CMPE 352 Milestone 2 Report

Group 11

12 May 2019

Contents

1 Executive Summary

1.1 Introduction

Our project is a social platform for people who consider themselves as traders. People can signup and log in the platform to take advantage of more advanced features such as buying and selling trading equipment. However, users can use searching functionality without login and signup. Guest users also can view articles, events, trading equipment, and comments. Fundamentally, there are two types of users: basic and trader. Basic users can create, comment and rank on the articles. In addition, they can create and share portfolios. Not only that, they can add trading equipment to the created portfolios. They can also chase economic events and filter them by country and significance levels. Our platform has more advanced features such as setting an alert, making a prediction about trading equipment and following trading equipment. Trader users have all the functionality that basic users have and they have extra features such as trading indices, stocks, ETFs, commodities, currencies, funds, bonds, and cryptocurrencies. They have an extra ”My Investments” section that enables them to set buy and sell order. Another cool feature of the platform is people can follow other people. In a nutshell, our platform creates a connection between people who are interested in trading and trading world. Stay tuned!

As we approach to the end of the semester and the course, our group's last assignment was to implement **useful and simple** APIs. We think that, reason we were assigned with this implementation task is to give a headstart to next term's actual project. Since our group is assigned to the *Traders' Platform* project, we had to implement simple and related API's to a potential *Traders' Platform* web application. So, in our meeting beginning of the assignment term, we gathered and did brainstorming on the APIs we could choose. Outcome was clear and straightforward. The crucial tools that could be helpful in the actual platform are **Login-Register system, stock indices system, exchange rates system and economic news and articles system.** Then, we divided our 11-people group to 3 groups to implement these simple APIs. We have implemented those APIs with some extended features and simple front-end. Following milestone report will be about the features of the APIs we implemented and the **documentation** of them.

1.2 What is done so far

- We set our base requirements for the platform based on customer's needs.
- We created 3 different personas and outlined different usage scenarios of the platform to show possible actions that can be taken.
- We supported these user scenarios by visualizing them as Mockups.
- We drew Use Case diagrams that explicitly shows which users can do which actions. The diagram made use of inheretence, extends and includes.
- We designed the Class Diagram in order to specify the basic design principles of the platform.

- We depicted some of the actions in Sequence diagrams which show what happens behind the scene step by step using the design made in Class Diagram.
- We implemented 4 different APIs: Stock Indexes API, Exchange Rates API, Login/Register API, and News API using express.js framework and Mongo database.
- We created different front ends for our APIs. We also create home page for our project.
- We wrote unit tests for our APIs using Chai.js framework.
- We deployed our APIs to Amazon Web Services.

We successfuly completed the tasks asked for the both 1st and 2nd Milestone. We are in close contact with our customer and frequently get feedback. We take the feedback serious and review our deliverables and make the necessary changes.

1.3 Road Ahead

Until now, we have been working on the plan of the project, hard and periodically. For the past weeks, we proceeded with the picture we have drawn in the first place and shaped our project via the incoming feedback from the customer. For the future, our team aims to stick to its plan and try to implement the project requirements. We will be collecting feedback from the customer and add the incoming requirements, project parts and knowledge from the course instructor to our project. Our team aims to turn the design and plan knowledge to reality in the next term. We will be dividing the work load evenly to group members and try to match the skills of the developers and designers to the right places in the development process. As a team, we believe that we will create a great platform to use and reach the customers according to our plan. We believe that communication and team work will bring the success in our Traders' Platform.

2 Status of Deliverables

Name	Delivery Date	Status
Communication Plan	Mar 4, 2019	Delivered
Requirements	Feb 25, 2019	Delivered
Personas & Mockups	Mar 4, 2019	Delivered
Milestone 1 Report	Apr 1, 2019	Delivered
Project Plan	Apr 8, 2019	Delivered
Implementation Assignment	May 6, 2019	Delivered

3 Evaluation Of The Status Of Deliverables

Project Plan

For the project plan, we first wrote the tasks we have already completed starting from the beginning of the semester. These completed tasks include, project initialization creating communication plan, creating draft requirements, creating wiki pages, revisions for the requirements, user scenarios and mock-ups, use case and sequence diagrams, revisions of diagrams and mock-ups, writing the milestone 1 report and preparing the project plan. After listing these tasks, we wrote our estimated plan for the future implementation task. Looking at it, we almost perfectly complied with the plan. We completed the tasks in the order they are given on the project plan and meet the deadline. Project is a vital part for a team of 11 people and we are happy to see that we were able to follow it.

Implementation Assignment

Web development was a new concept for most of the team. We first discussed what language and framework we should use for our implementation task. We first chose Django with Python because learning Python seemed easier compared to other languages. But, after a while, we thought that Django is too complex for a simple project like ours and made a switch to Express.js with Node.js. We planned to have APIs for sign up, sign in, stock indexes and exchange rates. We divided our team into 3 groups for these tasks. We started a bit late so we acted quickly to complete the APIs. In the end, we created the APIs and deployed our practice app to AWS. It was a good challenge to see what we are up to in the next semester with CMPE 451. We learned so many new concepts and certain that it will help us a lot in the next semester.

