CmpE451 Fall 2019 Group3 - MILESTONE 2

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

In our first milestone we had 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 had completed the outlines of the profile page for both web and android application.

For this milestone, firstly, we fixed our problems from the first milestone. We improved the UI in web and android. We also added the meanings of level ratings for users to understand. Then, moving forward, we finalized our work with profile page. Now, users can comment -and rate- on other people's profile pages on both web and Android app. Profile pages now can be edited in both web and mobile in the user's own profile page, which includes uploading new avatars and bio texts to database. We added user search functionality to find other users for chatting or viewing/commenting their page. In addition to these, we implemented chat functionality where you can view your chat history and message other users for both web and android. We implemented outline of the exercise page -with static content- on mobile. Additionally, exercise search and other exercise related endpoints are already available on the backend.

1.3) Road ahead

We will continue to our implementations following our requirements to add more features to our application. Our first concern right now is adding exercise functionality to web and mobile, which is the base of a learning platform. We plan to add level specific reading, writing, listening, grammar, vocabulary exercises and evaluate them to be effective on user's level.

We will also implement searching mechanism for exercises to find the desired content, filtered by type, language and level. We also plan to start working on the annotation feature after the exercises are done in both web and mobile.

2) List and status of deliverables

Note: See project plan for more.

Name	Delivery Date	Delivered
(Backend) Search API: User Search	23.10.19	DONE
(Backend)Search API: Exercise Search	23.10.19	DONE
(Backend) Updating User Profile	01.11.19	DONE
(Backend) Creating User Comment	07.11.19	DONE
(Backend) Exercise API: Get language level for choosen exercise type	15.11.19	DONE
(Backend) Exercise API: Get questions for an exercise	15.11.19	DONE
(Backend) Exercise API: Post answers	15.11.19	DONE
(Frontend) Writing	28.10.19	DONE
(Frontend) Navbar	01.11.19	DONE
(Frontend) Chat	08.11.19	DONE

Name	Delivery Date	Delivered
(Frontend) Setting: Change Bio	12.11.19	DONE
(Frontend) Setting: Change Avatar	12.11.19	DONE
(Frontend) Profile Page: Comment	12.11.19	DONE
(Frontend) Search: User Search	19.11.19	DONE
(Frontend) Search: Exercise Search	19.11.19	DONE
(Mobile) Chat: Chat screen	01.11.19	DONE
(Mobile) Chat: Get messages	31.10.19	DONE
(Mobile) Chat: Send messages	06.11.19	DONE
(Mobile) Chat: Chat history screen	07.11.19	DONE
(Mobile) Search: Design search screen	14.11.19	DONE
(Mobile) Search: Implement user seach	19.11.19	DONE
(Mobile) Comment: Send comment to other user	21.11.19	DONE
(Mobile) Profile Update: Designing profile page to adding setting button	22.11.19	DONE
(Mobile) Profile Update: Implementing permission request activity from device	26.11.19	DONE
(Mobile) Profile Update: Change bio	22.11.19	DONE
(Mobile) Profile Update: Change avatar	25.11.19	DONE
(Mobile) Exercise: Designing Exercise selection screen	25.11.19	DONE

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

Up to this milestone we managed to deliver the search, chat, user profile update and commenting properties. Our application is building up and we are happy to show some of our features like the chat and the search. We think that the most 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) Search

According to our requirements the users should be able to search other users and also search exercises semantically according to their levels, languages, types and contents. For this milestone we fully implemented the user search part for the users to be able to search for other users for them to be able to add comments or send messages later on. The exercise search part is only implemented by our backend team and it remains to be implemented by our frontend and mobile teams so that the users will be able to search and filter exercises rather than only selecting them from a list.

3.2) Chat

According to our requirements, the users can chat with each other. For this milestone, we implemented chat feature. User can chat with another user. When user sends messages to multiple users, the message box shows all users that have been chatted before. Users can send message from User profile.

