Milestone Report 2

12.05.2019

1. Executive Summary

1.1. Introduction

The purpose of this milestone report is to analyse implementation and project plan assignments. In order to improve our project in next steps, we believe examining the work done, the processes, tools chosen and current status is important. We prepared the report considering lectures and class documents. We will discuss what we had done after milestone 1 report (1.4.2019).

1.2. Work Done So Far

We believe that we have completed a great amount of work after the first milestone and until this one. Although, in terms of number of assignments, we had less things to do after the first milestone; the amount of work needed to achieve the last two assignments have certainly surpassed the previous assignments. Our first task after the first milestone was creating a project plan. This required us to retroactively review what we had done before that and what were the deadlines, who were the assignees and so on. Thankfully, since we tracked all the work using our GitHub repo in a systematic way, it was not very hard to obtain these info. Using all these, we created our project plan. In the following weeks, we updated our project plan with the new tasks.

The second and last (so far) assignment after the first milestone was the implementation assignment. Without a doubt, this assignment required the most work both individually and as a group, compared to the previous assignments. In this assignment, we believe we chose a very useful 3rd party API and we created a very useful API ourselves using that; and we will benefit significantly from this assignment in the next semester while we are working our actual app.

For this assignment, our first task was to find a useful API. We chose to use the free AlphaVantage API, which provides realtime and historic stock price data (and also various technical indicators). Using this, we developed various endpoints such as showing the current price of a stock, showing the moving average of a stock's price and so on. Also, we developed various other endpoints such as creating comments, price predictions etc about stocks; which we thought that can be useful for our project next semester. Lastly, we developed a frontend application that "drives" this API. After all, we had created a complete, full stack Web application that allows to you interact with other users and see various data about various trading

equipments. And, we have done other work, too, related to this assignment. We created documentation about our API, we deployed the application to an AWS server, we coded unit tests and so on.

1.3. Road Ahead

We have done some basic functionalities such as creating and reading comments, prediction about trading equipment' prices in our Web application side. It allows to user interact with other users. We have encountered some problems at the beginning of the implementation but we're gaining momentum and aiming to get faster in time. Firstly , we aim to have user registration system for our Web application with E-mail confirmation integrated and have user login system. Also we aim to have :

- Android application which has same functionalities with Web application

- User profile page with some features such as portfolio integrated

- Basic search functionalities implemented in all platforms

- Profit/Loss section which is private to the user in all platforms

and much more for our Final Milestone.

1.4. Challenges We Met As a Group

- For the project plan, indicating the exact name and duration of the future tasks were hard. We couldn't do it right away, first we waited for a while to understand the necessities of the next assignment.
- For the API implementation:
    - Most of the group members had not worked on REST API's before, so it was hard to create a deliverable in a field we're inexperienced.
    - It was challenging to write code as a group on GitHub. Most of us also had not used git features before, so we learned how to use git as well.
    - It was hard to divide the work as backend and frontend and at the same to create a program that is working as a whole.

2. Deliverables

REPOSITORY	Created	Revised After Feedback	Done
--**Project Documents**			
----**Group Meetings**	**13.02.2019**	--	✓
----**Customer Meetings**	**22.02.2019**	--	✓
----**PS's**	**26.02.2019**	--	✓
----**Github Research Assignment**	**29.02.2019**	--	✓
----**Member Pages**	**26.02.2019**	--	✓
----**Communication Plan**	**24.02.2019**	✓	✓
----**Readme**	**13.02.2019**	--	✓
--**Requirements**	**24.02.2019**	✓	✓
--**Personas**	**27.02.2019**	✓	✓
--**User Stories and Mockups**	**27.02.2019**	✓	✓
--**Diagrams**			
----**Use Case Diagram**	**11.03.2019**	✓	✓
----**Class Diagram**	**14.03.2019**	✓	✓
----**Sequence Diagrams**	**17.03.2019**	✓	✓
--**Issue Management**	--	--	--
--**practice-app**			
----**backend**	**04.05.2019**		✓

----frontend	02.05.2019		✓
----settings	02.05.2019		✓

3. Evaluation of Deliverables

We were able to deliver everything we planned for the second milestone except user registration and login system. Our first priority is implementation of these for the rest of the way. Since it was the beginning of the project, our production was slow , but each team member got the point and we hope that we will improve our speed and quality of work. For the rest of the way, We will continue to work on features that delivered to improve their quality and fix bugs. Finally , We will try to achieve our goals that are creating a proper Web and Android application which have all functionalities that are in the project description for our Traders Platform.

4. Work Done by Team Members

Ceren Tahtasız	● Reviewed project plan document ● Used PostgreSQL and pgAdmin for database features, planned table for comments in database ● Contributed to Implementation of backend using Flask with Sercan, Alper, Barış, Yusuf, Ahmet ● General issue management and tracking ● Filling introduction and challenges parts in Milestone 2 ● Created meeting note documents in Github ● Proofread API documentation ● Moderated weekly meetings
Mehmet Can Ünal	● Attended in group meetings and took part in discussions ● Took part in implementation of frontend with Alper, Ali and Ahmet
Alper Çakan	● Prepared the initial version of the project plan document ● Added a pull request template ● Initialized the frontend and backend application using Flask and React frameworks, created file layouts ● Set up the DB system (sqlalchemy with ActiveRecord mixin) and config systems of the backend app

