

**International School**

**Software Process & Quality Management**

**CMU-SE 433 SAIS**

**Database Design**

**Version 1.1**

**Date: June 25, 2025**

**Hotel Management System**

**Submitted by**

**Lanh, Do**

**Quyen, Truong Thi My**

**Hoang, Le Huy**

**Quoc, Pham Tan**

**Hung, Duong Van**

**Approved by**

**MSc Huy, Truong Dinh**

**Database Design Review Panel Representative:**

Name Signature Date

**Software Process & Quality Management - Mentor:**

Name Signature Date

**PROJECT INFORMATION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project acronym** | HMS | | | |
| **Project Title** | Hotel Management System | | | |
| **Start Date** | 12 Jun 2025 | | **End Date** | July 23, 2025 |
| **Lead Institution** | International School, Duy Tan University | | | |
| **Project Mentor** | MSc Huy, Truong Dinh | | | |
| **Scrum master / Project Leader & contact details** | Lanh, Do  Email: ddolanh0@gmail.com  Tel: 0936212369 | | | |
| **Partner Organization** | Duy Tan University | | | |
| **Project Web URL** |  | | | |
| **Team members** | Name | Email | | Tel |
| 28219129939 | Lanh, Do | ddolanh0@gmail.com | | 0936212369 |
| 28201100757 | Quyen, Truong Thi My | thanhmieu2112@gmail.com | | 0768585726 |
| 28219027592 | Hoang, Le Huy | lehuyhoang123789@gmail.com | | 0935563105 |
| 28212306085 | Quoc, Pham Tan | phamtanquoc1@gmail.com | | 0763739254 |
| 28210201891 | Hung, Duong Van | hd795033@gmail.com | | 0334231904 |

**DOCUMENT APPROVALS**

The following signatures are required for approval of this document.

|  |  |  |
| --- | --- | --- |
| Lanh, Do  Student ID: 28219129939  *Scrum Master* | Signature | Date |
| Quyen, Truong Thi My  Student ID: 28201100757  *Team Member* | Signature | Date |
| Hoang, Le Huy  Student ID: 28219027592  *Team Member* | Signature | Date |
| Quoc, Pham Tan  Student ID: 28212306085  *Team Member* | Signature | Date |
| Hung, Duong Van  Student ID: 28210201891  *Team Member* | Signature | Date |

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Comments** | **Author** |
| 1.0 | June 25, 2025 | Initial Release | Lanh,Do |

**Table of Contents**

1. Introduction ................................................................................................................7

1.1 Purpose .................................................................................................................7

1.2 Scope ....................................................................................................................7

1.3 Introduction about MongoDB ..............................................................................7

2. Database Diagram .......................................................................................................8

2.1 Table Overview ....................................................................................................8

2.2 Entity Relationship Diagram .........................................................................…...9

2.3 Table Relationship Diagram ...............................................................................10

3. Database Design for Sprint .......................................................................................10

3.1 Table Users…………………..............................................................................10

3.2 Table Rooms…………………............................................................................10

3.3 Table Staff……………………............................................................................11

3.4 Table Bookings……………….............................................................................11

3.5 Table Payments.....................................................................................................11

3.6 Table Services.........................................................................................................11

3.7 Table Service\_Requests……................................................................................12

3.8 Table Housekeeping..............................................................................................12

3.9 Table Financial\_Transactions...............................................................................12

4.Hardware and software Requirements..........................................................................13

**TABLE OF FIGURES**

Firure 1. Entity relationship diagram.………………………………………………..9

Firure 2. Relationship diagram………………………………………………………10

**TABLE OF TABLES**

Table 1. Overview…………………………………………………………………….8

Table 2: Users………………………………………………………………………..10  
  
Table 3: Rooms……………………………………………………………………....10

Table 4: Staff………………………………………………………………………....11

Table 5: Bookings……………………………………………………………………11

Table 6: Payments……………………………………………………………………11

Table 7: Services…………………………………………………….……………….11

Table 8: Service\_Requests……………………………….…………………………..12  
  
Table 9: Housekeeping……………………………….………………………………12  
  
Table 10: Financial\_Transactions……………………………….…………………....12  
.  
Table 11: Hardware and software Requirements………………………………..……13

1. Introduction

The Database Design 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 data constructs (e.g., tables...) of the target Database Management System.

