

AcProMS

Academic Project Management System



By
Muhammad Ali (Group Leader)
Daud Azim Khan
Haider Ali

Supervisor:
Asst. Prof Dr. Awais Majeed

Submitted to the Faculty of Computer Software Engineering
National University of Sciences and Technology, Rawalpindi in partial fulfillment for the
requirements of a B.E Degree in Computer Software Engineering

June 2012

ABSTRACT

The quest for quality in education highlights the role of planning and management. Hence educational programs, like Under-graduate and Master, may benefit from implementation of an academic project management system. In order to achieve that, it is reasonable to form a common framework that provides a basic structure, which would be useful for acquiring data and producing statistics to support the program's planning and management. The proposed system intends to be the framework that implements this basic structure. Moreover the management of various activities related to academic projects (UG Final year Project and MS Thesis) is carried out manually. There is no Information System available that can manage and keep record of related activities, deadlines/milestones and deliverables associated with these projects. Moreover, the communication and collaboration between the various stakeholders is difficult.

Our Project presents a complete web system that can be applied to support the educational planning and management demands of undergraduate and postgraduate final year projects and theses.

Supervisors will upload their projects with details. They will also mention the innovation level, and level of difficulty. They will also include description of the project, required deliverables, and skills (mathematical, programming etc.) and the required tool(s)/tech(s). Allocation of projects can be based on some algorithm or first-come-first serve basis in case more than one groups apply for a specific project. Students (their groups) will be given choices for more than one project. Out of these choices, a project will be allocated to them. Once a project is assigned, a respective workspace will be created for the syndicates where they can communicate and collaborate with other group members and the supervisor.

CERTIFICATE

Certified that the contents and form of project report entitled “**Academic Project Management System**” submitted by 1) Muhammad Ali, 2) Daud Azim Khan, and 3) Haider Ali have been found satisfactory for the requirement of the degree.

Supervisor: _____

Asst. Prof. Dr. Awais Majeed

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To our parents, without whose unflinching support and unstinting cooperation, a work of
this magnitude would not have been possible

ACKNOWLEDEMENTS

We are eternally grateful to Almighty Allah for bestowing us with the strength and resolve to undertake and complete the project.

We gratefully recognize the continuous supervision and motivation provided to us by our Project Supervisor, Asst Prof. Dr. Awais Majeed from Military College of Signals, NUST.

We are highly gratified to our committee members, Asst Prof. Dr. Hammad Afzal and Lec. Bilal Rauf for their continuous and valuable suggestions, guidance, and commitment. We are highly thankful to all of our professors whom had been guiding and supporting us throughout our course and research work. Their knowledge, guidance and training enabled us to carry out this project.

We would like to offer our admiration to all our classmates, and our seniors who had been supporting, helping and encouraging us throughout our project. We are also indebted to the MCS system administration for their help and support.

We deeply treasure the unparalleled support and tolerance that we received from our friends for their useful suggestions that helped us in completion of this project.

In the end we are also deeply obliged to our families for their never ending patience and support for our mental peace and to our parents for the strength that they gave us through their prayers.

Table of Contents

Introduction.....	2
1.1 Introduction	2
1.2 Problem Statement.....	2
1.3 Motivation	3
1.4 Proposed Solution.....	3
1.5 Goals and Objectives	5
1.5.1 Academic Objectives	5
1.5.1 Academic Objectives	5
1.6 Deliverables	5
1.7 Summary.....	6
Literature Review	8
1. Introduction	8
2. Background & existing work	8
2.2.1 4PM (For Project Management)	8
2.2.2 dotProject	9
2.2.3 Mantis Bug Tracker (MantisBT)	9
2.2.4 PHPProjekt	10
2.2.5 CA Clarity PPM.....	10
2.2.6 An Academic Project Management Web System	11
2.2.7 Version Control System	11
3. Summary.....	14
System Requirements Specification	16
3.1 Introduction	16
3.2 Use-Case Diagram	17
3.3 Scope of the Project.....	18
3.4 Stakeholders	18
3.5 Users of Product	18
3.6 Relevant Facts and Assumptions	19
3.7 Functional Requirements.....	19
3.7.1 Administrator Functionalities.....	20
3.7.2 Course Advisor Functionalities	25
3.7.3 Supervisor Functionalities	31

3.7.4	<i>Student Functionalities</i>	37
3.7.5	<i>Project Coordinator Functionalities</i>	42
3.8	<i>Non-Functional Requirements</i>	46
3.9	<i>Hardware Requirements</i>	47
3.10	<i>Tools and Technologies Requirements</i>	47
3.11	<i>Summary</i>	47
System Design		50
4.1	<i>Introduction</i>	50
4.2	<i>System Architecture</i>	51
4.3	<i>Detailed Design</i>	54
4.4	<i>Summary</i>	83
System Implementation		86
5.1	<i>Introduction</i>	86
5.2	<i>Implementation</i>	87
5.3	<i>Implementation Language</i>	88
5.4	<i>ASP.NET Page Life Cycle</i>	89
5.5	<i>Distribution of Modules</i>	93
5.6	<i>Distribution of Classes with respect to Modules</i>	96
5.7	<i>Summary</i>	98
Software Testing		100
6.1	<i>Introduction</i>	100
6.2	<i>Functional Testing</i>	100
6.3	<i>Summary</i>	105
Conclusion and Future Enhancements		107
7.1	<i>Introduction</i>	107
7.2	<i>Conclusion</i>	107
7.3	<i>Future Enhancements of AcProMS Functionalities</i>	108
7.4	<i>Summary</i>	108
Bibliography		109

LIST OF FIGURES

Figure	Page Number
1-1 Proposed Solution.....	6
3-1 Use-Case Diagram.....	19
4-1 Architecture Diagram.....	55
4-2 Abstract Use-Case Diagram.....	57
4-3 Project Management use-case Diagram	59
4-4 Syndicate Management use-case Diagram	60
4-5 Workflow Management use-case Diagram	60
4-6 Class Diagram	62
4-7 ER Diagram	63
4-8 State Transition Diagram	64
4-9 Login Sequence Diagram	65
4-10 Add Proposal Sequence Diagram	66
4-11 Chose Project Sequence Diagram	67
4-12 Create Activity Sequence Diagram	68
4-13 Submit Deliverable Sequence Diagram	69
4-14 Supervisor's View Hypertext Model	71
4-15 Course Advisor's View Hypertext Model	72
4-16 HoD's View Hypertext Model	73
4-17 Project Coordinator's View Hypertext Model.....	74
4-18 Student's View Hypertext Model	75
4-19 Supervisor's View Access Model	76
4-20 Course Advisor's View Access Model	77
4-21 HoD's View Access Model	78
4-22 Project Coordinator's View Access Model	79
4-23 Student's View Access Model	80

4-24 Home Page Presentation Model	81
4-25 Student Profile Page Presentation Model	82
4-26 Submit Deliverable Page Presentation Model	83
4-27 Create Activity Page Presentation Model	84
4-28 Add Proposal Page Presentation Model	85
5-1 Implementation structure.....	89
5-2 Master pages.....	90
5-3 Page Life Cycle.....	94

CHAPTER 1

Introduction

Introduction

1.1 Introduction

This chapter provides an introduction to its readers, future developers, software testers and project evaluators about the problem statement, proposed solution for this project, an overview of the system to be developed, motivation behind doing this project, scope of work in order to accomplish project objectives, research domain in which this project falls and summary of project objectives and goals.

Academic project management web system can be applied to support the educational planning and management demands of undergraduate and postgraduate final year projects and theses.

1.2 Problem Statement

The management of various activities related to the academic projects (UG Final year Project and MS Thesis) is carried out manually. The activities related to these projects are planned and executed manually and there is no Information System available that can manage and keep record of related activities, deadlines/milestones and deliverables associated with these projects. Moreover, the communication and collaboration between the various stakeholders is difficult. In order to achieve that, it is reasonable to form a common framework that provides a basic structure, which would be useful for acquiring data and producing statistics to support the program's planning and management.

1.3 Motivation

At the moment there is no Information System available through which the project synopsis can be published and made available to students for selection. Moreover, selection and allocation process is manual. Similarly, there is no repository for the previous projects that can be accessed for further information.

Academic projects involve different activities. The management of these activities is difficult and time consuming. An IT based solution can help in the management of these activities and can act as a repository of knowledge for the future projects. Hence educational programs, like Under-graduate and Master, may benefit from implementation of an academic project management system. In order to achieve that, it is reasonable to form a common framework that provides a basic structure, which would be useful for acquiring data and producing statistics to support the program's planning and management. The proposed system intends to be the framework that implements this basic structure.

1.4 Proposed Solution

The proposed solution is a complete web application that can be applied to support the educational planning and management demands of undergraduate and postgraduate final year projects and thesis. The overall system diagram is shown in Figure 1.1.

System Overview

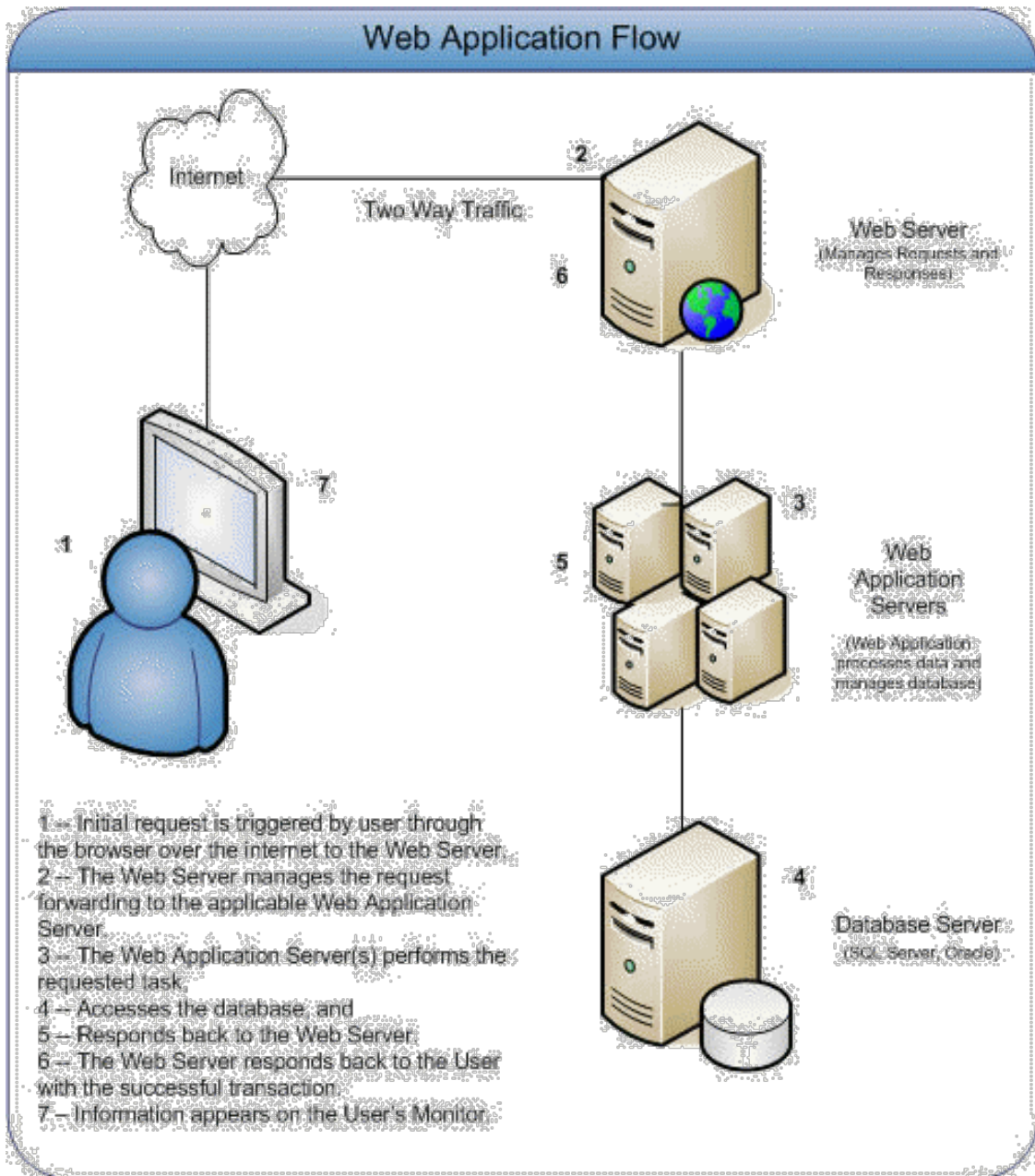


Figure 1-1: Proposed Solution

1.5 Goals and Objectives

1.5.1 Academic Objectives

Key objective of this project is to impart the knowledge about the development of Web Applications. It includes the architectural design, detailed design, implementation, usability analysis of the system and how such Information Technology systems can help in community building and collaboration.

