CROSS PLATFORM BOOKSTORE MANAGEMENT SYSTEM

HEVAR HOSHANG AHMED

UNIVERSITI TEKNOLOGI MALAYSIA

UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author's full name : HEVAR HOSHANG AHMED

Date of Birth : 9/12/2000

Title : Cross Platform Bookstore Management System

Academic Session :

I declare that this thesis is classified as:

☐	**CONFIDENTIAL**	(Contains confidential information under the Official Secret Act 1972)*
☐	**RESTRICTED**	(Contains restricted information as specified by the organization where research was done)*
☑	**OPEN ACCESS**	I agree that my thesis to be published as online open access (full text)

1. I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:
2. The thesis is the property of Universiti Teknologi Malaysia
3. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
4. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

SIGNATURE OF STUDENT

QU18SCSJ026
MATRIX NUMBER
Date: 14 FEB 2022

SIGNATURE OF SUPERVISOR

Abdul Salam Abdulla
NAME OF SUPERVISOR
Date: 14 FEB 2022

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

May 2023

QIU Library

Sir,

CLASSIFICATION OF THESIS AS OPEN
CROSS PLATFORM BOOKSTORE MANAGEMENT SYSTEM
HEVAR HOSHANG AHMED

Please be informed that the above-mentioned thesis entitled “CROSS PLATFORM BOOKSTORE MANAGEMENT SYSTEM” be classified as OPEN ACCESS.

Thank you.

Sincerely yours.

Abdul Salam Abdulla, As Sulaymaniyah Iraq, +964 770 547 2685

"I hereby declare that we have read this thesis and in my
opinion this thesis is suffcient in term of scope and quality for the
award of the degree of Bachelor of Computer Science (Software Engineering)"

Signature	:	Salam
Name of Supervisor	:	ABDUL SALAM ABDULLA
Date	:	14 FEBRUARY 2022

CROSS PLATFORM BOOKSTORE MANAGEMENT SYSTEM

HEVAR HOSHANG AHMED

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Bachelor of Computer Science (Software Engineering)

School of Computing
Faculty of Engineering
Universiti Teknologi Malaysia

FEB 2022

DECLARATION

I declare that this thesis entitled *"CROSS PLATFORM BOOKSTORE MANAGEMENT SYSTEM "* is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	 .....
Name	:	HEVAR HOSHANG AHMED
Date	:	14 FEBRUARY 2022

DEDICATION

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Professor Dr. Abdulsalam, for encouragement, guidance, critics and friendship. I am also very thankful to my co-supervisor Professor Dr Abdulasalam and Associate Professor Dr. Abdulsalam for their guidance, advices and motivation. Without their continued support and interest, this thesis would not have been the same as presented here.

My fellow student should also be recognised for their support. My sincere appreciation also extends to all my colleagues and others who have provided assistance at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all my family member.

ABSTRACT

The Cross Platform Bookstore Management System research aims to address the pressing need for a comprehensive digital platform for selling books online in Kurdistan while alleviating the difficulties faced by readers in locating specific books. In Chapter 1, the problem statement emphasizes the current absence of a suitable solution and highlights the challenges faced by administrators, owners, and readers. The chapter underscores the significance of developing a cross-platform bookstore management system that caters to the diverse needs of these stakeholders. Chapter 2 provides a detailed literature review examining the existing problems faced by bookstores in Kurdistan and exploring various systems and technologies employed to address these issues. Through case studies, the strengths and weaknesses of different systems, including manual operations and edition bookstores, are analyzed, offering insights into the industry and identifying gaps for improvement. Chapter 3 expounds upon the chosen Rapid Application Development (RAD) methodology, justifying its selection and describing the phases involved, such as requirements gathering, prototyping, rapid construction, and system implementation. The chapter discusses the technology stack employed, including SQL Server for the database, .NET Core for back-end development, and Vue.js for front-end development, while shedding light on system requirements and employing UML diagrams for system design. Chapter 4 focuses on requirements analysis and design, highlighting their crucial role in creating implementable diagrams like class diagrams and entity-relationship diagrams (ERD). This chapter ensures a seamless transition from design to implementation, laying a solid foundation for subsequent project phases. In conclusion, this chapter-by-chapter abstract offers a comprehensive overview of the document's content, addressing the lack of a digital platform for book sales in Kurdistan and providing detailed summaries of the problem statement, literature review, methodology, and requirements analysis and design.

ABSTRAK

Penyelidikan Sistem Pengurusan Kedai Buku Cross Platform bertujuan untuk menangani keperluan mendesak untuk platform digital yang komprehensif untuk menjual buku dalam talian di Kurdistan sambil mengurangkan kesukaran yang dihadapi oleh pembaca dalam mencari buku tertentu. Dalam Bab 1, penyataan masalah menekankan ketiadaan penyelesaian yang sesuai dan menyerlahkan cabaran yang dihadapi oleh pentadbir, pemilik dan pembaca. Bab ini menggariskan kepentingan membangunkan sistem pengurusan kedai buku merentas platform yang memenuhi keperluan pelbagai pihak berkepentingan ini. Bab 2 menyediakan tinjauan literatur terperinci yang mengkaji masalah sedia ada yang dihadapi oleh kedai buku di Kurdistan dan meneroka pelbagai sistem dan teknologi yang digunakan untuk menangani isu ini. Melalui kajian kes, kekuatan dan kelemahan sistem yang berbeza, termasuk operasi manual dan kedai buku edisi, dianalisis, menawarkan pandangan tentang industri dan mengenal pasti jurang untuk penambahbaikan. Bab 3 menerangkan metodologi Pembangunan Aplikasi Rapid (RAD) yang dipilih, mewajarkan pemilihannya dan menerangkan fasa yang terlibat, seperti pengumpulan keperluan, prototaip, pembinaan pantas dan pelaksanaan sistem. Bab ini membincangkan tindanan teknologi yang digunakan, termasuk SQL Server untuk pangkalan data, .NET Core untuk pembangunan back-end dan Vue.js untuk pembangunan front-end, sambil memberi penerangan tentang keperluan sistem dan menggunakan rajah UML untuk reka bentuk sistem. Bab 4 memfokuskan pada analisis dan reka bentuk keperluan, menonjolkan peranan penting mereka dalam mencipta rajah boleh dilaksanakan seperti rajah kelas dan rajah hubungan entiti (ERD). Bab ini memastikan peralihan yang lancar daripada reka bentuk kepada pelaksanaan, meletakkan asas yang kukuh untuk fasa projek seterusnya. Kesimpulannya, abstrak bab demi bab ini menawarkan gambaran menyeluruh kandungan dokumen, menangani kekurangan platform digital untuk penjualan buku di Kurdistan dan menyediakan ringkasan terperinci tentang penyataan masalah, kajian literatur, metodologi, serta analisis dan reka bentuk keperluan. .

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE NO. **TITLE** **PAGE**

LIST OF APPENDICES

Chapter 1

Introduction

1.1. Introduction

In Kurdistan we have so many Book stores, but none of them have digital platform, that they can sell Books online, in other hand there are many readers have problems of way finding books they will check each book store one by one to find the specific book because we don't have a search engine platform that just by searching for the book, you would know where you can purchase the book that you want, the platform will be Cross platform that can work on any devices, it is important to add all the information of the book into the platform and work efficiently, so readers and owners will be satisfied with the platform.

so basically we have three stakeholders admin who manage the platform, the owners who have their own the management system, and the readers who can search for the books, so reader will search for specific book for example fire & blood it gives him those bookstores that have this book he can choose one of the bookstore and order it, so we also have owners screen they add all the books and their information in the database, so it tells them the price and the count of the book, so this platform **Cross Platform Bookstore Management System** help both user who are readers and owners.

1.2. Problem background

During the purchase of the Books, the Readers will inform the Owners of the title of the Book that he or she wishes to purchase, and the Owners will promote it to the Readers, and occasionally the Owners will mix up the charges and sell the Book at an exclusive rate, as the Owners may promote Books for 15 unique Readers, and they will forget to test on the Books and run out of the Book while unaware of this, or they may have the book and they don't know because of so many books that they have, and that would be a problem.

Some Readers want to purchase a specific type of book and go to a bookstore only to discover that the bookstore either does not have the book or is out of stock, causing them to waste time looking for the book, and some Readers live in remote areas where there are either no bookstores or bookstores that only sell specific types of books, requiring them to visit multiple locations in order to find the desired books.

So **CPBMS (Cross Platform Bookstore Management System)** can solve these problems by making some functions available to Readers and Owners, so in the system we will provide a dashboard that different Bookstore will have a database that they will add their data to the database and it tells them the price, title, author, year, categories, and this system is also a search engine system so Readers can Type the title of the books and it tells them which book store have this specific book They won't waste time searching in the incorrect bookstores in that situation.

1.3. Project aim

The aim of this project is to create a user-friendly, efficient, and simple Cross Platform Bookstore Management System that will assist owners in managing their bookstores by showing available books. Also, assist readers in finding the appropriate books and determining where they may acquire them.

1.4. Project Objectives

i. To improve accuracy and efficiency, as well as safety and communication.
ii. To make certain the supply of Books and keep the database of available alternatives to update the information of the books.
iii. To manage Book list and Book movement within the Bookstores, and make certain the fast searching, deleting, and updating of Books.
iv. To provide activate reaction to Readers' requests for Books with the details of the Books.
v. To provide a search engine that help Readers find specific books.
vi. To provide a platform that can be used in any devices.

1.5. Project Scope

The Cross Platform Bookstore Management System are considered as a pc machine that is used to control and save the Books within the Bookstores, and offer search engine that help Readers look for particular book. The mission scope will improve each the running environment of the Owners and make sure the Readers' pleasure. We propose these assignments to show beneficial for the bookstores owners and to make certain that excellent device is given to the Readers as well. We think that this task will help the Readers to take find their Books and purchase them without transferring an inch from their residence. We can summarize the scope into the subsequent points:

i. The technology will concentrate development efforts in Kurdistan’s bookstores.
ii. The technology provide cross platform that can be used in each devices.

1.6. Project Importance

This project's importance is derived from personal experience. My father has a bookstores and in summer I help him, so I work In bookstores before, so the problems that we had is when a customer came and ask for specific book I couldn’t know either we have that book or not, I always had to look for the books and it took me more than 10 minutes sometimes more or less to find the book, sometimes I was looking for a book that either does not have the book or is out of stock, in other hand Readers came to my father bookstore and told me that this is the fifth book store that I visit to find this specific book, and I always thought how much time they waste to find this book, so Cross Platform Bookstore Management System will solve these kind of problems and help both users.

1.7. Report Organization

Figure 1-1 Gantt Chart

Chapter 2

Literature Review

2.1. Introduction

This literature review discusses the current problems faced by bookstores in Kurdistan and the available systems and technologies used in bookstores to solve them. It covers the weaknesses and strengths of the approaches and techniques currently in use in bookstores.

2.2. Case Study (If any)

Most bookstores in Kurdistan do not have a database and internet browser to display their books, which makes it difficult for readers to find the books they want. There is also no search engine that can find a specific book in one of the bookstores. The manual operation system is used in bookstores that offer specific categories of books. The system has both advantages and disadvantages.

2.2.1. Company Organization Structure

This section will address the available systems that have been explored or progressed in the past; the discussion will contain the title of the article, a short description of their weaknesses and strengths, and in the end.

2.2.2. Manual Operation

This idea for bookstores that offer specific category of books, which are have just one category of the books, for example they just have religion books, or they just have self-growth books, has both pros and downsides.

The advantage is that it will assist readers in locating those bookstores which they can find any books of that category that they may order or purchase on their own, while also assisting owners in purchasing more books and benefiting from it.

The problem is that maybe some books doesn't have barcodes like most of the religion books, therefore you must enter the title of the book into the database to compute, or create their own barcode for each book and add the price, and also you can just purchase those books in that specific category.

2.3. Current System Analysis

This idea is for edition bookstores, those bookstores that they sell books by their own publisher company so it also has the advantages and disadvantages. The advantages are they already have the data of the books, so they won't struggle by typing all if the data again they just can send the data and that is all. The disadvantages are you can't find all books you want; you just can find those books that they published.

2.4. Comparison Between Existing Systems

This proposal is for a university shop that only sells curricular books for students. It has both advantages and disadvantages. The benefit is that those books are already mentioned, and they will know how many copies they should have since they know how many students should purchase

this specific book. The drawback for students who do not attend this institution is that the university can only give books for their own students.

2.5. Literature Review of Technologies Used

This section will go through the technologies utilized in this project, including which database, languages, and web server were used and why they were chosen.

The SQL Server database can be used as the database in this assignment as it offers comprehensive guide for every software improvement need SQL Server is a utility software program for Relational Database Management System (RDBMS), from Microsoft, that can be used for creating, keeping, managing, and enforcing the RDBMS systems. It is a considerably used application as it allows a couple of users simultaneously to paintings at the database systems, wherein customers can variety from minor office-primarily based machines to massive Internet-based servers. Provisions any kind of SQL programming extending from ANSI SQL (for classic SQL) through SQL to T-SQL (Transact-SQL) used for superior relational databases.

Back-end: will be .net core, .NET Core is another variant of .NET Framework, that is a free, open-source, well known cause improvement stage kept up with via Microsoft. A pass-stage structure runs on Windows, macOS, and Linux working frameworks. .NET Core Framework might be utilized to assemble exceptional kinds of bundles alongside versatile, PC, net, cloud, Iota, framework getting to be aware, micro services, diversion, etc. .NET Core is composed without any preparation to make it measured, light-weight, quick, and pass-stage Framework. It comprises of the middle elements which are expected to run a principal .NET Core application. Different elements are outfitted as Nugget programs, which you might add it in your product on a case-by-case basis. Thusly, the .NET Core utility accelerates the general exhibition; diminish the memory impression and transforms into simple to keep.

Front-end: will be views, Vue.Js is a JavaScript shape for constructing consumer interfaces (UIs) and single-website web page applications (SPAs). Only like the chic of them, Vue.Js is open-supply. It makes use of a version-view-viewmodel (MVVM) compositional instance. Planned via

Evan You and despatched off in 2014, the shape is an immediate reaction to your time enjoyed running with Angular at Google.

You desired to separate the pieces and quantities he inclined towards from Angular and make an extremely new anyway outstandingly light-weight structure. The MVVM structure considers the modern-day endeavour intent, or version, to be splendid from the graphical UI, or view. In impact, the differentiation offers a purposely herbal example for the front-forestall and again-give up designers to craftsmanship with. The viewmodel is the shape maintaining such difficulties unbiased, intending as a contact. Further, Vue.Js has a middle library that centres fine across the view layer. Any extra usefulness needs to be accumulated 8tilizing the assisting libraries of Vue.Js. However, this tooling is excluded, accordingly the descriptive phrase ‘innovative’. In any case, Vue.Js’ most severe incredible detail is its mandates. Mandates are HTML credit that allows builders to enlarge HTML, a primary age of pages that comes to a decision the construction of internet internet web page content material fabric.

2.6. Chapter Summary

This section describes the technologies used in a project, including the SQL Server database, .NET Core for back-end development, and Vue.js for front-end development. SQL Server is an RDBMS system used for creating and managing database systems. .NET Core is a free, open-source platform for building various types of applications, and Vue.js is an open-source JavaScript framework for building user interfaces and single-page applications. Vue.js uses an MVVM compositional pattern and has a core library that focuses on the view layer, and mandates are HTML attributes that allow developers to extend HTML and define the structure of web page content.

Chapter 3

Methodology

3.1. Introduction

This chapter will discuss the methodology that will be used on this project and its justification, the stages that will be included in the chosen technique with a brief description of each segment and the work that has been completed in that segment, the design modelling, design tools, Gantt chart, describe the equipment and technology used within the task, the machine requirements which include software program requirements and hardware requirements, and finally a bankruptcy.

3.2. Methodology Choice and Justification

Specifications are situation to trade at any time: as an end result, if we are facing any limitations along the road at the same time as running at the task, we may also make changes fast.

- Makes purchaser comments a priority and encourages it.
- The responses are set off.
- The time necessary for manufacturing has been greatly reduced.
- Implementation is not an problem as it's been planned for since the beginning of the undertaking.
- The gap among ideas and implementations is slim.
- When You Can Efficiently Evaluate Your Designs: RAD is super whilst you are not in the very last model and want to use the enter statistics that is accurate for your prototype which you created.
- When You Have a Strategy: When compared to the possibility, RAD is a price-powerful option. Styles and methodologies.
- When You Need a Task Completed Right Away: If you have got a good deadline to finish your task, RAD is the manner to move.

3.3. Phases of the Chosen Methodology

We have four most important degrees and one trendy phase. The principal levels are:

1. Gathering Requirement Phase
2. Prototype (User Design) Phase
3. Rapid Construction (System Development and Testing) Phase
4. Finishing the software program (System Implementation) Phase

And the general phase is: Phase I of the Investigation

3.4. Technology Used Description

As you can see from the data most of the readers want that the bookstore needs a new system for bookstores.

As you could see from the have a look at above, gadget readers, whether or not owners or renters, require the subsequent functions:

- Display the guide date.
- Display the book's description.
- Control the stock mechanically.
- Search for books.
- Managing a book listing
- Calculating books
- Be on the lookout if the inventory stage is turning into low.

During the interview, the book shop proprietors cautioned the following capabilities:

- Display the maximum famous and least famous books a good way to choose which novels to transmit extra of.
- The alarm machine is now related no longer simplest to the computer however additionally to the mobile cellphone, so if they are far away from the bookshop and the stock degree is low, they'll receive a message.

3.5. System Requirement Analysis

We use UML Diagram to show classes of the system and how it should work:

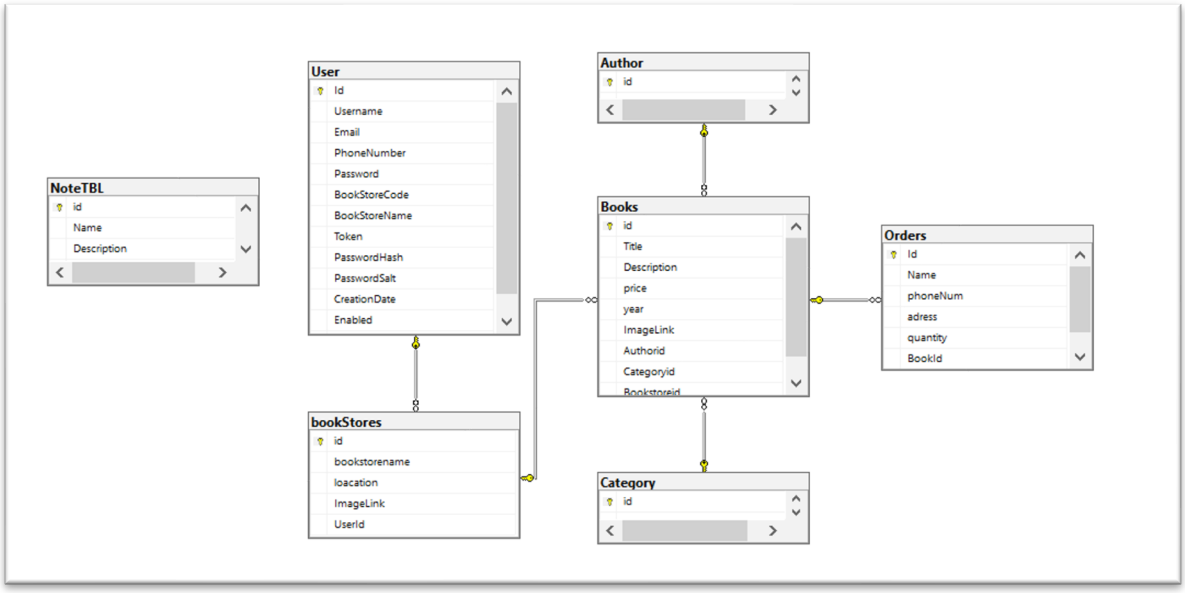

Figure 3-1 ERD

3.6. Chapter Summary

The methodology for the project is described in this section, including the stages involved, design modelling, design tools, Gantt chart, equipment and technology used, and software and hardware requirements. The chosen methodology is Rapid Application Development (RAD), which prioritizes customer feedback and allows for quick adjustments to specifications. The four main stages of the methodology are gathering requirements, prototyping, rapid construction, and finishing the software, with an additional investigation phase. The technology used includes features such as displaying book information, managing inventory, and sending alerts for low stock levels. Bookstore owners also suggested features such as displaying popular books and connecting the alarm system to mobile phones.

