

Student

Management System

\*\*Manage students efficiently with an all-in-one system. \*\*

Table of Content

[Executive Summary 2](#_Toc18561)

[Design Analysis Process 2](#_Toc1356)

[1. BRAINSTORMING: 2](#_Toc22873)

[2. USER STORIES: 4](#_Toc3473)

[3. USES CASES: 5](#_Toc12475)

[4. USE CASE DIAGRAM: 6](#_Toc21752)

[5. FUNCTIONAL AND NON – FUNCTIONAL: 7](#_Toc6015)

[6. ACTORS 7](#_Toc30945)

[7. EVENTS 7](#_Toc30154)

[8. AGILE METHODOLOGY 7](#_Toc2071)

[Entity Relationship Diagram 8](#_Toc12679)

[Table Designs – Data Dictionary 10](#_Toc25054)

[Contributions 12](#_Toc8655)

[References 13](#_Toc10830)

# Executive Summary

The **University Student Management System** (USMDS) is a **user-friendly** platform that will not assist academic staff to store and organize student’s information easily in **one place** but also help students to manage their profiles along with high level of security. We understand that how challenging it is, to store the information of the thousands of students and update it frequently when it is required. Therefore, we have made this database to to improve the overall experience for the students, staff members and administrators by utilization of just one centralized system.

By implementing University Student Management System, numerous fruitful benefits will arise that can enhance the overall academic experience: -

1. This system will help students to access their personal data and academic planning in which they will be able to check their personal information, academic records, enrolment status, course tenure and how much they are paying for it in just one place.
2. This system will also help academic staff to manage course information, grades and enrolment details just by one centralized system. It will also improve the accuracy and consistency across the institute regarding every aspect of students by better allocation of the data in system.
3. This system will play a predominant role for the admin to manage the student information effectively and efficiently. By the utilization of this system, administrators can keep thousands of students’ information in one database and can update it frequently whenever it is required.

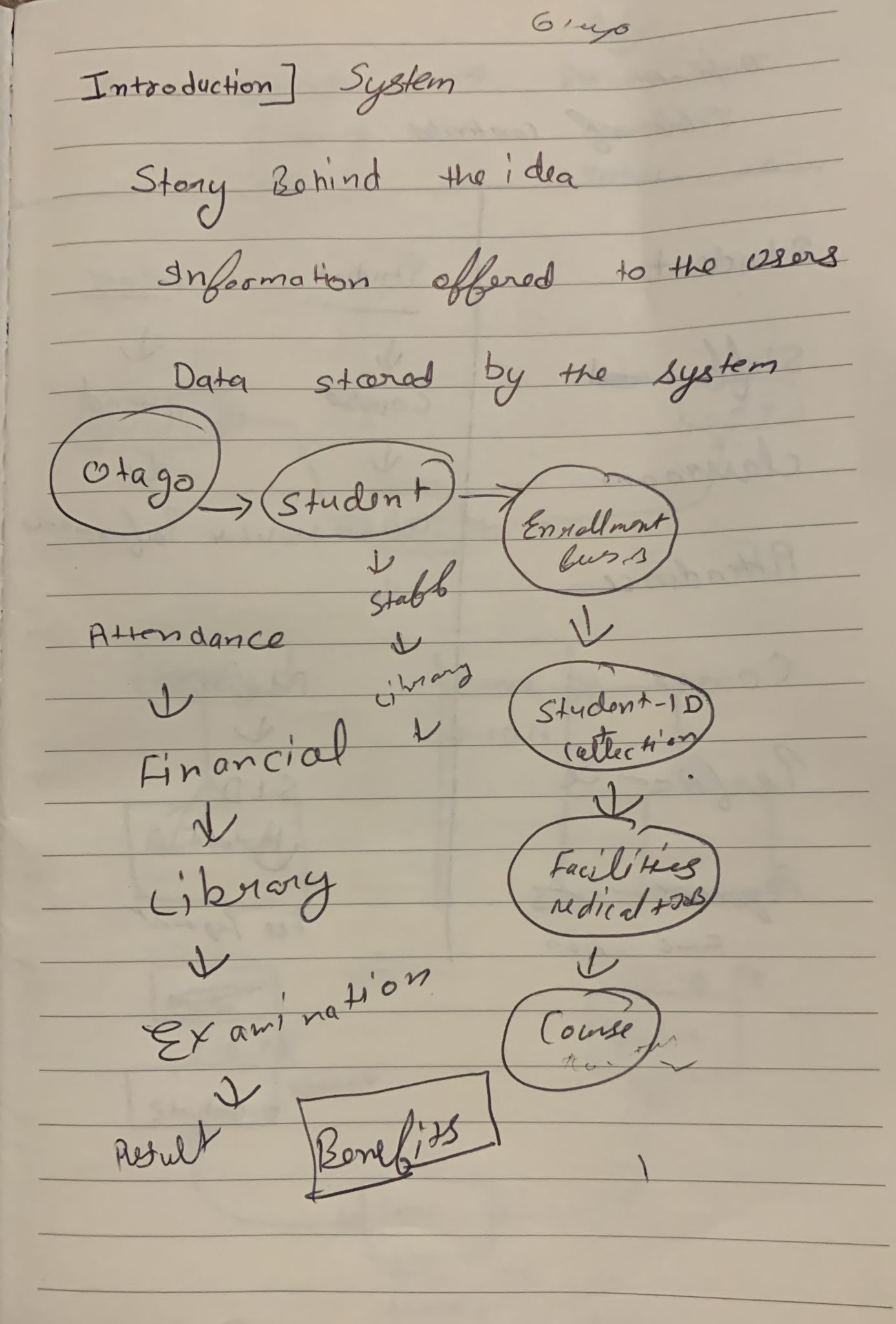
To put it in a nutshell, USMS is such a great platform that will make the overall process easier and smoother for the universities to manage the students effectively. And will also become handy for the learners to access all the vital resources throughout their study duration.