3.3) User Profile Update

Each user has a profile page where they can see their language levels, the comments done about them, their profile picture and biography. Users can update biography and profile picture. Also ,users has the accept writing / do not accept writing selection. It has been designed statically, we may change our decision about implementing selection part.

3.4) Comment

In the application according to the requirements users can chat with each other and also can review writing exercises of each other. After these processes we added a comment section in users' profiles with the comments that the user has received and and a new comment part for users to review their experiences so that when users are looking for another user to chat or send a writing they can check the comments written for that user.

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

Team Member	Contributions
Burak İlkay Akgün	I am working in frontend with Neval and Hilal.After milestone 1, I changed profile page's components that are comments and informations parts, from static to dynamic. I designed and implemented navbar component to our site's all page and also reorganised logout mechanism.I implemented message page with Neval and I implemented message sending functionalities to another user.I provided the live messaging by sending request to our db each 2 second .I made some arrangement on time showing, I used moment js format to show time more understandable. I implemented search functionalities to the search page with Hilal and I implemented search result page that show the result of search and provide some functionalities to user like going to searched user's profile.I implement the send comment to the other user functionality in the other user profile page.Finally, I edited the project plan according to the feedback.
Samet Demir	I am a member of the Backend team. Also, I am responsible for the communication between the backend and other teams (mobile & frontend). After Milestone 1, I first designed the Chat API that includes all the messaging functionality. I implemented one of the Chat endpoints (general chat history). Then, I designed the Exercise API that includes exercise-related functionalities, the Search API, the user profile update endpoint, and create comment endpoint. I implemented the Search API including the exercise search and user search. Moreover, I implemented the user profile update endpoint, which includes file uploading. In the meantime, I created issues for the backend tasks, followed how we, members of the backend team, are doing about tasks and reviewed the work from other members of the backend team.
Yağmur Kahyaoğlu	As the android team we implemented update profile, user search, chat history, chat, exercise selection and adding comment features with Ebru and Bartu. I implemented the backend connections and worked on the styling for search and exercise selection related endpoints. We implemented the backend connections and styling for chat history, chat and adding comments with Ebru and Bartu. I also added a swipe to refresh feature to the chat history page. I updated the requirements according to the feedback after the first milestone with Berkay.
Bartu Ören	I am working in the Android sub-team. For this milestone, I worked on profile pages’ layouts for adding comments, profile page edit screen layout, uploading image from your gallery with Glide. I also worked on Java part to implement profile page button functionalities and sending new profile images to database with Yağmur and Ebru. I also worked on chat feature, for both layout and java part of the job. We used RecylerView for both Chat and Chat History pages and implemented database connections for chat with Yağmur and Ebru.
Sabri Bayrakdar	I am a member of backend team.For the second milestone, I implemented exercise APIs. I created exercise, exercisechoice, exercisequestion and comment models.I created initial exercise seeder files which are completed by Berkay.I created exercise API documentation.I got help from Samet at some points, because I had technical problems about my personal computer.
Ebru Zorlu	I am working in the Android team. For this milestone, I worked on new functionalities of profile page such as sending comment to a user, editing avatar and bio with Yağmur and Bartu. We implement user search functionality. We also worked on chat feature by implementing chat and chat history pages and database connections.I also worked on permission requests that asks the user to allow access to the phone while chosing photo on the profile update page.
Neval Tüllük	I am working in frontend team with Hilal and İlkay. After Milestone 1, I changed profile page's components that are comments and informations parts and design of exam page according to Milestone feedbacks. I worked on send message, receive message part with Ilkay on messsage page, also I designed whole page. Additionally, I implemented User profile page with send message and send comment functionalities. I designed the table after search on Search page.
Berke Can Gürer	

Team Member	Contributions
Berkay Özerbay	I am working in backend team. For this milestone, I implemented the endpoint for creating user comments.I and Yağmur also edited the requirements page according to Cihat's review. Moreover, I also created questions their choices and answers in the exercise seeds(demo-exercise-*.js files).
Hilal Mente	I am working in frontend team. After Milestone 1, I helped Neval while designing exam page according to Instructor's feedback. For Milestone 2, I implemented Search Page with İlkay. Additionally, I implemented Settings page.

