**3. Design**

**3.1 Introduction:**

Design is the process of creating something on the basis of some perspectives. Generally design is done looking at the user perspective and needs. Design is different by looking at the things you are developing and requirements. Design is very necessary in any development. It is the 1st impression of any sort of program or application.

Benefits of designing:

* It sets the impression towards the customers or users.
* Sometimes a good design may provide a trust from user/customers.
* Its helps user to use the application with ease.

**3.2 Structural design**

Structural design is the method of investigating the strength, stability and rigidity. Generally we do it for looking the structural view of the system. Structural design also supports the architectural design.

Class diagram, Data flow diagram and flowchart are some examples of structural design.

For my project, I have made class diagram to find out the structure of my project.

1. **Class diagram:**

Class diagram is the static diagram which also represent the static view of the project. For my project I have made the class diagram to see the static view and how it will work. Class diagram also helps to analyze and design the static view of the application.

**Justification of my class diagram:**

I have made the class diagram by considering the future stability. After the register is done user can easily order the books with the categories they like. Also user can give the review of the books.

**Notation uses:**

*Aggregation notation:* This notation is used to configure the objects for making the complex type of object.

*Composition notation:* This notation is used for the composite objects.

*Association notation:* This type of notation is used to inter-relate the class.

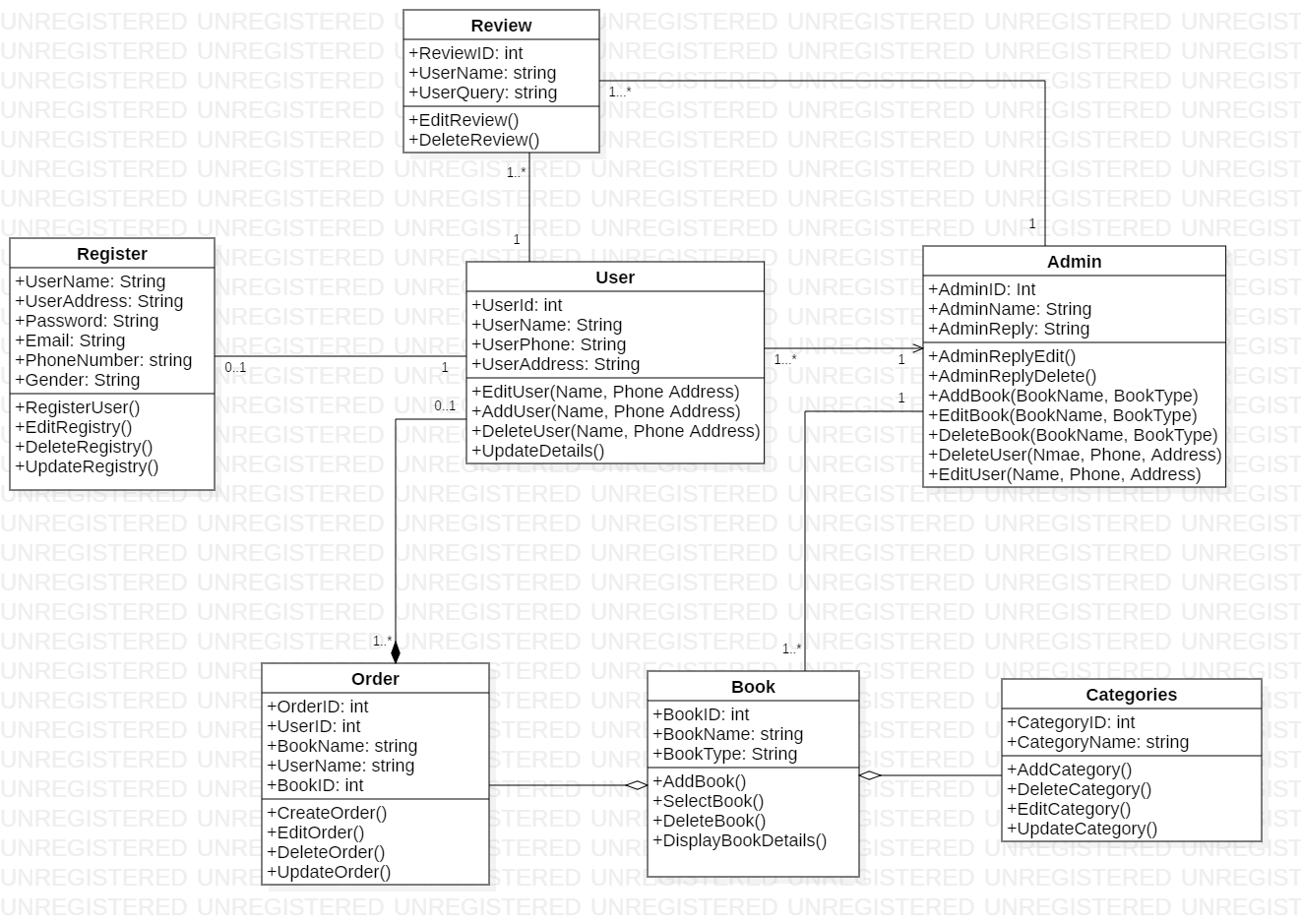


Fig: Final Class diagram