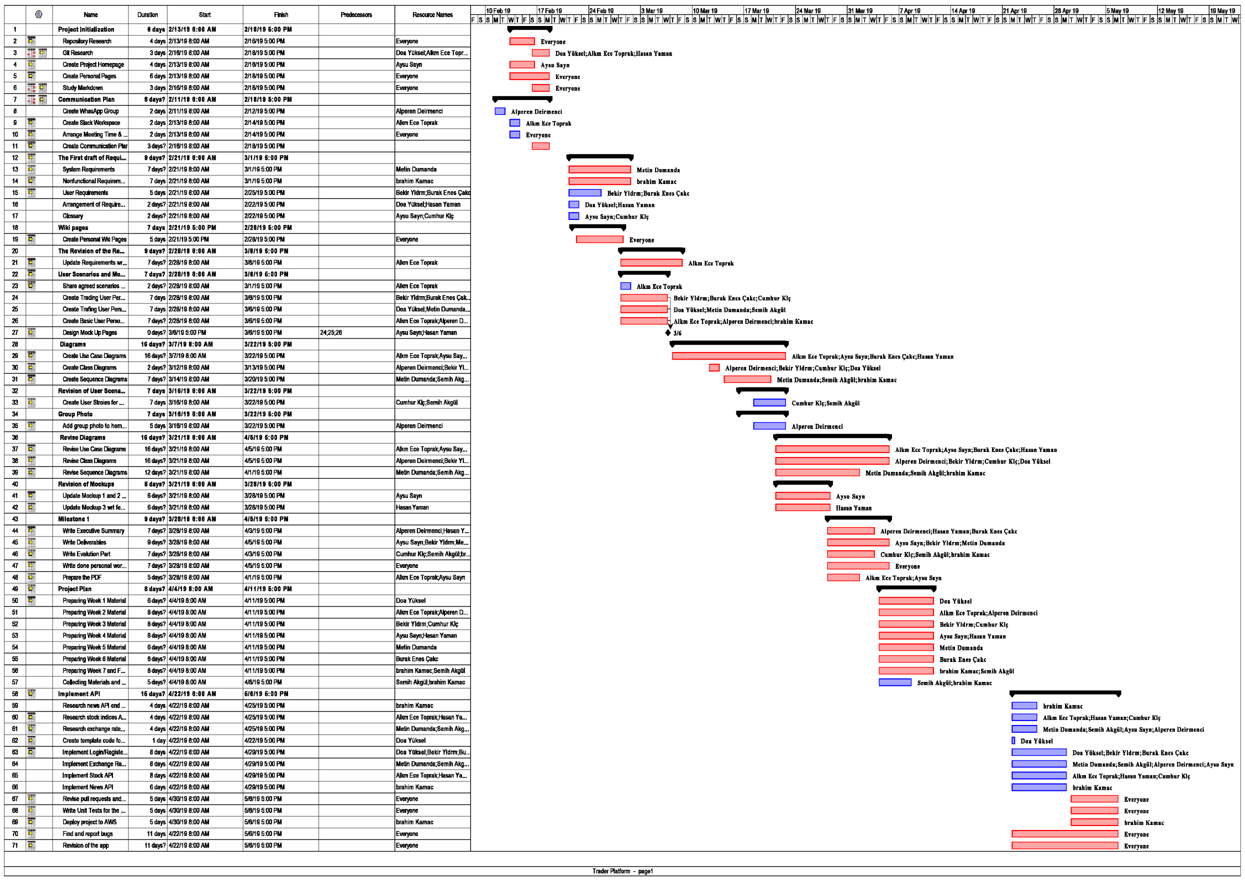
4 Work Done

Name	Work Done
Alkım Ece Toprak	Updated the requirements according to the first meeting with the customer. Helped create persona 1's back story and worked on the scenario. Worked on the use case diagram for signing up, logging in and requirements for the portfolios. Prepared and finalized the first Milestone Report. I added the database integration to the Stock Index API. Implemented a basic front end for the API. I also created a schema to be used in the creation of the database collection. Worked with Hasan to review and merge.
Alperen Değirmenci	I created the Communication plan page. Wrote down the first questions to the customer. Then, took part in writing the User Scenario 1 and its story. I also took part in the Class diagram group. And prepared the Road Ahead part for the Milestone report. I added the percentage change functionality to the exchange rates API. Added the front-end part for my API part. Reviewed each other's code with Metin, Aysu and Semih. Added unit tests for the part I implemented. Updated the Introduction part in the Executive Summary, for Milestone 2 report.
Aysu Sayın	Updated read.me according to the updates on the project. Took part in writing glossary with Cumhur. Created the mockup for persona 1. Created the events page for third mockup which I worked on with Hasan. Took part in making use case diagram: follow user, profile and improved some other parts using Lucidchart. Gathered the materials of our project and prepared the first Milestone Report. I added mockups week to project plan Implemented Exchange RatesAPI with Metin, Alperen and Semih. I worked on calculating average exchange rates. Added some unit tests for the part that I implemented. Also made additions to the frontend for my part of API.
Bekir Yıldırım	Wrote the first version of the User Requirements. Helped create the fabrication person, Persona 3, and make story and scenario for this. Worked on the class diagram and have drawn the diagram that we created with our teammates by LucidChart. Also helped write list, status and evaluation of deliverables. I worked on Sign Up system with Doğa and Bekir. Also worked on Login system. Created html page for Login. I added Register and Login buttons to home page. I contributed to Executive summary and wrote Register API Documentation parts of Milestone 2.
Burak Enes Çakıcı	I wrote the first version of the User Requirements under Functional Requirements. I drew the Use Case diagram for the Guest user and helped for Basic & Trader users'. I summarized the work done until the first Milestone for the report. Write down the introduction part in milestone report. I worked on Sign Up system with Doğa and Bekir. Also worked on Login system. Created html page for Login. I added Register and Login buttons to home page. I contributed to Executive summary and wrote Register API Documentation parts of Milestone 2.

