Design

Document

Online Hotel Booking.

**Index**

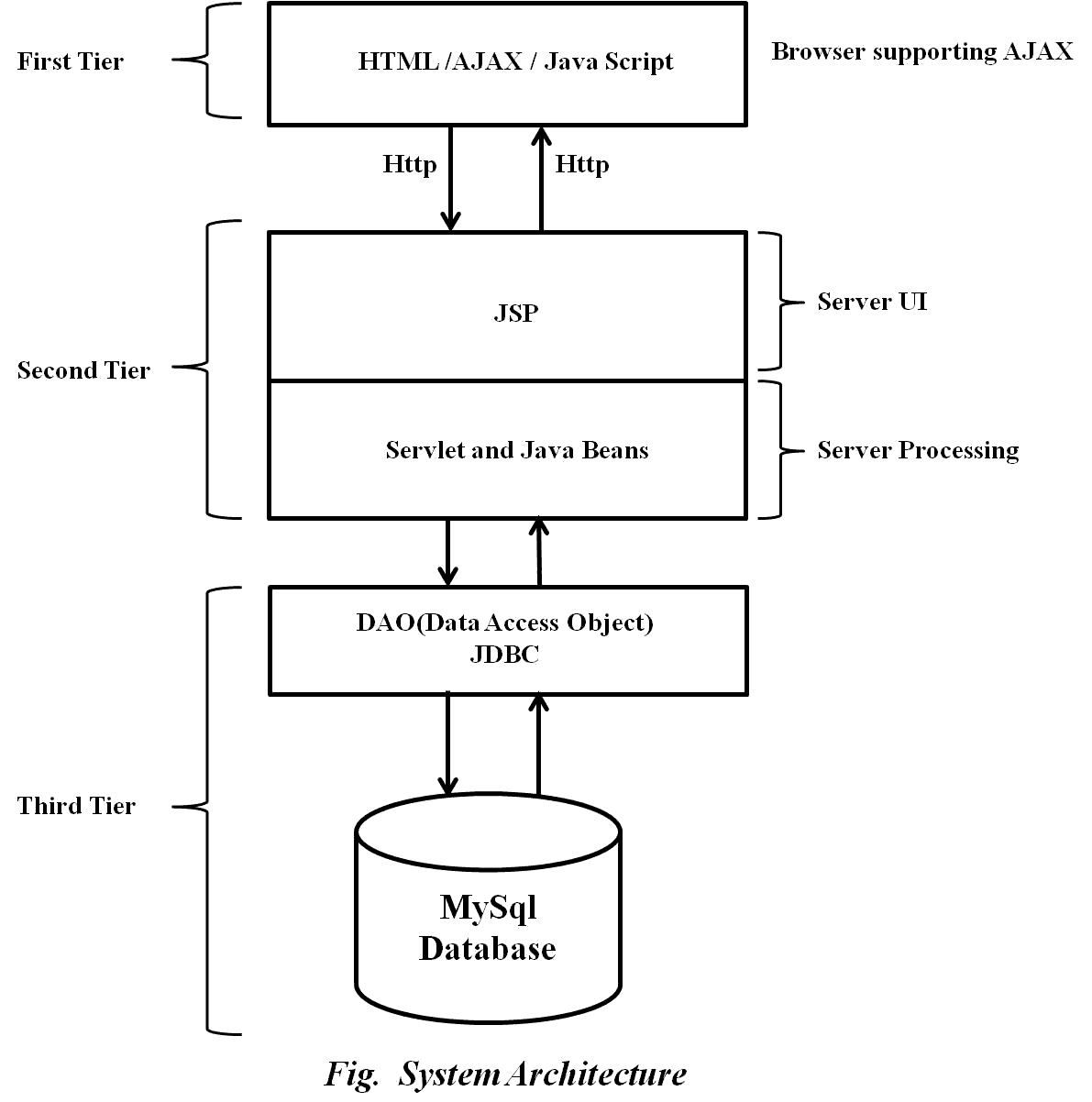
|  |  |  |
| --- | --- | --- |
| Sr No | Content | Page No |
| 1 | Introduction | 3 |
| 2 | Architectural Design | 3 |
| 3 | High Level Design | 6 |
|  | 3.1 E-R Diagram | 6 |
|  | 3.2 Page Navigation Diagram | 7 |
|  | 3.3 Data Flow Diagram | 9 |
|  | 3.4 Deployment Diagram | 12 |
| 4 | Low Level Design | 13 |
|  | 4.1 Database Design | 13 |
|  | 4.2 Stored Procedure | 16 |
|  | 4.3 Details Of Page Navigation | 17 |

* **Introduction:**

This document is meant for the description of the structure and the database which we are using in this project. This document gives brief description about Architecture of the system, E-R diagram of the system and the table descriptions, the page navigation diagrams and the detail description for the page navigation.

