

# CmpE451 Fall 2019 Group3 - FINAL MILESTONE

---

## 1) Final Project Assessment

---

### 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) Milestone Assessment

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 the second 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.

For the Final milestone, both android and web teams implemented the Exercise page. After that, search page implementation were updated to search an exercise. Exercises can be searched according to tags, exercise names, language levels, languages and exercise types. Exercises are filtered according to user's language level during exercise listing part. When a user wants to take an exercise, exercises which are higher than user's level are listed. Afterwards, we implemented the recommendation. Recommendation is a feature for sending writing, which is recommendation of the proficient users to send an essay. Then, we implemented the user progress. Progress is a feature for solving exercise, which gives the number of completed exercises and these exercises' success. The radar graph on the profile page, holds the exercise levels and presents these levels on a user friendly graph. Writing and annotation are huge parts of the project. We implemented writing and annotation part together. In writing part, a user can send an essay to a recommended user. Also, a proficient user can receive and annotate an essay. On annotation part, the user can annotate a text or an image. Additionally, we implemented create exercise feature for a user, we assumed that there exists an admin who controls exercises content. Finally, exercise page is updated as read-only for guest users on the Website. A guest user can sign up and see all the exercises, but they can not see content of these exercises.

### 1.3) Final Assessment

Our project was to develop a Language Learning Platform. This platform enables a user to take level determination exam and then continue to learn a language by solving exercises. There are various exercises that are listening, reading, grammar, vocabulary and writing. Platform is also provide to use annotations for evaluating essays.

We had some difficulties during implementation period because we started wrongly prioritizing, we had difficulty in completing important parts towards the end of the project. We could do exercise part before the profile page.

Besides, we gained experience in teamwork, meetings and division of labor. Except milestone 2, we finished our work on time. Our functionalities are mostly completed. Also, the interface of our application for both mobile and web is very well designed.

In the start of project we had meetings with the whole team. Additionally, each sub-team(android, frontend and backend) completed their work by setting small meetings on their own. Aside meeting, other means of communications were conducted to complete our deliverables. These methods are github issues, whatsapp groups.

## 2) List and status of deliverables

---

Note: See project plan for more.

Name	Delivery Date	Delivered
(Backend) Exam API: Get exam questions with choices	15.10.19	DONE
(Backend) Exam API: Post user answers and get evaluation grade	15.10.19	DONE
(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 chosen 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
(Backend) Writing API: Save, List, Get, Send, Recommendation to send for Writing	21.12.19	DONE
(Backend) Annotation API: List, Get, Create, Delete annotation	20.12.19	DONE
(Backend) Exercise API: Create new exercise	22.12.19	DONE
(Frontend) Comment: Send comment to a user from user's profile page	21.11.19	DONE
(Frontend) Chat: Send message to a user from user's profile page	21.11.19	DONE
(Frontend) Exercise: Designing Exercise selection page	25.11.19	DONE
(Frontend) Exercise: Designing Exercise list page	25.11.19	DONE
(Frontend) Exercise: Implementing exercises pages	26.11.19	DONE
(Frontend) Recommendation: Implementing user recommendation for sending writing	17.12.19	DONE
(Frontend) Progress: Implementing progress page	18.12.19	DONE
(Frontend) Writing: Designing writing pages for both sending and receiving essays	19.12.19	DONE
(Frontend) Writing: Implementing writing pages	23.12.19	DONE
(Frontend) Exercise: Implementing create exercise page	23.11.19	DONE
(Frontend) Annotation: Implementing text annotation	23.12.19	DONE
(Frontend) Annotation: Implementing image annotation	24.12.19	DONE
(Frontend) Profile Page Update: Implementing radar graph	24.12.19	DONE
(Frontend) Exercise Page Update: Implementing the exercise page as only read for guest users	24.12.19	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 search	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

Name	Delivery Date	Delivered
(Mobile) Exercise: Designing Exercise selection screen	25.11.19	DONE
(Mobile) Exercise: Implementing exercises pages	27.11.19	DONE
(Mobile) Profile Page : Implementing progress bar for exercises and language	09.12.19	DONE
(Mobile) Recommendation: Implementing user recommendation for sending writing	17.12.19	DONE
(Mobile) Writing: Designing writing screens for both sending and receiving essays	19.12.19	DONE
(Mobile) Writing: Implementing writing screen	23.12.19	DONE
(Mobile) Annotation: Implementing annotation	23.12.19	DONE

#### Details about deliverables - Android

Search : User can search for other users. They can search for exercises according to chosen language, type and level.

Recommendation : User based recommendation when user wants to send writing.

Comment : User can comment on other user's bio.

Chat: User can send and receive messages from/to other users.

Profile Update : Users can update their own profile pages by changing their avatar and bio.

Exercises: User can do listening, reading, grammar and vocabulary exercises.

Annotation : User can annotate writing exercises that sent to her/him.

### 3) A summary of coding work done by each team member

Team Member	Contributions
Burak İlkey Akgün	I am working in frontend with Neval and Hilal. After milestone 2, I designed and implemented the exercise search page. I and Neval designed and implemented writing sending page with file uploading functionality. I and Neval designed and implemented text annotation for the essays that are been written on browser for sending to other users. Also I and Neval designed and implemented image annotation for the essays that are been uploaded on browser to sending other users.
Samet Demir	I am a member of backend team. For final milestone, I fixed the problems of the Writing API implemented by Berkay (basically redesigned and rewrote most of the Writing API). Then, I designed and implemented the complete Annotation API. I updated "create comment endpoint" such that it calculates the overall ratings of the users from the average of the comment ratings. I added image uploading feature to Writing API. I implemented "create new exercise endpoint" and "language progress radar endpoint". Furthermore, I fixed some bugs in Writing API and Annotation API. Finally, I debugged image annotation feature of the android and frontend applications (synchronizing the image annotation between android and frontend) since coordinates of the annotations were not properly handled by android and frontend teams. I also help the frontend team in the development and debugging of exercise, writing and annotation features. For the delivery, I prepared the code, documentation, database config & dump and packaged web application.
Yağmur Kahyaoğlu	I am a part of the Android sub-team. After the previous milestones, we implemented exercise pages for listening, writing, vocabulary, grammar and reading exercises with Bartu and Ebru. We also implemented exercise search feature. I mostly worked on backend communications and view adapters for the features. Bartu and I implemented sent & received writing pages and writing reviewing page and also text annotation feature to make and display annotations.
Bartu Ören	I am in the Android sub-team. In addition to the previous milestones, we implemented exercise search with Yağmur and Ebru. I designed the filter layout and wrote the Java code for it. We implemented listening, grammar, vocabulary, writing, reading exercise pages with Yağmur and Ebru. I took part mostly in button functionalities and layouts for all of these exercise pages. For text annotation, I worked with Yağmur to implement it for sent & received writing pages. I also implemented the image annotation feature for android, working with Java, xml and handling database connections.

Team Member	Contributions
Sabri Bayrakdar	I am a member of the backend team with Samet and Berkay. As a continuation of previous milestones, I implemented language and exercise progress api endpoints which are used to get progress. Accordingly I created progress model and seeder files. After progress implementation, I rearranged exercise evaluate api endpoint. Because when a user complete an exercise, the progress information is needed to update (post progress) so that the user can keep tracking his/her own progress. Besides the progress implementation, I also implemented recommendation api endpoint. In the meantime, I realized and cleared unnecessary codes in some endpoints which I implemented previously. Finally, I added over 20 users in database in addition the existing users so that the project seems more realistic.
Ebru Zorlu	I am working in the Android sub-team. For final milestone, we implemented exercise search, listening, reading, grammar, vocabulary, writing exercise pages and annotation part with Yağmur and Bartu.
Neval Tüllük	I implemented the text and image annotation with İlkay and the send exercises part with Samet. I also helped the implementation of the send writing page.
Berke Can Gürer	
Berkay Özerbay	I was in the backend team. For final milestone, I implemented post writing and get writing by username endpoint. I also created writing model and a sample seed and edited time to time according to coming changes.
Hilal Mente	I am a member of frontend team. In addition to the previous milestones, I implemented listening exercise and exercise filtering with Neval. After second milestone feedbacks, I implemented indicating the correct answers of questions and the user's number of correct answers after submitting the exercises' answers. Also, I implemented the radar graph which holds the levels of exercises according to exercise results.

## 5) Annotation Implementation & W3C Standard Compliance

Our annotation implementation directly follows the W3C Web Annotation Data Model (<https://www.w3.org/TR/annotation-model/>). It consists of 4 endpoints for creating, deleting, listing and getting annotations (see API documentation for details). The creating annotation endpoint takes the annotation in the W3C Web Annotation Data Model format and saves it. The listing and getting annotations endpoints return the annotations in W3C Web Annotation Data Model format.

