

Chapter 11: Functional Design

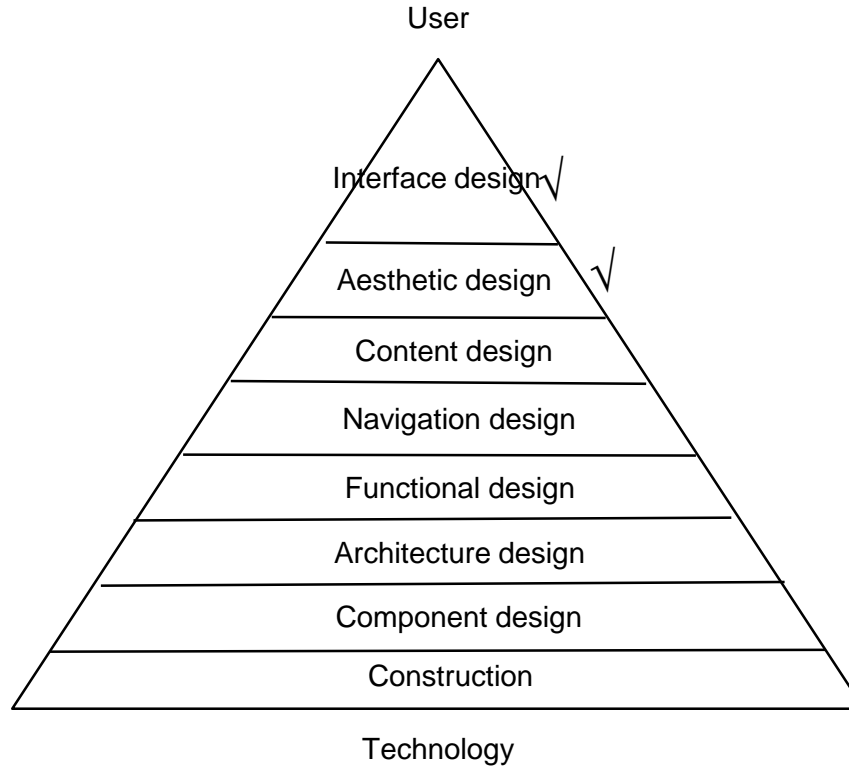
Software Engineering

Lecture-9

Functional Design Dr Shamsul huda



Functional design in Pyramid



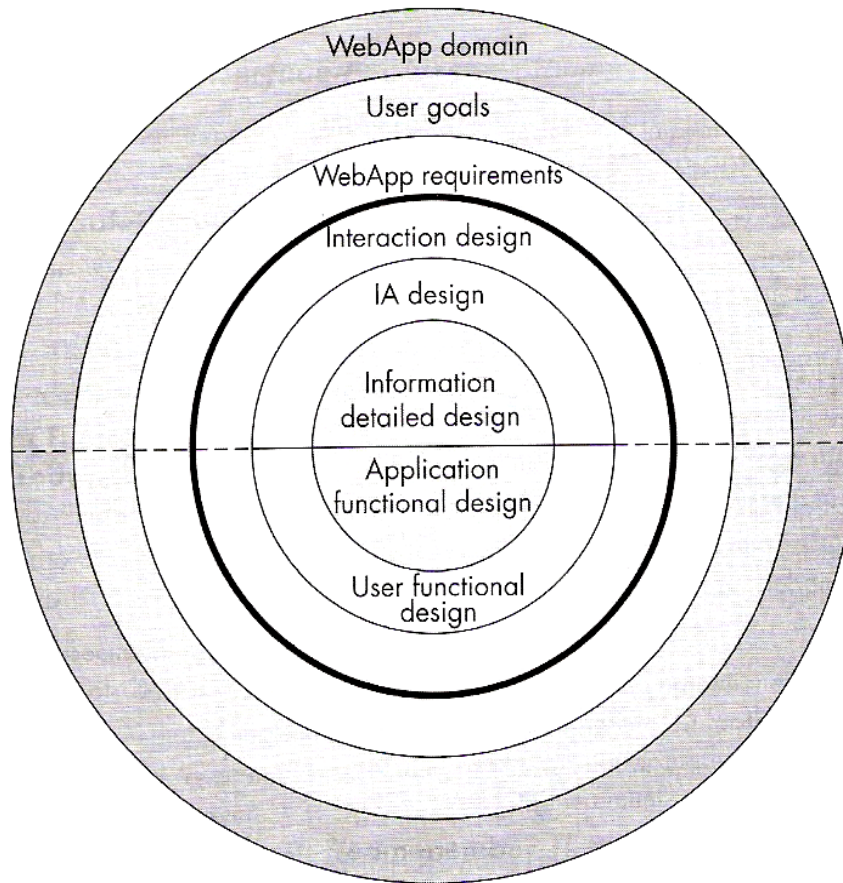
Why Functional Design

- ▶ Today, many WebApp's incorporate rich dynamic content, adaption and personalization for individual users, powerful interactivity, and intricate workflows and process support.
- ▶ All this has led to an increasing complex interplay between the content that underpins most WebApps and the functionality.



Functional Design in the Design Process

- Functional design is interwoven with other design activities



- **The outer layers specifies specification of WebApp**
- **Inner layers relate to design**
- **There is a clear relationship between different functionality levels and the various other associated design tasks**
- **Information Architecture (IA) and user functional design can progress in parallel**
- **Application level function design will follow and typically respond to user level function design**
-

Functional Design in the Design Process

- ▶ Commences with the design of the user functionality, derived from a definition of user goals (such as use cases!)
 - ▶ We have developed a process model for a use case
- ▶ The user functionality design is developed in parallel with the information architecture (IA) to ensure that they are consistent.
- ▶ The user functionality design, application functionality is combined with the information design (data and content) to create a functional architecture.
- ▶ Application functionality design follows the user functionality design, and will also affect the functional architecture.



A wider view of functional design

- *User-level functionality* is the expression of the WebApp capabilities that support users in achieving their goals.
- *Application-level functionality* represents lower-level internal functionality that may not be directly visible to users.
- *Application level functional design will usually follow, and respond to, the user-level functional design.*



Functionality Categories

▶ Group 1: User-Level (External) Functionality

- ▶ Category 1A: User Interaction Support. This provide the functionality in which a user might directly interact with the WebApp.
 - e.g. highlighting a link, drop-down menus, navigational structures and breadcrumbs
- ▶ Category 1B: User Information Support. Covers any functionality that affects the nature of the information and/or how it is presented to the user.
 - e.g. live or dynamic content updating, automatically including a site banner or forcing style sheets.
- ▶ Category 1C: User task Support. Includes any functionality that supports, guides, or controls the user in achieving specific tasks.
 - e.g. searching, login mechanism, access control.