* **Architecture Design:**

Following diagram shows the details of the e-faming system architecture.



This System consist of three tiers as listed below,

* First tier
* Second tier
* Third tier

**First Tier:**

This tier is used for user interface and it is called as client tier. In this tier we are using AJAX because of it provides better interactivity, easier navigation, compact. The use of java script facilities us for the client side validation. That’s why in first tier we are using the java script. We are using HTML for the presentation purpose.

**Second Tier:**

Second Tier is comprises of two parts listed below,

* **Server UI**

In this part of second tier we are using JSP, because it provides better UI to system, as well as it provides the dynamically designing of pages.

* **Server Process**

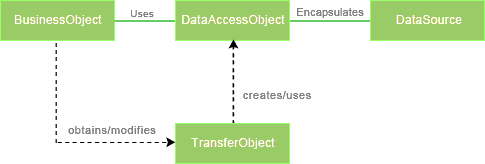
Servlet API is standard and freely available on the internet (like JSPs) servlets have the advantages like ease of development & platform independence (like Java) they can access all the J2SE and J2EE APIs can take the full advantage & capabilities of the Java programming language.

**Third Tier:**

Third tier consist of a Data Access Object (DAO) and the back end i.e. the database of e-farming system.

**Data Access Object (DAO):**

Data access object layer has proven good in separate business logic layer and persistent layer. The DAO design pattern completely hides the data access implementation from its clients.

