Database Design Document

For

Hostel Management System

Version 1.0

Prepared by,

Group Number: 4

Shaaheen A M	**B181134CS**	**shaaheen_b181134cs@nitc.ac.in**
Mathew Jose Mammoottil	**B180586CS**	**mathew_b180586cs@nitc.ac.in**
Emmanuel Marian Mathew	**B180347CS**	**emmanuel_b180347cs@nitc.ac.in**
Sidharth Menon	**B180561CS**	**sidharth_b180561cs@nitc.ac.in**
Vishnu Sajith	**B180474CS**	**vishnu_b180474cs@nitc.ac.in**

Instructor: **Abdul Nazeer**

Course: **Database Management System**

Date: **27 - 11 - 2020**

Contents

1. Purpose

The Database Design Document maps the logical data model to the target database management system with consideration to the system's performance requirements. The Database Design converts logical or conceptual data constructs to physical storage constructs (e.g., tables, files) of the target Database Management System (DBMS)

1.1. Document objectives

The Database Design Document has the following objectives:

- To describe the design of a database, that is, a collection of related data stored in one or more computerized files that can be accessed by users or computer developers via a DBMS.
- To serve as a basis for implementing the database and related software units. It provides the acquirer visibility into the design and provides information necessary for software development.

1.2. Intended Audience

This document is intended for the instructors of the course C3002D 'Database Management Systems' for their review and monitoring progress of the project. The document is also for the project team to use in order to analyze the system requirements and any additional details related to the project.

1.3. Acronyms and Abbreviations

- HMS - Hostel Management System
- PK - Primary Key
- FK - Foreign Key

2. Assumptions, Constraints and Dependencies

2.1. Assumptions and Dependencies

The following are the assumptions and dependencies that are imposed upon the implementation of the HMS:

- The application must have a user-friendly interface that is simple enough for all types of users to understand.
- The user of the application is expected to have a general knowledge of basic computer skills and the internet.
- It is assumed that the students who register for the hostel will have an email.
- It is assumed that all the visitors that visit a student will be having a phone number.

2.2. Constraints

The development of the system will be constrained by the software used to make the application. The Flutter software used to make the mobile application will only be compatible for making android apps for users. For the web app we need a PC with 4 GB RAM. These are the hardware and software constraints in the design and implementation of the web or mobile application of the Hostel Management System.

3. System Overview

The Hostel Management System is a web, or a mobile application developed to let the hostel authorities maintain a database of the occupants of the hostel, share information and keep track of the occupants or students. It also lets the students or the occupants lodge complaints, add suggestions, and access the latest updates from the authorities.

The application allows the user to sign up as an administrator or as a student. After signing up, each time they use the application they can login and continue to use the application in their preference. Depending on the user, the application will have a different user interface. For instance, if the user is an administrator, they will have certain features to allocate rooms to occupants, search for occupants, add visitor details, view suggestions and complaints and so on. If the user is a student, they will have features like check mess dues, search for lost and found, complaint or request for maintenance etc.

The main aim of implementing the Hostel Management System is to reduce human error, strength, and strain of manual labour, implement high security, avoid data redundancy, improve data consistency and to provide a platform to get things done in our fingertips.

3.1. Objectives we are trying to achieve through this project:

To shift from paper-based data handling to a more digital and computerized system. To make the hostel management data handling more easy and efficient. To make the synchronization with college data more efficient. Help Save time and effort.

