20.05.2022

CMPE 352.01 SPRING 2022

MILESTONE 2

GROUP 9 REPORT

Berke Çalışkan

Burak Ferit Aktan

Furkan Özdemir

Kutay Saran

Berkant Koç

Hüseyin Türker Erdem

Ömer Faruk Şişman

Can Bora Uğur

1- Executive Summary

1.1- Summary of the project

We are the members of bounswe Group 9. This report mainly contains our group work in CMPE-352 (Fundamentals of Software Engineering) course. In this report, we documented our works for the practice app project. In the practice app, we tried to stay as connected as possible to the previous works. So, we used the git versioning system to manage the team code-base effectively. We pulled the group repository into our locals, we created new branches to implement some functions and unit tests, we tried to gain skills to work with third-party services through a RESTful Application Programming Interface (API). We fetched some content from that API and processed what it returned in JSON format. We were also expected to build a RESTful API, dockerize our application and deploy our application to a server. Each team member had to write a post and a get method. Lastly, we provided a very simple front-end for an end user that shows the results we get from external API. In this report, you can see the work done by each team member in detail. We designed an application named eventapp. You can see functionality of our application in the "Basic Functionality of the Project" part of this report. We used:

- Git as team software development,
- Python as programming language,
- Django as python-based web framework,
- HTML for user interface design,
- Mysql for our database.

You can see a detailed evaluation report for these tools in "Evaluation of Tools and Processes".

1.2- Overall Status

- All team members used git as a version control system.
- All team members used a personal development environment (IDE) that is connected to the team software development platform.
- All team members created a function that uses at least one external API.
- All team members provide a simple user interface to their functions.
- All team members created unit tests for their functions.
- All team members developed an API with at least one GET and POST methods.
- We pushed all our work to the practice-app folder in our repository.
- All team members requested code review.
- All team members review the code of other team members.
- All team members created pull requests that implied code review.
- All team members updated their GitHub wiki page with a description of the API and the functionality they introduced.
- We couldn’t dockerize our application and couldn’t deploy it to AWS EC2.

This project was a great chance to gain experience in software development. We had the opportunity to work with our team. We learned to use Git commands, a python based web framework, mysql database management system and interface description language. Unfortunately, we couldn’t dockerize our application

1.3- URL of Deployed Application

Since we couldn't dockerize our application, there is no URL. We tried to dockerize this application for a long time but we get many different errors in our trials.

1.4- API URL

Since we couldn't dockerize our application, there is no URL.

1.5- Basic Functionalities of Project

- Users can sign in and sign up for the event app or they can view API's that we developed.
- Logged in users can add education records to their education history. An education record consists of 3 datas:institute name, degree type (like BSc), end year. As an example, a user can add he has graduated/will graduate from his master's degree at Bogazici University in 2025.
- Logged in users can view their education histories that they added to the system.
- Logged in users can view the currency that is being used in the specified country.
- Users can view the list of universities (and their websites) in a country by giving the name of the country. (An external API is used to fetch the university names)
- Logged users can query a Formula 1 race standing by entering race year and race round in the season.
- Logged users can add event to database .
- Logged users can view random useless fact and useless fact of today .
- Logged users can view github profiles with github username.
- Logged users can post github profiles, and get github profiles.
- Logged users can create football players with 4 data which are name,goal count,assist count and team.

- Logged users can view player informations that they added to system.
- Logged in users can get a summary for any subject that they submit.

1.6- Challenges

The first difficulty we encountered is to understand the project description. It was very ambiguous and we couldn't figure out what we would do exactly in the first place. Secondly, as can be expected, a frequently encountered problem in group projects is the creation of a collaborative working environment. Although we had 10 members in our team, 2 of them didn't join the conversations. We used git as a version control system. Sometimes it can be a challenging task to merge our work with the work of teammates. To overcome this challenge, firstly we created separate branches for our work. Then we merged these branches to the "master" branch one by one. In this way, we prevent conflicts which can occur while working in the same branch. Selecting tools was another challenge for us. We had to decide which framework and database we should use. Between Django and Flask, we decided to use Django in our project. We decided that way because we had some experience from the CMPE321 course. Another challenge was learning the tools that we used in our project. We used git, docker, mysql and some other programming tools. Some of our teammates had experience in these tools and they helped us learn to use them. We also shared the tutorials we found with each other.

