# Contents

# Analysis and Requirement Report

*Lingui: A Personalized Language Learning App Using Videos and Spaced Repetition*

## 1 Introduction

Most people use the same approach when it comes to learning and teaching new languages: the skill-building approach. This approach is quite intuitive at first, suggesting that you learn the language by consciously learning grammar and applying it when producing output [1]. However, fluency requires more than that: a person must be able to conjugate verbs, choose the correct words, put the words in the correct order, etc. in a very short time, which can only be achieved by acquiring the language. Therefore, we reject the skill-building approach all the other language learning apps and the conventional language learning methods follow and accept a completely different approach: the Comprehensible Input Hypothesis, which can be summarized as follows: “We all acquire language in the same way: by understanding messages” [1].

To speak a language fluently, one needs to effortlessly construct new sentences. This instinct can be developed by acquiring the language, not by consciously learning the grammar rules [2, 3, 4]. Most native speakers do not know the grammar of their native language, despite being fluent in the language. So, we know that fluency is not achieved by learning the grammar of the language. This brings us to our first point: fluency is only possible by developing an instinct for the language.

Learning the definition is not enough to develop an instinct for the words in a language. You also need to see the word in hundreds of different contexts to have a feel for how the word should be used. Every time you hear a word and understand it, the better your instinct becomes, and eventually, you know how the word is used so well that you can use it in your sentences, too. In other words, you acquire the word. Here is our second point: the meaning of a word is acquired when it is seen and understood many times in different contexts.

We did not learn our native language by checking the dictionary each time we heard a word we did not know. We heard it in different contexts, and eventually, we understood the word. After we developed an instinct for the language, we started to speak. You can get the meaning of a word you do not know, or a grammatical structure from context, if you understand the meaning of the overall sentence [1, 3, 4]. To sum it up, the third point is that we can understand the meaning of a word if it is in a sentence where we understand the general meaning.

Lingui combines this approach and the three ideas mentioned. We want our users to watch YouTube videos in such a way that the language level of the video is never too easy or too difficult for the user. We will achieve this by tracking the vocabulary of every user. To ensure that the user understands the meaning of an unknown word in

context, we want to give the word in sentences that do not contain any other unknown words. After seeing the words in such sentences, a certain number of times, a spaced repetition system will be used to test the user if they learned the word correctly, and to retain this vocabulary knowledge.

Our goal is not to teach the users some phrases they can use in situations they encounter. Language books and other language-learning apps already do that. Our goal is something bigger: to make our users fluent.

# 2 Proposed System

## 2.1 Overview

Lingui will be a mobile application designed for both Android and iOS users to learn vocabulary and practice their desired language by being exposed to words and sentences. For both accessibility and feedback purposes the application will be free and available on the mobile application stores of the given operating systems. For future profits, it is planned to put ads on different parts of the app.

In order to give you the reader a good preliminary understanding of the proposed system, we created a brief overview of the system using simpler terms. The application will require an account from each user in order to keep track of their learning process. Each new user will be assessed on their personal language knowledge and be presented with personalized content according to that assessment. The users will also have lists of words that they “do not know”, “have learnt” and are “in progress to learn”. According to these lists of words the application will find YouTube videos that contain these words. The user will be presented with these words in order both to teach new words and make the memory of the learnt words more concrete through a spaced repetition of the word. These videos will feature clickable subtitles separate from YouTube. While watching these videos, the user will also be able to select the words in the video that they do not know. If the user selects a word, they will first be presented with a small window showing the meaning of the word. The user will be able to add this word to a list if they wish to. After their learning process, the users will also be able to take short written quizzes in order to test their comprehension of the vocabulary.

The application will be developed using Flutter framework for frontend, and a microservices architecture which uses several different frameworks for different services in the backend. Also, several different technologies such as AWS and Docker will be used in the project.

In order to achieve the security of the user data, all accounts will be authenticated. A further explanation of the requirements of the application will be given in the next chapters.

## 2.2 Functional Requirements

**Login & Sign Up:** Users will be able to sign up to the application by providing an email and password or using their Google accounts. We will send verification mails to users to prevent bot/spam accounts. Name of the user and language they want to learn will be asked from the user when they sign up. After the user signs up, they will always be logged in to the app from the smartphone they are using. They will be asked to log in again if they sign out, or use a different device to use the app.

**Show Profile:** Users will be able to see their profile by clicking the profile icon in the navbar. Profile of the user will contain the achievements they got, how many hours they spent on the app, and how many words they have learned.

**Watch Video with Subtitles:** The main page of the app will be "display video" screen. Also, the user will be able to navigate to that screen from another screen by clicking the video icon from the navbar. A video will be displayed to the user according to the user's unknown words and their current language level. Our recommendation engine will find that video by analyzing the words the user doesn't know. Sentences in that video will contain a mixture of unknown and known words such that the user will be able to infer the meaning of the unknown words from the context. Also, each video will have subtitles in the language the user wants to learn. While the video is being watched, subtitles of the video will flow at the bottom half part of the screen. The user can tap the words they do not know in the subtitles while the video runs. When a word is tapped, the video will stop and the dictionary meaning of the tapped word will be displayed to the user so that the user can learn the meaning of the word. The user can continue to watch the video whenever he/she wants. Tapped words will be saved to the "unknown words" list, so that they can be teached the user later.

**Display Learning Progress:** The user will be able to see their individual progress by tapping the "Progress" icon in the navbar. Progress section will mostly consist of the words. Words that have been learned by the user, and words that are unknown to the user will be shown in that section, so that the user can see their overall progression in that language.

**Cloze Test:** Our program will have a "Cloze Test" section, which users can reach by clicking the test icon on the navbar. In those cloze tests, words that are being learned by the user will be asked, so that the user can reinforce their overall understanding about that word. Sentences that will be asked to the user will be gathered from both the video content and sources like Tatoeba.

**Notification System:** In order to apply spaced repetition system in our app, the user will be notified according to the learning intervals to take cloze tests about the words they are learning. This notification system will keep the users in contact with the app, and will maximize the retention rates of the words they are learning.

**Personalized Dictionary System:** The user might want to learn a word that he/she did not encounter in one of our videos. So, the user can enter that word to the dictionary, see its meaning and automatically add that word to its unknown words list. Also, when any word is clicked in that dictionary, its meaning and example uses will be displayed to the user.

## 2.3 Non-functional Requirements

### 2.3.1 Usability

The UI of the application will be Turkish for the initial users to understand. Additional languages will be supported as the application user base grows. Also, the user interface has to be very easy to understand both in order to make the user learn quicker and also for better user experience.

### 2.3.2 Scalability

Even though the application will start with very few users, it is intended to have many. In this case, the servers must be able do the complex tasks of video finding for every user separately.

### 2.3.3 Performance

Even though our servers will not be finding videos corresponding to the users' "words to learn list" in real time, performance is still an issue. Firstly, the video and the subtitles must be synchronous. Also, the app must have a low response time in order to achieve a good user experience.

### 2.3.4 Efficiency

Apps with video content can use big amounts of energy. The application must be energy efficient both for the environment and the user device's battery. In order to do that, the application will not perform the calculation for finding videos on device but rather on our server and only send the corresponding video information to the device.

### 2.3.5 Security

Even though the data that this app gathers is not sensitive, it is still user data and needs to be protected. To do that, we will use industry standard technologies in our servers and provide user authentication.

