

**INFORMATION SYSTEM DESIGN AND SOFTWARE ENGINEERING LAB**

Teacher-Student

Collaboration System

Submitted by

**Mahir Rashid** | 15-01-04-011

**Tasnim Mashrur mahee** | 15-01-04-013

**Atiqul Islam Chowdhury** | 15-01-04-014

**Md. Saiful Islam** | 15-01-04-027

**CONTENTS**

**Topic**  **Page no**

1. Executive Summary of Project 2

2. Reason of Selection of the Project 2

3. Project Goals 3

4. Feasibility Analysis 3

5. Project Scheduling 4

6. Gantt Chart 5

7. Risk Analysis 5

8. Data Flow Diagram 8

9. Use Case Diagram 13

10. ERD Diagram 15

11. Testing result of TSCS 21

12. Limitation 21

13. Future Plan 21

14. Contribution 21

15. Conclusion 21

**Executive Summary of the Project:**

Our project is dedicated to create digital bridge between the teachers and students. It will be great revolution if it will used throughout every educational institution. Because nowadays, there seems to be huge communication gap between teachers and students. So, it is causing a great hamper on the reinless flow of education. What if we create a medium through which teachers and students can have a unopposed communication and without any extra effort? That’s the point of our project. We will bring the teachers and students under the same sky of communication and we hope that this will be a great evolution in the education world.

**Reason of Selection of the Project:**

1. **Reducing Communication Gap:**

Because of being analog until now, students and teachers have a huge communication gap. This project will help to reduce the gap.

1. **Dynamic Share:**

This project will help to create a platform to share information between teachers and students dynamically. Neither teacher nor students need not meet physically to share any information/ material.

1. **Time Consuming:**

Nowadays, to share some information/materials face to face or to inform some notice to the students takes too much time. But this project will help to consume time and rearranging the hotchpotch things.

**Projects Goals /Targets:**

1. **Develop Teacher-Student Relationship:**

This project will shorten the distance between teachers and the students and digitalize the analog process of education.

1. **Covers The All Demands Of User/Buyers:**

We will develop this software in such a way which will fulfill all the primary demands of the user so that it can be usable instantly till the next modified update.

1. **Develop Efficiently:** With easy but standard user interface we will try to develop this software in an efficient way so that this software can run with minimum system requirements.
2. **Marketing:**

After developing we need a proper scope or chance to sell this software to the buyer. Proper advertisement is necessary to spread out among the buyers.

**Project Feasibilities:**

1. **Technical:**

It is possible to develop the system using the current technical resources :

a. .NET Framework

- To provide a consistent object-oriented programming environment

- To provide a code-execution environment that minimizes software deploying

and versioning conflicts

- To provide CLR that comes up for solving memory management, thread

management etc.

b. No need to hire additional programmers, testers or experts.

c. SQL server

- Data security, High performance

- Consistent modification of data

- Complete Workflow control

So, we can say that this project is technically feasible.

1. **Economical:**

We have huge possibilities to make it successful economically because of fastest growing computerized sector everywhere & demand. Economically it will be profitable. Resources used here mostly are free. But initially