Cumhur Kılıç	Helped the writing glossary and updating the glossary according to given feedback. Took part in writing i writing the User Scenario 3 and its story. Then I expanded the Persona 3 scenario and its user story. Determined the absent accaptence criterias in the Persona 3. Worked on the class diagram and figuring out relations between classes. I changed some parts of class diagram according to feedbacks. Worked on Stock API and rewiewed others pull request I wrote Evaluation Of Tools and Project Management parts of Milestone 2 report.
Doğa Yüksel	Wrote down remote repository part of git research page. Helped update the requirements according to feedback. Worked on the class diagram and figuring out relations between classes. Provided brief explanation of Express.js framework and a small template code for my group mates to look at. Worked on register view and wrote a test for that view. I also helped finalize the Milestone 2 report and helped update project plan.
Hasan Yaman	Did research and wrote down about git. Helped writing of requirements. Created mockup for scenario 2 and scenario 3 except events page. Create use case diagram for the trader user and some part of the basic user. Changed some parts of mockups and created new mockups according to feedbacks. Also reorganized some parts of use case according to feedbacks. I wrote the logic of Stock Index API and documentation of it. I wrote unit tests for Stock Index API. I reviewed others pull requests. I wrote the work done so far part in Milestone 2 report.
İbrahim Kamacı	Took responsibility in writing user scenario of basic user and wrote acceptance criteria of it.Prepared sequence diagram (follow a user) with Lucidchart tool according to class diagram of our project. Prepared evaluation of tools and project management part in our first milestone report.
Metin Dumandağ	Wrote the system requirements. Enumerated requirements. Wrote the preconditions, scenario and acceptance criteria for the persona 2. Created sequence diagrams for creating an article, searching and creating a portfolio. Wrote the evaluation for the requirements, personas & mockups and diagrams for the deliverables section of the milestone report. Added Meeting 5 to the project plan. Created the home page and layout for the project. Wrote the exchange rate API between the two currencies. Wrote a frontend for that API. Wrote the documentation for that API. Wrote frontend, API and database cache tests for that API. Fixed some bugs on the codes written by others and reviewd PRs. Added evaluation for the deliverables project plan and implementation assignment. for the milestone 2 report

Semih Akgül	Attended the first three customer meetings, informed my friends about the discussions during the meetings also updated Customer Meeting page and added the new notes. Made the second user's persona and created its user story. Determined the absent acceptance criterias in the Personas page. Created and updated the sequence diagrams about registering, making comments and making predictions. Prepared the evaluation part of the milestone report 1. Prepared project plan in project libre. Created All Exchange Rates API and front-end part

