CMPE352 - Milestone 2 Report

Group Members

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

Spring , 2019

Contents

1 Executive Summary

1.1 Introduction

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

1.2 Work done so far

The Group 6 is in process of a project which will design and produce a trader platform. Until Milestone 1, the team has produced two more deliverables: Project Plan and API implementations. As of Milestone 1, the team has designed the abstract level of the project. For now, we are initiating the implementations.
In the project plan, the team has documented the near future plans of the group. As stated, the projected future tasks have been carried out. The stated API implementations have been carried out. Not only the listed ones, but the group has also developed some additional APIs that we've found to be useful. The API implementations have been explained in detail below:
The Group 6 members have built the APIs in a correlation with our mission, that is The Trader Platform Project. Our priority has been building the APIs that are related to the project. The APIs have been listed as follows:
* Google Maps API: The Google Maps will be necessary for getting the address information of the user. The functionality of the Google Maps API is very essential for the project.
* Trading Economics News API: The API enables us to get economic news worldwide.
* Trading Economics Events API: The API enables us to get economic events worldwide.
* Currency API: The API by data.fixer.io enables us to get the currency rates.
* Twitter API: Twitter API enables us to publish the tweets of selected accounts. Therefore, the users can follow some top trend accounts and utilize them to make their decisions.

1.3 Road ahead

The group members are planning to build front-end parts of the APIs. Plus, in order to get prepared to the next semester, the group members plan to learn some basic concepts about Android and Web development. After the group members have earned some basic insight, according to their interests, we are planning to make a division of labor. Then the implementation of the project will be initiated.

1.4 Challenges you met as a group

The main challenges since the milestone 1 was to adapt the implementation of APIs and some general concepts for building APIs. The group members have adapted into use of Django and API implementations.

2 List and Status of Deliverables

#	Deliverable	Due Date	Status	Explanations
1	Project Plan	29.04.2029	+	The roadmap of the parctise app is drawn -including the example tutorial application-
2	UI	04.05.2019	+	The user interface of our app is implemented.
3	Currency API	04.05.2019	+	An API service that serves the parity results based on the selected base currency is implemented.
4	News API	04.05.2019	+	An API service that serves economic news is implemented.
5	Events API	04.05.2019	+	An API service that serves economic events is implemented.
6	Twitter API	04.05.2019	+	An API service that serves some links that lead to events and news on Twitter.
7	Google API	04.05.2019	+	An API service that serves autocomplete places for given input keyword and Google Geocode Object from given place id.

3 Evaluation of the Status of Deliverables

3.1 Project Plan

Project Plan has an important role in development of the project. It consists of jobs that are done by team members and deadlines and starting dates of each job, and a timeline of our future plan.

We believe that it helped us a lot to see the projects' progress and to analyze how much time we need to deliver a certain job. With this, seeing all of our old accomplishments in a graph, improved us in making future plans for our future expected accomplishments.

3.2 Application Programming Interface

3.2.1 News API

News API enables us to fetch some news from one source and feed the data to frontend. User can list the news and read a new among of them. One of the project requirement is providing news to user. This API enables us to satisfy the project requirement about news.

3.2.2 Currency Rate API

Currency Rate API enables us to fetch the daily values of different currents' ratios between them. This is an important part of the Traders' Platform app. With this API users are able to see live value of a current and make investments according to them. This API enables us to satisfy one of the project requirements.

3.2.3 Events API

Events API enables us to fetch an economic event that happened or will happen soon. Users can see events' names, times and places that the event occured and some economic values as well which represents the effect of the event. With this API's existence in the app it becomes way more trivial for users to make comments, predictions and investments in the traders' platform since it is possible to reach effects of the events (if the event hasn't occurred yet, a predicted effect is also fetched.)

3.2.4 Twitter API

Twitter API enables us to fetch Tweets that are about some economic events all around the world. Tweets that are followed are fetched from big economic news accounts. User can list links of lots of news and by clicking to them they can read the tweet and see the account that the tweet was fetched from. This API enables us to inform the customer more thoroughly.

