CmpE 352
Spring 2019
Milestone 2

Group 4
Bahadır Hocamoğlu
Baturalp Yörük
Elif Çalışkan
Ege Başural
Emre Demircioğlu
Gürkan Demir
İbrahim Özgürcan Öztaş
İsmail Levent Baş
Muhammed Bera Kaya
Taha Eyup Korkmaz

May 12, 2019

Contents

1 Executive Summary

1.1 Project Introduction

Our project Arken's aim is to establish an environment that users can socialize with each other, get information about any trading equipment, read articles and find out about economic events and also share their ideas about economic events and articles. There are three types of users in this project: Guest user-not registered-, Basic user and Trader.

Guest users can get information about any trading equipment, event or article. They can search users, trading equipment, events or articles. Basic users can make comments, chase events, create portfolios, rate articles, manually enter investments, make predictions about any trading equipment and follow users in addition to guest user's abilities. Their prediction rate for any equipment can be seen by other users. A user's profile can be private. Traders can make investments onsite with the IBAN number that they provide. Every user has profit/loss section and it shows the user's profit based on user's transactions with respect to the chosen trading equipment. Users can make annotations about events, trading equipment, articles etc...

Events have significance levels which represents their importance. Also events can have numeric values about the related trading equipment. Events can be filtered based on their significance level or country that it belongs. Articles can be rated, commented and shared by users.

Trading equipment should have functionalities such as: the previous close, percentage change with the percentage close, amount change with the previous close, day's range and moving averages.

The system provides semantic search and makes recommendations about articles or trading equipment based on their user history.

1.2 Work Done So Far

After the first milestone, we have done the project plan assignment with ProjectLibre.

For the implementation assignment, we divided into 3 groups. First group's mission was handling the signup and login part. The second group's mission was handling the economic events part. The last group's mission was handling the trading equipment part.

- We have written the APIs that get necessary data from related websites.
- We have completed the frontend part using React.
- We have tested the APIs with Mocha test library.

Our team has completed every assignment that we were given:

- Project Plan
- Implementation

1.3 Planned Changes

After the first Milestone report, we designed our logo and we created our project plan. Thanks to this API assignment we gained insight for the future tasks. We plan to introduce more features to our project such as articles and comments. These are our preferential plans for the road ahead.

In CMPE451, we will also consider different types of users, prediction rates, search and recommendation functionality. At the end of this semester we will divide the tasks of back-end, front-end and mobile development in order to gain a general idea during summer vacation.

1.4 Challenges We Met

During the implementation of our practice-app, we have met many challenges which are generally sourced from lack of experience in web development. Most of our group members have not written an API prior to the assignment and also didn't know how to write it. Thus, we first needed to learn many things. Moreover, most of us didn't know how to write the frontend and test part. Since we don't know much about those things, our decision on the languages we were going to use took quite long.

Therefore, after deciding to use **NodeJS** for backend, **React** for frontend, **MongoDB** for database, we took some time to read their documentation, watch their implementation videos on YouTube, Udemy etc. , read articles about general knowledge for web development.

2 List and status of deliverables

Deliverable	**Deadline**	**Status**
Project Plan	08.04.2019	Done
Implementation	06.05.2019	Done
API Documentation	06.05.2019	Done

Table 1: List and status of deliverables

3 Evaluation of the status of deliverables and its impact on project

Project Plan: Project plan helps us evaluate our progress on our project. We created a Gantt Chart to show where we currently are and where we should be in upcoming weeks. Essential elements like milestones and task are included in it. We have been updating it on every progress we make.

Implementation: Implementation part is the backbone of the process. We have group members that have no idea about frameworks and APIs. Therefore, we collected every language which we can use and elaborated each of them according to our group dynamics. Then, we decided to use JavaScript and its desired frameworks, such as NodeJS for back end, and React for front end. Furthermore, we split the tasks to certain people to utilize our efficiency. After every task is done, we regrouped to begin merging all different branches we opened for each task, and by each branch merged with main development branch, we encountered problems that based on synchronization. Several issues needs minor changing, that we've handled them without an effort. However, several issues needs more than minor changes, thus we've reconfigured our work to be compatible with each other. Finally, we've constructed the root our our project.

API Documentation: The API documentation is the map of our project. The purpose behind the API Documentation is to aid other developers and end-users to fetch information and transform it on their own goal. The API Documentation is explained elaborately in Section 7 and Section 8.

4 Summary of work done

Member Name	Work Done
Bahadır Hocamoğlu	Created initial files for web development and API parts of the practice API project. Booted up MongoDB database to use in the project with Gürkan Demir. Implemented login and signup functionalities with Gürkan Demir. Created the first versions of signup and login pages with Gürkan Demir. Wrote several unit tests for events controller functions. Done several reviews for my teammates' pull requests. Went through the final version of the whole project and solved several conflicts between branches.
Baturalp Yörük	I implemented the unit test of the trading equipment part. I opened a pull request to merge the branch that I created. I reviewed some of the pull requests and merged some of them. I contributed to Executive Summary part and list and status of deliverables part of this Milestone.
Elif Çalışkan	I assisted in preparing Project Plan draft. I implemented front-end part of event list. Contributed to unit test of event controller function. I opened pull requests in order to merge my code. I reviewed the code of my teammates, received reviews and updated my code according to them. I contributed to Executive Summary part of Milestone2.
Ege Başural	Created base structure of React frontend, prepared route and history structure. Created header and login-signup components together with İbrahim Özgürcan Öztaş.