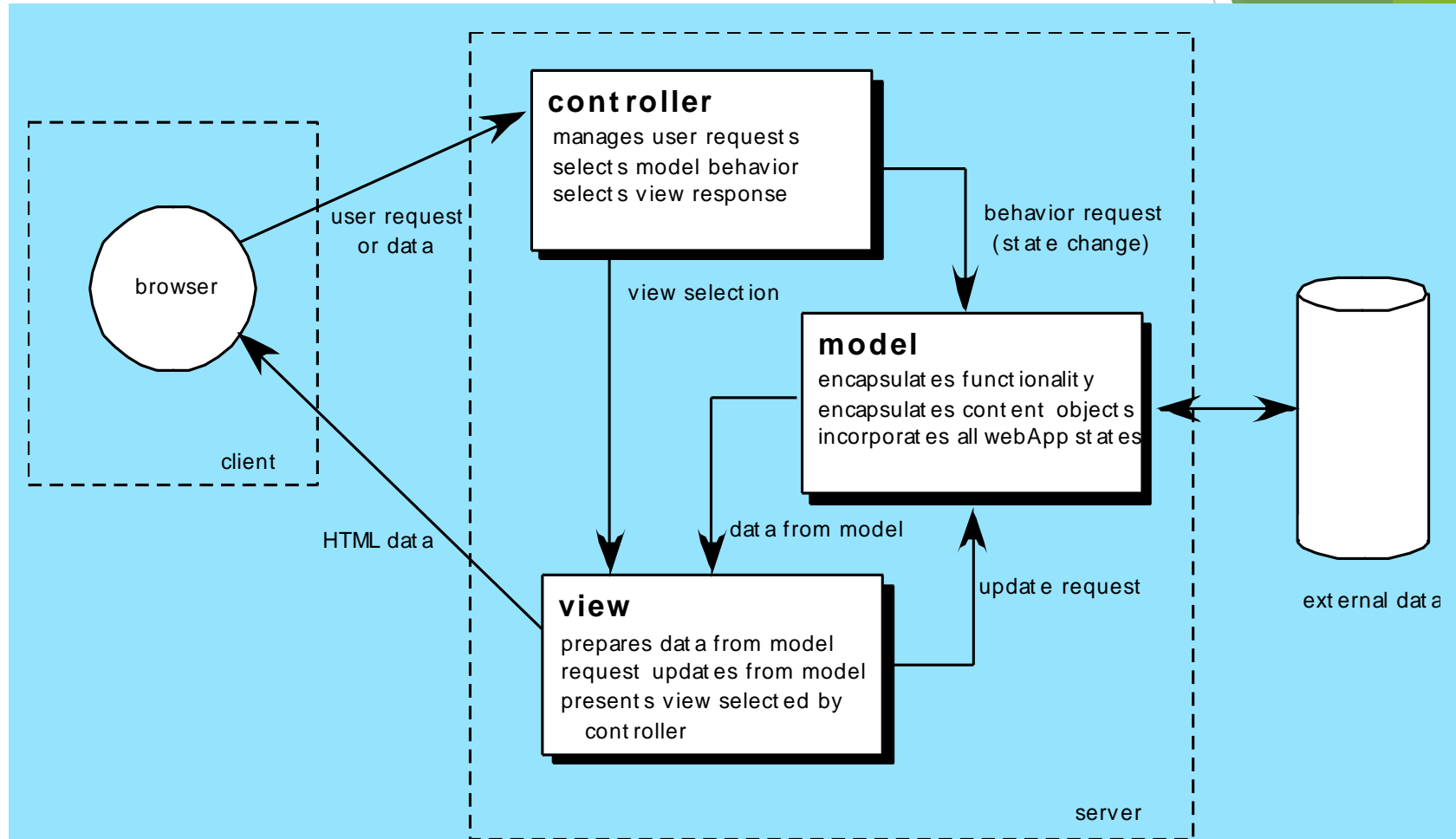
Functionality Categories (cont'd)

▶ Group 2: Application-Level (Internal) Functionality

- ▶ Category 2A: Application Interaction Support. **Functionality that is necessary to manage and maintain the interaction mechanism** but not to directly control the specific elements with which a user interacts.
 - e.g. tracking and recording of the interactions by particular users.
- ▶ Category 2B: Application Information Support. Any application-level management of information.
 - e.g. database maintenance, development and maintenance of content management system (CMS).
- ▶ Category 2C: Application task Support. The internal WebApp support for overall user tasks.
 - e.g. gateways to other applications (such as payment systems), user authentication, processing of submitted orders.

MVC Architecture for Functional Design

- The model-view-controller or MVC architecture
 - Separates the user interface from the WebApp functionality and the information content.



MVC Architecture for Functional Design

- ▶ The *model* part of MVC contains all application specific content and processing logic, including:
 - ▶ all content objects
 - ▶ access to corporate data, individual data file or information sources
 - ▶ all processing functionality that are application specific
- ▶ The *view* contains all interface specific functions and enables
 - ▶ the presentation of content and processing logic
 - ▶ all processing functionality required by the end-user
 - ▶ The data developed by the model must be formatted and organized by the appropriate view object and then transmitted from the application server back to the client-based browser for display on the customer's machine
- ▶ The *controller* manages access to the model and the view and coordinates the flow of data between them.
 - ▶ The controller also selects the view object that is applicable based on the user request

How to Develop the Functional Architecture using component models

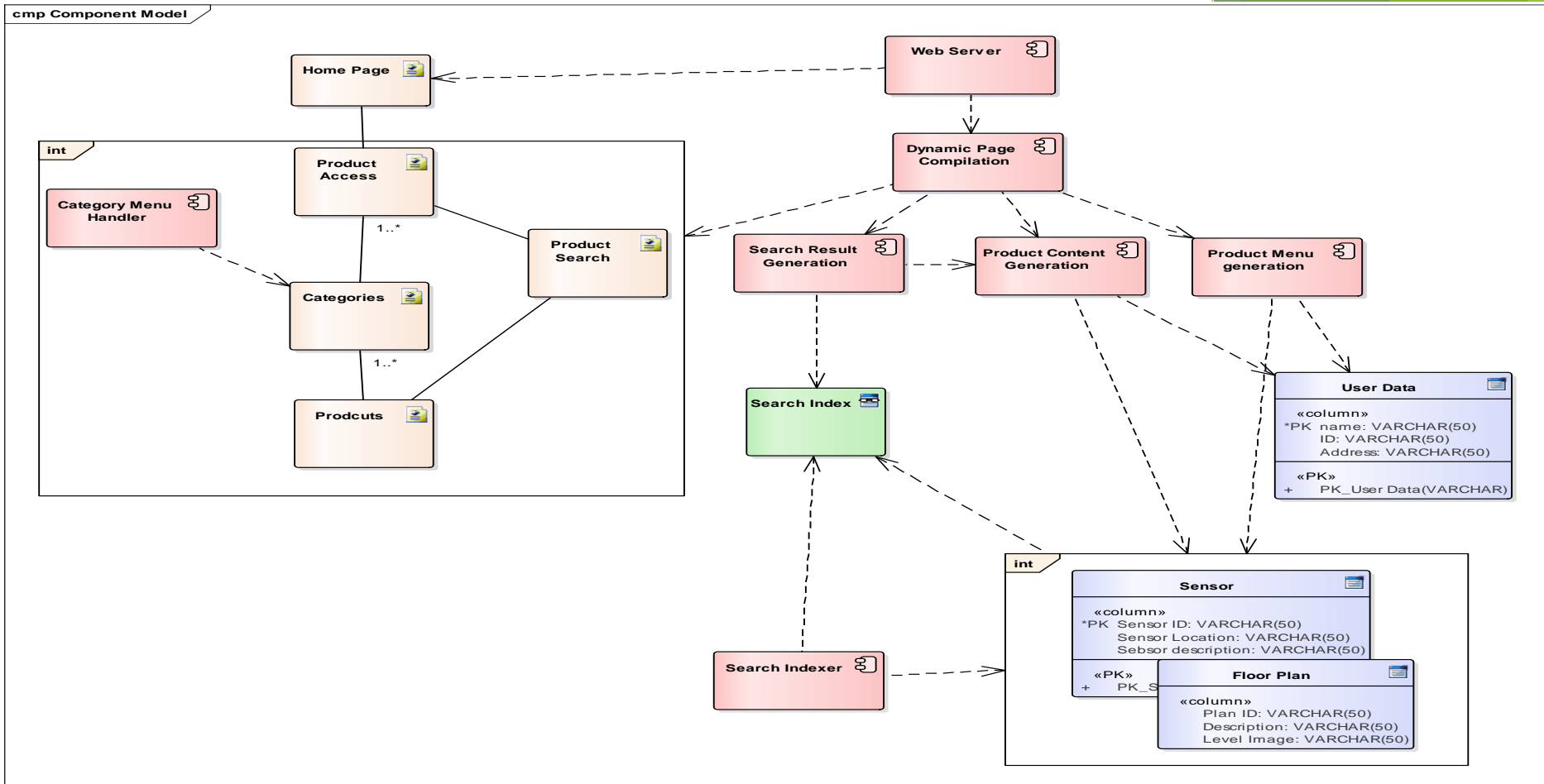
- ▶ Component based approach:
- ▶ A functional architecture is a representation of the functionality of the WebApp and describes the key functional components in the WebApp and how these components interact with each other.
- ▶ Steps:
- ▶ Identify user interaction scenarios from both the WebApp analysis model and the initial information architecture.
- ▶ Each scenario or part of the scenario can be decomposed into the following generic scenario component categories:
 - ❖ *System interaction* (i.e. functionality associated with interactions with other systems external to the WebApp, communication components, database components, task management component)
 - ❖ Process models
 - ❖ *Information selection* (i.e. functionality associated with the identification and/or selection of information to be presented to the user).
 - ❖ *Information processing* (i.e. the analysis or calculation of data).
 - ❖ *Information compilation* (i.e. functionality associated with merging information together to be presented to the user). Consider whether the specific function needs to be considered for dynamically content generation based on user request,

How to Develop the Functional Architecture using component models

- ▶ Then partition the functionality/processes (group the similar functions) and assign this to a particular component.
 - ▶ This component has clearly defined roles and interfaces and will perform those grouped functions
- ▶ Specify where does this functional component exists and what does it interact with.



