**ASSIGNMENT 1 FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **BTEC Level 5 HND Diploma in Business** | | |
| **Unit number and title** | **Unit 30: Application Development** | | |
| **Submission date** | Tuesday, 14 February 2023 | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Dang Nguyen Dang Khoi | **Student ID** | GCC200058 |
| **Class** | GCC0903 | **Assessor name** | Luong Hoang Huong |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
|  |  | **Student’s signature** | Khoi |

**Grading grid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **P1** | **P2** | **P3** | **M1** | **M2** | **D1** |
|  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Internal Verifier’s Comments:** | | |
| **Signature & Date:** | | |

**Assignment Brief 1 (RQF)**

**Higher National Certificate/Diploma in Computing**

|  |  |
| --- | --- |
| **Student Name/ID Number:** |  |
| **Unit Number and Title:** | **Unit 30: Application Development** |
| **Academic Year:** | **2021 – 2022** |
| **Unit Assessor:** | **Hoang Nhu Vinh** |
| **Assignment Title:** | **Analysis and Design solution for a given problem** |
| **Issue Date:** | **10 May 2022** |
| **Submission Date:** |  |
| **Internal Verifier Name:** |  |
| **Date:** |  |

|  |
| --- |
| **Submission Format:** |
| Format: The submission is in the form of two documents/files   * An individual SRS document in PDF format. Writing must be professional. * An individual evaluation document in PDF format. Writing must be professional.   Submission   * Students are compulsory to submit the assignment on the due date and in a way requested by the Tutor. * The form of submission will be a soft copy posted on <http://cms.greenwich.edu.vn/>. * Remember to convert the word file into a PDF file before the submission on CMS.   Note:   * The individual Assignment must be your work, and not copied by or from another student. * If you use ideas, quotes, or data (such as diagrams) from books, journals, or other sources, you must reference your sources, using the Harvard style. * Make sure that you understand and follow the guidelines to avoid plagiarism. Failure to comply with this requirement will result in a failed assignment. |
| **Unit Learning Outcomes:** |
| **LO1** Produce a Software Design Document by analyzing a business-related problem and deduce an appropriate solution including a set of initial requirements  **LO2** Use design and development methodologies with tools and techniques associated with the creation of a business application |
| **Assignment Brief and Guidance:** |
| **Scenario**: FPTBook will be a web-based software system to aid book store customers and owners to manage their book purchases and sales about simplifying and speeding up the process of selection, ordering, and purchasing books for customers as well as managing a database of users and a database of products for book store owners. The attached document can view the detailed description of those roles (Requirements.docx).  **Tasks**  You and your teammates (3 members/team maximum) need to prepare a software design document with the following sections:   * A requirement specification that explores the problem by a set of user and system requirements, as well as determines any risks related to the successful completion of your application. You are advised to use an SRS template or modify one to complete this task. * An evaluation section in which you research the use of software development tools and techniques and identify any that have been selected for the development of this application. You should compare them and justify your choices. * A design section in which you use chosen tools from the previous step to produce design diagrams for your solution based on the requirement specification. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Learning Outcomes and Assessment Criteria (Assignment 1):** | | | |
| Learning Outcome | Pass | Merit | Distinction |
| LO1 | **P1** Explore a business-related problem and produce a well-defined Problem Definition Statement supported by a set of user and system requirements.  **P2** Determine any areas of risk related to the successful completion of your application. | **M1** Analyse a business-related problem using appropriate methods and produce a well-structured Software Design Document that defines a proposed solution and includes relevant details on requirements, system analysis, system design, coding, testing and implementation. | **D1** Justify the tools and techniques chosen to realise a custom built website. Justify your preferred selection of tools and techniques in deducing an appropriate solution to a business related problem. |
| LO2 | **P3** Research the use of software development tools and techniques and identify any that have been selected for the development of this application. | **M2** Compare the differences between the various software development tools and techniques researched and justify your preferred selection as well as your preferred software development methodology. |

Software Requirements Specification

for

FPTBook

Version <1.0>

Prepared by

Group Name: KTK

|  |  |  |
| --- | --- | --- |
| Dang Nguyen Dang Khoi | GCC200058 | khoidndgcc200058@fpt.edu.vn |
| Mai Hoang Minh Thai | GCC200064 | thaimhmgcc200064@fpt.edu.vn |
| Truong Van Tuan Kiet | GCC200203 | kiettvtgcc200203@fpt.edu.vn |

|  |  |
| --- | --- |
| Instructor: | Luong Hoang Huong |
| Course: | Application Development |
| Date: | 4/1/2023 |

# Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | Dang Nguyen Dang Khoi | This is the documentation for the first version of the project. It serves as a general roadmap for project development. | 03/03/2023 |

# 

**Table of Contents**

[I. Introduction 10](#_Toc127281156)

[**1. Document Purpose 10**](#_Toc127281157)

[**2. Product Scope 10**](#_Toc127281158)

[**3. Intended Audience and Document Overview 10**](#_Toc127281159)

[**4. Definitions, Acronyms and Abbreviations 11**](#_Toc127281160)

[**5. References and Acknowledgments 12**](#_Toc127281161)

[II. Overall Description 13](#_Toc127281163)

[**1. Product Overview 13**](#_Toc127281164)

[**2. Product Functionality 14**](#_Toc127281165)

[III. Specific Requirements 15](#_Toc127281166)

[**1. Functional Requirements 15**](#_Toc127281167)

[**2. Non Functional Requirements 16**](#_Toc127281168)

[**3. Use Case Model 17**](#_Toc127281169)

[**a. Use case of guest 18**](#_Toc127281170)

[**b. Use case of customer 21**](#_Toc127281171)

[**c. Use case of owner 24**](#_Toc127281172)

[**d. Use case of admin 28**](#_Toc127281173)

[**4. Wireflow 32**](#_Toc127281174)

[**a. Wireframe 32**](#_Toc127281175)

[**b. Wireflow 58**](#_Toc127281176)

[IV. Technical Design 64](#_Toc127281177)

[**1. Entity Relationship Diagram (ERD) 64**](#_Toc127281178)

[**2. Class Diagram 67**](#_Toc127281179)

[**a. Class diagram of register function 69**](#_Toc127281180)

[**b. Class diagram of login function 70**](#_Toc127281181)

[**c. Class diagram of functions of BookController class 71**](#_Toc127281182)

[**d. Class diagram of functions of CategoryController class 73**](#_Toc127281183)

[**e. Class diagram of functions of PublisherController class 74**](#_Toc127281184)

[**f. Class diagram of functions of CartController class 76**](#_Toc127281185)

[**g. Class diagram of functions of OrderController class 78**](#_Toc127281186)

[**h. Class diagram of functions of UserController class 80**](#_Toc127281187)

[**3. Activity Diagram 81**](#_Toc127281188)

[**4. Gantt Chart 93**](#_Toc127281189)

[V. Risk Assessment 95](#_Toc127281190)

[VI. Technologies Evaluation 98](#_Toc127281191)

[**1. Design Tools 98**](#_Toc127281192)

[**2. Front End technology stack 99**](#_Toc127281193)

[**3. Back End technology stack 104**](#_Toc127281194)

[**4. Tools for source control management 110**](#_Toc127281195)

[**5. Software Development Models 112**](#_Toc127281196)

[**6. Illustrate all my findings on how to use these by drawing the overview 113**](#_Toc127281197)

[References 114](#_Toc127281198)

**Table of Figures**

[Figure 1. Product Overview 14](#_Toc127281637)

[Figure 2. Use case of guest 19](#_Toc127281638)

[Figure 3. Use case of customer 22](#_Toc127281639)

[Figure 4. Use case of owner 25](#_Toc127281640)

[Figure 5. Use case of admin 29](#_Toc127281641)

# I. Introduction

FPTBook is a web-based software system to help customers easily search, order and manage all information about their orders to simplify the customer's purchasing process row. In addition, it also helps store owners can easily manage a database of users, orders, books... Moreover, it also helps admins can easily review statistics. What books are best sellers? This helps the admin to make better business decisions about which books to import and which books to restrict. It also helps the admin to ban any account to avoid the customers who order but do not receive the goods affecting the business process of the store. The attached document can view the detailed description of those roles.

## 1. Document Purpose

The purpose of the Software Requirements Specification document is to clearly define the system under development, specifically the FPTBook. The version of the system is version 1.0. The intended audience of this document includes the owner of the FPTBook, the clerks of FPTBook and the end users of the FPTBook. Other intended audience includes the development team such as the requirements team, requirements analyst, design team, and other members of the developing organization.

## 2. Product Scope

The owner of FPTBook wants to develop an online book selling system. Therefore, FPTBook has the following functions. With guest, they can sign up for an account, log in, search for books, and view specifics of any books on sale at the store. With customer, they can perform all functions of guest, manage Cart, View order history, Cancel order, and manage profile. With owner, they can they can perform all customer functions, Manage order, manage Book, manage Category. With admin, they can they can perform all owner functions , manage account, Create admin account, Accept or Cancel order, manage Publisher, View best seller books, Accept Category, Search account by email. FPTBook aims to increase the owner's profit margin by increasing sales and attracting more customers to FPTBook through an online business model and allowing optimal number of employees compared to opening a business shop.

## 3. Intended Audience and Document Overview

This document is designed for people who are involved in development, use and need to better understand the project such as developer, user, admin, and owner’s project.

To carry out system construction, the Developer must comprehend this paper. To execute the FPT Book Web, they need to be aware of the technical and functional requirements.

To utilize the system efficiently, the user must comprehend this paper. Additionally, they require this paper to provide them with system usage instructions.

To use the system and manage tasks like managing books, publishers, categories, orders, and accounts, the owner and admin must comprehend this page. They also need to be aware of the technical details and specs of the system.

This document is divided into 5 parts including:

* Section I provides Document Purpose, Product Scope, Intended Audience and Document Overview, Definitions, Acronyms and Abbreviations, References and Acknowledgments
* Section II provides Product Overview, Product Functionality
* Section III provides Functional Requirements, Non Functional Requirements, Use Case Model, Wireflow
* Section IV provides diagrams for system design including: Entity Relationship Diagram, Class Diagram, Activity Diagram, Gantt Chart
* Section V provides Risk Assessment
* Section VI provides the following Technologies Evaluation: Design Tools, Front-End Technology stack, Backend Technology stack, Source Control Management Tools, Software Development Models, and illustrate all my findings on how to use these by drawing the overview.

## 4. Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| **No.** | **Abbreviations** | **Explanation** |
| 1 | SRS | Software Requirement Specification |
| 2 | ERD | Entity Relationship Diagram |
| 3 | MVC | Model-View-Controller |
| 4 | DFD | Data Flow Diagram |
| 5 | SDLC | Software Development Life Cycle |
| 6 | CSS | Cascading Style Sheet |
| 7 | HTML | Hypertext Markup Language |
| 8 | UML | Unified Modelling Language |

## 5. References and Acknowledgments

**References**

Gaba, I., 2023. *simplilearn.* [Online]   
Available at: https://www.simplilearn.com/tutorials/git-tutorial/what-is-github

geeksforgeeks, 2019. *geeksforgeeks.* [Online]   
Available at: https://www.geeksforgeeks.org/introduction-to-c-sharp/

Jackson, P., 2023. *guru99.* [Online]   
Available at: https://www.guru99.com/what-is-php-first-php-program.html

Kelley, K., 2023. *simplilearn.* [Online]   
Available at: https://www.simplilearn.com/tutorials/git-tutorial/what-is-gitlab#:~:text=GitLab%20is%20a%20web%2Dbased,management%20to%20monitoring%20and%20security.

KUMAR, R., 2022. *devopsschool.* [Online]   
Available at: https://www.devopsschool.com/blog/what-is-gitlab-and-how-it-works-an-overview-and-its-use-cases/

Martin, M., 2023. *guru99.* [Online]   
Available at: https://www.guru99.com/what-is-sdlc-or-waterfall-model.html

Morales, J., 2022. *mindonmap.* [Online]   
Available at: https://www.mindonmap.com/blog/drawio-review/

softwaretestinghelp, 2023. *softwaretestinghelp.* [Online]   
Available at: https://www.softwaretestinghelp.com/spiral-model-what-is-sdlc-spiral-model/

# II. Overall Description

## 1. Product Overview

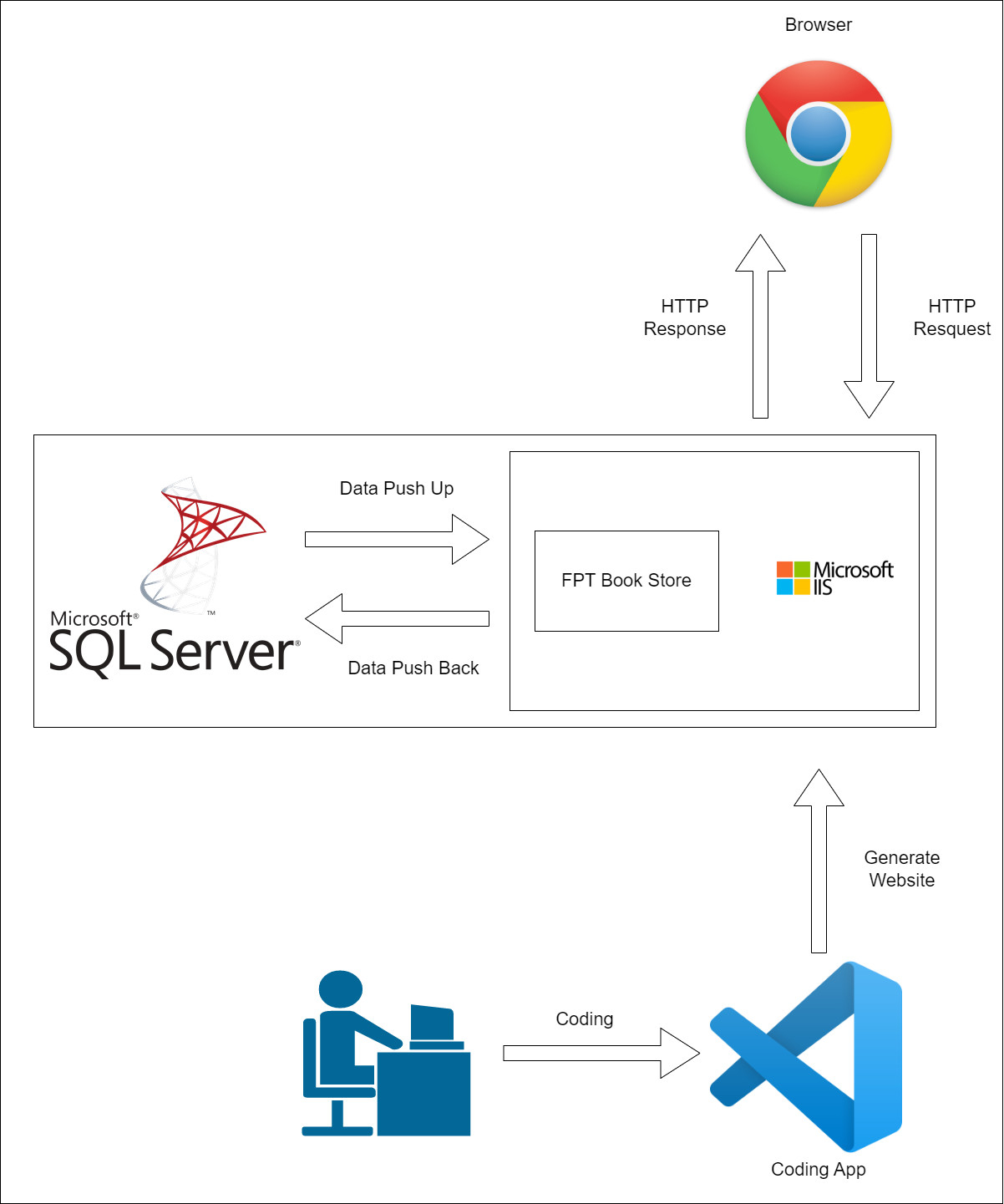


Figure . Product Overview

Nowadays, with the rapid development of technology. In addition, the social distancing due to the recent COVID-19 pandemic makes the demand for shopping and entertainment at home even more enhanced. Realizing this, FPTBook is a system that provides technology solutions for online sales. FPTBook is a website that makes it easy for users to buy goods when they are just at home and can still search and buy the books they want. It simplifies the buying, selling and payment process between customers and FPTBook. Since then, attracting more and more customers to FPTBook. In addition, the project also simplifies the management process of FPTBook administrators. When they can easily manage customer information, book information without having to check and manage it manually. All operations of customers and administrators can be performed on popular browsers such as Chrome, Opera on popular devices such as devices such as laptops, PCs, smartphones, tablets, and iPads.

## 2. Product Functionality

Function of guest:

* Register
* View all Book
* View Book Detail
* Search Book

Function of customer:

* All function of guest
* Login
* Add/Update/Delete/Order book in Cart
* View order history
* Cancel order
* Change information/ Change password/ View all profile

Function of owner:

* All function of customer
* Manage order
* View order detail
* Accept/Cancel/Update status order
* Add/Update/Delete/View Book
* Add/Update/Delete/View Category

Function of admin:

* All function of owner
* Set status/ View all account
* Create owner account
* Add/Update/Delete/View Publisher
* View best selller books
* Accept Category
* Search account by email

Update/ order số lượng quá số lượng trong kho chưa thông báo

# III. Specific Requirements

## 1. Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Type of User** | **Feature** | **Goal** |
| **1** | **Guest** | Register | The guest can register for an account to become a customer of FBTBook and perform customer functions |
| View All Book | When the guest accesses the website, they can view all book available at the FPTBook |
| View Book Detail | The guest can view detailed information of each book of FPTBook |
| Search Book | The guest can search for a book they are interested in |
| **2** | **Customer** ( the customer can use all function of guest) | Login | After successful registration, customer can login his account and perform customer functions |
| Add/Update/Delete/Order book in Cart | Customers can add products to their cart, can update products in the cart, remove products from the cart and place an order |
| View order history/ Cancel order | Customer can review all their orders and for orders pending confirmation they can cancel it |
| Change information/ Change password/ View all profile | Customers can see all their information. On this page, they can change their information and password |
| **3** | **Owner** ( the owner can use all function of customer) | Manage order / View order detail/ Accept/Cancel order | Owner can view the orders placed by the customer and view the details of that order. On the order detail page, they can accept or cancel the order |
| Add/Update/Delete/View Book | Owner can view all the books of the store. In addition, they can add a new book, update the book information, and delete the book |
| Add/Update/Delete/View Category | Owner can view all the book categories of the store. In addition, they can add a new book category, update the book category information, and delete the book category |
| **4** | **Admin** ( the owner can use all function of admin) | Set status/ View all account/ Create owner account/ Search account by email | Admin can view all accounts of customer and owner of FPTBook and search any account by email. Alternatively, they can create a new owner account and set the status for any one account. |
| Accept Category | Admin can accept any book category when the owner proceed to add a new book category |
| Add/Update/Delete/View Publisher | Owner can view all the publishers. In addition, they can add a new publisher, update the publisher information, and delete the publisher |
| View best selller books | Admin can see which book is best seller by selected time |

## 2. Non Functional Requirements

|  |  |
| --- | --- |
| **Non Functional Requirements** | **Goal** |
| Operational | - The system can run on handheld devices  - The system should be able to work on any Web browser |
| Performance | - Any interaction between the user and the system should not exceed 3 seconds  - The system should be available for use 24 hours per day , 365 days per year  - The system supports all simultaneous users |
| Security | - Only admin can accept category, manage customer information  - Functions must be clearly assigned roles and only suitable roles can use the functions corresponding to that role.  - The system includes all available safeguards from viruses , worms , Trojan horses, etc |

## 3. Use Case Model

* Notation of use case

|  |  |
| --- | --- |
| **Symbol** | **Description** |
|  | This symbol refers to a person, group, or outside system involved in one or more interactions with the system. |
| Diagram  Description automatically generated | This symbol represented a series of activities that an actor would take. |
|  | This arrow symbol represented the connections between use-cases, such as extend or include. |
|  | This symbol described associations between actors and use case |
|  | This is a rectangle around the use case, called the system boundary box |
|  | This is a descriptive notation for a set of use cases such as add, update, and delete a product. |
|  | This arrow symbol represented the connections between use-case and set of use cases |

### a. Use case of guest

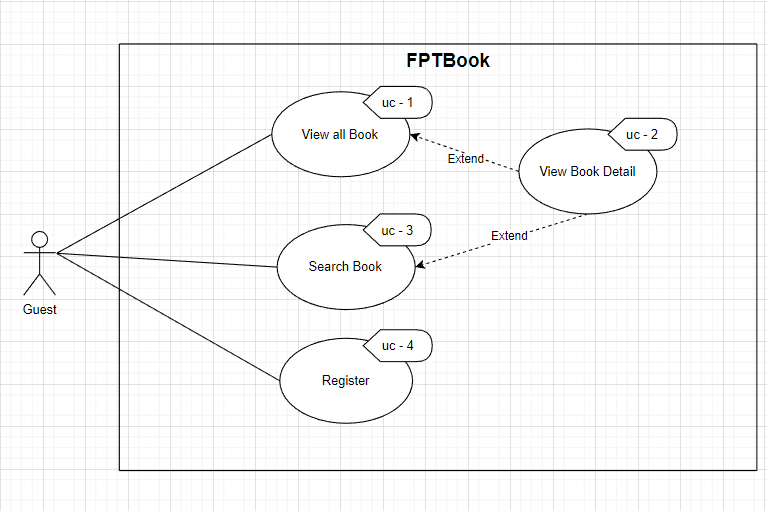


Figure . Use case of guest

* View all book

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** View all Book | **ID:** uc-1 | **Priority:** High |
| **Actor:** Guest | | |
| **Description:** The guest can view all the books in the Home page | | |
| **Trigger:**  When guest want to view all book, they can access into Home page of FPTBook  **Type:** External | | |
| **Pre-Condition:**  1. The guest’s devices must connect to internet  2. The FPT Book website must be available | | |
| **Normal Course:**  1. The guest access to the FPT Book website | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:** N/A | | |
| **Exception:** N/A | | |

* View book detail

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** View book detail | **ID:** uc-2 | **Priority:** Medium |
| **Actor:** Guest | | |
| **Description:** The guest wants to view the information detail of the book that they choose | | |
| **Trigger:** When guest want view the information detail of the book, they can click into that book  **Type:** External | | |
| **Pre-Condition:**  1. The guest’s devices must connect to internet  2. The FPT Book website must be available | | |
| **Normal Course:**  1. The guest choose any book that they want to view the information detail  2. Click on the book that they choose to view book’s detail | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:** Book detail page will be display | | |
| **Exception:** N/A | | |

* Search book

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Search Book | **ID:** uc-3 | **Priority:** Medium |
| **Actor:** Guest | | |
| **Description:** The guest wants to search the book in the FPT Book website | | |
| **Trigger:** When the guest wants to search the book in the FPT Book website, they enters name of Book and click search button  **Type:** External | | |
| **Pre-Condition:**  1. The guest’s devices must connect to internet  2. The FPT Book website must be available | | |
| **Normal Course:**  1. The guest enters name of the Book that they want to search in the search box and click on Search button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:**  1. The Book that the guest search will be display | | |
| **Exception:** There are no books that match the search guest's request | | |

* Register

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Register | **ID:** uc-4 | **Priority:** High |
| **Actor:** Guest | | |
| **Description:** Guest wants to buy goods and use all customer functions | | |
| **Trigger:** When the guest wants to buy goods and use all customer functions, they will sign up for an account  **Type:** External | | |
| **Pre-Condition:**  1. The guest’s devices must connect to internet  2. The FPT Book website must be available | | |
| **Normal Course:**  1. The guest click on the Register button in the navigation of the website  2. The guest perform filling all information that is required in the register form and click register button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:**  1. Display a successfully register message  2. Go to Login page | | |
| **Exception:**  1. The website will notify when the guest enters not enough field of register form  2. The website will notify when the guest enters wrong data | | |

### b. Use case of customer

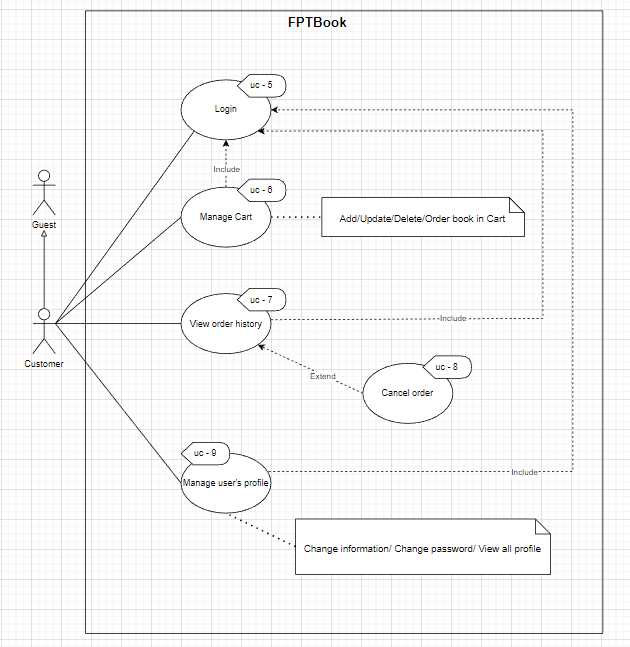


Figure . Use case of customer

* Login

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Login | **ID:** uc-5 | **Priority:** High |
| **Actor:** Customer | | |
| **Description:** Customer wants to login | | |
| **Trigger:** When the customer wants to login into their account  **Type:** External | | |
| **Pre-Condition:**  1. The customer’s devices must connect to internet  2. The FPT Book website must be available  3. They already have an account of FPTBook | | |
| **Normal Course:**  1. The guest click on the Login button in the navigation of the website  2. The guest perform filling email and password and click Submit button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:**  1. Login successfully and go to Home page | | |
| **Exception:**  1. Customer enter wrong password  2. Customer enter Wrong email | | |

* Manage Cart

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Manage Cart | **ID:** uc-6 | **Priority:** High |
| **Actor:** Customer | | |
| **Description:** Customer wants to manage their cart. They can Add/Update/Delete/Order book in Cart | | |
| **Trigger:** When customer wants to manage their cart  **Type:** External | | |
| **Pre-Condition:**  1. The customer’s devices must connect to internet  2. The FPT Book website must be available  3. Customer must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Action to manage cart | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:**  1. If customer add book into cart, that book will be have in cart  2. If customer update book in cart, information of that book will be change in cart  3. If customer delete book in cart, that book will be delete in cart  4. If customer order book, the message order success will be appeared | | |
| **Exception:**  1. Cannot update the number of books larger than the number of books in stock  2. Can't order while cart has no products | | |

* View order history

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** View order history | **ID:** uc-7 | **Priority:** High |
| **Actor:** Customer | | |
| **Description:** Customer wants to view order history | | |
| **Trigger:** When customer wants to view order history  **Type:** External | | |
| **Pre-Condition:**  1. The customer’s devices must connect to internet  2. The FPT Book website must be available  3. Customer must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click Order History button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:**  1. Order History page will be displayed | | |
| **Exception:** N/A | | |

* Cancel order

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Cancel order | **ID:** uc-8 | **Priority:** High |
| **Actor:** Customer | | |
| **Description:** Customer wants to cancel order | | |
| **Trigger:** When customer wants to cancel order  **Type:** External | | |
| **Pre-Condition:**  1. The customer’s devices must connect to internet  2. The FPT Book website must be available  3. Customer must login into FPTBook website  4. That order must have status is wait for confirmation | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click Order History button  3. Click Cancel button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:**  1. Status of order will be changed and become cancaled | | |
| **Exception:**  1. That order have status is delivering, customer cann’t cancel that order | | |

* Manage user's profile

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Manage user's profile | **ID:** uc-9 | **Priority:** High |
| **Actor:** Customer | | |
| **Description:** Customer wants to manage user's profile. They can Change information/ Change password/ View all profile | | |
| **Trigger:** When customer wants to manage their profile  **Type:** External | | |
| **Pre-Condition:**  1. The customer’s devices must connect to internet  2. The FPT Book website must be available  3. Customer must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Action to manage profile | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:** Information will be change successfully | | |
| **Exception:**  1. Customer entered wrong old password while changing password  2. customer duplicated an existing email while changing information | | |

