CmpE451 Fall 2019 Group3 - MILESTONE 1

1) Executive Summary

1.1) Introduction

Bulingo is a language learning platform. Since language is a tool for communication, Bulingo is also social. In Bulingo each user has a profile where other users can read about the users, also the comments done about them and their rating. Bulingo lets users practise with exercises for their levels, send writings to each other and evaluate the writings from different users to practice both reading and writing and provide a chat platform because we know that the best way to learn a language is to use it in the real life. Bulingo is also like a social platform where users can post comments about other users and rate them.

There are lots of language learning platforms in the market but in the most of them you are alone. Just complete the exercises and level up. We believe that this is not the right approach to learn a language. We believe the best way to learn a language properly is to communicate and Bulingo is a great app to learn a language with communication.

1.2) Work done so far

Firstly, we decided on our development platforms and languages. For the backend application, we decided to go with `NodeJS`, `ExpressJS`, `PostgreSQL` and `SequelizeJS`. For the web application of our product, we decided to use `ReactJS`, `HTML` and `CSS`. For the android application, we develop native android application with `Java`. Moreover, we make use of `nginx`, `Docker`, `Docker Compose` and `Travis` for infrastructural purposes. We deploy our applications to an `Amazon AWS EC2 t2.micro instance` which is provided as part of free-tier. We have configured `Continues Deployment` such that changes made on `production` branch will be deployed immediately. Our continues deployment configuration includes deployment of ngnix configurations, backend application, frontend application, and mobile application.

Up to this milestone we completed the login, sign-up, and log-out features of our web and mobile applications, and the language level determination feature to correctly identify how well the user knows a certain language. We also completed the profile page for both web and android application as the default page that the user will see when s/he sign-up or login, and after s/he completes the level determination test for a language.

1.3) Road ahead

We will continue to our implementations following our requirements to add more features to our application. We are planning to add features like sending an essay to another user, level specific reading, writing and listening exercises and a chatting feature for the users.

We are also planning to add a search feature, rating other users and commenting on their profile in the following weeks and finally we will be trying to make our application annotation-friendly.

2) List and status of deliverables

Name	Delivery Date	Delivered
Implementation of authentication (backend)	October 8,2019	DONE
Implementation of registration and login pages for web	October 8,2019	DONE
Implementation of registration and login pages for mobile	October 8,2019	DONE
Implementation of level exam (backend)	October 15,2019	DONE
Implementation of profile page for web	October 15,2019	DONE
Implementation of exam page for web	October 15,2019	DONE
Implementation of language selection screen for mobile	October 15,2019	DONE
Implementation of question screen for mobile	October 15,2019	DONE
Implementation of user profile (backend)	October 22,2019	DONE
Implementation of profile page for mobile	October 22,2019	DONE
Implementation of Logout for web	October 22,2019	DONE

3) Evaluation of the status of deliverables and its impact on your project plan

Up to this milestone we managed to deliver the sign-up login and logout properties that every application should have and some application specific features like our profile page and level determination exam. Our application is building up and we are happy to show some of our features like the profile page and the examination. We think that the skeleton of our application is ready so we can move forward in an agile plan where we can argue and decide to add a certain feature and do the implementation then review it before moving to the next feature.

3.1) Authentication Feature

According to our requirements there are certain things that a registered user can do but a guest user can not. To design our application according to this requirement we implemented authentication feature so after signing up a user can access all features a registered user can and if the user doesn't login or doesn't have an account, s/he only can access to the restiricted material.

3.2) Language Level Exam Feature

This feature implemented first because the first thing the user should do after signing up to start learning a language is to determine how well s/he knows the language. With this evaluation the user wont struggle for the exercises which are too hard for him/her or wont waste time on the exercises s/he already knows.

3.3) User Profile Feature

Each user has a profile page where they can see their language levels, the comments done about them, their profile picture and biography. Currently, users can also access language evaluation exam from profile page. In the future we are planning to update our profile page implementation as we add new features to our application.

