# CHAPTER FOUR

# SYSTEM DESIGN

## 

## 4.1. INTRODUCTION

We were trying to show you detail about the system feature of proposed system in the chapter three. In this chapter of the we will develop the system design. It includes and explains logic view, process view, deployment diagram, ER diagram and Database design. The transition of the analytical model into a system design model is known as system design. System design is the process of defining the architecture, interfaces, and data for a system to satisfy specific requirements [1]. It involves creating an architecture for different components, interfaces, and modules of a system, and providing data helpful in implementing the platform. It is an iterative process, and the design may change as new information is gathered and requirements evolve. It’s also important to communicate the design effectively to all stakeholders, including an administrator, the instructor, the learners to ensure that the system meets their needs and expectations.

The primary goal of system design is to demonstrate how the system is developed and both a student and a lecturer grasp a real information about how a new student can access the platform and also how an instructor can provide a quiz, lecture videos.

## 4.2. Purpose of the System Design Document (SDD)

The purpose of this System Design Document (SDD) is to specify the detailed architecture and system design of the Online Learning Platform (OLP). The OLP aims to provide a comprehensive, interactive, and user-friendly platform for learners of all ages and educators across various disciplines.

The SDD is intended for use by the project team to ensure a clear understanding of the system's components and their interactions. It will guide the development process, facilitate accurate estimation of costs, and serve as a reference for future system enhancements. The ultimate goal of the online learning platform, as guided by this SDD, is to create an accessible, engaging, and effective online learning environment that empowers learners and educators, promotes active learning, and fosters educational achievement.

## 4.3. Design Goal

A design goal is a purpose or intention which directs the design process. It is a formulation of the expected design characteristics. Design goals are usually solution oriented and user-driven, concentrating on the solving of a problem rather than the problem itself. Basically, a design goal is the target that a design process tries to attain. It can be used as a reference for all design considerations and checks whether the design is aligned with the needs of its potential users [2].  
The best aspects of our system that should be maximized are specified by the design goals. As well, develop a great quality, unique features, solid as well as efficient online learning platform. Some of the objectives are given below:

1. **User-Friendly Interface: The platform should have an intuitive and easy-to-navigate interface to ensure a seamless user experience.**
2. **Accessibility: A platform should work on several devices and browsers, and support disabled learners.**
3. **Engaging Content Delivery: The platform should have different types of content such as videos, slides, quizzes etc. to accommodate different learning styles and retain users.**
4. **Interactive Learning: The platform ought to provide means for learners and instructors to interact, as well as among learners, and this can be realized through features such as discussion forums.**
5. **Personalized Learning Experience: The platform should provide individualized learning paths taking into account the learner’s progress, interests and achievements.**
6. **Performance Tracking: The platform needs accurate tracking and reporting functions to trace the trajectory of learners’ progress and results**
7. **Security and Privacy: The platform should provide the security and privacy of the users’ data.**
8. **Scalability: The platform must be scalable-able to fit the needs of an increasing number of users and courses.**
9. **Support and Help: There should be reliable and prompt support on the platform to deal with any challenges that the users may encounter.**
10. **Continuous Improvement:** The platform should have mechanisms for collecting user feedback and making continuous improvements based on the feedback.

### 4.4. Architectural Design

### 4.4.1. Logical View of the Architecture

A **logical view** is a way of representing a system’s functionality from a high-level perspective [3]. It is concerned with the **functional requirements**of the system and how it provides value to the end-users, it is a way of describing the functionality and structure of a system from the perspective of different stakeholders, such as end-users, developers, or system engineers. It can be represented by various diagrams, such as class diagrams, state diagrams, or package diagrams, depending on the level of abstraction and the purpose of the view we will use package diagram to since it provides a clear and concise representation of our online learning platform.  
In our proposed system, we've structured the architecture into six distinct packages.

**The User Package:** encapsulates the system's roles, including administrators, educators, and learners. It delineates the permissions and actions associated with each user type, forming the backbone of user interactions.

**The User Interface Package** complements this by providing tailored interfaces for each user category, along with the essential sign-in and sign-up interfaces, facilitating seamless user interactions.

**The Content Management Package:** At the heart of our project lies on this package This package oversees the creation, organization, and management of learning materials. With subcomponents like course management, forum management, and resource management, it ensures an engaging and organized learning environment, overseen by administrators and educators.

**The** **Communication Package:** fosters collaboration and interaction among users through forums, feedback mechanisms, and blogs. This enhances the learning experience and promotes a sense of community within the platform.

**The Assessment Package:** contributes to the learning process covering functionalities like exams, grading, and certificates. It gives the tools to create, administer and assess which secures the accuracy of measurement of learner performance and progress.

**The Database Management Package:** is the backbone of the system, dealing with the data storage and authentication procedure. It ensures a secure access to the platform, holds user data, content assets and system configurations.

Every package is significant in molding the user experience, supporting content provision, and keeping the online learning platform functionally seamless. These packages allow for smooth integration and combine forces to importantly, achieve the platform's effectiveness, usability, and scalability.

