Software Requirement Specification (SRS) for

“PG Accommodation System”

Me tumse bohot jyada gussa hoon

# Introduction

### Purpose:

While living in a new place to build a new life seems to be an exciting prospect, there might be several obstacles along the way. The first one that comes to mind is finding a good and affordable place to stay. If you goes another place and you need to stay so how do you decide between paying guest accommodation and a flat on rent? So the Paying Guest Accommodation Project decided your accommodation in a new city according to you.

### 1.2 Scope:

In the Paying Guest Accommodation Project system, the user can register to get a login id and password. With login id and password, users can get logged in. After logging in, the user can post the paying guest post by adding details and pictures of the place. He/she can also view the interested users in his/her uploaded post. The uploaded post can be removed or deleted. Users can also see for the paying accommodation and after getting the desirable place he/she can select the place he/she is interested in. After selecting the desired place, the user will get the personal details of the owner, he/she can get in contact with the owner and go for further processing.

### 1.3 Definitions:

SRS- Software Requirement Specification

GUI- Graphical User Interface

**1.5 Overview:**

In paying guest accommodation project, you are bound to land up with two or more roommates, who might be in the mood for some fun and frolic, when all you need is sleep. in a rented house accommodation, you can binge-watch on Netflix all night or put on your nightshades at your own free will. Though it is the responsibility of a tenant to take care of daily maintenance of the rented property, you can get in touch with us for basic maintenance and get plumbing, electrical and house repair needs to be sorted easily

**EXISTING SYSTEM**

* Things going manually like going to PG rooms and checking the rooms ,facilties and buying the PG rooms.
* PG Management is also done by man by using man power so there is chance of demanding more money for rent.

**NEED FOR NEW SYSTEM**

* The new system is totally computerized system.
* A new system provides features like time efficiency to save cost and time of customer as customer can buy book online from anywhere .
* Customer can search the book according to specific category
* Provide facility to users to buy the book online.

**2.Overall Description**:

**2.1 Product Perspective**:

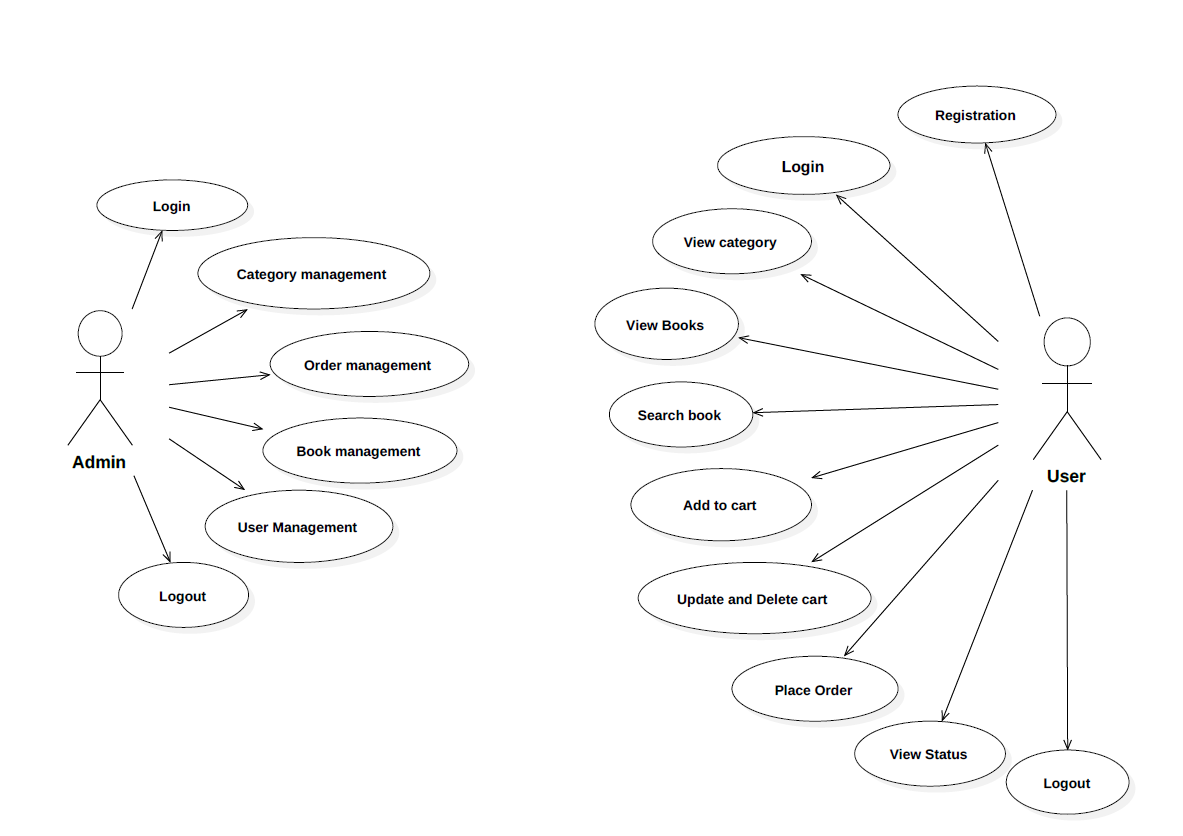
The online bookstore's main Actros are divided into two categories, one is the front user, one is the background user (Admin). Front-end users are mainly customers who will buy books from online bookstore. Front-end users can register, login, search the books ,add the book to shopping cart And place orders .The background user (Admin) will be able add, modify or delete the book details , can add the latest books. Also admin will be able to manage category i.e add , update ,delete category as per requirement .

