



الكلية الجامعية للعلوم التطبيقية
University College of Applied Sciences

رأس المجدد للإبداع Leader of Innovation

University College of Applied Sciences
Department of Engineering
Data Science and Artificial Intelligence Program

Intro. To Database Management System

COMP 2313

Database final project report

Hostel Management System

20/12/2022

Group members:

Name	ID	Section
Hebatallah AbuHarb	220210448	201
Shahd El-thalathini	220210527	201

Abstract

Hostel management system is a database application that aims to computerize the current procedures of allocating hostel rooms and simplify the process of communication between students and hostel managers.

Introduction

Student hostels are the place of residence for students who live in areas far from their universities. Usually, in order for a student to join a specific hostel, he fills out an application in which he applies to join the hostel. And as long as he lives there, there will be communication between him and the managers of this hostel. So, our goal was to computerize this process by implementing a web-enabled database with a richer feature set that makes it easier for students and hostel staff.

Scope of the Project

Right now, the website is only developed for student purposes. In the future, it can be expanded where Admin and Hostel Manager can also have a feature to login and allot students to their respective rooms.

Objective

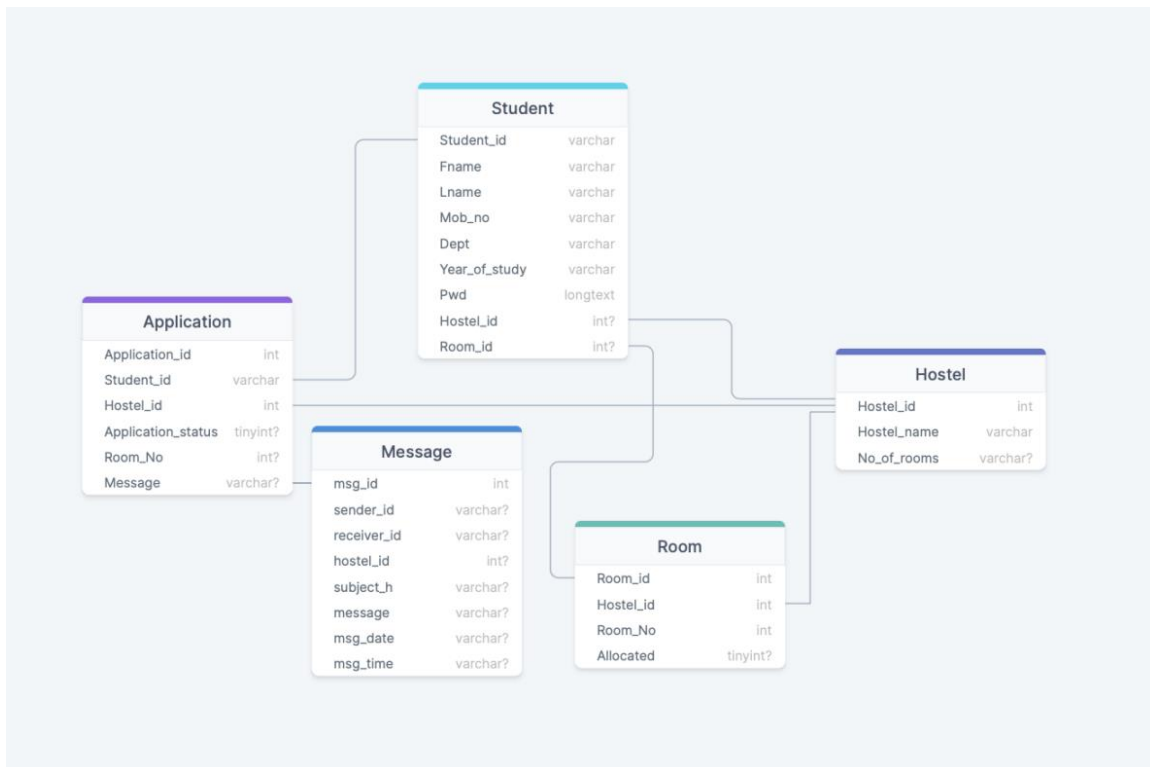
- To deal with Hostel Management System in an easy and efficient manner.
- Create a strong and private database that allows for any connection in a secret way, to prevent any outside or inside attacks.

Database Structure

The database consists of five tables. student, hostel, room, application, and message.