Chapter 4

Requirements Analysis and Design

4.1. Introduction

This chapter is most important chapter for this project that you make diagrams so later on you can implemented with code so in chapter I will make use case, sequence, activity diagrams, OPP class diagram, ERD with normalization, interface design and chapter summary.

4.2. Requirements Analysis

- Use case

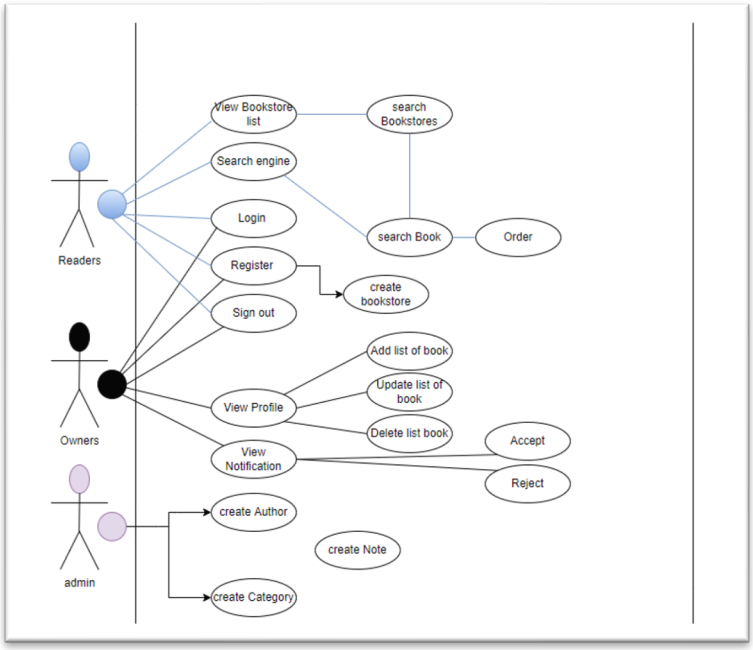

Figure 4-1 Use case diagram

- Sequence diagram

- Readers:

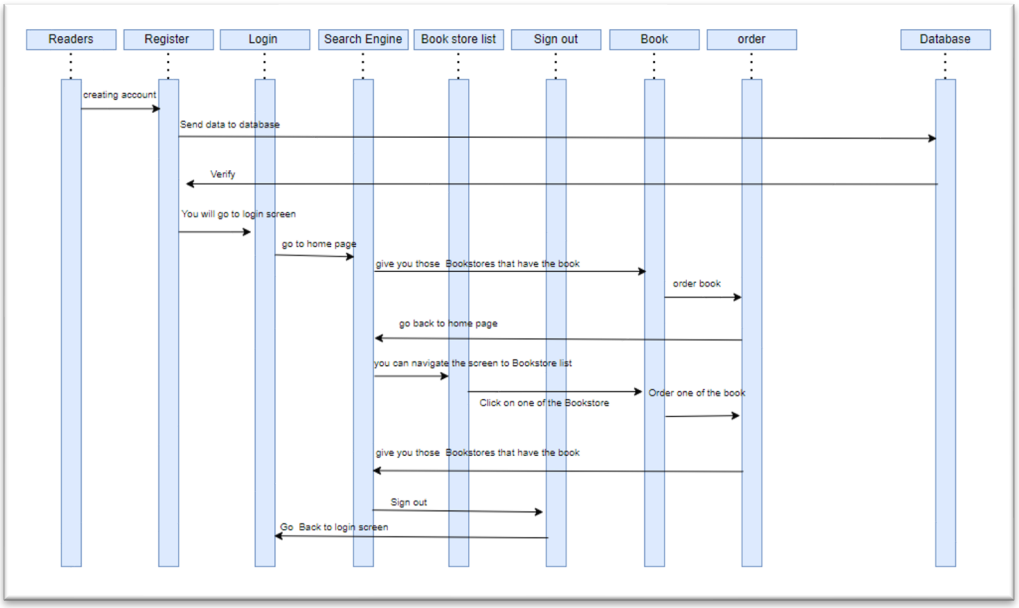

Figure 4-2 Sequence diagram for readers

- Owners:

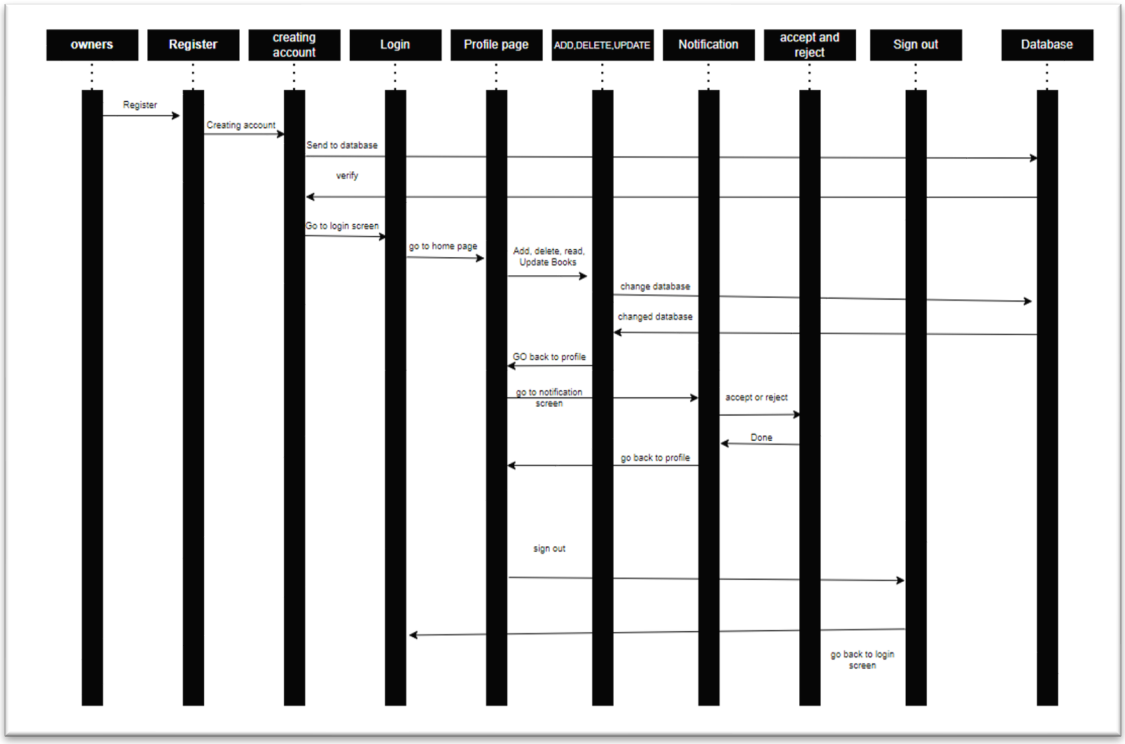

Figure 4-3 Sequence diagram for owners

- Admin

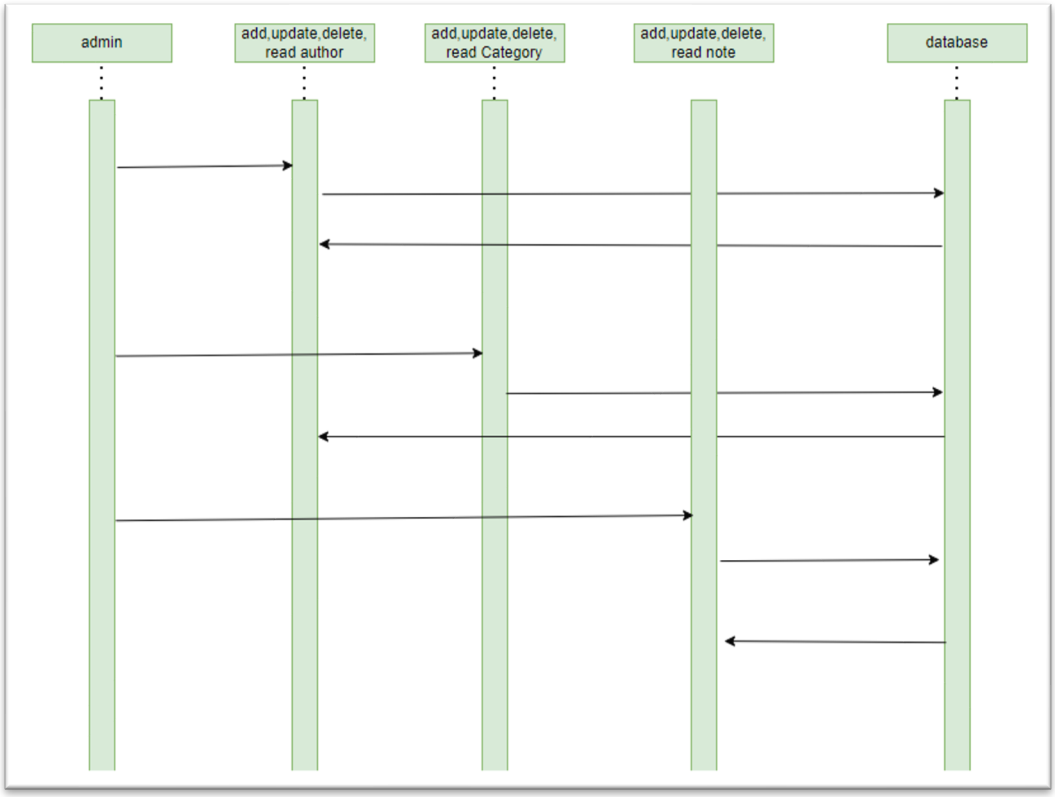

Figure 4-4 Sequence diagram for admin

- Activity diagram

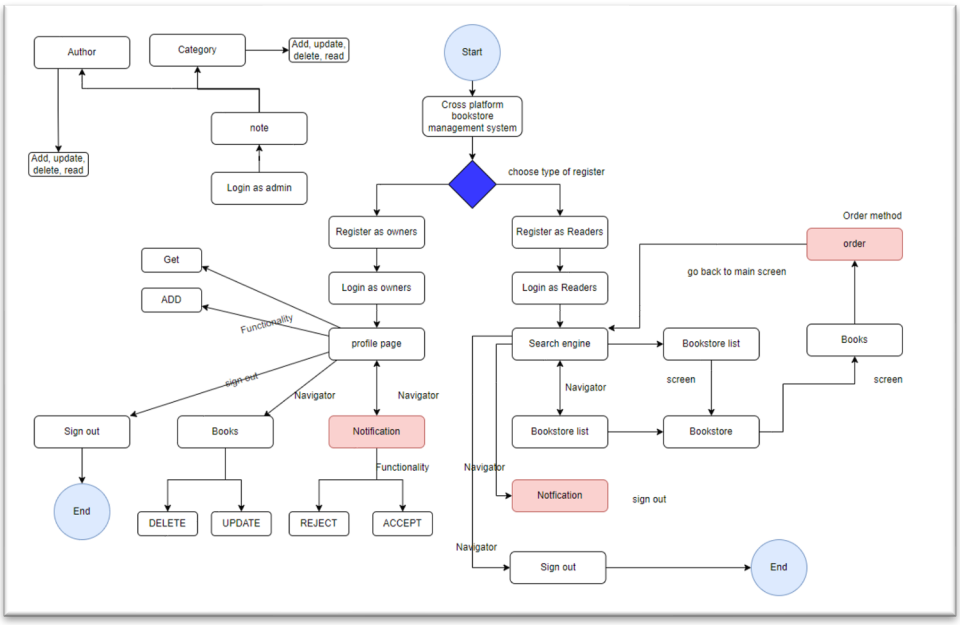

Figure 4-5 Activity diagram

4.3. Project Design

OOP class diagram

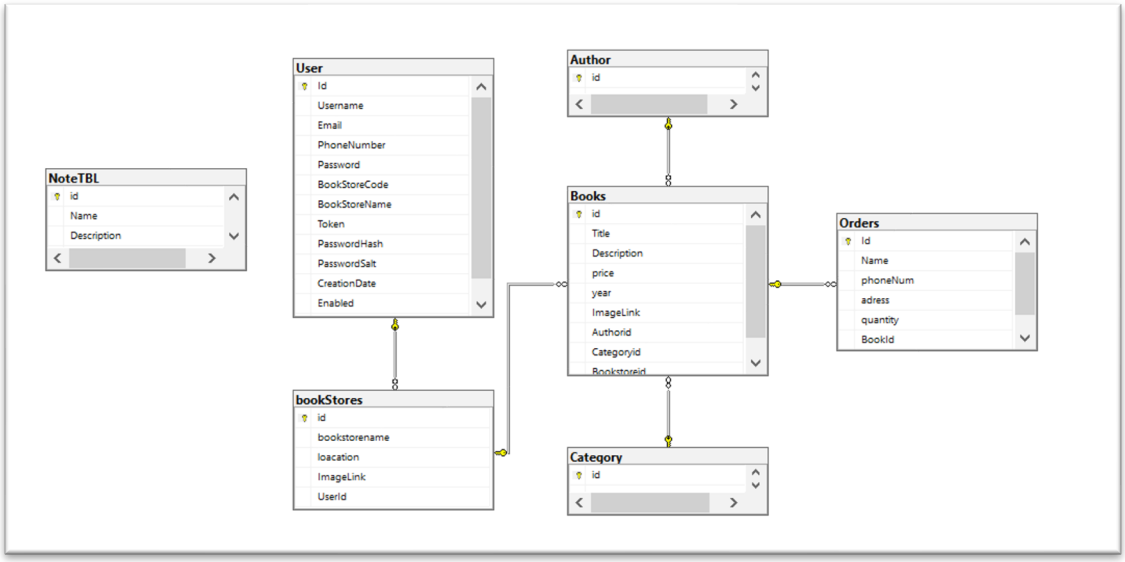

Figure 4-6 OOP Class diagram

4.4. Database Design

ERD

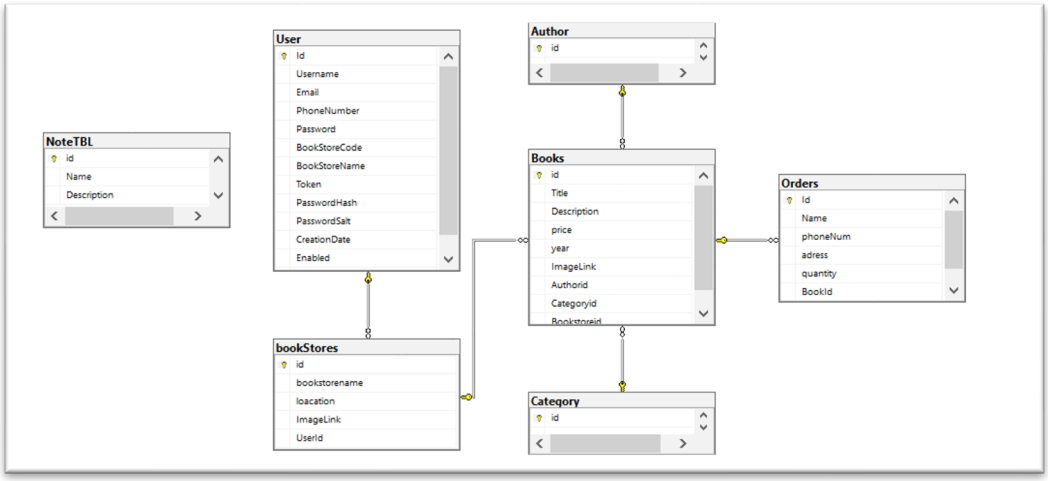

Figure 4-7 ERD

4.5. Interface Design

Readers:

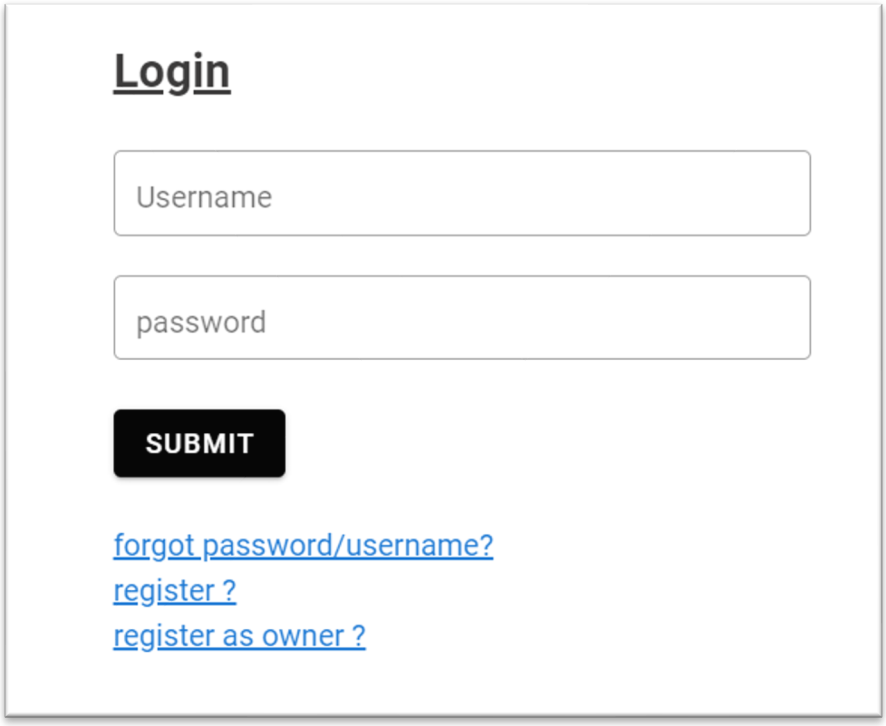

Figure 4-8 Login page

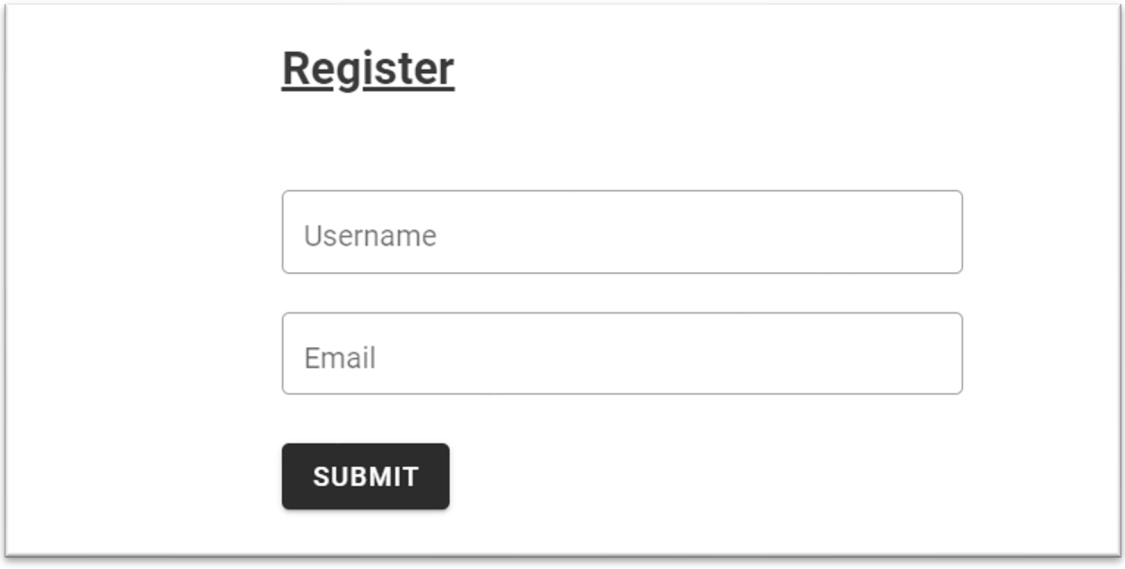

Figure 4-9 Register as a Reader

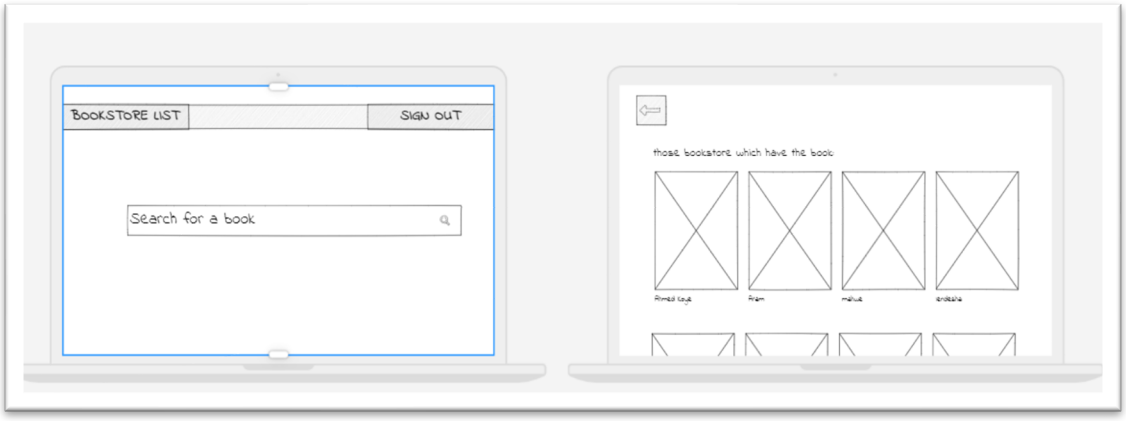

Figure 4-10 Search engine screen

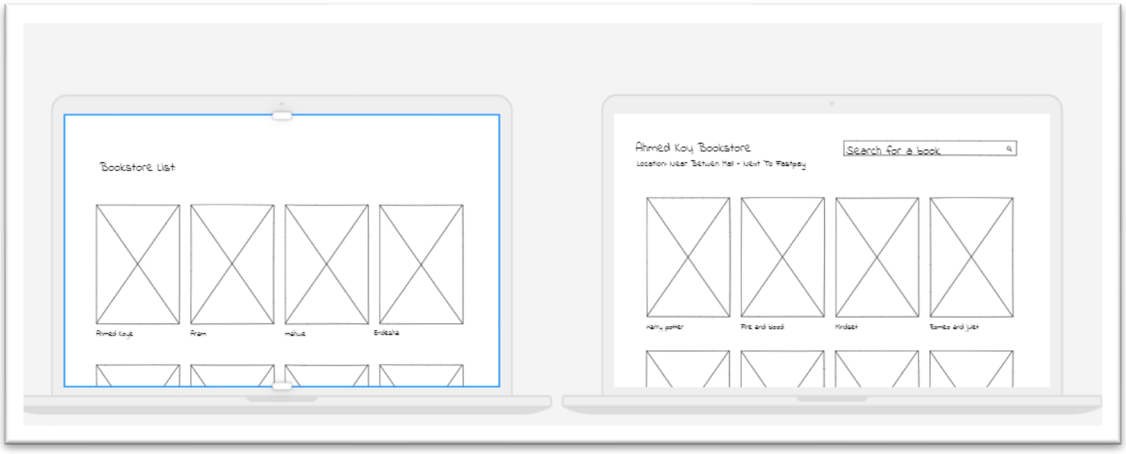

Figure 4-11 ERD

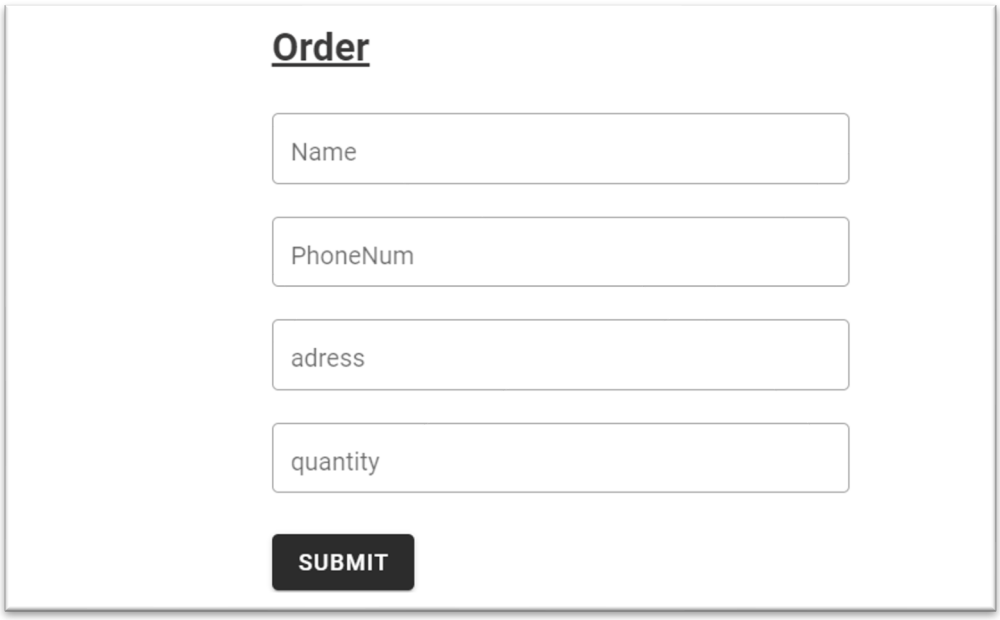

Figure 4-12 Order and payment

Owners:

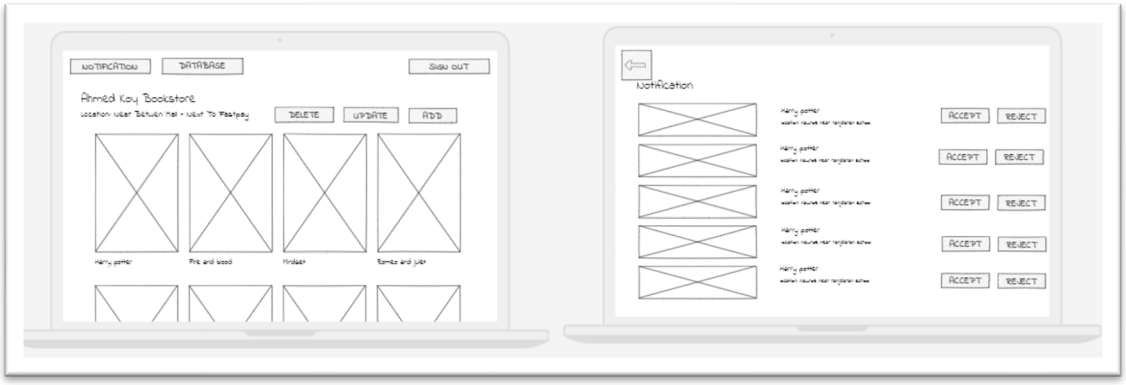

Figure 4-13 profile and notification

Admin:

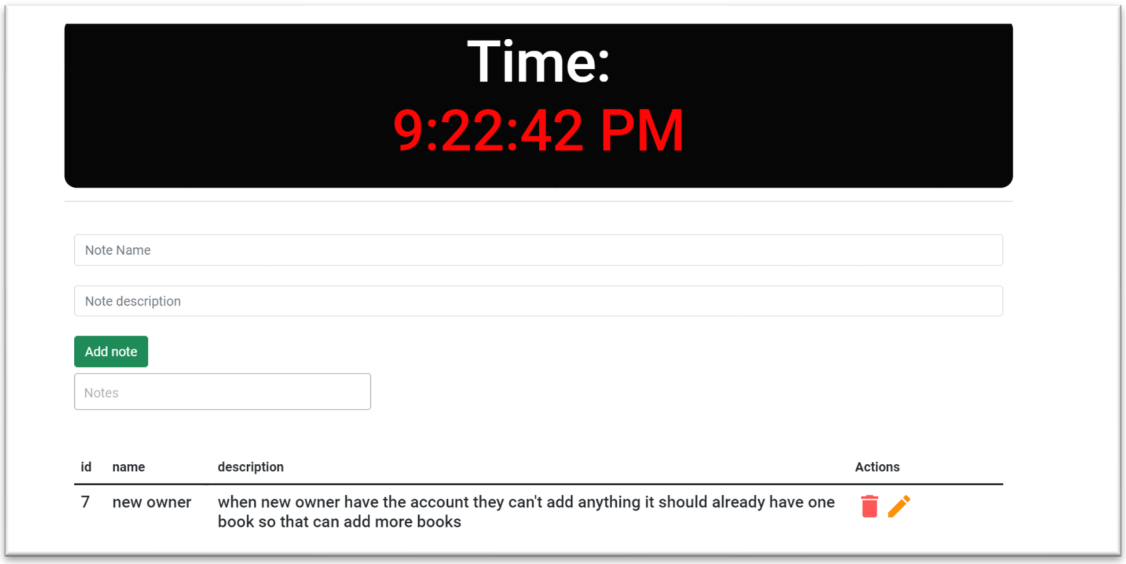

Figure 4-14 Note Page

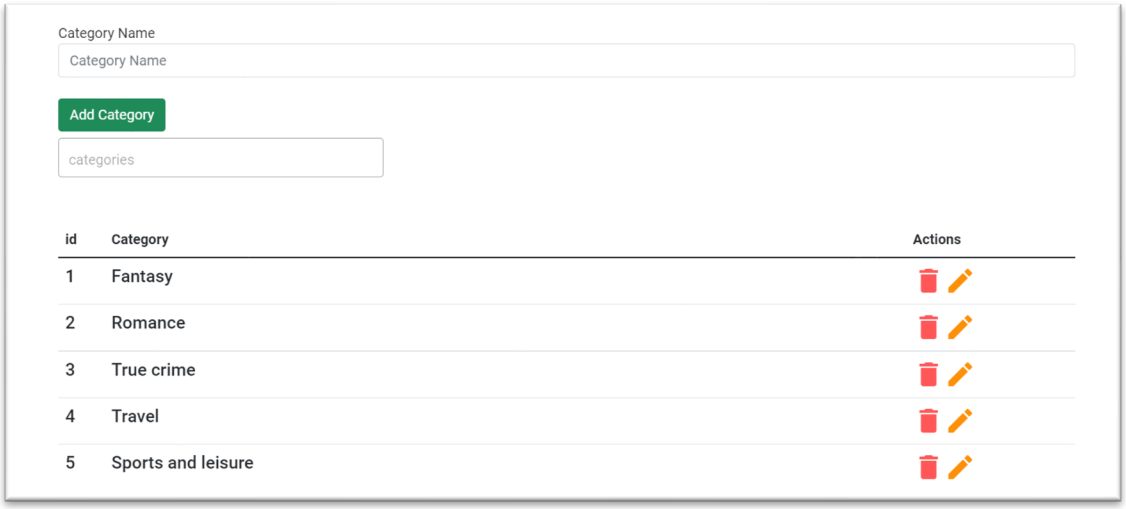

Figure 4-15 category page

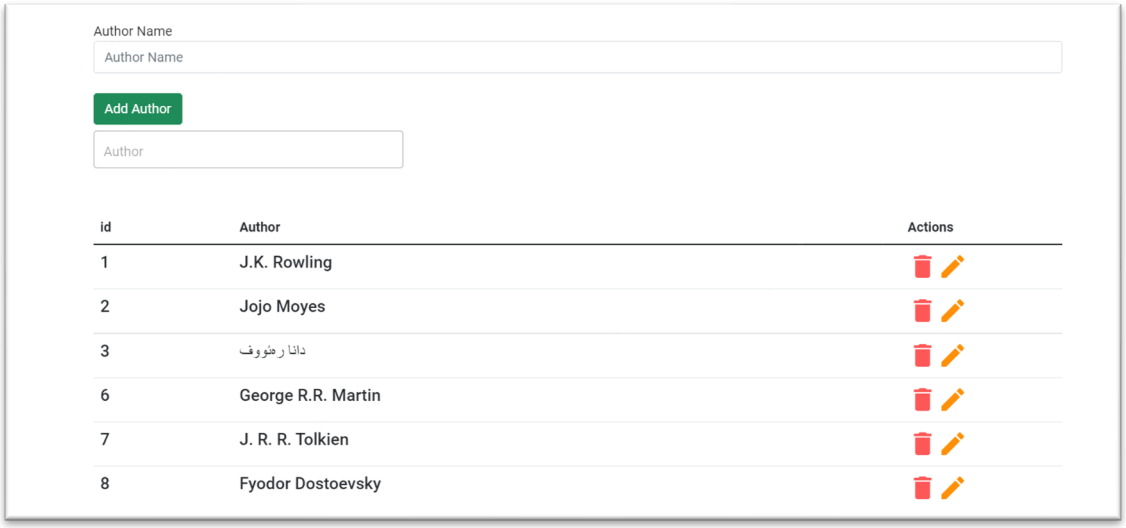

Figure 4-16 author page

4.6. Chapter Summary

In this chapter I made all of the designee that we need (use case, sequence, activity diagram, OOP class, ERD and interface design) and in project 2 we will implement all of those things with code.

Chapter 5

Implementation, Testing, Results, and Discussion

5.1. Introduction

In this final chapter of the final year project, which is the general conclusion and future works of the final year project, the PSM2 implements the Cross Platform bookstore management system. So let's talk about my project generally, let's start how I implement the code, so first I choose sql server for my database which is sql database, and It's easy to use and I can add data manually, so database is hard for some user to understand so they will create an interface to make it more easy to use in the same time publishing in world wide web in the same time, but front-end can't add data to database so there is a something between database and front-end which is back-end whc I used .net core, so back end can help front-end to add data to the database, so how I connect back-end with the database. In apps setting there is place that you can connect your sql server with back-end.

```
"DefaultConnection": "Data Source=name of source\\SQLEXPRESS;database=name of your database;Trusted_Connection=True"
```

So like that you connect your database with the back end and then you will make modules and you should again name those modules in data context for the name of the database like that

```
public DbSet<User> User { get; set; }
```

So now you are good to go to create controller and api. so you are good to go to create interface, which I used vue.js it's like react but in easy way so I used back end API and implemented in my interface and my website is ready to publish

5.2. Coding of System Main Functions

The version, the view, and the controller are the three main logical elements that make up an application according to the MVC architectural pattern. These parts are made with a certain application development factor in mind. MVC is a well-liked web development framework for making scalable and extendable activities that is on-trend in the business world.

All of the user's record-related logic is represented by the Model component. This could be some other enterprise good judgment-related information or the data that is being passed between the View and Controller components. A Customer object, for example, will collect customer information from the database, manage it, and update its records once again to the database or utilize it to render records.

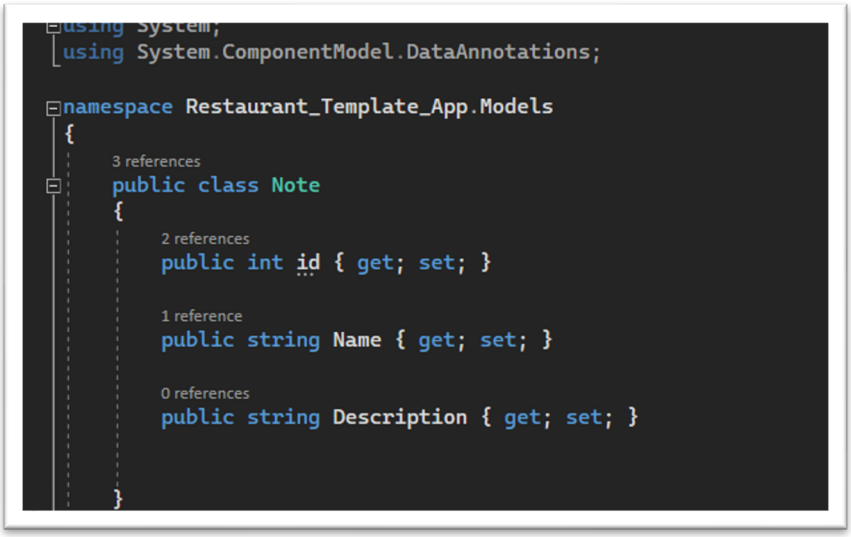

Figure 5-1 Note model

Model and view components are connected by controllers, which process all incoming requests and business logic, keep track of how the model component is being used, and work with views to provide the output. For instance, using the Customer Model to update the database, the Customer controller will handle all interactions and inputs from the Customer View. To access the customer information, the same controller will be employed.

```
[ApiController]
1 reference
public class NotesController : ControllerBase
{
    private readonly DataContext _context;
    private readonly IMapper _mapper;

    0 references
    public NotesController(DataContext context, IMapper mapper)
    {
        _context = context;
        _mapper = mapper;
    }


    [HttpPost("PostNotes")]
    0 references
    public async Task<IActionResult> PostNotes(Note note)
    {
        await _context.NoteTBL.AddAsync(note);
        _context.SaveChanges();
        return Ok("Creadted");

    }
```

Figure 5-2 Note controller

All of the utility's common-sense UI is implemented using the View element. For instance, All of the UI will be visible in the customer view that text elements, for instance, that the user interacts with, dropdown menus, etc.

```
            type="text"
            class="form-control"
            placeholder="Note description"
          />

          <div>

          </div>
        </div>
        <div class="form-group">
          <br />
          <button
            @click="data.id > 0 ? updateNotes(data.id) : postNotes()"
            :class="data.id > 0 ? 'btn btn-warning' : 'btn btn-success'"
          >
            {{ data.id > 0 ? "update note" : "Add note" }}
          </button>
        </div>
        <br />
      </v-row>

      <v-row>
        <v-col md="6" lg="4" sm="12" cols="12">
          <v-text-field placeholder="Notes" outlined dense v-model="search">
          </v-text-field>
        </v-col>
      </v-row>
```

Figure 5-3 Note View

5.3. Interfaces of System Main Functions

So, what I talk about in coding of system main function was about coding so now I will show you what I make with those codes.

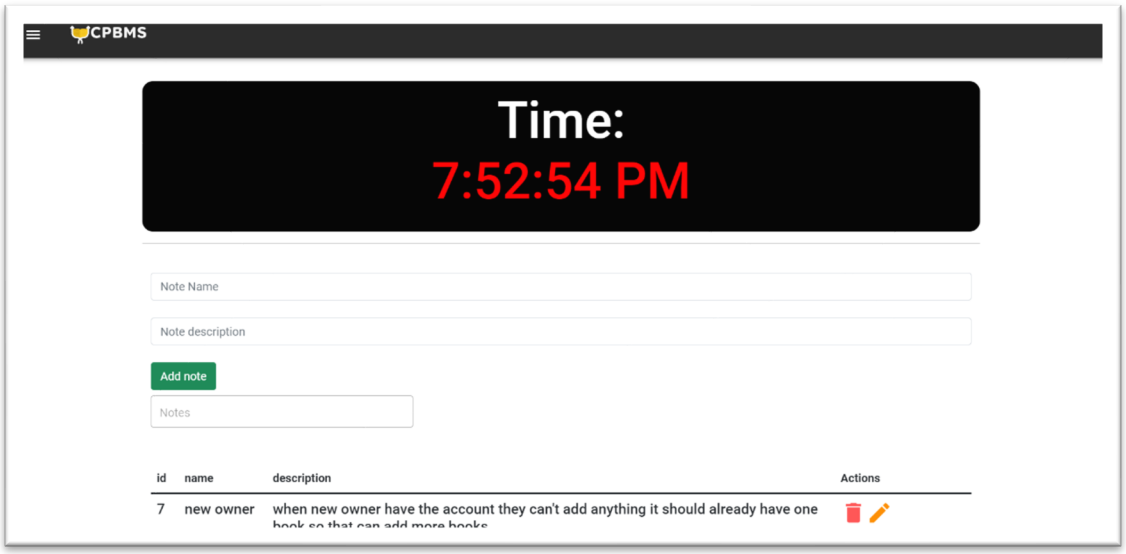

Figure 5-4 Note Result

5.4. Testing

Software testing is a way to determine whether or not the actual software product satisfies anticipated requirements and to ensure that it is error-free. It comprises the use of automated or manual equipment to execute software/device components in order to evaluate one or more potential homes. The goal of software program software testing is to identify errors, gaps, or missing requirements relative to actual needs.

Some people define testing as white box or black box testing Software testing, to put it simply, the application's verification is being tested (AUT). This software testing course explains the value of software testing to the target audience and introduces them to the process of trying out software programs.

5.5. Chapter Summary

This final chapter concludes the Cross Platform bookstore management system project, discussing its development using SQL server, .NET Core, and the MVC architectural pattern. It also showcases the interfaces of the system's main functions and emphasizes the importance of software testing to ensure the software product meets requirements.