**Explanation:**

My class diagram consists of 6 classes which is very important for developing the project. First of all, use will register themselves with the name, address and contact number. After the registering they can order the books with different categories of their interest. After registering themselves, they can also review the books, after they read it.

1. **DFD:**

DFD is the type of structural design. DFD is the known as the Data Flow Diagram. This diagram shows the flow of the data or process. Data Flow Diagram also provide the information about the outputs.

For my project too, I have made the data flow diagram to analyze the flow of my information. This diagram will give the enough knowledge to others about how the data will flow.

**Justification:**

Data flow diagrams are used to graphically represent the flow of data in a business information system***. (data-flow-diagram, 2019)***

DFD is very essential to my project because it shows the path of the program and how it will be working after the program is made.

**Notation used:**

*Process Notation:* This notation helps to transfer the incoming data flow into the out going data flow.

*Data stores Notation:* This notation are the repositories of the system.

*Data flow notation:* This notation helps to find the flow of data in what direction.

*External entity notations:* This notation gives a clue about the objects outside of the system.

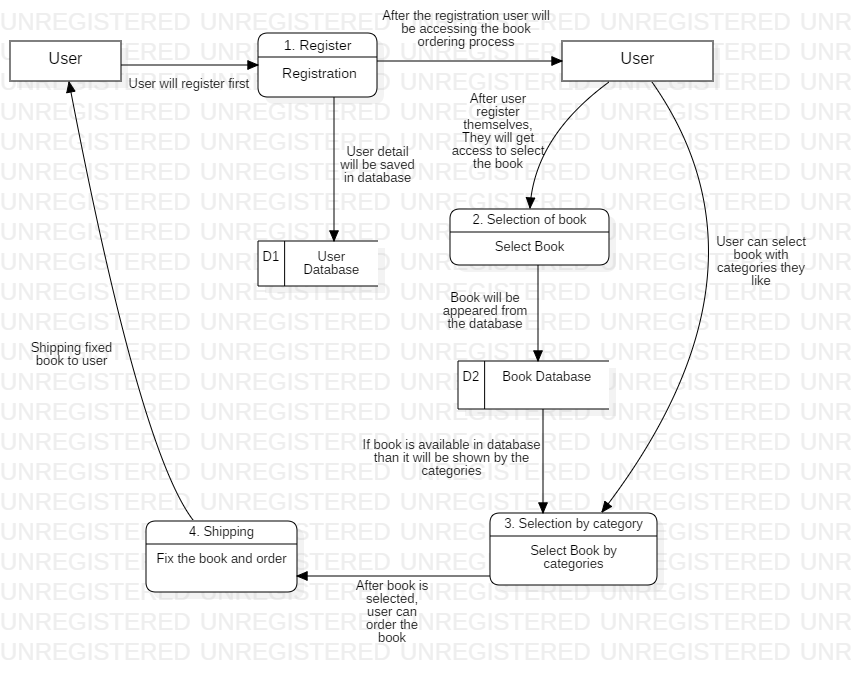


Fig: Data Flow Diagram

**Explanation:**

In this data flow diagram, User is the initial entity. User will be registering themselves before selecting the book. After the user will be registered, data will be saved in the database of the users. After the user data will be on the database, user can select the book. User can also select the book with the categories. The categories of the book will be appear from the book database. After the book will be selected by the user of their interest than book will be ordered than will be shipped to the user.

