

***Spring Framework***

**Training Assignments**

|  |  |
| --- | --- |
| Document Code | 25e-BM/HR/HDCV/FSOFT |
| Version | 1.1 |
| Effective Date | 20/11/2012 |

**Hanoi, 06/2019**

RECORD OF CHANGES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Effective Date | Change Description | Reason | Reviewer | Approver |
|  | 01/Oct/2018 | Add the new labs | Create new | DieuNT1 | VinhNV |
|  | 01/Jun/2019 | Update template | Fsoft template | DieuNT1 | VinhNV |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Contents

[Objectives 4](#_Toc19437912)

[Business needs 4](#_Toc19437913)

[Working Environments 5](#_Toc19437914)

[Product Architecture 5](#_Toc19437915)

[Technologies 5](#_Toc19437916)

[Database Relationship 6](#_Toc19437917)

[Assignment Descriptions 6](#_Toc19437918)

|  |  |
| --- | --- |
|  | **CODE: JSFW.L.A103 (CMS)**  **TYPE: LONG**  **LOC: N/A**  **DURATION: 360 MINUTES** |

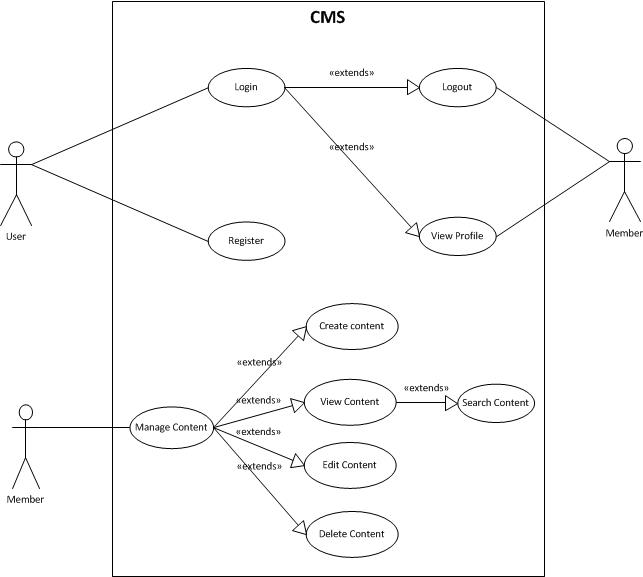
# **Objectives**

After finishing the following exercises, trainees will:

* Understand Spring Core (IoC, DI)
* Practice with JDBC Templates to connect database in spring
* Use JPA for object-relational mapping
* Practice with the Spring MVC framework architecture
* Know how to write a Web application with Spring MVC Framework and Spring Data JPA

# **Business needs**

* Create a Web application based on HTML, CSS, JavaScript, Ajax, Servlet, JSP, Spring MVC framework, Spring Data JPA to Manage Content System (CMS).
* Connect all the lessons into a complete whole so that students can gain an overall view of advanced Java.
* Provide adequate provisions for a special web system. In this assignment, we will learn about a content management system. The use case diagram is described as below:



A content management is a computer application that supports the creation and modification of digital content using a simple interface to abstract away low-level details unless required, usually supporting multiple users working in a collaborative environment.

CMS features vary widely. Most CMSes include Web-based publishing, format management, history editing and version control, indexing, search, and retrieval. By their nature, content management systems support the separation of content and presentation.

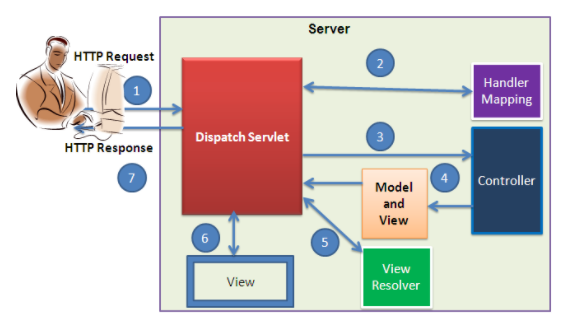
# Working Environments

* JDK 1.8 +
* Maven 3+
* Eclipse IDE or STS,
* MySQL DB Server/MySQL,
* Apache TomCat 7 or later,
* Internet connection,

Delivery: Source code, deployment and testing, reviewing evidences packaged in a compress archive.

