**Table of contents**

1. Data Flow Diagram. -----------------------------------------------------------------------------------1

1.1 Why Data Flow Diagram is Used? --------------------------------------------------------1

2. Components of a DFD. --------------------------------------------------------------------------------2

2.1. External Entities. ----------------------------------------------------------------------------2

2.2. Process. --------------------------------------------------------------------------------------2

2.3. Data Flow. -----------------------------------------------------------------------------------2

2.4. Data Store. -----------------------------------------------------------------------------------3

3.Rules of Data Flow Diagram (DFD). ---------------------------------------------------------------3

4.Levels of Data Flow Diagram (DFD). --------------------------------------------------------------4

4.1. Context Level (Level 0). -------------------------------------------------------------------4

4.2. Level 1. ---------------------------------------------------------------------------------------5

4.3. Level 2. ---------------------------------------------------------------------------------------6

5.Logical Data Flow Diagram**.** ------------------------------------------------------------------------10

5.1.Benefits of using Logical Data Flow Diagram (DFD). -------------------------------10

5.1.1. Clarity and Visualization. -----------------------------------------------------10

5.1.2. Abstraction of Complexity. ---------------------------------------------------10

5.1.3. Facilitates Requirements Analysis. ------------------------------------------10

5.1.4. Identification of Redundancies and Inefficiencies. ------------------------10

6.Physical Data Flow Diagram (DFD). --------------------------------------------------------------11

6.1.Benefits of using Physical Data Flow Diagram (DFD). ------------------------------11

6.1.1 Detailed Understanding. -------------------------------------------------------11

6.1.2 Facilitates System Implementation. ------------------------------------------11

6.1.3 Supports System Configuration. ----------------------------------------------11

6.1.4 Aids in System Maintenance. -------------------------------------------------11

6.1.5 Enhances Communication. ----------------------------------------------------11

7.Different between logical and physical Data Flow Diagram (DFD). --------------------------12

**List of figures**

Fig.1 Symbol of External Entities. ---------------------------------------------------------------------2

Fig.2 Symbol of Process. --------------------------------------------------------------------------------2

Fig.3 Symbol of Data Flow. -----------------------------------------------------------------------------2

Fig.4 Symbol of Data Store. -----------------------------------------------------------------------------3

Fig.5 Rules of Data Flow Diagram. --------------------------------------------------------------------3

Fig.6 Context Level (level 0) Data Flow Diagram. --------------------------------------------------4

Fig.7 level 1 Data Flow Diagram. ----------------------------------------------------------------------5

Fig.9. level 2 DFD for Student Enrollment process. -------------------------------------------------6

Fig.10. level 2 DFD for Course Management process. ----------------------------------------------7

Fig.11. level 2 DFD for Manage Attendance process. -----------------------------------------------8

Fig.12. level 2 DFD for Exam process. ----------------------------------------------------------------8

Fig.13. level 2 DFD for Grade Recording process. --------------------------------------------------9

1. **Components of a DFD:**
   1. **External Entity:**

External entities in a Data Flow Diagram (DFD) are the sources or destinations of data that interact with the system being modeled. These entities exist outside the system boundary and provide inputs to the system or receive outputs from it. They are represented by rectangles and represent real-world actors such as people, organizations, or other systems.

Fig.1. Symbol of External Entities.

**2.2 Process:**

Processes are the core components of a Data Flow Diagram (DFD) that represent the actions or functions where data is processed or transformed within the system. They are illustrated as circles or rounded rectangles, each labeled with a unique identifier and a descriptive name that conveys the specific task performed by the process. Each process must have at least one input and one output, indicating the data it receives and the data it produces.

Fig.2 Symbol of Process.

**2.3 Data Flow:**

Data flows in a Data Flow Diagram (DFD) represent the movement of data between processes, data stores, and external entities. Depicted by arrows, data flows show the direction in which data travels, along with the nature of the data being transferred. The arrows are labeled with the data they carry, providing clarity on the specific information being transmitted.

Fig.3 Symbol of Data Flow.

* 1. **Data Stores**

Data stores in a Data Flow Diagram (DFD) are locations where data is stored within the system for later retrieval and use. They are represented by open-ended rectangles or parallel lines, indicating that data is being held rather than processed or transferred. Examples of data stores include databases, files, logs, or registers that store information such as user accounts, transaction records, or inventory data.

