Software Design Document for BooXChange

Software Engineering Course, Sem 6
Prepared on 16th April 2020

PREPARED FOR
Prof. Khushru Doctor

Version 1.0

Prepared By :

Jainam Chhatbar - 1741002	Aman Dave - 1741003
Harshiv Patel - 1741005	Aditya Shah - 1741007
Dhairya Dudhatra - 1741058	Meet Modi - 1741071
Shantanu Sheth - 1741088	

Table of Contents

1. Introduction

The main aim of the project is to create a unified resource management system for all the students around the city. Every year many books are thrown away or kept unused in our desk and many people don't know what to do about those books. Keeping this thought in mind, we are building a platform where users will be able to buy, sell and rent their books. Other than books, users can also exchange their handwritten notes or book a tutoring session.

This design document presents the designs used or intended to be used to implement this project. The design described, follows the requirements specified in the Software Requirements Specifications document prepared for the project.

1.1 Purpose

The purpose of this document is to provide a detailed description of the designs of the BooXchange system. This document will also give a brief idea about the development process through the example of the requirements, constraints, and the system architecture of the product. This document will be used as a way to make sure that all the stakeholders of the system will have a complete and clear understanding of the requirements of the system which involves each and every process.

1.2 Scope

The introducing product, BookXchange is a community-driven society where only trusted kind of book transactions happen. The major task of this system is to streamline this procedure in a hassle-free manner. Our system is going to get used two kinds of user class:

- Customer base: Contains all the whole front facing of the business idea
- Admin base: Contains all the functionalities related to the management of the business.

The end users can be customers as well as employees both. Customers can search for, view and can purchase the books and can also make payments. They can also upload the details of the book for putting their book on sale/rent. The objective of this system is to make this process occur smoothly. The system also handles the issues regarding any complaints from the customer and resolves them in a quick manner. The admin/employees can search/track any order that has been processed earlier/in-transit. To run this smoothly, a dashboard will be developed with specific UI features for admin/employees only.

1.3 Process Methodology

The software development methodology we are implementing is SCRUM, as SCRUM is lightweight and has an easy learning curve. With every one new to this field, an easy method to learn is helpful. This is extremely suited if the team size is small i.e. with a team size of 7, this model is best suited. The concept of the sprint is very helpful and it keeps everyone accounted for. Backlogs act as a buffer and it will be really helpful when different team members have different submission deadlines for their respective electives. Also, it allows incremental development of features. After every iteration, a new feature request can be accommodated

1.4 Stakeholders Involved

There are two major stakeholders involved in this product and their characteristics

1. **Customer:**
   Who can be the customers:
   - Students across various universities.
   - Those who are independent readers.
   - Those who want to tutor various topics and chapters.
   - Those who are willing to lend/rent their books as well as handwritten notes.

   Customers can log in to the system. Furthermore, the customer can also view the products and services and their details on the site. After selecting a product to their liking, the customer purchases the product and makes the payment for the same. The customer has restricted privileges as they must not have access to the financial data as well as the stock, staff and room data. They also cannot update things like the book rate.

2. **Admin (Employee):**
   Admins have their own dashboard, where they can approve/reject the requests of books coming into the marketplace and handle financial complaints/product-related complaints. After the order has been received, it is the admin's job to process the order and after that pickup details are shared with the courier company and a tracking id is provided for each order. Admin can view accounting information of each customer.

2. Design Considerations

2.1 Assumptions

Various assumptions have been taken into consideration about the users. Users should have basic knowledge of surfing on the web and must be aware of the methodologies and terminologies in an e-commerce website, wherever not specifically specified. The users should have a valid e-mail for placing orders through the web-app.

2.2 Constraints

The system will always need an internet connection to work. And this system is built using ReactJS, Javascript and MongoDB technologies. This technology will work until the continued support of these three technologies.

2.3 System Environment

The BooXchage is designed to work on all operating systems. The system is accessible through all desktop/mobile environments through a browser. There is a requirement for hosting the web-app, the operating system has to be Linux based for hosting and implementing the tech-stack.

2.4 Design Methodology Based on NFRs

2.4.1 Capacity

The application is expected to scale itself and should serve the new users when there is a high amount of traffic on the website.

2.4.2 Response Time

The application should be designed in such a way that there is minimal access to the database for lesser latency and the uptime performance guarantee will be as per the guarantees provided by the Web and Database Hosting Service.

2.4.3 Roll-back

In case of bugs based development, the system should be able to implement a patch/fix easily or in unavoidable circumstances, a roll-back to the previous version should be done seamlessly.

2.4.4 Usability

User Interface: The design of the User interface is consistent through the system and has a minimum need for memoization for the User and maintains a high amount of focus on User Experience(UX).

Compatibility: The application will support all the screen with different screen resolutions and screen sizes with a relative font size according to the size. The User Experience will not be degraded due to different screen size.

2.4.5 Security

Authentication: Authentication of users while registering their account needs to be authenticated. For that, an API is created which should generate an OTP and send it to the user to their e-mail and the server will provide a token which will enable the user to stay logged in the system. Each time, a new token is generated and stored in the database.

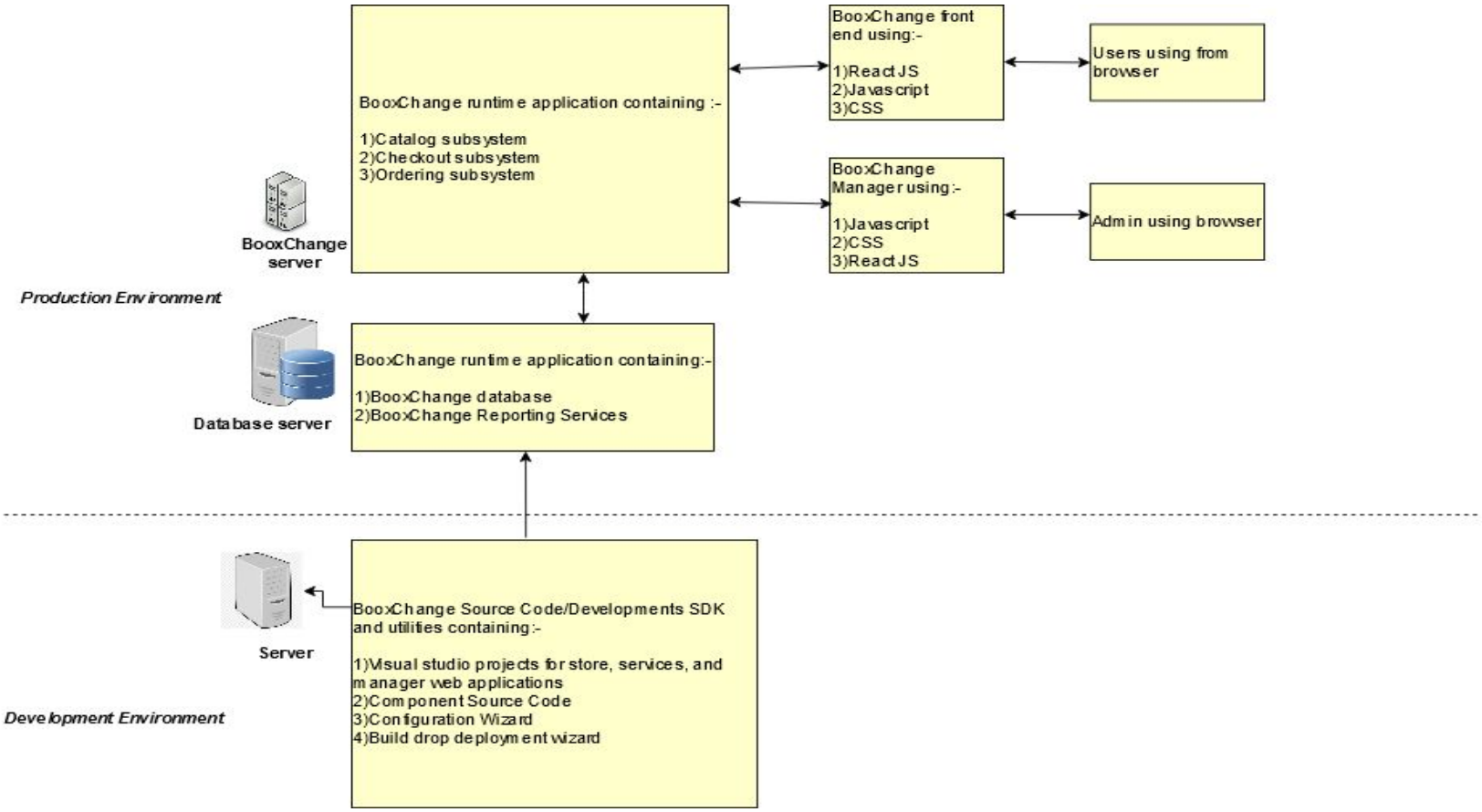
SSL Layer: Industry-level security filter will be added for all the interactions with the database or transmission of any personal data to the Server/Database. And all the interactions to the database are done through HTTPS requests only.

Log in: Each user will be able to log in through their Email-id.Each Email-id and Password is stored in an encrypted form in the database. The user will be able to create a new password using OTP sent to his mail-id.

Secured Data: The application will employ industry-stranded algorithms for data-transmission into databases.

3. Architecture

3.1 Technical Architecture

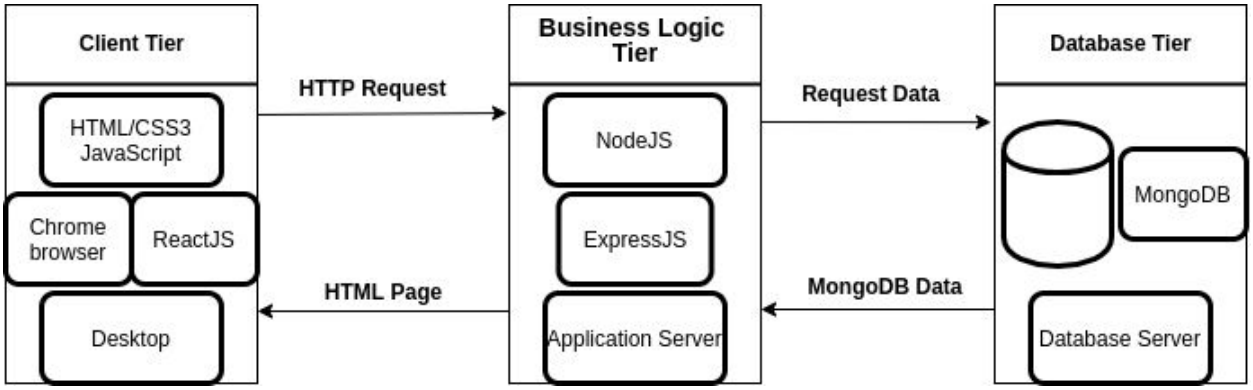


BookXchange Technical Architecture

The above figure depicts the whole software development cycle for the product BookXchange. Here there are two environments considered in this design :

- **Development Environment** includes the whole tech-stack on which the code has been developed using developer tools like IDE and Databases and all the configuration of various parameters have been done.This environment contains many version-controlled codes which are buggy and in not working condition. This environment includes only development server and is only meant for development use only.
- **Production Environment** includes the final release or version controlled release of the software which is meant for use of general public and the organization.This Environment incorporates business application needs and has a database server that will be used by businesses and the organization,also there will be a web server which will host the whole Application on the web.