**3.3 Behavioral Design**

Behavioral design is that type of design which help to communicate between the two objects. This design create the flexibility between the objects.

Behavioral design also helps to find out how the system works and does the work. Behavioral design examples are: Activity diagram and sequence diagram.

1. **Activity Diagram:**

Activity diagram are those type of diagram which show the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. ***(uml\_activity\_diagram, 2019)***

Activity diagram generally describes the operation of the system. The Control flow is generally drawn from the one operation to another operation of the system.

For my project too, I have made the activity diagram. In my activity diagram I have shown the flow of an operation which makes the system/ in a system.

**Justification:**

Activity diagram is made to see the dynamic aspects of the system. It means that, it is the systematic flow of process. With the help of this diagram, we can easily assume that what will happen in which step.

For my project too I have made the activity diagram. In that activity diagram, it clearly shows the step that will carry out to complete my project.

**Notation Used:**

*Start Point:* This is the point which represents the initial action state.

*Action State:* This point represents the non interruptible actions of the step.

*Action flow:* This notation helps to flow the object from one action state to another.

*Decision branching:* This notation helps to decide like if/else condition.

***Synchronization:***

*Fork node:* This notations help to split the one action state into two or many.

*Join node:* This notations helps to combine the two or many action state to one action state.

**Diagram:**



Fig: Activity Diagram

**Explanation:**

Explaining about the dynamic state of system is only possible with the help of activity diagram. So in this diagram, I have clearly described how my system will work. At first user will register themselves, after that they can login into the system. If password/email will be incorrect, they must start with login again. But, if the login is successfully done than user can easily search book and order for themselves.

1. **Sequence Diagram:**

Sequence diagram are the interaction between the diagrams which gives us detail about how the operation is carried out. Sequence diagram are generally the time focused and capture the interaction between the objects.

For my project too, I have a sequence diagram. With the help of this diagram, viewer can easily get the knowledge about the interaction between the object and how they deal with themselves.

**Justification:**

For my project I have made the sequence diagram to know the interaction between the objects. As sequence diagram show the flow of the data in sequence, I have made the flow of data of my project in sequence as shown in the figure.

**Notation Used:**

*Lifeline:* This notation helps to represent the all instance in every interaction.

*Message:* This notation help to send the text.

*Message reply:* This notation help to reply the text to the object.

*Self message:* This notation help to write the text to own self object.

*Actor:* This notation is for the object which is important for the system to run.

*Combine fragment:* This type of notation help to write down the if/else statement.

*Object:* This is the class like structure without which system flow isn't possible.

