|  |
| --- |
|  |
| Capstone Project Document |

**BOOKAHOLIC SOCIAL NETWORK**

Report #2 – Architecture Design

|  |  |  |
| --- | --- | --- |
| **Bookaholic Social Network** | | |
| **Group Members** | Thái Thị Cẩm Vân | SE03409 |
| Phạm Thị Huyền | SE03416 |
| Vũ Hải Đăng | SE03380 |
| Nguyễn Thị Hải Yến | SE03171 |
| Chu Minh Hải | SE03143 |
| **Supervisor** | Mr.Nguyễn Văn Sang | |
| **Project code** | BSN | |

**- Hanoi, 10/2016 -**

# SIGNATURE PAGE

AUTHOR: Thái Thị Cẩm Vân --/10/2016

Developer

REVIEWERS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -- /--/2016

Team member

APPROVAL: Nguyễn Văn Sang --/--/2016

Supervisor

Record of change

\*A - Added M - Modified D – Deleted

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Effective Date | Changed Item | A,M,D | Change Description | Reason for Change | Rev. Number |
| --/Nov/2016 | Create Architecture Design | A | First version | Create Architecture Design | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table of Contents

[SIGNATURE PAGE 1](#_Toc468613994)

[1 PROJECT OVERVIEW 3](#_Toc468613995)

[1.1 Purpose 3](#_Toc468613996)

[1.2 Scope 3](#_Toc468613997)

[1.3 Definitions, Acronyms, Abbreviations 3](#_Toc468613998)

[1.4 References 3](#_Toc468613999)

[1.5 Overview 3](#_Toc468614000)

[2 CHOICE OF ARCHITECTURE DESIGN 4](#_Toc468614001)

[2.1 MVC Model 4](#_Toc468614002)

[2.1.1 MVC Model Overview 4](#_Toc468614003)

[2.1.2 Advantages and disadvantages of MVC Model 5](#_Toc468614004)

[2.1.3 The reasons for choosing MVC Model 5](#_Toc468614005)

[2.2 .Net Framework 6](#_Toc468614006)

[2.2.1 .Net Framework Overview 6](#_Toc468614007)

[2.2.2 Advantages and disadvantages of .Net Framework 6](#_Toc468614008)

[2.2.3 The reasons for choosing .Net Framework 6](#_Toc468614009)

[3 ARCHITECTURAL REPRESENTATION 7](#_Toc468614010)

[4 ARCHITECTURAL GOALS AND CONSTRAINTS 8](#_Toc468614011)

[5 Use-case View 8](#_Toc468614012)

[5.1 User Group Function 9](#_Toc468614013)

[5.1.1 Guest Group Function 9](#_Toc468614014)

[5.1.2 Reader&Author Group Function 10](#_Toc468614015)

[5.2 Administrator Group Function 16](#_Toc468614016)

[5.2.1 Common Module 16](#_Toc468614017)

[5.2.2 Users Accounts Management 17](#_Toc468614018)

[5.2.3 Books Management 18](#_Toc468614019)

[5.2.4 Sliders Management 19](#_Toc468614020)

[5.2.5 Publishers Management 20](#_Toc468614021)

[5.2.6 Groups Management 21](#_Toc468614022)

[5.2.7 Statistics 21](#_Toc468614023)

[6 Logical View 21](#_Toc468614024)

[6.1 Overview 21](#_Toc468614025)

[6.2 Architecturally Significant Design Packages 22](#_Toc468614026)

[7 Process view 24](#_Toc468614027)

[8 Deployment View 26](#_Toc468614028)

[9 Quality 27](#_Toc468614029)

# PROJECT OVERVIEW

## Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions that have been made on the system.

## Scope

The scope of this document is to depict the architecture of the Bookaholic Social Network created by BSN capstone project team.

## Definitions, Acronyms, Abbreviations

|  |  |  |
| --- | --- | --- |
| **Acronym** | **Definition** | **Note** |
| BSN | Bookaholic Social Network |  |
| DB | Database |  |
| MVC | Model view control |  |
| IDE | Integrated development environment |  |
| Q&A | Question and answer |  |
| GUI | Graphic user interface |  |

Table 1: Definitions and Acronyms

## References

* BSN\_ Software Requirements Specification\_v1.0
* BSN\_Data Design\_v1.0

## Overview

The Software Architecture Document contains the following subsections:

* **Section 1**: Provide an overview of entire Software Architecture Document.
* **Section 2**: Choice of Architecture Design
* **Section 3**: Architectural Representation
* **Section 4**: Architectural Goals and Constraints
* **Section 5**: Use-Case view
* **Section 6**: Logical View
* **Section 7**: Process View
* **Section 8**: Deployment view
* **Section 9**: Quality

# CHOICE OF ARCHITECTURE DESIGN

## MVC Model

The purpose of BSN is developing a social network, where people can share and discus about book. The system of BSN is structured based on MVC combined with layered architecture.

### MVC Model Overview

The **model-view-controller or MVC** is software architecture commonly used for creating web applications or software. In other words, it's a structure for web applications to follow in order to ensure efficiency and consistency. Many of the most popular frameworks use the MVC architecture, including ASP.NET, CodeIgniter, Zend, Django, and Ruby on Rails. At the same time, there are many web developers who don't use a coding framework yet still set up their applications to follow the MVC structure.

The Model-View-Controller (MVC) design pattern assigns objects in an application one of  
three roles: model, view, or controller. The pattern defines not only the roles objects play in  
the application, it defines the way objects communicate with each other. Each of the three  
types of objects is separated from the others by abstract boundaries and communicates with  
objects of the other types across those boundaries. The collection of objects of a certain MVC  
type in an application is sometimes referred to as a layer—for example, model layer.

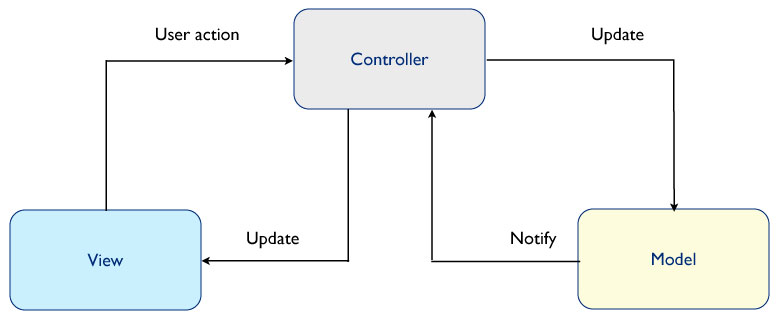


Figure 1: MVC Model

In addition to dividing the application into three kinds of components, the MVC design defines the interactions between them:

* **Controller:** The controller can be considered the "middle man" of the application. It works with the user, taking in data, and then working with the model to get the appropriate data or calculation, and then working with the view to show the response to the user.
* **Model:** A model is simply a representation of something we need to deal within our application. It is a "model" for something we must represent in code, such as a book, user, bank account, or whatever. The model is responsible for holding the functions and variables that are involved with whatever it's representing. You can think of a model's logic as the core concept to object oriented programming — models are just our "classes". However, don't let this confuse you as controllers are technically structured as classes as well.
* **View:** Finally, after the controller requests information from the model it sends it to a view. A view is just like the application's templating system — there might be a view for a certain type of page layout *(profile page)*, a mobile view, or a view for a particular theme/skin. A view will contain all of the markup, CSS, and etc. that you traditionally use with creating a static web page.