Here is a logical view look like

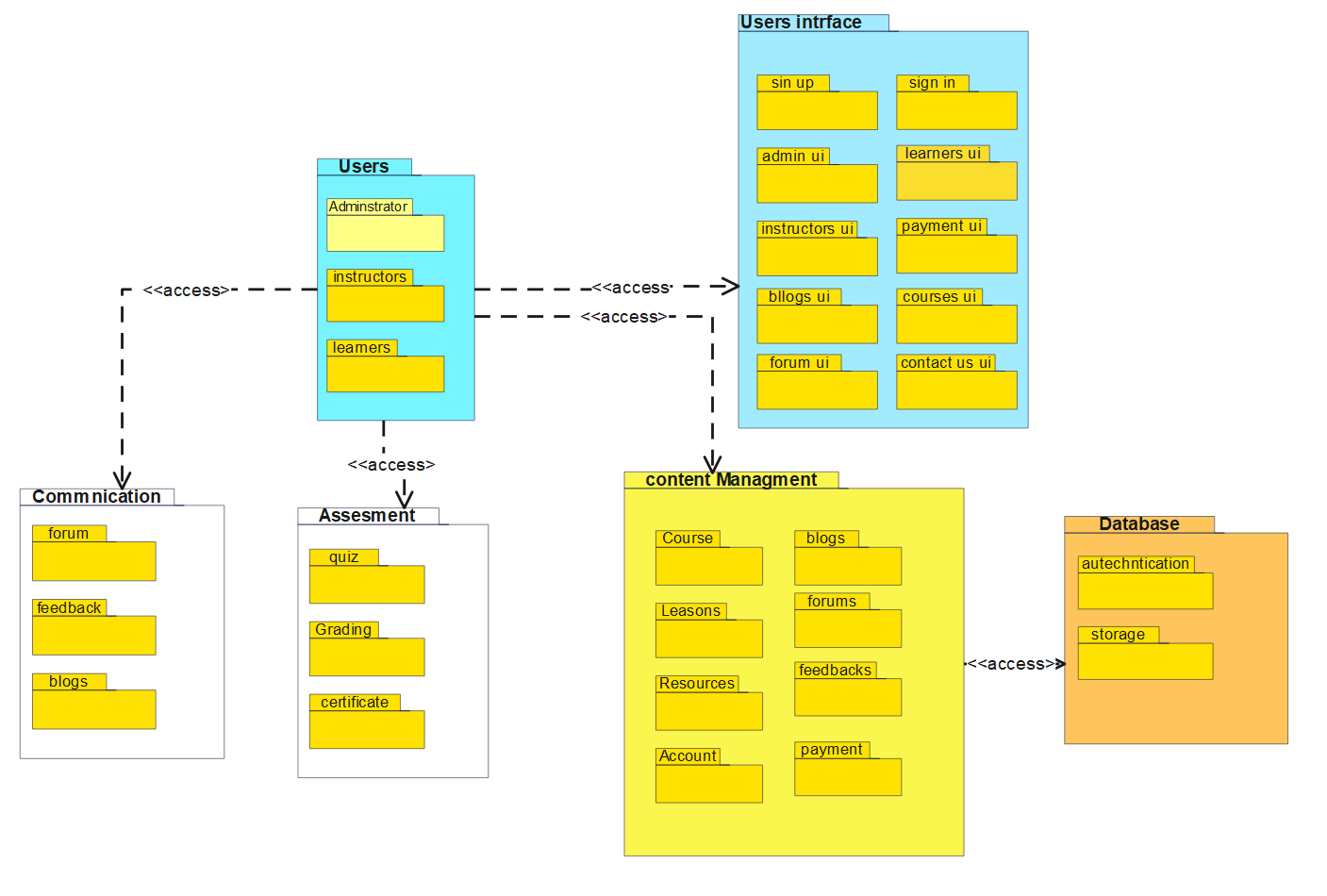


Figure: logical view

### 4.4.2. Process View

Process View is a way of looking at work as a series of steps or actions that transform inputs into outputs. It helps to understand how different processes interact and communicate with each other, and how they affect the performance and quality of the system. Process View can be used to design, analyze, improve, and manage work processes of our platforms.

