|  |  |
| --- | --- |
| Logo_FPT_University_doc | **MINISTRY OF EDUCATION AND TRAINING** |

**FPT UNIVERSITY**

**CAPSTONE PROJECT DOCUMENT**

**BSMS**

**Report #3 – Software Design Specification**

|  |  |
| --- | --- |
| **Pros** | |
| **Group Member** | <ID - Name> |
| <ID - Name> |
| <ID - Name> |
| <ID - Name> |
| **Supervisor** | <Name> |
| **Ext. Supervisor** |  |
| **Project Code** | BSMS |

- Hanoi, 05/2012 -

Table of Contents

[1 Purpose 3](#_Toc483625454)

[1.1 Architecture Overview 3](#_Toc483625455)

[2 ARCHITECTURE DESIGN 3](#_Toc483625456)

[2.1 Choice of Architecture Design 3](#_Toc483625457)

[2.2 Architectural Presentation 4](#_Toc483625458)

[2.3 Component/Package Design 4](#_Toc483625459)

[3 DETAILED DESIGN 4](#_Toc483625460)

[3.1 Common Design 4](#_Toc483625461)

[3.2 Authentication Management 4](#_Toc483625462)

[3.2.1 Class Diagram 4](#_Toc483625463)

[3.2.2 Class Description 4](#_Toc483625464)

[3.2.3 Register Customer/Staff 6](#_Toc483625465)

[3.2.4 Sequence Diagram 7](#_Toc483625466)

[4 DATABASE DESIGN 8](#_Toc483625467)

[4.1 Entity Relationship Diagram 8](#_Toc483625468)

[4.2 Database Diagram 8](#_Toc483625469)

[4.3 Table Descriptions 8](#_Toc483625470)

[4.3.1 Table xxx 8](#_Toc483625471)

[4.3.2 Table xxx 8](#_Toc483625472)

# Purpose

This document is proposed with intent of describing the technical design of our bookstore Management system, mainly to give development team an overall overview of the system’s architecture, thus bringing them to a level of enlightenment of how the system should work and how to implement it. This section contains the following

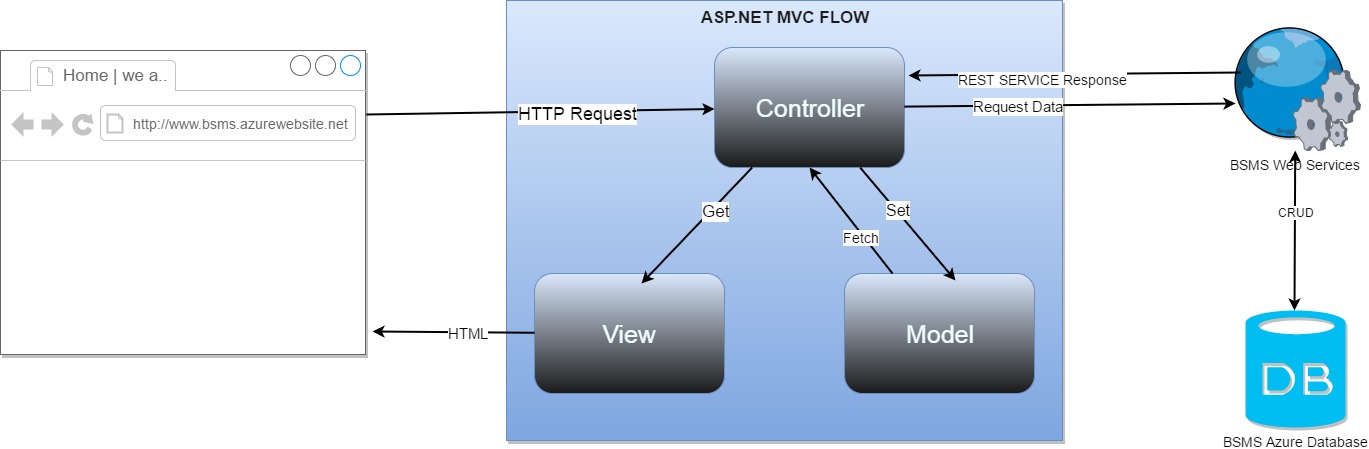
* System Architecture Design
* Database Design
* Detail Design

## Architecture Overview

# ARCHITECTURE DESIGN

## Choice of Architecture Design

BSMS web application uses the MVC (Model-View and Controller) architecture together with ASP.NET Web Services (XML,JSON), which has been one of the best software design pattern this days, This will allow our code to be more readable, more understandable and editable than the native coding pattern.