**2.1 Product Functions**:

**UML Diagrams :**

**Use Case Diagrams** : A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor - Sender, Secondary- Actor Receiver.

**Use case diagram :**



**2.3 User Characeristics**:

User should be familiar with the terms like login,register etc.

**2.4 Principle Actors**:

**Actors:**

• Admin

• User

**3. Specific Requirements**:

3.1 **FUNCTIONAL SPECIFICATION**

**Actors:**

• Admin

• User

**Functional Requirements :**

**Admin Module :**

**1. Admin Login :**

Admin can login using verified username and password.

1. **User Management :**

Admin can manage all the details of user , will manage all the user request.

1. **Book Management :**

Background Administrators can control book information such as adding books, editing book information, and deleting book information.

**4. Category Management :** Admin can update the category ,delete category or add a new category as per requirements.

**5. Order Management :**

View orders by order status, view order details, can cancel orders before they are shipped, and can be shipped after buyer payment.

**User Module :**

**1. User Registration Login :**

Users can register as a member, the user can become a member after login, modify

personal information, modify the password and exit.

**2 .Category display :**

User can view books as per specific category.

**3 .Book display:**

User can Search the books by title or keyword, search for books , able to see the details of book.

**4 . Cart Management:**

User can add the books in the cart which user wants to buy , modify quantity of books , can be able to remove the book from the cart .

**5 .Order Management:**

User can order the book ,review the details of existing order and place the order , cancel the order .

**3.2 Non-Functional Requirements**:

Following Non-Functional Requirements will be there in the

insurance to the internet:

(i) Secure access to consumer’s confidential data.

(ii) 24X7 availability.

(iii) Better component design to get better performance at peak

time.

(iv) Flexible service based architecture will be highly desirable for

future extension .Non-Functional Requirements define system

properties and constraints.

Various other Non-Functional Requirements are:

 Security

 Reliability

 Maintainability

 Portability

 Extensibility

 Reusability

 Compatibility

 Resource Utilization

**3.3 Performance Requirements**:

In order to maintain an acceptable speed at maximum number of uploads allowed from a particular customer as any number of users can access to the system at any time. Also the connections to the servers will be based on the attributes of the user like his location and server will be working 24X7 times.

**3.4 Technical Issues**:

This system will work on client-server architecture. It will require an internet server Apache tomcat . The system should support some commonly used browser such as IE ,mozzila , firefox , chrome etc.

**HARDWARE REQUIREMENT**

Hardware requirements for insurance on internet

will be same for both parties which are as follows:

|  |  |
| --- | --- |
| **RAM** | 2 GB |
| **Hard disk** | 320 GB |
| **Processor** | Dual Core |

**Software Requirements**

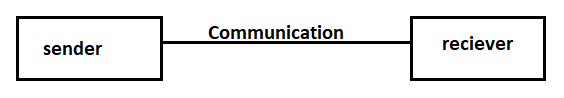
**Client side:**

|  |  |
| --- | --- |
| **Web Browser** | Google Chrome or any  compatible browser |
| **Operating System** | Windows or any equivalent OS |

**Server side:**

|  |  |
| --- | --- |
| **Web Server** | TOMCAT |
| **Server side Language** | Spring Boot (Java) |
| **Database Server** | MYSQL |
| **Web Browser** | Google Chrome or any  compatible browser |
| **Operating System** | Windows or any equivalent OS |