Fig.4 Symbol of Data Store.

1. **Rules of Data Flow Diagram (DFD)**
2. An entity cannot provide data to another entity without some processing occurred.
3. Data cannot move directly from an entity to a data store without being processed.
4. Data cannot move directly from data store to entity without being processed.
5. Data cannot move directly from one data store to another without being processed.

|  |  |  |
| --- | --- | --- |
| **Wrong** | **Right** | **Description** |
|  |  | An entity cannot provide data to another entity without some processing occurred. |
|  |  | Data cannot move directly from an entity to a data store without being processed. |
|  |  | Data cannot move directly from data store to entity store without being processed. |
|  |  | Data cannot move directly from one data store to another without being processed. |

Fig.5 Rules of Data Flow Diagram

1. **Levels of Data Flow Diagram (DFD).**

Data Flow Diagrams (DFDs) are structured into different levels to represent a system's data flow with varying degrees of detail. Each level provides a deeper and more detailed view of the system. The primary levels of DFDs are:

1. Context Level Diagram or Level 0 Diagram
2. Level 1 Data Flow Diagram
3. Level 2 Data Flow Diagram
   1. **Context Level Diagram (Level 0).**

The Context Level Data Flow Diagram (DFD), also known as the Level 0 DFD, serves as the highest-level abstraction of a system, providing an overview of its interactions with external entities. In this diagram, the system is depicted as a single process, encapsulating its entirety, while external entities are represented by rectangles, showcasing their inputs and outputs to the system.

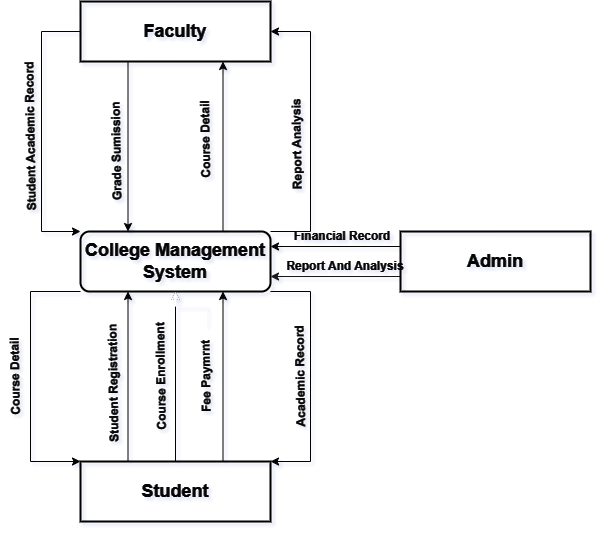


Fig.6 Context level DFD of College management system

This context level data flow diagram for the College Management System illustrates the interactions between faculty, students, and admin with the central system. Faculty members input student academic records, grades, attendance, and course details into the system, and in return, they receive analytical reports. Students engage with the system by registering, enrolling in courses, and making fee payments, while also accessing their academic records from the system. Admin staff input financial records and receive comprehensive financial reports and analyses. The central College Management System processes all these inputs and outputs, ensuring smooth and efficient data flow and communication among faculty, students, and admin, thereby facilitating effective management of academic and administrative tasks.

* 1. **Level 1 Data Flow Diagram (Level 1).**

The Level 1 Data Flow Diagram (DFD) provides a more detailed view of the system by breaking down the single process depicted in the context level diagram into major subprocesses. In this level of abstraction, each subprocess represents a significant functional area or module within the system, illustrating how data flows between them.

