs that might be useful for our project. I chose Bitfinex API and I created Javascript code files that gets information from that API. I took part in back-end implementation basically but also I tried to write some front-end codes for the API I have used on back-end side. Finally, I added new tasks to our Project Plan and attached it to Milestone Report 2.</p>

Team Member	Contributions
Mehmet Altay İnce	<p>As a member of this team, I gather with my teammates and discuss the weekly given assignments and feedbacks. I take responsibility for some share of the labor end try to accomplish it before its deadline. If any feedback is relevant with my part of the project, then I correct it. So far discluding small contributions, I created the mock-up 3 with the other 3 teammates, I created the class diagram with another 3 teammates and created the mock-up part of this milestone. After my tasks about milestone 1, We recieved the task on the running instance of our server. I participated in front-end part. Apart from implementing login page front end with my teammates , Me and Irem decided on API to to use in this project, which is the Bitfinex API . After that I implemented the login front-end page and google account log in for our project.</p>
Emirhan Saraç	<p>From the milestone report one to now, I have created "User scenarios and Mockups" part of the project planning part. Then created future part of it, practice app part. Then set all the predecessors of the project planning. I have implemented the all database operations. I have implemented backend for server running part(app.js), article, news, main, login, signup pages. I have implemented basic frontend table for article and news pages. I have implemented Javascript codes for news, article, login, main, signup pages. I have used Tradingeconomics API and used the functions to get latest news and articles. I have implemented two API GET request functions. One will get latest news with queries country and catalog and the other same but will get articles. Also I have created seven test cases, two for database, two for API that I use, three for API that I have written.</p>
İlker Özkan	<p>As a group member, I often attended group meetings and participated in discussions about assignments and feedback. I joined a part of the tasks and completed it on time. Since the last milestone, I helped my friends to complete the sequence diagrams. Then I contributed to complete some front-end parts.</p>
Ömer Faruk Toptas	<p>Since the last milestone , as a responsible group member I have joined almost every weekly meetings. In general, I take a part in the decision making part of these weekly meetings. I have created first part of the project plan which starts from the beginning of this semester and ends with mock-up part. Also contributed to further parts of the project plan. Then I helped to set some predecessors of the tasks, which are relevant with the features in the project timeline. Then I started to implement the features in the project timeline.</p>

Team Member	Contributions
Ufuk Yılmaz	<p>From the start of the semester, I participated in all of the group meetings(except one) and presented my ideas about the progress of the project. Since the last milestone, I have reviewed the sequence diagrams, and help my friends to complete them. I have created the first part of the project plan which contains the actions from the beginning of this semester to the mock-up part of the project. I learned the usage of ProjectLibre tool and I told my friends how to use this tool. I reviewed the next part of the project plan(We divided the project plan into 3 sections). In the practice-app implementation process, I took part in the front-end team. I implemented the navigation bar and added the top of all web pages. I have implemented the front-end of a part of login.html, homepage.html, currencies.html and I also implement Currency Rates API. I spent a time to learn how to send a requests in the desired form from front-end to back-end. I was responsible for the AWS initialization and the deployment our app on it. I deployed the application to the server. I served as a manual Travis, and provide the project nearly continuous integration.</p>
Eray Kurtuluş	<p>I have attended the group meetings and tried to help the group share the weekly assignments. I took part in the decision making process as much as I could. Apart from some minor personal tasks, I have created a user persona and the related user story, scenario and acceptance criteria. I have also contributed to the creation of sequence diagrams. Further, I have prepared the communication plan section of the milestone report. After the first milestone, I have worked with my teammates on improving the project according to the feedback. I have reviewed the second part of the project plan. I have partaken in the backend team. I researched the frameworks and API's my more experienced teammates have chosen and tried to understand how they worked. Currently I'm working on adding comment functionality and integrating an API to our project, but I failed to get it working. I will fix the problems and add my part to the project repository as soon as I can.</p>

