

**CIS)16-1 – Principles of Programming**

**CIS096-1 – Principles of Programming and Data Structures 2021-2022**

**Assignment 2 –****Hotel Online Customer Booking and Management System**

**GROUP / INDIVIDUAL REPORT**

**INDIVIDUAL WORK**

**Student ID: XXXXX Full Name XXXXXX**

Table of Contents

Introduction / Overview……………………………………………………………………… 3

Task Description………………………………………………………………………………... 3

Project Plan / Schedule….

Tasks:

Requirements Analysis

Overview of Functional, Technical (Non-Functional) and Usability Requirements

Design:

* UML Diagrams
* Use Case Diagrams
  + Use Case Specifications/Description
  + Use Case Scenario
* Activity Diagrams
* Class Diagram
* Database Design
* Entity Relationship Model
  + ERM Diagram
  + List of Entities
* Physical Database Design (including Data Dictionary)
  + Skeleton Tables
  + Data Dictionary
* User Interface Design

Implementation

Testing

List of Group Members – Roles and Responsibilities (if Group Work)

Discussion / Reflection / Critical Analysis

Conclusion

References

Appendix

* Complete code with comments and class / filenames.

CIS016-1 – Principles of Programming 2021-2022  
CIS096-1 – Principles of Programming and Data Structures 2021-2022  
Assignment 2 – Hotel Online Customer Booking and Management System

University ID 2147440 Full name Santosh Tamang

Introduction/Overview

The hotel Luton, situated in the town of Luton, provides three types of rooms i.e., single, double, and twin. The hotel has a bar and a restaurant where the guests can go for refreshments. The guests can also request room service. The bill will be added at the end when the guest is checking out. The guest can request any external service not provided by the user. This cost is also added at the checkout. There will be two types of customers i.e., individual, and corporate customers. The difference between these two users is that the individual customer should pay after the checkout, whereas the corporate customers are sent an invoice at the end of the month after the checkout.

Currently for booking, checking booking status and for cancelling the customer have to visit the hotel or send an email or call the hotel staff for the actions. The hotel staff stores all the details of the customer in hard copy format. It was hard for the customers to come to the hotel just to book and cancel a room they wanted. It was hard for the hotel staff to search the data of the specific customer from all the data stored in the hard copy.

To make the booking, cancelling, and viewing the booking much easier for the customer. For making the users’ working environment, and staff of the hotel, easier by making all data into electronic form. The given assignment is about creating a GUI desktop application for the Luton Hotel situated in the town of Luton. The desktop application is the Hotel Online Customer Booking and Management System. It allows different users like customers, corporate organizations, receptionists, and bar/restaurant staff. The application allows the users (customers, cooperative organizations) to log in and book a room through an online medium, cancelling a booking and checking the status of the booking. It allows the receptionist to assign a room to the booking according to the customer’s preference. The application also allows the receptionist to see all the expenses of the guest staying in a particular room.

The application to book, cancel and check the booking status from an online medium for the users was made using java. The application made calculates the total expenses of a guest staying in a particular room. The database to store the related data was created from MySQL with the required data dictionary and queries.

Task Description

The assignment asks for a desktop application with a database to be created for the Luton Hotel in the Town of Luton. The users are customers, corporate organizations, receptionists, and bar/restaurant staff. There should be a different login page for the customers and the hotel staff.

The receptionist after logging in, can see the booking and assign the room according to the room type (single, twin and double) requested by the customer if available for the date requested. The application should calculate the total expenses of the guest staying in a particular room, for the receptionist.

There are two types of customers that books the room in the hotel. The customers are individual (non-corporate) and corporate.

The individual who has not registered must register (providing their name, address, phone, email and credit card information) to use the online facilities. The registered customer should be able to view the check-in and check-out date and the room requested. The customers are allowed to view, change and cancel the booking made. Booking can be cancelled until the guest checks-in with valid reasons.

To book from companies, an account needs to be set up before making a corporate booking and billed at the end of each month since the last invoice was sent to the company. For each corporate, a discount is discussed when an account is set up

and is reviewed annually. Booking can be cancelled until the guest checks-in with valid reasons.

The guest can check-in any time after the booking is guaranteed. The check-in of the guest makes the booking active. The booking period ends after the guest check-out.