- It includes cost of programmers

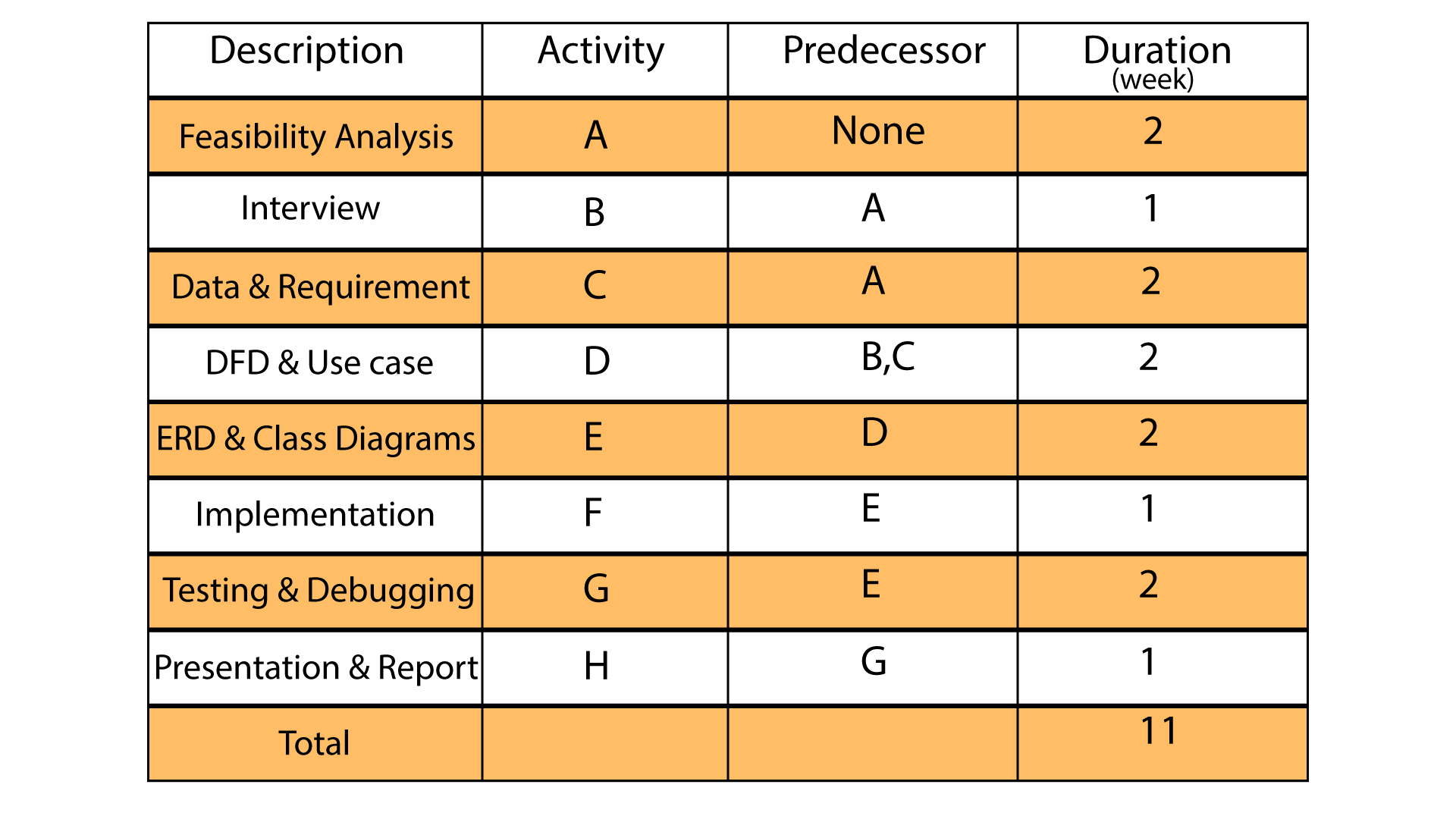
- Cost of studying the full scenario

- Cost of ISP & server

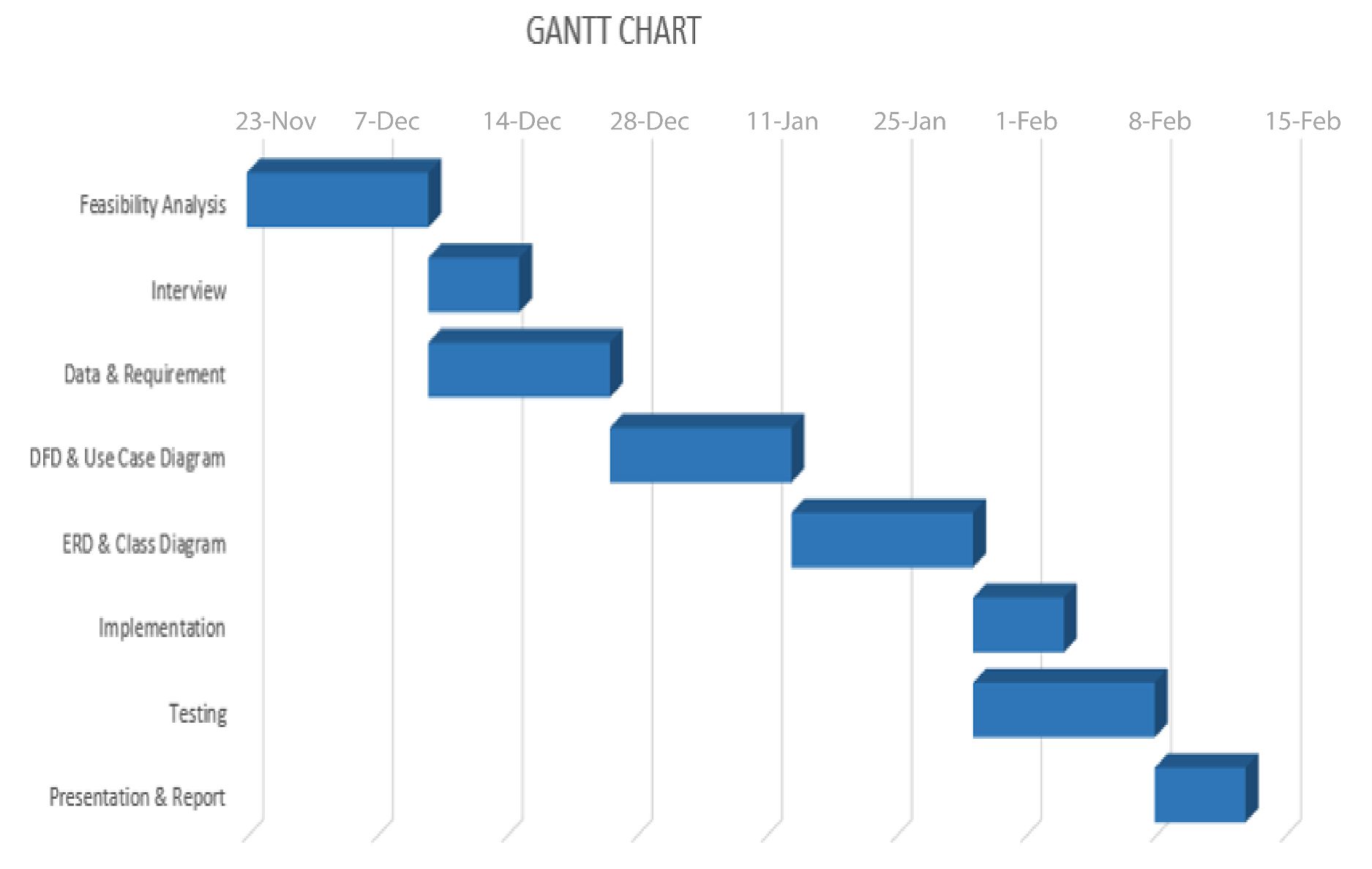
- Other costs like: electricity bill, transportation etc.

1. **Operational:** We will try to develop a 100% fully functional & operational software. But if any bug/error will arise while using, we may introduce user care system for the quick maintenance. As it is reducing time so user will welcome it for their interest. We will provide fast & reliable server. Services are flexible and expandable. We will increase the overall response. Support & user involvement will also be included.

**Project Scheduling:**



**Gantt Chart:**



**Risk Analysis:**

* Server failure
* Unforeseen bug and crush report
* Schedule overruns
* Market competition

# Questionnaires and Interview Pattern:

|  |  |  |  |
| --- | --- | --- | --- |
| **Teacher-Student Collaboration System** | | | |
| **Author:**  Mahir Rashid  Tasnim Mashrur Mahee  Atiqul Islam Chowdhury  MD. Saiful Islam | **Date:** 18/12/2017 | **Time:** 11.30am | **Duration:** 30 min |
| **Participant:**  Murad Hasan Anik  Lecturer, AUST  Ashraful Islam  Lecturer, DIU | | **Comments:**  The following interview has been conducted using a questionnaire to deduce the required information for developing the project. | |

**Selection of Interview Personnel:**

* We have chosen Murad Hasan Anik who is currently a lecturer at AUST and Ashraful Islam who is currently a lecturer at DIU as our honorable interview personnel.
* By talking to them we have come to know what types of procedures they follow, their expectations and their experience.

