SRH Hochschule
Heidelberg

# Content Management Systems (CMS)

**11005867 – Jay Bhargav Barbhaiya**
**11005851 – Prasad Pomaji**
**11005595 – Shehzar Javed**

**Guides:**
**Prof. Christoph Hahn**
**Prof. Ajinkya Prabhune**

## Table of Contents

## Introduction:

A content management system (CMS) is a system used to manage the content of a Web site. It allows the content manager or author, who may not know Scripting languages like Hypertext Markup Language (HTML), Cascading Style Sheets(CSS), PHP, etc. to manage the content for a Website without needing the expertise of a Programmer. The features of a CMS system vary, but most include Web-based publishing and format management. [1]

Without CMS, the organization will have a person (or group of people) responsible for writing the content, and another person or group (usually IT or web developer) responsible for implementing the changes. This process tends to be very inefficient and takes more time and costs more money than is necessary. In addition, as more and more pages are added to the website, it often becomes necessary to create new layouts and arrangements to accommodate this new content. Changes to the layout and appearance of the website often mean that a programmer will need to be involved in the creation of these new pages and layouts. In the end, the website is comprised of numerous page templates, countless pages of content, and multiple content authors with no real system to manage everything.

1. A CMS allows content to be controlled by the people who own the content (the content or subject matter experts). This means no more relying on developers to make changes to the web content; subject matter experts can add content without needing to know HTML or any code. This saves money and time.
2. Content can be added to the site much more quickly and efficiently. Nothing is worse than time-sensitive information not getting posted in time because the request to update the site is stuck in a development queue.
3. Keeping content controlled by the content authors frees up the developers' time to focus on other things like the design of the front end of the website or implementation of new features and functionality.
4. Content can be previewed from within the software.
5. By putting style rules in place, you can make sure that things such as font sizes and colors are used universally, without regard to how many content authors you have. This keeps things looking professional and helps maintain brand unity.
6. A CMS provides a wide range of functionality, functionality that would normally cost money to build from scratch.
7. Most CMS software is web based, which means that you can access and edit your site from any computer with an internet connection. No specific software needs to be installed.
8. Many enterprise CMS platforms can securely protect your data and information about your users or members.[2]

## Project Description:

Our Content Management System performs the following operations,

1. Creating
2. Reading
3. Updating
4. Deleting, content displayed on a web site's web pages.

The HTML is assembled and served to the end user only when requested. Dynamic web pages are generated by using SQL Server and ASP.NET MVC5 framework for managing and displaying the HTML content items.

The proposed CMS is an ASP.NET-based system for creating dynamic web pages. Its flexible and extensible structure allows functionality to be added using options for changing the content.[3]

**CRUD Operations:**

Create, retrieve, update and delete (CRUD) refers to the four major functions implemented in our application.

The CRUD functions are the user interfaces to databases, as they permit users to create, view, modify and alter data of the database.

The CRUD operations executed in this particular project are executed using LINQ. Language-Integrated Query (LINQ) is a set of features introduced in Visual Studio 2008 that extends powerful query capabilities to the language syntax of C# and Visual Basic. LINQ introduces standard, easily-learned patterns for querying and updating data, and the technology can be extended to support potentially any kind of data store. Visual Studio includes LINQ provider assemblies that enable the use of LINQ with .NET Framework collections, SQL Server databases, ADO.NET Datasets, and XML documents.[4]

The CMS is developed to allow an organization to have complete control over its website without the need to pay excessive and recurring development fees for almost daily, routine website updates.

Our intended audience is any person who would like to build his/her own website for,

E-commerce.
Blogs.
News Site.
Simple site that delivers content to the users.
A general website for talking about a service or business.

## Project Architecture:

### Model–view–controller (MVC)

**Model–view–controller** (**MVC**) is a software architectural pattern for implementing user interfaces. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. The central component, the *model*, consists of application data, business rules, logic and functions. A *view* can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants. The third part, the *controller*, accepts input and converts it to commands for the model or view.