## 5 Project Plan

	<b>Ald</b>	Süre	Başlat	Bitirme	Önceki	Kaynak Adları	
1	✓ <b>Documentation</b>	8 günler	12.02.2019 08:00	19.02.2019 17:00			
2	✓ Repository Initialization	6 günler	12.02.2019 08:00	19.02.2019 17:00	All Team Members		
3	✓ Personal Wikispace	5 günler	12.02.2019 08:00	18.02.2019 17:00	All Team Members		
4	✓ Communication Plan	5 günler	12.02.2019 08:00	18.02.2019 17:00	Omer Faruk Toptaş/Ufuk Yılmaz		
5	✓ Stakeholder Research	4 günler	12.02.2019 08:00	18.02.2019 17:00	All Team Members		
6	✓ Gantt	4 günler	13.02.2019 08:00	18.02.2019 17:00	All Team Members		
7	✓ Issue Customization	6 günler	13.02.2019 08:00	18.02.2019 17:00	Emrah Sarac/İrem Üstünboyacıoğlu/Emre Hoser		
8	✓ <b>The first draft of the Project Requirements</b>	7 günler	19.02.2019 08:00	25.02.2019 17:00	I		
9	✓ <b>Glossary</b>	1 günler	19.02.2019 08:00	22.02.2019 17:00	Mete Han Kurt/İlhan Hoser		
10	✓ <b>Functional Requirements</b>	10 günler	20.02.2019 08:00	27.02.2019 17:00			
11	✓ <b>User Requirements</b>	0,75 günler	20.02.2019 08:00	20.02.2019 15:00	Emrah Sarac/Omer Faruk Dogru/Omer Faruk Toptaş/Ufuk Yılmaz		
12	✓ System Requirements	5 günler	20.02.2019 08:00	22.02.2019 17:00	Emrah Sarac/Omer Faruk Dogru/Omer Faruk Toptaş/Ufuk Yılmaz		
13	✓ <b>Nonfunctional Requirements</b>	6 günler	20.02.2019 08:00	25.02.2019 17:00	Eray Kurkuluk/Mehmet Altay Ince/Ufuk Yılmaz/İrem Üstünboyacıoğlu		
14	✓ Availability and Accessibility	4 günler	20.02.2019 08:00	25.02.2019 17:00	Eray Kurkuluk/Mehmet Altay Ince/Ufuk Yılmaz/İrem Üstünboyacıoğlu		
15	✓ Reliability	4 günler	20.02.2019 08:00	25.02.2019 17:00	Eray Kurkuluk/Mehmet Altay Ince/Ufuk Yılmaz/İrem Üstünboyacıoğlu		
16	✓ Performance	4 günler	20.02.2019 08:00	25.02.2019 17:00	Eray Kurkuluk/Mehmet Altay Ince/Ufuk Yılmaz/İrem Üstünboyacıoğlu		
17	✓ Privacy	4 günler	20.02.2019 08:00	25.02.2019 17:00	Eray Kurkuluk/Mehmet Altay Ince/Ufuk Yılmaz/İrem Üstünboyacıoğlu		
18	✓ Security	4 günler	20.02.2019 08:00	25.02.2019 17:00	Eray Kurkuluk/Mehmet Altay Ince/Ufuk Yılmaz/İrem Üstünboyacıoğlu		
19	✓ <b>The second draft of the Project Requirements</b>	3 günler	26.02.2019 08:00	28.02.2019 17:00	B		
20	✓ <b>Project Requirements</b>	1 günler	26.02.2019 08:00	27.02.2019 17:00	10;13	Emrah Sarac/Omer Faruk Dogru	
21	✓ Update Glossary Section	2 günler	26.02.2019 08:00	27.02.2019 17:00	Emre Hoser		
22	✓ Update group repository	2 günler	27.02.2019 08:00	28.02.2019 17:00	2	Ufuk Yılmaz	
23	✓ <b>UI/UX Scenarios and Mockups</b>	10 günler	25.02.2019 08:00	06.03.2019 17:00	I		
24	✓ <b>Persona and User Scenarios</b>	4 günler	25.02.2019 08:00	28.02.2019 17:00	19		
25	✓ Create Persona	1 gün	25.02.2019 08:00	25.02.2019 17:00	Eray Kurkuluk/Omer Faruk Toptaş/İrem Üstünboyacıoğlu		
26	✓ Write The Personas	3 günler	25.02.2019 08:00	28.02.2019 17:00	25	Eray Kurkuluk/Omer Faruk Toptaş/İrem Üstünboyacıoğlu	