### Advantages and disadvantages of MVC Model

* Advantages:
* MVC separates system into components, which can be developed, maintained and upgraded individually without pausing system.
* Develop tools is useful and easy to use.
* Large of documentary sources.
* Disadvantages:
* For small projects that apply MVC model caused cumbersome, time consuming in development process.
* Time consuming to transits data between components.
* Not suitable for agent-oriented applications such as interactive mobile and robotics applications.
* Multiple pairs of controllers and views based on the same data model make data model change expensive.
* The division between the View and the Controller is not clear in some cases.

### The reasons for choosing MVC Model

* MVC makes parts of system can be developed individually and simultaneously to reduce developing time.
* Better support for test-driven development.
* Tools is useful and documentary source is large makes MVC is easy to develop.
* BSN system is not complete system, now. We built the system that towards extensibility and maintainability in the future.

## .Net Framework

### .Net Framework Overview

.NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library known as Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages) across several programming languages.

The .NET Framework is a technology that supports building and running the next generation of applications and XML Web services. The .NET Framework is designed to fulfill the following objectives:

* To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
* To provide a code-execution environment that minimizes software deployment and versioning conflicts.
* To provide a code-execution environment that promotes safe execution of code, including code created by an unknown or semi-trusted third party.
* To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
* To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
* To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

### Advantages and disadvantages of .Net Framework

* Advantages
* Less time to produce product
* Less Complexity.
* Easily to access complex O.S functions
* Easily to build Data Oriented Project, support huge DB functions.
* Managed
* Support Both Windows and Web Application.
* Easy to create Dynamic sites.
* Disadvantages
* Not suitable for High End Application
* Low performance compare to C, C++.
* Unavailability of build in methods.
* .NET framework is free to download but Code Editor is costly.
* Only few O.S supports .NET.

### The reasons for choosing .Net Framework

* Consistent with BSN system.
* There are many plugins and resources which support creating a website using MVC model.
* Many members can use and have experiences using C# (a language of .Net Framework)

# ARCHITECTURAL REPRESENTATION

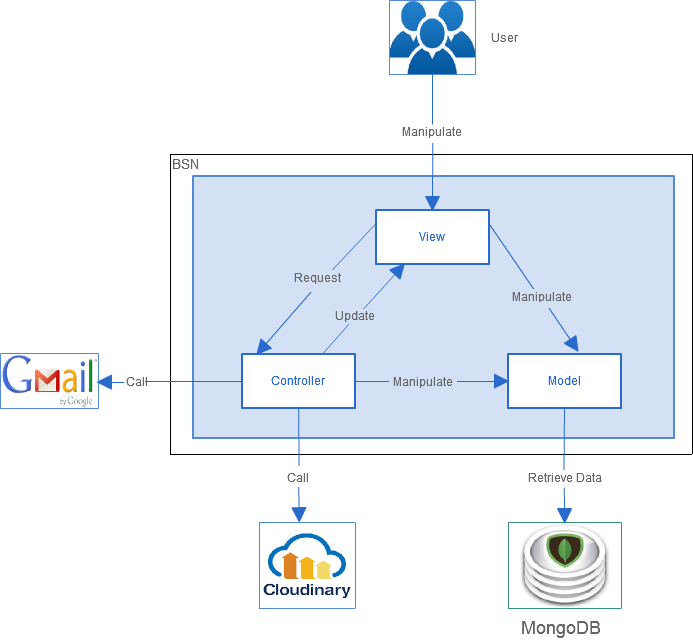


Figure 2: System Overview

* **Model** are include **Models** and **View Models**. View Models is an object that defines how the data will be sent over the network to remove circular references from Models, hide particular properties that clients are not supposed to view, omit some properties in order to reduce payload size, flatten object graphs that contain nested objects, to make them more convenient for clients, avoid “over-posting” vulnerabilities and decouple your service layer from your database layer. Models is where the application’s data objects are stored. A model object is in charge of encapsulating application state and one object could be related to other objects establishing a one-to-one or one-to-many relationship.
* **View** is what is presented to the users and how users interact with the system. The view is expected to render the model in a meaningful way to the user. In BSN, the view is made with .cshtml file including CSS, HTML, JavaScript and jQuery, it sends user gestures to controller and allows controller to select view.
* **Controller** is the decision maker and the glue between the model and view; it handles user actions and gestures, and responds to user events.

# ARCHITECTURAL GOALS AND CONSTRAINTS

* **Availability:**
* The application must be available 95% of time. Users can access to it everywhere from there .Web browser with internet connection.
* **Maintainability:**
  + Coding standards and naming conventions:
    - Output of the project must include coding standards and naming conventions documentations. Implementation code must be easy to maintain.
    - All code must be clearly commented, including class, method documentations.
    - If some components are reused, the documentations of those components must also be included.
* Design:
  + - The design of the system must be loosely coupled that chances on some module will not affect others.
* Logging:
  + - All the errors should be logged, supporting for bug fixing and maintenance.
    - All strange or sensitive situations should also be logged.
* **Usability:**
  + Intuitiveness: all help/error messages are simple to understand; user can know exactly how to do each feature after one time using it.
* **Capacity and scalability:**
  + Throughput, storage and growth requirements.

# Use-case View

* **This application includes two parts:**
* The first part is User module. User module includes Guest and Reader&Author.
* Next part is Administrator module.

## User Group Function

### Guest Group Function

#### Register Module

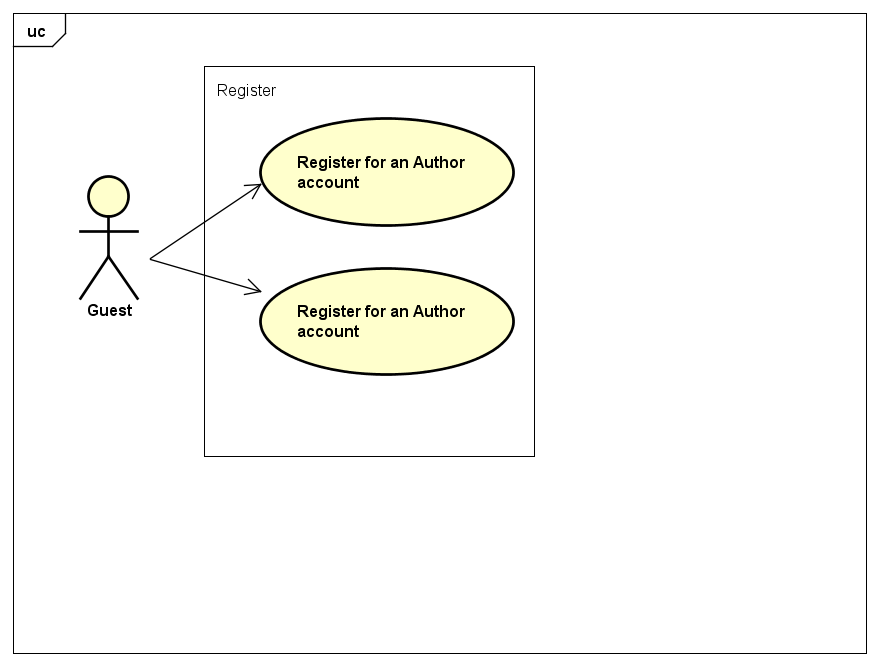


Figure 3: Register Module

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use-case name | Actor | Description |
| 1 | Register for a Reader account | Guest | Create new account for Guest as a Reader to use functions of Reader |
| 2 | Register for an Author account | Guest | Create new account for Guest as an Author to use functions of Author |