1.5.1 Academic Objectives

A complete Web application supporting the publishing, allocation and collaboration for final year projects. Our project has following objectives which needed to be fulfilled. The objectives include are 1) The development and deployment of full-fledge project management system for final year academic projects and theses efficiently and effectively, so that the students produce maximum output in minimum time. 2) The policies for the UG and PG projects will be exploited at maximum level upholding transparency at all levels. 3) The policy CGPA range for FYP group will be fulfilled and applied without exceptions. 4) The project will make use of the latest web technologies available.

1.6 Deliverables

We will provide Software Requirements Specification (SRS), which will include all functional and non-functional requirement of system. Analysis & Detailed Design document will describe the low level and high level design of system. At end there will a complete working system along with project report and user manual.

1.7 Summary

This chapter described the details about problem statement its objectives and project goals to be implemented. Moreover it highlighted an overview of the project scope of work, its structural diagram and the key parameters which needs to be considered. Research work related to the project and the main objectives and structural analysis has been discussed.

CHAPTER 2

Literature Review

Literature Review

1. Introduction

In this chapter different project management systems along with their basic objectives and short comings are discussed. There are different types of project management software systems or we can say different approaches including desktop, web-based, personal, single user, collaborative, & integrated.

2. Background & existing work

Many project management tools are available in the market. A brief description of these products is given in the following sections.

2.2.1 4PM (For Project Management)

2.2.1.1 Introduction

4PM is a web-based information system for project management and collaborative teamwork. It is written in the *Java 2 Platform, Enterprise Ed.* It uses advanced dynamic web interfaces. It provides multi-project management tailored to different project types in many areas like architecture, government agencies, educational institutes, co-financed projects, etc.

2.2.1.2 Features

Features of 4PM include project finance planning & monitoring, project planning & realization monitoring, resource allocation management, report generation, project's

analytical overviews, restricted user access, online document repository, multilingual support, templates.

2.2.2 dotProject

2.2.2.1 Introduction

dotProject is a web-based, multilingual, multiuser project management system. It is open-source and free application, developed in PHP, and initiated in 2000. Because of the fact that the dotProject is ultimately a project management tool, one can fit it to any scheduled workload. dotProject is similar to MS Project in sense that it also displays the timeline in Gantt chart. It differs from MS Project by being a web-based.

2.2.2.2 Features

Major features of PHPProjekt include Group Calendar, Project Management, Time Card System, File Management system, Contact Manager, Mail Client etc.

2.2.3 Mantis Bug Tracker (MantisBT)

2.2.3.1 Introduction

Mantis Bug Tracker is an open-source, web-based bug-tracking system. It is commonly used to detect software defect but it can be configured to work as an issue tracking system and project management system. It uses PHP scripting language, MySQL and PostgreSQL (experimental) as database system and Apache web server.

2.2.3.2 Features

It supports plug-ins, notifications and version controlling. Beside these minor features are Audit trails, revision control, roadmaps'changelogs, graphing of relationships between issues and many other.

2.2.4 PHProjekt

2.2.4.1 Introduction

PHProjekt is a free groupware, cross-platform, project management system. It is the only freeware which includes SimpleScripts script installerSimpleScripts is an installer for blogs, photo galleries, and forums. It is a modular application for the organization and management of group activities and to share information and various types of documents via web.PHProjekt supports many protocols like LDAP, SOAP and WebDAV (Web-based Distributed Authoring and Versioning) and is available for many languages and databases.

2.2.4.2 Features

Major features of PHProjekt include Group Calendar, Project Management, Time Card System, File Management system, Contact Manager, Mail Client etc.

2.2.5 CA Clarity PPM

2.2.5.1 Introduction

CA Clarity PPM is a project and portfolio management software. It is primarily used by large corporations across all industries to help manage projects, products, and services.

CA Clarity PPM has a strong presence in the IT Governance, New Product Development, Professional Services Automation, and US Federal markets.

2.2.5.2 Features

Portfolio Management: Allows organizations to adjust resources and investments according to corporate goals.

Project Management: Provides full project management functionality and best practice methodologies.

Other features include resource management, demand Management, project financial management, and process management etc.

2.2.6 An Academic Project Management Web System

2.2.6.1 Introduction

A research paper was published by P. Letouze, R. Ronzani, and A. Oliveira in Brazil at Federal University of Tocantins (UFT) (2010), entitled “An Academic Project Management Web System Developed through a Software House Simulation in a Classroom” [14] in which they explain their endeavor of developing an Academic Project Management Web System, initiating through a class simulation or mock-up model.

2.2.7 Version Control System

2.2.7.1 Introduction

Version Control System (VCS), also known as Revision Control or Source Control system, generally speaking, is a software system that helps user to track files over time.

Technically, it is a combination of various technologies and practices for tracking and controlling changes to files of a project e.g. source code, documentation, and web pages.

2.2.7.2 Terminologies

Repository

It is the database where the changes in a project are stored. Some VCSs are centralized that is, they store all the changes in single repository called **master repository**, while other are decentralized i.e. each developer has his own repository, and changes can be swapped between repositories randomly.

Checkouts

Checkout is a process of obtaining a copy of the project from a repository. A checkout usually produces a directory tree called a "working copy" which is a private directory tree of a developer containing source code files, and may also containing its web pages or other document, from which changes may be committed back to the original repository at any given time. In some decentralized version control systems, each working copy is itself a repository and changes can be pushed out to any repository.

2.2.7.3 Apache Subversion or SVN

Apache Subversion or SVN is currently the most popular centralized VCS, that is, it has a master repository. Clients can check-out the repository on their local machines, and after making changes, they can check-in the changes to the server. The latest stable version is 1.6.17. It is written in C programming language. It is open-source, cross-platform and has Apache license.

2.2.7.4 Git

Git is a distributed version control system or Decentralized VCS which means each developer has its own repository or copy of system or files. Distributed means that instead of running 'svn checkout (url)' to get the latest version of the repository, with Git you run 'git clone (url)', which gives user a complete copy of the entire history of that project. This means that immediately after the clone, there is basically no information about that project that the server one cloned from has that one do not have. The latest stable version is 1.7.6.4. It is open-source with GNU license.

So the sharp difference between SVN and Git is; in Subversion only the central repository stores the complete history while in Git all clients have all the information.

Git has an advantage over SVN because of its distributive nature, being less prone to loss of data. But it has a disadvantage also as it is more unsecure than SVN because of many copies of data while SVN is centralized and secure.

Limitations of existing systems:

The above described Project Management software are web-based, few open-source and freely available for Windows as well as Linux operating systems while some are SaaS or proprietary. However, none of these is an Academic Project Management software system, which is currently the need of all the major universities across the world. These are all generic project management systems and may or may not be modified into Academic project management systems (depending on license and extend to which these can be customized and modified) whereas the main focus of AcProMS will be academic projects management.

3. Summary

Literature overview has been discussed in this chapter. Different methodology and description of techniques have been discussed. It described the already built web-based project management systems and details of the technologies used to build them. We found that the all these systems provide general support for project management and are not designed for academic project management. These systems are complex, difficult to customize and need expertise to manage them. The study of CVS systems revealed that it is a good idea to employ a version control system in the system. We found Subversion version control system to be compatible with our needs.

CHAPTER 3

System Requirements Specification

System Requirements Specification

3.1 Introduction

Software requirements specification for **AcProMS** comprising of functional and non-functional requirements are discussed in this chapter. Use-case diagrams with user specific descriptions highlight requirement elicitation and analysis. Graphical user interface requirements for **AcProMS** enhances the application aesthetic look and user's ease of interaction with **AcProMS** application. Non-functional requirements describes the application security measures, stress handling, congestion control and other quality attributes which must be fulfilled in order to achieve robust and efficient application.

This System Requirement Specification Document (SRS) is the output report of Requirement gathering and Analysis phase. This document is developed in a way that it details the software product specifications in terms of its scope, requirement and Objective. Project Evaluation Team, AcProMS's Project Manager, Team Leader and developers, AcProMS's Test Engineers and The Users of AcProMS can get benefit and consult this part of document.

3.2 Use-Case Diagram

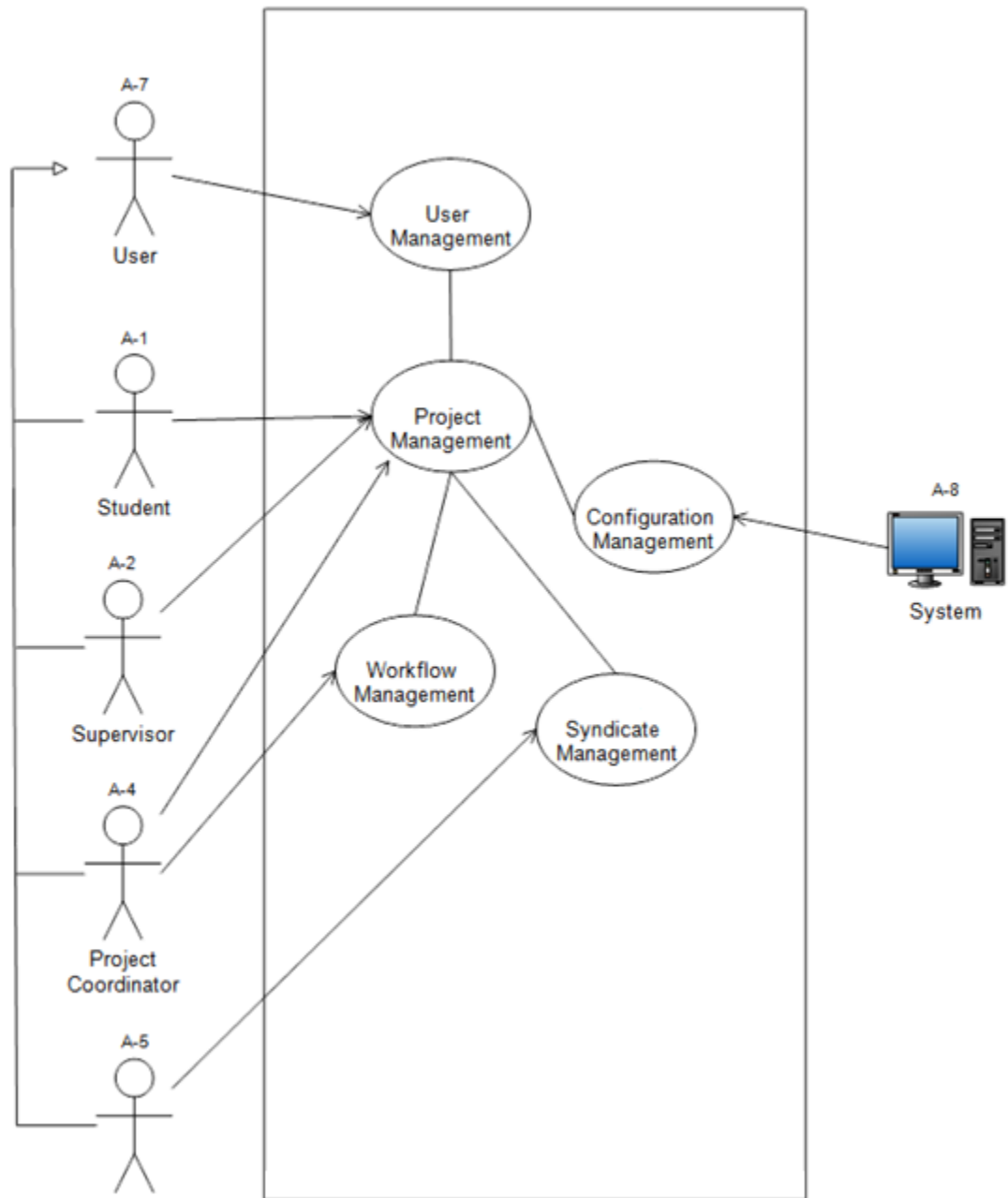


Figure 3-1: Use-Case Diagram

3.3 Scope of the Project

The scope of this project is restricted to various activities associated with the management and execution of projects assigned to students as part of the requirements for a graduate or postgraduate degrees offered in the engineering disciplines at National University of Sciences & Technology (NUST) under the given rules and regulations. The proposed system will not implement the configuration management as it is used for the software project for change management and version control of the source code/binary files. Rather the system will incorporate these features by integrating the existing tools like Visual SourceSafe, CVS or SVN. The complex project management activities like resource and cost management will not be covered by the proposed system as well.