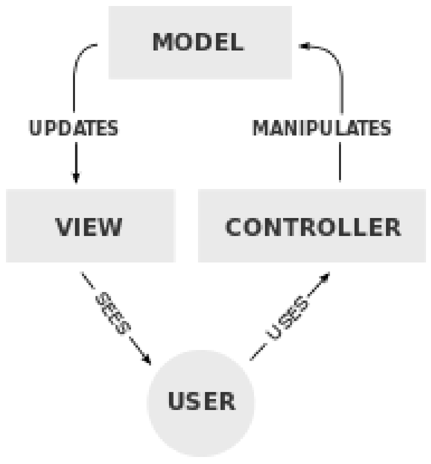


Fig. MVC Architecture [5]

In addition to dividing the application into three kinds of components, the model–view–controller design defines the interactions between them.

- A **controller** can send commands to the model to update the model's state (e.g., editing a document). It can also send commands to its associated view to change the view's presentation of the model (e.g., by scrolling through a document).
- A **model** notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. In some cases an MVC implementation might instead be "passive," so that other components must poll the model for updates rather than being notified.
- A **view** requests information from the model that it needs for generating an output representation to the user.[6]

SRH Hochschule
Heidelberg

## Project Implementation:

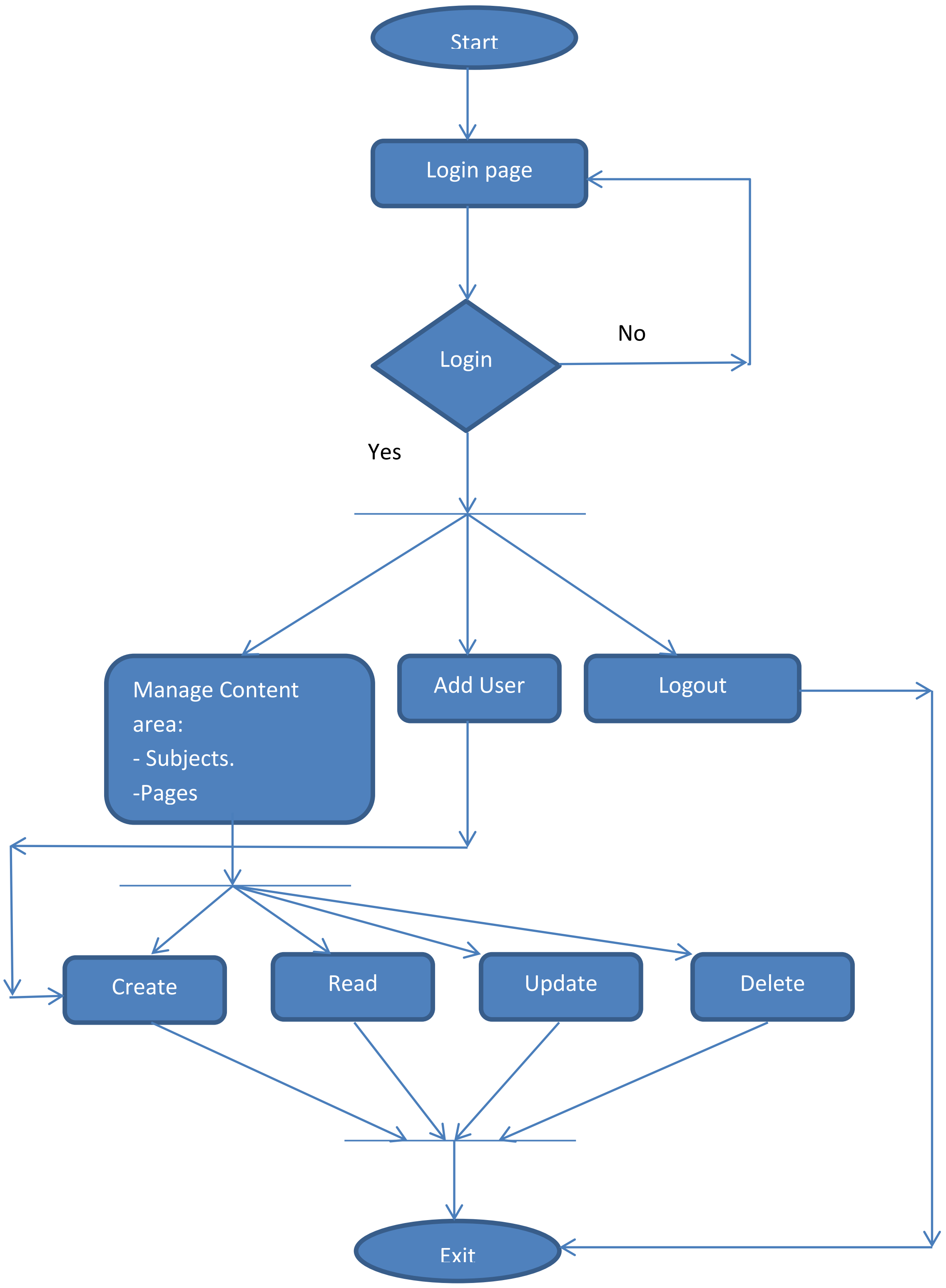


Fig. Flowchart for Administrator panel page

The Administrator panel operates in the following manner:

1. The user has to first successfully login or else the user is brought back to the login page.
2. Once the user has successfully logged in, access to the administrator panel is granted.
3. In the administrator panel the user can do the following:
    a. Manage the content for the website.
    b. Logout.
4. In manage content option, the user can create subjects and add pages to the subject.
5. In the Register option, the user can create username and password for other users.