### 2.3.6 Extensibility

Even though Lingui will start its lifecycle as a mobile app, it may also have a website and/or desktop application. Also, the app will probably grow with additional features. In order to achieve those, the documentation needs to be systematic and open to change, and platform specific tools should be used as little as possible during development. The source code will be written in a tidy manner, refactored as much as possible and comments will be added when necessary..

#### 2.3.7 Accessibility

Lingui will be available both on Google Play Store and App Store. The language of the app will be in Turkish in order for our initial users to use it easily. As the app grows, additional language support will be added. Also, the interface design of the app favors easy usage.

#### 2.3.8 Maintainability

The features given in in extensibility (refactoring, comments, etc.) will also provide maintainability for the application.

#### 2.3.9 Legality

The user data gathered from the application will only be used in our own servers for transaction related to our application and will not be shared with third parties.

### 2.4 Pseudo Requirements

#### 2.4.1 Implementation Constraints

- Lingui will be developed as a mobile application.
- Lingui will be available on both iOS and Android platforms.
- Flutter framework will be used for mobile application.
- All members will be using VS Code with linter extensions to maintain code quality.
- Git and GitHub will be used for version control and collaboration.
- Backend will be developed with micro service architecture.
- AWS services for databases, gateways, authentication will be used for supporting the backend micro services.
- Frontend of the application will contact only a single point for communication with microservices.
- Docker will be used for building, deploying and networking of services.
- Services will be deployed on EC2 instance(s).
- Each service will be put inside a Docker container.
- Services can be implemented in any language that the team member developing it is comfortable with.
- For databases, fully hosted database services from AWS will be used.
- General architecture of the project will aim for high availability, low latency and ease of extension for future changes.

2.4.2 Economic Constraints

- Lingui will be free in the first place with running ads to make up for the cloud expenses.
- Algorithms and architecture will aim to stay on free trial of services used in AWS.
- Free libraries and tools will be used.
- The web page of the application will be on the free GitHub domain.

2.4.3 Ethical Constraints

- User data should be in safe hands.
- Users will have multi factor authentication.
- User data will not be shared with third party platforms.

2.4.4 Sustainability Constraints

- User feedback will be considered for future improvements.

2.4.5 Social Constraints

- The main user group of the application will be Turkish students and white collar workers aiming to improve their English skills.

2.4.6 Reliability Constraints

- Since the application will only be used by end users, down time has to be minimized.
- Even though some services may be down, the mobile application will try its best to serve most of the functionality available.
- Mobile application, microservices and modules in each microservice will be tested by end to end testing and unit testing.

2.4.7 Language Constraints

- Mobile application's user interface will be in Turkish.

## 2.5 System Models

2.5.1 Scenarios

2.5.1.1 Sign Up

Actors: Language Learner

Entry Condition: User opens the application

Exit Condition: User is navigated to the main menu OR user closes the application

Flow of Events:

1. User clicks the signup button
2. The signup page opens
3. User enters their personal information such as username, email address and password
4. User clicks the signup button
5. A verification email is sent to the email address that the user provided
6. User uses the verification code sent in the email to fill the verification code area
7. User clicks the verify button on the verification page
8. The system creates a new account with the given information if the code is correct.
9. User is navigated to the initial quiz

2.5.1.2 Log In

Actors: Language Learner

Entry Condition: User opens the application

Exit Condition: User is navigated to the main menu OR user closes the application

Flow of Events:

1. User clicks the login button
2. The login page opens
3. User enters their email or username and password
4. User clicks the login button
5. The system checks the correctness of the email and password with the database
6. User is navigated to the main menu

2.5.1.3 Take Initial Test

Actors: Language Learner

Entry Condition: User is registered OR user opens logs in for the first time

Exit Condition: User completes the test OR user exits the test OR user closes the application

Flow of Events:

1. The app becomes horizontal
2. User is presented with five short videos on one side of the screen and the transcript on the other side of the screen

3. The user clicks the words they think they do not know
4. A brief explanation of the word comes up
5. User either clicks add word to unknown list button or cancel button.
6. User continues watching the videos

2.5.1.4 Watch Videos

Actors: Language Learner

Entry Condition: User clicks the video button on the navigation bar

Exit Condition: User closes the video OR user closes the application

Flow of Events:

1. The app becomes horizontal
2. User is presented with a video according to their word lists on one side of the screen and the transcript of the video on the other side

2.5.1.5 Watch Specific Video

Actors: Language Learner

Entry Condition: User clicks on one of the words in the word lists AND selects watch video option

Exit Condition: User closes the video OR user closes the application

Flow of Events:

1. The app becomes horizontal
2. User is presented with a video consisting the specific word they have chosen on one side of the screen and the transcript of the video on the other side

2.5.1.6 Add Word to Unknown List

Actors: Language Learner

Entry Condition: User is watching a video

Exit Condition: User is navigated to the main menu OR user closes the application

Flow of Events:

1. User hears an unfamiliar word in the video
2. User checks the word in the transcript
3. User clicks the word in the transcript

4. A brief explanation of the word comes up
5. User either clicks add word to unknown list button or cancel button
6. User continues watching the videos

2.5.1.7 Take Cloze Test

Actors: Language Learner

Entry Condition: User clicks the quiz button on the navigation bar

Exit Condition: User closes the quiz OR user closes the application

Flow of Events:

1. User is presented with “fill in the blank” questions according to their word lists
2. User writes down his answers in the blanks
3. User clicks check results button
4. User is presented with the results of the quiz
5. User is also suggested to switch some words between the word lists according to their results. (i.e. transfer a word from unknown list to known list if it is known or vice versa if the questions weren’t answered correctly)

2.5.1.8 See Progress

Actors: Language Learner

Entry Condition: User clicks the progress button

Exit Condition: User goes to another page OR user closes the application

Flow of Events:

1. User is presented with an overall progress they had consisting of the words they have learnt, are in progress of learning and do not know.

2.5.1.9 Use Dictionary

Actors: Language Learner

Entry Condition: User clicks the dictionary button

Exit Condition: User goes to another page OR user closes the application

Flow of Events:

1. User searches a word they are curious about in the dictionary search bar

2. The word was searched in the database, if it does not exist an error message is given
3. The word and its meaning is presented to the user
4. User checks the meaning of the word
5. If the user does not know the word, they add it to the unknown list
6. User is returned to the dictionary page and can continue searching other words