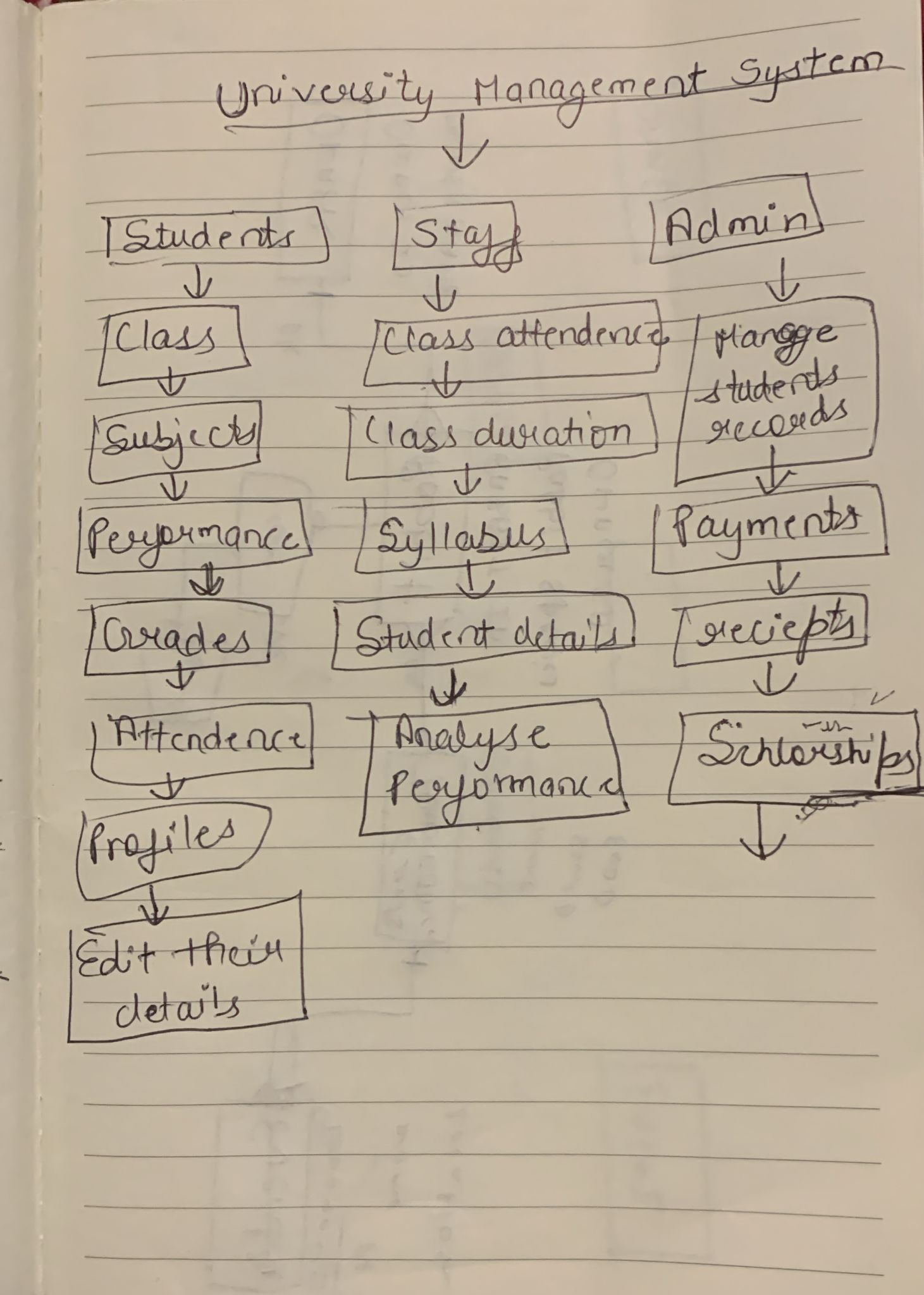
# Design Analysis Process

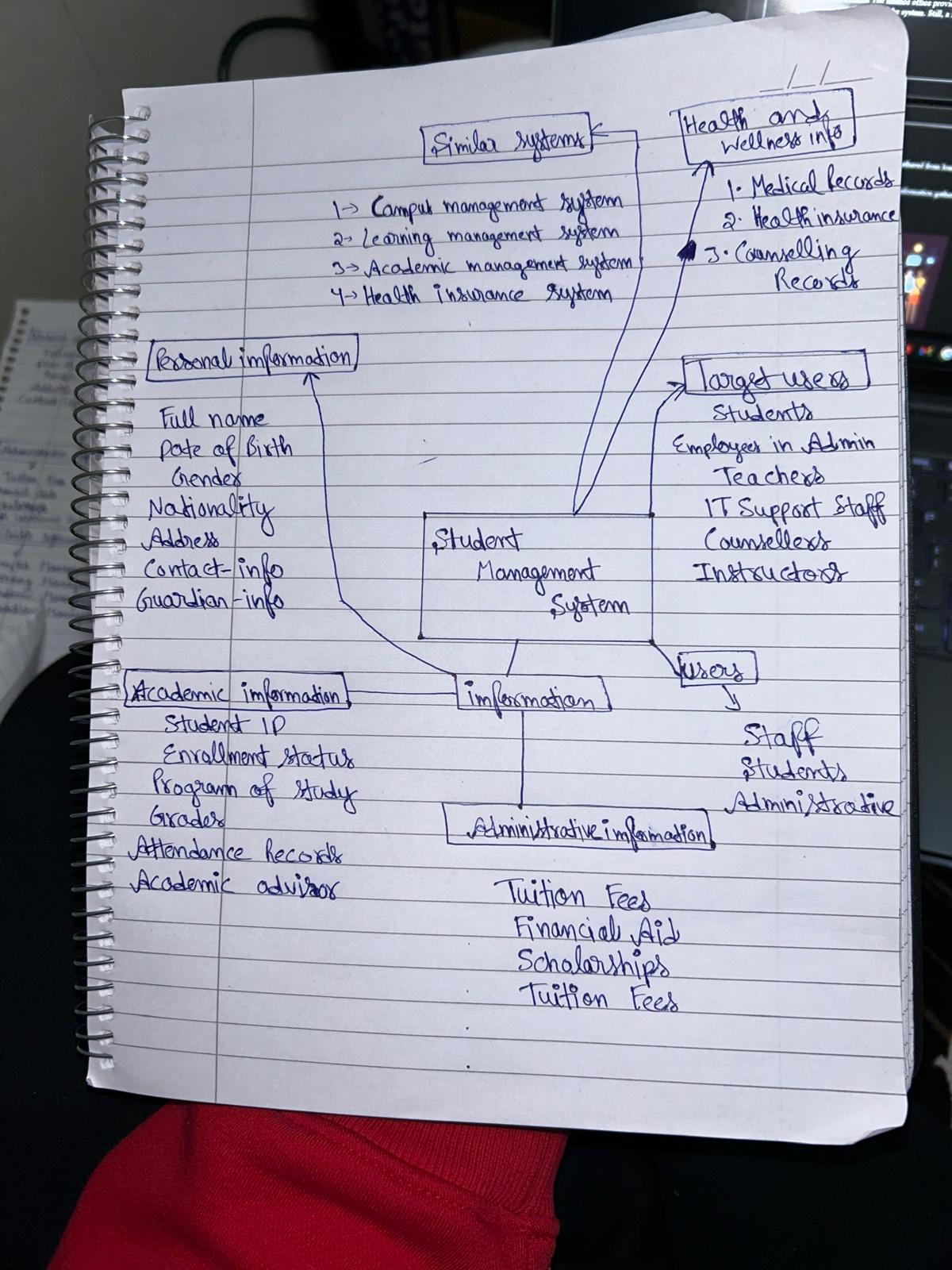
## BRAINSTORMING:

Before starting to make our system, we did Brainstorming just to get the overall idea of the system. We also got the requirements of the system by doing this activity like who are the users, what can be the benefits of the system, what are the events that will perform in the system, and many more ideas. Then, we started collecting the crucial points according to the time frame of the system and started working on them as a team.