* **Models :** The model lies in the heart of data layer, this is where data are temporally stored and set
* **Views :** This lies in the heart of the presentation layers which contain and render the model data into HTML.
* **Controllers :** This serves as the system handler, it handles user input and what the system do with them. When request is send from the client side the asp.net send a request to the controller, which after the controller will find the equivalent get method and them render the called view.
* **Web Service** : This is the service that connects to the database and fetch data for the controller to make use of, CRUD methods will be implemented using this web service.

## Architectural Presentation

<Khái quát về kiến trúc của hệ thống>

## Component/Package Design

<Khái quát kiến trúc của hệ thống dưới góc nhìn về component - các thành phần biểu diễn bằng UML >

# DETAILED DESIGN

## Common Design

<Phần mô tả thiết kế cho các thành phần chung của tất cả các chức năng>

## Authentication Management

### Class Diagram

<Sơ đồ quan hệ giữa các lớp của chức năng>

### Class Description

|  |  |  |
| --- | --- | --- |
| Class | Dependency | Description |
| User | Role | The user class holds all user data and information including all the usernames and passwords in the system, this is known as the BSMS user recognition methodology. |
| Role | N/A | This class is intended to hold unique set of roles for users, since the user uses this class as dependency, user has a role, which is how the system recognise the type of user coming into the system |

#### User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | User | | | |
| **Description** | User data, this class holds all the user information including the password and username | | | |
| **Base Class** | Authentication | | | |
| **Constructor** | Default | | | |
| **Prototype** |  | | | |
| **Source File** | User.cs | | | |
| **Namespace** | BSMS.Model | | | |
| **Attributes** | Name | Type | Description | |
| Name | String | N/A | |
| Username | String | Users username is contained in this attribute | |
| PasswordHash | String | This contains 64 digit MD5 encrypted password | |
| Email | String |  | |
| RoleID | int | This hold the role of user in number format ex. (1:”Staff” , 2:”Admin”) | |
|  | Thumbnail | String | This hold the thumbnail path in the system | |
| **Methods** | Name | Input | Output | Description |
| Authentication | Username  Password | User | This function check if user is valid or not, returns null if not valid and user information if valid |
| LoggedOn | N/A | Boolean | This function is intended to check if any user is logged on to the system |

#### Role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Role | | | |
| **Description** | This contain all the user roles that exists in the system | | | |
| **Base Class** | N/A | | | |
| **Constructor** | Default | | | |
| **Prototype** |  | | | |
| **Source File** | Role.cs | | | |
| **Namespace** | BSMS.Model | | | |
| **Attributes** | Name | Type | Description | |
| Roleid | int | This is a unique integer value that holds a unique id for each role that exist in the system | |
| Role | String | Role of this role e.g. (“Admin, Staff” .... ) | |
| **Methods** | Name | Input | Output | Description |
|  |  |  |  |
|  |  |  |  |

.....

.