2.5.1.10 See Profile

Actors: Language Learner

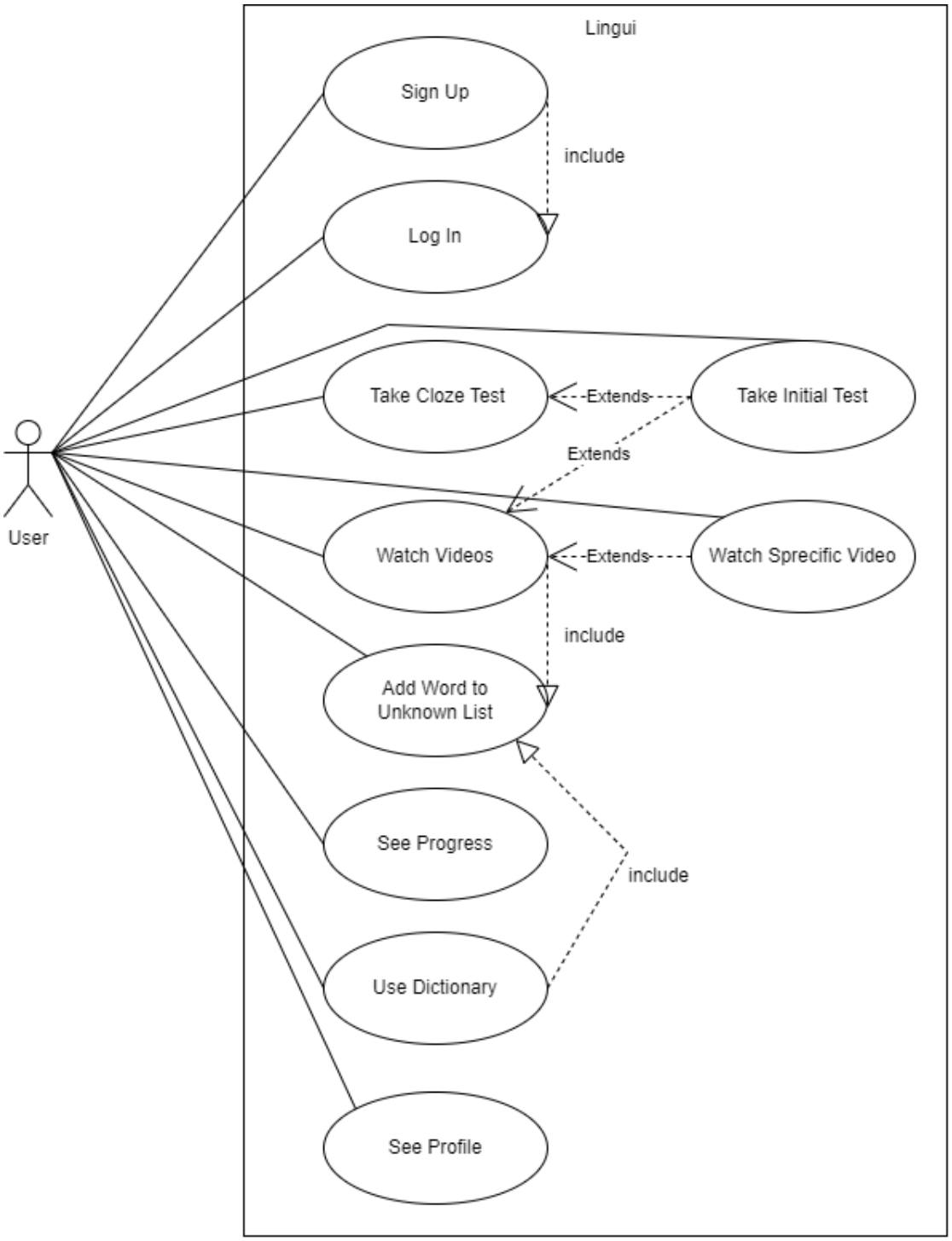
Entry Condition: User clicks the profile button on the navigation bar

Exit Condition: User goes to another page OR user closes the application

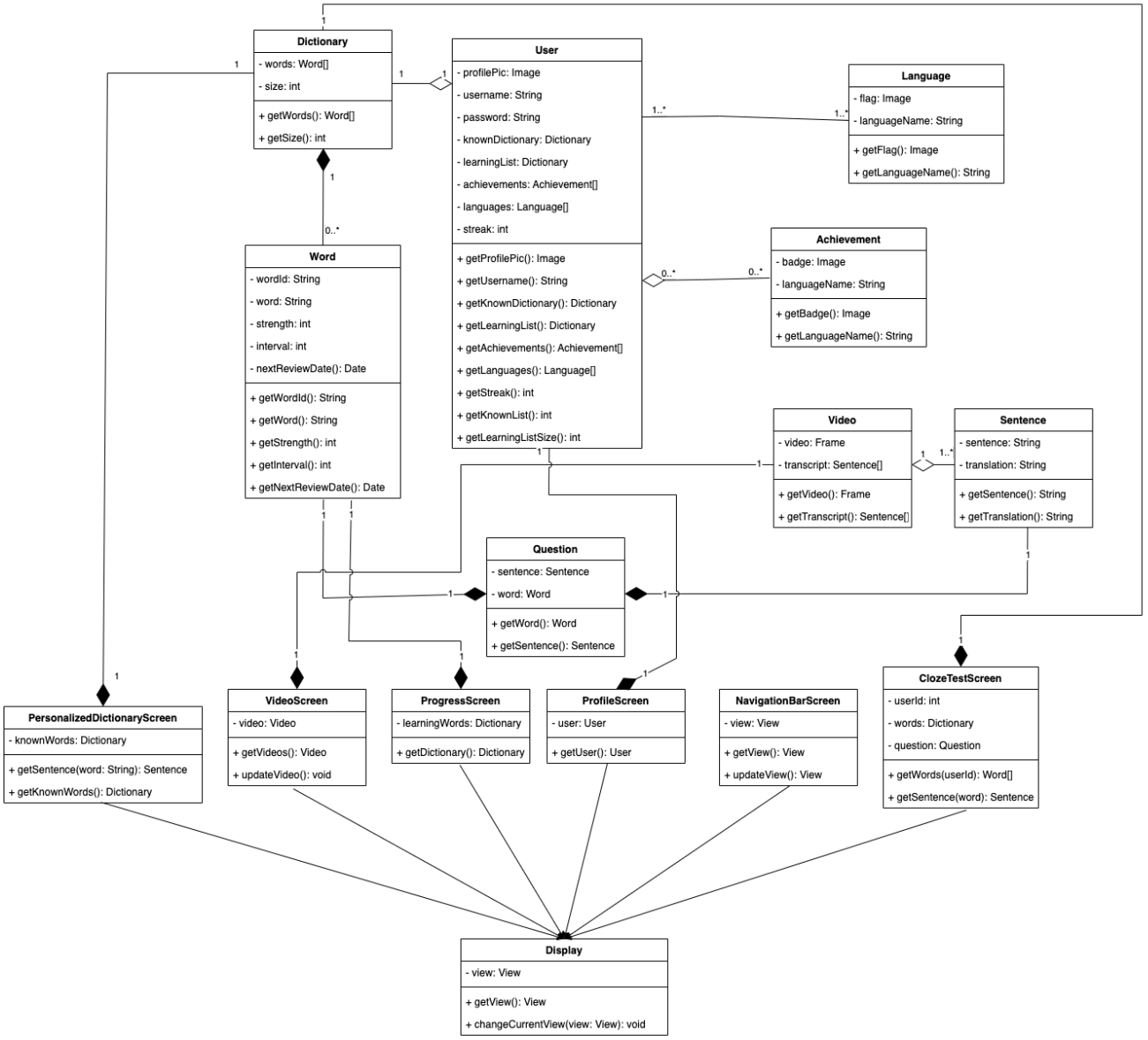
Flow of Events:

1. Users profile data such as username and email is displayed as well as their statistics like the number of hours they have spent in the application, their achievements and the number of words they have learnt

