WORCESTER POLYTECHNIC INSTITUTE

Database Management Systems Course – CS 542

Class Project Proposal

Online Classroom Management Application (OCMA)

BY RIMMA KAFTANCHIKOVA AND JOHN PERCY

OCMA – Online Class Management Application: Design and Implementation

Efficiency is doing better what is already being done. **Peter F. Drucker**

The Summary

This proposal describes the design and architectural decisions for the OCMA application.

What is OCMA?

We are endeavoring to build a complete, database-driven web application that will be helpful to teachers, and students in their every day work. Online Classroom Management Application (OCMA) would enable teachers to prepare courses containing web pages, forums, calendar, polls, online assessment and student record management. Students will be able check their grades online, post questions on the class forums, get automated emails about latest news and homework.

Key features:

- A logical, 3-tier architecture
- Security using Forms Authentication
- Retrieval of user account information from Database
- Data Grid with custom in-line editing
- Dynamic chart image creation with GDI+
- Microsoft Data Access Application Blocks for .NET
- Dynamic report creation
- Lightweight data classes used for the interface between the business logic layer and user interface layer

- Role-based security using a custom principal for role authorization

Goals

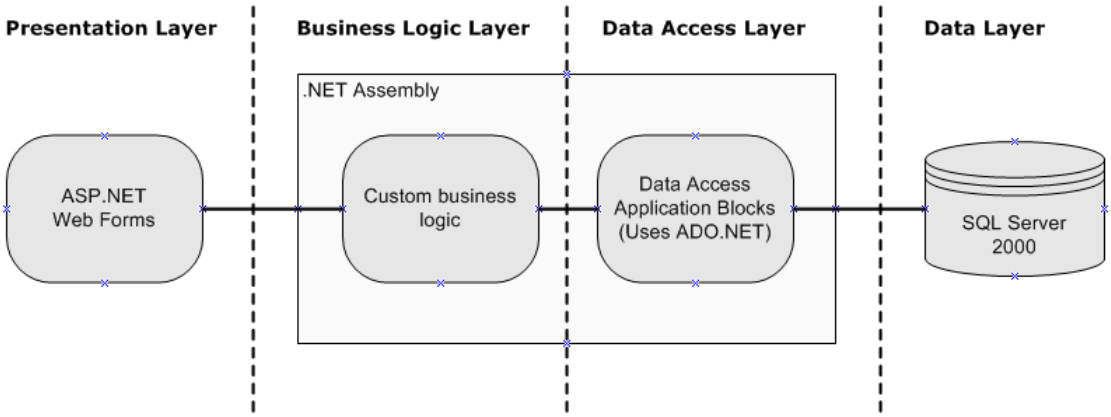
Since the OCMA is a line of business sample application that would typically be used over the Internet, a different set of challenges are presented to the developer:

- **Emphasis placed on maintenance instead of overall performance**: The application should be easy to manage. Adding new users to the application should be seamless. The OCMA application will use Forms authentication (i.e. users must sign in using their username and password). Once the user provides correct username and password, they are authorized to use the OCMA application. Anonymous users are allowed to view a main page, but not allowed to access other functionalities.

- **Clean separation between logical tiers**: In an Internet environment, it is likely that particular tiers or layers of the application will be shared or reused. The OCMA application will achieve this by using lightweight objects as the interface between the UI and business logic layer.

Application Architecture

The OCMA application will be an ASP.NET web application that will use a logical, 3-tier architecture.



