

Overall Software Plan

CS 346: Assignment-2A

Academic Section Management Software

Group 10A : 210101113-210101119

1 Problem Statement and Stakeholders

(Yashraj)

2 Software Model

The adoption of the **Structured Analysis and Design** (SAD) model for the development of this specific software is justified by the intricate design and the comprehensive requirements that demand careful consideration. This report documents the different components of this model, including the *Data Dictionary* (encompassing Requirements and Assumptions), *Data Flow Diagrams* (DFDs), and *Entity-Relationship* (E-R) Diagrams in the following sections below.

3 Software Requirements and Assumptions

The following are the specifications of the software that has to be developed that has been mentioned in the problem statement: (Agrawal)

- Item1
- Item2

4 Data Flow Diagrams

Short Desc (Yashraj) **Bold**, Verb, *Italics*

4.1 Level-0 DFD

(Yashraj)

4.2 Level-1 DFD

(Yashraj)

4.3 Level-2 DFDs

The following subsections highlight the detailed **Level-2** Dataflow diagrams of various processes in the Academic system Management software:

4.3.1 Admission Management Module

(Gupta)

4.3.2 Course Management Module

(Tanish)

4.3.3 Examination Management Module

(Pratyush)

4.3.4 Grade Management Module

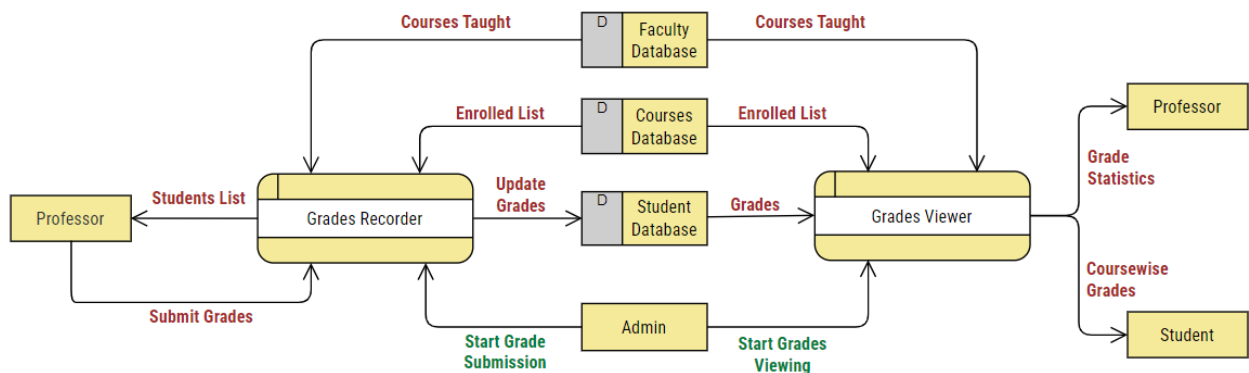


Figure 1: Level-2 DFD of Grade Management Module

- The **Grade Recorder** Process first gets the signal from the Admin side to start grade submission.
- The professors are displayed the various courses that they teach and they can upload the grade list of students enrolled coursewise, which are updated duly by the Grade Recorder process into **Grades** field of each course taken by student in **Student Database**.
- The **Grades Viewer** process first gets the signal from the Admin side after which the students are displayed the grades that they got in specific courses.
- The professors can get overall statistics about the grades like Class Average, number of students who got different grades like AA, AB, BB etc.
- This particular module interacts with the databases to fetch and update the results, as shown in the DFD diagram of Grade Management module.