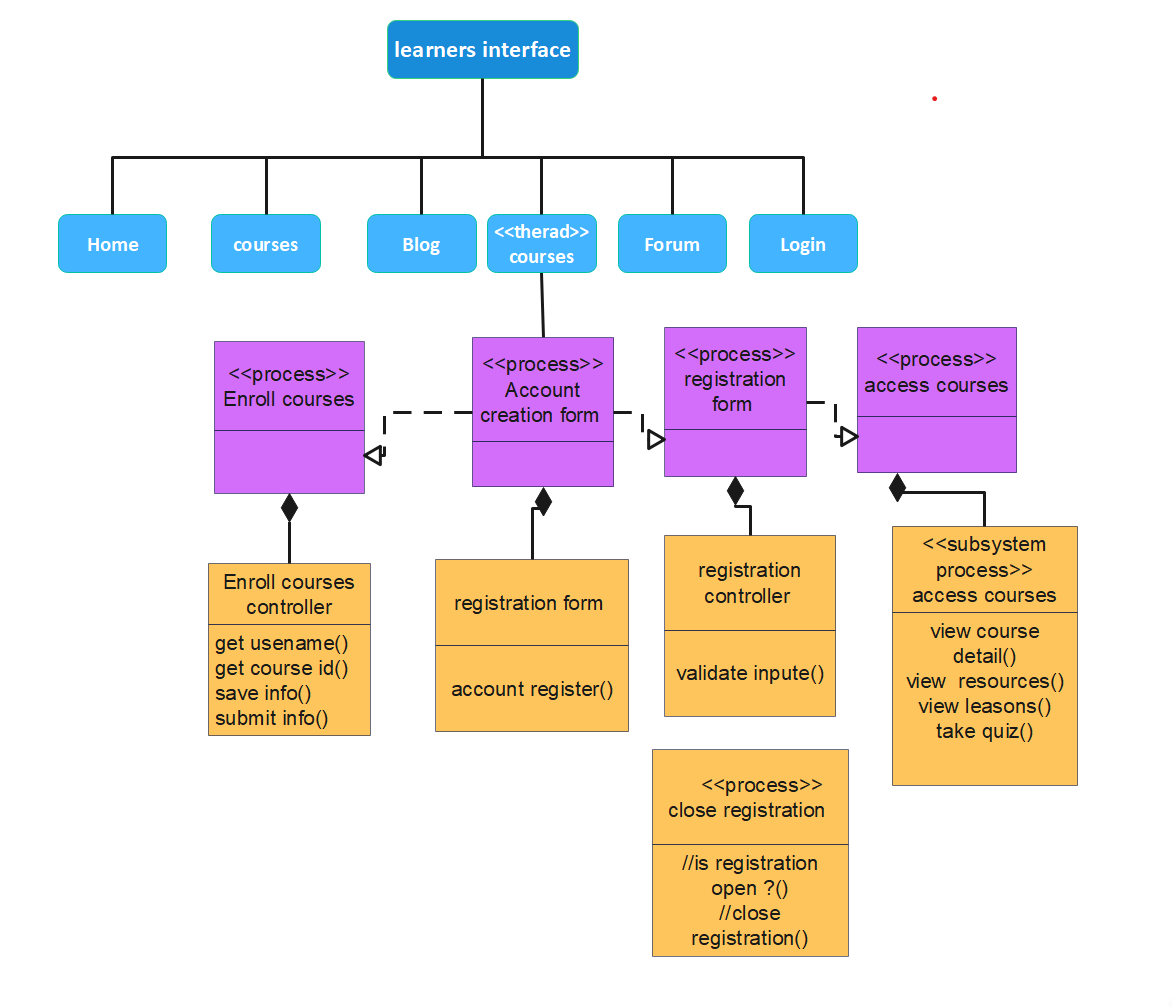


Figure: process view

### 4.4.3. Deployment View

A **deployment view** is a type of **structure diagram** that models the physical aspects of an object-oriented system. It is often used to represent the topology **of the hardware**and the **configuration of run time processing nodes** and the**components** that live on them [4].

A deployment view is a way of showing how a system is distributed and executed on different hardware and software components. It can also show the communication and dependencies between these components. A deployment view can help to plan, design, and manage the system’s infrastructure and performance. It is useful for visualizing, specifying, and documenting embedded, client, and distributed systems and also for managing executable systems. It describe how the operator can perform throughout the platform.