Project Plan/Schedule

|  |  |  |
| --- | --- | --- |
| Week No. | Tasks | Priority |
| 1 | Research in the related field | MUST |
| 2 | Requirement Analysis and GUI Prototype Design | MUST |
| 3 | Database Design | MUST |
| 4 | GUI Implementation | MUST |
| 5 | Database Implementation and Testing the Application | MUST |
| 6 | Finalizing Product | MUST |
| 7 | Submit Group Report, Project Code, and Individual Reflective Report, Video Recording | MUST |
| 8 | Project Presentation | MUST |

Overview of Functional, Technical (Non-Functional Requirements) and Usability Requirements

Functional Requirement

Individual Customer

1. A customer must register.
2. A customer must log in.
3. A customer should register with valid Credit Card Information for the guaranteed booking, if not registered booking is not guaranteed.
4. A customer should be able to select the room type of their preferences.
5. The customer should be able to view the booking status.
6. Customers can cancel the booking.
7. A customer should be able to make multiple bookings using the same id.
8. The bill is generated during the check-out process and the guest can pay the amount in any medium.

Corporate Customer

1. A customer must register.
2. A customer must log in.
3. A customer should register with valid Credit Card Information for the guaranteed booking, if not registered booking is not guaranteed.
4. A customer should be able to select the room type of their preferences.
5. The customer should be able to view the booking status.
6. Customers can cancel the booking.
7. A customer should be able to make multiple bookings using the same id.
8. The bill is generated during the check-out process. The invoices generated are sent to the company at the end of the month from the last invoices created.

Staff

1. A staff must login.
2. A staff should be able to view all the customers with the booking they made.
3. A staff should be able assign room according to the user preferences.
4. A staff should be able view room status.
5. A staff should be able mark the guest check-in and check-out.

Non-Functional Requirements

1. The system should respond as fast as possible.
2. The system must be platform independent.
3. The system must be online 24/7.
4. Future amendments must be allowed in the system.

Usability Requirements

1. The system should be user-friendly.
2. There should be validation to enter data and the forms should be short and easy to fill.
3. Enough information about the booking and lodging should be provided to meet users’ needs.
4. The navigation in the application should be easy the user should not need a guide to do the basics.

Design

* Uses Case Diagrams(s)

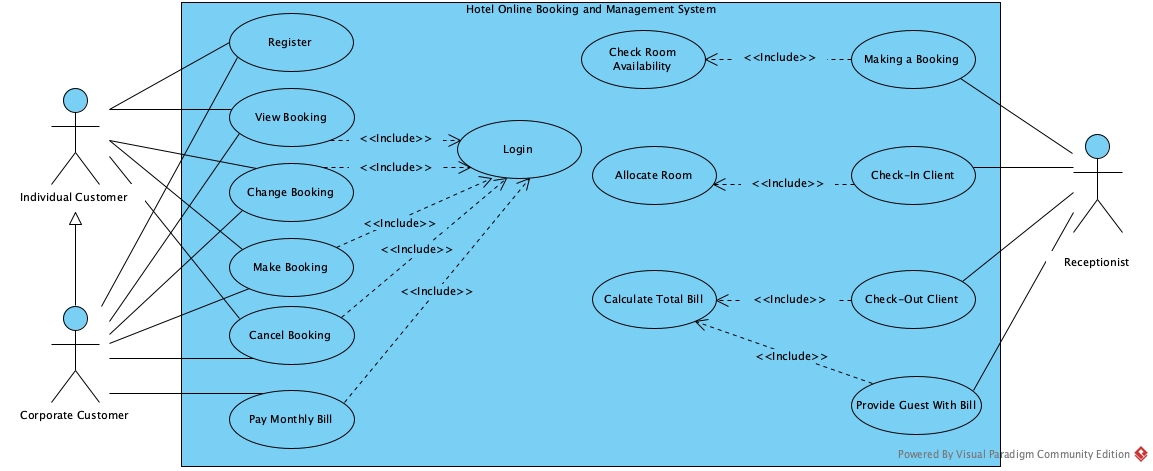


Fig 1: Hotel Online Customer Booking and Management System - Use Case Diagram (Sea Level)

Use Case Specifications/Description

Allocate Accommodation

A specific room is allocated to each guest.

Change Booking

An unguaranteed booking may become guaranteed on provision of credit card details. Other changes (for example to the composition of the party or to the dates booked) may be made on the customer’s request.

Calculate Bill Total

The total bill for a room comprises the cost of the room plus any items charged to it (for example, from the minibar, hotel bar or restaurant and any additional services provided) plus the appropriate rate of VAT.

Cancel Booking

A customer may cancel a booking any time before checking in. The booked accommodation is de-allocated.

Charge Item to Room

A guest may use items from the minibar in the room, make external telephone calls, make purchases from the hotel bar or restaurant or order additional services such as laundry. These items are charged to the guest’s room. An external service ordered by the guest may also be charged to their room (although some services, such as taxis, will be paid for directly by the guest).