Chapter 6

Conclusion

6.1. Introduction

Cross platform bookstore management system might be one of the greatest structures in Kurdistan I will work on this platform until it involves commercial enterprise and could assist all readers in Kurdistan this platform is so beneficial for readers and owners of bookstores if I speak about readers simply readers, can look for books and it tells them in which they can purchase the book and they can order it on the same time, in place of searching out books shop through shop and could not find the right book On the other hand, we are able to provide a database for bookstores proprietors for you to place all the records of the books and on the equal time control their own bookstores inside the internet site wherein they can add extra books, put off books, and update books so their bookstores turns into greater familiar to the customer.

Readers' difficulties are that they can't find the correct books until they waste so much time looking for books, and owners' problems are that they can't remember the names of all of the books and may be unaware that some of the books are finished, but with a database, they will fix all of the problems.

6.2. Achievement of Project Objectives

They are so many platforms looks like mine in other countries but in Kurdistan mine is the first, my project is much bigger it’s not just one specific thing in this project I will provide everything for owners and readers and what they need.

Basically, I finished two objectives out of 6 Objective:

- Provide activate reaction to Readers' requests for Books with the details of the Books.
- Provide a search engine that help Readers find specific books.

6.3. Suggestions for Future Improvement

My future work will be to create an application for the website that will make it easier for most users to use the platform. Another idea is to provide a map for readers so that they can find the exact location of a bookstore and also to provide the shortest path of bookstores so that they can find bookstores more easily. Another idea is to create another account called Editions Company, which will have a connection between bookstore owners and readers.

REFERENCES

Cervone, H.F., 2007. Standard methodology in digital library project management. OCLC Systems & Services: International digital library perspectives

Mirghaderi, P., 2009. Online database inventory for bookstore management system (Doctoral dissertation, Universiti Teknologi Malaysia).

Makcen, J., 2014. Bookstore Management System (Doctoral dissertation, University of Cincinnati. College of Education, Criminal Justice, and Human Services).

Mu, Z. and Jiang, L., 2018, August. Online Bookstore Management System Based on Android. In 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS) (pp. 498-500). IEEE.

Delbridge, R. and Fisher, S., 2007. The use of soft systems methodology (SSM) in the management of library and information services: a review. Library Management.

Rowley, J.E., 1994. Aspects of a library systems methodology. Journal of information science, 20(1), pp.41-45.

Appendix A

Software Requirement Specification

Software Requirements Specification

Project Title

Cross platform bookstore management system

Version 1.0

Department and Faculty

Computer science (software engineering)

Revision Page

a. Overview

Describe the content of the current version.

b. Target Audience

State the targeted audience.

c. Project Team Members

List the team members and respective assigned module.

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	**Hevar Hoshang**		

Table of Contents

1. Introduction

1.1 Purpose

A Software Requirement Specification (SRS) is a document that outlines the purposeful and non-practical requirements of a software machine in an exquisite element. This Software Requirement Specification (SRS) is intended to define Book Store Management machine that I create a search engine that can find each book in each bookshop in Kurdistan It also includes use eventualities that display how the user and the device interact. This document includes various diagrams, along with a series diagram and a go with the flow chart and pastime diagram.

1.2 Scope

The Cross Platform Bookstore Management System are considered as a pc machine that is used to control and save the Books within the Bookstores, and offer search engine that help Readers look for particular book. The mission scope will improve each the running environment of the Owners and make sure the Readers' pleasure. We propose these assignments to show beneficial for the bookstores owners and to make certain that excellent device are given to the Readers as well. We think that this task will help the Readers to take find their Books and purchase them without transferring an inch from their residence. We can summarize the scope into the subsequent points:

- The technology will concentrate development efforts in Kurdistan's bookstores.
- The technology provides cross platform that can be used in each devices.

1.3 Definitions, Acronyms and Abbreviation

Table 1.1 Definition, Acronyms and Abbreviation

Acronym/ Abbreviation/ Term Definition

CPBMS	Cross platform Bookstore Management System

1.4 References

System Design Architecture Lecture slides on QIU eLearning

1.5 Overview

This Software Requirement Specification (SRS) is divided into three sections, the first of which covers the introduction and provides an outline of the entire SRS. The second component is a modern description of the tool, which includes an outline of the requirements that will limit how the device is built and operated. The 0.33 section is a detailed requirement that goes into great detail about the system specification.

2. Overall Description

The CPBMS is made up of five (7) modules which are:

1. Author
2. Book
3. Bookstore
4. Category
5. Note
6. Order
7. User

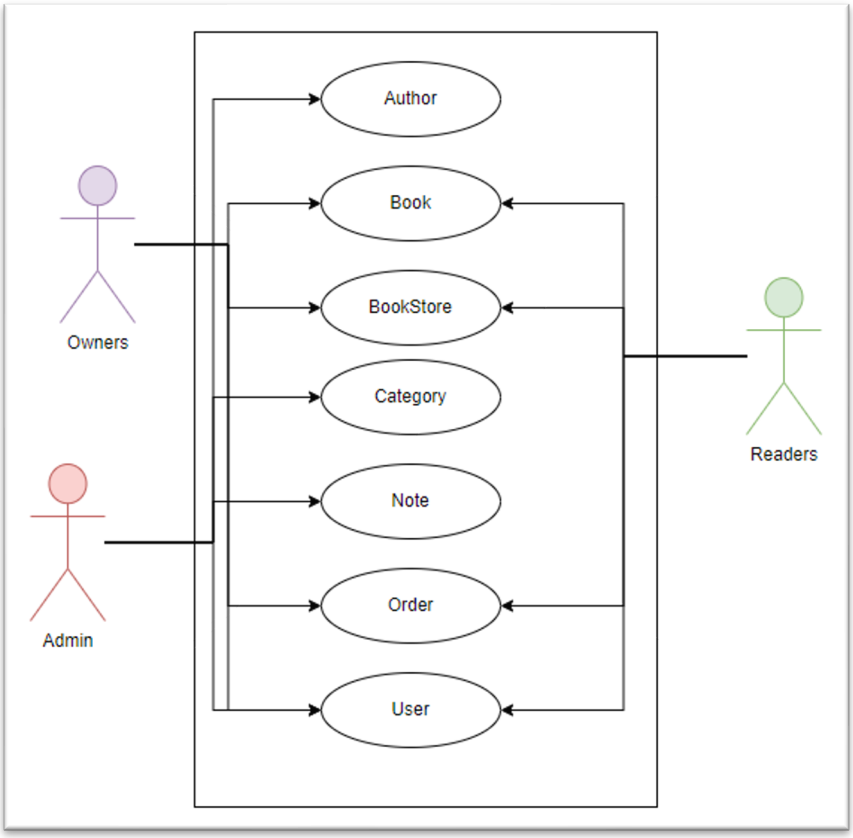

Figure 1.1 Use case Diagram of Cross Platform Bookstore Management system

2.1 Product Perspective

Cross Platform Bookstore Management System is a web-based system that made of for those readers who want to find books and visiting every bookstore in Kurdistan and owners can post their own books so could readers see it and order it, in the other hand admin will manage the system, and create note to not forget the tasks.

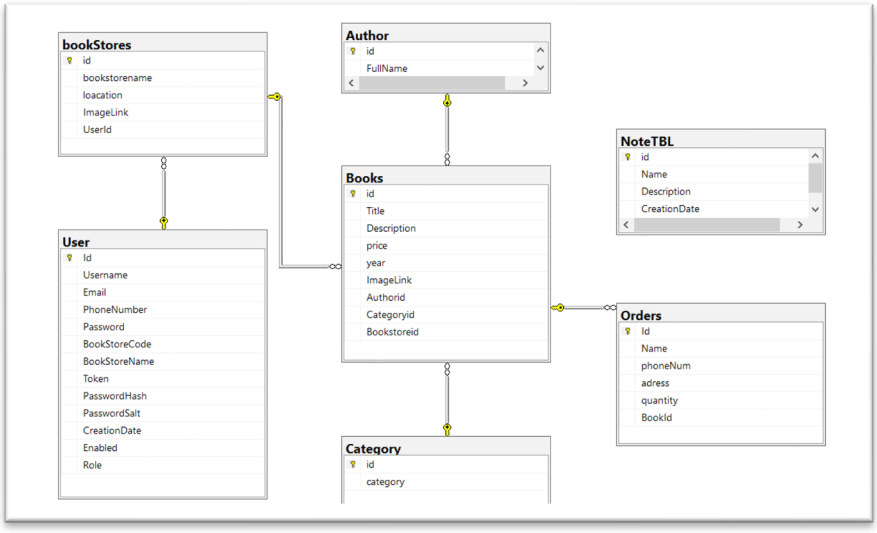

Figure 1.2 Entity Relationship Diagram for CPBMS

2.1.1 System Interfaces

I made 7 modules and I have 3 users each module do more than one functions basically readers and owners use these modules to make connection between them, and for the interface is very easy to use it's so simple and you can find things easily.

2.1.2 User Interfaces

Software: Device Specification:

1. OS Edition: Windows 10 Pro
2. Intergraded Development Environment: Visual Studio code, visual studio.
3. Database Management System: sql server
4. Framework: vue.js And .Net core
5. Web Browser: chrome
6. Visual Modelling & Design Tool: Draw.io
7. High Fidelity Prototype: balsamiq
8. Microsoft Power Point 2016: To create presentation slide
9. Microsoft Word 2010: To document project report, SRS and SDD

2.1.3 Hardware Interfaces

Minimum Requirements:

1. Windows 7 or later as the operating system
2. Intel Pentium 4 processor
3. Memory: 2GB minimum, 4GB preferred
4. Display Resolution: 1280x1024 or higher
5. Application window resolution: 1024x680 or higher
6. Internet access is required.
7. Input Method: Mouse

2.2 Product Functions

There are 7 use cases that represent the main functions performed by the proposed system:

Table 2.1 Use Case Table

No	Actor	Role
1	Author	Admin can create/update/remove/read author generally.
2	Book	Owner can add/update/remove/read books, readers can read books.
3	Bookstore	Admin can update/remove bookstores, owners can add/ read bookstores, readers can read bookstore
4	Category	Admin can create/update/remove/read category generally.
5	Note	Admin can add/update/remove/read note
6	Order	Readers can add order, owners can remove and read order.
7	user	Owners and readers can create account for theirself.

2.3 User Characteristics

In this system we have 3 users which they are admin readers and owner

Table 2.2 Software Requirements in process of developing the CPBMS

No	Actor	Role
1	Admin	Can add/remove/update/read authors, category, and author, In the other hand it can read/update/remove bookstores.
2	readers	It can read, bookstores, books, and can add users and order.
3	owner	It can add bookstore, it can add/remove/update/read books

2.4 Constraints

Table 2.3 Software Requirements in process of developing the CPBMS

Category	Software	Software Description
Operating System	Windows Operating System	Platform to run the CROSS PLATFORM BOOKSTORE MANAGEMENT SYSTEM
Source-code editor	Virtual Studio Code, Virtual Studio	Virtual studio code I used for vue.js and Virtual Studio I used for .net core.
Database	Sql server	I used for database and putting all of the data in sql sevrer.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Windows 10 pro was used for developing this project, and the code was implemented on VSCODE, VS and for the database sql server was used. The code is vue.js, .netcore and was run on chrome.

3.1.2 Hardware Interfaces

Fill in the blanks for Section 2. 1.3. Requirements Minimum:

1. Windows 7 or later as the operating system
2. Intel Pentium 4 processor
3. Memory: 2GB minimum, 4GB preferred
4. Display Resolution: 1280x1024 or higher
5. Application window resolution: 1024x680 or higher
7. Internet access is required.
8. Input Method: Mouse

3.2 System Features

3.2.1 Module <Crossplatformbookstoremanagmentsystem>

Briefly describe the module's functional requirements (use cases). It is preferable to include the diagram of the specific module (or the example of Customer Support System – via subsystem, see example below) in Figure 2.1 rather than the overall use case diagram.

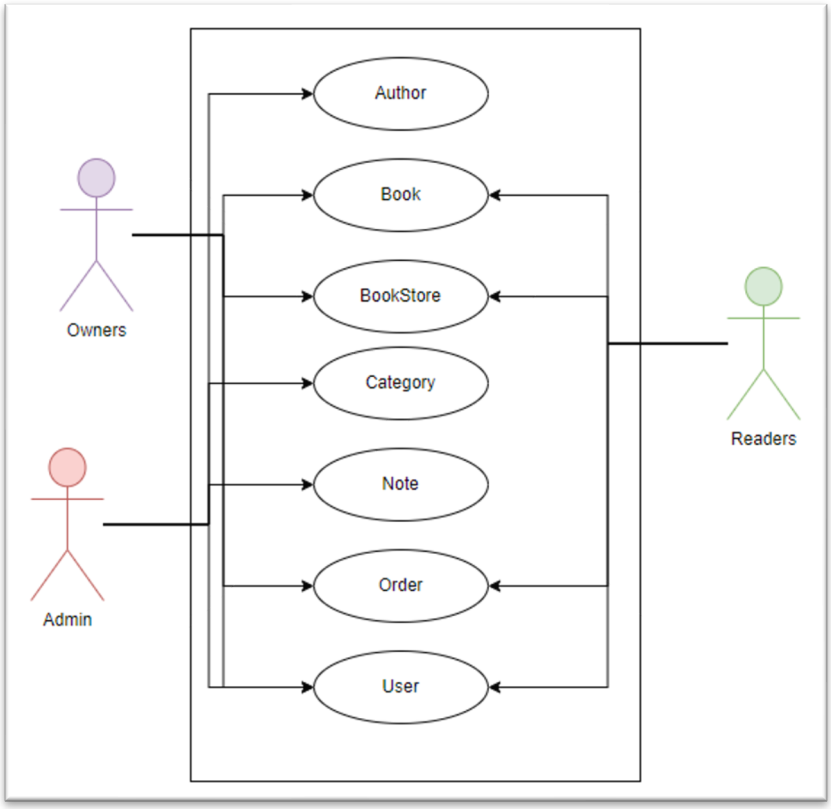

Figure 3.1 Use case Diagram of Employee Attendance Management System

3.2.1 UC001: Use Case <Author>

Table 3.1 Use Case Description for <Author>

User Case	**Author**
ID	UC01
Actors	Admin
Description	Admin can create/update/remove/read author generally.
Pre-Condition	Admin logged into the system and get authenticated
Normal Flow	1. The use case starts when the actor opens the system. 2. click Author in navbar 3. then you can add/remove/update/read in the same page
Alternative Flow	None
Exception Flow	None
Related Requirements	Possible Actions linked with other user case:
Post-Condition	Admin successfully can add remove update read author.

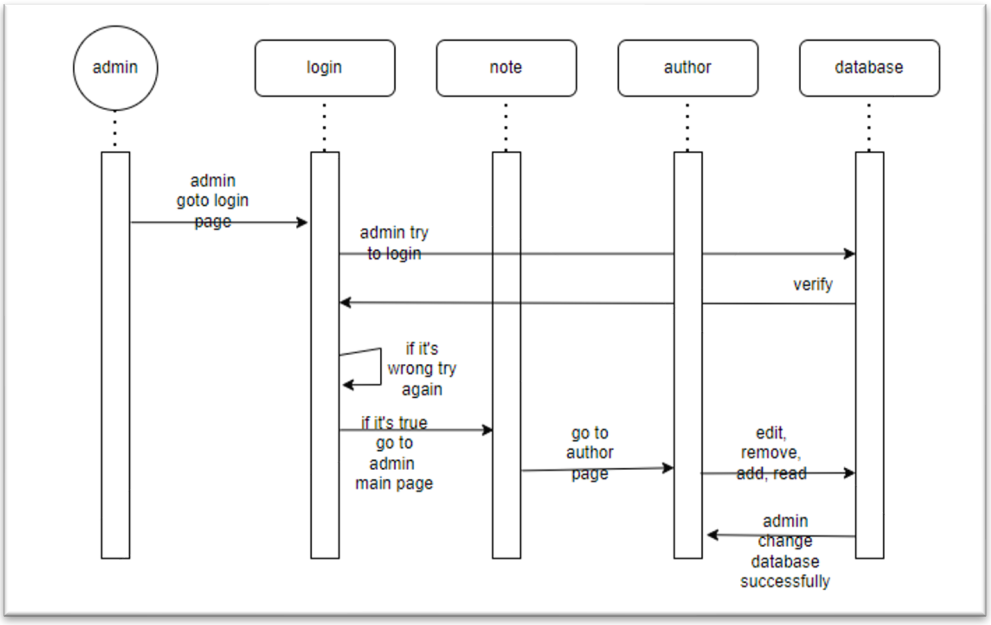

Figure 4.3 Sequence Diagram View for UC01 author Use case

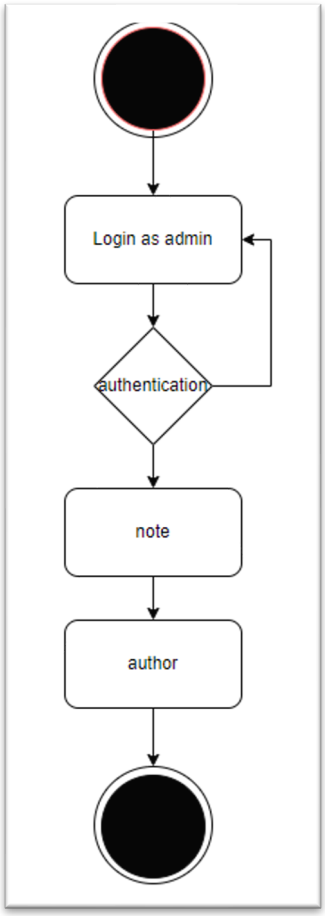

Figure 3.3 Activity Diagram for UC01 author

3.2.2 UC002: Use Case <Book>

Table 3.2 Use Case Description for <Book>

User Case	**Book**
ID	UC002
Actors	owners
Description	Owner can add/update/remove/read books, readers can read books.
Pre-Condition	owners logged into the system and get authenticated
Normal Flow	1. The use case starts when the actor opens the system. 2. You will add/read books in owner bookstore page. 3. Click of each books and you can remove and update
Alternative Flow	None
Exception Flow	None
Related Requirements	Possible Actions linked with other user case: -
Post-Condition	owners successfully add, remove, update, read books.