Table 2: Register Module UC

### Reader&Author Group Function

#### Account Management Module

#### 

Figure 4: Account Management Module

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use-case name | Actor | Description |
| 1 | Login by registed account | Reader, Author | User Login by registed account |
| 2 | Logout | Reader, Author | User want to logout their account |
| 3 | Edit Profile | Reader, Author | User want to edit profile information and user’s image |
| 4 | Reset Password | Reader, Author | User forgot password |

Table 3: Account Management Module UC

#### Post Management Module

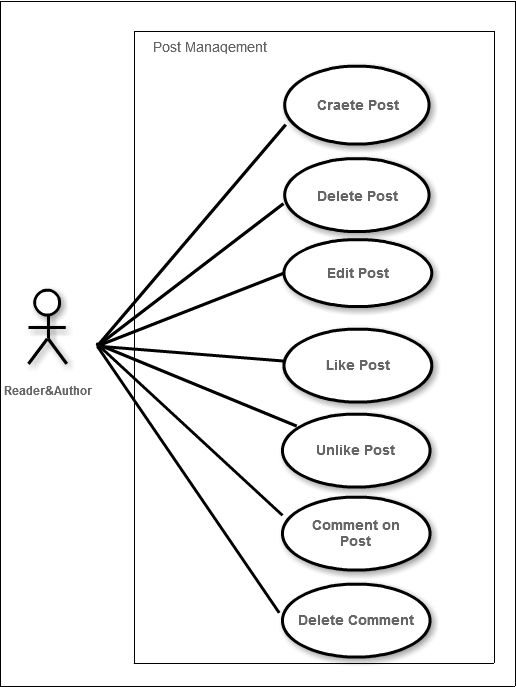


Figure 5: Post Management Module

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use-case name | Actor | Description |
| 1 | Create Post | Reader, Author | Allow user to create a post |
| 2 | Delete Post | Reader ,Author | Allow user to delete a post |
| 3 | Edit Post | Reader, Author | Allow user to edit a post |
| 4 | Like Post | Reader, Author | Allow user to like a post |
| 5 | Unlike Post | Reader, Author | Allow user to unlike a post |
| 6 | Comment on Post | Reader ,Author | Allow user to comment on Post |
| 7 | Delete comment | Reader, Author | Allow user to delete comment |

Table 4 : Post Management Module UC

#### Groups Management

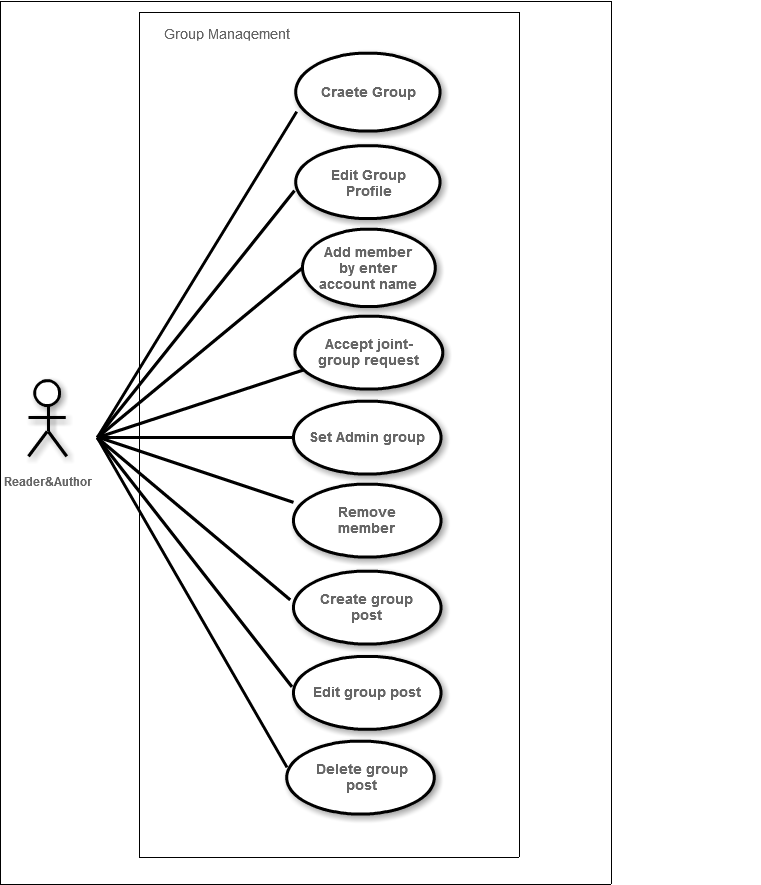


Figure 6: Groups Management Module

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use-case name | Actor | Description |
| 1 | Create Group | Reader, Author | Allow user to create a group |
| 2 | Edit Group Profile | Reader, Author | Allow user to edit group’s profile |
| 3 | Add member by enter account name | Reader, Author | Allow user(group creator) to add new member by search account name |
| 4 | Accept joint-group request | Reader, Author | Allow user(group creator) to accept joint-group request |
| 5 | Set Admin Group | Reader, Author | Allow user(group creator) to set other member as admin |
| 6 | Remove member | Reader, Author | Allow user(group creator) to remove member |
| 7 | Create Group Post | Reader, Author | Allow user(member of group ) to create group post |
| 8 | Edit Group Post | Reader, Author | Allow user to edit group’s post that created with their own |
| 9 | Delete Group Post | Reader, Author | Allow user to delete group’s post that created with their own |

