# CMPE 352:

## FUNDAMENTALS OF SOFTWARE ENGINEERING

---

# MILESTONE #2

---

Spring 2019

Traders Platform

# Group 7

12.05.2019

# 1.Summary

## 1.1 Introduction

The overall design process of the project is finished after the creation of the mock-ups, user scenarios, use cases, class and sequence diagrams as we have reported in the first milestone. After that we progress to the second phase of our project according to given two assignments and feedbacks. In this period, we also have decided the name and logo of our project by contribution of our team members. So, we agreed on "EyeTrade" as the name and half bitten eye as our logo. The second phase of the project mainly includes design of the project plan which plays a vital role in keeping track of time and creation of a simple yet elegant API that includes many functionalities which are requested by the customer.

In this documentation, one can find detailed explanations of each step and how the implementation phase of the project is managed by our team.

## 1.2 Work Done

After the first milestone, given assignments started to contain more implementation and planning tasks, because eventually we have finished the design part of our project.

First assignment after the first milestone was making our project plan, which is kind of a road map that shows what we have done so far and what we are going to do in a detailed and organized way.

However, the most important thing we have done was an API implementation. We have spent almost 3 weeks for the task and come up with a nice API which has also a front-end. First of all, we decided which functionalities that our API offer in our first API meeting. We have also arranged an online meeting in order to share that functionalities among team members. In short, our plan was not only to implement an API, but also to work together in harmony as a team. Therefore one of our goals was to write integrable code. Our API offers some trade-related functionalities like currency exchange calculation and economic events calendar.

## 1.3 Read Ahead

Up to now, we have accomplished lots of things. However, these things are only for having a background set before starting to implement our product. So our first aim for the future is to develop a product which can be represented to end-user.

For the summer, we are planning to enlarge our vision and widen our knowledge to produce a better product. Some of our members will be working at tech companies as software interns. For sure, this will be great for our team to gain experience and see how the things work in real life. Also we will be conducting research about mobile development which is an area that we have not interacted up to now. We will be searching for example projects on github, we will evaluate them with good and bad sides. As a result, we will extract the futures of a good projects and we will try to use them on our production stages.

For the next semester, in general, we are planning to develop our product. We will be coding, reviewing and testing for nearly 4 months. We will be more careful about variety of our test cases, procedure of pull requests and our deadlines. We were able

to detect our mistakes while we are developing the practice-app. So we will try to fix our mistakes and become a better team overall.

## 1.4 Challenges

At the start of the api project we had to choose technologies to use in our api. However, it was not easy for us because we did not know the advantages and disadvantages of the frameworks which we had to choose. After a rigorous research on the choices, we choose Django because of it's object relational mapping system, ease of migration, elegant routing system, and powerful serialization/deserialization libraries.

After we have chosen the framework to use, we had to find the functionalities to implement. In order to deciding the functionalities we made a brainstorming considering the restrictions such as the functionalities must be related to the Traders Project, some functionalities must use external apis, some must do crud operations on the database, and they should use different http methods(POST, GET, PUT,DELETE, etc) which are proper for the requirements. Also, we needed to share those functionalities as fair as possible so we decided to who will implement which functionality in our second meeting about the api project.

When we started to the implementation part, most of us did not know very well about the RESTful Apis, MVT architecture, and some important libraries of Django so we taught those important knowledge to each other. Furthermore, the bugs related to routing system, migrations, and response formats were frequently occurred so we helped each other to fix those bugs. Finally, we had a working api that is ready to use.

The api part of the project was not sufficient because we had to implement also a user interface. However, since learning to design a frontend application is long commitment we have chosen a subgroup consisting of two people who have experience on frontend development. Our subgroup, started to implement the frontend application using Reactjs. They encountered some challenges because of the asynchronous nature of the javascript. For example, they encountered some binding errors because bindings had to wait for the response from our api but they overcome that situation using the built-in lifecycle methods in Reactjs. At the end, we had both backend and frontend application which are working properly together.

## 2. List Status of Deliverables

| Name | Delivery Date | Delivery Status |
| --- | --- | --- |
| Project Plan | 8.04.2019 | Delivered |
| eyeTrade API Development | 6.05.2019 | Delivered |
| eyeTrade Front-End Design | 6.05.2019 | Delivered |

## 3. Evaluation of Status of Deliverables

- **Project Plan:** We prepared a project plan of what we had done prior to milestone 1, and what we would be doing until milestone 2. In a real world situation, the project plan would have been prepared from the start, but our late introduction to the concept required a design like this.