3.4 Stakeholders

3.4.1 Clients

Our main client is Department of CSE, Military college of Signals.

3.4.2 Project Stack-holders

Faculty of Engineering, departments under the faculty, head of departments, project coordinators, supervisors, program coordinators (training officers), students and the development team are the main stakeholders of the system.

3.5 Users of Product

3.5.1 System Administrator

System administrator will manage the whole system and will assign different roles to different users.

3.5.2 Head of Department

HoD will use the system to view the progress of project activities and different things will be approved from HoD.

3.5.3 Supervisors

Supervisors are one of the main users of the system. Supervisor will upload the projects and then he/she, after allocation of project to some group, will manage the all activities of project online.

3.5.4 Students

Students will select project and then will make communication with their supervisor.

3.5.5 Project coordinator

Project coordinator will manage all project activities.

3.6 Relevant Facts and Assumptions

The system will be deployed on MIS servers. The system is a Web based project management system. Version control features for the software projects will be provided through the integration of third party tools.

3.7 Functional Requirements

A **functional requirement** defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. **AcProMS** provides the user with the following functions:

3.7.1 Administrator Functionalities

3.7.1.1 Login

Using this feature of system, Administrator can log in to application and can further user other features of system.

3.7.1.1.1 Description

Application home page has a login box which takes user name and password to login the user to application. System validates the user credentials and redirects the user to his/her home page according to his/her role.

3.7.1.1.2 Basic Flow

Actor Actions	System Response
User redirect to home page of application	A form will be displayed with username and password fields
User fills the required fields and clicks on submit button	System will show web page according to user's roles and access.

3.7.1.1.3 Alternative Flow

1. User leaves the required fields empty.
2. Username and password are not verified. Invalid username and password.

3.7.1.2 User Registration

This Feature provides the ability to administrator to create accounts of different users other than students. Administrator can create account of Supervisors, Course Advisors, Project Coordinate and HOD.

3.7.1.2.1 Description

At administrator profile page there will be a link for user registration to add user information to create account for that user. On Clicking create user button system will save user data to database and now that particular user can access the application functionality.

3.7.1.2.2 Basis Flow

Actor Actions	System Response
User clicks on the link "User Registration"	System will display a web form with the "Name, User Name, Password, Email address, Role, Department" fields.
User fills the required fields and clicks on submit button	The form will be submitted for further processing and on successful registration a web page informing user account has been created.

3.7.1.2.3 Alternative Flow

1. User Leave some field empty.
2. Administrator clicks on close window button.
3. Administrator logged out.

3.7.1.3 Add New Department/Course

Administrator can add new department and course into database. This department and course information is used in other functionalities of system.

3.7.1.3.1 Description

Administrator will click on Add Department/Course link provided on his/her profile page. A web form having fields of department/course name will be displayed. Administrator will fill this information and on clicking button add, this information will be added to database.

3.7.1.3.2 Basis Flow

Actor Actions	System Response
User clicks on the link “Add Course and Department”	System will display a web form with the fields of Department Name and ShortName.
User fills the required fields and clicks on Add button	The form will be submitted for further processing and on successful Processing a web page informing department /course has been Added.

3.7.1.3.3 Alternative Flow

1. User Leave some field empty.
2. Administrator clicks on close window button.
3. Administrator logged out.

3.7.1.4 Modify Department/Course information

Administrator can modify department and course information.

3.7.1.4.1 Description

Administrator will click on Add Department/Course link provided on his/her profile page.

There is a section where all added department/course will be displayed. On clicking edit button system will fetch information from database. Administrator will modify this information and submit to database.

3.7.1.4.2 Basis Flow

Actor Actions	System Response
User clicks on the link “Modify Course and Dept.”	System will display a web form with the field of Department Name and ShortName. These fields will be populated from database.
User update the required fields and clicks on update button	System will add this updated information to database.

3.7.1.4.3 Alternative Flow

1. User Leave some field empty.
2. Administrator clicks on close window button.
3. Administrator logged out.

3.7.1.5 Add News

Administrator can add some news for other user of application.

3.7.1.5.1 Description

Administrator will click on Add News link provided on his/her profile page. A web form with required fields will be displayed. When user adds news system will save it in database and will display on home page of each and every user.

3.7.1.5.2 Basis Flow

Actor Actions	System Response
User clicks on the link “Add News.”	System will display a web form with the fields of Title, Description, and Attach file.
User fills the required fields and clicks on add button	System will add this information to database.

3.7.1.5.3 Alternative Flow

1. User Leave some field empty.
2. Administrator clicks on close window button.
3. Administrator logged out.

3.7.1.6 Functional Requirement

REQ-1: System should provide the user registration feature to create accounts for different types of users.

REQ-2: Access to various pages/features of the system should be restricted and will only be carried out after user authentication (user login) and authorization based on role assigned to the user.

REQ-3: System should provide the creation and customization of various roles. The system should support the following roles with the given responsibilities.

HoD	Approval of syndicates, project proposal, project activities.
Project Coordinator	Defining the activities of projects and deadlines for these activities.
Supervisor	Uploading the projects
Course Advisor	Create Account of students and make syndicate.

3.7.2 Course Advisor Functionalities

3.7.2.1 Login

Using this feature of system, Administrator can log in to application and can further user other features of system.

3.7.2.1.1 Description

Application home page has a login box which takes user name and password to login the user to application. System validates the user credentials and redirects the user to his/her home page according to his/her role.

3.7.2.1.2 Basic Flow

Actor Actions	System Response
User redirect to home page of application	A form will be displayed with username and password fields
User fills the required fields and clicks on submit button	System will show web page according to user's roles and access.

3.7.2.1.3 Alternative Flow

1. User leaves the required fields empty.
2. Username and password are not verified. Invalid username and password.

3.7.2.2 Student Registration

This Feature provides the ability to course advisor to create accounts of different students. Course advisor can create account through a web form or using a CSV file having fixed format.

File Format:

Registration Number, Name , Email, Username, Password

2008-NUST-BE-CSE-279,NCTanzeela Bilal, Tanzeela@mcs.edu.pk,Tanzeela,1234

2008-NUST-BE-CSE-01,NCRabia Sultan, RabiaSultan@gmail.com,RSA,1234

2008-NUST-BE-CSE-11,NC Muhammad Khan, khan@gmail.com,khan,1234

3.7.2.2.1 Description

At course advisor profile page there will be a link for user registration to add student information to create account for that student. On Clicking create user button system will save student's data to database and now that particular student can access the application functionality.

3.7.2.2.2 Basis Flow

Actor Actions	System Response
User clicks on the link "User Registration"	System will display a web form with the following fields of Name, User Name, Password, Email address.
User fills the required fields and clicks on submit button	The form will be submitted for further processing and on successful registration a web page informing user account has been created.
User uploads a CSV file using brows button.	System extracts information from file and creates accounts of students.

3.7.2.2.3 Alternative Flow

1. User Leave some field empty.
2. CSV file has some invalid entries.
3. User clicks on close window button.

4. User logged out.

3.7.2.3 Create Group

Course Advisor can create new Final year project group.

3.7.2.3.1 Description

Course Advisor will click on Add Create Group link provided on his/her profile page. A web form having fields of Group name will be displayed. Administrator will fill this information and on clicking button “create” this information will be added to database.

3.7.2.3.2 Basis Flow

Actor Actions	System Response
User clicks on the link “Create Group”	System will display a web form having field of group name.
User fills the required fields and clicks on “Create Group” button.	The form will be submitted for further processing and on successful Processing a web page informing Group has been Created.

3.7.2.3.3 Alternative Flow

1. User Leave some field empty.
2. User clicks on close window button.
3. User logged out.

3.7.2.4 Add Members to Group

Administrator can add different registered student to some already created group.

3.7.2.4.1 Description

Administrator will click on Add Members to Group link provided on his/her profile page.

System will display three boxes, one for group name other for unassigned students and third for assigned students.

On clicking some group name, system will display students name that are in that group.

User can add/ remove students from that group.

3.7.2.4.2 Basis Flow

Actor Actions	System Response
User clicks on the link “Add Member to group”	System will display a web page having three boxes, one for group name and other for unassigned student’s names and third for assigned student’s names.
User select some group name and can add/remove members to/from that group.	System performs respective action and adds/removes values from database.

3.7.2.4.3 Alternative Flow

1. User clicks on close window button.
2. User logged out.

3.7.2.5 Add News

Course Advisor can add some news for other user of application.

3.7.2.5.1 Description

Course Advisor will click on Add News link provided on his/her profile page. A web form with required fields will be displayed. When user adds news system will save it in database and will display on home page of each and every user.

3.7.2.5.2 Basis Flow

Actor Actions	System Response
User clicks on the link“Add News.”	System will display a web form with the fields of Title, Description, Attach file
User fills the required fields and clicks on add button	System will add this information to database.

3.7.2.5.3 Alternative Flow

1. User Leave some field empty.
2. User clicks on close window button.
3. User logged out.

3.7.2.6 Functional Requirement

REQ-1: System should provide the user registration feature to create accounts for students.

REQ-2:System should support the creation of syndicates (groups) by the authorized users (course advisor).

REQ-3: System should also support the changes of groups at any later stages if required.

REQ-4:User should be able to add or remove news.

3.7.3 Supervisor Functionalities

3.7.3.1 Login

Using this feature of system, Supervisor can log in to application and can further user other features of system.

3.7.3.1.1 Description

Application home page has a login box which takes user name and password to login the user to application. System validates the user credentials and redirects the user to his/her home page according to his/her role.

3.7.3.1.2 Basic Flow

Actor Actions	System Response
User redirect to home page of application	A form will be displayed with username and password fields
User fills the required fields and clicks on submit button	System will show web page according to user's roles and access.

3.7.3.1.3 Alternative Flow

1. User leaves the required fields empty.
2. Username and password are not verified. Invalid username and password.

3.7.3.2 Synopsis Publishing

This Feature provides the ability to Supervisor publish his/her project idea and this synopsis will be approved by respective authorities. After that these synopsis will be available to students to choose.

3.7.3.2.1 Description

At supervisor's home page there will be link of "add Synopsis". When user clicks that then there will be a web form which will take all required information from user and on clicking "Add" button system will save information to database.

3.7.3.2.2 Basis Flow

Actor Actions	System Response
User clicks on the link "Add Synopsis"	System will display a web form of format as defined by organization and supervisor will enter all required information. The form contains the information of Short title, Extended title, Brief description of the project, Scope of work, Academic objectives, Application goals,

	Previous work done on the subject, Required Resources, No of Students, Special Skills Required, Difficulty level
User enters the required information and presses the “Submit” button.	User fills the required information and on submits the information is added to the database in the relevant table with status “Submitted”. When a user will browse the list of the submitted projects the newly submitted project will be marked as “Submitted for Approval”.

3.7.3.2.3 Alternative Flow

1. User Leave some field empty.
2. User clicks on close window button.
3. User logged out.

3.7.3.3 Modify Proposal

Supervisor can modify some particular synopsis if required.

3.7.3.3.1 Description

Supervisor can modify some project proposal or he/she can delete some proposal if it is not selected by some group.

3.7.3.3.2 Basis Flow

Actor Actions	System Response
User clicks on the link “Edit”	System will display a web form all field of project synopsis and populate them from database.
User update information and clicks on update button.	System will update synopsis information in database.
User clicks on the link “Delete”	System will check if that proposal is not selected by some group and will not delete if selected by some group.

