Unit 4: Design of Web Applications (2/2)

Internal Design of WebApps

Unit 4: Design of Web Applications

- Web Application Engineering
- Web Application Analysis
- Web Application Design
  - External Design
    - The UX Model
    - UX Modeling by Example
  - Internal Design
    - The Web Application Extension for UML (WAE2)
    - The Model-View-Controller (MVC) Architectural Pattern

The Web Application Extension for UML (WAE2)

- The Web Application Extension (WAE2) for UML enables us to represent Web pages and other architecturally significant elements in the internal design model of 'Classic' Web Applications
- In 'Classic' Web Applications, *Web Page* may refer to:
  - Either a program/script/template executed on the server, which interact with other server-side resources before being sent to the browser as a completed (HTML) Web page.
  - Or the (generated HTML) Web Page that the browser gets from the server, which may include some (JavaScript) logic to be executed on the browser.

UML Mechanisms To Extend UML

- Stereotype: is an extension to the vocabulary of the language that allows to attach a new semantic meaning to a UML model element (a class, an association, etc).
- Tagged value: is the definition of a new property that can be associated with a model element.
  - UML Classes, for instance, have names, visibility, persistence, and other properties associated with them.
- Constraint: specifies the conditions under which the model can be considered well formed.

WAE2 Stereotypes

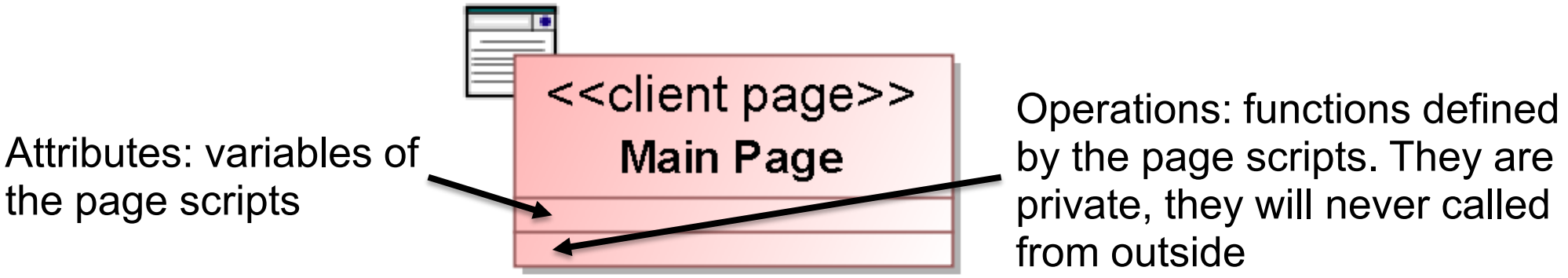
Main **Class Stereotypes**:

- Server Page
- Client Page
- Form

Main **Association Stereotypes**:

- Link
- Build
- Submit
- Redirect

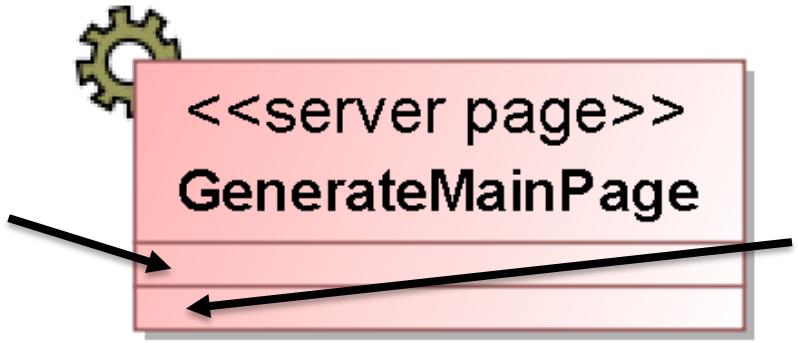
WAE2 Client Page



- A client page instance is an HTML-formatted Web page with a mix of data, presentation, and even logic.
- Constraints: none
- Tagged values:
  - TitleTag, the title of the page as displayed by the browser.
  - BaseTag, the base URL for dereferencing relative URLs.
  - BodyTag, the set of attributes for the <body> tag, which sets background and default text attributes.

WAE2 Server Page

Attributes: global variables accessible by all the operations

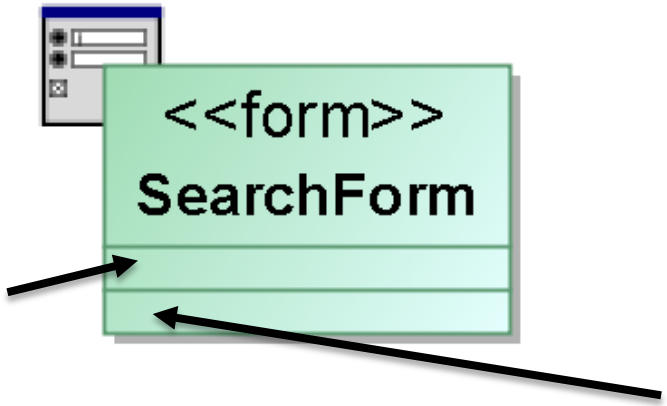


Operations: functions that interact with databases, business components, etc. and/or build dynamic content. These operations are private: they will never be called from outside.

- A server page represents a dynamic Web page that contains content assembled on the server each time it is requested. Later it can be implemented as a Servlet, JSP, ASP, or PHP page
- Constraints: Server pages can have only “normal” relationships with objects on the server
- Tagged values: none

WAE2 Form

Attributes: represent the HTML form's input fields: input boxes, text areas, radio buttons, check boxes, hidden fields, etc.



