Bilkent University
Department of Computer Engineering

Senior Design Project

2336

Lingui

Final Report

21902051, Mustafa Oğuz DUMAN, oguz.duman@ug.bilkent.edu.tr
21902356, Deniz Berkant DEMİRÖRS, berkant.demirors@ug.bilkent.edu.tr
21903103, Emirhan Nadir KARAMAN, nadir.karaman@ug.bilkent.edu.tr
21903031, Olcaytu GÜRKAN, olcaytu.gurkan@ug.bilkent.edu.tr
21901789, Enis ÖZER, enis.ozer@ug.bilkent.edu.tr

Uğur GÜDÜKBAY

Erhan DOLAK
Tağmaç TOPAL
Selim AKSOY

19.05.2023

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

Final 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 Requirements Analysis

2.1 Functional Requirements

Login & Sign Up: Users will be able to sign up for the application by providing an email and password or using their Google accounts. We will send verification emails to users to prevent bot/spam accounts. The name of the user and the 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. The 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 a “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. The 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.

2.2 Non-Functional Requirements

2.2.1 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.

2.2.2 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.

2.2.3 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.

2.2.4 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.

2.2.5 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.

2.2.6 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.

2.2.7 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.

2.2.8 Maintainability

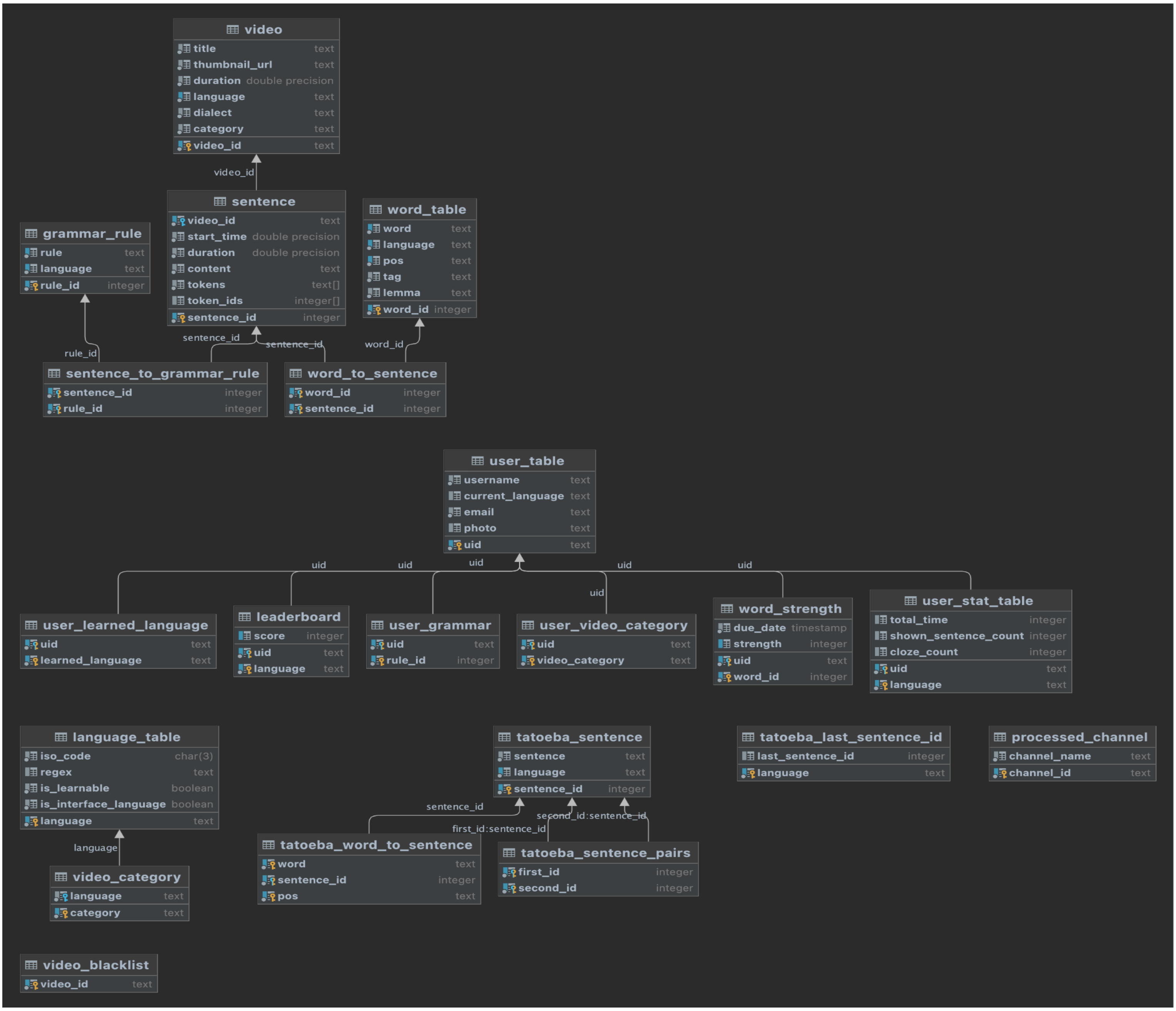
The features given in in extensibility (refactoring, comments, etc.) will also provide maintainability for the application.