3.2.5 Google Maps API

Google Map API provides us to get information from user about their location. We can show a map to user where s/he can select their location from map. Another way for user to describe his/her location is that s/he can type the location inside the user interface and see the suggestions about location. User can select a suggestion from the suggestions and we can identify the location's latitude and longitude.

This API meets our requirements and we have satisfied one of the project requirements.

3.3 User Interface (UI)

Our API development assignment contains different APIs which do different things from each other. The things, that are done by APIs, should be served to user as an interface in which the user can interact with the system itself. The User Interface part of the project is where the our APIs' functionalities are displayed and served to user. This part contains the integrations of our APIs into user environment.

Our UI has the ability to provide each capabilities of APIs to user. One of the important feature of UI is that it has error handling where the user can see what was happened behind the scene. Another feature UI has is that it is loaded one time into browser and do its job behind the scene, which is called Single Page Application (SAP). That enables user to navigate over pages easily, provide inputs to system and retrieve results from the system without loading page again.

Since our UI is where the user interacts with system, it has great impact on the system. Its usability describes the system itself.

4 Summary of Work done

Group Member	Contribution
Baran Deniz Korkmaz	Filled the past plans of the project with Irmak Güzey. Made articles and news api with Barış Ege Sevgili.
Barış Ege Sevgili	Filled the future plans of the project with Burak Yüksel and Enes Turan Özcan. Made articles and news api with Barış Baran Deniz Korkmaz.
Burak İkan Yıldız	Reviewed the project plan. Made currency rate api.
Burak Yüksel	Filled the future plans of the project with Barış Ege Sevgili and Enes Turan Özcan. Wrote some unit tests for the project.
Enes Turan Özcan	Filled the future plans of the project with Barış Ege Sevgili and Burak Yüksel. Mde economic events api.
Eray Sezgin	
Fatih Mustafa Kurt	Made Google Maps & Account api. Made user interface of the other api's. Made the deployment of the project to the aws with Sadullah Gültekin
Furkan Nane	
Irmak Güzey	Filled the past plans of the project with Baran Deniz Korkmaz. Made economic events api.
Sadullah Gültekin	Wrote an example API for team members to understand how api's work. Made the twitter api. Made the deployment of the project to the aws with Fatih Mustafa Kurt.
All Members	Downloaded and installed Django framework. Chose a random API and tried to make a test implementation with it in order to understand how does api's work. Read the Django tutorial and got familiar with the overall running of the Django.

5 Project plan

In the first week of the project, we decided to understand the framework that we choose to develop our API. Sadullah Gultekin told about Django at the meeting for us to get the basics and task of each person was to install and get familiar with the framework on his/her own. In addition to that, everyone was expected to look for useful APIs that can be used for our project.

In the next meeting, we allocated implementation of APIs that we came up with. Sadullah Gultekin initiated a sample project on Django. Each person/pair created a new branch and started implementing their API. After everything started to get in shape, Fatih Mustafa Kurt started to implement user interface for our API.

After the reviews are made, branches are merged into master and Fatih Mustafa Kurt and Sadullah Gultekin deploy our project to the AWS.

6 Evaluation of tools used

Our group members have used number of tools to create this project. These tools helped our team to communicate, develop and design the project.

One of the tools that is used is lucidchart.com to draw project diagrams. This tool is used effectively to present the design of the project. Similar tools are used to create mockups and project plans. One such website that helped our team to create reports is overleaf.com. These tools helped our team to clarify the design of the project. These web based tools offer such great potential without forcing one to setup programs to computer. Thus they make it easier to develop charts, diagrams, pdf files etc.

Communication tools were slacks and whatsapp. Slacks was not used as much but it is clear that whatsapp was an important tool for our team to communicate and share ideas as well as other thoughts. Having a dynamic tool like whatsapp is crucial to any project because it effectively increases the communication and thus helps team thought process immensely.

