CmpE 451 - Milestone II Report

Bonibon - Language Learning Platform

Group 5

November 30, 2019

Contents

1 Executive Summary

1.1 Project Introduction

Bonibon is language learning platform open to anyone. You can sign up for free, learn languages you don't know, do exercise anytime you want and help other people with languages you are good at. There are four types of exercises: listening, reading, writing, vocabulary. Bonibon also gives you an opportunity for chatting with other learners and experts or sending them essays to improve writing skills. You can follow up your progress and level. Bonibon accepts contributions from the community. If you want to contribute to our system, you can suggest new exercises so that other people can make use of your language skills. Contribution is not only limited by the suggestions, you can also have a conversation with someone whose level is lower than you or evaluate his/her essays and give feedback.

1.2 Project Status

API: We have designed and implemented the interfaces pertaining to the goals of the first milestone.
Requirements: We defined functional and non-functional requirements. Functional requirements
has user and system requirements. Non-functional requirements has availability and accessibility,
performance and response time, annotations, reliability, security, privacy and GDPR.
Mockup - User Stories - Personas: We defined two different personas who will use the system. We created scenarios for each one of them.
Backend: Completed
Android App: Is capable of performing tasks pertaining to the goals of the first milestone. Some features that have not been implemented yet exist only as non-functional placeholders. Is written in Java, using the native Android framework.
Frontend: Is capable of performing tasks pertaining to the goals of the first milestone. Some features that have not been implemented yet exist only as placeholders. Is written using React.

1.3 Future Plans

We need to design and implement the remaining part of the API to satisfy our requirements. We also need to develop the Android app and the frontend to take advantage of them.
Maybe as backend, we can make the API more convenient for our web and mobile teams.

2 List and Status of Deliverables

Name	Delivery Date	Status
Android: Login Page	October 22, 2019	Delivered
Android: Registration Page	October 22, 2019	Delivered
Android: Language Selection Page	October 22, 2019	Delivered
Android: Proficiency Exam Page	October 22, 2019	Delivered
Android: Proficiency Exam Result Page	October 22, 2019	Delivered
Android: Main Menu Page	October 22, 2019	Delivered
Android: Exercise and Exercise Result Pages	November 26, 2019	Delivered
Android: Profile Page	November 26, 2019	Delivered
Android: Essay functionality	November 26, 2019	Delivered
Android: Chat history and new message	November 26, 2019	Delivered
Android: User Search Page	November 26, 2019	Delivered
Android: Exercise Search Page	November 26, 2019	Delivered
Android: New Exercise Suggesting Page	November 26, 2019	Delivered
Android: Listening exercises		In progress
Android: Progress tracking		In progress
Web: Home Page	October 22, 2019	Delivered
Web: Login Page	October 22, 2019	Delivered
Web: Registration Page	October 22, 2019	Delivered
Web: Language Selection Page	October 22, 2019	Delivered
Web: Proficiency Test Page	October 22, 2019	Delivered

Web: Test Result Page	October 22, 2019	Delivered
Web: Grammar Exercise Page	November 26, 2019	Delivered
Web: Listening Exercise Page	November 26, 2019	Delivered
Web: Vocabulary Exercise Page	November 26, 2019	Delivered
Web: Writing List Page	November 26, 2019	Delivered
Web: Profile Page	November 26, 2019	Delivered
Web: Essay Functionality	November 26, 2019	Delivered
Web: Chat Page	November 26, 2019	Delivered
Web: Exercise Page with Search	November 26, 2019	Delivered
Web: Recommendation Page (user)	November 26, 2019	Delivered
Backend: All endpoints	November 26, 2019	Delivered

3 Evaluation of the Status of Deliverables and Their Impact of Plan

This is our second milestone in Cmpe 451 course. We set up small milestones within each team and tried to form much of the platform before the second customer milestone. Some functionalities are still missing and being worked on, but the vast majority of the requirements are now fulfilled. Then, we have evaluated our status on each deliverable one by one. All of our application pages have both Web and Android versions and their integration to the backend server is done efficiently. Both Frontend and Backend are deployed on different AWS servers. We paid extra attention to cross-test the Web and Android versions.