Table 5 : Groups Management Module UC

#### Interactions Module

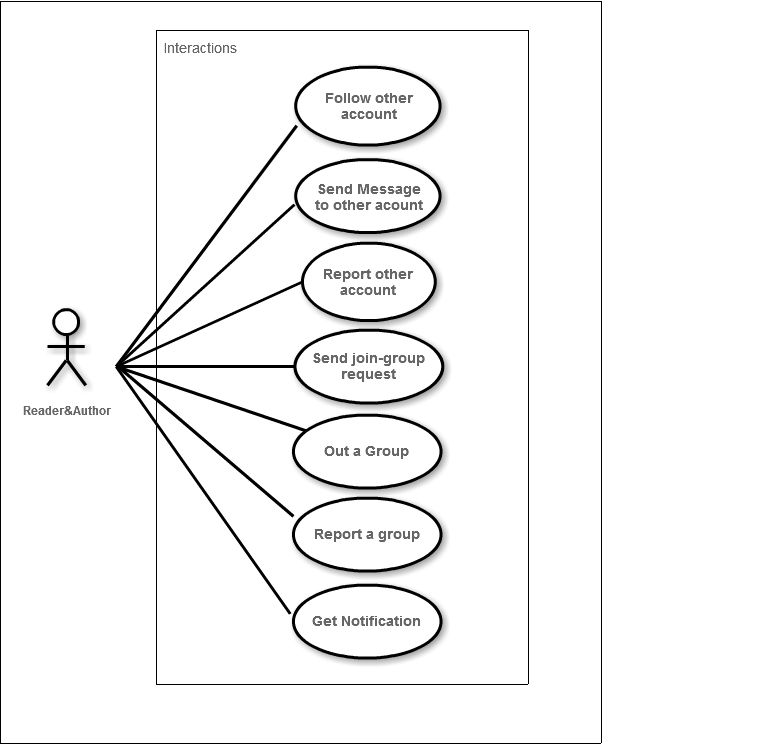


Figure 7: Interactions Module

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use-case name | Actor | Description |
| 1 | Follow other account | Reader, Author | Allow user to follow other account |
| 2 | Send message to other account | Reader, Author | Allow user to send message to other account |
| 3 | Report other account | Reader, Author | Allow user to report other account or group |
| 4 | Send join-group request | Reader, Author | Allow user to send request to join a group |
| 5 | Out a group | Reader, Author | Allow user to get out of group |
| 6 | Report a group | Reader, Author | Allow user to report a group |

Table 6 : Interactions Module UC

#### Book Management

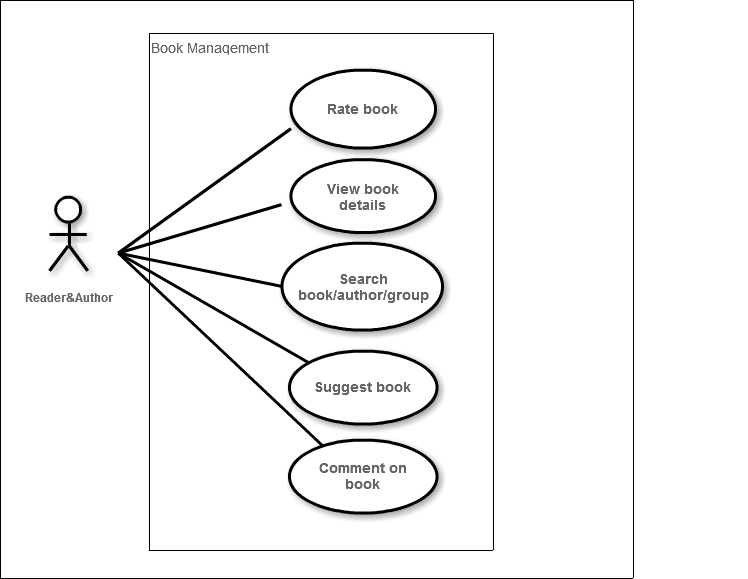


Figure 8 : Book Management Module

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use-case name | Actor | Description |
| 1 | Rate book | Reader, Author | Allow user to rate book |
| 2 | View book details | Reader, Author | Allow user to view book details |
| 3 | Search book/user/group | Reader, Author | Allow user to search about books,users,groups |
| 4 | Suggest book | Reader, Author | Suggest book to user |
| 5 | Comment on book | Reader, Author | Allow user to comment on book |

Table 7 : Book Management Module UC

## Administrator Group Function

### Common Module

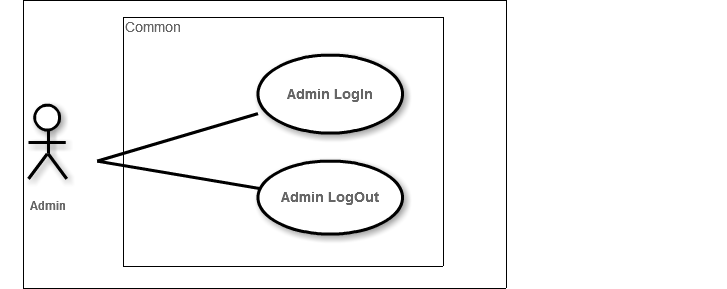


Figure 9 : Common Module

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use-case name | Actor | Description |
| 1 | Admin login | Administrator | This function allows Admin logins into website |
| 2 | Admin logout | Administrator | This function allows Admin logouts of website |

Table 8 : Common Module UC

### Users Accounts Management

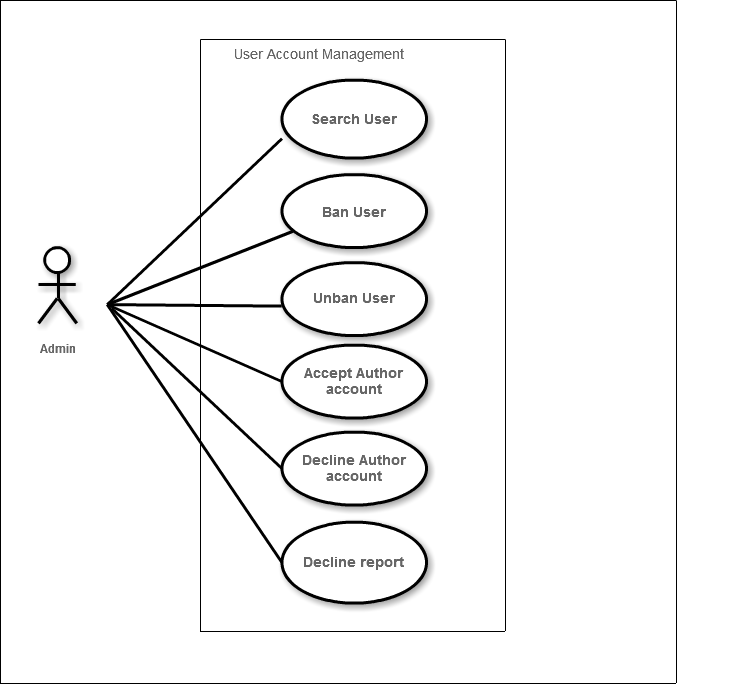


Figure 10 : User Account Management Module

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use-case name | Actor | Description |
| 1 | Search User | Administrator | Allow Administrator to search an user’s account |
| 2 | Ban User | Administrator | Allow Administrator to deactivate user’s account |
| 3 | Unban User | Administrator | Allow Administrator to activate an user’s account |
| 4 | Accept Author account | Administrator | Allow Administrator to accept request create author’s account |
| 5 | Decline Author account | Administrator | Allow Administrator to |
| 6 | Decline Report | Administrator | Allow Administration to |