*Most of the features of the W3C Web Annotation Data Model are supported. Here are the supported features:*

W3C Web Annotation Data Model Feature	Supported
Annotations may have @context, id, type, body, target, creator, motivation, created, and modified fields.	YES
The Annotation must have 1 or more @context values and <a href="http://www.w3.org/ns/anno.jsonld">http://www.w3.org/ns/anno.jsonld</a> must be one of them. If there is only one value, then it must be provided as a string.	1 @context only
An Annotation must have exactly 1 IRI that identifies it. (id)	YES
An Annotation must have 1 or more types	1 type only
Annotations may have 0 or more Bodies.	YES
Annotations may have 1 or more Targets.	YES
Annotations may have information about the context in which the Annotation and any External Web Resources were created, modified and used. (creator, motivation, created, and modified fields)	YES
The Target or Body resource may be a Web Resource.	YES
Resources may have id, value, source, type, format, language, processingLanguage, textDirection, creator, purpose and selector fields.	YES
Bodies or Targets which are External Web Resources must have exactly 1 id with the value of the resource's IRI.	YES
The Body or Target should have exactly 1 format associated with it, but may have 0 or more. The value of the property should be the media-type of the format, following the [rfc6838] specification.	YES

W3C Web Annotation Data Model Feature	Supported
The Body or Target should have exactly 1 language associated with it, but may have 0 or more, for example if the language cannot be identified or the resource contains a mix of languages. The value of the property should be a language code following the [bcp47] specification.	YES
Each Body and Target may have exactly 1 processingLanguage. The value of the property should be a language code following the [bcp47] specification. If this property is not present and the language property is present with a single value, then the client should use that language for processing requirements.	YES
The Body or Target may have exactly 1 textDirection associated with it. The value of the property must be one of the directions (ltr, rtl, or auto).	YES
The Body or Target may have 1 or more types (Dataset, Image, Video, Sound, Text)	YES
The Target or Body resource may be more specific than the entity identified by the resource's IRI alone.	YES
The Target or Body resource may be a specific segment of the resource. (Specific Resources)	YES
There may be 0 or more Motivations associated with the SpecificResource using purpose. (purpose)	YES
There may be 0 or more selector relationships associated with a Specific Resource. (FragmentSelector and TextQuote Selector)	YES
Selectors may have type, value, conformsTo, exact, prefix, suffix, start, and end fields.	YES
FragmentSelectors/TextQuoteSelectors must have exactly 1 type and the value must be FragmentSelector.	YES
The FragmentSelector must have exactly 1 value property.	YES
The Fragment Selector should have exactly 1 conformsTo link to the specification that defines the syntax of the fragment and must not have more than 1.	YES
Each TextQuoteSelector must have exactly 1 exact property.	YES
Each TextQuoteSelector should have exactly 1 prefix property, and must not have more than 1.	YES
Each TextQuoteSelector should have exactly 1 suffix property, and must not have more than 1.	YES

Here is an example annotation we use to annotate a writing image:

```
{
  "@context": "http://www.w3.org/ns/anno.jsonld",
  "id": "http://localhost/api/annotation/1",
  "type": "Annotation",
  "creator": "user",
  "created": "2019-12-21T04:21:07.387Z",
  "modified": "2019-12-21T04:21:07.387Z",
  "body": {
    "type": "TextualBody",
    "value": "'is' should be replaced with 'are'"
    "format": "text/plain"
  },
  "target": {
    "id": "http://18.184.207.248/api/writing/213312",
    "type": "Image",
    "creator": "berkay",
    "selector": {
      "type": "FragmentSelector",
      "value": "xywh=50,50,640,480",
      "conformsTo": "http://www.w3.org/TR/media-frags/"
    }
  }
}
```

Here is an example annotation we use to annotate a writing text:

```
{
  "@context": "http://www.w3.org/ns/anno.jsonld",
  "id": "http://localhost/api/annotation/2",
  "type": "Annotation",
  "creator": "user",
```

```

    "created": "2019-12-21T04:21:07.387Z",
    "modified": "2019-12-21T04:21:07.387Z",
    "body": {
        "type": "TextualBody",
        "value": "'is' should be replaced with 'are'"
        "format": "text/plain"
    },
    "target": {
        "id": "http://18.184.207.248/api/writing/6757",
        "creator": "berkay",
        "type": "Text",
        "selector": {
            "type": "FragmentSelector",
            "conformsTo": "http://tools.ietf.org/rfc/rfc5147",
            "value": "char=0,10"
        }
    }
}

```

## 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

- [annotation](#)
  - [create new annotation](#)
  - [deletes the annotation specified by the id](#)
  - [returns the annotation specified by the id](#)
  - [lists the annotations filtered by creator, target\\_creator, target\\_source](#)
- [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](#)
  - [return recommendation](#)
  - [evaluate the exercise](#)
  - [returns available languages](#)
  - [evaluates level determination exam](#)
  - [returns level determination exam questions](#)
- [search](#)
  - [search](#)
- [user](#)
  - [returns exercise progress](#)
  - [returns language progress](#)

- get language radar data
  - create comment for username
  - returns all users
  - returns user details
  - update user details
  - returns user comments
  - returns user language level details
- writing
    - save writing
    - get writing by id
    - list writing
    - set assignee

## annotation

---

### create new annotation

---

[Back to top](#)

This endpoint takes an annotation in the W3C Web Annotation Data Model format (<https://www.w3.org/TR/annotation-model/>) and saves it to the database

POST /api/annotation

#### Request body(JSON) Parameters

Name	Type	Description
annotation	Object	(see W3C Web Annotation Data Model for details)
annotation.type	String	
annotation.body	Object	
annotation.target	Object	
annotation.motivation	Object	

#### Success 200

Name	Type	Description
id	String	

### deletes the annotation specified by the id

---

[Back to top](#)

This endpoint deletes the annotation specified by the id

DELETE /api/annotation/:id

#### Request Param Parameters

Name	Type	Description
id	String	

## returns the annotation specified by the id

---

[Back to top](#)

This endpoint returns the annotation specified by the id in the W3C Web Annotation Data Model format (<https://www.w3.org/TR/annotation-model/>)

GET /api/annotation/:id

### Request Param Parameters

Name	Type	Description
id	String	

### body(JSON)

Name	Type	Description
annotation	Object	(see W3C Web Annotation Data Model for details)
annotation.type	String	
annotation.body	Object	
annotation.target	Object	
annotation.creator	Object	
annotation.motivation	Object	

## lists the annotations filtered by creator, target\_creator, target\_source

---

[Back to top](#)

This endpoint returns a list of annotations in the W3C Web Annotation Data Model format (<https://www.w3.org/TR/annotation-model/>) filtered by creator, target\_creator, target\_source

GET /api/annotation

### Request Query Parameters

Name	Type	Description
creator	String	
target_creator	String	
target_source	String	

### body(JSON)

Name	Type	Description
annotation	Object[]	(see W3C Web Annotation Data Model for details)
annotation.type	String	
annotation.body	Object	
annotation.target	Object	
annotation.creator	Object	

Name	Type	Description
annotation.motivation	Object	

## 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.exercise_id	Integer	exercise id
exercise.title	String	exercise title
exercise.language_abbr	String	exercise language abbreviation
exercise.exercise_type	String	exercise exercise type
exercise.level	String	exercise level
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

## return recommendation

---

[Back to top](#)

GET /api/language/:language\_abbr/recommendation/:id

### Request body(JSON)

Name	Type	Description
recommendation	Object[]	
recommendation.username	String	username
recommendation.rating	String	rating
recommendation.grade	String	grade

## evaluate the exercise

---

[Back to top](#)

POST /api/language/:language\_abbr/exercise/:exersice\_id/evaluate

### Parameter Parameters

Name	Type	Description
answers	Object[]	

Name	Type	Description
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
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

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.exercise_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)
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

---

### returns exercise progress

---

[Back to top](#)

GET /api/user/:username/exercise/:exercise\_id/progress

#### Success 200

Name	Type	Description
username	String	username of user
exercise_id	Integer	exercise id
question_done	Integer	number of questions answered correctly
questions	Integer	number of all questions
updatedAt	String	last created or updated time

### returns language progress

---

[Back to top](#)

GET /api/user/:username/language/:language\_abbr/progress

#### Success 200

Name	Type	Description
username	String	username of user

Name	Type	Description
lang_abbr	String	language abbreviation
exercise_done	Integer	number of completed exercises
exercises	Integer	number of all exercises
updatedAt	String	last created or updated time

## get language radar data

---

[Back to top](#)

GET /:username/language/:language\_abbr/radar

### Success 200

Name	Type	Description
radar	Object	
radar.listening	Integer	listening point (over 100)
radar.reading	Integer	reading point (over 100)
radar.grammar	Integer	grammar point (over 100)
radar.vocabulary	Integer	vocabulary point (over 100)
radar.writing	Integer	writing point (over 100)

## 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

## writing

---

### save writing

---

[Back to top](#)

POST /api/writing/

### Request body Parameters

Name	Type	Description
writing	Object	
writing.text	String	
writing.image	File	
writing.lang_abbr	String	
writing.title	String	(optional)

Name	Type	Description
writing.assignee	String	assignee username (optional)

## Success Response

Success-Response:

HTTP/1.1 204 OK

## get writing by id

---

[Back to top](#)

GET /api/writing/:id

## Success Response

Success-Response:

HTTP/1.1 200 OK

## Success 200

Name	Type	Description
writing	Object	writing
writing.writing_id	Integer	writing id
writing.title	String	title
writing.text	String	text
writing.image	String	image
writing.lang_abbr	String	lang_abbr
writing.written_by	String	author of writing
writing.assignee	String	assignee

## list writing

---

[Back to top](#)

GET /api/writing/

## Success Response

Success-Response:

HTTP/1.1 200 OK

## Success 200

Name	Type	Description
writings	Object[]	writings
writings.writing_id	Integer	writing id
writings.title	String	title
writings.text	String	text
writings.image	String	image
writings.lang_abbr	String	lang_abbr
writings.written_by	String	author of writing
writings.assignee	String	assignee

## set assignee

---

[Back to top](#)

PUT /api/writing/:id/assignee/:assignee\_username

### Success Response

Success-Response:

HTTP/1.1 204 OK

## 7) Requirements

---

### Glossary

- **Registered User:** a registered person whose purpose is to learn and/or to provide skill or knowledge in one or more languages
- **Guest User:** a person who have not logged in or registered yet
- **Admin:** a person whose purpose is to keep the platform intact
- **The platform:** the product with its whole functionality
- **Web Application:** a software which provides a user interface to browse the content on a Web Browser
- **Android Application:** a software which provides a user interface to browse the content on Android Mobile Operating System without using a Web Browser
- **API:** an Application Programming Interface
- **API Application:** a software which provides necessary data to the user interface applications
- **Level:** the difficulty of exercises or the language proficiency level of the user(6 categories : A1-A2-B1-B2-C1-C2 according to CEFR standards.