The profile page implemented for this milestone is an uncomplete version of the profile page. Since we use the profile page as the main page in web application, user see it after logging in, we implemented a basic profile page that lets the user access our current features. In mobile application, when a user registers and logs in, he / she sees the user name and mail address on the profile page. Then the rating and short biography are added. After the language evaluating exam, the levels are written across the relevant language. The user can also see comments from other users on his / her profile page. As we add new features to our application we will update the profile page so the user can use this features easily.

4) Brief and clear summary of work done by each team member

Team Member	Contributions
Burak İlkay Akgün	We implemented signup, signin, profile and exam pages with Neval and Hilal as frontend team, and I implemented connections of login, register, profile and exam pages with backend(databases).Moreover, I converted exam and profile pages to dynamic form from static version, also I used cookies to save some data and provide logout mechanism. I prepared the project plan according the process of works done.
Samet Demir	I am a member of the backend team. I manage the integrity between teams. I am also the one helping other teams when they need help. As a member of backend team, up to now, I took the lead in the determination of all API endpoints, I implemented authentication API endpoints, developed tests for them, helped Sabri & Berkay to learn the tools and processes we used for backend development, and reviewed the code developed by them. I configured Nginx, Docker, Docker-Compose & Travis. These configurations include containerization of the whole code base with Docker, Continues Deployment with Travis, redirection configurations for the frontend, backend, mobile(.apk file) applications and a maintenance page with Nginx. I configured the Amazon AWS EC2 instance for deployment with Sabri. I helped the android team to handle the cookie(authentication token). I helped the frontend team about CORS(Cross Origin Resource Sharing) problem they face in their development environment. I occasionally helped the frontend team about the usage of ReactJS, React-Router and Axios.

Team Member	Contributions
Yağmur Kahyaoğlu	As the android team we implemented sign up, sign in, profile page and exam page with Ebru and Bartu. I implemented the backend connections for sign in, sign up, logout and user profile related endpoints. I implemented the backend connections for language lists, exam questions and exam evaluations with Ebru. I solved the cookie management problem with Samet. I worked on the styling of the user profile page and added a dynamic comments section. I also converted the language selection page into a dynamic one using RecyclerView. I implemented the first version of the exam result page. I also implemented the code to store the username of a logged in user at local storage to use later for API requests.
Bartu Ören	I was part of the Android team. I helped Ebru designing xml files for our signin signup and question pages. I also worked with Yağmur to establish connection between java and xml files. I mainly worked with layouts and button functionalities until this milestone.
Sabri Bayrakdar	I work in backend team.At first, I didn't know anything about deployment of a product.I worked for production and learned a lot from Samet.I signed up on https://aws.amazon.com web site.I created Amazon AWS EC2 instance and learned some configurations from Samet.I called amazon customer services when we got a problem about our server to solve the problem.I didn't know how to use PostgreSQL, SequelizeJS and Docker but I learned as soon as possible.I created most of database models (comment, examchoice, examquestion, level, user) and their seeder files.I searched for web sites to find realistic data for seeder files.For example I got questions for evaluation exam from YDS (Yabancı Dil Bilgisi Seviye Tespit Sınavı) and realistic user data from https://randomuser.me web site.I contributed most of API endpoints.Especially I can say that I wrote almost, most of the API endpoints except authentication API and improved the code with the help of code reviews from Samet.I contributed API docs when it is needed (few times).Lastly, I wrote unit tests for language and user APIs.
Ebru Zorlu	As android team, we implemented sign up, sign in, profile page and exam page with Yağmur and Bartu. I implemented API responses for languages and questions with Yağmur. Moreover, I have coded some warning messages and have taken care to make our screen designs compatible with all phone screens.
Neval Tüllük	We implemented signup, signin, profile and exam pages with İlkay and Hilal as frontend team, the sidebar navigation and sending answers to database logic with Hilal. I also implemented the router.
Hilal Mente	We implemented signup, signin, profile and exam pages with ilkay and neval as a frontend team, I implemented sidebar navigation and sending answers/showing grade with neval,additionally I created a new component which is popup to show grade, I completed error handling for signin page.
Berke Can Gürer	-
Berkay Özerbay	- I am in the backend team,so I learned the technologies s.t PostgreSQL,SequelizeJS and Docker.I created some database models (comment, level) and their seeder files, and tested their functionalaties in Postman.Additionally, I was one of the member who presented the milestone and prepared its use-case story.