Here I am providing some pictures of the brainstorming process:







## USER STORIES:

**User stories for the student**

* As a student, I want to update my contact information, including email, address, and phone number, so that I can receive correct information from the college and stay informed about new announcements and important updates without any difficulty. This will assist to stay connected with my studies if I am not able to attend lectures in any emergency.
* As a student, I want to view grades, including scores on individual assignments or exam so that I can easily monitor my academic progress and performance which will help me to stay on track with my educational goals. Additionally, knowing my grades will also help me to plan my studies accordingly if I am not able to match with passing criteria then I can change my strategies and can improve my performance.
* As a student, I want to know what classes I will be taking so I can organize my calendar and align with my academic and career goals as well as benefits include improved time management, greater planning goal alignment, and conflict avoidance. As a result, students are better prepared and engaged sin their next coursework.

**User stories for the teacher or academic staff member**

* As a teacher, I want to view the students grades so that I can help students to improve their performance by providing them extra support such as refer them for the extra classes during free time or by providing additional resources, giving feedback's and more. This will help me to understand the challenges faced by the students during their studies and later help them in improving the academic performance.
* As a teacher, I want to manage the lecture schedules according to my and students timetable so that I manage my lectures efficiently and can ensure that all the students can attend the classes without any scheduling conflicts. This will help me to improve the overall performance of the overall class just by organizing the lectures properly.
* As a teacher, I want to know how many students are enrolled in my course. So that I can manage classroom dynamics and plan resources effectively. This will help me to improved class management, better resource allocation and tailored instruction. As a result, it will be enhanced teaching effectiveness and improved student experience.

**User stories for the administrator:**

* As an administrator, I want to view the academic records of each student so that I can monitor their progress and identify any academic issues. This will help me to provide targeted support and interventions, such as academic support or additional tutoring, ensuring students have the resources they need to succeed.
* As an administrator, I want to update student contact information and personal details so that all communications and records are accurate and up to date. This will help in maintaining effective communication with students and their families, ensuring that important information reaches them without any issues.
* As an administrator, I want to manage student enrolment and course registrations so that I can ensure that students are enrolled in the right courses. This will help in balancing class loads and providing students with the courses they need to progress in their academic programs.

## USES CASES:

**Use cases for students:**

1. Update personal information.
2. View courses along with teachers’ name for a particular block.
3. View my enrollment history for a particular year.
4. View grades in particular subject.
5. View grades for all subject in a year sorted by grades
6. View lecture schedules