2.2.9 Legality

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

3 Final Architecture and Design Details

Firstly, we will describe our objects. Here is our object diagram which we have generated from our PostgreSQL database:



video	
title	text
thumbnail_url	text
duration	double precision
language	text
dialect	text
category	text
video_id	text

sentence	
video_id	text
start_time	double precision
duration	double precision
content	text
tokens	text[]
token_ids	integer[]
sentence_id	integer

grammar_rule	
rule	text
language	text
rule_id	integer

word_table	
word	text
language	text
pos	text
tag	text
lemma	text
word_id	integer

sentence_to_grammar_rule	
sentence_id	integer
rule_id	integer

word_to_sentence	
word_id	integer
sentence_id	integer

user_table	
username	text
current_language	text
email	text
photo	text
uid	text

user_learned_language	
uid	text
learned_language	text

leaderboard	
score	integer
uid	text
language	text

user_grammar	
uid	text
rule_id	integer

user_video_category	
uid	text
video_category	text

word_strength	
due_date	timestamp
strength	integer
uid	text
word_id	integer

user_stat_table	
total_time	integer
shown_sentence_count	integer
cloze_count	integer
uid	text
language	text

language_table	
iso_code	char(3)
regex	text
is_learnable	boolean
is_interface_language	boolean
language	text

tatoeba_sentence	
sentence	text
language	text
sentence_id	integer

tatoeba_last_sentence_id	
last_sentence_id	integer
language	text

processed_channel	
channel_name	text
channel_id	text

video_category	
language	text
category	text

tatoeba_word_to_sentence	
word	text
sentence_id	integer
pos	text

tatoeba_sentence_pairs	
first_id	integer
second_id	integer

video_blacklist	
video_id	text

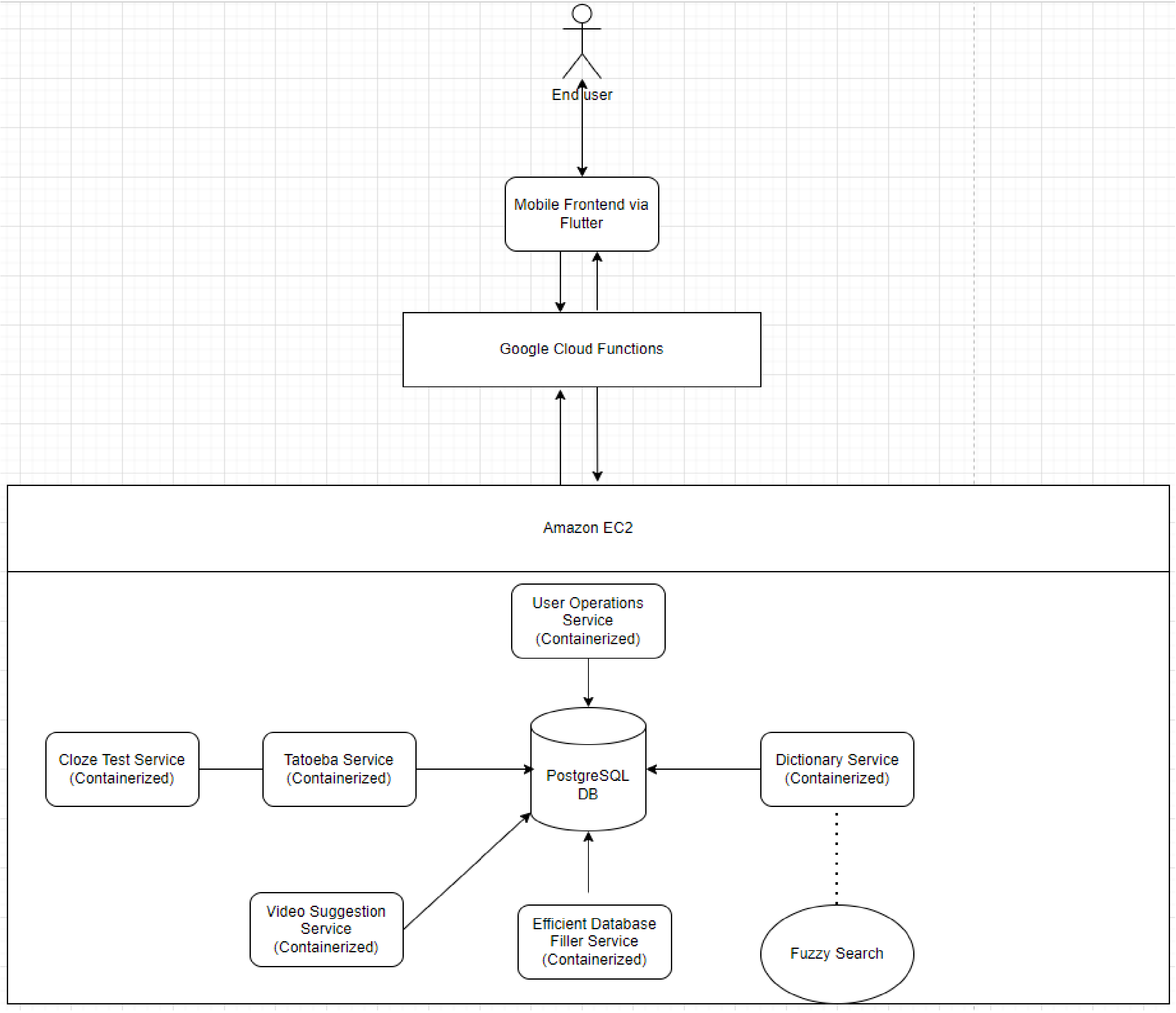
Main element of our database is the *video* table. We hold necessary information about videos in that table in order to reach that video in YouTube in necessity. *sentence* table holds the mapping of the sentence to the respective video in the *video* table. Also, we have mapper tables like *sentence_to_grammar_rule* and *word_to_sentence* which acts as a bridge between different tables.