# Product Architecture

Web applications are by nature distributed applications, meaning that they are programs that run on more than one computer and communicate through a network or server. Specifically, web applications are accessed with a web browser and are popular because of the ease of using the browser as a user client.



The product is implemented using MVC Pattern base on Spring Web MVC.

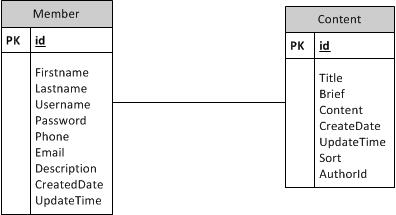
# Technologies

The product implements one or more technology:

* HTML & CSS
* JavaScript & Ajax
* MVC & JSP Model
* Spring Framework: IoC, Bean, SpEL, JdbcTemplate, Spring Data JPA, Interceptor, Validation, Spring Sercutity

# Database Relationship

Based on the analysis of the requirements, we decide to use the following database tables to store the persistent data for our CMS application:



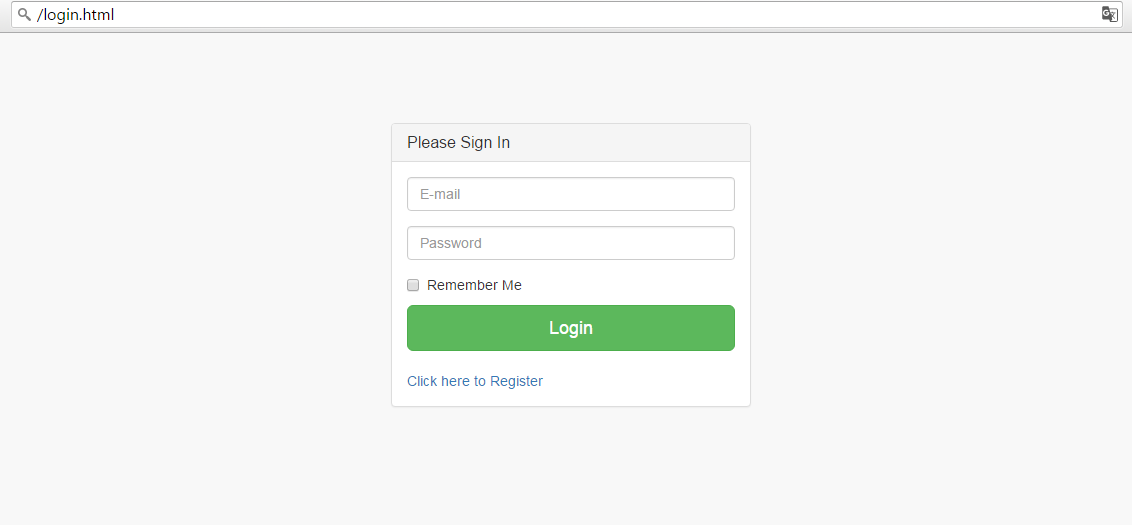
* ***Member*** stores the user information, including username and password.
* ***Content*** stores the post content information. Each comment is associated with a post and mainly consists of the following columns:
* authorID: required, the author id;
* Title: required, the title of the content;
* Brief: required, it is the description of content;
* Content: required, the comment content in plain text format.
* Sort: option, it use to other the post in the list.
* CreateDate, UpdateTime are date time create and update this post