**Total Interview Procedure:**

**Time:** 18th December, 2017, 11.30 am

**Interview process:** Face to face

**Personnel:** Murad Hasan Anik and Ashraful Islam

**Interview Pattern:** Diamond Pattern

**Questions:**

**Question 1:** Is there any need to create a system for student and teacher communication?

**Ans:** Yes, definitely.

**Question 2:** Is only material (pdf, result, quiz marks) sharing enough?

**Ans**: Quite enough, but it will be better if you add some more features

**Question 3:** Is there any need of creating individual student profile?

**Ans:** Not necessary

**Question 4:** Do you want it for all departments in AUST now?

**Ans:** Do it only for CSE now but make an option for further extension to other department

**Question 5:** Do you want attendance system on it?

**Ans:** Yes, if you can that will be better

**Question 6:** Do you want only desktop version?

**Ans:** It’s difficult to carry laptop always but if you can make it dynamic for mobile user that would be more useable

**Question 7:** Any suggestion?

**Ans:** Its overall a good idea. Make sure you maintain the server and other staffs regularly and make the interface more user friendly.

**Question 8**: From your perspective what should be there?

**Ans:** Comment System

**Question 9:** What do u think about the overall project? Is it beneficial for both?

**Ans:** Yes. I think so.

**Revised Requirements after Collection of Data:**

After the data collection we have come to a decision of adding features like-

* Separate department and individual space for teachers against courses.
* Sharing the study materials provided by teachers.
* Download option for student without overwrite it.
* Giving the results and next update about the upcoming days
* Students will be included against the courses of any teacher.
* As simple process will be delivered for taking attendance in the earliest possible time
* At last teachers can easily deliver their student’s marks against attendance or quiz marks.

**Processes Followed for the Creation of the DFD:**

* Creation of a list of activities
* Construction of a Context Level DFD containing the main process and the External Entities
* Construction of level-0 DFD identifying manageable sub processes and and basic data stores
* Construction of level-1 DFD
* Cross-check to see if all properties are maintained

**Types of Data Flow Diagram:**

* Context Diagram
* Level 1 Diagram
* Level 2 Diagram

**Context Diagram (Level 0):**

DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modeled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

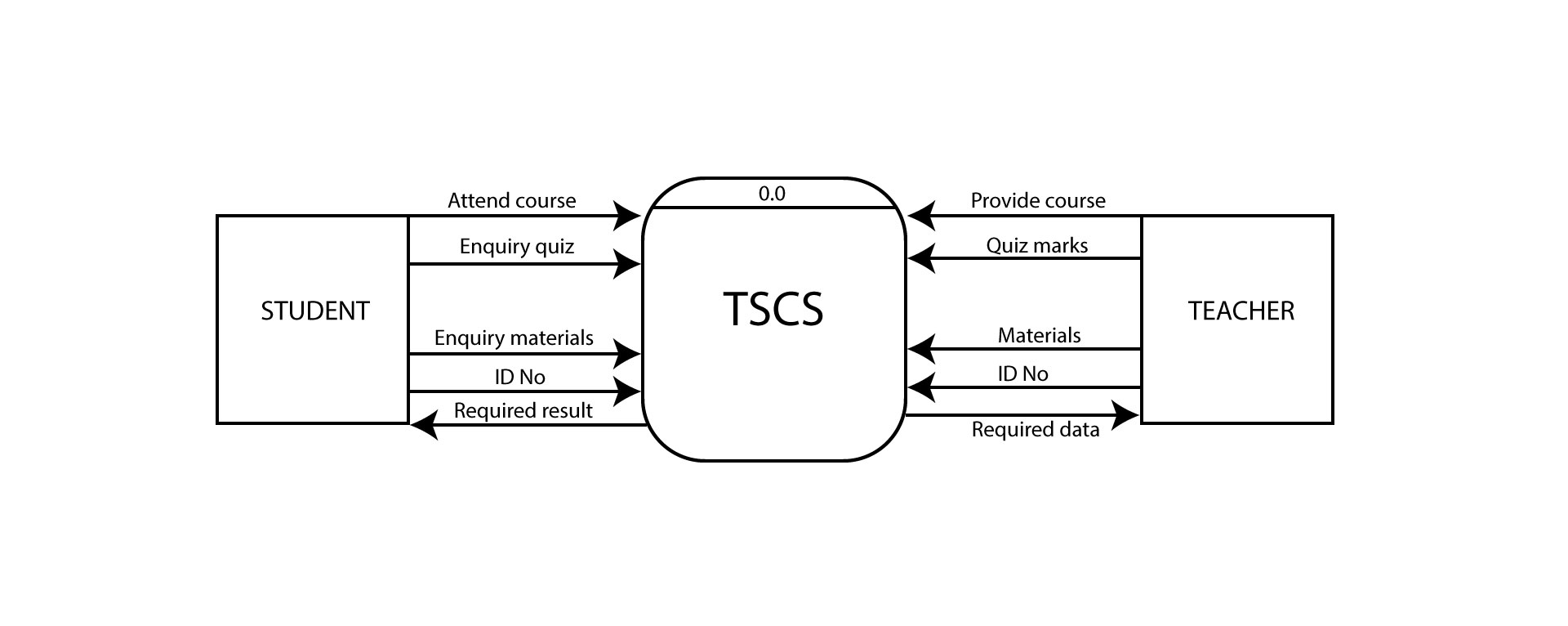


Fig: Context Level Diagram

**Description:**

Here, we have three entities in our TSCS system. They are –Teacher, Student. Student can add course to the system. They can also enquiry quiz, enquiry materials and add an ID to the system. On the other hand, Teacher can provide course, quiz marks, attendance marks, materials to the system, but system can give teacher the required information/data.