Our *user_table* contains the information of our users, and many tables are bridged to *user_table* via uid foreign key. We hold information like user stats, leaderboard, user's learned words and their respective strengths, the languages users are learning etc. in those tables.

Also, *tatoeba_sentence, tatoeba_word_to_sentence, tatoeba_sentence_pairs* tables hold the information that we have gathered from the Tatoeba API. We use those tables to present the sentences to our users in our cloze tests.

video_blacklist table holds the videos that we know do not have any subtitles, so we blacklisted them to increase the quality of the user experience. We also hold the processed channels in our *processed_channels* table, in order to quickly skip the channels of which we have processed all videos.

Our architecture diagram:


End user
Mobile Frontend via Flutter
Google Cloud Functions
Amazon EC2
User Operations Service (Containerized)
Cloze Test Service (Containerized)
Tatoeba Service (Containerized)
PostgreSQL DB
Dictionary Service (Containerized)
Video Suggestion Service (Containerized)
Efficient Database Filler Service (Containerized)
Fuzzy Search

End user reaches our app through our frontend, which is written in Flutter. All of our microservices (Cloze Test, Dictionary, Tatoeba, User Operations, Database Filler, and Video Suggestion) are dockerized and deployed to an Amazon EC2 machine via AWS. Also, our database resides in AWS too. Whenever our frontend wants to use a microservice in our backend, it issues a request through Google Cloud Functions. Google Cloud Functions interacts with our services in Amazon EC2, and gets the response it wants. Then, it returns that response to our frontend, and that is displayed to our user.

4 Development/Implementation Details

Our development process was divided into two main factors: backend and frontend. The team was split into two groups as one frontend member, Deniz, and four backend members, Emirhan, Enis, Oğuz and Olcaytu. At frontend we have used Flutter since Deniz had development experience over three years with it and at backend we have used Flask to write our microservices, deploy them to Amazon EC2 and access them through the Google Cloud Functions which checks the integrity of the end-user.

For the frontend architecture, we have used the MVC pattern since it speeds up the development time significantly. As our system is heavily reliant on the internet connection and consistency is important, we are caching the results from the requests that are done through Google Cloud Functions so same requests in a short period of time, between one and ten minutes depending on the service, do not pressure the services so the availability concerns are lowered.

At the backend, we are using the microservice architecture which is running on the Amazon infrastructure. We have decided to use microservices instead of a monolithic application since the objective of each service is different and the application needs to be available at all times. All of our services are written in Python using the Flask framework. The reason why we chose Flask is, it is a lightweight framework which ensures that the response time is minimal since to increase the retention of the end-users we have to prevent any kind of delay that may lose us customers.

5 Test Cases and Results

Test ID	FTC-1
Test Type/Category	Functional
Title	“Next” button should work as expected.
Procedure of testing steps	1. Enter the cloze test screen. 2. Type the correct answer. 3. Tap the next button.
Expected results	The next question should be shown, and the strength of the word should be increased by 1.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Cloze Test

Test ID	FTC-2
Test Type/Category	Functional
Title	Next button should be unclickable if the typed word is not correct.
Procedure of testing steps	1. Enter the cloze test screen. 2. Type an incorrect answer. 3. Tap the next button.
Expected results	Next button should not be clickable.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Cloze Test

Test ID	FTC-3
Test Type/Category	Functional
Title	Skip button should always be clickable, and work as expected.
Procedure of testing steps	1. Enter the cloze test screen. 2. Type an incorrect answer. 3. Tap the skip button. 4. Type a correct answer. 5. Tap the skip button.
Expected results	The skip button should be clickable regardless of what is typed, and the strength level of the words should be set to 1.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Cloze Test

Test ID	FTC-4
Test Type/Category	Functional, Visual
Title	The correct letters should be written in green, and the incorrect ones in red.
Procedure of testing steps	1. Enter the cloze test screen. 2. Start typing the answer, and add some incorrect letters.
Expected results	Correct letters should be green, and the rest should be red.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Cloze Test

Test ID	FTC-5
Test Type/Category	Functional
Title	The generated cloze test question should be different each time.
Procedure of testing steps	1. Enter the cloze test screen. 2. Note the question. 3. Refresh the page by clicking on the top right button.
Expected results	The newly generated question should be different.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Cloze Test

Test ID	FTC-6
Test Type/Category	Functional
Title	The questions should be regenerated when the refresh button is clicked.
Procedure of testing steps	1. Enter the cloze test screen. 2. Check the questions. 3. Click the refresh button.
Expected results	New questions should be generated for the words.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Cloze Test

Test ID	FTC-7
Test Type/Category	Functional
Title	The questions should be regenerated when new words are added.
Procedure of testing steps	1. Enter the cloze test screen. 2. Check the questions. 3. Enter the suggester screen and click on a video. 4. Click on a word in the subtitles, and click the “Add Word to List” button. 5. Wait for a minute. 6. Enter the cloze test screen.
Expected results	A question should be generated for the newly added word.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Failure
Service Tested	Cloze Test