5 Project Plan



		Name	Duration	Start	Finish	Predecessors	Resource Names
1		**Project Initialization**	**6 days**	**2/13/19 8:00 AM**	**2/18/19 5:00 PM**		
2		Repository Research	4 days	2/13/19 8:00 AM	2/16/19 5:00 PM		Everyone
3		Git Research	3 days	2/16/19 8:00 AM	2/18/19 5:00 PM		Doa Yüksel;Alkm Ece Topr...
4		Create Project Homepage	4 days	2/13/19 8:00 AM	2/16/19 5:00 PM		Aysu Sayn
5		Create Personal Pages	6 days	2/13/19 8:00 AM	2/18/19 5:00 PM		Everyone
6		Study Markdown	3 days	2/16/19 8:00 AM	2/18/19 5:00 PM		Everyone
7		**Communication Plan**	**8 days?**	**2/11/19 8:00 AM**	**2/18/19 5:00 PM**		
8		Create WhasApp Group	2 days	2/11/19 8:00 AM	2/12/19 5:00 PM		Alperen Deirmenci
9		Create Slack Workspace	2 days	2/13/19 8:00 AM	2/14/19 5:00 PM		Alkm Ece Toprak
10		Arrange Meeting Time & ...	2 days	2/13/19 8:00 AM	2/14/19 5:00 PM		Everyone
11		Create Communication Plar	3 days?	2/16/19 8:00 AM	2/18/19 5:00 PM		
12		**The First draft of Requi...**	**9 days?**	**2/21/19 8:00 AM**	**3/1/19 5:00 PM**		
13		System Requirements	7 days?	2/21/19 8:00 AM	3/1/19 5:00 PM		Metin Dumanda
14		Nonfunctional Requirem...	7 days	2/21/19 8:00 AM	3/1/19 5:00 PM		brahim Kamac
15		User Requirements	5 days	2/21/19 8:00 AM	2/25/19 5:00 PM		Bekir Yldrm;Burak Enes Çakc
16		Arrangement of Require...	2 days?	2/21/19 8:00 AM	2/22/19 5:00 PM		Doa Yüksel;Hasan Yaman
17		Glossary	2 days?	2/21/19 8:00 AM	2/22/19 5:00 PM		Aysu Sayn;Cumhur Klç
18		**Wiki pages**	**7 days**	**2/21/19 5:00 PM**	**2/28/19 5:00 PM**		
19		Create Personal Wiki Pages	5 days	2/21/19 5:00 PM	2/28/19 5:00 PM		Everyone
20		**The Revision of the Re...**	**9 days?**	**2/28/19 8:00 AM**	**3/8/19 5:00 PM**		
21		Update Requirements wr...	7 days?	2/28/19 8:00 AM	3/8/19 5:00 PM		Alkm Ece Toprak
22		**User Scenarios and Mo...**	**7 days?**	**2/28/19 8:00 AM**	**3/6/19 5:00 PM**		
23		Share agreed scenarios ...	2 days?	2/28/19 8:00 AM	3/1/19 5:00 PM		Alkm Ece Toprak
24		Create Trading User Per...	7 days	2/28/19 8:00 AM	3/6/19 5:00 PM		Bekir Yldrm;Burak Enes Çak...
25		Create Trafing User Pers...	7 days	2/28/19 8:00 AM	3/6/19 5:00 PM		Doa Yüksel;Metin Dumanda...
26		Create Basic User Perso...	7 days?	2/28/19 8:00 AM	3/6/19 5:00 PM		Alkm Ece Toprak;Alperen D...
27		Design Mock Up Pages	0 days?	3/6/19 5:00 PM	3/6/19 5:00 PM	24;25;26	Aysu Sayn;Hasan Yaman
28		**Diagrams**	**16 days?**	**3/7/19 8:00 AM**	**3/22/19 5:00 PM**		
29		Create Use Case Diagrams	16 days?	3/7/19 8:00 AM	3/22/19 5:00 PM		Alkm Ece Toprak;Aysu Say...
30		Create Class Diagrams	2 days?	3/12/19 8:00 AM	3/13/19 5:00 PM		Alperen Deirmenci;Bekir Yl...
31		Create Sequence Diagrams	7 days?	3/14/19 8:00 AM	3/20/19 5:00 PM		Metin Dumanda;Semih Akg...
32		**Revision of User Scena...**	**7 days**	**3/16/19 8:00 AM**	**3/22/19 5:00 PM**		
33		Create User Stroies for ...	7 days	3/16/19 8:00 AM	3/22/19 5:00 PM		Cumhur Klç;Semih Akgül
34		**Group Photo**	**7 days**	**3/16/19 8:00 AM**	**3/22/19 5:00 PM**		
35		Add group photo to hom...	5 days	3/16/19 8:00 AM	3/22/19 5:00 PM		Alperen Deirmenci
36		**Revise Diagrams**	**16 days?**	**3/21/19 8:00 AM**	**4/5/19 5:00 PM**		