1.1 Purpose

The purpose of the Database Design is to ensure that every database transaction meets or exceeds its performance requirements. This document takes into account data and transaction volume to produce a schema and environment that will meet necessary performance.

1.2 Scope

- 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 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.

1.3 Introduction about SQL Server

SQL Server, developed by Microsoft, is a relational database management system (RDBMS) that stores data in structured tables with rows and columns. Known for its robustness, performance, and scalability, SQL Server is widely used in enterprise environments on platforms like Windows and Linux, making it suitable for mission-critical applications.

Key features of SQL Server include scalability, high availability, transaction management, and backup and recovery. Data integrity is maintained through relationships and constraints using primary and foreign keys. Unlike NoSQL databases, SQL Server uses Structured Query Language (SQL) for querying, which is ideal for complex queries and data consistency.

For instance, a table for a Student entity can be defined as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **StudentId** | **Name** | **RollNo** | **Subjects** |
| 1 | Study\_tonight | 1 | C Language, C++, Core Java |

Relationships between tables are established through joins and foreign keys, promoting data normalization and reducing redundancy.

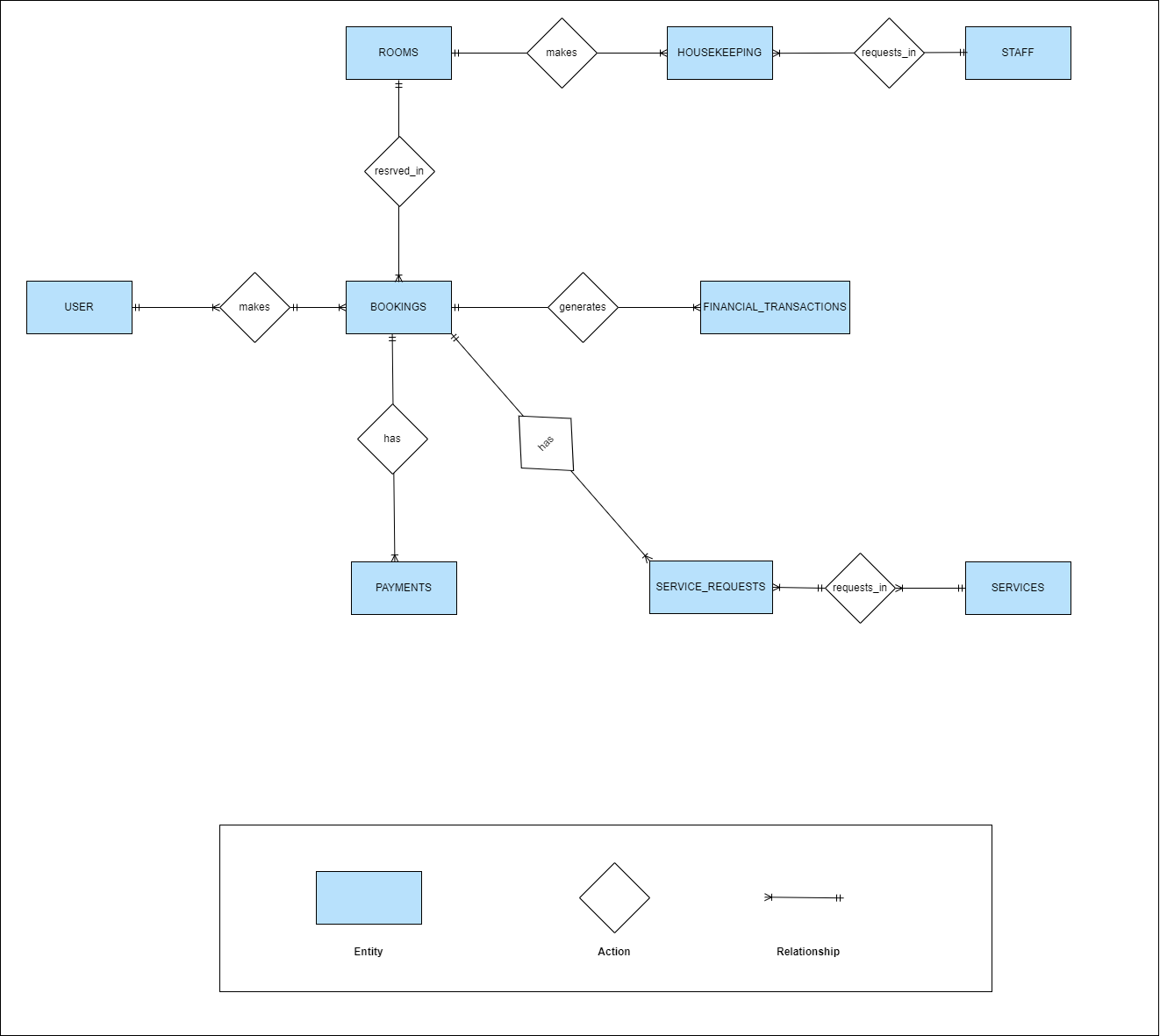
**2. Database Diagram**

2.1 Table Overview

Table 1. *Overview*

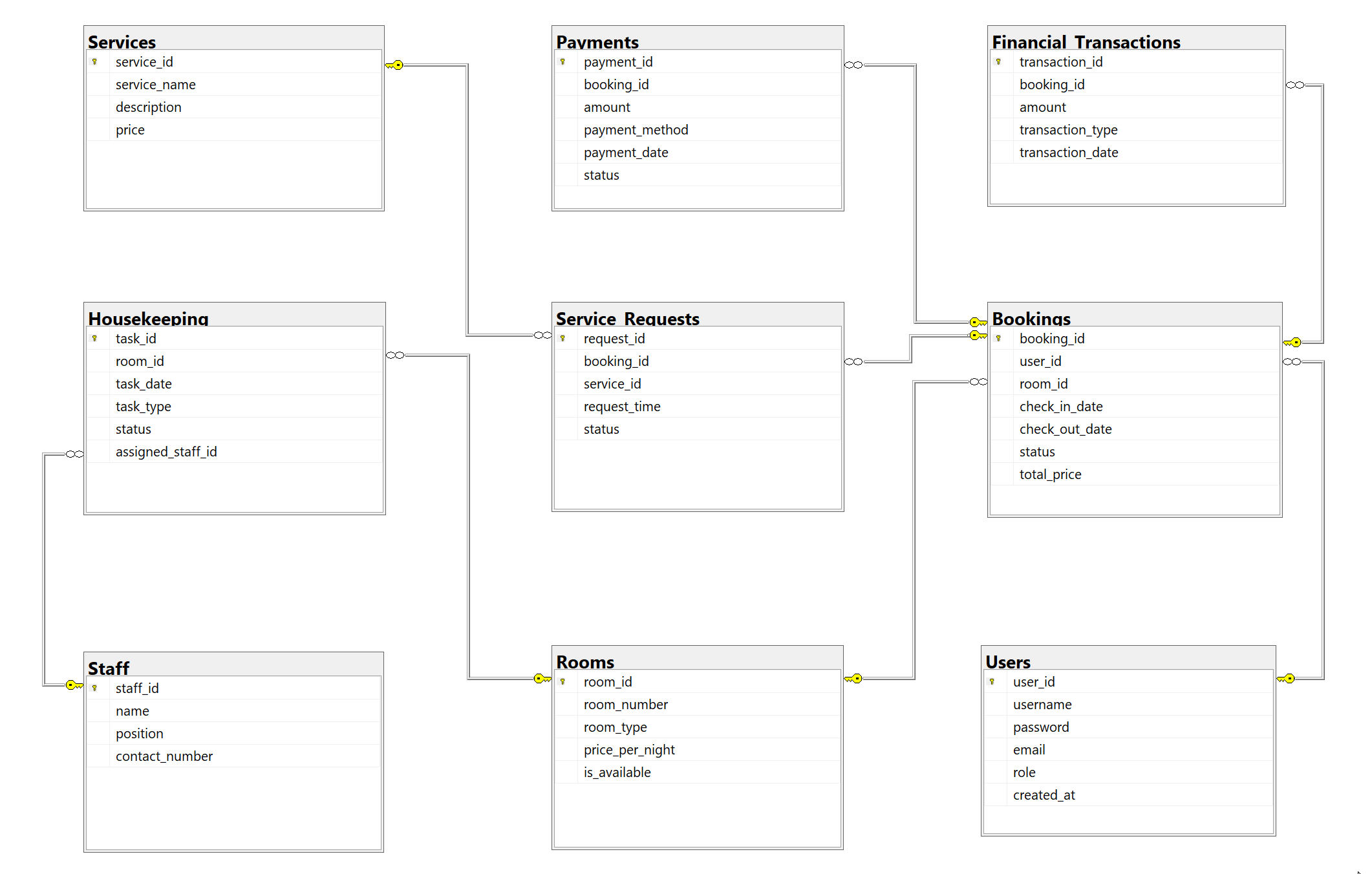
|  |  |
| --- | --- |
| **Table Name** | **Short Description** |
| **Users** | This table stores information about each user, including details such as username, password, and role. |
| **Rooms** | This table contains information about the rooms available, including room type, price, and availability. |
| **Staff** | This table holds details about the staff members, including their names, positions, and contact numbers. |
| **Bookings** | This table records bookings made by users, including check-in/check-out dates, room details, and status. |
| **Payments** | This table captures payment information related to bookings, including amount, payment method, and status. |
| **Services** | This table lists all available services provided by the property, including service names and prices. |
| **Service\_Requests** | This table records requests made by users for specific services, linking them to bookings and services. |
| **Housekeeping** | This table schedules housekeeping tasks for rooms, including assigned staff, task types, and statuses. |
| **Financial\_Transactions** | This table logs financial transactions related to bookings, such as payments, refunds, and transaction dates. |

