CmpE 352 - Milestone 2 Report

Bonibon - Language Learning Platform
Group 5

12 May 2019

Contents

1. Executive Summary

1.1 Introduction :

Bonibon is a language learning platform open to anyone. You can sign up for free, learn languages you want to learn, 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 our system, you can suggest new exercises so that other people can make use of your language learning 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

Project Plan:

Project Plan is the diagram, or to do list for our project which we plan, track, and check our status while creating our program. We have created our project plan and it is updated with necessary changes.

Implementation Decisions:

As we discussed which languages and tools we use for the implementation of the project at the Future Plans section in our Milestone 1, we decided to use NodeJS for backend operations with some frameworks like Express for our API configurations. We have started our work to learn NodeJS with some tutorials. We have decided to use React for the frontend operations of our project. We have used MySQL for database operations in our API configurations.

Database Model Creation:

As the first step of our Implementation work, we have decided to create a database model with states and relations. This was crucial due to inside communication of our project API. We have decided our tables with fields and also relations. Using MySQL we have created our model and shared inside our group.

API Implementation:

We have decided to start with vocabulary exercises. In our exercise, we supply users 4 pictures and 1 correct answer for corresponding picture. We have 3 mini API functionalities. Category listing, creating exercise and result checking. We are using Wikidata API to get pictures and names according to categories. These categories have unique ID's. We post these ids with database queries and it returns related words for it. We have the maintable code for Oxford Dictionary API to get definitions yet we didn't implement this part for now. We are using React for the frontend operations, React app calls our API and it responses for the required operations.

1.3 Future Plans

We talked that we can use our Oxford dictionary API code for the definitions and sentences for future sections. There may be a dictionary part for our project for this manner. Our design can be enhanced. Due to the fact that most of our group members are new in coding on these languages, we believe that we can be better in time. Because of inexperience, we have faced with some problems. Technical inability, planning and dividing the whole work and communication were the main problems. We discussed these and in time we believe that we can be better.

2. List and status of deliverables

Deliverable	Delivery Status
Project Plan	✓
Implementation Assignment	✓

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

- Project Plan: First version of Project Plan was done using ProjectLibre and it had some deficiencies. Then second and qualified version was completed using SmartSheets. Now our Project Plan shows us the work done and the work ahead, giving us valuable information on how to plan and move forward.

- Implementation Assignment: We agreed upon the idea that the API that we will choose should be related to our original project, which is language learning platform. Wikidata API provides lots of utilities and we used those to create a image test. We have made a detailed meeting and we determined the technologies that we will use and tasks to do. Every individual tried to write his/her own. Sometimes we encountered conflicts and we solved them with additional meetings. Effective usage of github repository was crucial during the coding period. As we discussed during the lectures, unit tests is crucial to check a specific portion of our code works as expected or not. In this project, members wrote unit tests for the code that they write themselves. We have used mocha and chai libraries. In order to provide a better understanding on our project we created use case, class and sequence diagrams. Also we have specified the API URL with its own document.

