Project Proposal
On
Student Grading System

Guided By:

Mr. Anuj Kumar

Created By:

Shashi Maddheshiya

AFId: AF04991143

Shifa Parveen

AFId:AF04991258

Batch Code:ANP-D2406

Course Code:ITPR

Table of Contents

1. Title of the Project
2. Introduction
3. Objective
4. Project Category
5. Analysis
   - Database Design
   - Modules and Description
   - ER Diagram
   - Data Flow Diagram
6. Complete Structure
   - Process Logical Diagram
7. Platform Used
   - Hardware Requirement
   - Software Requirement
8. Future Scope
9. Bibliography

Title of the Project:

Student Grading System

Introduction to Student Grading System:

- Student Grading System is a faster and accurate system that stores student data and marks
- It calculates grades automatically.
- Helps teachers save time and reduce errors.

Objective:

- It reduce manual work.
- Avoid calculation errors.
- Store all student and marks data in one place.
- Auto-calculate total, percentage, and grade.
- Provide a simple and secure system.

Project Category:

- Application-Based Project using Java, JDBC, and MySQL

Analysis:

Modules and Description

1. Login Management

    Handles secure login

    Authenticates admin/teacher users

2. Student Management

    Add, update, delete, view student details

3. Subject Management

    Manage subject code and subject name

4. Marks Management

    Enter and update marks for each student

5. Grade Calculation

    Automatically calculates total, percentage, and gra

6. Result Module

    Generates student's final result (pass/fail)

DATABASE DESIGN – Student Grading System

Table 1: Student

Fields	Datatype	Properties
student_id	Varchar(30)	Primary key
student_name	Varchar(200)	Not null
email	Varchar(200)	Not null
course	Varchar(100)	Not null
dob	date	Not null
created_at	date	Not null

Table 2: Marks

Fields	Datatype	Properties
student_id	Varchar(30)	Not null, foreign key
subject	Varchar(100)	Not null
exam_type	Varchar(50)	Not null
marks	int	Not null
Max_marks	int	Not null
created_at	date	Not null

Relation:

one student has many marks.

one mark belongs to one student.

Table 3: Grade

Fields	Datatype	Properties
student_id	Varchar(30)	Not null, foreign key
total_marks	int	Not null
percentage	float	Not null
grade	Varchar(5)	Not null
created_at	date	Not null

Relation:

One student has one grade only

Grade is calculated from marks

Table 4: Result

Fields	Datatype	Properties
student_id	Varchar(30)	Not null, foreign key
total_marks	int	Not null
percentage	float	Not null
grade	Varchar(5)	Not null
status	Varchar(10)	Not null

Relation:

One student has one result only.
Result is calculted from marks.

Table 5: User_Login

Fields	Datatype	Properties
user_id	Varchar(30)	Primary key, not null
username	Varchar(200)	Not null
password	Varchar(200)	Not null
role	Varchar(50)	Not null

Relation:

Login table only manages system users
Student are not stored here

Table 6: Subjects

Fields	Datatype	Properties
subject_code	Varchar(30)	Primary key, not null
subject_name	Varchar(200)	Not null
max_marks	int	Not null