Operations: none

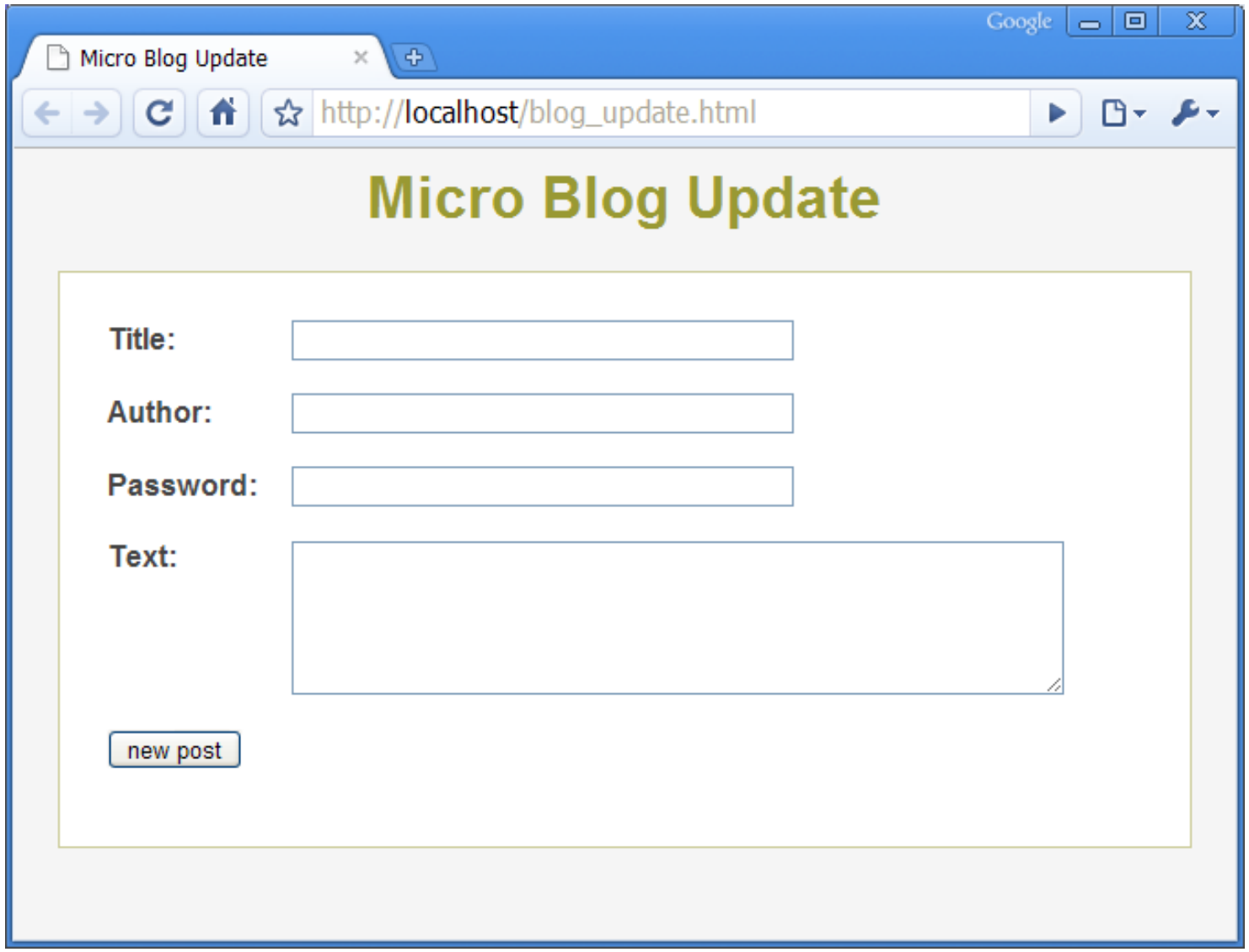
- A form instance represents a HTML form in a client page.
- Constraints: none.
- Tagged values:
  - Either GET or POST: the method used to submit data to the action URL.

