



Module02

Designing ASP.NET MVC 4 Web Applications

- Planning in the Project Design Phase
- Designing Models, Controllers, and Views

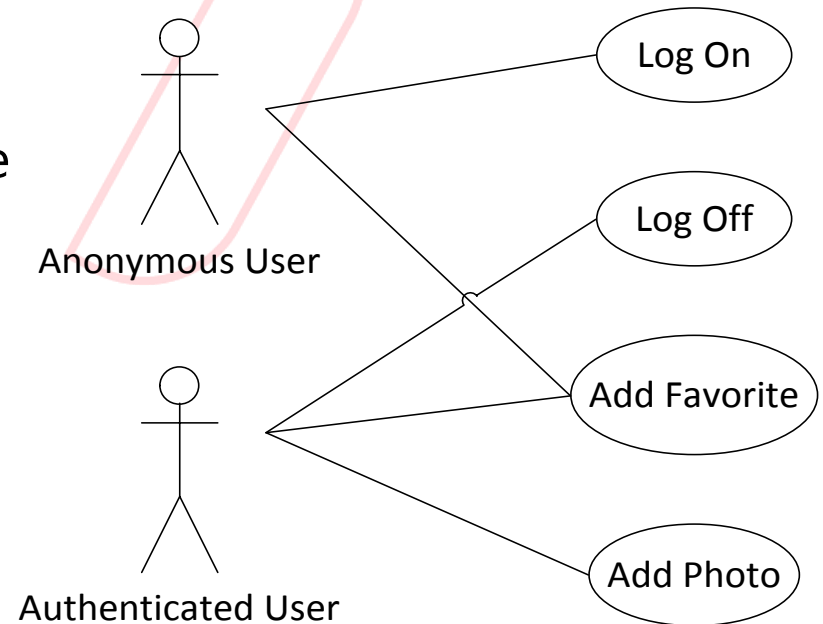
Lesson 1: Planning in the Project Design Phase

- Project Development Methodologies
- Gathering Requirements
- Planning the Database Design
- Planning for Distributed Applications
- Planning State Management
- Planning Globalization and Localization

Development Model	Description
Waterfall Model	Activities for building an application are performed sequentially in distinct phases with clear deliverables.
Iterative Development Model	An application is built iteratively in parts, by using working versions that are thoroughly tested, until it is finalized.
Prototype Model	Based on a few business requirements, a prototype is made. Feedback on the prototype is used as input to develop the final application.
Agile Development Model	An application is built in rapid cycles, integrating changing circumstances and requirements in the development process.
Extreme Programming	Begins with solving a few critical tasks. Developers test the simplified solution and obtain feedback from stakeholders to derive the detailed requirements, which evolve over the project life cycle.
Test-Driven Development	A test project is created and you can test changes to the code singly or as a group, throughout the project.
Unified Modeling Language	UML diagrams are used for planning and documenting purposes, across all project development models.

Gathering Requirements

- Functional requirements describe how the application responds to users
- Technical requirements describe the technical features of an application, such as availability, security, or performance
- You can build functional requirements by using:
 - Usage scenarios
 - Use cases
 - Requirements modeling in agile
 - User stories in extreme programming