27	✓ Defining Preconditions	3 günler	26.02.2019 08:00	28.02.2019 17:00	25	Eray Kurkuluk/Omer Faruk Toptaş/İrem Üstünboyacıoğlu	
28	✓ Defining Acceptance Criteria	3 günler	26.02.2019 08:00	28.02.2019 17:00	25	Eray Kurkuluk/Omer Faruk Toptaş/İrem Üstünboyacıoğlu	
29	✓ <b>Mockups</b>	5 günler	01.03.2019 08:00	05.03.2019 17:00			
30	✓ <b>Web Mockups 1</b>	5 günler	01.03.2019 08:00	05.03.2019 17:00	24		
31	✓ Login Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
32	✓ Search Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
33	✓ User Profile Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
34	✓ Equipment Information Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
35	✓ Profile Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
36	✓ Event Information Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
37	✓ <b>Web Mockups 2</b>	5 günler	01.03.2019 08:00	05.03.2019 17:00	24		
38	✓ User Profile	5 günler	01.03.2019 08:00	05.03.2019 17:00	Emrah Sarac/Omer Faruk Dogru		
39	✓ Search Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Emrah Sarac/Omer Faruk Dogru		
40	✓ Equipment Information Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Emrah Sarac/Omer Faruk Dogru		
41	✓ Profile Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Emrah Sarac/Omer Faruk Dogru		
42	✓ Alert Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Emrah Sarac/Omer Faruk Dogru		
43	✓ Comment Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Emrah Sarac/Omer Faruk Dogru		
44	✓ <b>Mobile Mockups</b>	5 günler	01.03.2019 08:00	05.03.2019 17:00	24		
45	✓ Login Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
46	✓ Welcome Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
47	✓ Signin Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
48	✓ Location Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
49	✓ Investment Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
50	✓ Profile Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
51	✓ Event Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
52	✓ Review Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
53	✓ Equipment Information Page	5 günler	01.03.2019 08:00	05.03.2019 17:00	Mete Han Kurt/İlhan Hoser		
54	✓ Review Mockups	1 gün	04.03.2019 08:00	04.03.2019 17:00	29	All Team Members	
55	✓ Review User Scenarios	1 gün	06.03.2019 08:00	06.03.2019 17:00	29	All Team Members	
56	✓ Enhance Requirements	1 gün	04.03.2019 08:00	04.03.2019 17:00	29	All Team Members	
57	✓ <b>Review Requirements</b>	22 günler	04.03.2019 08:00	25.03.2019 17:00	I		
58	✓ <b>Use Case Diagrams</b>	6 günler	05.03.2019 08:00	08.03.2019 17:00	Emre Hoser/Mete Han Kurt/Omer Faruk Toptaş		
59	✓ Class Diagram	7 günler	09.03.2019 08:00	15.03.2019 17:00	58	Emrah Sarac/Mehmet Altay Ince/Ufuk Yılmaz/İrem Üstünboyacıoğlu	