# Assignment Descriptions

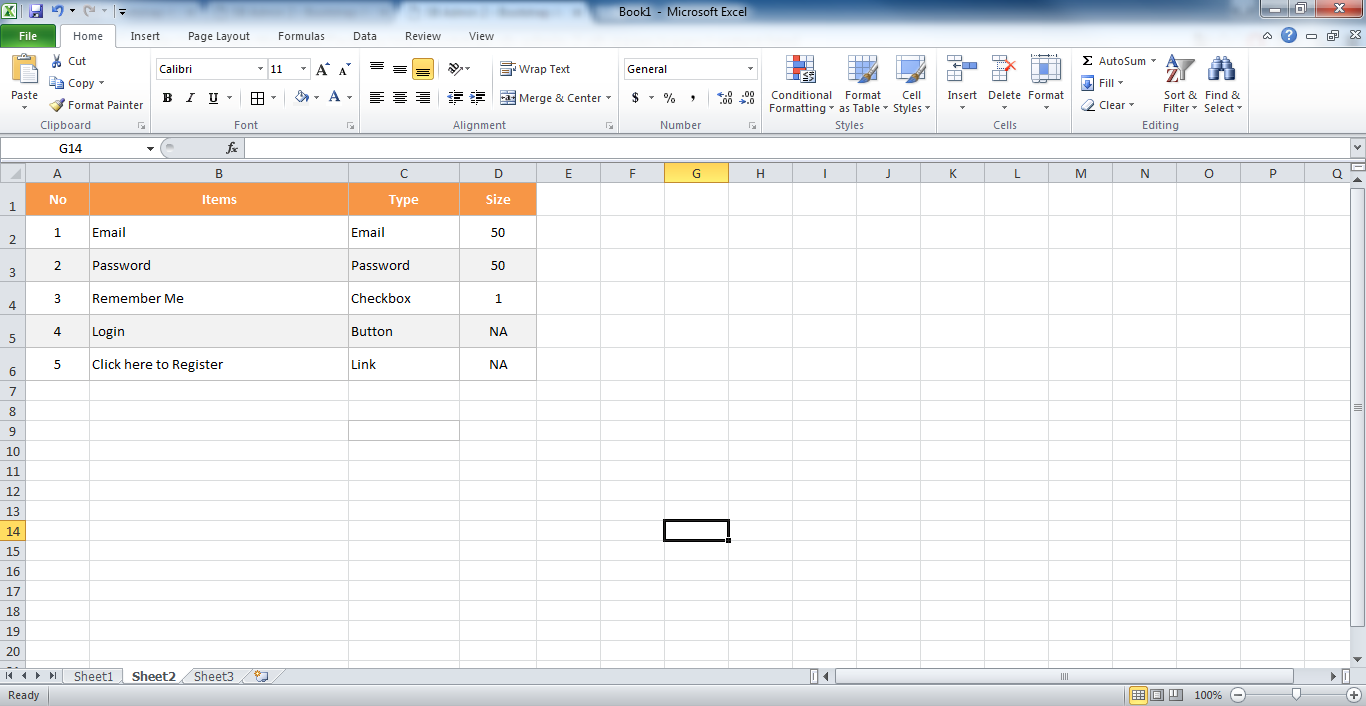
1. ***Create the pages as below:***

Start with this basic HTML template in attachment, and modify the pages as below:

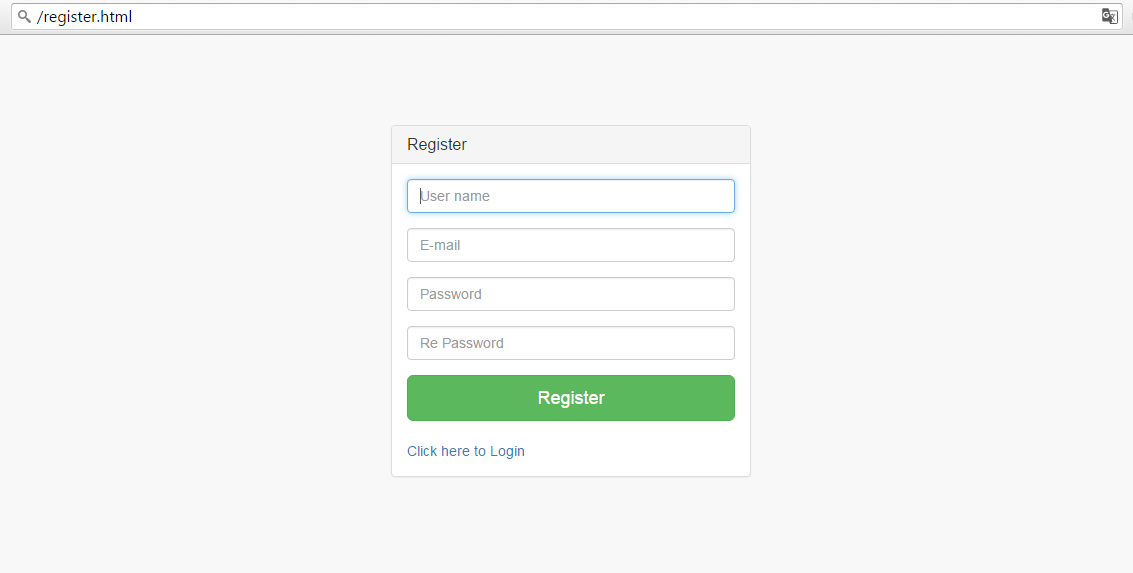
* Login page



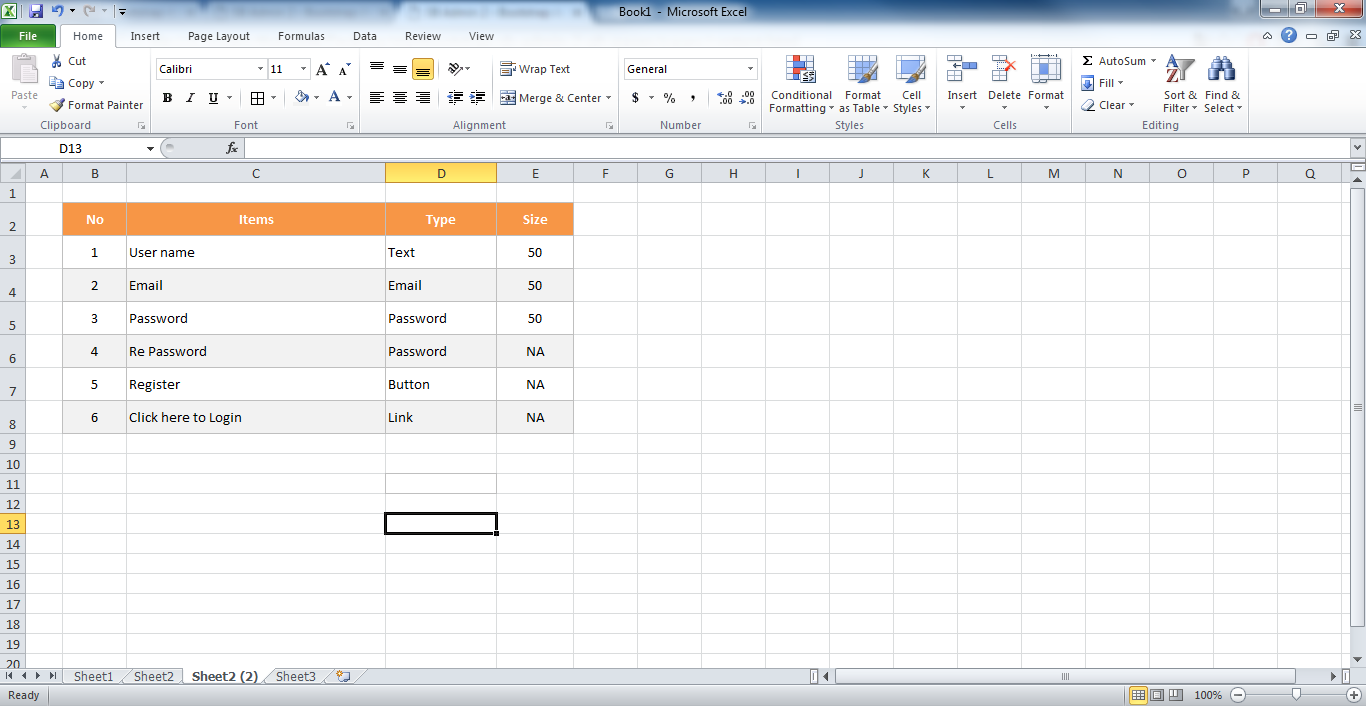
List item definition:



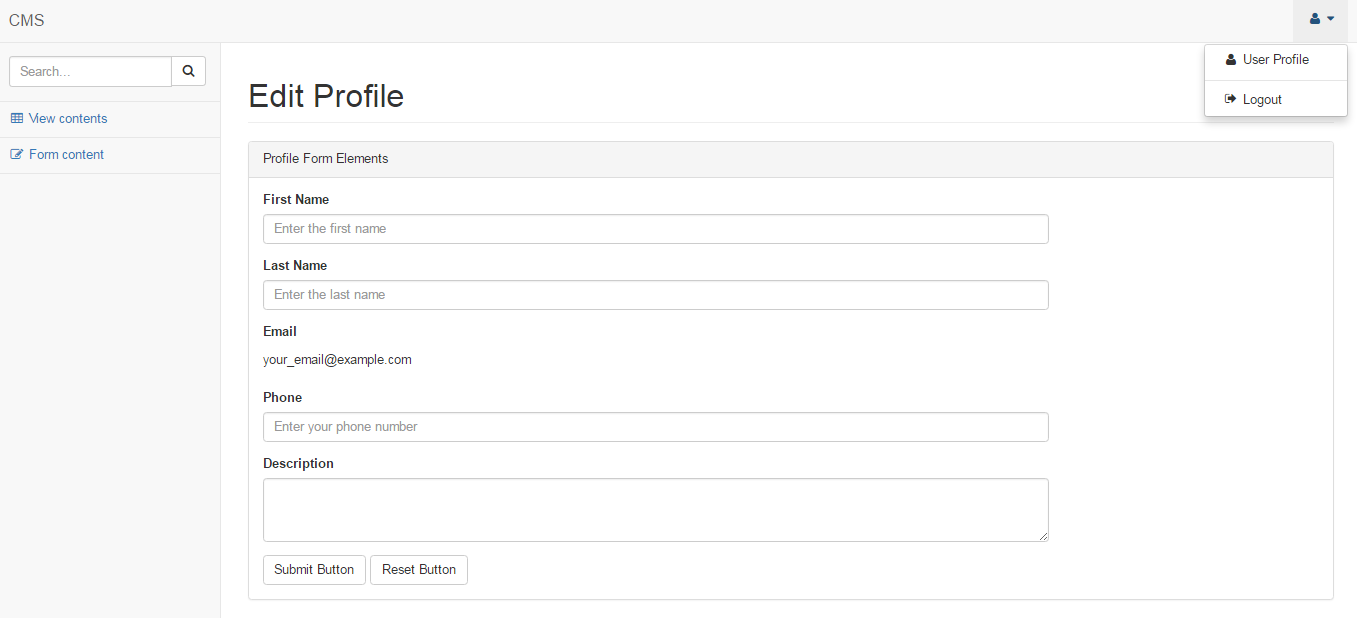
* Register page