3.1 Requirements

The requirements are largely unchanged since the first milestone, except for the addition of 1.1.2.2 describing how the users level up.

3.2 Project Plan

The subgroups' progresses are largely independent from one another. Each subgroup sets up intermediate milestones within itself and distributes available tasks fairly and appropriately among its members.

3.3 Home Page

Both Frontend and Android teams designed Home page while Backend team was constructing the backend server of our project. We tried to create an easy-to-navigate homepage in order to help users.

3.4 Login & Registration Page

We created our registration pages as described in the requirements. After giving their credentials, users can be able to get into the application and start learning. At the backend, the required APIs and tests for registration were implemented. At the android side, Login and Registration pages are implemented and improved.

3.5 Language Selection Page

After registration is completed, we direct our user into a Language Selection page in order to ask which language he/she wants to learn. Our application supports three languages that are Turkish, English and German.

3.6 Proficiency Exam Page

After Language Selection is completed, we canalize our user into a Proficiency Test in order to measure his/her language level. The questions are fetched from the backend. After the user chooses to submit his answers, they are sent to the backend which records the result and sends a copy in response, which is used by the frontends to display to the user his calculated level. User levels are used to filter exercises so that exercises with higher levels do not get offered to users with lower levels.

3.7 Proficiency Result Page

There are six levels starting from A1 to C2 in the application. After Proficiency Test is completed, we lead our user into a Test Result page where he/she can be able to see his/her level.

3.8 Exercise and Exercise Result Pages

To start an exercise, the user can either request a random exercise of a specific type or choose one of the results he got as the result of an exercise search. The questions are fetched from the backend. After the user chooses to submit his answers, they are sent to the backend which records the result and sends a copy in response, which is used by the frontends to display to the user the number of correct vs incorrect answers and correct answers for incorrectly answered questions.

3.9 Profile Page

Users can view other users' profiles. Among the information in the profile page are the user's name, native language, comments and average rating. A user profile page can be accessed from User Search and after annotation screen.

3.10 Essay Functionality

Users can upload essays to be reviewed by other users. Essays can be in both text and image format. The system suggests a list of reviewers, and leaves the decision up to the user. Once the upload is complete, the essay appears in the essay feed of the reviewer, who can choose to reject it, or accept and start annotating it. The author can see the annotations created by the reviewer.For Android and front-end teams, we have implemented the text and image annotation functionality and that made us ahead of the schedule. Yet we will improve these functionalities soon according to feedbacks from the demo

3.11 User Search Page

Users can use this page to search for other users based on their native language and partial username. It can be used to initiate chats with other users and view their profile.

3.12 Exercise Search Page

Users can use this page to search for exercises based on its language, tags and keywords. The user can select one of the results to start solving that exercise.For

android side this section is completed and returns exercises related language, tag and keywords.

3.13 Listening Exercises

The backend's facilities for listening exercises are in place. The frontend and android teams will be striving to complete this feature as soon as possible.

3.14 Progress Tracking

We are working to make this feature as mature and intuitive as possible.

3.15 All backend endpoints

All endpoints should be evaluated by the web and mobile. As a backend, we can make them feel comfortable as much as we can.

3.16 Android Functionalities

As Android team, firstly we stick to the communication among team and coordinated ourself according to the load sharing. Once there is an update on the backend side, we drew a schedule for our in-team organization and shared the issues. Meetings which includes all the team and the internal communication may have lead some changes for some features but we figured out how we could handle these as a team. Fortunately we had faced minor problems thanks to the communication of the team. Having in-team deliverables made-us keep planned as much as possible. Therefore we hadn't been fall behind of the schedule. That made us stick to the project plan itself.