5) Requirements

1. Functional Requirements

1.1 User Requirements

1.1.1 Registration and Login

1.1.1.1 Registration

A guest user shall be able register with their e-mail, pick a unique username and password or continue as a guest user.

1.1.1.2 Login

- **1.1.1.2.1** Registered users log in by entering their username/e-mail and password.

1.1.1.3 Guest User

- **1.1.1.3.1** A guest user shall be able to access limited material, in order to get whole material, he/she must register first.

1.1.2 Multi-Language Selection

A registered user shall be able to choose one or more languages in order to learn or provide expert skill or knowledge.

1.1.3 Access to Materials and Exercises

1.1.3.1 Registered User

- **1.1.3.1.1** A registered user shall be able to access the listening materials according to his/her level.
- **1.1.3.1.2** A registered user shall be able to access the exercises related to the listening materials.
- **1.1.3.1.3** A registered user shall be able to access the reading materials according to his/her level.
- **1.1.3.1.4** A registered user shall be able to access the exercises related to the reading materials.
- **1.1.3.1.5** A registered user shall be able to access the grammar topics according to his/her level
- **1.1.3.1.6** A registered user shall be able to access the exercises related to the grammar materials.
- **1.1.3.1.7** A registered user shall be able to access the vocabulary materials according to his/her level
- **1.1.3.1.8** A registered user shall be able to access the exercises related to the vocabulary materials.
- **1.1.3.1.9** A registered user shall be able to access the writing materials.
- **1.1.3.1.10** A registered user shall be able to access the exercises related to the writing materials.
- **1.1.3.1.11** A registered user shall be able to upload handwritten writing exercises.
- **1.1.3.1.12** A registered user shall be able to accept providing writing consultancy to other users.

1.1.3.2 Guest User

- **1.1.3.2.1** A guest user shall be able to access to a directory of listening materials categorized by language level.
- **1.1.3.2.2** A guest user shall be able to access to a directory of reading materials categorized by language level.
- **1.1.3.2.3** A guest user shall be able to access to a directory of grammar materials categorized by language level.
- **1.1.3.2.4** A guest user shall be able to access to a directory of vocabulary materials categorized by language level.

1.1.4 Customized Access to Materials and Exercises

- **1.1.4.1** A registered user shall be able reach the materials and exercises that are for their current level.
- **1.1.4.2** A registered user shall be able to reach materials and exercises of the lower levels.

1.1.5 Consultancy for Writing Exercises

- **1.1.5.1** A registered user shall be able to choose a user from the recommended list of users or by searching a user by his/her username in order to consult.
- **1.1.5.2** A registered user shall be able to send his/her writing exercises to other users who accept to provide writing consultancy in order to get feedback.

1.1.6 User Profiles

1.1.6.1 Username

Username of a registered user shall be able to seen from the profile of the user.

1.1.6.2 Progress in the languages

Registered users shall be able to see the progress of themselves or other users in the languages in terms of the percentage of correct answers, and the current level.

1.1.6.3 Rating

Registered user's rating in terms of average response time, appreciation by registered users shall be able to seen other users.

1.1.6.4 Avatar

- **1.1.6.4.1** A registered user should have an avatar.
- **1.1.6.4.2** Registered users should be able to change their own avatars.

1.1.6.5 Bio

Registered users should be able to add the information about their biography.

1.1.6.6 Guest User

A guest user should have no profile.

1.1.7 Annotation

Registered users shall be able to annotate texts and images within the system.

1.1.8 New Materials

- **1.1.8.1** Registered users in the corresponding language shall be able to upload new learning materials and exercises.
- **1.1.8.2** Newly-added materials shall be evaluated by the admin users.
- **1.1.8.3** Materials which are approved by any admin user shall be integrated into the system.

1.1.9 Messaging

- **1.1.9.1** Registered users shall be able to send a chat request to other users in the corresponding language that user want to learn.
- **1.1.9.2** Registered users shall be able to accept the chat request received from other registered users.
- **1.1.9.3** Registered users shall be able to send text messages to other registered users.
- **1.1.9.4** Registered users shall be able to receive text mes sages from other registered users.