**Level 0 diagram:**

DFD Level 0 provides a more detailed breakout of pieces of the Context Level Diagram. We have highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub processes. Entities are same here, just the system are broken into some sub systems.

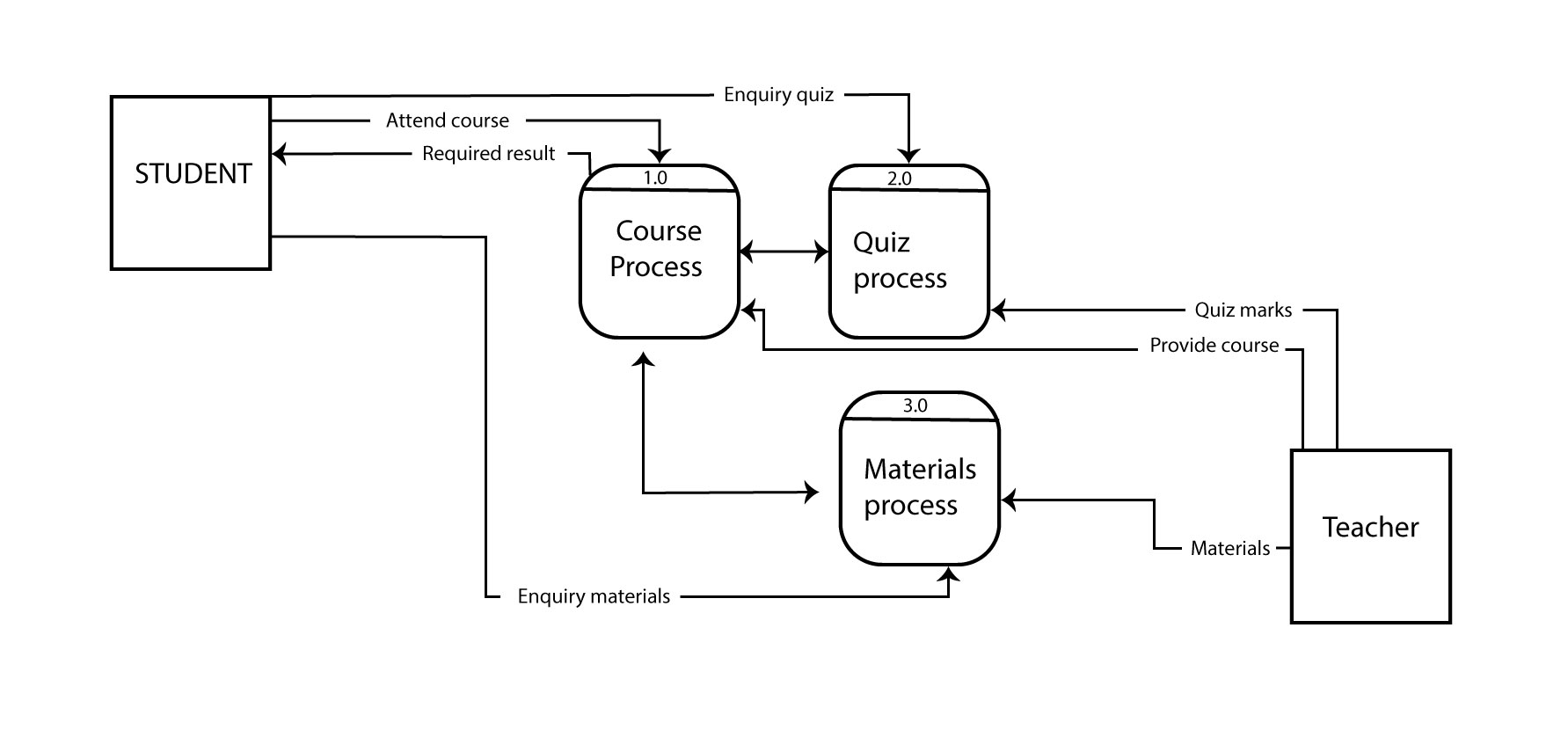


Fig : Level-0 Diagram

**Description:**

Here, the main system is converted into five sub systems. They are – Course process, Quiz process, Materials process, Data accumulation. Entities remain same. First of all, student can attend course and it will be saved to the course process, and course process system provide the result to the student. Similarly, student entity is connected to the quiz process, materials process. There is a connection from both entity and system. Then, for teacher entity, same things are shown here. Teacher can provide materials to the materials process system, give quiz marks to the quiz process system etc.

**Level 1 Diagram:**

DFD Level 1 then goes one step deeper into parts of Level 0. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole.