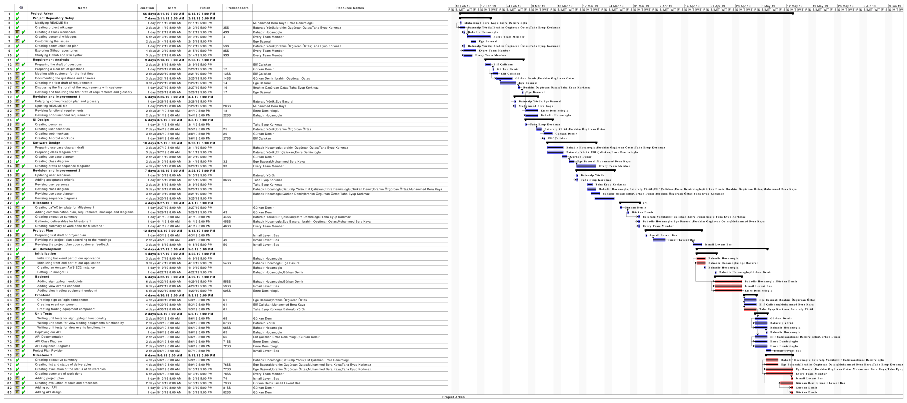
Emre Demircioğlu	I got authentication for external APIs that provides forex data. I implemented trading equipment endpoint which contains two get functionalities for our practice-app. I made class diagram and 3 sequence diagrams(Signup, view events and using currency converter) for our practice-app.I created pull request to merge my implementation and updated it according my team members reviews.I reviewed the code segments that my team members asked me to review and made necessary warnings and advise.
Gürkan Demir	I contributed to prepare Project Plan with other team members. I implemented signup endpoint for our practice-app. I implemented login endpoint with Bahadir for our practise-app. I booted up MongoDB with our server with Bahadir. I implemented frontend side of signup, trading equipment and currency converter for our practice-app. I fixed some bugs in login and events endpoint. I wrote unit tests for signup endpoint. I created lots of pull requests in order to merge what I had done. I reviewed almost everyone's code and made necessary comments. I contributed to prepare Project Plan, Evaluation of Tools and Processes, API parts of this milestone with Levent.
İbrahim Özgürcan Öztaş	I implemented the frontend parts of login and sign up with Ege Başural. Also contributed the API parts of login and sign up even if it is very little contribution. Furthermore, I've reviewed my friends' works and opened a pull request to merge our developed parts from our branch to main development branch, then from here to main streamline of our project. I've contributed in composing the Project Plan with my colleagues. Although I do not know much about these concepts, and their applications on each part, I've given my best to understand, elaborate and apply all frameworks that we use in our projects.

Table 2: Summary of work done by each team member

İsmail Levent Baş	I created all of the project plan versions (v1, v2, v3 and v4) starting from the first plan using ProjectLibre. I added all of the necessary tasks, milestones, resources, start and finish time of all of the tasks with the help of meeting notes written by Elif Çalışkan and all of the assigned issues. I designed all of the predecessors and successors of all tasks. I added FS, SS, FF, SF links between tasks where deemed necessary. I created the view events functionality endpoint which uses a get request to show the ten latest economic events in the world. I created another branch to implement this functionality and then I created a pull request to merge this branch. I reviewed the written code by our team members. I added the newly created version of the project plan which is revised by me to the Milestone 2 report. I added two evaluations for the tools we have used during our implementation and project planning phase to the Milestone 2 report.
Muhammed Bera Kaya	I implemented the page where the properties of an event are demonstrated.I added links to these pages from the page where all events are listed.I reviewed some codes of my team members.I created a branch for the changes I made to project and created a pull request to merge this branch.I contributed to ”listing and evaluation of deliverables” sections of the milestone with Ege,Taha and Özgürcan.
Taha Eyup Korkmaz	I implemented the frontend parts of trading equipment page with Gürkan Demir. I’ve reviewed my friends’ works and opened a pull request to merge our developed parts from our branch to main development branch, then from here to main streamline of our project. I’ve contributed in composing the Deliverables of the Project with my Bera Kaya, Ege Başural and Özgürcan Öztaş. Although I do not know much about these concepts, and their applications on each part, I’ve given my best to understand, discuss, apply and learn all the tools and frameworks that we use in our project’s parts.

