



Bilkent University

Department of Computer Engineering

CS 353 Database Systems

Group 1

Project Proposal

Abdullah Can Alpay 21702686 - Section 2

Cemre Biltekin - 21802369 - Section 1

Mustafa Yaşar - 21702808 - Section 1

Oğuz Kaan İmamoğlu - 21702233 - Section 1

Instructor: Uğur Güdükbay

Teaching Assistant: Mustafa Can Çavdar

Table of Contents

1. Introduction	2
2. Project Description	2
2.1. Why There Is A Need For Database	3
2.2. How Database Will Be Used	4
3. Requirements	4
3.1. Functional Requirements	4
3.1.1. Student	4
3.1.2. Premium Student	6
3.1.3. Course Creator	6
3.1.4. Site Administrator	7
3.2. Non-Functional Requirements	7
3.2.1. Quality Requirements	7
3.2.1.1. Usability	7
3.2.1.2. Reliability	8
3.2.1.3. Performance	8
3.2.1.4. Access Security	9
3.2.1.5. Supportability	9
3.2.2. Pseudo-Requirements (Constraints)	9
3.2.2.1. Implementation	9
4. Limitations	10
5. Entity Relationship Diagram	11
6. Website	12

1. Introduction

This report is the project proposal report of the online course platform project, and it gives information about the purpose, basic functionalities, requirements, limitations, database model of the project, and how the project will be designed and implemented.

In the second chapter, the overview of the project is explained, and the need for a database in the project and how the database will be integrated into the project is discussed.

In the third chapter, functional requirements, non-functional requirements and constraints of the project are described. Functional requirements are written considering how different end users interact with the application and how the system is supposed to respond to certain operations. Non-functional requirements are concerned with the system aspects except its functionalities. Constraints show the technologies that will be used for implementation.

In the fourth chapter titled as Limitations, actions that end users can and cannot do are listed; in other words, the limitations that should be put to users in the implementation are discussed.

In the fifth chapter, the E/R (Entity Relationship) Diagram of the project is given. The database design of the project will be based on this model. Finally, in the last chapter, a link to the website containing information about this project and its reports is given.

2. Project Description

The project is a web-based database application which is an online course platform that provides courses for students and enables course creation for instructors. There are four types of end users. The first one is the students. Students must be able to search for courses, add courses to their wishlists and carts, buy the courses that they desire, add funds to their wallets. They can keep track of their progress on different courses, once they complete a course, they can give feedback and ratings to that

course and receive a certificate of completion. Also, they are able to request a refund for a bought course, with valid reasons. If a student purchases a membership, he/she becomes a premium student which is another end user. Premium students can perform anything a student can and they have certain benefits. Another end user in the platform is the course creator. Course creators can publish courses, add lectures and assignments to these courses and make announcements for these courses. They can keep track of how much they earn via their own wallets. Students and course creators can participate in Q&A sessions where students ask questions and the course creator answers. The last type of users is admin. Admin can offer discounts for courses, and it is up to the course creator of that course to decide whether to accept the discount or not. Site admins also must take care of complaints from users and should accept or decline refund requests of students.

A course consists of several lectures and may contain assignments. Each of these lectures and assignments must be completed by students if they want to get a completion certificate for those courses. Students can take notes for lectures and associate these notes with lectures. Courses have question & answer sections. In these sections, students can ask their questions related to courses and the creator answers these questions. Some courses are only open to premium students. Premium students do not have to pay for the certificate. Memberships expire after a year.

2.1. Why There Is A Need For Database

An online course platform has a large amount of data about its users and the courses it contains. The passwords of the users, which students take which courses, which courses consist of which lectures, who is the course creator of a particular course, which courses contain which announcements can be shown as examples of this data. The volume of this data grows as new students and course creators enroll on the platform, new courses are added, and students take new courses. In order to maintain the system properly and process and fetch accurate data (avoiding false entities), the data must be stored in an organized manner and when necessary, the data must be

easily retrieved in a short time for the importance of response time. There are different end users that can perform different operations, and a database system can help regulate authorization like how only a premium student can access courses that are special to premium students. For these reasons, choosing a database for data management will be a wise choice.

2.2. How Database Will Be Used

Database is going to store all data that is related to the course platform and provide these data to the application when it receives queries. There are many use cases of the database. When users log in to the system, their emails and passwords will be checked whether they are correct or not; students can see their courses and their progress, lectures, and announcements of these courses. The database will provide information about the courses in the system, their contents, rates and prices and whether they have certificates or not. By looking at this information, students can decide on buying a course to improve themselves. They can prefer a course creator's teaching style, and with the help of a database system, the students can view courses offered by that course creator. These functionalities and more information will be stored in the database and provided by it.

3. Requirements

3.1. Functional Requirements

3.1.1. Student

- Students should register to the application by specifying a unique email address, password, unique username information, and by selecting the student role on the sign-up page.
- Students should login to the application with appropriate credentials.