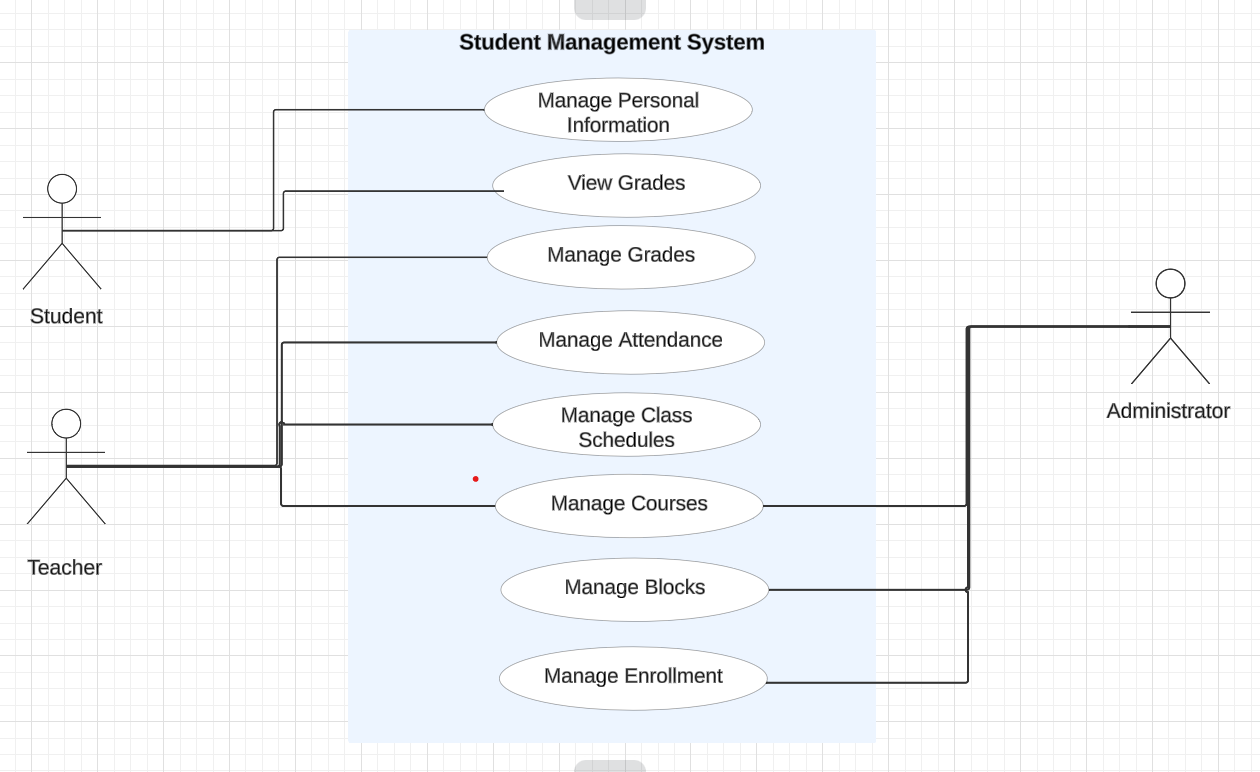
**Use cases for teachers or academic staff members:**

1. View number of students enrolled in courses for a specific year.
2. Update student’s grades.
3. View best grades of students in specific course.
4. Manage their courses. (If they want to update course details)
5. View which students are enrolled in a particular course.

**Use cases for the administrator:**

1. View which teachers are assigned to which courses and blocks.
2. Manage teacher and students information.
3. View the student performance in each course.
4. View number of students per block.
5. Manage courses. (By updating or deleting them).

## USE CASE DIAGRAM:



Retrieved from: [https://lucid.app/lucidchart/d4148ee5-d3fd-4378-a6e0-7220da95fc34/edit?page=.Q4MUjXso07N&invitationId=inv\_ed4ac094-9a50-496a-9a8e-2b9b785d58ac#](https://lucid.app/lucidchart/d4148ee5-d3fd-4378-a6e0-7220da95fc34/edit?page=.Q4MUjXso07N&invitationId=inv_ed4ac094-9a50-496a-9a8e-2b9b785d58ac)

## FUNCTIONAL AND NON – FUNCTIONAL:

**Functional Requirements:**

* Ability to store and manage student information, including personal details academic courses and activities, and schedules. Use Cases ability to schedule classes and register students for lectures.
* Ability to create and manage student and parent profiles, including their contact information.
* Ability to manage university activities and its instructors.
* Ability to provide a user- friendly interface for all system users.

**Non – functional requirements:**

* The system should handle many users and transactions efficiently.
* The system should be easy to update and maintain.
* Ensure integration with other systems such as academic record systems financial system.
* Implement backup and recovery solutions to prevent data loss.

## ACTORS

1. **Students**: Students would be the primary users of the system they could check their class schedule, grades, sign them up for activities, see the upcoming events or changes in their curriculum, register for courses and they can update their information.
2. **Teachers**: Tutors or instructors would use the system to manage class roaster, input grades, create assignments and they have the access to communicate with students.
3. **Administrator**: The administrator can manage the overall system management, including user accounts, course offerings and reporting.

## EVENTS

* **Update information:** Students can update their information.
* **Grade Submission**: Teachers could submit or update grades of the students.
* **Course Creation**: Administrators or faculty create or modify courses.
* **Schedule Update**s: Changes to class schedules are made.
* **Attendance**: Teachers record attendance.
* **Assignment Submission**: Students submit assignments.
* **Profile Update**: Users update their personal or professional information.

## AGILE METHODOLOGY

We have used **Agile Methodology** to build our **University Student Management System**. We have chosen this model because it is based on iterative methodology which basically means to start with basic skeleton and make regular changes until the final release. Moreover, it also allows user to understand the **requirements** of system and then later follow up with **planning** by evaluating the existing system**, design**, **developmen**t, **testing** and finally **deploymen**t which is quite easy and straightforward to utilize by the user in building the new system. Firstly, we started by collecting the requirements from the user then after making changes throughout the process we are able the deliver the result. Apart from that, this model allowed us to start by building basic structure with limited features and then follow up with continuous changes and delivery to the users.