1.7- List and Status of Deliverables

Deliverable Name	Delivery Status	Due Date	Delivery Date
Practice App	Delivered	20/05/2022	20/05/2022
Requirements	Delivered	20/05/2022	20/05/2022
Milestone 2 Report	Delivered	20/05/2022	20/05/2022

2- Project Plan

3- Summary of Personal Works

3.1. Ömer Faruk Şişman

Task	Link
Adding random activity functionality to the practice-app.	#133
Research on Bored API.	#135
Research on git branches and pull requests.	#136
Research on Django framework.	#137
Adding “back” and “find another” links to random activity.	#167 #168
Adding Geolocation API.	#171
Adding IP address information functionality.	#180
Research on Country API.	#181
Fixing bug in View IP Information Page.	#183
Writing unit tests for implemented functionalities.	#185 #186 #187 #188
Making reviews for other team members’ codes.	#189 #190 #191 #192 #193
Research on Docker and dockerizing a Django app.	#208
Writing requirements part of the Milestone 2 group report.	#209
Writing all of the Milestone 2 individual report.	#210

3.2 Burak Ferit Aktan

Task	Link
Reviewed #199 pull request	**#211** **#199**
Reviewed #159 pull request	**#212** **#159**
Implemented “show universities” functionality	**#144** **#146**

Implemented API endpoints/functions for university API	**#203**
Implemented UI for show_universities functionality	#158
Added unit tests to "show universities" functionality and "university API" .	**#172** **#163** **#162** **#161** **#205**
Implemented “add and view education history” functionalities	**#153**
Created a new table and stored procedure at the database and create_db python file for “add and view education history” functionality	**#156** **#157**
Implemented education API	**#204**
Implemented UI pages for “add and view education history” functionalities	**#158**
Added unit tests to "add and view education history" functionality and education API	**#205** **#163** **#162** **#161**
Contributed to Group Report	
Writing all of the Milestone 2 Individual Report	**#222**

3.3 Berke Çalışkan

Title	Link
Research on API’s and how to implement them	
Research for external API’s that I can use for the app	
Design and create an API for the app	link
Add findCurrency function to the app	link
Add unit tests for findCurrency function to the app	link
Making a review for other members branches	link
Adding requirements for findCurrency function	link

3.4 Kutay Saran

Task	Link
Adding random useless fact function to the practice-app.	#152
Research on external APIs	
Research on pull requests and git branches	#149
Research on Django Framework	#148
Adding add event functionality to the practice-app	#150
Writing unit tests for adding an event and view random useless fact functionalities	#194
Creating new table and procedure at the database for add event function	
Making reviews for team members' codes	#227 , #226
Adding Player Information API.	#224
Writing Milestone 2 individual report.	#223
Contributing to group report.	

3.5 Berkant Koç

Task	Link
Watching tutorials	• Link
Created HTML page for input to race standing	• Link
Created HTML page for standings	• Link
Created race standing functionality	• Link
Created unit test	• Link
Write API for My program	• Link
Prepared my personal deliverable	• Link
Contributing to group report	

3.6 Furkan Özdemir

Issues	Link
Documenting Meeting Notes 6.1	#119
Updating the Readme file of the practice app.	#120
Research on git branches and pull requests.	#122
Creating initial files of the practice app.	#123
Adding login and sign up functionalities.	#124 #130
Adding homepage to practice app.	#125
Creating database files of the practice app.	#126
Research on Github API.	#127
Adding github user info functionality to the practice app.	#128
Fixing the bug in add event functionality.	#141
Resolving conflict between branches	#142
Adding unit tests for the practice app.	#160
Research on building a REST API with django framework.	#175
Creating the github api for the practice app.	#173 #174
Reviewing the codes of other team members.	#176
Creating API page template.	#179 #229
Preparing group 9 milestone report.	#231
Preparing the individual report.	#230

Pull Requests	Link
Creating initial files for practice application.	#121
Adding github information function.	#129
Adding sign up functionality.	#131
Adding unit tests.	#159

Adding github API.	#165
Adding migration files.	#166
Adding API page template.	#178

Code Reviews	Link
Adding random fun activities functionality	#132
Adding the add event functionality.	#140
Adding back and another links.	#169
Adding geolocation API.	#170
Adding IP address information functionality.	#177
Unit tests for IP info and View Activity	#184

3.7 Can Bora Uğur

Issues	Link
Documenting Meeting Notes 6.2	#118
Adding “Wikipedia summary” functionality to site	#139
Adding API for the “Wikipedia summary” functionality	#200
Adding unit tests for the API	#207
Preparing individual report	#228

Pull Requests	Link
Adding “Wikipedia summary” functionality to site	#143
Adding API for the “Wikipedia summary” functionality	#201

Code Reviews	Link
“Show universities” functionality	#144

Adding unit tests to “show universities” functionality	#172
Burak API	#206

4- Evaluation of Tools and Processes

4.1. Evaluation Of Tools

4.1.1. Github

Github is our most essential platform. It is so important because every member in our group uploads the work done by themselves to Github. The job done includes but is not limited to meeting notes, requirements, scenarios, mock-ups, sequences and use-case diagrams. It helps us to track the work of others and our project. It also makes reviewing the job done by other members super easy. Besides those qualifications, the issue system available on Github also dramatically helps us in our project. Thanks to the issue system, any work currently being worked on or completed can be viewed, commented on, and reviewed with ease. The tags also help to identify the tasks. It is easy to filter issues by tags to find the specific problems. Since there are many tools to help us with the progress of our Project, Github is a great tool and working place that benefits us greatly.

