Adventure Works Photo Sharing Application (Proposed)

Detailed Planning Document

*09/03/2019*

Authors: *Jingjing Dong*

# Introduction

The author has examined the initial investigation document by Hines, Raghav, and Khan. Based on the use cases, technical requirements, and other content in that document, the author has created the detailed plans below. The board has already agreed that the photo sharing application will be built as a website based on Microsoft’s ASP.NET MVC technology. Therefore the details presented here include the names and properties of model classes and controllers developers must create. Views have also been identified and wireframe diagrams included to help envision the user interface for important parts of the site.

The application design is likely to evolve throughout the development process as requirements change. The development team will adopt Agile practices to ensure such changes are reflected in the final product. Therefore this document should not be considered a complete definition of the final application.

# MVC Model

Developers will create a model with the following model classes. For each model class, properties have been listed and descriptions given.

Table 1: MVC Model

|  |  |  |  |
| --- | --- | --- | --- |
| Model Class | Description | Properties | Data Types |
| Photo | The photo model class represents a photo which authenticated users can upload and delete to the website. | PhotoID (PK) | Integer |
| UserID (FK) | Integer |
| Description | String |
| CreatedDate | Date |
| Title | String |
| PhotoFile | Binary |
| Comment | The comment model class represents a comment that authenticated users can add to photos which enables users to discuss photos with others. Each comment is associated with only one photo and one user. | CommentID (PK) | Integer |
| UserID (FK) | Integer |
| PhotoID (FK) | Integer |
| Subject | String |
| Body | String |
| User | This user model class represents an authenticated user who can register in the website in order to logon/logoff. | UserID (PK) | Integer |
| Username | String |
| Password | String |

# MVC Controllers

Developers will create the following controllers. For each controller, actions have been listed and descriptions given.

Table 2: MVC Controllers

|  |  |  |
| --- | --- | --- |
| Controller | Action | Description |
| Photo | AddPhoto (GET) | Creates a new instance of the Photo model class, sets default values such as the created date, and passes it to the correct view |
| AddPhoto (POST) | Calls the Photo model class methods to save the photo values to the database and redirects the browser to the DisplayAll view |
| DisplayPhoto (GET) | Requests display a detailed view of selected photo |
| DisplayAll (GET) | Requests displays all the photos stored in the database |
| DeletePhoto (GET) | After the user clicks “Delete this photo” link in the DisplayPhoto view, this action displays the DeletePhoto view which requests confirmation for deletion. |
| DeletePhoto (POST) | This action deletes the current Photo along with all associated comments from the database after user clicks “Delete” in the DeletePhoto view and redirects the user to the DisplayAll view. |
| Comment | AddComment (GET) | Creates a new instance of the Comment model class, sets default values such as PhotoID it comments to, and passes it to the correct view |
| AddComment (POST) | Calls the Comment model class methods to save the comment values such as subject and body to the database and redirects the browser to the DisplayPhoto view |
| User | Logon (GET) | Displays a view that an anonymous user can enter credentials into |
| Logon (POST) | Checks user credentials against the membership database. If the credentials are correct, the logon action authenticates and redirects the user to the originally requested page. |
| Logoff (POST) | Logs the authenticated user off the website, displays message ‘You’re successfully logged off’, and redirects user to Logon view |

# MVC Views

Developers will create the following views. Each view has been listed together with the controller it is associated with.

Table 3: MVC Views

|  |  |  |
| --- | --- | --- |
| Controller | View | Description |
|  |  |  |
|  |  |  |
|  |  |  |

# Hosting Recommendations

Since the photo sharing application will be developed in ASP.NET MVC, it must be hosted on a Microsoft web server. The author recommends the following hosting configuration:

## Web Server

*Insert web server recommendations here.*

## Database

*Insert database server recommendations here.*