5 Entity Relationship Diagram

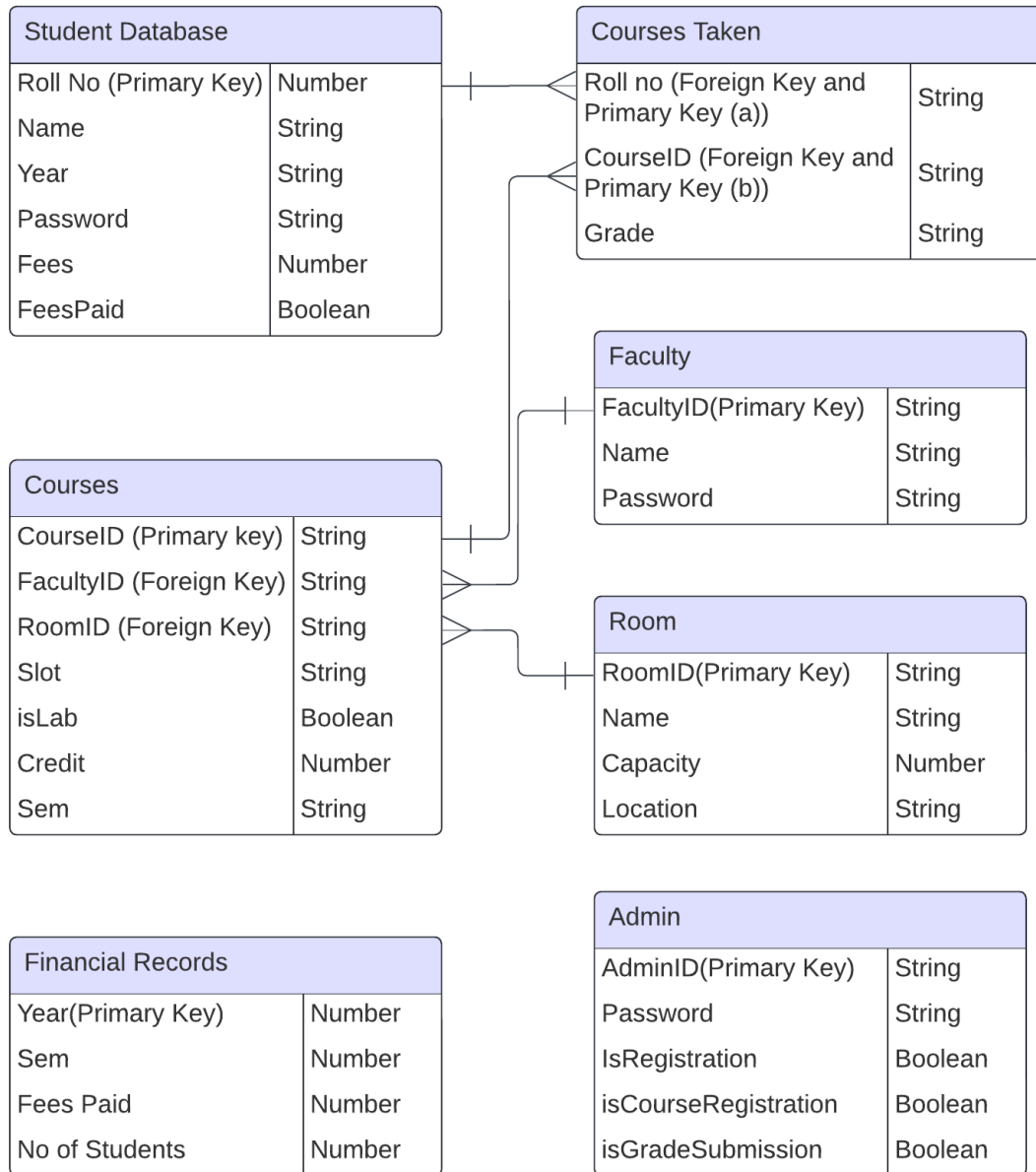


Figure 2: **ER-Diagram**, Each Arrow represents a relation. A multiple-forked arrow represents Many relationships, whereas a single bar arrow represents one-one relationship.

5.1 Student Database

This database is used to store all information regarding students. The description of Each key is as follows:

- **Roll No:** Roll Number of student serves as Primary Key
- **Name:** Name of Student
- **Year:** Year Of Joining
- **Password:** Password Used for authentication Purposes

- **Fees:** Amount to be Paid By student
- **FeesPaid:** Boolean to store if Fees paid or not

5.2 Courses Database

This database is used to store all information regarding Courses. The description of Each key is as follows:

- **CourseID:** Unique Identifier for Each Course. Serves as Primary key.
- **FacultyID:** ID of faculty taking this course serves as a Foreign Key to faculty Table.
- **RoomID:** ID of Room where this course will be held serves as a Foreign Key to Room Database.
- **Slot:** Slot in which this course lectures are held. Assigned by TimeTable generator.
- **isLab:** Boolean to store if It is a lab or not.
- **Credit:** Credits of this course.
- **Sem:** String to store sems in which this course is offered.