4 A Summary of Work Done by Each Team Member

Team Member	Contribution
Hasan Öztürk	As an Android developer, I - Created a chat screen with

	message balloons where user can see previous conversations and type a new message. - Created a screen where user can see the list of users who he/she previously chatted with - Connected the existing profile page to the backend - Wrote some exercise data - Found several bugs and created issues for reporting them. - Reviewed and approved several PR's
Furkan Kadıoğlu	- All endpoints of the backend have developed. - All bug reports related to the backend have fixed
Kartal Kaan Bozdoğan	- Created the Android app proficiency exam, Exercise, commenting/rating and writing essay functionalities - Solved merge conflicts - Reviewed merge requests - Created the Android app skeleton
Ali Meriç Deşer	I have worked on Android side : - I have designed and created the Search Page and it's backend integration - I have implemented the Search Result page - I have fixed some bugs occuring from other issues and some integration problems - Improved the language selection screen - Worked on Main Menu views - Tested some features, opened issues related them contributed fixing them, merged some PRs - Created some Exercise Data

Meltem Suiçmez	- Contributed as a frontend team member, using react.js - Created profile page of self and other users - Created exercises page that shows selected language and exercise type's available exercises id, also filters them by tag and keyword - Did tests and suggestions to other functionalities of our project to teammates
Nevzat Ersoy	- Contributed as a backend team member, using django on python. - After Milestone 1 I have contributed to backend as tester, writing texts and fixing any detected problems in the code. - I have updated the wiki requirements. - I have created exercise data for backend. - Prepared the project plan.
Selamettin Dirik	- Contributed as a frontend team member, using react.js - Created vocabulary test page that shows a random test according to users selected language and level - Created grammar test page similarly. - Created listening test that plays audio file. - Reviewed and merged Mert's pull request for recommendation page - Have worked for solving the bugs concerning test results.
Mahmut Uzunpostalcı	After the first milestone I have developed following features in Android app: - An activity to list essays that are related to the user. Clicking on the essay will dırect user to the essay vıewıng page. - Series of activities that let users to suggest new exercise to the application. An activity to create

	any number of questions to the exercise. - An actıvıty that dynamically search users and lists them in the same activity for ease of use. User can change his/her search query in the same activity and search again. Clicking on the user will direct the user to the profile page of searched user. - I have corrected incorrect users in the database that don't have a native language and thus raised an error. - I have reviewed and merged multiple number of pull requests. - I have created issues for the bugs I have found and assigned them to the related teammates.
Abdullah Enes ÖNCÜ	I mainly worked on chat and annotations: -Chat page. -Image annotation. -Text annotation. -Writing file upload page. -And api connection of them. -Some bug fixes related to previous jobs.
Yusuf Mert Bila	I worked in both backend and frontend. Backend: - I added test cases. - I fixed problems in deployment and also deployed the backend on another ec2 instance. - I reviewed pull requests and gave feedback. Frontend: - I designed and implemented the user recommendation page. I also helped integrating it to sending essay page as a pop-up. - I designed and implemented writing list page, where a user can see sent and incoming writings. I also added the API call function to

	accept or reject an incoming writing for future use. I also added some exercises to the system.

5. Requirements

1. Functional Requirements

- 1.1. User Requirements
  - 1.1.1. Registration and Login
    - 1.1.1.1. Users shall be able to either create an account or continue as a guest user.
    - 1.1.1.2 Registered User
      - 1.1.1.2.1. Users shall be able to sign up by providing a username with at most 10 characters, an email, at least six character long password and native language.
      - 1.1.1.2.2. Users shall be able to login to the application by providing his/her username or email and password.
    - 1.1.1.3 Guest User
      - 1.1.1.3.1. Guest users shall have limited access to content.(Only first exercises of exercise types)
    - 1.1.1.4 Admin User
      - 1.1.1.4.1. Admin users shall be able to delete abusive content and users.
      - 1.1.1.4.2. Admin users shall be able to alter the content of exercises.
  - 1.1.2. Course Material
    - 1.1.2.1. Users shall be able to apply to any language he/she wants to learn.
    - 1.1.2.2. Users shall be able to request a proficiency exam, with respect to users' level, to level up. Users shall be able to level up if their number of correct answers is at least twice as many as their incorrect answers