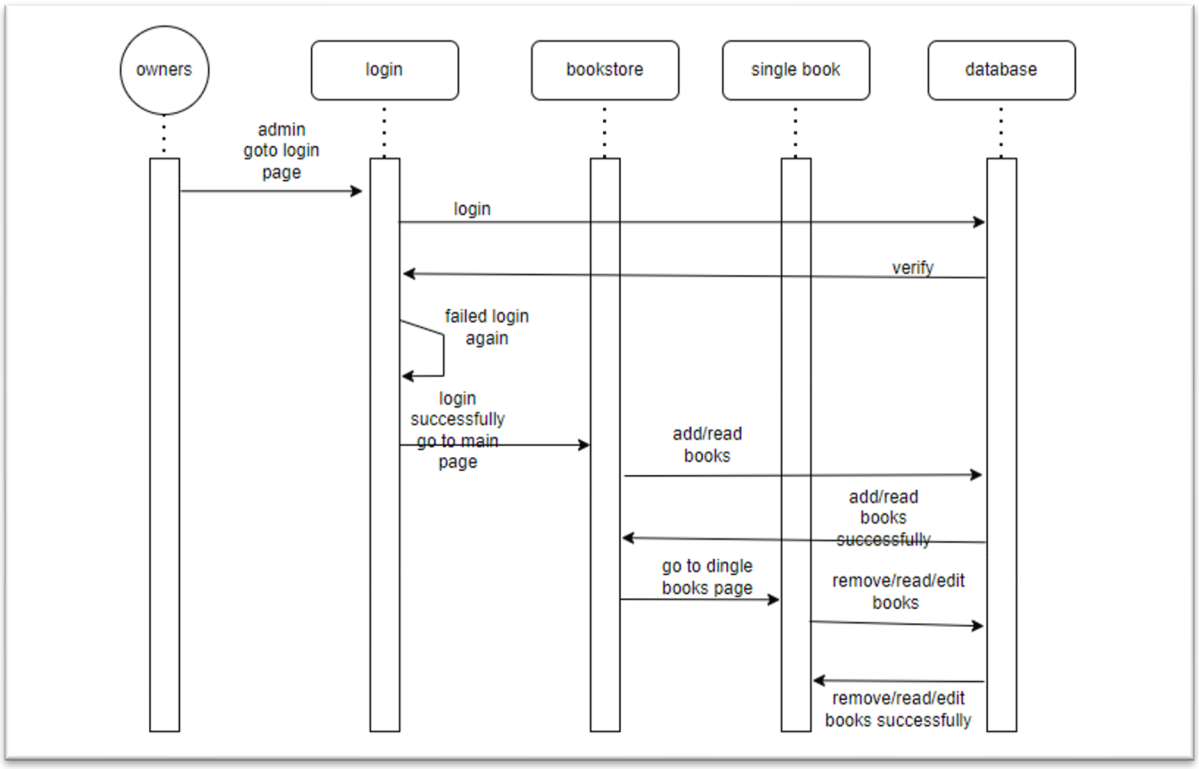

Figure 3.4 Sequence Diagram for UC002 books Use case

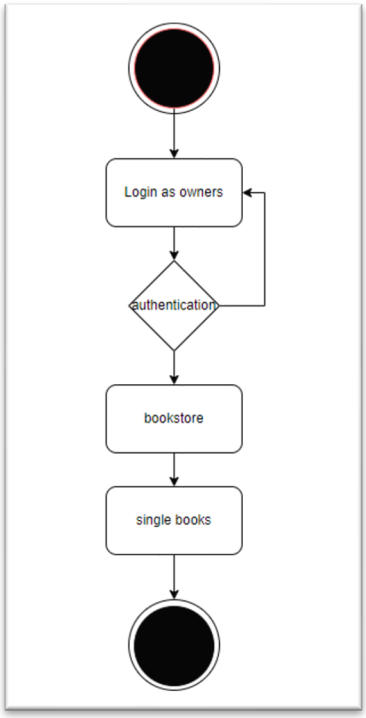

Figure 3.5 Activity Diagram for UC002 books Use case

3.2.3UC003: Use Case <Bookstore>

Table 3.3 Use Case Description for < Bookstore >

User Case	**Bookstore**
ID	UC003
Actors	Owners and admin
Description	Admin can update/remove bookstores; owners can add/ read bookstores.
Pre-Condition	1. Admin and owners logged into the system and get authenticated
Normal Flow	1. owners 1.1. Click on register bookstore 1.2. Register bookstore 1.3. Login as owners 1.4. In main page will read bookstore 2. Admin 2.1. Click bookstore on navbar 2.2. Update and remove bookstore
Alternative Flow	None
Exception Flow	None
Related Requirements	Possible Actions linked with other user case:
Post-Condition	Owners successfully add, read bookstore, and admin can update, remove successfully.

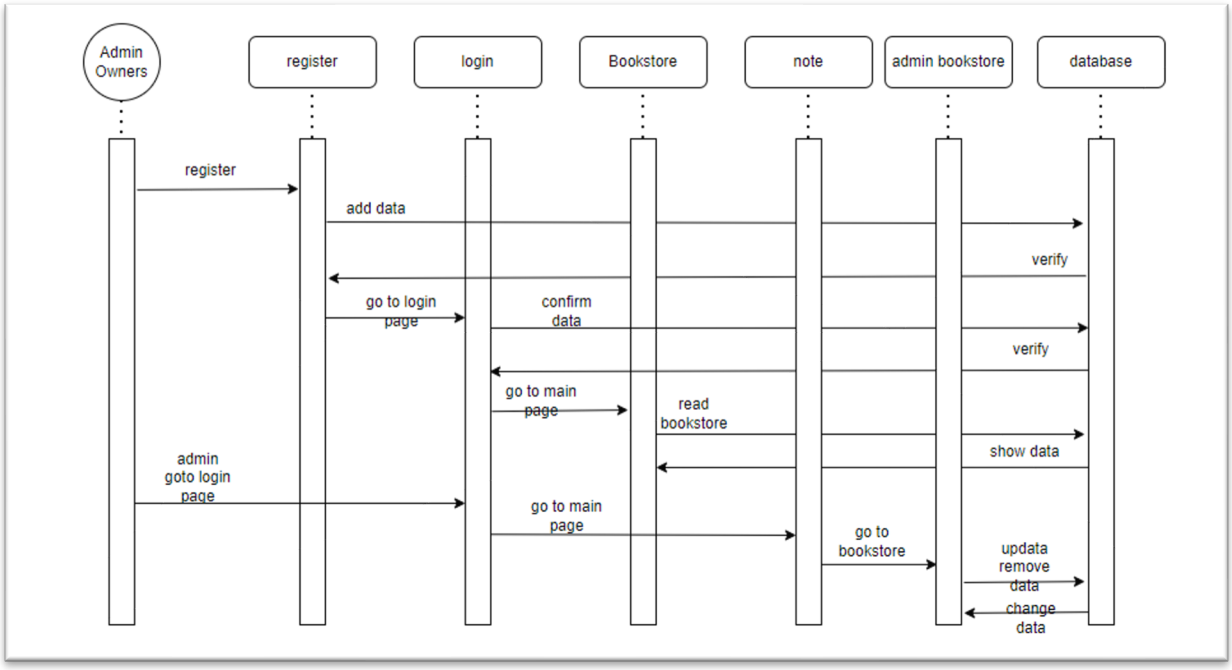

Figure 3.6 Sequence Diagram for UC003 bookstore Use case

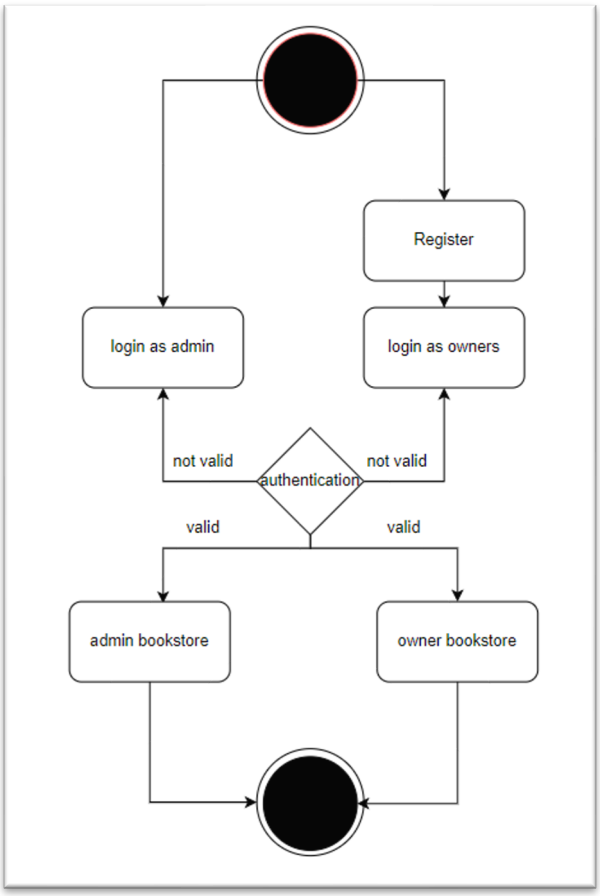

Figure 3.7 Activity Diagram for UC003 bookstore Use case

3.2.4 UC004: Use Case <category>

Table 3.4 Use Case Description for <category>

User Case	**Category**
ID	UC01
Actors	Admin
Description	Admin can create/update/remove/read category generally.
Pre-Condition	Admin logged into the system and get authenticated
Normal Flow	4. The use case starts when the actor opens the system. 5. click category in navbar 6. then you can add/remove/update/read in the same page
Alternative Flow	None
Exception Flow	None
Related Requirements	Possible Actions linked with other user case:
Post-Condition	Admin successfully can add remove update read category

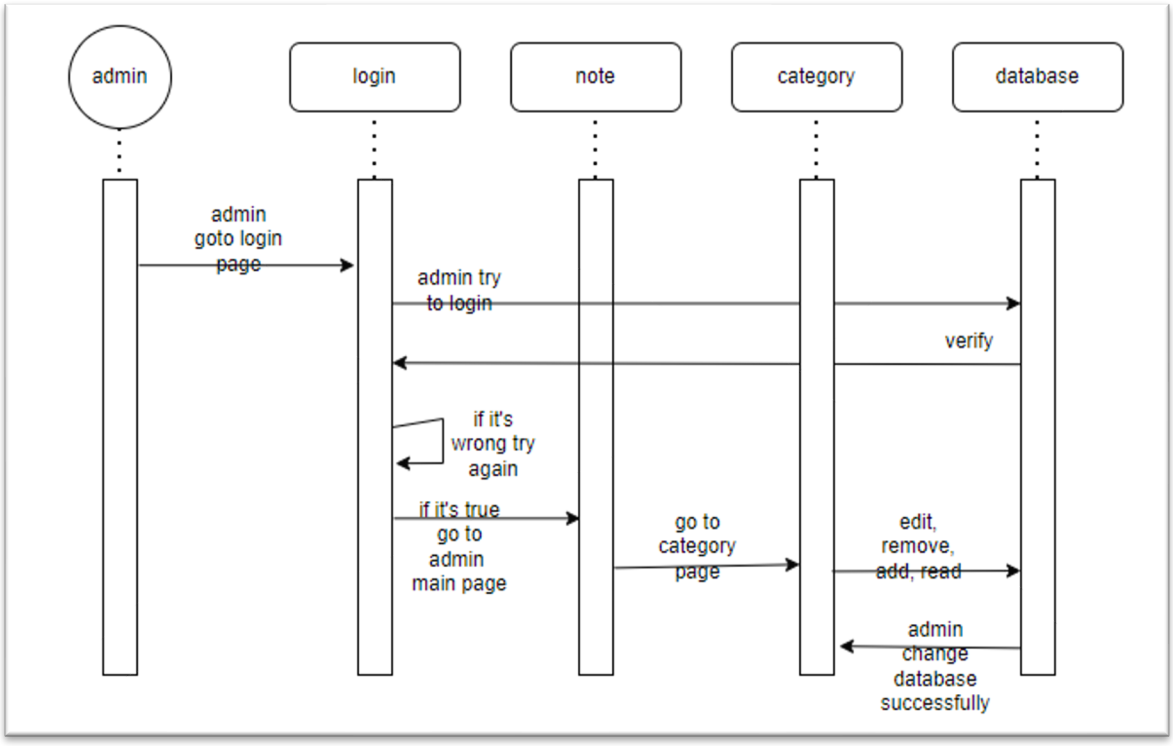

Figure 3.8 Sequence Diagram Update for UC004 category Use case

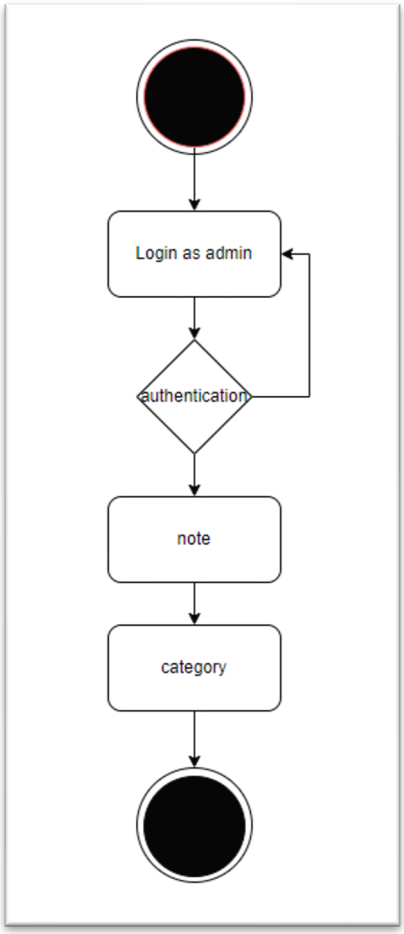

Figure 4.3 Sequence Diagram View for UC004 Update category Use case

3.2.5 UC005: Use Case <Note>

Table 3.5 Use Case Description for <Note>

User Case	**Note**
ID	UC005
Actors	Admin
Description	Admin can add/update/remove/read note
Pre-Condition	1. Admin logged into the system and get authenticated
Normal Flow	1. The use case starts when the actor opens the system. 2. In main page can add, remove, update, read note.
Alternative Flow	None
Exception Flow	None
Related Requirements	Possible Actions linked with other user case:
Post-Condition	Admin successfully add, remove, update, read notes.

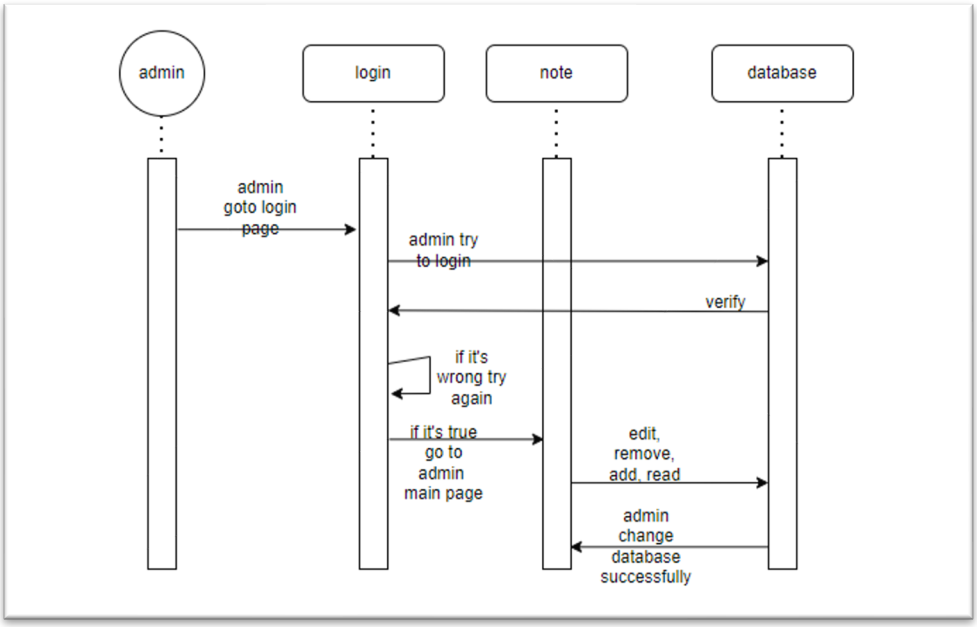

Figure 3.5 Sequence Diagram for UC005 Note Use case

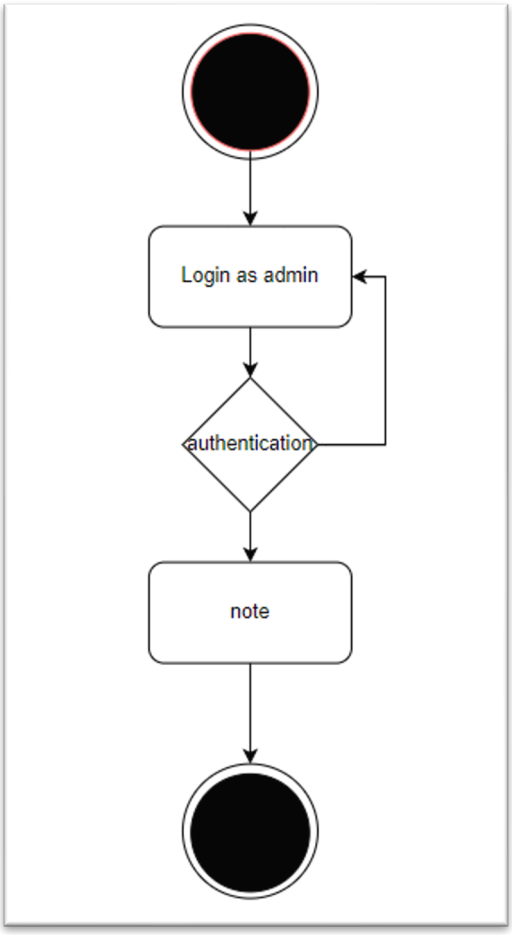

Figure 3.5 Activity Diagram for UC005 note Use case

3.2.6. UC006: Use Case <order>

Table 3.6 Use Case Description for <order>

User Case	**order**
ID	UC006
Actors	Readers and owners
Description	Readers can add order; owners can remove and read order.
Pre-Condition	1. Owners and readers logged into the system and get authenticated
Normal Flow	1. The use case starts when the actor opens the system. 2. Readers 2.1. Click of any books 2.2. Click on purchase 2.3. Click on order 3. Owners 3.1. Click on notification on the navbar
Alternative Flow	None
Exception Flow	None
Related Requirements	Possible Actions linked with other user case:
Post-Condition	Owners can see the notifications; readers can make an orders.

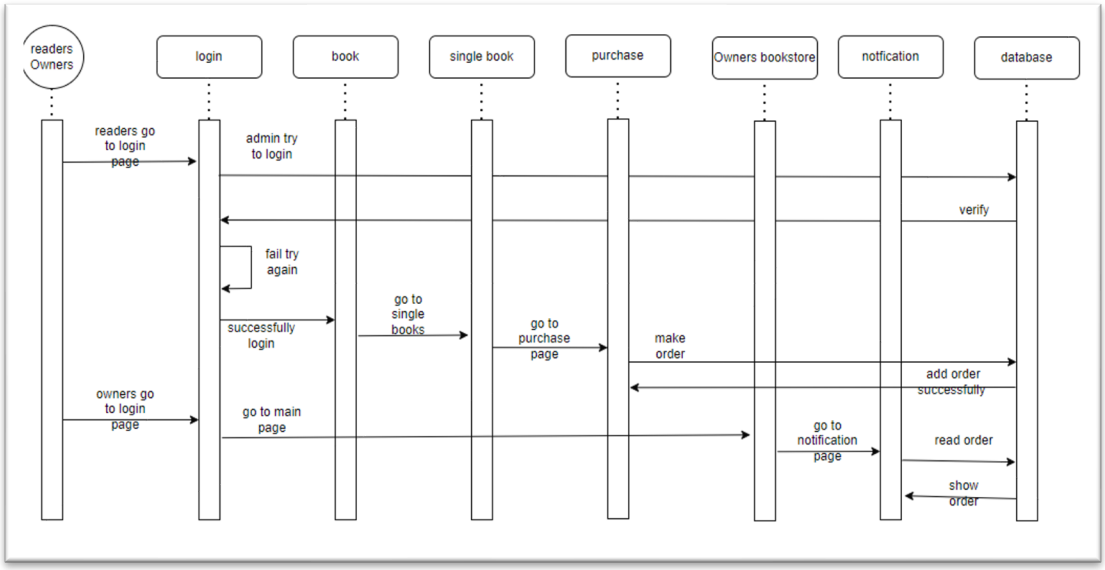

Figure 3.6 Sequence Diagram for UC006 order Use case

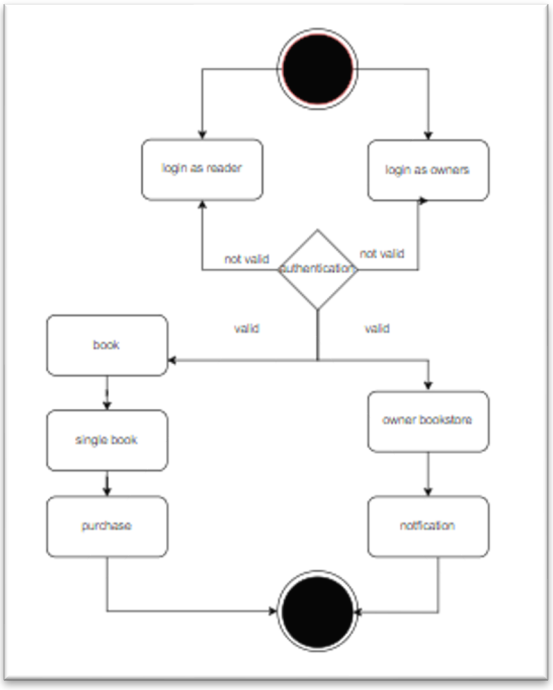

Figure 3.6 Activity Diagram for UC006 order Use case

3.2.7. UC007: Use Case <user>

Table 3.7 Use Case Description for < **user** >

User Case	**user**
ID	UC007
Actors	Owners and readers
Description	Owners and readers can create account for their self.
Pre-Condition	None
Normal Flow	1. The use case starts when the actor opens the system. 2. Readers 2.1. Click on register 2.2. Register 3. Owners 3.1. Owners click on register bookstore 3.2. Register bookstore
Alternative Flow	None
Exception Flow	None
Related Requirements	Possible Actions linked with other user case:
Post-Condition	Owners and readers successfully can add their account.