Example Functional Architecture: component models example (SafeHomeAssured.com)



- ▶ A functional architecture has been presented for SafehomeAssured.com
- ▶ Text book:S.Presman (chapter-11,page-280-281)
- ▶ This has been prepared using EA. In EA, you also can use Webmodeling tools and component model tool.
- ▶ Left hand side is the client part , right hand side server part
- ▶ Data model has been integrated with functional architecture

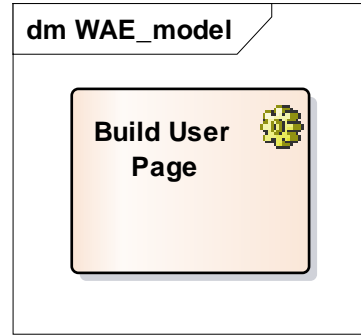
Functional design using WAE

- ▶ Unified Modeling Language (UML) is a standard in the development of software systems.
- ▶ UML-Based Web Engineering (UWE): is to adhere to standards, and extend the UML to model Web-specific features in WebApp design.
- ▶ WAE (Web Application Extension for UML) is a design approach that links information with functional WebApp Components.
- ▶ *WAE models the connection between client-side content and behaviour and server-side functionality.*
- ▶ *More on physical implementation specification*
- ▶ *Can connect functions are defined in the process model*

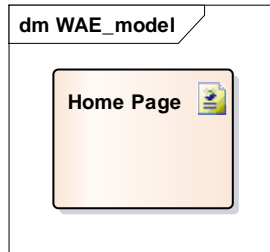


WAE model Stereotypes: Server Page and Client page [1]

A server page : (server page stereotype class) : This is a dynamic web page that has content generated each time it is requested. (the page's scripts, subroutines, and functions represent the class's methods; variables declared in the script that have page scope represent the class's attributes.

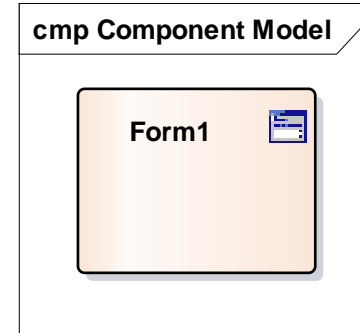
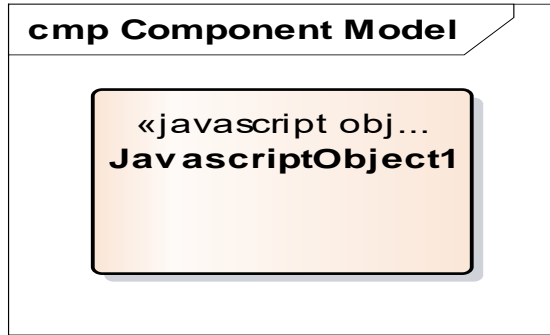


A client page: This is an HTML- page with presentation, and logic. Client page attributes map to variables declared in the page's script tags that have page scope (accessible by any function in the page).

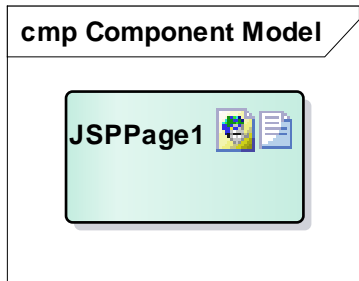


WAE model: javaScriptObject and form [1]

A JavaScript Object: This is represented by a class of stereotype JavaScript. JavaScript object instances exist only in the context of client pages



A form: This is represented by a class of stereotype Form. It is a collection of input fields that are part of the client page. In HTML, a form is often associated with Submit button

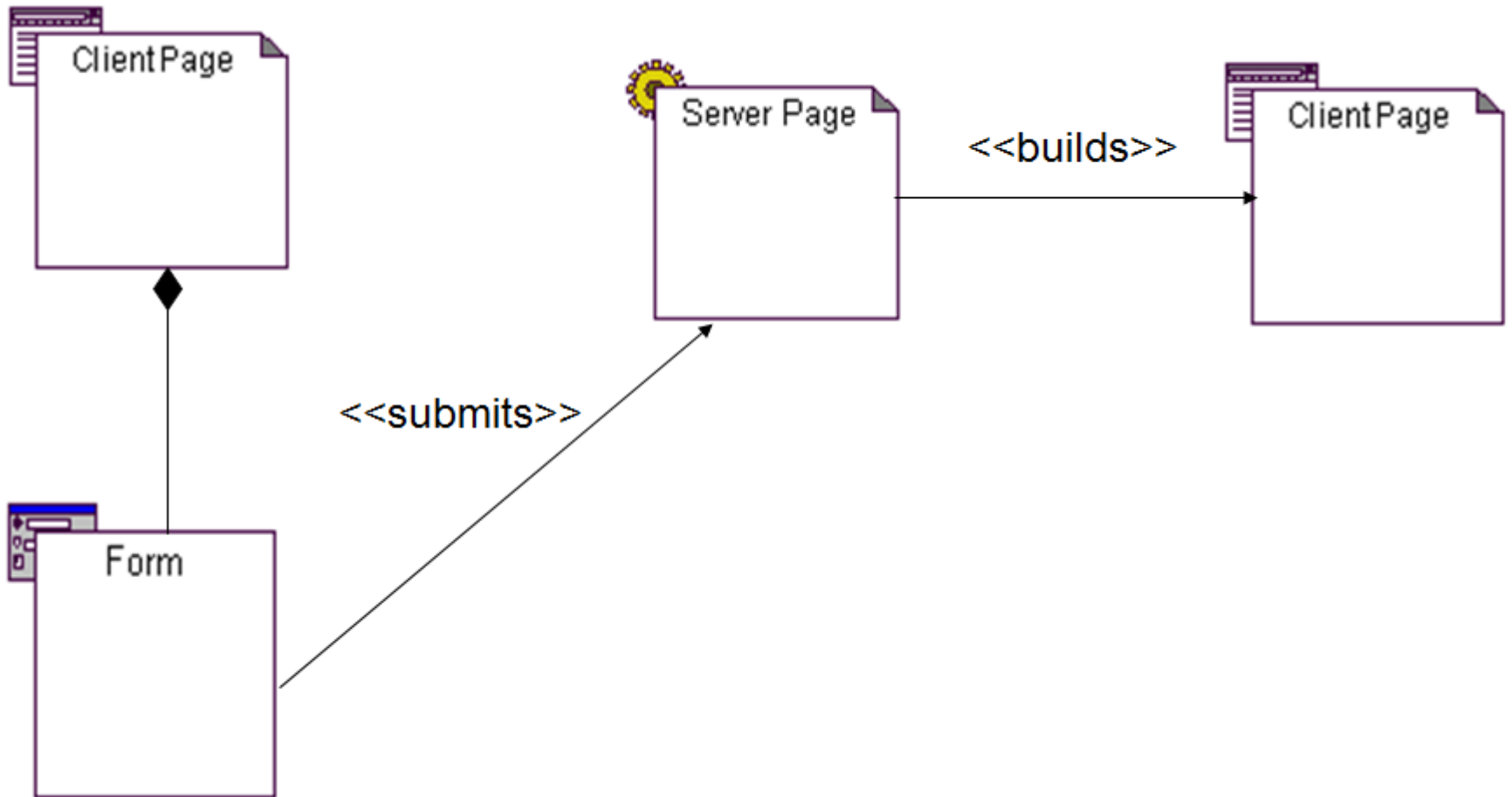


A Java server Page (JSP): JSP Page is a web page that implements JSP server-side code.