2.2 Entity Relationship Diagram

****

**Figure 1.** Entity relationship diagram

2.3 Table Relationship Diagram



**Figure 2**. *Relationship diagram*

3. Database Design for Sprint

3.1 Table Users

Table 2: *Users*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Group** | | | | |
| **Id** | **Field** | **Type** | **Constrain** | **Description** |
| 1 | user\_id | INT | Primary Key | Unique identifier for each user |
| 2 | username | NVARCHAR(50) | NOT NULL | Username of the user |
| 3 | password | NVARCHAR(255) | NOT NULL | Hashed password for user authentication |
| 4 | email | NVARCHAR(100) | NOT NULL | Email address of the user |
| 5 | role | NVARCHAR(20) | Optional | Role of the user (e.g., admin, guest) |
| 6 | created\_at | DATETIME | Default: GETDATE() | Timestamp of when the user account was created |

3.2 Table Rooms

Table 3: *Rooms*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User** | | | | |
| **Id** | **Field** | **Type** | **Constrain** | **Description** |
| 1 | room\_id | INT | Primary Key | Unique identifier for each room |
| 2 | room\_number | NVARCHAR(10) | NOT NULL | Room number within the property |
| 3 | room\_type | NVARCHAR(50) | Optional | Type of room (e.g., single, double, suite) |
| 4 | price\_per\_night | DECIMAL(10, 2) | Optional | Cost per night for the room |
| 5 | is\_available | BIT | Default: 1 | Availability status of the room (1 = available, 0 = not) |

3.3 Table Staff

Table 4: *Staff*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Frame** | | | | |
| **Id** | **Field** | **Type** | **Constrain** | **Description** |
| 1 | staff\_id | INT | Primary Key | Unique identifier for each staff member |
| 2 | name | NVARCHAR(100) | NOT NULL | Full name of the staff member |
| 3 | position | NVARCHAR(50) | Optional | Job position/title of the staff member |
| 4 | contact\_number | NVARCHAR(15) | Optional | Contact number for the staff member |