A diagram of a process

Description automatically generated

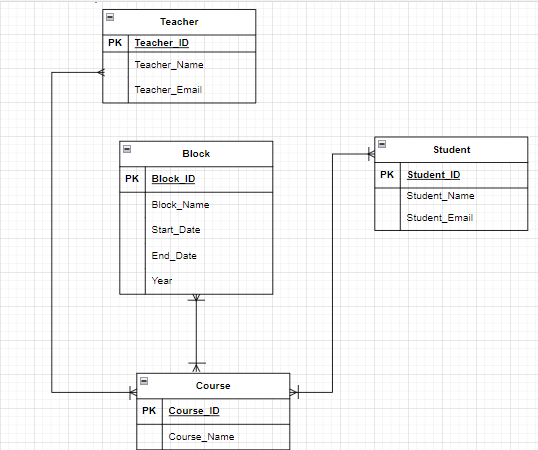
Retrieved from: <https://asana.com/resources/agile-methodology>

# Entity Relationship Diagram

1. **LOGICAL ERD**

**Relationships among different entities:-**

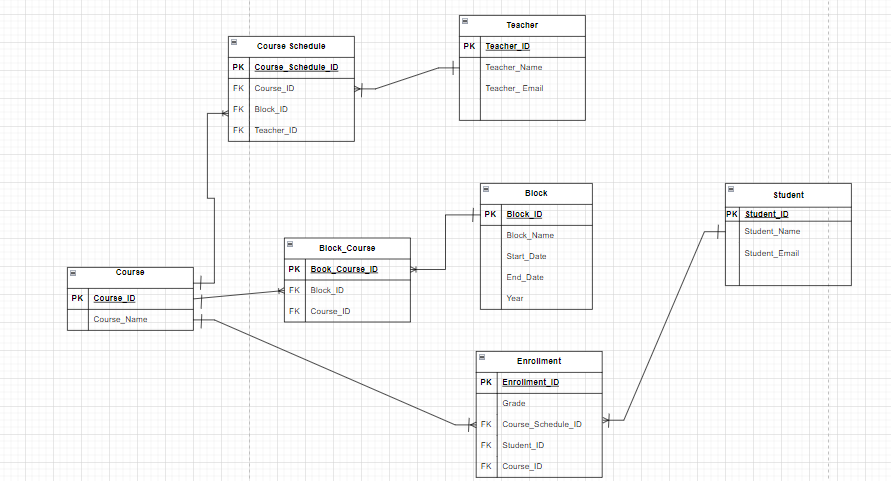
* **Teacher to Course -** teacher can teach many courses to the students and a course can be teach by many teachers. So it is a **many to many relationship.**
* **Block to Courses -** A block can have many courses at a time similarly a course can be taught in many blocks. Therefore, It is also a **many to many relationship.**
* **Student to Course -** A student can study multiple courses at a time and in the same way, a course can be studied by many students. Clearly, it is another **many to many relationship.**

****

1. **PHSYSICAL ERD 1**

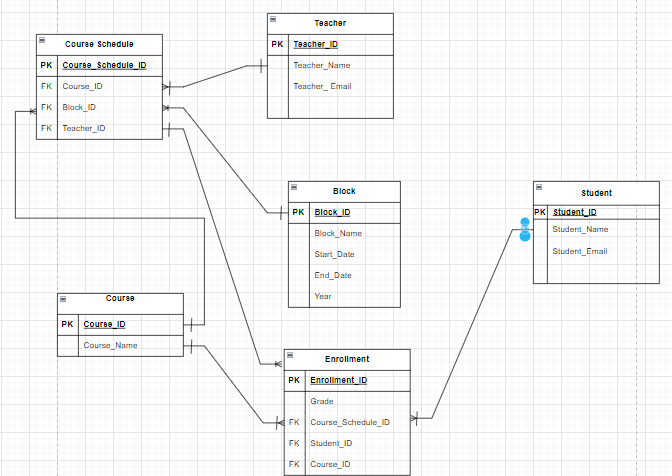
To break these many to many relationships we have added three associative entities name:-

1. Course\_Schedule - Between Course and Teacher.
2. Block\_Course - Between Block and Course.
3. Enrollment - Between Student and Course.