### c. Use case of owner



Figure . Use case of owner

* Manage book

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Manage book | **ID:** uc-10 | **Priority:** High |
| **Actor:** Owner | | |
| **Description:** Owner wants to manage book. They can Add/Update/Delete book | | |
| **Trigger:** When owner wants to manage book  **Type:** External | | |
| **Pre-Condition:**  1. The owner’s devices must connect to internet  2. The FPT Book website must be available  3. Owner must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Action to manage book | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:**  1. If owner add book, that book will be have in database of FPTBook  2. If owner update book, information of that book will be change  3. If owner delete book, that book will be change status is unavailable | | |
| **Exception:** N/A | | |

* Manage order

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Manage order | **ID:** uc-11 | **Priority:** High |
| **Actor:** Owner | | |
| **Description:** Owner wants to manage order | | |
| **Trigger:** When owner wants to manage order  **Type:** External | | |
| **Pre-Condition:**  1. The owner’s devices must connect to internet  2. The FPT Book website must be available  3. Owner must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click into manage order button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:**  1. Manage order page will be display | | |
| **Exception:** N/A | | |

* View order detail

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** View order detail | **ID:** uc-12 | **Priority:** High |
| **Actor:** Owner | | |
| **Description:** Owner wants to view order detail. In addition, they can Accept/Cancel order | | |
| **Trigger:** When owner wants to view order detail, Accept/Cancel order  **Type:** External | | |
| **Pre-Condition:**  1. The owner’s devices must connect to internet  2. The FPT Book website must be available  3. Owner must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click into manage order button  3. Click detail button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:**  1. View order detail page will be display  2. Status of order will be changed if owner choose status of order and click submit | | |
| **Exception:** N/A | | |

* Set status

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Set status | **ID:** uc-13 | **Priority:** High |
| **Actor:** Owner | | |
| **Description:** Owner wants to set status of order | | |
| **Trigger:** When owner wants to set status of order  **Type:** External | | |
| **Pre-Condition:**  1. The owner’s devices must connect to internet  2. The FPT Book website must be available  3. Owner must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click into manage order button  3. Click detail button  4. Choose status for order and click submit button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:**  1. Status of order will be changed | | |
| **Exception:** N/A | | |

* Manage category

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Manage category | **ID:** uc-14 | **Priority:** High |
| **Actor:** Owner | | |
| **Description:** Owner wants to manage category. They can Update/delete/view Category | | |
| **Trigger:** When owner wants to manage category  **Type:** External | | |
| **Pre-Condition:**  1. The owner’s devices must connect to internet  2. The FPT Book website must be available  3. Owner must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Action to manage category | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:**  1. If owner view book category, all book category have in database of FPTBook will be display  2. If owner update book category, information of that book category will be change  3. If owner delete book category, that book category will be change status is unavailable | | |
| **Exception:** N/A | | |

* Add category

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Add category | **ID:** uc-15 | **Priority:** High |
| **Actor:** Owner | | |
| **Description:** Owner wants to add category | | |
| **Trigger:** When owner wants to add category  **Type:** External | | |
| **Pre-Condition:**  1. The owner’s devices must connect to internet  2. The FPT Book website must be available  3. Owner must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click Manage category button  3. Click Insert button  4. Fill in all the information of the book category and click add button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:** A book category will be added into database of FPTBook with status is unavailable | | |
| **Exception:** N/A | | |

* Accept category

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Accept category | **ID:** uc-16 | **Priority:** High |
| **Actor:** Admin | | |
| **Description:** Admin wants to accept category | | |
| **Trigger:** When admin wants to accept category  **Type:** External | | |
| **Pre-Condition:**  1. The admin’s devices must connect to internet  2. The FPT Book website must be available  3. Admin must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click Manage category button  3. Choose status of book category and click submit button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:** Status of book category will be changed | | |
| **Exception:** N/A | | |

### d. Use case of admin

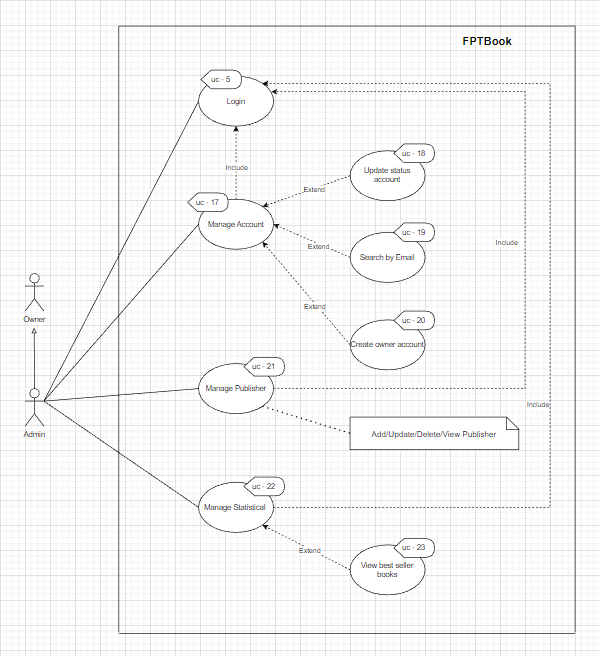


Figure . Use case of admin

* Manage account

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Manage account | **ID:** uc-17 | **Priority:** High |
| **Actor:** Admin | | |
| **Description:** Admin wants to manage account | | |
| **Trigger:** When admin wants to manage account  **Type:** External | | |
| **Pre-Condition:**  1. The admin’s devices must connect to internet  2. The FPT Book website must be available  3. Admin must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click Manage account button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:** Manage account page will be display | | |
| **Exception:** N/A | | |

* Update status account

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Update status account | **ID:** uc-18 | **Priority:** High |
| **Actor:** Admin | | |
| **Description:** Admin wants to manage account | | |
| **Trigger:** When admin wants to manage account  **Type:** External | | |
| **Pre-Condition:**  1. The admin’s devices must connect to internet  2. The FPT Book website must be available  3. Admin must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click Manage account button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:** Manage account page will be display | | |
| **Exception:** N/A | | |

* Search by email

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Search by email | **ID:** uc-19 | **Priority:** High |
| **Actor:** Admin | | |
| **Description:** Admin wants to search account by email | | |
| **Trigger:** When admin wants to search account by email  **Type:** External | | |
| **Pre-Condition:**  1. The admin’s devices must connect to internet  2. The FPT Book website must be available  3. Admin must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click Manage account button  3. Enter email and click search button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:** Information of account will be display | | |
| **Exception:** There are no account that match the search admin's request | | |

* Create owner account

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Create owner account | **ID:** uc-20 | **Priority:** High |
| **Actor:** Admin | | |
| **Description:** Admin wants to create owner account | | |
| **Trigger:** When admin wants to create owner account  **Type:** External | | |
| **Pre-Condition:**  1. The admin’s devices must connect to internet  2. The FPT Book website must be available  3. Admin must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click Manage account button  3. Click Insert button  4. Fill out all the information and click add button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:** A new account with role is owner will be added into database of FPTBook | | |
| **Exception:** N/A | | |

* Manage publisher

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Manage publisher | **ID:** uc-21 | **Priority:** High |
| **Actor:** Admin | | |
| **Description:** Admin wants to manage publisher | | |
| **Trigger:** When admin wants to manage publisher  **Type:** External | | |
| **Pre-Condition:**  1. The admin’s devices must connect to internet  2. The FPT Book website must be available  3. Admin must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Action to manage publisher | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:**  1. If admin add publisher, that publisher will be have in database of FPTBook  2. If admin update publisher, information of that publisher will be change  3. If admin delete publisher, that book will be change status is unavailable | | |
| **Exception:** N/A | | |

* Manage statistical

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Manage statistical | **ID:** uc-22 | **Priority:** High |
| **Actor:** Admin | | |
| **Description:** Admin wants to manage statistical | | |
| **Trigger:** When admin wants to manage statistical  **Type:** External | | |
| **Pre-Condition:**  1. The admin’s devices must connect to internet  2. The FPT Book website must be available  3. Admin must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click Manage statisticalbutton | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:** Manage statistical page will be display | | |
| **Exception:** N/A | | |

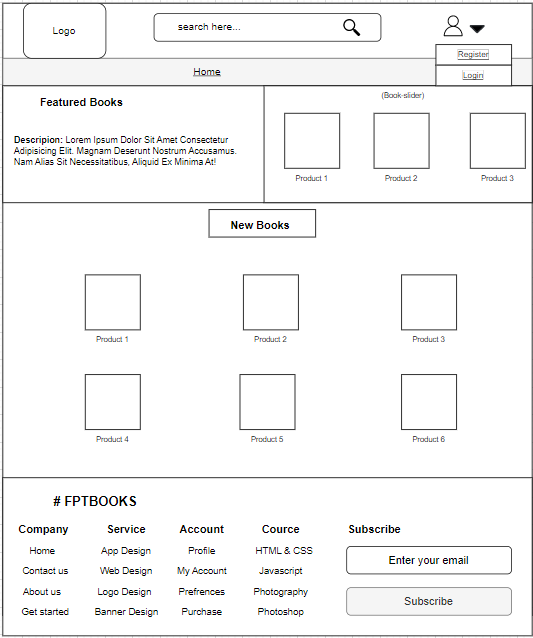
* View best seller books

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** View best seller books | **ID:** uc-23 | **Priority:** High |
| **Actor:** Admin | | |
| **Description:** Admin wants to view best seller books | | |
| **Trigger:** When admin wants to view best seller books  **Type:** External | | |
| **Pre-Condition:**  1. The admin’s devices must connect to internet  2. The FPT Book website must be available  3. Admin must login into FPTBook website | | |
| **Normal Course:**  1. Login into FPTBook website  2. Click Manage account button  3. Choose time and click submit button | | |
| **Alternative Flow:** N/A | | |
| **Post-Condition:** Information of best seller book will be display | | |
| **Exception:** N/A | | |

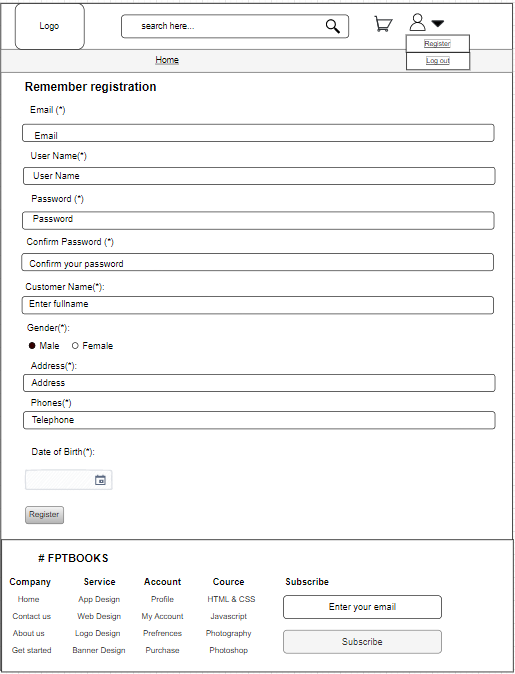
## 4. Wireflow

### a. Wireframe

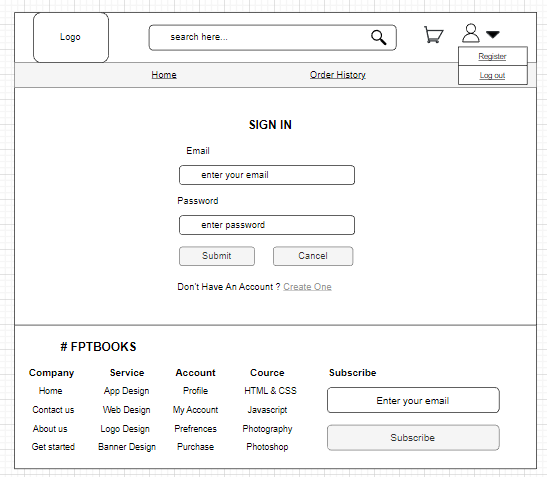
* When users access the FPTBook website, this is the first interface they see. Here, the user can browse the books and perform a search by the book's title. Additionally, consumers have the option of creating a customer account or logging in with one

****

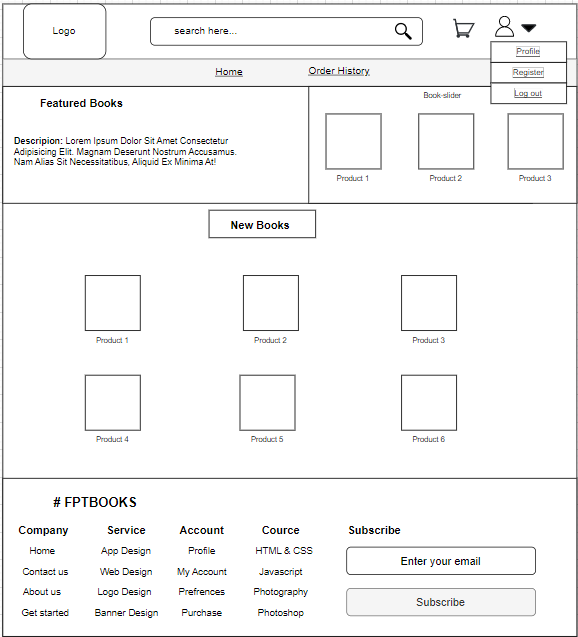
* This is the interface for creating new customer accounts. To create a new customer account, a user simply needs to fill out the registration form with their name, email address, user name, password, confirm password, customer name, gender, address, phone number, and date of birth before clicking the Register button



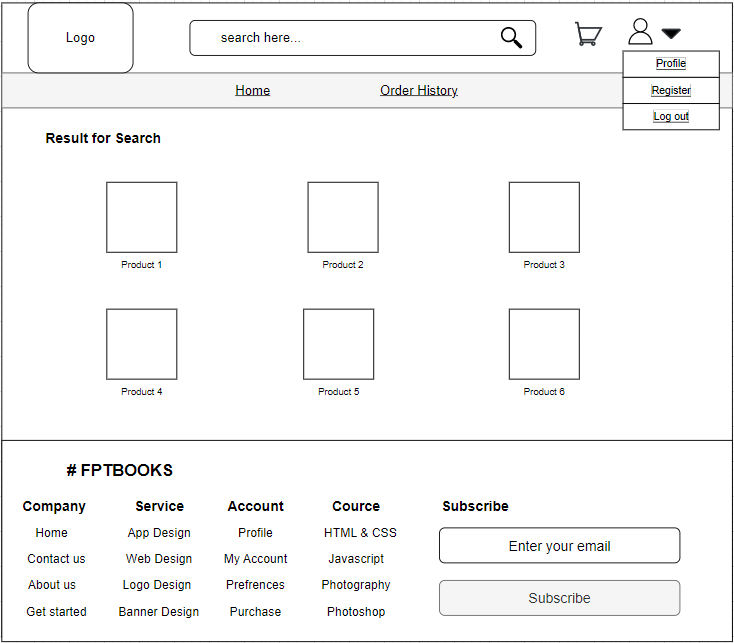
* Users only need to input their email address and password here, then click the Submit button to log in to the website



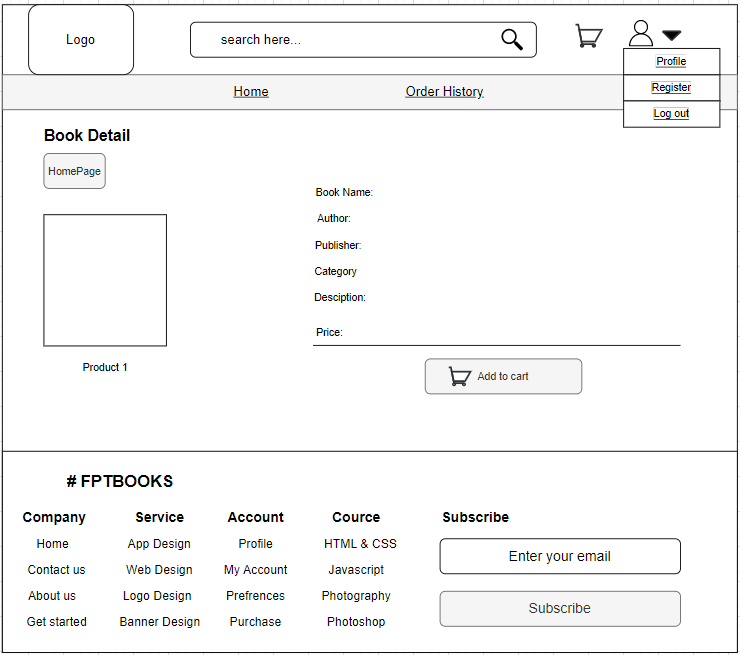
* When using a customer account to log in, you will see this as your home page interface. Here, users can view books, search for books by title, view their carts, add books, edit books in their carts, delete books from their carts, order, cancel orders that are awaiting order confirmation, examine order history, view their profiles, update their profiles, and change their passwords



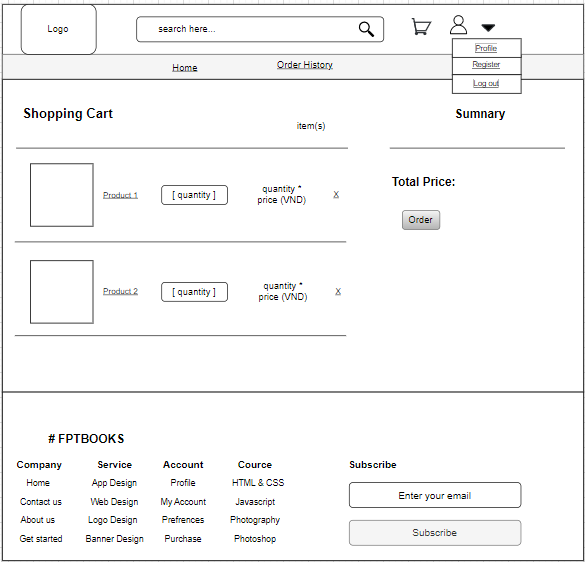
* This is the search results page; it lists books according to the keywords that the user enters into the search bar



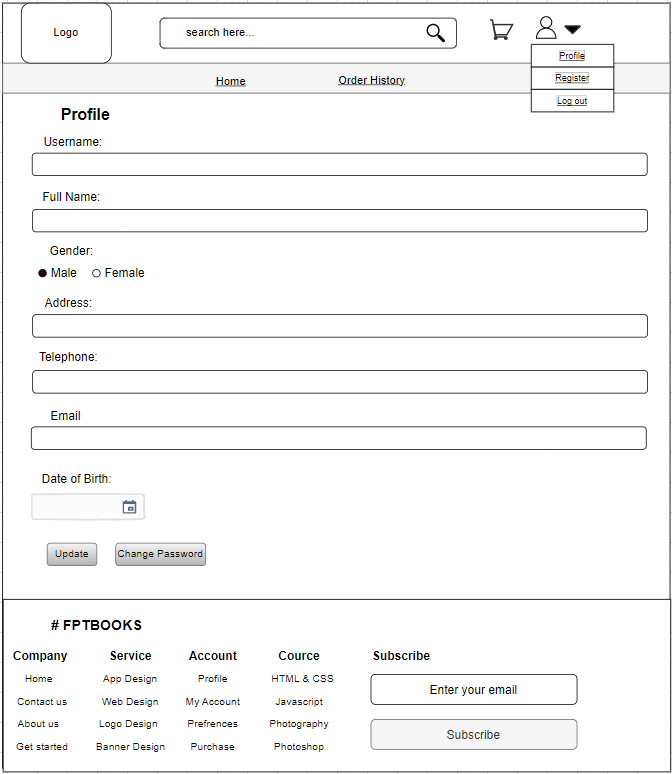
* This is the interface for the book detail; it shows the book's specific details, including its name, author, publisher, category, description, price, and cover image, and allows the user to add the book to their shopping basket.



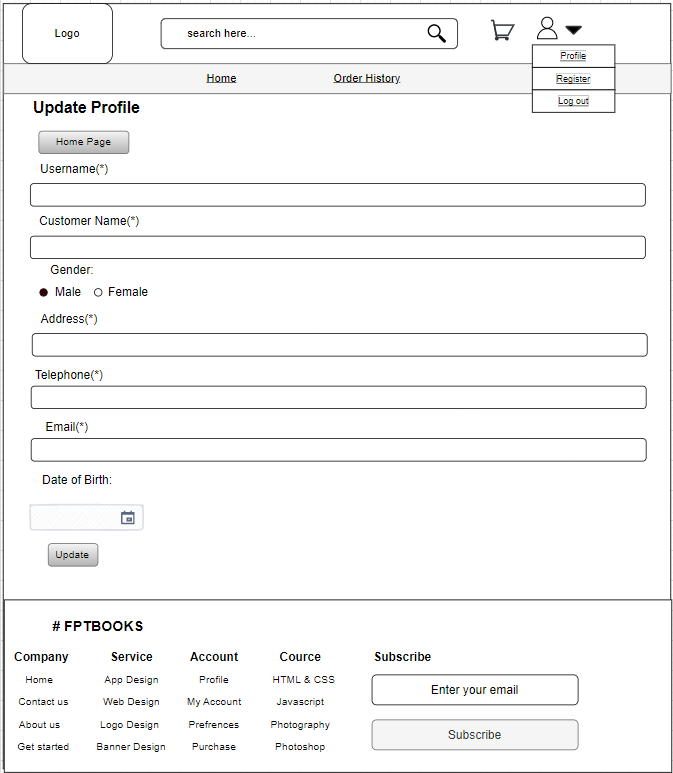
* This is the shopping cart interface, which will provide information about the books that the user has added to the cart, including the book's image, the quantity purchased, the price, and the order's total cost. Additionally, clients can delete orders or their shopping cart through this interface.



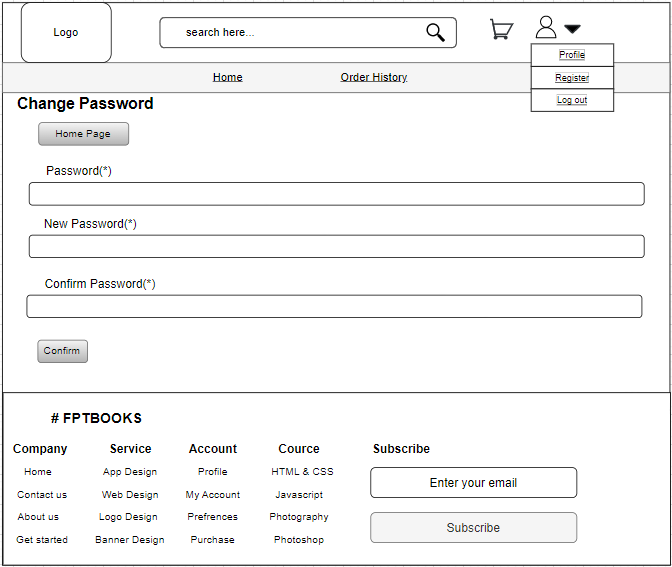
* The person's full name, user name, gender, address, phone number, email, and date of birth are all displayed in this user profile window. Users can also update the information on their profiles by clicking the Update button, which causes the system to go to the Update Profile screen. Users can also change their account's password by clicking the Change Password button, which will move the system to the password-changing interface for them.



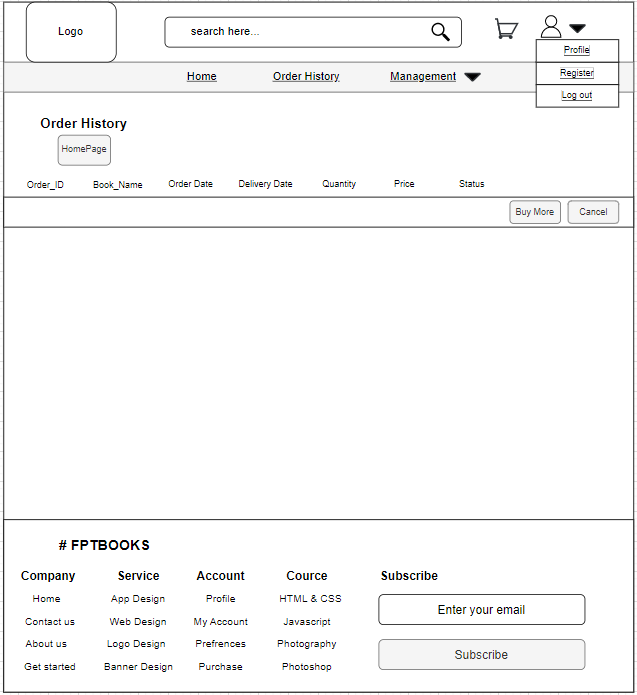
* The user only needs to make changes to the information on the form and then click the Update button to update their profile.



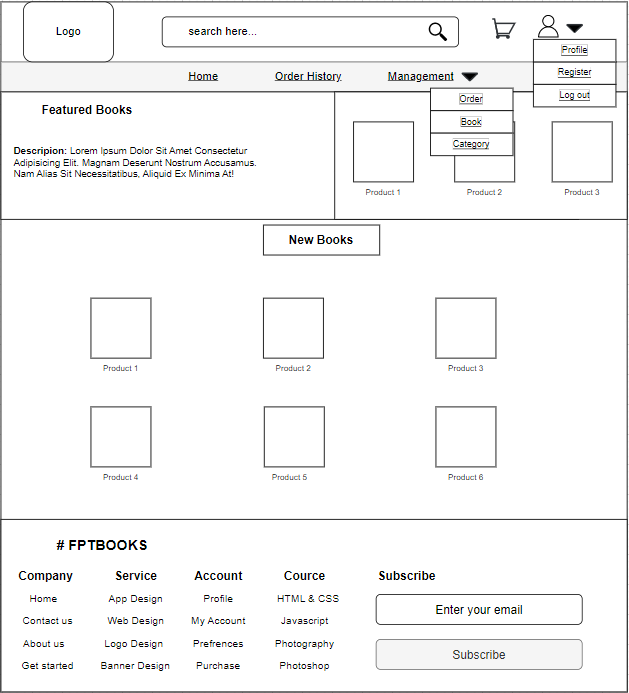
* The user only needs to enter the old password, the new password, then the new password again to confirm the new password before pressing the Confirm button to change the password.



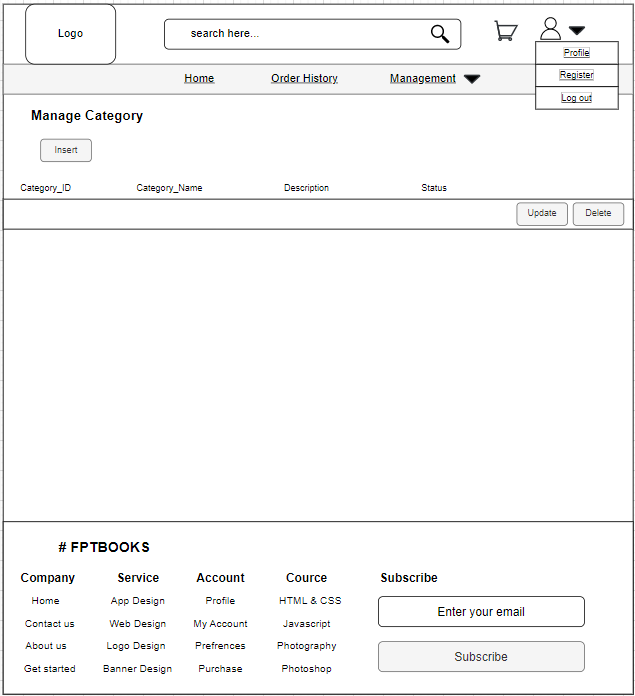
* The Order History interface shows details and the current state of orders placed by customers, including Order ID, Book Name, Order Date, Delivery Date, Quantity, Price, and Status. Additionally, by selecting the Cancel button while the order is in the status of pending confirmation, the user can cancel it. Additionally, consumers can add to their order of books by clicking the Buy more button, which will take them to the book's detail page.