**Diagram:**

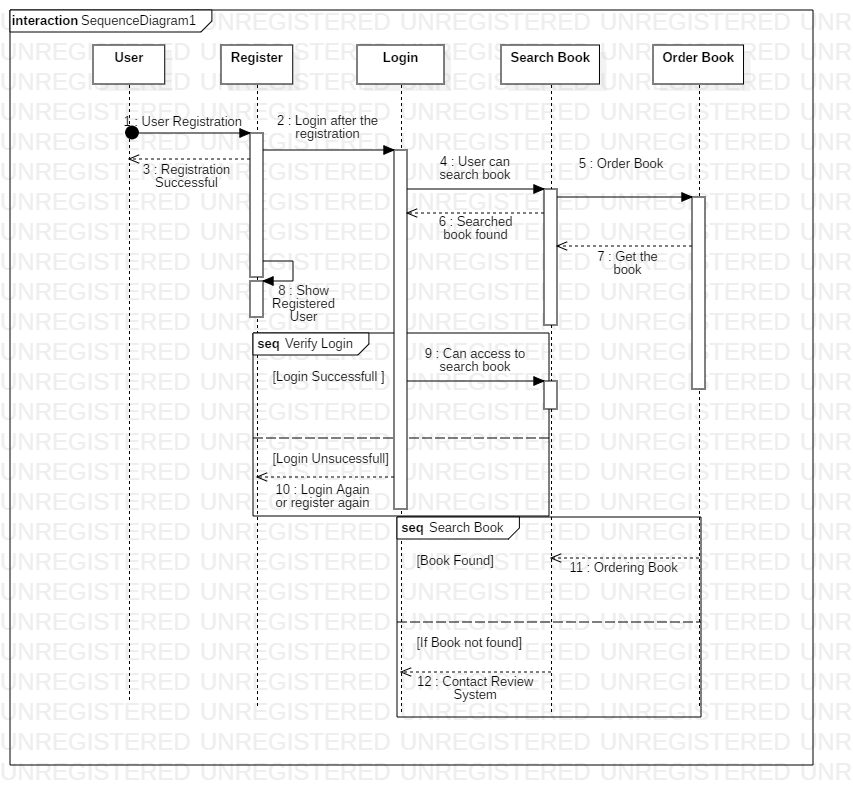


Fig: Sequence Diagram

**Explanation:**

In this sequence diagram, I have clearly shown the object that flow through the entire system. In this sequence diagram I have started with the actor and finished with the ordering the book. In this sequence diagram, I have also mentioned the word of combined fragment, which shows the if/else like statement. As the figure is Cristal and clear, anyone can know what will be happening in the system. Here in the above. There is no much work of Admin than to reply in the review system, so that it is not quite necessary to give admin the space in the figure above.

**3.4 Database Management System**

Database management system is the management software to manage the database. This system also help to create the database. This system creates the way to end-user to create, edit and delete the data.

There are more advantages of the DBMS:

* Data security and independence
* Robust data integrity
* Auditing of activity
* Simple access using API
* Data abstraction

1. **Data Dictionary:**

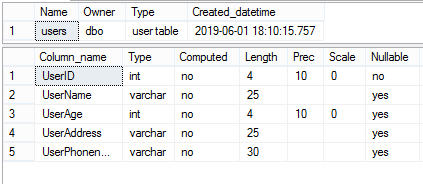
A data dictionary is a file or a set of files that contains a database's metadata. ***(data-dictionary, 2019)***

Data dictionary also contains the data of other objects. Such as data ownership, data relationships. Etc. Data dictionary is the crucial component of the relational database. Data dictionary generally contains of Name, type, length and many more.

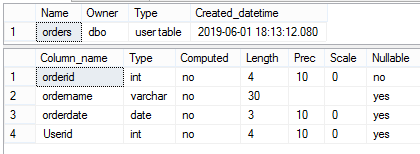
**Justification:**

In my data-dictionary, I have taken 8 entities. All 8 entities do have the metadata. As data dictionary give the detail information about the Meta data, here is the data dictionary of my project.

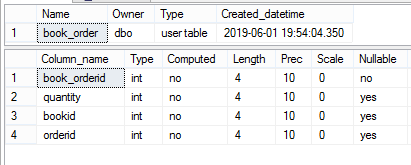
**Data-Dictionary of entity Users:**



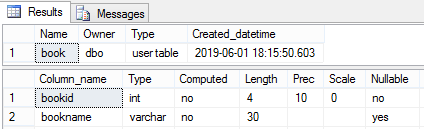
**Data-Dictionary of entity Order:**



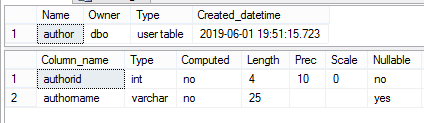
**Data-Dictionary of entity Order\_Book:**

