Adventure Works Photo Sharing Application (Proposed)

Detailed Planning Document

*Insert Date Here*

Authors: *Insert Your Name Here*

# Introduction

The author has examined the initial investigation document. 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 |
| 1. | Photo | The Photo model class represents a photo that authenticated users can upload to the website | PhotoID | Integer |
| 2. | Title | String |
| 3. | PhotoFile | Binary |
| 4. | Description | String |
| 5. | CreatedDate | DateTime |
| 6. | Owner | Integer |
| 7. | Comment  The comment model class represents a comment that authenticated users can add to photos. This enables users to discuss others’ photos. Tach comment is associated with just one photo. | | CommentID | Integer |
| 8. | Subject | String |
| 9. | Body | String |
| 10. | PhotoID | Integer |
| 11. |  |  |  |  |

# 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 |
| 1. | PhotoController | DisplayGallery(Get) | The action runs when the user requests the PhotoGallery page. The action obtains all the photos from the database and passes them to the DisplayGallerly View. |
| 2. |  | DisplayRecent(GET) | The action is similar to the DisplayGallery action except that only the most recent photos are obtained from the database. This is smaller collection of photos is passed to the DisplayGallery View |
| 3. |  | DisplayPhoto |  |
| 4. |  | AddPhoto(Get) |  |
| 5. |  | DeletePhtoto(Get) |  |
| 6. |  | DeletePhtoto(Post) |  |
| 7. |  | AddPhoto(post) |  |
| 8. |  |  |  |
| 9. |  |  |  |
| 10. |  |  |  |

# 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 |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
| 4. |  |  |  |
| 5. |  |  |  |
| 6. |  |  |  |

# Hosting Recommendations

Since the photo sharing application will be developed in ASP.NET Core 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.*