CHAPTER-1

INTRODUCTION

## OBJECTIVES:

* + - The main objective of the project is to design and develop a user friendly-system
    - Easy to use and an efficient computerized system.
    - To develop an accurate and flexible system, it will eliminate data redundancy.
    - To study the functioning of Students management System.
    - To make a software fast in processing, with good user interface.
    - To make software with good user interface so that user can change it and it should be used for a long time without error and maintenance.
    - To provide synchronized and centralized farmer and seller database.
    - Computerization can be helpful as a means of saving time and money.
    - To provide better Graphical User Interface (GUI).
    - Less chances of information leakage.
    - Provides Security to the data by using login and password method.
    - To provide immediate storage and retrieval of data and information.
    - Improving arrangements for students’ coordination.
    - Reducing paperwork.

## LIMITATIONS:

* + - Time consumption in data entry as the records are to be manually maintained faculties a lot of time.
    - Lot of paper work is involved as the records are maintained in the files and registers.
    - Storage Requires as files and registers are used the storage space requirement is increased.
    - Less Reliable use of papers for storing valuable data information is not at all reliable.
    - Aadhar linkage with the official Aadhar database has not been done.

CHAPTER-2

## PROPOSED SYSTEM:

While there has been no consensus on the definition of Students Management in the literature, they have proposed that researchers adopt the below definition to allow for the coherent development of theory in the colleges. In order to have a successful students management, we need to make many decisions related to the flow of marks, attendance, and data. Each record should be added in a way to increase the scalability. Student management is more complex in colleges and other universities because of the impact on people’s number requiring adequate and accurate information of students need.

**CHAPTER 3**

**3. DATABASE DESIGN**

**Software Requirements Specification**

* **SOFTWARE REQUIREMENTS:**

Frontend- HTML, CSS, Java Script, Bootstrap

Backend-Python flask (Python 3.7) , SQLAlchemy,

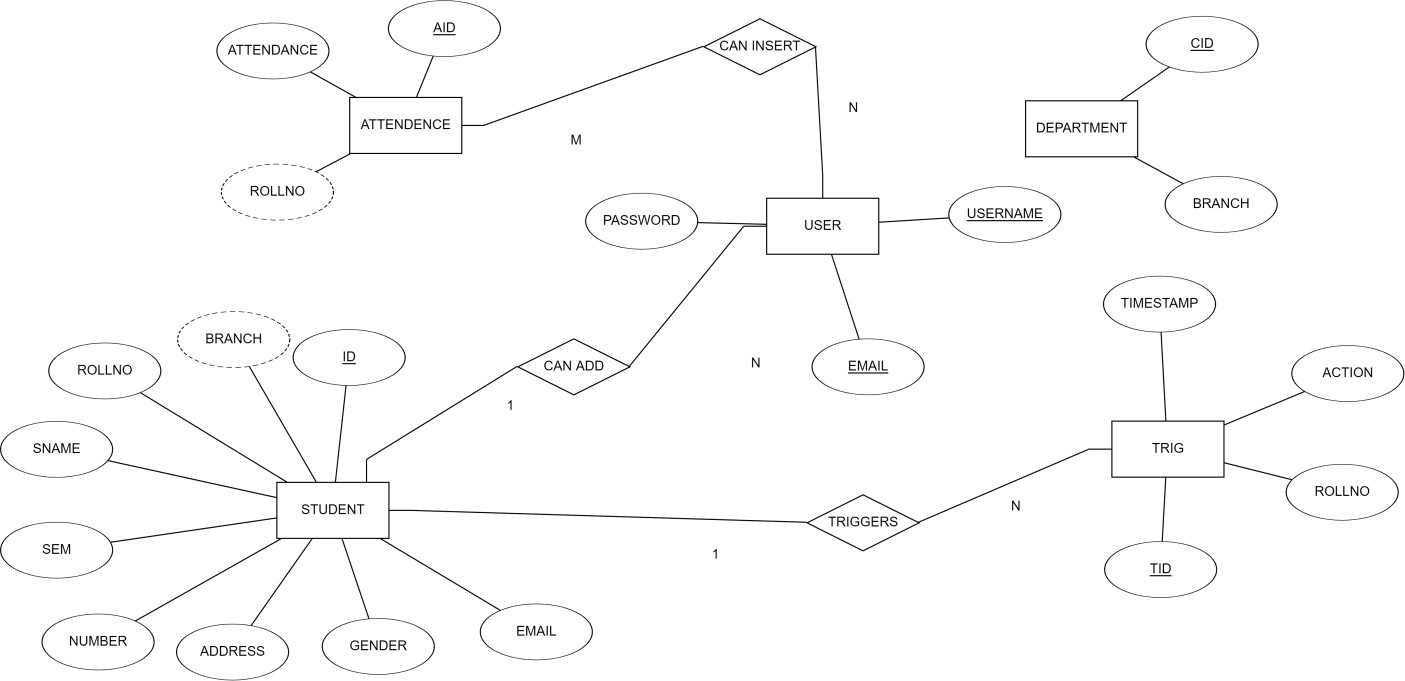
* + - * Operating System: Windows 10
      * Google Chrome/Internet Explorer
      * XAMPP (Version-3.7)
      * Python main editor (user interface): PyCharm Community
      * workspace editor: Sublime text 3