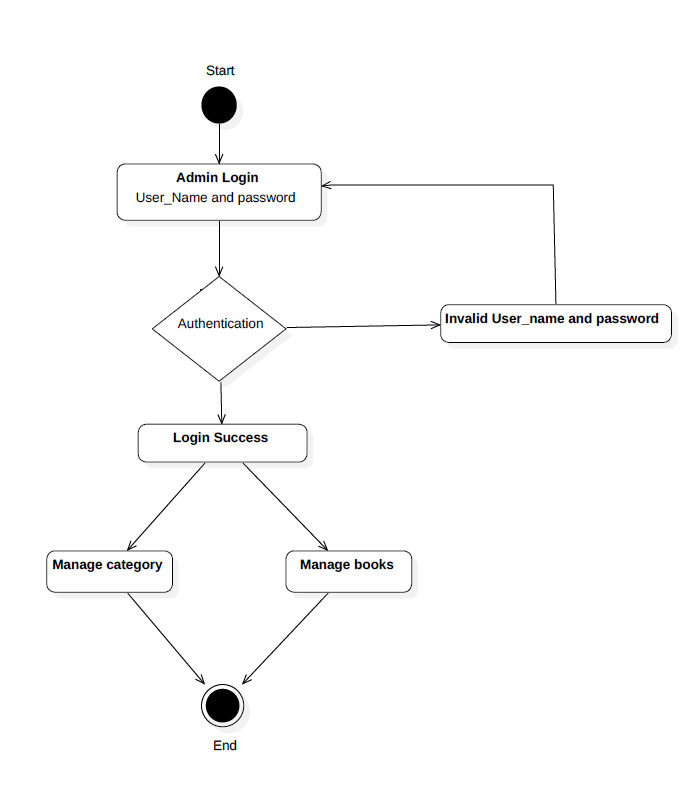
**Communication Interfaces**:

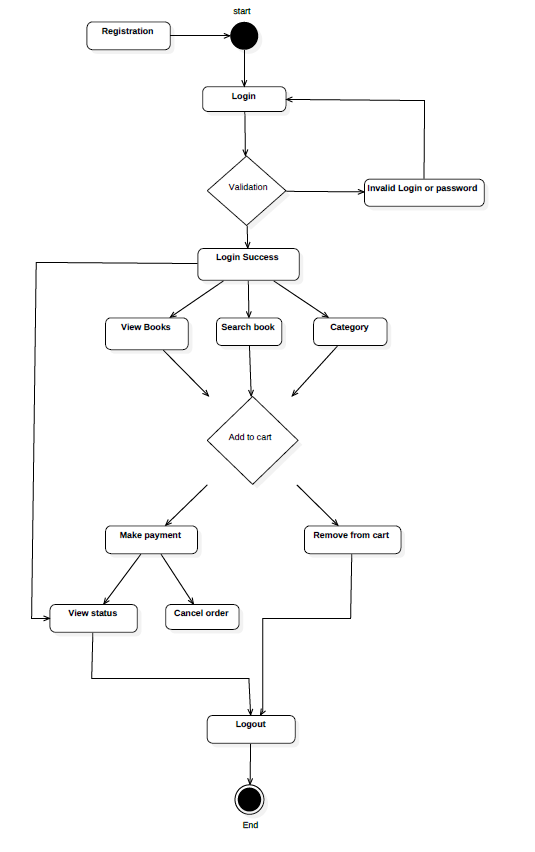
The two parties should be connected by LAN or WAN for the communication purpose.