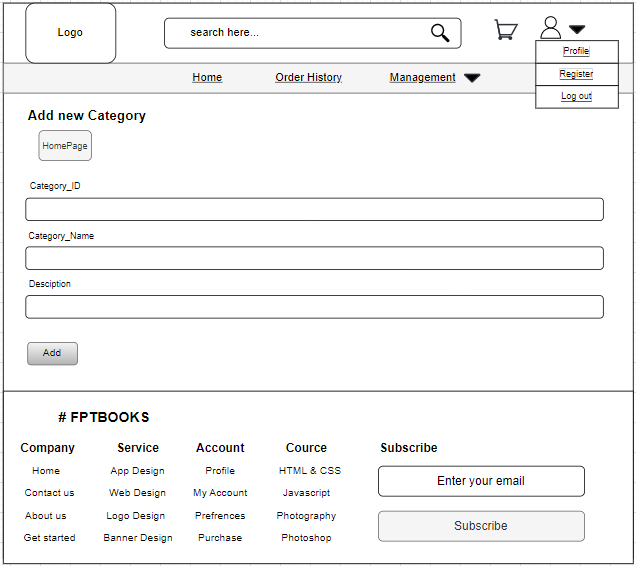
* When using the owner account to log in, this is the homepage interface. The user can access all the features available on the customer account using the owner account. Additionally, there are management tasks including managing orders, books, and categories.



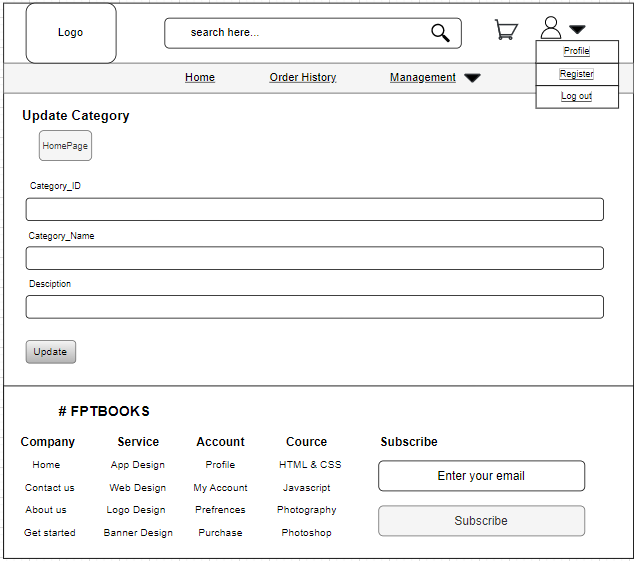
* This is the interface for managing categories; it shows data about each category, including its Category ID, Category Name, Description, and Status. By clicking the Insert, Update, or Delete buttons, users can add, update, or delete categories here.



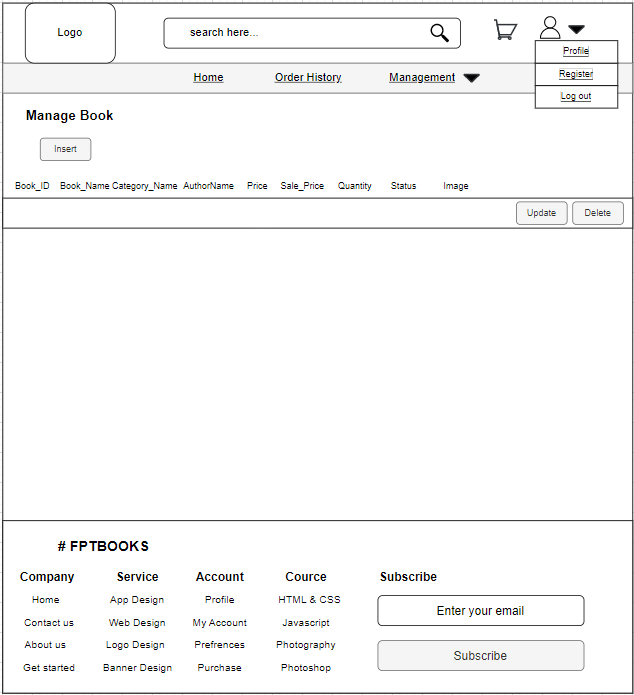
* This is the interface for adding categories. To add a category, a user must first provide its Category ID, Category Name, and Description, after which they must click the Add button. In particular, the rights of the Admin account and the Admin account used by the FPTBook store owner must be confirmed when adding a new category to the system of the FPTBook store with the owner account.



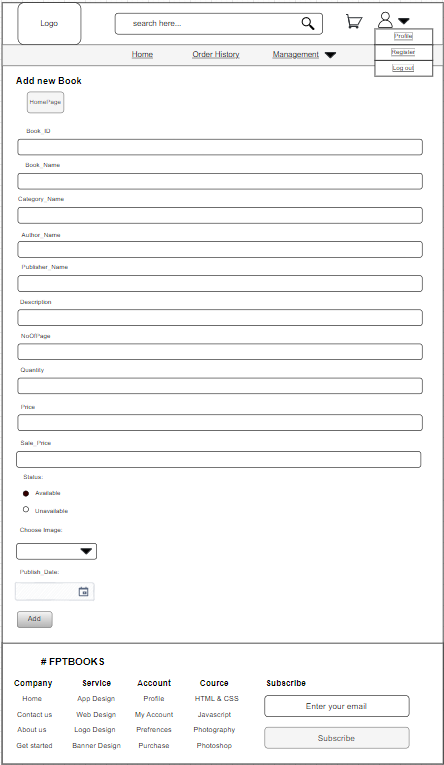
* The user must re-enter all of the category's information except for Category ID and then click the Update button in order to update the information of the category using this interface.



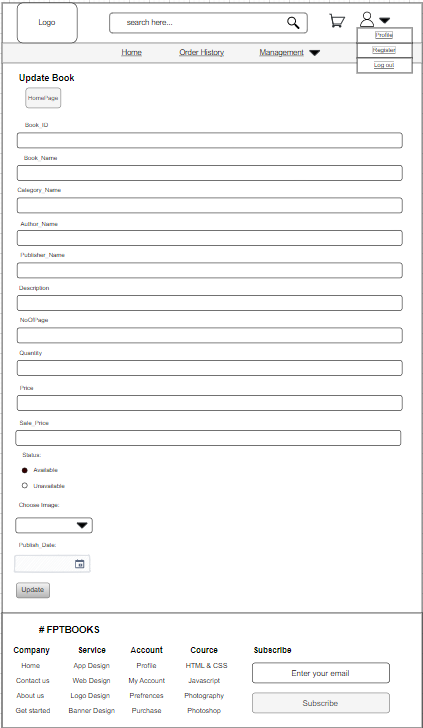
* This is the interface for managing books; it shows details about each book, including its ID, name, category, author, price, sale price, quantity, status, and image. By clicking the Insert, Update, or Delete buttons, users can add, update, or delete books from this page.



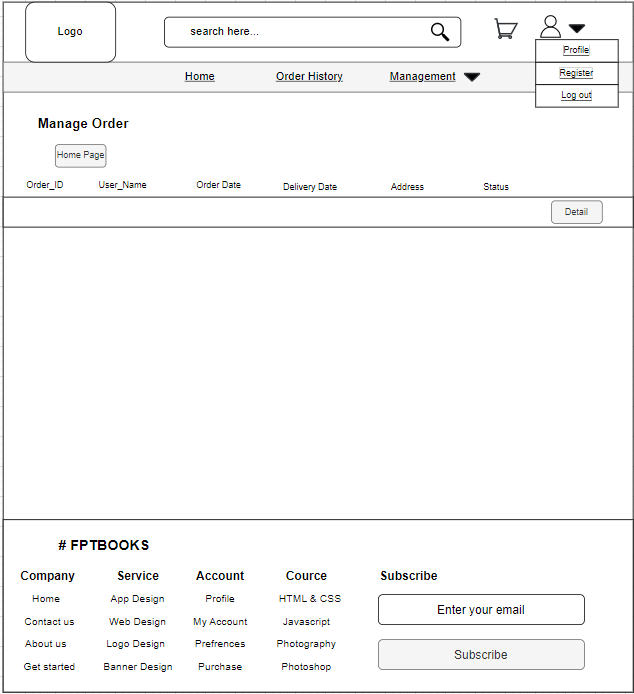
* The user simply has to enter the book's Book ID, Book Name, Category Name, AuthorName, PublisherName, Description, NoOfPage, Quantity, Price, Sale Price, Status Image, and Publisher Date in order to add a new book.



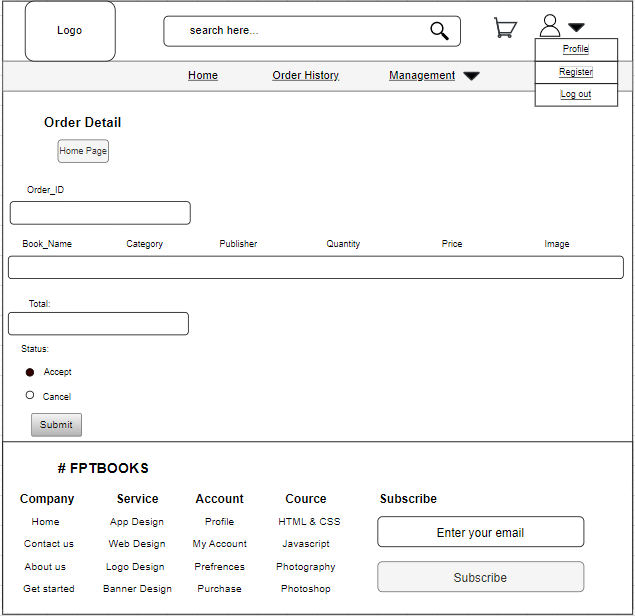
* This is the interface for updating books; to update a book, users only have to enter all of its information again, excluding Book ID, and then click the Update button.

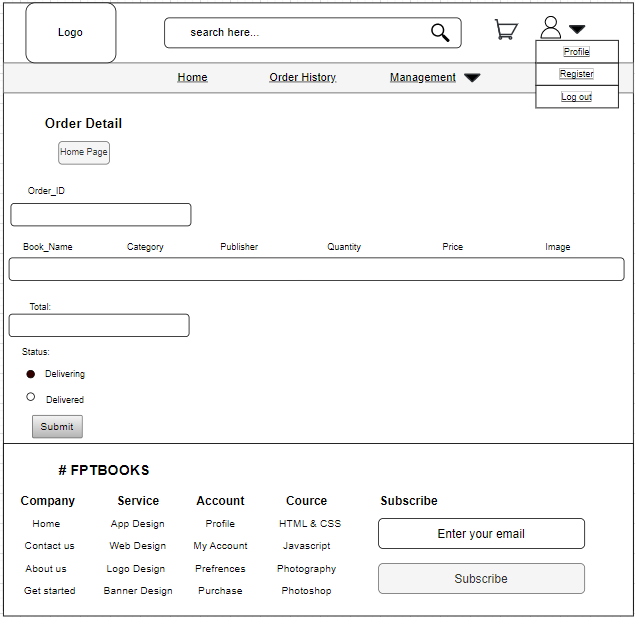


* This is the interface for managing orders; it shows details about client orders, including Order ID, User Name, Order Date, Address, and Status. Users can also view order specifics by selecting the Detail button next to that order.

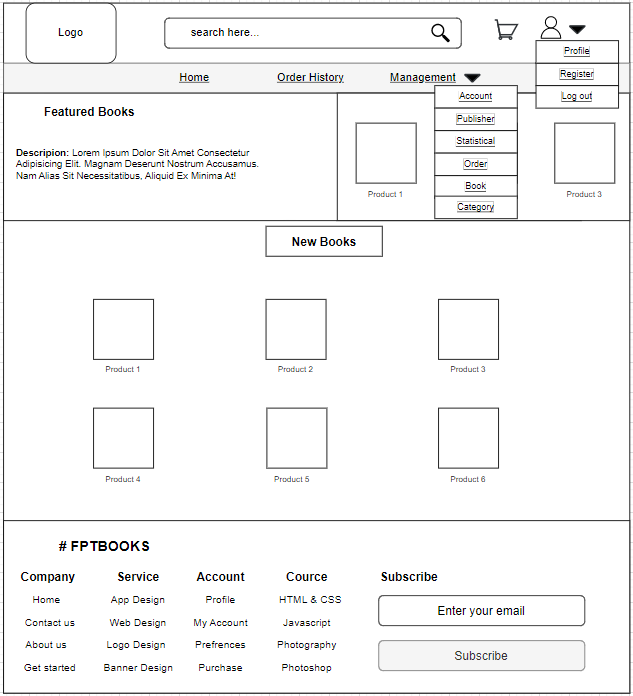


* There are two Order Detail interfaces. The first Order Detail screen is used to confirm or cancel the customer's order. The second Order Detail interface will appear if the user choose to confirm the Order and clicks the Submit button. Updates to the order's shipment or delivery status are made using this Order Detail.

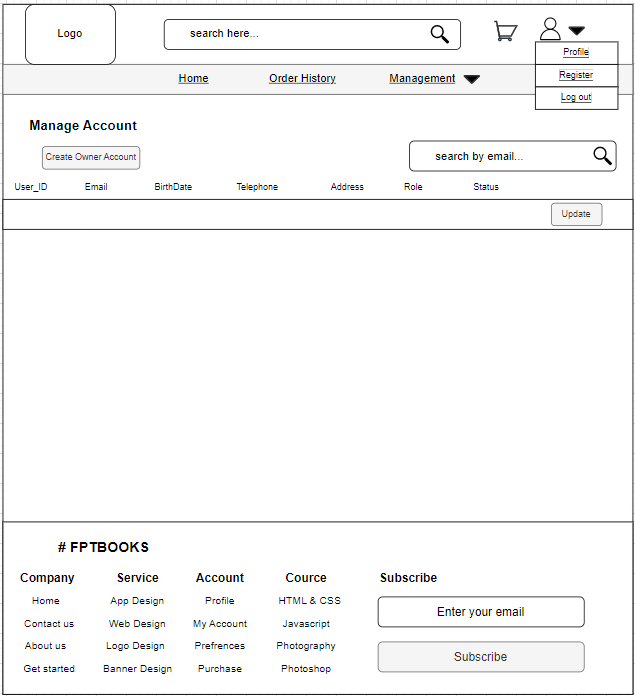




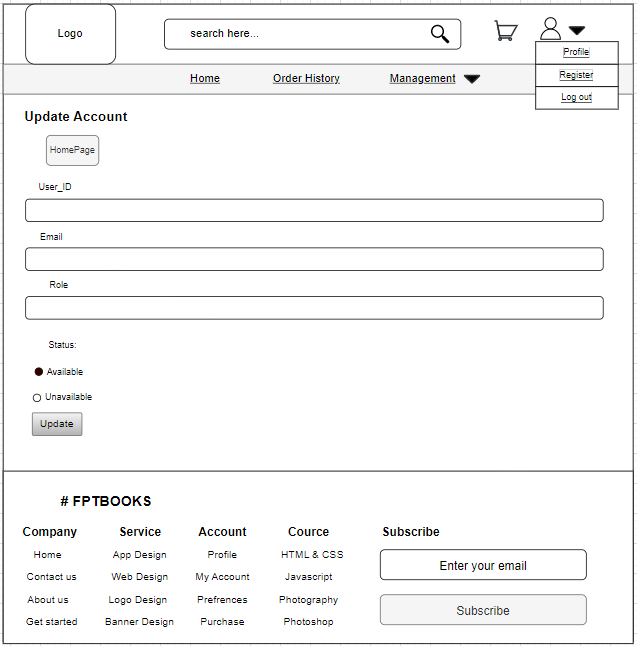
* When using the admin account to log in, this is the homepage interface. The user has access to all of the owner account's features here. In addition, there are additional administrative tasks like managing accounts, publishers, and analytics.



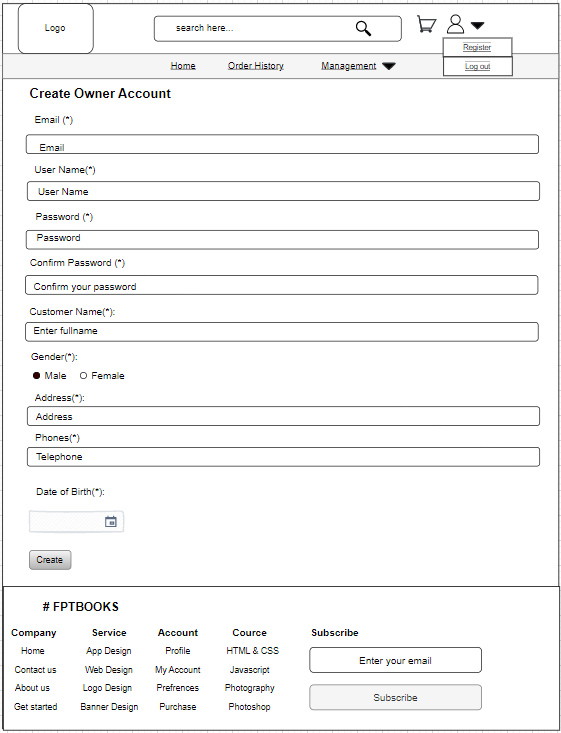
* This is the interface for managing accounts; it shows all account information, including User ID, Email, BirthDate, Telephone, Address, Role, and Status. There are more features like search by email, update status accounts, and create owner accounts. Users can utilize the Search by Email feature by entering the account's email address in the search bar and clicking the Search icon. The system will then list accounts that match the user's specified keywords. Users only need to click the Update button and Create Owner Account for the system to transition to the Update and Create Owner Account interface, where they may use the Update status Account and Create Owner Account features.



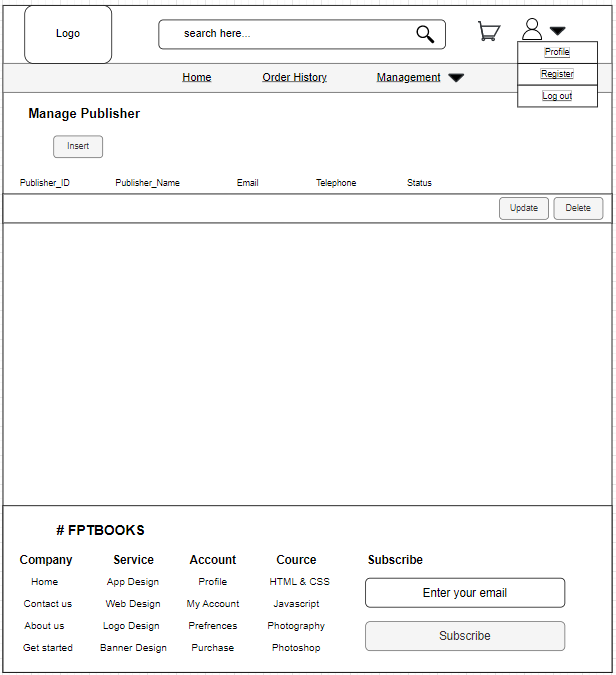
* The User ID, Email, Role, and Status of the Account are displayed on this interface, which is for updating the Account. Users must select one of the two statuses, Available or Unavailable, and then click the Update button in order to change the status of the Account.



* The customer account is this. The user merely needs to complete all the data on the registration form, including Email, User Name, Password, Confirm Password, Customer Name, Gender, Address, Number of Phone, and Date of Birth, before pressing the Create button, in order to create an owner account.



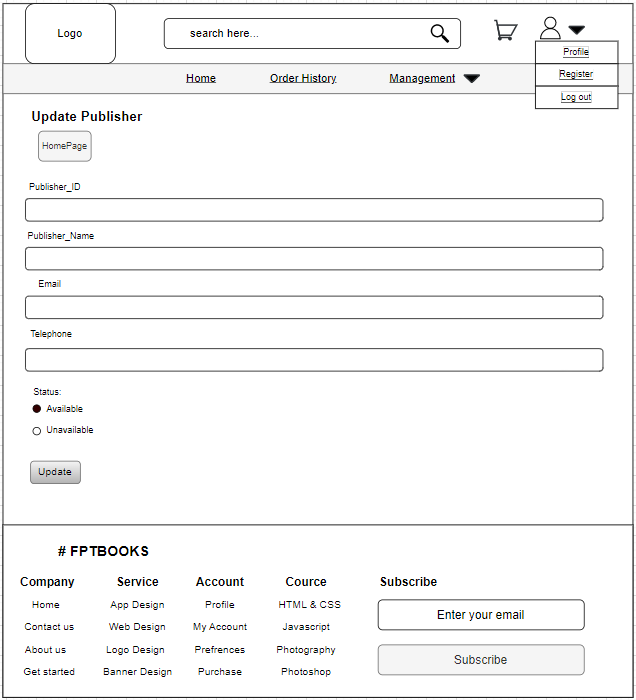
* This is the interface for managing publishers. It shows details about each publisher, including their Publisher ID, Publisher Name, email address, phone number, and status. There are features like Add, Update, and Delete in this interface. Users only need to press the Add, Update, and Delete buttons to access these features.



* Users only need to provide all the necessary details, including Publisher ID, Publisher Name, Email, Telephone, and Status, and then click the Add button to be able to add a new publisher.



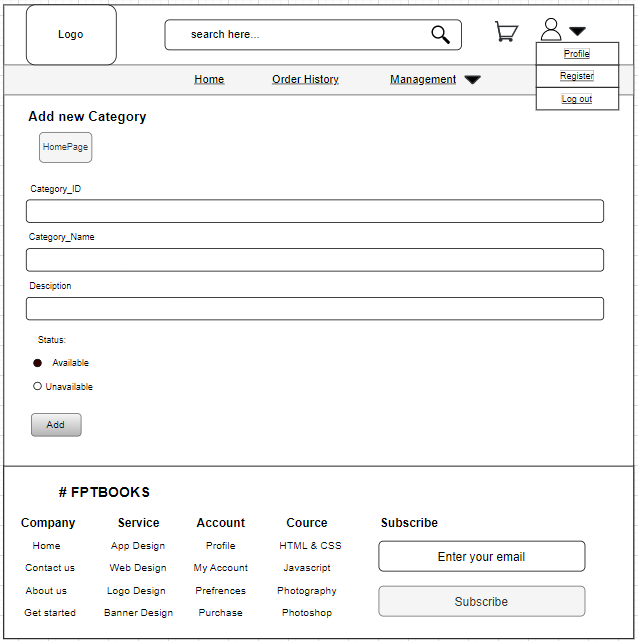
* Users only need to re-enter information such Publisher ID, Publisher Name, Email, Telephone, and Status before pressing the Update button to update Publisher. This is the Update Publisher interface.



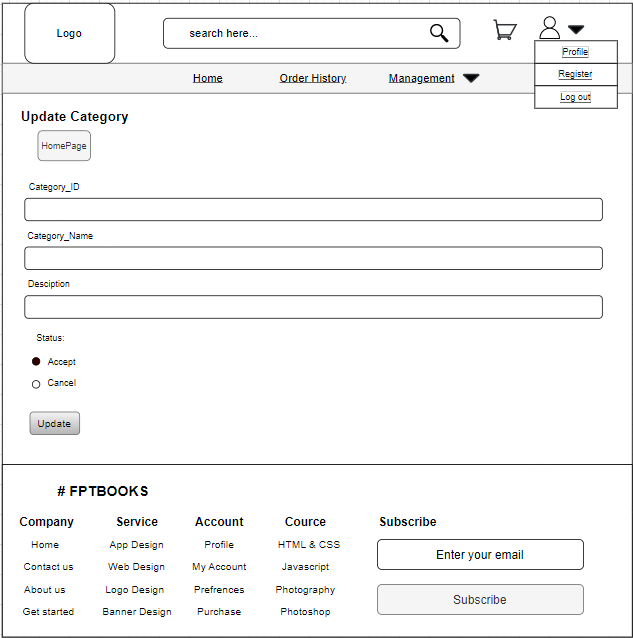
* This is the statistics management interface, which shows data such the book's ID, name, category, publisher, sell price, quantity, and sell quantity. Users have the option of selecting statistics by time, statistics sorted by high to low, or vice versa.

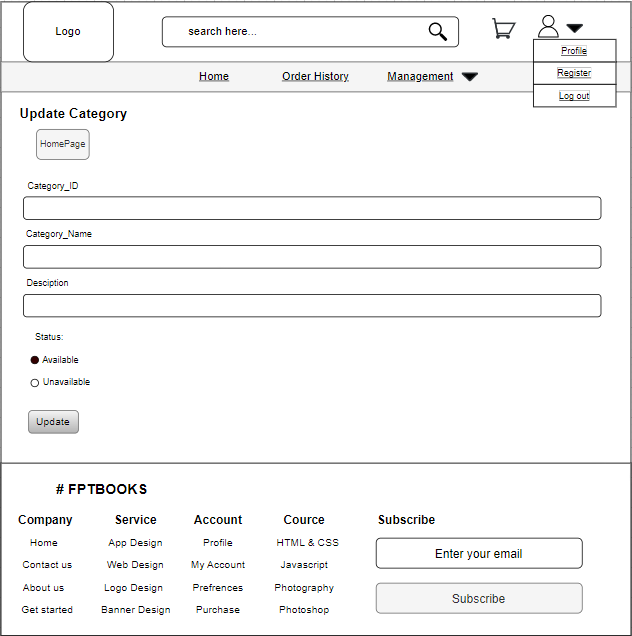


* When using an Admin account to log in, this is the Add Category interface. Users only need to enter all the necessary data, such as Category ID, Category Name, Description, and Status, and then click the Add button in order to add a new Category with an Admin account.



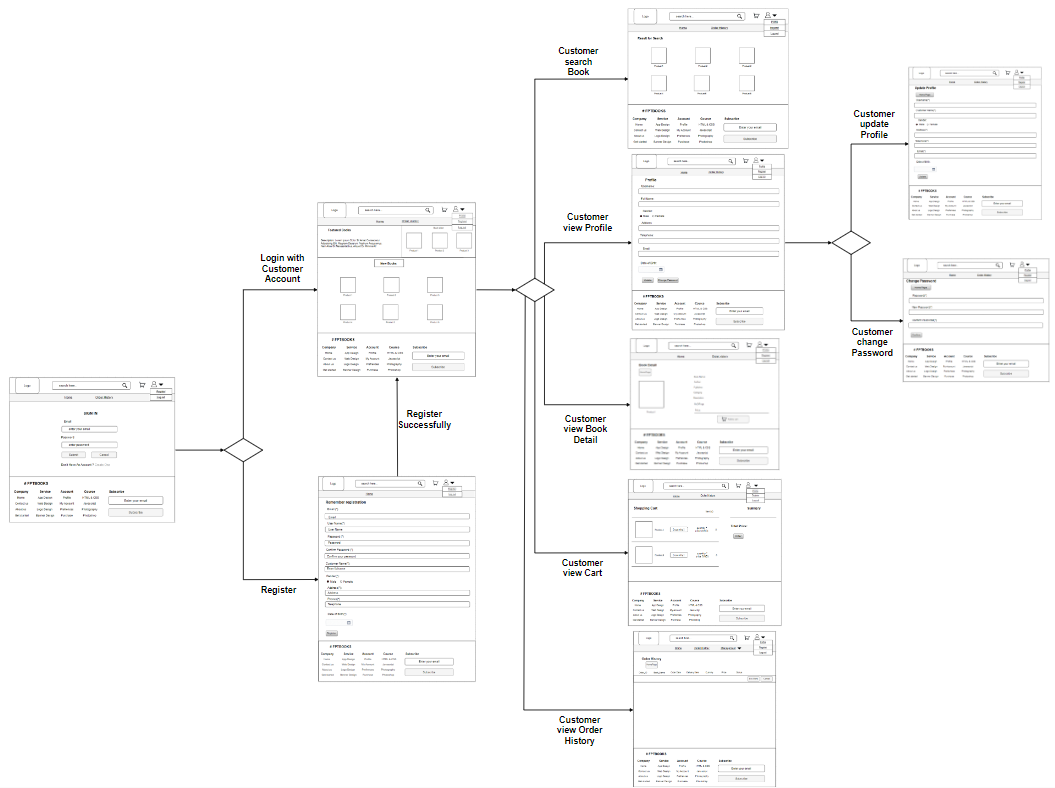
* When using an Admin account to log in, you will see these 2 Update Category interfaces. The user can accept or reject the Categories that the Owner account added to the shop and are in the status of pending confirmation using the first interface. The system will transition to the second interface after you confirm the Category. Users can update the status of the Category as Available and Unavailable using the second Update Category interface.