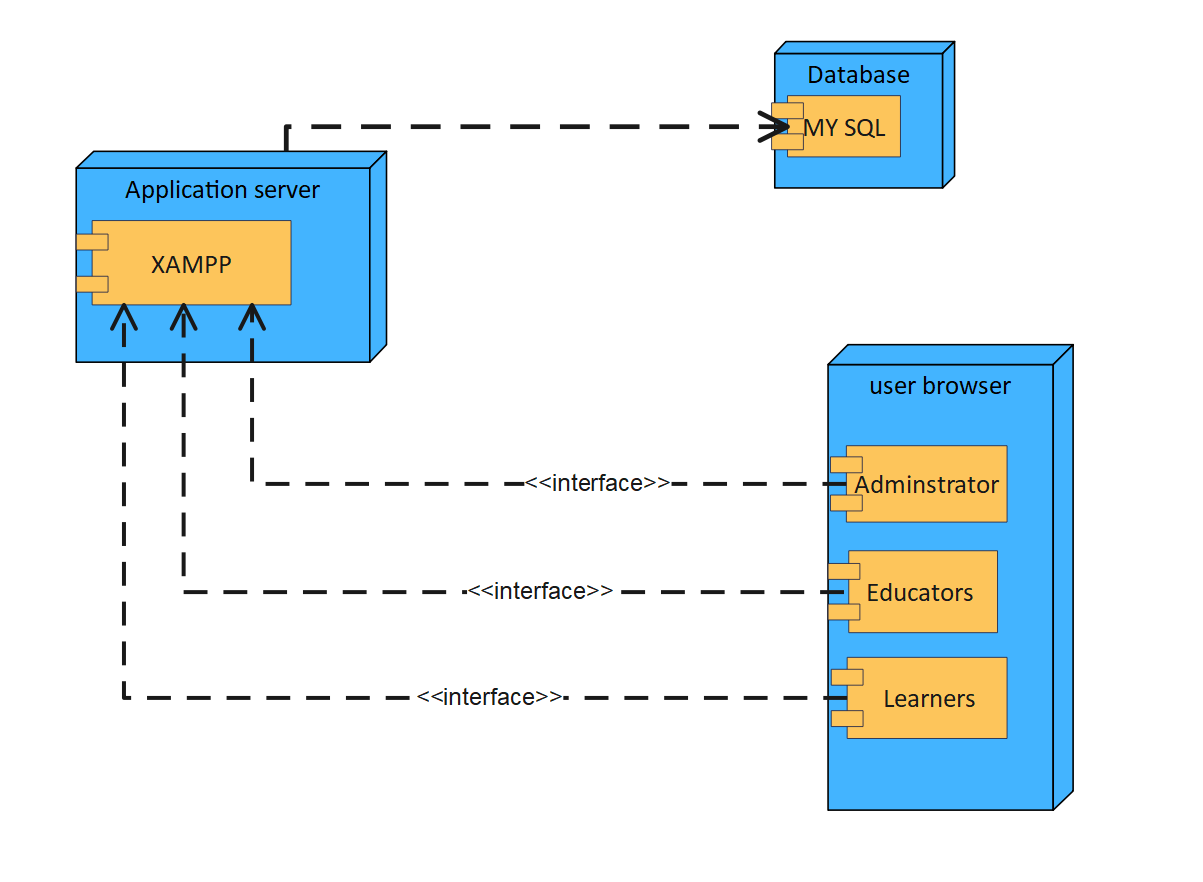


Figure: deployment view

## 4.5. Database Design

**Database design** is the process of organizing data according to a database model. The designer or database administrator can determine what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. A **database management system** manages the data accordingly [5].

**The process of database design involves classifying data and identifying interrelationships. By determining the relationships and dependencies among the different pieces of data it is possible to organize the data into a logical structure which can then be mapped into the storage objects supported by the database management system. The data represent types as courses, course categories, students, course enrollments teachers, classes, attendance, exams, and scores.  
The main objectives of database design in DBMS are to construct logical and physical designs models of the database system under consideration. A well-designed database should be capable of handling a large amount of data and at the same time provide faster access to the data. It should also provide data integrity and security. To elaborate Logical design involves mapping the conceptual design to a specific data model, such as relational, hierarchical, or network, and defining the constraints, keys, and indexes for each entity. Physical design involves choosing the storage structures, file organizations, and access methods that optimize the performance, security, and availability of the database. the logical model is largely focused on data needs, and considerations must be made in terms of monolithic concerns, and so the stored physical data must be stored independently of physical conditions. The physical database design model, on the other hand, involves a translation of the logical design model of the database by maintaining control of physical media utilizing hardware resources and software systems such as Database Management System (DBMS).**

The following points describe the significant considerations that may be taken into account while stressing the importance of database design.

* Ensures **simplicity**: A well-designed database makes it easy to write queries and access data in a user-friendly way.
* Eliminates **redundancies**: A good database design avoids unnecessary data duplication, which saves disk space.
* Enables **analysis**: A structured database design allows for effective data retrieval, reporting, and analytics, which can help online learning platforms measure and improve their performance.
* Maintains **accuracy**: A reliable database design ensures that the data stored is valid, complete, and up-to-date.

|  |  |  |
| --- | --- | --- |
|  | **Instructors** | **Data Type** |
| **PK** | l\_id | Int (11) |
|  | F\_name | Varchar (255) |
|  | L\_name | Varchar (255) |
|  | l\_qualification | Varchar (255) |
|  | l\_img | text |
|  | email | text |
|  | password | text |

|  |  |  |
| --- | --- | --- |
|  | **Admin** | **Data Type** |
| **PK** | A\_id | Int (11) |
|  | Username | Varchar (255) |
|  | email | text |
|  | password | Varchar (255) |

|  |  |  |
| --- | --- | --- |
|  | **courses** | **Data Type** |
| PK | course\_id | Int (11) |
|  | course\_name | Varchar (255) |
|  | course\_desc | text |
| FK | c\_instructor\_id | Int (11) |
|  | course\_img | text |
|  | course\_duration | Int (11) |
|  | course\_price | float |
|  | course lessons | Int (11) |

|  |  |  |
| --- | --- | --- |
|  | **students** | **Type** |
| **PK** | stu\_id | Int (11) |
|  | f\_name | Varchar (255) |
|  | l\_name | Varchar (255) |
|  | Birth day | Datetime |
|  | City | Varchar (255) |
|  | stu\_email | Varchar (255) |
|  | stu\_pass | Varchar (255) |
|  | stu\_occ | Varchar (255) |
|  | stu\_img | Text |