**5.System Design Specification:**

**System Flow Chart**

**Admin :**

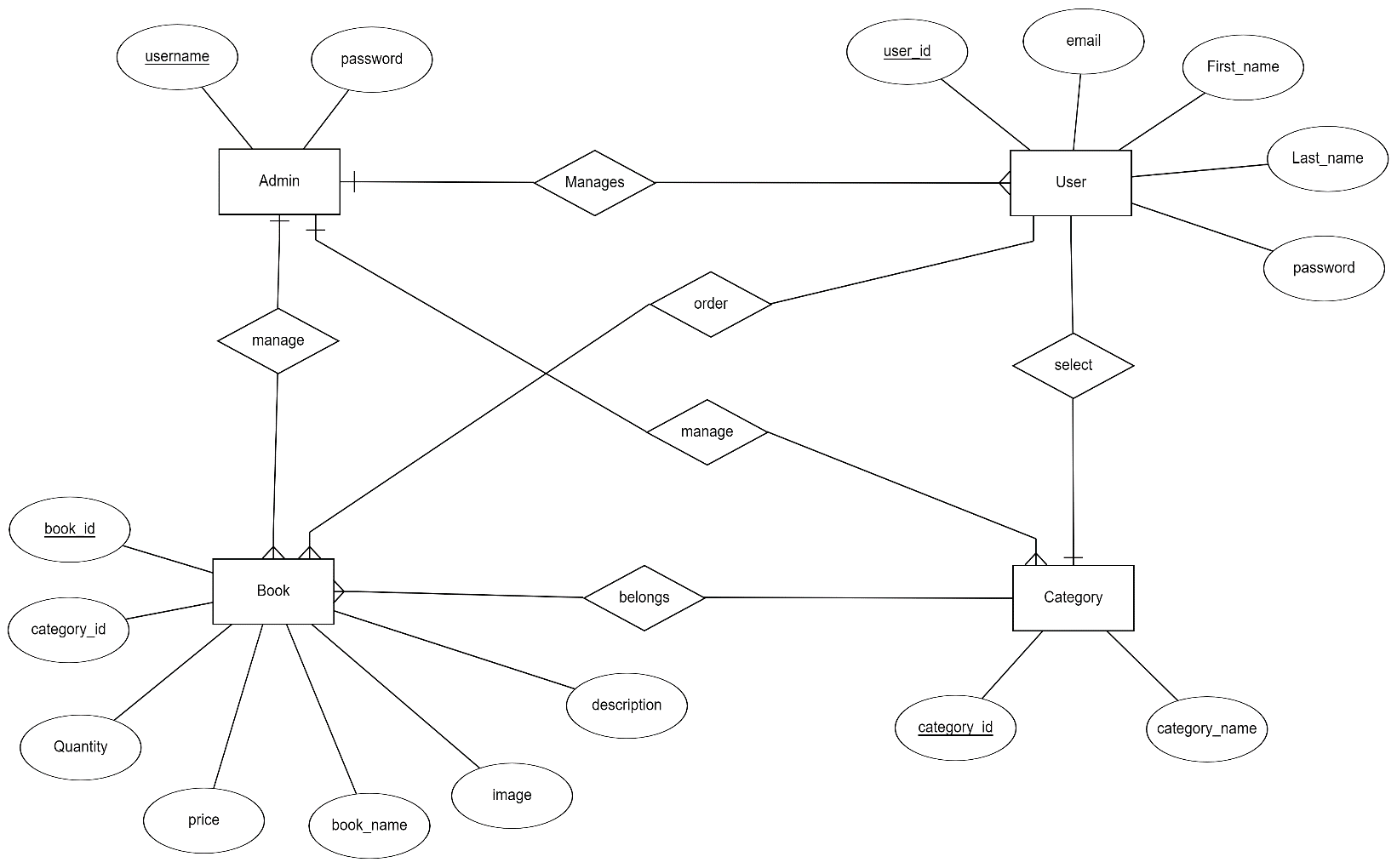


**User :**

**ER DIAGRAM**

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

* It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
* It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
* In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.



**DATABASE DESIGN**

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases.

Book table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| Book\_id | int | NO | PRI | NULL |  |
| Book\_description | Varchar(255) | YES |  | NULL |  |
| image | Blob | YES |  | NULL |  |
| Book\_name | Varchar(255) | YES |  | NULL |  |
| Book\_price | int | YES |  | NULL |  |
| Book\_quantity | Int | YES |  | NULL |  |
| Category\_id | int | YES | MUL | NULL |  |

Category table :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| Category\_id | int | NO | PRI | NULL |  |
| Category\_name | Varchar(255) | yes |  | NULL |  |

**User table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **NULL** | **Key** | **Default** | **Extra** |
| User\_id | bigint | NO | PRI | NULL |  |
| Email | Varchar(255) | YES |  | NULL |  |
| First\_name | Varchar(255) | YES |  | NULL |  |
| Last\_name | Varchar(255) | YES |  | NULL |  |
| Password | Varchar(255) | YES |  | NULL |  |
| role | Varchar(255) | YES | MUL | NULL |  |

Add to Cart :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| id | int | NO | pri | NULL |  |
| User\_id | BIGINT | NO | MUL | NULL |  |
| Book\_id | BIGINT | NO | MUL | NULL |  |
| quantity | int | YES |  | NULL |  |
| Added\_date | timestamp | YES |  | NULL |  |
| price | int | YES |  | NULL |  |

Checkout Cart:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| id | int | NO | pri | NULL |  |
| User\_id | BIGINT | NO | MUL | NULL |  |
| Book\_id | BIGINT | NO | MUL | NULL |  |
| quantity | int | YES |  | NULL |  |
| order\_date | timestamp | YES |  | NULL |  |
| price | int | YES |  | NULL |  |
| address | Varchar | NO |  | NULL |  |