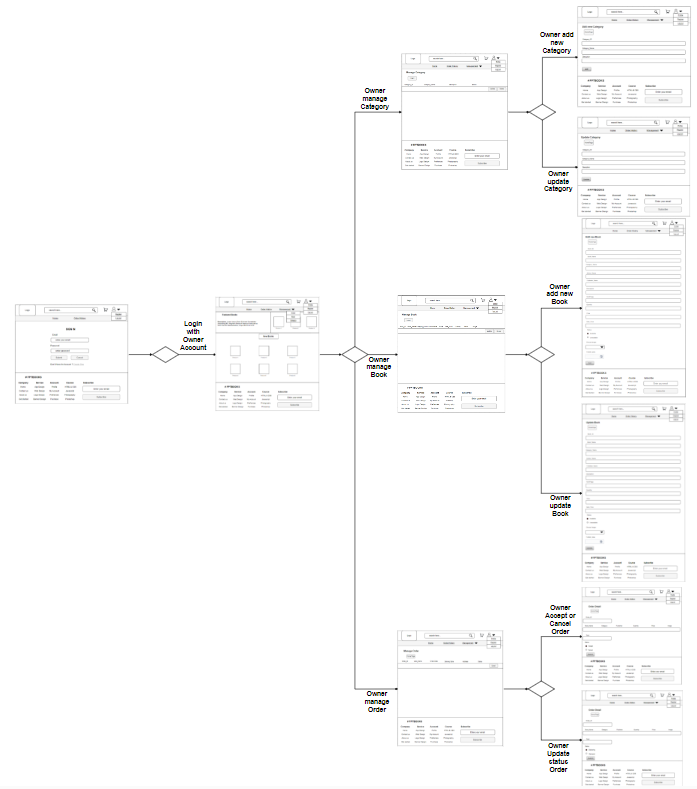
### b. Wireflow

* **Login with Customer**

****

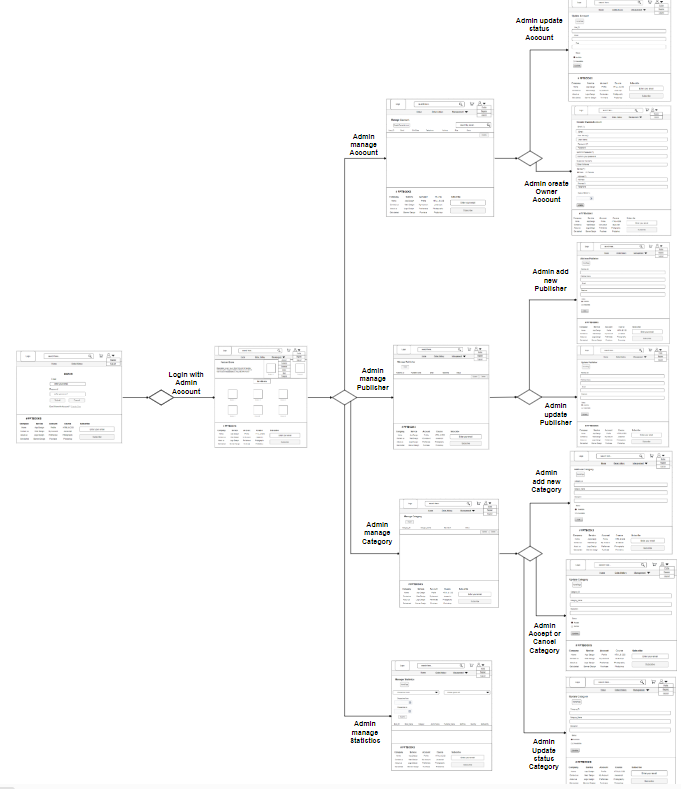
We begin by using the Login interface. The system will move to the registration interface if the user chooses the Register item on the Home Page interface. This interface will show a registration form for the user to sign up for a Customer account. The system will change to the customer's home page after successful login using the customer account. There are 5 options on the customer's homepage interface. The first is the Search feature; following the customer's search of the books on the homepage, the system will change to the Result for Search interface, which lists all the books according to the phrase the customer searched for. The system will move to the Profile interface and display the entire Customer's profile information if the Customer picks the Profile area on the homepage. The second function is the View Profile option. There are two additional alternatives available. The system will transition to the Change Profile interface after the customer selects the Update button on the Profile screen, where they can update their profile information. This is the first choice. The system will transition to the Change Password interface after the customer selects the Alter Password button in the Profile page, where he can change the password for his account. The second option is to change the username. The View Book Detail function is the next. When a user clicks on a book's name on the homepage interface, the system automatically switches to the book's Book Detail interface. The customer can add the book to their cart after viewing all of the information on this interface. The system will move to the customer's shopping cart interface when the user hits the cart icon on the home page interface, which is followed by the View Cart function. Information about the books that the customer has added to their cart will be shown on this interface. Customers may also add to, edit, and delete their orders and shopping carts. The system will display the Order History interface once the user selects the Order History item on the homepage interface for the View Order History function. The orders that the customer has placed will be shown on this interface. Additionally, the customer has the option to purchase back or cancel the order while it is still awaiting confirmation.

* **Login with Owner**



After successfully logging in with the Owner account, the system will switch to the Owner's homepage screen. All the choices available for the Customer account can be found here. There are additionally 3 choices. The system will move to the Category management interface once the Owner selects the Category management item in the first function. Three operations are Add, Update, and Delete on this interface, which shows information about Categories. The system will move to the Add Category interface and display the create category form for the Owner to add a new Category if the owner clicks the Insert button. If the owner selects the Update button, the system will switch to the Update Category interface, this interface will display the Update Category form for the owner to update. The system will move to the Change Category interface if the owner chooses the Update button. This interface will show the Update Category form for the owner to update. The system will move to the Book management interface once the Owner selects the Book management item in the second function. This interface features three functions: Add, Update, and Delete, which display information about Books. The system will move to the Add Book interface if the owner chooses the Insert button; this interface will show the add book form so the owner can add a new book. When the owner clicks the Update button, the system switches to the Update Book interface, where the owner can update the Update Book form. Finally, the system will display the Order management interface when the Owner selects the Order management item in the Order management function. Information about the customer's order will be shown on this interface. Additionally, there are functions to update the customer's order status and accept or cancel the customer's order. The system will move to the Order Update interface after the Owner clicks the Detail button on the Order Management page. This interface will provide the customer's order information, and the Owner can then accept or cancel the order before updating. The system will move to the Update Order interface after accepting an order, returning to the Order management interface, and hitting the Detail button once again on that order. Here, the Owner can change the order's status.

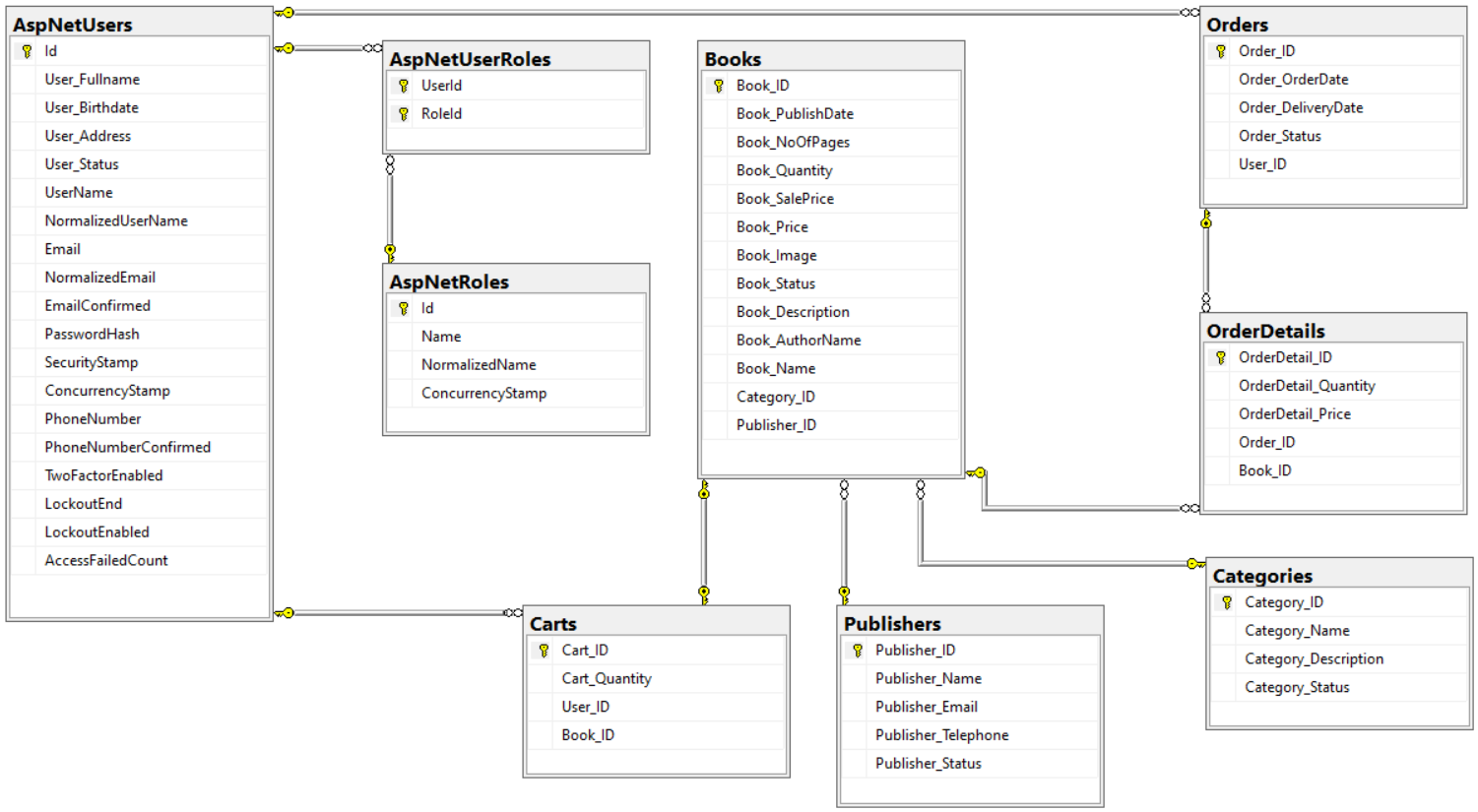
* **Login with Admin**



After successfully logging in using the Admin account, the system will switch to the Admin's Homepage interface. All the choices available in the Owner account are available here. There are additionally 4 choices. Management of accounts comes first. The system will move to the Account management interface after the administrator picks the Account management item. This interface will show all accounts and allow the administrator to search for an account by email. Additionally, there are 2 features: Create Owner account and Update status Account. The system will move to the Update Account interface after the Admin hits an account's Update button; this interface shows the account's information and lets the Admin change the account's state. The system will move to the Create Owner Account interface and display the Owner account registration form if the user hits the Create Owner Account button. The system will transition to the Publisher management interface as soon as the administrator selects the Publisher management item in the menu. Three functions—Add, Update, and Delete—are available on this interface, which shows information about the Publishers. The system will switch to the Add Publisher interface and present the add Publisher form for the admin to add a new Publisher if they choose the Insert button. The system will move to the Update Publisher interface and present the Update Publisher form for the owner to update if the admin clicks the Update button. The system will then show the Category management interface after the Admin picks the Category management item in the previous function. Information about Categories will be shown on this interface. Additionally, there are functions to update the category state and accept or cancel the category. The system will move to the Update Category interface once the Admin hits the Update button on the Category management interface. This interface will display the Category information, and the Admin can accept or reject the Category before updating. The system will transition to the Update Category interface where the Admin can update the status of the Category after accepting a Category, going back to the Category management page, and then pressing the Update button once more on that Category. Finally, manage Statistics. The system will transition to the Statistics administration interface after the Admin clicks on the Statistics management item. This interface will display sales data for books, and it will also allow the administrator to compile statistics over time.

# IV. Technical Design

## 1. Entity Relationship Diagram (ERD)



* AspNetUsers

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Data Type** | **Constraint** |
| 1 | Id | nvarchar(450), Primary key | Not null |
| 2 | User\_Address | nvarchar(100) | Not null |
| 3 | User\_Fullname | nvarchar(MAX) | Allow Null |
| 4 | User\_Birthdate | datetime2(7) | Not null |
| 5 | User\_Status | int |  |
| 6 | UserName | nvarchar(256) | Allow Null |
| 7 | NormalizedUserName | nvarchar(256) | Allow Null |
| 8 | Email | nvarchar(256) | Allow Null |
| 9 | NormalizedEmail | nvarchar(256) | Allow Null |
| 10 | EmailConfirmed | bit | Not null |
| 11 | PasswordHash | nvarchar(MAX) | Allow Null |
| 12 | SecurityStamp | nvarchar(MAX) | Allow Null |
| 13 | ConcurrencyStamp | nvarchar(MAX) | Allow Null |
| 14 | PhoneNumber | nvarchar(MAX) | Allow Null |
| 15 | PhoneNumberConfirmed | bit | Not null |
| 16 | TwoFactorEnabled | bit | Not null |
| 17 | LockoutEnd | datetimeoffset(7) | Allow Null |
| 18 | LockoutEnabled | bit | Allow Null |
| 19 | AccessFailedCount | int | Allow Null |

* AspNetUserRoles

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Data Type** | **Constraint** |
| 1 | UserId | nvarchar(450), Foreign key of AspNetUsers table | Not null |
| 2 | RoleId | nvarchar(450), Foreign key of AspNetRoles table | Not null |
| UserId and RoleId are the primary key of AspNetUserRoles table | | | |

* AspNetRoles

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Data Type** | **Constraint** |
| 1 | Id | nvarchar(450), Primary key | Not null |
| 2 | Name | nvarchar(256) | Allow Null |
| 3 | NormalizedName | nvarchar(256) | Allow Null |
| 4 | ConcurrencyStamp | nvarchar(MAX) | Allow Null |

* Books

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Field Name** | **Data Type** | **Constraint** |
| 1 | Book\_ID | int, Primary key | Not null |
| 2 | Book\_PublishDate | datetime2(7) | Not null |
| 3 | Book\_NoOfPages | int | Not null |
| 4 | Book\_Quantity | int | Not null |
| 5 | Book\_SalePrice | decimal | Not null |
| 6 | Book\_Price | decimal | Not null |
| 7 | Book\_Image | nvarchar(MAX) | Not null |
| 8 | Book\_Status | int | Not null |
| 9 | Book\_Description | nvarchar(MAX) | Not null |
| 10 | Book\_AuthorName | nvarchar(MAX) | Not null |
| 11 | Book\_Name | nvarchar(MAX) | Not null |
| 12 | Category\_ID | int, Foreign key of Category table | Not null |
| 13 | Publisher\_ID | int, Foreign key of Publisher table | Not null |

* Carts

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Field Name** | **Data Type** | **Constraint** |
| 1 | Cart\_ID | Int, Primary key | Not null |
| 2 | Cart\_Quantity | int | Not null |
| 3 | User\_ID | nvarchar(450), Foreign key of AspNetUsers table | Not null |
| 4 | Book\_ID | Int, Foreign key of Book table | Not null |

* Orders

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Field Name** | **Data Type** | **Constraint** |
| 1 | Order\_ID | Int, Primary key | Not null |
| 2 | Order\_OrderDate | datetime2(7) | Not null |
| 3 | Order\_DeliveryDate | datetime2(7) | Not null |
| 4 | Order\_Status | int | Not null |
| 5 | User\_ID | nvarchar(450), Foreign key of AspNetUsers table | Not null |

* OrderDetails

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Field Name** | **Data Type** | **Constraint** |
| 1 | OrderDetail\_ID | Int, Primary key | Not null |
| 2 | OrderDetail\_Quantity | int | Not null |
| 3 | OrderDetail\_Price | Int, decimal | Not null |
| 4 | Order\_ID | Int, Foreign key of Order table | Not null |
| 5 | Book\_ID | Int, Foreign key of Book table | Not null |

* Publishers

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Field Name** | **Data Type** | **Constraint** |
| 1 | Publisher\_ID | Int, Primary key | Not null |
| 2 | Publisher\_Name | nvarchar(MAX) | Not null |
| 3 | Publisher\_Email | nvarchar(MAX) | Not null |
| 4 | Publisher\_Telephone | nvarchar(MAX) | Not null |
| 5 | Publisher\_Status | int | Not null |

* Categories

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Field Name** | **Data Type** | **Constraint** |
| 1 | Category\_ID | Int, Primary key | Not null |
| 2 | Category\_Name | nvarchar(MAX) | Not null |
| 3 | Category\_Description | nvarchar(MAX) | Not null |
| 4 | Category\_Status | int | Not null |

Relationships between tables:

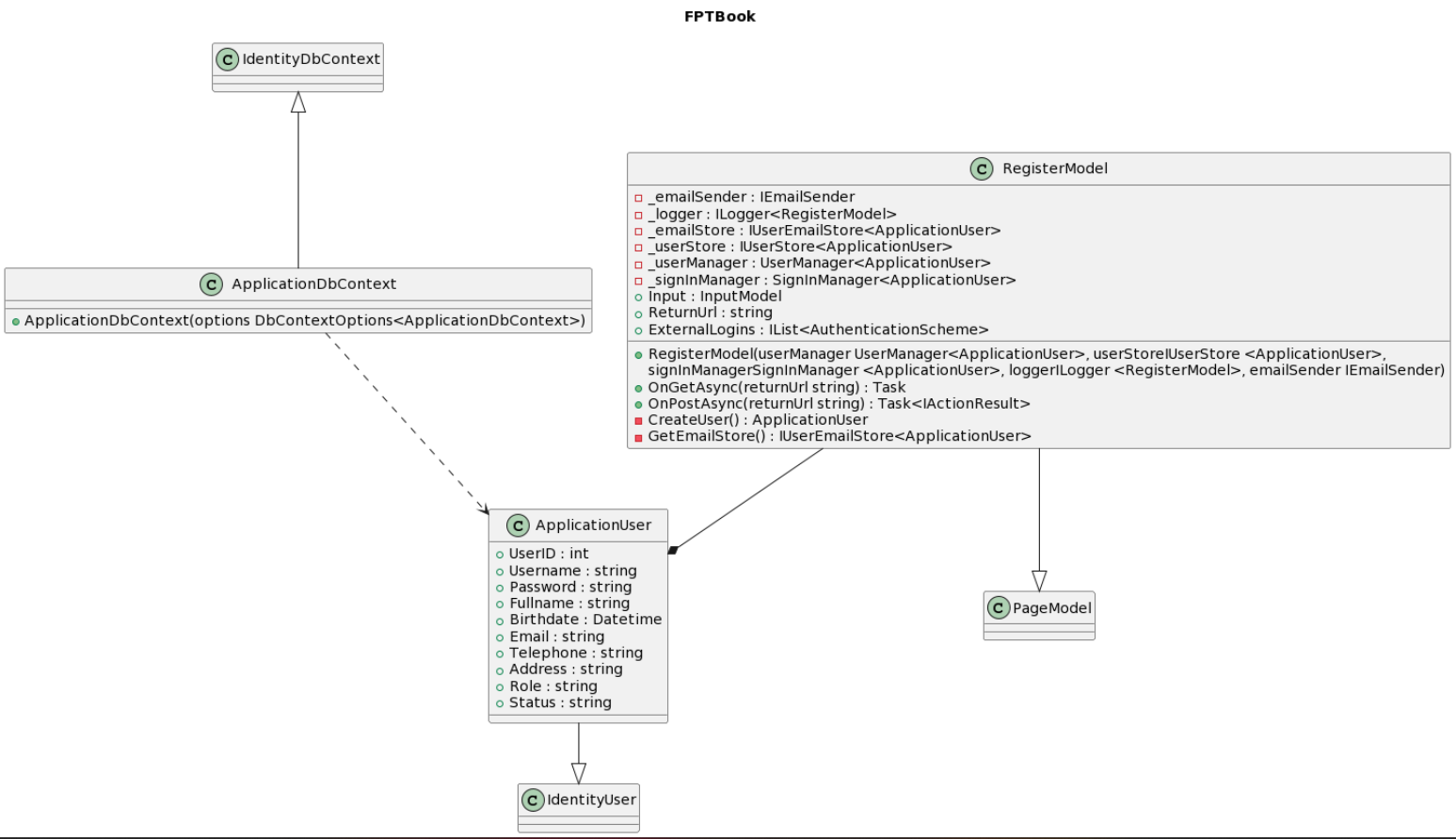
* AspNetUsers – AspNetUserRoles( 1 : n ): One AspNetUsers will have many AspNetUserRoles. In contrast, one AspNetUserRoles will have one AspNetUsers.
* AspNetUsers – Carts( 1 : n ): One AspNetUsers will have many AspNetUserRoles. In contrast, one AspNetUserRoles will have one AspNetUsers.
* AspNetUsers – Orders( 1 : n ): One AspNetUsers will have many Orders. In contrast, one Orders will have one AspNetUsers.
* AspNetRoles – AspNetUserRoles( 1 : n ): One AspNetRoles will have many AspNetUserRoles. In contrast, one AspNetUserRoles will have one AspNetRoles.
* Orders – OrderDetails( 1 : n ): One Orders will have many OrderDetails. In contrast, one OrderDetails will have one Orders.
* Books - OrderDetails( 1 : n ): One Books will have many OrderDetails. In contrast, one OrderDetails will have one Books.
* Publishers – Books( 1 : n ): One Publishers will have many Books. In contrast, one Books will have one Publishers.
* Categories - Books( 1 : n ): One Categories will have many Books. In contrast, one Books will have one Categories.
* Books – Carts( 1 - 1 ): One Carts will have One Books. In contrast, one Books will have one Carts.

## 2. Class Diagram

* Notation of Class Diagram

|  |  |
| --- | --- |
| **Symbol** | **Description** |
| https://plantuml.com/img/private-method.png | Access modifier is private for method |
| https://plantuml.com/img/private-field.png | Access modifier is private for properties |
| https://plantuml.com/img/public-method.png | Access modifier is public for method |
| https://plantuml.com/img/public-field.png | Access modifier is public for properties |
|  | This is class |
|  | Meaningful arrows: Class02 inherited Class01 |
|  | Meaningful arrows: Class15 dependent Class16 |
|  | Meaningful arrows: Class04 composition Class03 |

### a. Class diagram of register function

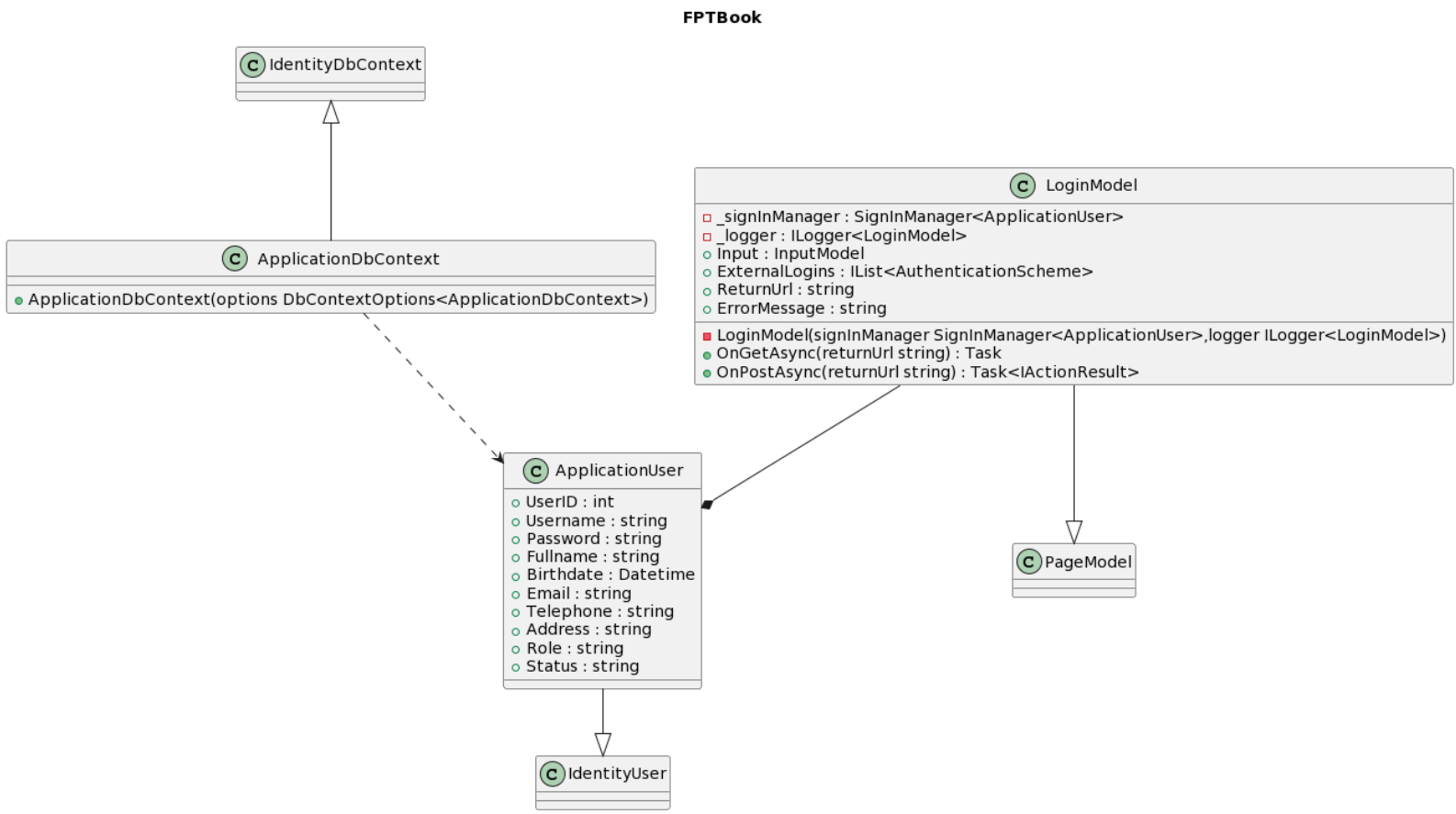


- ApplicationDbContext class: This class inherits the IdentityDbContext class, has the constructor of the class and depends on the ApplicationUser class.

- ApplicationUser class: has properties such as: UserID, Username, Password, Fullname, Birthdate, Email, Telephone, Address, Role, Status with data types respectively int, string, string, string, Datetime, string, string , string, string, int. Also, it inherits the IdentityUser class. Furthermore, it has a relationship that if this class is lost then the RegisterModel class cannot be used

- RegisterModel class: This class inherits the PageModel class. In addition, it has properties such as \_emailSender, \_logger, \_emailStore, \_userStore, \_userManager, \_signInManager with data types IEmailSender, ILogger<RegisterModel>, IUserEmailStore<ApplicationUser>, IUserStore<ApplicationUser>, UserManager<ApplicationUser>, SignInManager<ApplicationUser> respectively with private accessor, and Input, ReturnUrl, ExternalLogins properties have data types InputModel, string, IList<AuthenticationScheme> respectively with public access method. In addition, this class has RegisterModel constructors whose function is UserManager<ApplicationUser>, IUserStore<ApplicationUser>, SignInManager<ApplicationUser>, ILogger<RegisterModel>, IEmailSender, OnGetAsync function that takes returnUrl as argument. with a data type of string and a return type of , the OnPostAsync function has an input parameter of returnUrl with a data type of string and a return type of Task<IActionResult>. They have access methods that are private. In addition, the CreateUser() function has a return type of ApplicationUser and the GetEmailStore() has a return type of IUserEmailStore<ApplicationUser> whose access method is public.

### b. Class diagram of login function

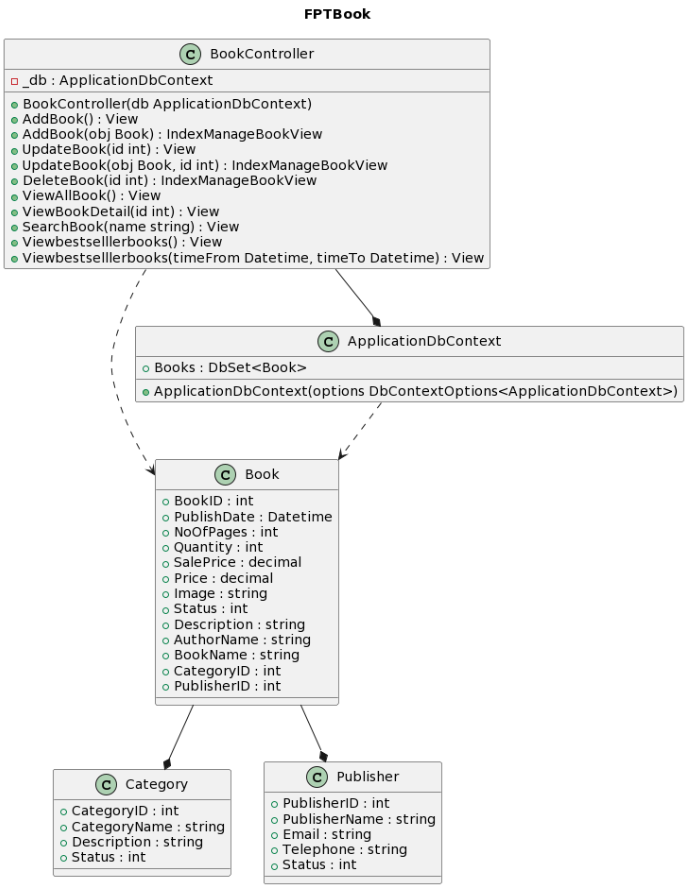


- ApplicationDbContext class: This class inherits the IdentityDbContext class, has the constructor of the class and depends on the ApplicationUser class.