****

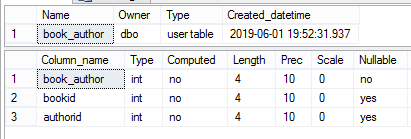
**Data-Dictionary of entity Book:**

****

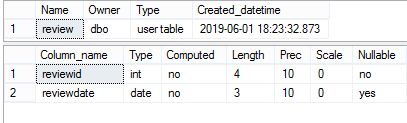
**Data-Dictionary of entity Author:**

****

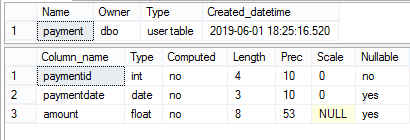
**Data-Dictionary of entity Author\_book:**

****

**Data-Dictionary of entity Review:**

****

**Data-Dictionary of entity Payment:**

****

**Explanation:**

As explaining these data dictionary, all have a proper data with valuable length. Data dictionary clearly shows the datatype, length, null able or non-null able value, etc.

Due to the above figure, it clearly shows the data dictionary of my project.

1. **ER (Entity Relationship):**

ER is known as Entity relationship. ER diagram generally gives knowledge about the relation between the entities. They are also very important to modeling anything from simple to complex database.

**Justification:**

For my project, I have relate the 8 entities. In the 8 entities, I have given the column name which is proper to them.

Here are right Entities:

* Users
* Order
* Book
* Book\_Order
* Author
* Book\_author
* Review
* Payment

**Notation Used:**

*Fields:* This notation is used to classify the entity and their columns.

***Keys***

*Primary key:* This key uniquely identifies the column.

*Foreign key:* This key are created any time an attribute relates to another entity.

***Cardinality and Ordinality***

Cardinality and Ordinality are the notations that are used to relate the one instance of one entity to another instance of another entity.

In this types, I have used these 3 notations:

*One to many:* This notation helps to relate the one instance of one entity with the many instance of another entity.

*Many to many:* This notation helps to relate the maximum number of times an instance in one entity with other many instance of another entity.