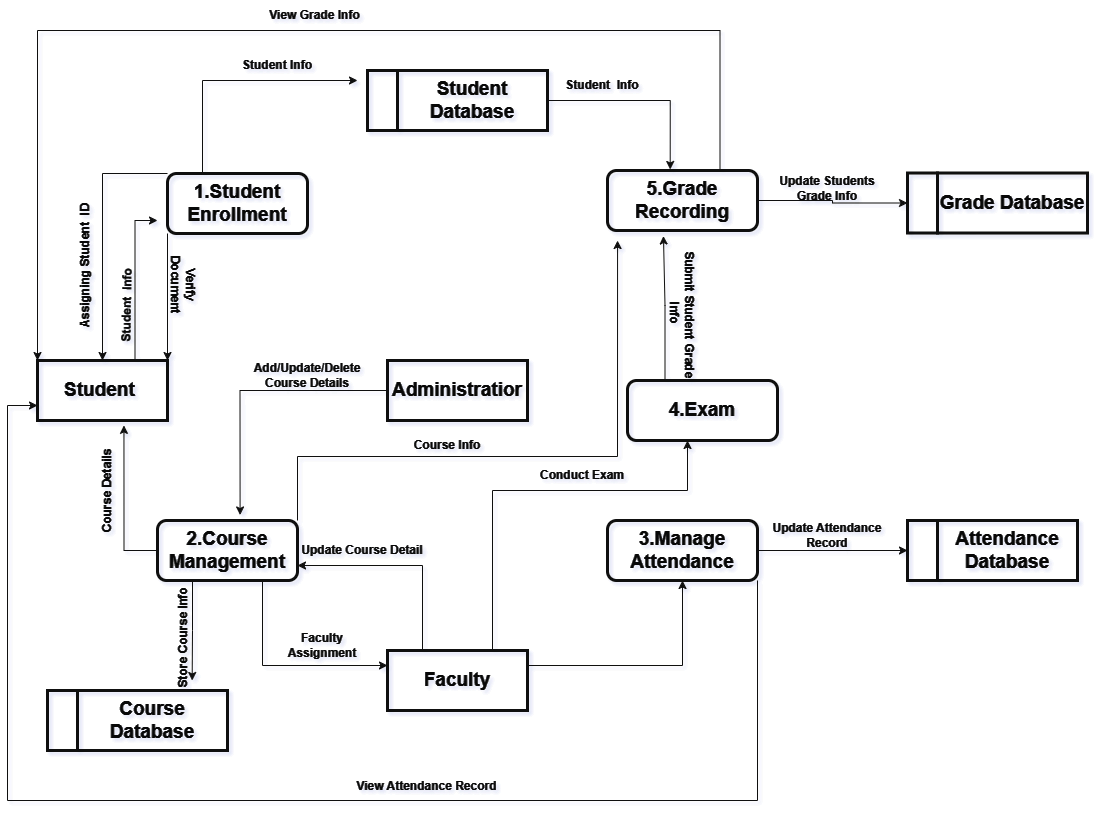


Fig.7 Level 1 DFD for college management system.

The Level 1 Data Flow Diagram for the College Management System outlines key interactions between students, faculty, and administrators with subsystems and databases. It starts with "1. Student Enrollment," where student information is entered into the Student Database. Administrators handle "2. Course Management," updating course details in the Course Database for access by students and faculty. Faculty manage course details, attendance, and grades.

The "3. Manage Attendance" process records and updates student attendance in the Attendance Database, viewable by faculty and administrators. "4. Exam" covers conducting exams and recording results in the Student Database. Lastly, "5. Grade Recording" updates student grades in the Grade Database. This system ensures efficient operations and accurate record-keeping.

* 1. **Level 2 Data Flow Diagram (Level 2).**

The Level 2 Data Flow Diagram (DFD) further decomposes the processes identified in the Level 1 DFD into more detailed subprocesses, providing a comprehensive view of the system's functionality. Each major process from the Level 1 DFD is broken down into smaller, more specific tasks or operations, illustrating the detailed flow of data and interactions within the system.

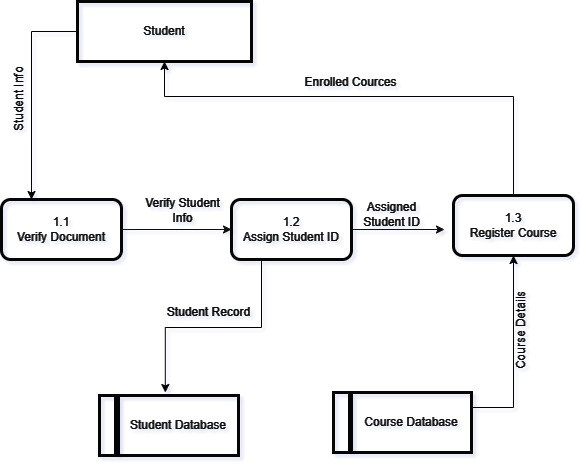
****

Fig.9. Level 2 DFD for Student Enrollment process.