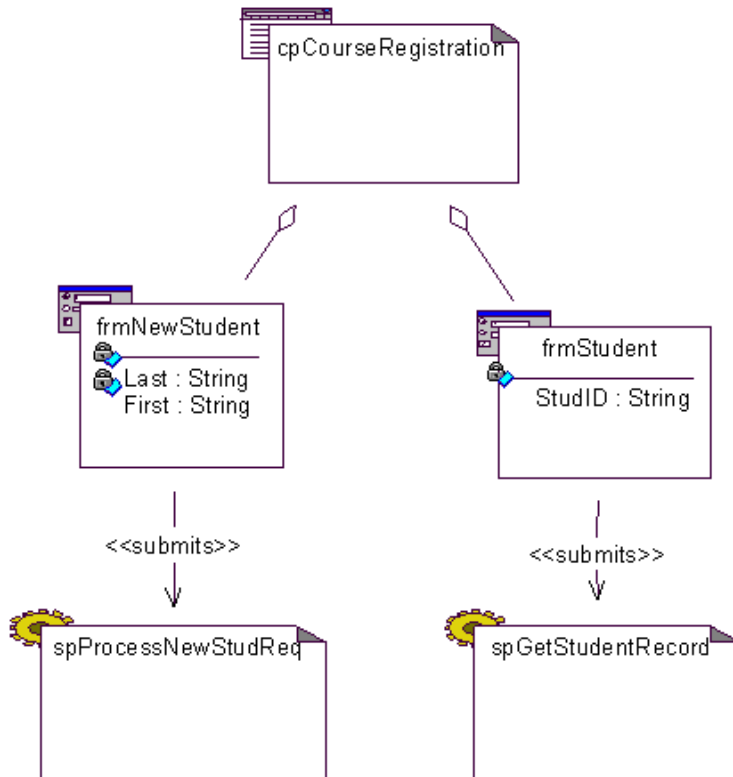
WAE: Association Stereotypes Relationships [1]

STEREOTYPE	DESCRIPTION
<<build>>	A directional relationship between a server page and a client page. It identifies the HTML output of a server page's execution.
<<link>>	A relationship between a client page and a server page resource, or Web page. The target may be a client page or a server page. This is an abstraction of the HTML anchor element (href).
<<submit>>	A directional relationship between an <<HTML form>> and a server page. All the forms attributes are submitted.
<<redirect>>	A directional relationship between a client page or a server page and another page. This association indicates a command to the client to request another resource.
<<forward>>	A directional relationship between one server page and another server page or a client page. This association represents the delegation of processing a client's request for a resource to another server-side page

Example: How WAE model shows implementation detail [1]



Example: WAE Model: Client page: HTML Form [1]

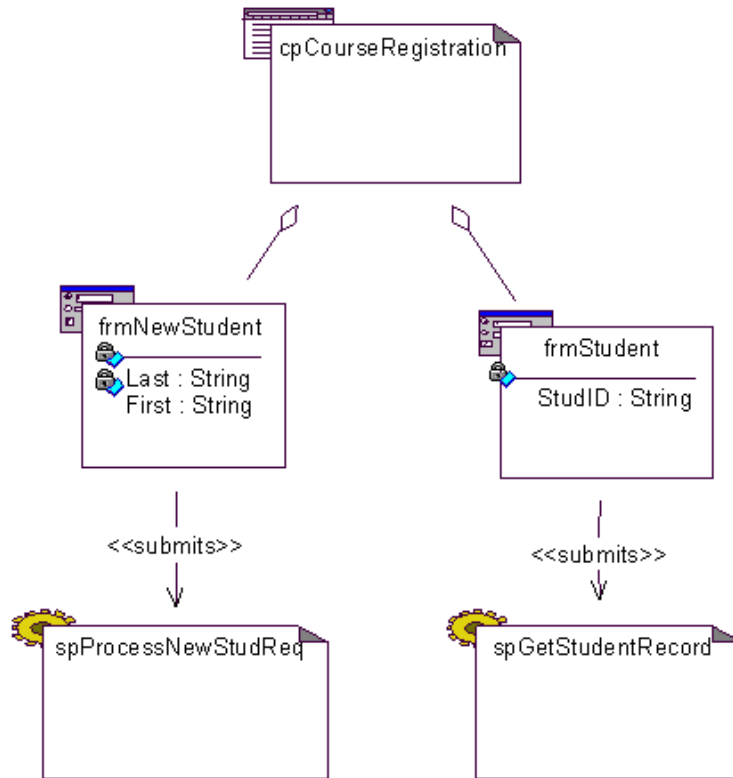


The screenshot shows a Mozilla Firefox browser window displaying an HTML form. The form is titled "New Student" and "Current Student". It contains the following fields and buttons:

- New Student**:
 - Last Name:
 - First Name:
 - Submit button
- Current Student**:
 - Student ID:
 - Submit button

The status bar at the bottom of the browser window shows "Done".

Example: WAE Model: Client page: HTML Form [1]



CourseRegistration.html

```
<html>
<head></head>
<body>
<p><b>New Student</b></p>

<form method=post action="ProcessNewStudReq.jsp">

<p>Last Name: <input type=text name="Last"></p>
<p>First Name: <input type=text name="First"></p>
<p><input type=submit name="Submit"></p>

</form>
<br>
<p><b>Current Student</b></p>

<form method=post action="GetStudentRecord.jsp">

<p>Student ID: <input type=text name="StudID"></p>
<p><input type=submit name="Submit"></p>

</form>

</body>
</html>
```

Case study: Music portal web application

Melbourne Entertainment Pty Ltd will develop a music portal web application

User will be able to search album by their name and also by the artist name. The search results by the search process will provide a list of matching albums. The album will have a link to a detail page for each album which shows the title of the album, the name of the artist, the list of songs and the album's price. Each registered user has a credit account that is used to buy albums. The credit account can be recharged by credit card payment. Both user and artists can maintain their address book to search other user or artists. Each named Contact should store an email address, two phone, two postal addresses and a picture. Name and email, Phone, Picture and Address should be maintained. Address may represent more information. The links for logging in or out, for registering and to the user's account page are always shown. This also holds for the album search box. The functionalities of address book will be as follows. You can extend this description based on your assumption for other functionalities of the music portal.

User need to have an account to buy an album and need to be logged in.