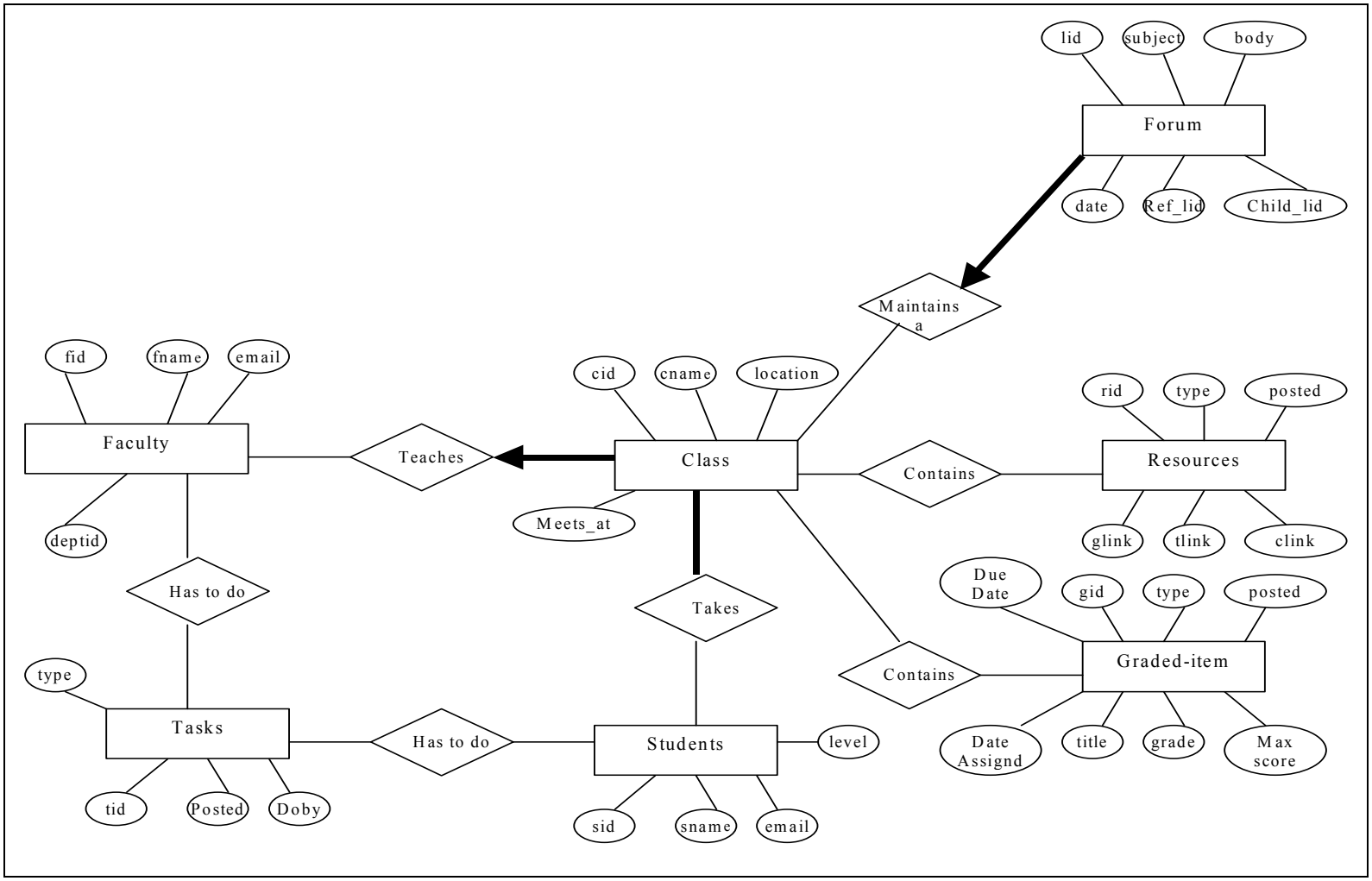
Data access, business logic, and user interface logic will be separated into different classes. N-tier architectures have many advantages, including:

- There is a clean separation between the user interface, business logic, and data access layers. This isolation promotes code re-use and makes maintaining and enhancing the code easier.
- Business rules are centralized into one component that is easy to re-use.
- Data access code is centralized in one place making development and maintenance easier.

Data Layer

The OCMA application will use a Microsoft SQL Server 2000 database.