  A project plan is crucial to any project because of the resources involved. Aside from money and time, human resources play a big role in software projects. We cannot move to the next stage unless we finish the ones required for it. Therefore, we need a project plan in order to make sure the project progresses smoothly.

- **eyeTrade API Development:** We used Django to build a RESTful API. In addition to developing an API of our own, we had to use another API in our project. We exposed certain functionalities so that they could be used from the front-end. However, we also needed live data for exchange rates. Therefore, we used an external API and integrated it into our system. We had limited access to the external API, therefore we saved the results in our database to read them later.

  Had we created our project without an API, it would be impossible to communicate with the front-end, and thus crippling our project greatly. The whole project would have to be contained on a single machine, which defeats the purpose of creating a server with dynamic content. APIs are necessary for using functionalities from different machines, or different programs on a single machine.

- **eyeTrade Front-End Design:** Our front-end is an application that communicates with our API and displays the requested information. It acts as an interface between the user and the server, tying the two ends together. Inputs from the user are turned into API requests and the results are displayed as HTML elements.

  A front-end must be made available to the end user because of the impracticalities involving using an API directly, such as having to install additional programs other than a standard web browser, the data not being easily understandable and visually unappealing, and having to know the API endpoints prior to browsing. The front-end deals with these problems and presents us an interface for the users.
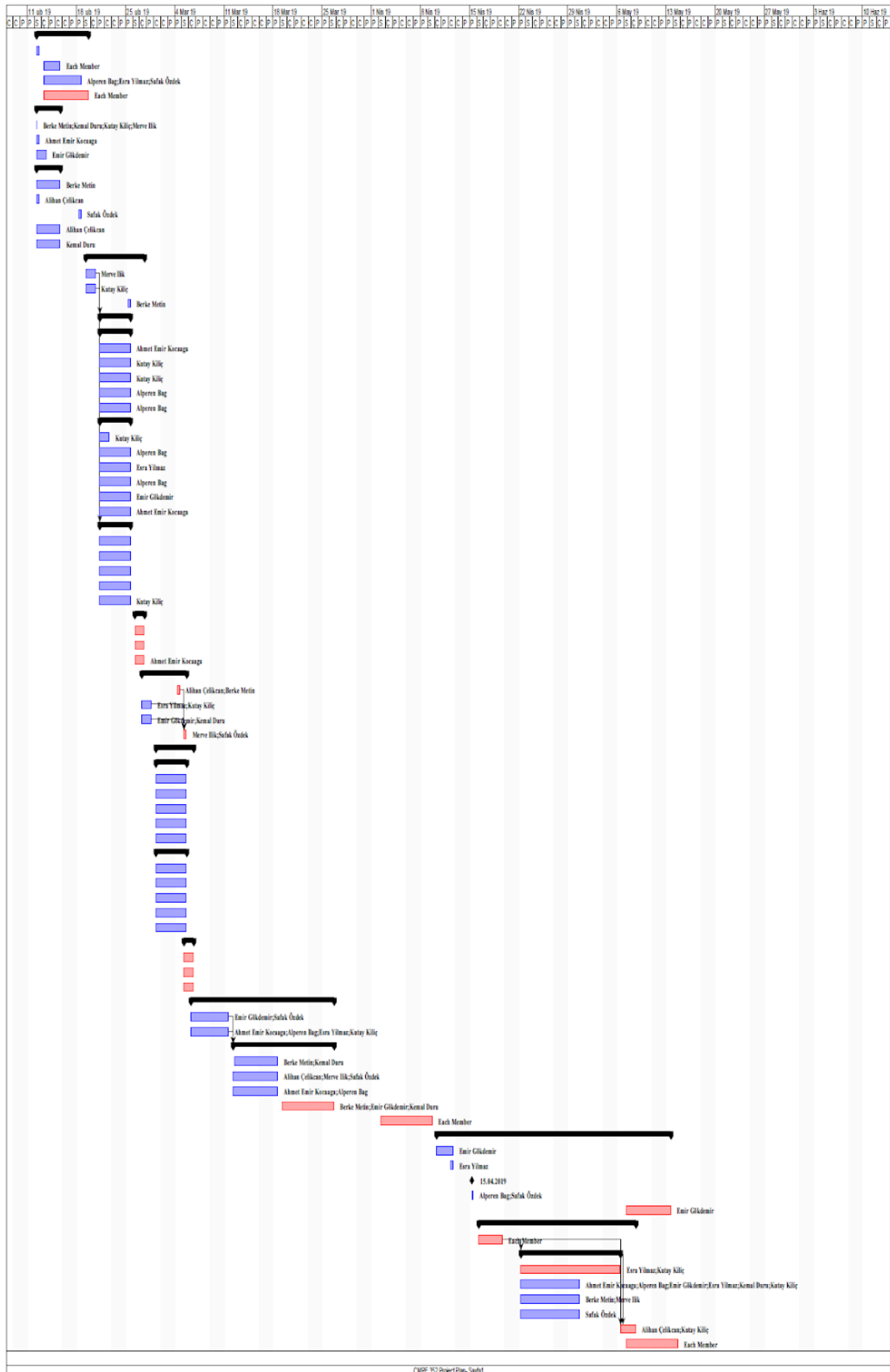
## 4. Work Done by Each Member