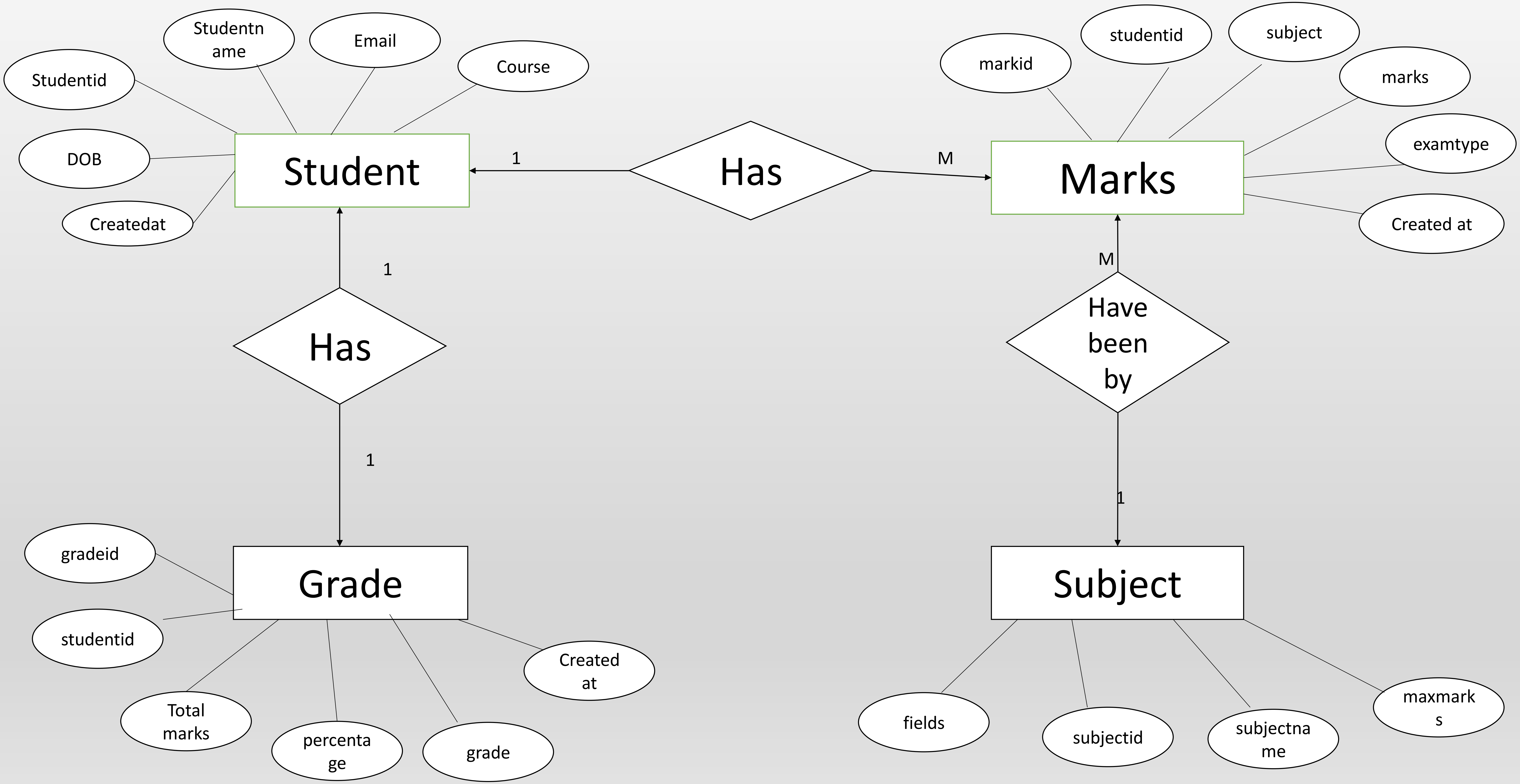
Relation:

One subject can have many student marks.

Final relationship summary:

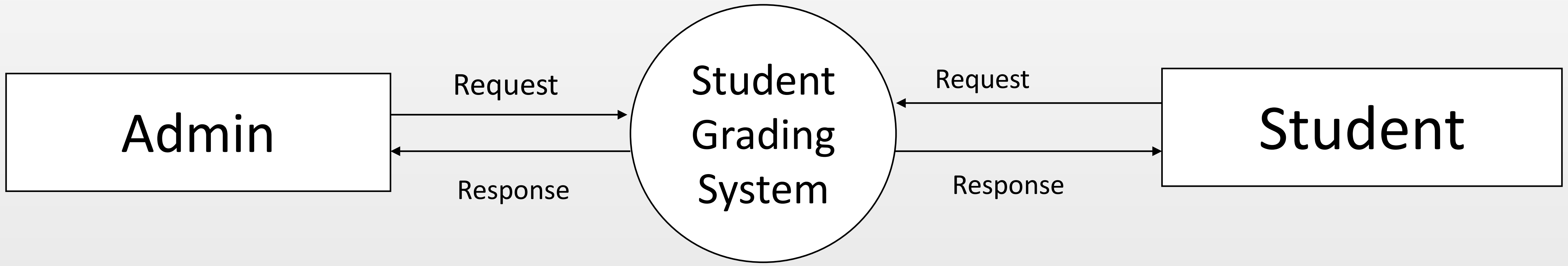
One student → many marks
One student → one grade
One grade → belongs to one student
One subject → many marks
Login table independent