ER Diagram



Relational Schemas

Faculty

Schema: Faculty(fid integer, fname string, dept_id integer, email string);

Table Field	Meaning
Fid	Unique ID for a faculty member
Fname	Unique name of a faculty member
Dept_id	Id of dept faculty member works in
Email	Email address of faculty member

Students

Schema: Students(sid integer, sname string, level integer, email string);

Table Field	Meaning
sid	Unique ID for a student
sname	Unique name of a student
level	Level(grade) of student
Email	Email address of student

Class

Schema: Class(cid integer, cname string, location string, meets_at integer, fid, integer);

Table Field	Meaning
cid	Unique ID for a class
cname	Unique name of a class
location	Location (room) of a class
Meets_at	Meeting time of a class
fid	Unique ID for a faculty member who teaches class

Tasks

Schema: Tasks(tid integer, tstring string, posted_date string, do_by_date string);

Table Field	Meaning
tid	Unique ID for a task
tstring	Description of a task
Posted_date	Date task was posted
Do_by_date	To be done date of a task

Resources

Schema: Resources(rid integer, type integer, rstring string, posted string, grade_link gid, task_link tid, class_link cid);

Table Field	Meaning
rid	Unique ID for a resource
Type	Type of resource
posted	Date resource was posted
String	Description of resource
Grade_link	Graded item ID resource is linked to
task_link	Task ID resource is linked to
class_link	Class ID resource is linked to

Graded_Item

Schema: Graded_Item(gid integer, type integer, title string, grade integer, max_score integer,
date_assigned string, date_due string, sid integer);

Table Field	Meaning
gid	Unique ID for a graded item
Type	Type of graded item
Title	Description of graded item
Grade	Grade received on graded item
Max_score	Graded item max possible score
Date_Assigned	Date graded item was assigned
Date_Due	Date graded item was due/done
Sid	SID of student who did graded item

Enrolled

Schema: Enrolled(sid integer, cid integer)

Table Field	Meaning
Sid	SID of student enrolled in class
cid	CID of class enrolled in

Forum_Listing

Schema: Forum_listings(lid integer, subject string, body string, date integer, ref_lid integer, child_lid integer)

Table Field	Meaning
lid	Unique ID for a forum listing
Subject	Subject description of listing
Body	Body text of listing
Date	Date listing was posted
Ref_lid	Parent listing of the listing
Child_lid	Child listing of the listing

Stored Procedures

The OCMA application will use stored procedures to encapsulate all of the database queries. The use of stored procedures provides a clean separation between the database and the data access layer. There are performance benefits by using stored procedures because they are optimized the first time they are run and then retained in memory for subsequent calls. Strongly-typed parameters in stored procedures and the ability to set permissions on each stored procedure result in improved security. Furthermore, the user accessing the stored procedures only needs rights to the stored procedures and not the underlying tables.

Depending upon the type of changes made to tables in the database, stored procedures can be modified without requiring changes to the data layer.

Data Access Layer

A data access layer will be used to encapsulate database-specific code, separating the details of the database from the business logic layer. The Microsoft Data Access Application Blocks (DAAB) will be used to implement the data access layer.

The majority of the methods in the data access component will follow a very similar pattern. The pattern is outlined as follows:

1. Create Connection to the Database
2. Create Command Object
3. Set Command Type to Stored Procedure
4. Create and Populate Parameters
5. Execute the Command
6. Close the Connection