Search Album using artist

Search by song index

Download Album

Recharge your account, deposited advanced

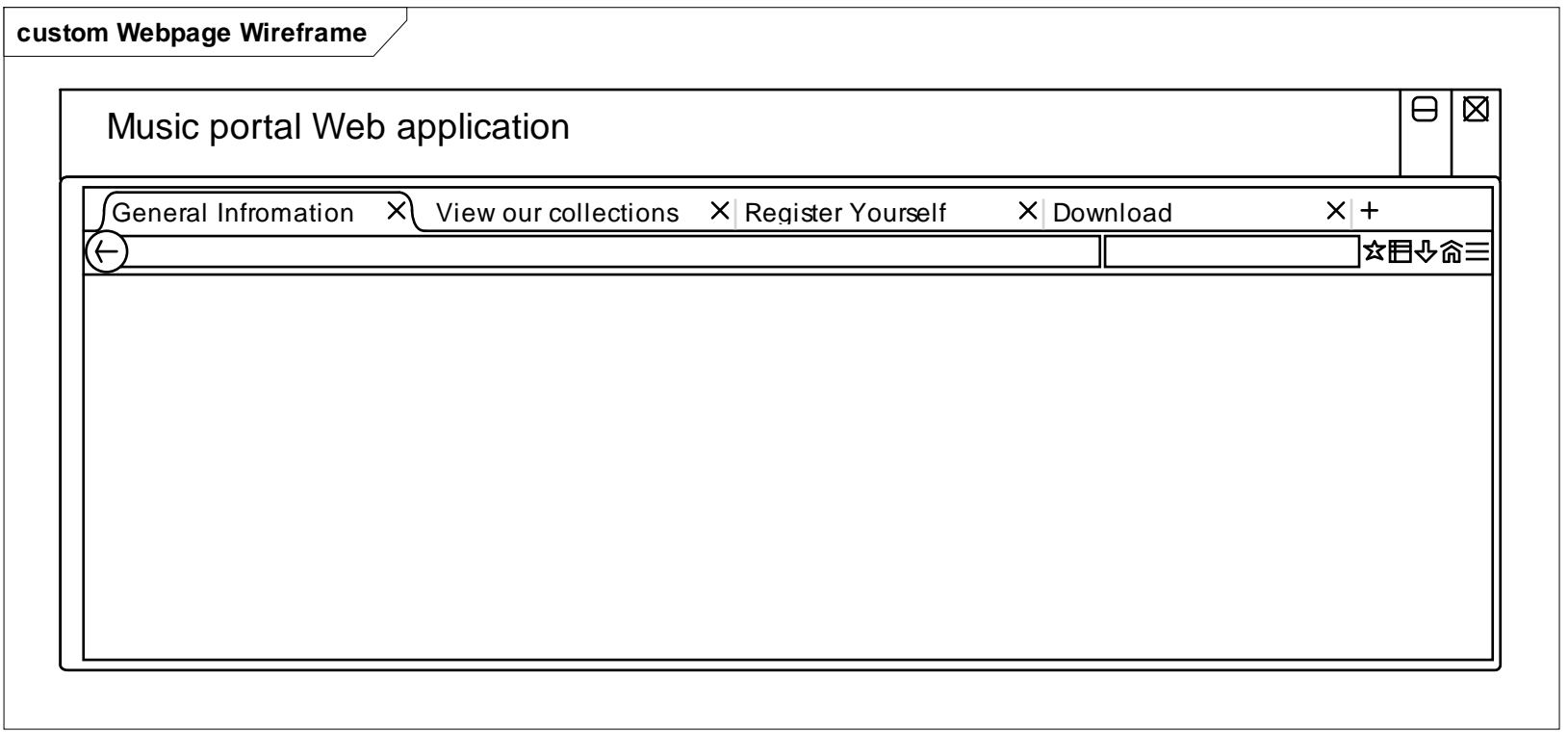
Credit check facility

Register user

View Albums

Shopping Cart facility

Music Portal Web Application: Wireframe



Music Portal Web Application: Wireframe

custom Webpage Wireframe

Download (you need to be logged in)

User Name

User 1

E-mail

user1@gmail.com

Cancel

Log-in

Music Portal Web Application: Wireframe

custom Webpage Wireframe

Down Load Music and Buy



Detail:
Artist Name
Description
Price per unit
Status (In stock/Out stock)
Delivery charge
Delivery time

Add to cart

List: Song

Item1

Item2

Item3

Item4

Submit: Show Detail

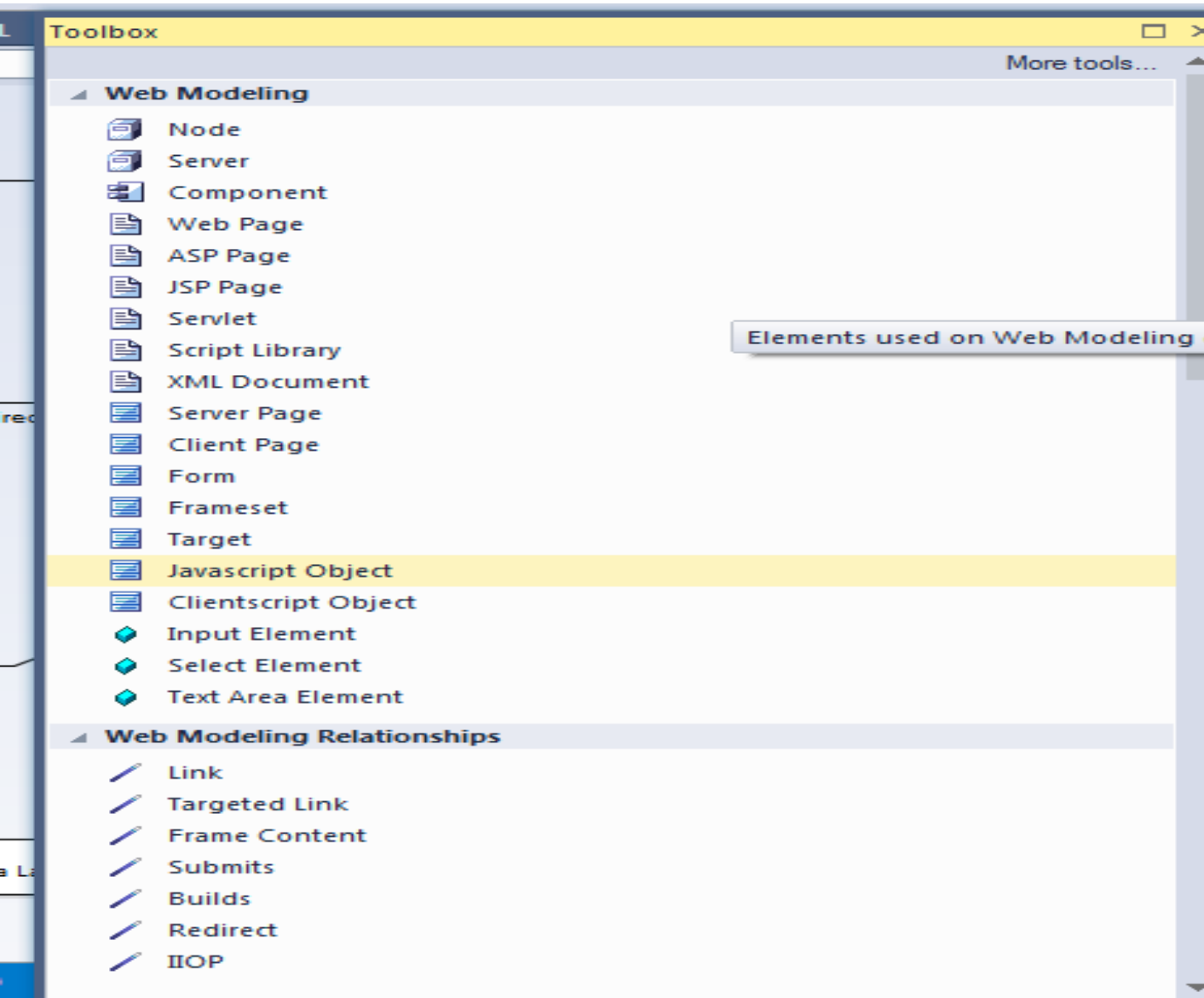
Shopping Cart

Item Index	Item Name	Item ID	Item Unit Price	Item Status	Quantity	Edit	Delete
1	Song1	111-222	\$200	In Stock	1	Edit	Delete
2	Song2	R-3344-99	\$50	In Stock	4	Edit	Delete
3	Song3	L-111-777	\$100	Out of Stock	1	Edit	Delete