| Alperen Bağ | <ul><li>I wrote the API implementation section of our project plan with Şafak, and put a link for the project plan in our Wiki.</li><li>I implemented the currency related part of the API with Ahmet Emir.</li><li>I reviewed Şafak's code which is about Email subscribers list.</li></ul> |
|---|---|

| | |
|---|---|
| **Alican Şafak Özdek** | • **I wrote the API implementation section of our project plan with Alperen.**<br>• **I have created an Mailchimp account to manage mailing lists.**<br>• **I implemented the backend/frontend of Subscribe page which uses Mailchimp's API.**<br>• **I have created an EC2 device on AWS and deployed the system on this device.**<br>• **I have created a MySQL Server using AWS RDS and connected it's ports to EC2.**<br>• **I have reviewed some of the backend codes, gave feedback under related issues.**<br>• **I wrote "Road Ahead" and "Tools" section of milestone report with Ahmet Emir.** |
| **Ahmet Emir Kocaağa** | • **I wrote the Tools section of our milestone report with Şafak.**<br>• **I implemented the currency related part of the API with Alperen.**<br>• **I have created an Alpha Vantage key for the real time exchange rates.**<br>• **I have made the review of our project plan with Kutay and gave feedbacks about it under related issues.**<br>• **I reviewed Merve's code which was about Event tests.** |
| **Merve İlik** | • **Detection of duration and resources of Persona, User Scenarios and Sequence Diagrams for project plan.**<br>• **Adding detected parts to the project plan.**<br>• **Reviewing project plan according to feedback.**<br>• **Implemented Event app in project API with Berke.**<br>• **Added a unit test for Event app.** |
| **M.Emir Gökdemir** | • **I designed first half part of predecessor of project report.**<br>• **I reviewed the and redesigned of second part after other team members' review process.**<br>• **I presented our platform in class with Kutay and Berke.** |
| **Alihan Çelikcan** | • **I coded the frontend with Kutay.**<br>• **I wrote the Evaluation of Status of Deliverables section of the milestone report.** |

| | |
|---|---|
| **Seyfi Kutay Kılıç** | • **Developed Article App in the api project which handles createArticle(), listArticles(), getArticle(), deleteArticle(), and updateArticle() functionalities. Also, created a function to convert sql query results to python dictionary objects.**<br>• **Teached my teammates about Django and MVT architecture on a meeting.**<br>• **Helped for the implementation of eventcrawler class in Event app.**<br>• **Reviewed and gave feedback for Event app in our api.**<br>• **Researched for a frontend template and cropped it to use in the project.**<br>• **Set the CORS settings of the api.**<br>• **Designed the architecture of the frontend application, considering the restrictions of the template that we use.**<br>• **Developed the frontend with Alihan.**<br>• **Implemented the Navigation, Routing, Home Page, Article Page, and Documentation Page parts of the frontend.**<br>• **Wrote some unit tests for the Article app.**<br>• **Wrote "Challenges We Encountered as a Group" part of this milestone document.** |
| **Kemal Duru** | • **Tried to learn fundamentals of Django and React frameworks for both currency related part of API development and future implementation tasks.**<br>• **Made a research on PEP8 convention which is used throughout API implementation and share the important parts which are followed during coding.**<br>• **Wrote "Introduction" part of this milestone document.** |
| **Nazım Berke Metin** | • **Implemented Events app with Merve.**<br>• **Created the table 'works done by each member'.** |
| **Esra Yılmaz** | • **Implementation of searching a keyword in title or content for Articles.**<br>• **Implementation of testcases for searching operations.**<br>• **Review of code belonging Kutay Kılıç.**<br>• **Doing the Project Plan with Merve İlik, Alperen Bağ, Emir Gökdemir.** |