6. The logout option expires the user's session and redirect to the login page.

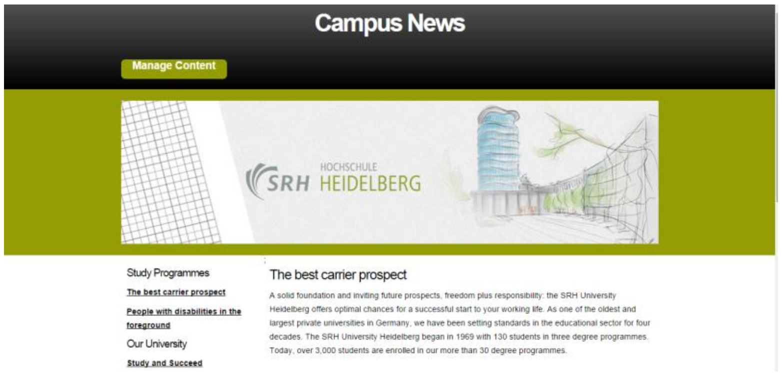
The above mentioned operation in manage content area is supported with CRUD operations using LINQ.

Keeping the MVC pattern in mind the implementation was divided into 3 steps:

1. **Model:** The model class forms the basis for the database schema.
2. **View:** The GUI developed using HTML5 and CSS3 forms the View for the application and creating the view was the second step for building the application.
3. **Controller:** ASP.NET plays the role of controller. Depending on the user's actions received through the browser, C# interacts with the Model and returns the data back to the browser for the user to view.

Another aspect kept in mind during the implementation was to **maximize the reusability of the code**. This was important because the CMS contains of two parts:

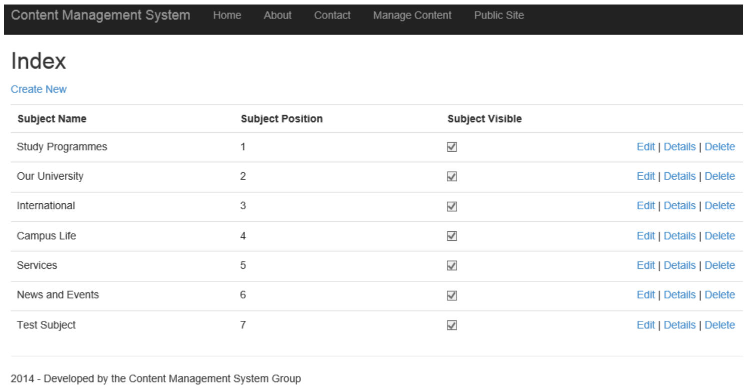
1. Website
2. Administrator Panel



The best carrier prospect

A solid foundation and inviting future prospects, freedom plus responsibility: the SRH University Heidelberg offers optimal chances for a successful start to your working life. As one of the oldest and largest private universities in Germany, we have been setting standards in the educational sector for four decades. The SRH University Heidelberg began in 1969 with 130 students in three degree programmes. Today, over 3,000 students are enrolled in our more than 30 degree programmes.