- Students should be able to search for a course by course name.
- Students should be able to filter courses according to their corresponding categories (i.e. Biology, Computer Science, etc.).
- Students should be able to view all courses offered by a particular course creator on that course creator's page.
- Students should be able to add courses to and remove courses from their own wishlist via a button and view those courses on the wishlist page at any time.
- Students should add the desired courses to the cart in order to purchase them. They should be able to remove courses from the cart if they change their decision.
- Students should be able to enroll in a course after purchasing it.
- Students should be able to see their progress in a particular enrolled course which is the percentage of completed lectures of the course.
- Students should receive a personal certificate of completion if they paid the certificate price and completed the course.
- Students should rate an enrolled course by using a 5-star rating system and send written feedback after completing it.
- Students should be able to see the course name, course description, course creator name, price, rating, language and the number of enrolled students of a particular course on the course page.
- Students should be able to view course announcements of a course on the course page.
- Students should be able to request a refund for any purchased course by specifying a reason.
- Students should use the lecture page to create a personalized note(s) for any lecture of any course.
- Students should be able to ask many questions to the course creator and view questions and answers on the course's Q&A page.
- Students should be able to send a complaint about an enrolled course at any time.

- Students should pay for a premium membership to upgrade to a premium student.
- Students should receive the appropriate money to their wallets after a refund is processed and approved.
- Students should be able to add funds to their wallets so that they can use these funds while purchasing a course.
- Students should be able to send the answers of the assignments if the deadline of the assignments did not expire.

3.1.2. Premium Student

- Premium students should have access to all features a regular student has and also some special courses special to the premium students.
- Premium students' membership should be canceled by the system when it expires.
- Premium students should receive free certificates when they finish the courses.

3.1.3. Course Creator

- Course creators should register to the application by specifying a unique email address, password, unique username information and by selecting the course creator role on the sign-up page.
- Course creators should login to the application with appropriate credentials.
- Course creators should be able to create courses and lectures of courses.
- Course creators should be able to answer the questions posted on the Q&A page.
- Course creators should be able to post announcements on the announcement page of the relevant course.
- Course creators should be able to allow discounts on their courses.

- Course creators should approve written feedback to display it on the course page.
- Course creators should be able to view student feedback of his/her course.
- Course creators should receive their course creator rating based on the average course rating of all offered courses by them.
- Course creators should be able to view their total earnings.
- Course creators should be able to create assignments for their courses and publish these assignments.

3.1.4. Site Administrator

- Admins should be able to login into the application with appropriate credentials.
- Admins should be able to offer discounts on courses and wait for course creators' approval.
- Admins should be able to view student complaints and see the number of complaints a particular course receives.
- Admins should be able to view all refund requests in detail and reject or accept the request given the textual reason is valid.

3.2. Non-Functional Requirements

3.2.1. Quality Requirements

3.2.1.1. Usability

- For ease of navigation between pages, there will be a navigation bar at the top of the page from which they will be able to change pages.
- Any of the user types whose English proficiency level is A1 or A2 (beginner-level) should understand button labels, top bar menu names and content of pages without having to refer to a dictionary.

- The menu names and button names are self-explanatory to appeal to users' intuition and at most be 10 words.
- The system should display a buffering icon when there is an ongoing request and disable appropriate buttons to prevent multiples of the same request by a user.

3.2.1.2. Reliability

- The system should be 99% reliable for the duration of being logged in by the user.
- Should there be a crash in the system, the progress in a particular course followed by a particular student (where a student has left off) is saved by the system and retrieved immediately when the application is ready to use again.
- The system should not crash during payments to ensure a successful transaction.

3.2.1.3. Performance

- The system, which is a web application, should support at least 1000 users concurrently.
- The maximum response time between click and reaction must be two seconds.
- The average response time between click and reaction must be between 0.1 seconds to 1 second.
- The loading time of each individual page must be between 0.1 seconds to 2 seconds.
- The system should successfully handle the storage and process of large amounts of data.

3.2.1.4. Access Security

- The system should protect the visibility of passwords on the database at any time.
- The system should impose password requirements in setting up the password to ensure the strength of credentials.
- The system should check the authorization of a user before performing an operation on the application to ensure the integrity of roles.

3.2.1.5. Supportability

- The bugs should be maintained by the developers. Developers should solve the issue within 24 hours.

3.2.2. Pseudo-Requirements (Constraints)

3.2.2.1. Implementation

- MySQL should be used for the database management system.
- PHP should be used to develop the back-end of the web application.
- Javascript, CSS, HTML5 should be used to develop the front-end of the web application.