|  |  |  |
| --- | --- | --- |
|  | **Lesson** | **Data Type** |
| **PK** | lesson\_id | Int (11) |
| **FK** | L\_course\_id | Int (11) |
|  | lesson\_name | Varchar (255) |
|  | lesson\_link | Varchar (255) |

|  |  |  |
| --- | --- | --- |
|  | **Course order** | **Type** |
| PK | order\_id | Varchar (255) |
| FK | C\_stu\_id | Int (11) |
| FK | course\_id | Int (11) |
|  | amount | int (11) |
|  | date | datetime |
|  | L\_stu\_name | Varchar (255) |
|  | course\_name | varchar (255) |

|  |  |  |
| --- | --- | --- |
|  | **Exam question** | **Data Type** |
| **PK** | id | int (11) |
| FK | e\_category | Varchar (255) |
|  | ques\_no | Int (11) |
|  | question | Varchar (255) |
|  | opt1 | Varchar (255) |
|  | opt2 | Varchar (255) |
|  | opt3 | Varchar (255) |
|  | opt4 | Varchar (255) |
|  | answer | Varchar (255) |

|  |  |  |
| --- | --- | --- |
|  | **Exam Result** | **Data Type** |
| PK | id | Int (11) |
| FK | E\_stu\_id | Int (11) |
|  | exam type | Varchar (255) |
|  | total\_question | Int (11) |
|  | correct\_answer | Int (11) |
|  | wrong\_answer | Int (11) |
|  | exam\_time | Datetime |
|  | mark | Int (11) |

|  |  |  |
| --- | --- | --- |
|  | **Exam Category** | **Data Type** |
| PK | id | Int (11) |
|  | exam\_name | Varchar (255) |
|  | exam\_time | Varchar (255) |

|  |  |  |
| --- | --- | --- |
|  | **Materials** | **Data Type** |
| **PK** | material\_id | Int (11) |
| **FK** | M\_course\_id | Int (11) |
| **FK** | M\_lesson\_id | Int (11) |
|  | material\_type | Varchar (50) |
|  | material\_url | text |
|  | upload\_date | datetime |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | **Answer** | **Data type** |
| PK | A\_id | Int (11) |
| FK | Q\_id | Int (11) |
| FK | A\_stu\_id | Int (11) |
|  | A\_body | long text |
|  | likes | Int (11) |
|  | a timestamp | Datetime |

|  |  |  |
| --- | --- | --- |
|  | **Contact** | **Data Type** |
| **PK** | id | int (11) |
|  | f\_name | Varchar (255) |
|  | l\_name | Varchar (255) |
|  | email | text |
|  | msg | text |

|  |  |  |
| --- | --- | --- |
|  | **Form\_question** | **Data Type** |
| PK | Q\_id | int(11) |
| FK | Q\_stu\_id | int(11) |
|  | q\_body | text |
| FK | q\_course\_id | int(11) |
|  | q\_timestamp | datetime |
|  | resolved | varchar(4) |

|  |  |  |
| --- | --- | --- |
|  | **Certificate** | **Data Type** |
| **PK** | C\_id | Int (11) |
| **FK** | Course\_id | Int(11) |
| **FK** | stu\_id | Int (11) |
|  | certificate | text |
|  | course\_name | Varchar (255) |
|  | issue\_date | date |
|  | completion\_status | Boolean |

|  |  |  |
| --- | --- | --- |
|  | **Notification** | **Data Type** |
| **PK** | notification\_id | Int (11) |
| **FK** | N\_stu\_id | Int (11) |
| **FK** | material\_id | Int (11) |
|  | is\_read | Boolean |
|  | notification\_dat e | datetime |

|  |  |  |
| --- | --- | --- |
|  | **Feedback** | **Data Type** |
| **PK** | f\_id | Int (11) |
|  | f\_content | text |
| **FK** | f\_stu\_id | Int (11) |

|  |  |  |
| --- | --- | --- |
|  | **Blogs** | **Data Type** |
| **PK** | b\_id | Int (11) |
|  | b\_title | text |
|  | b\_dec | text |
|  | b\_img | text |
|  | Posted\_time | datetime |

|  |  |  |
| --- | --- | --- |
|  | **students’ progress** | **Data Type** |
| **PK** | progress\_id | Int (11) |
| **FK** | P\_stu\_id | Int (11) |
| **FK** | P\_course\_id | Int (11) |
| **FK** | P\_lesson\_id | Int (11) |
|  | completed | Booleans |
|  | progress\_date | date |

### 4.4.1. ER Diagram

An entity relationship model is a way of representing the data and its relationships in a database.  It uses symbols to show the entities, attributes, and relationships that are relevant to a specific domain of knowledge. An entity relationship model can help to design and understand a database by showing the logical and physical structure of the data and how it can be manipulated [6]. An E-R diagram, or entity-relationship diagram, is a way of representing the data and its relationships in a database.