## 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 messages 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 Harassment

- 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 insulting 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**

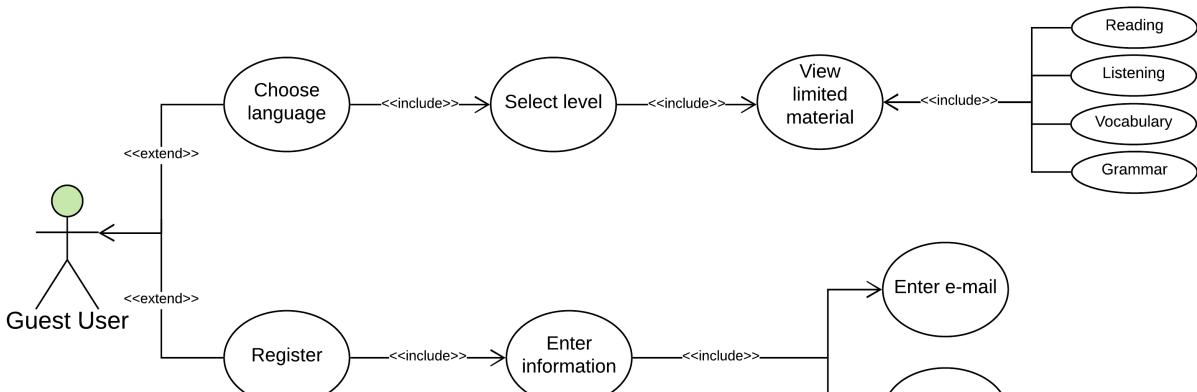
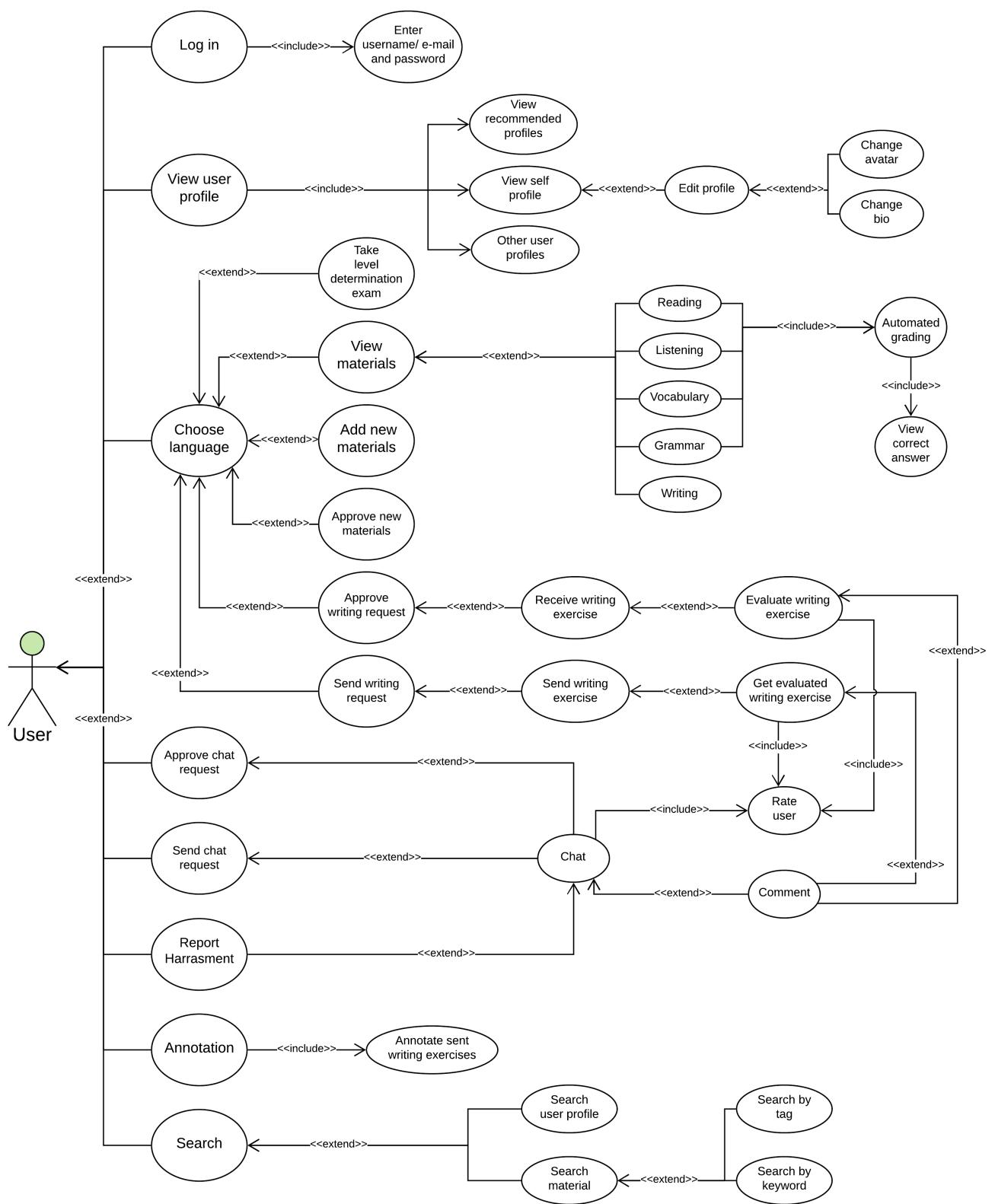
- 2.6.1 W3C Web Annotation Data Model shall be used for Annotation.
- 2.6.2 The implementation of this system shall follow the standards introduced by the World Wide Web Consortium (W3C).