WAE2 Association Stereotypes

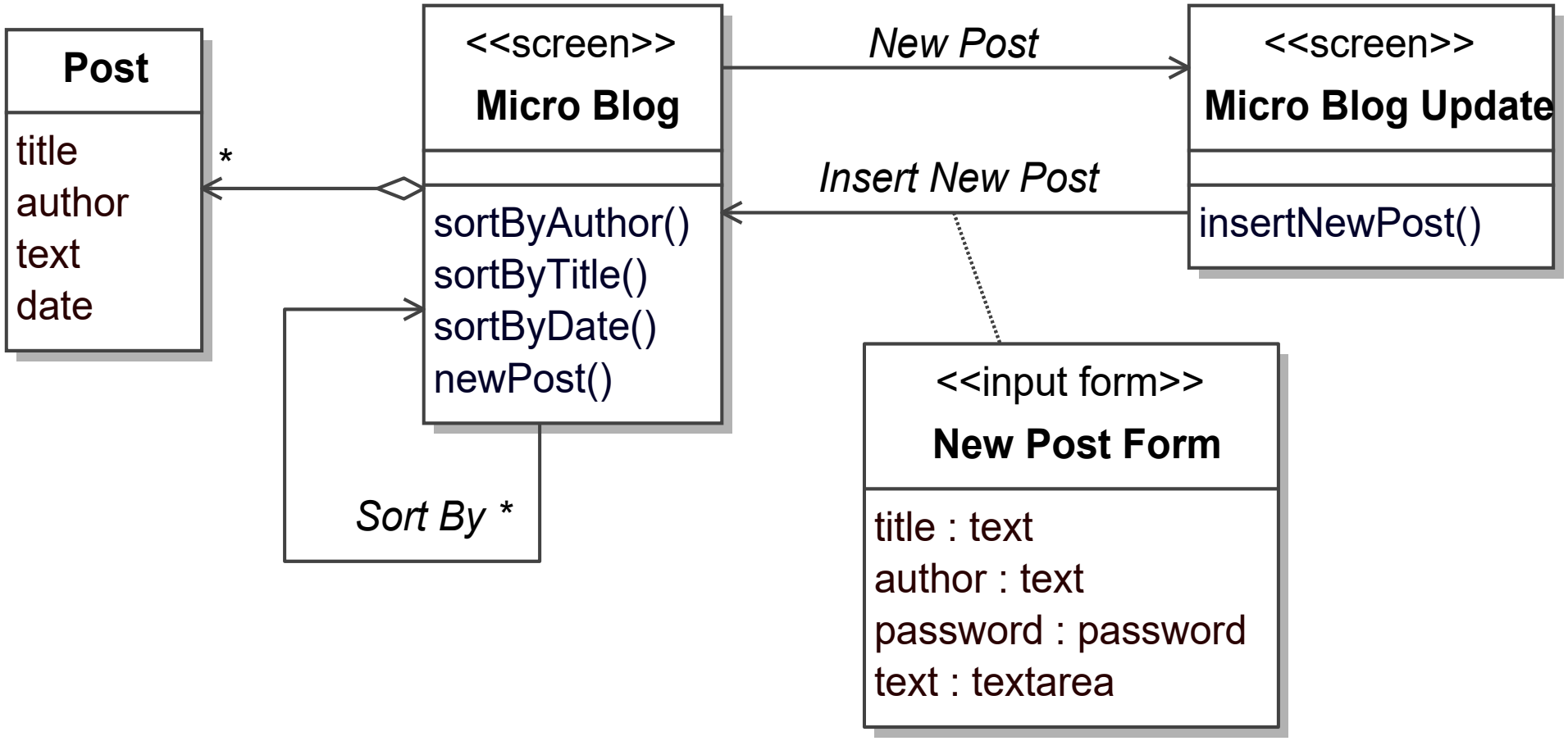
Stereotype	Source	Target	Description
<<link>>	<<Client Page>>	<<Client Page>> <<Server Page>>	Abstraction of <a href= ...>
<<builds>>	<<Server Page>>	<<Client Page>>	Identifies the HTML output of a server page's execution
<<submit>>	<<Form>>	<<Server Page>>	Form data submission
<<redirect>>	<<Client Page>> <<Server Page>>	<<Client Page>> <<Server Page>>	Makes the browser request the target resource

Case Study: Micro Blog Example

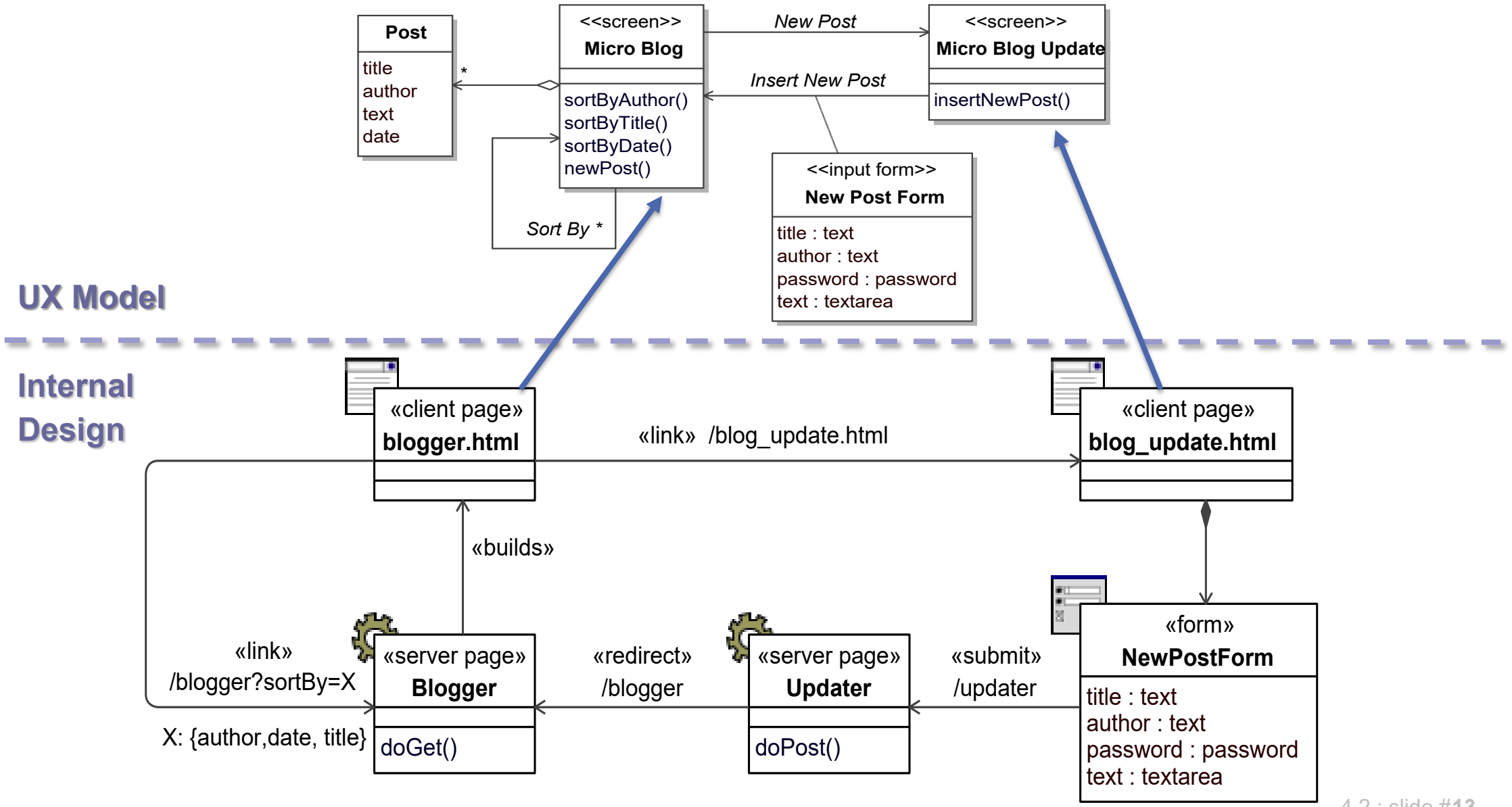
Micro Blog Example (cont.)