3.4 Table Bookings

Table 5: *Bookings*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PrivateMessage** | | | | |
| **Id** | **Field** | **Type** | **Constrain** | **Description** |
| 1 | booking\_id | INT | Primary Key | Unique identifier for each booking |
| 2 | user\_id | INT | Foreign Key | ID of the user making the booking |
| 3 | room\_id | INT | Foreign Key | ID of the booked room |
| 4 | check\_in\_date | DATE | Optional | Date of check-in |
| 5 | check\_out\_date | DATE | Optional | Date of check-out |
| 6 | status | NVARCHAR(20) | Optional | Booking status (e.g., confirmed, canceled) |
| 7 | total\_price | DECIMAL(10, 2) | Optional | Total price for the booking |

3.5 Table Payments

Table 6: *Payments*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PublicMessage** | | | | |
| **Id** | **Field** | **Type** | **Constrain** | **Description** |
| 1 | payment\_id | INT | Primary Key | Unique identifier for each payment |
| 2 | booking\_id | INT | Foreign Key | ID of the associated booking |
| 3 | amount | DECIMAL(10, 2) | NOT NULL | Amount paid |
| 4 | payment\_method | NVARCHAR(50) | Optional | Method of payment (e.g., cash, credit card) |
| 5 | payment\_date | DATE | Optional | Date of payment |
| 6 | status | NVARCHAR(20) | Optional | Status of the payment (e.g., completed) |

3.6 Table Services

Table 7: *Services*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Event** | | | | |
| **Id** | **Field** | **Type** | **Constrain** | **Description** |
| 1 | service\_id | INT | Primary Key | Unique identifier for each service |
| 2 | service\_name | NVARCHAR(100) | Optional | Name of the service provided |
| 3 | description | TEXT | Optional | Description of the service |
| 4 | price | DECIMAL(10, 2) | Optional | Price for the service |

3.7 Table Service\_Requests

Table 8: *Service\_Requests*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course** | | | | |
| **Id** | **Field** | **Type** | **Constrain** | **Description** |
| 1 | request\_id | INT | Primary Key | Unique identifier for each service request |
| 2 | booking\_id | INT | Foreign Key | ID of the associated booking |
| 3 | service\_id | INT | Foreign Key | ID of the requested service |
| 4 | request\_time | DATETIME | Default: GETDATE() | Timestamp when the service was requested |
| 5 | status | NVARCHAR(20) | Optional | Status of the service request |

3.8 Table Housekeeping

Table 9: *Housekeeping*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CourseVocabulary** | | | | |
| **Id** | **Field** | **Type** | **Constrain** | **Description** |
| 1 | task\_id | INT | Primary Key | Unique identifier for each housekeeping task |
| 2 | room\_id | INT | Foreign Key | ID of the room for housekeeping |
| 3 | task\_date | DATE | Optional | Date when the housekeeping task is scheduled |
| 4 | task\_type | NVARCHAR(50) | Optional | Type of housekeeping task (e.g., cleaning) |
| 5 | status | NVARCHAR(20) | Optional | Status of the task |
| 6 | assigned\_staff\_id | INT | Foreign Key | ID of the staff member assigned to the task |

3.9 Table Financial\_Transactions

Table 10: *Financial\_Transactions*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CourseOfUsser** | | | | |
| **Id** | **Field** | **Type** | **Constrain** | **Description** |
| 1 | transaction\_id | INT | Primary Key | Unique identifier for each transaction |
| 2 | booking\_id | INT | Foreign Key | ID of the associated booking |
| 3 | amount | DECIMAL(10, 2) | NOT NULL | Transaction amount |
| 4 | transaction\_type | NVARCHAR(50) | Optional | Type of transaction (e.g., credit, refund) |
| 5 | transaction\_date | DATE | Optional | Date of the transaction |

4. Hardware and software Requirements

This section provides an overview of hardware and software requirements. Below are descriptions of the technological components of the HMS - Hotel Management System:

Table 11: *Hardware and software Requirements*

|  |  |
| --- | --- |
| **Attributes of Easy English WEBSITE** | |
| **Attributes** | **Descriptions** |
| **Database** | SQL Server |
| **Software** | JavaScript, Python , Reactjs, Nodejs, Polling, Redis, Redis, WebSocket, Data Training and Machine Learning |
| **Hardware** | Computer |
| **Library** | Hook, Redux, React hook form, Font awesome, Material UI, Express.js, TensorFlow.js, Socket.io, JWT |