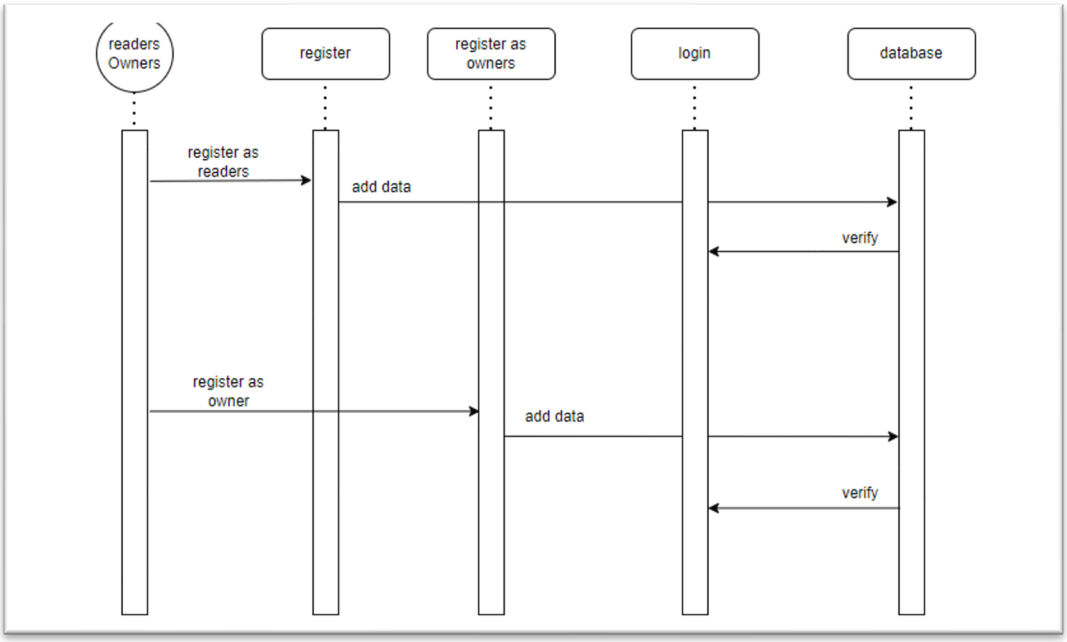

Figure 3.7 Sequence Diagram for UC007 user Use case

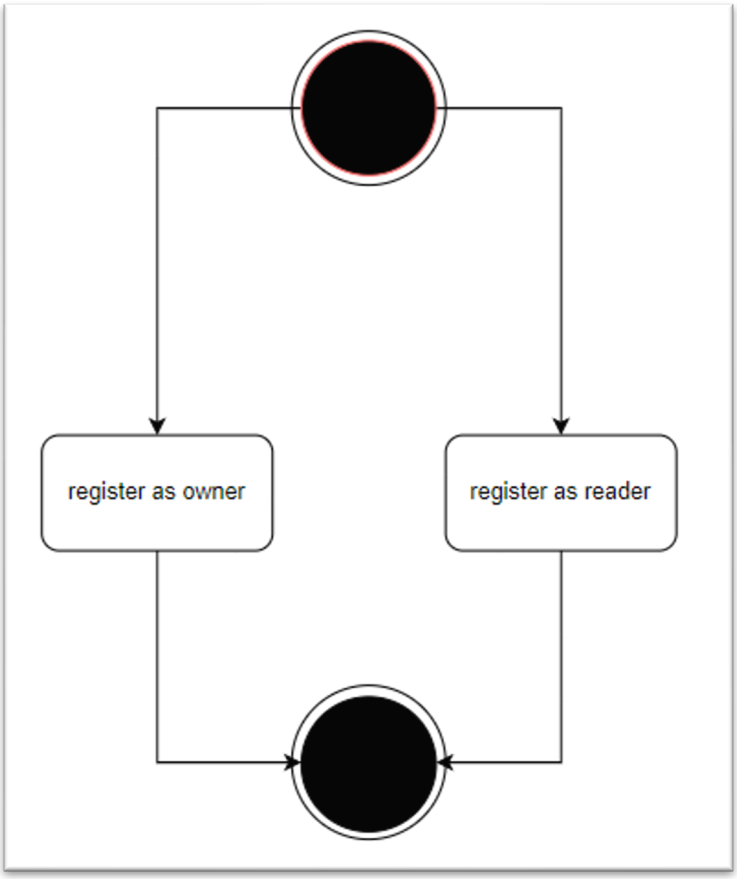

Figure 3.7 Activity Diagram for UC007 user Use case

3.3 Performance Requirements

Table 3.8 Non-Functional Requirements

Non-Functional Requirements	Description
Usability	The user interface should be appropriate for a system that requires no additional training or assistance for the target user to understand.
Availability	The proposed system should be accessible from anywhere that has access to the internet and a computer.
Performance	All of the pages have full performance; the pictures reduce performance, but you can still get to the pages quickly.

Appendix B

Software Design Document

SCSJ3323: Software Design and Architecture

Software Design Document

Project Title

Cross platform bookstore management system

Version 1.0

Printing Date

Department and Faculty

Computer science (software engineering)

Prepared by: Hevar Hoshang Ahmed

Revision Page

e. Overview

Describe the content of the current version.

f. Target Audience

State the targeted audience.

g. Project Team Members

List the team members and respective assigned module.

h. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	**Hevar Hoshang**		

Table of Contents

1. Introduction

1.1 Purpose

The purpose of Software Design Documentation (SDD) is to offer a clear view on how the DCMS is going to construct and quick description of the system's glide. Software design description together with architecture style, database description and interface design additionally being explained on this file.

1.2 Scope

The Cross Platform Bookstore Management System are considered as a pc machine that is used to control and save the Books within the Bookstores, and offer search engine that help Readers look for particular book. The mission scope will improve each the running environment of the Owners and make sure the Readers' pleasure. We propose these assignments to show beneficial for the Bookstores owners and to make certain that excellent device are given to the Readers as well. We think that this task will help the Readers to take find their Books and purchase them without transferring an inch from their residence. We can summarize the scope into the subsequent points:

- The technology will concentrate development efforts in Kurdistan's bookstores.
- The technology provides cross platform that can be used in each devices.

1.3 Definitions, Acronyms and Abbreviation

Table 1.1 Definition, Acronyms and Abbreviation

Acronym/ Abbreviation/ Term	Definition
CPBMS	Cross Platform Bookstore management system
MVC	Model View Controller
ERD	Entity Relationship Diagram

1.4 References

Krasner, G.E. and Pope, S.T., 1988. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object-oriented programming*, *1*(3), pp.26-49.

1.5 Overview

This document describes the overview of the system into various sections. The section of this document as sated as below:

2. Introduction
3. System Overview
4. System Architecture
5. Database Design
6. Interface Design

2. System Overview.

In Kurdistan we have so many Book stores, but none of them have digital platform, that they can sell Books online, in other hand there are many readers have problems of way finding books they will check each book store one by one to find the specific book because we don't have a search engine platform that just by searching for the book, you would know where you can purchase the book that you want, the platform will be Cross platform that can work on any devices, it is important to add all the information of the book into the platform and work efficiently, so readers and owners will be satisfied with the platform.

so basically we have three stakeholders admin who manage the platform, the owners who have their own the management system, and the readers who can search for the books, so reader will search for specific book for example fire & blood it gives him those bookstores that have this book he can choose one of the bookstore and order it, so we also have owners screen they add all the books and their information in the database, so it tells them the price and the count of the book, so this platform **Cross Platform Bookstore Management System** help both user who are readers and owners.

3. System Architecture Design

3.1 MVS Architecture

MVC is an architectural pattern that divides an application into three major logical additives: the version, the view, and the controller. Each of these components is designed to address specific application development factors. MVC is a popular enterprise-trendy web development framework for creating scalable and extensible tasks.

Let's talking about Modules first

The Model component corresponds to all of the records-related logic that the user works with. This can constitute either the information that is being transferred among the View and Controller components or some other enterprise good judgment-associated information. For instance, a Customer object will retrieve the customer information from the database, manage it and update it records again to the database or use it to render records.

Second controller

Controllers act as an interface between Model and View additives to system all the business common sense and incoming requests, control information the usage of the Model component and interact with the Views to render the final output. For example, the Customer controller will take care of all the interactions and inputs from the Customer View and update the database the use of the Customer Model. The equal controller will be used to view the Customer information.

Third is view

The View element is used for all of the UI common sense of the utility. For instance, the Customer view will include all the UI components such as textual content packing containers, dropdowns, etc. That the final person interacts with.

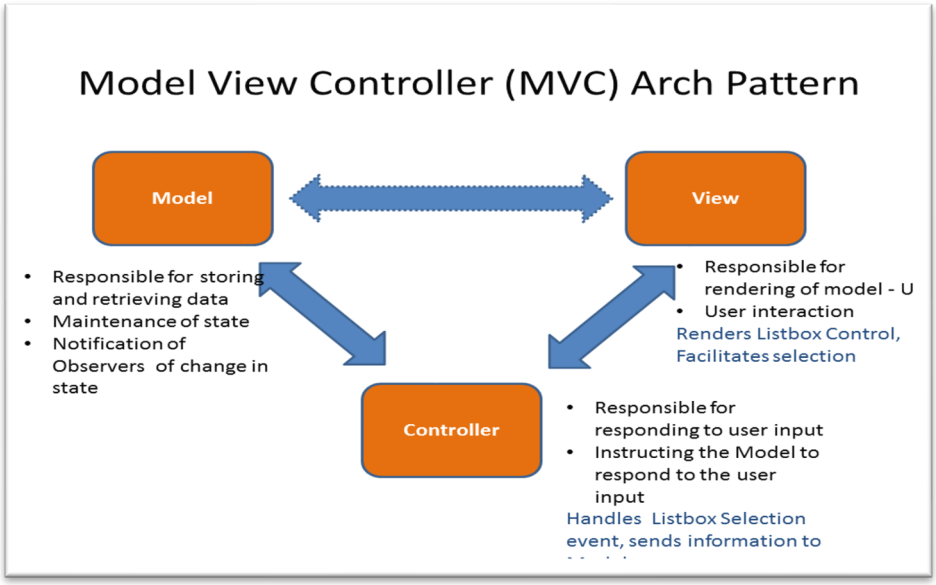

Figure 3.1 MVC Architecture

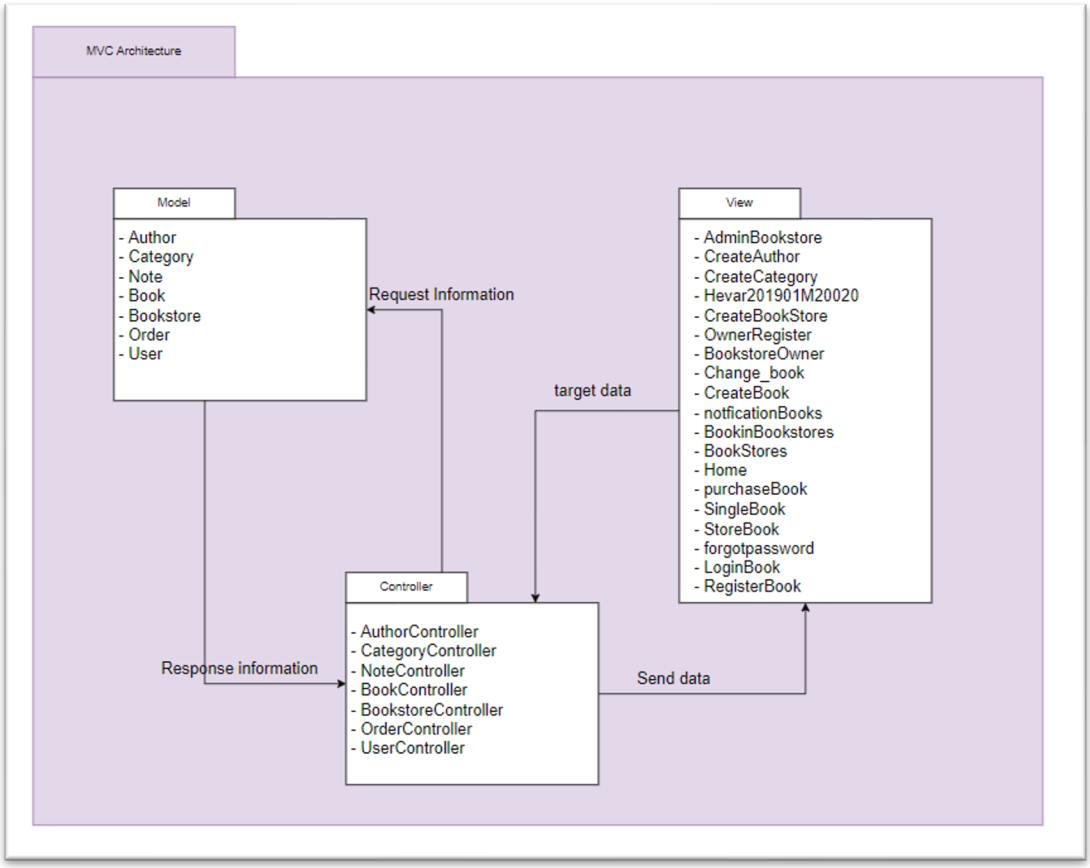

Figure 3.2 MVC Architecture of CPBMS

4. Data Design

4.1 Entity Relationship Diagram

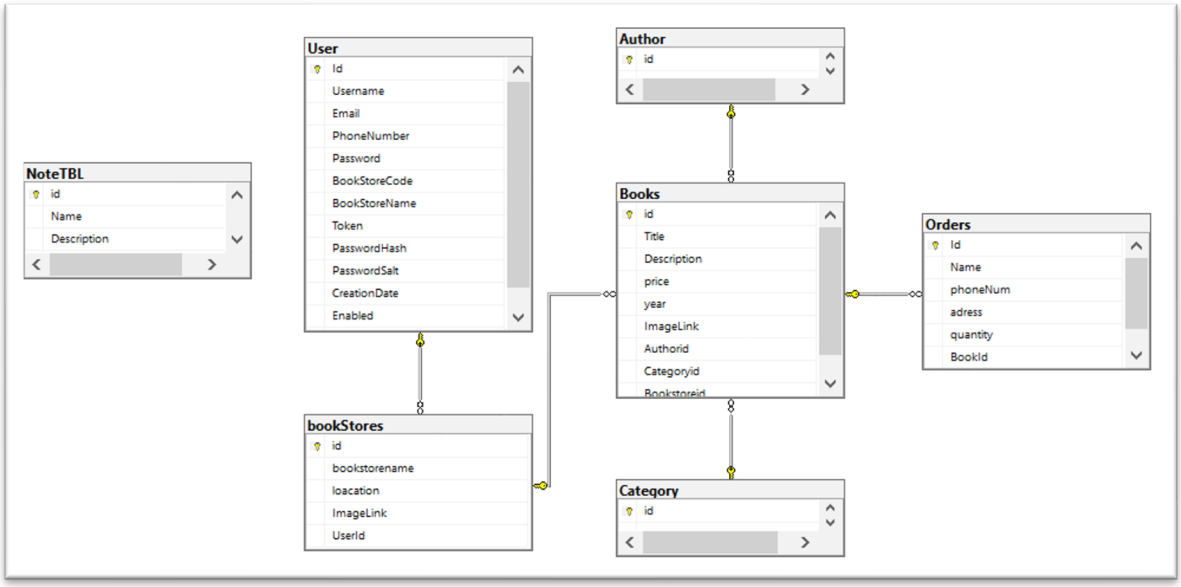

Figure 4.1 ERD for CPBMS

4.2 Data Dictionary

Below is the description of the data dictionary in the CPBMS database system.

Table 4.1 Database Table: Authors

Attribute	Data Type	Description
Id	INTEGER	Primary Key
FullName	VARCHAR	Name of the authors

Table 4.2 Database Table: Categories

Attribute	Data Type	Description
Id	INTEGER	Primary Key

Category	VARCHAR	Name of the category

Table 4.3 Database Table: Book

Attribute	**Data Type**	**Description**
Id	INTEGER	Primary Key
Title	VARCHAR	Title of the book
Description	VARCHAR	Description of the book
Price	DOUBLE	Price of the book
year	INTEGER	year of the book
ImageLink	VARCHAR	iMAGE of the book
Authorid	INTEGER	Primary Key
Categoryid	INTEGER	Primary Key
Bookstoreid	INTEGER	Primary Key

Table 4.4 Database Table: bookstore

Attribute	**Data Type**	**Description**
Id	INTEGER	Primary Key
bookstorename	VARCHAR	bookstorename of the bookstore
location	VARCHAR	location of the bookstore
ImageLink	VARCHAR	iMAGE of the bookstore
Userid	INTEGER	Primary Key

Table 4.5 Database Table: Note

Attribute	**Data Type**	**Description**
Id	INTEGER	Primary Key
Name	VARCHAR	Name of the note
Description	VARCHAR	Description of the note

Table 4.6 Database Table: order

Attribute	**Data Type**	**Description**
Id	INTEGER	Primary Key
Name	VARCHAR	Name of person
phoneNum	VARCHAR	Phone of person
adress	VARCHAR	Address of the person
quantity	INTEGER	Quantity of book
BookId	INTEGER	Primary Key

Table 4.7 Database Table: User

Attribute	**Data Type**	**Description**
Id	INTEGER	Primary Key
Username	VARCHAR	username of person
PhoneNumber	VARCHAR	Phone of person
Email	VARCHAR	email of the person
token	VARCHAR	Token to get authrazation
passwordHash	BYTE	To hash the password

passwordSalt	BYTE	To salt the password
CreationDate	DateTime	Time of creation
Enabled	BOOL	To enable or disable user
Role	VARCHAR	What kind of rule we have

5. User Interface Design

5.1 Overview of User Interface

This system is primarily concerned with desktop systems. A user-friendly interface has been created for the target to use without any additional training.