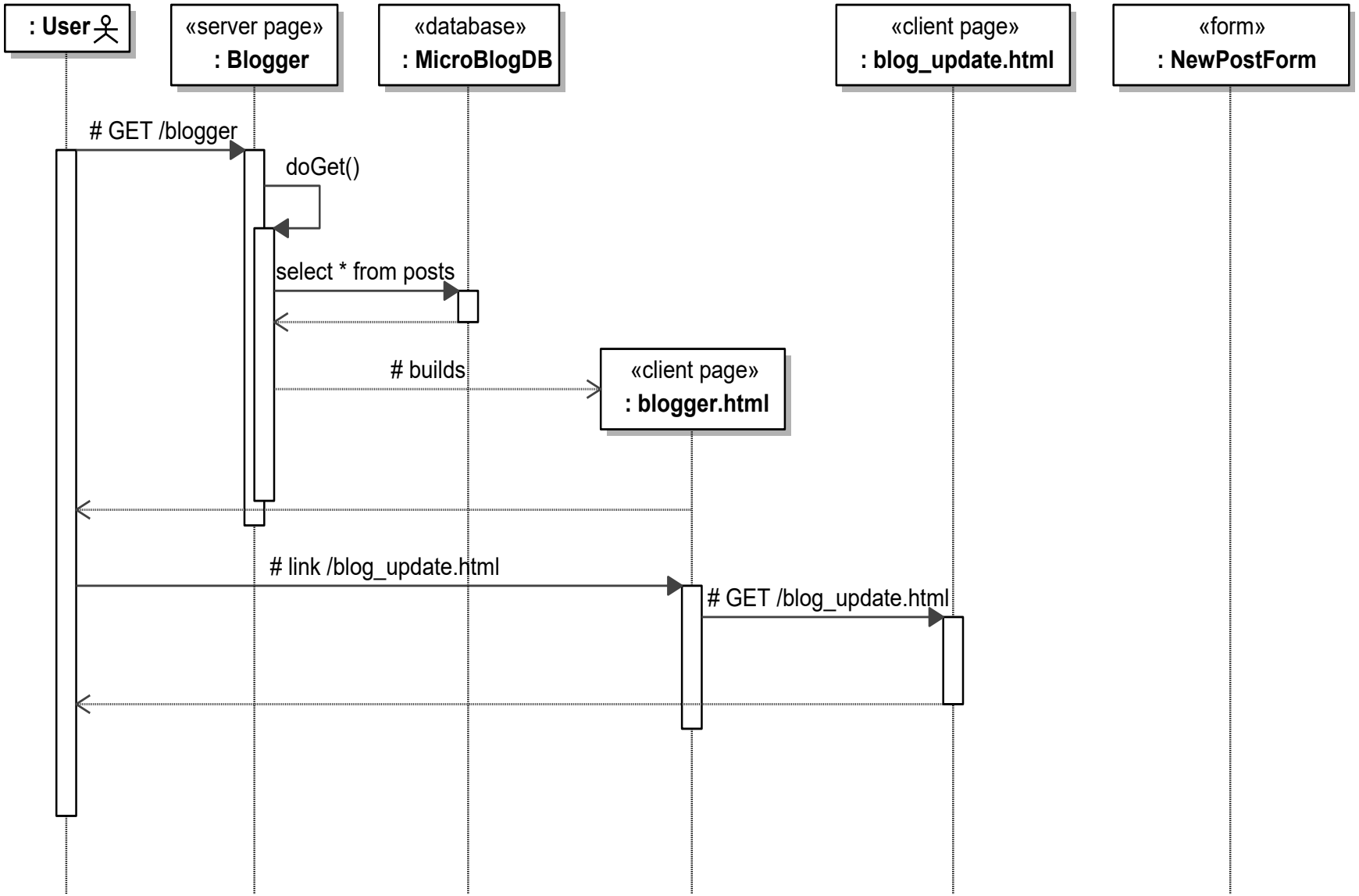
Micro Blog Example: UX Model



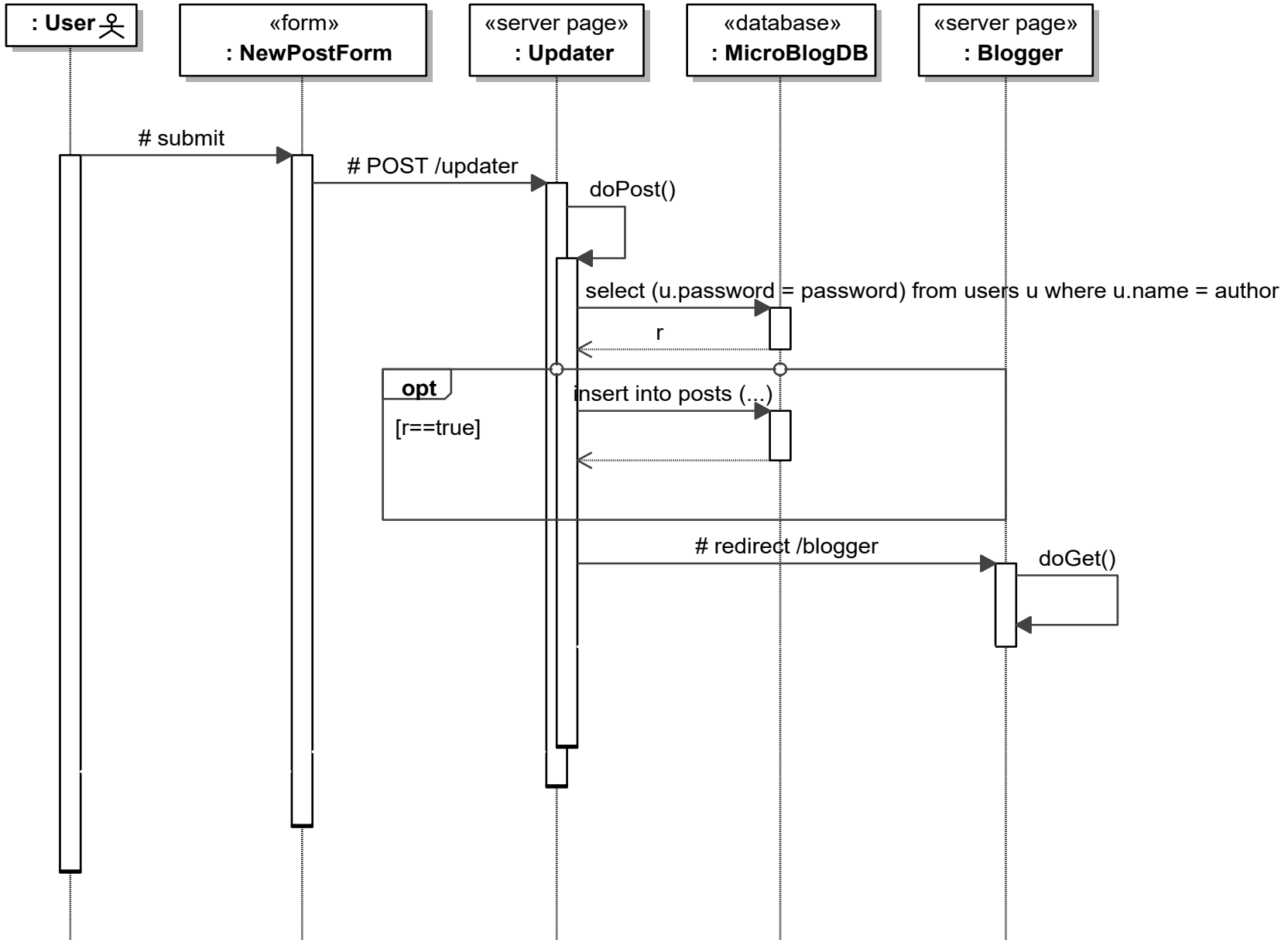
Main WAE2 Stereotypes: Micro Blog Example



Internal Design Sequence Diagram: Micro Blog Example



Internal Design Sequence Diagram: Micro Blog Example



The MVC Architectural Pattern

Divides an interactive application into three components/levels:

- **Model**:
  - Contains the functional core of the application
  - Encapsulates the appropriate data, and exports procedures that perform application-specific processing
- **View**:
  - Displays information to the user
  - Obtains the data from the model
  - Different views present the model's information in different ways
- **Controller**:
  - Accepts user input as events
  - Translates events to service requests for the model or display requests for the view

MVC for Web Apps: The RoR way

- Models
  - 1 model class for each type of entity manipulated by the app
  - Active Record pattern
- Controllers
  - Each controller class corresponds to 1 model class
  - Each controller action is handled by a particular method within that controller.
    - RESTful routes
- Views
  - Many view classes for each model class
  - “Rendered” by controllers