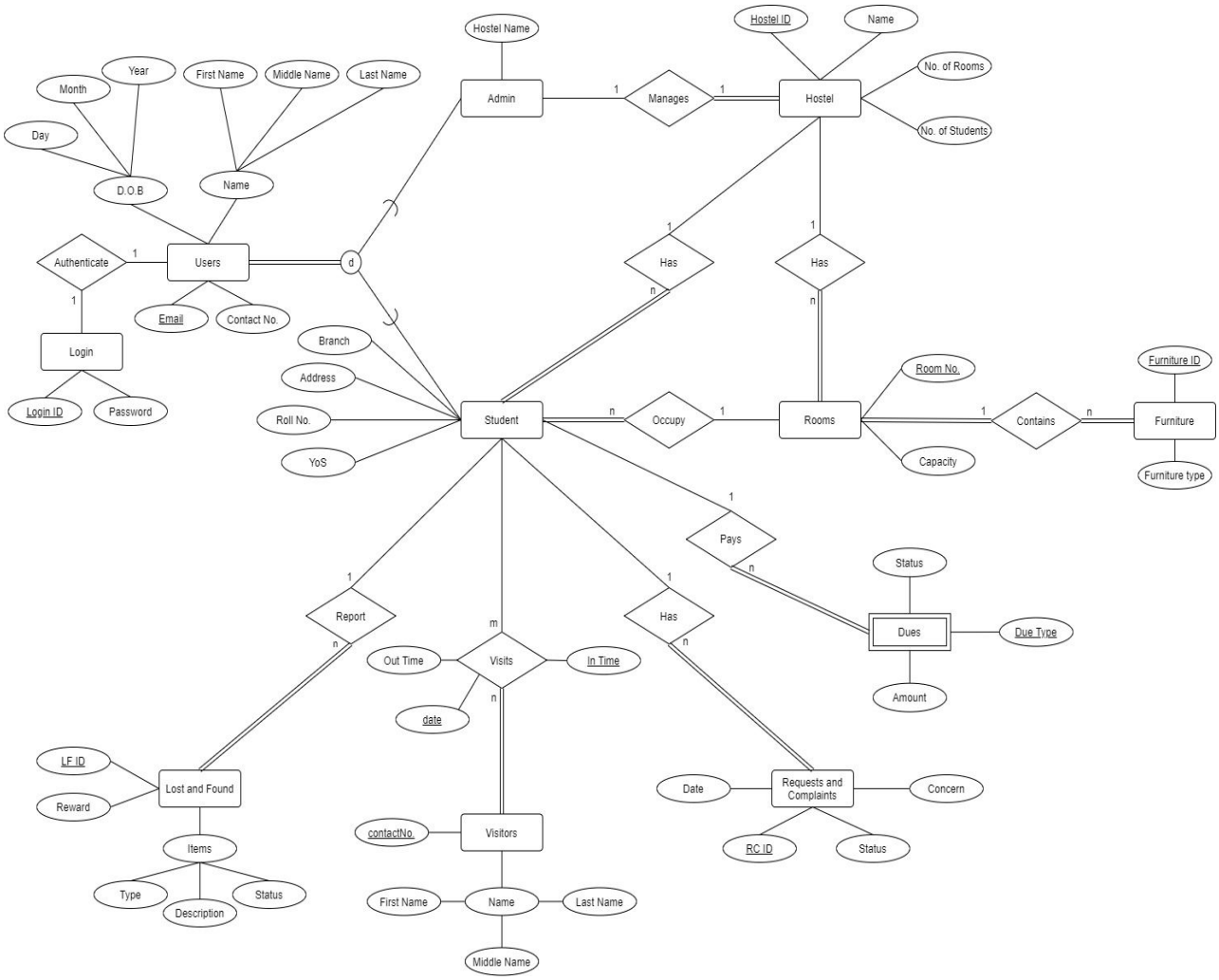
3.2. Scope of this project includes but not limited to:

- Providing a friendly interface for the users and the admin
- Allowing the easy search for occupants and their details
- Minimizing human error while managing the database
- Avoiding data redundancies in the database