37		Revise Use Case Diagrams	16 days?	3/21/19 8:00 AM	4/5/19 5:00 PM		Alkm Ece Toprak;Aysu Say...
38		Revise Class Diagrams	16 days?	3/21/19 8:00 AM	4/5/19 5:00 PM		Alperen Deirmenci;Bekir Yl...
39		Revise Sequence Diagrams	12 days?	3/21/19 8:00 AM	4/1/19 5:00 PM		Metin Dumanda;Semih Akg...
40		**Revision of Mockups**	**8 days?**	**3/21/19 8:00 AM**	**3/28/19 5:00 PM**		
41		Update Mockup 1 and 2 ...	6 days?	3/21/19 8:00 AM	3/28/19 5:00 PM		Aysu Sayn
42		Update Mockup 3 wrt fe...	6 days?	3/21/19 8:00 AM	3/28/19 5:00 PM		Hasan Yaman
43		**Milestone 1**	**9 days?**	**3/28/19 8:00 AM**	**4/5/19 5:00 PM**		
44		Write Executive Summary	7 days?	3/28/19 8:00 AM	4/3/19 5:00 PM		Alperen Deirmenci;Hasan Y...
45		Write Deliverables	9 days?	3/28/19 8:00 AM	4/5/19 5:00 PM		Aysu Sayn;Bekir Yldrm;Me...
46		Write Evalution Part	7 days?	3/28/19 8:00 AM	4/3/19 5:00 PM		Cumhur Klç;Semih Akgül;br...
47		Write done personal wor...	7 days?	3/28/19 8:00 AM	4/5/19 5:00 PM		Everyone
48		Prepare the PDF	5 days?	3/28/19 8:00 AM	4/1/19 5:00 PM		Alkm Ece Toprak;Aysu Sayn
49		**Project Plan**	**8 days?**	**4/4/19 8:00 AM**	**4/11/19 5:00 PM**		
50		Preparing Week 1 Material	6 days?	4/4/19 8:00 AM	4/11/19 5:00 PM		Doa Yüksel
51		Preparing Week 2 Material	8 days?	4/4/19 8:00 AM	4/11/19 5:00 PM		Alkm Ece Toprak;Alperen D...
52		Preparing Week 3 Material	8 days?	4/4/19 8:00 AM	4/11/19 5:00 PM		Bekir Yldrm;Cumhur Klç
53		Preparing Week 4 Material	8 days?	4/4/19 8:00 AM	4/11/19 5:00 PM		Aysu Sayn;Hasan Yaman
54		Preparing Week 5 Material	6 days?	4/4/19 8:00 AM	4/11/19 5:00 PM		Metin Dumanda
55		Preparing Week 6 Material	6 days?	4/4/19 8:00 AM	4/11/19 5:00 PM		Burak Enes Çakc
56		Preparing Week 7 and F...	8 days?	4/4/19 8:00 AM	4/11/19 5:00 PM		brahim Kamac;Semih Akgül
57		Collecting Materials and ...	5 days?	4/4/19 8:00 AM	4/8/19 5:00 PM		Semih Akgül;brahim Kamac
58		**Implement API**	**15 days?**	**4/22/19 8:00 AM**	**5/6/19 5:00 PM**		
59		Research news API end ...	4 days	4/22/19 8:00 AM	4/25/19 5:00 PM		brahim Kamac
60		Research stock indices A...	4 days	4/22/19 8:00 AM	4/25/19 5:00 PM		Alkm Ece Toprak;Hasan Ya...
61		Research exchange rate...	4 days	4/22/19 8:00 AM	4/25/19 5:00 PM		Metin Dumanda;Semih Akg...
62		Create template code fo...	1 day	4/22/19 8:00 AM	4/22/19 5:00 PM		Doa Yüksel
63		Implement Login/Registe...	8 days	4/22/19 8:00 AM	4/29/19 5:00 PM		Doa Yüksel;Bekir Yldrm;Bu...
64		Implement Exchange Ra...	8 days	4/22/19 8:00 AM	4/29/19 5:00 PM		Metin Dumanda;Semih Akg...
65		Implement Stock API	8 days	4/22/19 8:00 AM	4/29/19 5:00 PM		Alkm Ece Toprak;Hasan Ya...
66		Implement News API	6 days	4/22/19 8:00 AM	4/29/19 5:00 PM		brahim Kamac
67		Revise pull requests and...	5 days	4/30/19 8:00 AM	5/6/19 5:00 PM		Everyone
68		Write Unit Tests for the ...	5 days	4/30/19 8:00 AM	5/6/19 5:00 PM		Everyone
69		Deploy project to AWS	5 days	4/30/19 8:00 AM	5/6/19 5:00 PM		brahim Kamac
70		Find and report bugs	11 days	4/22/19 8:00 AM	5/6/19 5:00 PM		Everyone
71		Revision of the app	11 days?	4/22/19 8:00 AM	5/6/19 5:00 PM		Everyone