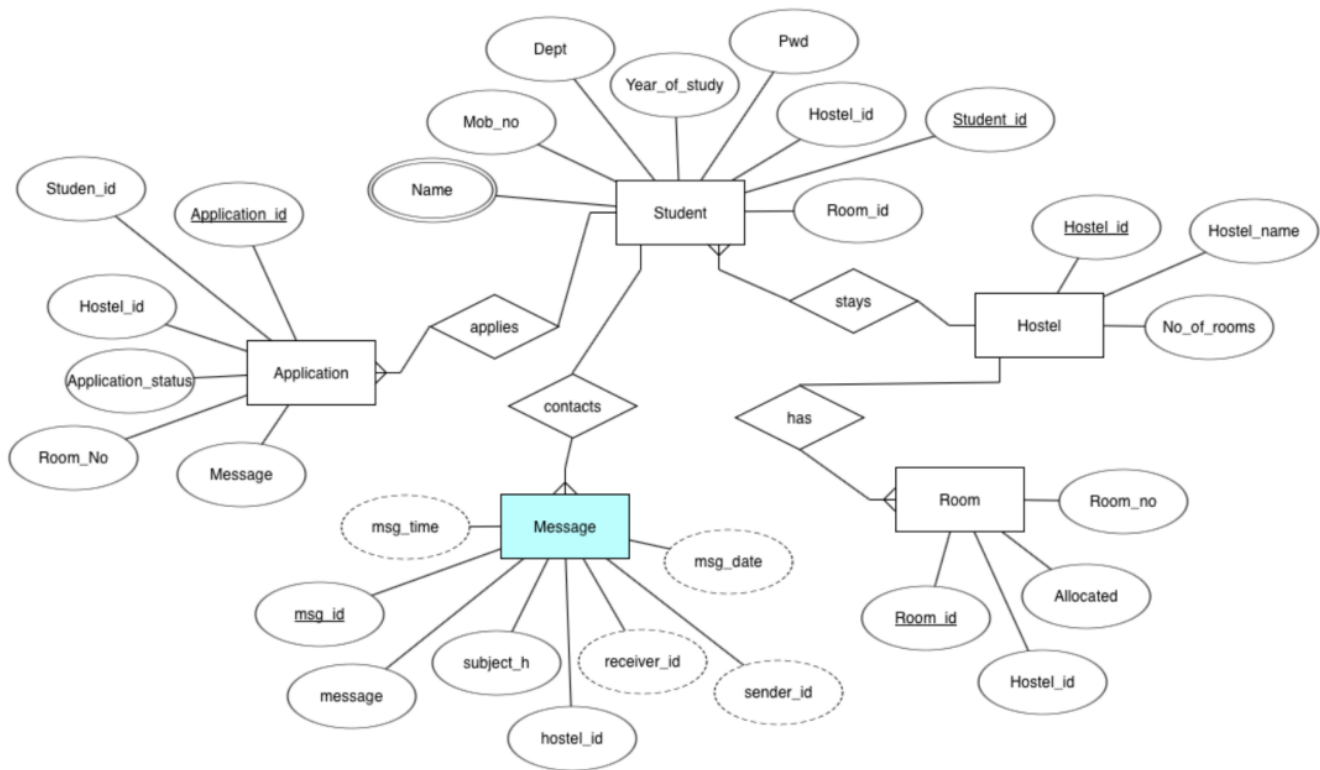
DATABASE TABLE	DESCRIPTION
Student	Contains all of the student's information.
Hostel	Contains information about the hostel.
Room	Contains information about each room in the hostel.
Application	Contains the data that the student entered when he fills in the application.
message	Contains information about the sender and the content of the message itself.

conceptual Schema



The primary key of **Student** table is **Student id** , and it's a foreign key in **Application** table. But the primary key of **Application** table is **Application id** . Moving to **Hostel** table, **Hostel id** is the primary key, when it's a foreign key in **Application**, **Student**, and **Room** tables. **Room** table has the **Room id** attribute as the primary key, while it's a foreign key in **Student** table. Finally, the primary key of **Message** table is **msg id**, and it's a foreign key in **Application** table.

ER Diagram



Relations between entities are described as the following. Many students stay at one hostel, so it's a one_to_many relation between student and hostel entities.

One hostel contains many rooms, and many rooms belong to one hostel, which means it's an one_to_many relation.

Student can send several messages, so it's one_to_many.

Finally, if the student has been in the hostel, then he went out for a while, and after that, he wants to join the hostel again, so he applies another time, so one student can have multiple applications, which means it's one_to_many relation.