### 2.5.2 Use-Case Model

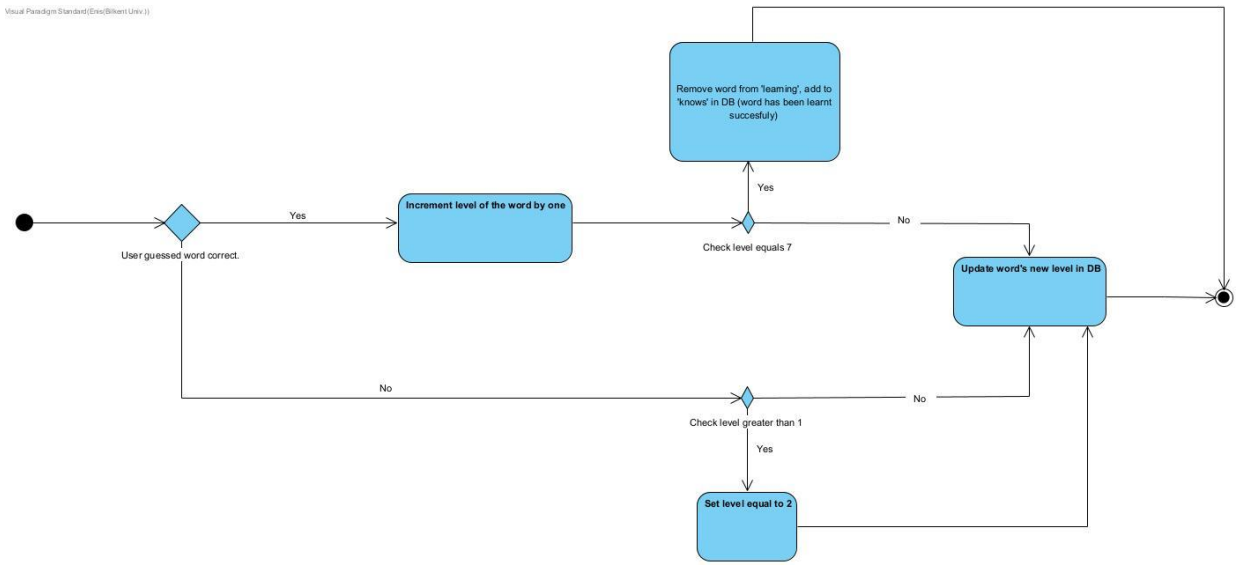


### 2.5.3 Object and Class Model

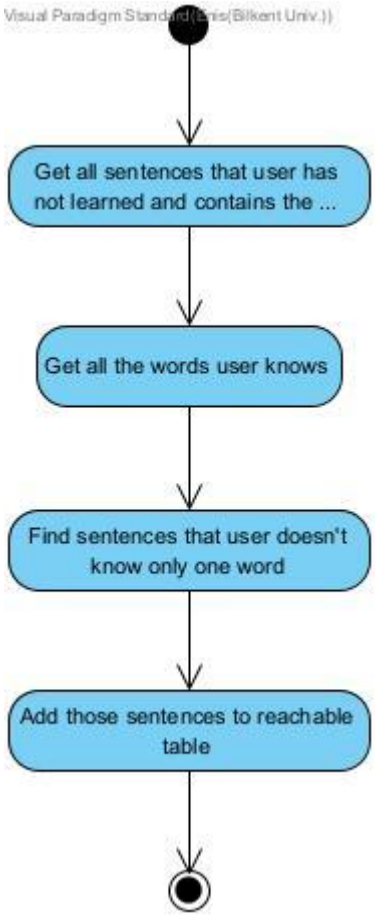


### 2.5.4 Dynamic Models

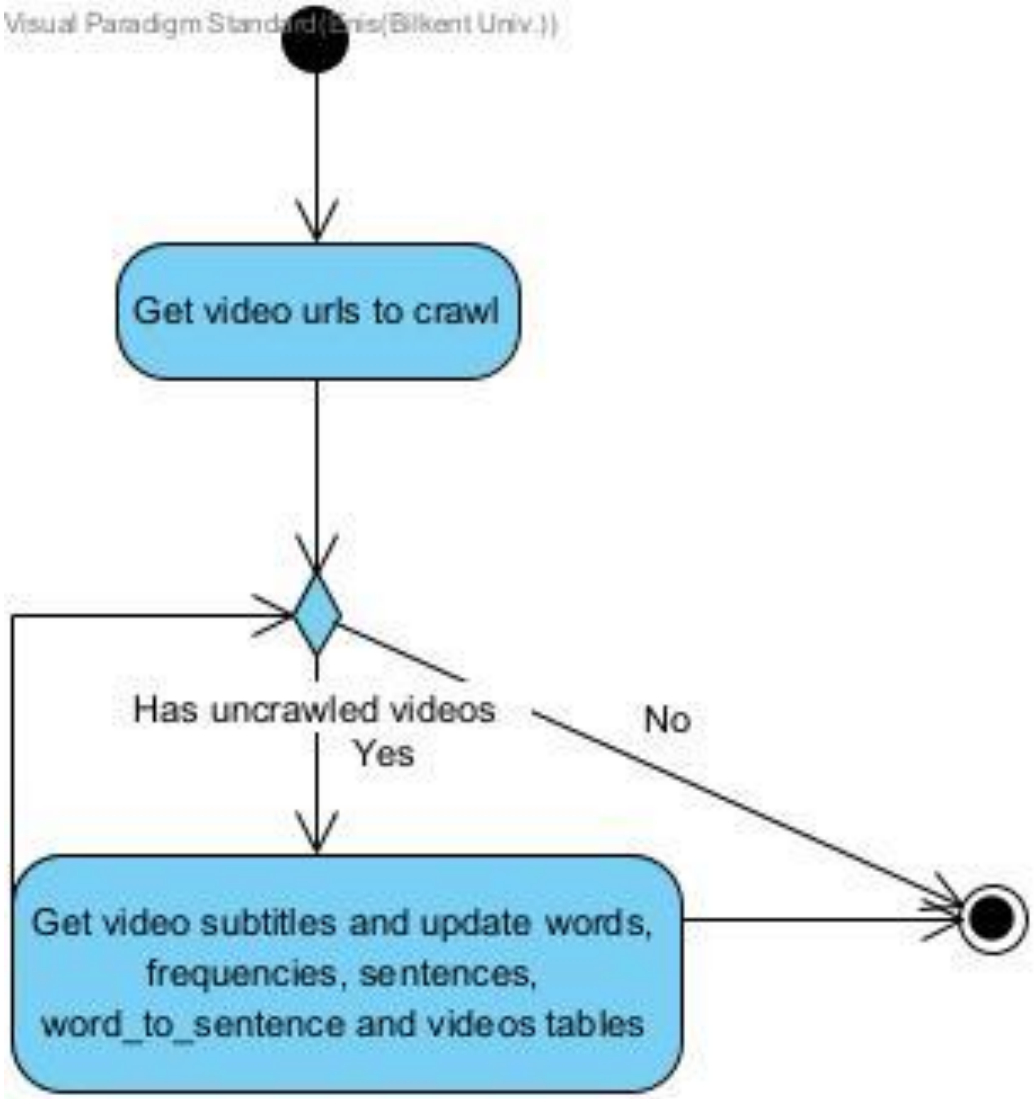
Activity diagram for the spaced repetition systempaced Repetition System:



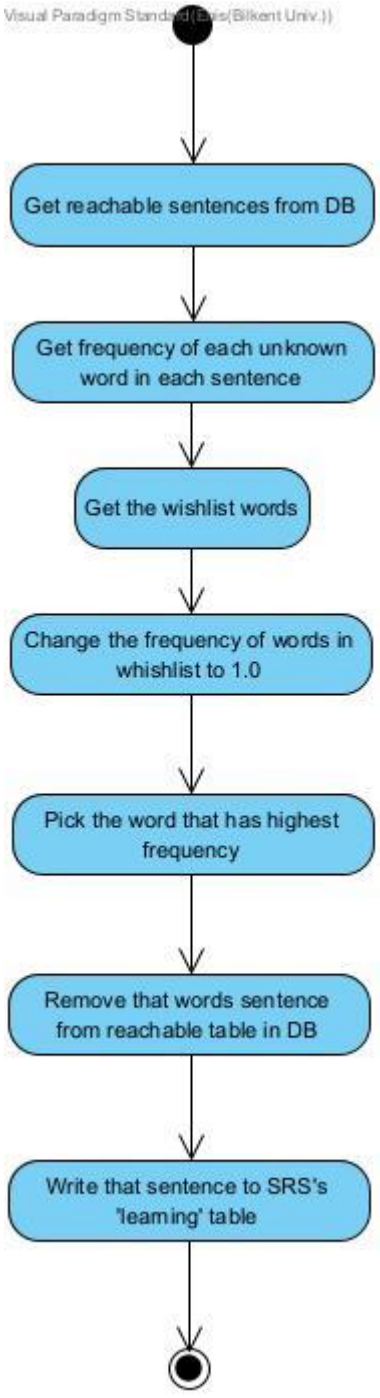
Update of reachable table for a user by Table Manager service:



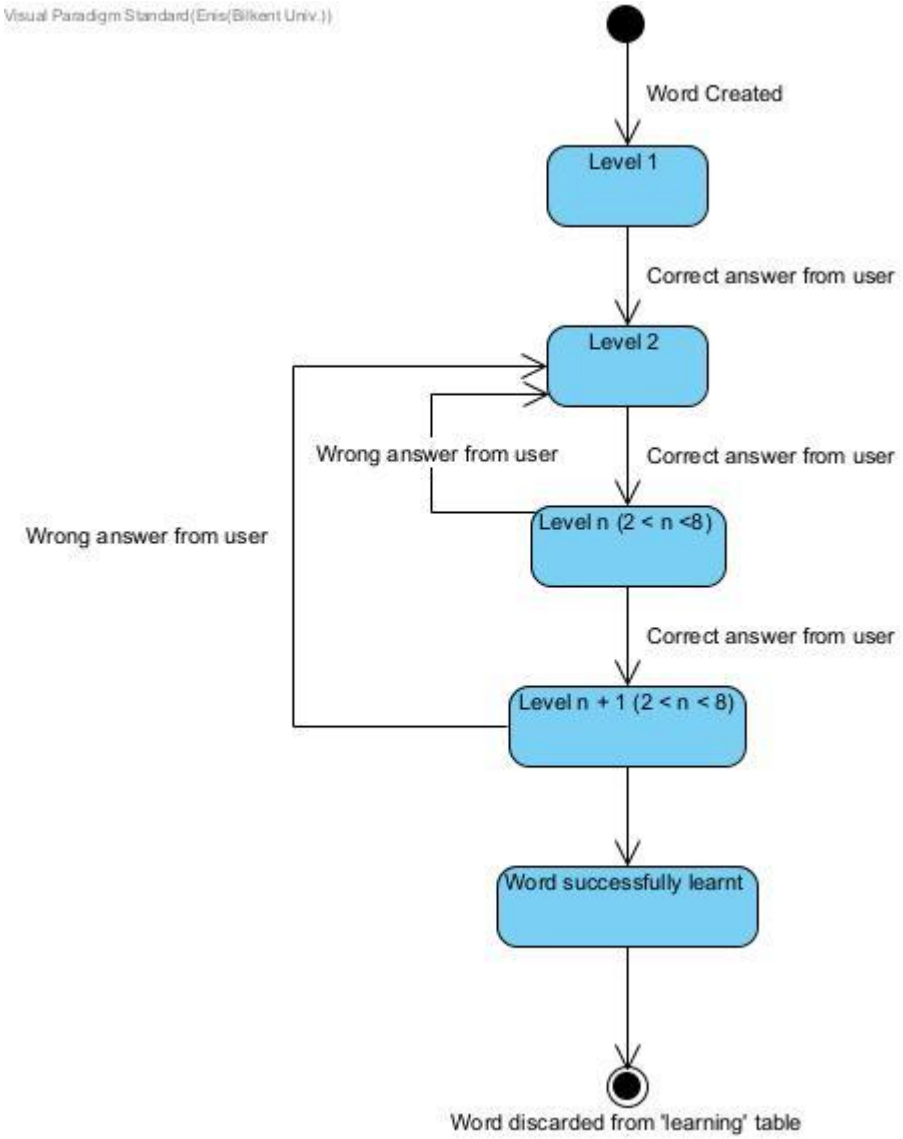
Crawler crawling videos subtitles and updating DB tables activity diagram:



Activity diagram of Feeder service finding a sentence that is optimal for the user:

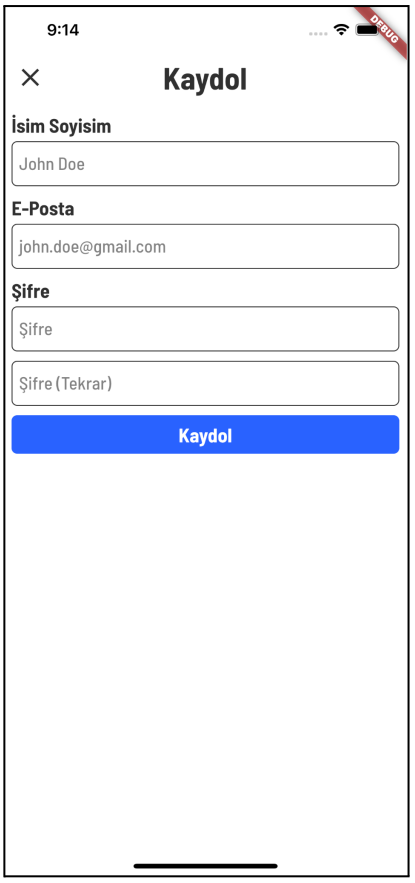


State machine diagram of a word that is currently being learnt by a user:

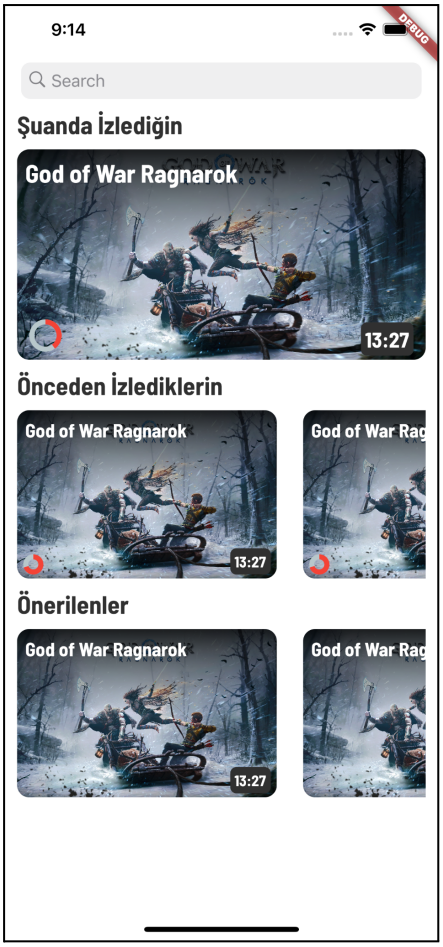


### 2.5.5 User Interface

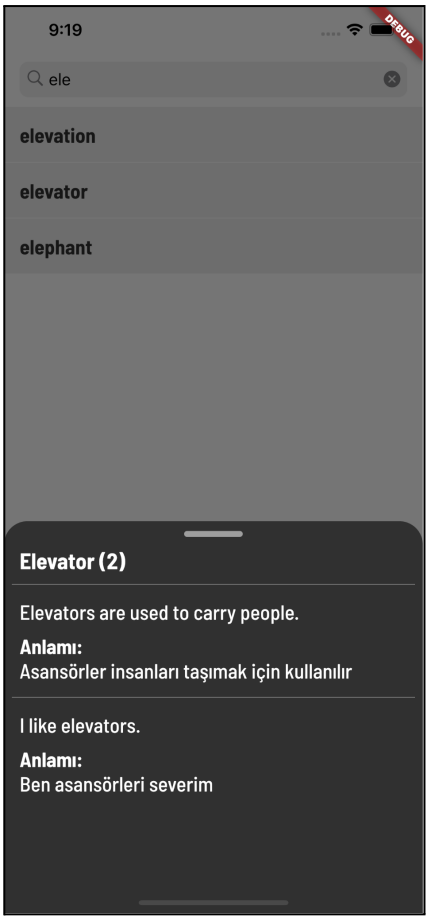

*Log In Screen*



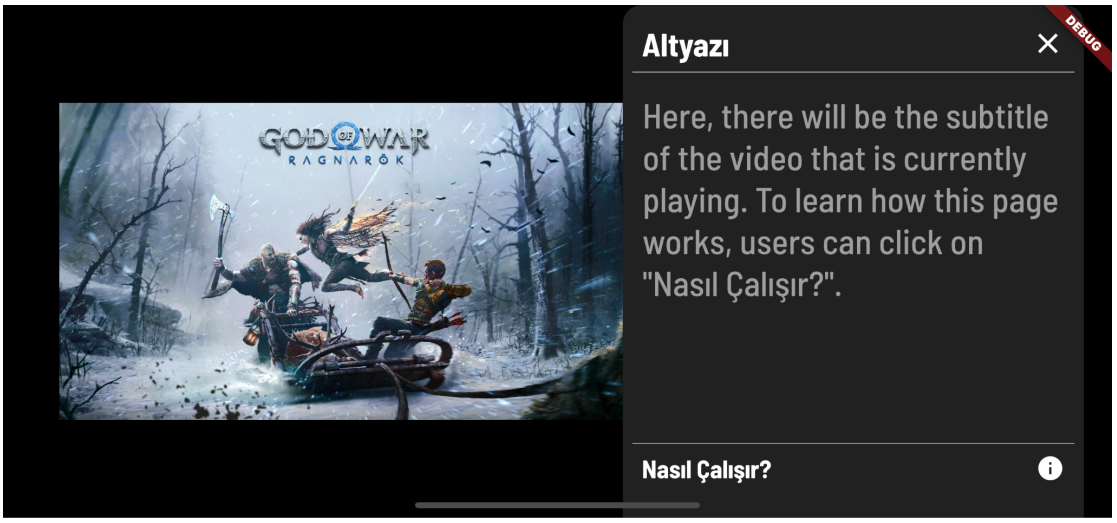

*Sign Up Screen*




*Videos Screen*

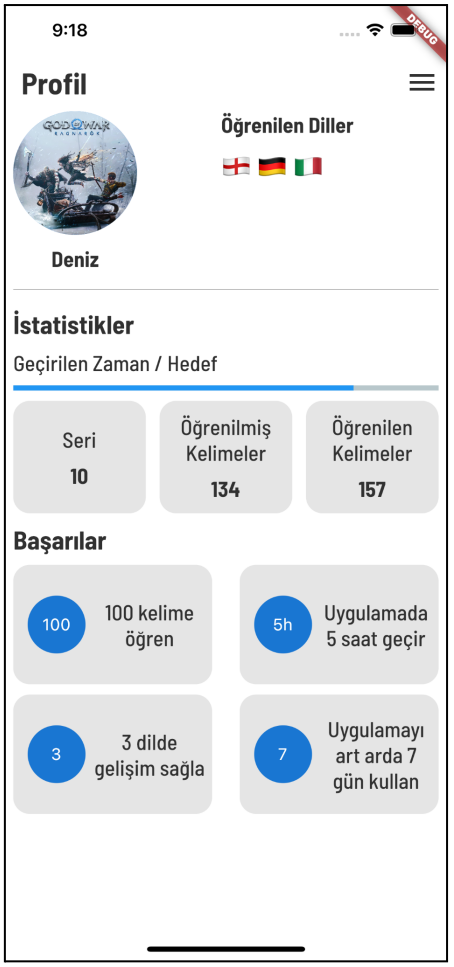



*Dictionary Screen*




*Video Play Screen*


*Progress Screen*




*Profile Screen*

# 3 Other Analysis Elements

## 3.1 Consideration of Various Factors in Engineering Design

Our system's main purpose is to teach English to the native Turkish speakers (initially). Since language learning is a domain that is not really affected by the general ongoing of a person's daily life, our system does get affected on things like public health and safety. Public welfare can affect our system, because if public welfare is too low, learning a language might not be the first priority of a person. So, our system should be able to run on old phones and should not consume too much internet. Global factors do not really affect our system, but English being the global language of the world makes our app more desirable. In the Turkish education system, English is taught by a hard emphasis on rules, grammar and test questions. Since education is strongly connected with cultural and social factors, this is the main paradigm in Turkish society. We want to change the emphasis on rules and grammar with an exposure system, but to keep the cultural familiarity, we will keep questions in the form of cloze tests. Also, socially, people are more inclined to watch short videos because of the decline of the attention span. Thus, we might want to keep our videos short.

Table 1: Factors that can affect analysis and design.

| | Effect level | Effect |
|---|---|---|
| Public health | 0 | Public health has no effect on our system. |
| Public safety | 0 | Public safety has no effect on our system. |
| Public welfare | 5 | Public welfare can affect the usage of our system because if the general welfare is very low, learning a new language might not be the priority of the people. |
| Global factors | 4 | English being the global language of the world can attract/lure users to our system because it is a desired language to learn. |
| Cultural factors | 6 | In the Turkish education system, English is not taught as we will teach in our system. Since education and culture are closely related, it might take a time for users to get used to our system. |
| Social factors | 6 | Since people like short videos more due to the decline of the attention span, we might choose videos with short durations. |

### 3.2 Risks and Alternatives

We want to present videos to the user which include the user's personal interests. If we fail to do so, users might get random videos that do not interest them, which can lead users to exit the app. Our B plan is to handpick videos that users generally like, from the very popular YouTube channels like Kurzgesagt. Another risk we envision is to not being able to create a word dictionary to each user due to the storage overhead. This affects our project really badly, because this is our main marketing point. Our plan B is to maybe select 1000-2000 main popular words to teach to all of the users. Lastly, we will use third party services to get the subtitles of a video. If that service closes down, we have to write our own web-scraping solution to get the subtitles. If we fail to do so, our entire project gets useless since we have no subtitles.