# 5. Project Plan

| | | Ad | Süre? | Balat | Bitirme | Önceki | Kaynak Adlar |
|---|---|---|---|---|---|---|---|
| 1 | | **Documentation** | **6 günler** | **12.02.2019** | **19.02.2019** | | |
| 2 | | Repository Initialize | 1 gün | 12.02.2019 ... | 12.02.2019 ... | | |
| 3 | | Favourite Git Repos | 3 günler | 13.02.2019 ... | 15.02.2019 ... | | Each Member |
| 4 | | Research about Git version control system | 4 günler | 13.02.2019 ... | 18.02.2019 ... | | Alperen Bag;Esra Yilmaz;Safak Özdek |
| 5 | | Personal Wiki's | 5 günler | 13.02.2019 ... | 19.02.2019 ... | | Each Member |
| 6 | | **Guidelines** | **4 günler** | **12.02.2019** | **15.02.2019** | | |
| 7 | | Labels | 0,25 günler | 12.02.2019 ... | 12.02.2019 ... | | Berke Metin;Kemal Duru;Kutay Kiliç;Merve Ilik |
| 8 | | Issues | 1 gün | 12.02.2019 ... | 12.02.2019 ... | | Ahmet Emir Kocaaga |
| 9 | | Readme | 2 günler | 12.02.2019 ... | 13.02.2019 ... | | Emir Gökdemir |
| 10 | | **Communication Plan** | **4 günler** | **12.02.2019** | **15.02.2019** | | |
| 11 | | Communication Page | 4 günler | 12.02.2019 ... | 15.02.2019 ... | | Berke Metin |
| 12 | | Slack Page | 1 gün | 12.02.2019 ... | 12.02.2019 ... | | Alihan Çelikcan |
| 13 | | Wiki Custom Sidebar | 1 gün | 18.02.2019 ... | 18.02.2019 ... | | Safak Özdek |
| 14 | | Research About Effort System | 4 günler | 12.02.2019 ... | 15.02.2019 ... | | Alihan Çelikcan |
| 15 | | Research About W3C | 4 günler | 12.02.2019 ... | 15.02.2019 ... | | Kemal Duru |
| 16 | | **Requirements** | **7 günler** | **19.02.2019** | **27.02.2019** | | |
| 17 | | Decision of Functional Requirements | 2 günler | 19.02.2019 ... | 20.02.2019 ... | | Merve Ilik |
| 18 | | Decision of Non-Functional Requirements | 2 günler | 19.02.2019 ... | 20.02.2019 ... | | Kutay Kiliç |
| 19 | | Glossary | 1 gün | 23.02.2019 ... | 25.02.2019 ... | | Berke Metin |
| 20 | | **Functional Requirements** | **3 günler** | **21.02.2019** | **25.02.2019** | **17** | |
| 21 | | **User Requirements** | **3 günler** | **21.02.2019** | **25.02.2019** | | |
| 22 | | User Types | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | Ahmet Emir Kocaaga |
| 23 | | Authentication | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | Kutay Kiliç |
| 24 | | Profile | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | Kutay Kiliç |
| 25 | | Portfolio | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | Alperen Bag |
| 26 | | Profit/Loss Section | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | Alperen Bag |
| 27 | | **System Requirements** | **3 günler** | **21.02.2019** | **25.02.2019** | | |
| 28 | | Notifications | 2 günler | 21.02.2019 ... | 22.02.2019 ... | | Kutay Kiliç |
| 29 | | Predictions | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | Alperen Bag |
| 30 | | Searching | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | Esra Yilmaz |
| 31 | | Trading Equipment | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | Alperen Bag |
| 32 | | Recommendation System | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | Emir Gökdemir |
| 33 | | Economic Events | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | Ahmet Emir Kocaaga |
| 34 | | **Non-Functional Requirements** | **3 günler** | **21.02.2019** | **25.02.2019** | **18** | **Merve Ilik** |
| 35 | | Security and Privacy | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | |
| 36 | | Performance | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | |
| 37 | | Availability | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | |
| 38 | | Annotations | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | |
| 39 | | Database | 3 günler | 21.02.2019 ... | 25.02.2019 ... | | Kutay Kiliç |
| 40 | | **Refinement of Requirements** | **2 günler** | **26.02.2019** | **27.02.2019** | | **Alperen Bag** |
| 41 | | Update wrt Feecback | 2 günler | 26.02.2019 ... | 27.02.2019 ... | | |
| 42 | | Update wrt Customer | 2 günler | 26.02.2019 ... | 27.02.2019 ... | | |
| 43 | | Preparation of Project Page | 2 günler | 26.02.2019 ... | 27.02.2019 ... | | Ahmet Emir Kocaaga |
| 44 | | **User Scenarios & Personas** | **5 günler?** | **27.02.2019** | **05.03.2019** | | |
| 45 | | Basic User Scenario and Persona | 1 gün | 04.03.2019 ... | 04.03.2019 ... | | Alihan Çelikcan;Berke Metin |