**HARDWARE REQUIREMENTS:**

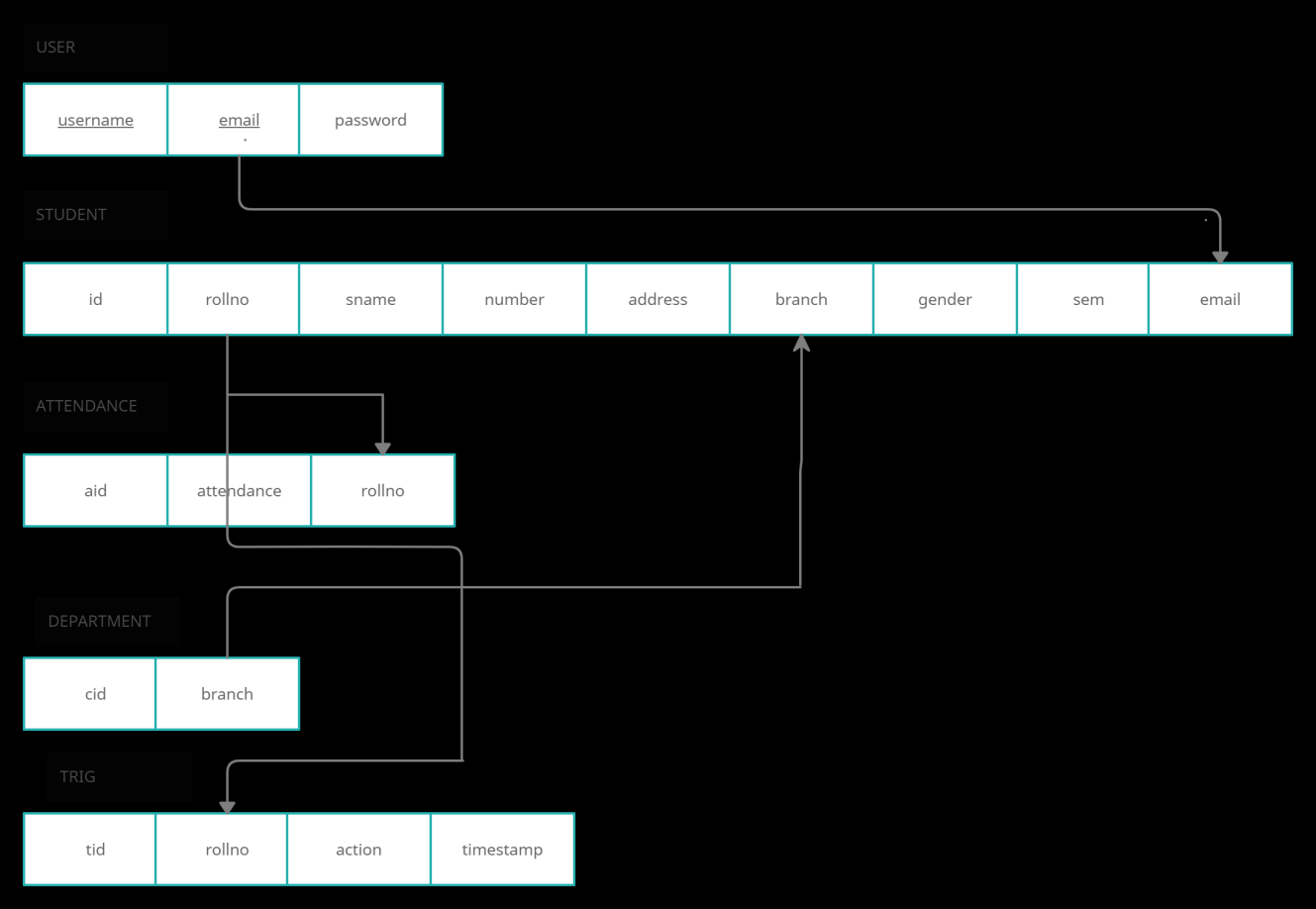
* + - * Computer with a 1.1 GHz or faster processor
      * Minimum 2GB of RAM or more
      * 2.5 GB of available hard-disk space
      * 5400 RPM hard drive
      * 1366 × 768 or higher-resolution display
      * DVD-ROM drive