Cancel Order

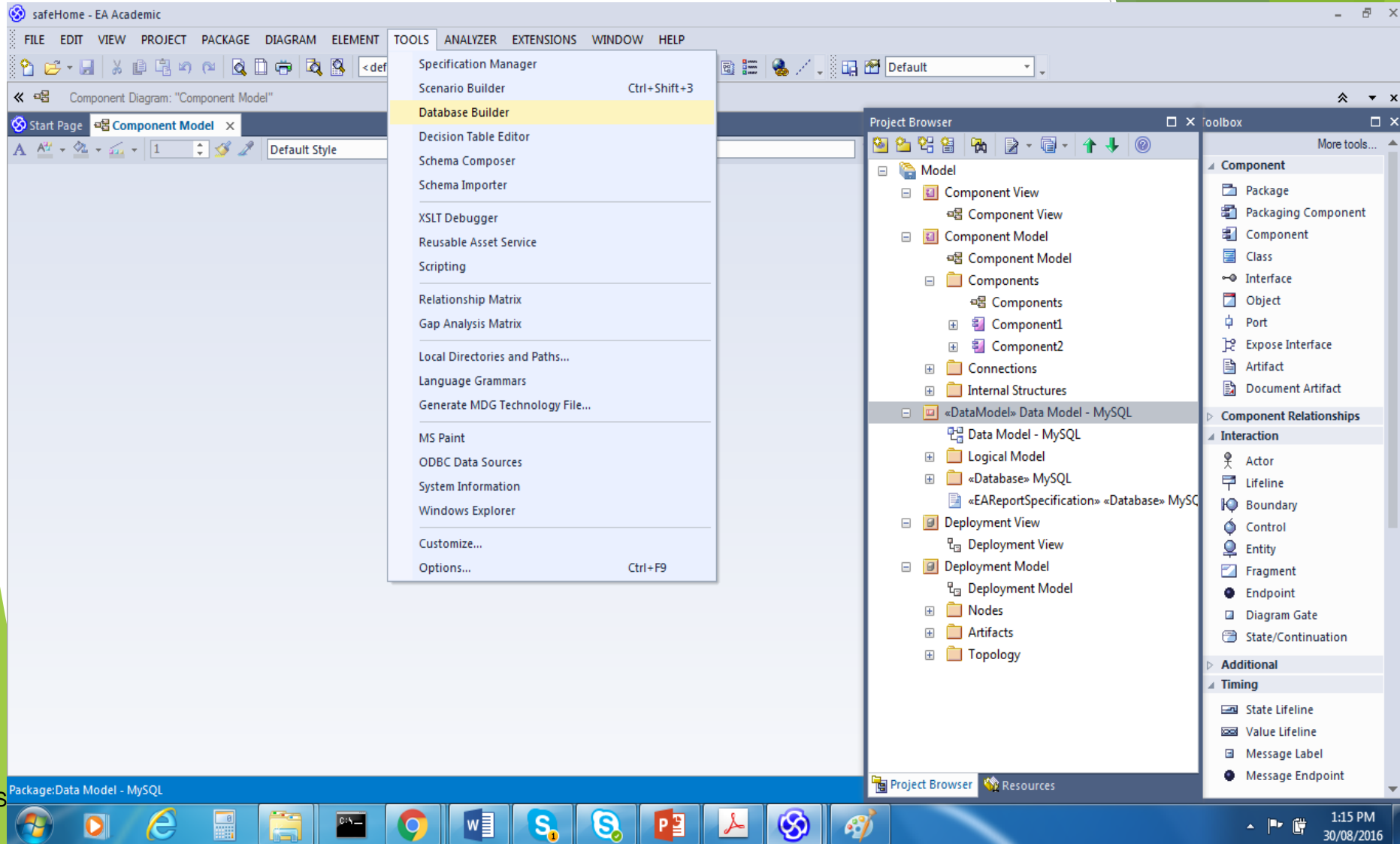
Confirm Order

Functional Architecture - and Enterprise Architect



Webmodeling Tool

Data model in Enterprise Architect



Data Model in enterprise Architect:

safeHome - EA Academic

FILE EDIT VIEW PROJECT PACKAGE DIAGRAM ELEMENT TOOLS ANALYZER EXTENSIONS WINDOW HELP

Component Diagram: "Component Model"

Start Page Component Model

Model Wizard

Model Patterns Application Patterns VEA Examples

Add to Package: Model

Technology

- Core Modeling
- BPMN
- SysML
- Business
- Database**
- Software
- Services
- Geospatial
- Framework
- Others

Database Engineering

- ☐ Data Model - DB2
- ☐ Data Model - Firebird
- ☐ Data Model - Informix
- ☐ Data Model - Ingres
- ☐ Data Model - InterBase
- ☐ Data Model - MSAccess
- ☐ Data Model - MSAccess2007
- ☐ Data Model - MySQL
- ☐ Data Model - Oracle
- ☐ Data Model - PostgreSQL
- ☐ Data Model - SQLServer2012
- ☐ Data Model - SQLServer2008
- ☐ Data Model - SQLServer2005
- ☐ Data Model - SQLServer2000
- ☐ Data Model - SQLServer7
- ☐ Data Model - SQLite

OK Cancel Help

Package: Data Model - MySQL

Project Browser Resources

Component

- Package
- Packaging Component
- Component
- Class
- Interface
- Object
- Port
- Expose Interface
- Artifact
- Document Artifact

Component Relationships

Interaction

- Actor
- Lifeline
- Boundary
- Control
- Entity
- Fragment
- Endpoint
- Diagram Gate
- State/Continuation

Additional

Timing

- State Lifeline
- Value Lifeline
- Message Label
- Message Endpoint

1:18 PM 30/08/2016

Data Model In Enterprise Architect

The screenshot displays the safeHome - EA Academic software interface, specifically the Database Builder window. The main window shows a table definition with the following columns:

Name	Type	Length	Scale	PK	Not Null	Alias
Plan ID	VARCHAR	50		<input type="checkbox"/>	<input type="checkbox"/>	
Description	VARCHAR	50		<input type="checkbox"/>	<input type="checkbox"/>	
Level Image	VARCHAR	50		<input type="checkbox"/>	<input type="checkbox"/>	
New Column...						

The Project Browser on the right shows the following structure:

- Component1
- Component2
- Connections
- Internal Structures
- «DataModel» Data Model - MySQL
 - Data Model - MySQL
- Logical Model
- «Database» MySQL
 - MySQL
 - Functions
 - Procedures
 - Queries
 - Tables
 - «table» Events
 - «table» Floor Plan
 - «table» Order
 - «table» Sensor
 - «table» User Data
- Views
 - Views