Table 3: Summary of work done by each team member

5 Project Plan



		Name	Duration	Start	Finish	Predecessors	Resource Names
1	✓	**Project Arken**	**66 days**	**2/11/19 8:00 AM**	**5/13/19 5:00 PM**		
2	✓	**Project Repository Setup**	**7 days**	**2/11/19 8:00 AM**	**2/19/19 5:00 PM**		
3	✓	Modifying README file	1 day	2/11/19 8:00 AM	2/11/19 5:00 PM		Muhammed Bera Kaya;Emre Demircioglu
4	✓	Creating project wikipage	2 days	2/11/19 8:00 AM	2/12/19 5:00 PM	3SS	Baturalp Yörük;Ibrahim Özgürcan Öztas;Taha Eyup Korkmaz
5	✓	Creating a Slack workspace	1 day	2/12/19 8:00 AM	2/12/19 5:00 PM	4SS	Bahadir Hocamoglu
6	✓	Creating personal wikipages	5 days	2/13/19 8:00 AM	2/19/19 5:00 PM	4	Every Team Member
7	✓	Customizing the issues	2 days	2/14/19 8:00 AM	2/15/19 5:00 PM		Ege Basural
8	✓	Creating communication plan	1 day	2/12/19 8:00 AM	2/12/19 5:00 PM	5SS	Baturalp Yörük;Ibrahim Özgürcan Öztas;Taha Eyup Korkmaz
9	✓	Exploring Github repositories	4 days	2/12/19 8:00 AM	2/15/19 5:00 PM	8SS	Every Team Member
10	✓	Studying Github and wiki syntax	3 days	2/12/19 8:00 AM	2/14/19 5:00 PM	9SS	Every Team Member
11	✓	**Requirement Analysis**	**9 days**	**2/18/19 8:00 AM**	**2/28/19 5:00 PM**		
12	✓	Preparing the draft of questions	2 days	2/18/19 8:00 AM	2/19/19 5:00 PM		Elif Çaliskan
13	✓	Preparing a clear list of questions	1 day	2/20/19 8:00 AM	2/20/19 5:00 PM	12	Gürkan Demir
14	✓	Meeting with customer for the first time	2 days	2/20/19 8:00 AM	2/21/19 5:00 PM	13SS	Elif Çaliskan
15	✓	Documenting the questions and answers	3 days	2/21/19 8:00 AM	2/25/19 5:00 PM	14SS	Gürkan Demir;Ibrahim Özgürcan Öztas
16	✓	Creating the first draft of requirements	3 days	2/22/19 8:00 AM	2/26/19 5:00 PM	14	Ege Basural
17	✓	Discussing the first draft of the requirements with customer	1 day	2/27/19 8:00 AM	2/27/19 5:00 PM	16	Ibrahim Özgürcan Öztas;Taha Eyup Korkmaz
18	✓	Revising and finalizing the first draft of requirements and glossary	1 day	2/28/19 8:00 AM	2/28/19 5:00 PM	17	Ege Basural
19	✓	**Revision and Improvement 1**	**5 days**	**2/26/19 8:00 AM**	**3/4/19 5:00 PM**		
20	✓	Enlarging communication plan and glossary	1 day	2/26/19 8:00 AM	2/26/19 5:00 PM		Baturalp Yörük;Ege Basural
21	✓	Updating README file	1 day	2/26/19 8:00 AM	2/26/19 5:00 PM	20SS	Muhammed Bera Kaya
22	✓	Revising functional requirements	2 days	3/1/19 8:00 AM	3/4/19 5:00 PM	18	Emre Demircioglu
23	✓	Revising non-functional requirements	2 days	3/1/19 8:00 AM	3/4/19 5:00 PM	22SS	Bahadir Hocamoglu
24	✓	**UI Design**	**6 days**	**3/1/19 8:00 AM**	**3/8/19 5:00 PM**		
25	✓	Creating personas	1 day	3/1/19 8:00 AM	3/1/19 5:00 PM		Taha Eyup Korkmaz
26	✓	Creating user scenarios	2 days	3/4/19 8:00 AM	3/5/19 5:00 PM	25	Baturalp Yörük;Ibrahim Özgürcan Öztas
27	✓	Creating web mockups	3 days	3/6/19 8:00 AM	3/8/19 5:00 PM	26	Gürkan Demir
28	✓	Creating Android mockups	1 day	3/6/19 8:00 AM	3/6/19 5:00 PM	27SS	Elif Çaliskan
29	✓	**Software Design**	**10 days**	**3/7/19 8:00 AM**	**3/20/19 5:00 PM**		
30	✓	Preparing use case diagram draft	3 days	3/7/19 8:00 AM	3/11/19 5:00 PM		Bahadir Hocamoglu;Ibrahim Özgürcan Öztas;Taha Eyup Korkmaz
31	✓	Preparing class diagram draft	3 days	3/7/19 8:00 AM	3/11/19 5:00 PM		Baturalp Yörük;Elif Çaliskan;Emre Demircioglu
32	✓	Creating use case diagram	2 days	3/11/19 8:00 AM	3/12/19 5:00 PM		Gürkan Demir
33	✓	Creating class diagram	2 days	3/13/19 8:00 AM	3/14/19 5:00 PM	32	Ege Basural;Muhammed Bera Kaya
34	✓	Creating drafts of sequence diagrams	4 days	3/15/19 8:00 AM	3/20/19 5:00 PM	33	Every Team Member
35	✓	**Revision and Improvement 2**	**7 days**	**3/15/19 8:00 AM**	**3/25/19 5:00 PM**		
36	✓	Updating user scenarios	1 day	3/15/19 8:00 AM	3/15/19 5:00 PM		Baturalp Yörük
37	✓	Adding acceptance criteria	1 day	3/15/19 8:00 AM	3/15/19 5:00 PM	36SS	Taha Eyup Korkmaz
38	✓	Revising user personas	2 days	3/18/19 8:00 AM	3/19/19 5:00 PM		Taha Eyup Korkmaz
39	✓	Revising class diagram	3 days	3/18/19 8:00 AM	3/20/19 5:00 PM		Bahadir Hocamoglu;Baturalp Yörük;Elif Çaliskan;Emre Demircioglu;Gürkan Demir;Ibrahim Özgürcan Öztas;Muhammed Bera Kaya
40	✓	Revising use case diagram	3 days	3/19/19 8:00 AM	3/21/19 5:00 PM		Bahadir Hocamoglu;Gürkan Demir;Ibrahim Özgürcan Öztas;Taha Eyup Korkmaz
41	✓	Revising sequence diagrams	4 days	3/20/19 8:00 AM	3/25/19 5:00 PM		