Test ID	FTC-8
Test Type/Category	Functional, Visual, Usability
Title	The user can enter at most the number of characters the correct answer has.
Procedure of testing steps	1. Enter the cloze test screen. 2. Type an answer.
Expected results	The user should not be able to write more than the number of characters of the correct answer.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Cloze Test

Test ID	NFTC-1
Test Type/Category	Non-Functional, Performance
Title	The questions should be cached in the cloze test screen.
Procedure of testing steps	1. The number of words that are due should be checked from the database. 2. The number of questions received from the API should be checked.
Expected results	Questions for all due words should be received.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Cloze Test

Test ID	NFTC-2
Test Type/Category	Non-Functional, Performance
Title	All questions should be received in under 1 second.
Procedure of testing steps	1. Enter the suggester screen. 2. Add some words to the user list by clicking on the subtitles. 3. Wait for a minute. 4. Enter the cloze test screen.
Expected results	All questions should be received in under 1 second.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Failure (Questions are not updated unless refresh is clicked)
Service Tested	Cloze Test

Test ID	NFTC-3
Test Type/Category	Non-Functional, Performance
Title	Questions should be returned in under 1 second when the refresh button in cloze test screen is clicked.
Procedure of testing steps	1. Enter the cloze test screen. 2. Tap the refresh button. 3. Wait for the questions to be returned.
Expected results	The questions should be returned in under 1 second.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Cloze Test

Test ID	FTC-9
Test Type/Category	Functional
Title	When a query is entered into the search bar, close words should be listed.
Procedure of testing steps	1. Enter the dictionary screen. 2. Type something into the search bar.
Expected results	Suggested results should be returned.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Dictionary

Test ID	FTC-10
Test Type/Category	Functional
Title	The suggested results should be clickable, and when clicked, sentence count, the sentences, and the number of unknown words in each sentence should be listed.
Procedure of testing steps	1. Enter the dictionary screen. 2. Type a word into the search bar. 3. Click a suggested result.
Expected results	The mentioned information should be returned.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Dictionary

Test ID	FTC-11
Test Type/Category	Functional
Title	The returned sentences should be sorted.
Procedure of testing steps	1. Enter the dictionary screen. 2. Type a word into the search bar. 3. Click a suggested result.
Expected results	The returned sentences should be sorted in ascending order, according to the number of unknown words in them.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Dictionary

Test ID	NFTC-4

Test Type/Category	Non-Functional, Usability
Title	The suggested words should be listed in under 0.5 seconds.
Procedure of testing steps	1. Enter the dictionary screen. 2. Type into the search bar. 3. Wait for the results to appear.
Expected results	The suggested words should be listed in under 0.5 seconds.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Dictionary

Test ID	FTC-12
Test Type/Category	Functional
Title	Videos/sentences should be suggested according to the user's unknown words.
Procedure of testing steps	1. Enter the suggester screen. 2. Click on a video / sentence.
Expected results	The suggested sentence should contain only one unknown word. All the other words should be in the user's known words list.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Suggester

Test ID	FTC-13
Test Type/Category	Functional, Visual
Title	Each sentence should also show the unknown word, for which the sentence is suggested.
Procedure of testing steps	1. Enter the suggester screen. 2. Check the unknown word, and the sentence by clicking on the video.
Expected results	The unknown word in the sentence should be shown on the suggester screen.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Suggester

Test ID	FTC-14
Test Type/Category	Functional
Title	Previous and next buttons should change the suggested video.
Procedure of testing steps	1. Enter the suggester screen. 2. Click on a video. 3. Click on the video player, and click the previous / next button.
Expected results	The buttons should work as expected.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Suggester

Test ID	FTC-15
Test Type/Category	Functional
Title	Every video should be started from the starting point of the suggested sentence.
Procedure of testing steps	1. Enter the suggester page, and click on a video.
Expected results	Video should be started from the starting time of the suggested sentence.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Suggester

Test ID	FTC-16
Test Type/Category	Functional
Title	When the sentence ends, the video should be automatically paused.
Procedure of testing steps	1. Enter the suggester screen, and click on a video. 2. Wait for the sentence to end.
Expected results	The video should be automatically paused when the sentence ends.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Failure
Service Tested	Suggester

Test ID	FTC-17
Test Type/Category	Functional
Title	There should be restart, previous and next buttons on the video player.
Procedure of testing steps	1. Enter the suggester screen, and click on a video. 2. Try the mentioned buttons.
Expected results	The buttons should work as expected.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Failure
Service Tested	Suggester

Test ID	FTC-18
Test Type/Category	Functional
Title	Only the part of the video where the suggested sentence is playing should be displayed.
Procedure of testing steps	1. Enter the suggester screen, and click on a video.
Expected results	The video duration should be equal to the sentence duration.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Failure
Service Tested	Suggester

Test ID	FTC-19
Test Type/Category	Functional
Title	The progress bar at the bottom of the video should be movable.
Procedure of testing steps	1. Enter the suggester screen, and click on a video.
Expected results	Video progress bar should be movable.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Cloze Test

Test ID	FTC-20
Test Type/Category	Functional
Title	When the transcript button is clicked, the video player mode should disappear and a transcript mode should appear.
Procedure of testing steps	1. Enter the suggester screen, and click on a video. 2. Click on the transcript mode, which is shown by letter T on the screen.
Expected results	A transcript screen, in which only the video transcript is shown, should be entered.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Failure
Service Tested	Suggester