3.7.3.3.3 Alternative Flow

1. User Leave some field empty.
2. User clicks on close window button.
3. User logged out.

3.7.3.4 Assign Project

Supervisor can assign some particular project to one of the groups who chose that project.

3.7.3.4.1 Description

Supervisor can see all his/her projects that are chosen by some group. If some project is chosen by more than one group then it is on supervisor hand to assign that project to group to which he/she wants to. The assign project will be added to “projects in progress” list.

3.7.3.4.2 Basis Flow

Actor Actions	System Response
User clicks on the link “Selected projects”	System will display a web page having a list of projects chosen by some group with that group name.
User clicks on assign button.	System will assign that project to some particular group. System will add this project from available to choose projects list and add it into “projects in progress” list.

3.7.3.4.3 Alternative Flow

1. User clicks on close window button.
2. User logged out.

3.7.3.5 Work Space Creation

When a project is assigned to some group then a work space is created for that project through which supervisor can interact with his/her group.

3.7.3.5.1 Description

Supervisor can assign some task to some particular member of group. He/she also can add some resource file for his/her group. Complete information of group is also available at work space page.

3.7.3.5.2 Basis Flow

Actor Actions	System Response
User clicks on the link some particular selected project.	System will display a web page through which supervisor can perform task like Assign task or upload resource.

3.7.3.5.3 Alternative Flow

1. User clicks on close window button.
2. User logged out.

3.7.3.6 Add News

Supervisor can add some news for other user of application.

3.7.3.6.1 Description

Supervisor will click on Add News link provided on his/her profile page. A web form with required fields will be displayed. When user adds news system will save it in database and will display on home page of each and every user.

3.7.3.6.2 Basis Flow

Actor Actions	System Response
User clicks on the link “Add News.”	System will display a web form with the fields of Title, Description, Attach file

User fills the required fields and clicks on add button	System will add this information to database.
---	---

3.7.3.6.3 Alternative Flow

1. User Leave some field empty.
2. User clicks on close window button.
3. User logged out.

3.7.3.7 Functional Requirement

REQ-1:System should provide functionality of adding project proposals by supervisors.

REQ-2:Supervisors can remove project proposals if project is not assigned to any group.

REQ-3: Project proposals should only be visible to students for selection only if it is approved by the appropriate forum.

REQ-4: System must create a work space for supervisor and his/her group.

REQ-5:System should restrict access of any user to work space of other user.

REQ-6: Supervisor can upload a file from their computer to their workspace.

REQ-7:System should allow supervisors to allocate the project to any group.

3.7.4 Student Functionalities

3.7.4.1 Login

Using this feature of system, Student can log in to application and can further user other features of system.

3.7.4.1.1 Description

Application home page has a login box which takes user name and password to login the user to application. System validates the user credentials and redirects the user to his/her home page according to his/her role.

3.7.4.1.2 Basic Flow

Actor Actions	System Response
User redirect to home page of application	A form will be displayed with username and password fields
User fills the required fields and clicks on submit button	System will show web page according to user's roles and access.

3.7.4.1.3 Alternative Flow

1. User leaves the required fields empty.
2. Username and password are not verified. Invalid username and password.

3.7.4.2 Chose Project

This feature will facilitate the students to choose a project of their interest.

Students can reserve more than one project at a time of different supervisors.

3.7.4.2.1 Description

At student's home page there will be link of "choose project". When user clicks that then there will be a list of available projects from which user can chose one or more projects. System will only allow group leader to choose project.

3.7.4.2.2 Basis Flow

Actor Actions	System Response
User clicks on the link "choose project"	System will display a web page having list of all project of all supervisors.
User clicks one project from available projects list.	System will display a web page containing a information of project. If it is of student's interest then he/she can choose this project by clicking on choose project button.

3.7.4.2.3 Alternative Flow

1. User is not group leader.
2. User clicks on close window button.
3. User logged out.

3.7.4.3 Work Space Creation

When a project is assigned to group then a work space is created for that project through which student can interact with his/her supervisor.

3.7.4.3.1 Description

Student can view if some task is assign to him and can update the supervisor about status of work. He/she also can add some resource file for his/her group. Complete information of group is also available at work space page.

3.7.4.3.2 Basis Flow

Actor Actions	System Response
User clicks on the link selected project.	System will display a web page through which student can perform taskupdate work status or upload resource.

3.7.4.3.3 Alternative Flow

1. User clicks on close window button.
2. User logged out.

3.7.4.4 View Project Schedule

Student can view complete project schedule which will be added by project coordinator at the start of project.

3.7.4.4.1 Description

Studentwill click on view schedule link provided on his/her profile page. A well formatted page will be displayed which will contain information of different activities of project throughout the year.

3.7.4.4.2 Basis Flow

Actor Actions	System Response
User clicks on the link “View Schedule”	System will display a web page having whole year project activities. A brief description of event, start date and end date will be displayed to user.

3.7.4.4.3 Alternative Flow

1. User clicks on close window button.
2. User logged out.

3.7.4.5 Functional Requirement

REQ-1:System should provide functionality to students to choose one or more projects at the same time.

REQ-2:System should allow the student to remove the projects from their selected projects list

REQ-3: system should restrict the students to remove the project once project has been allocated to that group.

REQ-4: Supervisors can send notification to groups who choose his/her project for interview/meeting.

3.7.5 Project Coordinator Functionalities

3.7.5.1 Login

Using this feature of system, Project Coordinator can log in to application and can further user other features of system.

3.7.5.1.1 Description

Application home page has a login box which takes user name and password to login the user to application. System validates the user credentials and redirects the user to his/her home page according to his/her role.

3.7.5.1.2 Basic Flow

Actor Actions	System Response
User redirect to home page of application	A form will be displayed with username and password fields
User fills the required fields and clicks on submit button	System will show web page according to user's roles and access.

3.7.5.1.3 Alternative Flow

1. User leaves the required fields empty.
2. Username and password are not verified. Invalid username and password.

3.7.5.2 Add Project Schedule

This Feature provides the ability to Project Coordinator to add a project schedule which will have all activities and events of whole year. Once Project Coordinator will add this schedule to database then it will be available to each and every user of application.

3.7.5.2.1 Description

At supervisor's home page there will be link of "Project schedule". When user clicks that then there will be a web form which will take all required information from user and on clicking "Add" button system will save information to database.

3.7.5.2.2 Basis Flow

Actor Actions	System Response
User clicks on the link "Project schedule"	System will display a web and Project Coordinator will enter all required information. The form contains the information of Activity Title, Description, Start Date, End Date
User enters the required information and presses the "Add" button.	User fills in the required information and on submits the information is added to the database in the relevant table. This schedule will be available to each and every user domain.

3.7.5.2.3 Alternative Flow

1. User Leave some field empty.

2. Start date is greater than End Date.
3. User clicks on close window button.
4. User logged out.

3.7.5.3 Approve Projects

System will enable project Coordinator to approve project on behalf of R&D wing.

3.7.5.3.1 Description

Project Coordinator can approve projects. A list of project will be available and Project Coordinator can approve project which he/she wants.

3.7.5.3.2 Basis Flow

Actor Actions	System Response
User clicks on the link "Project Status"	System will displays a web page containing all projects and approve button.
User click "Approve" button	System will mark that project as approved and this project is now available to students for selection.

3.7.5.3.3 Alternative Flow

1. User clicks on close window button.
2. User logged out.

3.7.5.4 Add News

Project Coordinator can add some news for other user of application.

3.7.5.4.1 Description

Project Coordinator will click on Add News link provided on his/her profile page. A web form with required fields will be displayed. When user adds news system will save it in database and will display on home page of each and every user.

3.7.5.4.2 Basis Flow

Actor Actions	System Response
User clicks on the link “Add News.”	System will display a web form with the fields of Title, Description, Attach file
User fills the required fields and clicks on add button	System will add this information to database.

3.7.5.4.3 Alternative Flow

1. User Leave some field empty.
2. User clicks on close window button.
3. User logged out.

3.7.5.5 Functional Requirement

REQ-1:System must provide functionality to project coordinator to set timeline for different project activities.

REQ-2:Project coordinator can modify the timeline at any time.

3.8 Non-Functional Requirements

The non-functional requirements of **AcProMS** are listed below.

3.8.1 Performance Requirements

3.8.1.1 Response Time

The web pages should be loaded in reasonable response time. System should meet the load time as specified by different Web standards.

3.8.1.2 Scalability

System should be able to support increasing number of users every year due to increasing number of students and their related project. Moreover, the storage space should be able to increase easily whenever required.

3.8.2 User Interface Requirements

The system should be easy to use and users should be able to use the system and complete their tasks without any formal training. For this purpose help should be provided where required.

3.8.3 Stress Handling and Congestion Control

AcProMS should compliance with the requirements of stress handling once multiple requests are processed. It must also conform to the congestion control requirement multiple requests and responses will be exchangebetween client and server.

3.8.4 Software Quality Attributes

The client application must be compatible with all browsers. All users' passwords should be encrypted. **AcProMS** must be available when required. **AcProMS** should be reliable and no data loss should occur. GUI should be user friendly.

3.9 Hardware Requirements

The hardware requirements for **AcProMS** should be high enough so that the system can perform optimally without any compromise in performance under different levels of loads and stress.

3.10 Tools and Technologies Requirements

1. Microsoft Visual C# (Visual Studio 2008 or greater).
2. Microsoft ASP.NET
3. SQL Server for database
4. Microsoft Expression Web/ Adobe Dreamweaver

3.11 Summary

The main tasks in terms of requirements specification need to be done in order to accomplish desired project goals have been highlighted in this chapter. These requirements have given the clear understanding of system design which can be implemented for the **AcProMS** application development. Graphical interface measures need to be adopted while designing the proposed **AcProMS** underlying structure. Non-functional requirements mentioned in this chapter have reflected upon us a clear picture of performance and quality attributes

which must be considered while designing this system to ensure its efficiency, reliability, integrity, availability and security.

CHAPTER 4

System Design

System Design

4.1 Introduction

This chapter provides documentation that will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case diagrams, sequence diagrams, state diagrams, class diagram, architectural design, WebML Designs and Design Patterns.

i. Purpose

The purpose of this document is to provide a description of the design of “Academic Project Management System (AcProMS)” fully enough to allow for development to proceed with an understanding of what is to be built and how can we build it.

ii. Audience

The target audiences are implementers, but this document should be useful to anyone who wishes to understand the overall structure of each component.

iii. Scope

This design document contains the static view, logical view, and dynamic view of the system. Static view is depicted and class diagram and ER diagram, while dynamic view is depicted by statechart and sequence diagrams. Class and statechart diagrams are in fact the content model for web applications. This also covers presentation and hypertext models for our project.

iv. Goal

The ultimate goal of this document is a technical and functional design specification that helps the development team to build the system according to user's requirements. It will also help in achieving the required quality characteristics defined in the Software Requirements Specification (SRS) document. For this reason, we tried our best to make this document:

Complete: Everything here reflects the latest information. We do not want readers to be several versions behind by the time they are ready to dive into the code.

Clear: Avoiding ambiguity as much as possible.

4.2 System Architecture

Architectural style is concerned with the subsystems of an application with their relationships and collaborations with each other. The architecture of a software system is almost never limited to a single architectural style, but is often a combination of architectural styles that make up the complete system. A combination of architecture styles is also useful for building a Web application like AcProMS.

In our project we can achieve effective separation of concerns by using the layered architecture style. This will separate our presentation logic from business logic and data access logic. On presentation layer, we have decided to use Model-View-Controller (MVC) as a separated interactive system pattern.

The main benefits of the layered architectural style, and the use of a separated interactive system pattern, are:

Abstraction: Layers allow changes to be made at the abstract level. You can increase or decrease the level of abstraction you use in each layer of the hierarchical stack.

Isolation: Allows you to isolate technology upgrades to individual layers in order to reduce risk and minimize impact on the overall system.

Manageability: Separation of core concerns helps to identify dependencies, and organizes the code into more manageable sections.

Performance: Distributing the layers over multiple physical tiers can improve scalability, fault tolerance, and performance.

Reusability: Roles promote reusability. For example, in MVC, the Controller can often be reused with other compatible Views in order to provide a role specific or a user-customized view on to the same data and functionality.