4. Limitations

- A user must have an email and password to login. They can also use their username to login.
- Username can contain at least 6, at most 30 characters.
- Passwords must contain at least 8 characters.
- An email and username can only be used by one user.
- A student can earn one certificate from a course.
- A student can request a refund for a specific course for once.
- A refund must be made within 15 days after purchase.
- Course creators can't create lectures if they do not have any courses.
- A course can be created by only one course creator.
- A student can add at most 50 courses to their wishlist.
- A student can add at most 50 courses to their cart.
- A student can add at most 30 notes to a lecture.
- A student cannot add the same course more than once to their wishlist and cart.
- A course must contain at least 1 lecture.
- Only the course creator of the course can answer questions on its Q&A page.
- A course creator can only send one answer to a posted question.
- A student can earn the certificate of the course only if the progress is complete and all of the assignments of the course is done.
- A membership must be valid for 365 days after purchase and cancellation cannot be made.

5. Entity Relationship Diagram

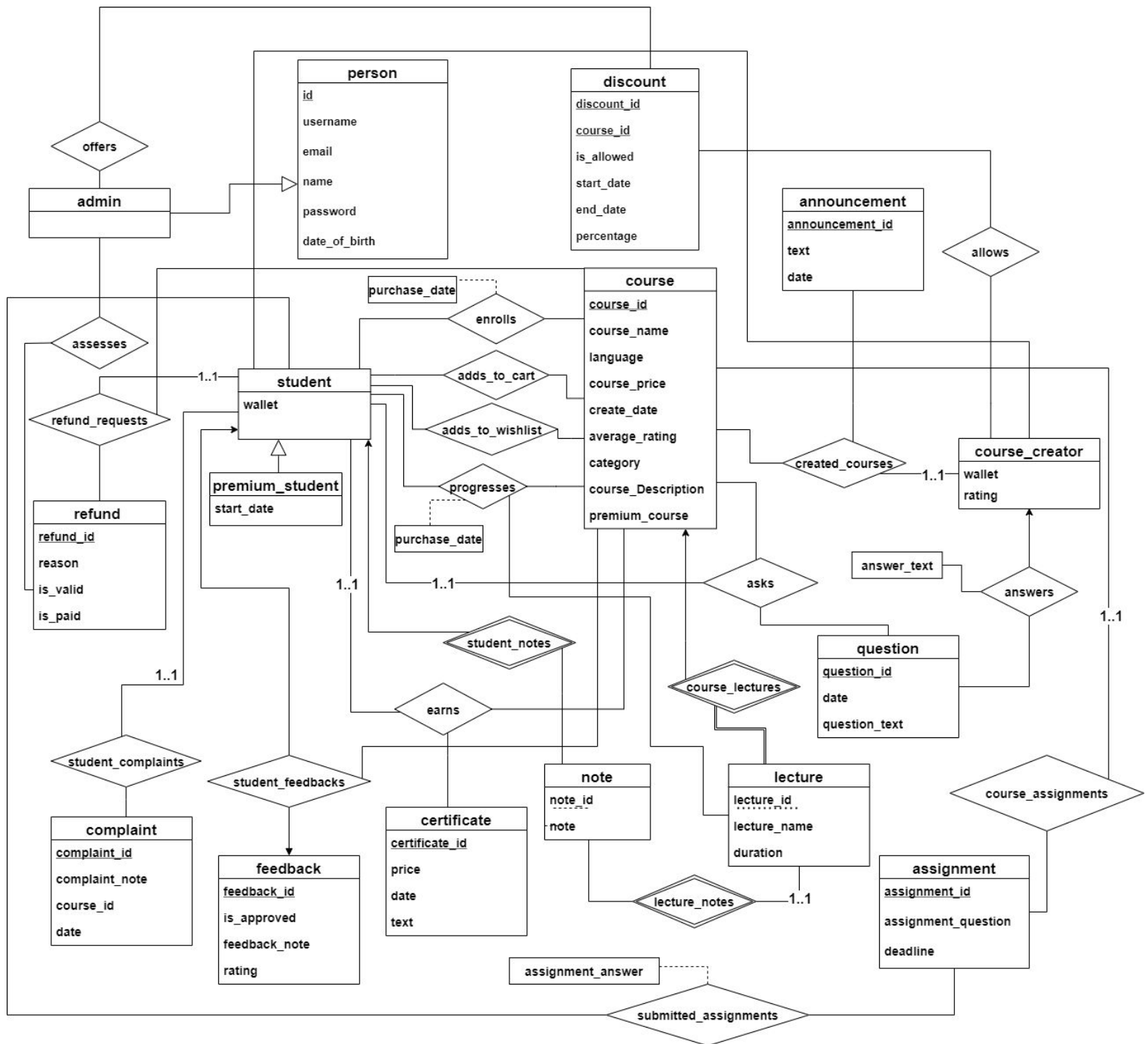


Fig.1: Entity Relationship Diagram for the online course platform database system.

6. Website

Information and updates about the project can be found at the following link:

<https://oguzkaanimamoglu.github.io/Online-Course-Platform/>