Test ID	FTC-21
Test Type/Category	Functional
Title	The subtitle button should not work if we are currently in the Subtitle half screen.
Procedure of testing steps	1. Enter the suggester screen, and click on a video. 2. The Subtitle half screen should be entered by default.
Expected results	The subtitle button should have no effect (or it should not be clickable).
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Failure
Service Tested	Suggester

Test ID	FTC-22
Test Type/Category	Functional
Title	Subtitles should be able to be turned on and off when the video is full screen.
Procedure of testing steps	1. Enter the suggester page and click on a video. 2. Click the full screen button at the bottom right corner. 3. Click the subtitle button.
Expected results	Subtitles should be able to be turned on and off when the video is full screen.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Suggester

Test ID	FTC-23
Test Type/Category	Functional
Title	When a word in the subtitles is clicked, a screen with example sentences should be displayed.
Procedure of testing steps	1. Enter the suggester screen, and click on a video. 2. Click a word in the subtitles.
Expected results	The dictionary / example sentences screen should be displayed.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Suggester / Dictionary

Test ID	FTC-24
Test Type/Category	Functional
Title	If the word is already known by the user, the “Learn Word” and “Add to Word List” buttons should be unclickable.
Procedure of testing steps	1. Enter the suggester screen, and click on a video. 2. Click on a word in the subtitles, which is known by the user.
Expected results	The buttons should be unclickable.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Failure
Service Tested	Suggester

Test ID	FTC-25
Test Type/Category	Functional
Title	If the word is already known by the user, there should be a check mark right next to the “Learn Word” button.
Procedure of testing steps	1. Enter the suggester screen, and click on a video. 2. Click on a word in the subtitles, which is known by the user.
Expected results	There should be a check mark next to the “Learn Word” button.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Failure
Service Tested	Suggester

Test ID	FTC-26
Test Type/Category	Functional
Title	The “Learn Word” and “Add Word to List” buttons should be clickable and work as expected, if the word is not currently in the user’s vocab list.
Procedure of testing steps	1. Enter the suggester screen, and click on a video. 2. Click on a word in the subtitles, which is not in user vocab list. 3. Click on the “Add Word to List” and “Learn Word” buttons.
Expected results	The former button should set the word strength to 1, and the latter to 8.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success

Service Tested	Suggester

Test ID	FTC-27
Test Type/Category	Functional
Title	When the back button on the top left is clicked, the app should return to the suggester main screen.
Procedure of testing steps	1. Enter the suggester screen, and click on a video. 2. Click the “Back” button on the top left.
Expected results	The app should return to the suggester main screen.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Suggester

Test ID	FTC-28
Test Type/Category	Functional, Visual, Usability
Title	The unknown words in the subtitles and transcript modes should be red, and the known words should be white.
Procedure of testing steps	1. Enter the suggester screen, and click on a video. 2. Check the subtitle colors. 3. Enter full screen, and check the subtitle colors. 4. Enter transcript mode, and check the transcript colors.
Expected results	Unknown words should be written in red, and the rest in white color.
Priority/Severity	Minor

Date Tested / Test Result	18.05.2023 / Failure
Service Tested	Suggester / Transcript

Test ID	FTC-29
Test Type/Category	Functional
Title	Words that are currently being learned by the user should be listed.
Procedure of testing steps	1. Enter the progress screen.
Expected results	The words that are currently being learned by the user should be listed in the progress screen.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Progress

Test ID	FTC-30
Test Type/Category	Functional
Title	Once a word is fully learned, the word should no longer be displayed.
Procedure of testing steps	1. Enter the progress screen, and choose a word from that list with strength = 7. 2. Choose one of the two: a. Go to the cloze test screen (if the word is due) and answer the question correctly. b. Go to the suggester screen and click on the word in the subtitles in any video, and click on “Learn Word”. 3. Enter the progress screen.
Expected results	The word should no longer appear.

Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Progress

Test ID	FTC-31
Test Type/Category	Functional
Title	Leaderboard should list the users in a descending order according to their score.
Procedure of testing steps	1. Enter the Profile screen, and scroll down to the leaderboard.
Expected results	The users should be listed according to their score in that language, in descending order.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Profile

Test ID	FTC-32
Test Type/Category	Functional
Title	Badges, achievements and progress bar should work successfully
Procedure of testing steps	1. Go to profile page 2. Check badges, achievements and progress bar
Expected results	Badges, achievements and progress bars work successfully
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Failure

Service Tested	Profile

Test ID	FTC-33
Test Type/Category	Functional, Usability
Title	Switching learnt language should work as expected
Procedure of testing steps	1. Go to profile page 2. Change learnt language 3. Check for results to reflect new language
Expected results	Sentences with new language should be shown on cloze test and suggester tabs
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Profile

Test ID	FTC-34
Test Type/Category	Functional, Usability
Title	After a new language is added, the most frequent word list should be shown to users with expected features.
Procedure of testing steps	1. Add a new language 2. Check for most frequent words list as a user 3. Check for clicking to the words and marking them as known
Expected results	Users should see the new list and be able to mark words as known.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Suggester

Test ID	FTC-35
Test Type/Category	Functional
Title	A user should be able to start learning a new language.
Procedure of testing steps	1. Enter the Profile page. 2. Click on the language button on the top right. 3. Click on “Learn a New Language” button. 4. Click on a suggested language. 5. Click on the words that are known.
Expected results	The words should be added to the user’s vocab list, language should be added to the user’s learned languages.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Profile