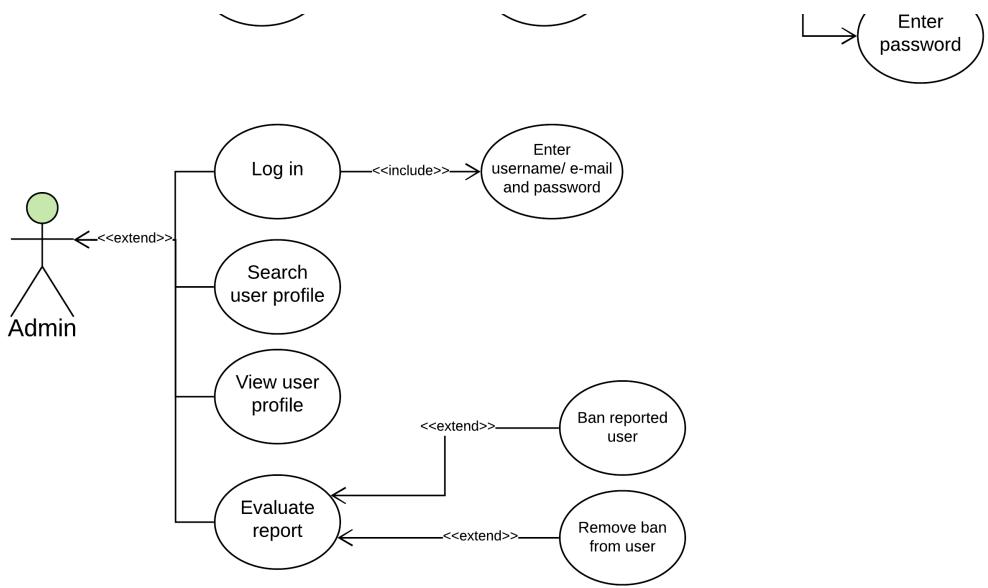
## 8) Design Documents

---

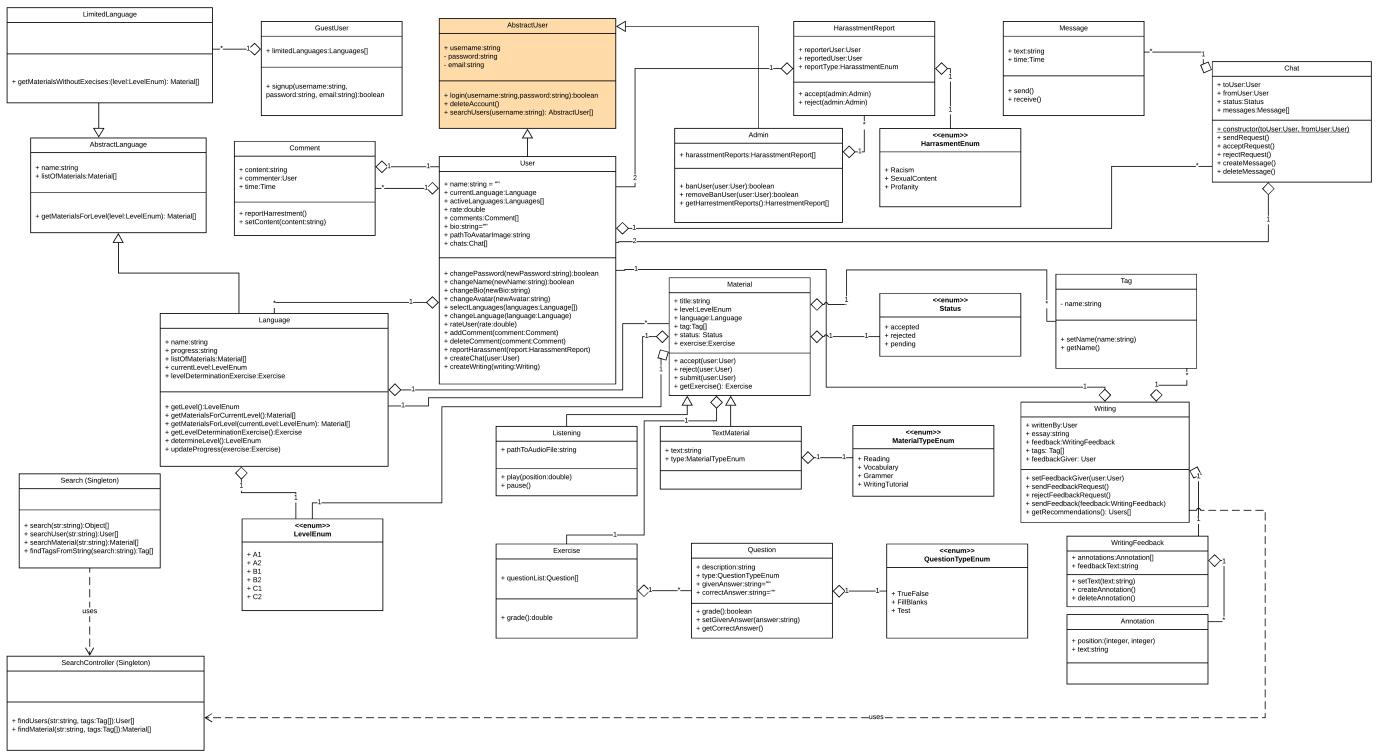
### 8.1) Use Case Diagram

---



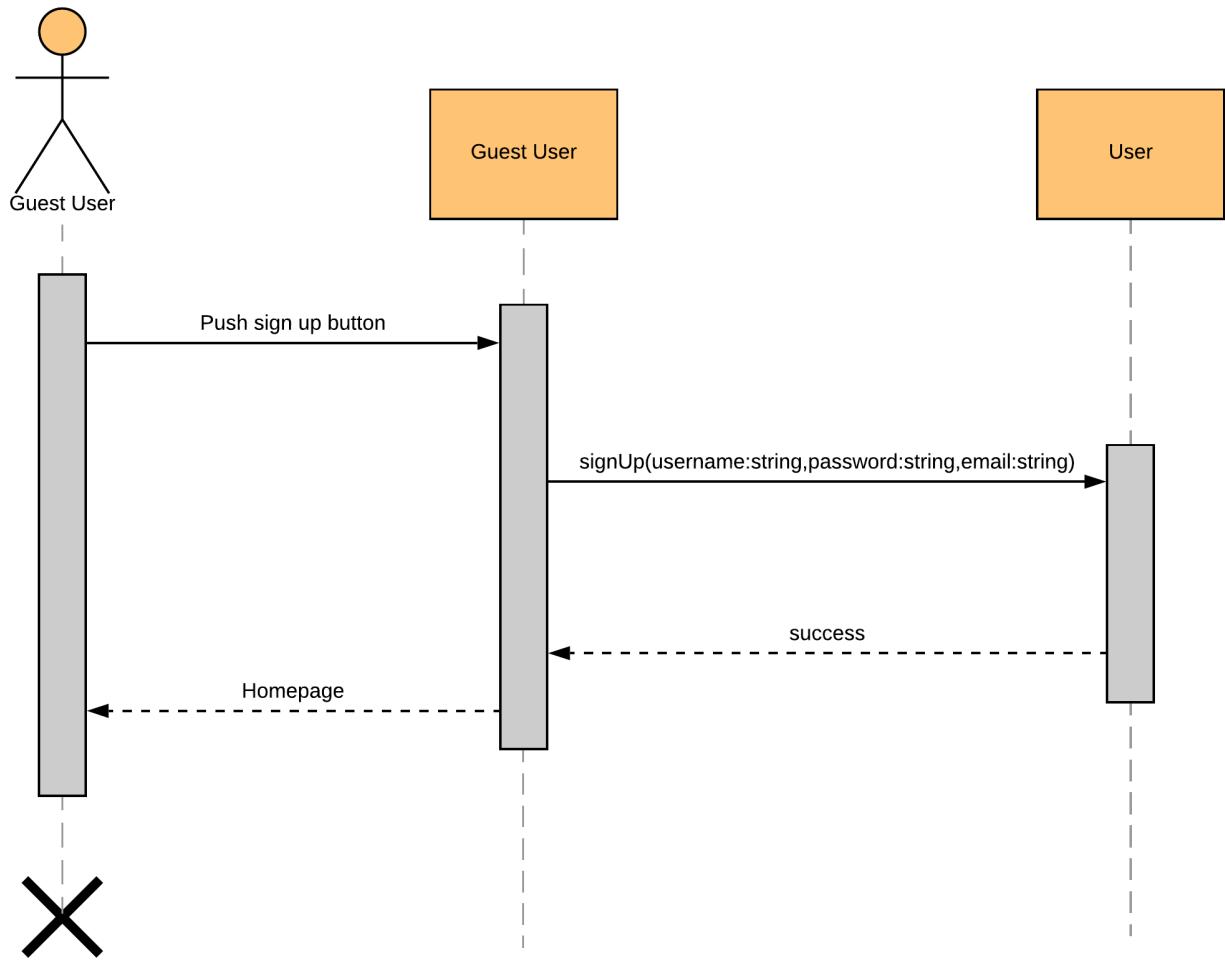


## 8.2) Class Diagram

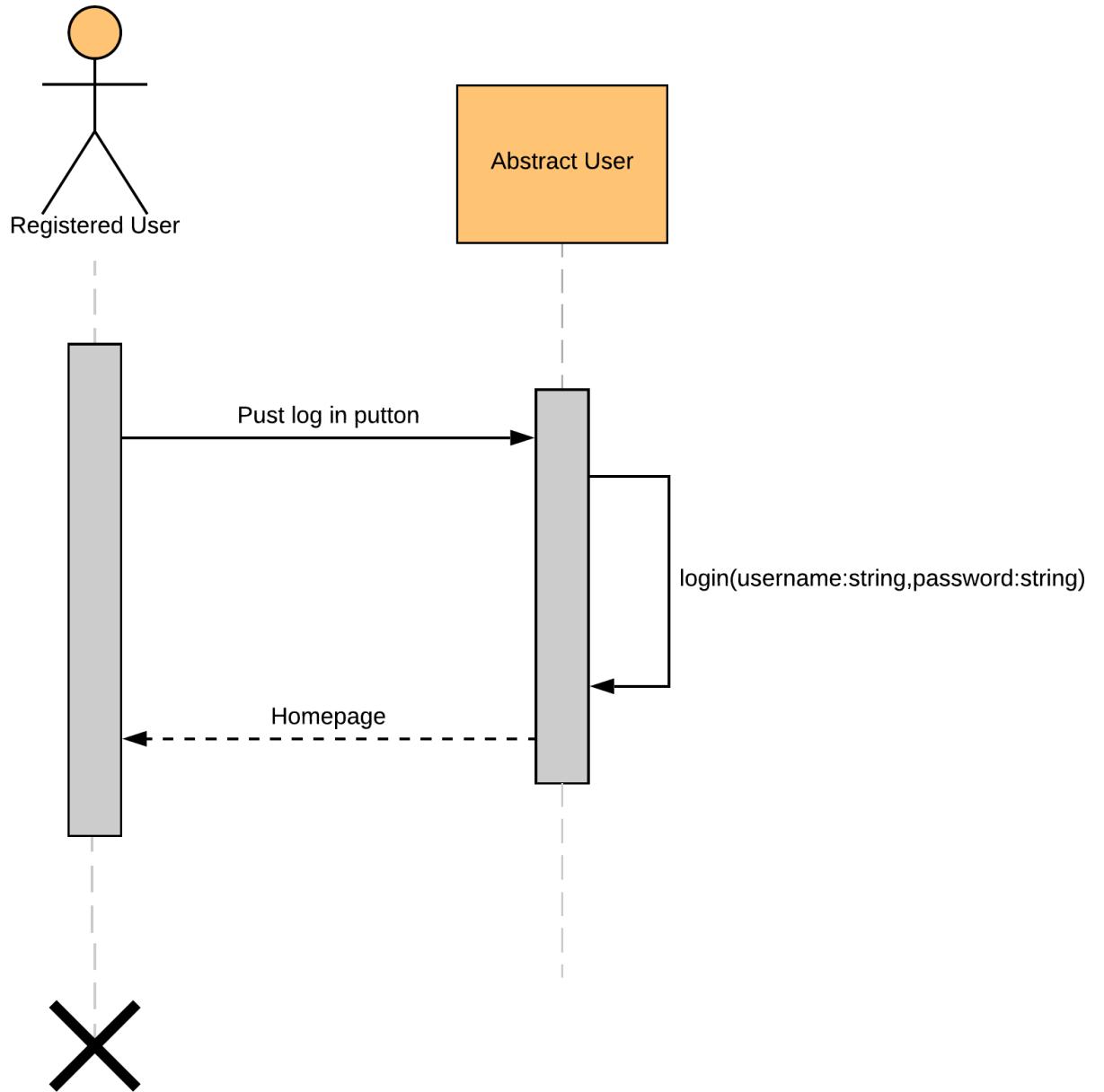


## 8.3) Sequence Diagrams

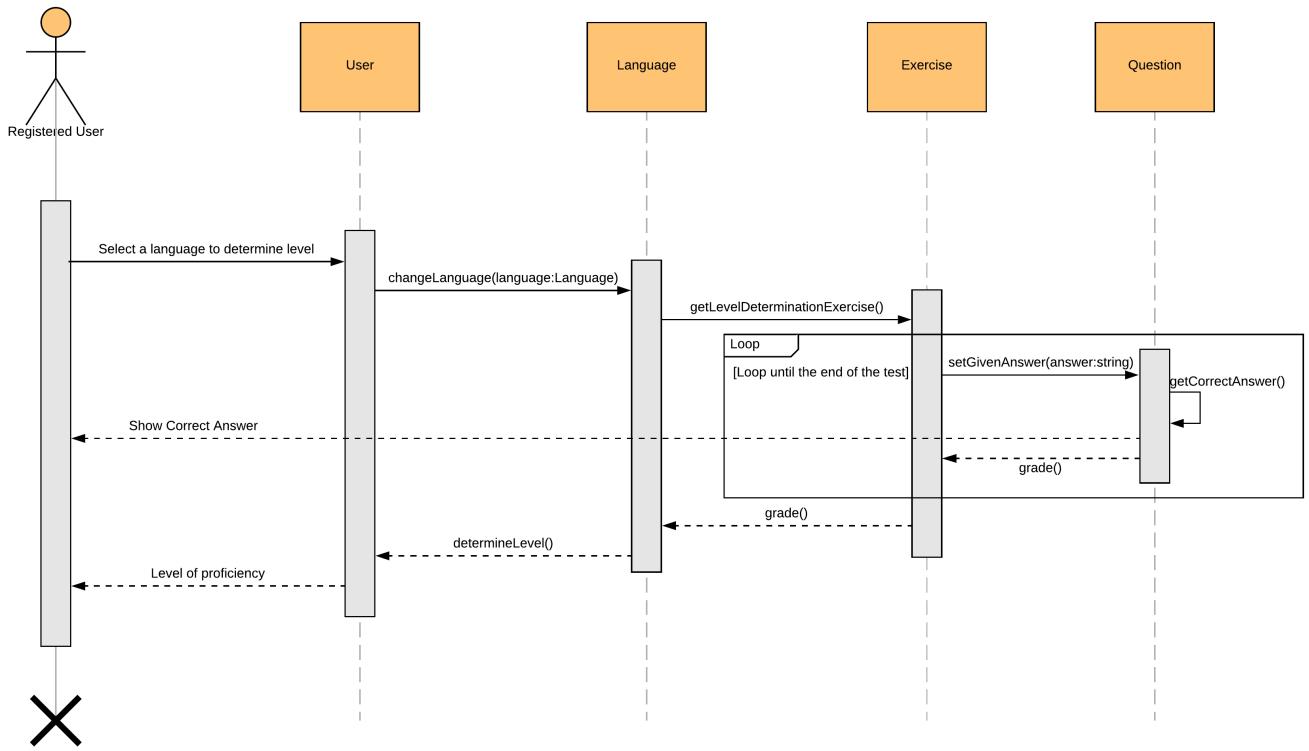
### 8.3.1) Sign Up



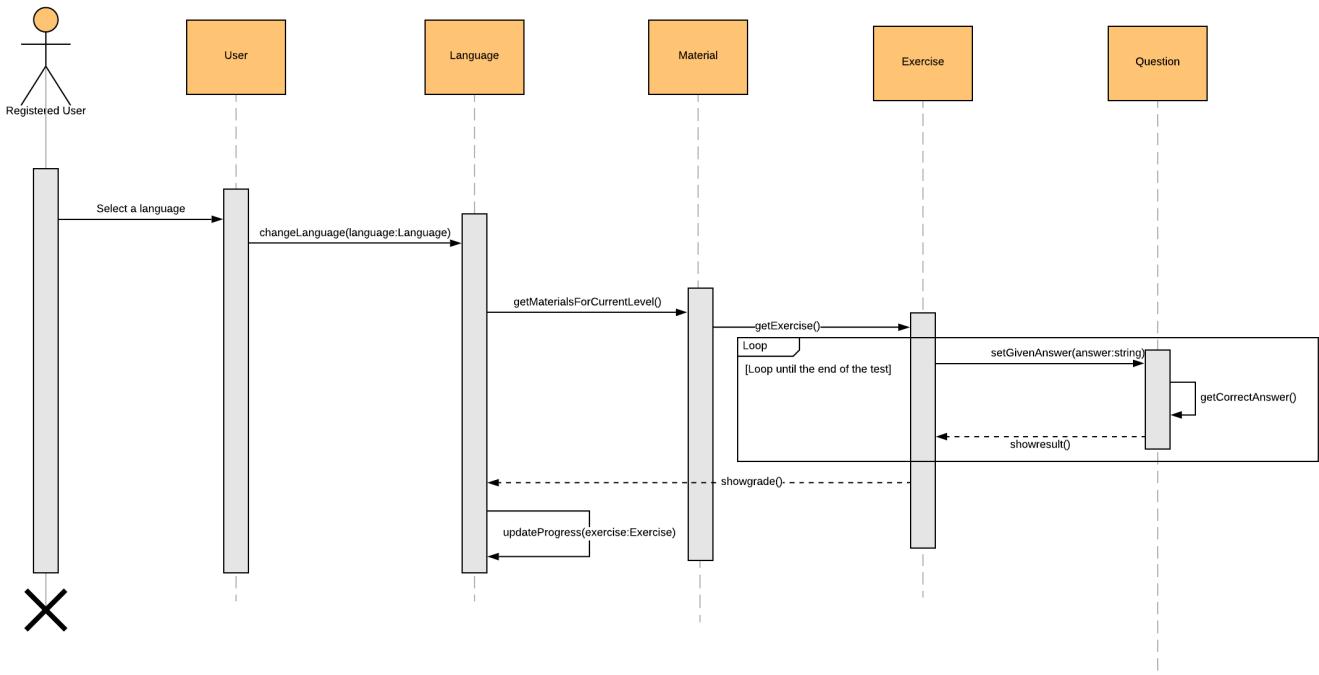
### 8.3.2) Login



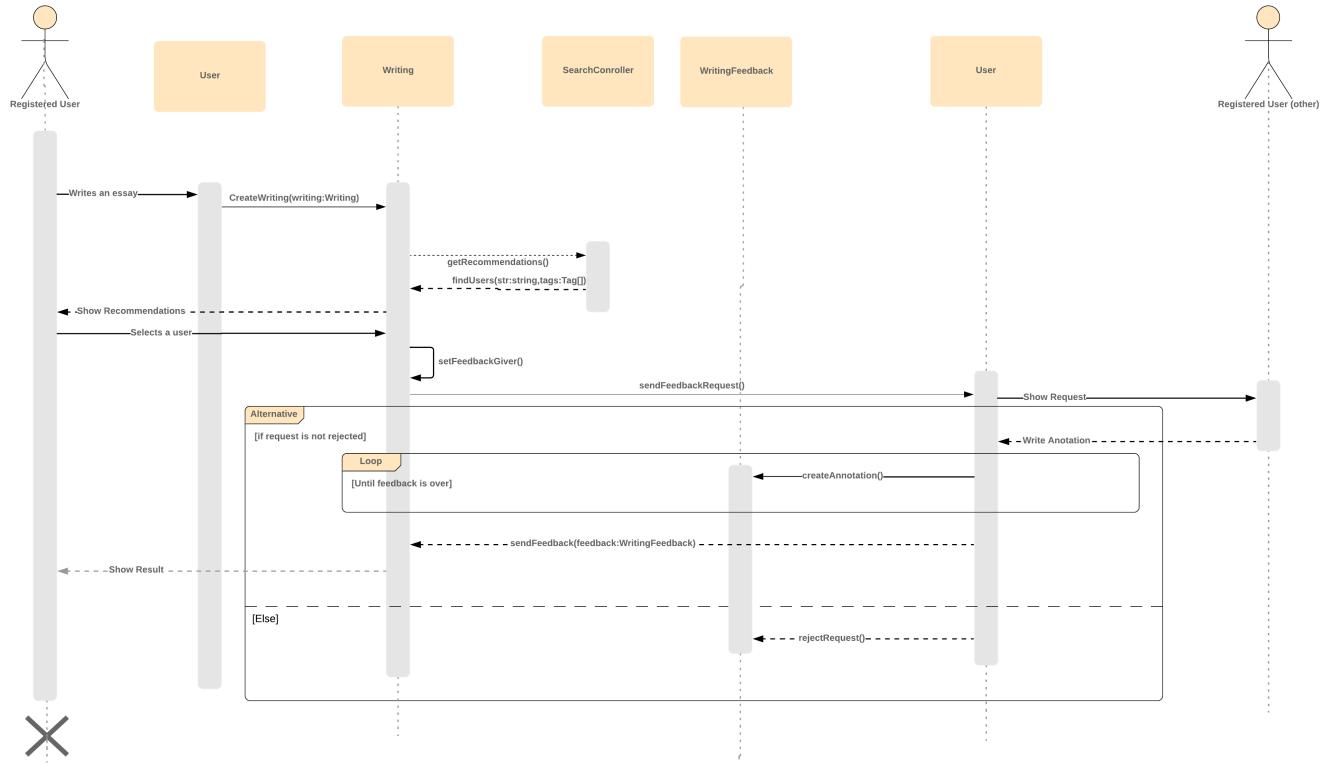
### 8.3.3) Level Determination



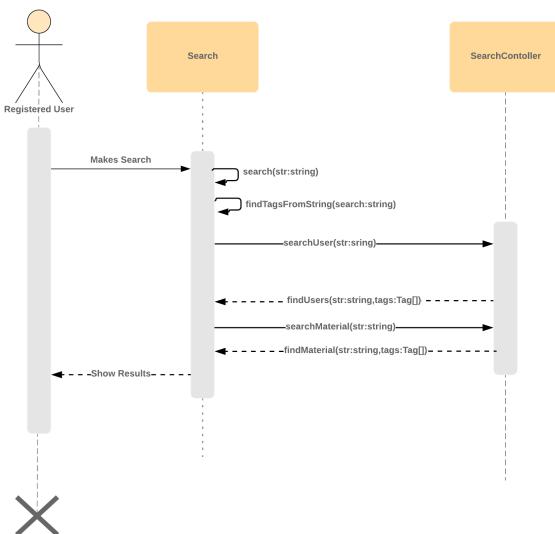
### 8.3.4) Use Language Materials



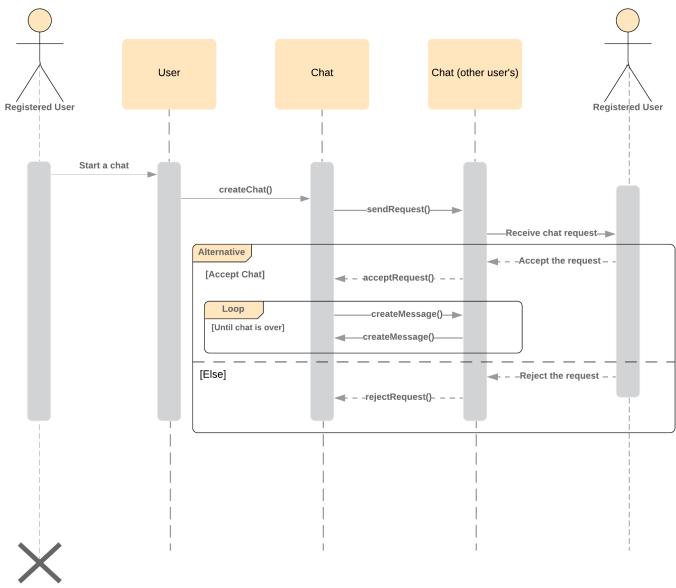
### 8.3.5) Writing and Annotation



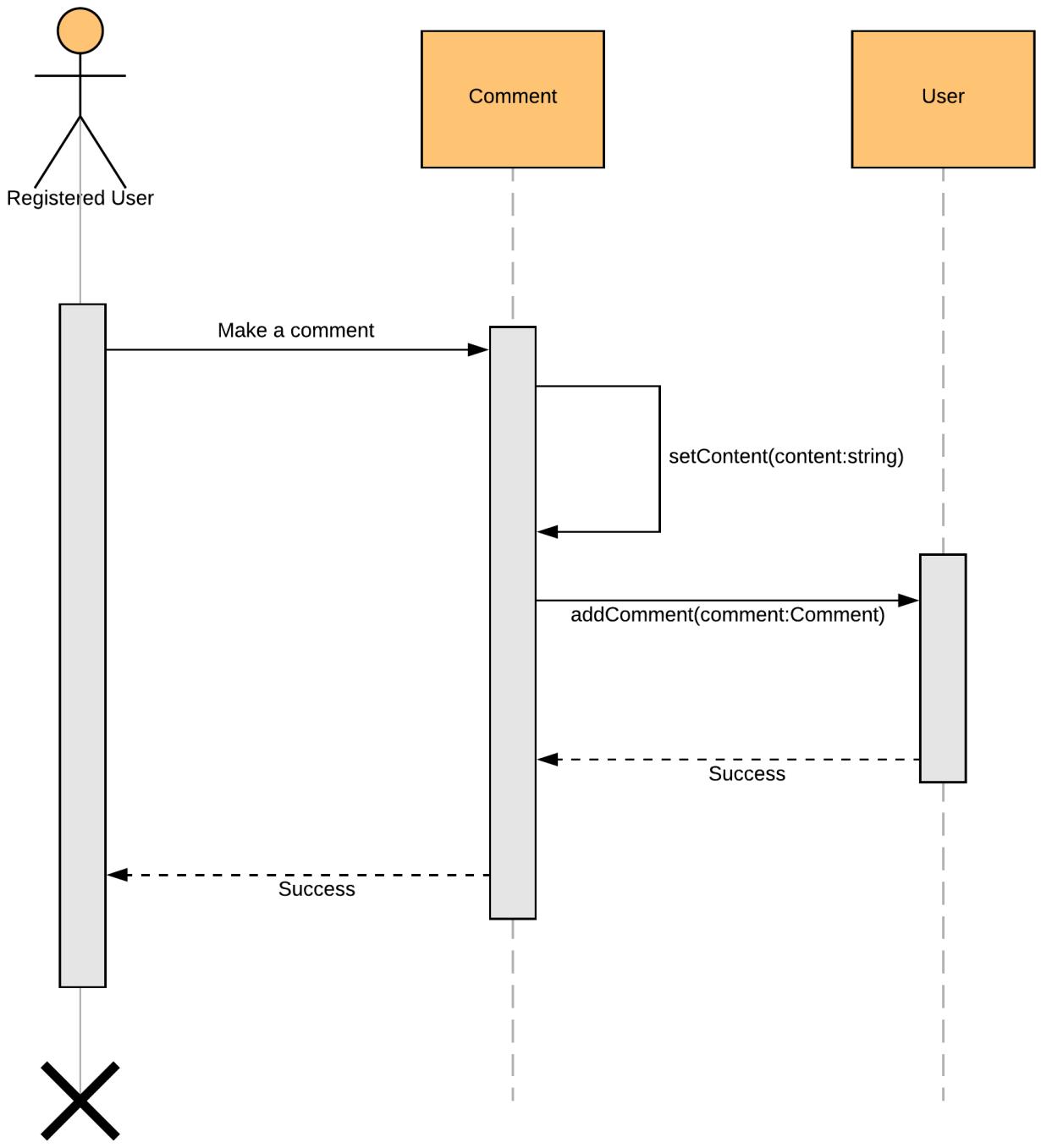
### 8.3.6) Search



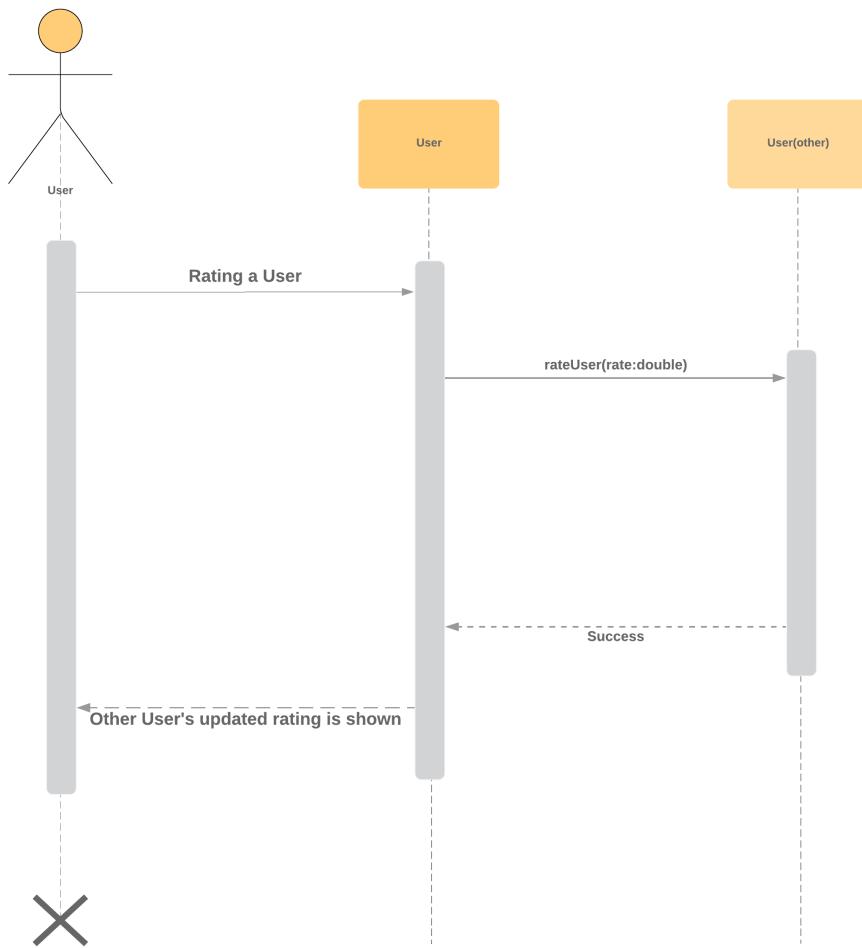
### 8.3.7) Chat



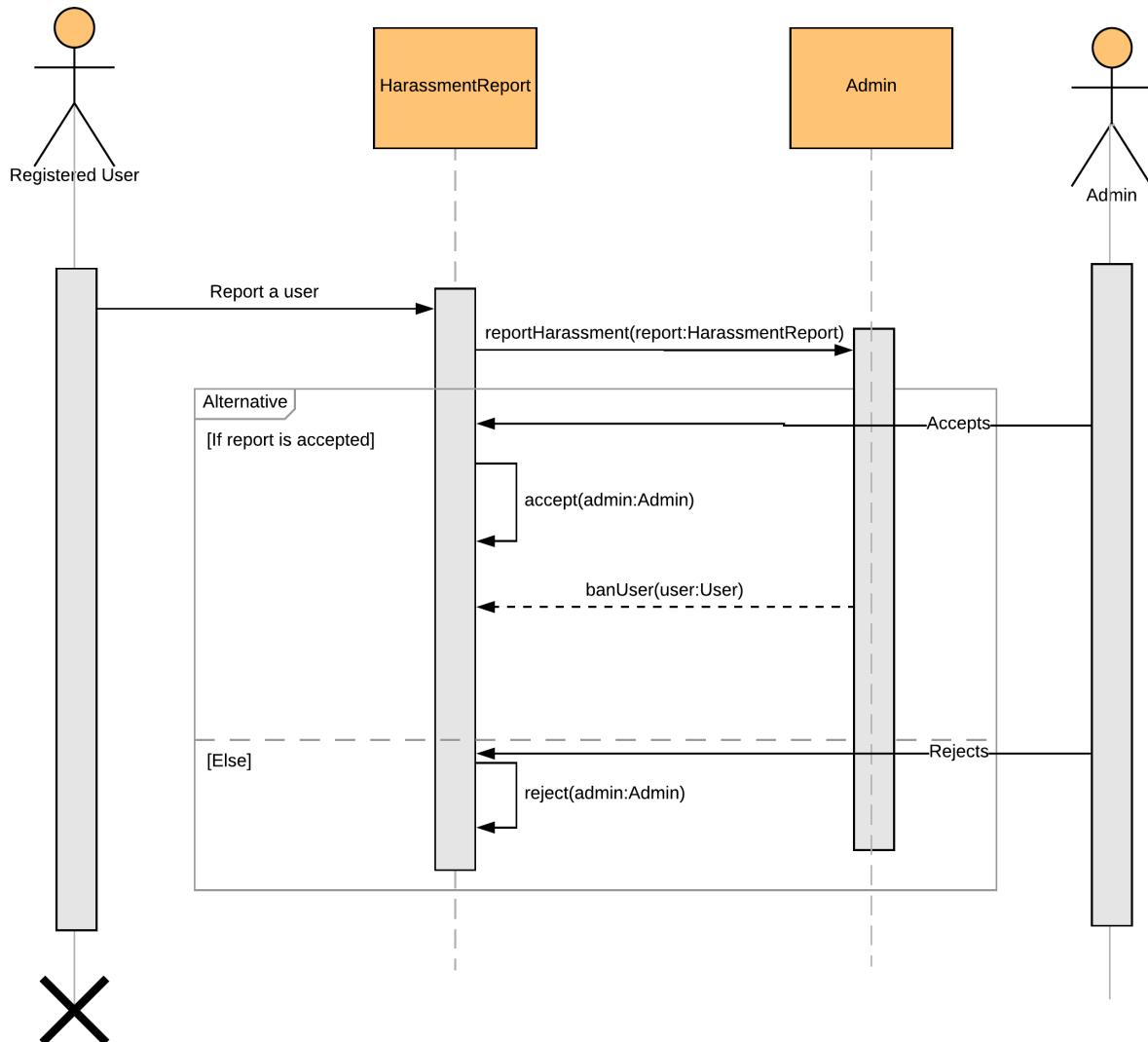
### 8.3.8) Comment



#### 8.3.9) Rate



#### 8.3.10) Report Harassment



## 8.4) Mockups

### 8.4.1) Scenario 1

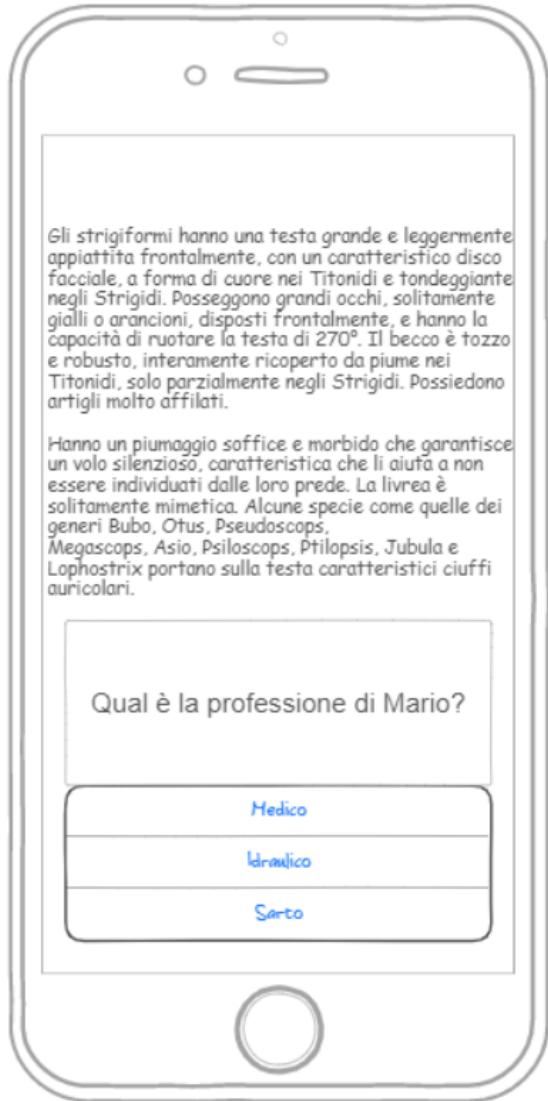
Preconditions:

- User has registered
- User has chosen Italian to work on.

Esra wants to work on vocabulary because if she knows the words of Italian, she can understand most of the sentences without knowing the grammar. She wants to work on reading materials also. If she gets used to Italian readings, she can work on Italian articles to enlarge her horizon. Apex Lingue provides best experience, that is why she chooses Apex Lingue.



Esra practices with flashcards, in that way she can learn new words easily.



She reads readings and answers questions based on the reading

#### 8.4.2) Scenario 2

##### Precondition

- User has registered

Cihat makes some minor mistakes in his essays and he wants to get rid of them. These minor mistakes makes him feel uncomfortable, and might cause a failure on TOEFL exam. He prefers our program because we provide direct interaction between user and expert.

Chooses which language he/she wants to work on.

## English

### What do you want to do?

- Writing
- Grammar