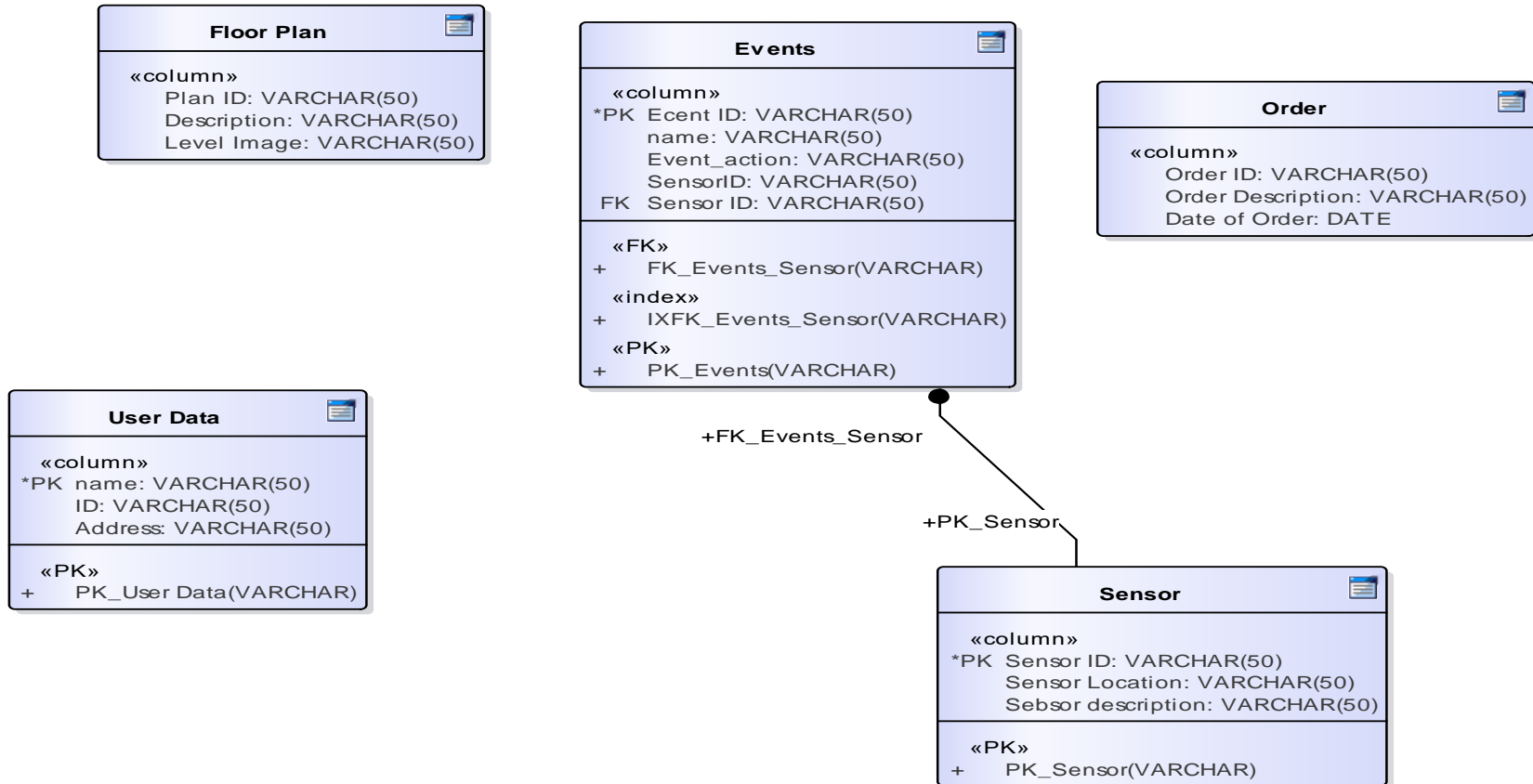
The Toolbox on the far right lists the following elements:

- Data Modeling
 - Table
 - View
 - Procedure
 - Sequence
 - Function
 - Association
 - Database Connection
 - SQL Query
- Interaction
 - Actor
 - Lifeline
 - Boundary
 - Control
 - Entity
 - Fragment
 - Endpoint
 - Diagram Gate
 - State/Continuation
- Additional
- Timing
 - State Lifeline
 - Value Lifeline
 - Message Label
 - Message Endpoint
 - Diagram Gate
- State

The status bar at the bottom indicates "Line: 66 Column: 1". The system clock shows 1:44 PM on 30/08/2016.

Partial data Model in EA

dm Tables



Example (cont'd)

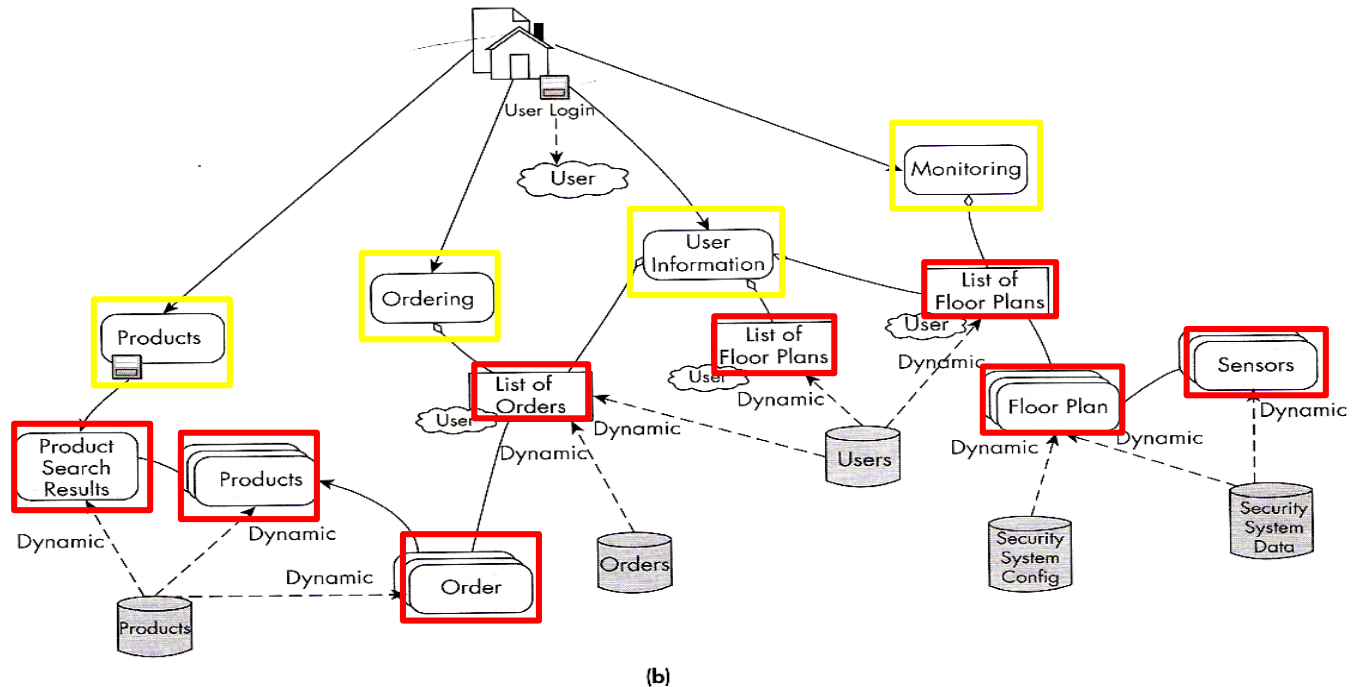
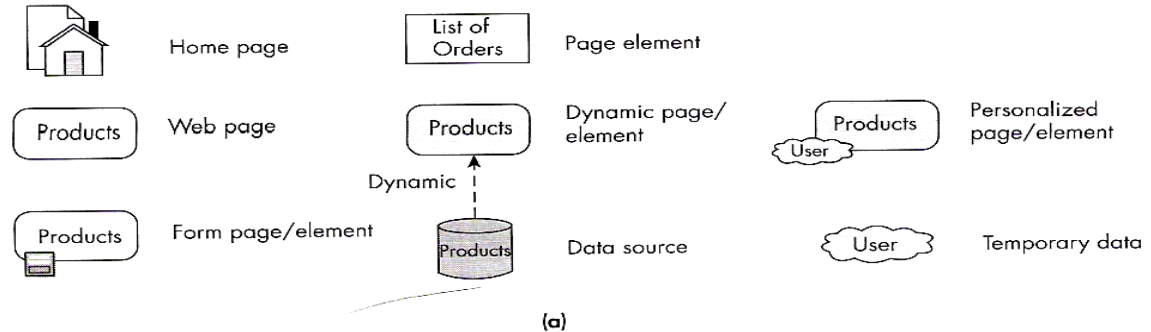
A high-level blueprint:

Dark cylinders = data sources

Red icons = dynamically constructed pages.