**Process-1: Course Process:**

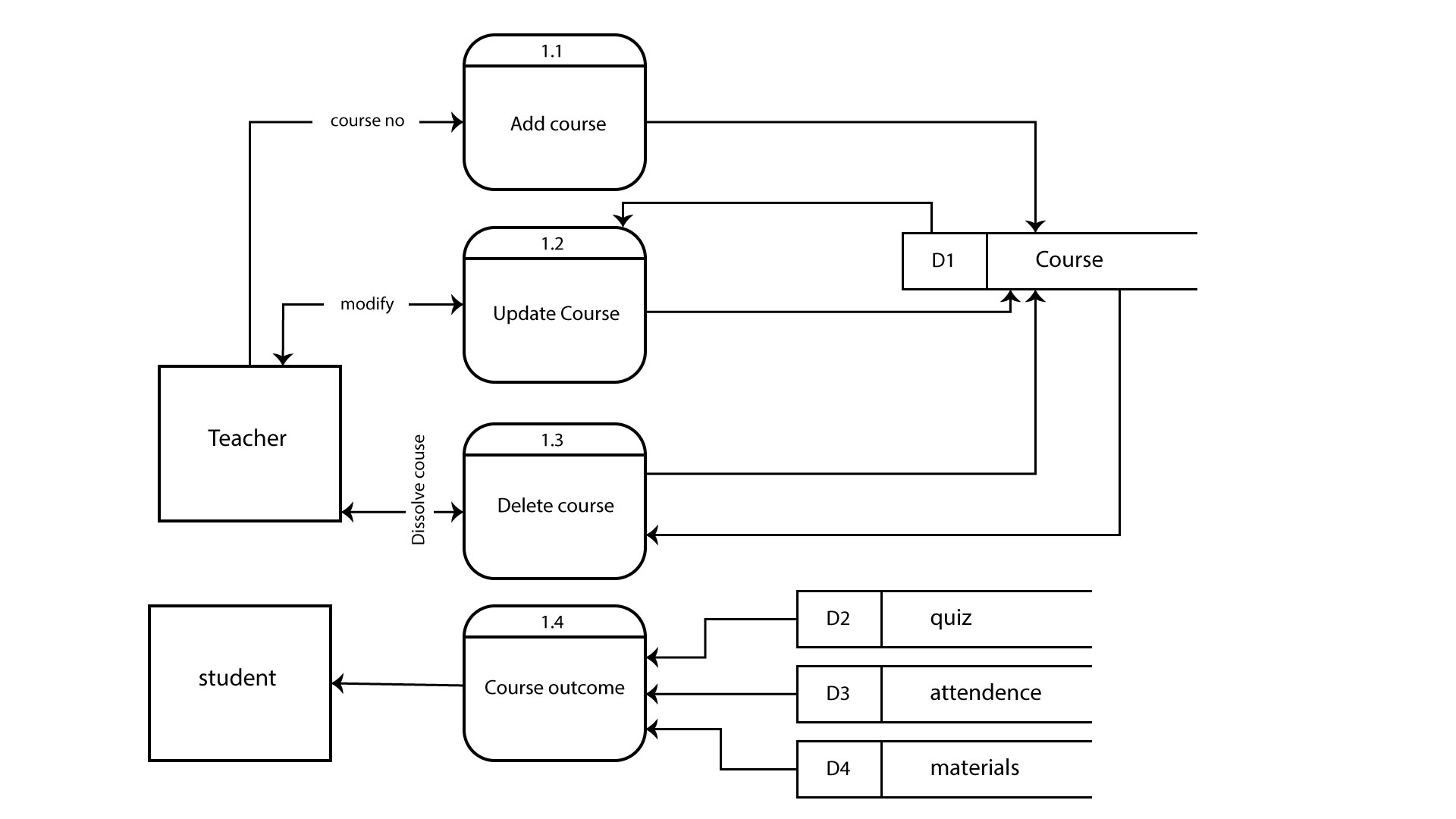
****

Fig : Course Process

In Level-1, the course process is converted into three process- add course, update course, delete course. Teacher can add, update, delete course if he/she has no interest to take it. Then the course related information will be saved in database D1. Teacher can update it later, change it or remove it anytime.

**Process-2: Quiz Process:**

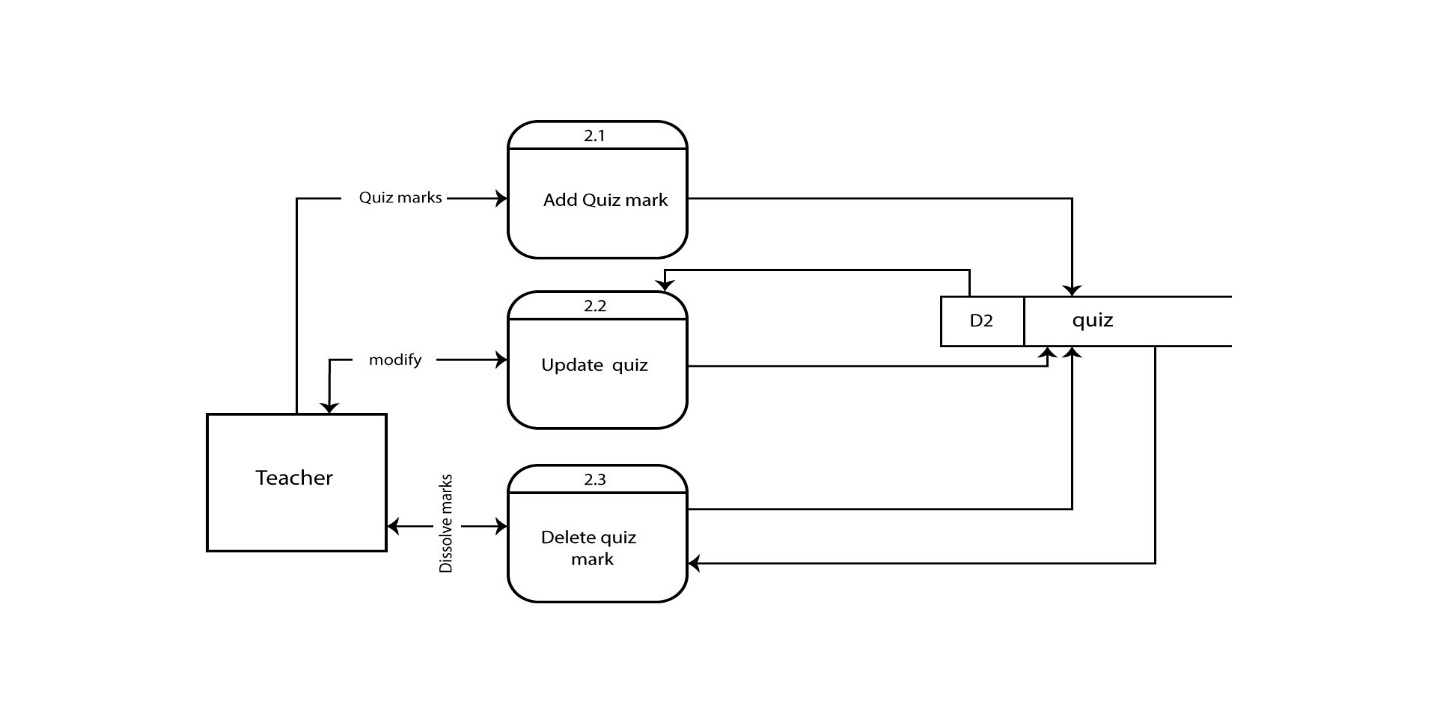


Fig : Quiz Process

In this process, teacher can add quiz mark, delete quiz mark and update quiz mark to the database.