42	✓	**Milestone 1**	**4 days**	**3/27/19 8:00 AM**	**4/1/19 5:00 PM**		
43	✓	Creating LaTeX template for Milestone 1	1 day	3/27/19 8:00 AM	3/27/19 5:00 PM		Gürkan Demir
44	✓	Adding communication plan, requirements, mockups and diagrams	1 day	3/29/19 8:00 AM	3/29/19 5:00 PM	43	Gürkan Demir
45	✓	Creating executive summary	1 day	4/1/19 8:00 AM	4/1/19 5:00 PM	44SS	Baturalp Yörük;Elif Çaliskan;Emre Demircioglu;Taha Eyup Korkmaz
46	✓	Gathering deliverables for Milestone 1	1 day	4/1/19 8:00 AM	4/1/19 5:00 PM	45SS	Bahadir Hocamoglu;Ege Basural;Ibrahim Özgürcan Öztas;Muhammed Bera Kaya
47	✓	Creating summary of work done for Milestone 1	1 day	4/1/19 8:00 AM	4/1/19 5:00 PM	46SS	Every Team Member
48	✓	**Project Plan**	**12 days**	**4/3/19 8:00 AM**	**4/18/19 5:00 PM**		
49	✓	Preparing first draft of project plan	1 day	4/3/19 8:00 AM	4/3/19 5:00 PM		Ismail Levent Bas
50	✓	Revising the project plan according to the meetings	2 days	4/5/19 8:00 AM	4/8/19 5:00 PM	49	Ismail Levent Bas
51	✓	Revising the project plan upon customer feedback	3 days	4/16/19 8:00 AM	4/18/19 5:00 PM	50	Ismail Levent Bas
52	✓	**API Development**	**14 days**	**4/17/19 8:00 AM**	**5/6/19 5:00 PM**		
53	✓	**Initialization**	**4 days**	**4/17/19 8:00 AM**	**4/22/19 5:00 PM**		
54	✓	Initializing back-end part of our application	3 days	4/17/19 8:00 AM	4/19/19 5:00 PM		Bahadir Hocamoglu
55	✓	Initializing front-end part of our application	3 days	4/17/19 8:00 AM	4/19/19 5:00 PM	54SS	Bahadir Hocamoglu;Ege Basural
56	✓	Creating an Amazon AWS EC2 instance	1 day	4/19/19 8:00 AM	4/19/19 5:00 PM		Bahadir Hocamoglu
57	✓	Setting up mongoDB	1 day	4/22/19 8:00 AM	4/22/19 5:00 PM		Bahadir Hocamoglu;Gürkan Demir
58	✓	**Backend**	**6 days**	**4/22/19 8:00 AM**	**4/29/19 5:00 PM**		
59	✓	Adding sign up/login endpoints	6 days	4/22/19 8:00 AM	4/29/19 5:00 PM	55SS	Bahadir Hocamoglu;Gürkan Demir
60	✓	Adding view events endpoint	6 days	4/22/19 8:00 AM	4/29/19 5:00 PM	59SS	Ismail Levent Bas
61	✓	Adding view trading equipment endpoint	6 days	4/22/19 8:00 AM	4/29/19 5:00 PM	60SS	Emre Demircioglu
62	✓	**Frontend**	**4 days**	**4/30/19 8:00 AM**	**5/3/19 5:00 PM**		
63	✓	Creating sign up/login components	4 days	4/30/19 8:00 AM	5/3/19 5:00 PM	61	Ege Basural;Ibrahim Özgürcan Öztas
64	✓	Creating event component	4 days	4/30/19 8:00 AM	5/3/19 5:00 PM	61	Elif Çaliskan;Muhammed Bera Kaya
65	✓	Creating trading equipment component	4 days	4/30/19 8:00 AM	5/3/19 5:00 PM	61	Taha Eyup Korkmaz;Baturalp Yörük
66	✓	**Unit Tests**	**2 days**	**5/3/19 8:00 AM**	**5/6/19 5:00 PM**		
67	✓	Writing unit tests for sign up/login functionality	2 days	5/3/19 8:00 AM	5/6/19 5:00 PM	65	Gürkan Demir
68	✓	Writing unit tests for view trading equipments functionality	2 days	5/3/19 8:00 AM	5/6/19 5:00 PM	67SS	Baturalp Yörük
69	✓	Writing unit tests for view events functionality	2 days	5/3/19 8:00 AM	5/6/19 5:00 PM	68SS	Bahadir Hocamoglu
70	✓	Deploying our API	1 day	5/6/19 8:00 AM	5/6/19 5:00 PM	65	Bahadir Hocamoglu
71	✓	API Documentation	2 days	5/3/19 8:00 AM	5/6/19 5:00 PM	65	Elif Çaliskan;Emre Demircioglu;Gürkan Demir
72	✓	API Class Diagram	2 days	5/3/19 8:00 AM	5/6/19 5:00 PM	71SS	Emre Demircioglu
73	✓	API Sequence Diagrams	2 days	5/3/19 8:00 AM	5/6/19 5:00 PM	72SS	Emre Demircioglu
74	✓	Project Plan Revision	2 days	5/6/19 8:00 AM	5/7/19 5:00 PM		Ismail Levent Bas
75	✓	**Milestone 2**	**6 days**	**5/6/19 8:00 AM**	**5/13/19 5:00 PM**		
76	✓	Creating executive summary	4 days	5/6/19 8:00 AM	5/9/19 5:00 PM		Bahadir Hocamoglu;Baturalp Yörük;Elif Çaliskan;Emre Demircioglu
77	✓	Creating list and status of deliverables	4 days	5/6/19 8:00 AM	5/9/19 5:00 PM	76SS	Ege Basural;Ibrahim Özgürcan Öztas;Muhammed Bera Kaya;Taha Eyup Korkmaz
78	✓	Creating evaluation of the status of deliverables	6 days	5/6/19 8:00 AM	5/13/19 5:00 PM	77SS	Ege Basural;Ibrahim Özgürcan Öztas;Muhammed Bera Kaya;Taha Eyup Korkmaz
79	✓	Creating summary of work done	6 days	5/6/19 8:00 AM	5/13/19 5:00 PM	78SS	Every Team Member
80	✓	Adding project plan	1 day	5/13/19 8:00 AM	5/13/19 5:00 PM	74	Ismail Levent Bas
81	✓	Creating evaluation of tools and processes	2 days	5/10/19 8:00 AM	5/13/19 5:00 PM	79SS	Gürkan Demir;Ismail Levent Bas
82	✓	Adding our API	1 day	5/13/19 8:00 AM	5/13/19 5:00 PM	81SS	Gürkan Demir
83	✓	Adding API design	1 day	5/13/19 8:00 AM	5/13/19 5:00 PM	82SS	Gürkan Demir