Fig. Website



Index

Create New

| Subject Name | Subject Position | Subject Visible | |
|---|---|---|---|
| Study Programmes | 1 | ✓ | Edit \| Details \| Delete |
| Our University | 2 | ✓ | Edit \| Details \| Delete |
| International | 3 | ✓ | Edit \| Details \| Delete |
| Campus Life | 4 | ✓ | Edit \| Details \| Delete |
| Services | 5 | ✓ | Edit \| Details \| Delete |
| News and Events | 6 | ✓ | Edit \| Details \| Delete |
| Test Subject | 7 | ✓ | Edit \| Details \| Delete |

2014 - Developed by the Content Management System Group

Fig. Administration Panel

## Example code snippets:

Some of the example  code snippets which are crucial in the project are given below:

```
public static void RegisterRoutes(RouteCollection routes) {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}")
                routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Public", action = "Index", id =
                UrlParameter.Optional }
            );
        }
```

The above code snippet creates a routing to navigate to a particular page. If the URL exists then user will be granted access to the pages in the application.

## Contribution

**Jay Bhargav Barbhaiya**

- Database configuration with SQL Server LocalDB.
- Project setup for CMS.
- Webpage to view the subjects and Pages.
- Admin Panel design modifications.
- GIT repository creation for project.
- Controller class for Subject, Page and Public.

**Shehzar Javed**

- Trello Board creation and task assignment.
- Implementation of security.
- Web Page design for the Public Area.
- Documentation.

**Prasad Pomaji**

- Model class for Subjects, Pages and Public.
- Database and tables creation using Entity Framework.
- Scaffolding of Models with their respective Views.
- Login and Logout functionality.

**Steps to implement the CMS:**

Requirements:

1. Install Visual Studio 2013 Express for Web.
2. SQL Server is in-build in the above mentioned installation package, it is referred as LocalDb.
3. The WebConfig file in the root of the directory should contain the connection string to the database as follows:

```
<connectionStrings>
         <add name="DefaultConnection"
              connectionString="Data
            Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|\CMSDatabase.
            mdf;Initial Catalog=aspnet-CMS-20140921115440;Integrated
            Security=True"
            providerName="System.Data.SqlClient"
         />
</connectionStrings>
```

4. A folder named APP_DATA contains the database files which can be include into project.
5. The Routes that the application should follow are specified in the RouterConfig.cs file.
6. Whenever a controller is created, the corresponding View file inside 'View' folder is also created. The Models lie in a separate folder. This helps in code reusability as well as separation of code along with systematic application development.
7. Access to Mange Content of the CMS can be done by registering a new user.

**Git repository link:**

https://github.com/jaybarbhaiya/simple_cms.git

Install Visual Studio 2013 Express for Web and build and run the project after cloning the git repository.

## Conclusion:

The proposed CMS is enabled for serving the user with creation, editing and deletion of the website content. Also security of the user's content is given priority and thus direct access to the pages in the administration panel is restricted unless the user logs in.

The system can be expended to support images and media files. Also make the CMS to be flexible to the user with respect to template selection. The user would be able to select as well as change the template according to his/her choice. This template design is reflected on the main website. Also create logs to monitor the user activities. This feature will be useful in order to manage which user has created, deleted or updated the content.

## References:

[1] - http://searchsoa.techtarget.com/definition/content-management-system
[2] - http://www.wsol.com/why-choose-a-cms/
Fig. MVC Architecture - http://upload.wikimedia.org/wikipedia/commons/thumb/a/a0/MVC-Process.svg/200px-MVC-Process.svg.png
[3] - http://docs.joomla.org/Content_Management_System
[4] - http://msdn.microsoft.com/en-us/library/bb397926.aspx
[5]- http://upload.wikimedia.org/wikipedia/commons/thumb/a/a0/MVC-Process.svg/200px-MVC-Process.svg.png
[6] - http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

Template used for website - http://www.opendesigns.org/design/html5-streets/