5.2 Screen Images

5.2.1 Zlogin

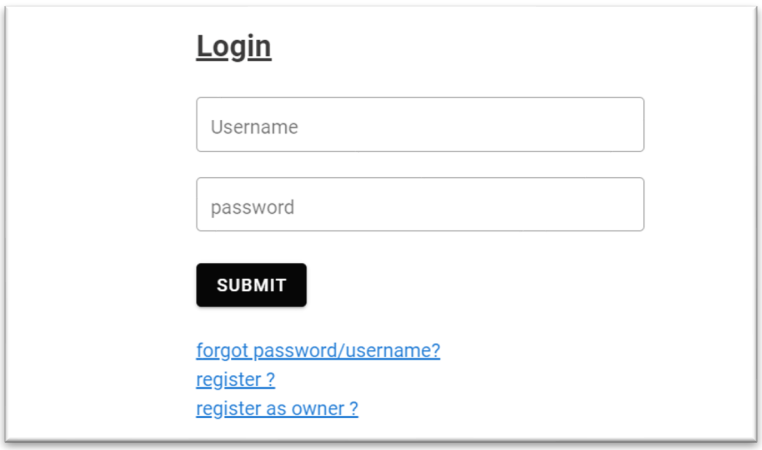

Figure 5.1 Login Page Interface

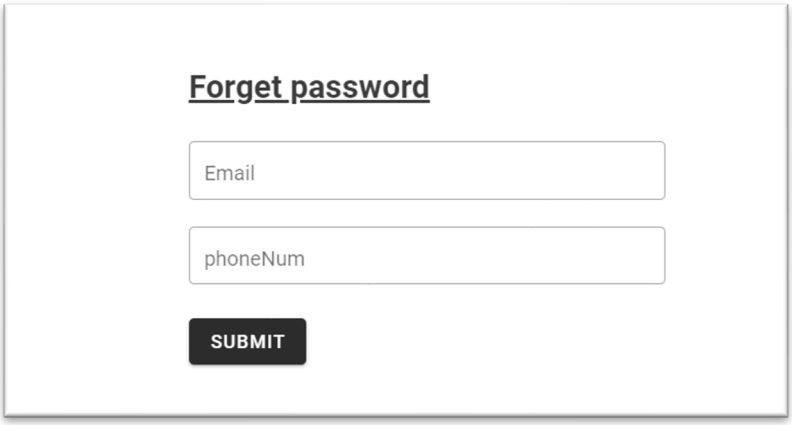

Figure 5.2 forget password Page Interface

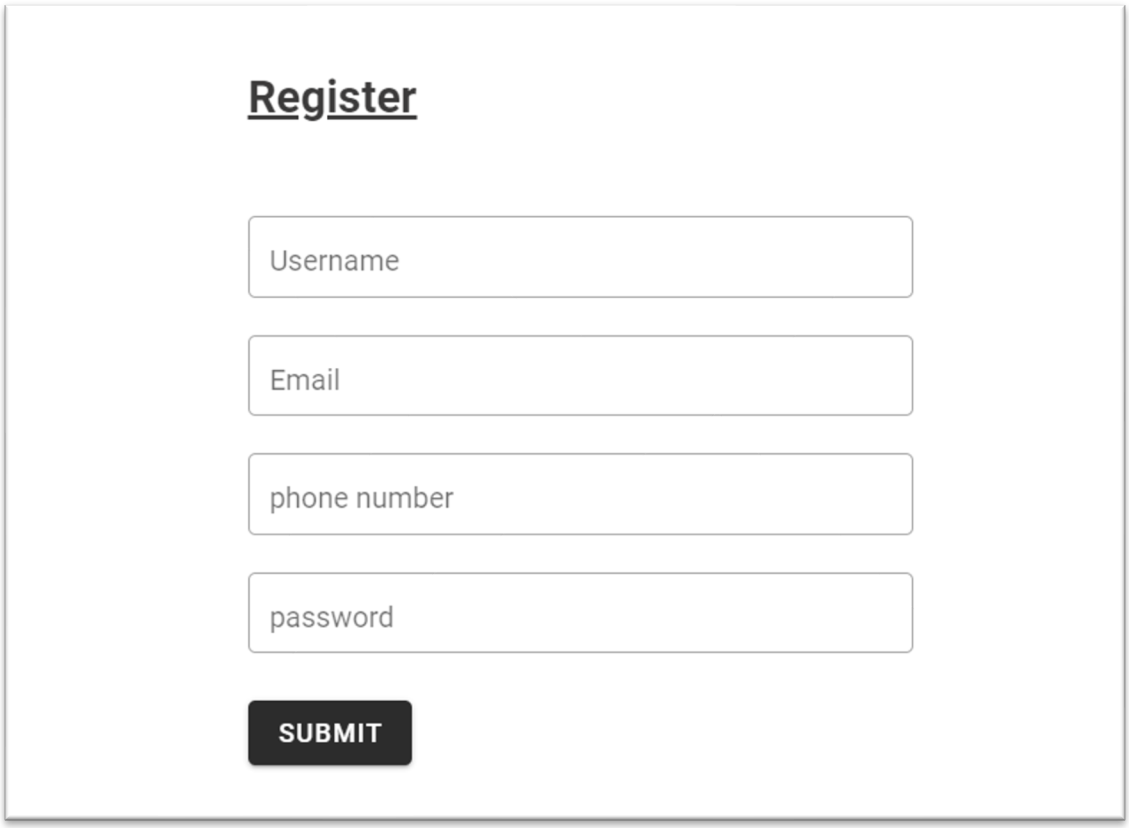

Figure 5.4 register Page Interface

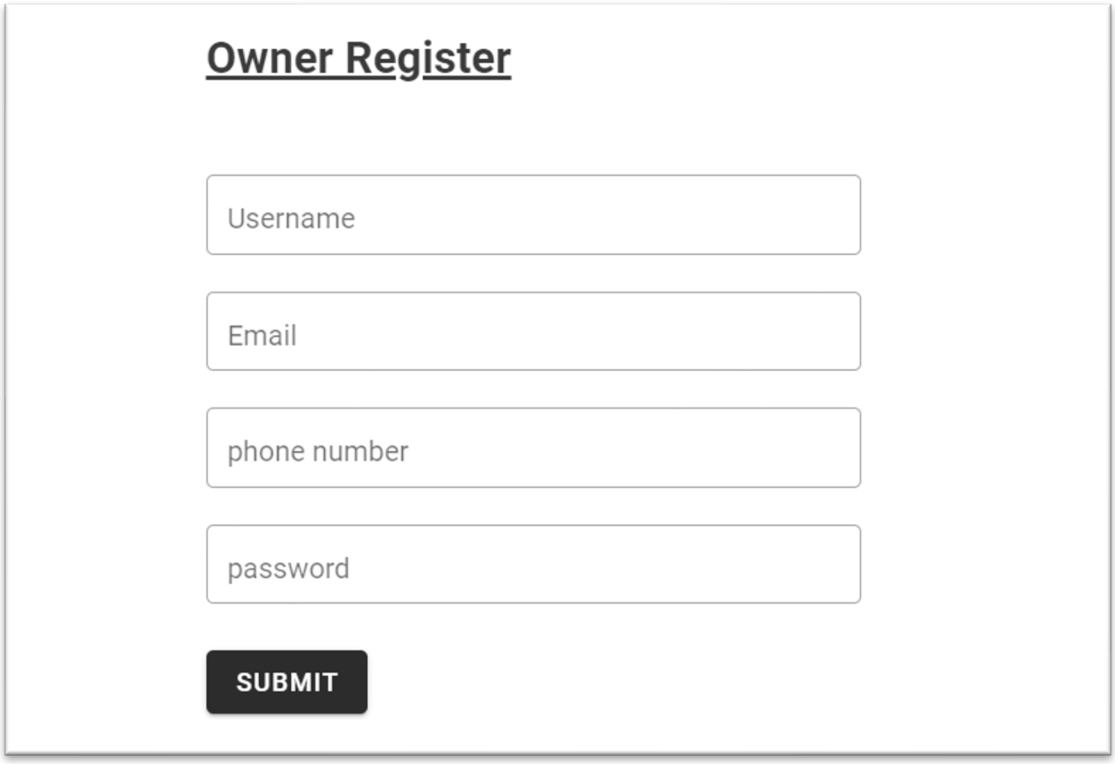

Figure 5.5 register owners Page Interface

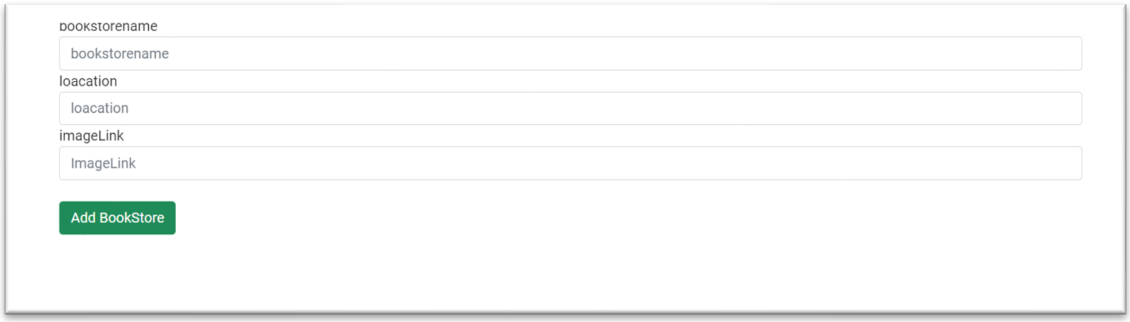

Figure 5.6 Create bookstore Page Interface

5.2.2 Readers

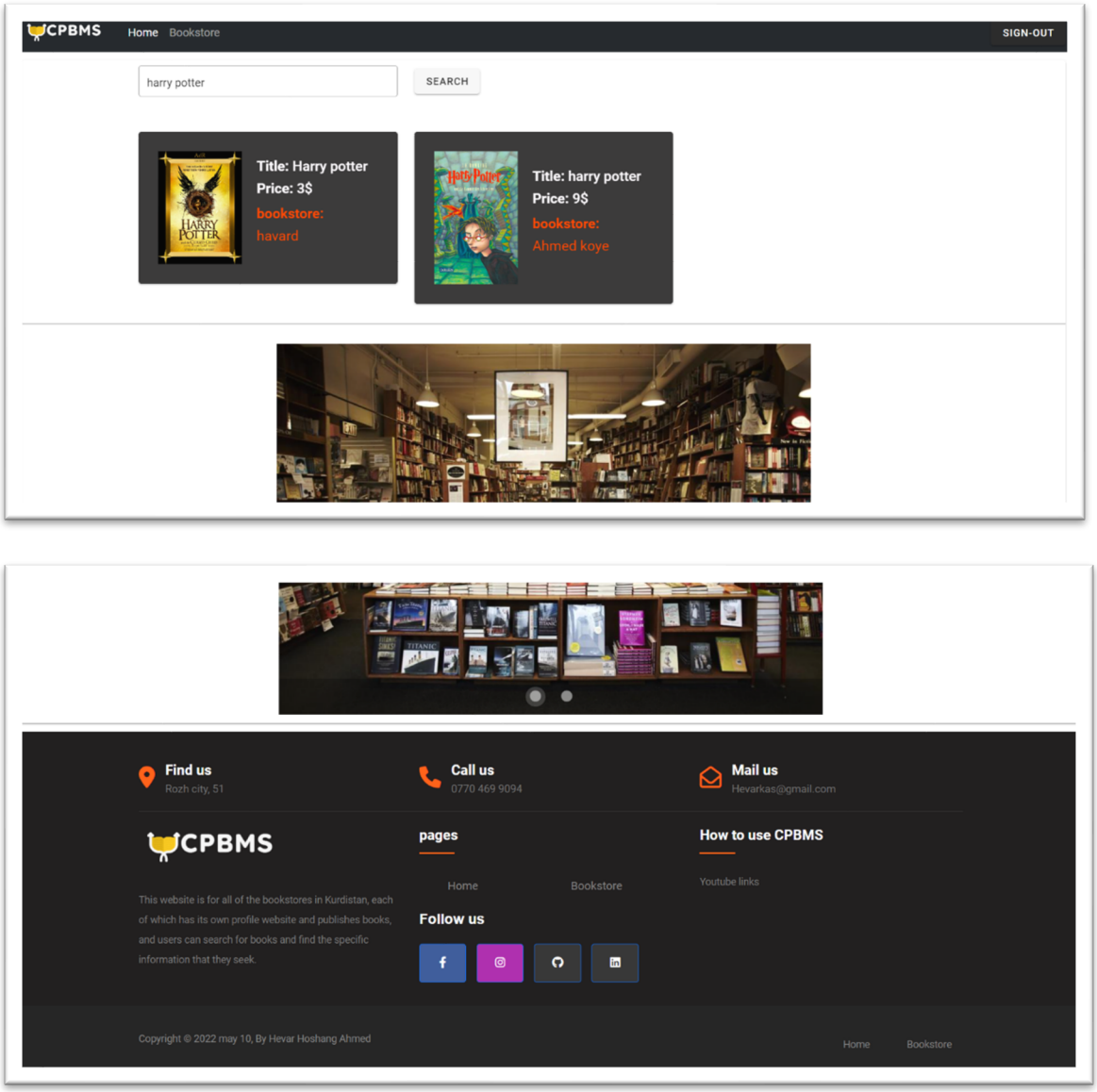

Figure 5.7 homepage Page Interface

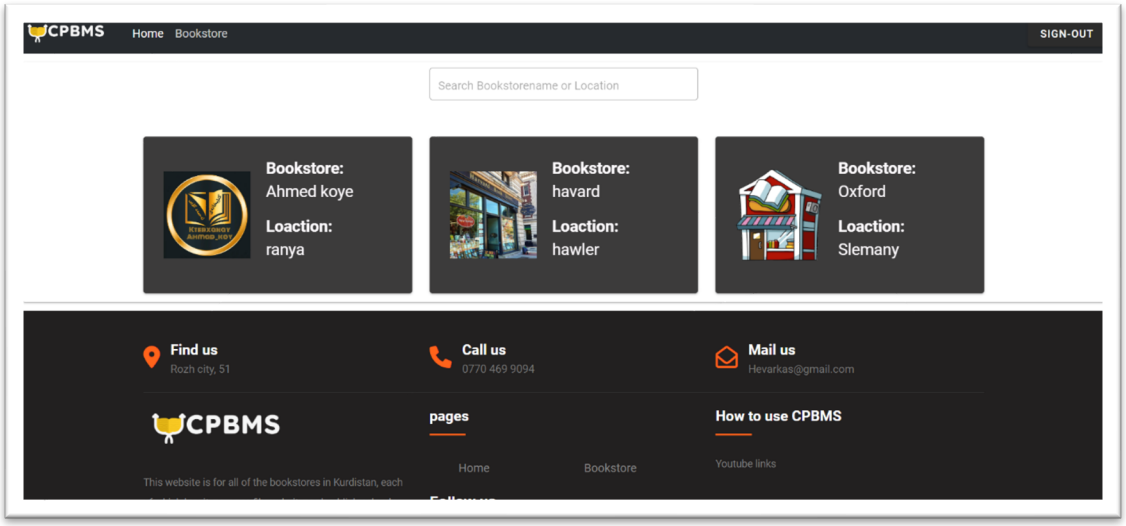

Figure 5.8 bookstore Page Interface

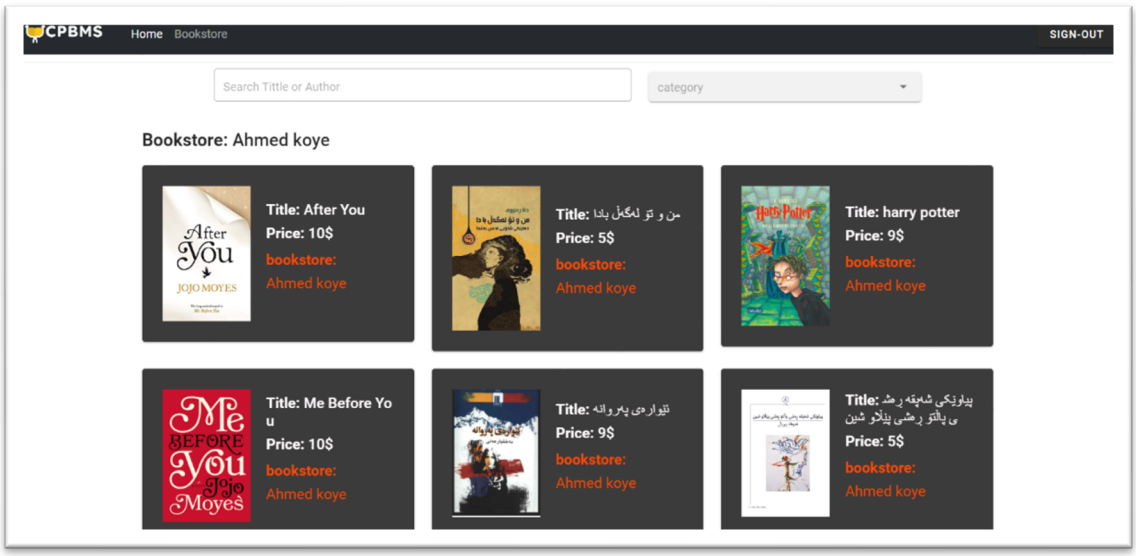

Figure 5.9 bookinbooks Page Interface

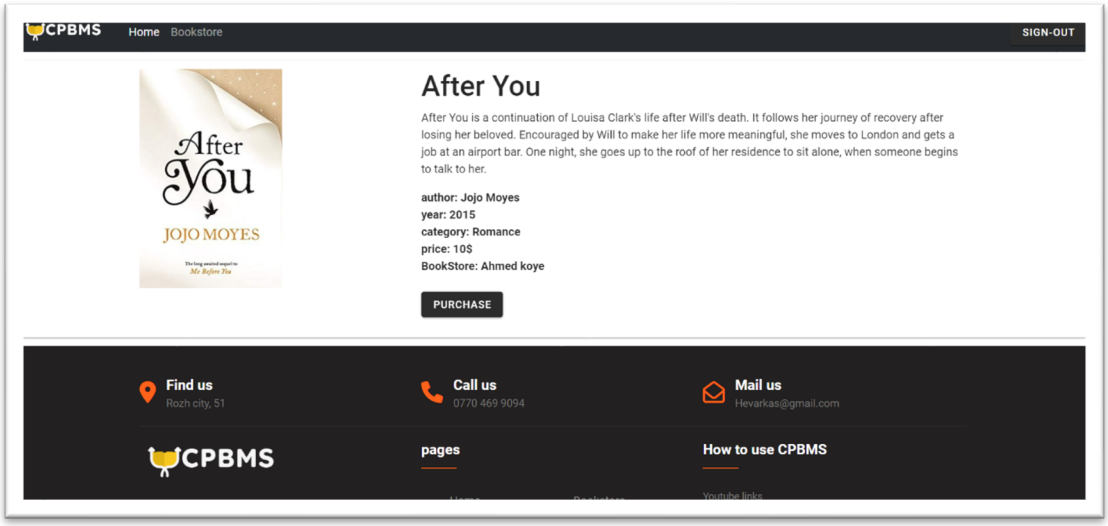

Figure 6 single books Page Interface

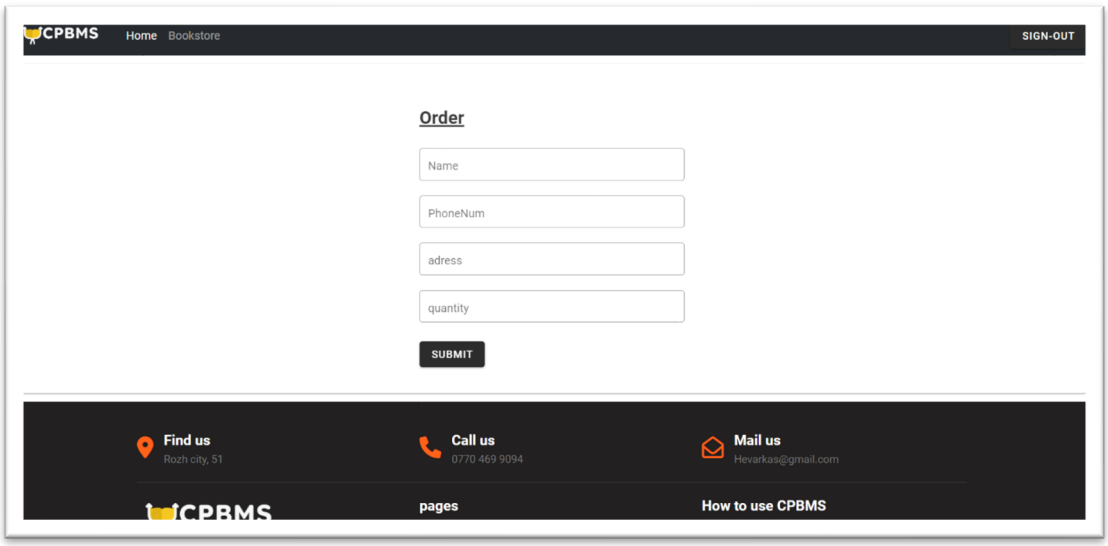

Figure 6 purchase books Page Interface

5.2.3 owners

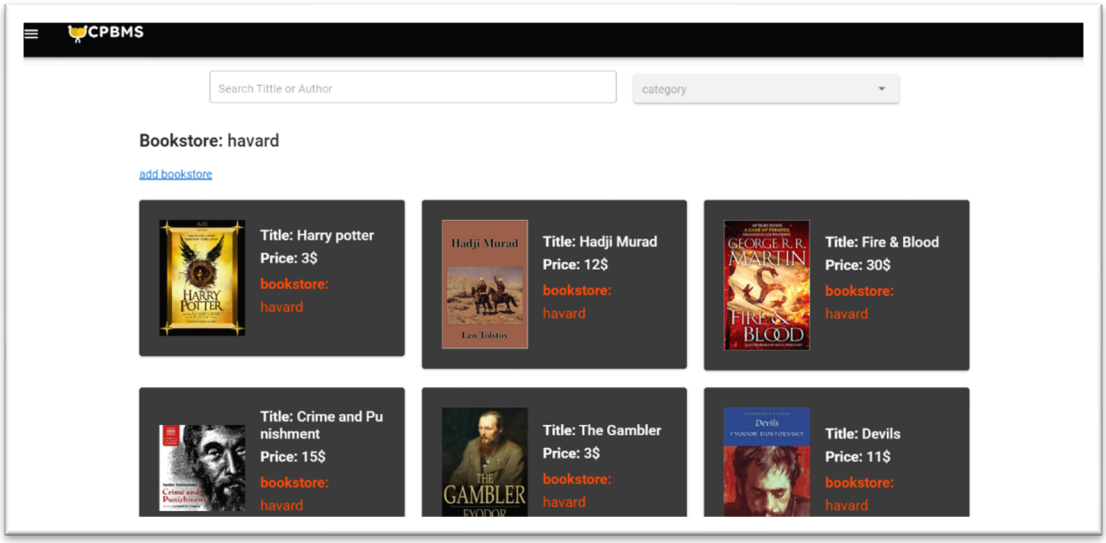

Figure 6.1 ownersbookstore Page Interface

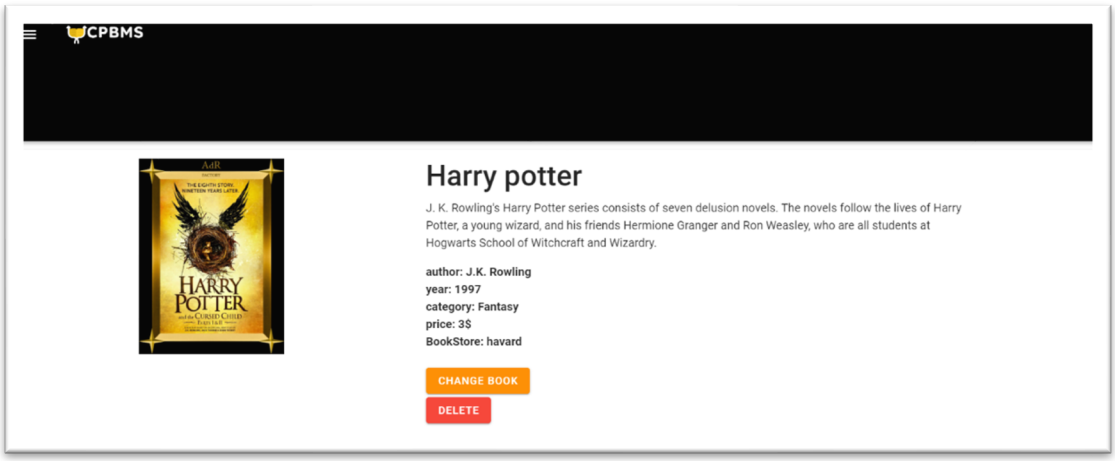

Figure 6.2 singleownersbookstore Page Interface

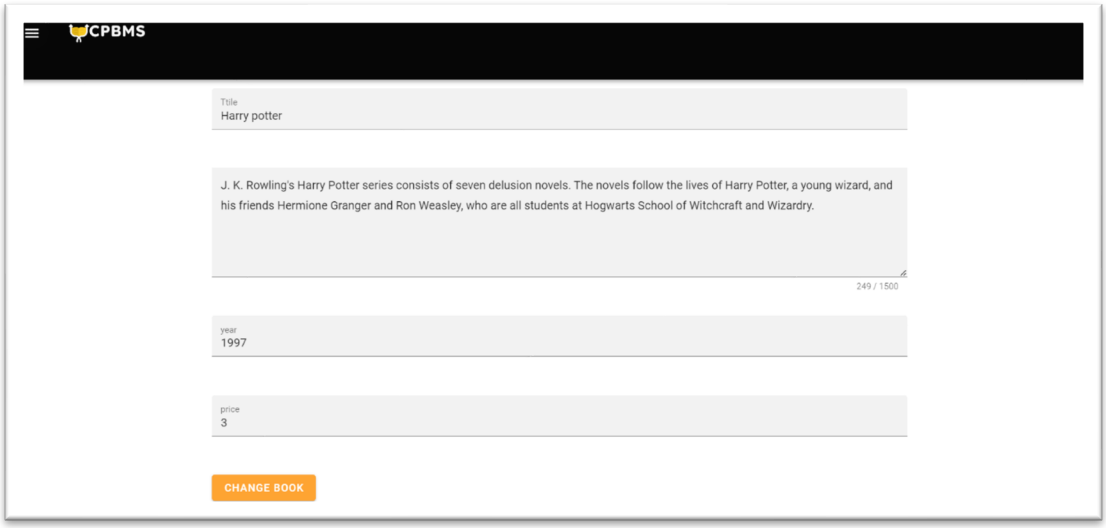

Figure 6.2 changebook Page Interface

5.2.3 admin

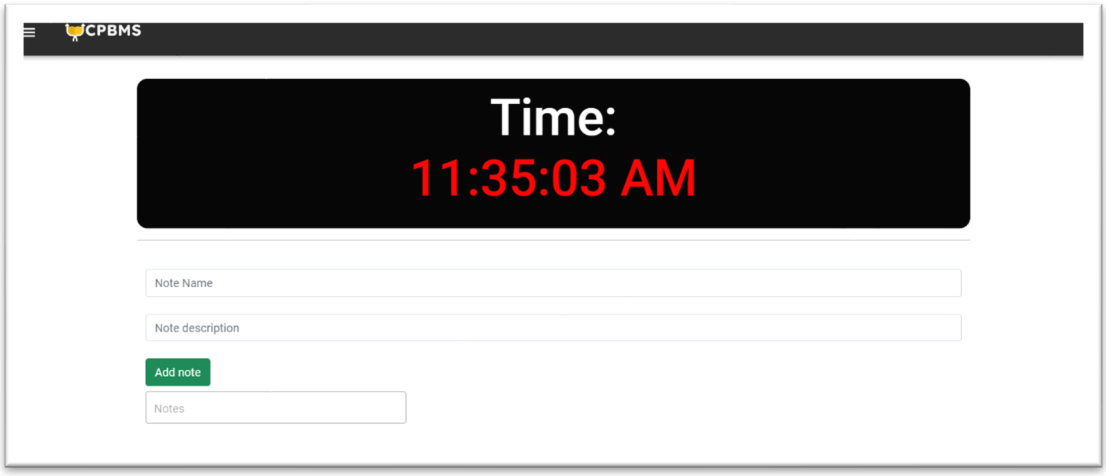

Figure 6.3 admin Page Interface

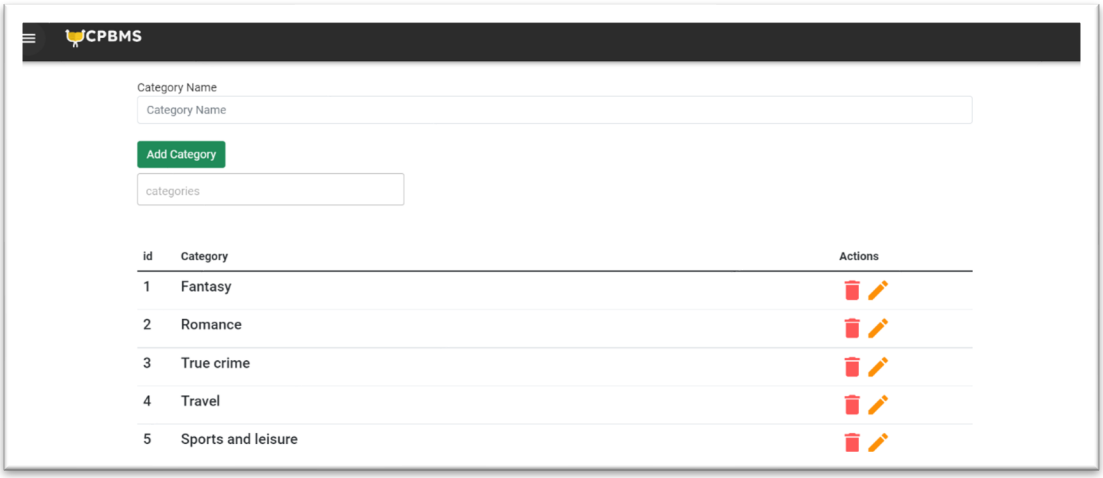

Figure 6.4 category Page Interface

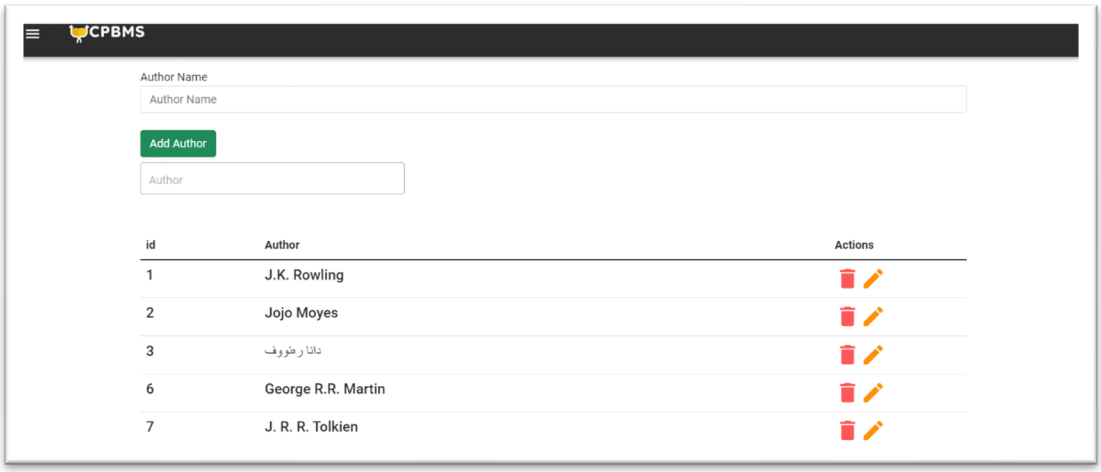

Figure 6.5 author Page Interface

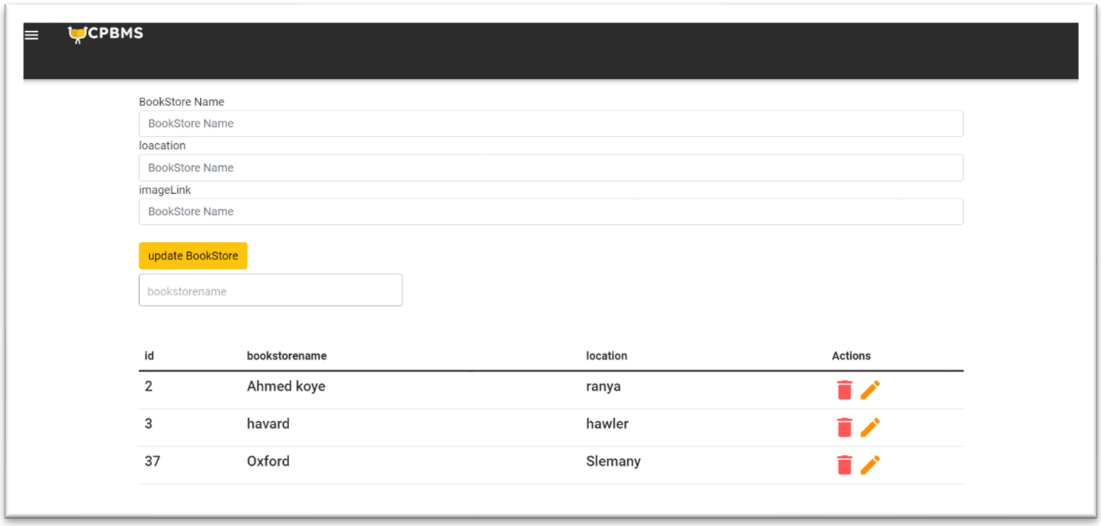

Figure 6.6 bookstoreadmin Page Interface

Appendix C

Software Testing Documentation

Software Testing Documentation

Project Title

Cross Platform Bookstore Management System

Version 1.0

Printing Date

Department and Faculty

Computer science (software engineering)

Prepared by:

Hevar Hoshang Ahmed

Revision Page

i. Overview

Describe the content of the current version.

j. Target Audience

State the targeted audience.

k. Project Team Members

List the team members and respective assigned module.

l. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
<Current Version>			

Table of Contents

4. Introduction

Programming testing is a method of identifying flaws in an item's requirements, design, and execution. It is used to check programming program items for correctness, completion, security, and fineness in comparison to a decision. Programming examination is a method of determining the quality of a program that has been created. It reveals all of the product's flaws, errors, and shortcomings. There are various testing procedures that can be used, such as White Box Testing, Black Box Testing, Gray Box Testing, and Agile Testing. Aside from methodologies, there are also evaluating stages such as Unit Testing, System Testing, Acceptance Testing, and Integration Testing. To complete the testing for this mission, the DCMS task could be completed using the Black Box Testing technique and the User Acceptance Testing stage.

4.1 Purpose

This STD manages the cost of significant data in terms of looking at sports that include test portrayal and investigation results while looking at the DCMS. This STD also stores the UAT completed by an individual. The DCMS mission could be completed using the Black Box Testing method and the User Acceptance Testing level.

4.2 Scope

The Cross Platform Bookstore Management System are considered as a pc machine that is used to control and save the Books within the Bookstores, and offer search engine that help Readers look for particular book. The mission scope will improve each the running environment of the Owners and make sure the Readers' pleasure. We propose these assignments to show beneficial for the Bookstores owners and to make certain that excellent device are given to the Readers as well. We think that this task will help the Readers to take find their Books and purchase them without transferring an inch from their residence. We can summarize the scope into the subsequent points:

- The technology will concentrate development efforts in Kurdistan's bookstores.
- The technology provide cross platform that can be used in each devices.

4.3 Definitions, Acronyms and Abbreviation

Table 1.1 Definition, Acronyms and Abbreviation

Acronym/ Abbreviation/ Term	Definition
CPBMS	Cross platform Bookstore management system
STD	Software Testing Documentation
TC	Test Case
UAT	User Acceptance Testing
SC	School of Computing

Test Cases, Data and Expected Results

4.4 Test TC001 for Module View Profile: <View Profile (UC001)>

UC001_01: e.g. users login

Test Case ID	Input data	Expected result	Actual result	Pass Fail
TC001_01_01	Logged in as hhqu	The system displays the owner's profile.	The system successfully displayed the owner profile.	Pass
TC001_01_02	Logged in as hevarkas	The system displays the user's profile.	The system successfully displayed the user profile.	Pass
TC001_01_03	Logged in as Awear	The system displays the administrator's profile..	The system successfully displayed the admin profile.	Pass
TC001_01_04	Logged in as ahmed	System shows the profile of the user.	The system failed to display the user's profile because the user was not logged in.	Fail

4.5 Test TC002 for Module category: < View Attendance (UC002)>

UC001_01:. category

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC002_01_01	Logged as a admin	Show the data	System successfully show all the categorys that you can change delete add read.	Pass