        - 1.1.2.3. Users shall be able to take listening, reading, grammar, vocabulary exercises.
        - 1.1.2.4. Users shall be able to access the lower level contents.
        - 1.1.2.5. Users shall be able to suggest new material for a language.
        - 1.1.2.6. Admins shall review and approve suggested new materials to the system.
        - 1.1.2.7. Users shall be able to see the *correct* answers after the *exercise*.
    - 1.1.3 Writing Assignment
        - 1.1.3.1. Users shall be able to upload a text file or image file that includes text as a writing exercise
        - 1.1.3.2. Users shall be able to select one user and see recommended users who are provided by the application.
        - 1.1.3.3. Users shall be able to communicate with the users that are currently reviewing his/her writing exercise through a built-in communication mechanism
        - 1.1.3.4. Users shall be able to reject an essay request.
    - 1.1.4 Annotation in writing exercise
        - 1.1.4.1 Users shall be able to annotate essays that are in either text or image form, as a feedback for the writing exercises
        - 1.1.4.2 Users shall be able to annotate a text or image in a course content.
    - 1.1.5 Chat
        - 1.1.5.1 Users shall be able to chat with other users.
    - 1.1.6 Tracking for progress
        - 1.1.6.1 Users shall be able to track progress and what she/he accomplished so far.
        - 1.1.6.2 Users shall be able to see statistics about the ratio of completed exercises.
        - 1.1.6.3 Users shall be able to view average grade.
    - 1.1.7 Rating & Comment
        - 1.1.7.1 Users shall be able to view user's rates and comments about her/him.
        - 1.1.7.2 Users shall be able to vote users to whom they've sent an essay.
        - 1.1.7.3 Users shall be able to comment about users to whom they have sent an essay.
    - 1.1.8 Search

        - 1.1.8.1. Users shall be able to search a user and a content semantically through keywords.
        - 1.1.8.2. Users shall be able to do advanced search a user and a content through tags.
        - 1.1.8.3. Users shall be able to suggest tags and keywords about material which they suggest.
        - 1.1.8.4. Admin users shall be able to approve or reject the suggested tags for the materials.
- 1.2. System requirements
    - 1.2.1 Recommendation
        - 1.2.1.1 The system shall provide a recommendation mechanism for users in order to suggest a set of users that are eligible to review a writing exercise.
    - 1.2.2 Search
        - 1.2.1.1 The system shall provide a mechanism for searching a content semantically.
        - 1.2.1.2 The system shall provide a mechanism for searching a content with exact matching.
    - 1.2.3 Education Language
        - 1.2.3.1 System shall supply materials for learners in Turkish, English and German.
    - 1.2.4 Commenting and Rating
        - 1.2.4.1 System should keep comments and votes mentioned in 1.1.7 and evaluate overall ratings for the relevant user profiles.
    - 1.2.5 Annotation
        - 1.2.5.1 The system shall provide an annotation mechanism for users to annotate a text, an image file or a part of an image file.
    - 1.2.6 Communication System
        - 1.2.6.1 System should provide an interface for communication between *users.*
    - 1.2.7 Adding New Learning Materials
        - 1.2.7.1 System should redirect the materials suggested from users mentioned in 1.1.2.5 to admins in order to get approval about materials.
    - 1.2.8 Apparency & Privacy
        - 1.2.8.1 System should maintain the privacy by restricting the accessibility to user profiles from guest users.
        - 1.2.8.2 System should restrict guest users from rating & commenting for the user profiles.