5) Requirements

1. Functional Requirements

1.1 User Requirements

1.1.1 Registration and Login

1.1.1.1 Registration

- **1.1.1.1.1** A guest user shall be able register with their e-mail, 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(available languages:English,German)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 do writing exercises by typing in the application.
- **1.1.3.1.13** 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 all listening exercises of a selected language categorized by language level.
- **1.1.3.2.2** A guest user shall be able to access to a directory of all reading exercises of a selected language categorized by language level.
- **1.1.3.2.3** A guest user shall be able to access to a directory of all grammar exercises of a selected language categorized by language level.
- **1.1.3.2.4** A guest user shall be able to access to a directory of all vocabulary exercises of a selected language 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 shall have an avatar.
- **1.1.6.4.2** Registered users shall 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 registered 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 Harresment

- **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 insults or inappropriate contents.

1.1.12 Commenting

Registered users should be able to make a comment to any user .It will be appended to the commented user’s profile

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 all the exercises for their current proficiency level 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 delete the racist or insultive contents that was reported by the users.
- **1.2.7.4** An admin shall evaluate the new materials that are added by users and shall accept or reject the material to be integrated 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/.

bounswe2019group3-backend v1.0.0

backend project

- auth
  - login
  - logout
  - signup
- chat
  - create new message for username
  - general chat history
  - chat history with username
- language
  - return all exercise of type
  - return the exercise
  - evaluate the exercise
  - returns available languages
  - evaluates level determination exam
  - returns level determination exam questions
- search
  - search
- user
  - create comment for username
  - returns all users
  - returns user 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.

logout

Back to top

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

signup

Back to top

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

Request body(JSON) Parameters

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

chat

create new message for username

Back to top

```
POST /api/chat/:username
```

Request body(JSON) Parameters

Name	Type	Description
body	`Object`	

Name	Type	Description
body.message	`String`	message text

Success Response

Success-Response:

```
HTTP/1.1 204 OK
```

general chat history

Back to top

```
GET /api/chat/
```

Success Response

Success-Response:

```
HTTP/1.1 200 OK
{
  "nb_new_messages": 3,
  "history": [
     {
         "username": "user",
         "last_message": "hello world",
         "nb_new_messages": 1,
         "last_message_date": "2013-10-21T13:28:06.419Z"
     },
     {
         "username": "admin",
         "last_message": "welcome to bulingo",
         "nb_new_messages": 2,
         "last_message_date": "2013-10-20T11:10:04.222Z"
     }
  ]
}
```

Success 200

Name	Type	Description
chat	`Object`	chat
chat.nb_new_messages	`Integer`	number of new messages (all)
chat.history	`Object[]`	chat history with all users (ordered by date)
chat.history.username	`String`	username
chat.history.last_message	`String`	last message
chat.history.nb_new_messages	`Integer`	number of new messages from that user
chat.history.last_message_date	`String`	date of the last message

chat history with username

Back to top

```
GET /api/chat/:username
```

URL Parameters

Name	Type	Description
username	`String`	opponent user's username
skip	`String`	number of messages to skip
limit	`String`	number of messages to return

Success 200

Name	Type	Description
messages	`Object[]`	list of messages
messages.to_username	`String`	message receiver
messages.from_username	`String`	message sender
messages.message	`String`	message text
messages.new	`Boolean`	message is read boolean

language

return all exercise of type

Back to top

```
GET /api/language/:language_abbr/exercise
```

Success 200

Name	Type	Description
exercise	`Object[]`	exercises
exercise.exersice_id	`Integer`	exercise id
exercise.title	`String`	exercise title
exercise.language_abbr	`String`	exercise language abbreviation
exercise.exercise_type	`Stirng`	exercise exercise type
exercise.level	`String`	exercise level

Name	Type	Description
exercise.tags	`String`	exercise tags

return the exercise

Back to top

```
GET /api/language/:language_abbr/exercise/:exersice_id/questions
```