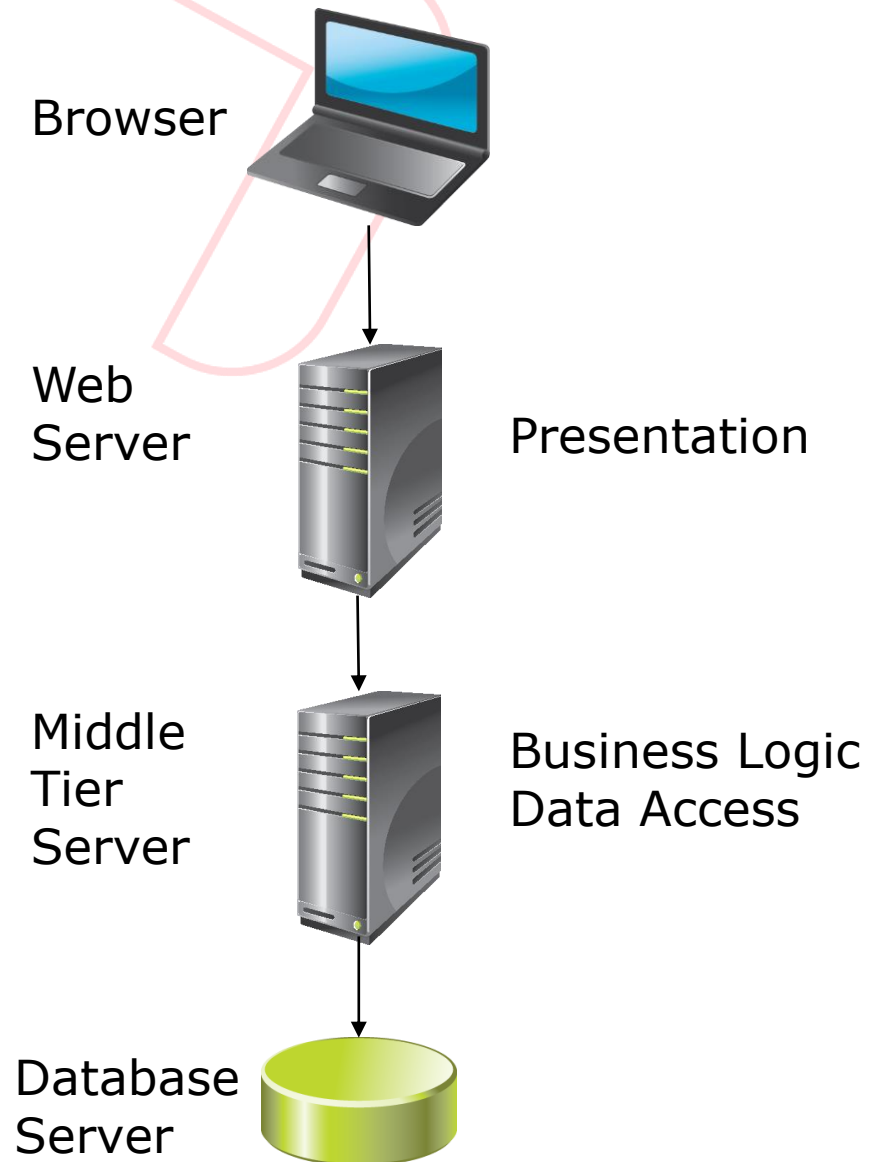
Sample UML Use Case Diagram

Planning the Database Design

- Logical Modeling
- Physical Database Structure
- Working with DBAs
- Database Design in Agile and Extreme Programming

Planning for Distributed Applications

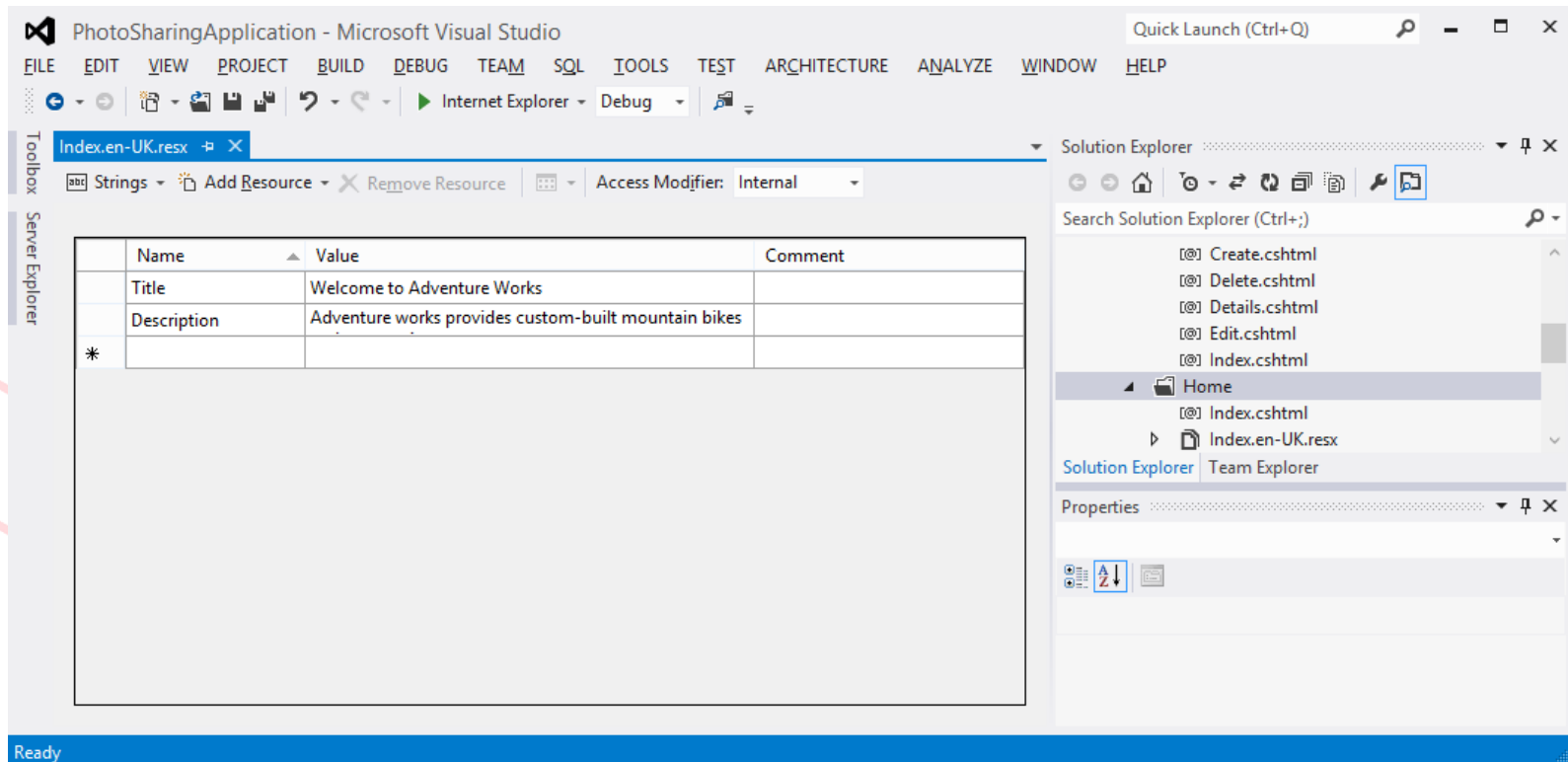
- Layers
 - Presentation
 - Business Logic
 - Data Access
 - Database
- Communication
- Security