6 Evaluation Of Tools and Project Management

6.1 Evaluation Of Tools

- **Github:** GitHub is a web-based version-control and collaboration platform for software developers.Github is our main base while designing our Trader project. All tasks that are agreed upon in weekly meeting assigned to team members via Github platform.

We used wikipages like a gathering point especially when a task consists of subtasks. We did issue management properly in our Github page and it made our work follow easy.

- **Mockups:** We designed and created our mockups by using moqups.com. Moqups is a streamlined web app that helps developers create and collaborate on wireframes, mockups, diagrams and prototypes. The platform has mockup template therefore even we had not known how to design and create a mockup, We had created mobile and web mockups.

- **Lucidchart:** Lucidchart is an online diagramming tool that provides individuals with a web-based flowchart platform loaded with multiple features and capabilities. We used while creating the diagrams. After creating the first diagram you became familiar with the tool quickly and the following ones are done easily.

- **ProjectLibre:** ProjectLibre is an open source, freely available project management software system intended ultimately as a standalone replacement for Microsoft Project. We used it for generating project plans. It is actually hard to learn and it has a really bad UI. However, it is open-source and once it is learned, project management can be done quickly in later times. Also, it is easy to show assignees of a task, dependencies among tasks, start-finish times of tasks and their visual representation.

- **Node.js Express:** Nodejs is our server side strategy for this application. This is framework used for back-end development. Express is very handy for api and basic server development. It's too early in the development phase to see if the choice was absolutely correct, but so far it did not cause any problems during coding. Express' ability to deploy a working app very quickly is a boon, as well.

- **Postman:** Easy-to-use and powerful tool for testing API's.

- **MongoDB:** MongoDB is an open source database management system (DBMS) that uses a document-oriented database model which supports various forms of data. It is one of numerous nonrelational database technologies which arose in the mid-2000s under the NoSQL banner for use in big data applications and other processing jobs involving data that doesn't fit well in a rigid relational model. Instead of using tables and rows as in relational databases, the MongoDB architecture is made up of collections and documents.

6.2 Project Management

Working as a team is a lot harder than working as individuals. First of all, we were newly met people and had to adapt to each other and feel comfortable enough to express our ideas and ask questions about the current subjects if existed. On the contrary of timidity, there may be conflicts among group members maybe rising from problems such as lack of participation of some members or unequal distribution of roles. For example, when we chose Node.js Express as a framework, We could not decide which one to choose for a short time.

However, after our research and discussions, our final decision is to use express.js for the API.

In our group, we have a warm environment that any member can freely contribute to the project and ask questions at any time. Duties were distributed based on volunteering. Knowing and adapting to each other or making jobs done were not challenging for our group. What most challenged us during this process was setting a proper time and coming together because each of 11 team member had to set a common free time for weekly meetings. We could decide on that time but that was the only option.

Understanding given assignments were sometimes confusing and they had ambiguous points which necessitate contacting the customer. Within PS hours we could ask our questions. Another problem rising at this point was the inconvenient PS hours for us. We could not attend every PS but we contacted the assistants via Piazza when we had questions. This is a faster and easier way of getting answers.

We believe we've achieved all the assignments and built a core team who're skilled to do work or open to learning new things. We believe the next part of course will be much smoother for us. We're aiming for lean development so changes will be easier

7 API - Documentation

Here is our demo app: http://18.223.112.209:3000

7.1 Exchange Rate API

7.1.1 URL

/api/exchangerate

7.1.2 Parameters

- **from**: Symbol of the base currency. Examples: EUR, USD, TRY
- **to**: Symbol of the currency to be converted. Examples: EUR, USD, TRY

from parameter is optional. When it is not given, it will be assumed as TRY.

7.1.3 Return Format

```
{
  "from": FROM-PARAMETER,
  "to": {
    TO-PARAMETER: RATE
  }
}
```

Example

```
{
  "from": "USD",
  "to": {
    "TRY": 5.974361273
  }
}
```

7.1.4 Request Type

Only GET allowed.

7.1.5 Examples

1. Request

   ```
   GET example.com/api/exchangerate?from=USD&to=EUR
   ```

   Response

   ```
   {
     "from": "USD",
     "to": {
       "EUR": 0.896458987
     }
   }
   ```

2. Request

   ```
   GET example.com/api/exchangerate?to=JPY
   ```

   Response

   ```
   {
     "from": "TRY",
     "to": {
       "JPY": 18.6663465578
     }
   }
   ```

7.2 Average Exchange Rate API

Usage of the API is also described in th README.MD

7.2.1 URL

/api/exchangerate/avg

7.2.2 Parameters

- **from:** Symbol of the base currency. Examples: ‘EUR‘, ‘USD‘, ‘TRY‘
- **to:** Symbol of the currency to be converted. Examples: ‘EUR‘, ‘USD‘, ‘TRY‘
- **start_date:** The date to start the average calculation from. In YYYY-MM-DD format.
- **end_date:** The date to end the average calculation. In YYYY-MM-DD format.
- **format:** This is return format. It can be ’json’ or ’html’. If you set this to ’json’, it will return a JSON object. Otherwise, it will return a HTML page which shows the val

from parameter is optional. When it is not given, it will be assumed as ‘TRY‘. start_date parameter is optional. When it is not given, it will be assumed as 7 days before current date. end_date parameter is optional. When it is not given, it will be assumed as current date

7.2.3 Return Format

```
{
  "from": FROM-PARAMETER,
  "to": TO-PARAMETER,
  "start_date": START_DATE-PARAMETER,
  "end_date": END_DATE-PARAMETER,
  "average": AVERAGE-PARAMETER
}
```

Example

```
{
  "from":"USD",
  "to":"TRY",
  "start_date":"2019-04-28",
  "end_date":"2019-05-05",
  "average":5.960330369399999
}
```

7.2.4 Request Type

Only ‘GET‘ allowed.