60	✓ Sequence Diagram	6 günler	14.03.2019 08:00	19.03.2019 17:00	59	Eray Kurkuluk/Omer Faruk Dogru/Biker Çoban	
61	✓ Review Diagrams	6 günler	21.03.2019 08:00	26.03.2019 17:00	58;59;60	All Team Members	
62	✓ <b>Executive Summary</b>	1 günler	01.04.2019 08:00	02.04.2019 17:00	19;23;57		
63	✓ <b>Executive Summary of Deliverables</b>	7 günler	01.04.2019 08:00	03.04.2019 17:00	Emrah Sarac		
64	✓ List and status of deliverables	7 günler	03.04.2019 08:00	02.04.2019 17:00	Ömer Faruk Toptaş		
65	✓ Evaluation of the Status of Deliverables and Their Impact on the Project	5 günler	20.03.2019 08:00	02.04.2019 17:00	Mete Han Kurt/İrem Üstünboyacıoğlu		
66	✓ Communication plan	7 günler	20.03.2019 08:00	02.04.2019 17:00	Eray Kurkuluk		
67	✓ Modules	7 günler	20.03.2019 08:00	02.04.2019 17:00	Ufuk Yılmaz		
68	✓ Test Cases	7 günler	20.03.2019 08:00	02.04.2019 17:00	Mehmet Altay Ince		
69	✓ Requirements	7 günler	20.03.2019 08:00	02.04.2019 17:00	Omer Faruk Dogru		
70	✓ A summary of work done by each team member	7 günler	20.03.2019 08:00	02.04.2019 17:00	All Team Members		
71	✓ <b>Project Planning</b>	7 günler	01.04.2019 08:00	09.04.2019 17:00	All Team Members		
72	✓ <b>Executive Skeleton of Traders Platform</b>	17 günler	11.04.2019 08:00	29.04.2019 17:00	19;23;57;62		
73	✓ <b>Create Frontend</b>	6 günler	06.04.2019 08:00	03.04.2019 17:00	Ufuk Yılmaz		
74	✓ <b>Skeleton for Frontend</b>	10 günler	20.04.2019 08:00	29.04.2019 17:00	Biker Çoban/Mehmet Altay Ince		
75	✓ <b>Skeleton for Backend</b>	10 günler	20.04.2019 08:00	29.04.2019 17:00	Omer Faruk Dogru/Mete Han Kurt/Eray Kurkuluk		
76	✓ <b>Finding a API</b>	10 günler	20.04.2019 08:00	29.04.2019 17:00	İrem Üstünboyacıoğlu/Mehmet Altay Ince		
77	✓ <b>Implementing Practice-app</b>	0,875 günler	06.05.2019 08:00	07.05.2019 17:00			
78	✓ <b>Unit Test Cases</b>	2 günler	06.05.2019 08:00	08.05.2019 17:00	All Team Members		
79	✓ <b>API Documentation</b>	4 günler	07.05.2019 08:00	07.05.2019 17:00	All Team Members		
80	✓ <b>Integrating found API to our application</b>	3 günler	07.05.2019 08:00	06.05.2019 17:00	76	All Team Members	
81	✓ <b>Milestone 2</b>	5 günler	06.05.2019 08:00	10.05.2019 17:00			
82	✓ <b>Executive Summary</b>	1 günler	06.05.2019 08:00	10.05.2019 17:00			
83	✓ <b>Status of Deliverables</b>	1 günler	06.05.2019 08:00	10.05.2019 17:00	Mete Han Kurt		
84	✓ <b>Evaluation of the Status of Deliverables and Their Impact on the Project</b>	4,875 günler	06.05.2019 08:00	10.05.2019 17:00	Mete Han Kurt		
85	✓ A summary of work done by each team member	5 günler	06.05.2019 08:00	10.05.2019 17:00	All Team Members		
86	✓ <b>Project plan</b>	5 günler	06.05.2019 08:00	10.05.2019 17:00	İrem Üstünboyacıoğlu		
87	✓ <b>Evaluation of tools and processes</b>	5 günler	06.05.2019 08:00	10.05.2019 17:00	Ufuk Yılmaz		
88	✓ <b>API URLs (documented)</b>	5 günler	06.05.2019 08:00	10.05.2019 17:00	79	All Team Members	