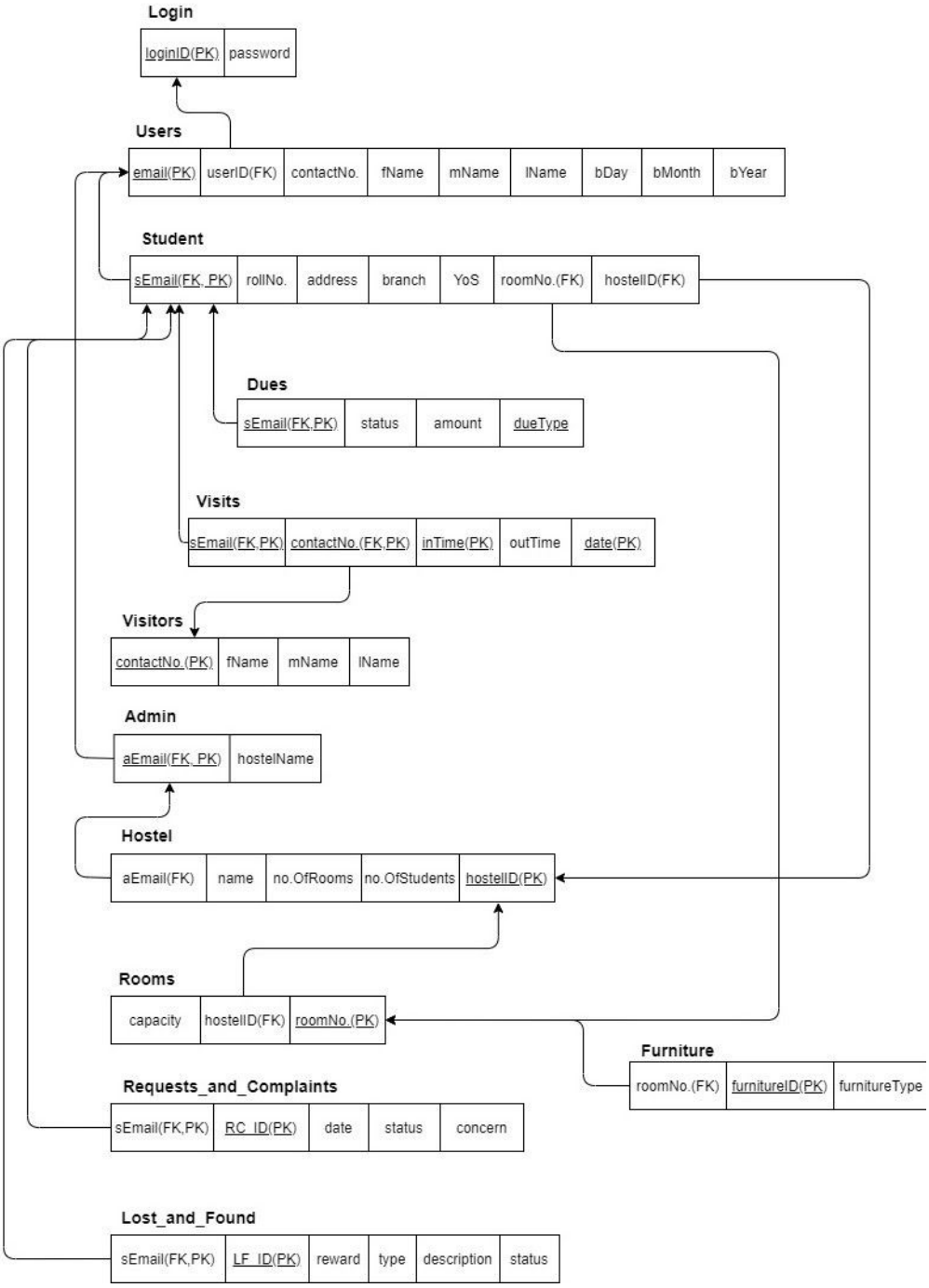
- Ensuring high security of the data
- Facilitating the easy access to all the details of any student for the administration
- Recognizing complaints or requests for maintenance by the occupants
- Listing of Lost and Found items in the hostel
- Potentially developing the Database Management System for full-time use to maintain the database of the NIT Calicut hostels.