1.1.10 Search

- **1.1.10.1** All users shall be able to do a basic search by using keywords.
- **1.1.10.2** Registered users shall be able to do an advanced search to filter the content by topic, difficulty, scope, and tag.

1.1.11 Report Harrasment

- **1.1.11.1** Registered users should be able to report disturbing behavioral acts.
- **1.1.11.2** All users should be able to report materials, if they contain racist insultions or inappropriate contents.

1.1.12 Commenting

Registered users should be able to send a comment which will be appended to the commented user’s profile page, to any user.

1.2 System Requirements

1.2.1 Level Determination

1.2.1.1 Level Determination for New Users

The system shall provide a test to new users to determine their level of knowledge about language.

1.2.1.2 Level Determination for Users

The system shall provide a test to registered users after completing a certain amount of exercises or any arbitrary time that is chosen by the user to determine their new level of knowledge about language.

1.2.2 Automated Grading

- **1.2.2.1** The system shall automatically grade exercises except for writing exercises
- **1.2.2.2** The system shall highlight the correct answer if the provided one by the user is wrong.

1.2.3 User Recommendation

The system shall recommend other users to users to consult for writing according to their proficiency level in the corresponding language.

1.2.4 Progress Statistics

The system shall provide statistics about completed/uncompleted exercises, achievement.

1.2.5 Accepting Consultancy

The system shall provide an option for users to accept if they are willing to provide writing consultancy for other users.

1.2.6 Semantic Searching

The system shall provide a semantic searching mechanism.

1.2.7 Admin

- **1.2.7.1** The system shall have some admins who are responsible of reports and new materials.

- **1.2.7.2** An admin shall evaluate the report that come from the users.
- **1.2.7.3** An admin shall evaluate the new materials that are added by users and shall accept or reject the material to be entegrated to the system.

2. Nonfunctional Requirements

2.1 Availability Requirements

The platform and the materials should be accessible at any time, from anywhere.

2.2 Portability Requirements

2.2.1 Web application

- **2.2.1.1** The Web application shall support Google Chrome 48 and later.
- **2.2.1.2** The Web application should support Google Chrome 32 and later.

2.2.2 Android

- **2.2.2.1** The Android Application shall support Android 6: Marshmallow and later in order to support modern devices.
- **2.2.2.2** The Android Application should support Android 4.4: KitKat and later in order to support older devices.

2.2.3 API

- **2.2.3.1** The API Application shall be deployable on a remote and manually configurable remote server.
- **2.2.3.2** The API Application should be deployed to Amazon EC2 or DigitalOcean remote server.

2.3 Performance Requirements

The system should respond to any request in 5 seconds at most.

2.4 Security Requirements

2.4.1 Encryption

The system should encrypt the traffic by using HTTPS.

2.4.2 Password

The system should force the users to pick at least six characters long password.

2.5 Privacy Requirements

The personal information, contact information, copyrighted contents, license issues and everything related to these paradigms shall be respected and considered.

2.6 Standards

- **2.6.1** W3C Web Annotation Data Model shall be used for Annotation.
- **2.6.2** The implementation of this system shall follow the standards introduced by the World Wide Web Consortium (W3C).

6) API documentation

Note: API doc is also available online on http://18.184.207.248/api/docs/. However, online version includes some work in progress elements. The version given here provides complete documentation of working features as part of Milestone 1 and excludes the work in progress.

bounswe2019group3-backend v1.0.0

backend project

- auth
  - login
  - logout
  - signup
- language
  - returns available languages
  - evaluates level determination exam
  - returns level determination exam questions
- user
  - returns all users
  - returns user details
  - returns user comments
  - returns user language level details

auth

login

Back to top

```
POST /api/auth/login
```

Request body(JSON) Parameters

Name	Type	Description
id	`String`	username or email of the user.
password	`String`	password of the user.