****

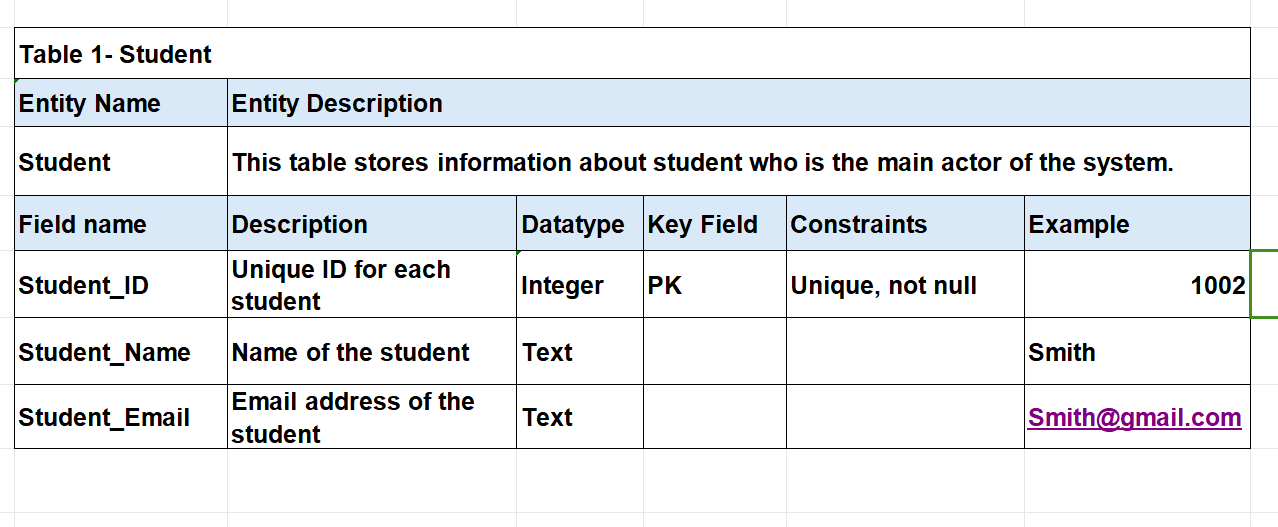
1. **PHSYSICAL ERD 2**

To further simplify the Physical ERD we have created the another ERD with just six entities.