| 46 | | Trading User Scenario and Persona | 2 günler | 27.02.2019 ... | 28.02.2019 ... | | Esra Yilmaz;Kutay Kiliç |
| 47 | | Guest User Scenario and Persona | 2 günler? | 27.02.2019 ... | 28.02.2019 ... | | Emir Gökdemir;Kemal Duru |
| 48 | | Obtaining acceptance criteria | 1 gün | 05.03.2019 ... | 05.03.2019 ... | 46;45;47 | Merve Ilik;Safak Özdek |
| 49 | | **Mock-ups** | **4 günler?** | **01.03.2019** | **06.03.2019** | | |
| 50 | | **Web Mock-ups** | **3 günler** | **01.03.2019** | **05.03.2019** | | **Alihan Çelikcan;Berke Metin;Esra Yilmaz;Kutay Kiliç** |
| 51 | | Welcome Page, Sign-up and Login | 3 günler | 01.03.2019 ... | 05.03.2019 ... | | |
| 52 | | User Profile Page | 3 günler | 01.03.2019 ... | 05.03.2019 ... | | |
| 53 | | Home Page | 3 günler | 01.03.2019 ... | 05.03.2019 ... | | |
| 54 | | Currencies Page | 3 günler | 01.03.2019 ... | 05.03.2019 ... | | |
| 55 | | Comment Page | 3 günler | 01.03.2019 ... | 05.03.2019 ... | | |
| 56 | | **Mobile Mock-ups** | **3 günler?** | **01.03.2019** | **05.03.2019** | | **Emir Gökdemir;Kemal Duru** |
| 57 | | Welcome Page, Sign-up and Login | 3 günler | 01.03.2019 ... | 05.03.2019 ... | | |
| 58 | | User Profile | 3 günler | 01.03.2019 ... | 05.03.2019 ... | | |
| 59 | | Home Page | 3 günler | 01.03.2019 ... | 05.03.2019 ... | | |
| 60 | | Event Page | 3 günler | 01.03.2019 ... | 05.03.2019 ... | | |
| 61 | | Article Page | 3 günler | 01.03.2019 ... | 05.03.2019 ... | | |
| 62 | | **Preconditions** | **2 günler?** | **05.03.2019** | **06.03.2019** | | **Emir Gökdemir** |
| 63 | | Preconditions of Guest | 2 günler | 05.03.2019 ... | 06.03.2019 ... | | |
| 64 | | Preconditions of Basic User | 2 günler | 05.03.2019 ... | 06.03.2019 ... | | |
| 65 | | Preconditions of Guest | 2 günler | 05.03.2019 ... | 06.03.2019 ... | | |
| 66 | | **Diagrams** | **15 günler** | **06.03.2019** | **26.03.2019** | | |
| 67 | | Use Case Diagram | 4 günler | 06.03.2019 ... | 11.03.2019 ... | | Emir Gökdemir;Safak Özdek |
| 68 | | Class Diagram | 4 günler | 06.03.2019 ... | 11.03.2019 ... | | Ahmet Emir Kocaaga;Alperen Bag;Esra Yilmaz;Kutay Kiliç |
| 69 | | **Sequence Diagrams** | **11 günler?** | **12.03.2019** | **26.03.2019** | **67;68** | |
| 70 | | Preparation of 6 sequence diagrams | 4,5 günler | 12.03.2019 ... | 18.03.2019 ... | | Berke Metin;Kemal Duru |
| 71 | | Preparation of 4 sequence diagrams | 5 günler | 12.03.2019 ... | 18.03.2019 ... | | Alihan Çelikcan;Merve Ilik;Safak Özdek |
| 72 | | Preparation of 2 sequence diagrams | 5 günler | 12.03.2019 ... | 18.03.2019 ... | | Ahmet Emir Kocaaga;Alperen Bag |
| 73 | | Update Diagrams for Feedback | 6 günler | 19.03.2019 ... | 26.03.2019 ... | | Berke Metin;Emir Gökdemir;Kemal Duru |
| 74 | | Milestone Report 1 | 6 günler | 02.04.2019 ... | 09.04.2019 ... | | Each Member |
| 75 | | **Preparing Project Planning** | **24 günler?** | **10.04.2019** | **13.05.2019** | | |
| 76 | | Beginning to Mockups | 3 günler? | 10.04.2019 ... | 12.04.2019 ... | | Emir Gökdemir |
| 77 | | Mockups to Diagrams | 1 gün? | 12.04.2019 ... | 12.04.2019 ... | | Esra Yilmaz |
| 78 | | Diagrams to API | 0 günler? | 13.04.2019 ... | 15.04.2019 ... | | Merve Ilik |
| 79 | | API Implemantation | 0,5 günler? | 15.04.2019 ... | 15.04.2019 ... | | Alperen Bag;Safak Özdek |
| 80 | | Repreparing Project Planning | 5 günler? | 07.05.2019 ... | 13.05.2019 ... | | Emir Gökdemir |
| 81 | | **API Implementation** | **17 günler?** | **16.04.2019** | **08.05.2019** | | |
| 82 | | Learning Tools | 4 günler? | 16.04.2019 ... | 19.04.2019 ... | | Each Member |
| 83 | | **Backend Implementation** | **10,5 günl...** | **22.04.2019** | **06.05.2019** | **82** | |
| 84 | | createArticle(), deleteArticle(), getArtic... | 10,5 günler? | 22.04.2019 ... | 06.05.2019 ... | | Esra Yilmaz;Kutay Kiliç |
| 85 | | getCurrency(), convertCurrency() | 7 günler? | 22.04.2019 ... | 30.04.2019 ... | | Ahmet Emir Kocaaga;Alperen Bag;Emir Gökdemir;Esra Yilmaz;Kemal Duru;Kutay Kiliç |
| 86 | | getEvents() | 7 günler? | 22.04.2019 ... | 30.04.2019 ... | | Berke Metin;Merve Ilik |
| 87 | | addToMailList() | 7 günler? | 22.04.2019 ... | 30.04.2019 ... | | Safak Özdek |
| 88 | | Frontend Implementation | 2,5 günler? | 06.05.2019 ... | 08.05.2019 ... | 82;83 | Alihan Çelikcan;Kutay Kiliç |
| 89 | | Milestone Report 2 | 6 günler? | 07.05.2019 ... | 14.05.2019 ... | | Each Member |