The Level 2 Data Flow Diagram for the student enrollment process in the College Management System outlines the specific steps of enrollment. It starts with students submitting their information and documents, followed by the "1.0 Verify Document" process to verify these details. Once verified, "1.2 Assign Student ID" assigns a unique ID to the student, and this information is stored in the Student Database.

Next, in "1.3 Register Course," the student uses their ID to register for courses, updating the Course Database with the enrollment details. This ensures accurate recording and seamless course enrollment, maintaining up-to-date records of all students and their course selections.

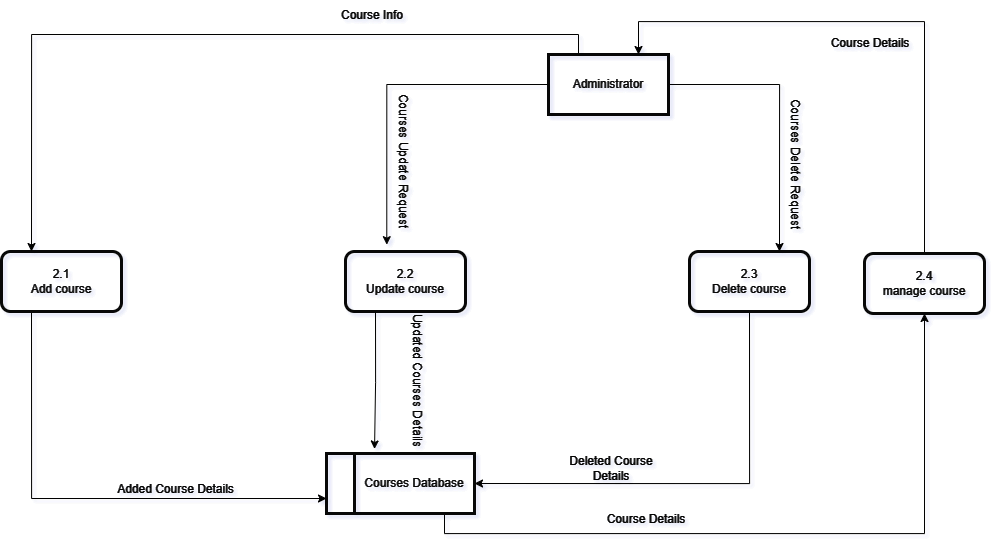


Fig.10. Level 2 DFD for Course management process.

The Level 2 Data Flow Diagram (DFD) for managing courses in a college management system includes several subprocesses. It begins with "Add Course" (2.1), where new course information is entered into the Courses Database. The "Update Course" (2.2) subprocess allows administrators to modify existing course details. The "Delete Course" (2.3) subprocess handles the removal of courses, updating the Courses Database accordingly. "Manage Course" (2.4) provides a comprehensive interface for course management, enabling administrators to fetch or update course details. These subprocesses ensure the Courses Database remains accurate and up to date through administrator interactions.