    - 1.2.9 Progress & Statistics
        - 1.2.9.1 System should provide an interface to let user track and monitor his/her progress.
        - 1.2.9.2 System should also provide statistics of ratio of completed exercises, average grade.
    - 1.2.10 Automatic Grading
        - 1.2.10.1 System shall automatically grade listening, reading, grammar and vocabulary exercises and shall be capable of highlighting the correct answer if the provided one by the user is wrong.

2. Nonfunctional Requirements

- 2.1. Availability and Accessibility
    - 2.1.1. The system should work on any web browsers except IE10 and older versions.
    - 2.1.2. The application shall be provided for Android 7 and later versions.
    - 2.1.3. The system should inform the user if the application is not able to provide a service.
    - 2.1.4. If a failure happens, the problem should be fixed within 1 hour.
- 2.2. Performance and Response time
    - 2.2.1. The system shall block an excessive amount of access requests made in a short amount of time.
    - 2.2.2. The system shall back-up user-expert messages once a month.
    - 2.2.3. The system shall use encryption for user messages and passwords.
- 2.3. Annotations
    - 2.3.1. W3C Web Annotation Data Model shall be used for writing exercises which are annotated by learners and experts.
    - 2.3.2. Implementation of this system will follow the standards introduced by the World Wide Web Consortium (W3C).
- 2.4. Reliability
    - 2.4.1. The system shall endure up to 250 users without breaking.
    - 2.4.2. The system shall provide materials to searches within 3 seconds.
- 2.5. Security
    - 2.5.1. The system should use input validation.
    - 2.5.2. The authorization should always be done on the server side.
    - 2.5.3. The system should not use redirect to prevent header injection.
    - 2.5.4. The system should sanitize the user input.

- 2.6. Privacy and GDPR
    - 2.6.1. The system should have a privacy policy and this policy should be seen easily by users.
    - 2.6.2. The user should have right to object processing of self data for direct marketing.
    - 2.6.3. The user should have right to object processing that is based on profiling.
    - 2.6.4. The system should not collect or try to predict users personal data like religion, gender, identity, race, political leanings.

6 API Documentation

http://35.158.176.194/docs/
http://35.158.176.194/redocs/

7 Project Plan

{TASKS}	Duration(Days)	Start	Finish	Predecessors	Resource Names
Revision of the project	5	9.27.2019	10.3.2019		Everyone
Review requirements	5	9.27.2019	10.3.2019		Everyone
Review project plan	5	9.27.2019	10.3.2019		Everyone
Review diagrams	5	9.27.2019	10.3.2019		Everyone
[Mobile]					
Login	10	10.5.2019	10.15.2019	52	Hasan Ozturk
Sign up	10	10.7.2019	10.17.2019	53	Hasan Ozturk, Ali Meric Deser
Proficiency Exam	10	10.8.2019	10.18.2019		Kartal Kagan Bozdogan, Mahmut Uzunpostalci
Profile Page	10	10.9.2019	10.19.2019		Hasan Ozturk
Creating and deploying the app skeleton	10	10.10.2019	10.20.2019		Kartal Kagan Bozdogan, Ali Meric Deser
Main Menu Page	10	10.11.2019	10.21.2019		Mahmut Uzunpostalci
Bug fix	Always				Everyone
~Milestone 1~		10.22.2019			Everyone
Chat	10	11.3.2019	11.13.2019	67	Hasan Ozturk
Profile Page Connection to Backend	10	11.4.2019	11.14.2019	59	Hasan Ozturk
Wrote Exercise Data	10	11.12.2019	11.22.2019		Everyone
Proficiency Exam	10	10.25.2019	11.4.2019	54	Kartal Kagan Bozdogan
Comment	10	11.5.2019	11.15.2019		Kartal Kagan Bozdogan
Exercise	10	11.7.2019	11.17.2019		Kartal Kagan Bozdogan
Writing Essay	10	11.8.2019	11.18.2019		Kartal Kagan Bozdogan
Search	10	11.7.2019	11.17.2019	67	Ali Meric Deser, Mahmut Uzunpostalci
Improved the language selection screen	10	11.8.2019	11.18.2019		Ali Meric Deser
Worked on Main Menu views	10	11.10.2019	11.20.2019		Ali Meric Deser
List essays that are related to the user	10	11.12.2019	11.22.2019	67	Mahmut Uzunpostalci
User exercise suggestion	10	11.13.2019	11.23.2019	67	Mahmut Uzunpostalci
Bug Fix	Always				Everyone
~Milestone 2~		11.26.2019			Everyone