It uses symbols to show the entities, attributes, and relationships that are relevant to a specific domain of knowledge. An E-R diagram can help to design and understand a database by showing its logical and physical structure. An E-R diagram consists of symbols that represent entities, attributes, and relationships. An entity is a thing of interest that can be identified and distinguished from others. An attribute is a property or characteristic of an entity. A relationship is an association or connection between two or more entities.

The following diagram on the next page is the entity relationship:

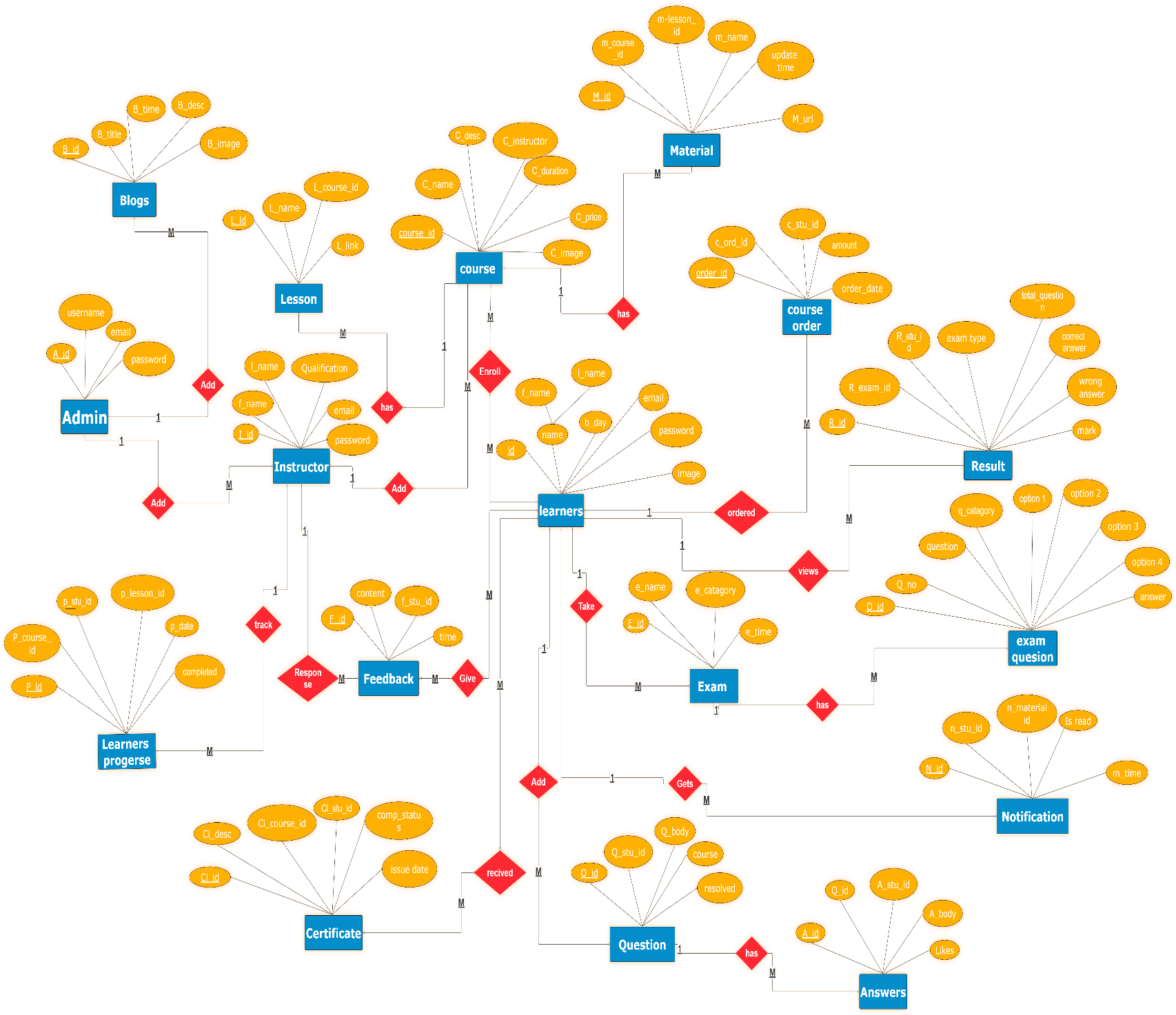


Figure: ER-diagram

### 4.5.2. Relational Mapping & Database Normalization

Database Normalization is a stepwise formal process that allows us to decompose database tables in such a way that both data dependency and update anomalies are minimized. It makes use of functional dependency that exists in the table and the primary key or candidate key in analyzing the tables [7].