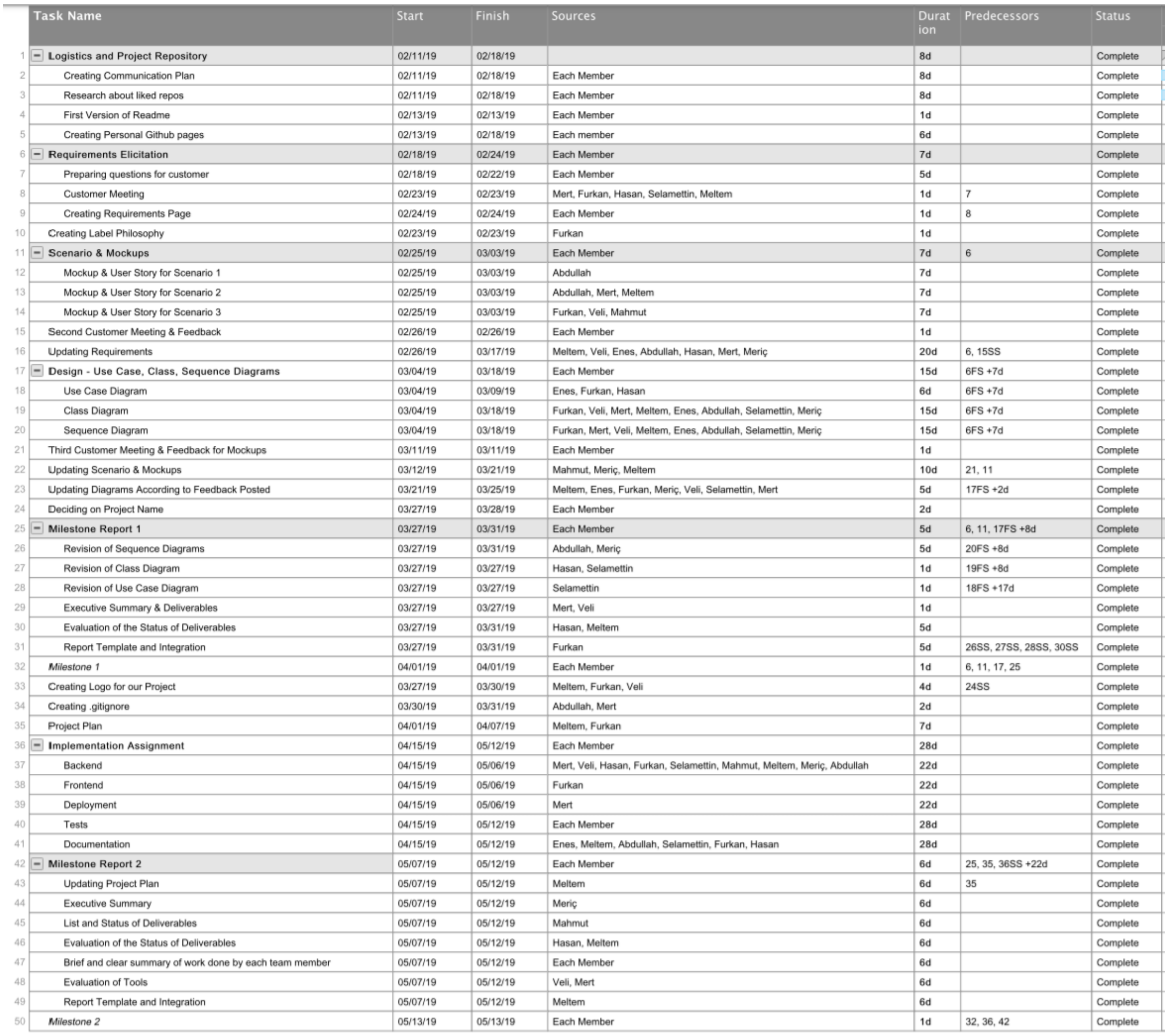
4. Summary of work done by each team member

Team Member	Contributions
Veli Can Ünal:	At the second sprint, we took some baby steps for development of our project.We use node.js for all development process. First, we take some data from wiki data by using some post request. After that Abdullah Yıldız and I parsed it and print it on an HTML in our local. I wrote code for getting data from Oxford dictionary api. I connected the wiki data api and Oxford dictionary api each other. Hasan and I made data usable for front end part. Since we quit the idea of using Oxford dictionary api, I removed the integration of the Oxford dictionary api from our main api and made our code work without it. We didn’t have any test environment. I searched usage of travis, how we integrate to our project.
Furkan Kadıoğlu	I have researched api to find best api for language learning platform. I have learnt how can i make react app for practice-app.I have written frontend part and primitive backend apis for integration. I reviewed Mert's code for database. I wrote unit test for getWordsOfAClass function. I have reviewed Enes unit test by observing chai framework. I also reviewed Selamettin's unit test and and suggestion about informative feedback for frontend. I tested last version of backend by using postman. I created branches and corresponding pull request to merge frontend and basic api works.
Hasan Öztürk	For the implementation project, I took over the part which looks for Wikidata API and fetches item which has 16 or more subclasses. This is important because we are creating tests which includes 4 questions. Therefore we need 16 words belonging to a class. After I fetching the items by writing the necessary queries, I inserted to our mySQL database. Afterwards, I updated the frontend end code for our home page. When user clicks for the home page, our app fetches items from database which I put earlier. This is very critical, because Wikidata has millions of items but only a small portion of them is needed for us. I wrote the necessary software which handles this functionalities.