3.2 System Architecture

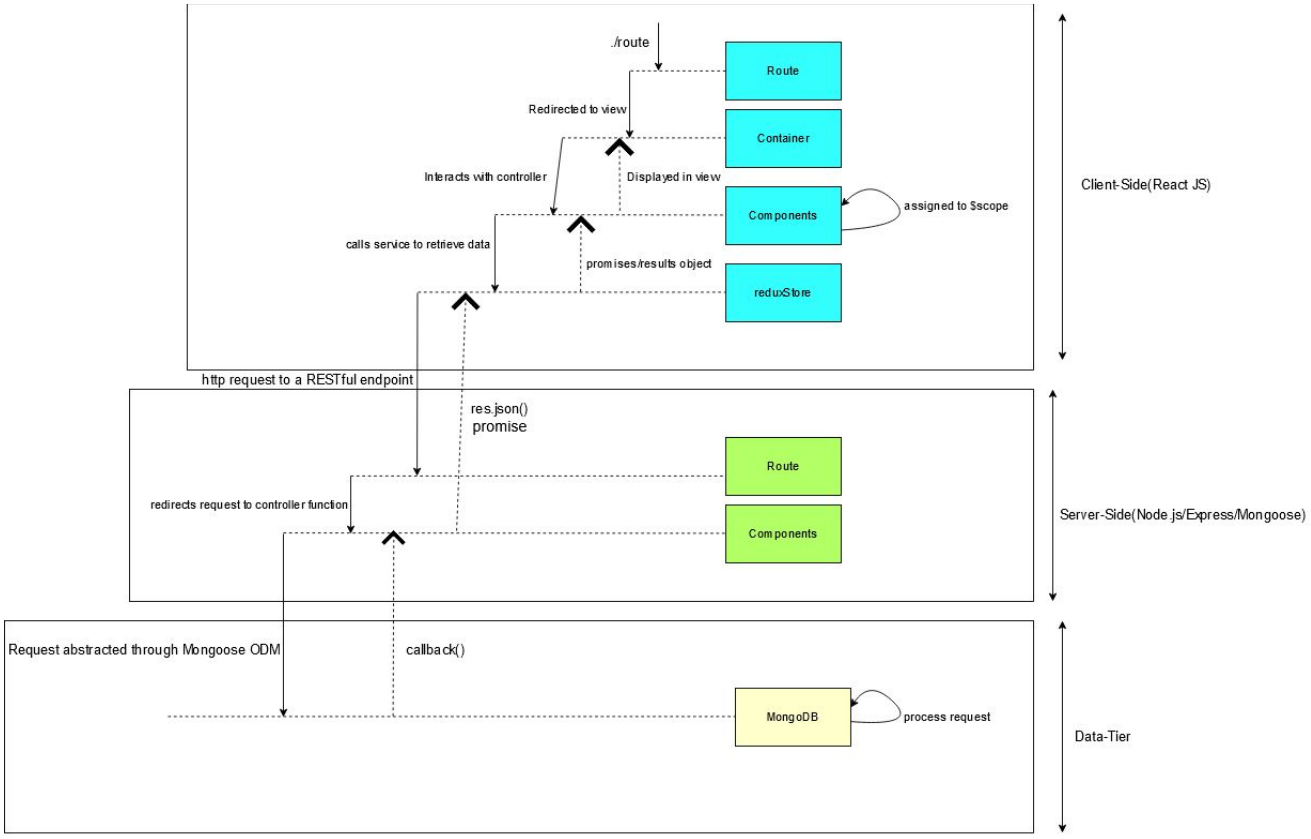


Here, the design of BookXChange Software is done in consideration of NFR's stated in the above chapter 2.We have proposed using NoSQL architecture,for faster development of thee product.BookXchange product will be using the MERN (MongoDB,Express.js,React,Node) for development.

- **MongoDB :** MongoDB is a document database, which means it stores data in JSON-like documents.This has a great advantage for handling data in JSON objects and these data objects can be transferred throughout to the front-end without any interference or conversion through different technologies.
- **Express.js :** It is a modular web framework for Node.js. It makes it easier to write secure, modular and fast applications
- **React.js :** It enables developers to create web applications in a scalable, simple and fast manner
- **Node :** Easier way to write back-end APIs and handle the server.

The team was familiar with Javascript, and so we all decided to stick with MERN stack.

3.3 **Architecture Process**



In the process of our architecture, we have divided the system into 3 parts: Client (User), Server and Data tier.

Client:

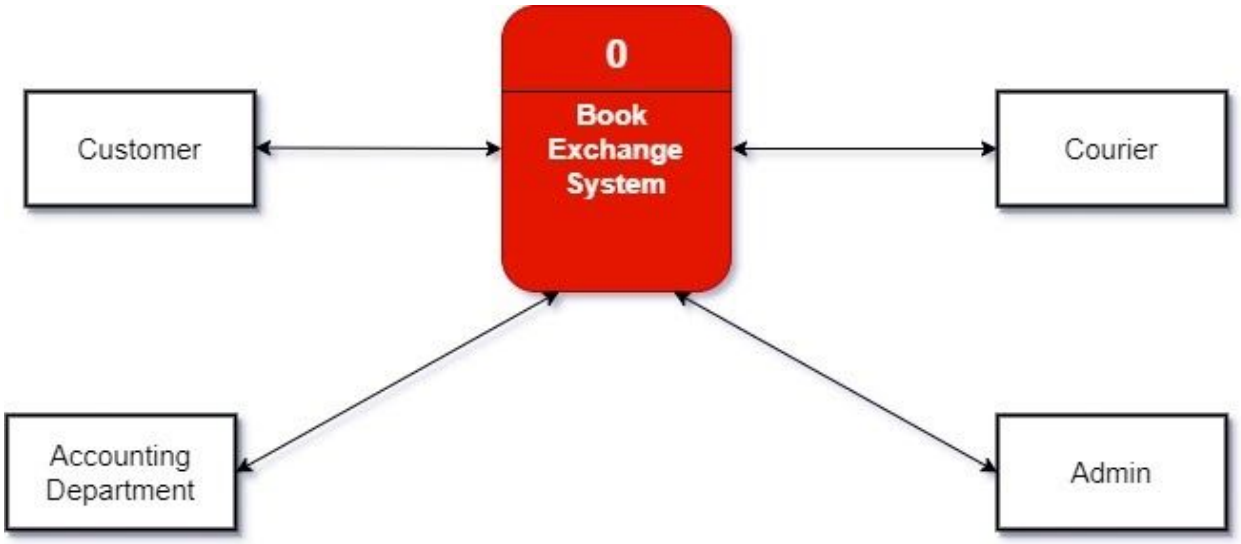
The flow starts from Frontend. Users visit the website on a particular **route**. On hitting that route, a main **container** will be called which contains the logic of individual **components**. State allocation will also happen from here. Based on individual components, **API call** will be made with the help of action-reducer.

Server and Data:

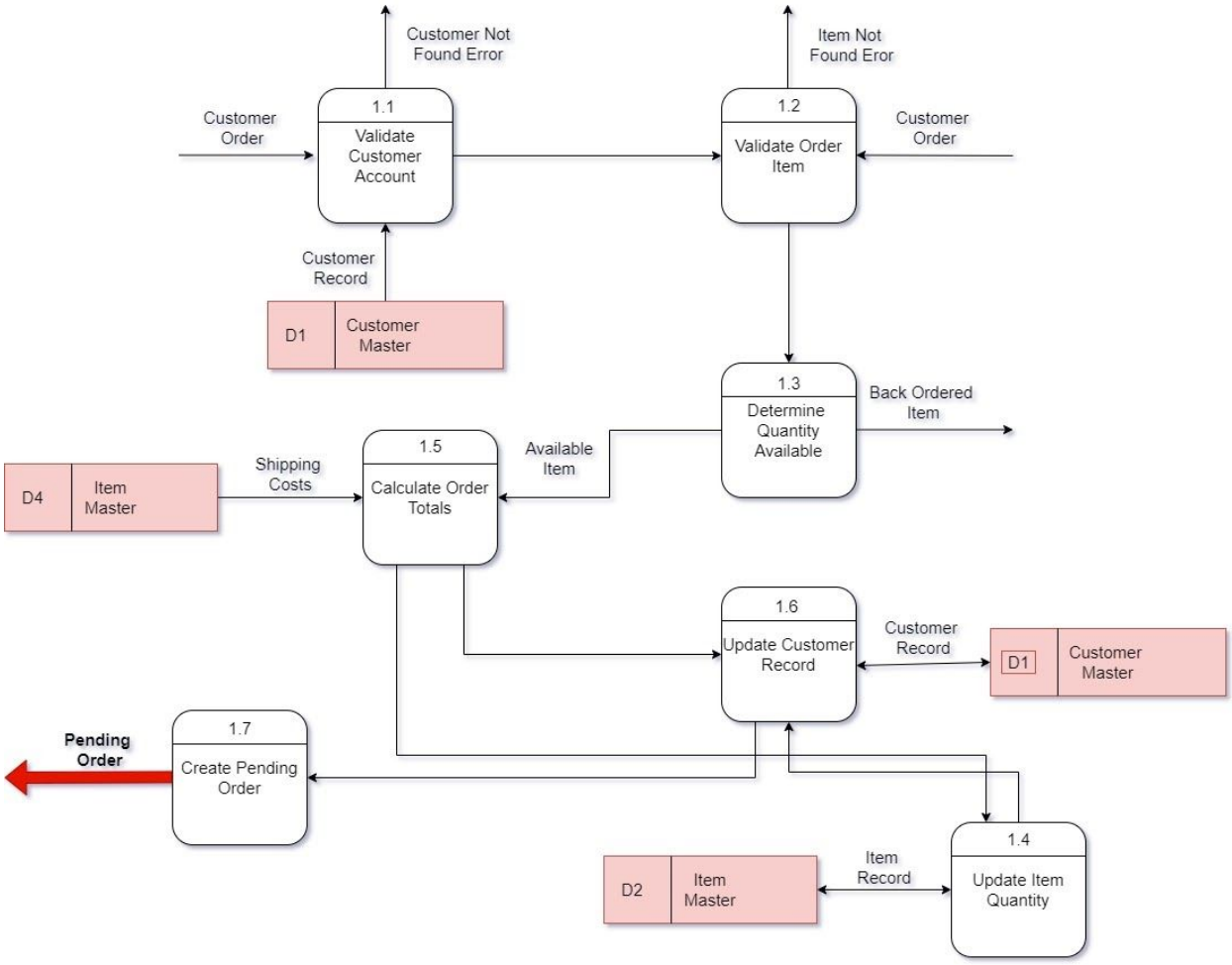
The API call is redirected using a proxy. It will check the **route** and call its corresponding **controller**. The controller contains the individual logic for an API. If it needs, it will make a call to the MongoDB cluster deployed on Atlas.