### Register Customer/Staff

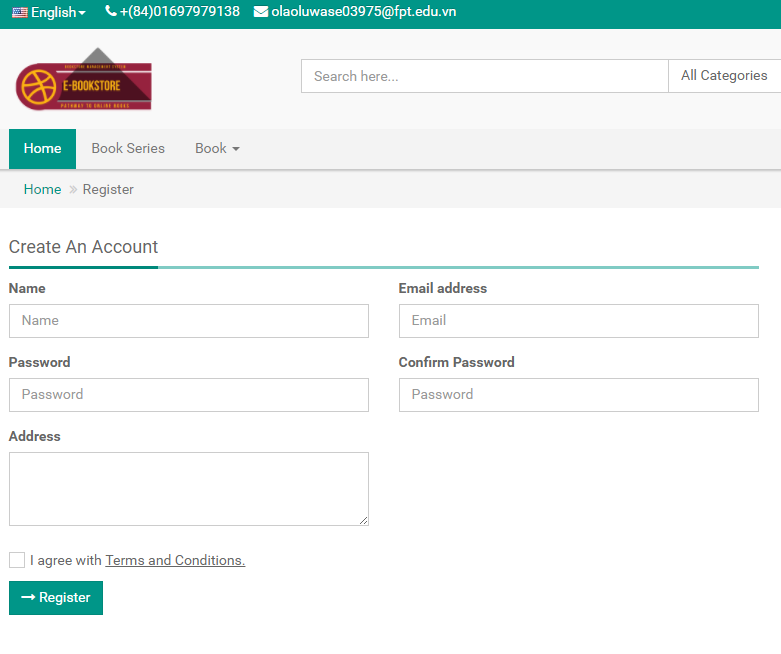
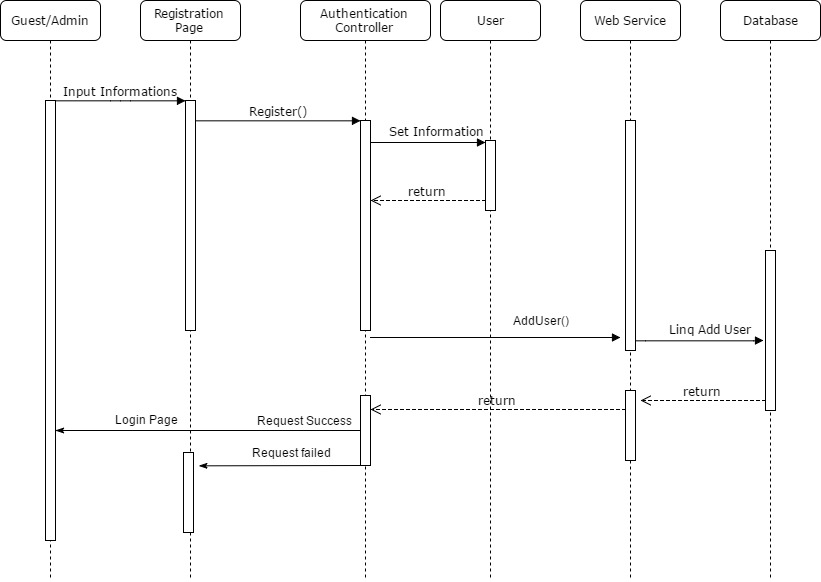


Figure 3.2.3

**Table 4-x:** Manage news- Delete news

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Object/Control Name** | **Type** | **Required** | **Length** | **Description** |
| 1 | Name | Text-Field | Yes | 3-20 | Customer Name |
| 2 | Email | Email-Field | Yes | 7-30 | Customer Email Address |
| 3 | Password | Password-Field | Yes | > 6 | User password |
| 4 | Confirm Password | Password-Field | Yes | > 6 | Password verification |
| 5 | Address | MultiLine-Field | No | Max 50 | Customers Address |
| 6 | cbAccept | CheckBox | N/A | N/A | Check i agree with terms and condition to activate Register button |
| 7 | Register | Button | N/A | N/A | Registration button |

### Sequence Diagram



#### <Use-case ID - Use case name>

...

#### <Use-case ID - Use case name>

................

# DATABASE DESIGN

## Entity Relationship Diagram

## Database Diagram

## Table Descriptions

### Table xxx

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Attribute | Type | Constraints | Description |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

### Table xxx

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Attribute | Type | Constraints | Description |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |
| 12 |  |  |  |  |
| 13 |  |  |  |  |
| 14 |  |  |  |  |
| 15 |  |  |  |  |