Testability: Increased testability arises from having well-defined layer interfaces, as well as the ability to switch between different implementations of the layer interfaces. Separated Presentation patterns allow you to build mock objects that mimic the behavior of concrete objects such as the Model, Controller, or View during testing.

Presentation/user interface can be changed easily without affecting the functionality of the system. Moreover, user is able to customize the user interface.

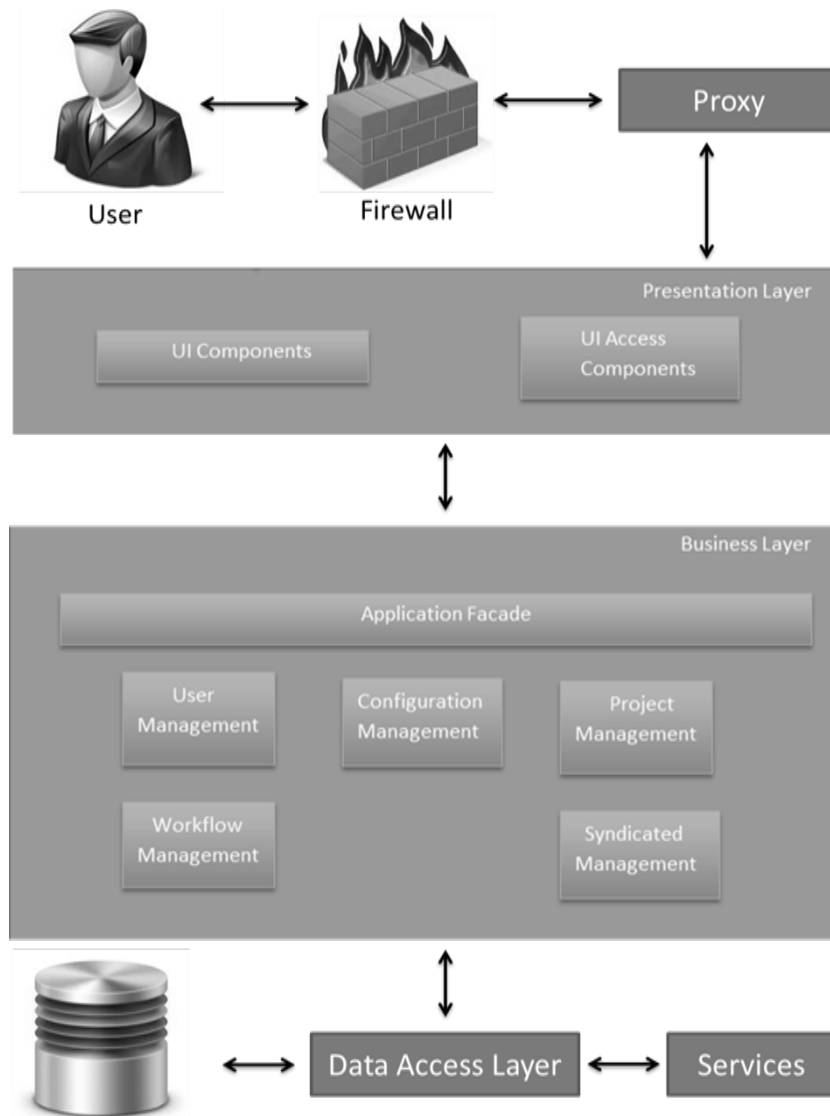


Figure 4-1: Architecture Diagram

The presentation layer provides the application's user interface (UI). Typically, this involves the use of Windows Forms for smart client interaction.

The business layer implements the business functionality of the application. The domain layer is typically composed of a number of components implemented using one or more programming languages. Business layer is further subdivided into many sub-components.

User management: User management includes tasks like registration, roles assignment, and login.

Configuration management: In configuration management, tasks like file version control and workspace creation are included.

Project management: In project management, tasks like add project proposal, remove project proposal, choose project(s), and allocate project are included.

Workflow management: Workflow management includes tasks like send message, add activity or news.

Syndicate management: Syndicate management includes tasks related to group i.e. group formation etc.

The data layer provides access to external systems such as databases.

4.3 Detailed Design

Overall system functionality is depicted with the help of use case diagrams given in the behavioral design.

Logical view shows that how the overall project is structured and also shows the relationship between the different modules of system.

4.3.1 Functional Model

4.3.1.1 Use Case Diagrams

Overall System: This diagram shows the general functionality of our system. It shows all the actors that will interact with the product. Inside the system boundary lie the major use cases.

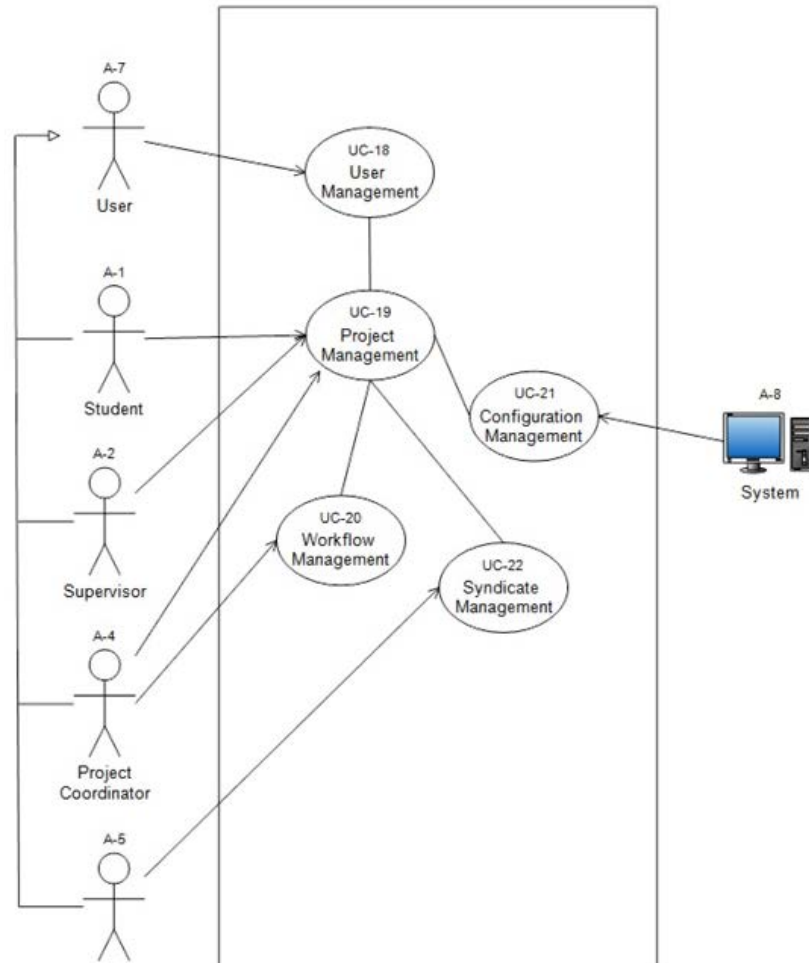


Figure 4-2: Abstract use-case Diagram

Following is a brief description of main modules of our system.

User Management

User management provides the functionality of registering new users that they can use functionalities of system. Registration can be manual or self. Course advisor can register students using CSV file. This module also provides functionality of authenticating users.

Role Assignment & Customization

Administrator can assign different role to different users of system so that they can be redirected to their authorize pages.

Syndicate Management

System must support the creation of syndicates (groups) by the authorized users. System must also support the changes of groups at any later stages if required.

Project Management

System must provide the functionality to supervisor to add or remove the project proposal. Student must be able to choose one or more project of different. System must ensure that student can't remove project once allocated. System must allow supervisors to allocate the project to any group.

Workflow Management

System must provide functionality to project coordinator to set and modify timeline for different project activities. System should have functionality to create an alarming news and send an email to group leaders if due date of any submission is coming up.

Configuration Management

Once project is allocated to some group, system must create a workspace for better communication between supervisor and group. System must provide the functionality of change management and version control so that it's easy for student to keep track of their work.