**Diagram:**

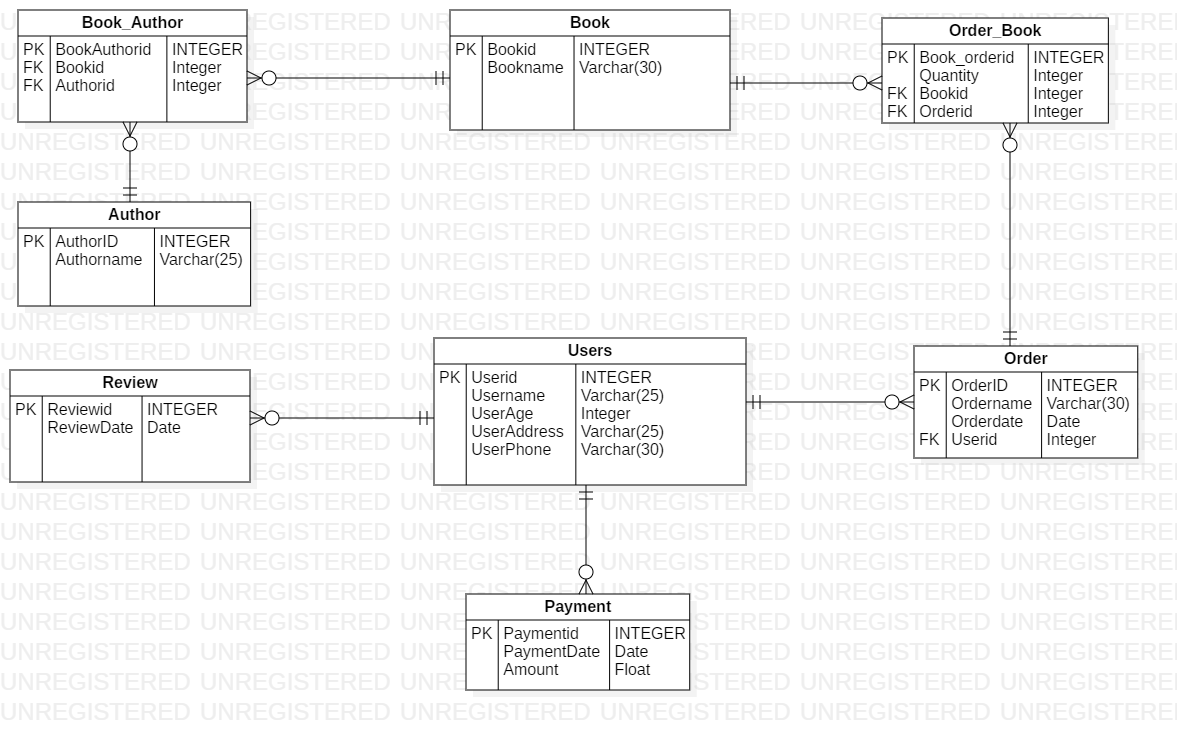


Fig: Entity relationship diagram

**Explanation:**

As I have created the entity relationship diagram, it gives the knowledge about the entity and how they act with each other. This entity relationship diagram also gives an idea about which entity is dependent with which entity.

**3.5 Architecture**

Architecture is the fundamental structure that must be made to give shape to the system. Architecture is generally called the blue print of the system. Architecture helps to design the system, edit it if necessary and proceed to develop the system.

I have used 3-tier architecture for my project.

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. ***(dbms\_architecture, 2019)***

3-tier architecture is mostly used architecture in the DBMS.



3-tier architecture works with the help of Presentation, Application and Database tier.

In the **presentation tier**, user operating will be carry on. They only know about the things that they see, they won't be access to the database.

In the **Application tier**, the connection between the application server and the database will be connected. For the user, this tier will show the abstracted version of the database.

In the **database tier**, all the data of the system will be stored. In this tier query processing language will also be stored.

**Justification:**

As completing the talk about the architecture, now I have made some of the paper prototyping which completely define my system. As paper prototyping must be clear, I have made the prototyping in the clear way.