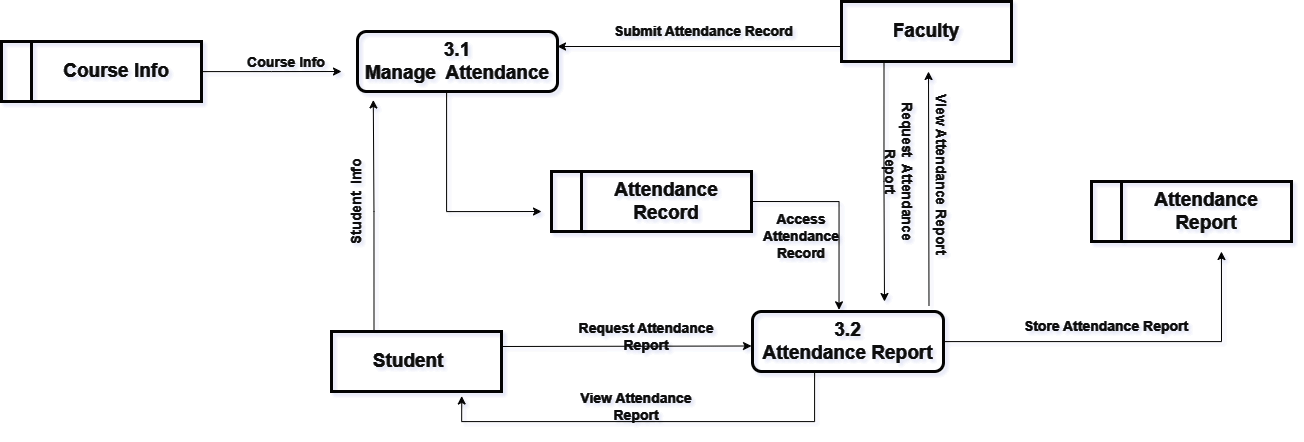


Fig.11. Level 2 DFD for Manage Attendance process.

The Level 2 Data Flow Diagram (DFD) for managing attendance in a college management system includes two key subprocesses. First, "Manage Attendance" (3.1) uses course and student information to handle attendance records, which faculty members submit and store in the Attendance Record database. Students can also view their attendance through this subprocess. Second, "Attendance Report" (3.2) generates and stores reports based on the attendance records, allowing faculty to access up-to-date attendance information. This system ensures accurate maintenance and accessibility of attendance data for both students and faculty.

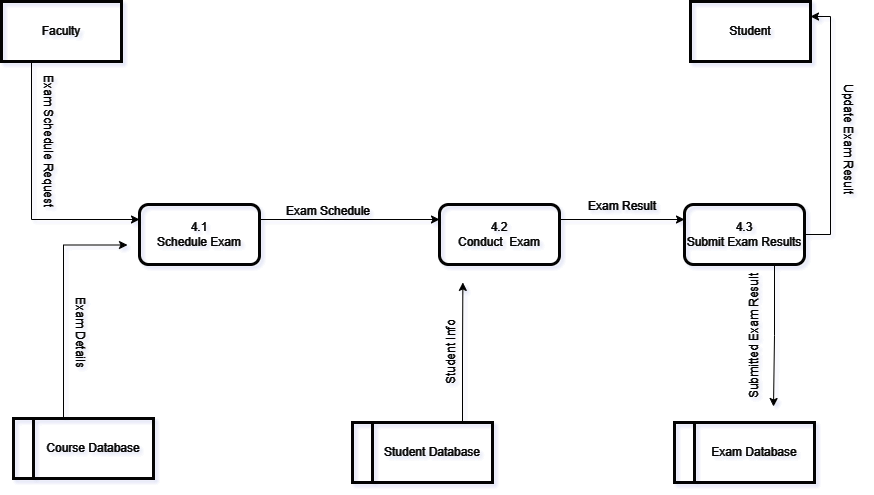


Fig.12. Level 2 DFD for Exam process.

This Level 2 Data Flow Diagram (DFD) illustrates the subprocesses for managing exams in a college management system. It begins with the faculty scheduling exams by inputting exam details, which are verified against the course database. The "Schedule Exam" process then generates an exam schedule. The "Conduct Exam" process uses this schedule to manage the actual exam, ensuring that eligible students from the student database participate. Finally, the "Submit Exam Results" process collects and records the exam results, updating the exam database and communicating the results to both students and faculty. This ensures a streamlined and organized exam management system.