4. Database Design

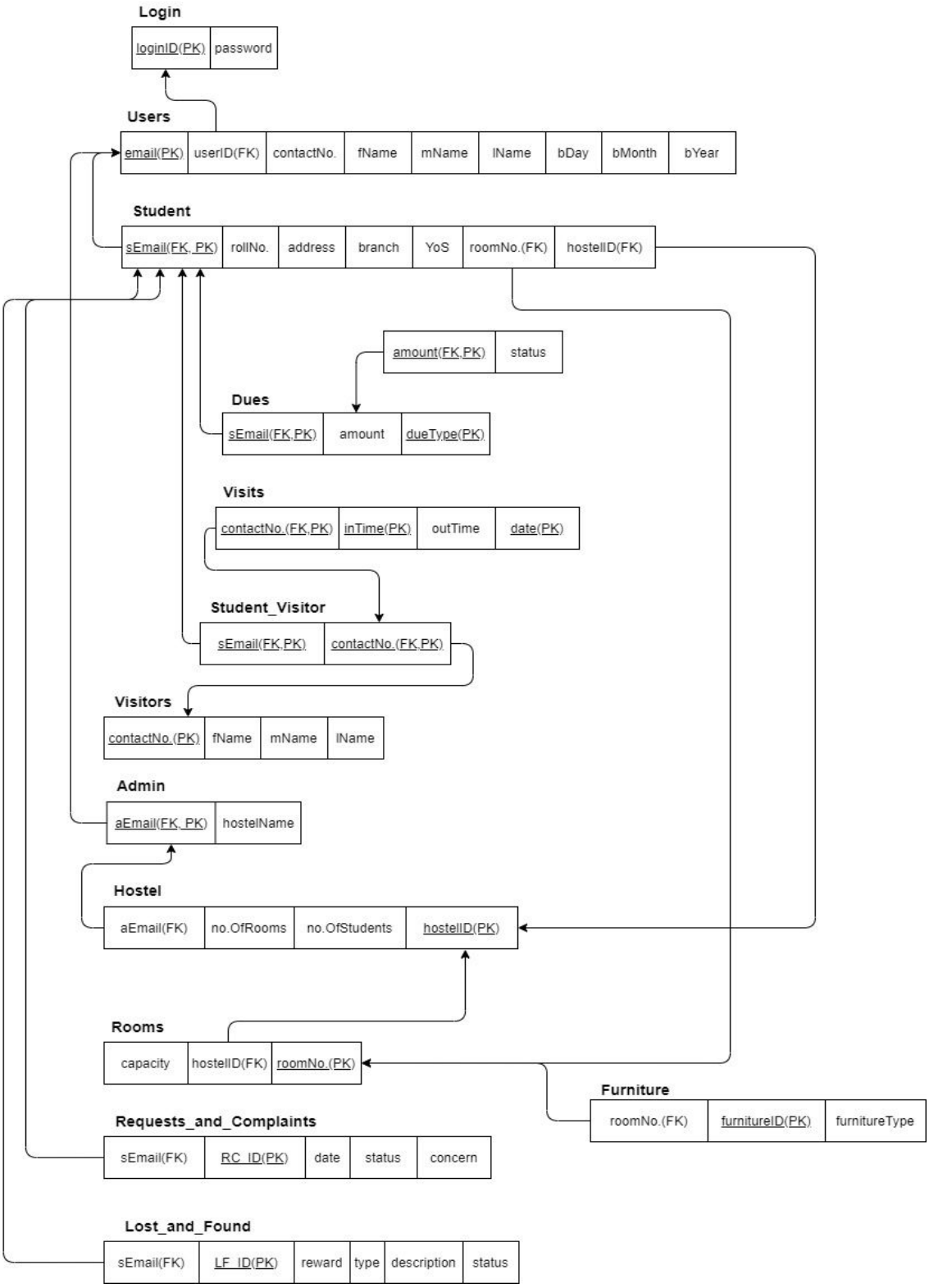
4.1. Entity Relationship Diagram



4.2. Relational Schema



4.3. Normalized Form



5. Entities and Attributes

This section of the document explains the various entities involved in the project, their attributes, and how they work together. This section is intended to make the design more clear and understandable.