One of the tools that helped our team to write code and design the code is Django. Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Ease of use and rapid development tools were highly recommended and offered a common place for our team to communicate and write the code. Having a high-level Python Web framework like Django helps a team to understand each other and think in common terms rather than be split and not understand each other.

We have decided to implement the backend in Python. We have decided that our common IDE will be PyCharm. These tools are well known code languages and IDEs. They are used efficiently and they based a common place for our team to develop the project with ease. Also some other IDES and languages are considered but as a general conclusion these tools are found to be the easiest to use.

From general point of view it is safe to say that our team used these tools efficiently and successfully completed given tasks as well as designing the project. Although it is always possible to use the tools more efficiently we can not discard the fact that our team had no problems completing the given tasks so as a conclusion we can say that our team was pretty efficient with the use of the above mentioned tools.

7 API URL (documented)

BASE URL: http://54.89.235.179/

Our API can be divided into 5 parts as following:

7.1 Events API

Events

Get the details of economic events perfoming today.

URL : `/events`

Method : `GET`

Auth required : NO

Permissions required : None

Parameters : None

Success Response

Code : `200 OK`

The return type is a list of dictionaries of which elements contain event information.

Content examples


```
[{
    "name": "General Elections",
    "country": "Panama",
    "date": "2019-05-05",
    "rep": 2,
    "actual": "",
    "prev": "",
    "forecast": ""
},
{
    "name": "Emirates NBD PMI",
    "country": "Egypt",
    "date": "2019-05-05",
    "rep": 1,
    "actual": "50.8",
    "prev": "49.9",
    "forecast": "50.1"
}]
```


Notes

- "rep" key resresents the importance of an economic event ranging from 1 to 3.
- "actual", "prev" and "forecast" keys might have empty values for some events.

7.2 Currency Rate API

Currency Rate

After obtaining request from user, it respond the requested currency translation or all currency translations.

URL : `/currencyrate/<str:source>to<str:target>`

Method : `POST`

Auth required : NO

Permissions required : None

Parameters : source, target

Success Response

Code : `200 OK`

Content examples

```
{
    "response": success,
    "datetime": "2019-05-06 11:49:05",
    "Source to Target: "6.708424",
}
```

7.3 News API

News

Having information about the newly emerged financial news

URL : `/news`

Method : `GET`

Auth required : NO

Permissions required : None

Parameters : None

Success Response

Code : `200 OK`

Content examples

List of dictionaries is returned, each are itemized by their id's. Information about the news are categorized accordingly.


```
"77111":
   {"id": "77111",
    "title": "Rwanda Inflation Rate Lowest in 3 Months in April",
    "description": "The annual inflation rate in Rwanda eased to 0.2 percent in April 2019 from 1.1 percent in
                    the lowest inflation since January, as prices fell further for food & non-alcoholic beverag
                    March) and slowed for transport (3% vs 6.7%). Also, cost was flat for housing & utilities (
                    monthly basis, consumer prices went up 0.3 percent, following a 1.1 percent rise in the pre
    "date": "2019-05-10",
    "country": "Rwanda",
    "symbol": "RwandaIR",
    "url": "/rwanda/inflation-cpi"},
"77107":
   {"id": "77107",
   "title": "Kosovo Inflation Rate Highest in Over 6 Years",
   "description": "Annual inflation rate in Kosovo edged up to 3.4 percent in April of 2019 from 3.3 percent ir
                   the highest inflation rate since January of 2013, as cost rose faster for food & non-alcohol
                   from 6.7 percent in March) and for furniture & household equipment (0.9 percent vs 0.3 perce
                   prices went up at a softer pace for transport (2.2 percent vs 2.7 percent); alcoholic bevera
                   vs 2.8 percent) and miscellaneous goods & services (1.0 percent vs 1.3 percent) while cost f
                   (-1.4 percent vs -1.3 percent); recreation & culture (-0.3 percent vs -0.4 percent) and comm
                   -0.5 percent). On a monthly basis, consumer prices went up 0.1 percent, the same as in March
   "date": "2019-05-10",
   "country": "Kosovo",
   "symbol": "KOSSOVOINFNRATE",
   "url": "/kosovo/inflation-cpi"}
```