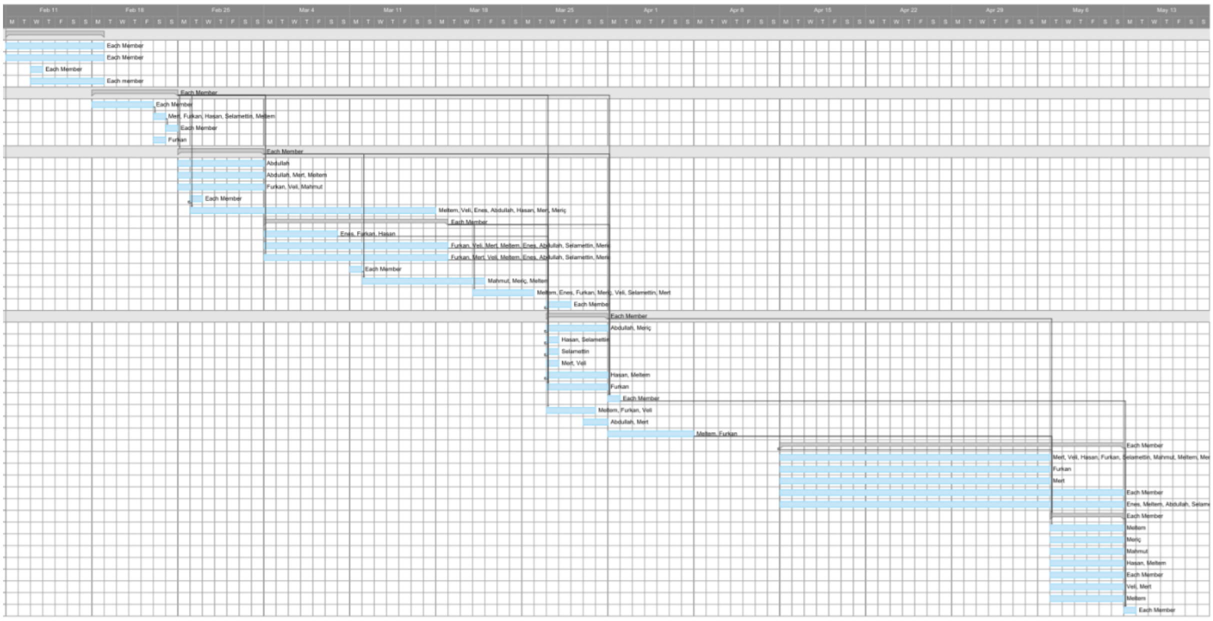
Ali Meriç Deşer	I created database model for our project. Created tables and relations with necessary fields and keys.I suggested my team the exercise types, and we decided to implement. I tried to learn SQL, Node.JS, JavaScript with tutorials. I have tried to handle the database communication part of our project. I tried to write queries to maintain this communication. I have written the sections of table filling methods. I have tested the API integrations with wikidata.Made some reviews and gave some feedbacks to my teammates.
Abdullah Enes ÖNCÜ	We had a createExercise function in order to provide an exercise to user. However, it wasn’t randomly generated and always the first options were correct answer. I edited it and randomized questions options. On the other hand, I added language selection option to frontend and I rearranged createExercise function with language selection. In order to translate vocabs to a language I used Google Translate API’s. I write a Unit Test for controlling randomization of createExercise.
Meltem Suiçmez	I created a draft for project plan on Word because my projectlibre was Turkish and I couldn't turn it to English. In my draft there were all the tasks, dates, sources and predecessors. Then Furkan and I put them together on Projectlibre. I attended PS on 16th April and informed my teammates about the contents of our implementation assignment. I tried to learn javascript, node.js, mysql from scratch for our assignment but it was challenging for me. I tried to write methods for modifying database using data but it had some errors. I created meeting note 13’s 1st 2nd and 4th parts. I did design of API with Abdullah. I created part 3 of milestone 2 report with Hasan. I updated project plan using a different platform called SmartSheets. This was easier to use and i added new assignments here.
Mahmut Uzunpostalcı	For the practice app I took part in wiriting getDatafromWikiData for wikiData query. Previously out wikiData query could return duplicates of same objects. After modifying it, now returns unique objects for our exercise. Before deciding on react as our frontend software I have worked on pug for frontend. Apart from that I have learnt javascript from scratch and it was hard for me. Also I have created list and status of deliverables section of our report.

Abdullah Yıldız	I integrated our app with Travis.CI, added a .travis.yml file to a new branch Travis. I prepared meeting notes 11 and 12. Me and Velican worked together that we pulled info from Wikidata then printed them on html page as keyword-URL pairs. Me and Meltem worked on design of our app. We created class/use case diagrams.
Yusuf Mert Bila	I wrote the getClassDataFromWikiData function get image and word data from Wikidata API. Then I fixed the function generateRenderedArray that parses this data and function sendRequestForOxfordApi that gets data from Oxford API to work asynchronously. I wrote getWordObjectsFromOxfordApi function to send requests to Oxford API for each word that comes from Wikidata API. I helped Hasan to write insertAnswers function to insert answers of generated exercise to database and later I modified it to work asynchronously. I helped Hasan to debug the script that fetches classes with more than 16 subclasses in Wikidata. I fixed a bug in createExercise method which causes the answers in database and answers sent to frontend be different. I wrote connectToDatabase.js to connect to database easier. I wrote the function getExerciseAnswers to fetch the answers of an exercise and function checkExercise to evaluate results of a completed exercise. I wrote the Word model in Word.js to represent a Word object, when we get the whole data for an object from Wikidata and Oxford APIs. Then I wrote testGenerateRenderedArray.js test file using chai framework to test the generateRenderedArray function. I opened an EC2 instance on AWS, I set up the database and installed the dependencies on EC2 and deployed the application.