**1st normalization form**

In the first normal form, each table must have a unique identifier, and each attribute must contain atomic values. The provided tables below demonstrate the 1st normal form of the proposed system tables, as each table has a primary key, and attributes contain atomic values.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Exam question | | | | | | | | |  |  |  |  |
| id | ques\_no | question | opt1 | opt2 | opt3 | opt4 | answer | category | correct\_answer | wrong\_answer | exam\_time | mark |

|  |  |  |  |
| --- | --- | --- | --- |
| admin | | | |
| id | email | password | username |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| questions | | | | | | |
| Q\_id | q\_body | q\_timestamp | resolved | answer | Answer\_body | Answer time |

|  |  |  |  |
| --- | --- | --- | --- |
| blog | | | |
| b\_id | b\_title | b\_dec | b\_img |

|  |  |  |  |
| --- | --- | --- | --- |
| certificates | | | |
| id | certificate | issue\_date | completion\_status |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| contact | | | | |
| id | f\_name | l\_name | email | msg |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| course | | | | | | | | |
| course\_id | course\_name | course\_desc | Course order | amount | date | course\_img | course\_duration | course\_price |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| instructors | | | | | |
| l\_id | l\_name | l\_qualification | l\_img | email | password |

|  |  |  |
| --- | --- | --- |
| lesson | | |
| lesson\_id | lesson\_name | lesson\_link |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| materials | | | | | |  |
| material\_id | material\_type | material\_url | upload\_date | notification | notification\_date | notification\_date |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| students | | | | | | | | |
| stu\_id | f\_name | l\_name | stu\_email | stu\_pass | stu\_phone | stu\_img | feedback | F\_content |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| student\_progress | | | | | |
| progress\_id | stu\_id | course\_id | lesson\_id | completed | progress\_date |

**2nd normalization form**

In the second normal form, a table is in 2NF if it is in 1NF, and every non-prime attribute is fully functionally dependent on the entire primary key.

To modify the following tables from 1NF to 2NF:

1. Splitting learners Table into Feedback and Learner: The Learner table contains attributes related to both the learner and their feedback. Therefore, we create a new table called Feedback with attributes dependent on feedback.

2. Splitting Exam Question Table into Exam Question, Exam Result, and Exam Category: The Exam Question table contains attributes related to both the question and the exam result. We create a new table called Exam Result with attributes dependent on the exam result and Exam Category to store information about exam categories.

3.Splitting Question Table into Questions and Answers: Since the Question table contains both questions and answers, we create a new table called Answers with attributes related to answers.

4. Splitting Courses Table into Course and Course Order: The Courses table contains both course information and order information. We create a new table called Course Order with attributes related to course orders.

5. Splitting Material Table into Material and Notifications: The Material table contains both material information and notification information. We create a new table called Notifications with attributes related to notifications and remove the notification-related attributes from the Material table.

By following these steps, each table contains only attributes that are fully functionally dependent on the primary key, thus achieving the second normal form. The 2nd normal form tables are provided below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| students | | | | | | | |
| stu\_id | f\_name | l\_name | stu\_email | stu\_pass | stu\_phone | stu\_img | stu\_course(fk) |

|  |  |  |
| --- | --- | --- |
| feedback | | |
| f\_id | f\_content | f\_stu\_id (FK) |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Exam question | | | | | | | |
| id | ques\_no | question | Option 1 | Option 2 | Option 3 | Option 4 | category \_id (fk) |

|  |  |  |
| --- | --- | --- |
| exam category | | |
| id | exam\_name | exam time |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| exam result | | | | | | | |
| r\_\_id | r\_stu\_id (FK) | Question\_id (fk) | total question | correct\_  answer | wrong\_answer | exam\_time | mark |

|  |  |  |  |
| --- | --- | --- | --- |
| questions | | | |
| Q\_id | Q\_stu\_id (FK) | q\_body | course\_id (FK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| answers | | | | | |
| A\_id | Q\_id (FK) | A\_stu\_id  (FK) | A\_body | likes | a\_timestamp |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| course | | | | | | |
| course\_id | course\_name | course\_desc | course\_  instructor(FK) | course\_img | course\_duration | course\_price  (FK) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| courseorder | | | | |
| order\_id | stu\_id (FK) | course\_id (FK) | amount | date |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| materials | | | | | |
| material\_id | course\_id (FK) | lesson\_id (FK) | material\_type | material\_url | upload\_date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| notifications | | | | |
| notification\_id | stu\_id (FK) | material\_id (FK) | is\_read | notification\_date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| contact | | | | |
| id | f\_name | l\_name | email | msg |

|  |  |  |  |
| --- | --- | --- | --- |
| admin | | | |
| id | email | password | username |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| instructors | | | | | |
| l\_id | l\_name | l\_qualification | l\_img | email | password |

|  |  |  |  |
| --- | --- | --- | --- |
| blog | | | |
| b\_id | b\_title | b\_dec | b\_img |

|  |
| --- |
| student\_progress |
| progress\_id | stu\_id (FK) | course\_id (FK) | lesson\_id (FK) | completed | progress\_date |

|  |  |  |  |
| --- | --- | --- | --- |
| lesson | | | |
| lesson\_id | lesson\_name | lesson\_link | course\_id (FK) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| certificates | | | | | |
| id | certificate | issue\_date | completion\_status | stu\_id (FK) | course\_id (FK) |