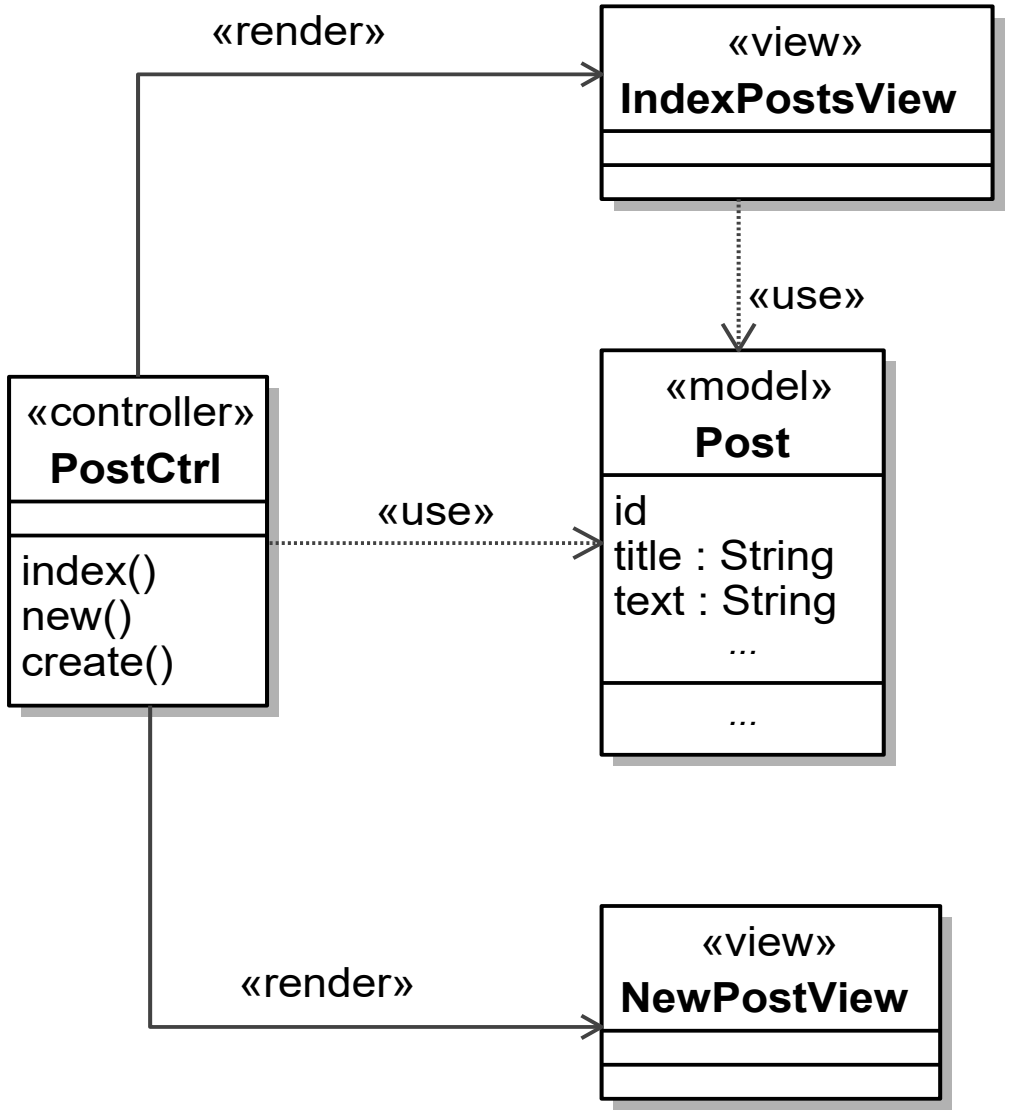
RESTful routes

- A **route** maps an incoming HTTP request to the appropriate controller and method.
- REST (**REpresentational State Transfer**) is an *architectural style* for designing networked applications
- RESTful routes:
  - Each entity manipulated by a Web app (model) is a *resource*
  - Any HTTP request should contain all the information necessary to identify both a particular resource and the action to be performed on it.