Table 9 : User Account Management Module UC

### Books Management

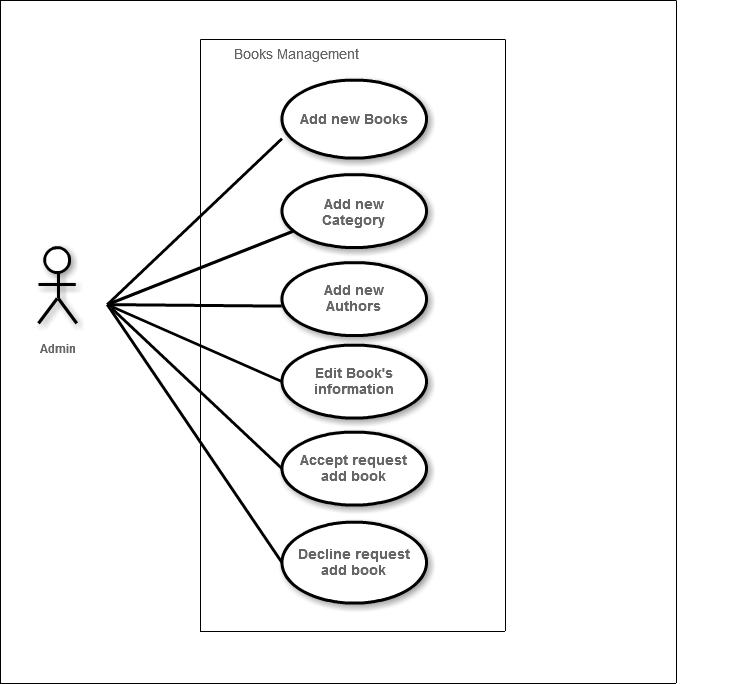


Figure 11 : Books Management Module

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use-case name | Actor | Description |
| 1 | Add new books | Administrator | Allow Administrator to add new books |
| 2 | Add new category | Administrator | Allow Administrator to add new category |
| 3 | Edit book information | Administrator | Allow Administrator to edit book’s information |
| 4 | Accept request add book | Administrator | Allows Administrator to add new book from user’s request |
| 5 | Decline request add book | Administrator | Allow Administrator to decline request add book from user |
| 6 | Add new Author | Administrator | Allows Administrator to add new Author |

Table 10 : Books Management Module UC

### Sliders Management

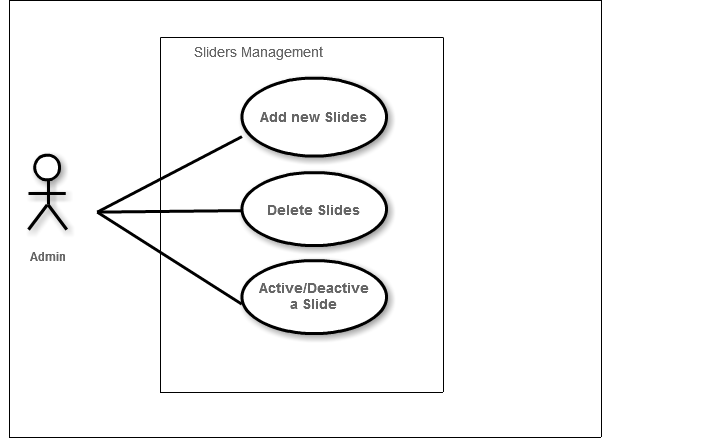


Figure 12 : Sliders Management Module

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use-case name | Actor | Description |
| 1 | Add new Slide | Administrator | Allow Administrator to add new slides |
| 2 | Delete Slides | Administrator | Allow Administrator to delete slides |
| 3 | Active/Deactive Slides | Administrator | Allows Administrator to active or deactive Slides |

Table 11 : Sliders Management Module UC

### Publishers Management

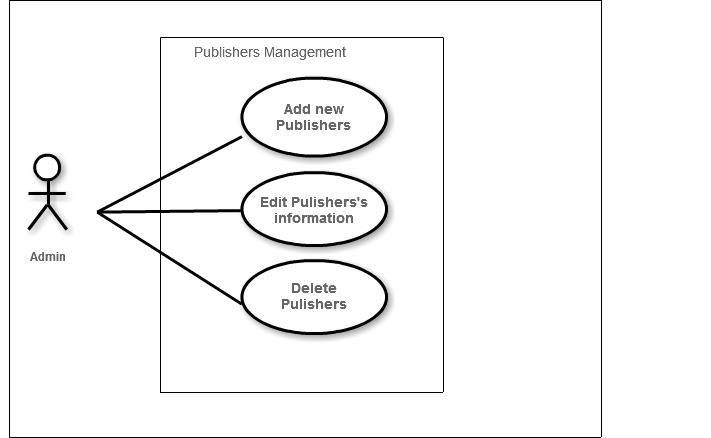


Figure 13 : Publishers Management Module

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use-case name | Actor | Description |
| 1 | Add new Publishers | Administrator | Allow Administrator to add new pulishers |
| 2 | Edit Pulisher’s information | Administrator | Allow Administrator to edit pulisher’s information |
| 3 | Delete Pulishers | Administrator | Allows Administrator to delete pulishers |

Table 12 : Pulishers Management Module UC

### Groups Management

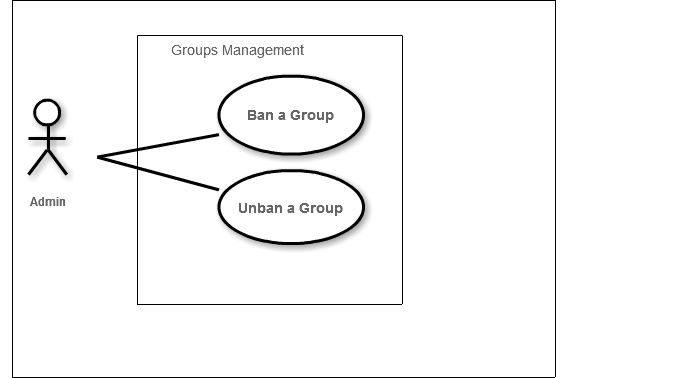


Figure 14 : Groups Management Module

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use-case name | Actor | Description |
| 1 | Ban a Group | Administrator | Allow Administrator to deactive a group |
| 2 | Unban a Group | Administrator | Allow Administrator to activate a group |

Table 13: Groups Management Module UC

### Statistics

# Logical View

## Overview

Logical View includes Package diagram and Class diagram. Package diagram describes the organization of packages and elements. Class Diagram provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. It provides a wide variety of usages; from modeling the domain-specific data structure to detailed design of the target system

* + Controller contain the interface between:
    - Associated models
    - Associated views
    - The input devices (e.g., keyboard, pointing device, time).
    - Send commands to the model to update the model's state.
    - Create queries to DB
    - Process data.
  + Model is:
    - the domain-specific software simulation
    - Or implementation of the application's central structure.
  + View: deal with everything graphical:
    - Requests data from their model
    - Display the data.