Project Arken

For full view, please visit our page named Project plan under page Wiki.

6 Evaluation of Tools and Processes

- **Tools**
  - **Project Plan**
    - **ProjectLibre:** One of the most used open source project management tools out there where you can create tasks and assign resources to those tasks as well as viewing the project as a Gantt chart provided by ProjectLibre. It was suggested to us by our instructor, hence we have used this project management tool as we didn't need a much more complex and advanced planning tool.
  - **Development**
    - **Visual Studio Code:** Most of us used that simple yet very extensible code editor as IDE. The work we were doing didn't require a more sophisticated development environment, so we settled with that simple solution.
    - **Express:** Express is an unopinionated, minimalist, extensible framework running on NodeJS. We have selected that API framework as it's very well-suited for fast prototyping and debugging. Also, as the most commonly-used API framework in NodeJS framework, it has top quality documentation and a very huge community behind it, which was very helpful as the team was new to API development.
    - **React:** React is the most popular and efficient framework that a front end project can be done. This happens because in React, there is a layer of a Virtual DOM that analyzes the current

and previous DOMs and find differences, thus renders only those differences. Furthermore, it reduces the load of whole page components to a much smaller scale and agile to implement and run.

    * **Postman:** Postman is one of the most popular tools used in API testing. Postman played a vital role in our project especially in the backend part where we had to test certain endpoints to make sure we had written a particular API functionality correctly. Thanks to Postman we have used HTTP requests to test our API.
  - **Unit Testing**
    * **Mocha and Chai:** These two libraries are widely used by Javascript projects, and as a result, they have become mature and well-documented over time. Besides that, we used those for unit testing as they are simple and straightforward, which made testing easier.

- **Processes**
  - **Pull Requests:** The process that we encountered during development as following:
    1. Create a new branch from dev(main development branch)
    2. Make some changes according to assigned issue
    3. Commit those changes
    4. Create pull-request
       (a) Fill the title
       (b) Fill the description
       (c) Label the pr
       (d) Fill reviewers
    5. Make more changes if needed
    6. Merge with dev
  - **Reviews:** The process that we encountered during reviewing other codes as following:
    1. Checkout to that branch.
    2. Build the project.
    3. Check whether it is successfully built or not.
    4. Establish main work of assigned individual.
    5. Control whether it does what is needed.
    6. Control whether there exists unnecessary variables.
    7. Control whether there exists a forgotten corner case.
    8. Control whether code is well commented.
    9. Run the tests.
    10. Control whether the branch has no conflict with base branch.
    11. If this all passes, approve. Else, emphasize the missing part via the help of comments, or our communication channel.

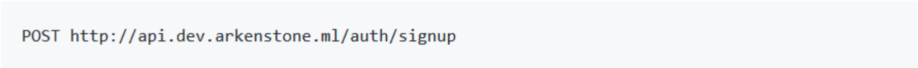
7 API

- POST Signup User

  Users send their personal information required to signup to the system.

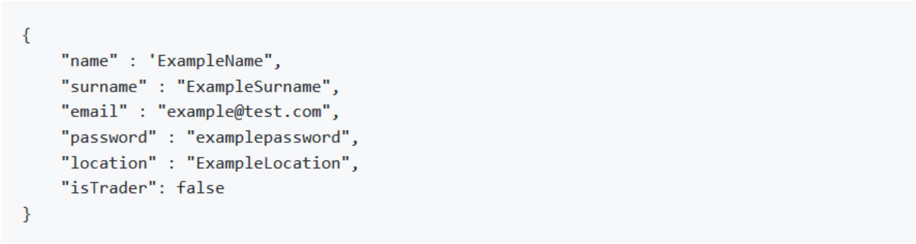
  - Endpoint: /auth/signup
  - Request body: Application/JSON
  - Required parameters:
    - name: String
    - surname: String
    - email: String
    - password: String
    - location: String
  - Possible error codes:
    - 400: Duplicate email or missing parameter in request body

  Sample API Request:

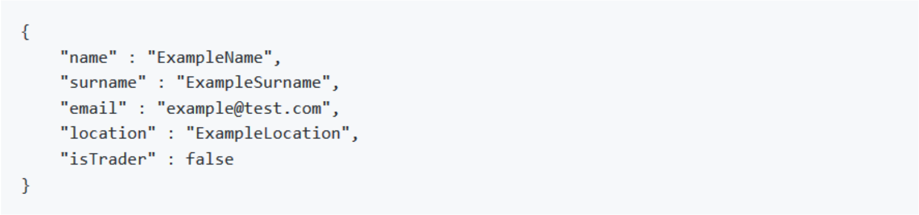
  ```
  POST http://api.dev.arkenstone.ml/auth/signup
  ```