**Process-3: Materials Process:**

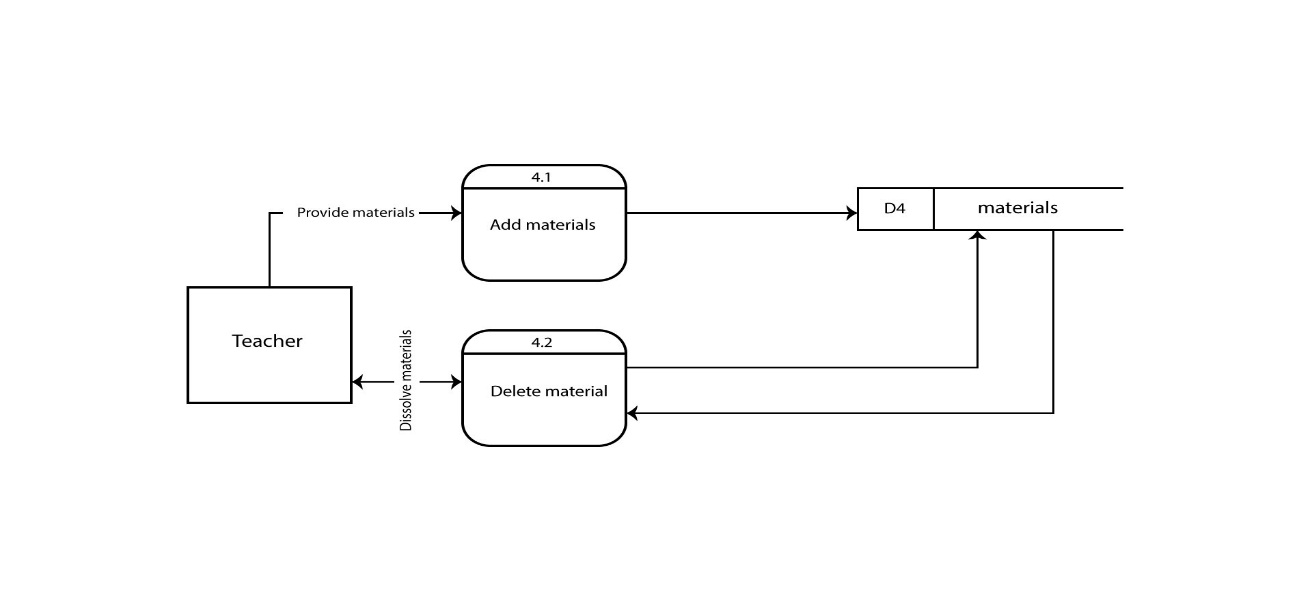


Fig : Material Process

This process depends on teacher. Teacher can add course materials to the system, and if there is no need of some materials, then he/she can delete it easily.