- ApplicationUser class: has properties such as: UserID, Username, Password, Fullname, Birthdate, Email, Telephone, Address, Role, Status with data types respectively int, string, string, string, Datetime, string, string , string, string, int. Also, it inherits the IdentityUser class. Furthermore, it has a relationship that if this class is lost then the LoginModel class cannot be used

- LoginModel class: This class inherits the PageModel class. In addition, it has properties such as \_signInManager, \_logger with data type SignInManager<ApplicationUser>, ILogger<LoginModel> with private accessor method, and Input, ExterrnalLogins, ReturnUrl, ErrorMessage properties with data type times turn is IInputModel, IList<AuthenticationScheme>, string, string with public access method. In addition, this class has LoginModel constructors whose input function is SignInManager<ApplicationUser>, ILogger<LoginModel> with private accessor method, and OnGetAsync function whose input parameter is returnUrl with data type of string and has a return type of , the OnPostAsync function has an input parameter of returnUrl with a data type of string and a return type of Task<IActionResult> with an access method of public.

### c. Class diagram of functions of BookController class



- BookController class: has a \_db property with data type ApplicationDbContext with private access method. The BookController constructor takes the ApplicationDbContext as the parameter. There are functions AddBook, AddBook with parameters passed in Book, AddBook with parameters passed in Book, UpdateBook with parameters passed in int, UpdateBook with parameters passed in Book and int, DeleteBook with parameters passed in are int, ViewAllBook, ViewBookDetail with int as parameter, SearchBook with string as input, Viewbestselllerbooks, Viewbestselllerbooks with Datetime and Datetime as input parameters respectively View, IndexManageBookView, View, IndexManageBookView, IndexManageBookView, View, View, View, View. They have a return type of public. This class has a dependency relationship with the Book class, and if the ApplicationDbContext class is lost, the class cannot be used.

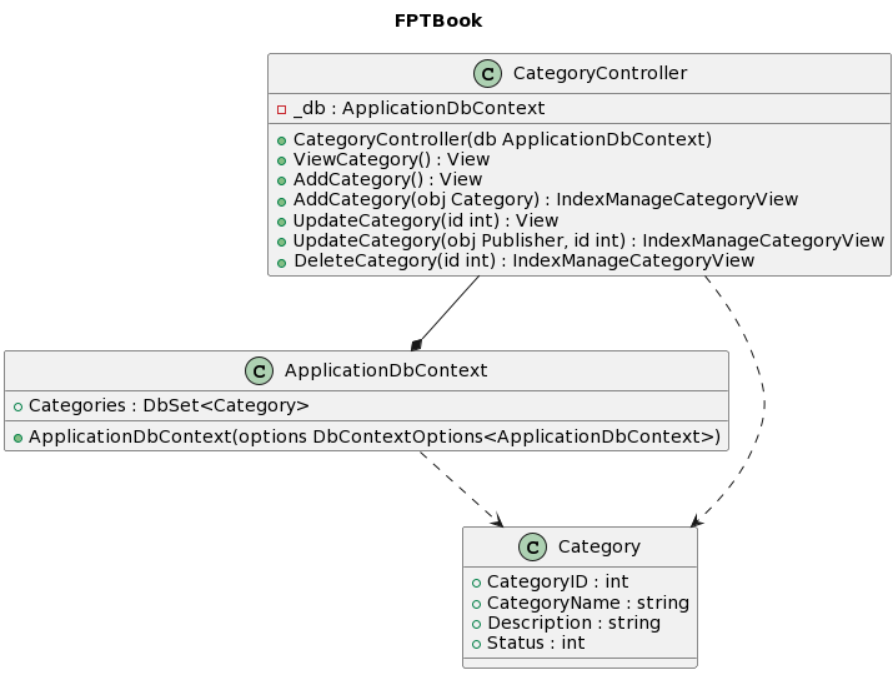
- ApplicationDbContext class: has a Books property with a data type of DbSet<Book> with an access method of public, has the constructor of the class. In addition, this class has a dependency relationship with the Book class

- Book class: has properties BookID, PublishDate, NoOfPages, Quantity, SalePrice, Price, Image, Status, Description, AuthorName, BookName, CategoryID, PublisherID, respectively, data types are int, Datetime, int, int , decimal, decimal, string, int, string, string, string have public access method. It has a relationship with Category class and Publisher class that if any class is lost, this class will not be usable

- Category class: has properties CategoryID, CategoryName, Description, Status respectively with data types of int, string, string, int respectively with public access method

- Publisher class: has properties PublisherID, PublisherName, Email, Telephone, Status, respectively, with data types of int, string, string, string, int respectively with public access methods

### d. Class diagram of functions of CategoryController class

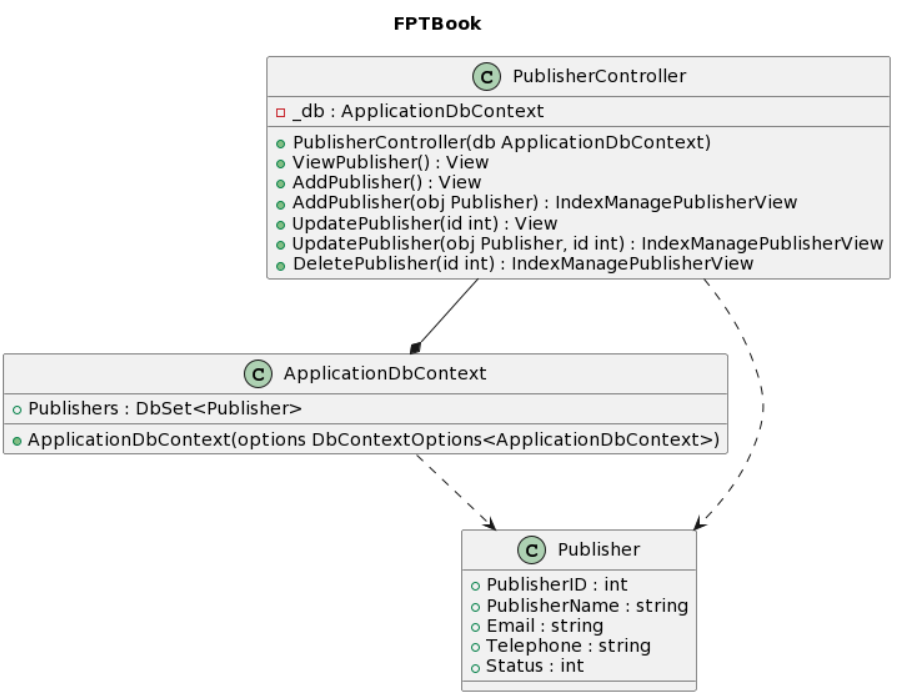


- CategoryController class: has a \_db property with data type ApplicationDbContext with private access method. The CategoryController constructor takes the ApplicationDbContext as the parameter. There are functions ViewCategory, AddCategory, AddCategory with parameters passed in Category, UpdateCategory with parameters passed in int, UpdateCategory with parameters passed in Category and int, DeleteCategory with parameters passed in are int respectively View, View, IndexManageCategoryView, View, IndexManageCategoryView, IndexManageCategoryView. They have a return type of public. This class has a dependency relationship with the Category class and has a relationship if the ApplicationDbContext class is lost then this class cannot be used.

- ApplicationDbContext class: has a Categories property with a data type of DbSet<Category> with an access method of public, has the constructor of the class. In addition, this class has a dependency relationship with Category class

- Category class: has properties CategoryID, CategoryName, Description, Status respectively with data types of int, string, string, int respectively with public access method

### e. Class diagram of functions of PublisherController class

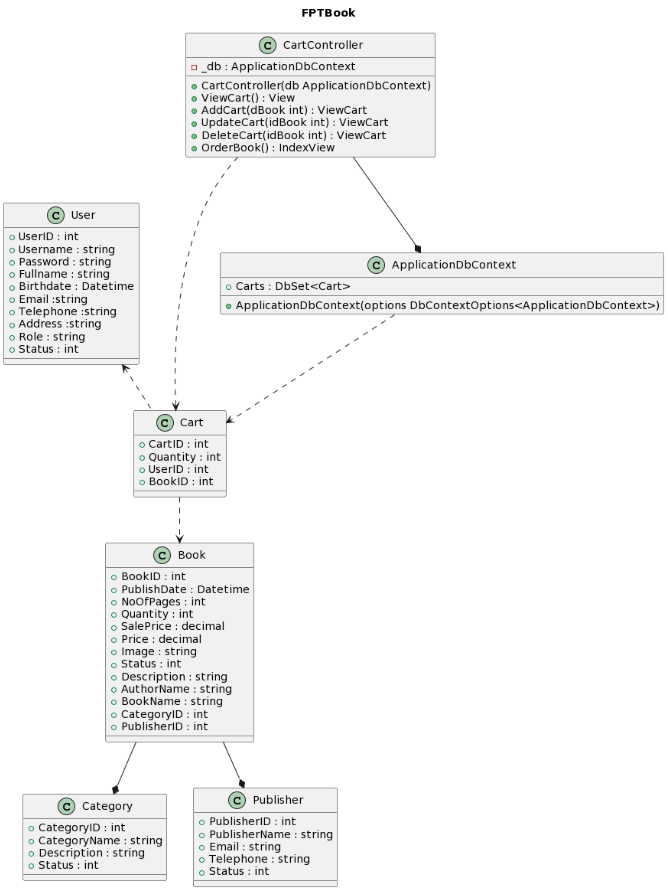


- PublisherController class: has a \_db property with data type ApplicationDbContext with private access method. The PublisherController constructor takes the ApplicationDbContext as the parameter. There are functions ViewPublisher AddPublisher, AddPublisher with parameters passed in Publisher, UpdatePublisher with parameters in int, UpdatePublisher with parameters passed in Publisher and int, DeletePublisher with parameters passed in are int respectively View, View, IndexManagePublisherView, View, IndexManagePublisherView , IndexManagePublisherView. They have a return type of public. This class has a dependency relationship with the Publisher class and if the ApplicationDbContext class is lost, this class cannot be used.

- ApplicationDbContext class: has a Publishers property with a data type of DbSet<Publisher> with an access method of public, has the constructor of the class. In addition, this class has a dependency relationship with the Publisher class

- Publisher class: has properties PublisherID, PublisherName, Email, Telephone, Status, respectively, with data types of int, string, string, string, int respectively with public access methods

### f. Class diagram of functions of CartController class



- CartController class: has a \_db property with data type ApplicationDbContext with private access method. The CartController constructor takes the ApplicationDbContext as the parameter. There are functions ViewCart, AddCart with parameters passed in int, UpdateCart with parameters passed in int, DeleteCart with parameters passed in int, OrderBook respectively View, ViewCart, ViewCart, ViewCart, IndexView. They have a return type of public. This class has a dependency relationship with the Cart class, and if the ApplicationDbContext class is lost, the class cannot be used.

- ApplicationDbContext class: has a Carts property with a data type of DbSet<Cart> with an access method of public, has the constructor of the class. In addition, this class has a dependency relationship with the Cart class

- Book class: has properties BookID, PublishDate, NoOfPages, Quantity, SalePrice, Price, Image, Status, Description, AuthorName, BookName, CategoryID, PublisherID, respectively, data types are int, Datetime, int, int , decimal, decimal, string , int, string, string, string have public access method. It has a relationship with Category class and Publisher class that if any class is lost, this class will not be usable

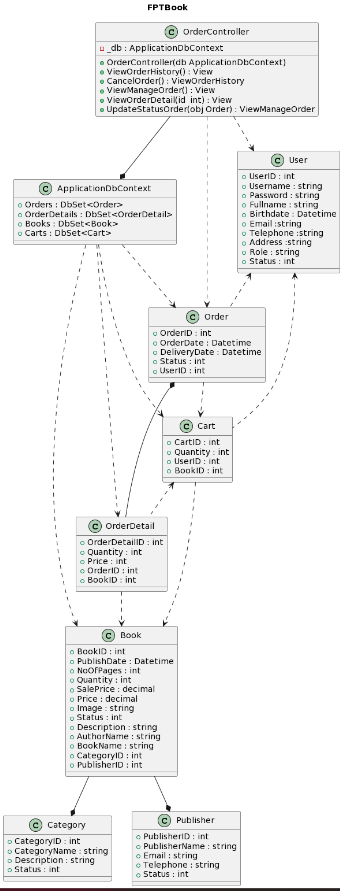
- Category class: has properties CategoryID, CategoryName, Description, Status respectively with data types of int, string, string, int respectively with public access method

- Publisher class: has properties PublisherID, PublisherName, Email, Telephone, Status, respectively, with data types of int, string, string, string, int respectively with public access method

- ApplicationUser class: has properties such as: UserID, Username, Password, Fullname, Birthdate, Email, Telephone, Address, Role, Status with data types respectively int, string, string, string, Datetime, string, string, string, string, and int.

- Cart class: has properties CartID, Quantity, UserID, BookID respectively with data types of int, int, int, int respectively with public access method. This class has a dependency relationship with the ApplicationUser class

### g. Class diagram of functions of OrderController class



- Book class: has properties BookID, PublishDate, NoOfPages, Quantity, SalePrice, Price, Image, Status, Description, AuthorName, BookName, CategoryID, PublisherID, respectively, data types are int, Datetime, int, int , decimal, decimal, string, int, string, string, string have public access method. It has a relationship with Category class and Publisher class that if any class is lost, this class will not be usable

- Category class: has properties CategoryID, CategoryName, Description, Status respectively with data types of int, string, string, int respectively with public access method

- Publisher class: has properties PublisherID, PublisherName, Email, Telephone, Status, respectively, with data types of int, string, string, string, int respectively with public access methods

- ApplicationDbContext class: has a Carts property with a data type of DbSet<Cart>, Orders property with a data type of DbSet<Order>, OrderDetails property with a data type of DbSet<OrderDetail>, Books property with a data type of DbSet<Book> with an access method of public, has the constructor of the class. This class has a dependency relationship with the Cart class, Order class, OrderDetail class, Book class

- ApplicationUser class: has properties such as: UserID, Username, Password, Fullname, Birthdate, Email, Telephone, Address, Role, Status with data types respectively int, string, string, string, Datetime, string, string , string, string, int

- Cart class: has properties such as: CartID, Quantity, UserID, BookID with data types respectively int, int, int, and int. This class has a dependency relationship with the Book class and ApplicationUser class

- Order class: has properties such as: OrderID, OrderDate, DeliveryDate, Status, UserID with data types respectively int, Datetime, Datetime, int, int. This class has a dependency relationship with the Cart class and ApplicationUser class

- OrderDetail class: has properties such as: OrderDetailID, Quantity, Price, OrderID, BookID with data types respectively int, int, int, int and int. This class has a dependency relationship with the Cart class and Book class. It has a relationship with Order class if Order class is lost, this class will not be usable

- OrderController class: has a \_db property with data type ApplicationDbContext with private access method. The OrderController constructor takes the ApplicationDbContext as the parameter. There are functions ViewOrderHistory, CancelOrder, ViewManageOrder, ViewOrderDetail with parameters passed in int, UpdateStatusOrder with parameters passed in Order respectively View, ViewOrderHistory, View, View, ViewManageOrder. They have a return type of public. This class has a dependency relationship with the ApplicationUser class and Order class, and if the ApplicationDbContext class is lost, the class cannot be used.

### h. Class diagram of functions of UserController class



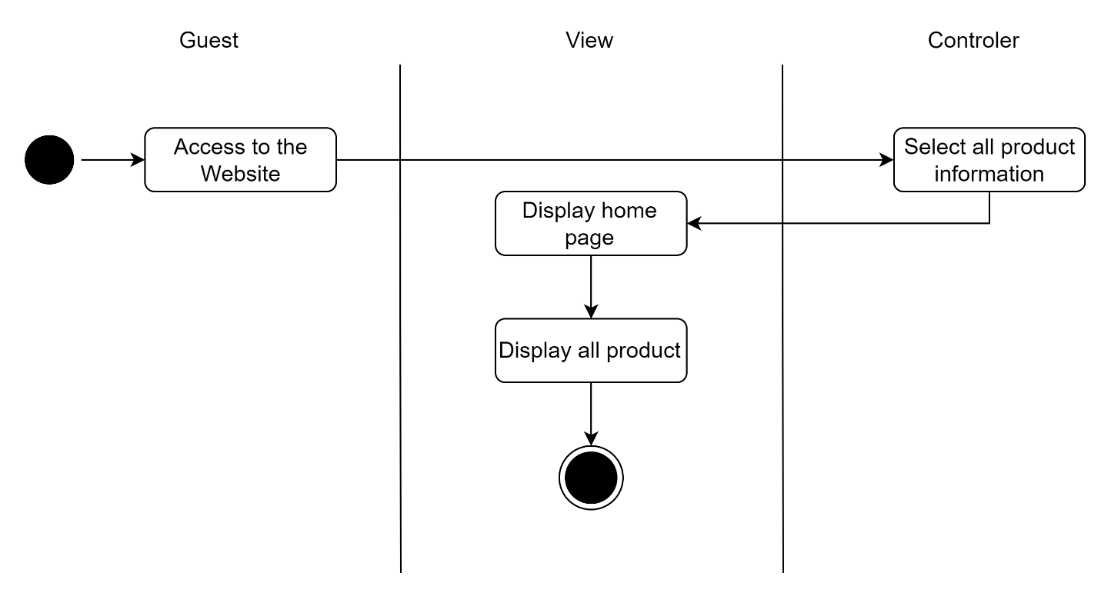
- ApplicationDbContext class: This class inherits the IdentityDbContext class, has the constructor of the class and depends on the ApplicationUser class.

- ApplicationUser class: has properties such as: UserID, Username, Password, Fullname, Birthdate, Email, Telephone, Address, Role, Status with data types respectively int, string, string, string, Datetime, string, string , string, string, int. Also, it inherits the IdentityUser class.

- UserController class: has a \_db property with data type ApplicationDbContext with private access method. The UserController constructor takes the ApplicationDbContext as the parameter. There are functions ViewAllProfile, ChangeInformation with parameters passed in User, ChangePassword with parameters passed in User, ViewAllAccount, UpdateStatusAccount with parameters passed in int, UpdateStatusAccount with parameters passed in User, CreateOwnerAccount, CreateOwnerAccount with parameters passed in User, SearchAccount with parameters passed in string respectively View, ViewAllProfile, ViewAllProfile, View, View, ViewAllAccount, View, ViewAllAccount, ViewAllAccount. They have a return type of public. This class has a dependency relationship with the ApplicationUser class, and if the ApplicationDbContext class is lost, the class cannot be used.

## 3. Activity Diagram

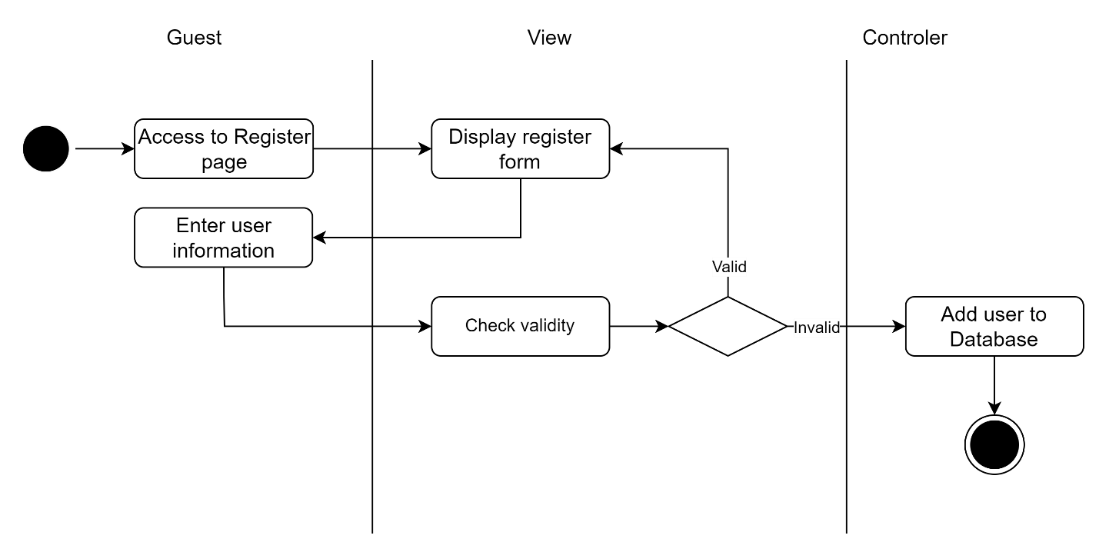
* Home page: Start when a visitor views the website, the Controller obtains all of the Product's data, which the View then shows before closing



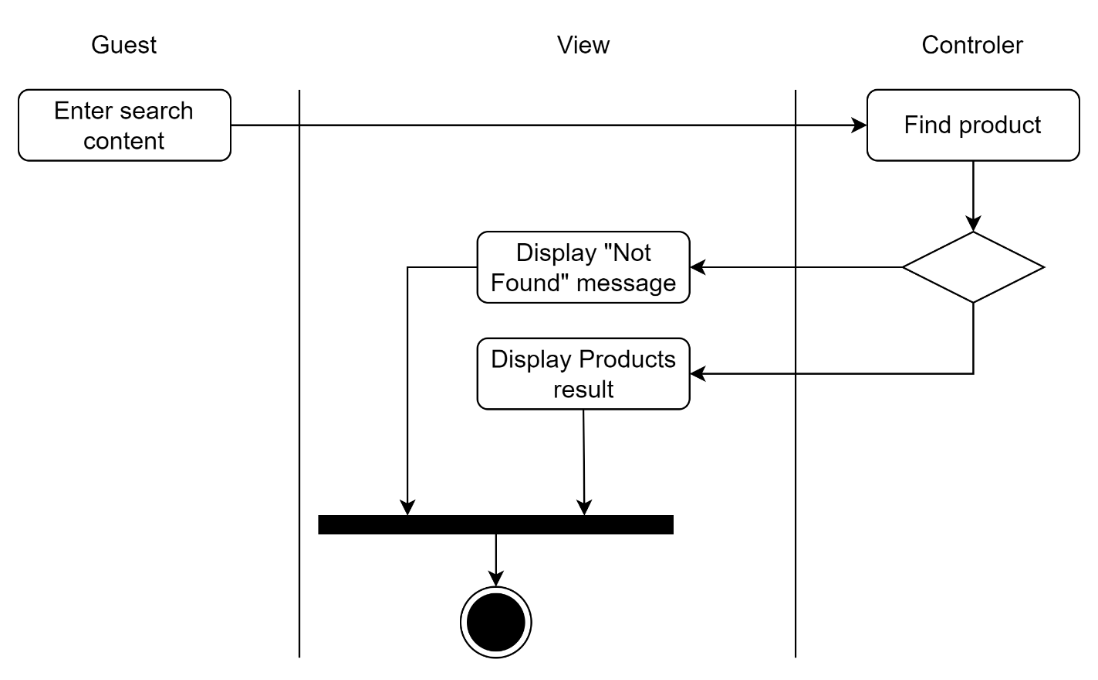
* Product detail: A visitor selects a product View then shows the information about that product after the Controller has obtained the data for it



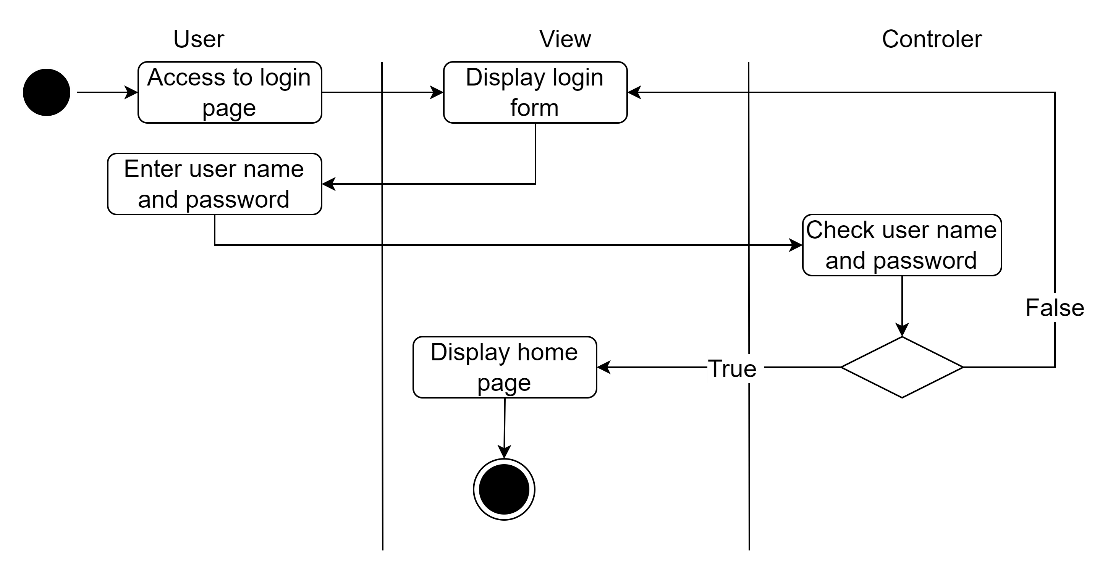
* Register: The Registrer page is accessed by Star the Guest, who then views the Registr page. The visitor submits all the information required to register, the view verifies its accuracy, and the controller adds the new user's information to the database; otherwise, the visitor is sent back to the registration page, where the registration process ends.



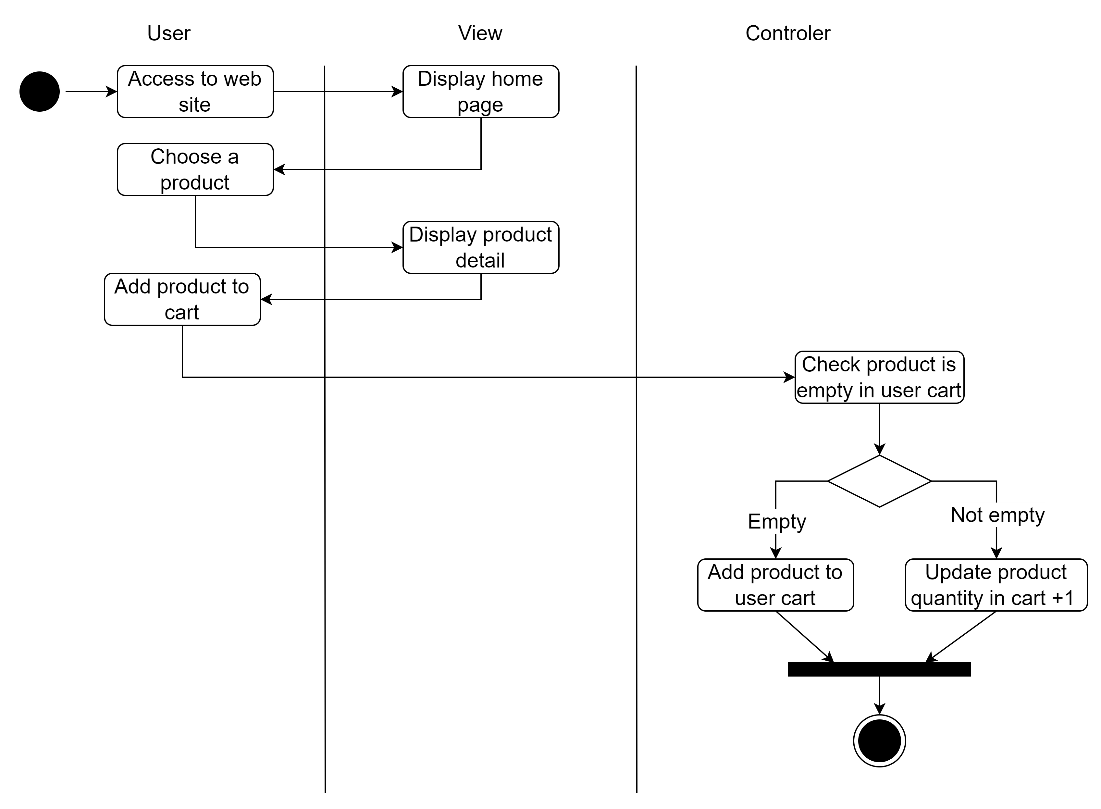
* Search: If there are no outcomes, the Controller first looks for a product with the same or a similar name as the search content entered by the guest. If there are no results, View will finish after displaying a "Not found" message.