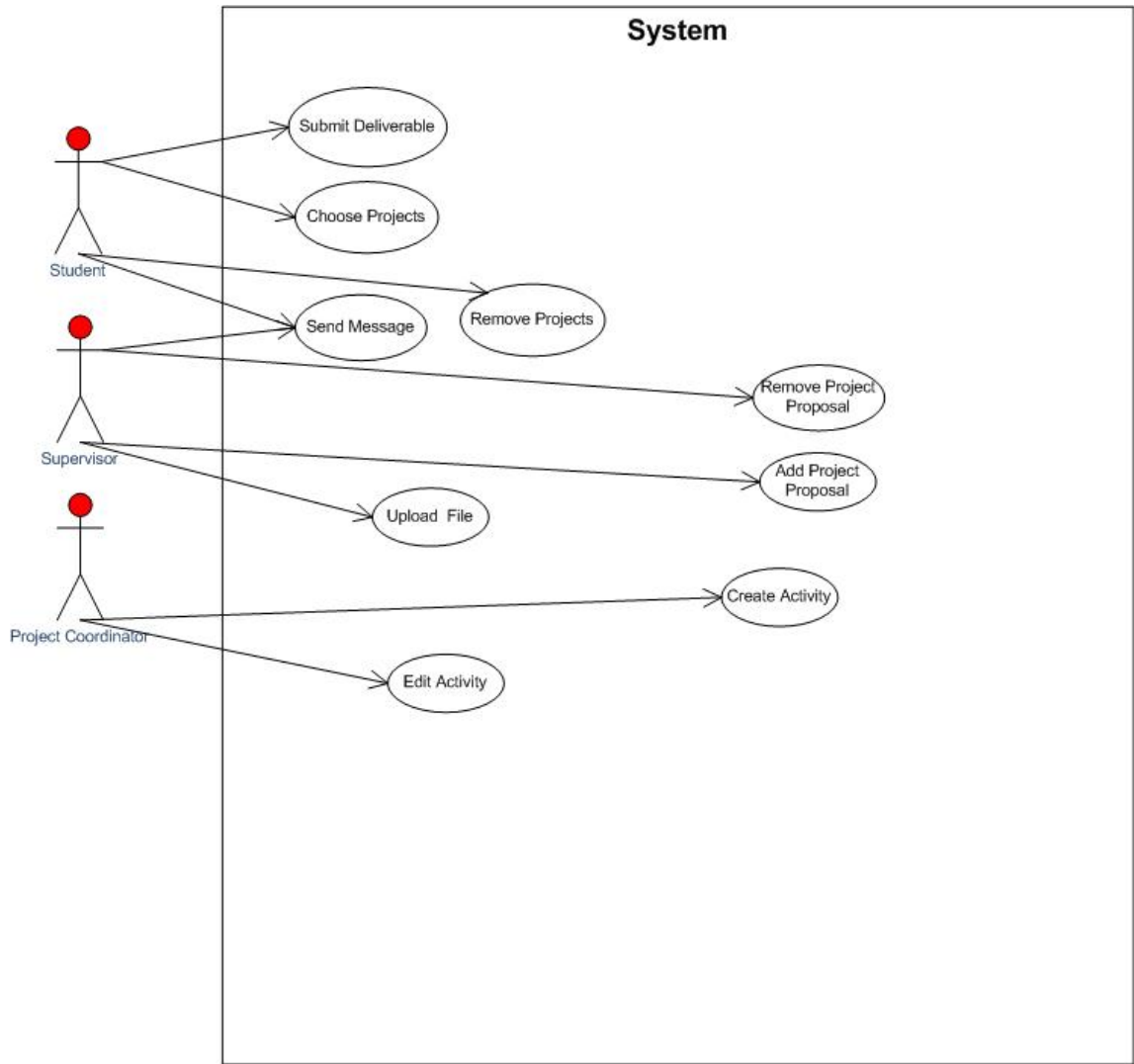


Figure 4-3: Project Management use-case Diagram

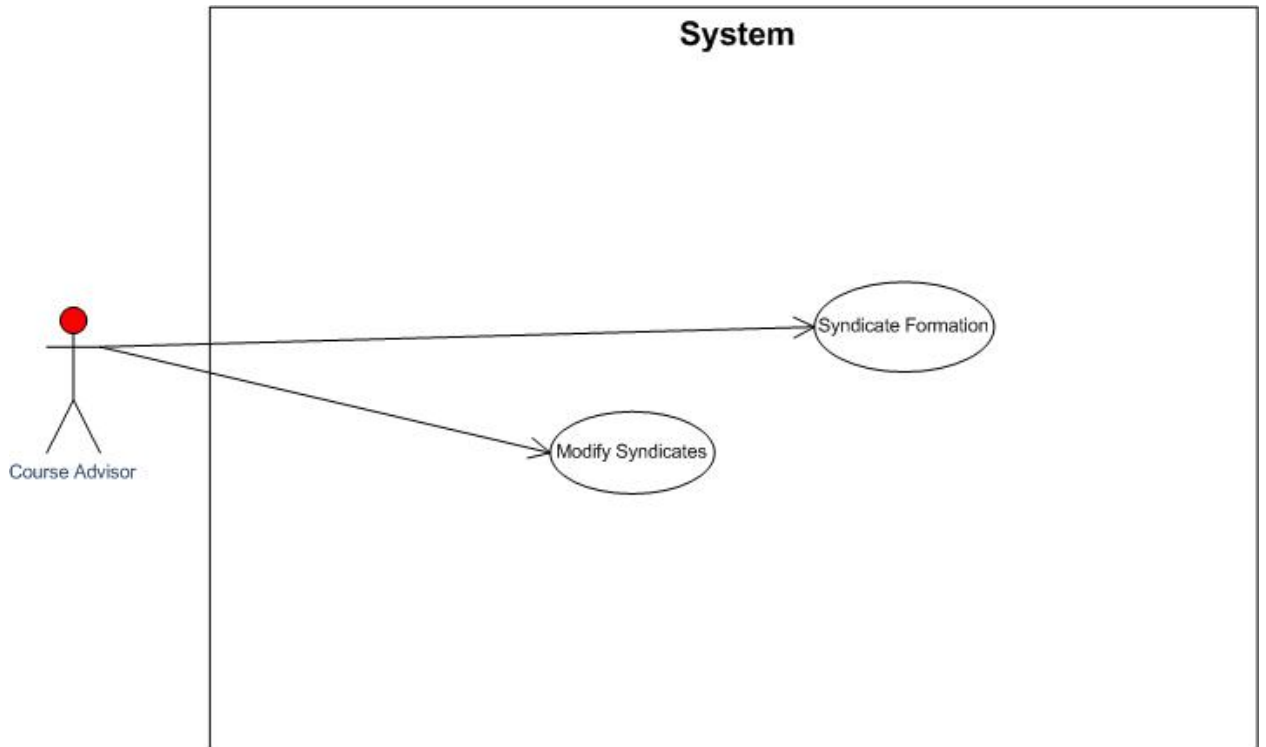


Figure 4-4: Syndicate Management use-case Diagram

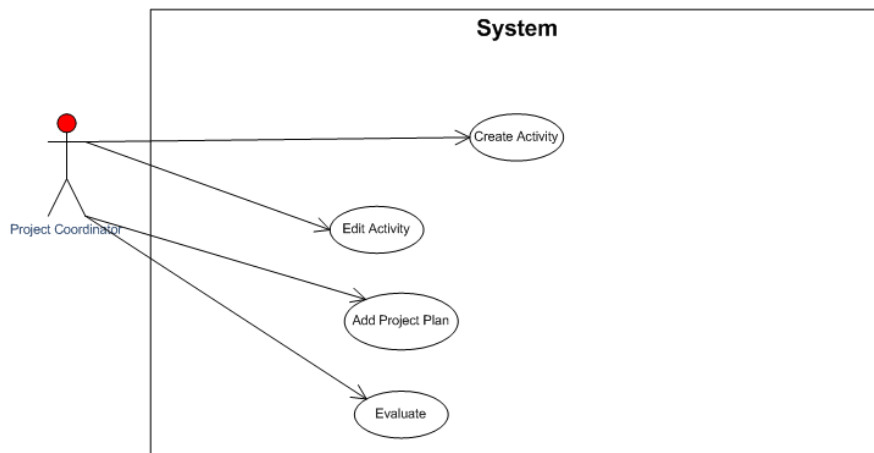


Figure 4-5: Workflow Management use-case Diagram

4.3.2 Logical View

4.3.2.1 Class Diagram:

Structural design is represented by a set of diagrams including class diagram. Class diagram shows a collection of static model elements such as classes and types, their contents, and their relationships. The main data in our project that needs to be encapsulated is of system user, project, and message. The system users are those actors who will interact with the system in order to either use it or manage it. The project supervisor, coordinator, head of department, administrator, course advisor, and student are inherited from user class. Similarly project can have one or many resources, including hardware and software etc., a project can have only one project proposal. Faculty can submit none or more project proposals. A project has one or more activities. A project can be performed by one group at a time.

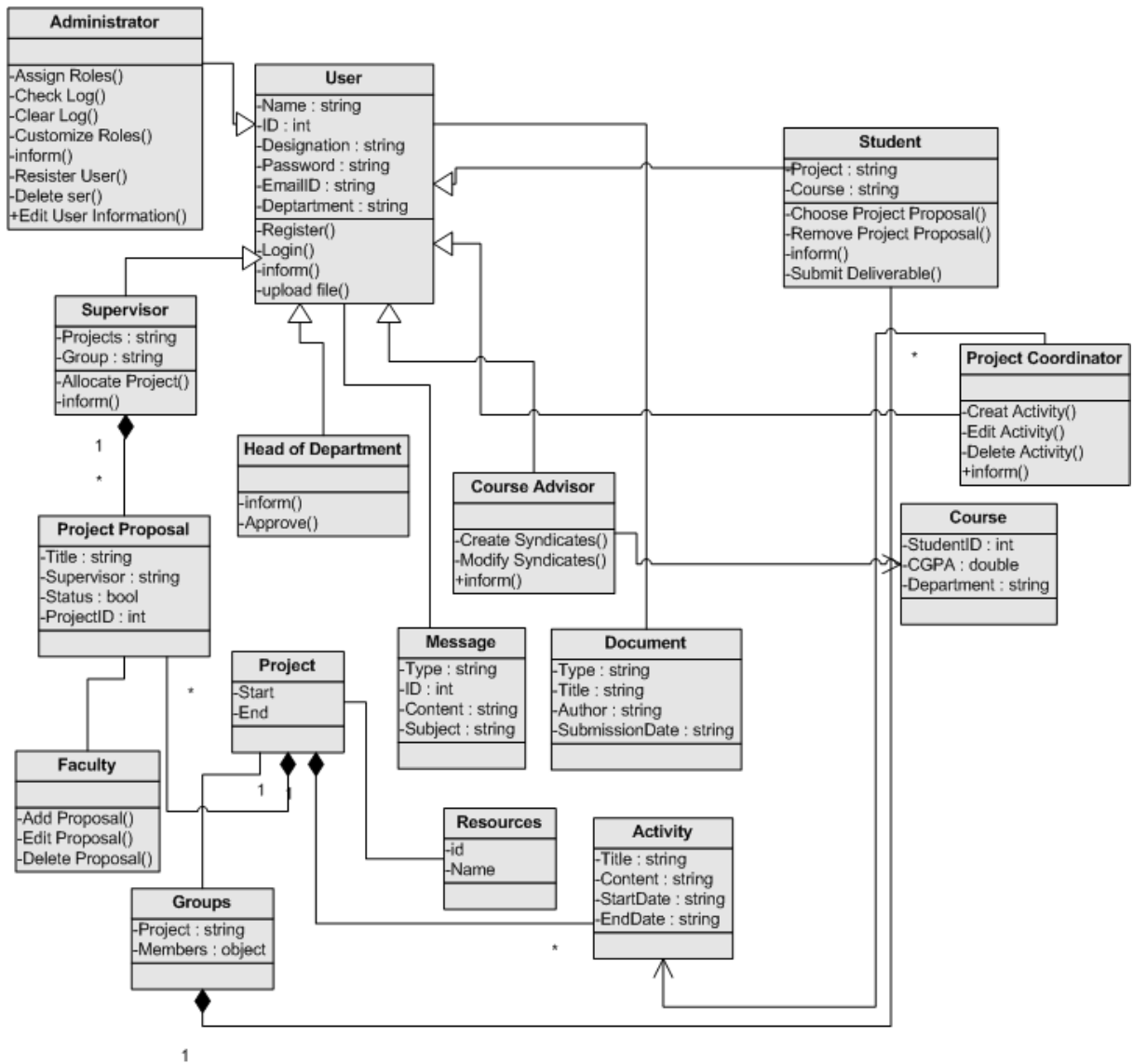


Figure 4-6: Class Diagram

The Above diagram also acts as a content model for the Web Application under development.

4.3.2.2 ER Diagram:

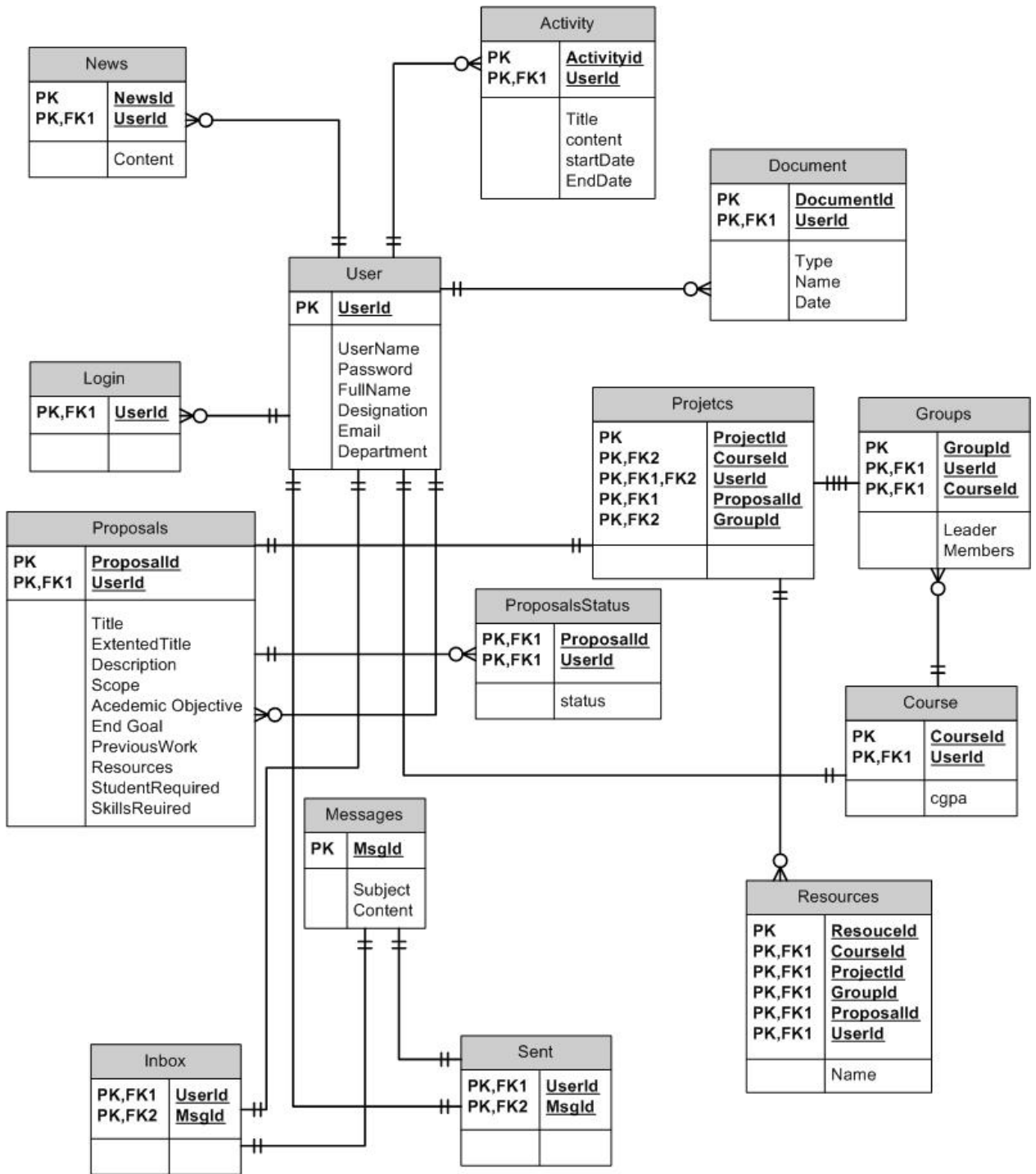


Figure 4-7: ER Diagram

4.3.2.3 State Transition Diagram

State transition diagram shows the state of an object during its lifecycle. Statechart has an entry point and an exit. In this project the main object of our concern is the project. The state transition diagram of our system shows the lifecycle of a project from un-proposed state to proposed state after supervisor submit the project, approved or disapproved by R&D, project is selected by syndicates (group of students) and after assignment of project, the activities of project start until its successful completion.