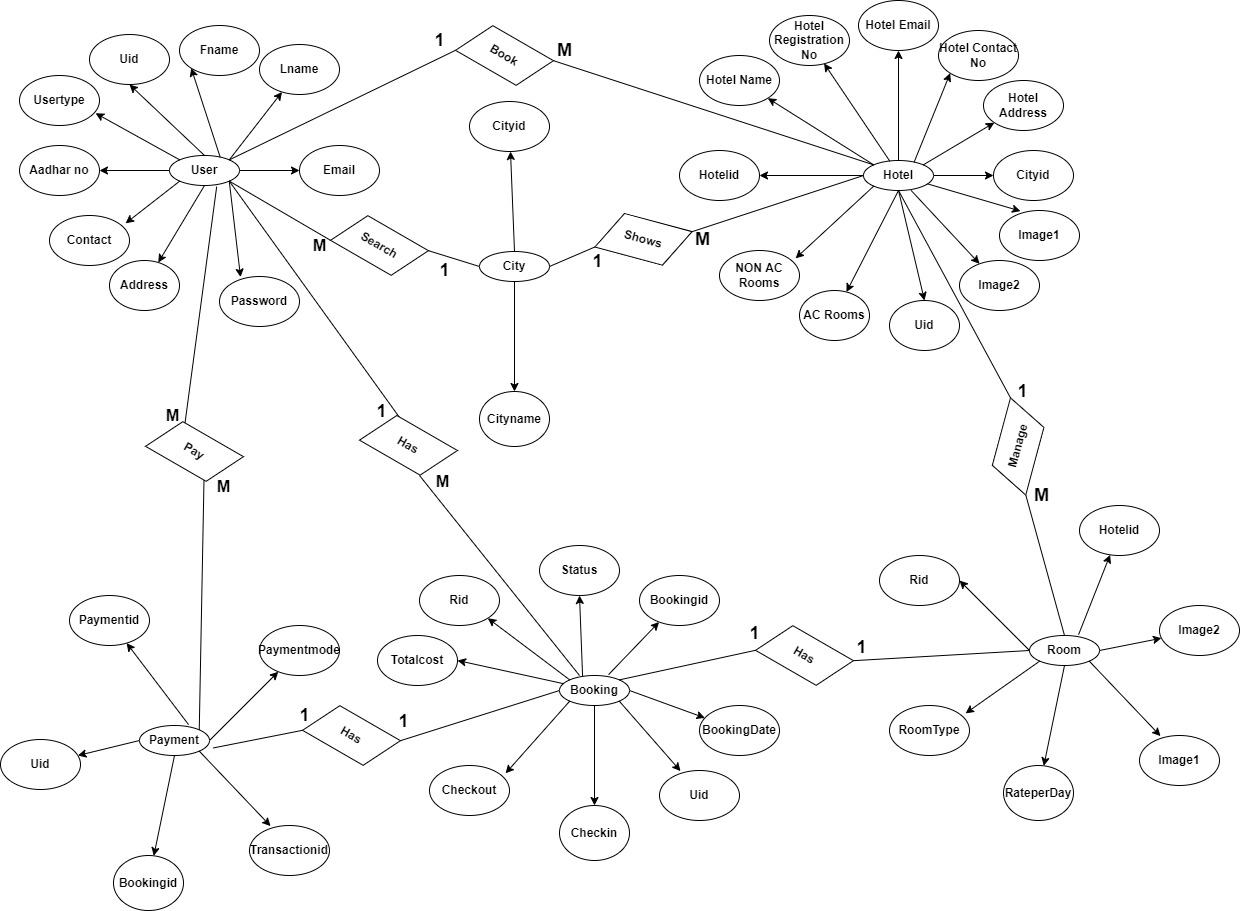


*Fig Data Access Object (DAO) Mechanism*

**Java Database Connectivity (JDBC):**

JDBC is used to provide database connectivity from java to database. Using Java database connectivity we can update/retrieve data to/from database with java programs. The main advantage of using JDBC is we can execute database queries by the program so that we can utilize the functionality provided by the database (with the queries). More over we can use triggers too. JDBC provides much other functionality (like the functions provided by CallableStatemtent class) to manage the data. Additionally, loading the driver will be different to different databases.

* **High Level Design:**
* **E-R Diagram:**



Above E-R Diagram shows that database of Online Hotel Booking system consist of following entities:

* **Low Level Design:**
* **Database Design:**

**1] Tbl\_User**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Uid | Integer | No | Primary key |  | User ID |
| Fname | Varchar(45) | Yes |  | Null | First Name |
| Lname | Varchar(45) | Yes |  | Null | Last Name |
| Email | Varchar(45) | Yes |  | Null | Email ID |
| Password | Varchar(15) | Yes |  | Null | Account Password |
| ContactNo | Varchar(10) | Yes |  | Null | Contact No. |
| AadharNo | Varchar(12) | Yes |  | Null | Aadhar No. |
| Address | Varchar(100) | Yes |  | Null | Permanent Address |
| UserType | Varchar(10) | Yes |  | Null | Customer/Service Provider |

**2] Tbl\_Hotel**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Hotelid | Integer | No | Primary key |  | Hotel id |
| HotelRegNo | Integer | Yes |  | Null | Hotel Registration No |
| Hname | Varchar(45) | Yes |  | Null | Hotel Name |
| Address | Varchar(45) | Yes |  | Null | Hotel Address |
| Email | Varchar(45) | Yes |  | Null | Email Address |
| ContactNo | Varchar(45) | Yes |  | Null | Contact Number |
| Image1 | Varchar(45) | Yes |  | Null | Hotel Image File name 1 |
| Image2 | Varchar(45) | Yes |  | Null | Hotel Image File name 2 |
| Ac\_Room | Integer | Yes |  | Null | Total ac rooms |
| Non\_Ac\_Room | Integer | Yes |  | Null | Total non ac rooms |
| Cityid | Integer | No | Foreign key |  | Reference to Cityid (Tbl\_City) |
| Uid | Integer | No | Foreign key |  | Reference to Uid(Tbl\_User) |

**3] Tbl\_City**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Cityid | Integer | No | Primary key |  | City ID |
| CityName | Varchar(20) | Yes |  | Null | City name |

**4] Tbl\_Room**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Rid | Integer | No | Primary key |  | Room ID |
| RoomType | Varchar(20) | Yes |  | Null | Room name |
| RateperDay | Double | Yes |  | Null | Room rate per day |
| Image1 | Varchar(20) | Yes |  | Null | Room image 1 |
| Image2 | Varchar(20) | Yes |  | Null | Room image 2 |
| Hotelid | Integer | No | Foreign key |  | Reference to Hotelid (Tbl\_Hotel) |