- Client-side locations to store state data:
 - Cookies
 - Query Strings
- Server-side locations to store state data:
 - TempData
 - Application State
 - Session State
 - Profile Properties
 - Database Tables

Planning Globalization and Localization

- You can use the internationally-recognized set of language codes available in browsers to present content customized to suit a user's language or region
- You can use resource files to provide a localized response suitable to a user's culture
- You can use separate views to suit each language code



Lesson 2: Designing Models, Controllers, and Views

- Designing Models
- Designing Controllers
- Designing Views

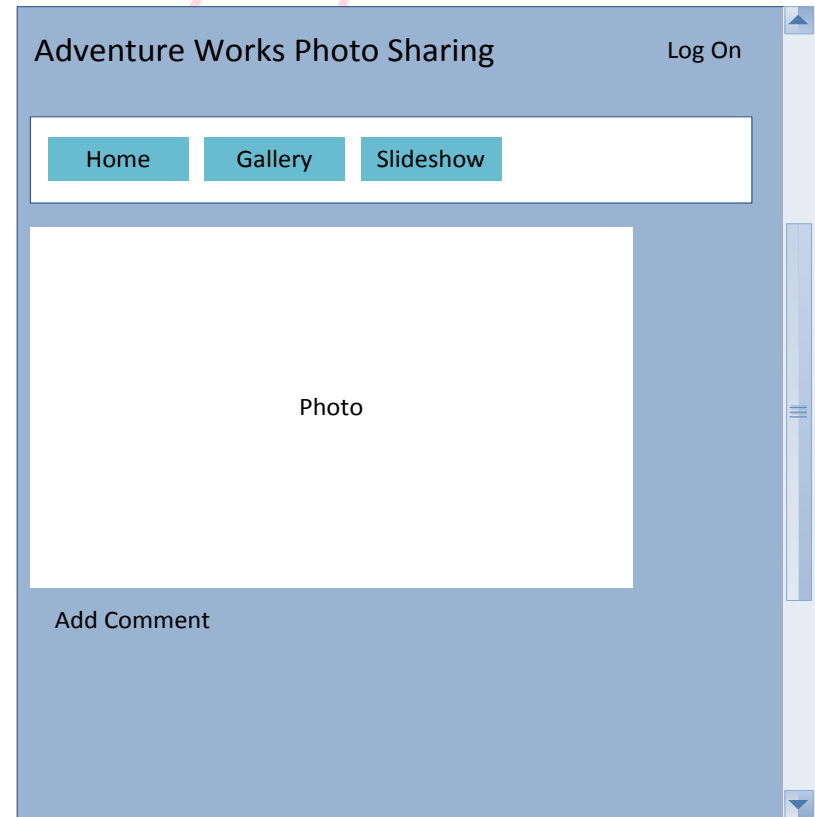
- Model Classes and Properties
- Domain Model and Logical Data Model Diagrams
- Relationships and Aggregates
- Entity Framework
- Design in Agile and Extreme Programming

Controller	Action
Photo	AddPhoto (GET)
	AddPhoto (POST)
	DisplayGallery (GET)
User	Logon (GET)
	Logon (POST)

- Identify Controllers and Actions
- Design in Agile and Extreme Programming

Designing Views

- Views, Templates, and Partial Views
- Wire-Framing
- Design in Agile and Extreme-Programming



Lab: Designing ASP.NET MVC 4 Web Applications

- Exercise 1: Planning Model Classes
- Exercise 2: Planning Controllers
- Exercise 3: Planning Views
- Exercise 4: Architecting an MVC Web Application

Estimated Time: 40 minutes

Your team has chosen ASP.NET MVC 4 as the most appropriate ASP.NET programming model to create the photo sharing application for the Adventure Works web application. You need to create a detailed project design for the application, and have been given a set of functional and technical requirements with other information. You have to plan:

- An MVC model that you can use to implement the desired functionality.
- One or more controllers and controller actions that respond to users actions.
- A set of views to implement the user interface.
- The locations for hosting and data storage.

- What model classes should be created for the photo sharing application based on the initial investigation?
- What controllers should be created for the photo sharing application based on the initial investigation?
- What views should be created for the photo sharing application?

Module Review and Takeaways

- Review Question(s)
- Real-world Issues and Scenarios
- Tools
- Best Practice
- Common Issues and Troubleshooting Tips