- Vocabulary
- Reading
- Listening

Chooses which course type he/she wants to work on.

## English

**Writing**

Grammar  
Vocabulary  
Reading  
Listening

Eut  
 Euthanasia  
 Eutrophication  
 Eutectic

Personal Comment:  
I am preparing for TOEFL exam...

**Euthanasia**  
In this day and time, people face many several diseases. Some kind of diseases can't be cured in our day. Such as Aids still has not any cures. With the widening the number of diseases, the opportunities that can be done has been changed, too. One of those opportunities is Euthanasia. Euthanasia is a process which is used when there is nothing to do for saving the patient. It is like killing the patient in his sleep. It could be considered as painless death. There are 2 groups of people. These 2 groups can be defined, the people who think euthanasia is needed, the people who think it's a murder.  
The first group's idea is about ending pain. When people in coma or something like that also doctors have nothing to wake him up and cure whatever happened to him...

**Choose Expert**

Writes an essay for his/her preparation about TOEFL exam.

**Expert Overview**

NAME	Tracy Irving	Kyle Irving	Teresa Vacaro	Marc Olsen
JOB TITLE	Head Instructor	Teacher	Chef	Teacher
BACKGROUND	I have a Masters degree in education and I am a certified English language teacher.	I have a Bachelor degree from University College London as well as TESL (Teaching English as a Foreign Language).	I have a Bachelor's Degree in English and has been teaching English as a foreign language since 2008.	I have a bachelor's degree in TESOL (Teaching English to Speakers of Other Languages) and Linguistics, and I have taught thousands of students all over the world for the past 10 years.