****

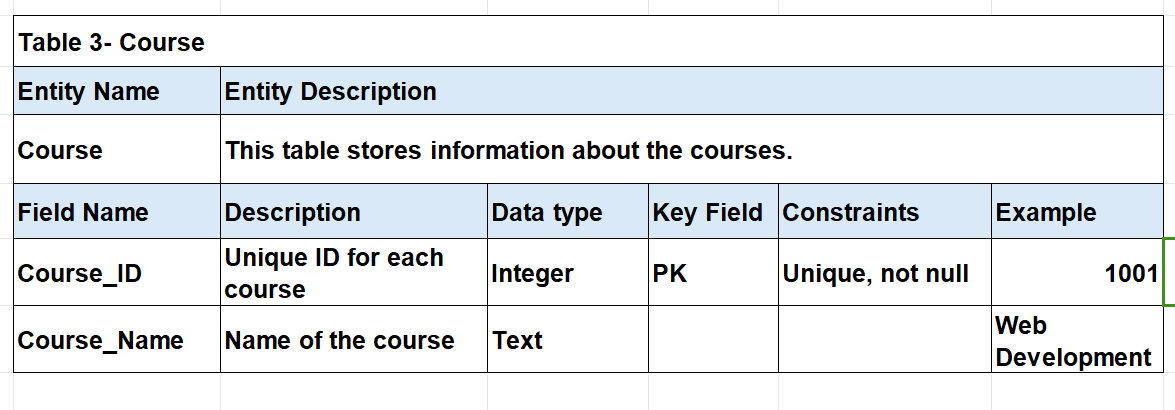
# Table Designs – Data Dictionary

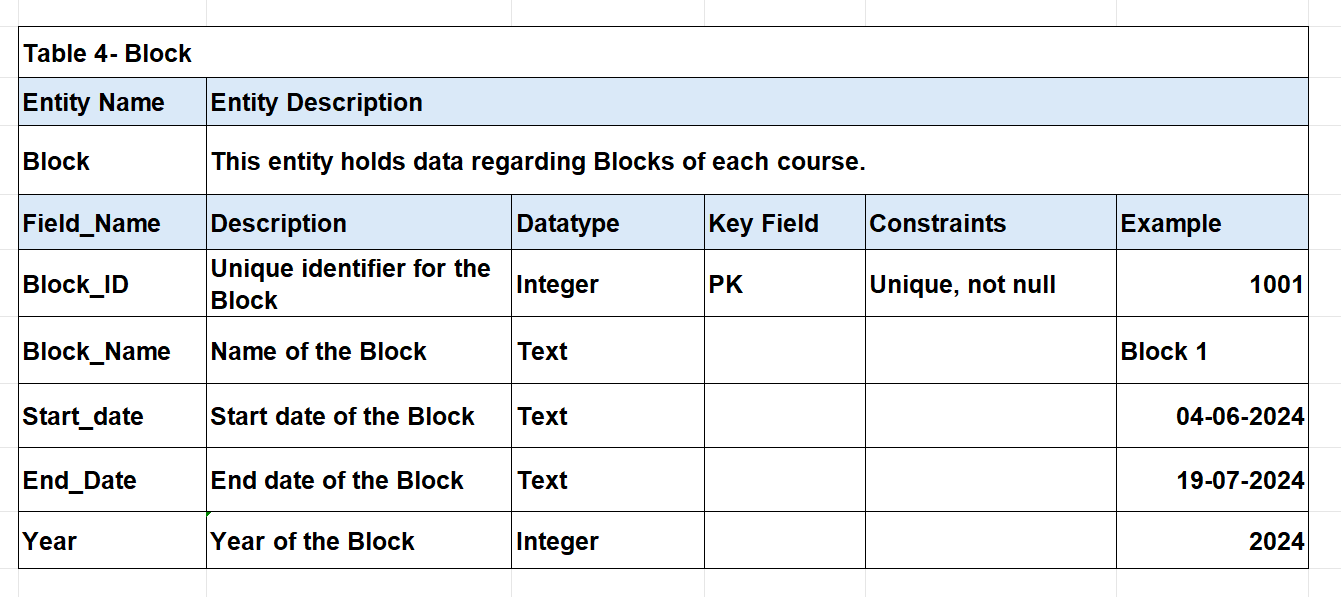
So we have created 6 data dictionary table according to our Physical ERD 2



A screenshot of a computer

Description automatically generated





A screenshot of a document

Description automatically generated

A screenshot of a computer

Description automatically generated

# Contributions

**Paras**

* I have written executive summary for the University Student Management System.
* I have also written user stories and use cases for the Teacher.
* I have created ERD in the notebook, then my team mates helped me in finalize the ERD after discussing with the lecturers.
* I have created data dictionary table for the Student and Teacher.
* I have also made use case diagram in the Lucid Chart.
* I have created tables and inserted data in the Notepad and then after collecting the other tables I created them in sqlite3.
* I have formatted and edited the whole report.

**Gursehbaj Singh**

* I have written user stories and use cases and Agile methodology.
* I have created the Data Dictionary of two entities which are Course and Course Schedule.
* I helped in creating the ERD diagram then finalize that diagram after discussing with the group members.
* I have created the physical ERD diagram using lucid chart the draw.io website.
* I Similarly, I have created the logical diagram using Draw website and create the third EDR just for the explanation.
* I have written the MySQL code for creating table and inserting data in that tables.

**Harpreet Kaur**

* In this project, I have written uses cases and user stories for Student..
* I have also written the functional and non-functional requirements for the system.
* Moreover, I have created Block and Enrolment table on notepad and then send it to the leader to run them in sqlite3.
* I also helped to the team leader to modify the ERD Diagram according to the requirements of the system.
* I helped the group leader in creating the use cases diagram in the lucid chart.

**Prabhjot Singh**

* In this project, I have written actors point for the university student management system.
* I have also written Events.
* Furthermore, I created the flow chart for the system.
* I also helped my team members for finalize the use cases diagram in the lucid chart.
* I have also added the References to acknowledge the original content.

# References

Laoyan, S. (2024). *What is Agile methodology?* Retrieved from Asana: https://asana.com/resources/agile-methodology

*Student Managent System*. (n.d.). Retrieved from LovelyCoding.Org: https://www.lovelycoding.org/student-management-system/#:~:text=Hardware%20Requirements%20of%20Student%20Management%20System&text=Processor%3A%20Intel%20P%2DIV%20System,Hard%20Disk%3A%2040%20Gb

*Use Case Diagram*. (n.d.). Retrieved from LucidChart: https://lucid.app/lucidchart/d4148ee5-d3fd-4378-a6e0-7220da95fc34/edit?invitationId=inv\_ed4ac094-9a50-496a-9a8e-2b9b785d58ac&page=.Q4MUjXso07N#