Request body(JSON)

Name	Type	Description
question_id	`String`	question id
desc	`String`	question description
media_url	`String`	media url related to question (optional)
media_type	`String`	media type related to question (optional)
media_start_time	`String`	media start time related to question (optional)
media_end_time	`String`	media end time related to question (optional)
choices	`Object[]`	answer choices (optional: not available for writing)
choices.id	`String`	choice id
choices.desc	`String`	choice description

evaluate the exercise

Back to top

```
POST /api/language/:language_abbr/exercise/:exersice_id/evaluate
```

Parameter Parameters

Name	Type	Description
answers	`Object[]`	
answers.question_id	`Integer`	answer question id
answers.choice_id	`Integer`	answer choice id
answers.text	`String`	(optional: only available for writing)

Success 200

Name	Type	Description
nb_correct_answers	`Integer`	number of correct answers

Name	Type	Description
nb_questions	`Integer`	number of questions
answers	`Object[]`	media related to question (e.g. listening material) (optional)
answers.question_id	`Integer`	question id
answers.choice_id	`Integer`	correct choices id

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

search

search

Back to top

```
GET /api/search/
```

URL Parameters

Name	Type	Description
text	`String`	text to be searched
type	`String`	what type of data to be searched (exercise/user)
lang_abbr	`String`	lang_abbr (exercise)
level	`String`	level (exercise)
exercise_type	`String`	exercise_type (exercise)

Success 200

Name	Type	Description
result	`Object[]`	result array
result.type	`String`	type of data
result.username	`String`	username of the user (optional: only for user type)
result.exersice_id	`Integer`	id of the exercise (optional: only for exercise)
result.title	`String`	title of the exercise (optional: only for exercise)
result.lang_abbr	`String`	language of the exercise (optional: only for exercise)

Name	Type	Description
result.exercises_type	`String`	type of the exercise (optional: only for exercise)
result.level	`Integer`	level of the exercise (optional: only for exercise)
result.tags	`String`	tags of the exercise (optional: only for exercise)

user

create comment for username

Back to top

```
POST /api/user/:username/comments/
```

Request body(JSON) Parameters

Name	Type	Description
comment	`Object`	
comment.text	`String`	text
comment.rating	`Integer`	rating (1,2,3,4,5)

Success Response

Success-Response:

```
HTTP/1.1 204 OK
```

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

update user details

Back to top

```
POST /api/user/:username
```

Request body(JSON) Parameters

Name	Type	Description
user	`Object`	user object
user.bio	`String`	biography text
user.avatar	`File`	avatar image file