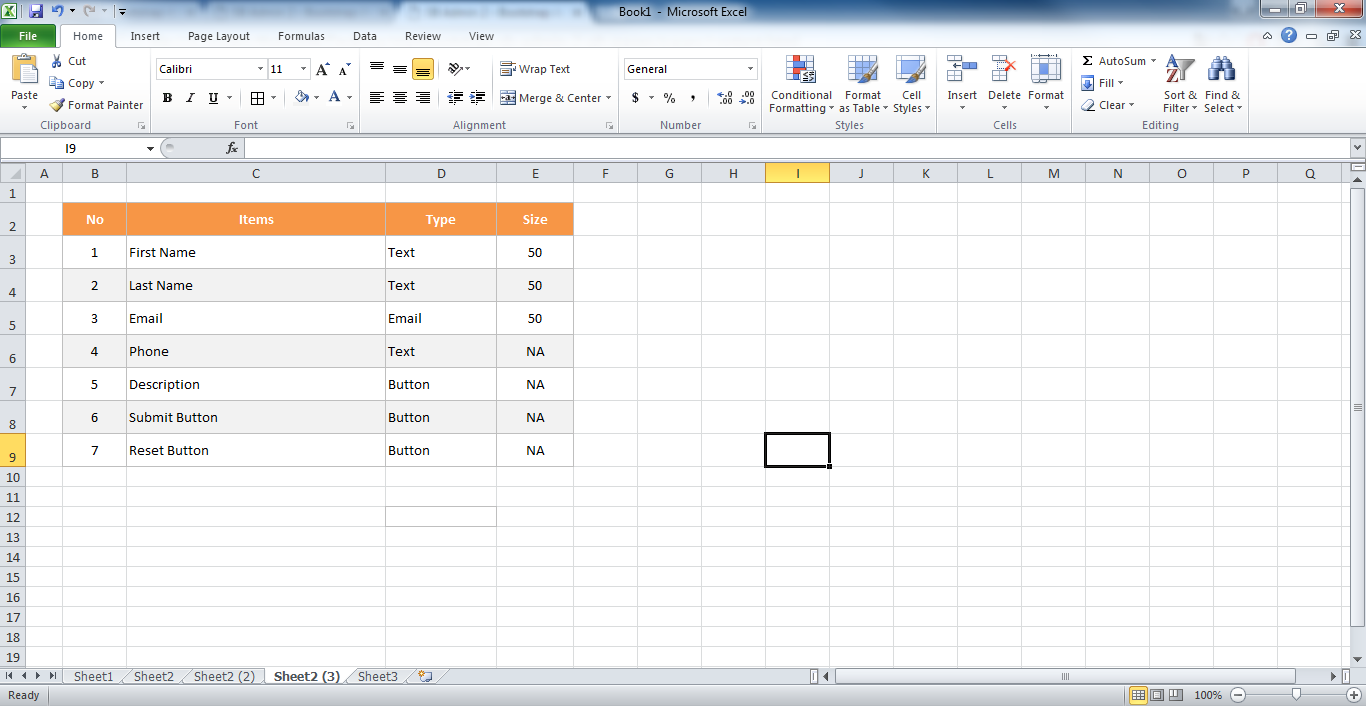
List item definition:



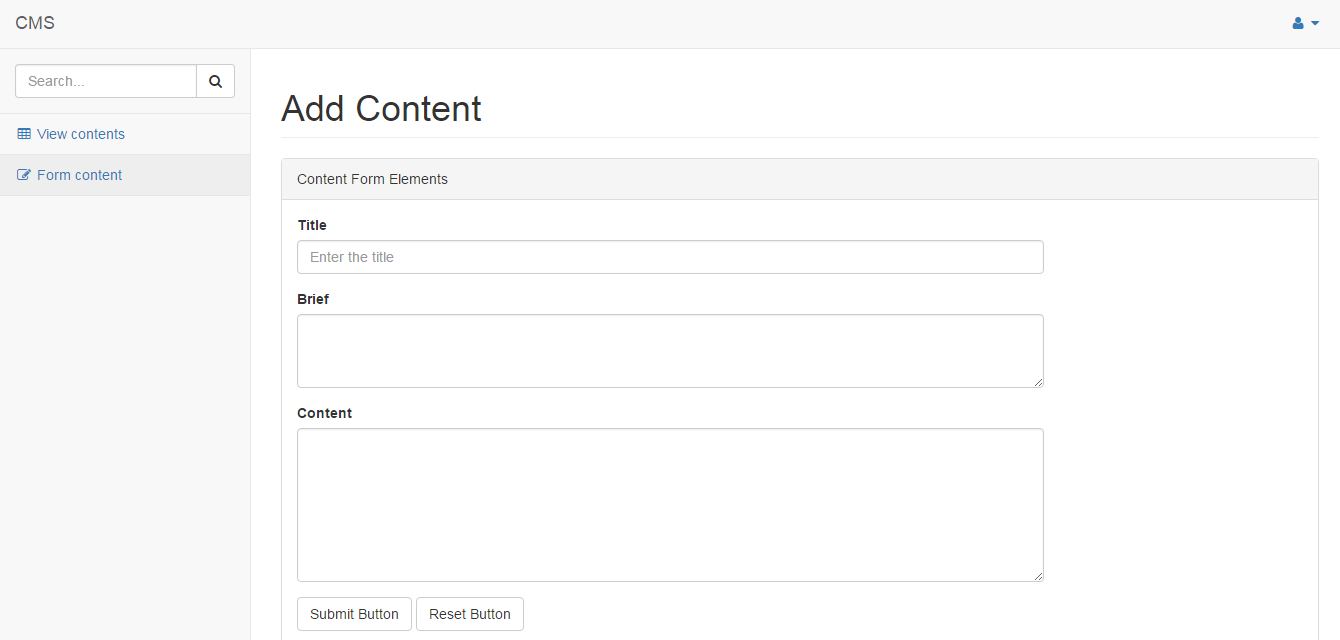
* Edit Profile



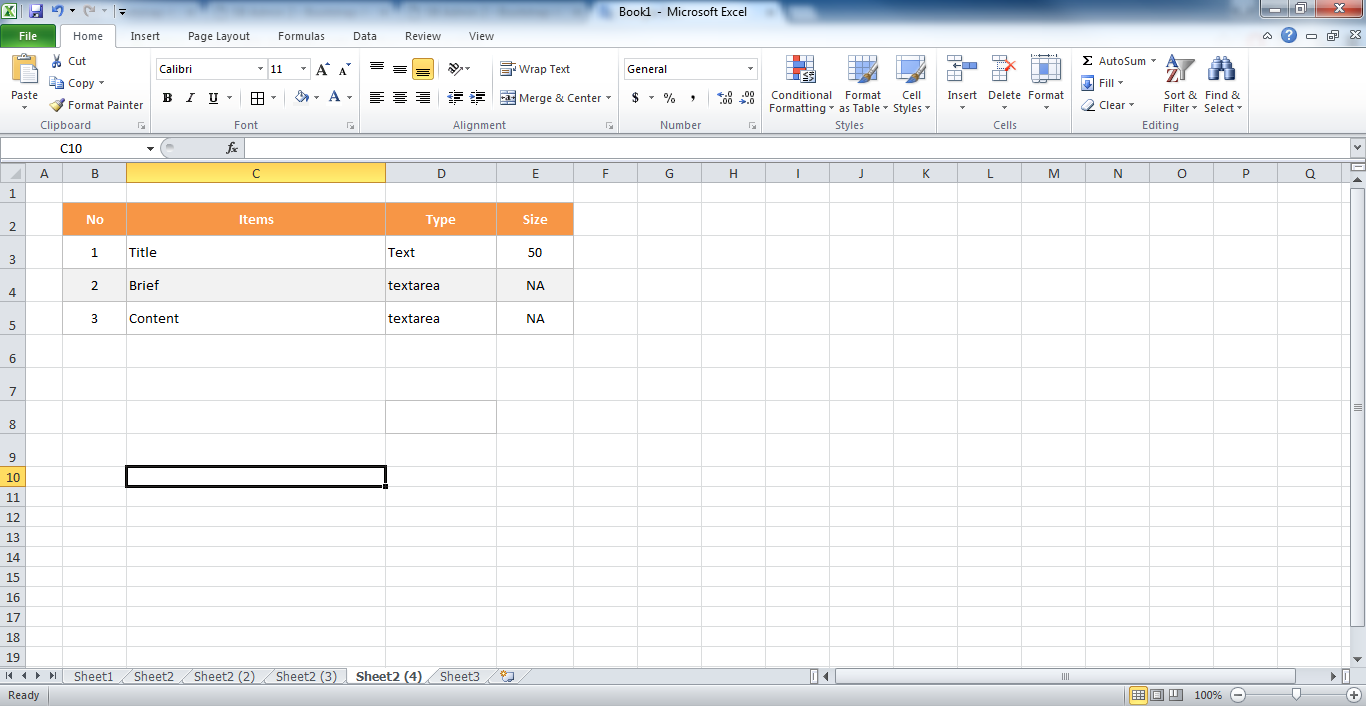
List item definition:



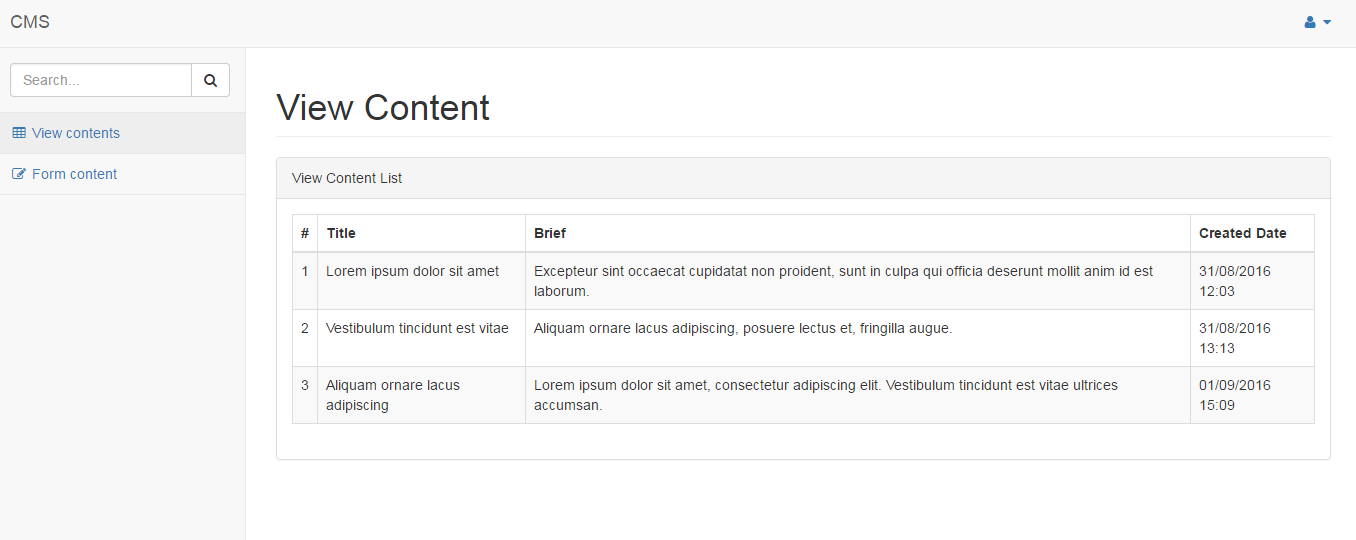
* Form content



List item definition:



* View contents

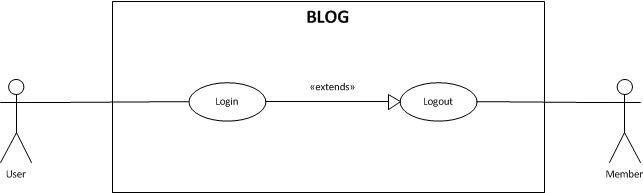


**Requirement:**

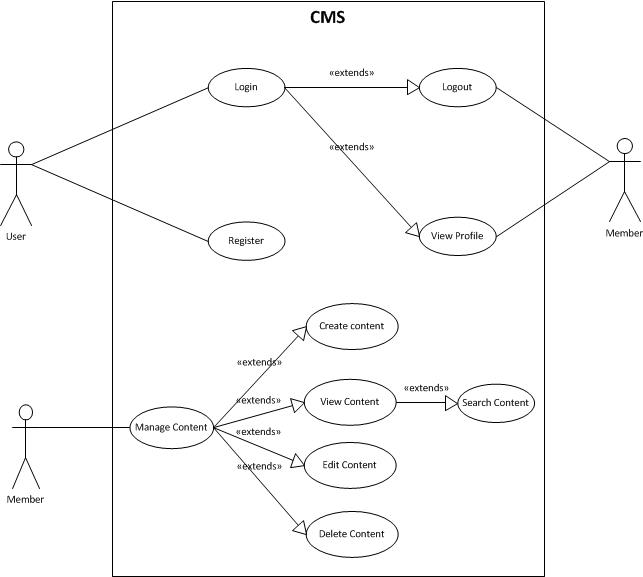
* All buttons have background in green color.
* Placeholder text in inputs has a light gray color. You need to define the placeholder for input items same as above screen.
* Add a new file fStyle.css to customize your screen, it will be based on convention of CSS style.

1. ***Functional Requirement:***

* As you may have found that the skeleton application already provides user authentication by checking if the username and password are both ***demo*** or ***admin***. In this section, we will modify the corresponding code so that the authentication is done against the User database table.



* Use Spring Framework framework connect to Database to finish all feature in bellow Use case:



**Requirement**

* Implement a program that manages Content and it will follow the above use case.
  + The columns “CreateDate” and “UpdateTime” are automatically increases the value by current date and time.
  + ID column is a primary key column
* Deploy your program on the Tomcat server.
* View Content and Search content page will be use ajax to load data from controller.
* User need use java code to validate data before insert, update user information to the database. So that, we were validated data on the client by Javascript and you will validate by Java on the server.

**---THE END---**