5.3 Courses Taken Database

This database is used to store all information regarding which student took which Course. The pair (Roll No, CourseID) serves as a Primary Key. The description of Each key is as follows:

- **Roll No:** Roll No of student who has taken this course. Serves as Foreign Key to student Database.
- **CourseID:** ID of Course taken. Serves as a Foreign Key to Courses Table.
- **Grade:** Grade Received by student in that particular Course

5.4 Faculty Database

This database is used to store all information regarding Faculties. The description of Each key is as follows:

- **FacultyID:** Unique Identifier for Each Faculty. Serves as Primary key.
- **Name:** Name of Faculty
- **Password:** Password Used for authentication Purposes

5.5 Room Database

This database is used to store all information regarding Rooms. The description of Each key is as follows:

- **RoomID:** Unique Identifier for Each Room. Serves as Primary key.
- **Name:** Name of Room
- **Capacity:** Capacity of Room
- **Location:** Location of Room

5.6 Admin Database

This database is used to store all information regarding Admin who controls the system. The description of Each key is as follows:

- **AdminID:** Unique Identifier for Admins. Serves as Primary key.
- **Password:** Password Used for authentication Purposes
- **isRegistration:** Boolean to start admission and Registration of new students.
- **isCourseRegistration:** Boolean to start course Registration.
- **isGradeSubmission:** Boolean to start Grade Submission.

5.7 Financial Records

This database is used to store all information related to Finances. The description of Each key is as follows:

- **Year:** Year of Financial Record. Serves as Primary key.
- **Sem:** Semester
- **FeesPaid:** How much Fees is paid.
- **No of Students:** Number of Registered Students.

6 Feature Enhancement

(Shivam)

- hrtrt
- rtrtrtrtrtrt

7 User Interface

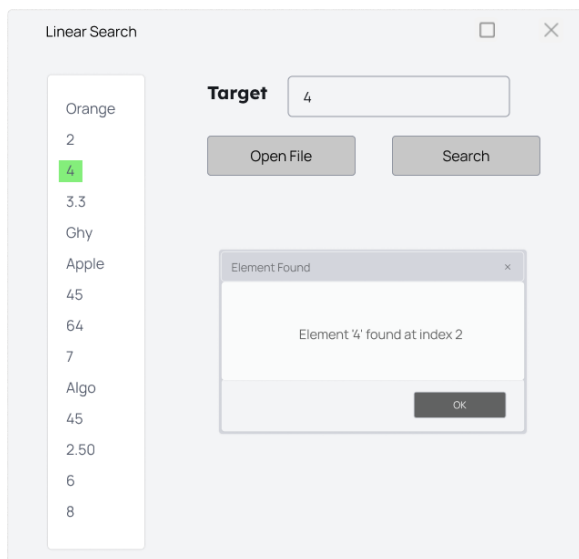
(Shivam Agrawal) The user interface for this software can be developed using **Visual Basic**. The following are the tentative list of toolbox components needed for the basic implementation (without any feature enhancement):

- **Text Box:** sdfeererererer
- And so on....

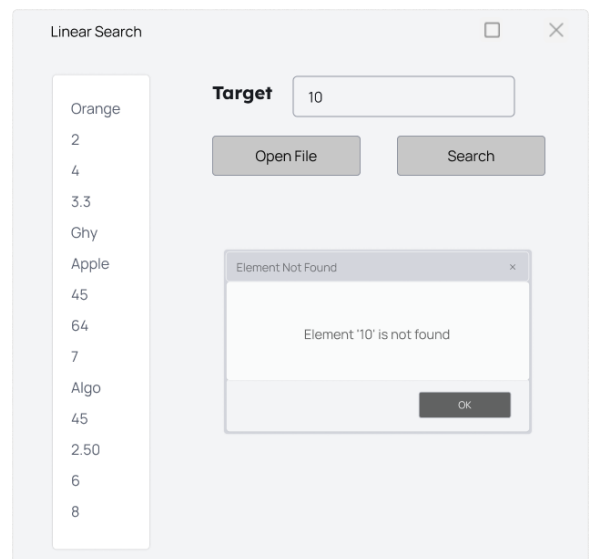
7.1 Basic Design

This is the tentative basic design of the user interface which is to be designed in **Visual Basic** using the components listed above:

7.2 Subsections



(a) Message box output when element is found.



(b) Message box output when element is not found.

Figure 3: Basic design of the software application