returns user comments

Back to top

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

Success 200

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

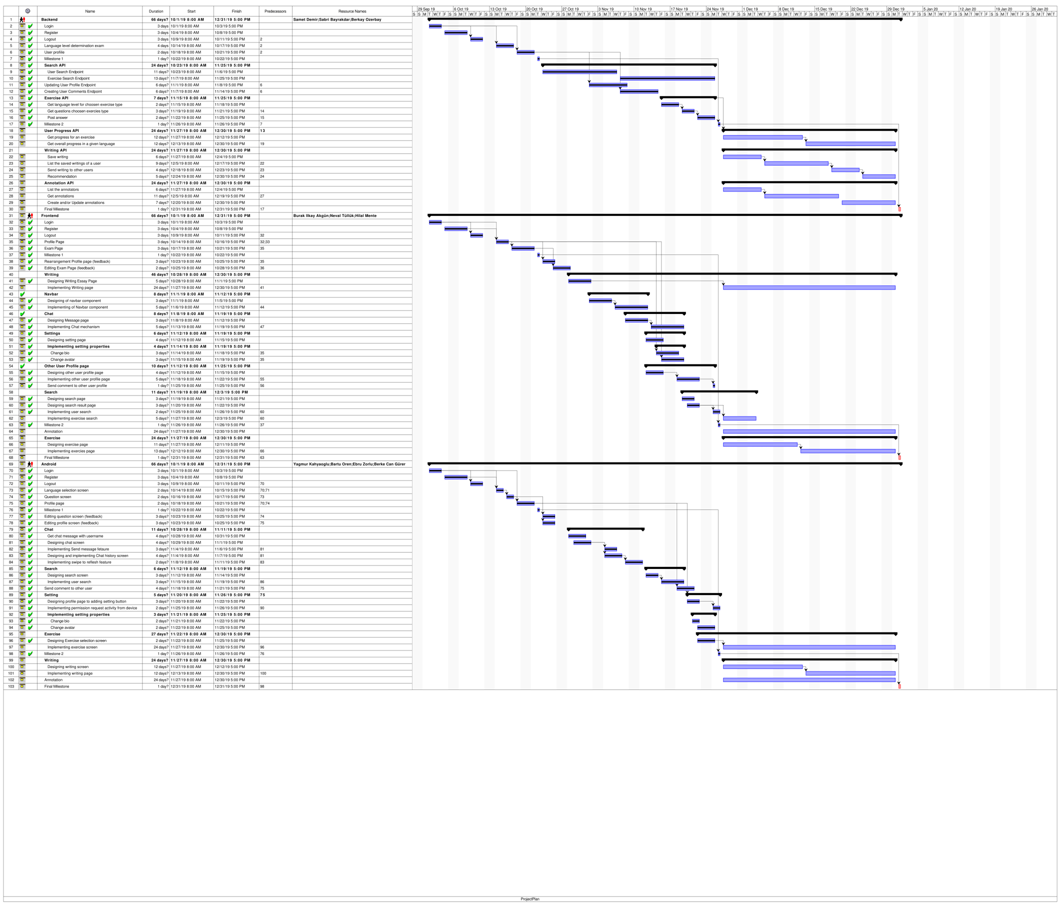
Back to top

```
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**	**66 days?**	**10/1/19 8:00 AM**	**12/31/19 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 profile	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		
8	✓	**Search API**	**24 days?**	**10/23/19 8:00 AM**	**11/25/19 5:00 PM**		
9	✓	User Search Endpoint	11 days?	10/23/19 8:00 AM	11/6/19 5:00 PM		
10	✓	Exercise Search Endpoint	13 days?	11/7/19 8:00 AM	11/25/19 5:00 PM		
11	✓	Updating User Profile Endpoint	6 days?	11/1/19 8:00 AM	11/8/19 5:00 PM	6	
12	✓	Creating User Comments Endpoint	6 days?	11/7/19 8:00 AM	11/14/19 5:00 PM	6	
13	✓	**Exercise API**	**7 days?**	**11/15/19 8:00 AM**	**11/25/19 5:00 PM**		
14	✓	Get language level for choosen exercise type	2 days?	11/15/19 8:00 AM	11/18/19 5:00 PM		
15	✓	Get questions choosen exercies type	3 days?	11/19/19 8:00 AM	11/21/19 5:00 PM	14	
16	✓	Post answer	2 days?	11/22/19 8:00 AM	11/25/19 5:00 PM	15	
17	✓	Milestone 2	1 day?	11/26/19 8:00 AM	11/26/19 5:00 PM	7	
18		**User Progress API**	**24 days?**	**11/27/19 8:00 AM**	**12/30/19 5:00 PM**	**13**	
19		Get progress for an exercise	12 days?	11/27/19 8:00 AM	12/12/19 5:00 PM		
20		Get overall progress in a given language	12 days?	12/13/19 8:00 AM	12/30/19 5:00 PM	19	
21		**Writing API**	**24 days?**	**11/27/19 8:00 AM**	**12/30/19 5:00 PM**		
22		Save writing	6 days?	11/27/19 8:00 AM	12/4/19 5:00 PM		
23		List the saved writings of a user	9 days?	12/5/19 8:00 AM	12/17/19 5:00 PM	22	
24		Send writing to other users	4 days?	12/18/19 8:00 AM	12/23/19 5:00 PM	23	
25		Recommendation	5 days?	12/24/19 8:00 AM	12/30/19 5:00 PM	24	
26		**Annotation API**	**24 days?**	**11/27/19 8:00 AM**	**12/30/19 5:00 PM**		
27		List the annotators	6 days?	11/27/19 8:00 AM	12/4/19 5:00 PM		
28		Get annotations	11 days?	12/5/19 8:00 AM	12/19/19 5:00 PM	27	
29		Create and/or Update annotations	7 days?	12/20/19 8:00 AM	12/30/19 5:00 PM		
30		Final Milestone	1 day?	12/31/19 8:00 AM	12/31/19 5:00 PM	17	
31		**Frontend**	**66 days?**	**10/1/19 8:00 AM**	**12/31/19 5:00 PM**		**Burak Ilkay Akgün;Neval Tüllük;Hilal Mente**
32	✓	Login	3 days	10/1/19 8:00 AM	10/3/19 5:00 PM		
33	✓	Register	3 days	10/4/19 8:00 AM	10/8/19 5:00 PM		
34	✓	Logout	3 days	10/9/19 8:00 AM	10/11/19 5:00 PM	32	
35	✓	Profile Page	3 days	10/14/19 8:00 AM	10/16/19 5:00 PM	32;33	
36	✓	Exam Page	3 days	10/17/19 8:00 AM	10/21/19 5:00 PM	35	
37	✓	Milestone 1	1 day?	10/22/19 8:00 AM	10/22/19 5:00 PM		
38	✓	Rearrangement Profile page (feedback)	3 days?	10/23/19 8:00 AM	10/25/19 5:00 PM	35	
39	✓	Editing Exam Page (feedback)	2 days?	10/25/19 8:00 AM	10/28/19 5:00 PM	36	
