



## Database Systems Project Proposal

# Online Language Learning Platform

Project Group no: 36

Selahattin Cem Öztürk - 21802856

Lamia Başak Amaç - 21601930

Webpage: <https://github.com/selahattinozturk/cs353-project>

<b>Database Systems Project Proposal</b>	<b>0</b>
<b>Introduction</b>	<b>2</b>
<b>Description</b>	<b>2</b>
<b>Requirements</b>	<b>3</b>
Functional Requirements	3
Teacher Side	3
Native Side	3
Student Side	3
Admin Side	4
Non-Functional Requirements	4
Limitations	4
<b>Entity-Relationship Diagram</b>	<b>5</b>
<b>References</b>	<b>6</b>

# Introduction

In this project, we are implementing a database system for an online language learning platform. Since the learning process has been overtaken by online platforms these days (especially after the corona pandemic), it is crucial to have a feasible database system for learning platforms. Connecting teachers and students for online lectures and native language speakers with students for online speaking practice is the primary aim for an online language learning platform.

In the rest of the report, we will discuss the description of the project, requirements (functional and non-functional), limitations, and entity relationship diagram in a detailed way. In the description part, we will address the primary concerns of our project.

While the non-functional requirements are general to the whole application, functional requirements are related to the specific usage of the teachers, natives, students, and admins.

At the end of the report we will show an entity-relationship diagram that shows the entities and their relationships with each other in a detailed way.

## Description

In the project, Dualingoingto, there are three fundamental features that will be considered.

Firstly, the student could take a lesson from a teacher according to their level. Lessons also contain homeworks and exams that evaluate their knowledge. Teachers of the corresponding course can also grade those exams and homeworks. Also teachers can associate some words that they defined and show them as the word of the day.

Secondly, the student could request an online meeting from a native speaker for a specific date and time. If the native speaker accepts the request, after the meeting they can give a grade to the student.

Last but not least, the admin of the platform could be able to view some analytics about the courses and students such as viewing the average grade and viewing the enrolled student to a specific course.

It is crucial to use a database to store the required data, such as s\_id, t\_id, l\_id, m\_id, etc., to provide security and easy access [1]. Otherwise, we could not trace the lessons and meetings so that the main functionalities could not be performed.

# Requirements

In this section we discuss the functional, non-functional requirements, and limitations of this project.

## Functional Requirements

### Teacher Side

**-Ability to give lessons:** The teacher should have the ability to give lessons when the student requests.

**-Ability to assign homework:** The teacher should have the ability to assign homework to a specific student by indicating their student ID.

**-Ability to grade exams and homework:** The teacher should have the ability to grade the homeworks that is specifically given to a student and also the exams that are given to the whole class.

**-Ability to see how many students are taking their class:** The teacher should have the ability to check how many students are taking their course.

**-Ability to view student activities on student's accounts:** The teacher should have the ability to view the activities of their students by entering their own profile.

**-Ability to create their own dictionaries:** The teacher should have the ability to create their own words in order to show them in their course page as the word of the day. They should be able to define a word and give an example usage for that word. They can also use those words to define phrasal verbs.

**-Ability to upload certificates:** Teachers should be able to upload certificates to prove that they have competence for teaching a language.

### Native Side

**-Ability to hold meetings:** The natives should have the ability to hold speaking exercises by accepting the request which is sent by the student.

**-Ability to grade speaking exercises:** The natives should have the ability to grade the speaking exercise after the meeting ended.

### Student Side

**-Ability to pick a language:** After login to the website, the student should have the ability to pick a language in order to take lessons from a teacher or request a meeting from a native speaker.

**-Ability to request class from teacher:** The student should have the ability to request classes from the teacher by specifying their language levels.

**-Ability to request online meetings from native language speakers:** The student should have the ability to request an online meeting from a native language speaker for speaking exercises by specifying the date and the time.

#### Admin Side

**-Ability to view how many students are learning a specific language:** Because we will hold the data about registered students, we can easily take the information in terms of counts of them. Admin can reach this information and can see how many students are learning a specific language from a specified teacher.

**-Ability to view the grade average of the lesson:** Since we will hold the corresponding grades of the student's exams and homeworks, Admin should be able to view the average grade of a specific course taught by a teacher.

**-Ability to approve the certificate:** Since the teachers can upload fake certificates, the admin should be able to see the uploaded certificates and approve them if everything is okay.

#### Non-Functional Requirements

- **Usability:** System should be responsive and easy to use so that users from all backgrounds can use it without needing to learn anything new.
- **Reliability:** System should ensure that there is no data lost and that all errors are handled gracefully.
- **Performance:** System should handle all requests within less than 2 seconds.
- **Maintainability:** System should be easy to maintain in the future.
- **User-Friendly:** System should not lead to any misunderstanding for users such as appropriate buttons for each function.
- **Scalability:** System should be able to handle big surges in traffic and scale depending on the traffic.
- **Security:** As the system will store the link for the online meeting, also the users enter some personal information, thus we should make sure that the data is stored and accessed securely.