4.1.2. Discord

Discord is our go-to app for team meetings. Since we are reaching the post-Covid era, it is still not easy to gather up physically as a group of 10 people. Discord makes it possible to meet as if we are sitting in a room together. It is even better because we can use the screen-sharing option to share our screens with other team members so everyone can work on the same thing. Discord also has chat

channels to communicate via text and share stuff with our team. It also helps us find sources we have used before because chat messages are not deleted and can be filtered easily. Since there can be more than one voice and text channel, we can also work concurrently. Overall, Discord is a great tool that helps us greatly in communicating.

4.1.3. Django

Django is a Python framework to develop web applications.At our team meetings we decided to Django because it makes developing web application fun and easy.

4.1.4. Zoom

We have used Zoom for our first two team meetings. It is a well-known app across our team, so it was easy to use at first. Unfortunately, it is pretty hard to archive chat messages, and we need a Premium account to save our meetings. Because of these reasons, even though we were familiar with the app, we decided to use Discord to plan our team meetings.

4.1.5. Slack

Slack is a communication platform that is primarily focused on texting. We use Slack to communicate with our professor and TA. It is an easy to learn and easy to use app. It also notifies every team member whenever one of our teachers sends us a message. It is a great app that helps us with communication

4.1.7. Google Docs

We have used Google Sheets when working on document-related issues as a group of members. We also have used it for the milestone report. It is a great app because it makes it possible to form a document with input from all of us. Everyone can see

every addition immediately and check on others' work. It is easy to use since we all had Google accounts before, and it is a great free app.

4.1.8. ProjectLibre

ProjectLibre is open-source software that we have used for Project planning. It is an excellent tool capable of creating a well-documented Project plan. Even though it is sufficient, working with ProjectLibre is a bit hard. Its UI is not very user friendly, and some of its tools are a bit complicated to use. It also does not let its users work on the Project concurrently, which is a negative feature. However, it is a useful app that helped us immensely when creating the Project plan.

4.2. Evaluation of Processes

4.2.1. Team Meetings

We have used team meetings to better understand topics discussed in class, decide on our short and long-term roadmap, distribute our workload until the next meeting, and check on each other. We have used Discord to gather up and used screen-sharing excessively. We also had a dedicated note-taker for each meeting so every meeting could be documented and can be reviewed later on. Team meetings are our most basic and essential process.

4.2.2. Issues

We have used issues to track our process on the project. Everyone used the issue system to document the job they had done. The issues include the job that needs to be done and the deadline and tags. It was an essential part of the

4.2.3 Branches and Pull Requests

When we are adding new codes to the practice app, instead of pushing directly into the master branch, we first create a new branch and do the development in that branch. After we finish the development in the related branches, we make pull requests to merge our branch with the master branch. Each pull request is reviewed by at least one team member other than those who made development at that branch.

4.2.4. PS Meetings

We had also gathered up as a group in PS hours. We had client meetings whenever our TA was available. At other times, we discussed the state of our project and tasks. These meetings were incredibly beneficial for our project since it is one of the most efficient times to ask our questions to our TA and get feedback from her.