40		**Writing**	**46 days?**	**10/28/19 8:00 AM**	**12/30/19 5:00 PM**		
41	✓	Designing Writing Essay Page	5 days?	10/28/19 8:00 AM	11/1/19 5:00 PM		
42		Implementing Writing page	24 days?	11/27/19 8:00 AM	12/30/19 5:00 PM	41	
43	✓	**Navbar**	**8 days?**	**11/1/19 8:00 AM**	**11/12/19 5:00 PM**		
44	✓	Designing of navbar component	3 days?	11/1/19 8:00 AM	11/5/19 5:00 PM		
45	✓	Implementing of Navbar component	5 days?	11/6/19 8:00 AM	11/12/19 5:00 PM	44	
46	✓	**Chat**	**8 days?**	**11/8/19 8:00 AM**	**11/19/19 5:00 PM**		
47	✓	Designing Message page	3 days?	11/8/19 8:00 AM	11/12/19 5:00 PM		
48	✓	Implementing Chat mechanism	5 days?	11/13/19 8:00 AM	11/19/19 5:00 PM	47	
49	✓	**Settings**	**6 days?**	**11/12/19 8:00 AM**	**11/19/19 5:00 PM**		
50	✓	Designing setting page	4 days?	11/12/19 8:00 AM	11/15/19 5:00 PM		
51	✓	**Implementing setting properties**	**4 days?**	**11/14/19 8:00 AM**	**11/19/19 5:00 PM**		
52	✓	Change bio	3 days?	11/14/19 8:00 AM	11/18/19 5:00 PM	35	
53	✓	Change avatar	3 days?	11/15/19 8:00 AM	11/19/19 5:00 PM	35	
54	✓	**Other User Profile page**	**10 days?**	**11/12/19 8:00 AM**	**11/25/19 5:00 PM**		
55	✓	Designing other user profile page	4 days?	11/12/19 8:00 AM	11/15/19 5:00 PM		
56	✓	Implementing other user profile page	5 days?	11/18/19 8:00 AM	11/22/19 5:00 PM	55	
57	✓	Send comment to other user profile	1 day?	11/25/19 8:00 AM	11/25/19 5:00 PM	56	
58		**Search**	**11 days?**	**11/19/19 8:00 AM**	**12/3/19 5:00 PM**		
59	✓	Designing search page	3 days?	11/19/19 8:00 AM	11/21/19 5:00 PM		
60	✓	Designing search result page	3 days?	11/20/19 8:00 AM	11/22/19 5:00 PM		
61	✓	Implementing user search	2 days?	11/25/19 8:00 AM	11/26/19 5:00 PM	60	
62		Implementing exercise search	5 days?	11/27/19 8:00 AM	12/3/19 5:00 PM	60	
63	✓	Milestone 2	1 day?	11/26/19 8:00 AM	11/26/19 5:00 PM	37	
64		Annotation	24 days?	11/27/19 8:00 AM	12/30/19 5:00 PM		
65		**Exercise**	**24 days?**	**11/27/19 8:00 AM**	**12/30/19 5:00 PM**		
66		Designing exercise page	11 days?	11/27/19 8:00 AM	12/11/19 5:00 PM		
67		Implementing exercies page	13 days?	12/12/19 8:00 AM	12/30/19 5:00 PM	66	
68		Final Milestone	1 day?	12/31/19 8:00 AM	12/31/19 5:00 PM	63	
69		**Android**	**66 days?**	**10/1/19 8:00 AM**	**12/31/19 5:00 PM**		**Yagmur Kahyaoglu;Bartu Oren;Ebru Zorlu;Berke Can Gürer**
70	✓	Login	3 days	10/1/19 8:00 AM	10/3/19 5:00 PM		
71	✓	Register	3 days	10/4/19 8:00 AM	10/8/19 5:00 PM		
72	✓	Logout	3 days	10/9/19 8:00 AM	10/11/19 5:00 PM	70	
73	✓	Language selection screen	2 days	10/14/19 8:00 AM	10/15/19 5:00 PM	70;71	
74	✓	Question screen	2 days	10/16/19 8:00 AM	10/17/19 5:00 PM	73	
75	✓	Profile page	2 days	10/18/19 8:00 AM	10/21/19 5:00 PM	70;74	
76	✓	Milestone 1	1 day?	10/22/19 8:00 AM	10/22/19 5:00 PM		
77	✓	Editing question screen (feedback)	3 days?	10/23/19 8:00 AM	10/25/19 5:00 PM	74	