Success 200

Name	Type	Description
username	`String`	
email	`String`	

logout

Back to top

```
POST /api/auth/logout
```

Success 204

signup

Back to top

```
POST /api/auth/signup
```

Request body(JSON) Parameters

Name	Type	Description
username	`String`	
email	`String`	
password	`String`	

Success 204

language

returns available languages

Back to top

```
GET /api/language/
```

Success 200

Name	Type	Description
language	`Object[]`	list of languages
language.name	`String`	language name
language.abbr	`String`	language abbreviation

evaluates level determination exam

Back to top

```
POST /api/language/:language_abbr/exam/evaluate
```

Request body(JSON) Parameters

Name	Type	Description
answers	`Object[]`	list of answers
answers.question_id	`Integer`	question id
answers.choices_id	`Integer`	choices id

Success 200

Name	Type	Description
grade	`String`	result of evaluation

returns level determination exam questions

Back to top

```
GET /api/language/:language_abbr/exam/questions
```

Success 200

Name	Type	Description
questions	`Object[]`	list of exam questions
questions.id	`Integer`	question id
questions.desc	`String`	question description
questions.choices	`Object[]`	answer choices
questions.choices.id	`Integer`	answer choices id
questions.choices.desc	`String`	answer choices description

user

returns all users

Back to top

```
GET /api/user/
```

Success 200

Name	Type	Description
user	`Object`	user object
user.username	`String`	username

returns user details

Back to top

```
GET /api/user/:username
```

Success 200

Name	Type	Description
user	`Object`	user object
user.username	`String`	username
user.email	`String`	email
user.bio	`String`	biography text
user.avatar	`String`	avatar url
user.rating	`Float`	rating from comments

returns user comments

Back to top

```
GET /api/user/:username/comments
```

Success 200

Name	Type	Description
comments	`Object[]`	comments
comments.text	`String`	comment text

Name	Type	Description
comments.rating	`String`	comment rating
comments.comment_by	`String`	author of comment
comments.comment_to	`String`	target of comment
comments.createdAt	`String`	creation time of comment

returns user language level details

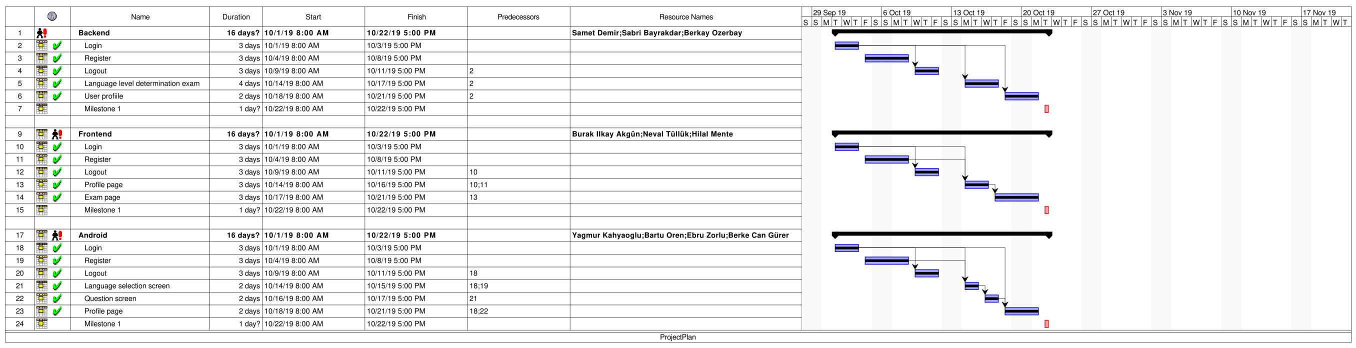
Back to top

```
GET /api/user/:username/language/level
```

Success 200

Name	Type	Description
language_levels	`Object[]`	language levels
language_levels.lang_abbr	`String`	language abbr
language_levels.grade	`String`	language level