[Frontend]					
Guest		10	10.5.2019	10.15.2019	Meltem Suicmez
Login		10	10.7.2019	10.17.2019	52 Meltem Suicmez, Abullah Enes Oncu
Visual Design		10	10.8.2019	10.18.2019	Meltem Suicmez, Selamettin Dirik
Registration		10	10.9.2019	10.19.2019	53 Selamettin Dirik
Proficiency Exam		10	10.10.2019	10.20.2019	54 Selamettin Dirik, Abdullah Enes Oncu
Building the architecture of the Web App		10	10.11.2019	10.21.2019	Abdullah Enes Oncu
Bug fix	Always				Everyone
~Milestone 1~			10.22.2019		Everyone
Profile Page		10	11.3.2019	11.13.2019	59 Meltem Suicmez
Exercise Page		10	11.4.2019	11.14.2019	Meltem Suicmez
Grammar Test Page		10	11.5.2019	11.15.2019	Selamettin Dirik
Vocabulary Test Page		10	10.25.2019	11.4.2019	Selamettin Dirik
Listening Test Page		10	11.5.2019	11.15.2019	Selamettin Dirik
Chat Page		10	11.7.2019	11.17.2019	67 Abdullah Enes Oncu
Image and Text Annotation		10	11.8.2019	11.18.2019	67 Abdullah Enes Oncu
File Upload Page		10	11.7.2019	11.17.2019	Abdullah Enes Oncu
User Recommendation Page		10	11.8.2019	11.18.2019	67 Mert Bila
Writing List Page		10	11.10.2019	11.20.2019	67 Mert Bila
Wrote Exercise Data		10	11.12.2019	11.22.2019	Everyone
Bug fix	Always				Everyone
~Milestone 2~			11.26.2019		Everyone

[Backend]					
Login		10	10.5.2019	10.15.2019	Furkan Kadioglu
Register		10	10.7.2019	10.17.2019	Furkan Kadioglu
Proficiency Exam		10	10.8.2019	10.18.2019	Furkan Kadioglu, Mert Bila
API Documentation		10	10.9.2019	10.19.2019	Furkan Kadioglu
Structure Design		10	10.9.2019	10.19.2019	Furkan Kadioglu
User model		10	10.9.2019	10.19.2019	Nevzat Ersoy
Comment model		10	10.9.2019	10.19.2019	Nevzat Ersoy
Profile Page		10	10.10.2019	10.20.2019	Nevzat Ersoy
Dockerizing Backend		10	10.10.2019	10.20.2019	Mert Bila
Integrating Travis		10	10.10.2019	10.20.2019	Mert Bila
Deployment on AWS EC2 instance		10	10.11.2019	10.21.2019	Mert Bila
Exam Model		10	10.11.2019	10.21.2019	Mert Bila
Question Model		10	10.11.2019	10.21.2019	Mert Bila
Bug fix	Always				Everyone
~Milestone 1~			10.22.2019		Everyone
Backend completion		20	11.3.2019	11.23.2019	Furkan Kadioglu
Bug report fixing	Always		11.4.2019	11.14.2019	Furkan Kadioglu
Test Cases		10	11.5.2019	11.15.2019	Nevzat Ersoy, Mert Bila
Second Deployment		10	10.25.2019	11.4.2019	Mert Bila
Bug fix from tests		20	11.5.2019	11.25.2019	Nevzat Ersoy
Wrote Exercise Data		10	11.12.2019	11.22.2019	Nevzat Ersoy
~Milestone 2~			11.26.2019		Everyone

8 User Scenarios

8.1 Web Scenario

Persona: Enes Öncü

Enes is a fresh graduate, looking to improve his English using Bonibon app. He is a user of some time.