## Architecturally Significant Design Packages

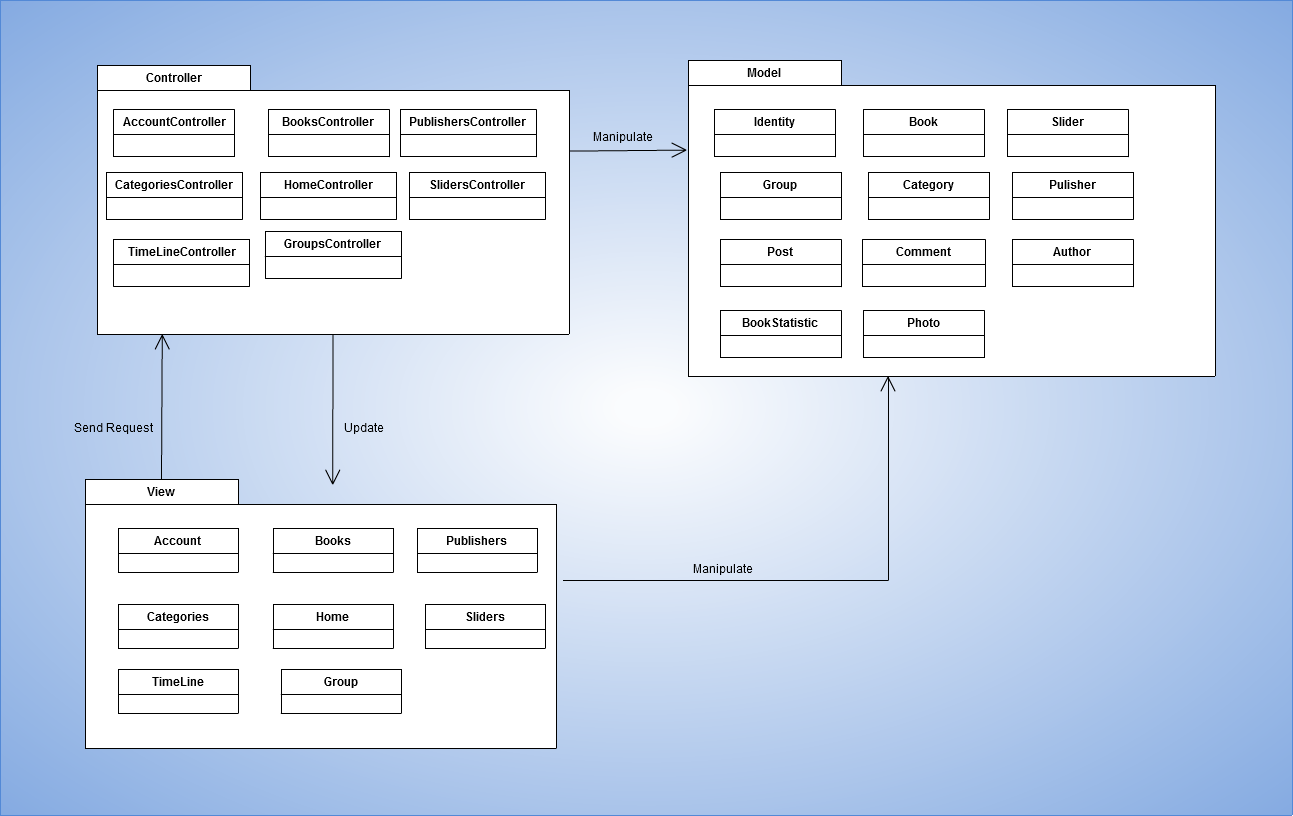


Figure 15: Package Diagram

* **Model:**

|  |  |  |
| --- | --- | --- |
| No | Model class | Role |
|  | Identity | Description entity of User in database |
|  | Book | Description entity of Book in database |
|  | Slider | Description entity of Conversation in database |
|  | Group | Description entity of Group in database |
|  | Publisher | Description entity of Publisher in database |
|  | Post | Description entity of Post in database |
|  | Comment | Description entity of Comment in database |
|  | Author | Description entity of Author in database |

Table 14 : Model list

* **Controller:**

|  |  |  |
| --- | --- | --- |
| No | Controller class | Role |
|  | AccountController | * Receive request login, logout, register from client then process data and respond login view, logout view, register view and login, logout, register status. |
|  | HomeController | * Receive request to home page from client. * Handle request from client and process data to get all data of home page. * Respond data back to Home View. |
|  | BooksController | * Receive request about book from client. Process data and respond to client. |
|  | PublishersController | * Receive request about publisher from client. Process data and respond to client. |
|  | CategoriesController | * Receive request about category from client. Process data and respond to client |
|  | SlidersController | * Receive request about slider from client. Process data and respond to client. |
|  | TimeLineController | * Receive request about user’s page from client. Process data to get all data of user’s page and respond to client |
|  | GroupsController | * Receive request about group from client. Process data and respond to client. |