78	✓	Editing profile screen (feedback)	3 days?	10/23/19 8:00 AM	10/25/19 5:00 PM	75	
79	✓	**Chat**	**11 days?**	**10/28/19 8:00 AM**	**11/11/19 5:00 PM**		
80	✓	Get chat message with username	4 days?	10/28/19 8:00 AM	10/31/19 5:00 PM		
81	✓	Designing chat screen	4 days?	10/29/19 8:00 AM	11/1/19 5:00 PM		
82	✓	Implementing Send message fetaure	3 days?	11/4/19 8:00 AM	11/6/19 5:00 PM	81	
83	✓	Designing and implementing Chat history screen	4 days?	11/4/19 8:00 AM	11/7/19 5:00 PM	81	
84	✓	Implementing swipe to reflesh feature	2 days?	11/8/19 8:00 AM	11/11/19 5:00 PM	83	
85	✓	**Search**	**6 days?**	**11/12/19 8:00 AM**	**11/19/19 5:00 PM**		
86	✓	Designing search screen	3 days?	11/12/19 8:00 AM	11/14/19 5:00 PM		
87	✓	Implementing user search	3 days?	11/15/19 8:00 AM	11/19/19 5:00 PM	86	
88	✓	Send comment to other user	4 days?	11/18/19 8:00 AM	11/21/19 5:00 PM	75	
89	✓	**Setting**	**5 days?**	**11/20/19 8:00 AM**	**11/26/19 5:00 PM**	**75**	
90	✓	Designing profile page to adding setting button	3 days?	11/20/19 8:00 AM	11/22/19 5:00 PM		
91	✓	Implementing permission request activity from device	2 days?	11/25/19 8:00 AM	11/26/19 5:00 PM	90	
92	✓	**Implementing setting properties**	**3 days?**	**11/21/19 8:00 AM**	**11/25/19 5:00 PM**		
93	✓	Change bio	2 days?	11/21/19 8:00 AM	11/22/19 5:00 PM		
94	✓	Change avatar	2 days?	11/22/19 8:00 AM	11/25/19 5:00 PM		
95		**Exercise**	**27 days?**	**11/22/19 8:00 AM**	**12/30/19 5:00 PM**		
96	✓	Designing Exercise selection screen	2 days?	11/22/19 8:00 AM	11/25/19 5:00 PM		
97		Implementing exercise screen	24 days?	11/27/19 8:00 AM	12/30/19 5:00 PM	96	
98	✓	Milestone 2	1 day?	11/26/19 8:00 AM	11/26/19 5:00 PM	76	
99		**Writing**	**24 days?**	**11/27/19 8:00 AM**	**12/30/19 5:00 PM**		
100		Designing writing screen	12 days?	11/27/19 8:00 AM	12/12/19 5:00 PM		
101		Implementing writing page	12 days?	12/13/19 8:00 AM	12/30/19 5:00 PM	100	
102		Annotation	24 days?	11/27/19 8:00 AM	12/30/19 5:00 PM		
103		Final Milestone	1 day?	12/31/19 8:00 AM	12/31/19 5:00 PM	98	

8) User scenarios you presented during the milestone presentation

Persona 1

Elza is a German architect who is currently living in Berlin, Germany. She uses her advanced German skills in her daily life in Germany to communicate with the locals. She also wants to improve her English skills for communicating at work.

Persona 2

Zachary is a Dutch student living in Amsterdam, Netherlands. He is confident with his English skills. Since he wants to live in Germany after graduating, he wants to improve his proficiency in German.

Scenario 1