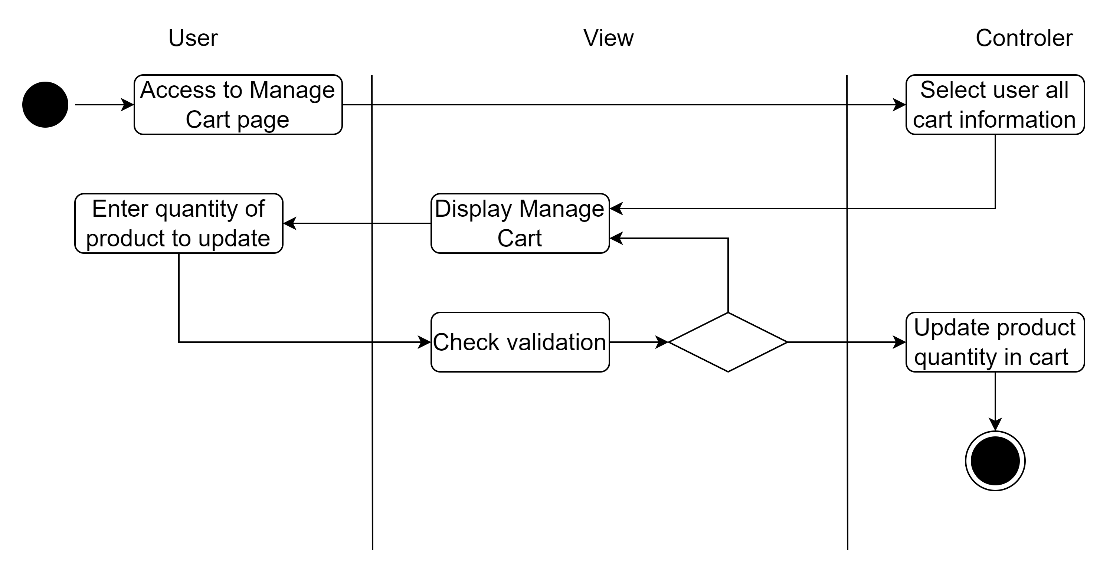
* Login: Start by granting the user access to the login page. The user should then view and fill out the login form. View displays the Home and end pages. The controller validates the user name and password and returns to the login form if either is incorrect.



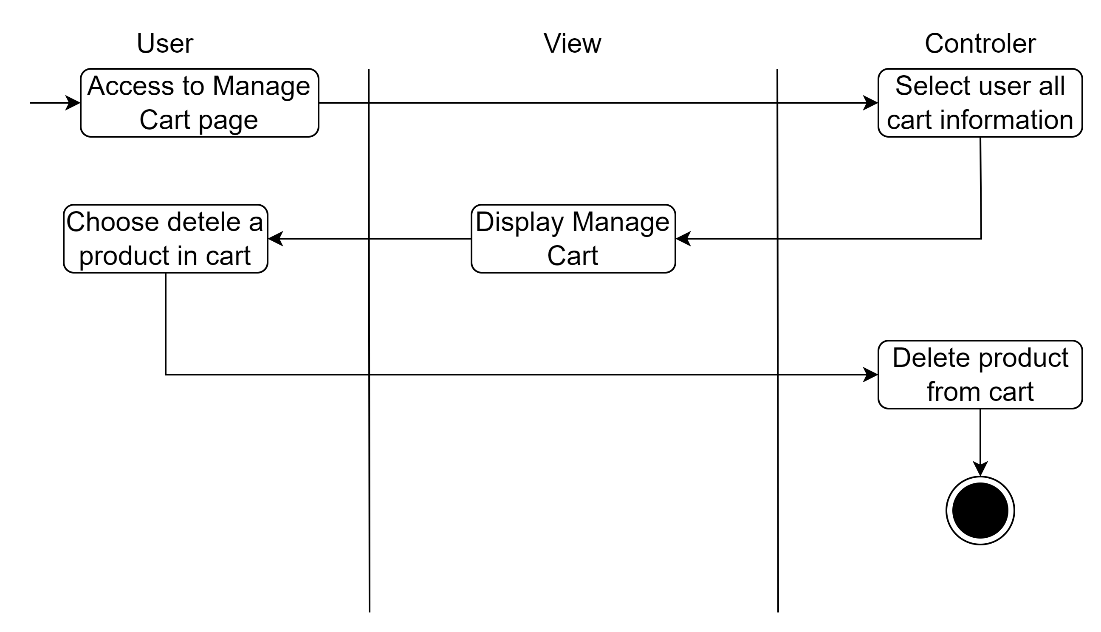
* Add to cart: Start access by users to the website, User selects a product to view product detail after the view displays the home page. The consumer will add the item to their shopping cart on the product detail page. The controller determines whether the product is already in the cart; if so, it updates the number of products in the cart by 1; otherwise, it adds the product to the cart and ends



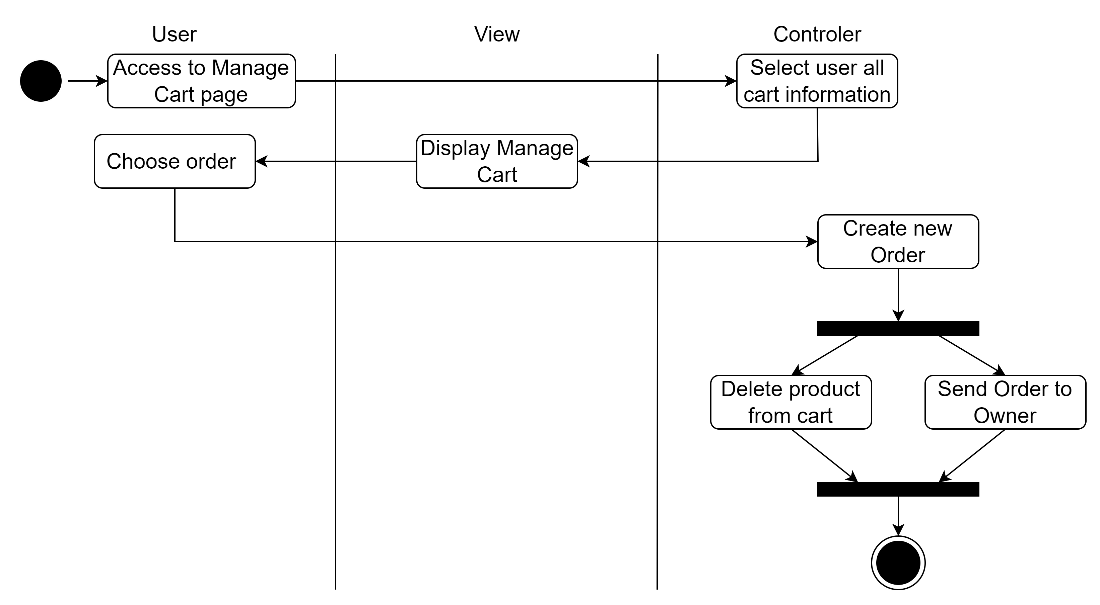
* Update product quantity: Start User accessing the Manage Cart page. After viewing the Manage Cart page, User enters the updated product quantity. If a product is legitimate, the controller updates the number of that product in the basket, and the view checks for invalidity before ending.



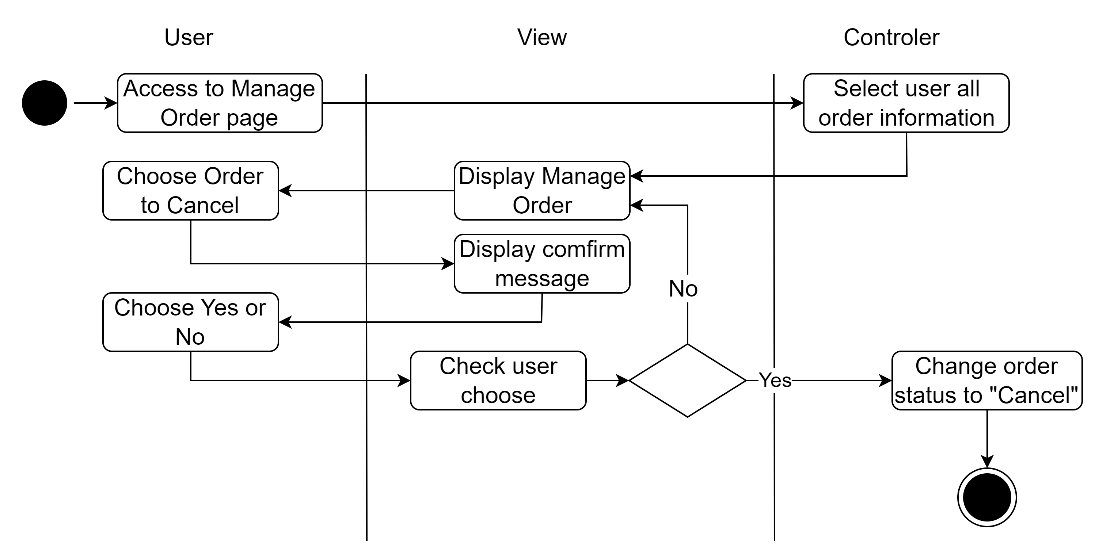
* Delete product from cart: User first accesses the Manage Cart page, which is displayed, after which User selects a product to delete. The controller ends by removing the item from the shopping cart.



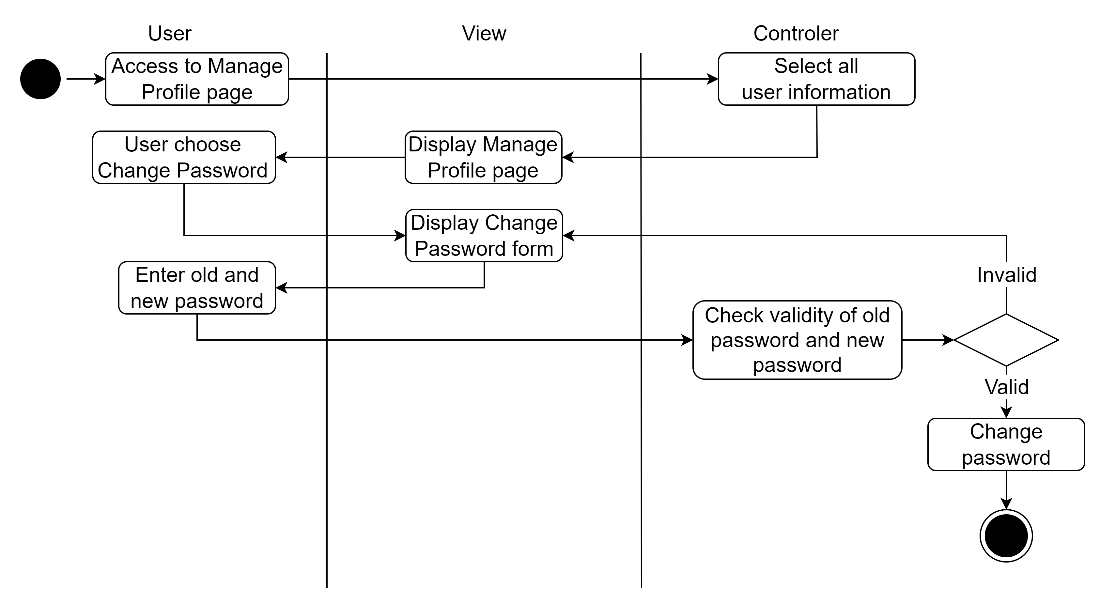
* Order: Start when a user accesses the Manage Cart page, the controller chooses all of the user cart data before the View shows the Manage Cart page. On the Manage cart page, the user selects Order. The Controller then generates a new Order and removes the ordered items from the cart, and it ends.



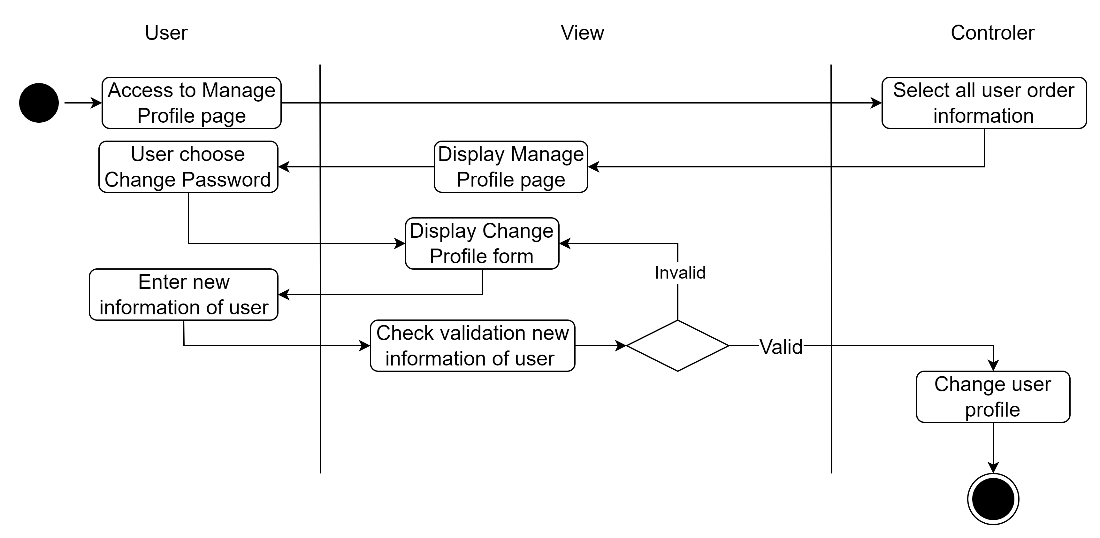
* Delete order: The Controller chooses all order data and presents it in the View when the user first accesses the Manage Order page. The View shows a confirm form after the User selects an order to cancel. The order status will change to Cancel if the user selects yes; otherwise, they will be taken back to the Manage Order page.



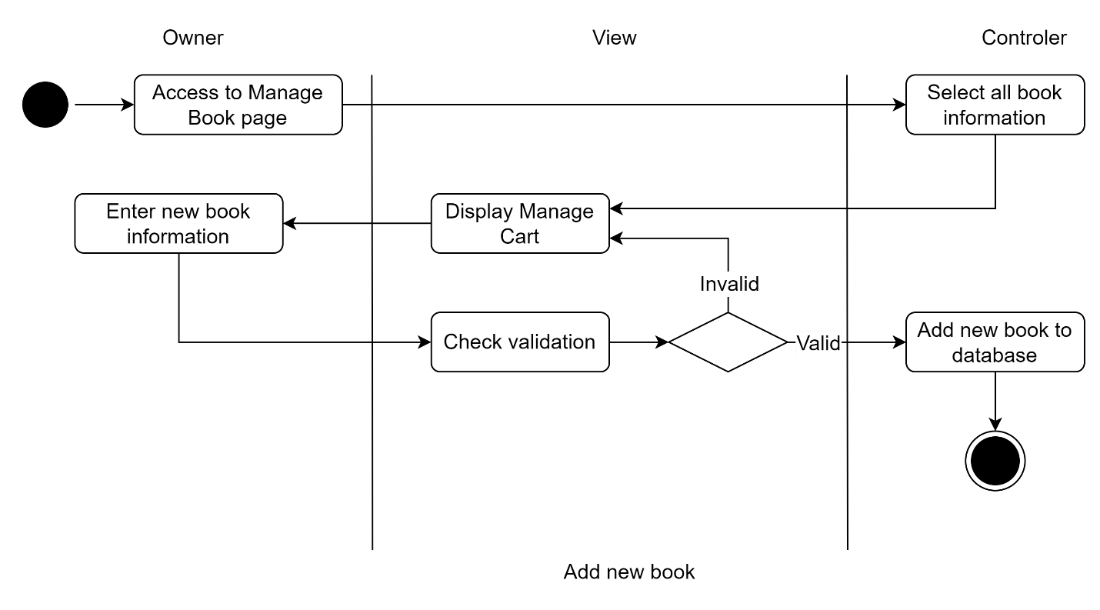
* Change password: Start when a user accesses the Manage Profile page, the Controller chooses all the user account data before displaying the Manage Profile page on the View. When the user selects "change password," the view shows the password-change form. The user enters an old password and a new one. The controller then checks the validity of the old and new passwords. If either is false, the controller returns the user to the password-change form.



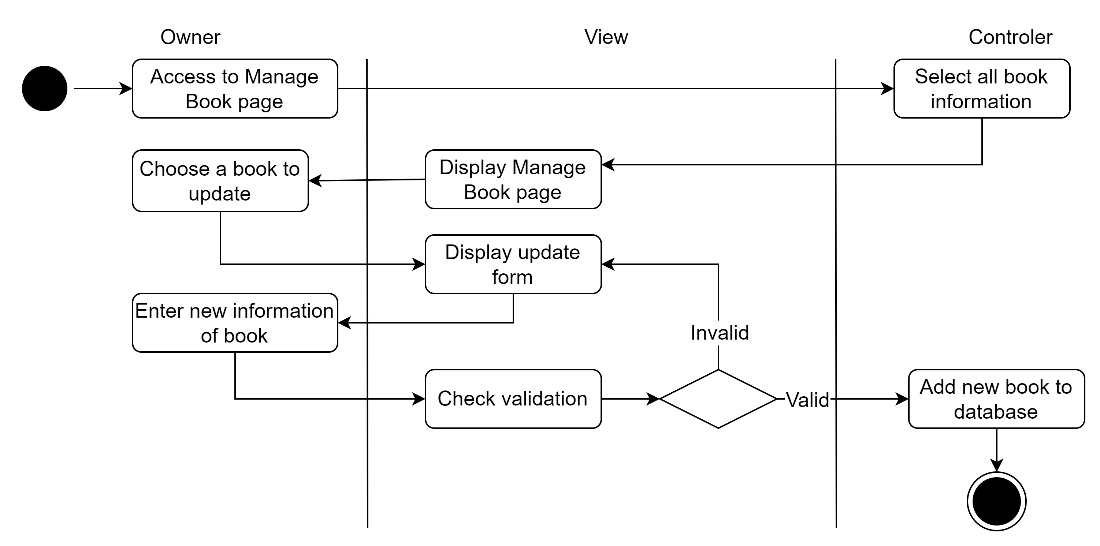
* Change profile: Beginning with the user's access to the Manage Profile page, the Controller chooses all user account data before the Manage Profile page is shown in the View. The View then shows the Change profile form after the User selects Change profile. The User enters new profile information, the View verifies its accuracy, if it is false, it returns to the Change profile form, and if true, the Controller changes the user's profile before coming to an end.



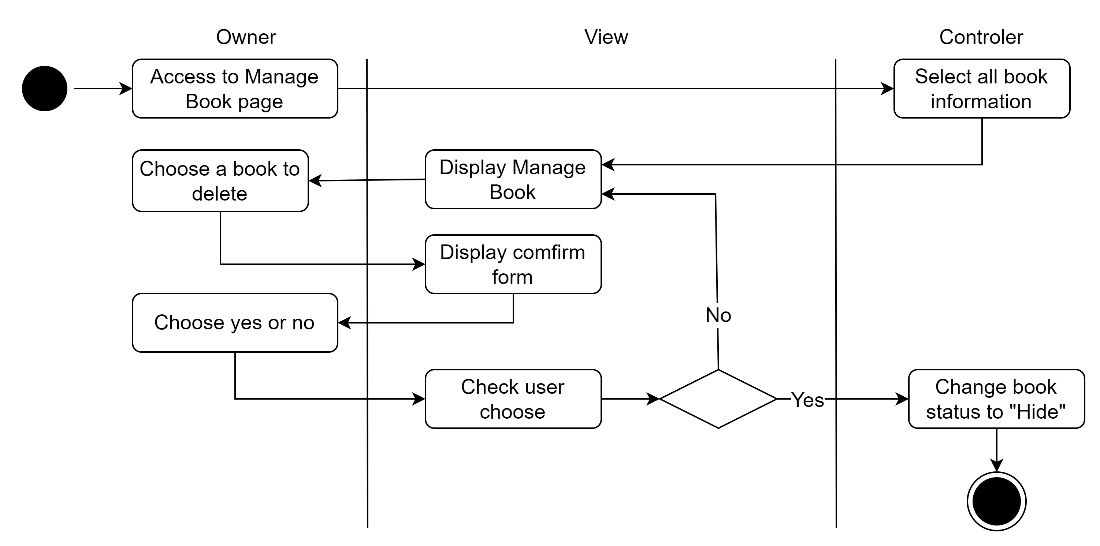
* Add book: Open the Manage book page with Owner access, then have the Controller choose all of the book's data and display it in View. After choosing Add new book, the owner's view will appear. new book form add. Owner completes the form with all the details for the new book, and the View verifies the accuracy of the information. If data is invalid return to Add new book form if data is valid Controller add new book to Database and terminate.



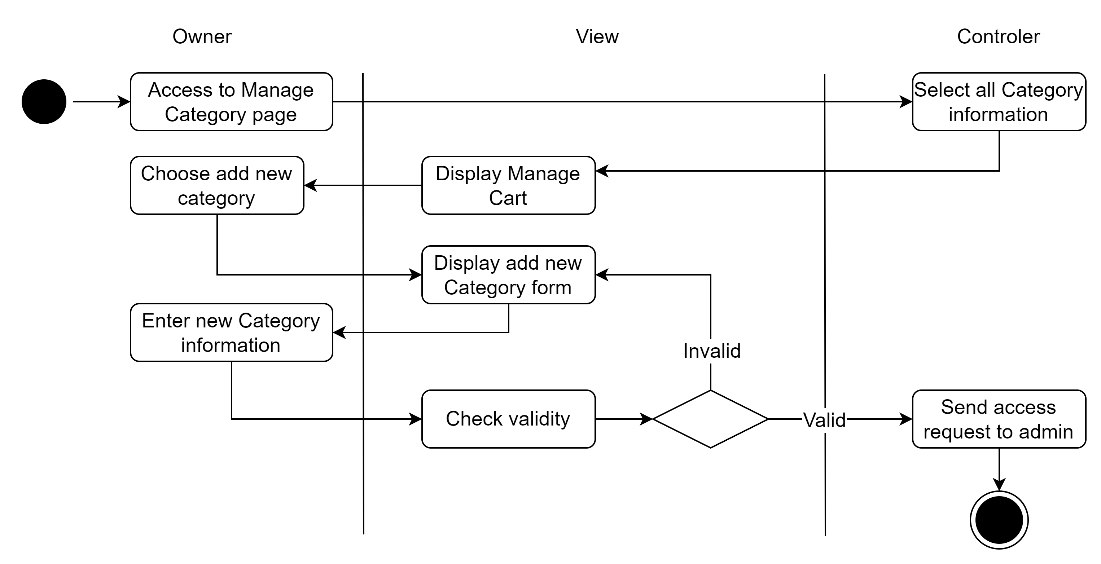
* Update book: Owner should first access the Manage Book page, after which the Controller should choose all of the book's data and display it in View. The following owner choose a book to update. The Update book form will be displayed in the view. Owner updates the book's information, and the View then verifies the accuracy of the information. If the data is correct, the controller will update the book's information in the database and the process will finish if the data is incorrect.



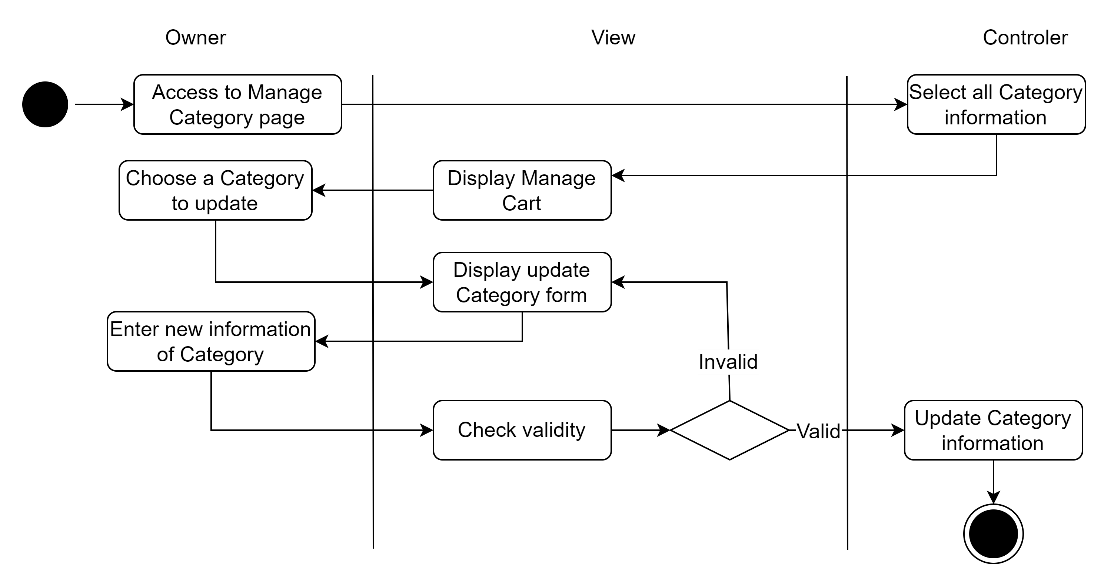
* Delete book: Owner should first access the Manage Book page, after which the Controller should choose all of the book's data and display it in View. The next owner choose a book to get rid of. The confirm form will appear in the view. The owner selects Yes or No; if No, the controller switches the book's status to Hide and ends; if Yes, the owner returns to the Manage Book page.



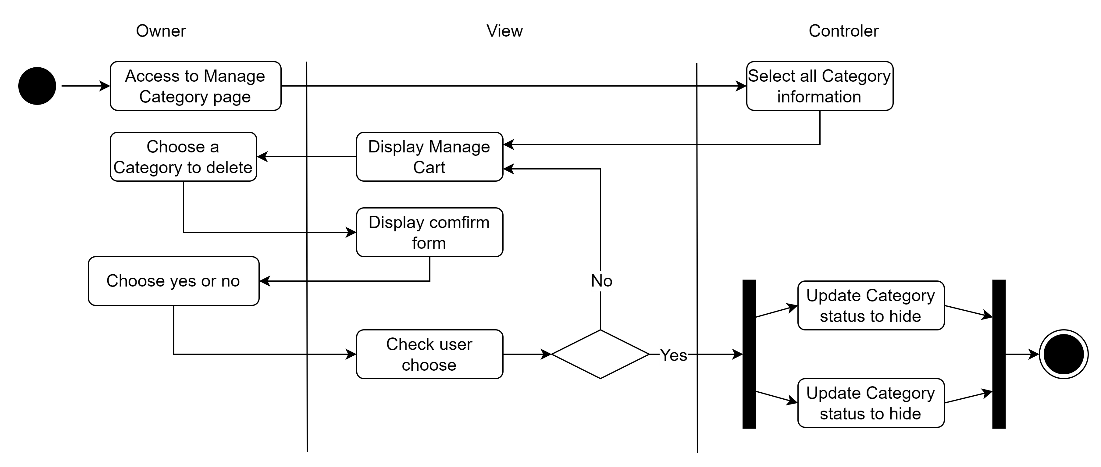
* Add new category: Owner should first access the Manage Category page, after which the Controller should choose all of the Category information and display it in View. The Add new category form will appear when the next Owner chooses to Add new category. Owner completes the form with all the data for the new category before the View verifies the accuracy of the information. If the data is correct, the controller should add the new category to the database, and the process should finish if the data is incorrect.