Table 15: Controller list

* **View:**

Include many .cshtml file

# Process view

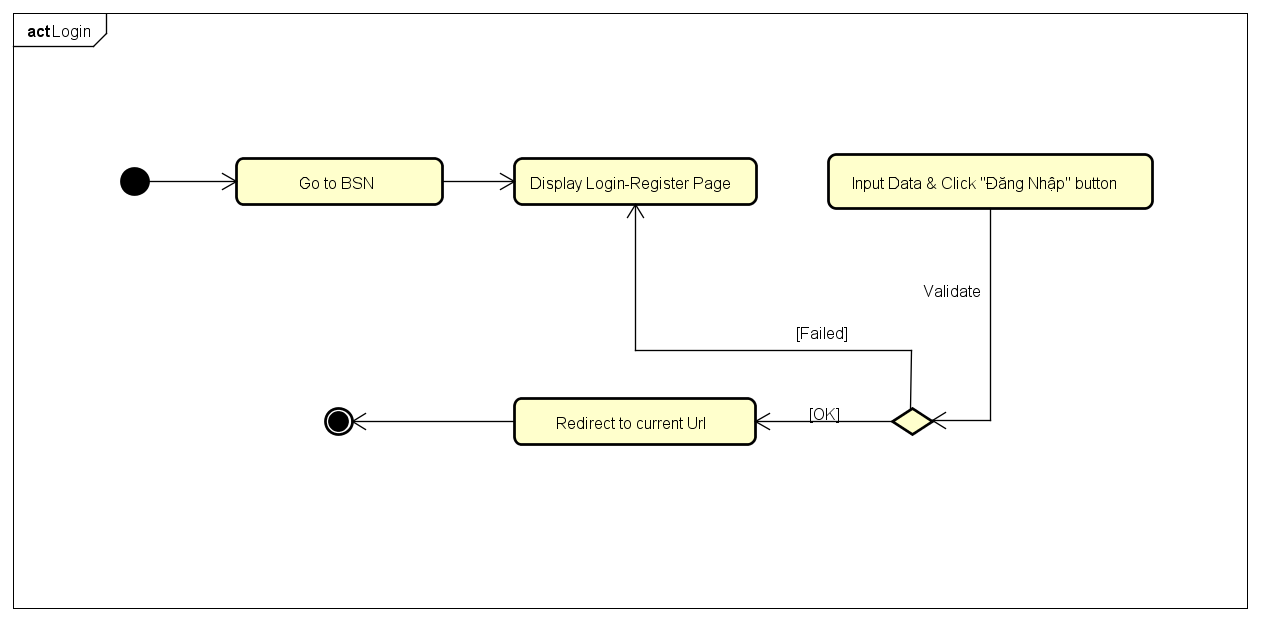


Figure 17: Login activity diagram

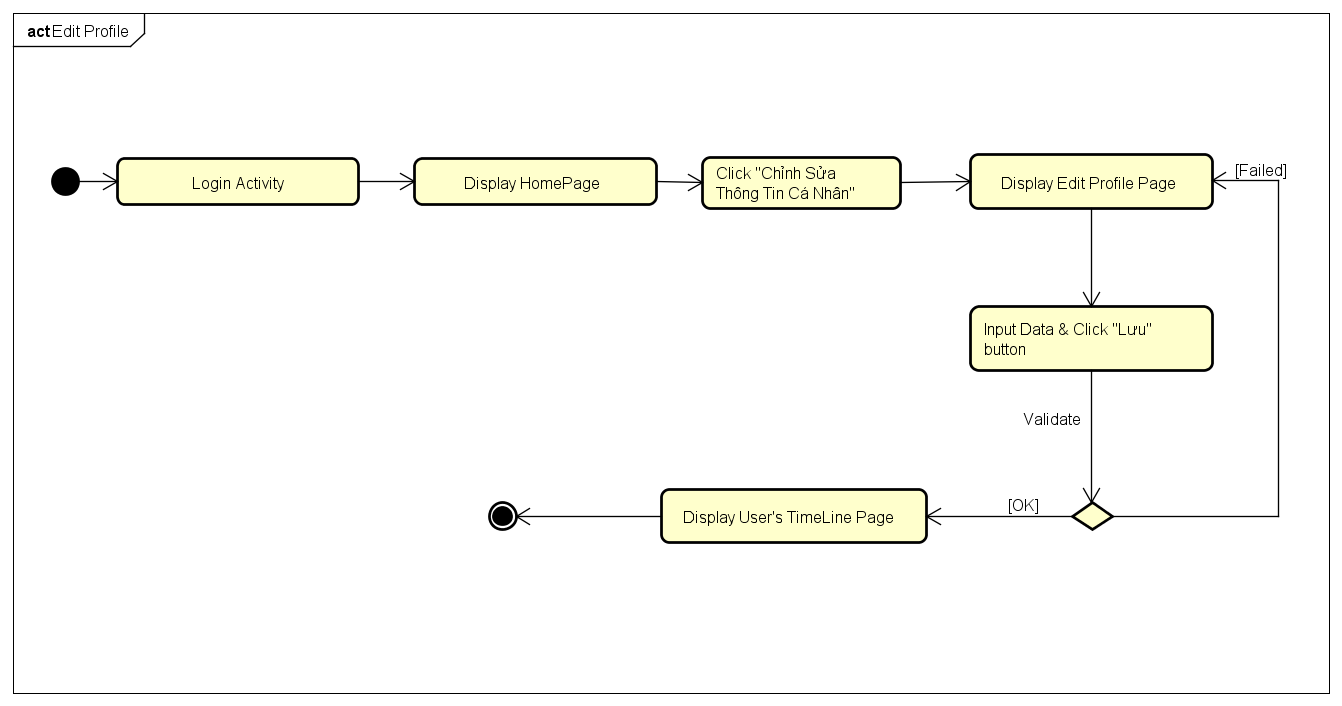


Figure 18 : Edit Profile activity diagram

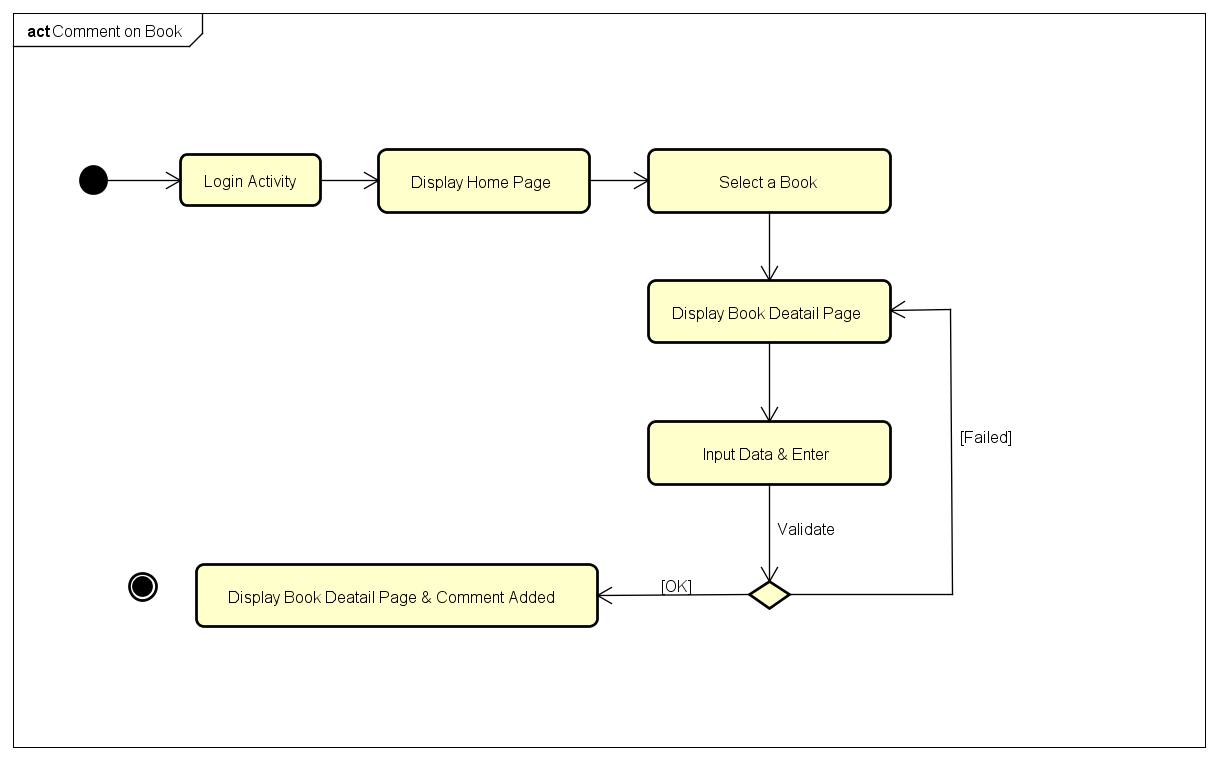


Figure 19: Comment on book activity diagram

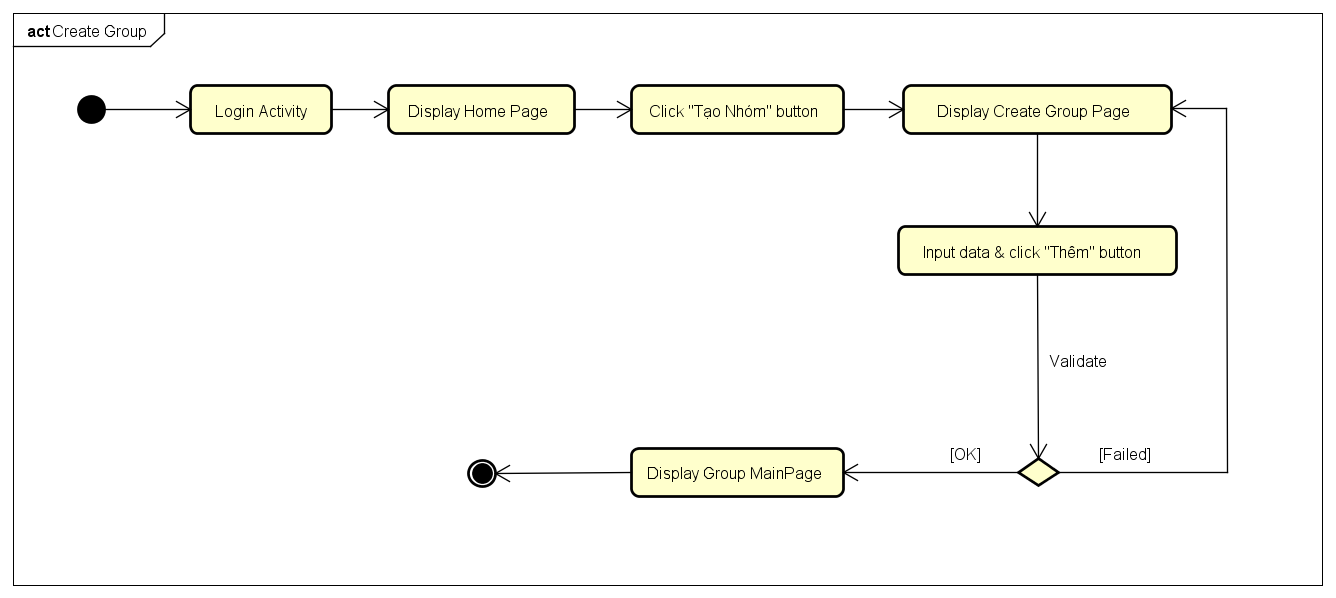


Figure 20: Create Group activity diagram

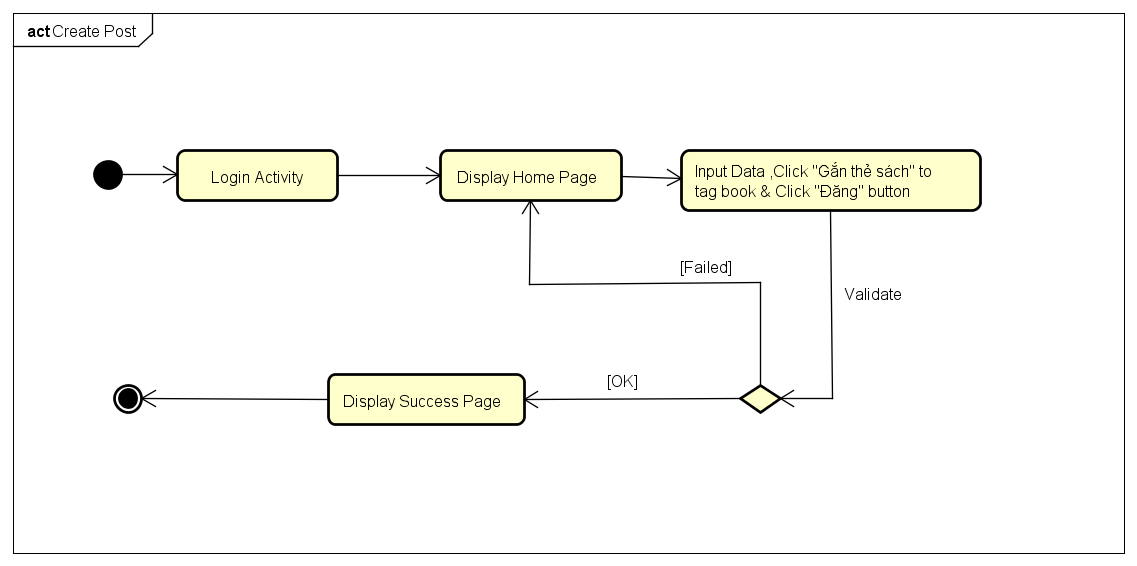


Figure 21 : Create Post activity diagram