4. Data Design

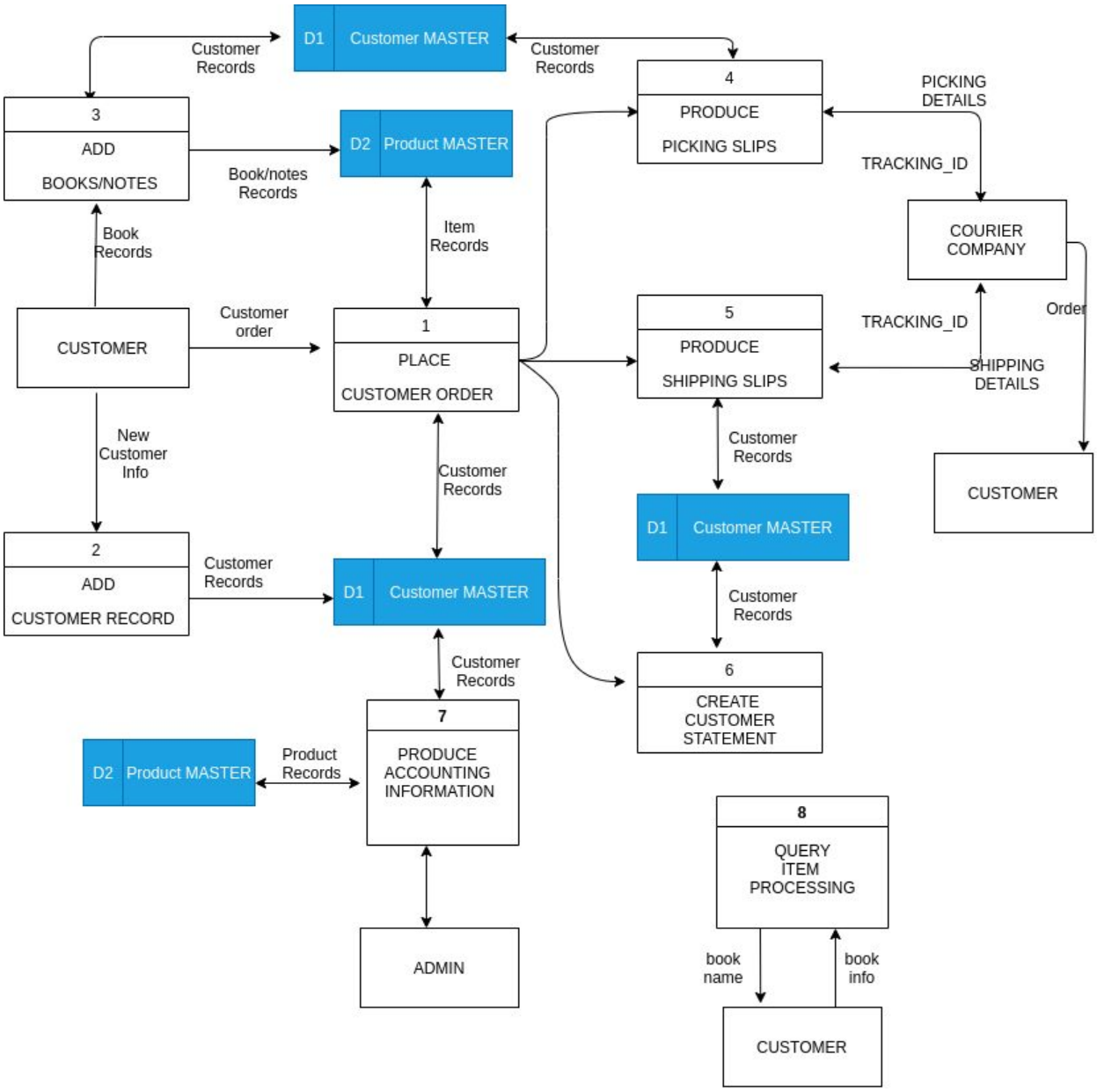
In this design, first every stake-holder which will be using the system in direct or indirect way has been addressed and their functions and use case scenarios while using the process.For this process, Data Flow diagrams were made :



Level 0 Diagram : Every Stake-holder Involved



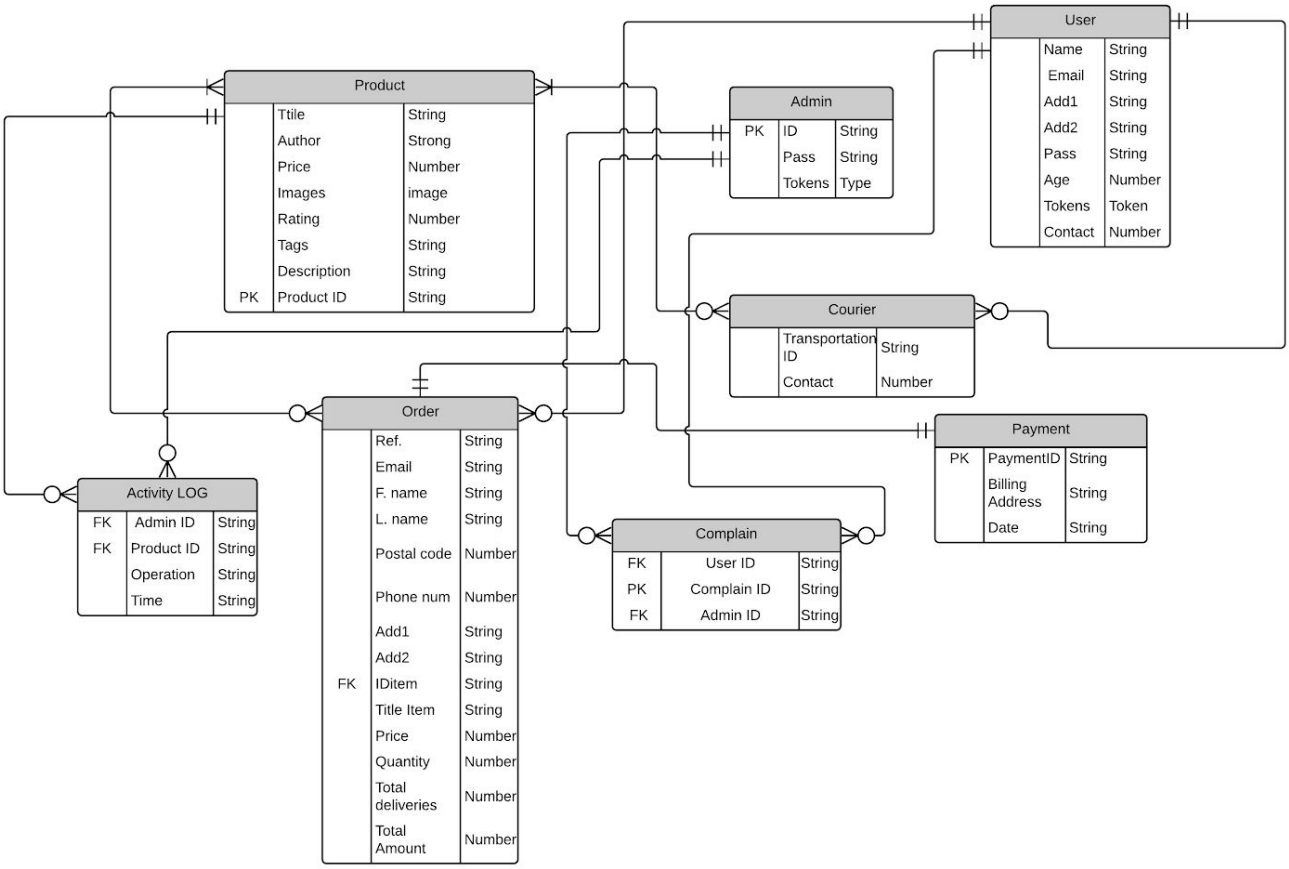
Level 1 Diagram: Processes Involved



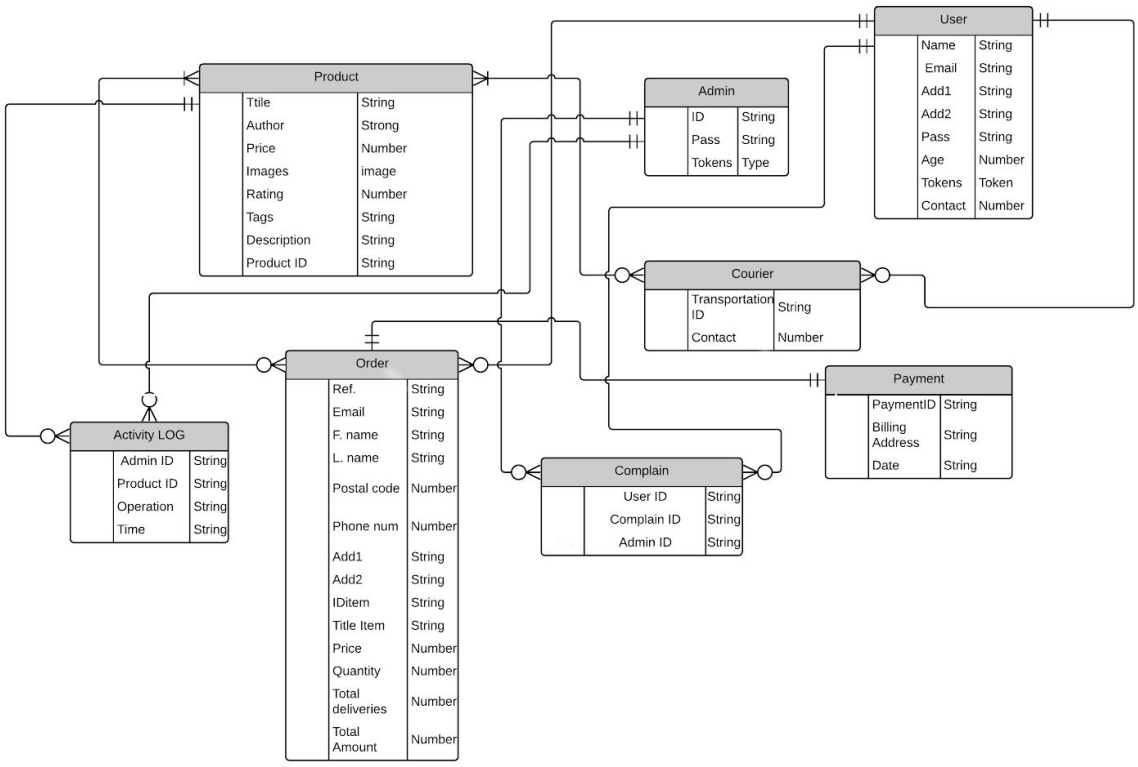
Level 2 Diagram: More detailed Lookup of Processes Involved

4.1 Data Description

Here the Data is represented in NoSQL database.In NoSQL, the data-types are number and string instead of int and varchar.PK and FK are primary and foreign key respectively.This physical ER Diagram is designed from the above processes discussed in the chapter.