ER Diagram:

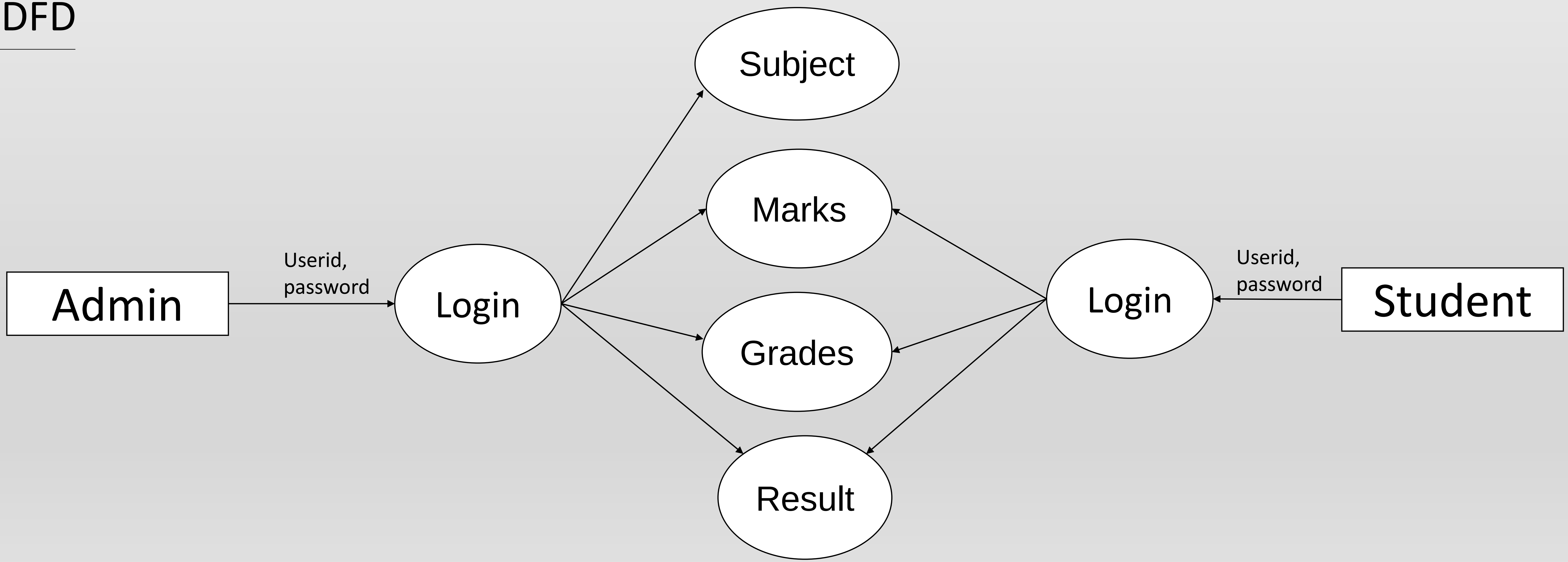

Studentname
Email
Course
Studentid
DOB
Createdat
Student
1
Has
M
Marks
markid
studentid
subject
marks
examtype
Created at
1
Has
1
M
Have been by
1
Grade
gradeid
studentid
Total marks
percentage
grade
Created at
Subject
fields
subjectid
subjectname
maxmarks

Data Flow Diagram

Zero Level DFD



First Level DFD



Complete Structure:

The system structure includes:

- User login
- Student & subject management
- Marks entry
- Automatic grade calculation
- Result generation
- Database operations

Process Logical Diagram

Shows flow from:

Login → Students → Subjects → Marks → Grade → Result

Platform Used:

Hardware Requirements

- Minimum 4GB RAM
- i3 Processor
- 500MB free disk space

Software Requirements

- java JDK
- MySQL
- JDBC Connector
- Eclipse or IntelliJ
- Maven

Future Scope:

- Can be upgraded to web or mobile application
- Can include attendance and fees module
- Can generate detailed performance reports
- Can add multi-user roles (Admin, Teacher, HOD)

Bibliography :

- Java Documentation
- MySQL Documentation
- Online Tutorials (Oracle, W3Schools)
- Trainer Notes

Thank you