5. Project Plan



	Task Name	Start	Finish	Sources	Duration	Predecessors	Status
1	**Logistics and Project Repository**	02/11/19	02/18/19		8d		Complete
2	Creating Communication Plan	02/11/19	02/18/19	Each Member	8d		Complete
3	Research about liked repos	02/11/19	02/18/19	Each Member	8d		Complete
4	First Version of Readme	02/13/19	02/13/19	Each Member	1d		Complete
5	Creating Personal Github pages	02/13/19	02/18/19	Each member	6d		Complete
6	**Requirements Elicitation**	02/18/19	02/24/19	Each Member	7d		Complete
7	Preparing questions for customer	02/18/19	02/22/19	Each Member	5d		Complete
8	Customer Meeting	02/23/19	02/23/19	Mert, Furkan, Hasan, Selamettin, Meltem	1d	7	Complete
9	Creating Requirements Page	02/24/19	02/24/19	Each Member	1d	8	Complete
10	Creating Label Philosophy	02/23/19	02/23/19	Furkan	1d		Complete
11	**Scenario & Mockups**	02/25/19	03/03/19	Each Member	7d	6	Complete
12	Mockup & User Story for Scenario 1	02/25/19	03/03/19	Abdullah	7d		Complete
13	Mockup & User Story for Scenario 2	02/25/19	03/03/19	Abdullah, Mert, Meltem	7d		Complete
14	Mockup & User Story for Scenario 3	02/25/19	03/03/19	Furkan, Veli, Mahmut	7d		Complete
15	Second Customer Meeting & Feedback	02/26/19	02/26/19	Each Member	1d		Complete
16	Updating Requirements	02/26/19	03/17/19	Meltem, Veli, Enes, Abdullah, Hasan, Mert, Meriç	20d	6, 15SS	Complete
17	**Design - Use Case, Class, Sequence Diagrams**	03/04/19	03/18/19	Each Member	15d	6FS +7d	Complete
18	Use Case Diagram	03/04/19	03/09/19	Enes, Furkan, Hasan	6d	6FS +7d	Complete
19	Class Diagram	03/04/19	03/18/19	Furkan, Veli, Mert, Meltem, Enes, Abdullah, Selamettin, Meriç	15d	6FS +7d	Complete
20	Sequence Diagram	03/04/19	03/18/19	Furkan, Mert, Veli, Meltem, Enes, Abdullah, Selamettin, Meriç	15d	6FS +7d	Complete
21	Third Customer Meeting & Feedback for Mockups	03/11/19	03/11/19	Each Member	1d		Complete
22	Updating Scenario & Mockups	03/12/19	03/21/19	Mahmut, Meriç, Meltem	10d	21, 11	Complete
23	Updating Diagrams According to Feedback Posted	03/21/19	03/25/19	Meltem, Enes, Furkan, Meriç, Veli, Selamettin, Mert	5d	17FS +2d	Complete
24	Deciding on Project Name	03/27/19	03/28/19	Each Member	2d		Complete
25	**Milestone Report 1**	03/27/19	03/31/19	Each Member	5d	6, 11, 17FS +8d	Complete
26	Revision of Sequence Diagrams	03/27/19	03/31/19	Abdullah, Meriç	5d	20FS +8d	Complete
27	Revision of Class Diagram	03/27/19	03/27/19	Hasan, Selamettin	1d	19FS +8d	Complete
28	Revision of Use Case Diagram	03/27/19	03/27/19	Selamettin	1d	18FS +17d	Complete
29	Executive Summary & Deliverables	03/27/19	03/27/19	Mert, Veli	1d		Complete
30	Evaluation of the Status of Deliverables	03/27/19	03/31/19	Hasan, Meltem	5d		Complete
31	Report Template and Integration	03/27/19	03/31/19	Furkan	5d	26SS, 27SS, 28SS, 30SS	Complete
32	*Milestone 1*	04/01/19	04/01/19	Each Member	1d	6, 11, 17, 25	Complete
33	Creating Logo for our Project	03/27/19	03/30/19	Meltem, Furkan, Veli	4d	24SS	Complete
34	Creating .gitignore	03/30/19	03/31/19	Abdullah, Mert	2d		Complete
35	Project Plan	04/01/19	04/07/19	Meltem, Furkan	7d		Complete
36	**Implementation Assignment**	04/15/19	05/12/19	Each Member	28d		Complete
37	Backend	04/15/19	05/06/19	Mert, Veli, Hasan, Furkan, Selamettin, Mahmut, Meltem, Meriç, Abdullah	22d		Complete
38	Frontend	04/15/19	05/06/19	Furkan	22d		Complete
39	Deployment	04/15/19	05/06/19	Mert	22d		Complete
40	Tests	04/15/19	05/12/19	Each Member	28d		Complete
41	Documentation	04/15/19	05/12/19	Enes, Meltem, Abdullah, Selamettin, Furkan, Hasan	28d		Complete
42	**Milestone Report 2**	05/07/19	05/12/19	Each Member	6d	25, 35, 36SS +22d	Complete
43	Updating Project Plan	05/07/19	05/12/19	Meltem	6d	35	Complete
44	Executive Summary	05/07/19	05/12/19	Meriç	6d		Complete
45	List and Status of Deliverables	05/07/19	05/12/19	Mahmut	6d		Complete
46	Evaluation of the Status of Deliverables	05/07/19	05/12/19	Hasan, Meltem	6d		Complete
47	Brief and clear summary of work done by each team member	05/07/19	05/12/19	Each Member	6d		Complete
48	Evaluation of Tools	05/07/19	05/12/19	Veli, Mert	6d		Complete
49	Report Template and Integration	05/07/19	05/12/19	Meltem	6d		Complete
50	*Milestone 2*	05/13/19	05/13/19	Each Member	1d	32, 36, 42	Complete