5- Deliverables

5.1. Requirements

5.1.1. Functional Requirements

5.1.1.1 User Requirements

5.1.1.1.1 Registration

5.1.1.1.1.1 Guest users shall be able to sign up to the practice-app with a username and password.

5.1.1.1.1.2. Guest users should have access only to registration, login and API pages.

5.1.1.1.2. Login

5.1.1.1.2.1. Users shall be able to log in with their username and their password.

5.1.1.1.2.2. Users shall be able to log out.

5.1.1.1.3. Github Account Information

5.1.1.1.3.1. Users shall be able to get the number of public repositories of a github user from a github username input.

5.1.1.1.3.2. Users shall be able to get the number of followers of a github user from a github username input.

5.1.1.1.3.3. Users shall be able to get the email account of a github user from a github username input.

5.1.1.1.3.4. Users shall be able to get the name of a github user from a github username input.

5.1.1.1.4. IP Address Information

5.1.1.1.4.1. Users shall be able to get the country location of an IP address from an IP address input.

5.1.1.1.4.2. Users shall be able to get the country code of an IP address from an IP address input.

5.1.1.1.4.3. Users shall be able to get the last update date of the location of an IP address for an IP address input.

5.1.1.1.5. Random Funny Activities

5.1.1.1.5.1. Users shall be able to get random activity.

5.1.1.1.5.1. Users shall be able to get the type of random activity.

5.1.1.1.5.1. Users shall be able to get the number of participants of random activity.

5.1.1.1.5.1. Users shall be able to get the price of random activity.

5.1.1.1.5.1. Users shall be able to get the related link of random activity.

5.1.1.1.5.1. Users shall be able to get the accessibility of random activity.

5.1.1.1.6. Random Useless Facts

5.1.1.1.6.1. Users shall be able to get random useless fact.

5.1.1.1.6.1. Users shall be able to get useless fact of today.

5.1.1.1.7. Subject Summary Information

5.1.1.1.7.1 Users shall be able to get a wikipedia summary of any subject that has at most 100 characters and a wikipedia page.

5.1.1.1.8. Universities

5.1.1.1.9.1 Users shall be able to get names of the universities in the country specified by the user.

5.1.1.1.9.2 Users shall be able to get the website URL of the universities in the country specified by the user.

5.1.1.1.9. Race Standing

5.1.1.1.9.1 Users shall be able to get driver names of the race specified by year and round by the user.

5.1.1.1.9.2 Users shall be able to get the nationality of the drivers of the race specified by year and round by the user.

5.1.1.1.9.3 Users shall be able to get standing of the race specified by year and round by the user.

5.1.1.1.9.4 Users shall be able to get the circuit name where the race takes place.

5.1.1.1.10. Currency

5.1.1.1.10.1 Users shall be able to select any country in the world from the dropdown list.

5.1.1.1.10.2 Users shall be able to get the currency's code that is being used in the specified country.

5.1.1.1.10.3 Users shall be able to get the currency’s full name that is being used in the specified country.

5.1.1.1.10.4 Users shall be able to get the currency’s symbol that is being used in the specified country.

5.1.1.1.11. Education History

5.1.1.1.11.1 Users shall be able to save their education to the system by entering institute name, degree type (like BSc) and end year.

5.1.1.1.11.2 Users shall be able to get the institute name, degree type, end year of education records they saved to the system.

5.1.1.1.12. Player Informations

5.1.1.1.12.1 Users shall be able to get goal counts of the football players specified by name.

5.1.1.1.12.2 Users shall be able to get assist counts of the football players specified by name.

5.1.1.1.12.3 Users shall be able to get team information of the football players specified by name.

5.1.1.1.13. Add Event

5.1.1.1.13.1 Users shall be able to add event to the practice-app with an event form.

5.1.1.1.13.3

5.1.1.2. System Requirements

5.1.1.2.1. Verification

5.1.1.2.1.1. When a user wants to search information for a github account, s/he must provide a valid github username. The github username field must not be empty.

5.1.1.2.1.2. When a user wants to search information for an IP address, s/he must provide a valid IP address. The IP address field must not be empty.

5.1.1.2.2. Database

5.1.1.2.2.1. The platform shall include a user DB to keep users' profile information.

5.1.1.2.2.1. The platform shall include an event DB to keep events’ information.

5.1.1.2.2.1. The platform shall include a github users DB to keep github users' information.

5.1.1.2.2.1. The platform shall include an IP address DB to keep IP address’ information.

5.1.2. Non-Functional Requirements

.5.1.2.1. Security

5.1.2.1.1. User password data shall be kept safe from possible attacks via encryption algorithms such as SHA-256. All of these passwords should be encrypted in the database.

5.1.2.2. Portability

5.1.2.2.1. The program shall have an API for frontend applications to use.

5.1.2.3. Implementation

5.1.2.3.1. The Django framework shall be used for the backend side.

5.1.2.3.2. MySQL shall be used for the database.

5.1.2.3.3. HTML pages shall be rendered for the response pages.

5.1.2.3.4. Developed APIs shall have a JSON format.

Appendix

Appendix A: Glossary

1. practice-app: Event application.

2. Users: People who are already signed up to the practice-app and have an account.

3. Guest users: People who haven't signed up to the practice-app yet.

4. SHA-256: A widely used encryption algorithm which is very fast, reliable and secure compared to most other encryption algorithms.

5. Encryption: A change from a readable data to a masked data. This masked data must be decrypted to get the original readable data.