Test ID	FTC-36
Test Type/Category	Functional
Title	A user should be able to change the interface language.
Procedure of testing steps	1. Enter the Profile screen. 2. Click on the “Change Language” button.
Expected results	The interface language should change to the selected language.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Failure
Service Tested	Profile

Test ID	FTC-37
Test Type/Category	Functional, Usability
Title	Third party authentication signup should work as expected
Procedure of testing steps	1. Sign up to the app using a third party option. 2. Check databases and confirm that signup worked as expected.
Expected results	A new user is created and changes are reflected to our databases and firebase database.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Signup

Test ID	NFTC-5
Test Type/Category	Non-Functional
Title	All transactions should ensure integrity.
Procedure of testing steps	1. Inspect code for possible disintegrity sources.
Expected results	All code complies with integrity requirements.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	DB

Test ID	NFTC-6
Test Type/Category	Non-Functional, Effectiveness
Title	Database should have sufficient resources to run suggest words for users
Procedure of testing steps	1. Fill the DB using DB filler script 2. Check for sufficient suggestions using the user application.
Expected results	There should be a sufficient number of suggested words for the user.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	DB, DB filler script

Test ID	NFTC-7
Test Type/Category	Non-Functional, Reliability
Title	DB filler script should work as intended
Procedure of testing steps	1. Run the DB filler on a fresh test DB. 2. Check for the intended result.
Expected results	Intended results are found on DB
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Backend, DB

Test ID	NFTC-8
Test Type/Category	Non-Functional, Portability
Title	User app should work on both Android and iOS
Procedure of testing steps	1. Test the app thoroughly in Android emulator 2. Test the app thoroughly in iOS emulator
Expected results	The app works as expected in both operating systems
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	User application

Test ID	NFTC-9
Test Type/Category	Non-Functional, Efficiency
Title	All AWS resources are deployed in the Frankfurt region.
Procedure of testing steps	1. Log in AWS console 2. Check region for EC2 and RDS services
Expected results	Both services are deployed to the Frankfurt region.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	AWS, Host, DB

Test ID	NFTC-10
Test Type/Category	Non-Functional, Security
Title	Check certificate based auth. is used for SSH connection to host machine.
Procedure of testing steps	1. Log in AWS console 2. Check for certificate in EC2 attributes
Expected results	Host has a certificate configured.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Success
Service Tested	AWS, EC2

Test ID	NFTC-11
Test Type/Category	Non-Functional, Security
Title	Check a limited (5000-5100) numbers of ports are publicly accessible of the host machine
Procedure of testing steps	1. Log in to the AWS console. 2. Check security rules for the host machine.
Expected results	Only ports 5000-5100 are open for public traffic.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	AWS, network security rules

Test ID	NFTC-12
Test Type/Category	Non-Functional, Security
Title	Check IAM users are created for all team members.
Procedure of testing steps	1. Log in to the AWS IAM service. 2. Check IAM users for all team members
Expected results	All members have access to the AWS console.
Priority/Severity	Minor
Date Tested / Test Result	18.05.2023 / Success
Service Tested	AWS, IAM service

Test ID	NFTC-13
Test Type/Category	Non-Functional, Financial
Title	Check whether billing alerts for all aws services used are enabled
Procedure of testing steps	1. Log in to the AWS console. 2. Check CloudWatch service for billing alerts.
Expected results	Billing alerts enabled
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	AWS, Billing

Test ID	NFTC-14
Test Type/Category	Non-Functional, Security
Title	Test postgres password is not default.
Procedure of testing steps	1. Open postgres console 2. Try logging in with the default password ‘postgres’.
Expected results	Unable to login
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	AWS, RDS

Test ID	NFTC-15
Test Type/Category	Non-Functional, Efficiency
Title	All Firebase resources must be deployed to europe-west3 region.
Procedure of testing steps	1. Enter firebase console 2. Check region for auth and Cloud Functions.
Expected results	All Firebase resources are deployed to europe-west3 region.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	Firebase authentication, Cloud Functions

Test ID	NFTC-16
Test Type/Category	Non-Functional, Reliability
Title	Check host machine has static IP address
Procedure of testing steps	1. Check console for elastic IP 2. Additionally, reboot instance and see if ID address remained the same.
Expected results	Elastic IP enabled, host has same IP address after rebooting.
Priority/Severity	Major
Date Tested / Test Result	18.05.2023 / Success
Service Tested	AWS, EC2, Network

6 Maintenance Plan and Details

Our maintenance plans were shaped according to two factors: new features and user number. Both these events may show system limitations and may cause changes.

6.1 System Limitations and Upgrades

Currently we are using AWS EC2's free-tier version. Even though this system worked pretty well during our development phase we know that this system has its limitations. Thus as our user number increases and if new and more complicated features are added, we will need to upgrade our cloud system to another system with much greater computing power.

6.2 Problem Handling and Updates

Even though we ran comprehensive tests on our system, we may still get feedback about problems in our application from customers. After we finalize our product we will begin a support project for Lingui as the whole team and act upon this feedback. We will regularly work on the problems and update the app. We will use the GitHub Issues page to keep track of these fixes.

6.3 Version Checking

Currently we are using GitHub for both sharing code and version control. In the future, we will use this platform also for checking between versions of the application and detecting possible errors, and correcting them.