  Sample Request Body:

  ```
  {
      "name" : 'ExampleName",
      "surname" : "ExampleSurname",
      "email" : "example@test.com",
      "password" : "examplepassword",
      "location" : "ExampleLocation",
      "isTrader": false
  }
  ```

  Sample API Response:

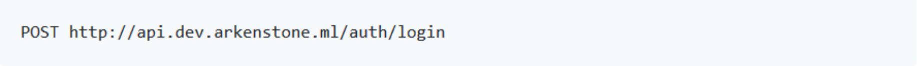
  ```
  {
      "name" : "ExampleName",
      "surname" : "ExampleSurname",
      "email" : "example@test.com",
      "location" : "ExampleLocation",
      "isTrader" : false
  }
  ```

- POST Login User

  Users send their login credentials, which are email and password, to login to the system. The response contains a cookie they may use for further operations.

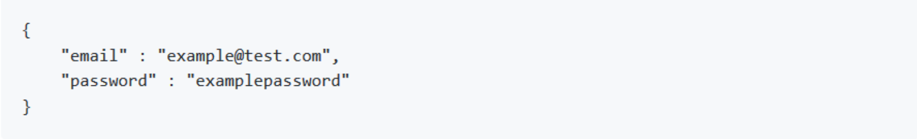
  - Endpoint: /auth/login
  - Request body: Application/JSON
  - Required parameters:
    - email: String
    - password: String
  - Possible error codes:
    - 400: Missing parameter or nonexistent email-password pair

  Sample API Request:



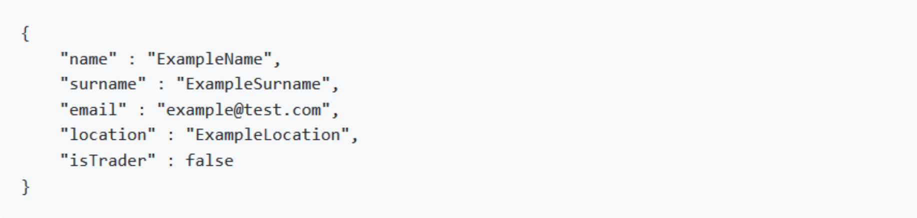
```
POST http://api.dev.arkenstone.ml/auth/login
```

  Sample Request Body:



```
{
    "email" : "example@test.com",
    "password" : "examplepassword"
}
```

  Sample API Response:



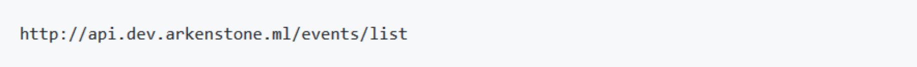
```
{
    "name" : "ExampleName",
    "surname" : "ExampleSurname",
    "email" : "example@test.com",
    "location" : "ExampleLocation",
    "isTrader" : false
}
```

- GET List events

  Gives a list of events with details of significance, country and forecast values.

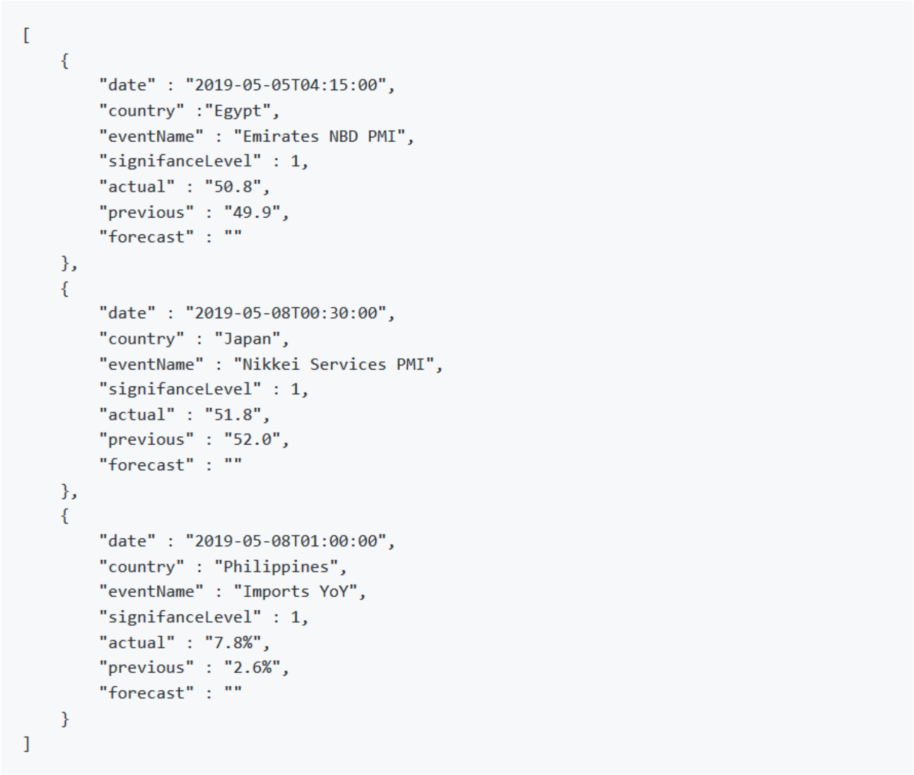
  - Endpoint: /events/list

  Sample API Request:



```
http://api.dev.arkenstone.ml/events/list
```

  Sample API Response:



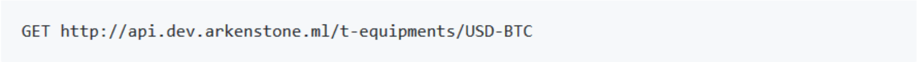
```
[
    {
        "date" : "2019-05-05T04:15:00",
        "country" :"Egypt",
        "eventName" : "Emirates NBD PMI",
        "signifanceLevel" : 1,
        "actual" : "50.8",
        "previous" : "49.9",
        "forecast" : ""
    },
    {
        "date" : "2019-05-08T00:30:00",
        "country" : "Japan",
        "eventName" : "Nikkei Services PMI",
        "signifanceLevel" : 1,
        "actual" : "51.8",
        "previous" : "52.0",
        "forecast" : ""
    },
    {
        "date" : "2019-05-08T01:00:00",
        "country" : "Philippines",
        "eventName" : "Imports YoY",
        "signifanceLevel" : 1,
        "actual" : "7.8%",
        "previous" : "2.6%",
        "forecast" : ""
    }
]
```

- GET Retrieve a certain exchange rate

  Users use that endpoint to retrieve the exchange rate of given two currencies. The currency codes should be sent via the URL as 'currencyA' and 'currencyB'.

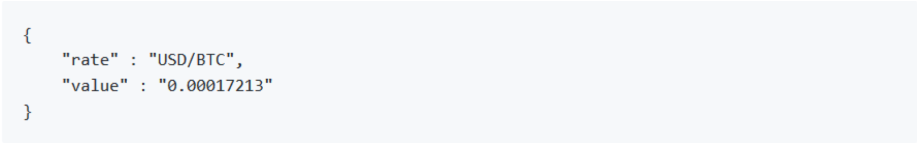
  - Endpoint: /t-equipments/:currencyA-:currencyB

    Sample API Request:

    ```
    GET http://api.dev.arkenstone.ml/t-equipments/USD-BTC
    ```

    Sample API Response:

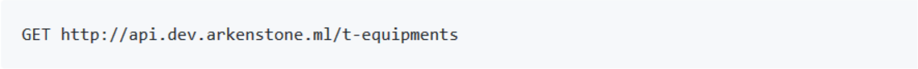
    ```
    {
        "rate" : "USD/BTC",
        "value" : "0.00017213"
    }
    ```

- GET List exchange rates of currencies against USD

Since USD is commonly used in the market as a base currency, that endpoint makes it easier for users to retrieve all exchange rates of currencies against USD.

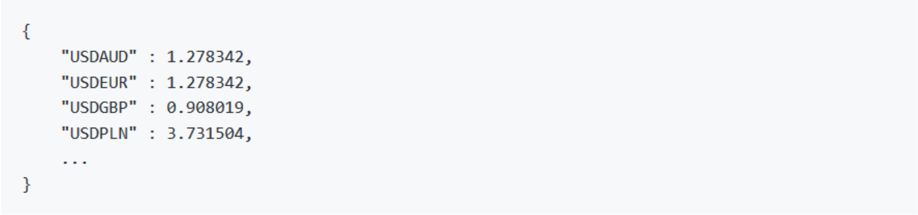
- Endpoint: /t-equipments

  Sample API Request:

  ```
  GET http://api.dev.arkenstone.ml/t-equipments
  ```

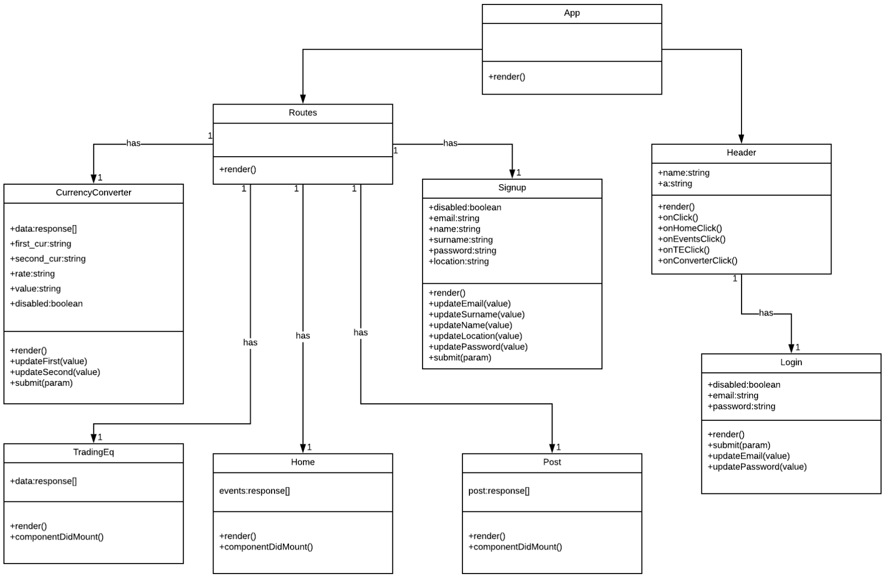
  Sample API Response:

  ```
  {
      "USDAUD" : 1.278342,
      "USDEUR" : 1.278342,
      "USDGBP" : 0.908019,
      "USDPLN" : 3.731504,
      ...
  }
  ```