**Use Case Diagram:**

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

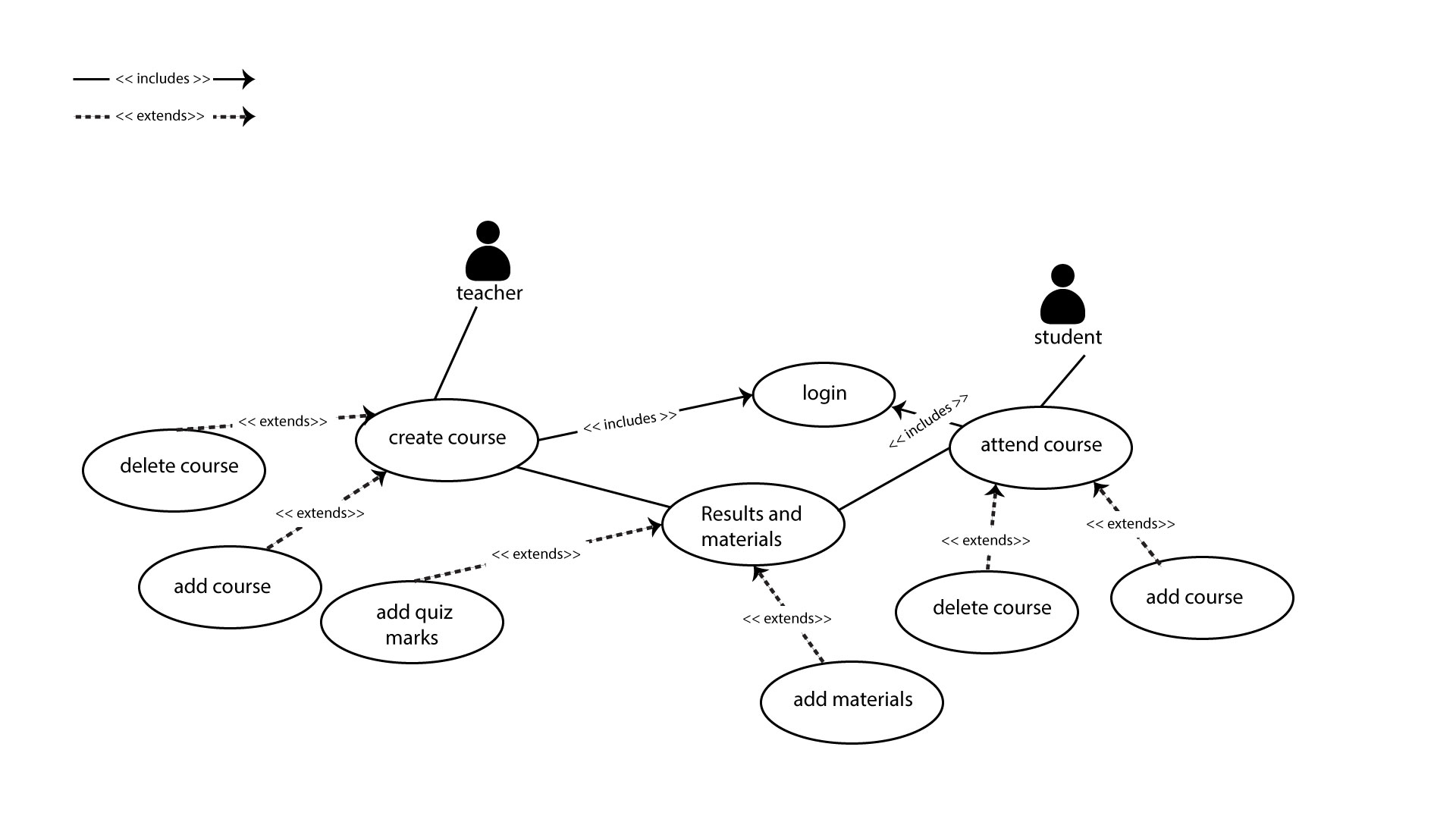


Fig : Use Case Diagram

**Actors:**

* **Teacher:**

Teacher is the most important actor in this system. Any teacher can create course, add marks of quiz, add materials to the system.

* **Student:**

Student can take courses, get marks of quiz and attendance from the system and also get course related information and materials from the system.

**Use Case:**

* **Login**

Actor: Teacher, Student

Pre-Condition: None

Primary path:

* Enter user id
* Enter email id
* Enter password

Alternative path:

* Wrong user id
* Wrong email id
* Wrong password
* **Create Course**

Actor: Teacher

Pre-Condition: Login

Primary path:

* Create a new course
* Provide course information
* Add/Delete course

Alternative path:

* No course taken
* **Attend Course**

Actor: Student

Pre-Condition: Login

Primary path:

* Take a new course
* Get course information
* Add/Delete course

Alternative path:

* No course taken
* **Results and Materials**

Actor: Teacher

Pre-Condition: Create course

Primary path:

* Share materials
* Share quiz and attendance results

Alternative path:

* No materials shared
* No quiz and attendance taken
* **Results and Materials**

Actor: Student

Pre-Condition: Attend course

Primary path:

* Get materials
* Get the quiz and attendance results

Alternative path:

* Found no materials
* Found no quiz and attendance results

**ENTITY-RELATIONSHIP DIAGRAM (ERD):**

The entity-relationship model (or ER model) is a way of graphically representing the logical relationships of entities (or objects) in order to create a database that shows the relationship between people, objects, places, concepts or events within that system. An ERD is a [data modeling](http://searchdatamanagement.techtarget.com/definition/data-modeling) technique that can help define business processes and can be used as the foundation for a [relational database](http://searchsqlserver.techtarget.com/definition/relational-database).

**The elements of an ERD are:**

* Entities
* Relationships
* Attributes

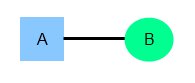
**Uses:**

* Entities are the "things" for which we want to store information. It is a piece of data-an object or concept about which data is stored. An entity is a person, place, thing or event.
* A relationship is how the data is shared between entities. There are three types of relationships between entities:

**1.One-to-One:**

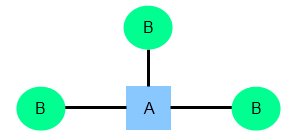
A one-to-one relationship is a type of cardinality that refers to the relationship between two entities (see also entity–relationship model) A and B in which one element of A may only be linked to one element of B.

For example, in a database of employees, each employee name (A) is associated with only one social security number (B).



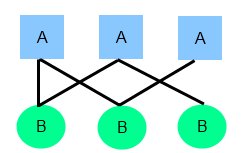
**2. One-to-Many:**

One instance of an entity (A) is associated with zero, one or many instances of another entity (B), but for one instance of entity B there is only one instance of entity A. For example, for a company with all employees working in one building, the building name (A) is associated with many different employees (B), but those employees all share the same singular association with entity A.



**3. Many-to-Many:**

Many-to-Many relationship is a type of cardinality that refers to the relationship between two entities A and B in which A may contain a parent instance for which there are many children in B and vice versa. For example, for a company in which all of its employees work on multiple projects, each instance of an employee (A) is associated with many instances of a project (B), and at the same time, each instance of a project (B) has multiple employees (A) associated with it.



**Attributes** are the data we want to collect for an entity.

There are different types of attribute:

1. **Optional attributes:**

A required attribute is an attribute that must have a value in it, while an optional attribute may not have a value in it and can be left blank. The reasoning for making an attribute required is to ensure that data are collected for that specific characteristic.

**2.** **Keys and non-keys Attributes:**

In every entity an attribute or grouped attributes uniquely identify that entity. These attributes are the key attributes and range from Primary key (single attribute identifier) to a Composite Key (Multi attribute Identifier). The rest of the attributes after the identifier are considered the non-key attributes or descriptors, which just describe the entity.

1. **Simple and Composite Attributes:**

Attributes can be classified as having many parts to them or a single unbreakable attribute. The composite attribute is an attribute that can be subdivided into other single attributes with meanings of their own. A simple attribute is just an attribute that cannot be subdivided into parts.

1. **Single-valued and multi-valued Attributes:**

Attributes can be classified as single or multivalued. The single-value attribute can only have one value, while the multi-valued attributes usually can store multiple data in them.

1. **Derived Attributes:**

The last category that attributes can be defined is called a derived attribute, where one attribute is calculated from another attribute. The derived attribute may not be stored in the database but rather calculated using algorithm.