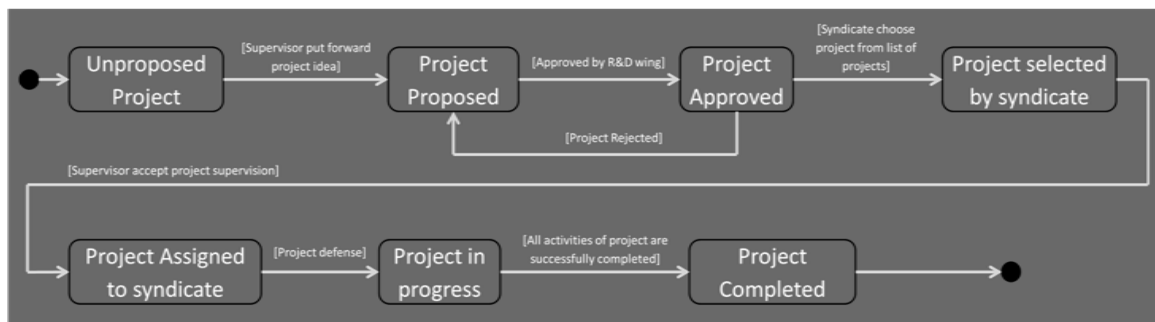


Figure 4-8: State Transition Diagram

4.3.3 Navigational View

4.3.3.1 Sequence Diagram

Login

User navigates to home page of AcProMS and enters his/her username and password in login box. System will authenticate these information and redirect user to his/her profile page otherwise show error message that either username or password is incorrect.

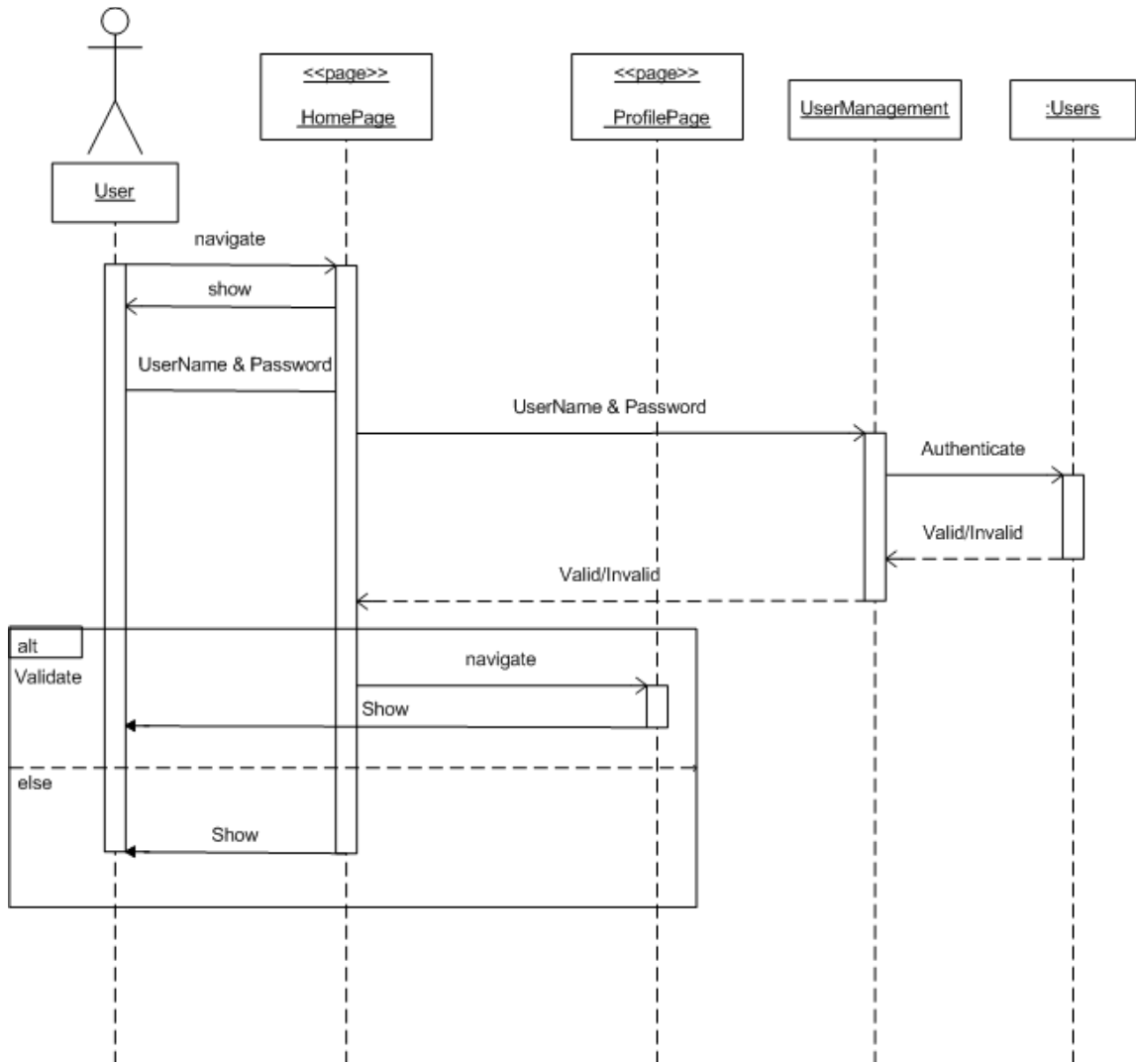


Figure 4-9: Login Sequence Diagram

Add proposal

User navigates to profile page by logging in to the system. For proposal upload, supervisor clicks the link “Add Proposal” from menu. System will display a web form for collecting required detail of proposal. The web form is customizable that means users can add field according to their need. After entering all information, when supervisor clicks submit button, system will validate all field and request the “addproposal” object of server to add this proposal to “proposals” table of database. System shows a message of successful action after adding this proposal to database.

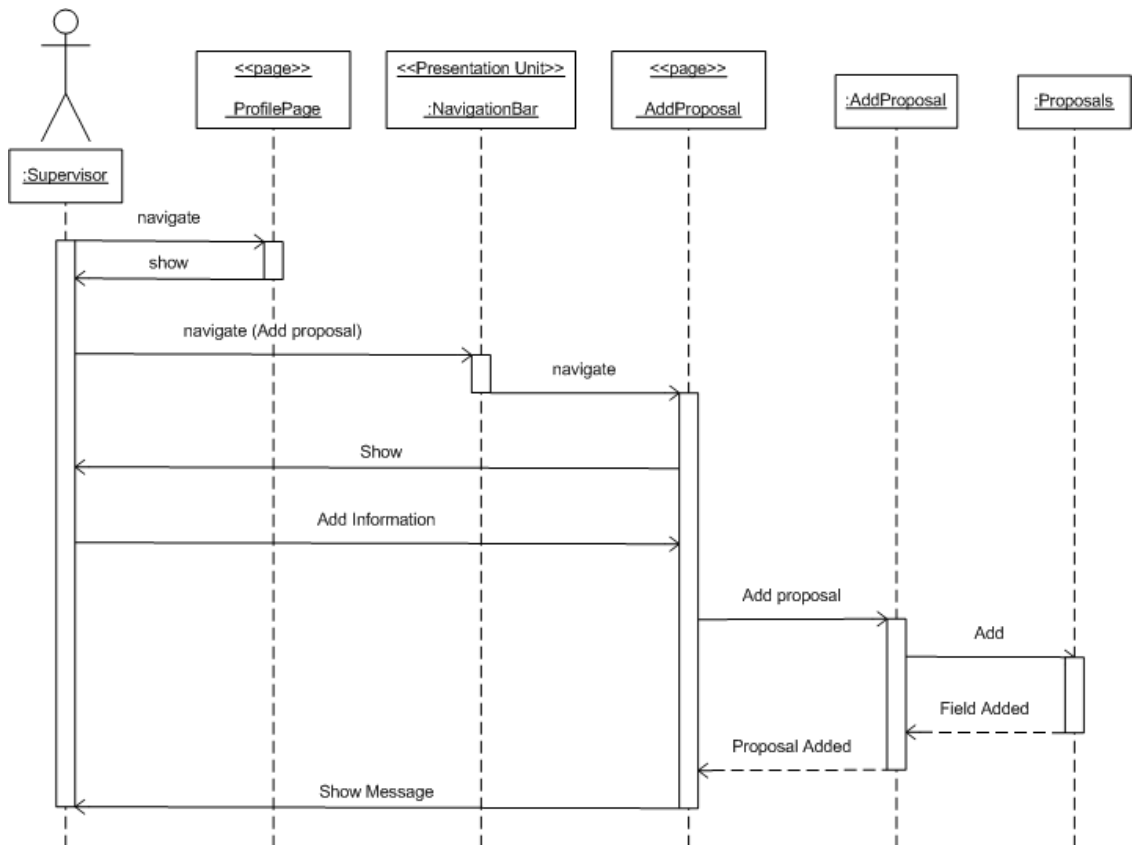


Figure 4-10: AddProposal Sequence Diagram

Choose projects

Student clicks on “choose Projects” link in a menu on his/her profile page. System will request to server to retrieve list of all uploaded proposals. System create page of proposal dynamically according to uploaded proposals. For given list of proposals student can choose of his/her interest and click on “select project” button. System will add these projects to database and prompt a message of successful action.

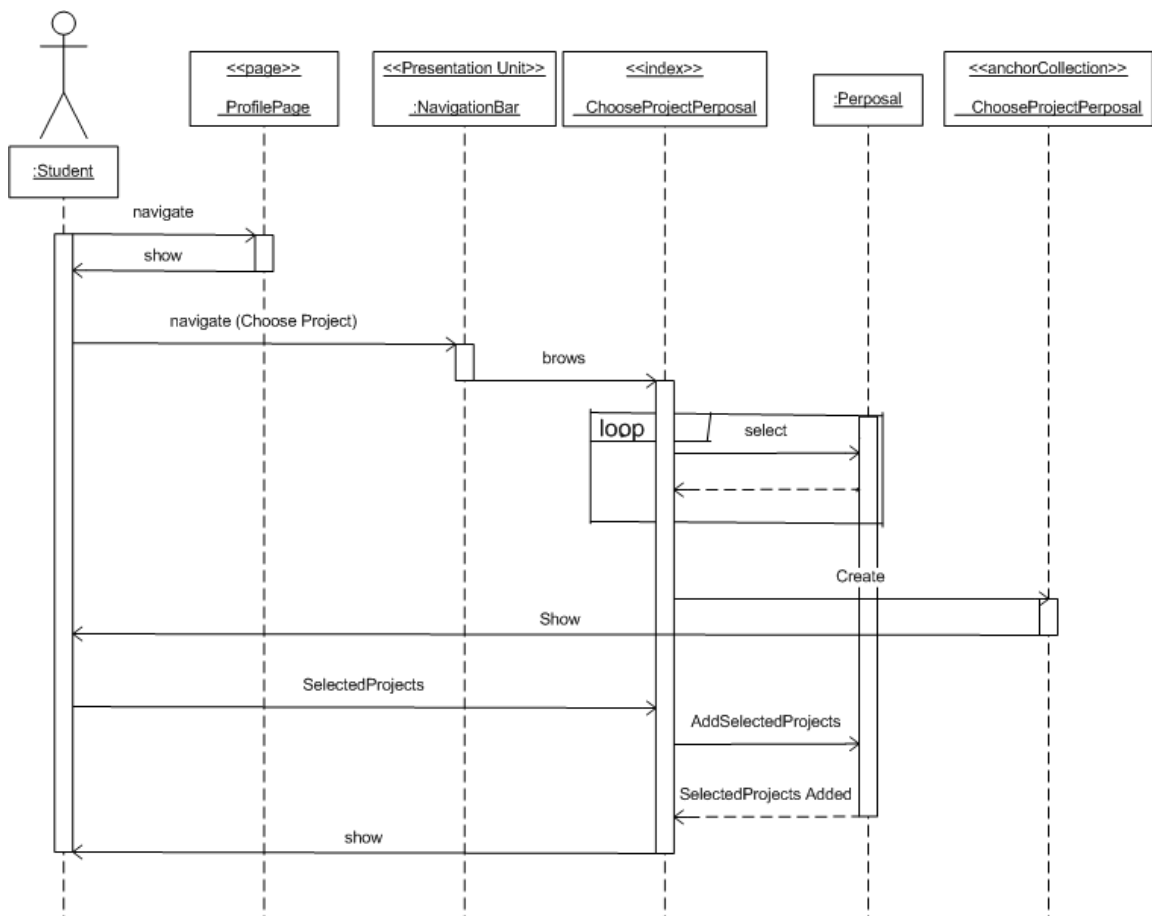


Figure 4-11: ChoseProject Sequence Diagram

Create activity

As there are many activities and milestones in whole project period so project coordinate can create an activity by clicking on “Create Activity” link. System will display a web form where user can set title, content, start and end date in addition to other fields. The activity will be added on database and displayed on relevant pages.