7.2.5 Examples

1. **Request**

```
GET example.com/api/exchangerate/avg?from=EUR&to=TRY&start\
   _date=2019-05-01&end\_date=2019-05-05&format=json
```

Response

```
{
  "from":"EUR",
  "to":"TRY",
  "start_date":"2019-05-01",
  "end_date":"2019-05-05",
  "average":6.6758
}
```

2. **Request**

```
GET example.com/api/exchangerate/avg?from=EUR&to=AUD&start\
   _date=&end\_date=&format=json
```

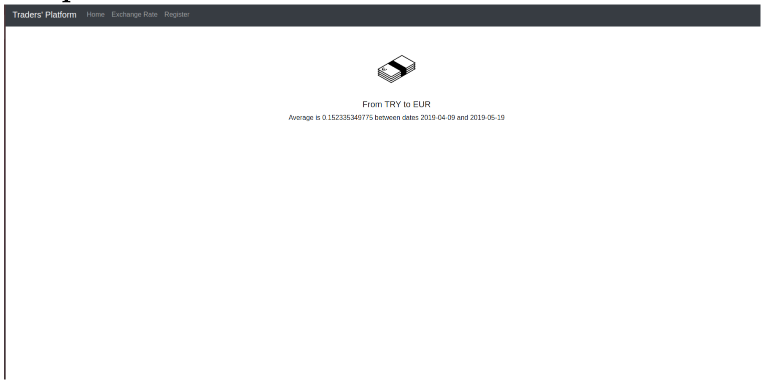
Response

```
{
  "from":"EUR",
  "to":"AUD",
  "start_date":
  "2019-04-28",
  "end_date":"2019-05-05",
  "average":1.5912250000000001
}
```

3. **Request**

```
GET example.com/api/exchangerate/avg?from=TRY&to=EUR&start\
   _date=2019-04-09&end\_date=2019-05-19&format=html
```

Response



From TRY to EUR

Average is 0.152335349775 between dates 2019-04-09 and 2019-05-19

7.3 Percentage change Exchange Rate API

7.3.1 URL

/api/exchangerate/percentage

7.3.2 Parameters

- **from**: Symbol of the base currency. Examples: EUR, USD, TRY
- **to**: Symbol of the currency to be converted. Examples: EUR, USD, TRY
- **change_date**: The date to see the exchange rate's percentage change from the previous day. In YYYY-MM-DD format.
- **prev_day**: The previous day of the chosen day to use. In YYYY-MM-DD format.
- **format**: This is return format. It can be 'json' or 'html'. If you set this to 'json', it will return a JSON object. Otherwise, it will return a HTML page which shows the values.

from parameter is optional. When it is not given, it will be assumed as TRY. change_date parameter is optional. When it is not given, it will be assumed as the current day.

7.3.3 Return Format

```
{
  "from": FROM-PARAMETER,
  "to": TO-PARAMETER,
  "change_date": CHANGE_DATE-PARAMETER,
}
```

Example

```
{
  "from":"USD",
  "to":"TRY",
  "change_date":"2019-05-08",
  "% change":0.6048985065072967
}
```

7.3.4 Request Type

Only GET allowed.

7.3.5 Examples

Request

```
GET example.com/api/exchangerate/percentage?from=USD&to=TRY&
   change_date=2019-05-08&format=json
```

Response

```
{
  "from":"USD",
  "to":"TRY",
  "change_date":"2019-05-08",
  "% change":0.6048985065072967
}
```

7.4 All Exchange Rates API

7.4.1 URL

/api/allrates

7.4.2 Parameters

- **from**: Symbol of the base currency. Examples: EUR, USD, TRY

when ’from’ parameter is not given, it is set to ’TRY’ as default

7.4.3 Return Format

```
{
  "from": FROM-PARAMETER,
  result: {
    result: RATES
  }
}
```

Example

```
{
  "from":"USD",
  result: {
    "BGN":0.2841204585,"NZD":0.2473233871,"ILS":0.5811409562,"
       RUB":10.6555050336, ... , "SEK":1.570492613,"EUR":0.1452
       70712
  }
}
```

7.4.4 Request Type

Only GET allowed.