6. Evaluation of tools and processes:

- Slack:
  Slack is the very versatile tool for not also communication but also making follow ups. We integrated our github account with slack and It notifies every team member when someone changes some part of our project. This feature makes our group really responsive.

- NodeJs:
  NodeJs is javascript run-time environment. Node Js works asynchronously.Since It has non-blocking input/output model, you can make some real time changes. The programmer needs to execute a bunch of code written in Javascript for using Nodejs. As a team, we didn't have so much information about the designing web application. We searched it and we understood that besides all advantages of Node js, It is also easy to learn. The most challenging part for us was writing our code asynchronously. At first, we added some callbacks to our codes. When it didn't work, we added some setTimeout functions that gives our function some delay. At the end we learn how to use promises, await and async in our functions.

- Postman:
  Postman is a tool for sending requests to HTTP clients by using an endpoint and several parameters.
  The main advantage of using postman is you don't need to write code or any environment. You just write the parameters and put the endpoint accordingly, It will returns the result.

- Travis:
  Travis has very good technical environment for test, deployment and integration. At first we thought about using just Mocha as testing environment. However, Travis gives us more than running our test codes.It can make deployment and basic integrations with other tools. It also supports aws environment. We just need to add script to our code and whole other things are handled by travis. It also notifies the coder for the unsuccessful unit tests.

- Amazon Web Services:
  Amazon Web Services (AWS) is a cloud platform and it has almost everything a developer needs for developing and deploying an application. We used a EC2 instance in AWS to deploy our application. After getting used to it, EC2 is easy to manage. What we did for deployment is basically cloning the repo to EC2, adding the credentials and running the program. The only tricky part was daemonizing the frontend and backend applications but we used a program called "forever" and it was smooth sailing after that.

- MySQL:
  There wasn't a strong choice about relational databases in our group but some of us had experience with MySQL beforehand so we went with it. It was comfortable to use. Running it on local machines or EC2 was easy. We could also use MySQL Workbench program to visualize our tables better and that helped to speed up our development progress.

- Dividing the team into groups of 2:
  This idea sounded good at the start but we realized it was making it hard to communicate within the team. Groups didn't have much idea of what other groups were doing, so we dropped the idea.

7.API URL (documented)

http://ec2-18-195-228-58.eu-central-1.compute.amazonaws.com:3000/home
This is our home page. It shows the topics you can choose to create exercise questions about. We have 20 topics.
http://ec2-18-195-228-58.eu-central-1.compute.amazonaws.com:3000/exercise
When you click a topic, exercise page opens and it shows related questions. There are four questions. Each of them consists of a picture and four options. There is only one correct answer which is the name of the object in the picture. There is a submit exercise button that you can click.
http://ec2-18-195-228-58.eu-central-1.compute.amazonaws.com:3000/result
This is the result page you go when you click submit exercise. It shows how many questions you solved were correct and how many were wrong. It also provides a informative feedback, showing each question's correct answer.