7) Project plan



		Name	Duration	Start	Finish	Predecessors	Resource Names
1		**Backend**	**16 days?**	**10/1/19 8:00 AM**	**10/22/19 5:00 PM**		**Samet Demir;Sabri Bayrakdar;Berkay Ozerbay**
2	✓	Login	3 days	10/1/19 8:00 AM	10/3/19 5:00 PM		
3	✓	Register	3 days	10/4/19 8:00 AM	10/8/19 5:00 PM		
4	✓	Logout	3 days	10/9/19 8:00 AM	10/11/19 5:00 PM	2	
5	✓	Language level determination exam	4 days	10/14/19 8:00 AM	10/17/19 5:00 PM	2	
6	✓	User profiile	2 days	10/18/19 8:00 AM	10/21/19 5:00 PM	2	
7		Milestone 1	1 day?	10/22/19 8:00 AM	10/22/19 5:00 PM		
9		**Frontend**	**16 days?**	**10/1/19 8:00 AM**	**10/22/19 5:00 PM**		**Burak Ilkay Akgün;Neval Tüllük;Hilal Mente**
10	✓	Login	3 days	10/1/19 8:00 AM	10/3/19 5:00 PM		
11	✓	Register	3 days	10/4/19 8:00 AM	10/8/19 5:00 PM		
12	✓	Logout	3 days	10/9/19 8:00 AM	10/11/19 5:00 PM	10	
13	✓	Profile page	3 days	10/14/19 8:00 AM	10/16/19 5:00 PM	10;11	
14	✓	Exam page	3 days	10/17/19 8:00 AM	10/21/19 5:00 PM	13	
15		Milestone 1	1 day?	10/22/19 8:00 AM	10/22/19 5:00 PM		
17		**Android**	**16 days?**	**10/1/19 8:00 AM**	**10/22/19 5:00 PM**		**Yagmur Kahyaoglu;Bartu Oren;Ebru Zorlu;Berke Can Gürer**
18	✓	Login	3 days	10/1/19 8:00 AM	10/3/19 5:00 PM		
19	✓	Register	3 days	10/4/19 8:00 AM	10/8/19 5:00 PM		
20	✓	Logout	3 days	10/9/19 8:00 AM	10/11/19 5:00 PM	18	
21	✓	Language selection screen	2 days	10/14/19 8:00 AM	10/15/19 5:00 PM	18;19	
22	✓	Question screen	2 days	10/16/19 8:00 AM	10/17/19 5:00 PM	21	
23	✓	Profile page	2 days	10/18/19 8:00 AM	10/21/19 5:00 PM	18;22	
24		Milestone 1	1 day?	10/22/19 8:00 AM	10/22/19 5:00 PM		

8) User scenarios you presented during the milestone presentation

Persona 1

Elza is a Brazillian newly graduated architect who recently started her new job in Berlin, Germany. She uses her advanced English skills for communicating at work. She also wants to improve her German skills to use in her daily life in Germany to communicate with the locals.

Persona 2

Hans is a German architect living in Berlin, Germany. He is a colleague of Elza. Since he is working at an international company, he uses English at work but he is not very confident with his English skills and he wants to test his English knowledge.

Scenario 1

Elza has been using Bulingo web and mobile application for a while to test her English skills. She also wants to learn her proficiency level in German language hence she opens the app from her mobile phone. Elza enters her username and password and clicks the "Sing in" button in the appearing Sign In page to be directed to the Home page. At the Home page to take a German proficiency test she clicks the "Language Selection" button and selects "German" in the appearing page. After selecting her language she takes the test by clicking the answers she chose for every question and finally sees her calculated German level and her percentage of success. She click "Go Back" to go ack to the Home page. To check her profile after taking the test she clicks "Profile". Elza sees her German level on her profile.

Scenario 2

While passing by, Hans sees Elza taking something like a quiz on her phone. He wonders what the quiz is about and asks her about it and learns about the app Bulingo. Since he has been wanting to test his English level, he find the application very interesting and asks Elza if he can try it. Elza is at her Home page at the moment so she clicks the "Log Out" button before giving her phone to Hans. To register as a new user Hans clicks "Sign Up" button to be redirected to the Sign Up page. He enters his new username, his email and his password. He has to enter the password twice, the second one for the conformation but he enters a different password to the second password field. Hence after clicking "Sign Up", he gets a warning. Then, he reenters the second password and clicks "Sign Up" again. In the Home page he clicks "Language Selection" button and selects "English" for his language. After the exam starts while at the first question he realizes that the exam will take time and decides to take it later. He clicks the back button on the phone and clicks "OK" for the warning telling his progress will be lost. He logs out clicking "Logout" and returns the phone back. Elza tells him that Bulingo also has a web application. After going his home, Hans decides to take the test on his computer. He opens the web page of Bulingo and enters his credentials to sign in. He clicks "Login" and sees his profile page. He clicks "Exams" and selects "English" from the opening page after reading explanations about the proficiency exam. He takes the exam selecting the answers for the questions and clicks "Submit" button to finish the process. He sees his English level at the top left corner of the screen. Hans clicks the "Close" button to be redirected to his profile page and he also sees his language levels on his profile.