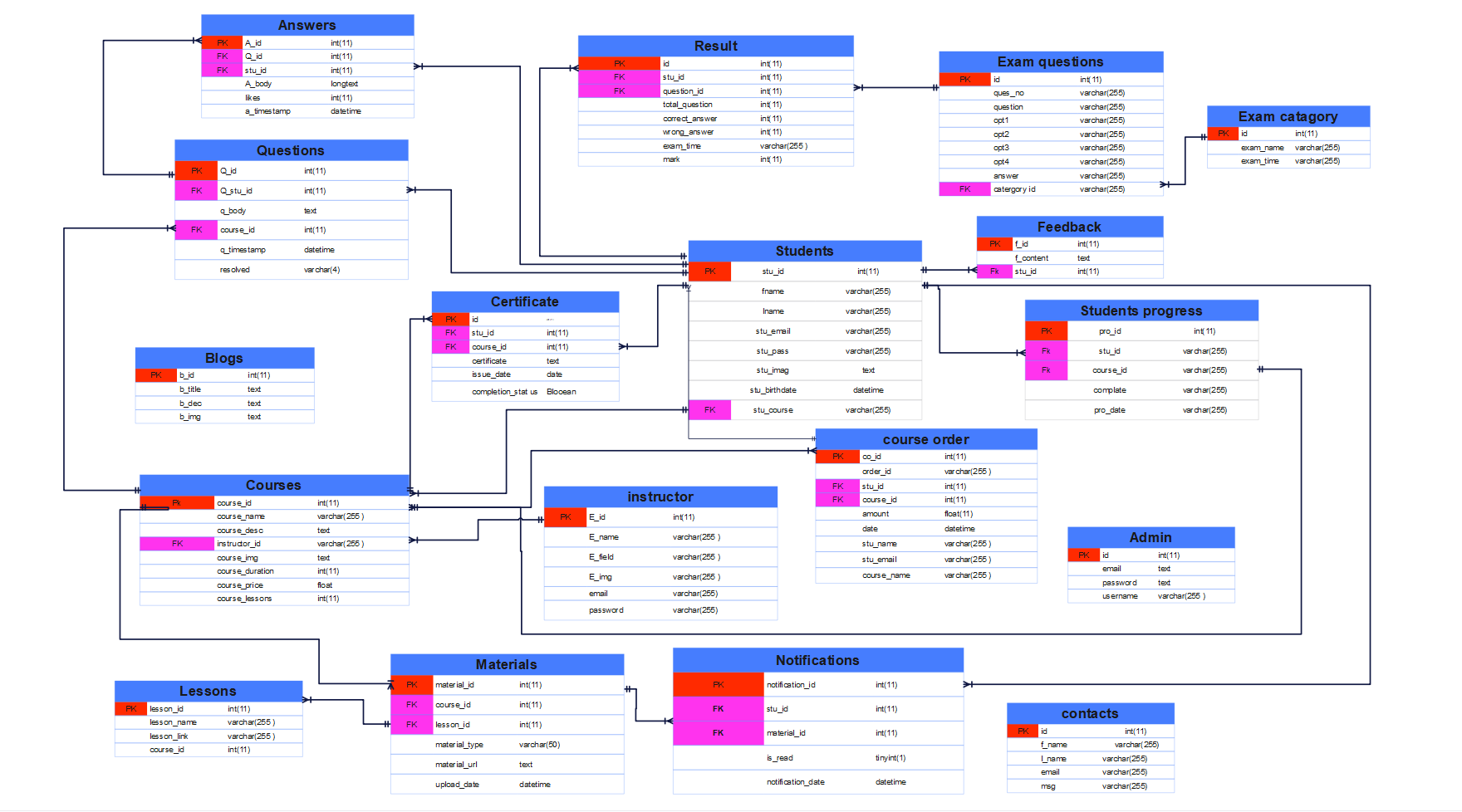


Figure: Relational Mapping

# References

|  |  |
| --- | --- |
| [1] | E. Schaffer, "Educative," 13 septamber 2024. [Online]. Available: https://www.educative.io/blog/complete-guide-to-system-design. [Accessed 9 feburary 2024]. |
| [2] | 28 january 2020. [Online]. Available: https://www.linkedin.com/pulse/what-design-diane-bowen/. [Accessed 10 feburary 2024]. |
| [3] | "Visual Paradigm Community Circle," 03 march 2020. [Online]. Available: https://circle.visual-paradigm.com/docs/visual-paradigm-getting-started/user-interface/logical-view/. [Accessed 10 feburary 2024]. |
| [4] | "visual paradigm," [Online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/. [Accessed 11 feburary 2024]. |
| [5] | "Wikipedia," january 2024. [Online]. Available: https://en.wikipedia.org/wiki/Database\_design. [Accessed 11 feburary 2024]. |
| [6] | [Online]. Available: https://en.wikipedia.org/wiki/Entity%E2%80%93relationship\_model. [Accessed 12 feburary 2024]. |
| [7] | "GeeksforGeeks," 06 may 2023. [Online]. Available: https://www.geeksforgeeks.org/normalization-process-in-dbms/. [Accessed 12 feburary 2024]. |

|