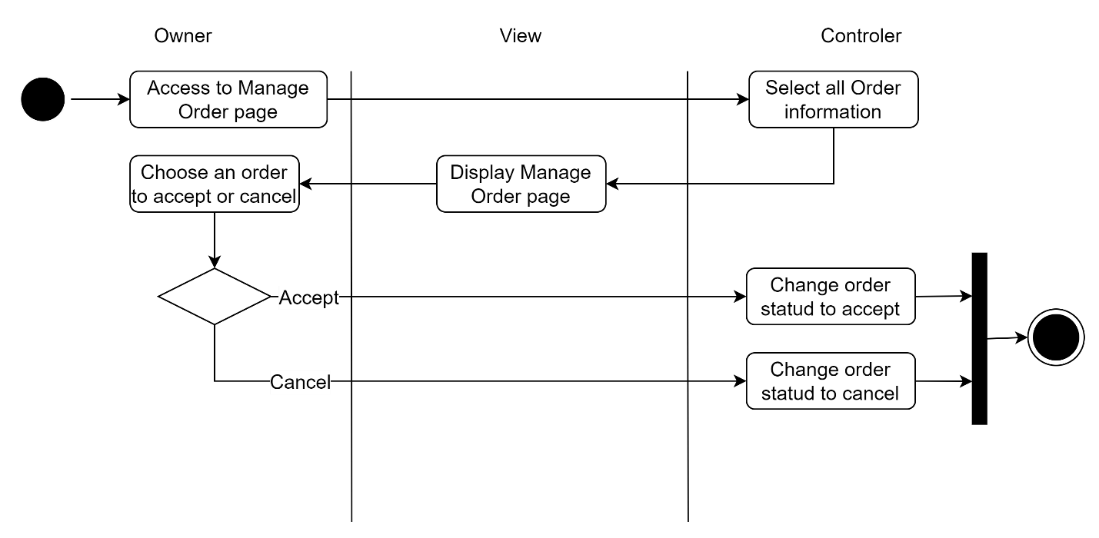
* Update category: Owner should first access the Manage Category page, after which the Controller should choose all of the category information and display it in View. Next Owner decides which category needs updating. The view will show the category form for updates. Owner updates the category's information, and the View then confirms the accuracy of the information. If the data is accurate, go back to the Update Category form; otherwise, the Controller will update the database with the category information and terminate.



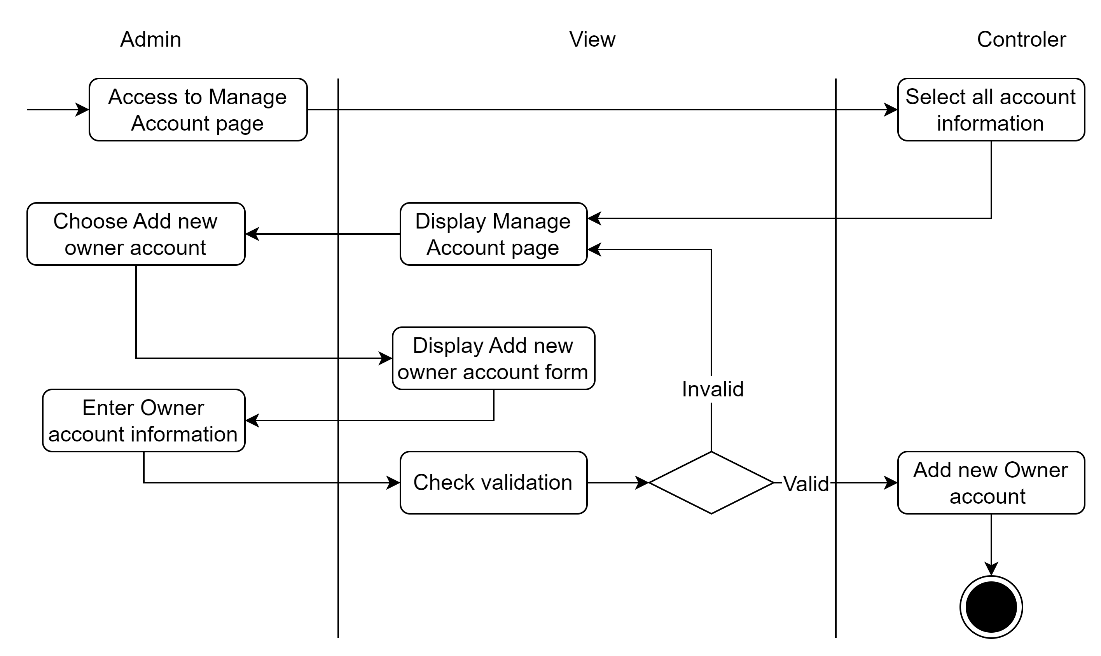
* Delete category: Owner should first access the Manage Category page, after which the Controller should choose all of the category information and display it in View. The next owner picks which category to eliminate. The Owner can select Yes or No after the view provides a confirmation. If the Owner chooses Yes, the Controller will change the category's status to Hide as well as the status of any books that belong to it.



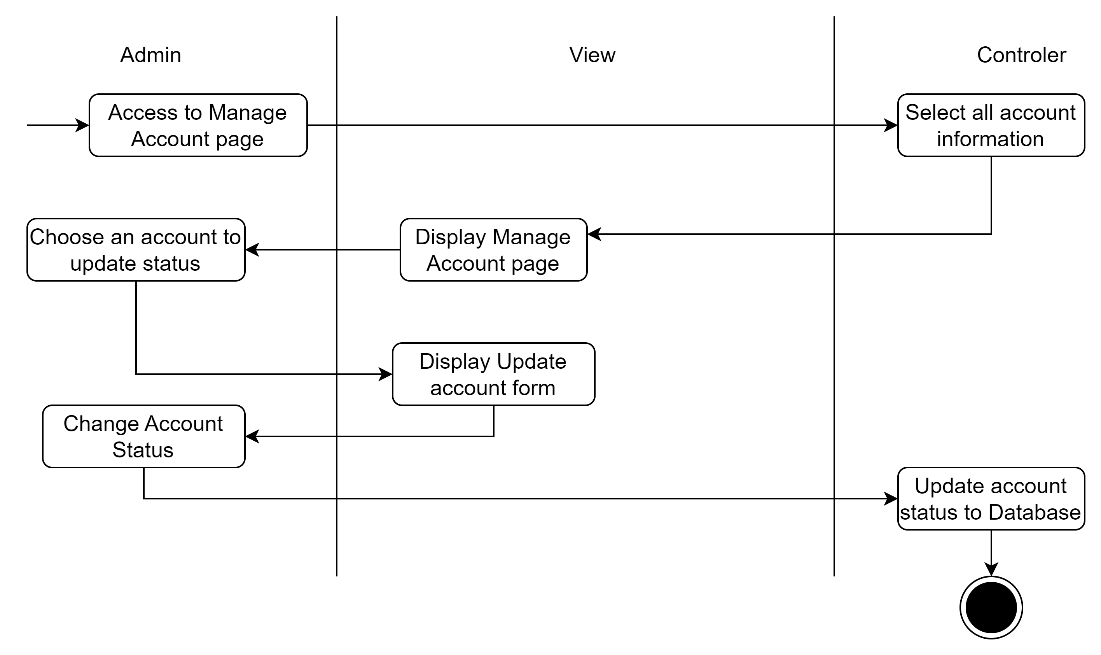
* Accetp order: Owner accesses the Order Management page, after which the Controller chooses every piece of Order data and shows it in View. The owner selects which orders to accept or reject. If the Owner decides to accept, the Controller updates the order status to Accept; if the Owner decides to cancel, the order status is updated to Cancel.



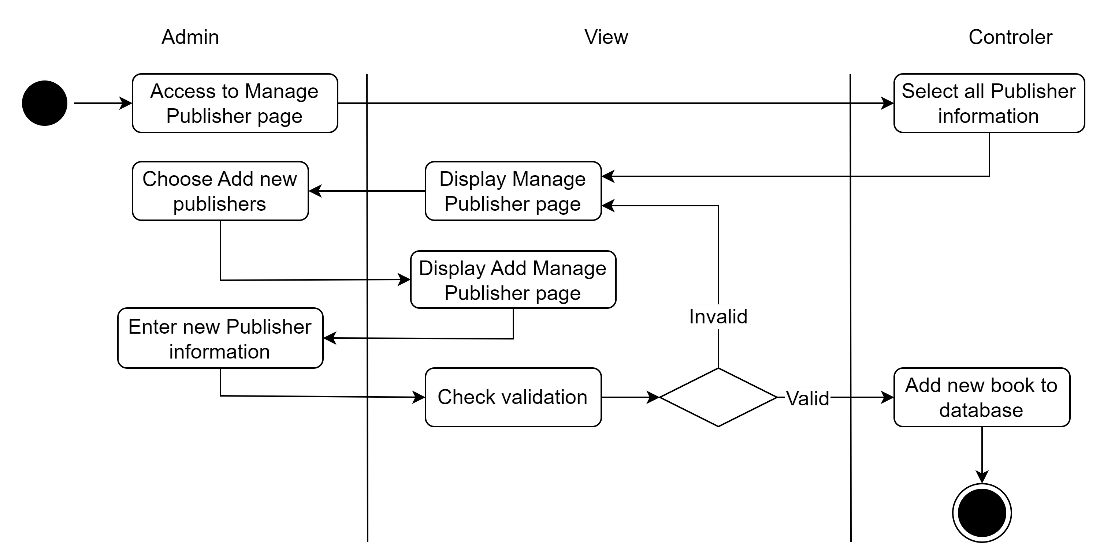
* Add new owner: When an administrator first gains access to the Manage Account page, the Controller chooses all of the Account data and displays it by View on the Manage Account page. When the admin decides to add a new owner account, the view will show the form for adding a new owner account. The new Owner account's complete information is entered by the administrator, and the data accuracy is then verified by the view. In the event that the data is accurate, the user should return to the Add New Owner Account form.



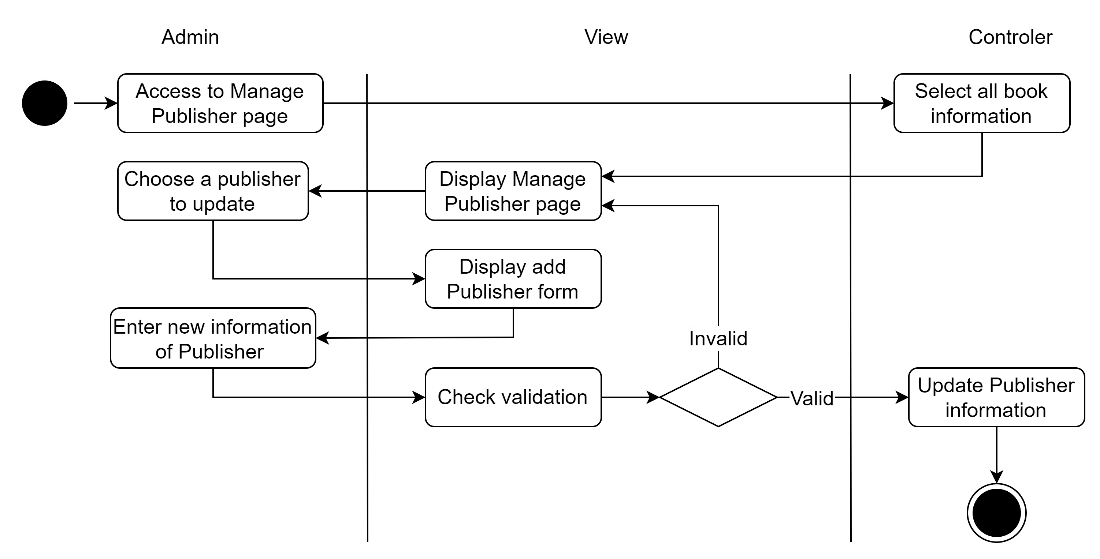
* Update account status: To access the Manage Account page, launch the Admin. The Controller then chooses all the Account information and shows it with View on the Manage Account page. Admin then decides which account will have its status updated. Admin can alter the status of an account from Available to Unavailable or vice versa. The controller stops after updating the database with the account status.



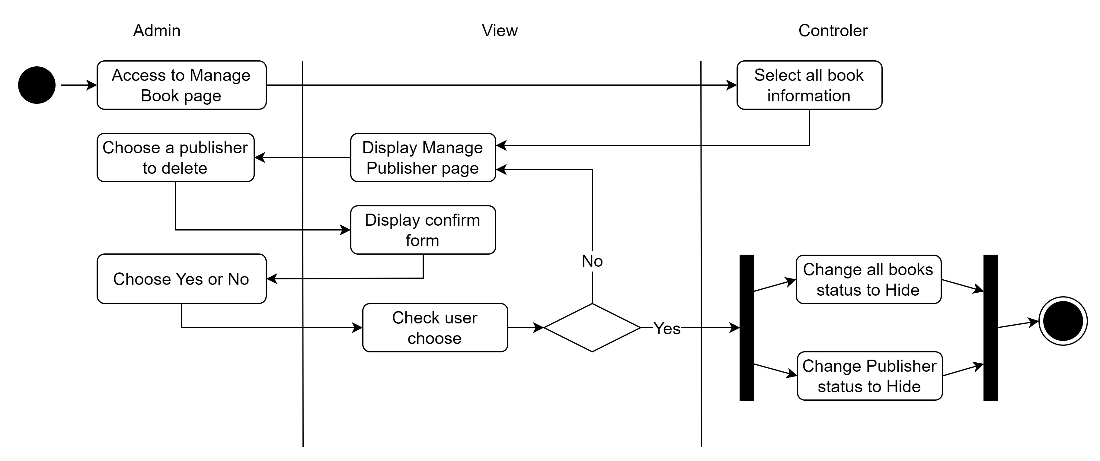
* Add new publisher: Launching the Manage Publisher page with Admin access causes the Controller to pick all publisher data and display the Manage Publisher page by View. The Add new publisher form will appear if the administrator selects Add new publisher. A new publisher's complete information is entered by the administrator into the form, and the data is then verified by the view. If the data is accurate, the controller should add the new publisher to the database and proceed; otherwise, return to the Add New Publisher form.



* Update publisher: Launching the Manage Publisher page with Admin permission causes the Controller to pick all publisher data and display it in View. The administrator then selects a publisher to update. The Update publisher form will be visible in the view. After the admin updates the publisher's information, the view verifies the accuracy of the information. If the data is correct, the controller will update the publisher's information in the database and then return to the Update Publisher form.

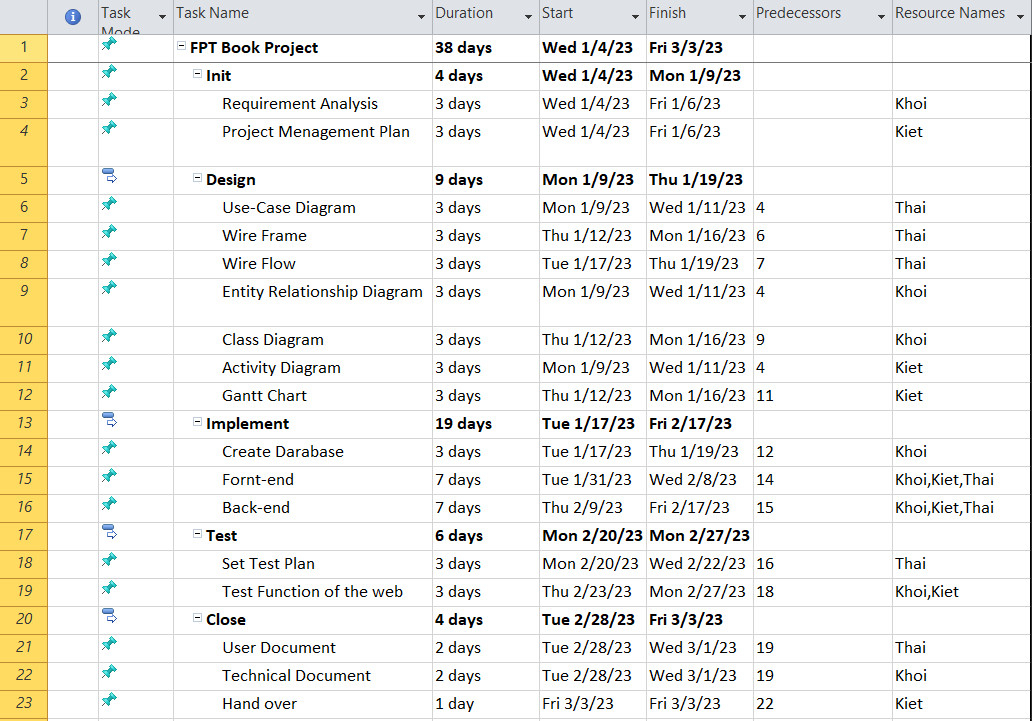


* Delete published: Start by granting admin access to the Manage Publisher page, after which the Controller picks out all publisher data and shows it in View. The publisher to delete is then chosen by the admin. The view shows a confirmation, and the administrator then chooses Yes or No. If the administrator chooses Yes, Controller will modify the publisher's status to Hide as well as the status of all books with this publisher.

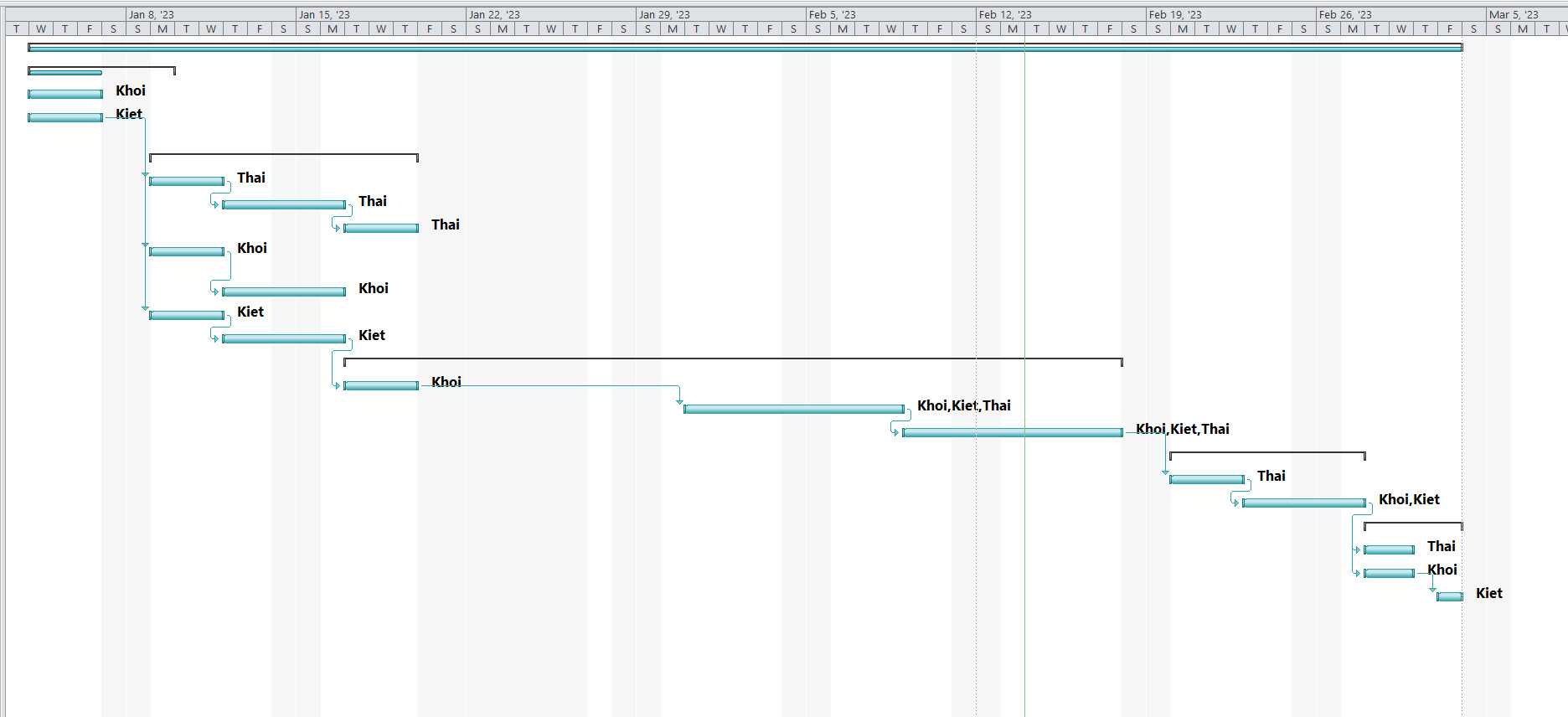


## 4. Gantt Chart

Work Breakdown Structure of Project:



With Work Breakdown Structure, gantt chart of project:



The FPT Book project will run from January 4 to March 3, 2023. The KTK developers group, which also includes Dang Khoi, Minh Thai, and Tuan Kiet, is in charge of carrying out the project. The project has five primary jobs: initiate, design, implement, test, and close, along with related supporting duties. From January 4 to January 6, 2023, Dang Khoi and Tuan Kiet worked concurrently on the primary task Init and its two subtasks, Requirement Analysis and Project Management Plan. The project is up next. Seven sub-tasks from the primary design project must be distributed equally among the team members for implementation. On January 9–11, 2023, Minh Thai will deploy Use–Case Diagrams, on January 12–16, 2023, Wire Frames, and on January 17–19, 2023, Wire Flows. From January 9–11, 2023, Dang Khoi will implement the Entity Relationship Diagram, and from January 12–16, 2023, it will implement the Class Diagram. Tuan Kiet Acting on the Gantt Chart from January 12 to 16, 2023, and the Activity Diagram from January 9 to 11, 2023. Main work Implement is followed by Fornt-End, which is completed by all 3 members from January 31 to February 8, 2023, Fornt-End, which is also completed by all 3 members from February 9 to 17, 2023. Create Database was completed by Dang Khoi from January 17 to January 19, 2023. After the primary work of Implement is the major job of Test, Dang Khoi and Tuan Kiet implemented the Test function of the website from February 23 to 27, 2023, and Minh Thai implemented the Set Test Plan from February 20 to 22, 2023. After two sub-tasks, User Document and Technical Document, were completed concurrently by Minh Thai and Dang Khoi from February 28 to March 1, 2023, the project finally progressed to Close, with the final sub-task, Hand over, being completed by Tuan Kiet on March 3, 2023.

# V. Risk Assessment

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NAME** | Dang Nguyen Dang Khoi | | | | **OBJECTIVE** | FPTBook web-based software system | | | | |
| **REF / ID** | **P R E - M I T I G A T I O N** | | | | **DEPARTMENT / LOCATION** | **MITIGATIONS / WARNINGS / REMEDIES** | **P O S T - M I T I G A T I O N** | | | |
| **RISK** | **RISK SEVERITY** | **RISK LIKELIHOOD** | **RISK LEVEL** | **RISK SEVERITY** | **RISK LIKELIHOOD** | **RISK LEVEL** | **ACCEPTABLE TO PROCEED?** |
| REF/01 | Project completion time exceeded deadline | UNDESIRABLE | POSSIBLE | **HIGH** | Leader | 1. Before starting the project, it is necessary to make a plan and specific time for each task for each person  2. During the implementation of the project, regularly monitor the progress of the members and recalculate the time if any changes occur. | UNDESIRABLE | POSSIBLE | **HIGH** | **YES** |
| REF/02 | The programmer can't work because of health reasons | UNDESIRABLE | POSSIBLE | **HIGH** | Programmers involved in project development | The team leader needs to talk to other team leaders to borrow members or quickly recruit members with the right qualifications for the position. | ACCEPTABLE | POSSIBLE | **MEDIUM** | **YES** |
| REF/03 | Expenditures during project implementation exceed the initial capital level | INTOLERABLE | POSSIBLE | **HIGH** | Leader | Cost worksheets, risks that may affect the cost of the project should be calculated before starting | INTOLERABLE | POSSIBLE | **HIGH** | **NO** |
| REF/04 | The interface design does not match the customer's requirements | UNDESIRABLE | POSSIBLE | **HIGH** | Customer | Customer care department needs to exchange and understand and absorb customers' opinions. After that, it is necessary to closely coordinate with the design department to have the same design as the customer's wishes | ACCEPTABLE | POSSIBLE | **MEDIUM** | **YES** |
| REF/05 | Hardware device damaged | UNDESIRABLE | POSSIBLE | **HIGH** |  | Maintenance personnel need to regularly check the equipment. In addition, to avoid affecting the work progress, it is necessary to have hardware devices ready to be replaced if any equipment is damaged. | ACCEPTABLE | POSSIBLE | **LOW** | **YES** |

# VI. Technologies Evaluation

## 1. Design Tools

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Draw.io** | **Figma** | **Why I choose?** |
| **Definition** | Online and desktop app called Draw.io has open-source code. It is software for flowcharts and diagrams that was created with the modern obligations and sensitivities of professionals in mind. The user-friendly interface of this tool, which enables users to put out their data in a more palatable form, can also leave a positive impression on users. This is due to the fact that its interface has features and settings that are simple to use and comprehend for people of all skill levels. We can also anticipate that this flowchart generator will be a flexible program. Draw.io is a free tool since it offers users both an online and offline option for using it, as well as a variety of templates and layouts for any art requirement they may desire (Morales, 2022). | An online tool for designing user interfaces and altering visuals is called Figma. It may be used for a variety of graphic design tasks, including wireframing websites, creating user interfaces for mobile applications, creating design prototypes, creating social media posts, and everything in between (theme-junkie, 2021). | I have chosen Draw.io to design UML and design User Interface for our team project because of the following reasons. First of all, Draw.io lets me drag and drop icons with ease. In addition, it supports a lot of built-in icons that I need without me having to install additional libraries. |
| **Characteristic** | Using Draw.io's online tool, you may create flowcharts and diagrams. You can drag and drop shapes onto the canvas while creating diagrams or maps, which will then be stored to the cloud. Additionally, you may share and export your diagrams as well as import and export data with this application (Morales, 2022). | Figma operates entirely within your browser. As a result, you can access your projects and begin designing from any computer or platform without having to install additional software or acquire additional licenses. The fact that Figma offers a large free plan that allows you to create and store three active projects at once is another reason why designers adore this tool. You have more than enough time to learn new things, try new things, and work on minor tasks (theme-junkie, 2021). |

## 2. Front End technology stack

* **Programming Language**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **JavaScript** | **TypeScript** | **Why I choose?** |
| **Definition** | A dynamic computer programming language is called JavaScript. Its implementations enable client-side script to interact with users and create dynamic pages, and it is most frequently used as a component of web pages. It is an object-oriented programming language that may be interpreted (tutorialspoint, 2023). | Strongly typed, object-oriented, and compiled describe TypeScript. Anders Hejlsberg, the man behind C# at Microsoft, created it. Both a language and a collection of tools are part of TypeScript. A typed superset of JavaScript that has been compiled to JavaScript is TypeScript. To put it another way, TypeScript is JavaScript plus a few extra features (tutorialspoint, 2023). |  |
| **Characteristic** | - Less server interaction: Before transmitting the page to the server, you can check the user input. Because of the reduced server traffic, your server will be under less stress  - Immediate feedback to the visitors: They can check to see whether they forgot to type something without having to wait for the website to reload  - Increased interactivity: You can design user interfaces that respond when a user moves their mouse cursor over them or presses a key on their keyboard  - Richer interfaces: For your website's visitors to experience a Rich Interface, you can use JavaScript to include things like drag-and-drop components and sliders  (tutorialspoint, 2023) | - Compilation: An interpreted language is JavaScript. As a result, it needs to be used to verify its validity. In case there is an error, that implies you write all the scripts merely to find no output. As a result, you must spend hours looking for faults in the code. The error-checking feature is offered by the TypeScript transpiler. If TypeScript detects any syntax mistakes, it will compile the code and produce compilation errors. This makes faults more obvious before the script is executed  - Strong Static Typing: There is no strong typing in JavaScript. Through the TLS, TypeScript offers an optional static typing and type inference system (TypeScript Language Service). The TLS may deduce a variable's type based on its value when a variable is defined without a type  - Supports type definitions: Type definitions for already-existing JavaScript libraries are supported by TypeScript. External JavaScript libraries are defined in TypeScript Definition files (with the.d.ts extension). Therefore, these libraries can be included in TypeScript code  - Supports Object Oriented Programming: Classes, interfaces, inheritance, and other Object Oriented Programming principles are supported by TypeScript  (tutorialspoint, 2023) |