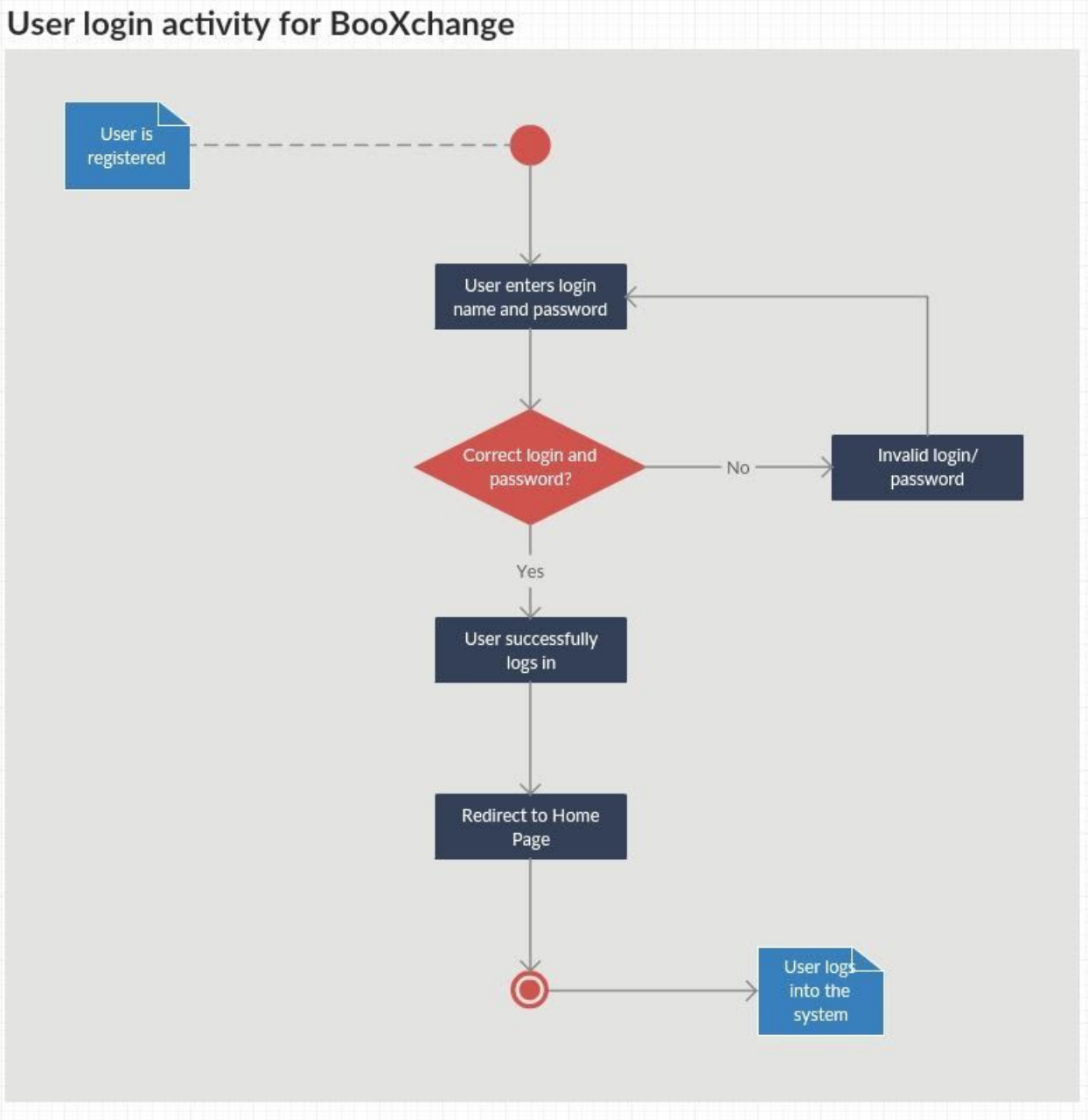
Physical ER Diagram



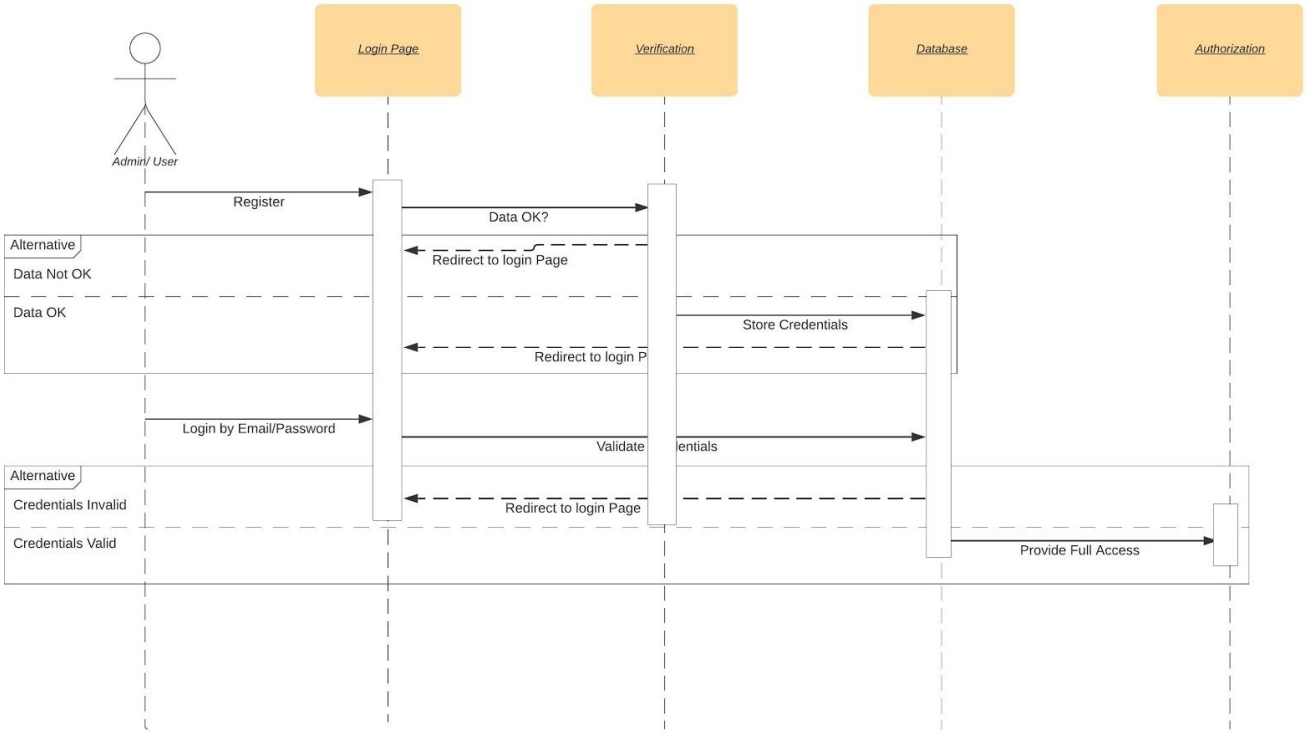
Logical ER Diagram

5. Component Design

5.1 Login

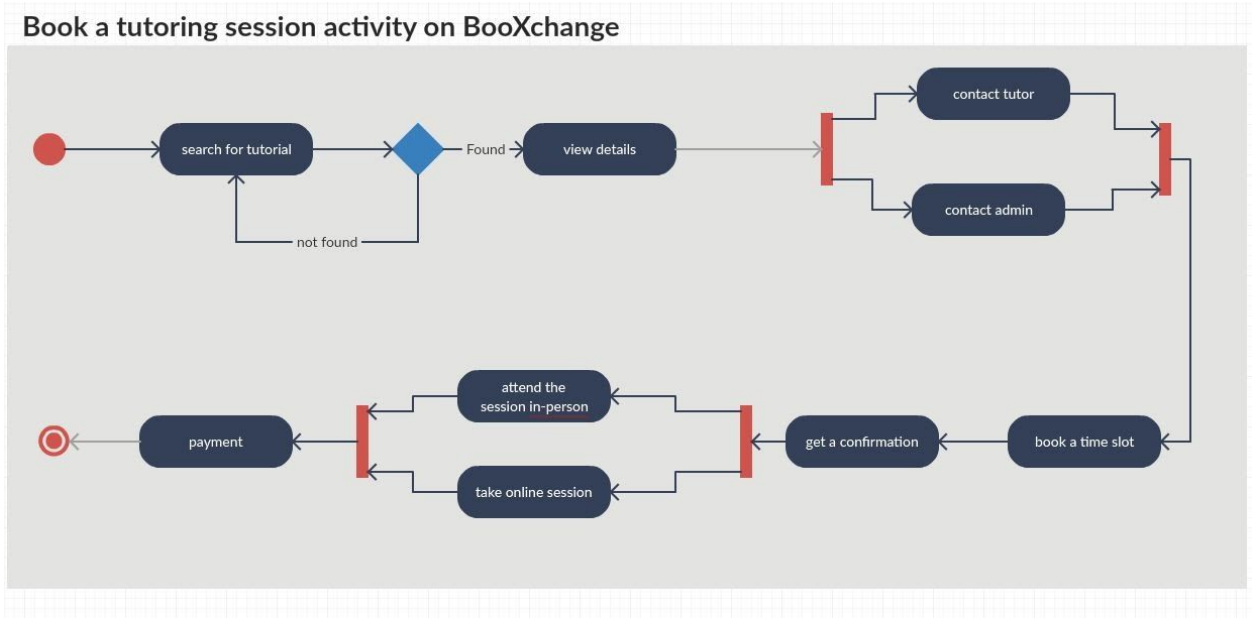


Activity Diagram for User Login



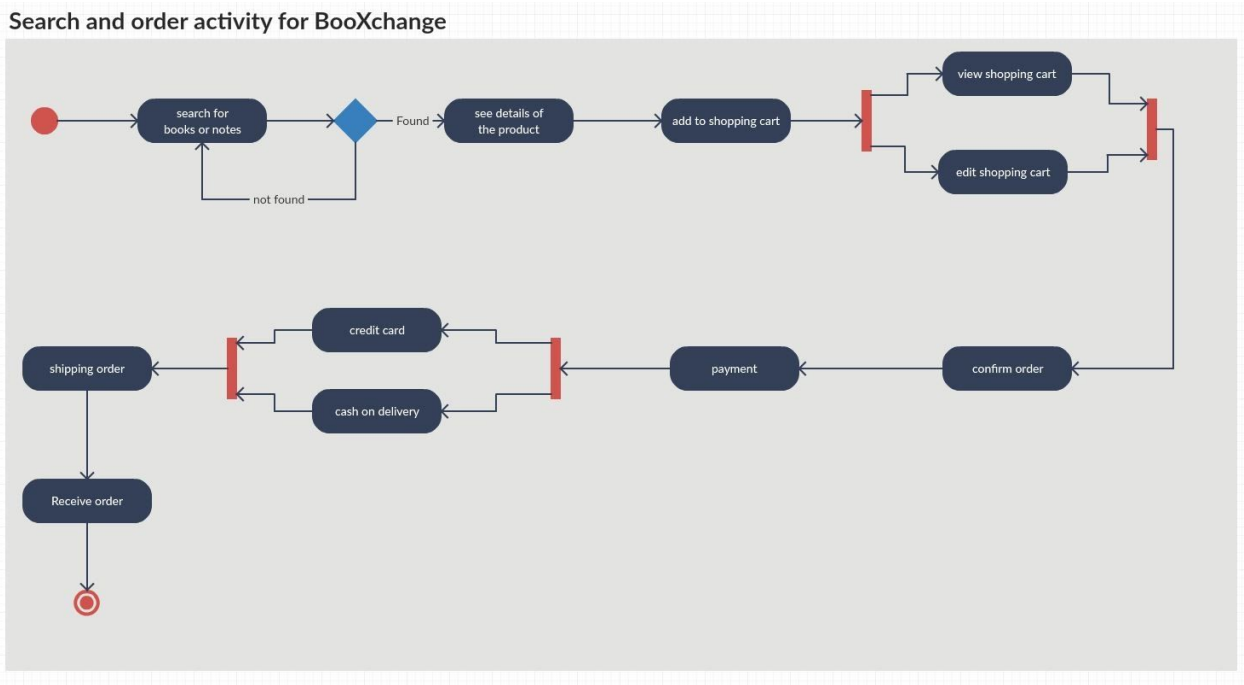
Sequence Diagram for Login

5.2 Book Tutoring Session

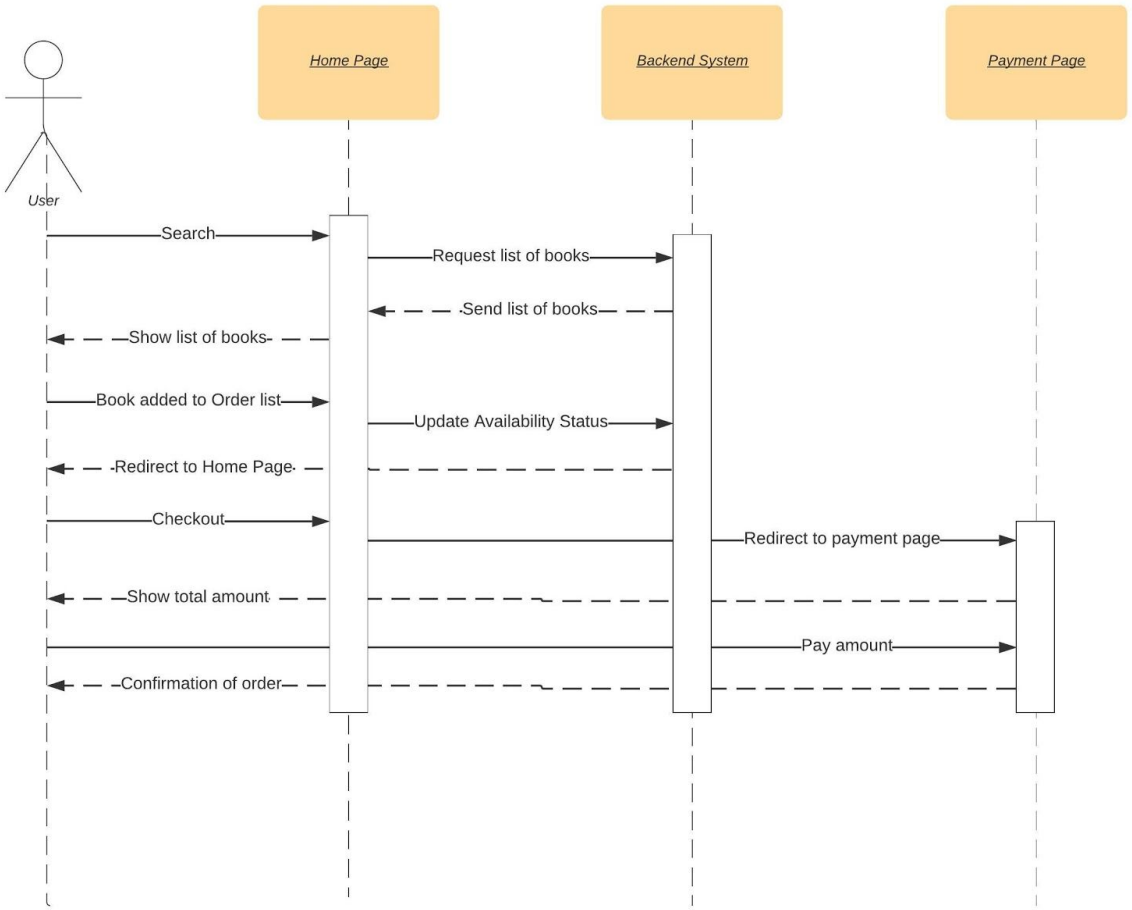


Activity Diagram for Booking a Tutoring Session

5.3 Search and Order

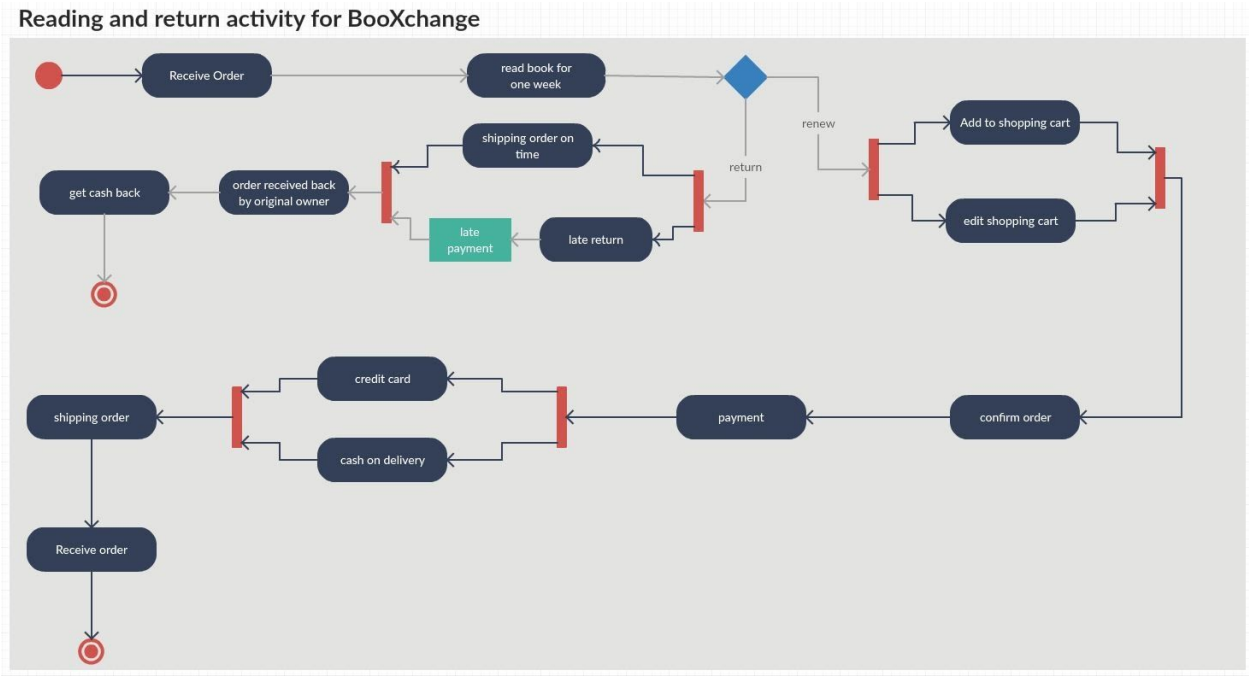


Activity Diagram for Search and Order



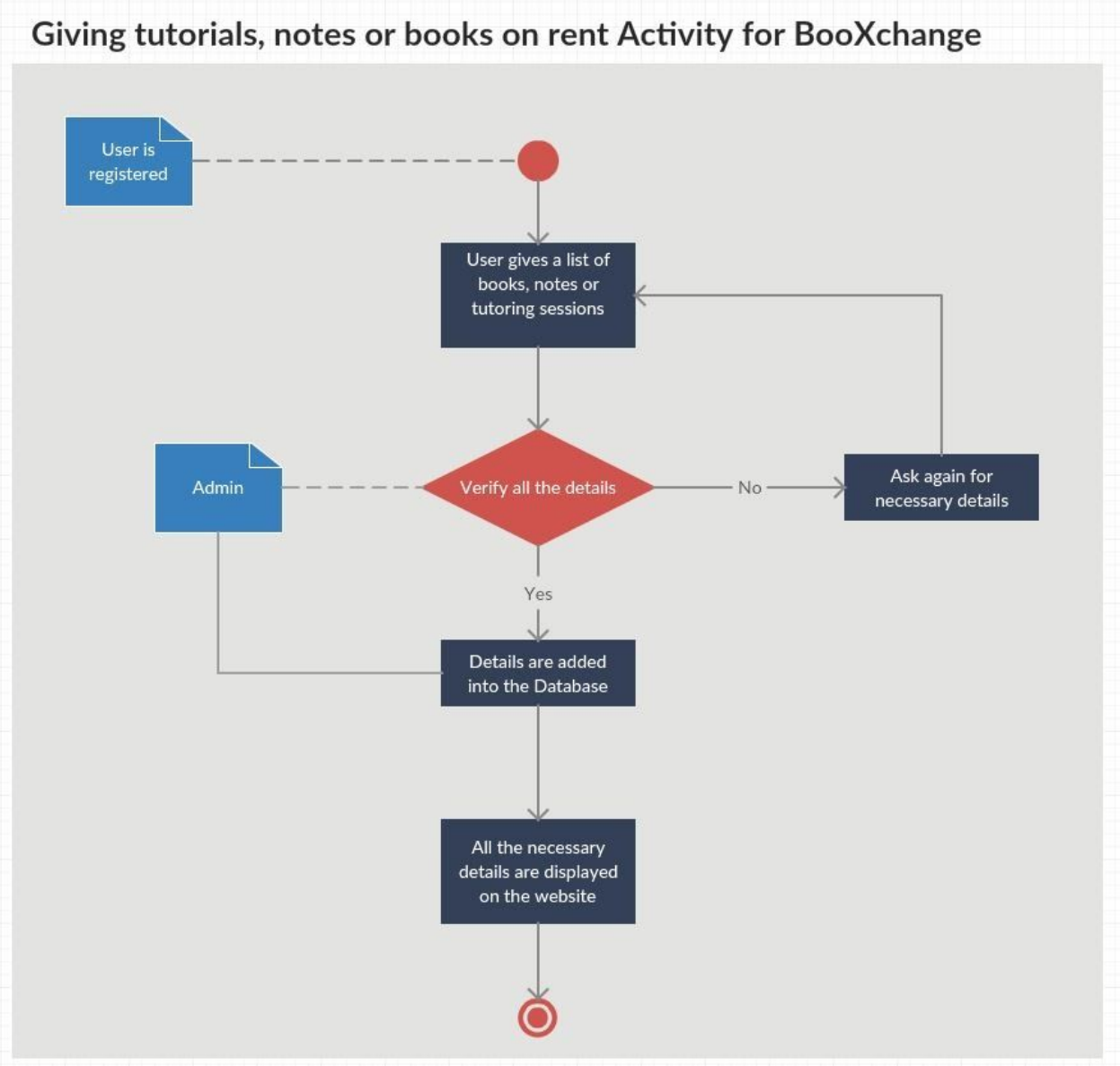
Sequence Diagram for Order

5.4 Reading and Return



Activity Diagram for Reading and Return

5.5 Rent Book/Notes/tutorials

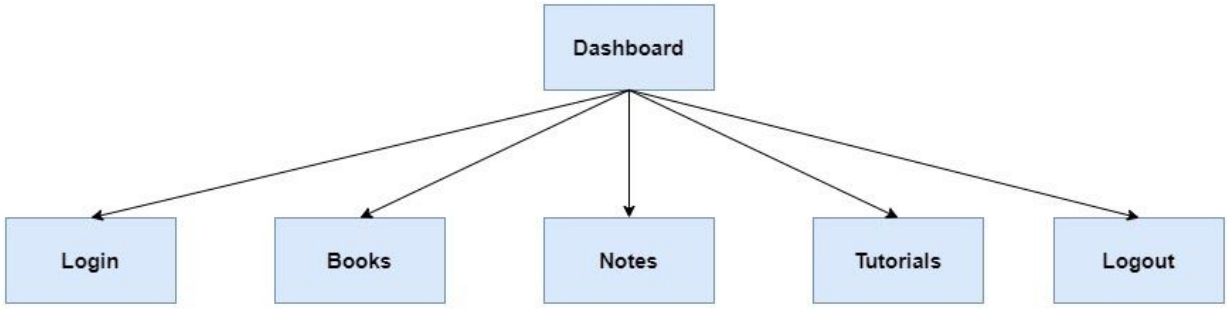


Activity Diagram for renting Book/Notes/Tutorials

6. Software Interface Design

6.1 Web Pages in a Tree

The system's web pages are presented in a tree form. From the main dashboard, the user can reach the following pages: "Login", "Books", "Notes", "Tutorials", "Logout". All these pages cover the necessary functionality of the system. The user can navigate between these pages without any hassle.



6.1.1 Description

The main "**Dashboard**" page contains the navigation bar which contains various options that covers the main functionalities of the system.

The "**Login**" page has all the required operations through which the user can login into the system.

The "**Books**" page has all the required operations through which the user can rent the book they like.

The "**Notes**" page has all the required operations through which the user can rent the notes they like.

The "**Tutorials**" page has all the required operations through which the user can book the tutorial session they want to study.

6.2 **Web Pages**

6.2.1 **Description**

The main "**Dashboard**" page is available to the user where the user has the following menu: "Login", "Books", "Notes", "Tutorials", "Logout". User clicks on the necessary item to go to the next page.

The "**Login**" page has two options:

- Signin: The user can sign into the system with the existing account.
- Signup: The user can create a new account by giving the required information.

The "**Books**" page has the following submenu on the left side of the page:

- Categories: There is the category option in which we have different types of categories like Sci-fi, Novel, etc. The user can select any category according to the type of book they want to read.
- Price: The user can adjust the pricing ranging from 0 (minimum) to 500 (maximum). According to that the books will be displayed.
- Add to Cart: According to the category and the price set by the user, the user can add the book to the cart.