TC002_01_02	Logged as a owners	Won't show the data	System failed to show data to the owners	Fail
TC002_01_03	Logged as a readers	Won't show the data	System failed to show data to the readers	Fail

4.6 Test TC003 for Module author: <author (UC003)>

UC002_01: author

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC003_02_01	Logged as a admin	Show the data	System successfully show all the categorys that you can change delete add read.	Pass
TC003_02_02	Logged as a owners	Won't show the data	System failed to show data to the owners	Fail
TC003_02_03	Logged as a readers	Won't show the data	System failed to show data to the readers	Fail

4.7 Test TC004 for Module book: < book (UC004)>

UC004_01: book

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC004_01_01	Loged as admin	Won't show anything	System fail to show or change books	fail

TC004_01_02	Loged as owners	Showing books	System successfully can see all the books and make changes	pass
TC004_01_03	Loged as reader	Showing books	System successfully let reader to see all of the books	pass

4.8 Test TC005 for Module Manage Attendance: <Book store (UC005)>

UC005_01: Bookstore

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC005_01_01	“username” : “Awear” “password” : “jL9VM1dZ” “role” : “admin”	Can show the bookstore	System successfully showing bookstore edit bookstore and delete bookstore	Pass
TC005_01_02	“username” : “hevar” “password” : “123456” “role” : “owner”	Can show the bookstore	System successfully get to his own bookstore	Pass
TC005_01_03	“username” : “hevarkas” “password” : “jhD5b3?L” “role” : “reader”	Can show the bookstore	System successfully can read bookstore	Pass
TC005_01_04	Not authenticate	Can’t show bookstore	System unsuccessfully to showing data	fail

4.9 Test TC006 for note: < note (UC006)>

UC004_06: note

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC006_01_01	“username” : “Awear” “password” : “jL9VM1dZ” “role” : “admin”	System can see note	System successfully can read and change note	Pass

TC006_01_02	“username” : “hevar” “password” : “123456” “role” : “owner”	System can’t show the note	System un unsuccessfully will show the note	Fail
TC006_01_03	“username” : “hevarkas” “password” : “jhD5b3?L” “role” : “reader”	System won’t show the note	System un unsuccessfully will show the note	Fail
TC006_01_04	Not authenticate	System won’t show the note	System un unsuccessfully will show the note	Fail

4.10 Test TC007 for Module order: < order (UC007)>

UC007_01: order

Test Case ID	**Input data**	**Expected result**	**Actual result**	**Pass / Fail**
TC007_01_01	“username” : “Awear” “password” : “jL9VM1dZ” “role” : “admin”	Won’t show data	System unsuccessfully showing data	fail
TC007_01_02	“username” : “hevar” “password” : “123456” “role” : “owner”	System shows the order	System successfully showed the data	Pass
TC007_01_03	“username” : “hevarkas” “password” : “jhD5b3?L” “role” : “reader”	System shows the order	System successfully post the data	pass
TC007_01_04	Not authenticate	Won’t show data	System unsuccessfully showing data	Fail

Appendix D

User Acceptance Testing for plus the edge company

User Acceptance Testing for plus the edge company

Purpose :To test user acceptance of CPBMS meets user needs and expectations.

Testing Date :13/06/2022

Role:	✓	Admin (section 1 & 2)	**Name: Dawar Fuad**
		Employee (section 3)	**Name: Wrya Muhamad**

Scale of experience:

1: Very bad 2: Bad 3: Neutral 4: Good 5: Very good

Section 1	System design evaluation					
Purpose	To see if the system design meets the requirements.					
No	**Aspect**	**Score**				
		1	**2**	**3**	**4**	**5**
1.	Linking and navigating				✓	
2.	Color scheme for the menu				✓	
3.	Text color System				✓	
4.	background color				✓	
5.	Design of the entire system					✓
6.	System user-friendliness					✓
7.	System content information				✓	
Comment and Suggestion	-					

Section 2	System evaluation (Admin)					
Purpose	CPBMS Admin system functionality must be tested.					
No	**Aspect (Functionality)**	**Score**				
		1	**2**	**3**	**4**	**5**
1.	Login and logout to the system					✓
2.	View the Dashboard page					✓
3.	Add note					✓
4.	Update note					✓
5.	Remove note				✓	
6.	Read note					✓
7.	Add author					✓
8.	Update author				✓	
9.	Remove author					✓
10.	Read author					✓
11.	Add category					✓
12.	Update category					✓
13.	Remove category				✓	
14	Read category					✓
15	Update bookstore					✓
Comment and Suggestion	-					

Section 3	System testing (owners)					
Purpose	To test the system's functionality for CPBMS_Owners					
No	**Aspect (Functionality)**	**Score**				
		1	**2**	**3**	**4**	**5**
1.	Log in and out of the system					✓
2.	View the dashboard page					✓
3.	Add book		✓			
4.	remove book					✓
5.	change book					✓
6.	read book					✓
7.	Remove notification					✓
8.	Read notification			✓		
Comment and Suggestion						

Section 4	System testing (readers)					
Purpose	To test the system's functionality for CPBMS_readers					
No	**Aspect (Functionality)**	**Score**				
		1	**2**	**3**	**4**	**5**
1.	Log in and out of the system					✓
2.	View the dashboard page					✓
3.	Read book					✓
4.	Read bookstore					✓
5.	post order					✓
6.	Search for book					✓
7.	Search for bookstore					✓
8.	navigation					✓
Comment and Suggestion	-					