## **6 Evaluation of Tools and Processes**

### **6.1 Node.js and its Libraries**

We decided to use Node.js framework in the implementation of our app. Node.js is an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive applications. When you executes something asynchronously, you can move on to another task before it finishes. It has a package manager called Node Package Manager (NPM) provides online repositories for Node.js packages/modules and command line utility to install Node.js packages, do version management and dependency management of Node.js packages. With npm, deployment of the application is very easy process. We used different Node.js modules in our project. One of the most used one is Express.js that is a web application framework for Node.js. We used it in the implementation back-end operations(while giving response the API calls, routing etc.)

### **6.2 MongoDB(DB)**

MongoDB is a database program that is classified as a NoSQL database. It uses JSON-like documents to schema. Main features of the MongoDB can be listed as Ad hoc queries, Indexing, Replication, Load balancing, File storage, Aggregation, Server-side Javascript execution, Transaction which generate a convenience and ease of use.

### **6.3 ProjectLibre**

ProjectLibre is a open-source project management software product that is used to create the project plan of our group. It helps us tasks that are developing a schedule, assigning resources to tasks, tracking progress etc. Schedules can be resource leveled, and chains are visualized in a Gantt chart.

### **6.4 WebStorm**

Since we decided to implement the application in Node.js, and Javascript, we need to IDE that will make this JavaScript implementation process easier. We search for different IDEs and finally decided to use WebStorm which is one of the JetBrains tools. It can be used free with student license. The code completion of the WebStorm is ahead of any other IDE. It has blazingly fast debugger.

### **6.5 Postman**

We used Postman while developing our practice-app. Postman is a application for interacting with HTTP APIs. Through design, testing and full production(It supports

all stages of the API Production), Postman is there for faster, easier API development.

## 6.6 Commit Procedure

### 6.6.1 Motivation

High performance and efficiency requires well documentation and clearly explained commits. Below we will explain "How should we commit" and what are the procedures we should obey in steps.

- Always use your own branches (Never commit to master)
- Atomicity is key for good commit → Each commit should include small functionality.
- All commits should be related(link) to an already open issue(s).
- Each commit should have a clear and short description.

### 6.6.2 Commit Message Template

```
git commit -m " ISSUE_LINK COMMIT_TYPE DESCRIPTION"
ISSUE_LINK -> Related issue number e.g. #3 (All issues have a number like
Database update #70)
COMMIT_TYPE -> Fix: —— Docs: —— Feat: —— Refactor: —— Test:
Fix: -> Fixing a bug or another type of problem
Docs: -> Documentation or commenting
Feat: -> New feature, functionality, is added
Refactor: -> Renaming variables and code re-arranging to improve the maintainability and readability(Functionality can change but behavior should be the same)
Test: -> Testing
```

Example good commits:

```
git commit -m "#23 #12 Feat: Send mail functionality is added."
git commit -m "#12 Docs: Added documentation for select_last_x_commits."
git commit -m "#3 Test: Test cases for bring_the_action is added."
```

For this file to be added -> git commit -m "#72 Docs: Commit template is added"

## 6.7 Pull Request Procedure

Pull requests is a feature of GitHub repository which enable us to tell other members about changes we have pushed to a branch in a repository. This feature of GitHub gives us kind of code reviewing mechanism that before merged into the base branch.

We used pull request while developing practice-app. We follow the sequence that firstly the issue of a task is opened. Then try to resolve this issue in a branch separate from master. Doing necessary changes in this branch, and then commit the changes according to commit procedure(link the commit by issue). When the owner of the branch is thinking that the branch can be merged into master, he/she create the pull request for it. Wait for review pull requests. After getting approval, merge it into the master branch.

## 7 API URL Documentation

**Base URL = 35.166.169.167:8080**

### Getting Articles

Filters by country or/and category and returns articles from last 10. If no filter is defined, returns last 10 articles.

**URL :** /news&articles/articles <br>

**METHOD :** GET <br>

**AUTHORIZATION :** - <br>

**PARAMETERS :** ?country={country name}&category={category name}

### RESPONSE

Returned JSON Object from request

BASE\_URL/news&articles/articles?country=Switzerland

```
{"status":200,"news":[{"title":"Swiss April Inflation Rate Matches Estimates","description":"Switzerland's annual inflation rate came in at 0.7 percent in April 2019, unchanged from the previous month and in line with market expectations. Health and housing prices increased at softer pace while transport inflation accelerated."}]} 
```

### Getting News

Filters by country or/and category and returns news from last 10. If no filter is defined, returns last 10 news

**URL :** /news&articles/news <br>

**METHOD :** GET <br>

**AUTHORIZATION :** - <br>

**PARAMETERS :** ?country={country name}&category={category name}

## RESPONSE

Returned JSON Object from request BASE\_URL/news&articles/news?  
category=Inflation Rate&country=Tunisia

```
{"status":200,"news":[{"title":"Tunisia Inflation Rate at  
14-Month Low of 6.9% in April","description":"The annual i  
nflation rate in Tunisia declined to 6.9 percent in April  
2019 from 7.1 percent in the previous month. It was the lo  
west inflation since February 2018, as inflation slowed ma  
inly for food and non-alcoholic beverages (6.6% vs 7.5% in  
March), in particular (meat (9.8% vs 14.3%), vegetables (8.6% vs 10.2); transport (9.5% vs 9.7%), and recreation &  
culture (4.9% vs 5.2%). Inflation was steady for both hous  
ing & utilities (at 5.3%) and furnishings (at 9.1%). On th  
e other hand, cost rose faster for clothing & footwear (9.  
0% vs 8.9%); restaurants & hotels (10.2% vs 8.8%). On a mo  
nthly basis, consumer prices went up 0.8 percent after inc
```

```
reasing 0.4 percent in the previous month."}]}{
```