**Conceptual Design:**

**ER-Diagram:**



**Schema Diagram:**



**Implementation**

An "implementation" of Python should be taken to mean a program or environment which provides support for the execution of programs written in the Python language, as represented by the [CPython](https://wiki.python.org/moin/CPython) reference implementation.

There have been and are several distinct software packages providing of what we all recognize as Python, although some of those are more like distributions or variants of some existing implementation than a completely new implementation of the language.

**Back End (MySQL) Database:**

A Database Management System (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems. Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. A DBMS includes:

* A modeling language to define the schema of each database hosted in the DBMS, according to the DBMS data model.
* The dominant model in use today is the ad hoc one embedded in SQL, despite the objections of purists who believe this model is a corruption of the relational model, since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.
* Data structures (fields, records, files and objects) optimized to deal with very large amounts of data stored on a permanent data storage device (which implies relatively slow access compared to volatile main memory).A database query language and report

writer to allow users to interactively interrogate the database, analyze its data and update it according to the users’ privileges on data.

* Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called sub schemas. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and student data.
* If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it may not leave an audit trail of actions or provide the kinds of controls necessary in a multi-user organization. These controls are only available when a set of application programs are customized for each data entry and updating function.
* A transaction mechanism, that ideally would guarantee the ACID properties, in order to ensure data integrity, despite concurrent user accesses (concurrency control), and faults (fault tolerance).
* It also maintains the integrity of the data in the database.
* The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can help prevent duplicate records via unique index constraints; for example, no two customers with the same customer numbers (key fields) can be entered into the database. See ACID properties for more information (Redundancy avoidance).

When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. to the Organizations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and analysis. Overall systems design decisions are performed by data administrators and systems analysts. Detailed database design is performed by database administrators.

**SQL:**

Structured Query Language (SQL) is the language used to manipulate relational databases. SQL is tied very closely with the relational model.

* In the relational model, data is stored in structures called relations or tables*.*

SQL statements are issued for the purpose of:

* Data definition: Defining tables and structures in the database (DDL used to create, alter and drop schema objects such as tables and indexes)