Elza has been using Bulingo web and mobile application for a while to test her English and German skills with the username orangelion929. She wants to communicate people through the app that wants to learn German so, she wants to add the information that she is living in Germany to her bio. Since she has been using the app for some time, she also wants to update her avatar. She opens Bulingo on her mobile phone and clicks "Profile Page" button. She clicks the update symbol on the top left corner on her profile. She clicks the update button on the opening screen to select her new avatar and she enters a new bio and clicks "Save Changes". Then, she goes back to the home page and checks her chat history by clicking the mail symbol to see if she received any new messages. She sees that there is a new message from a user called angrydog556. She clicks the chat reads the message. She types a new message to reply and clicks "Send". After receiving and sending couple of messages she wants to add a comment on angrydog556's profile. She goes back to the home page and clicks search symbol. She chooses user to search and enters the username she wants to search. The users appears and she clicks on angrydog556. The profile of the user opens. She scrolls down and types a comment on the new comment layout. She also gives a rating of 5 and hits "Send" button. After talking with angrydog556 in English, she wants to see if there is any new exercises for her level. She clicks "Language Selection" button on the home page and selects "English". She clicks to the reading symbol on the bottom and checks the exercise list. Since there wasn't anything new she goes back and closes the app.

Scenario 2

Zachary has been using Bulingo web and mobile application for a while to test her English and German skills with the username angrydog556. He wants to communicate people through the app that are proficient in German so, he wants to add the information that he wants to learn German to his bio. He also wants to change his avatar since he decides not to use his own photo so anymore. He opens the app on his web browser and logins. He clicks the "Settings" button on the top right corner. He uploads a new avatar and enters his new bio and clicks "Change". He wants to chat with a user that knows German so to find this user he clicks the "Search" button on top right. He enters "German" and selects "User" to see if there is a user that used a word starting with "German". He sees in the results the user orangelion929. He clicks on the result to see her profile. After seeing her profile he decides to send her a message. He enters his message to the text box on the bottom right corner and hits "Send Message". He sees his sent message on the opening message screen. Then he receives a message from orangelion929. They send and receive a couple of messages. After he thinks it is enough for the day, he closes the web browser.

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