# Deployment View

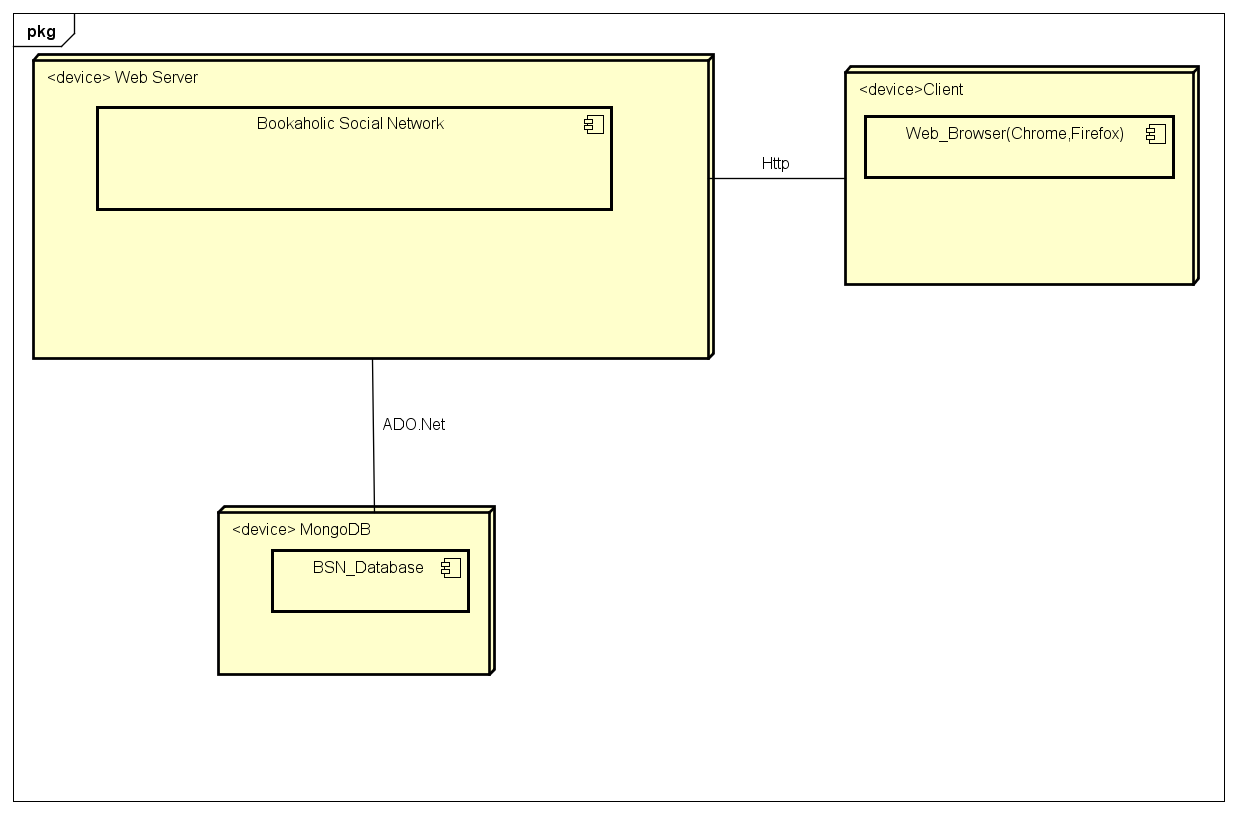


Figure 22: Deployment Diagram

|  |  |  |
| --- | --- | --- |
| **No** | **Name** | **Description** |
| 1 | MongoDB | MongoDB use to store system’s data.  Using MongoDB version 3.0.12. |
| 2 | Client | Client is web browser to use system. Chrome 50 or higher. |
| 5 | Web Server | Web server is hosted by … Web Service. |

Table 16 : Deployment Diagram Description

# Quality

Reference to: BSN\_Software requirement specification\_v1.0