6.4 NLP Optimization

In this project we used a third-party NLP library called spaCy. spaCy works pretty well with the customizations we made on it for now. However, it is still not perfect. In the future, we may add languages that are not featured in spaCy. Thus, we will write our own NLP library in order to both support these new languages and solve the problems caused by spaCy that we are now solving via customizations.

6.5 Enhanced UI/UX

For now we are satisfied with the performance and functionality of our UI. However, in the future as our team expands, we may want to add advanced elements such as custom animations or 3D models in order to enhance the user experience and have a lead over our rivals in the industry.

7 Other Project Elements

7.1 Consideration of Various Factors in Engineering Design

We came across many important factors during the research phase which affected the design and development of Lingui. These factors were given below.

7.1.1 Public Health

While there is no direct link between Lingui and public health we can mention a few side benefits of using Lingui which is also beneficial for users' overall physical and mental well-being. First of all, Lingui is a language-learning app. Language learning has many sound effects on the brain such as improved concentration, a more powerful memory, more vital communication skills, and enhanced creativity[5]. It also helps with mental health by boosting confidence, enhancing self-esteem, decreasing the effects of depression, and delaying cognitive decline[6]. Another minor benefit of our app is that it has a dark theme. Dark themes are also beneficial for eye health since they lower blue light exposure[7].

7.1.2 Public Safety

The most important part of public safety in Lingui is data privacy. Even though we do not store any sensitive data, the email addresses and passwords that our users use may probably be used elsewhere. Because of this, we used a strong encryption algorithm while hashing user information in our database and we made sure no single part of our application is vulnerable to SQL Injection attacks. We also only used YouTube videos from manually selected channels in order to protect our audience from violent, offensive, or harmful content.

7.1.3 Public Welfare

As a free-to-use app, Lingui contributes to the educational needs of the public to a great extent. Regardless of geographical location and socioeconomic background, Lingui provides everyone with an opportunity to learn new languages boosting education accessibility. Lingui also provides a language-learning expence via listening to native speakers which is not accessible in most schools around the world. Also, in many cases, people do not have the opportunity to learn new languages after they graduate from school. However, Lingui is accessible to everyone and requires a relatively short time every day. Thus, Lingui provides this opportunity and gives people a chance for lifelong learning which is crucially important in today's world.

7.1.4 Global Factors

There are only two constraints that limit Lingui's global reach. First, the application UI must be in the user's native language or at least in a language that the user knows very well. Second, the number of languages featured as languages to learn must increase which is very hard after some point since there are not many desirable videos (e.g. with sufficient subtitles) or big YouTube channels in languages with a smaller speaker base. This fact makes content moderation very hard. Also, for newer languages especially with very different grammar, the implementation of an NLP model becomes very hard since we will also need to work with many speakers of that language.

7.1.5 Cultural Factors

Lingui is an app that introduces people to the cultures of the people speaking their desired language. Thus, providing a platform for cultures to interact with each other. Because of this, we strived not to make our application design biased toward one or a group of particular cultures. In the future, if our application gains a considerable audience, we plan to celebrate these cultures by changing our UI on important days such as Dia de Muertos or the 4th of July.

7.1.6 Social Factors

The ultimate goal of our project is to create a more integrated society regardless of nationality or socioeconomic status. Thus, we are serving people with a free-to-use app and trying to boost their language learning efforts.

7.1.7 Economic Factors

All the hardware our application uses (databases, servers, etc.) is free tier. These hardware elements were sufficient for us at the start of our project but, we think as the user number increases we will need to make newer procurement plans for these services.

	Effect level	Effect
Public health	1	Public health only has side effects on our system
Public safety	9	Data privacy and safety of the system are big priorities
Public welfare	7	Our project is very important in terms of making language learning available to groups of people without language education available to them
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.
Economic Factors	3	Economic factors may only have future implications on our system.

Table 1: Factors that can affect analysis and design.

7.2 Ethics and Professional Responsibilities

In terms of ethics, we paid immense attention to user and data privacy. First, we do not intend to collect any unnecessary user data such as location, camera data, device files and etc. We have pledged not to share any confidential user data with third parties and we took careful technical precautions to prevent any data leak. If we need to collect or share any user data we will ask for user permission first and foremost. We also paid utmost attention to not cause any copyright infringements or violations. Apart from all these, we used all our third part services, hardware, or components abiding by the corresponding licenses and terms of service.

As for professional responsibilities, we paid utmost attention and care to create a product while abiding by professional standards such as the Joint Statement of Ethical Principles of the Engineering Council[8]. We valued learning, leadership, communication, and acting upon knowledge. We made use of scientific literature from the start till the end of our project while giving required references and avoiding plagiarism in any reports.

7.3 Teamwork Details

7.3.1 Contributing and Functioning Effectively on the Team

Since our team's mindset is like "From each according to his ability, to each according to his needs", we did not have any problems such as forcing a teammate to contribute to the project. Each member was proficient in an area before the project, so we just distributed people to the right parts, and each one of us functioned really well. Deniz is experienced and proficient in frontend development, so nearly carried all the work on the front side of the project. He used Flutter to create a very eye-appealing mobile app. Enis is interested and experienced in DevOps and infrastructure, so we let him deal with all of the work on that part of the project. He deployed our project's servers to AWS. Emirhan, Oğuz, and Olcaytu are proficient in backend development. So they did a great job of effectively dividing the workload on backend, and created functioning backend services using Flask and Spring.