**Here are the prototypes of my project:**

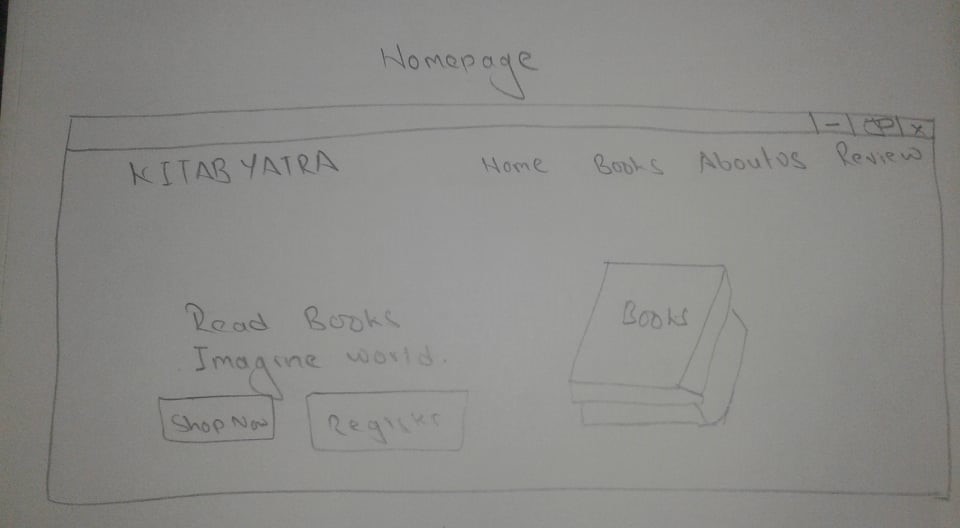


Fig: Homepage of Website

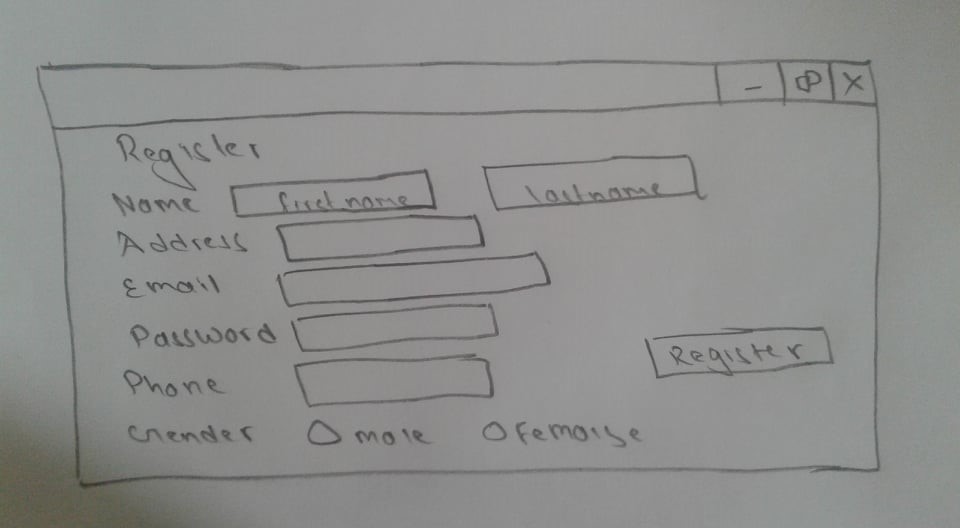


Fig: Registration Form

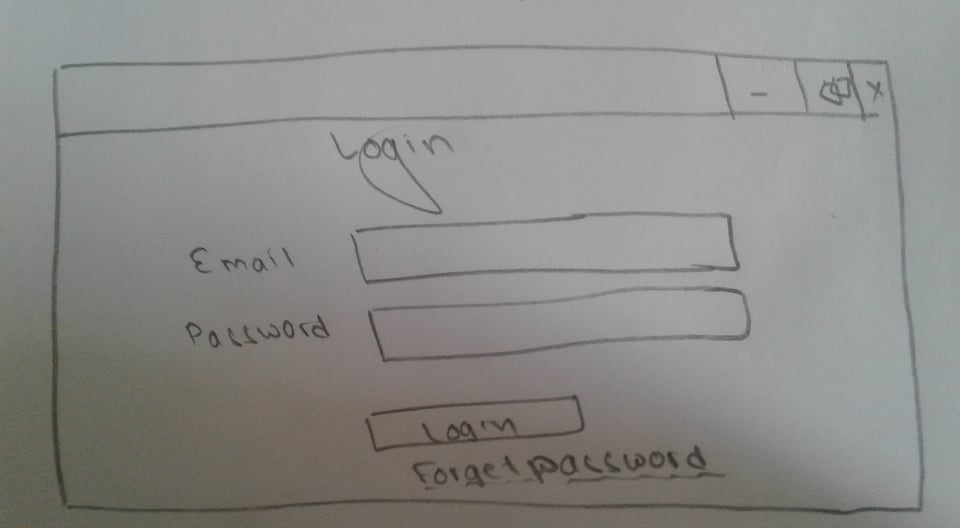


Fig: Login Form

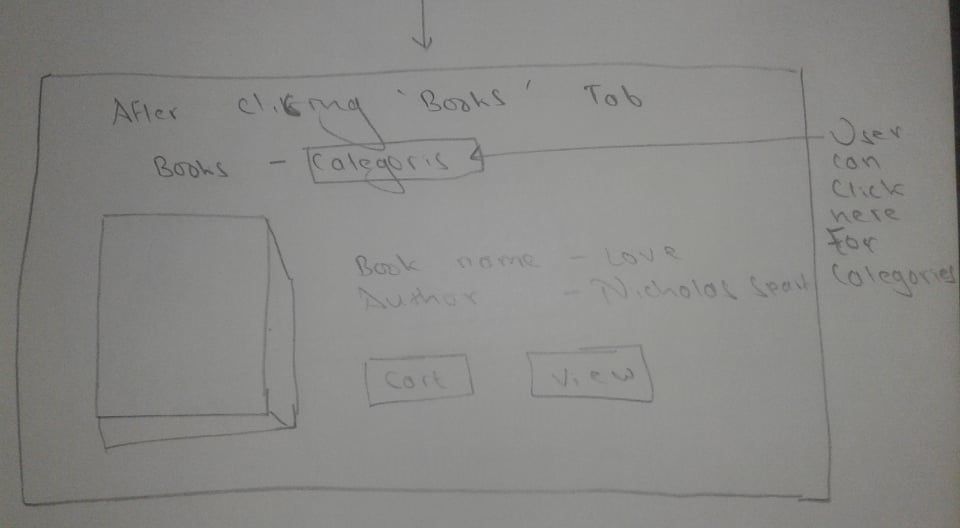


Fig: Books section of website

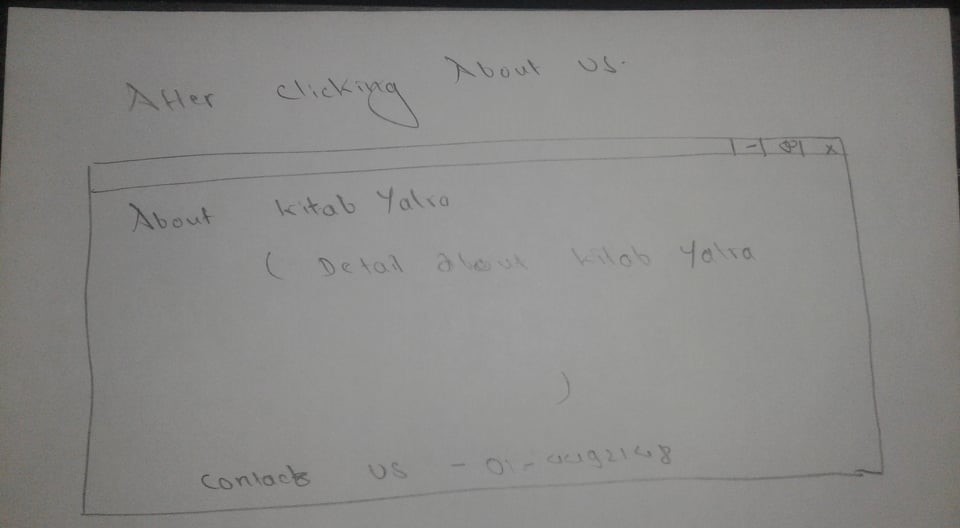


Fig: About Us Section of Website