Check In Guest

When a party (one or more guests) arrives at the hotel each guest is checked in and the booking becomes registered. Details of all the guests are confirmed, or if they have not been given at the time of booking they are recorded. The method of payment (by which guest(s) or whether to be charged to a corporate account) is confirmed. Accommodation is allocated if not already done at the time of booking and special requests are recorded.

Check Out Guest

Before the guest leaves the hotel, they need to check out and the booking becomes completed. The receptionist provides them with the total room bill, which an individual customer must pay. If the bill is to be charged to a corporate customer the guest is asked to check and sign the bill, it is then added to the appropriate corporate account.

Make a Booking

When a customer contacts the receptionist with a booking request the receptionist checks the room occupancy for the required period and if the customer's needs can be met, makes a booking.

A guaranteed booking is only made for corporate customers or for individual customers who supply credit card details.

Accommodation may be allocated at this stage; alternatively, where it is a group booking (for example, for 6 twin-bedded rooms) the allocation of each guest in the party to a specific room may be done on checking in. Either way, occupancy records are updated to avoid double booking.

Order External Service

Services not offered by the hotel itself can be ordered, once their availability has been checked. The guest’s name, room number and details of the service required are taken when the service is ordered.

Provide Guest with Current Bill

On request, a guest may be provided with the current total of their bill.

* Activity Diagram(s)
* Class Diagram(s)

This COULD be a screen shot from BlueJ. This is not ideal but a compromise.

* Database Design
  + Entity Relationship Model (ERM)
  + ERM Diagram
  + List of Entities

An Entity Relationship Model consists of a Entity Relationship Diagram (ERD) and a list of Entities with their attributes, Primary Key and Foreign Keys.

Example:

**Booking** (BookingID, CustomerID\*, RoomiD\*, ManagerID\*, DateBooked, ArrivalDate, DepartureDate, BookingStatus, Paid)

Primary and Foreign Keys and other attributes for each entity should be included in this list.

You are STRONGLY ADVISED to use VISUAL PARADIGM CE to create your ERM. Visual Paradigm Community Edition software is a free download. You should not use Visio or equivalent software.

Your ER Diagram should include all identified entities and the relationships between them. Relationship should include consideration of both cardinality and optionality, and a textual description of each relationship should be included.

* Physical Database Design (including Data Dictionary)

At this point you will have decided what database platform (RDMS) you will use – ie. Oracle, MySQL, SQLite.

This includes:

* + Skeleton Tables

List of tables with their attributes, Primary Key and Foreign Keys.

Example:

**Booking** (BookingID, CustomerID\*, RoomiD\*, ManagerID\*, DateBooked, ArrivalDate, DepartureDate, BookingStatus, Paid)

* + Data Dictionary (for each table)

Example:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Customer | | | | | | | |
| Description: Customer details | | | | | | | |
| Field Name | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| custid *(Primary)* | int (11)  unsigned | 11 | PK | No |  |  | Autoincremented Uniquely identifies every customer |
| firstname | varchar (30) | 30 |  | No |  |  | First name of customer |
| lastname | varchar (30) | 30 |  | No |  |  | Last name of customer |
| email | varchar (100) | 100 |  | No |  | Must be email format containing an @ and a ‘.’  Regex expression used | Email of customer |
| password | varchar (30) | 30 |  | No |  |  | Customer password |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Customer | | | | | | | |
| Description: Customer details | | | | | | | |
| Field Name | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| custid *(Primary)* | int (11)  unsigned | 11 | PK | No |  |  | Autoincremented Uniquely identifies every customer |
| firstname | varchar (30) | 30 |  | No |  |  | First name of customer |
| lastname | varchar (30) | 30 |  | No |  |  | Last name of customer |
| email | varchar (100) | 100 |  | No |  | Must be email format containing an @ and a ‘.’  Regex expression used | Email of customer |
| password | varchar (30) | 30 |  | No |  |  | Customer password |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Customer | | | | | | | |
| Description: Customer details | | | | | | | |
| Field Name | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| custid *(Primary)* | int (11)  unsigned | 11 | PK | No |  |  | Autoincremented Uniquely identifies every customer |
| firstname | varchar (30) | 30 |  | No |  |  | First name of customer |
| lastname | varchar (30) | 30 |  | No |  |  | Last name of customer |
| email | varchar (100) | 100 |  | No |  | Must be email format containing an @ and a ‘.’  Regex expression used | Email of customer |
| password | varchar (30) | 30 |  | No |  |  | Customer password |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Customer | | | | | | | |
| Description: Customer details | | | | | | | |
| Field Name | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| custid *(Primary)* | int (11)  unsigned | 11 | PK | No |  |  | Autoincremented Uniquely identifies every customer |
| firstname | varchar (30) | 30 |  | No |  |  | First name of customer |
| lastname | varchar (30) | 30 |  | No |  |  | Last name of customer |
| email | varchar (100) | 100 |  | No |  | Must be email format containing an @ and a ‘.’  Regex expression used | Email of customer |
| password | varchar (30) | 30 |  | No |  |  | Customer password |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Customer | | | | | | | |
| Description: Customer details | | | | | | | |
| Field Name | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| custid *(Primary)* | int (11)  unsigned | 11 | PK | No |  |  | Autoincremented Uniquely identifies every customer |
| firstname | varchar (30) | 30 |  | No |  |  | First name of customer |
| lastname | varchar (30) | 30 |  | No |  |  | Last name of customer |
| email | varchar (100) | 100 |  | No |  | Must be email format containing an @ and a ‘.’  Regex expression used | Email of customer |
| password | varchar (30) | 30 |  | No |  |  | Customer password |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Column** | **Null** |
| PRIMARY | BTREE | Yes | custid | No |