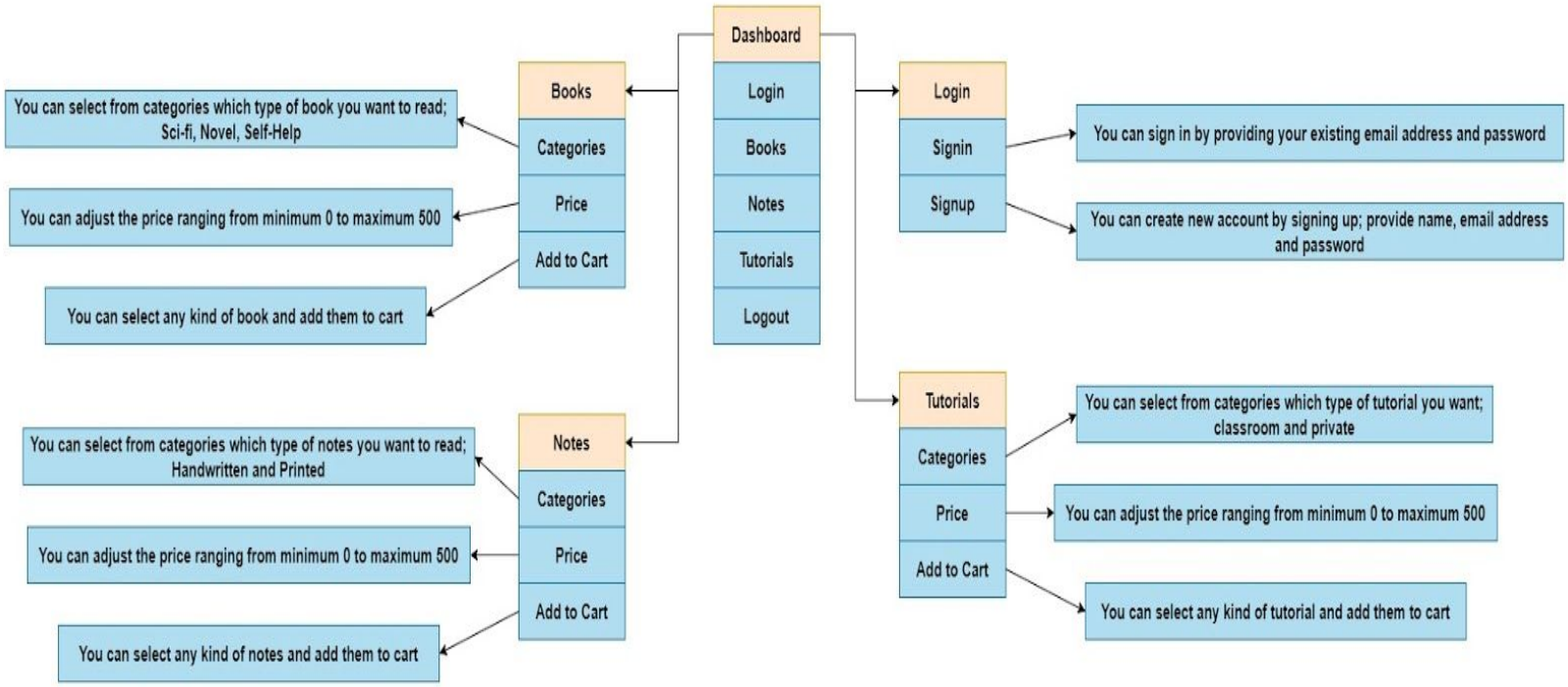
The "**Notes**" page has the following submenu on the left side of the page:

- Categories: There is the category option in which we have different types of categories like handwritten, printedl, etc. The user can select any category according to the type of notes they want to read.
- Price: The user can adjust the pricing ranging from 0 (minimum) to 500 (maximum). According to that the notes will be displayed.
- Add to Cart: According to the category and the price set by the user, the user can add the notes to the cart.

The "**Tutorials**" page has the following submenu on the left side of the page:

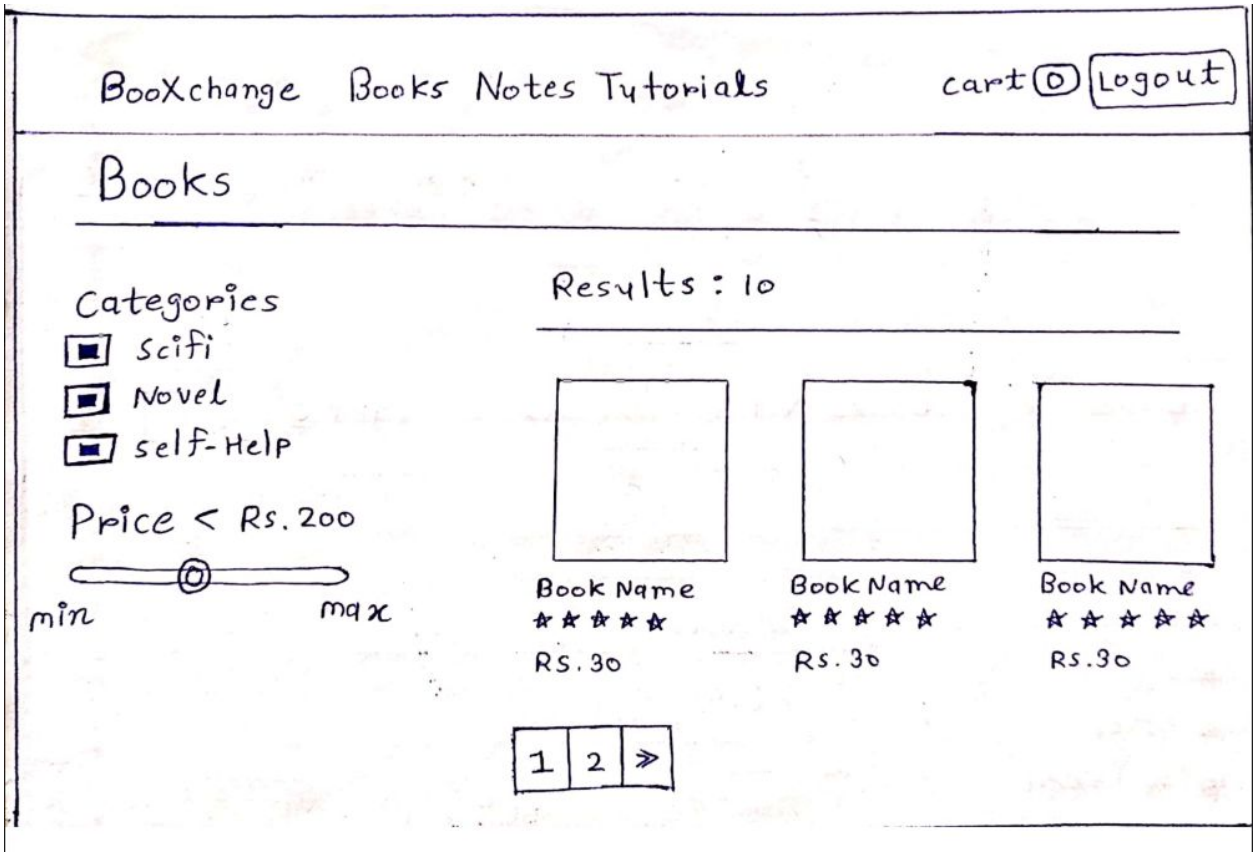
- Categories: There is the category option in which we have different types of categories like Classroom, private, etc. The user can select any category according to the type of tutorial they want like If they want the tutorial in group or in private.
- Price: The user can adjust the pricing ranging from 0 (minimum) to 500 (maximum).
- Add to Cart: According to the category and the price set by the user, the user can book the tutorials.

The user can logout of the system whenever they want. If the user is not logged in the system, they can view and add the books or notes in the cart but at the time of payment, they have to login into the system. If the account is not made then they have to first create the account.

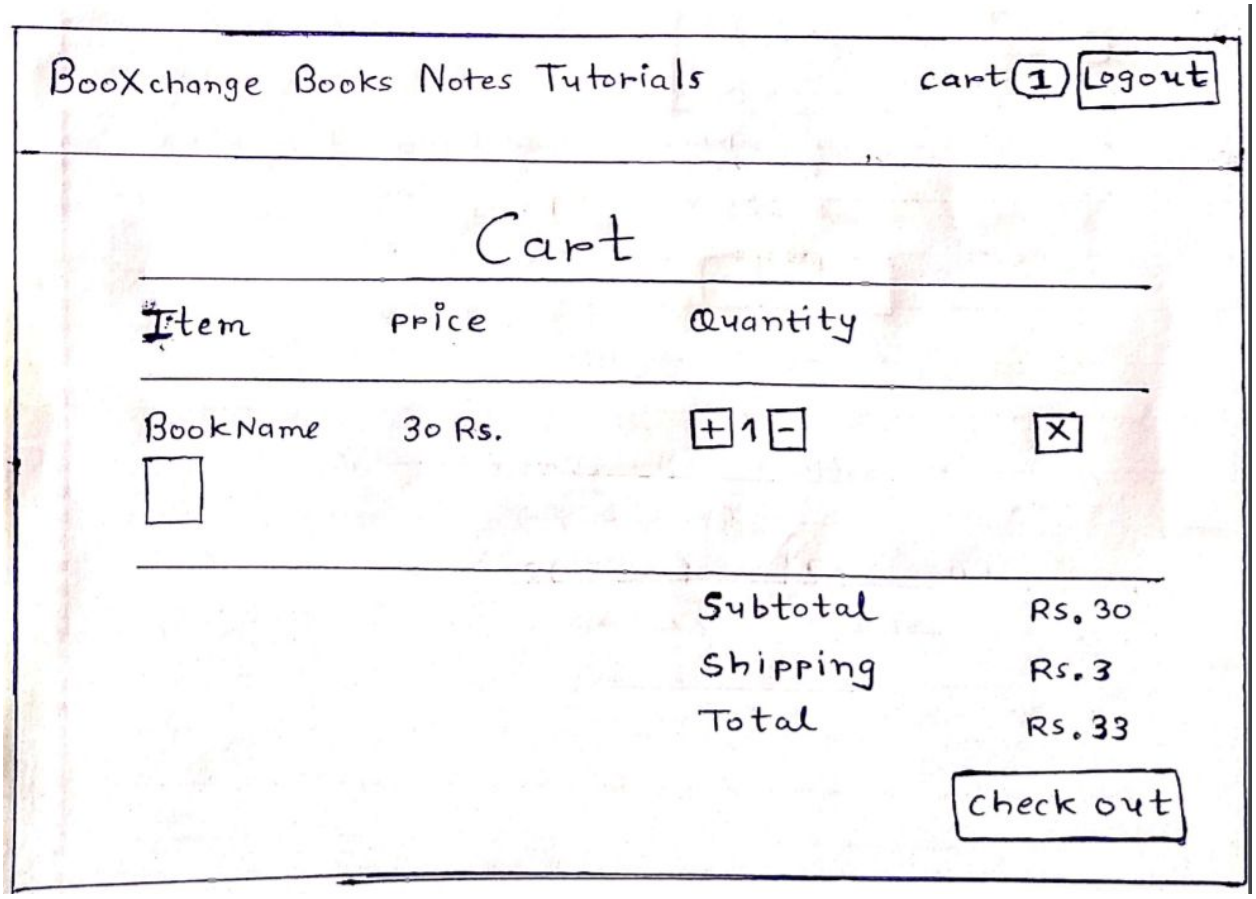

Dashboard
Login
Books
Notes
Tutorials
Logout
Books
Categories
Price
Add to Cart
You can select from categories which type of book you want to read; Sci-fi, Novel, Self-Help
You can adjust the price ranging from minimum 0 to maximum 500
You can select any kind of book and add them to cart
Notes
Categories
Price
Add to Cart
You can select from categories which type of notes you want to read; Handwritten and Printed
You can adjust the price ranging from minimum 0 to maximum 500
You can select any kind of notes and add them to cart
Login
Signin
Signup
You can sign in by providing your existing email address and password
You can create new account by signing up; provide name, email address and password
Tutorials
Categories
Price
Add to Cart
You can select from categories which type of tutorial you want; classroom and private
You can adjust the price ranging from minimum 0 to maximum 500
You can select any kind of tutorial and add them to cart