RESTful routes: Example

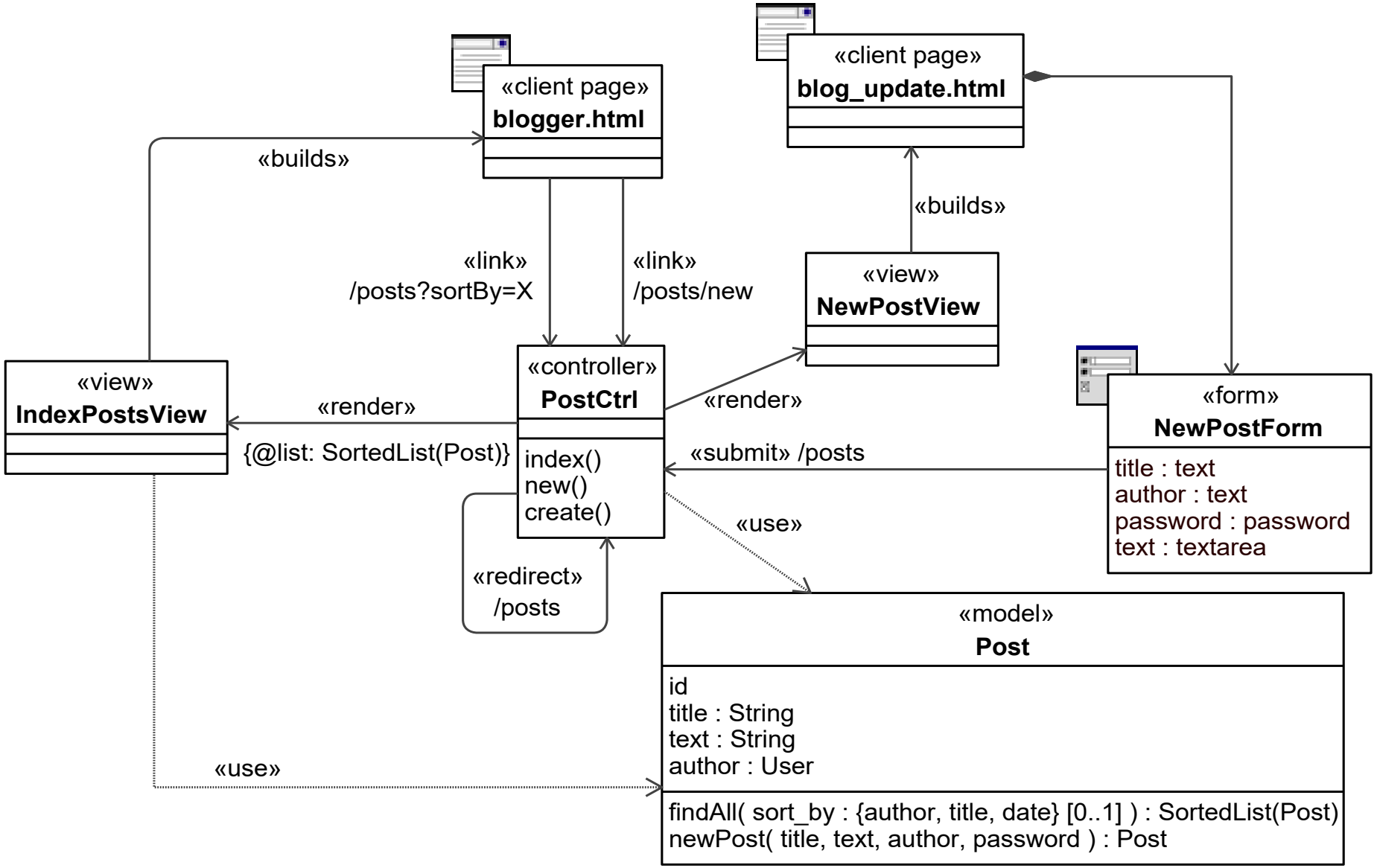
Operation on resource	Method & URI		Controller method	Rendered view
List (index) of posts	GET	/posts	index	IndexPostsView
Show just one post instance	GET	/posts/:id	show	ShowPostView
Display a fill-in form for a new post	GET	/posts/new	new	NewPostView
Create a new post from filled-in form	POST	/posts	create	-
Display from to edit existing post	GET	/posts/:id/edit	edit	EditPostView
Update post from fill-in form	PUT	/posts/:id	update	-
Destroy existing post	DELETE	/posts/:id	destroy	-