## API THAT ALLOWS GETTING CURRENCY RATES

### GET REQUEST

It simply returns the html page that lists the most used 6 currency rates(TRY, USD, GBP, CAD, SEK, CHF).

**URL :** /currencies <br>

**METHOD :** GET <br>

**AUTHORIZATION :** - <br>

Returned HTML page of currencies [35.166.169.167:8080/currencies](http://35.166.169.167:8080/currencies)

```
{<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css" href=".style3.css">
    <link rel="stylesheet" type="text/css" href=".style2.css">
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js" type="text/javascript"></script>
```

```
        <title>Currencies</title>
    </head>
    <body>
        <!-- NAVIGATION BAR STARTS -->
        <ul>
            <li>
                <a href="homepage.html">Home</a>
            </li>
            <li>
                <a href="article.html">Articles</a>
            </li>
            <li class="dropdown">
                <a href="javascript:void(0)" class="dropb
tn">Trading Equipment</a>
                <div class="dropdown-content">
                    <a href="currencies.html">Currencie</a>
                    <a href="cryptocurrencies.html">Crypt
ocurrencies</a>
                </div>
            <li>
                <a href="news.html">News</a>
            </li>
            <li style="float:right">
                <a href="about.html">About</a>
            </li>
        </ul>
        <!-- NAVIGATION BAR ENDS -->
```

```
<h1>
    <span class="yellow">Currencies (Base: EUR
)</span>
</h1>
<table class="container">
    <thead>
        <tr>
            <th>
                <h1>Currency Name</h1>
            </th>
            <th>
                <h1>Exchange Rate</h1>
            </th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>TRY</td>
            <td id="TRY_ER"></td>
        </tr>
        <tr>
            <td>USD</td>
            <td id="USD_ER"></td>
        </tr>
        <tr>
            <td>GBP</td>
            <td id="GBP_ER"></td>
        </tr>
    </tbody>
</table>
```

```

        </tr>
        <tr>
            <td>CAD</td>
            <td id="CAD_ER"></td>
        </tr>
        <tr>
            <td>SEK</td>
            <td id="SEK_ER"></td>
        </tr>
        <tr>
            <td>CHF</td>
            <td id="CHF_ER"></td>
        </tr>
    </tbody>
</table>
<script>
$(document).ready(function(){
    console.log("dsadas")
    $.post("/currencies",
    {
        },
        function (data, status) {
            var curr = JSON.parse(data);
            console.log(curr)
            $("#TRY_ER").text(curr.rates.TRY);
            $("#USD_ER").text(curr.rates.USD);
            $("#GBP_ER").text(curr.rates.GBP);
            $("#CAD_ER").text(curr.rates.CAD);
        }
    );
});

```

```

        $("#SEK_ER").text(curr.rates.SEK);
        $("#CHF_ER").text(curr.rates.CHF);
    });
});

</script>
</body>
</html>

```

## Getting Currency Rates (POST REQUEST)

Returns real-time exchange rate data for all available currencies.

**URL :** /currencies <br>

**METHOD :** POST <br>

**AUTHORIZATION :** - <br>

## RESPONSE

Returned JSON Object from request POST`localhost:8080/currencies

```
{"success":true,"timestamp":1557682325,"base":"EUR","date"
:"2019-05-12","rates":{"AED":4.127029,"AFN":87.917805,"ALL"
:123.372846,"AMD":541.742872,"ANG":2.106548,"AOA":365.305
877,"ARS":50.349688,"AUD":1.604883,"AWG":2.022392,"AZN":1.
915665,"BAM":1.955994,"BBD":2.248788,"BDT":94.785016,"BGN"
:1.955656,"BHD":0.42368,"BIF":2062.840225,"BMD":1.123551,"
```