**5] Tbl\_Booking**

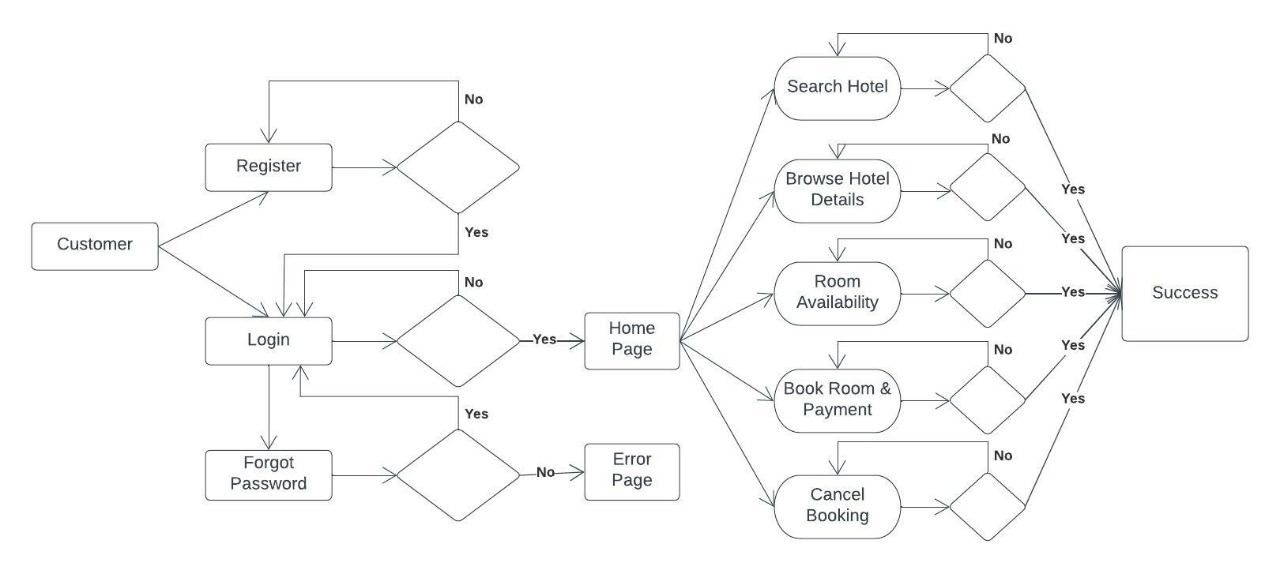
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Bookingid | Integer | No | Primary key |  | Booking ID |
| BookingDate | Date | Yes |  | Null | Booking Date |
| Rid | Integer | No | Foreign key |  | Hotel ID |
| Checkin | Date | Yes |  | Null | Check in Date |
| Checkout | Date | Yes |  |  | Check out Date |
| Uid | Int | No | Foreign Key |  | Uid Reference to Uid(Tbl\_User) |
| TotalCost | Double | Yes |  | Null | Total Cost of Booking |
| Status | Varchar(20) | Yes |  | Null | Booking Status |

**6] Tbl\_Payment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Paymentid | Integer | No | Primary key |  | Payment ID |
| PaymentMode | Varchar(20) | Yes |  | Null | Payment Mode |
| Bookingid | Integer | No | Foreign Key |  | Bookingid Reference to Bookingid(Tbl\_Booking) |
| Transactionid | Integer | Yes |  | Null | Transcation id |
| Uid | Integer | No | Foreign key |  | Reference to Uid(Tbl\_User) |