WAE2 + MVC: Changes

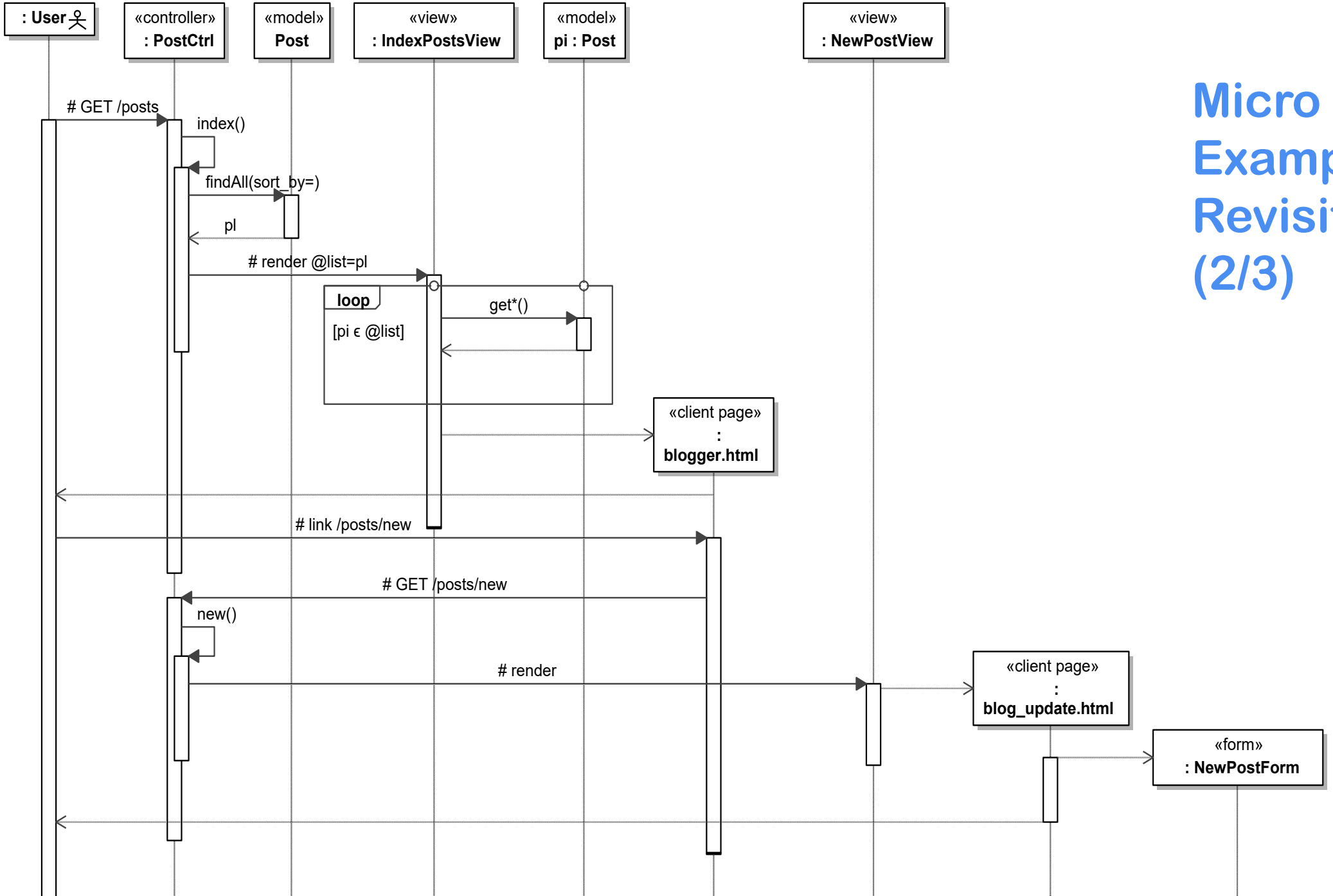


- Class Stereotypes:
  - Deleted: <<Server Page>>
  - **Added: <<Controller>> & <<View>>**
- Association Stereotypes:
  - **Added: <<render>>**

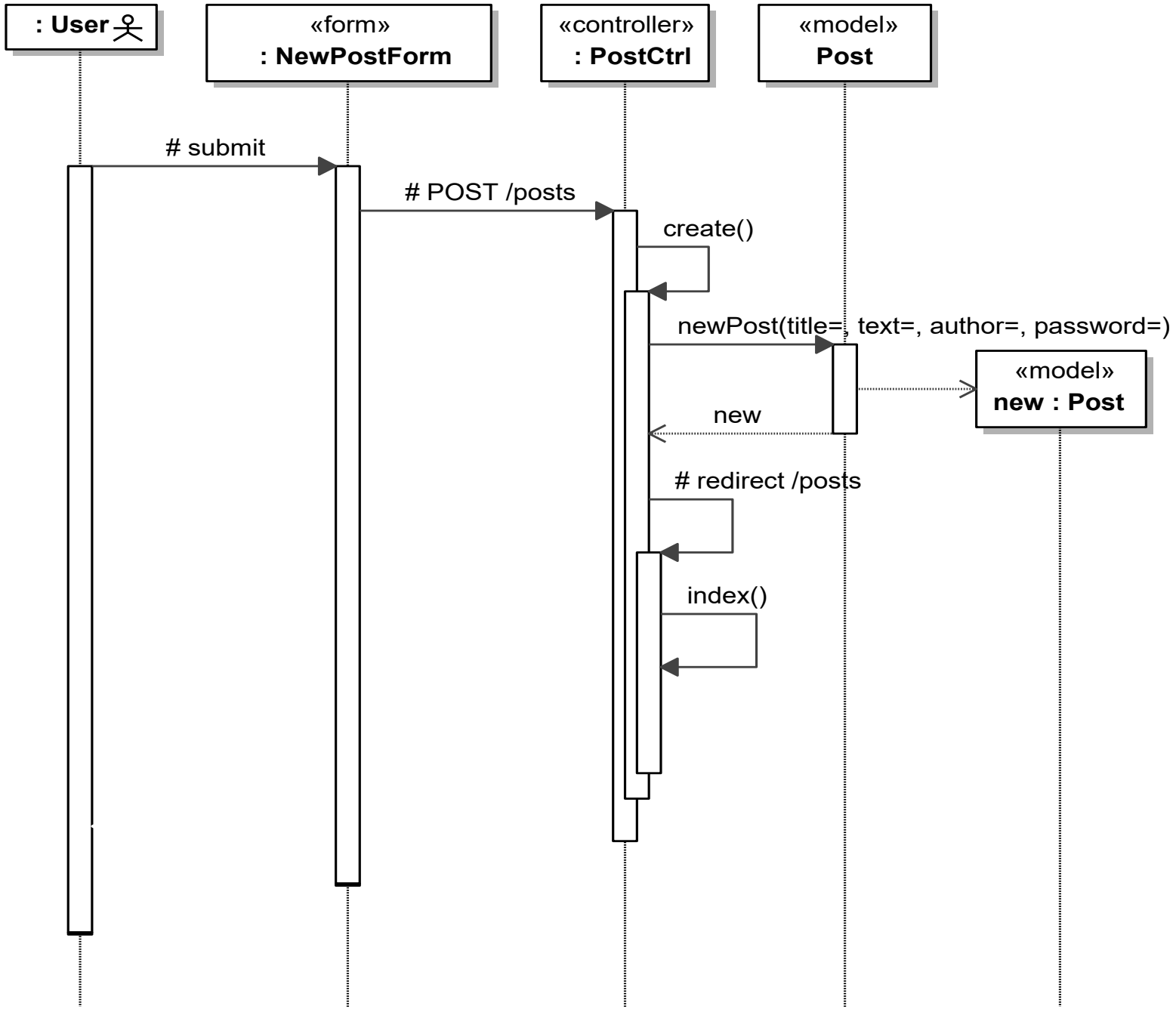
Micro Blog Example Revisited (1/3)



Micro Blog Example Revisited (2/3)



Micro Blog Example Revisited (3/3)



References

- CONALLEN, Jim Building **Web Applications with UML, 2on Edition**, Addison-Wesley, 2002.
- FOX, Armando; PATTERSON, David. **Engineering Software as a Service: An Agile Approach Using Cloud Computing**,Strawberry Canyon LLC.
- FOWLER, Martin **Patterns of Enterprise Application Architecture**, Addison Wesley, 2002.