Business Logic Layer

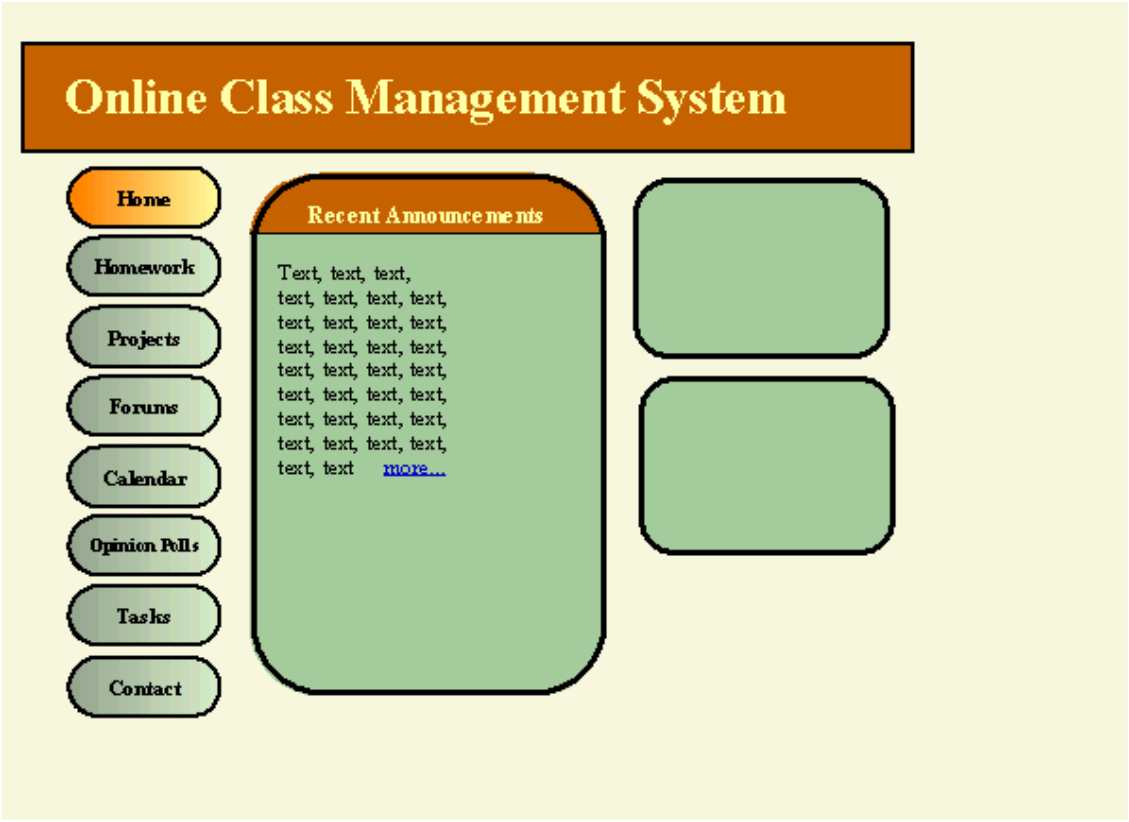
The business logic layer will separate the code specific to the application from the user interface and the database-specific code.

The OCMA will use lightweight classes to wrap information that is returned from the database access layer.

Presentation Layer

The presentation layer will be responsible for the user interface and will communicate directly with the business logic layer. Separating the presentation layer from the rest of the application will enable the development of different user interfaces (i.e. Web Forms, Windows Forms, Mobile Devices) that all use the same business logic and database access code (Who knows maybe in the future students might want to access their grades from a cell phone ☺).

Here is an example of a possible user interface (Student Version):

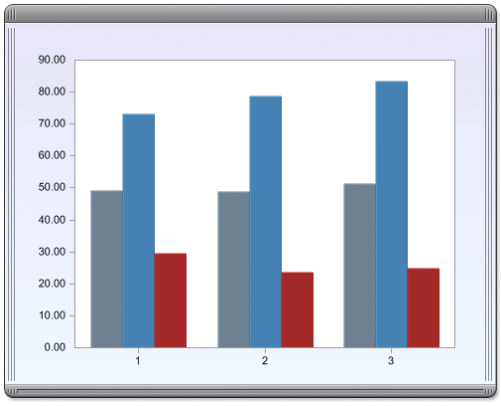


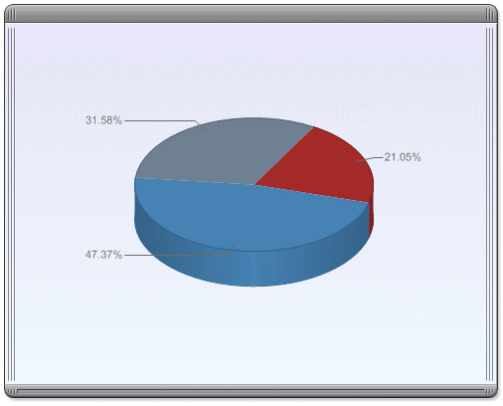
Security

The security design makes use of both authentication and authorization. Authentication is the process in which the application verifies a user's identity and credentials. Authorization will actually verify the authenticated user's permissions for a requested resource. These checks prevent a user from accessing functionality they should not be able to access. Similarly, a user's role in the OCMA application determines which tasks they can and cannot do. For instance, only a teacher can view the grades for the entire class.