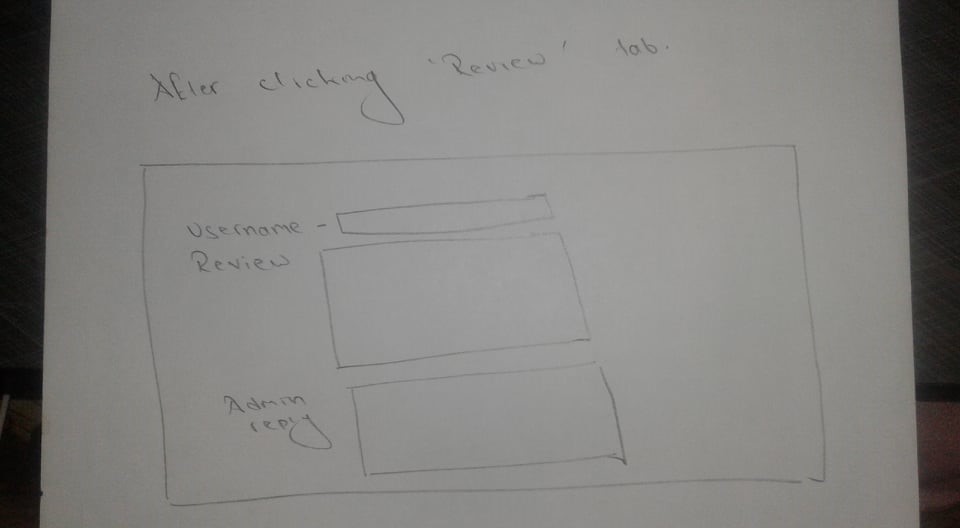


Fig: Review section of website

**Explanation:**

Above in the figure, I have made 4 prototypes of my project. Prototypes are only the blue print of the system and if the system require more feature than it will be edited and updated. As 4 prototypes looks in the figure above about how my website looks, it will be made in the same way.