6.3 **User Interface**

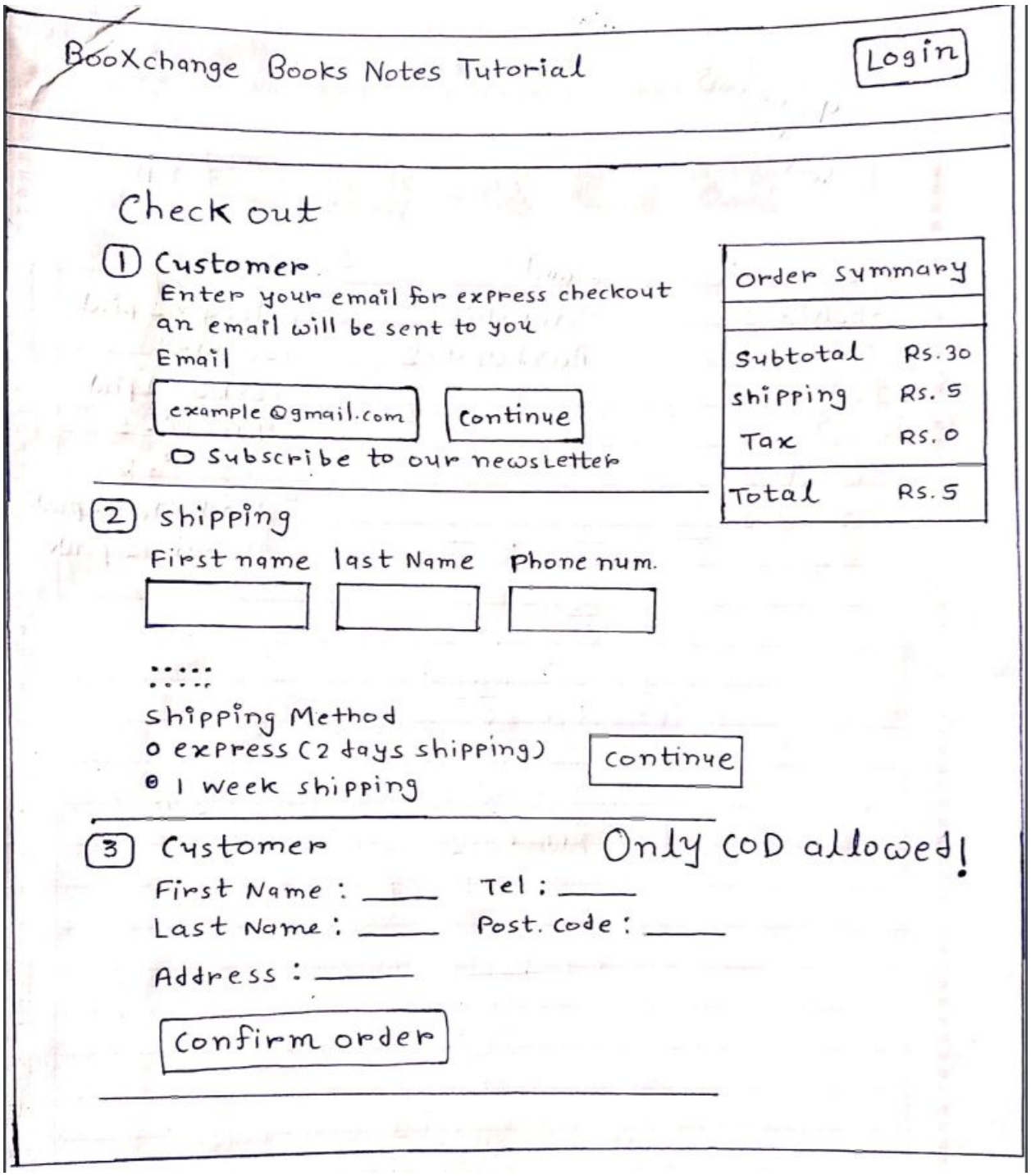
6.3.1 **Browse Window**



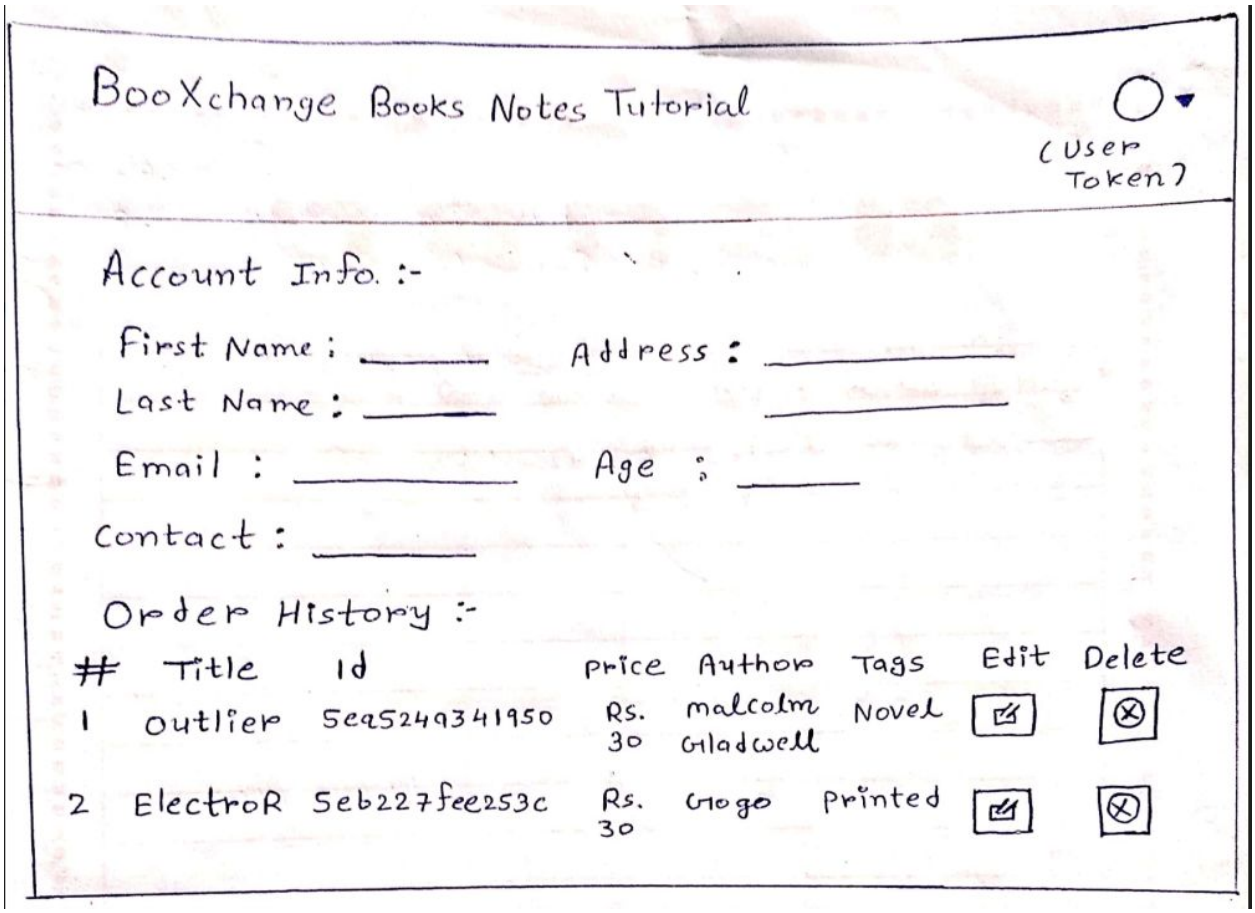
6.3.2 **Cart Window**



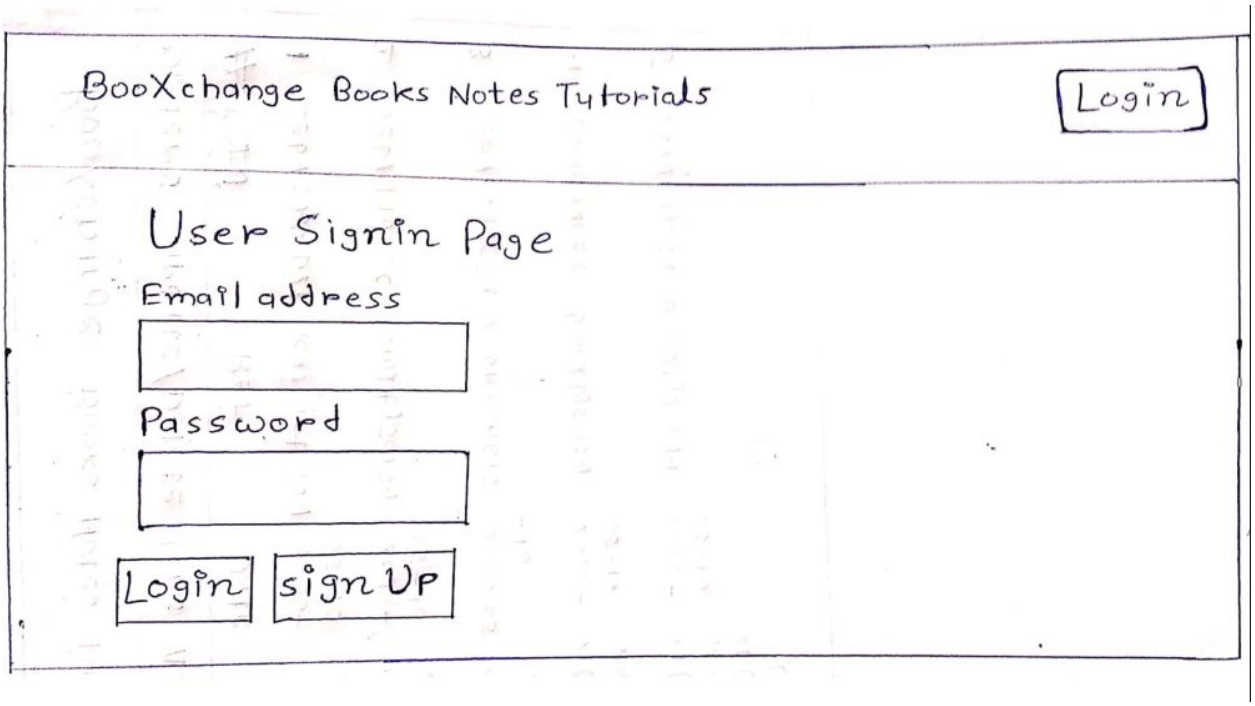
6.3.3 **Checkout Window**



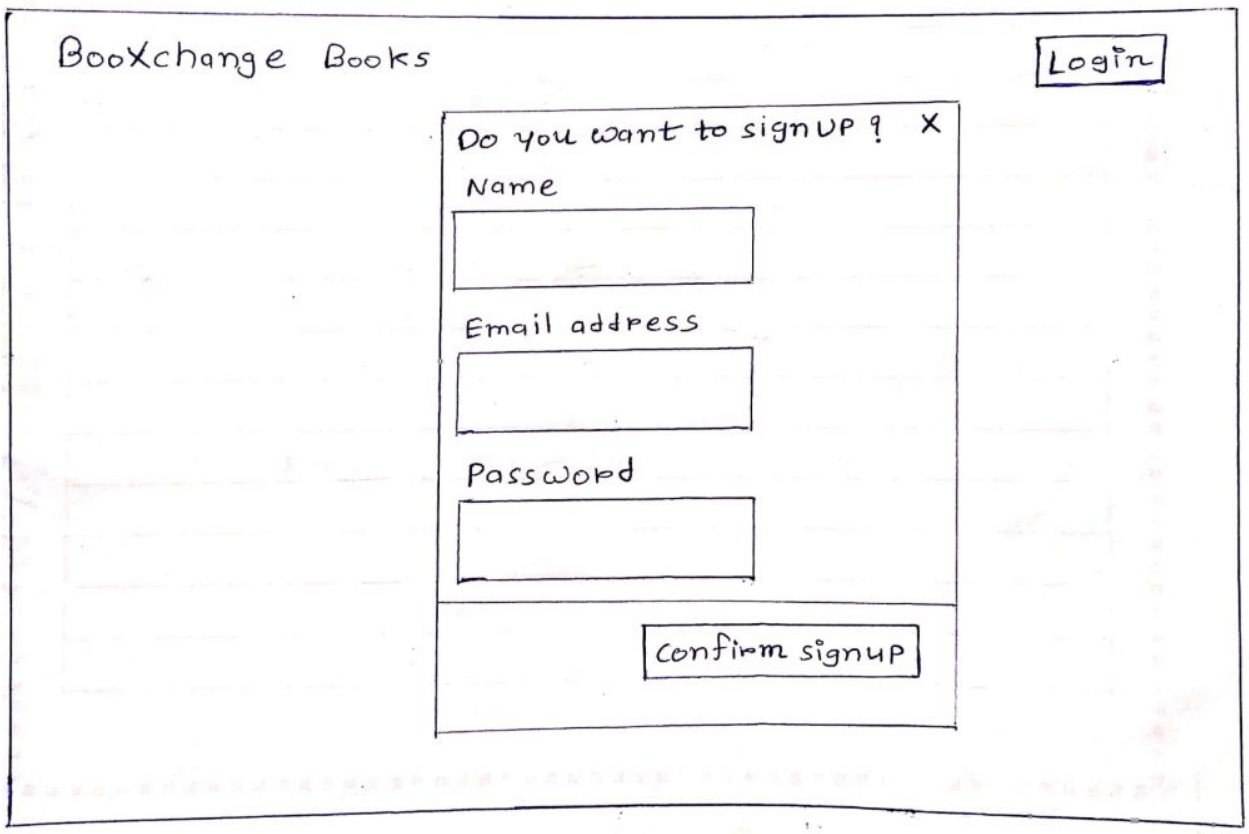
6.3.4 **User Detail and Order History Window**