9) The code structure and group process

the code structure:

- /project
  - /project/backend includes backend application
  - /project/frontend includes frontend application
  - /project/mobile includes mobile application
  - /project/nginx includes nginx configurations
  - /project/docker-compose.yml includes docker-compose configurations for production environment
  - /project/docker-compose.dev.yml includes docker-compose configurations for development environment
  - /project/docker-compose.test.yml includes docker-compose configurations for test environment
- /.travis.yml includes travis configurations

the branches

- backend-dev is used for backend development
- frontend is used for frontend development
- mobile is used for mobile development
- master is used to keep the code together
- production is used for continues (auto) deployment

the workflow

- a) Whole group discusses what functionalities are needed to provide the next feature
- b) Backend team determines the API endpoints related to the functionalities and provides API documentation for them
- c) Each team implements their side with respect to the specified API endpoints on their own branch
- d) Each team merges its code to master branch regularly
- e) Master branch is merged to the production branch in order to deploy the feature

teams

- backend: Samet, Sabri, Berkay
- frontend: İlkay, Neval, Hilal
- mobile: Yağmur, Bartu, Ebru, Berke

10) Evaluation of tools and managing the project

Tools

These are the tools and services used by our team which are not essentially required.

Sequelize

This is an Object Relational Mapping(ORL) solution that the backend team uses in order to avoid writing SQL queries directly. It helped us write less, do more. It is good to abstract away the database details with an ORL solution.

Docker / Docker-Compose

These are used to containerize the code so that we do not need to deal with environment-specific situations such as version management and installation of tools for each machine. It helps us focus more on development and avoid configuration problems such as database connection configurations.

Travis

We currently use Travis for Continues Deployment purposes. It helps save time when the code needs to deployed by automating the deployment. We almost never need to ssh into the server or do anything on the server excluding some special problems such as the disc space problem that we have recently experienced.

Deployment: Amazon AWS EC2 Instance

An interesting problem that we have recently experienced was kind of related to the overuse of the free-tier Amazon AWS EC2 instance. We were trying to compile the android application inside a docker container on the server for each time the production branch is changed. The purpose is to generate the latest android application apk so that it could be accessible online. The problem was that the android compilation process takes much of computational resources for a few minutes and the EC2 instance that we are using was only allowed to give %10 of the CPU to our usage. This means that if we operate over %10 for enough time Amazon system blocks our CPU usage and our server becomes unresponsive. It took some time to understand the issue. We resolved the problem by limiting the computational resources of the docker container. Now, the android application apk is compiled and updated approximately 5-8 hours for each time there is a change in the production branch. While we were working on this, we had to restart the instance a few time which result in IP address change and we had to change the IP address configurations on our applications. The takeaway is to know the limit of the available resources before you attempt to do something unusual. Another takeaway is to configure static IP address for server before anything.

Challenges you met as a group

Even though we work properly as a group, sometimes we have a lack of communication.And these tiny conflict may cause some problems.In our case, we have a misunderstanding for describing id of question choices to check whether the answer is true or false.
Some of us think that:
"The id of choices should be unique (independent from question) and has a autoincrement property"
On the other hand the others think like that:
"The id of choices should be a representation for each question as 0,1,2,3 for A,B,C,D."
Result of this conflict was a waste of time.Because some members of group who send request to server got wrong evaluation grade.After all, it didn't take so long, as soon as we find out that situation, we synchronize the server with mobile and web apps.

What kind of changes are planned

We don't want to experience the problem which is mentioned above.Therefore we need to take a precaution against communication problems.So we decide to share an example use of completed api endpoints to mobile and web groups.