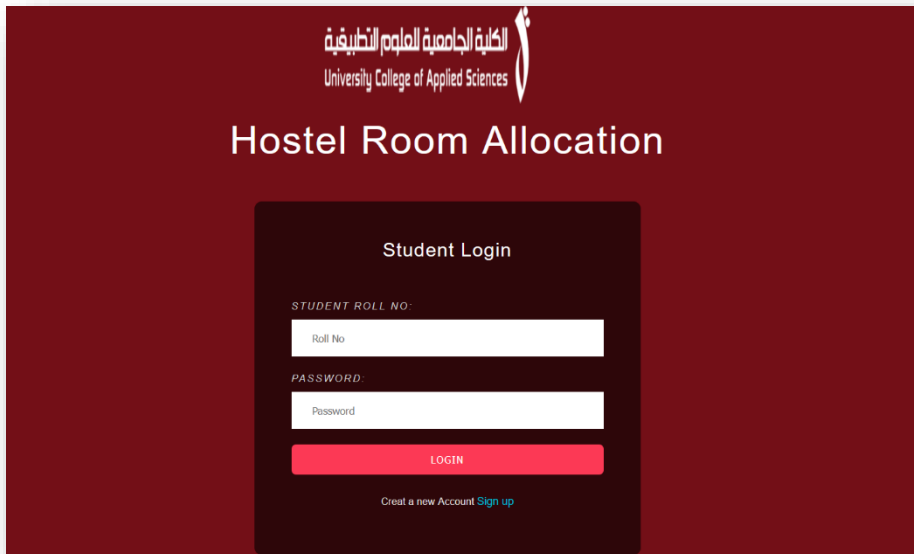
Functionality

The database and web application allow technicians to complete the following functions:

- Enter the data about a new student.
- Modify an existing student's data.
- Apply for the hostel using the request form.
- Contact the hostel managers and faculty.

The website is password-protected and requires a login. There is also session control that automatically logs out the person if the session is idle for too long.

Login Page



The screenshot shows a web application interface for 'Hostel Room Allocation' at the 'University College of Applied Sciences'. The page has a dark red background. At the top, there is a logo with Arabic text 'الكلية الجامعية العلوم التطبيقية' and English text 'University College of Applied Sciences'. Below the logo, the title 'Hostel Room Allocation' is displayed in white. In the center, there is a dark gray login box titled 'Student Login'. Inside this box, there are two input fields: 'STUDENT ROLL NO:' with a placeholder 'Roll No' and 'PASSWORD:' with a placeholder 'Password'. Below these fields is a red 'LOGIN' button. At the bottom of the login box, there is a link that says 'Create a new Account Sign up'.

This is what a **Student** sees when they first open the website. Students can also click on sign up or login with his/her credential.

To this page, Student Database is connected. When a student tries to login with his/her credential, the backend fetches the Student_id and Pwd from the Student Table, and checks if the credentials are correct or incorrect.

Sign Up Page

Hostel Room Allocation System

Sign Up Here

STUDENT ROLL NO

Roll No

FIRST NAME

First Name

LAST NAME

Last Name

MOBILE NO

Mobile No

DEPARTMENT

Department

YEAR OF STUDY

Year of Study

PASSWORD:

Password

CONFIRM PASSWORD:

Confirm Password


SIGN UP

Already a member? [Login](#)

When you enter all the details and click on the signup button, Student Table stores the details.

Student_id should be unique as it's a Primary key.

There are columns like Hostel_id and Room_id which hold no data. These were included as they can be used in the future by Hostel Admin to allot students their respective rooms.

Student		
	Student_id	varchar(255)
	Fname	varchar(255)
	Lname	varchar(255)
	Mop_no	varchar(255)
	Dept	varchar(255)
	Year_of_Study	varchar(255)
	Pwd	varchar(255)
	Hostal_id	varchar(255)
	Room_id	varchar(255)


Application form Page

Application Form

Gaurav Reddy	Hello Please Allot Me To MH2 Block. I Have Scored GPA : 9.5
19BCE7137	
MH2	

Click To Apply	

The data we enter in text fields is then saved to the Application-table. The table has Application_id as its primary key and it is auto-incremented.

Application		
	Application_id	integer
	Student_id	varchar(255)
	Hostel_id	integer(255)
	Application_status	varchar(255)
	Room_no	integer(255)
	Message	varchar(255)


There are a few foreign keys like Student_id and Hostel_id which are fetched from other tables.

In the future, if there's an Admin Page, he/she can approve the Application by changing the Application_status.

Contact Page

Contact Us

Hostel Name	Message...
Gaurav Reddy	
19BCE7137	
Subject	Submit

Message		
	msg_id	integer
	sender_id	varchar(255)
	reciver_id	varchar(255)
	Hostel_id	integer(255)
	subject_h	varchar(255)
	Message	varchar(255)
	msg_date	date(255)
	msg_time	varchar(255)

By contact page, students can contact the faculty if he/she has any issue with the rooms, bathrooms, lights, and other hostel-related issues.

Whatever the values entered will be added to the database Message-table.

Future of the Database

The database is currently functioning as a web application with an Oracle back-end and can begin to be used. However, the ideal situation would be for the database to be a college-supported production application.

It is anticipated that the following tasks will need to be accomplished in order to achieve the goals stated above:

1. Gather feedback (both direct and observational) from users.
2. Develop information architecture for needed forms, web pages, and database tables to support the purpose.
3. Code web pages using ASP.NET to interact with database tables.
4. Expanded the database where Admin and Hostel Manager can also have a feature to login and allot students to their respective rooms.