Each Member

Alperen Bag;Esra Yilmaz;Safak Özdek

Each Member

Berke Metin;Kemal Duru;Kutay Kiliç;Merve Ilik

Ahmet Emir Kocaaga

Emir Gökdemir

Berke Metin

Alihan Çelikcan

Safak Özdek

Alihan Çelikcan

Kemal Duru

Merve Ilik

Kutay Kiliç

Berke Metin

Ahmet Emir Kocaaga

Kutay Kiliç

Kutay Kiliç

Alperen Bag

Alperen Bag

Kutay Kiliç

Alperen Bag

Esra Yilmaz

Alperen Bag

Emir Gökdemir

Ahmet Emir Kocaaga

Kutay Kiliç

Ahmet Emir Kocaaga

Alihan Çelikcan;Berke Metin

Esra Yilmaz;Kutay Kiliç

Emir Gökdemir;Kemal Duru

Merve Ilik;Safak Özdek

Emir Gökdemir;Safak Özdek

Ahmet Emir Kocaaga;Alperen Bag;Esra Yilmaz;Kutay Kiliç

Berke Metin;Kemal Duru

Alihan Çelikcan;Merve Ilik;Safak Özdek

Ahmet Emir Kocaaga;Alperen Bag

Berke Metin;Emir Gökdemir;Kemal Duru

Each Member

Emir Gökdemir

Esra Yilmaz

15.04.2019

Alperen Bag;Safak Özdek

Emir Gökdemir

Each Member

Esra Yilmaz;Kutay Kiliç

Ahmet Emir Kocaaga;Alperen Bag;Emir Gökdemir;Esra Yilmaz;Kemal Duru;Kutay Kiliç

Berke Metin;Merve Ilik

Safak Özdek

Alihan Çelikcan;Kutay Kiliç

Each Member

# 6. Tools

It is a fact that we are new-learners, not professionals, as a group. Because of that, it was a good choice to use Django REST Framework. While Django took care of the much of the hassle of development, we had the chance to focus only to our API's functionalities. So, we did not need to reinvent the wheel as it is stated in Django's official website.