GOALS	Achieve a Band Score 6 or Higher Pass all sections of the IELTS Exam Score 7+ in the IELTS exam Confidence answer IELTS Reading Successfully overcome IELTS Listening tricks and traps Write IELTS essays with excellence	Achieve a band score of 7 and above by using and understanding a wide range of English grammar structures Boost your IELTS Writing band score by writing essays using a wide range of vocabulary Speak more naturally in the IELTS Speaking Test Improve your reading and listening skills for the IELTS Listening and IELTS Reading Tests	Use advanced vocabulary and synonyms Improve your grammar and pronunciation Give opinions and connect your ideas Deal with difficult questions Improve your speaking confidence Develop your answers and manage your time Answer any question, even if you don't know what to say!	Create coherent, organised responses to a variety of different question types Speak clearly and accurately and quickly for your audience Use a range of vocabulary appropriately and with precision for a variety of questions Create a variety of complex grammatical constructions with flexibility and accuracy

**>>**

**Details** **Details** **Details** **Details**

Chooses an expert from the expert pool to send his/her essay.

#### Expert Detail



**Key Characteristics**  
I am an IELTS and TOEFL specialist who has developed courses for international exams for 10 years. I have lived and taught in multiple countries, including China, Brazil, The Ivory Coast, Kazakhstan and Georgia. This experience provides me with a true insight into the needs and view points of all of my international students.

**NAME** Tracy Irving  
**TITLE** Head Instructor of \*\*\*

**BACKGROUND**  
28 years old  
Instructor  
5/5

**TECH PREFERENCES**  
Computer  
Mobile

**Goals**  
Achieve a Band Score 6 or Higher  
Pass all sections of the IELTS Exam  
Schedule an actual IELTS Exam  
Confidently answer IELTS Reading  
Successfully overcome IELTS Listening tricks and traps  
Write IELTS essays with excellence

**Send Writing**

Gains detailed information about the expert before sending his/her essay.

Expert Review

**Read Text**  
Write Answer



Cihat Kapusuz  
21 years old  