Summary

The project was a learning experience for both of us and allowed us to improve upon our SQL. We developed a usable database for a hostel and are proud of the results of our efforts. We look forward to seeing the database being put into actual use later.

References

- Fundamentals of Database Systems 7th Edition.
- Apps we used for drawings:
 - <https://app.dbdesigner.net/designer>
 - <https://creately.com>

Appendix

Source Code

```
CREATE TABLE `Application` (  
  `Application_id` int(100) NOT NULL,  
  `Student_id` varchar(255) NOT NULL,  
  `Hostel_id` int(10) NOT NULL,  
  `Application_status` tinyint(1) DEFAULT NULL,  
  `Room_No` int(10) DEFAULT NULL,  
  `Message` varchar(255) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `Hostel` (  
  `Hostel_id` int(10) NOT NULL,  
  `Hostel_name` varchar(255) NOT NULL,  
  `No_of_rooms` varchar(255) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
INSERT INTO `Hostel` (`Hostel_id`, `Hostel_name`, `No_of_rooms`) VALUES  
(1, 'MH1', '400'),  
(2, 'MH2', '400'),  
(3, 'MH3', '400'),  
(4, 'LH1', '400');  
  
CREATE TABLE `Message` (  
  `msg_id` int(10) NOT NULL,  
  `sender_id` varchar(255) DEFAULT NULL,  
  `receiver_id` varchar(255) DEFAULT NULL,  
  `hostel_id` int(10) DEFAULT NULL,  
  `subject_h` varchar(255) DEFAULT NULL,  
  `message` varchar(255) DEFAULT NULL,  
  `msg_date` varchar(255) DEFAULT NULL,  
  `msg_time` varchar(255) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `Room` (  
  `Room_id` int(10) NOT NULL,  
  `Hostel_id` int(10) NOT NULL,  
  `Room_No` int(10) NOT NULL,  
  `Allocated` tinyint(1) DEFAULT '0'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `Student` (  
  `Student_id` varchar(255) NOT NULL,  
  `Fname` varchar(255) NOT NULL,  
  `Lname` varchar(255) NOT NULL,  
  `Mob_no` varchar(255) NOT NULL,  
  `Dept` varchar(255) NOT NULL,  
  `Year_of_study` varchar(255) NOT NULL,  
  `Pwd` longtext NOT NULL,  
  `Hostel_id` int(10) DEFAULT NULL,  
  `Room_id` int(10) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```

ALTER TABLE `Application`
  ADD PRIMARY KEY (`Application_id`),
  ADD KEY `Student_id` (`Student_id`),
  ADD KEY `Hostel_id` (`Hostel_id`);

ALTER TABLE `Hostel`
  ADD PRIMARY KEY (`Hostel_id`);

ALTER TABLE `Message`
  ADD PRIMARY KEY (`msg_id`),
  ADD KEY `hostel_id` (`hostel_id`);

ALTER TABLE `Room`
  ADD PRIMARY KEY (`Room_id`),
  ADD KEY `Hostel_id` (`Hostel_id`);

ALTER TABLE `Student`
  ADD PRIMARY KEY (`Student_id`),
  ADD KEY `Hostel_id` (`Hostel_id`),
  ADD KEY `Room_id` (`Room_id`);

ALTER TABLE `Application`
  MODIFY `Application_id` int(100) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

ALTER TABLE `Hostel`
  MODIFY `Hostel_id` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

ALTER TABLE `Message`
  MODIFY `msg_id` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

ALTER TABLE `Room`
  MODIFY `Room_id` int(10) NOT NULL AUTO_INCREMENT;

ALTER TABLE `Application`
  ADD CONSTRAINT `Application_ibfk_1` FOREIGN KEY (`Student_id`) REFERENCES `Student` (`student_id`),
  ADD CONSTRAINT `Application_ibfk_2` FOREIGN KEY (`Hostel_id`) REFERENCES `Hostel` (`hostel_id`);

ALTER TABLE `Message`
  ADD CONSTRAINT `Message_ibfk_1` FOREIGN KEY (`hostel_id`) REFERENCES `Hostel` (`hostel_id`);

ALTER TABLE `Room`
  ADD CONSTRAINT `Room_ibfk_1` FOREIGN KEY (`Hostel_id`) REFERENCES `Hostel` (`hostel_id`);

ALTER TABLE `Student`
  ADD CONSTRAINT `Student_ibfk_1` FOREIGN KEY (`Hostel_id`) REFERENCES `Hostel` (`hostel_id`),
  ADD CONSTRAINT `Student_ibfk_2` FOREIGN KEY (`Room_id`) REFERENCES `Room` (`room_id`);

```