	- Added health check, current price, list comments and list stocks endpoints - Wrote the API docs - Fixed small bugs of various endpoints - Refactored the API to have a better separation of concerns (moved the price API calls to another file etc) - Added a unit test for list comments endpoint - Helped implement the frontend components for various API features such as showing the price, showing the comments, creating comments, listing stocks, creating predictions etc. - General issue maintenance, tracking and coordination
Yusuf Yüksel	- Contributed preparing project plan document - Implementation of comment delete endpoint using Flask framework in Python - Reviewing comment delete and comment create methods - Testing comment delete and comment create methods manually with help of pgAdmin with Sercan - Testing last version of Web application - General issue maintenance and tracking - Contribution to weekly meetings - Filling Road Ahead part of the Milestone 2 - Filling Evaluation of Deliverables part of the Milestone 2
Efe Önal	- Updated project plan document by elaborating milestone report. - Contributed to API choice discussions at the meeting. - Testing comment functionality via Postman with Sercan, Ali, Furkan, Ahmet at meeting. - Updated deliverables on this document.
Sercan Ersoy	
Barış Zöngür	- General issue maintenance and tracking - Assignment 7 in project plan - Assignment 7 in Deliverables part - Attended in group meetings - Review of a part in back-end - Took a part in back-end implementation
Ahmet Yiğit Gedik	- Added Project Plan to our Wiki. - Prepared agendas for Weekly Meetings. - Attended all of the meetings and at these meetings took part in discussions about progress of practice-app. - Organized meetings to implement frontend.

	● Took part in implementation of frontend with Alper, Ali and Mehmet.
Furkan Aydar	● Attended to the three meetings before we started developing our API. ● Contributed to the process of deciding the tech we will use to develop the API. ● Discussed functionalities and the API we will include in our application and the rough version of our user interface styling. ● Implementation of Prediction model to our database ● Implementation of create prediction, get prediction methods ● Implementation of get_sma method which uses AlphaVantage API ● Testing comment delete and comment create methods manually with help of pgAdmin, with team members ● Testing the function get_sma for different trading equipments and parameters ● Testing the deployed application with user interface and the API commands
Ali Özden	● Contributed to meeting which we choose our API's. ● Took part in testing comment functionality with Sercan, Furkan, Ahmet, Efe. ● Took part in implementation of frontend with Alper, Ahmet and Mehmet Can.

5. Project Plan
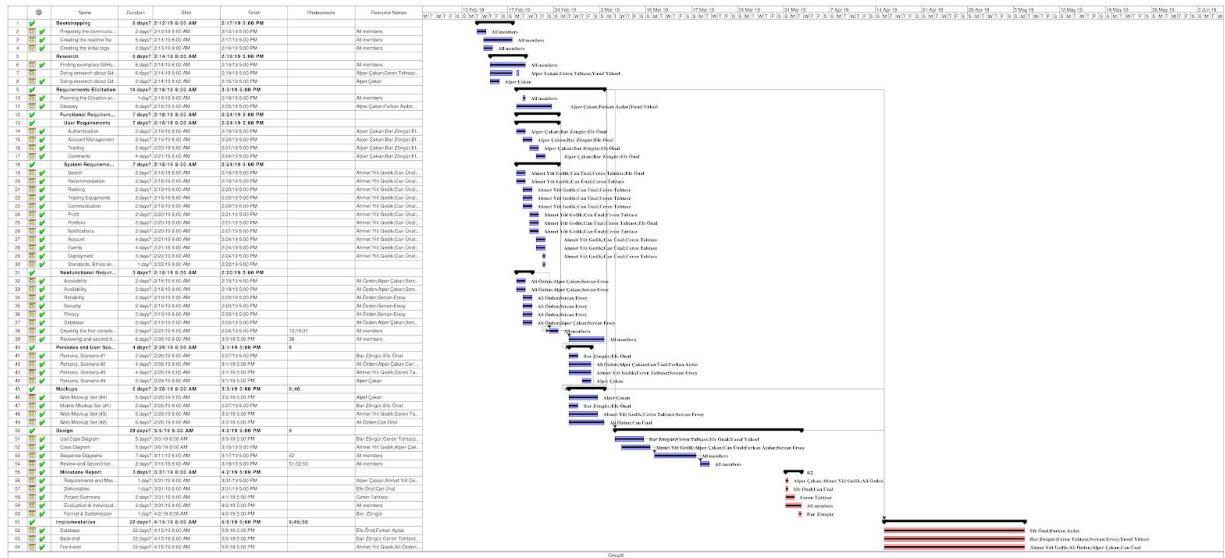
6. Evaluation of Tools and Processes

React: It was a simple and efficient framework of JS that combines basic html, css and js. Its difficulty is coming basically from the fact that if you do not know a little bit of html, css and js, you cannot proceed fast. One should have a fundamental knowledge about these languages. One of its advantages is if you know a little bit of these languages, you can proceed faster than working with those languages separately. Another advantage is that when you run the app and make changes to the .js files, application recognizes the changes and updates itself automatically.

Flask: Flask was also a simple and efficient framework of Python. One of its advantages is that it is a Python framework and we all acknowledged with Python from some of the required CMPE courses. Another advantage is that it is compatible with the most of the SQL databases. Database language we used was Postgresql and it was also compatible with it.
Its difficulty was not very much but getting accustomed to it took a while. Compatibility issues can be hard for someone who is not experienced with database systems, basic API's and Python frameworks.

7. API

The API can be reached at http://18.188.149.114:8080/

The frontend application can be reached at http://18.188.149.114/

The API documentation can be reached at
https://documenter.getpostman.com/view/2365580/S1LySmQB