He is student at BOUN

**Euthanasia**  
In this day and time, people face many several diseases. Some kind of diseases can't be cured in our day. Such as Aids still has not any cures. With the widening the number of diseases, the opportunities that can be done has been changed, too. One of those opportunities is Euthanasia. Euthanasia is a process which is used when there is nothing to do for saving the patient. It is like killing the patient in his sleep. It could be considered as painless death. There are 2 groups of people. These 2 groups can be defined, the people who think euthanasia is needed, the people who think it's a murder.  
The first group's idea is about ending pain. When people in coma or something like that also doctors have nothing to wake him up and cure whatever happened to him...

**Send**

Expert reads the essay then gives the feedback.

Review Answer

**Read**  
Ask



Tracy Irving  
28 years old  
Spanish, English expert

This essay has some minor mistakes, I marked them and wrote some corrections next to your mistakes. The topic was covered well. This essay might get 75-90 over 120. If you want to get more points, you should take my advices into account carefully. If there is still some questionmarks in your head after my feedback, you can ask...

**Feedback**

**Send**

Reads the given feedback from the expert.

#### 8.4.3) Scenario 3

##### Preconditions

- Mert already has a registered account.
- He took a level determination test and placed in his appropriate level.

Mert Kara is a freshman student and studying Computer Engineering. He has a decent English, but he wants to learn French since he wants to work in France in the future. He uses his smartphone to learn French while he uses Metrobus everyday to go to his school. He is mostly interested in vocabulary for the time being.

### Mockups

1. Mert logs in to his account using his credentials.



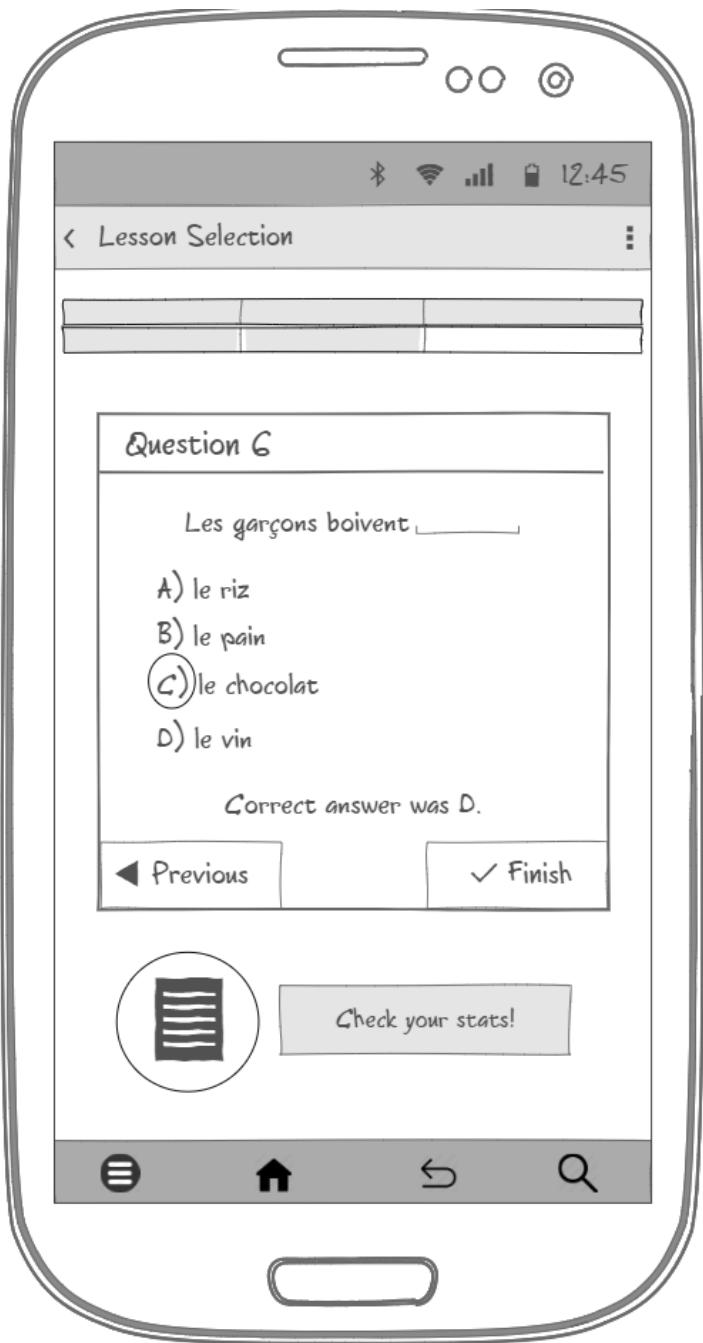
2. Mert chooses to learn French.