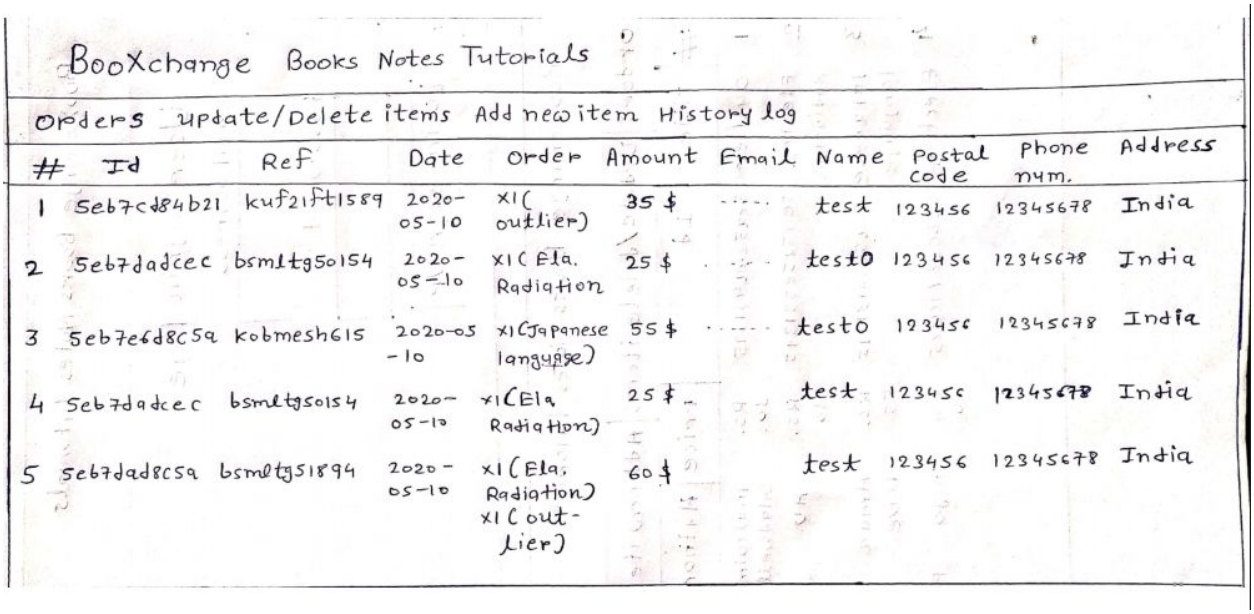
6.3.5 **Login Window**



6.3.6 **Signup Window**



6.3.7 **Order Window (Admin Side)**

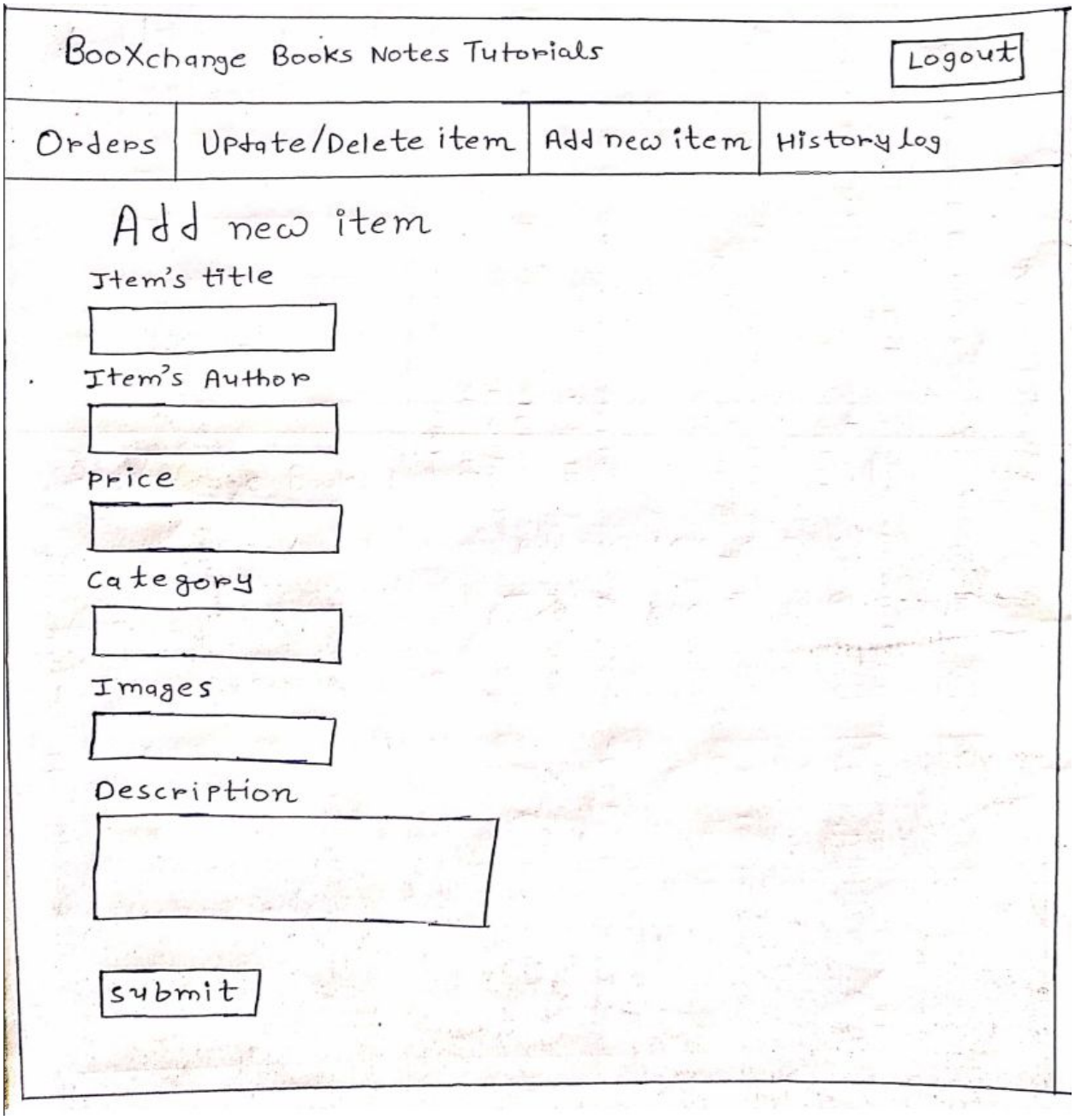


BooXchange Books Notes Tutorials

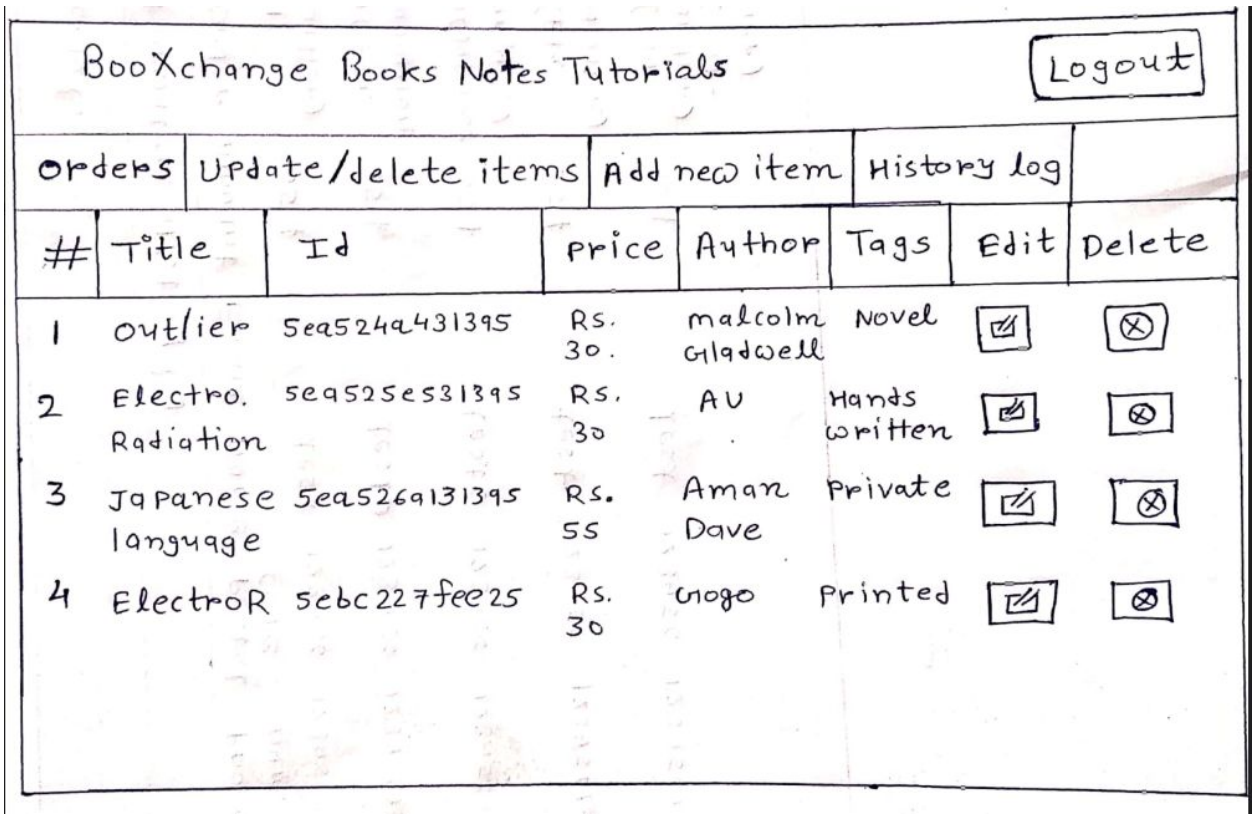
Orders update/Delete items Add new item History log

#	Id	Ref	Date	Order	Amount	Email	Name	Postal code	Phone num.	Address
1	5eb7cd84b21	kuf21ft1589	2020-05-10	x1(outlier)	35 $	.....	test	123456	12345678	India
2	5eb7dadcec	bsmltg50154	2020-05-10	x1(Ela. Radiation	25 $	.....	test0	123456	12345678	India
3	5eb7e6d8c5a	kobmesh615	2020-05-10	x1(Japanese language)	55 $	.....	test0	123456	12345678	India
4	5eb7dadcec	bsmltg50154	2020-05-10	x1(Ela Radiation)	25 $	.....	test	123456	12345678	India
5	5eb7dad8c5a	bsmltg51894	2020-05-10	x1(Ela. Radiation) x1(out-lier)	60 $	.....	test	123456	12345678	India

6.3.8 **Add New Item (Admin Side)**



6.3.9 **Update/Delete Window (Admin Side)**



6.3.10 **History Window**