Visual Report

Students and teachers will be able to view some information (grades and other statistics) in visual form in three different views: Pie Chart, Bar Graph, and Tabular. This report will either use GDI+ to dynamically generate pie charts and bar graphs. (GDI+ is an infrastructure for creating images and graphics in .NET.), or another option might be using ChartFX controls for .NET (This will be decided later).





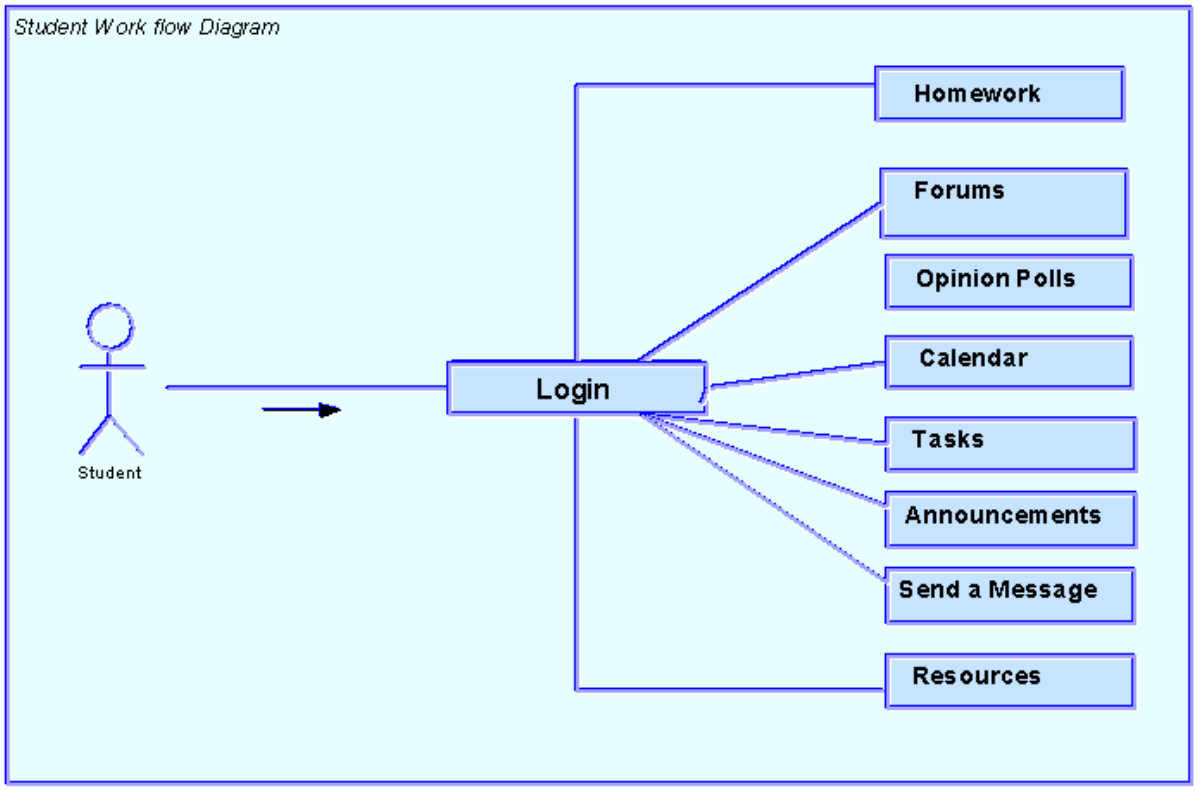
Areas of Functionality

The areas of functionality are listed below in no particular order:

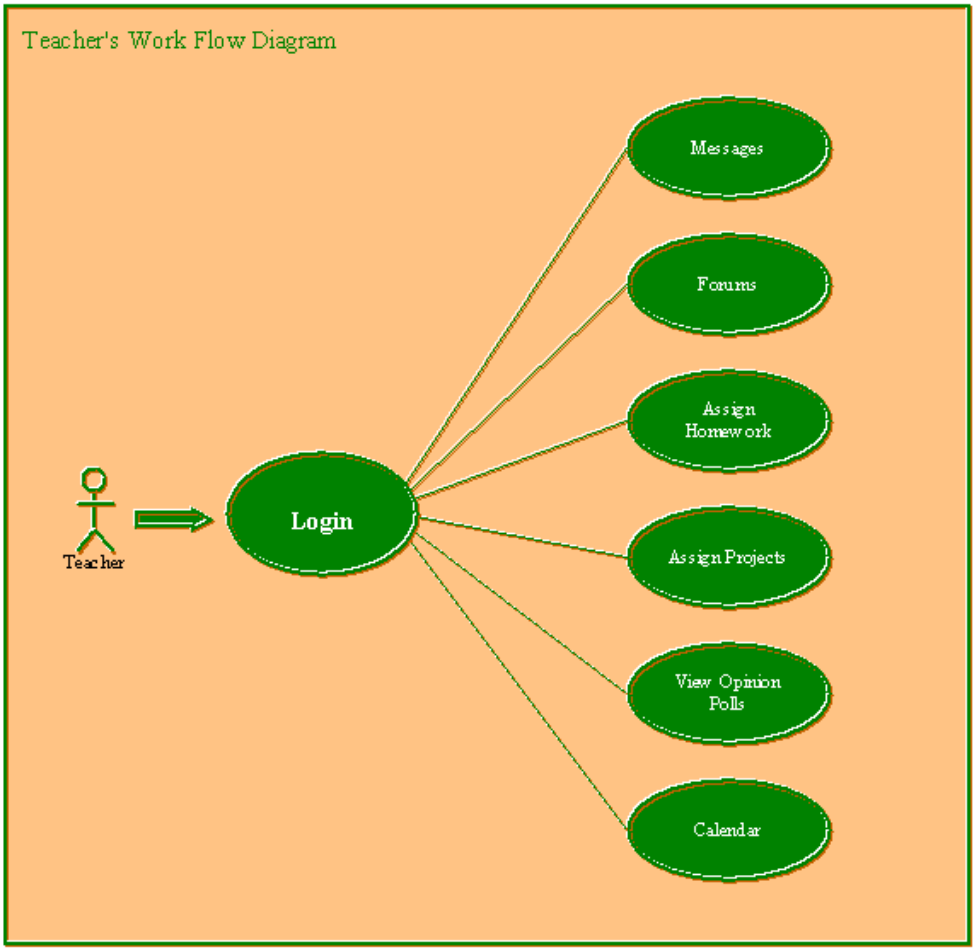
- Email newsletter
- News and class content system
- Opinion polls (voting)
- Forums

- Users and security
- Web site basics (homepages, navigation, headers, footers, etc.)
- Administration (uploading changes to files)