**ENTITY SET NAMES:**

1. Courses

2. Teacher

3. Materials

4. Students

5. Quizes

**ATTRIBUTES OF EACH ENTITY:**

|  |  |
| --- | --- |
| **COURSES** | |
| Course no  Course\_name  Teacher\_id  Student\_id | Varchar (5000)  Varchar (5000)  Varchar (5000)  Varchar (5000) |

|  |  |
| --- | --- |
| **TEACHER** | |
| Teacher id  Teacher\_name  Teacher\_mail  Teacher\_cell\_ no | Varchar (5000)  Varchar (5000)  Varchar (5000)  Varchar (5000) |

|  |  |
| --- | --- |
| **Materials** | |
| Material\_name  Course\_no  Student\_id  Teacher\_id | Varchar (5000)  Varchar (5000)  Varchar (5000)  Varchar (5000) |

|  |  |
| --- | --- |
| **STUDENTS** | |
| Student\_id  Student\_name  Student\_mail  Student\_cell\_no | Varchar (5000)  Varchar (5000)  Varchar (5000)  Varchar (5000) |

|  |  |
| --- | --- |
| **QUIZES** | |
| Quiz\_no  Course\_no  Marks  Student\_id | Varchar (5000)  Varchar (5000)  Varchar (5000)  Varchar (5000) |

**Entity Relationship Diagram:**

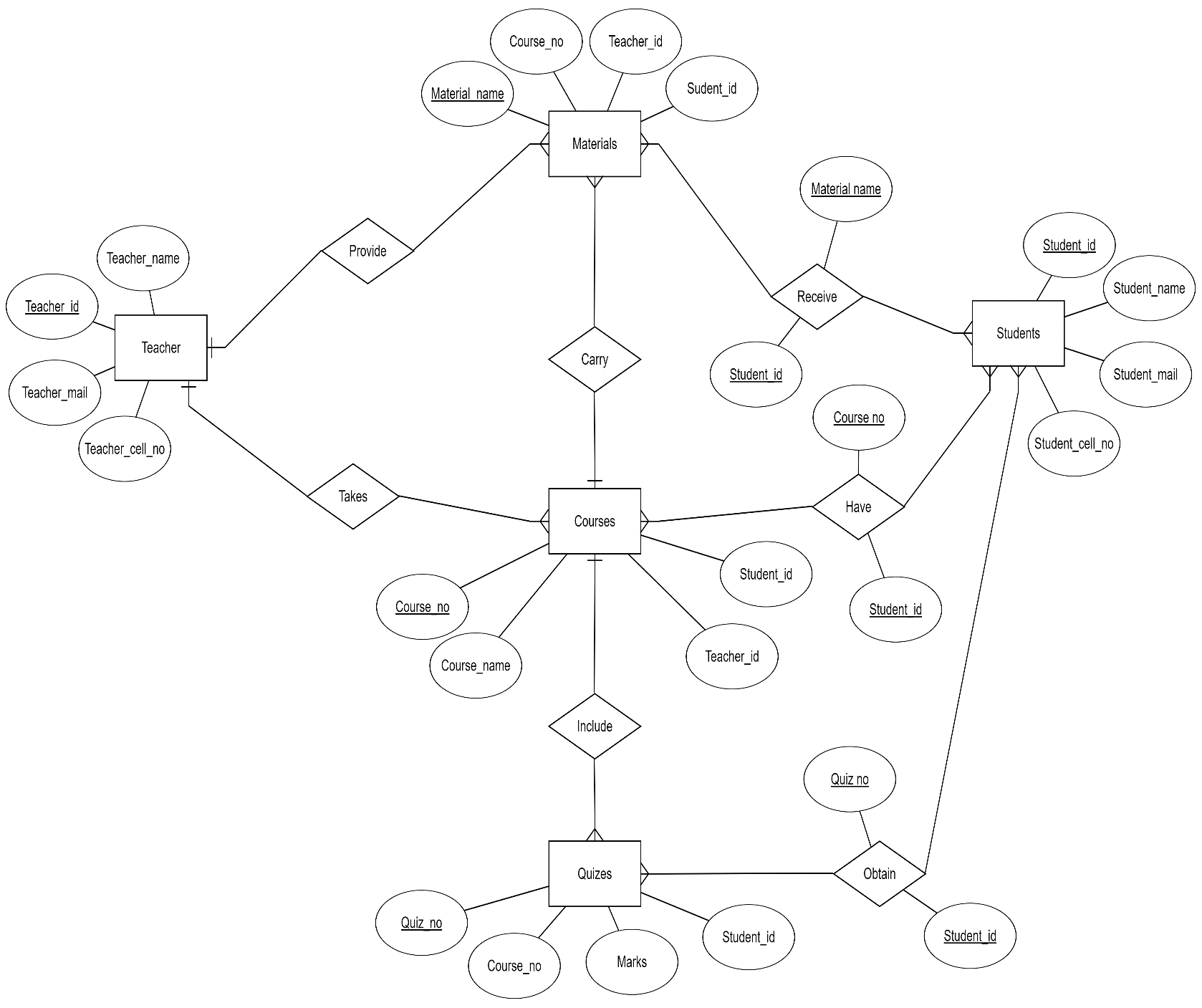
****

Fig : ERD Diagram

**Testing & Debugging:**

For testing our application we used fiddler web debugger. In there we found certain amount of problems on posting info and exception handling problem. We successfully overcome those problems. Now our web application is more or less a complete one without extreme bugs that can cause problem to user.

**Limitation:**

* We couldn’t add the individual student user part.
* Teacher is unable to see who are seeing his course materials.
* We couldn’t add password system for attending course.

**Future Plan:**

* Individual profile for students.
* Real-time problem solution by leaving comments and reply.
* Sending results of students in their profile dynamically without showing it publicly.

**Contribution:**

* Mahir or Rashid – 25%
* Tasnim Mashrur – 25%
* Atiqul Islam Chowdhury – 25%
* Md. Saiful Islam– 25%

**Conclusion:**

At the end we can say it's all about solving the problems of communication between teachers and students. With its dynamic process hope we can deport sharing problems. So all the sides can get complication free information gathering way. Though it has many shortcomings but in future we will try to improve it.