* **CSS/SCSS**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SCSS** | **CSS** | **Why I choose?** |
| **Definition** | An abbreviation for Sassy Cascading Style Sheets is SCSS. In essence, it is an improved and evolved form of the CSS language. It was made by Natalie Weizenbaum, Chris Eppstein, and Hampton Catlin. It is frequently referred to as Sassy CSS because it has more advanced features. We must translate or pre-process this pre-processor language into the CSS language. The file extension for the SCSS language is .scss (byjus, 2023). | The term "CSS" stands for Cascading Style Sheet. We may develop and create many web pages using this kind of scripting language. A developer can style web pages with CSS to make them more presentable and appealing overall. In combination with HTML and JavaScript, CSS is the most widely used web technology. It has the .css extension (byjus, 2023). |  |
| **Characteristic** | - Since it is a preprocessor language, we must compile or interrupt it to create CSS  - SCSS uses far less lines of code than other languages  - The SCSS programming language encourages appropriately nested rules  - SCSS is a unique file format in an SASS software that requires Ruby programming  (byjus, 2023) | - CSS reduces time: CSS can be created once and reused in various HTML pages  - Simple Upkeep: Simply changing the style will cause all elements across all webpages to be updated when a global modification is made  - Search Engines: CSS is regarded as a clean coding style, therefore it won't be difficult for search engines to "understand" its material  - Superior styles to HTML: Compared to HTML attributes, CSS provides a far wider range of properties, allowing you to give your HTML page a lot nicer appearance  - Offline browsing: CSS uses an offline cache to enable local storage of web apps. We can visit offline websites using this  (byjus, 2023) |

* **JavaScript Library/Framework**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **jQuery** | **Angular** | **Why I choose?** |
| **Definition** | John Resig developed the quick and condensed jQuery JavaScript library in 2006 with the catchphrase "Write less, accomplish more." For quick web development, jQuery makes navigating HTML documents, event handling, animation, and Ajax interactions simpler (tutorialspoint, 2023). | Angular is a JavaScript framework that is open-source and developed in TypeScript. It is maintained by Google, and its main use is to create single-page applications. Angular is a framework with undeniable benefits that also gives programmers a uniform framework to work with. It gives users the ability to make huge apps in a manageable way (Deshpande, 2023). |  |
| **Characteristic** | - By utilizing the cross-browser open source Sizzle selection engine, jQuery made it simple to choose DOM elements, negotiate with them, and modify their content  - Without having to overcrowd the HTML code with event handlers, jQuery provides a simple approach to capture a range of occurrences, such as a user clicking a link  - Using AJAX technology, jQuery greatly aids in the development of a responsive and feature-rich website  - You can use the many built-in animation effects that jQuery has to offer on your webpages.  - The jQuery library is quite small, about 19KB in size (Minified and gzipped)  - jQuery is cross-browser compatible and functions well on IE 6.0+, FF 2.0+, Safari 3.0+, Chrome, and Opera 9.0+  (tutorialspoint, 2023) | - Angular gives users the ability to create custom components that can combine rendering logic and functionality into reusable parts. It works well with web components as well  - Using Angular, users can easily transfer data from JavaScript code to the screen and respond to user events without manually writing any code  - With Angular, users may create modular services and inject them as needed. This enhances the services' capacity to be tested and reused  - Tests are first-class tools, and Angular was created with testability in mind from the very beginning. You will be able to test each component of your application, which is strongly advised  - Angular is a complete framework that offers ready-to-use options for server connectivity, application routing, and more  (Deshpande, 2023) |

* **CSS Framework**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Bootstrap** | **Foundation** | **Why I choose?** |
| **Definition** | The mobile-first front-end framework Bootstrap makes building websites more efficient and quick. Javascript, HTML, and CSS are all used (tutorialspoint, 2023). | One of the top front-end frameworks for creating stunning, responsive websites is Foundation. It supports all platforms and offers JavaScript, CSS, and HTML plugins (tutorialspoint, 2023). |  |
| **Characteristic** | - Mobile first styles are incorporated into the Bootstrap 3 framework throughout the entire library rather than in separate files  - It is compatible with all widely used browsers  - Anyone can start using Bootstrap with just a working grasp of HTML and CSS. The official Bootstrap website also provides excellent info  - The responsive CSS in Bootstrap adapts to Desktops, Tablets, and Mobiles  (tutorialspoint, 2023) | - It offers quicker development thanks to the Sass compiler, which operates considerably more quickly than the default compiler  - Pricing tables, switches, joyride, range sliders, lightboxes, and many more features are added to your website  For quicker coding and control, it includes development packages like Grunt and Libsass.  - Foundation for sites gives you access to HTML, CSS, and JS so you can create websites rapidly  - You may get responsive HTML emails from the Email Framework that function on any device  - You can create completely responsive web apps using Foundation for Apps  (tutorialspoint, 2023) |

* **HTML/ Razor**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **HTML** | **Razor** | **Why I choose?** |
| **Definition** | HyperText Markup Language, or HTML. With the aid of a markup language, it is used to design web pages. The acronym HTML stands for Hypertext Markup Language. The link between web pages is defined by hypertext. The text document inside the tag that specifies the structure of web pages is defined using a markup language. This language is used to annotate (add notes to) material so that a computer can comprehend it and modify the content as necessary. Most markup languages, like HTML, can be read by people. The language employs tags to specify what text processing is required (geeksforgeeks, 2022). | With the help of the markup language Razor, you may incorporate Visual Basic and C# server-based code into web pages. While a web page is being written to the browser, server-based programming has the ability to generate dynamic web content instantly. Prior to returning the page to the browser after a web page request, the server runs any server-based code present on the page. The programs may carry out complicated operations, such accessing databases, by operating on the server. For building web applications, Razor is based on ASP.NET. It is simpler to use and to learn, yet it has the same capability as conventional ASP.NET markup (w3schools, 2023). |  |
| **Characteristic** | - It is simple to use and simple to learn  - It is independent of platforms  - A web page can include images, videos, and audio  - The text may be supplemented using hypertext  - Markup language is what it is  (geeksforgeeks, 2022) | All levels of developers, from entry-level to enterprise, can use Razor Pages. Its page-centric development style provides web developers with expertise with other page-centric frameworks, such as PHP, Classic ASP, Java Server Pages, ASP.NET Web Pages, and ASP.NET Web Forms, with a sense of familiarity. It also incorporates all of the sophisticated features of ASP.NET Core (including dependency injection), making it reasonably simple for beginners to learn and equally ideal for large, scalable, team-based projects (learnrazorpages, 2023). |

## 3. Back End technology stack

* **Programming Language**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **C#** | **PHP** | **Why I choose?** |
| **Definition** | Modern, object-oriented, general-purpose programming language C# is pronounced "C sharp." The European Computer Manufacturers Association (ECMA) and the International Standards Organization both gave their approval for its development by Microsoft under the direction of Anders Hejlsberg and his team inside the.Net program (ISO). Version 7.2 of C#, one of the languages for Common Language Infrastructure, is currently available. For users who are familiar with C, C++, or Java, C# is simple since it shares many syntactical similarities with Java (geeksforgeeks, 2019). | A server-side programming language called PHP is used to create either static or dynamic webpages or online apps. PHP, which formerly stood for Personal Home Pages, is now known as Hypertext Pre-processor. Only servers with PHP installed can interpret PHP scripts. Only a web browser is needed on the client machines to view the PHP scripts. A PHP file has the ".php" suffix and contains PHP tags (Jackson, 2023). |  |
| **Characteristic** | - Because C# is a high-level language, it is more similar to other well-known programming languages like C, C++, and Java and is therefore simple for everyone to learn  - Both desktop and web apps are frequently created in C#. It is one of the most widely used languages on business desktops. C# is the language of choice for anyone looking to develop Microsoft applications  - The more people involved, the better, as new software and tools will be created to improve the community. Due to the sizeable C# community, advancements are made to keep the language alive and prevent its extinction  - In game development, C# is widely utilized and will remain dominant. Because C# interfaces with Microsoft, it has a big target market. C# is a well-known language for creating video games because of its features like Automatic Garbage Collection, interfaces, object-oriented programming  (geeksforgeeks, 2019). | - PHP is open source and cost nothing  - Less difficult to learn than other languages like JSP and ASP, etc.  - Significant local document  - In contrast to other languages like ASP that require IIS, the majority of web hosting servers offer PHP by default. Because of this, PHP is a sensible decision  - PHP is regularly updated to reflect the newest developments in technology  - Another advantage of PHP is that it is a server-side scripting language, which means that client computers that request resources from the server do not require PHP to be installed; instead, just a web browser will do  - Although PHP has built-in support for collaborating with MySQL, this does not preclude the use of PHP with other database management systems. PHP is still compatible with  (Jackson, 2023) |

* **Operating System**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Linux** | **Windows** | **Why I choose?** |
| **Definition** | An operating system is Linux. In truth, Linux is the operating system that runs one of the world's most widely used platforms, Android. A desktop or laptop's operating system is a piece of software that controls all of the hardware resources connected to it. Simply explained, the operating system controls how your software and hardware communicate with one another. The operating system (OS) is necessary for the software to run (linux, 2023). | Microsoft created the operating system called Windows. You are only able to utilize a computer thanks to the operating system. Windows is the most widely used operating system in the world since it comes preloaded on the majority of new personal computers (PCs) (gcfglobal, 2023). |  |
| **Characteristic** | Open source software includes Linux. The source code for Linux is open source and free for anybody to examine, modify, and—for people with the necessary skills—contribute to. Additionally, Linux is unique in that there are numerous distributions of the operating system, each of which offers a variety of software options even if the fundamental components of Linux are typically universal. This demonstrates how flexible Linux is since it allows for the replacement of more than just software, such as word processors and web browsers. Linux users can also select various user-interface elements and essential components, such as which system shows graphics (opensource, 2023). | - Ease of use: Users accustomed to earlier versions of Windows will likely find the more recent ones to be simple to use as well. This can be attributed to a variety of factors, including the virtually universally standardized appearance and feel of Windows programs and the way the file system has been shown ever since MS-DOS (disk A:, disk C:, etc.). One of the main reasons Windows users are frequently hesitant to switch operating systems is due to this  - Available software: For Windows, there is a sizable assortment of software. Microsoft's dominance of the global market for PC operating systems and office applications is both a result of and the cause of this. The likelihood is that if a program exists and would serve your company needs, there will be a Windows version of it someplace  - Backwards compatibility: The chances are good (though not guaranteed) that those programs will also function with a newer version of Windows if you're currently using an older version of Windows and need something more modern but don't want to lose the use of some older programs that are only available for Windows and are essential to your business needs  - Support for new hardware:  - Plug & Play: In terms of Plug & Play compatibility for PC hardware, Windows still has an advantage over the competition as an operating system for the typical home user. When installing the proper drivers, Windows will typically do an excellent job of identifying new hardware. While Plug & Play functionality is available in other operating systems as well, it is generally less robust and necessitates more user involvement  (rjsystems, 2023) |

* **Web Server and Hosting**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **IIS** | **Apache** | **Why I choose?** |
| **Definition** | A single web platform called IIS Web Server unifies IIS, FTP services, PHP, ASP.NET, and Windows Communication Foundation (WCF). Because it automatically isolates applications, comes with a pre-configured sandbox, and has a smaller server footprint, you can use it to host your websites and services with the highest level of security. To speed up your website, it also incorporates dynamic caching and improved compression. Developers can add unique modules to the modular platform to increase its capabilities (Morris, 2022). | Approximately 40% of websites worldwide utilize the free and open-source Apache web server software. The name of the server is Apache HTTP Server. The Apache Software Foundation is in charge of its creation and upkeep. Apache enables website owners to serve material across the internet. It is referred to as a "web server" for this reason (javatpoint, 2021). |  |
| **Characteristic** | - Because it is closely connected with the Windows Operating System, IIS differs from other web servers (OS). This means that all of Windows' security features, including Active Directory and Group Policy, are accessible to IIS  - IIS also comes with a number of administrative tools for setting up and maintaining websites. These include the Graphical User Interface (GUI) for configuring IIS settings, Internet Services Manager (ISM)  - IIS 8 also comes with IIS Manager, an improved version of ISM that makes it easier to manage your websites and server settings. The command-line utility AppCdmd.exe and PowerShell cmdlets are also present on the web server  - IIS's ability to host ASP.NET apps is among its most important benefits. A Microsoft technology called ASP.NET allows programmers to construct dynamic web applications. Only IIS is capable of hosting ASP.NET apps without the need for extra software  (Morris, 2022) | - Reliable software that is stable  - Open-source and free, even for commercial use  - Regular security updates and patches  - Simple to configure and good for beginners  - Flexible thanks to its module-based design.  - Works with WordPress sites right out of the box  - Multiplatform (implements on Windows servers and Unix  - A sizable community and easily accessible support in the event of any problems  (javatpoint, 2021) |

* **Database**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MySQL** | **SQL Server** | **Why I choose?** |
| **Definition** | Based on Structured Query Language, MySQL is an open source relational database management system (RDBMS) sponsored by Oracle (SQL). Almost every platform, including Linux, UNIX, and Windows, supports MySQL. Although it has a wide range of uses, MySQL is most frequently related to websites and online publications (Moore, 2018). | One of the most popular relational database management systems on the market, Microsoft SQL Server supports a variety of software programs for business intelligence and analysis in corporate settings. It is based on the Transact-SQL language, integrates a number of standard language programming extensions, and offers both on-premises and cloud usage options for its application (PÉREZ, 2021). |  |
| **Characteristic** | - Data can be saved and accessed using MySQL across a variety of storage engines, including InnoDB, CSV, and NDB. For improved performance and durability, MySQL is also capable of duplicating data and splitting tables. Users of MySQL don't need to learn any new SQL commands in order to access their data  - Developed in C and C++, MySQL runs on more than 20 different operating systems, including Mac, Windows, Linux, and Unix  - MySQL provides a host-based verification feature that is enabled by an access privilege and encrypted password scheme. On any platform, MySQL clients can connect to MySQL Server via a variety of protocols, including TCP/IP sockets. Several client and utility programs, command-line programs, and administration tools like MySQL Workbench are also supported by MySQL  (Moore, 2018) | - Intelligence on all your data with Big Data clusters: The ability to perform replication-free queries across your complete data estate, from SQL Serve to Oracle  - Choice of Language and Platform: From Linux or Windows to deployments using Kubernetes  - Intelligent database capabilities: Support for persistent memory in-memory and an efficient tempbd  - Data encryption and compliance: For nine years, the National Institute of Standards and Technology database ranked it as one of the top most secure platforms due to its data protection, monitoring, and classification system  - Mobile BI and scalability: Making it possible for you to quickly integrate your database management systems with any device and Azure services for improved performance and data analytics |

* **Frameworks**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Laravel** | **.NET** | **Why I choose?** |
| **Definition** | Laravel is a reliable and simple to use open-source PHP framework. It adheres to the model-view-controller pattern of design. Laravel makes use of pre-existing parts from other frameworks to build online applications. The resulting web application is more organized and practical. Incorporating the fundamental components of PHP frameworks like CodeIgniter and Yii as well as other programming languages like Ruby on Rails, Laravel provides a wide range of functionalities. The extensive feature set of Laravel will accelerate web development (tutorialspoint, 2023). | Many web developers use the open-source ASP.NET framework to create cutting-edge web apps, services, and hubs that let connected clients access fresh material in real time. The Hypertext Transfer Protocol (HTTP), the common protocol used by all web applications, is compatible with this framework. The server-side framework can be used by web developers to install scripts on the server and provide a unique user interface (indeed, 2022). |  |
| **Characteristic** | - The Laravel framework makes the web application more scalable  - Designing the web application takes much less time because Laravel reuses components from other framework  - It has namespaces and interfaces, which aids in resource management and organization  (tutorialspoint, 2023) | Web developers benefit from ASP.NET's quick performance, affordable price, and extensive language support. The platform also requires very little setup and configuration; developers can download it with no requirements in as little as five minutes. Benefits of ASP.NET include the fact that it tends to be faster and more effective than websites and web applications built with PHP and other frameworks.  Code is compiled by ASP.NET into object code, which executes more quickly than interpreted code. With ASP.NET, problems are discovered during compilation, allowing programmers to address any coding issues right away (indeed, 2022). |

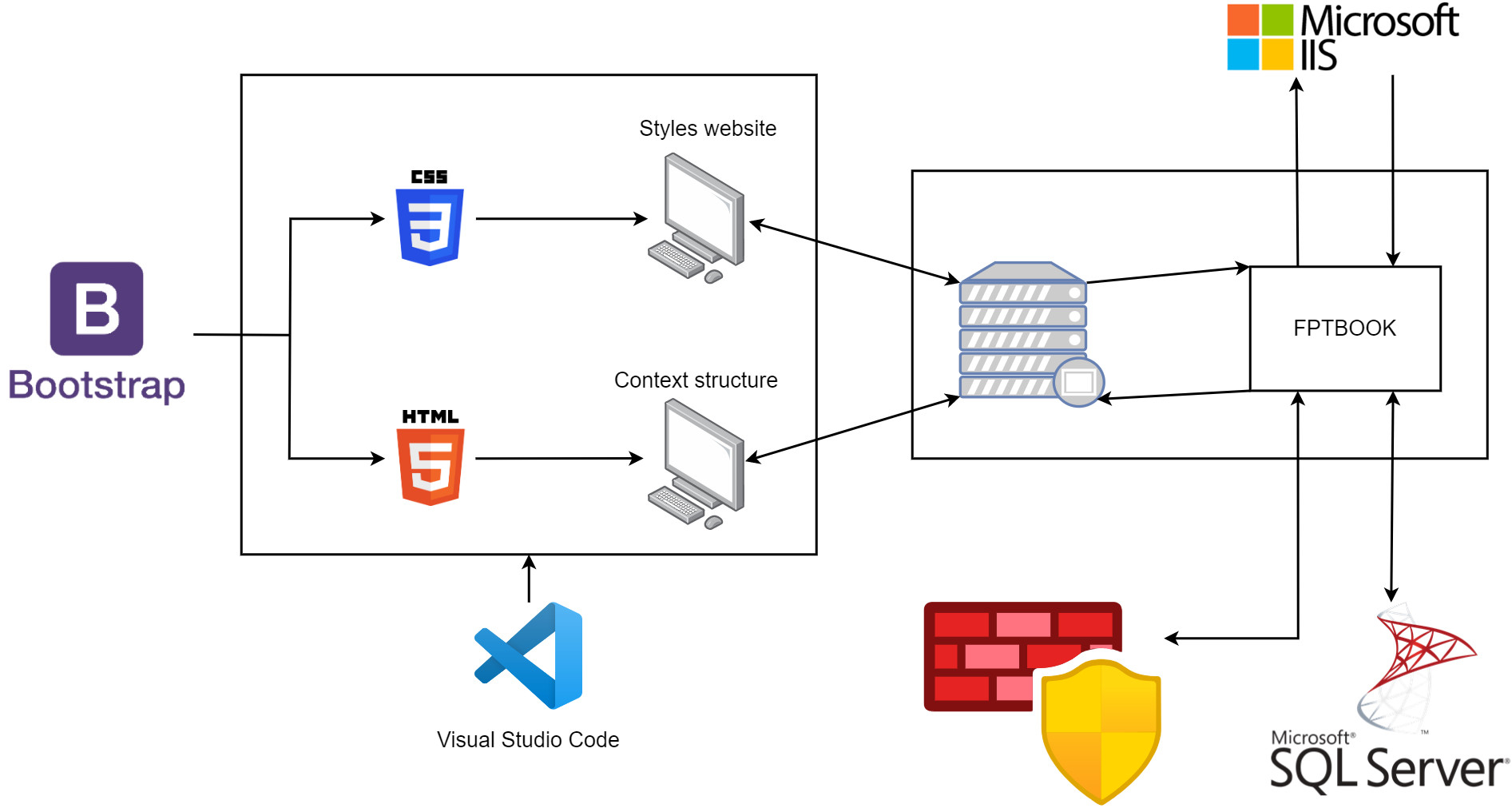
## 4. Tools for source control management

|  |  |  |  |
| --- | --- | --- | --- |
|  | **GitHub** | **GitLab** | **Why I choose?** |
| **Definition** | GitHub is a hosting service for Git repositories that offers a web-based graphical user interface. It is the biggest coding community in the planet. A project or piece of code that is uploaded to GitHub is given more exposure. Programmers can find source codes in a wide variety of languages and make and track modifications using the command-line interface Git (Gaba, 2023). | A free open and private repository, issue-following tools, and wikis are all included in the web-based Git repository known as GitLab. It is a comprehensive DevOps platform that enables experts to carry out every project work, from planning and managing the source code to monitoring and security. Teams can collaborate and create better software as a result (Kelley, 2023). | I chose Github because its advantages are as follows. First of all, Github makes it easy for us to recover data, if during development we encounter some errors, we can revert back to the previous state of the code. This saves us a lot of time when we don't have to delete each newly written code or rewrite deleted code when needed. The second, all members can work together, making the members' activities easily manageable and transparent. |
| **Characteristic** | - Easy Project Management: Project managers and developers collaborate on projects on GitHub to plan, monitor, and update their work in order to keep projects transparent and on time  - Increased Safety With Packages: Packages may be made public to the open-source community, private to the team, or both. You can use or reuse the packages by getting them from GitHub  - Effective Team Management: Using GitHub, the team can stay coordinated and on the same page. The team may concentrate on the code with the help of moderating tools like Issue and Pull Request Locking  - Improved Code Writing: Organizations use pull requests to examine, create, and suggest new code. Before making any source code changes, team members can discuss any implementations and suggestions using these  - Increased Code Safety: GitHub utilizes specialized tools to find and examine code vulnerabilities that other tools frequently overlook. Worldwide development teams collaborate to safeguard the entire software supply chain  - Easy Code Hosting: The documentation and all the code are in one location. On GitHub, there are millions of repositories, and each one has its own set of tools for hosting and disseminating code  (Gaba, 2023) | - Git-based version control and repository management  - Boards, bug tracking, and issue management.  - The Review Apps tool and the Code Review feature.  - The Continuous Integration and Deployment (CI/CD) tool GitLab  - Code Reliability (Code Climate)  - Tool ChatOp (Mattermost)  - Help Desk (ticketing system)  (KUMAR, 2022) |

## 5. Software Development Models

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Spiral** | **Watefall** | **Why I choose?** |
| **Definition** | A spiral model is a combination of a sequential model and a prototype model. This model is ideal for large projects with continuous improvement. There are certain activities that are performed iteratively (spiral), resulting in a small prototype of large software. Then the same activity is repeated for all spirals until all software is built (softwaretestinghelp, 2023). | The waterfall model is a sequential model that divides software development into predefined phases. Each phase must be completed before starting the next phase without overlapping phases. Each phase is designed to perform specific activities during the SDLC phase (Martin, 2023). | I have chosen Waterfall model to be the Software Development Models for my team's project because of the following reasons. First of all, our project is a small project with fairly simple requirements, so this software development model is a great fit for us. Also, because we do 2 reports, the first one is on the theoretical and design level, the second one starts coding. This is perfect for waterfall because it requires work to be done sequentially |
| **Characteristic** | - Development is fast  - Large projects / software are strategically created and processed  - The risk assessment is correct  - Control all phases of development  - More and more features are systematically added  - Software is created early  - There is room for customer feedback and changes will be implemented more quickly  (softwaretestinghelp, 2023) | - Each phase must be completed before the next development phase  - Suitable for small projects with well-defined requirements  - Quality Assurance Testing (Verification and Validation) must be performed before completing each step  - Detailed documentation is provided at each phase of the software development cycle  - The project is completely dependent on the project team and client intervention is minimized  - Software changes are made during the development process  (Martin, 2023) |

## 6. Illustrate all my findings on how to use these by drawing the overview



To build FPTBook website, using Visual Studio Code, I used programming languages and tools such as: CSS, HTML, Bootstrap framework, SQL Server, IIS and allow other devices to view it.

# References

byjus, 2023. *byjus.* [Online]   
Available at: https://byjus.com/gate/difference-between-css-and-scss/#:~:text=What%20is%20SCSS%3F,thus%20often%20called%20Sassy%20CSS.

Deshpande, C., 2023. *simplilearn.* [Online]   
Available at: https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular#what\_is\_angular

Gaba, I., 2023. *simplilearn.* [Online]   
Available at: https://www.simplilearn.com/tutorials/git-tutorial/what-is-github

gcfglobal, 2023. *gcfglobal.* [Online]   
Available at: https://edu.gcfglobal.org/en/windowsbasics/all-about-windows/1/

geeksforgeeks, 2019. *geeksforgeeks.* [Online]   
Available at: https://www.geeksforgeeks.org/introduction-to-c-sharp/

geeksforgeeks, 2022. *geeksforgeeks.* [Online]   
Available at: https://www.geeksforgeeks.org/html-introduction/

indeed, 2022. *indeed.* [Online]   
Available at: https://www.indeed.com/career-advice/career-development/what-is-aspnet

Jackson, P., 2023. *guru99.* [Online]   
Available at: https://www.guru99.com/what-is-php-first-php-program.html

javatpoint, 2021. *javatpoint.* [Online]   
Available at: https://www.javatpoint.com/what-is-apache

Kelley, K., 2023. *simplilearn.* [Online]   
Available at: https://www.simplilearn.com/tutorials/git-tutorial/what-is-gitlab#:~:text=GitLab%20is%20a%20web%2Dbased,management%20to%20monitoring%20and%20security.

KUMAR, R., 2022. *devopsschool.* [Online]   
Available at: https://www.devopsschool.com/blog/what-is-gitlab-and-how-it-works-an-overview-and-its-use-cases/

learnrazorpages, 2023. *learnrazorpages.* [Online]   
Available at: https://www.learnrazorpages.com/

linux, 2023. *linux.* [Online]   
Available at: https://www.linux.com/what-is-linux/

Martin, M., 2023. *guru99.* [Online]   
Available at: https://www.guru99.com/what-is-sdlc-or-waterfall-model.html

Moore, L., 2018. *techtarget.* [Online]   
Available at: https://www.techtarget.com/searchoracle/definition/MySQL

Morales, J., 2022. *mindonmap.* [Online]   
Available at: https://www.mindonmap.com/blog/drawio-review/

Morris, W., 2022. *elegantthemes.* [Online]   
Available at: https://www.elegantthemes.com/blog/wordpress/microsoft-iis?utm\_source=Blog&utm\_medium=Manual%20Divi%20Targets&utm\_campaign=Google%20Search&retargeting=off&gclid=CjwKCAiAuaKfBhBtEiwAht6H7-JzEKEyO2mAqsOWpIzRKCaoU33FWcvSf1-2ot-\_zHuUf9ViVuXR1xoChecQAvD\_BwE#wh

opensource, 2023. *opensource.* [Online]   
Available at: https://opensource.com/resources/linux

PÉREZ, S. D., 2021. *intelequia.* [Online]   
Available at: https://intelequia.com/en/blog/post/2948/what-is-microsoft-sql-server-and-what-is-it-for

rjsystems, 2023. *rjsystems.* [Online]   
Available at: https://www.rjsystems.nl/en/3200.php

softwaretestinghelp, 2023. *softwaretestinghelp.* [Online]   
Available at: https://www.softwaretestinghelp.com/spiral-model-what-is-sdlc-spiral-model/

theme-junkie, 2021. *theme-junkie.* [Online]   
Available at: https://www.theme-junkie.com/what-is-figma/

tutorialspoint, 2023. *tutorialspoint.* [Online]   
Available at: https://www.tutorialspoint.com/laravel/laravel\_overview.htm#

tutorialspoint, 2023. *tutorialspoint.* [Online]   
Available at: https://www.tutorialspoint.com/typescript/typescript\_overview.htm

tutorialspoint, 2023. *tutorialspoint.* [Online]   
Available at: https://www.tutorialspoint.com/jquery/jquery-overview.htm

tutorialspoint, 2023. *tutorialspoint.* [Online]   
Available at: https://www.tutorialspoint.com/bootstrap/bootstrap\_overview.htm#:~:text=Bootstrap%20is%20a%20sleek%2C%20intuitive,uses%20HTML%2C%20CSS%20and%20Javascript.

tutorialspoint, 2023. *tutorialspoint.* [Online]   
Available at: https://www.tutorialspoint.com/foundation/foundation\_overview.htm

tutorialspoint, 2023. *tutorialspoint.* [Online]   
Available at: https://www.tutorialspoint.com/javascript/javascript\_overview.htm

w3schools, 2023. *w3schools.* [Online]   
Available at: https://www.w3schools.com/asp/razor\_intro.asp