7.4 Twitter API

Tweets

- Returns some links that leads to events and news

URL : `/twitter`

Method : `GET`

Auth required : NO

Permissions required : None

Parameters : None

Success Response

Code : `200 OK`

Content examples


```
{
    "all_links": [
        "https://go.shr.lc/2L9MKtJ",
        "https://www.publish0x.com/earning-crypto/what-are-security-tokens-and-stos-xllynk?a=mWZdPqweKg",
        "https://go.shr.lc/2VeKaGv",
        "http://bit.ly/2U1f8gz",
        "https://www.publish0x.com/consensus-report/along-with-bitcos-prices-lightg-network-numbers-explod-xe
    ]
}
```

7.5 Google API

7.5.1 Google Place

Google Place

Suggest autocomplete places for given input keyword

URL : `/google/map/place`

Method : `GET`

Auth required : NO

Permissions required : None

Parameters : input

Success Response

Place predictions about given input

Code : `200 OK`

Content examples


```
{
    "predictions" : [
        {
            "description" : "Ankara, Turkey",
            "id" : "908fdf0efc46fb81721d9b06ff54ee23e8703a4f",
            "matched_substrings" : [
                {
                    "length" : 6,
                    "offset" : 0
                }
            ],
            "place_id" : "ChIJsS1zINVH0xQRjSuEwLBX3As",
            "reference" : "ChIJsS1zINVH0xQRjSuEwLBX3As",
            "structured_formatting" : {
                "main_text" : "Ankara",
                "main_text_matched_substrings" : [
                    {
                        "length" : 6,
                        "offset" : 0
                    }
                ],
                "secondary_text" : "Turkey"
            },
            "terms" : [
                {
                    "offset" : 0,
                    "value" : "Ankara"
                },
                {
                    "offset" : 8,
                    "value" : "Turkey"
                }
            ],
            "types" : [ "locality", "political", "geocode" ]
        }
    ],
    "status" : "OK"
}
```

7.5.2 Google Geocode

Google Geocode

Returns Google Geocode Object from given place_id

URL : `/google/map/geocode`

Method : `GET`

Auth required : NO

Permissions required : None

Parameters : input

Success Response

Geocode Object

Code : `200 OK`

Content examples


```
{
   "results" : [
      {
         "address_components" : [
            {
               "long_name" : "Ankara",
               "short_name" : "Ankara",
               "types" : [ "locality", "political" ]
            },
            {
               "long_name" : "Ankara",
               "short_name" : "Ankara",
               "types" : [ "administrative_area_level_1", "political" ]
            },
            {
               "long_name" : "Turkey",
               "short_name" : "TR",
               "types" : [ "country", "political" ]
            }
         ],
         "formatted_address" : "Ankara, Turkey",
         "geometry" : {
            "bounds" : {
               "northeast" : {
                  "lat" : 40.076332,
                  "lng" : 33.007056
               },
               "southwest" : {
                  "lat" : 39.7304211,
                  "lng" : 32.5184735
               }
            },
            "location" : {
               "lat" : 39.9333635,
               "lng" : 32.8597419
            },
            "location_type" : "APPROXIMATE",
            "viewport" : {
               "northeast" : {
                  "lat" : 40.076332,
                  "lng" : 33.007056
               },
               "southwest" : {
                  "lat" : 39.7304211,
                  "lng" : 32.5184735
               }
            }
         },
         "place_id" : "ChIJsS1zINVH0xQRjSuEwLBX3As",
         "types" : [ "locality", "political" ]
      }
   ],
   "status" : "OK"
}
```