Student Workflow



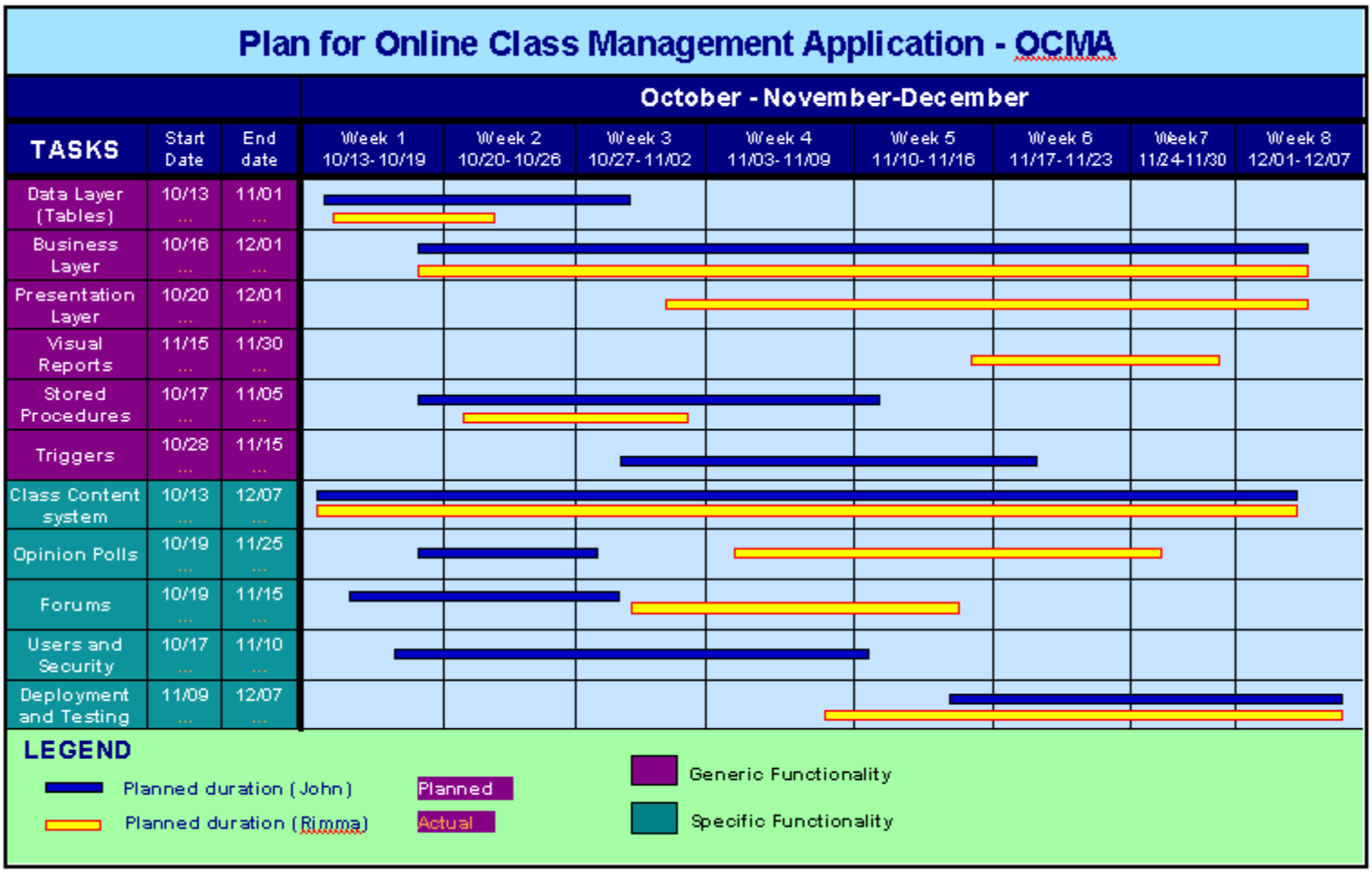
Teacher Workflow



Environment and Software Tools

- Visual Studio.NET (7.0)
- SQL Server 2000
- Adobe Photoshop (for graphics)
- ChartFX (maybe) for graphs

Plan of Deliverables



Background Material and Required Reading

Books:
ASP.NET Website Programming: Problem - Design - Solution, C# Edition
Professional C#, 2nd Edition
Professional SQL Server 2000 Programming
Professional ASP.NET 1.0, Special Edition
…and many more

Websites:
http://www.asp.net
http://www.DotNetJunkies.com
http://DevAsp.net

http://msdn.microsoft.com/netframework
http://www.programmersheaven.com/
http://www.4guysfromrolla.com/
…and many more

For More Information

For more information contact

Rimma at rimma3@hotmail.com or
John at john-and-susan@juno.com