The following is the list of entities involved in the project:

1. **LOGIN**
2. **USERS**
3. **ADMIN**
4. **STUDENT**
5. **DUES**
6. **HOSTEL**
7. **ROOMS**
8. **FURNITURE**
9. **REQUESTS_AND_COMPLAINTS**
10. **VISITORS**
11. **VISITS**
12. **LOST_AND_FOUND**

5.1 LOGIN

Every user will have a Login ID. This entity represents the Login details of each user. The Login entity takes part in the following relationship:

- User : The Login authenticates the Users of the database.

The following are the attributes of the Login entity:

NAME	DATA TYPE	TYPE
loginID	string	Primary Key
password	string	Non-Key

5.2 USERS

This entity represents each user and stores the details of each user. The Users entity is divided into two disjoint entities, namely the Admin entity and the Student entity. The Users entity takes part in the following relationship:

- Login : The Login authenticates the Users of the database.

The following are the attributes of the Users entity:

NAME	DATA TYPE	TYPE
email	string	Primary Key
userID	string	Foreign Key
contactNo.	long integer	Candidate Key
fName	string	Non-Key
mName	string	Non-Key
lName	string	Non-Key
bDay	integer	Non-Key
bMonth	integer	Non-Key
bYear	integer	Non-Key

5.3 ADMIN

Every hostel will have an administrator and each administrator will be represented by this entity. The Admin entity takes part in the following relationships:

- Hostel : The Admin manages the Hostel.

The following are the attributes of the Admin entity:

NAME	DATA TYPE	TYPE
aEmail	string	Primary Key, Foreign Key
hostelName	string	Non-Key

5.4 STUDENT

Each student in the hostel is represented by the ‘Student’ entity. The Student entity takes part in the following relationships:

- Hostel : Every Student in the system will have a Hostel.
- Rooms : Every Student in the system will occupy a Room.
- Requests and Complaints : Every student can have Requests and/or Complaints to report.
- Visitors : Every student can have visits from their Visitors in the hostel.
- Lost and Found : Every student can report Lost and Found items.

The following are the attributes of the Student entity:

NAME	DATA TYPE	TYPE
sEmail	string	Primary Key, Foreign Key
rollNo.	string	Non-key
address	string	Non-key
branch	string	Non-key
YoS	integer	Non-key
roomNo.	integer	Foreign Key
hostelID	string	Foreign Key

5.5 DUES

Every student will have a certain amount of dues for either the hostel or the mess, or as a fine that has been imposed on the student due to a violation. The dues for each student will be represented by this entity. The Dues entity takes part in the following relationships:

- Student : The Student has to pay the Dues.

The following are the attributes of the Admin entity:

NAME	DATA TYPE	TYPE
aEmail	string	Primary Key, Foreign Key

amount	long integer	Non-Key
status	string	Non-Key
dueType	string	Primary Key

5.6 HOSTEL

An institution has multiple hostels. Each hostel will be represented by this entity. The Hostel entity takes part in the following relationships:

- Admin : The Admin manages the Hostel.
- Student : The Hostel has Students.
- Rooms : The Hostel has Rooms.

The following are the attributes of the Hostel entity:

NAME	DATA TYPE	TYPE
aEmail	string	Foreign Key
name	string	Non-Key
no.OfRooms	integer	Non-Key
no.OfStudents	integer	Non-Key
hostelID	string	Primary Key

5.7 ROOMS

Every hostel has multiple rooms and each room is represented by this entity. The Rooms entity takes part in the following relationships:

- Hostel : The Hostel has Rooms.
- Student : Every Student in the system will occupy a Room.
- Furniture : The Room contains the Furniture.