Enes logs in to Bonibon and then clicks his profile. He views his statistics of his learning of English.

He wants to get some users recommended to chat and send his essays to. He goes to the recommendation page and lists English learning/native users that are recommended. He sees kartalkaan and clicks to see his profile. He is impressed with the rating and reviews of kaan.

He goes back to the recommendation page and this time selects to chat with kartalkaan. He sends a message and asks him if he could send an essay to him.

He goes back to the recommendation page again and this time selects to send an essay to kartalkaan. He choses to upload an essay using txt format. After uploading, he annotates his essay and asks kartalkaan for reviews.

He uploads an essay similarly again using picture format this time and again he annotates the image and waits kartalkaan reviews.

8.2 Android Scenario

Persona: Halil Yılmaz

Halil is a high school student living in Adıyaman, Turkey. He thinks that he cannot learn English at school, thus he looks for an alternative ways to learn. Halil spends most of this time in transportation, since his home is very far away from the school and he plans to utilize this time.

He previously registered to our system and took a proficiency exam to learn his level. Now he just wants to learn new stuff!

He opens up the app and navigates to exercise search screen, because he wants to solve an exercise to learn about new words about “vehicles”. He types “vehicles” as tag and “car” as a keyword and searches for results. He randomly chooses a vocabulary test out of the results. After the test ends, he sees his overall result and looks for his mistakes and the correct answers.

Then, Halil wants to solve a grammar test. For that, he clicks exercises button and chooses the grammar test option. Then a randomly selected grammar test appears and Halil solves it and sees the results similar to the previous test. Furthermore, he solves randomly selected reading and vocabulary tests.

Now, he wants to write an essay and send for a review. He types the essay in the application and clicks the send button. Then our system recommends top reviewers to Halil. Halil chooses user “kbozdogan” as reviewer and send his essay. “Kbozdogan” checks for his essays and sees the incoming essay from Halil. Then he corrects Halil’s

typos by making annotations on the words. Afterwards Halil checks his essays and sees the “kbozdogan”’s corrections.

Halil is very pleased with the review of “kbozdogan” and he wants to make a positive comment about him on his profile. So he clicks the profile icon and writes a positive comment and gives a 5 as a rating.

After all the exercises and the writings, Halil now wants to get interact with someone. To do that, he uses our user search and chat features. He searches for a user named “Tom” and picks a user whose native language is English. Then he writes a “Hello” message to Tom. Tom replies as “Nice to meet you” and They talk for a while.

After spending plenty of good time in our app, Halil wants to make a contribution to the app by uploading a new exercise. So he creates a new grammar test about “propositions” and uploads to the system so that other users can benefit from the test. Of course this new test is reviewed by the app admins before making it available.

9. Code Structure

9.1 Branch Structure

The project has three phases as known:

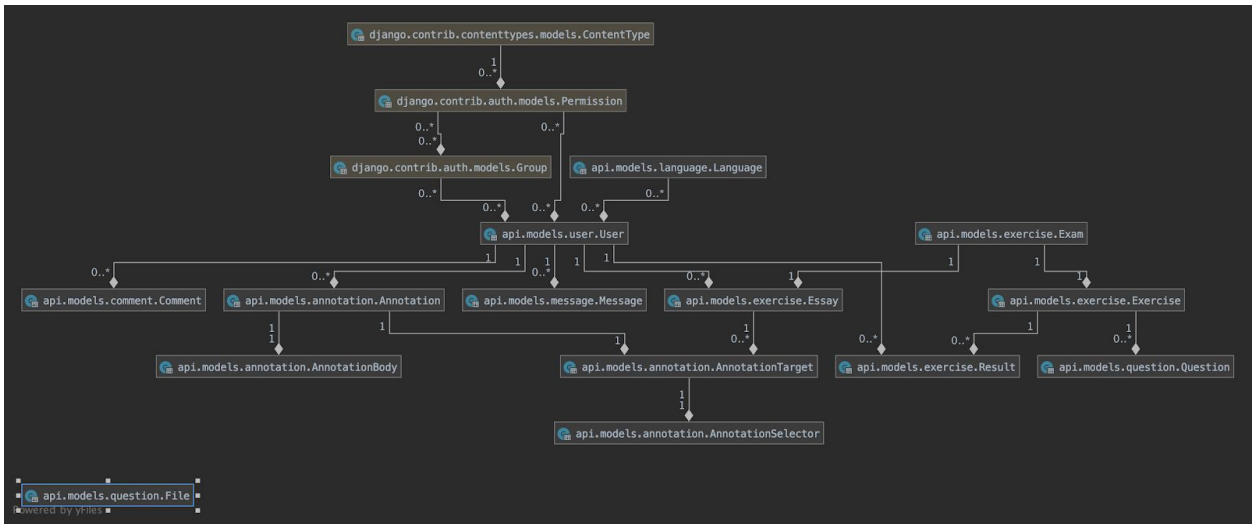
- Backend
- Frontend
- Mobile (Android)

We, as a team have started to develop all of these phases from scratch. So, it Is very likely to have code congestions while working with a 10 people team. Therefore, we came up with an idea that we should create 3 branches the same Name with phases (backend,frontend,mobile) we thought that this isolation may let Us work more clearly and we can combine our work when we finish coding.
This main branches can also have sub-branches for development purposes, we name them as <branch_name>_issue<#number>_<feature_summary> and we Keep our updated work in the main branches by merging when the changes are done.

9.2 Pull Request Structure

As mentioned above, we have branches and sub branches, Because of the fact that we work separately, we may have conflicts. We talk as a team and assign reviewers for pull requests and if there are conflicts, we try to resolve them with communicating each other.So far, we haven't had any disasters and we handled arising conflicts easily.

9.3 Django Module Dependency Diagram



Class diagram can be accessed
https://github.com/bounswe/bounswe2019group5/tree/backend/backend/diagrams

10. Evaluation of Tools

The project has many phases and thus, we needed more tools for each coding, testing deployment and documentary issues.For example the android phase is developed with Java programming language using Android Studio and also emulators. The project has evolved with the help of the tools. By holding to given requirements for tool selection, we also try to use most modern technologies for developing and deployment purposes. Some of these are changeable for person to person(IDE choice for example). But the main technologies are common. Backend is developed Django restful API by using Pycharm and Travis. Pycharm is very useful for python environment, tracking database changes and the changes of version. Front-end is developed by using React. We have also deployed our project. We have considered the ease of use, scalability, collaborative(While working together) while selecting tools. So far, we have managed to work together in harmony without having problems with the help of the tools we have used.

11. Managing the Project

The project is being developed using the modern software development approaches. Working in a crowded team makes development a little challenging, that's why we have implemented some methods to make it easier.

Groups: We have 3 groups for 3 phases of the project. These are Backend, Frontend and Android teams. Keeping groups smaller makes development easier. Also communication is easier. For sure, we keep the communication constant as a whole communication group.

Meetings: We have weekly meetings inside groups and also as a whole project team. We share the current status of our phases, discuss what to do next and how to integrate together. We try to be clean as much as possible.

Other Communication: Besides our meetings, we also communicate from other channels like slack and whatsapp groups.

Git & Branches and PRs : We use Git as version control management system. As mentioned above, we have branches and sub branches just to keep our project simple and easy to control. We assign reviewers and merge requests to merge our work together without having conflicts.

All of these above are the techniques that we use to manage our project. As our instructor said, communication is the key. That's why we try to manage as a team at first.