Hostel Management System

Project Report : CS315 - Introduction to Database Systems

Akshay Kumar & Abhijit Sharang
Advisor : Dr. Harish Karnick
Department of Computer Science and Engineering
{kakshay,abhisg}@iitk.ac.in

April 9, 2013

We present to you Hostel Management System - a fully automated portal to cater to various affairs of a hostel. This is capable of handling Book Club, Complains, Washerman Transaction, Canteen and Mess Transactions. The Relational Database System used is mysql whereas the frontend is powered by php, ajax and other internet technologies.

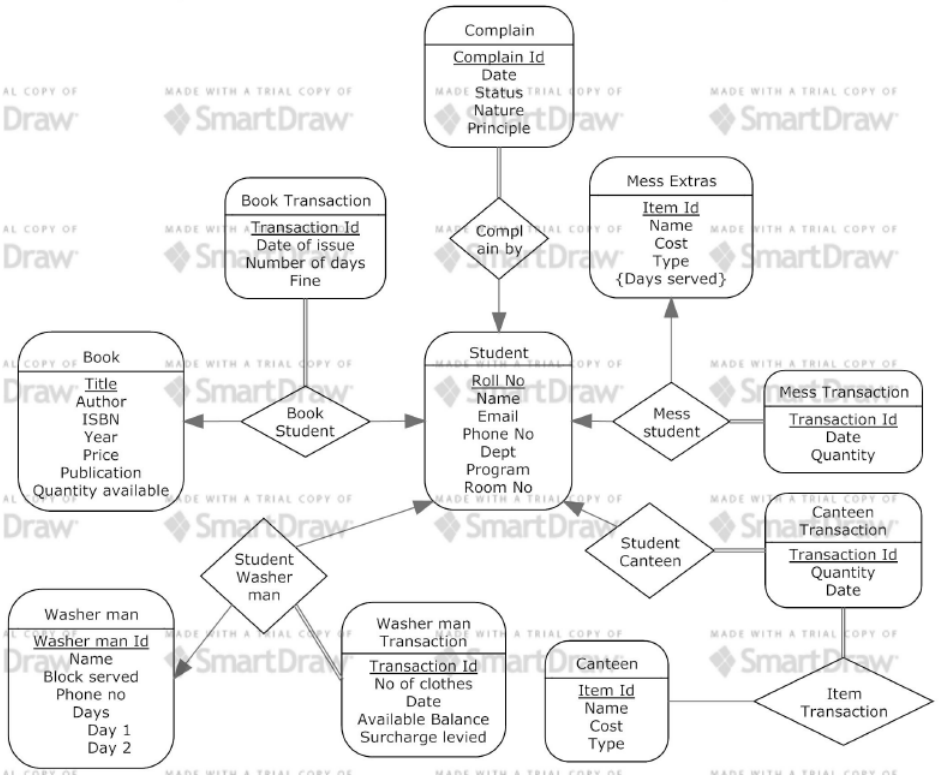
INTRODUCTION

Our project involved developing a fully automated hostel management system which can cater to the needs of students & administrators. Presently, there's no such system in place and most of the work done is not computerized. With the help of this portal, the time spent in such redundant chores can be heavily cut down and will also ensure transparency. For now, the day to day tasks of mess, canteen, library, complaints, washerman, etc. have been incorporated into the framework but it can be extended to suit the exact needs of a hostel. Since the amount of work is large, we have decided to split the work between two people.

FUNCTIONAL SPECIFICATIONS AND FEATURES

- **Database :** We created the database for this project right from scratch. All the information has been manually entered. To implement it actually, it can easily be imported from a csv file.
- **Login :** The interface provides login only to select authorities. More specifically, no student has been provided access to the database. Only select individuals can modify/update the database.
- **Session Management :** PHP session variable is used to pass on information from one page to another while a user is logged in.

ER DIAGRAM



RELATIONSHIP SCHEMA

Following tables were used:

1. **Student** (RollNo, Name, RoomNo, Department, Program, Email, CellNo, WasherManBalance)
2. **Book** (Title, ISBN, Year, Price, Publication, QuantityAvailable)
3. **BookAuthor** (ISBN, Author)
4. **BookTransaction** (TransactionId, DateOfIssue, NoOfDays, Fine, DateOfReturn)
5. **Student_Book_Transaction** (TransactionId, RollNo, ISBN)
6. **Complain** (ComplainId, RollNo, Nature, Date, Description, Status, Feedback)
7. **WasherMan** (WasherManId, Name, BlockServed, CellNo, Day1, Day2)
8. **WasherManTransaction** (TransactionId, NoOfClothes, Cost, Date, Surcharge)
9. **Student_WasherMan_Transaction** (TransactionId, RollNo, WasherManId)
10. **Canteen** (ItemId, Name, Cost, Type)
11. **CanteenTransaction** (TransactionId, Date)

12. **Item_Transaction** (ItemId, TransactionID, Quantity)
13. **Student_Canteen** (TransactionId, RollNo)
14. **MessExtras** (ItemId, Name, Cost, Type)
15. **MessExtrasDays** (ItemId, Day)
16. **MessTransaction** (TransactionID, Date, Quantity)
17. **Student_Mess_Transaction** (TransactionID, RollNo, ItemID)

Note:

1. All the tables have been normalized according to **BCNF : Boyce-Codd Normal Form**.
2. Aforementioned tables are part of the relational database schema. Apart from this few other tables to take care necessities like login, etc. were used.
3. Tables **BookAuthor** and **MessExtrasDays** were used to support multivalued attribute.
4. Tables whose name contain an underscore (_) correspond to a relationship between two entity sets.

Future Scope

1. Because of the relatively simple schema, it can be imporvised and extended to incorporate many other features.
2. The Graphical Interface is plain and simple. It can be improved slightly.
3. The problem of same student having multiple Roll Number has not been addressed. In this scenario, the primary key for student must be changed from Roll Number to Email.