Table 2: Risks

| | Likelihood | Effect on the project | B Plan Summary |
|---|---|---|---|
| Risk: Not being able to recommend videos with the interest of the user | Low | Users might be served with the videos that do not interest them, which can lead to user loss. | Selecting generally liked videos from the very popular YouTube channels. |
| Risk: Not being able to create a dictionary for every user | Low | Our main marketing point, “personalized learning” takes damage and our effectiveness gets down. | Creating a 1000-2000 dictionary with popular words to teach the user. |
| Risk: Closure of subtitle scraping service | Very Low | Our app entirely becomes useless due to lack of subtitles. | We have to write our own web-scraping service to get subtitles. |

### 3.3 Project Plan

Table 3: List of work packages

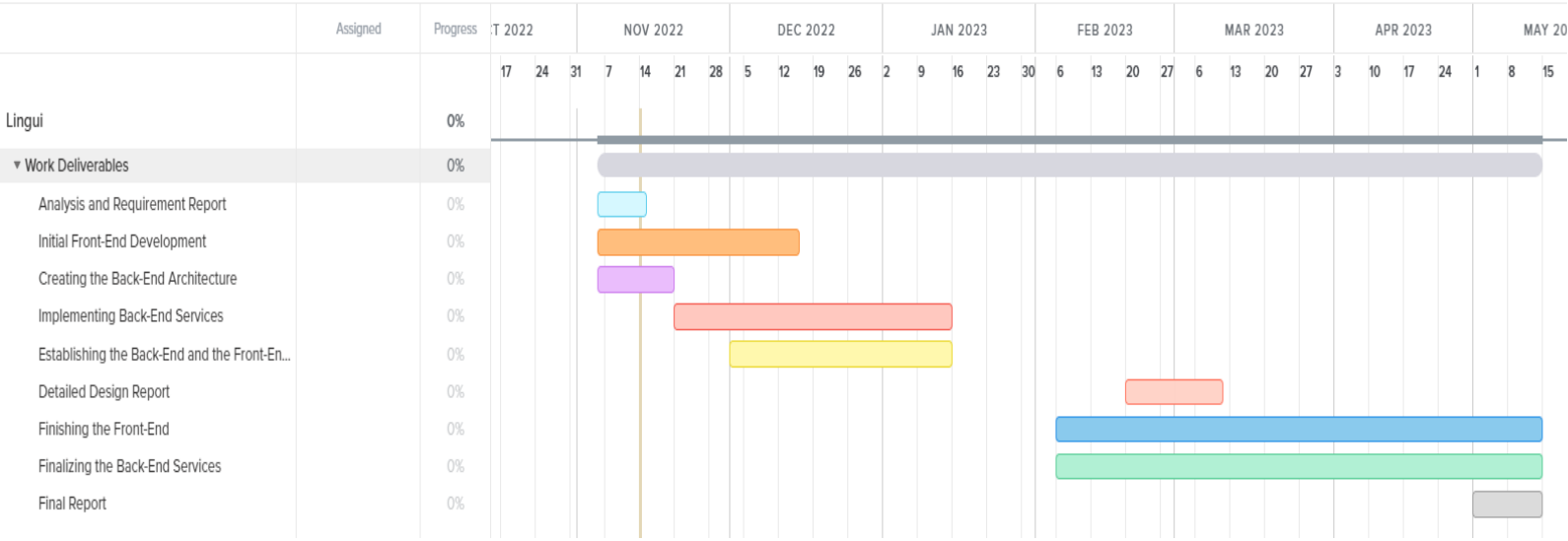
| WP# | Work package title | Leader | Members involved |
|---|---|---|---|
| WP1 | Analysis and Requirements Report | Oğuz | Emirhan,Oğuz, Deniz, Olcaytu, Enis |
| WP2 | Initial Front-End Development | Deniz | Deniz, Emirhan |
| WP3 | Creating Back-End Architecture | Enis | Enis, Oğuz, Olcaytu, Emirhan |
| WP4 | Implementing Back-End Services | Enis | Enis, Oğuz, Olcaytu, Emirhan |
| WP5 | Establishing the Back-End and Front-End Connection | Deniz | Deniz, Enis, Oğuz, Olcaytu, Emirhan |
| WP6 | Detailed Design Report | Olcaytu | Emirhan,Oğuz, Deniz, Olcaytu, Enis |
| WP7 | Finishing the Front-End | Deniz | Deniz, Emirhan |
| WP8 | Finalizing the Back-End Services | Enis | Enis, Oğuz, Olcaytu, Emirhan |
| WP9 | Final Report | Emirhan | Emirhan,Oğuz, Deniz, Olcaytu, Enis |

| | | | |
|---|---|---|---|
| **WP 1:** *Analysis and Requirements Report* | | | |
| **Start date:** *05.11.2022* **End date:** *13.11.2022* | | | |
| **Leader:** | *Oğuz* | **Members involved:** | *Emirhan, Oğuz, Deniz, Olcaytu, Enis* |
| **Objectives:** *Objective is to finish the Analysis and Requirements Report and submit it to the course instructors.* | | | |
| **Tasks:**<br>***Task 1.1 Introduction*** **:** *Finish the "Introduction" part.*<br>***Task 1.2 System Requirements*** **:** *Decide the system requirements and finish that part.*<br>***Task 1.3 Diagrams*** **:** *Draw the class, use case and dynamic diagrams.*<br>***Task 1.4 Other Analysis Elements*** **:** *Finish the "Other Analysis Elements" part.* | | | |
| **Deliverables**<br>***D1.1:*** *Analysis and Requirements Report* | | | |
| **WP 2:** *Initial Front-End Development* | | | |
| **Start date:** *05.11.2022* **End date:** *15.12.2022* | | | |
| **Leader:** | *Deniz* | **Members involved:** | *Deniz, Emirhan* |
| **Objectives:** *Start the Front-End development of the project. Decide on the initial design, and how to implement it. Create necessary pages in our app.* | | | |
| **Tasks:**<br>***Task 2.1 Deciding Design*** **:** *Briefly draw the UI mock-ups.*<br>***Task 2.2 Coding the Home Page*** **:** *Implement the home page of the app.*<br>***Task 2.3 Coding the Video Page*** **:** *Implement the watching video page of the app.*<br>***Task 2.4 Coding the Cloze Test Page*** **:** *Implement the cloze test page of the app.* | | | |
| **Deliverables**<br>***D2.1:*** *UI mock-ups.*<br>***D2.2:*** *Home Page UI.*<br>***D2.3:*** *Video Page UI.*<br>***D2.4:*** *Cloze Test Page UI.* | | | |
| **WP 3:** *Creating Back-End Architecture* | | | |
| **Start date:** *05.11.2022* **End date:** *18.11.2022* | | | |
| **Leader:** | *Enis* | **Members involved:** | *Enis, Oğuz, Olcaytu, Emirhan* |
| **Objectives:** *Initial services of the back-end will be decided. Code base of the back-end will be deployed. Services will be assigned to the members.* | | | |
| **Tasks:**<br>***Task 3.1 Architecture Design*** **:** *Architecture of the back-end will be designed.*<br>***Task 3.2 Code Base Deployment*** **:** *Code base for the back-end will be deployed on GitHub.*<br>***Task 3.3 Service Assignment*** **:** *Services will be assigned to the back-end members.* | | | |
| **Deliverables**<br>***D3.1:*** *Back-End Architecture*<br>***D3.2:*** *Back-End Code Base* | | | |
| **WP 4:** *Implementing Back-End Services* | | | |
| **Start date:** *18.11.2022* **End date:** *End of the First Semester* | | | |