The following are the attributes of the Rooms entity:

NAME	DATA TYPE	TYPE
roomNo.	integer	Primary Key
capacity	integer	Non-Key
hostelID	string	Foreign Key

5.8 FURNITURE

Every hostel room will have furniture and each piece of furniture will have a furniture ID and is represented by this entity. The Furniture entity takes part in the following relationships:

- Rooms : The Room contains the Furniture.

The following are the attributes of the Admin entity:

NAME	DATA TYPE	TYPE
roomNo.	integer	Foreign Key
furnitureID	string	Primary Key
furnitureType	string	Non-Key

5.9 REQUESTS_AND_COMPLAINTS

Every student has the ability to lodge requests or complaints regarding their stay in the hostel rooms. This entity represents each of those requests and/or complaints. The Requests and Complaints entity takes part in the following relationship:

- Student : Every student can have Requests and/or Complaints to report.

The following are the attributes of the Requests and Complaints entity:

NAME	DATA TYPE	TYPE
sEmail	string	Foreign Key, Primary Key
RC_ID	string	Primary Key

date	string	Non-Key
status	string	Non-Key
concern	string	Non-Key

5.10 VISITORS

Every student in the hostel can have visitors in the hostel. The Visitors entity represents the visitors related to the particular student in the hostel. The Visitors entity takes part in the following relationship:

- Student : Every student can have visits from their Visitors in the hostel.

The following are the attributes of the Visitors entity:

NAME	DATA TYPE	TYPE
fName	string	Non-Key
mName	string	Non-Key
lName	string	Non-Key
contactNo.	long integer	Primary Key

5.11 VISITS

The details of every visit by any student's visitor is stored in the database by the means of this relationship. Any student or a group of students can have a certain number of visits from visitors. The Visits relationship involves the entities :

- Student
- Visitors

The following are the attributes of the Visits relationship:

NAME	DATA TYPE	TYPE
sEmail	string	Primary Key, Foreign Key
contactNo.	long integer	Candidate Key

inTime	string	Primary Key
outTime	string	Non-Key
date	string	Primary Key

5.12 LOST_AND_FOUND

Many items can be misplaced in the hostel or found by any member of the hostel. The Lost and Found entity will represent all of the items in the hostel Lost and Found list. The Lost and Found entity takes part in the following relationships:

- Student : Every student can report Lost and Found items.

NAME	DATA TYPE	TYPE
LF_ID	string	Primary Key
sEmail	string	Primary Key, Foreign Key
reward	string	Non-Key
type	string	Non-Key
description	string	Non-Key
status	string	Non-Key

6. Other Non-functional Requirements

6.1. Performance Requirements

- The database should be able to store information of a large collection of records and the changes made from one user should be updated for all users in real time.
- The system must be responsive and should have less delays.
- The application should be able to support multiple users at a time.

6.2. Safety and Security Requirements

- The database may take time to load or crash in case a lot of users are trying to access together. In such cases, the restoration of the database might take time.
- Due to some virus, the operating system might take time to boot up which would cause certain delays.
- Critical information should be encrypted and sent to be stored in the server.
- Certain tasks or access to certain functionalities should be authorized and only those members should be able to use those functions.
- Keep a log on who modified data and restrict communication between certain parts of the application.
- Must prevent the accidental access, modification and destruction of the database.

6.3. Software Quality Attributes

- Availability Requirement: The system should be available to the user 24/7. The changes in relation to the last date to pay bills or the postponing of an event should reach all students. Hence the availability of the system must be high.

- Efficiency Requirement: In case of a system repair, scheduled software update or a system crash, the system should be able to restore and recover quickly.

- Accuracy: It should be able to provide precise real time data about the various users. It should regularly update the *dues* lies and the Administration should be able to know whether a student needs to pay fees or a *mess* card can be issued directly.