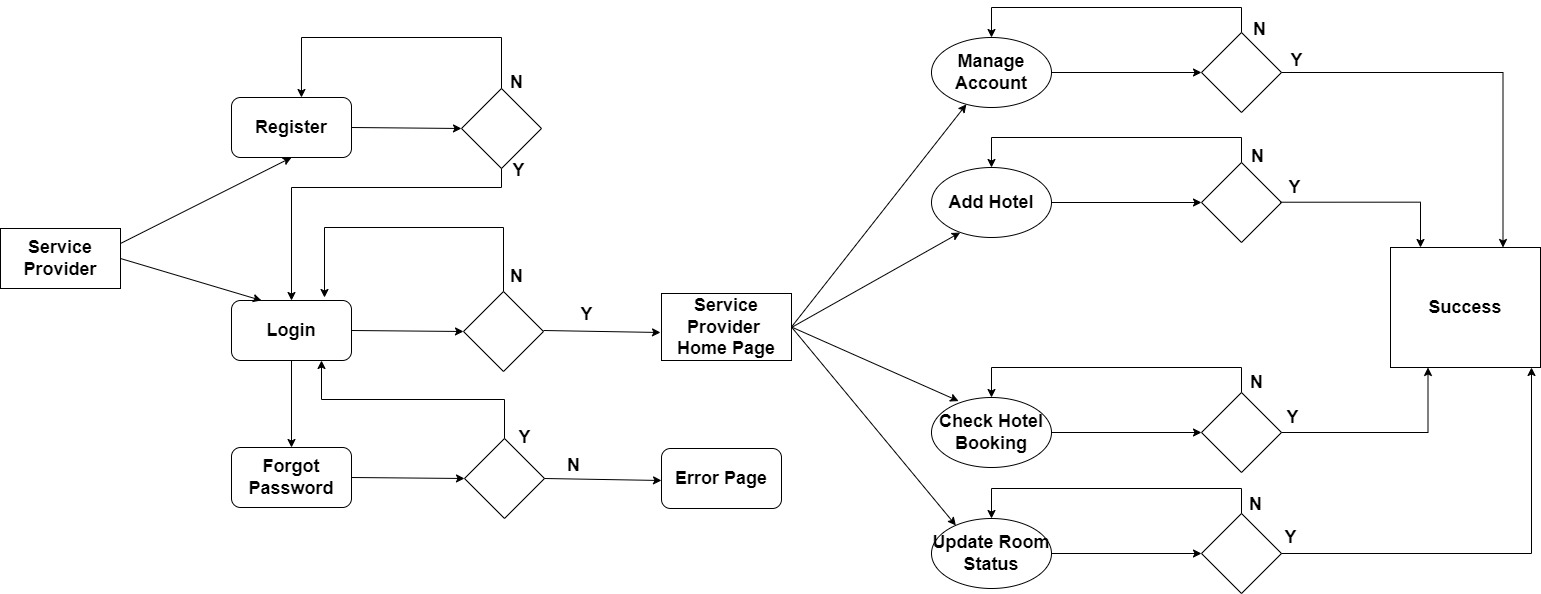
* **Page Navigation Diagram:**
* **Customer**

Following diagram explains the page navigation for the customer module:

****

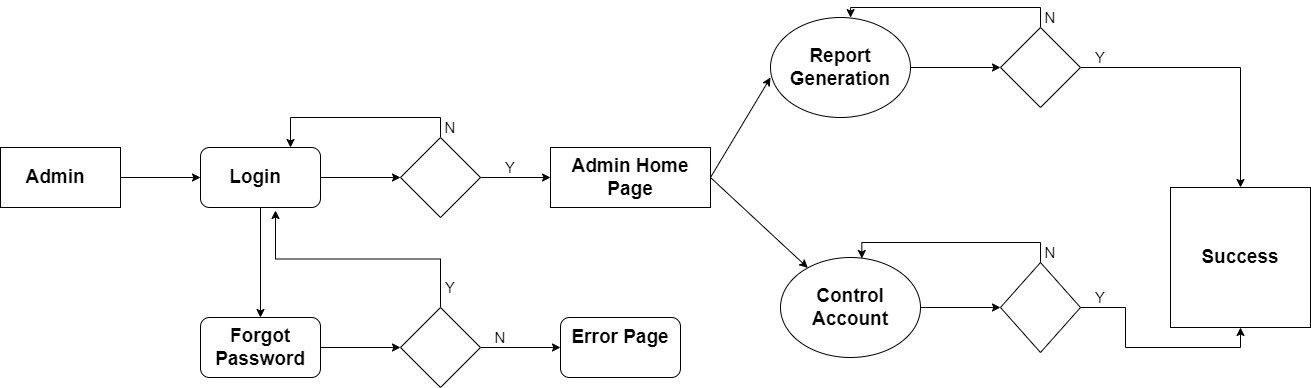
* **Service Provider**

Following diagram explains the page navigation for the wholesaler module:

****

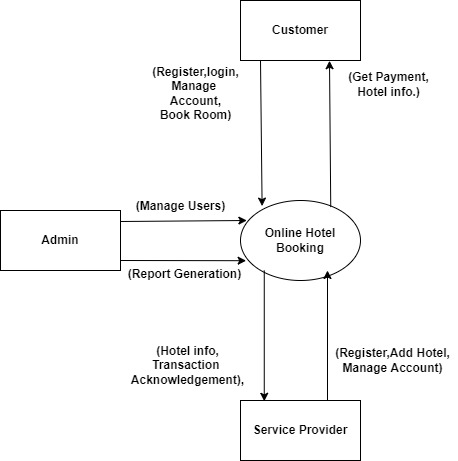
* **Admin**

Following diagram explains the page navigation for the Admin module:

****

**Data Flow Diagram:**

* **0-Level DFD:**



* In 0-Level DFD, there are three Entities:
* Customer
* Service Provider
* Admin