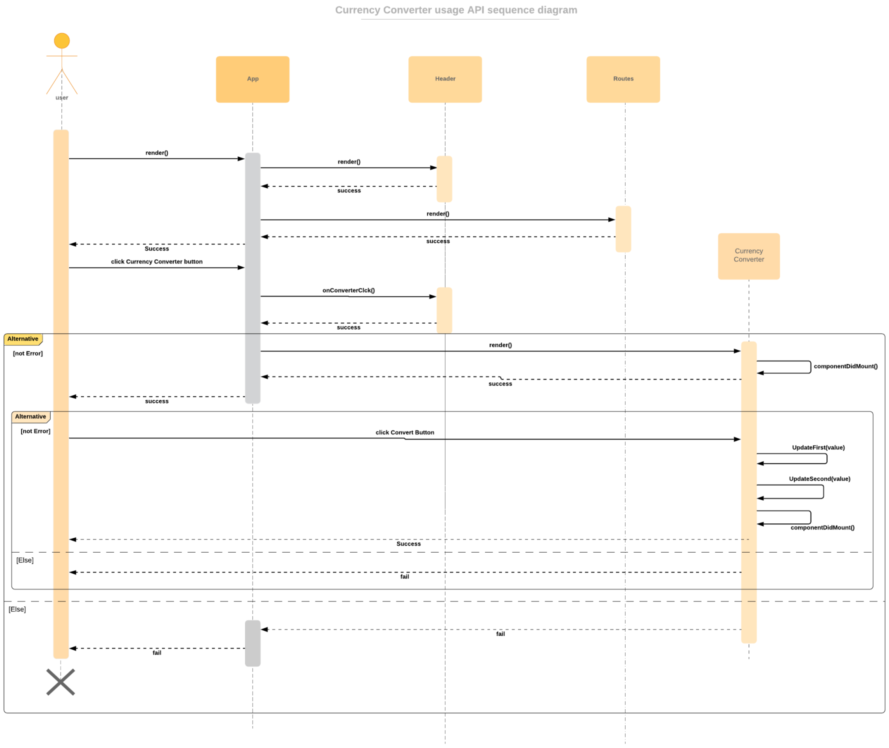
8 API Design

8.1 Class Diagram

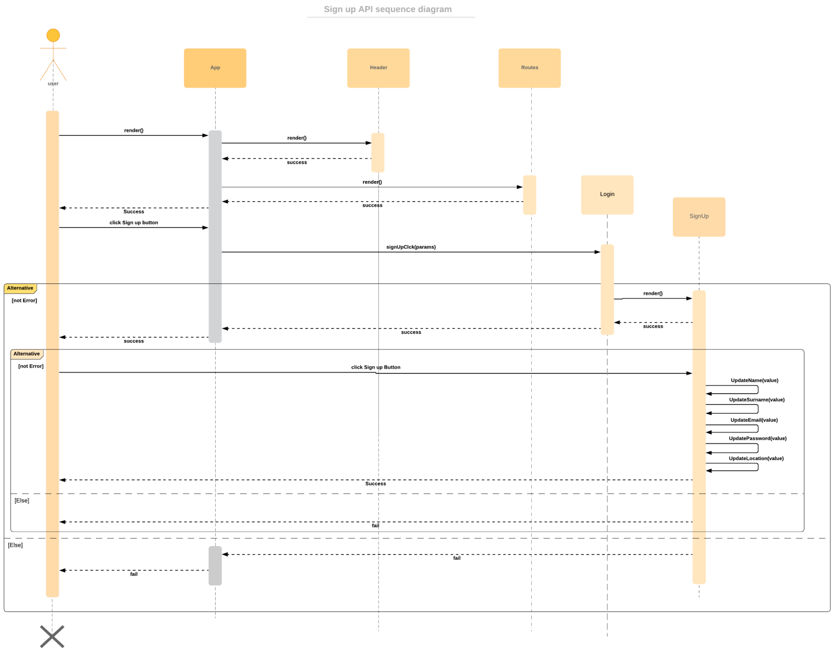


8.2 Sequence Diagrams

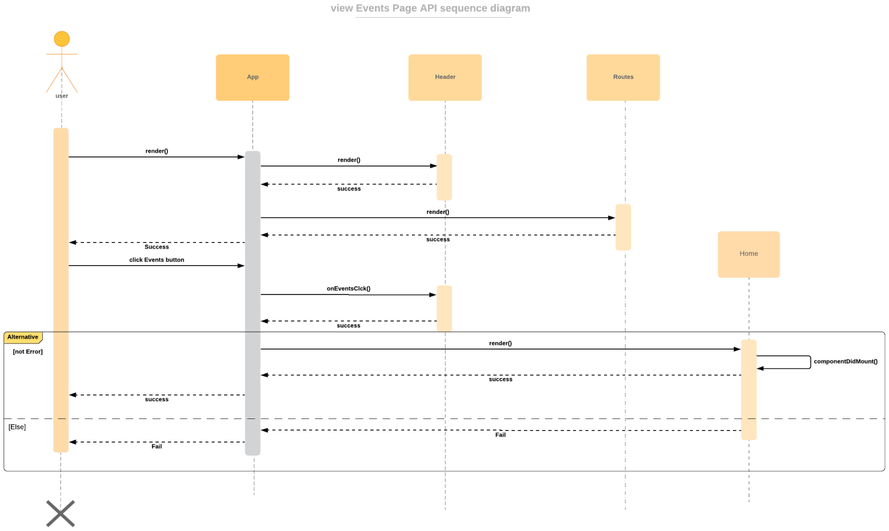
8.2.1 Currency Converter



8.2.2 Sign up



8.2.3 Events


view Events Page API sequence diagram
user
App
Header
Routes
Home
render()
render()
success
render()
success
Success
click Events button
onEventsClck()
success
Alternative
[not Error]
render()
componentDidMount()
success
success
[Else]
Fail
Fail