7.3.2 Helping Creating a Collaborative and Inclusive Environment

Every member of the team has been friends since freshman year, and have been on numerous projects before together. So, working as a team with this group of people was not anything new, or surprising. Before even picking a group meeting time, every member's availability was asked, and every member tried to create availability to match the other members' availability. Since some members worked part-time jobs, if a member was tight on his other schedules, other members picked up his work with the premise that the busy member will do "overwork" in the future for that member. Even in small matters like choosing the authentication framework, we took all member's ideas and respected their past experiences with the frameworks so that we can pick the fittest tools for our project, aiming that every team member would be comfortable in the project domain.

7.3.3 Taking Lead Role and Sharing Leadership on the Team

We do not have any specified leader. Every member has full leadership responsibility on the project, and is willing to call the shots if necessary. If a service is needed for the frontend, Deniz sets a deadline for the backend team for that service. If a service needs to be deployed to AWS EC2, the backend team collaboratively sets a deadline for Enis. If the deadline for a report is coming, whoever notices it first alerts the team through the WhatsApp group and sets a meeting. While doing such stuff, we also try to respect other members' schedules and availability, which prevents toxic leadership amongst the team. Since this team has been on numerous projects before, we hardly take offense from each other when taking "orders", and this helps us to get things done faster, and clearer.

7.3.4 Meeting Objectives

The first meetings we had at the start of the semester were about which project idea we were going to implement. These meetings were more chaotic since we did not have good meeting practices and in the end, we needed to consult with our instructors on the topic.

After this phase, we created a project design and plan at the Analysis Stage. This design and plans were evaluated and changed in future meetings as required while we were in the Development Phase.

When we entered the Development Phase, we continued with having two types of meetings. Team Meetings were made only between project team members, and Progress Meetings which were done with team members and instructors. Team Meetings were sometimes done face-to-face since we preferred it but they were mostly done online through our Discord server or Zoom. In Team Meetings we discussed what we had previously done, what was going to be done, and which things were our impediments. This allowed good communication between team members and we could finish tasks collaboratively. Progress Meetings on the other hand, were conducted online through Google Meet. In these meetings, the project group members showed what they had done, what they are planning to do, and their problems to the instructors. Then instructors gave feedback and recommendations about those issues.

In addition to these kinda formal meetings, many times a subset of the project group would meet face to face in order to solve problems or do pair programming when they were looking at the same or a similar problem. These meetings enabled us to be much more efficient and contributed to communication since the issues were not communicated through writing or speech but were understood while having hands-on experience.

7.4 New Knowledge Acquired and Applied

Before this project, we did not have used nor know how to use Docker and containerization. We took some courses from YouTube and Udemy in order to get familiar with it. The subject is not that hard actually after you know the fundamentals and basics. We also did not have previously used technologies such as NLP and YouTube Data API. In order to learn these topics we needed to read a lot of documentation and watch some more YouTube videos. Some of us have previously used cloud servers and compute engines, however, this time we got familiar with Amazon Web Services. We used many APIs and thus had many dependencies in this project. We learned how to manage the risks that these dependencies cause and take necessary precautions. We also had real-time streaming this project thus needed to put the performance of the system as one of our main priorities. We learned how to optimize our systems and prevent delays.

This project was the longest project all of us had in our lives. Thus we needed to manage this project effectively and efficiently. We learned how to communicate better, how to have more effective meetings, and how to resolve problems together.

Since we moderate the content on our application ourselves we got to know YouTube and its algorithm much better. We surfed through hundreds of YouTube channels, learned which type of subtitles is better, what the common mistakes made while writing captions and etc.

We also got to know our domain. As a group, we conducted in-depth research about language learning especially in the initiation step of our project. While doing this research we used forum discussions and feedback from the industry in addition to academic literature.

8 Conclusion and Future Work

In conclusion, we have built an application for language learners to learn vocabulary in their desired languages by watching personalized YouTube videos. Using this application language learners will be able to learn languages while doing an entertainment activity most of them do every day regardless of language learning. Furthermore, our application provides quizzes for its users in order to let them test their knowledge of the vocabulary they are trying to learn.

In the near future, we are planning to add video categories to increase personalization and give a better user experience. Users will be able to choose their desired categories and be more entertained while using Lingui. According to the success of Lingui and demand from international markets, new UI language support and languages to learn may be added. This will increase Lingui’s market share and reach into the global market. Also, we are currently using free-tier cloud services. We project that these services will not be sufficient as our user base and video count gets larger. We plan to upgrade these systems in the future.

9 Glossary

SRS: Spaced Repetition System.
AWS: Amazon Web Services
API: Application Programming Interface
UI: User Interface
NLP: Natural Language Processing

10 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.

[5] Spence, C. (2022). How learning a new language changes your brain: Cambridge English. Retrieved from https://www.cambridge.org/elt/blog/2022/04/29/learning-language-changes-your-brain/#:~:text=Language%20learning%20skills%20will%20help,%2Dsolving%20and%20decision%E2%80%93making.

[6] Tan, A. (2021). Retrieved from https://medium.com/live-your-life-on-purpose/why-you-can-and-should-learn-a-new-language-to-improve-your-mental-health-fc50a47f669a

[7] Porter, D. (2022). Retrieved from https://www.aao.org/eye-health/tips-prevention/digital-devices-your-eyes

[8] (N.d.). Retrieved from https://www.engc.org.uk/standards-guidance/guidance/statement-of-ethical-principles/