Alpha Vantage is an easy to use and well documented API, but we had also problems with it. Because it is a free API, it provides only 5 distinct data in a minute. So, we needed to reduce the number of currencies we will use in our API. So, probably we will not be using it in for our real project.

Mailchimp is a great tool to manage your mailing lists and send emails. It offers many different templates for your emails and it is very easy to use. The reason we decided to use mailchimp was being able to manage mailing lists in a better way because email marketing is very popular nowadays, so we wanted use a tool for this job and mailchimp is one of the most popular email tools out there.

As for the frontend, we decided to code it in React (react.js) for its ease of use and beginner-friendliness. It allows for dynamic use of HTML elements via JSX. We also used CoreUI along with React to create an appealing UI. The main deciding factor for us was the ease of creating graphs and dashboards which are both essential considering the scope of our project. It also integrates easily into React.

**Django:** Django is a free and open source, high-level Python web framework that eases the creation of complex database-driven websites.

**Django REST Framework:** Django Rest Framework is a flexible toolkit for building Web APIs which is a feature-filled extension to the Django Framework.

**Alpha Vantage:** It is a free API that provides real time and historical data on stocks, forex and crypto currencies. For more information: https://www.alphavantage.co/#page-top

**Mailchimp.com and Mailchimp's API:** Mailchimp is a platform that you can manage your email lists, send out emails with templates, keep useful information related to your subscribers. It provides an API to add a new user to your email list, delete someone from email list and update information about subscribers. For more information about mailchimp's API: https://developer.mailchimp.com/

**React:** React (also known as react.js) is a free JavaScript library that helps you build websites in a fast and reliable manner.

**CoreUI:** CoreUI is a free Bootstrap template that helps create better user interfaces. It is mainly used for creating admin pages.

# 7. API URL Documentations:

### 7.1 Currency

#### 7.1.1. Retrieve All Currency Exchange Rates
*http://52.87.206.237:8000/currencies/*

It will return all currency exchange rate information in our server. Currently, we offer only TRY/USD and USD/TRY.

### 7.1.2. Retrieve Only One Currency Exchange Rate
*http://52.87.206.237:8000/currencies/<id>*
It will return the currency exchange rate according to the given id. ID of USD/TRY is 0, ID of TRY/USD is 1.

### 7.1.3. Conversion Between Currencies
*http://52.87.206.237:8000/currencies/convert/<from>/<to>/<amount>*
It will convert to given amount of money (<amount>) from one currency (<from>) to other currency (<to>). For example, in order to convert 10 TRY to USD;
*http://52.87.206.237:8000/currencies/convert/TRY/USD/10*

## 7.2 Article

### 7.2.1. Retrieve All Articles
GET request to http://52.87.206.237:8000/articles/
It will return all the articles in our database.

### 7.2.2. Retrieve a Spesific Article
GET request to http://52.87.206.237:8000/articles/id
It will return the article which has the given id.

### 7.2.3. Create an Article
POST request to http://52.87.206.237:8000/articles/ with an article object in the body
It will create an article which has the given title and content.

### 7.2.4. Update an Article
PUT request to http://52.87.206.237:8000/articles/id with an article object in the body
It will update the article object having the given id, according to the given information on the body.

### 7.2.5. Delete an Article
DELETE request to http://52.87.206.237:8000/articles/id
It will delete the article which has the given id.

### 7.2.6. Retrieve the Articles which Contain the Given Keyword in Title
GET request to
http://52.87.206.237:8000/articles/getbykeywordsintitle/keyword
It will retrieve the articles which contain the given keyword in it's title.

### 7.2.7. Retrieve the Articles which Contain the Given Keyword in Content
GET request to
http://52.87.206.237:8000/articles/getbykeywordsincontent/keyword
It will retrieve the articles which contain the given keyword in it's content.