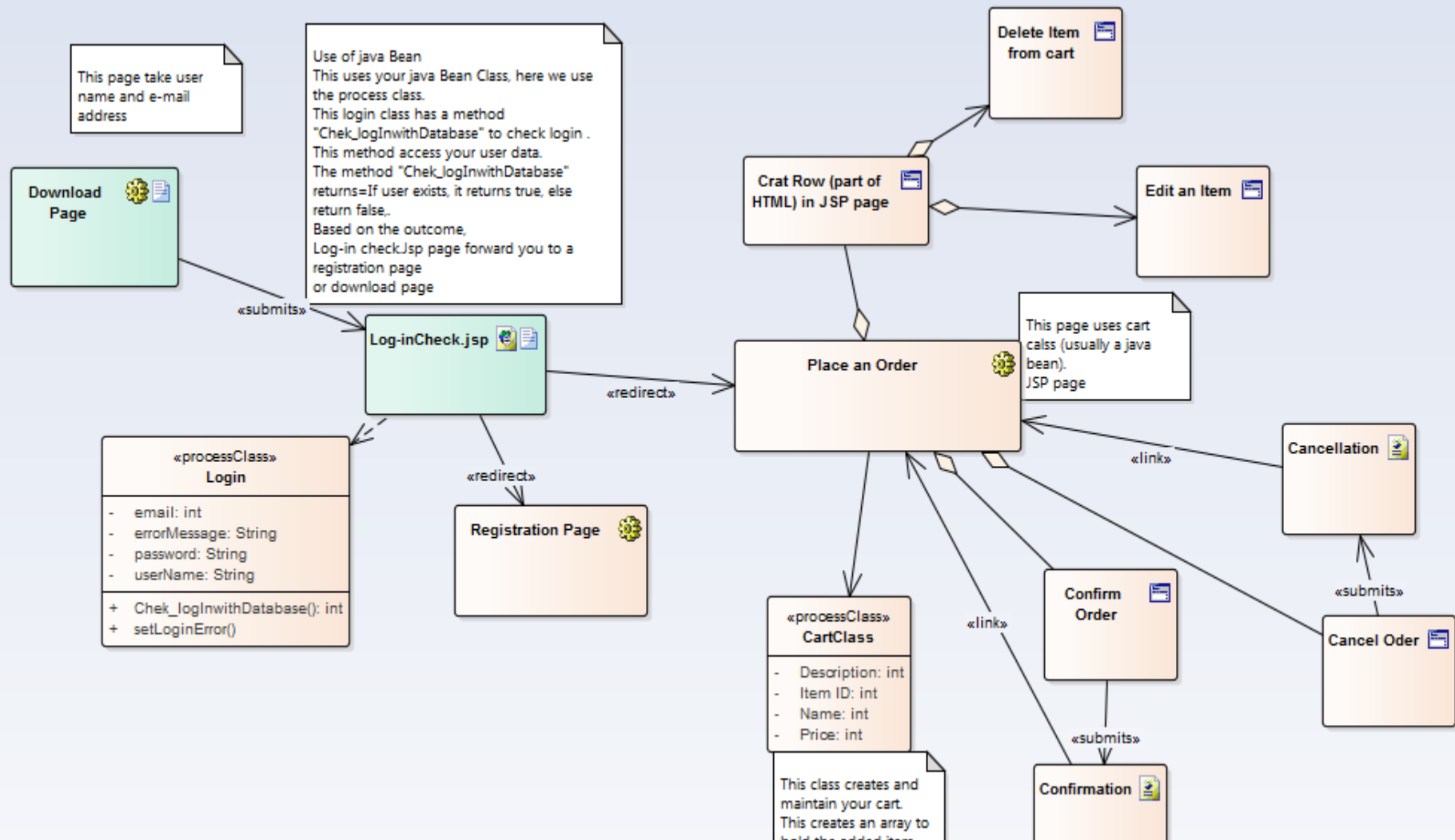
Yellow = our static main pages

(a) Example notation for use in blueprints. (b) High-level blueprint for **SafeHomeAssured.com**.



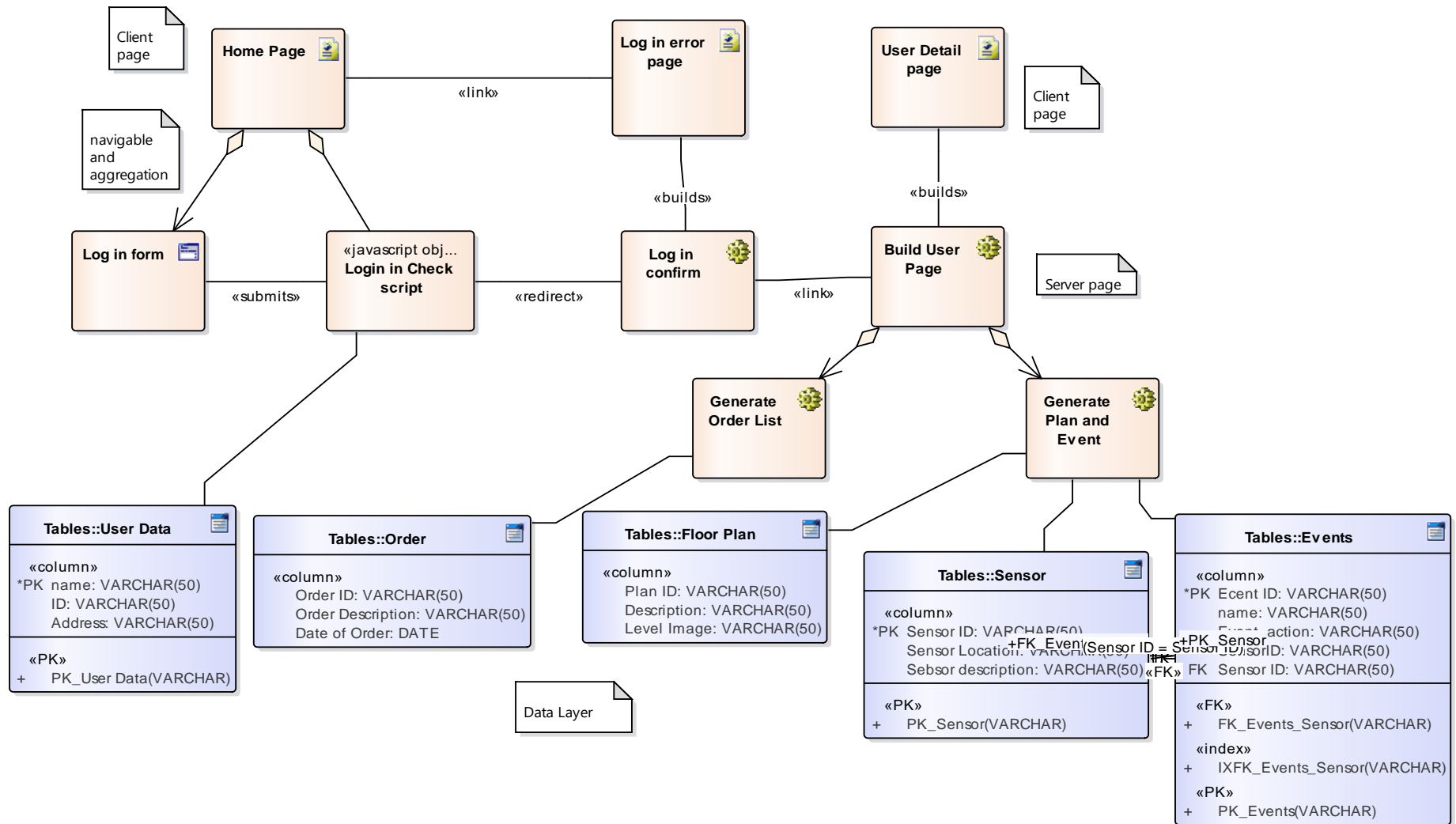
WAE model: Information and functional design detail

Music Portal Web Application



WAE model: functional design detail (safehome)

dm WAE_model



Summary

- ▶ Functional design does not occur in isolation!
 - ▶ It has strong connections to the analysis model, the information design, and the selection of the technologies that are being used to implement the designs.
- ▶ The functional architecture is strongly influenced by the information architecture.
- ▶ WAE provides a vehicle for considering how we weave detailed navigation paths together with the functionality required to generate the pages in a path.
- ▶ WebML allows a designer to model complex process workflows in a manner that will lead to more effective functional design.



Additional Readings: References

R. S. Pressman and D. Lowe: *Web Engineering, A Practitioner's Approach*, McGraw-Hill, 2009.

- Chapter 11: Functional Design

(concentrate on the topics covered in the lecture)

[1] Building Web Applications with UML (2nd Edition) Oct 13, 2002,
by Jim Conallen

References:

http://www.ibm.com/support/knowledgecenter/SS6RBX_11.4.3/com.ibm.sa.oomethod.doc/topics/c_Web_app_Extensions_WAE.html.