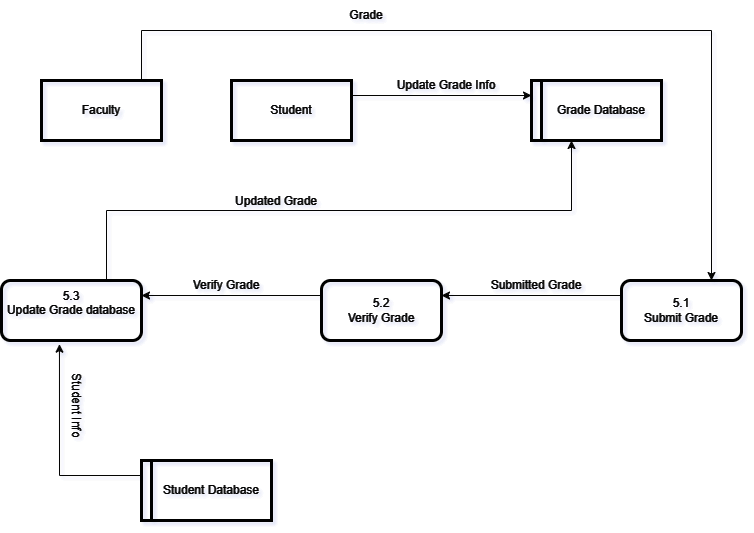


Fig.13. Level 2 DFD for Grade Recording process.

This Level 2 Data Flow Diagram (DFD) illustrates the subprocesses involved in recording grades within a college management system. The process starts with the faculty entering grades, which are then updated in the Grade Database through the "Update Grade Database" subprocess (S3). The updated grades are subsequently verified in the "Verify Grade" subprocess (S2). Once the grades are verified, they are submitted in the "Submit Grade" subprocess (S1), updating the Grade Database accordingly. The Student Database interacts with the "Update Grade Database" subprocess, providing student information needed for updating grades. The entire flow ensures grades are accurately recorded, verified, and stored in the system.

1. **Logical Data Flow Diagram.**

A Logical Data Flow Diagram (DFD) graphically represents the flow of data within a system, focusing on the logical relationships between processes, data stores, and external entities. Unlike a Physical DFD, which details hardware and software implementation, a Logical DFD emphasizes the system's functional aspects, ignoring specific implementation details. It helps stakeholders visualize data movement and interactions within the system, using standardized symbols to represent processes, data flows, data stores, and external entities. This facilitates communication and analysis during system design and development.

* 1. **Benefits of using Logical Data Flow Diagram.**

Logical Data Flow Diagrams (DFDs) offer several significant benefits that enhance the understanding, analysis, and design of systems. Some of them are explain below: -

* + 1. **Clarity and Visualization:** Logical DFDs clearly show data flow within a system, aiding stakeholder understanding and enhancing communication among developers, analysts, and users.
    2. **Abstraction of Complexity:** They focus on essential functions, reducing complexity and allowing stakeholders to concentrate on data flow logic without technical details.
    3. **Facilitates Requirements Analysis:** Logical DFDs help identify data sources, processing needs, and destinations, aiding in requirement elicitation and validation.
    4. **Identification of Redundancies and Inefficiencies:** By visualizing data flow, Logical DFDs reveal redundancies and inefficiencies, highlighting areas for optimization.
    5. **Supports System Design and Architecture:** They define high-level structures of processes and data stores, providing a foundation for detailed system design.

1. **Physical Data Flow Diagram (DFD).**

A Physical DFD extends a Logical DFD by adding details about hardware, software, networks, and data storage. It shows how data flows through a system's physical components, such as servers, workstations, web servers, databases, and networks. For example, in an online bookstore system, a Physical DFD would depict servers, internet connections, and cloud storage. This detailed view aids in system implementation, configuration, and maintenance, offering insights crucial for effective deployment and management.

* 1. **. Benefits of using Physical Data Flow Diagram (DFD).**

Logical Data Flow Diagrams (DFDs) offer several significant benefits that enhance the understanding, analysis, and design of systems. Some of them are explained below:-

* + 1. **Detailed Understanding:** Physical DFDs provide a comprehensive view of data flow through the system's physical components, aiding stakeholders in understanding the implementation details.
    2. **Facilitates System Implementation:** They serve as blueprints for system deployment, detailing the required hardware and software components and their interactions, ensuring accurate setup and integration.
    3. **Supports System Configuration:** These diagrams assist in configuring networks, servers, software installations, and databases, ensuring the system meets performance, security, and reliability standards.
    4. **Aids in System Maintenance:** Physical DFDs are valuable for system maintenance tasks like troubleshooting, upgrades, and repairs, offering a visual guide for diagnosing and addressing issues.
    5. **Enhances Communication:** They improve communication among developers, system administrators, and other stakeholders by providing a common visual language for discussing the system's physical setup.