For each table, for each attribute show the datatype, length, and other properties. You may want to include any other information such as data entry constraints.

Take care with the datatypes as these vary from database system to database system.

Read the assignment brief for more details.

* User Interface Design

Sketches of your GUI interfaces

You should use Balsamiq Wireframes (see under Announcements) or an online prototyping tool (for example, <https://marvelapp.com/>)

Interactive prototypes could be created using the Balsimiq Mockups 3 / Wireframe Prototyping tool (<https://balsamiq.com/>) or an suitable online prototyping tool

Implementation

You should discuss YOUR application. How was the program created?

You should describe the development stages.

Explain how you manged the implementation. How did you ensure all group members actively contributed to writing code? How did you distribute the workload? Was it successful?

What Integrated Development Environment (IDE) did you use? Justify your choice of IDE, Java Framework (if used), packages / libraries.

Did you embed SQL in your client application or did you implement a Client-Server (Sockets Programming architecture). Justify you chosen method, and discuss the advantages and disadvantages of both.

What problems did you encounter during the coding, and explain how you overcame these?

Include snippets of code here – that either caused problems or show original, novel approaches.

What rDBMS did you use? Justify your choice of database system. Did you use the University’s MySQL database? How did you overcome problems associated with not being able to access this from outside University?

Did you use WAMP / XAMPP / MAMP or an equivalents stack on your own PC?

Detailed discussion of your experience coding is required here.

Did you incorporate any novel approaches to coding and meeting the Assignment requirements?

Testing

Include evidence of detailed and robust testing.

There are various ways to document this…

Test Plan / Test Log (tabular format) with test number, date, input, expected output/actual output and screen shot of result.

OR

Documentation of each test – one after the other

You need to include:

* Test No
* Test Date
* Purpose of test
* Input data or action
* Expected result
* Actual result
* Action if the above are not the same
* Screenshot of output

Group Member Contribution List

|  |  |  |
| --- | --- | --- |
| Student ID | Student Name | Role / Contribution |
|  |  |  |
|  |  |  |
|  |  |  |

Discussion / Refection / Critical Analysis

* What went well and what went wrong?
* What would you have done differently?
* How would you improve your web application in the future?

This part is very important. You must discuss the project, group work or individual work, and time management skills.

Basically, you are evaluating what you have done. Did you achieve all the required specification of the Assignment brief? If not, why not? Discuss the problems you encountered and how you overcame them.

What stopped you achieving what you had hoped to achieve and to the standard you know you are capable of achieving? Remote learning, lack of access to BREO, COVID, lack of suitable devices remotely, Internet connection issues etc.

Did you give yourselves enough time to accomplish this work? Was it harder than you expected? What have you learned from tackling this Assignment?

If working as a group, did your group work well together? What did you learn about the dynamics of group work?

Did this work enhance your understanding of the Java programming language? Did it motivate you to learn more Java, and other programming languages?

If you had to undertake this or something similar again would you tackle it differently?

What improvements or enhancements would you recommend for a future iteration of this project?

Conclusion

What were you asked to do? Did you achieve it?

References

Use the Harvard Referencing System. Any reference must be cited in text -otherwise it should go in a Bibliography section.

Appendix

You MUST include:

Complete project code

Include file/class names

Make sure you code is fully commented and well presented with correct indentation and colour coding.

You could include:

Details of you database/table structure and content

Screen shots from PHPMyAdmin.

Screenshots of you database / table structure / table data (PHPMyAdmin).

Screen shots of your GUI interfaces.

Anything else that you might want to include that does not fit well into the main report body.