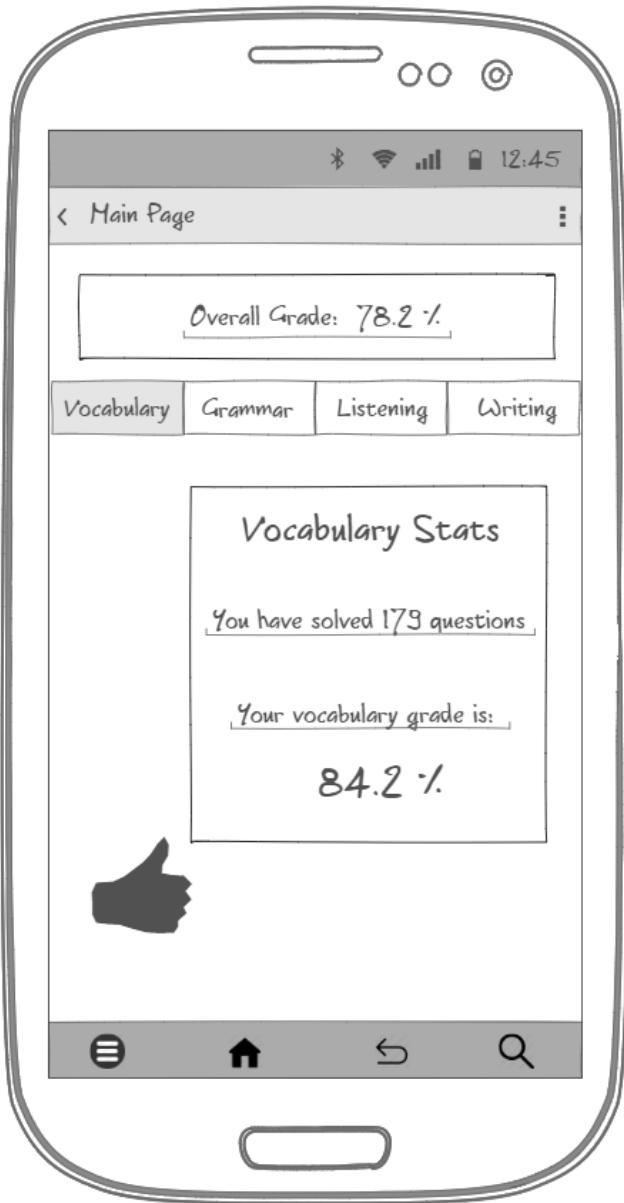
3. Mert selects vocabulary exercises to work on.



4. As Mert marks questions, he sees that his answer is wrong. Later he finds out that vine is the only beverage in the choices.



5. Mert finishes the exercise and decides to view his stats.
6. He can see both his overall grade and individual grades for various exercises.



## 9) System and User Manuals

---

### Run the complete project with Docker Compose

---

#### 1. System Manual

---

##### 1.1 Requirements

- Docker
- Docker Compose

##### 1.2 Setup

1. Clone the project from Github Repository.
2. Change directory to the project folder (bounswe2019group3/project)

##### 1.3 Run the project

1. Run the project using the following command:

```
docker-compose up
```

2. Database Setup - Initialize the database from the dump using one of the following commands:

- SQL: `docker exec -it bulingo_database bash -c "psql -U postgres bulingo < /home/db_dump.sql"`
- TAR: `docker exec -it bulingo_database bash -c "pg_restore -d bulingo /home/db_dump.tar -c -U postgres"`

3. Access the applications using the following URLs:

- frontend: <http://localhost/>
- backend: <http://localhost/api>
- android: <http://localhost/android/app.apk>

# Frontend

---

User Manual Web Application

## 1. System Manual

---

To use our web application, go to <http://18.184.207.248/>

## 2. User Manual

---

### 2.1 Login

The first page you encounter is the login page, If you are a registered user sign in with your username and password that you specify during the sign up process.

If you are a guest user you can go to Sign Up page by clicking the blue Sign Up label next to the LOGIN button or see a list of the exercises as a guest by clicking the Exercise button on the navigation bar.

### 2.2 Exercises List For Guest User

In the exercises page for a guest user first you can select the language and the see a list of all exercises for that language, to access these exercises you must create an account.

### 2.3 Sign Up

You can sign from the sign up page. It is accessible from the login page, when you click on the "sign up" text it redirects you to the sign up page. To sign up you have to provide a unique username, an email and password.

### 2.4 Profile

After you register and log in, you can see your profile page. From the profile page most of the functionalities of the application are available. From the profile page you can see your language progress in the radar chart (right side of the page), the comments done to you (middle section) and your profile picture and biography (right side of the page). From the navigation bar you can access to exercises (only the exercises in your level is accessible here), send exercise page to create and submit your own exercises, exam page to take level determination exam, my writing page to see and annotate the writings you sent to other people and the writings other people sent to you, messages page to see your personal messages, search page that you can search exercises related to a keyword or users, and the settings page where you can customize your biography.

### 2.4 Settings

In the settings page you can change your biography, profile picture and specify your preferences about accepting the writings from other users or not. By clicking the CHANGE button, you can submit your changes.

### 2.5 Level Determination Exam

Firstly you must choose the language of the level determination test with a similar process to choose exercise language selection stated above(2.2). After the selection there are multiple choice questions, and after choosing the correct answer for all the questions you should click to the submit button for evaluation. Your score will appear on the top of the page.

### 2.6 Exercises

After completing the level determination exam, you can see the exercises in your own level and work on your skills. To do so click on the exercises in the navigation bar and select the language you want to proceed. In the list of the available exercises you can click on the exercise title to go to exercise. Completing the exercise is like the level determination exam but now you can see the correct answers after completing and submitting.

## 2.7 Send Exercises

You can send exercises from send exercises page, here you should specify title, tags, language, level and exercise type of the exercise you are going to add. After that you should fill the body parts and the choices, pick the true answer from the choices and submit after completing all the questions and answers.

## 2.8 Display Writings

In my writings page you can see the writings you sent, and the writings assigned to you.

## 2.9 Display and Annotate a Writing

To go to a writing, click on the title from the list in the page My Writings, its accessible from the navigation bar. After going in the writing page you can read and annotate the writing,

### 2.9.2 Image Annotation

To annotate a text you can write your annotation to the text box and select the text you want to annotate. You can delete the yellow annotations by clicking on them once. After editing, send your annotations by pressing submit.

### 2.9.1 Text Annotation

To annotate an image, click and drag until the box covers the area you want to annotate. Release the mouse and write your annotation to the textbox. When you are ready, submit by clicking on the submit button. You can see the existing annotation by clicking on them.

## 2.10 Send Writing

To send writing, you can write your essay in the textbox, if you would like to upload a file, you should select upload essay option from the drop-down menu. After you select the language, the recommended users will appear and you can select by looking at their rating and language level.

## 2.11 Messages

To see and answer your messages go to messages page from navbar in the left you can see the list of the people you are chatting with and by clicking on the boxes on the left you can see the message history and send a message on the right

## 2.12 Search

To search a user or an exercise click on the search button on the navigation bar. First select if you want to search a user or exercise, then write your keyword for the search. For exercise you can further specify language, language level, and exercise type. Click on the search button to search to proceed.

## 2.13 Comment, Rate, Initiate Message to Another User

To make a comment and rate a user you should search them from user search and by clicking on the username from a list of users you can go to their profile. Here you can make a comment and rate the user and send a message. After sending a message you get redirected to the messages page.

# Backend

---

## 1. System Manual

---

### 1.1 Requirements

- NodeJS 10
- NPM
- PostgreSQL

## **1.2 Setup**

1. Clone the project from Github Repository.
2. Change directory to the backend folder (bounswe2019group3/project/backend)
3. Install dependencies using the following command: `npm install`
4. Start PostgreSQL database server
5. Database Setup - Initialize the database from the dump using one of the following commands:
  - SQL: `psql -U postgres bulingo < db_dump.sql`
  - TAR: `pg_restore -d bulingo db_dump.tar -c -U postgres`

## **1.3 Run the project**

1. Run the following command to start the backend server: `npm start`
2. Access the backend server on <http://localhost:3000>

# **Android**

---

## **1. System Manual**

### **1.1 Requirements**

- Android Studio
- Android SDK Tools
- JDK
- Android 6.0 and above (API Level 23)

### **1.2 Setup**

1. Clone the project from Github Repository.
2. Open the source code in Android Studio.
3. Install a virtual device on Android Studio or connect an Android phone.

### **1.3 Run the project**

1. Select a virtual device or connected Android phone in Android Studio.
2. Build and run the project on the selected device.

## **2. User Manual**

### **2.1 Requirements**

- Android 6.0+
- Internet Connection
- User permissions

### **2.2 Usage**

1. Setup APK file on your Android device.
2. Open the app.
3. Register or login with your existing account.
4. Never stop learning!

## **10) Project Plan**