BND":1.517637,"BOB":7.763347,"BRL":4.446117,"BSD":1.123383  
,"BTC":0.000177,"BTN":78.672405,"BWP":11.977191,"BYN":2.35  
8728,"BYR":22021.605892,"BZD":2.264631,"CAD":1.507975,"CDF":  
1837.00591,"CHF":1.136697,"CLF":0.027933,"CLP":770.75663  
7,"CNY":7.667225,"COP":3686.147174,"CRC":668.490242,"CUC":  
1.123551,"CUP":29.77411,"CVE":110.595088,"CZK":25.725501,"  
DJF":199.677422,"DKK":7.467235,"DOP":56.947252,"DZD":134.1  
96812,"EGP":19.234754,"ERN":16.853376,"ETB":32.588607,"EUR":  
1,"FJD":2.422882,"FKP":0.862781,"GBP":0.864539,"GEL":3.0  
84144,"GGP":0.864496,"GHS":5.80318,"GIP":0.862781,"GMD":55  
.733467,"GNF":10364.761224,"GTQ":8.606066,"GYD":234.777425  
, "HKD":8.818474,"HNL":27.752098,"HRK":7.390694,"HTG":98.26  
0744,"HUF":323.144615,"IDR":16088.693141,"ILS":3.999787,"IM  
P":0.864496,"INR":78.572758,"IQD":1337.026072,"IRR":47307  
.12856,"ISK":136.803852,"JEP":0.864496,"JMD":153.42106,"JO  
D":0.796646,"JPY":123.528872,"KES":113.6019,"KGS":78.47849  
, "KHR":4556.000426,"KMF":492.230006,"KPW":1011.777925,"KRW":  
1320.397132,"KWD":0.341676,"KYD":0.936283,"KZT":427.0172  
32,"LAK":9690.629918,"LBP":1694.97717,"LKR":198.126674,"LR  
D":198.138251,"LSL":16.13442,"LTL":3.317555,"LVL":0.679625  
, "LYD":1.561816,"MAD":10.825082,"MDL":20.131809,"MGA":4044  
.784449,"MKD":61.716113,"MMK":1721.73394,"MNT":2955.537626  
, "MOP":9.081496,"MRO":401.107126,"MUR":39.219244,"MVR":17.  
302643,"MWK":811.883772,"MXN":21.46354,"MYR":4.672854,"MZN":  
71.766844,"NAD":16.134177,"NGN":404.478343,"NIO":37.2459  
37,"NOK":9.788944,"NPR":125.56248,"NZD":1.703864,"OMR":0.4  
3264,"PAB":1.123495,"PEN":3.724599,"PGK":3.776184,"PHP":58  
.643757,"PKR":159.122986,"PLN":4.296795,"PYG":7150.83174,"

```
QAR":4.090871,"RON":4.760147,"RSD":117.972904,"RUB":73.163  
078,"RWF":1016.813946,"SAR":4.213711,"SBD":9.197671,"SCR":  
15.367905,"SDG":50.692952,"SEK":10.816895,"SGD":1.531288,"  
SHP":1.484103,"SLL":9943.429442,"SOS":654.466189,"SRD":8.3  
79442,"STD":23651.429,"SVC":9.830906,"SYP":578.629233,"SZL":  
16.134204,"THB":35.453098,"TJS":10.597111,"TMT":3.943665  
, "TND":3.372119,"TOP":2.568495,"TRY":6.723223,"TTD":7.6144  
76,"TWD":34.774996,"TZS":2584.63099,"UAH":29.417961,"UGX":  
4241.237147,"USD":1.123551,"UYU":39.39194,"UZS":9499.62657  
7,"VEF":11.221469,"VND":26232.114467,"VUV":127.760523,"WST":  
3.001761,"XAF":656.164825,"XAG":0.076008,"XAU":0.000874,  
"XCD":3.036453,"XDR":0.810655,"XOF":656.154055,"XPF":119.9  
38978,"YER":281.227461,"ZAR":15.909268,"ZMK":10113.309813,  
"ZMW":14.438015,"ZWL":362.182398} }
```