#### Limitations

- Students can't request any class above their levels.

- If a native speaker is not a teacher at the same time, they can't give lessons to the students.
- Teachers can't give the same homework to another student.
- Students can't take the exams and upload homeworks after their due date.
- Teachers can't view the students' activities if they are not enrolled in any of their lessons.
- Teachers can't see other teachers' defined words.

## Entity-Relationship Diagram

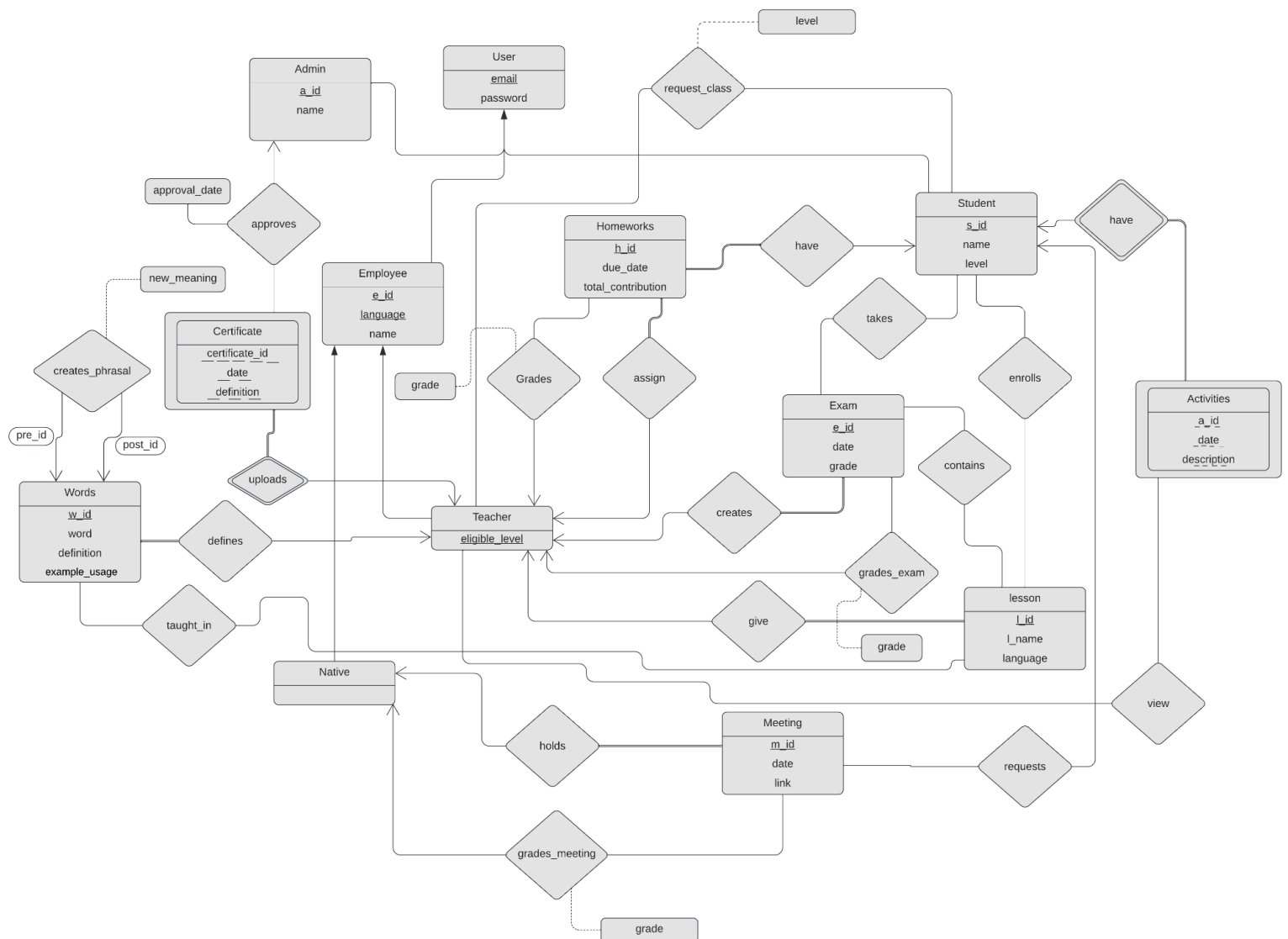


Figure in above shows the initial conceptual design of our project. We have drawn the diagram by using an online tool named LucidChart [2].

## References

- [1] H. F. Korth and A. Silberschatz, Database system concepts. Singapore: McGraw-Hill, 1991.
- [2] "Intelligent diagramming," Lucidchart. [Online]. Available: <https://www.lucidchart.com/pages/> . [Accessed: 27-Feb-2022].