7.4.5 Examples

Request

```
GET example.com/api/exchangerate?from=USD
```

Response

```
{
  "from": "USD",
  rates: {
    "BGN":1.7415850401,"NZD":1.5160284951,"ILS":3.5622439893,
       ... , "SEK":9.6267141585,"EUR":0.8904719501
  }
}
```

Request

```
GET example.com/api/exchangerate?
```

Response

```
{
  "from": "TRY",
  rates: {
    "BGN":0.2841204585,"NZD":0.2473233871,"ILS":0.5811409562,
       ... ,"SEK":1.570492613,"EUR":0.145270712
  }
}
```

7.5 Stock Indexes API

7.5.1 URL

/stock

7.5.2 Parameters

- **function**: The time series of choice. The following values are supported:
  - TIME_SERIES_INTRADAY
  - TIME_SERIES_DAILY
  - TIME_SERIES_DAILY_ADJUSTED
  - TIME_SERIES_WEEKLY
  - TIME_SERIES_WEEKLY_ADJUSTED
  - TIME_SERIES_MONTHLY
  - TIME_SERIES_MONTHLY_ADJUSTED
- **symbol**: The name of the stock index. Examples: MSFT
- **interval**: Time interval between two consecutive data points in the time series.The following values are supported: 1min,5min,15min,30min,60min

function and symbol parameters are required. The interval parameter is required only when function=TIME_SERIES_INTRADAY

7.5.3 Return Format

```
{
    "Meta Data": {
        "1. Information": GENERAL INFORMATION ABOUT RESPONSE
           SUCH AS FIELDS ETC.,
        "2. Symbol": SYMBOL PARAMETER,
        "3. Last Refreshed": LAST REFRESH TIME OF DATA,
        "4. Interval": INTERVAL PARAMETER IF GIVEN,
        "5. Output Size": Compact IS DEFAULT OUTPUT SIZE.
           Compact RETURNS ONLY THE LATEST 100 DATA POINTS IN
           THE INTRADAY TIME SERIES,
        "6. Time Zone": "US/Eastern" IS TIME ZONE OF LAST
           REFRESHED FIELD
    },
    TIME SERIES: {
        TIME: {
            "1. open": OPEN VALUE,
            "2. high": HIGH VALUE,
            "3. low": LOW VALUE,
            "4. close": CLOSE VALUE,
            "5. volume": VOLUME VALUE
        },
}
```

7.5.4 Request Type

Only GET allowed.

7.5.5 Examples

1. Request

   ```
   GET example.com/stock?function=TIME_SERIES_INTRADAY&symbol=
      MSFT&interval=5min
   ```

   Response

   ```
   {
       "Meta Data": {
           "1. Information": "Intraday (5min) open, high, low,
              close prices and volume",
           "2. Symbol": "MSFT",
           "3. Last Refreshed": "2019-05-03 16:00:00",
           "4. Interval": "5min",
           "5. Output Size": "Compact",
           "6. Time Zone": "US/Eastern"
       },
   ```

```
    "Time Series (5min)": {
        "2019-05-03 16:00:00": {
            "1. open": "128.9200",
            "2. high": "129.0400",
            "3. low": "128.8000",
            "4. close": "128.8900",
            "5. volume": "1163055"
        },
        "2019-05-03 15:55:00": {
            "1. open": "128.9300",
            "2. high": "128.9700",
            "3. low": "128.8600",
            "4. close": "128.9150",
            "5. volume": "373828"
        },
        ...
}
```

2. Request

```
GET example.com/stock?function=TIME_SERIES_DAILY&symbol=
   MSFT
```

Response

```
{
    "Meta Data": {
        "1. Information": "Daily Prices (open, high, low,
           close) and Volumes",
        "2. Symbol": "MSFT",
        "3. Last Refreshed": "2019-05-03",
        "4. Output Size": "Compact",
        "5. Time Zone": "US/Eastern"
    },
    "Time Series (Daily)": {
        "2019-05-03": {
            "1. open": "127.3600",
            "2. high": "129.4300",
            "3. low": "127.2500",
            "4. close": "128.9000",
            "5. volume": "24911126"
        },
        "2019-05-02": {
            "1. open": "127.9800",
            "2. high": "128.0000",
            "3. low": "125.5200",
            "4. close": "126.2100",
            "5. volume": "27350161"
        },
        ...
}
```

7.6 Register/Sign Up API

7.6.1 URL

/register

7.6.2 Request Format

```
body:
{
  "name": "NAME_OF_USER",
  "email": "EMAIL_OF_USER (unique)",
  "username": "USERNAME_OF_USER (unique)",
  "password": "PASSWORD_OF_USER"
}
```

7.6.3 Return Format

```
{
  "name": "NAME_OF_USER",
  "email": "EMAIL_OF_USER",
  "username": "USERNAME_OF_USER",
  "createdAt": _creationTime_
}
```

7.6.4 Request Type

Only POST allowed.

7.6.5 Examples

1. Request

   [POST] example.com/register

2. Body

```
{
  "name": "Name Surname",
  "email": "name@mail.com",
  "username": "crazy_boi",
  "password": "passw0rd"
}
```

3. Response

```
{
  "name": "Name Surname",
  "email": "name@mail.com",
  "username": "crazy_boi",
  "createdAt": 2019-05-12T18:20:16.633Z
}
```