| **Leader:** | *Enis* | **Members involved:** | *Enis, Oğuz, Olcaytu, Emirhan* |
|---|---|---|---|
| **Objectives:** *Services of the back-end will be implemented. Services will be connected with Docker. Services will be tested.* | | | |
| **Tasks:**<br>***Task 4.1 Service Implementation*** **:** *Each member will implement their own services. Those services are Recommendation, Spaced Repetition System, Cloze Tests, User Profile Data, and Personalized User Dictionary. Main task is to finish as much as possible before CS492.*<br>***Task 4.2 Connect Services*** **:** *Services will be connected with Docker. After those services are connected, we will connect whole system to the database.*<br>***Task 4.3 Testing*** **:** *Services will be tested with unit-tests.* | | | |
| **Deliverables**<br>***D4.1:*** *Service Codes*<br>***D4.2:*** *Docker Containers*<br>***D4.3:*** *Unit-Test Code* | | | |
| **WP 5:** *Establishing the Back-End and the Front-End Connection* | | | |
| **Start date:** *01.12.2022* **End date:** *End of the First Semester* | | | |
| **Leader:** | *Deniz* | **Members involved:** | *Deniz, Enis, Olcaytu, Oğuz, Emirhan* |
| **Objectives:** *Back-end and the front-end will be connected. Back-end services will be tested on the front-end.* | | | |
| **Tasks:**<br>***Task 5.1 Establish Connection*** **:** *Connection of the back-end and the front-end will be established (data and the service flow).*<br>***Task 5.2 Testing Services*** **:** *Each service and the UI will be tested.* | | | |
| **Deliverables**<br>***D5.1:*** *A Partially Working App*<br>***D5.2:*** *Established Data Flow Between the Front-End and the Back-End* | | | |
| **WP 6:** *Detailed Design Report* | | | |
| **Start date:** *2nd Week of the Spring Semester* **End date:** *3rd Week of the Spring Semester* | | | |
| **Leader:** | *Olcaytu* | **Members involved:** | *Deniz, Enis, Olcaytu, Oğuz, Emirhan* |
| **Objectives:** *Final Design will be decided, and the report will be delivered.* | | | |
| **Tasks:**<br>***Task 6.1 Decide Final Design*** **:** *Based on the experiences and the feedback we have on the demo, finish the final design.*<br>***Task 6.2 Finish the Report Parts*** **:** *Assign the report parts to the members.* | | | |
| **Deliverables**<br>***D6.1:*** *A detailed final app design.*<br>***D6.2:*** *Detailed Design Report.* | | | |
| **WP 7:** *Finishing the Front-End* | | | |
| **Start date:** *Start of the Spring Semester* **End date:** *End of the Spring Semester* | | | |
| **Leader:** | *Deniz* | **Members involved:** | *Deniz, Emirhan* |
| **Objectives:** *Based on the feedbacks from the users and the inspection, we will finalize the front-end development of our app.* | | | |
| **Tasks:** | | | |

| | | | |
|---|---|---|---|
| ***Task 7.1 Review Feedback*** **:** *Get and review feedback from the users.*<br>***Task 7.2 Finalize and Develop Front-End*** **:** *Finalize the design and finish each of the pages of our UI.* | | | |
| **Deliverables**<br>***D7.1:*** *Finalized User Interface of Our App* | | | |
| **WP 8:** *Finalizing the Back-End Services* | | | |
| **Start date:** *Start of the Spring Semester* **End date:** *End of the Spring Semester* | | | |
| **Leader :** | *Enis* | **Members involved:** | *Enis, Oğuz, Olcaytu, Emirhan* |
| **Objectives:** *Based on the tests and the feedback we receive, we will finalize the back-end services of our project.* | | | |
| **Tasks:**<br>***Task 8.1 Receive feedback*** **:** *We will receive feedback from our users and the course instructors about our back-end.*<br>***Task 8.2 Finish the Back-End Features*** **:** *We will finish our uncompleted features from the first semester and add any additional ones if necessary.* | | | |
| **Deliverables**<br>***D8.1:*** *Finished back-end services that are connected to each other, and the database.* | | | |
| **WP 9:** *Final Report* | | | |
| **Start date:** *12th Week of the Spring Semester* **End date:** *13th Week of the Spring Semester* | | | |
| **Leader :** | *Emirhan* | **Members involved:** | *Deniz, Enis, Olcaytu, Oğuz, Emirhan* |
| **Objectives:** *Based on the finished project, we will prepare our final report.* | | | |
| **Tasks:**<br>***Task 9.1 Distribute the Parts to the Members*** **:** *Each member will receive its own part.*<br>***Task 9.2 Finishing the Parts*** **:** *Parts will be finished by their respective members.* | | | |
| **Deliverables**<br>***D9.1:*** *A finished Final Report.* | | | |

**Gantt Chart of Our Work Deliverables**




## 3.4 Ensuring Proper Teamwork

To ensure proper teamwork, we embraced the "from each according to his ability, to each according to his needs" principles. Members that are experienced in and want to pursue a career in the front-end were assigned to the front-end part of the project, and members that are experienced in and want to pursue a career in the back-end were assigned to the back-end part of the project. Since everyone is passionate about the career path they chose, proper teamwork and inclusion is nearly totally ensured, because everyone thinks that this project is a good chance to fill their CV's with the technologies they want to work in their professional life. Also, we will use collaboration tools like GitHub, which will allow us to see each member's contribution to the project. If there is a general consensus about something like "a team member is slacking", we can talk to that person since we have been friends for four years. Also, we always try to arrange meetings when everyone is available so that no member feels left out from the project, and we always decide on something collaboratively.

## 3.5 Ethics and Professional Responsibilities

Our system does not really pose ethical challenges to us, since we are just a language learning app. The only thing we considered was whether to show foul words to the user or not. We decided not to show them, because each person's ethical values are different and they might feel uncomfortable from those words. Also, each team member has ethical and professional responsibilities to the team. Each member is obliged to attend the team meetings and finish the work packages assigned to them. Even though we are long time friends, it is easy to get heated up during any argument. So, every member should act professional during the team decisions and meetings.

### 3.6 Planning for New Knowledge and Learning Strategies

Even though the team members are experienced in the parts that they are assigned, each member has a lot to learn in the course of the project. Some members learned new programming languages from Udemy for the front-end development. A member in our project is experienced in microservice architecture, so he is teaching it to the rest of the back-end team. We will use Docker in the back-end, so all of the back-end members will start to learn Docker online very soon. Also, generally our team lacks architectural design knowledge for our project. We plan to enhance our architectural knowledge from the meetings with our innovation expert, Haluk Altunel. Back-end team also started to learn Golang from the Udemy, so that we can help each other to debug things.

## 4 References

[1] Krashen, S. (2002). The comprehension hypothesis and its rivals. Selected papers from the Eleventh International Symposium on English Teaching/Fourth Pan Asian Conference. pp. 395-404. Taipei: Crane Publishing Company.

[2] Krashen, S. (2004). Applying the comprehension hypothesis: Some suggestions. Paper presented at 13th international symposium and book fair on language teaching (English Teachers Association of the Republic of China), Taipei, Taiwan. Retrieved on 17 Oct 2022 from http://www.sdkrashen.com/content/articles/eta_paper.pdf.

[3] Rodrigo, V (2006). The amount of input matters: Incidental acquisition of grammar through listening and reading. The International Journal of Foreign Language Teaching, 2 (1). 10-13.

[4] Ponniah, R. J. (2008). Free voluntary reading and the acquisition of grammar by adult ESL students. The International Journal of Foreign Language Teaching, 4 (1) 20-22.