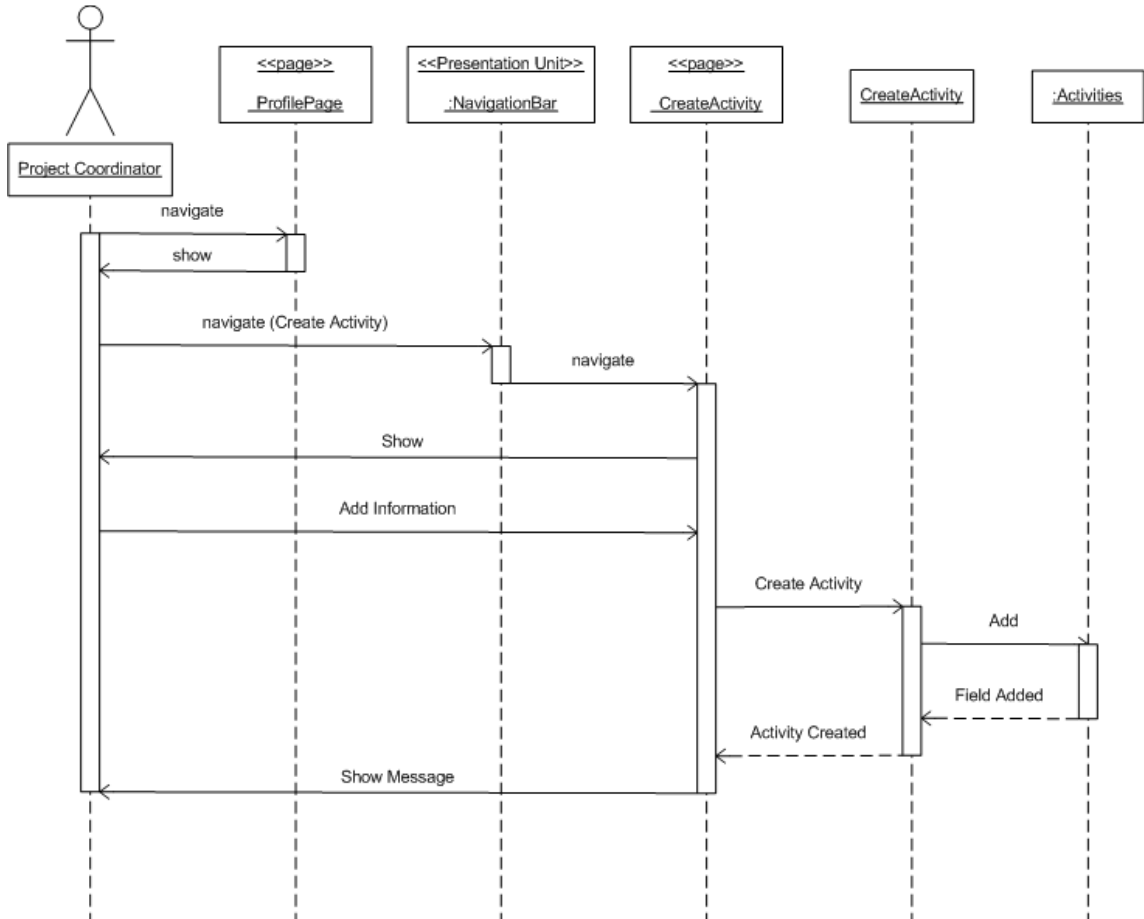


Figure 4-12: Create Activity Sequence Diagram

Submit Deliverable

Student navigates to profile page by logging in and clicks on “Submit Deliverable” link. System will display a web form. Student enters some required information and clicks on “Browse” button and select file on upload file dialogue. On submitting system will upload the document on server and save information in database.

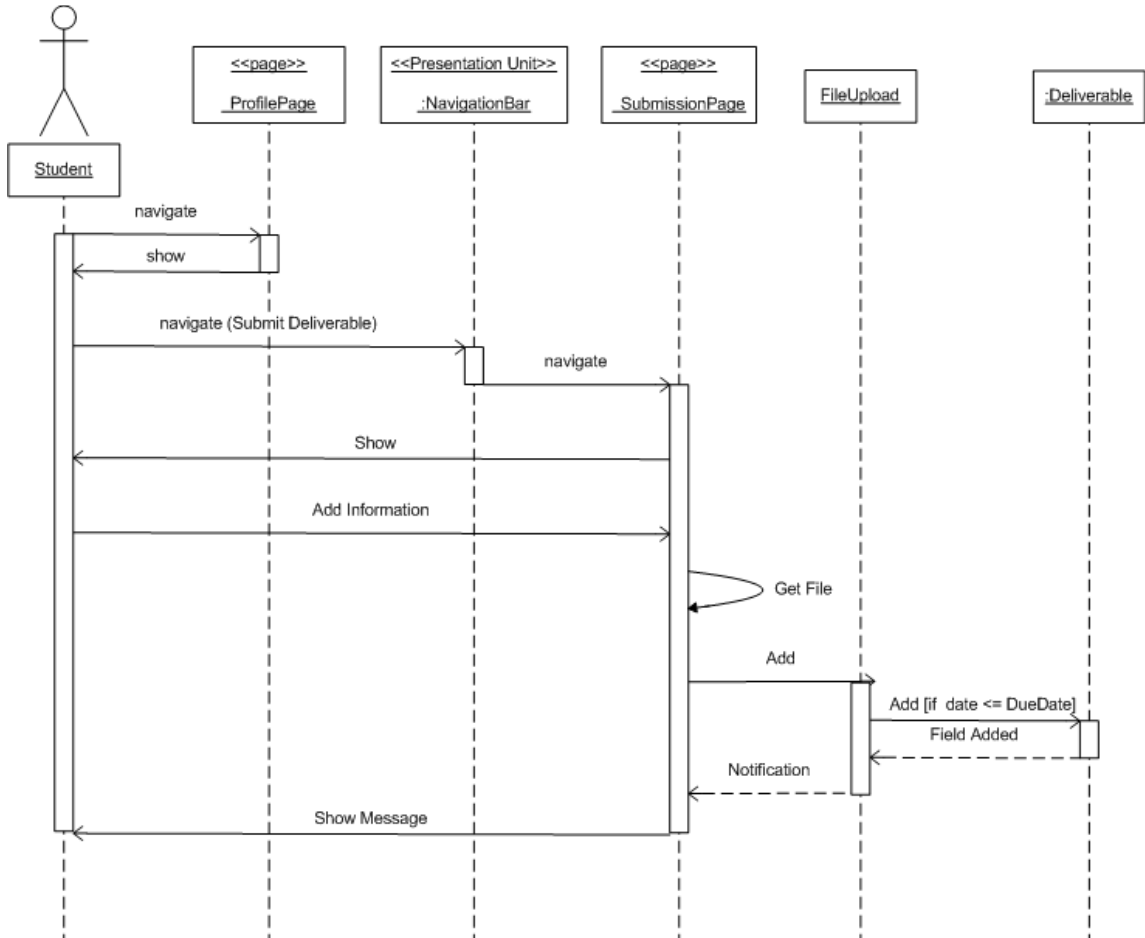


Figure 4-13: Submit Deliverable Sequence Diagram

4.3.3.2 Hypertext Model

Hypertext model specifies composition and navigation of the site. Composition describes which pages compose the hypertext, and which content units make up a page. The pages of the Web application are the containers of information actually delivered to the reader. Navigation is specified through links. Links can be defined between the units inside a single page, between units placed in different pages, and between pages. The hypertext models for our project are given:

Supervisor's View

A supervisor can access pages of his/her user profile, submit or modify project proposal, read and send messages, view projects in progress and set milestone of projects.

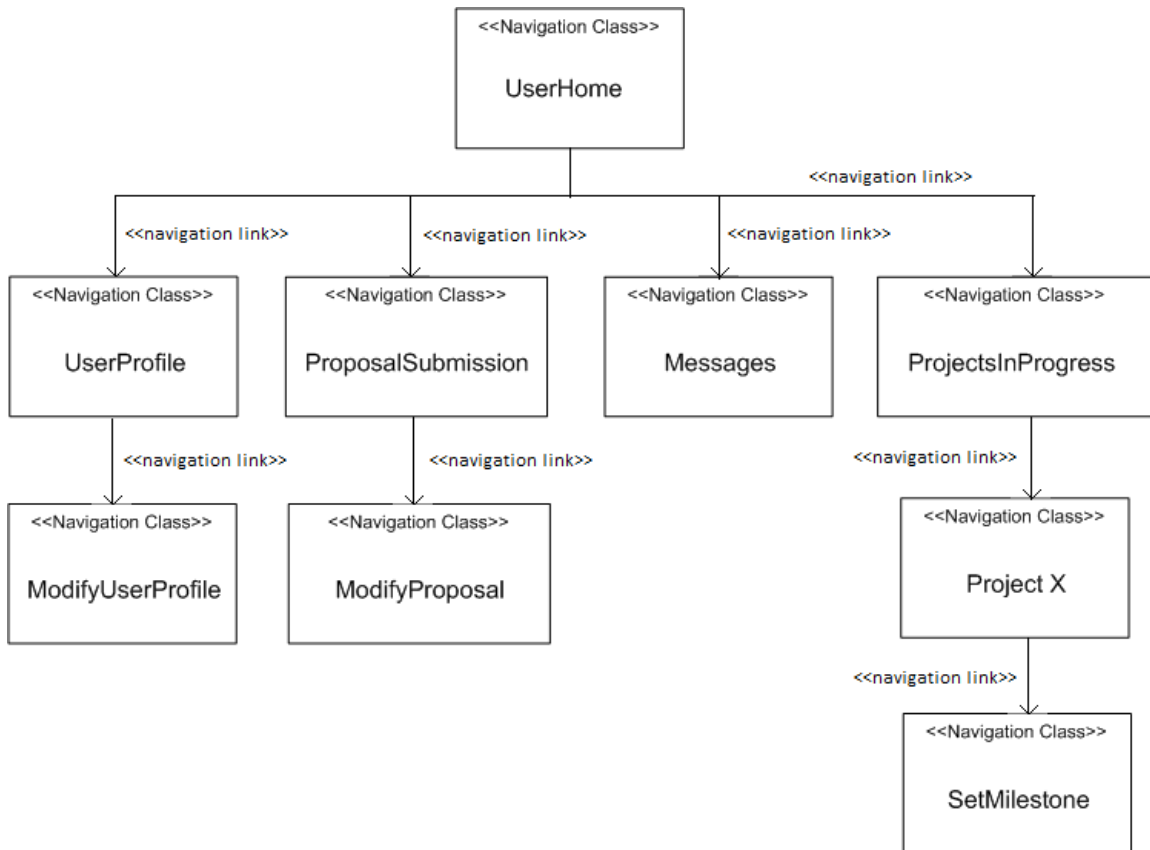


Figure 4-14: Supervisor's View Hypertext Model

Course Advisor's View

A course advisor can access pages of his/her user profile, syndicate formation or modification and read & send messages.

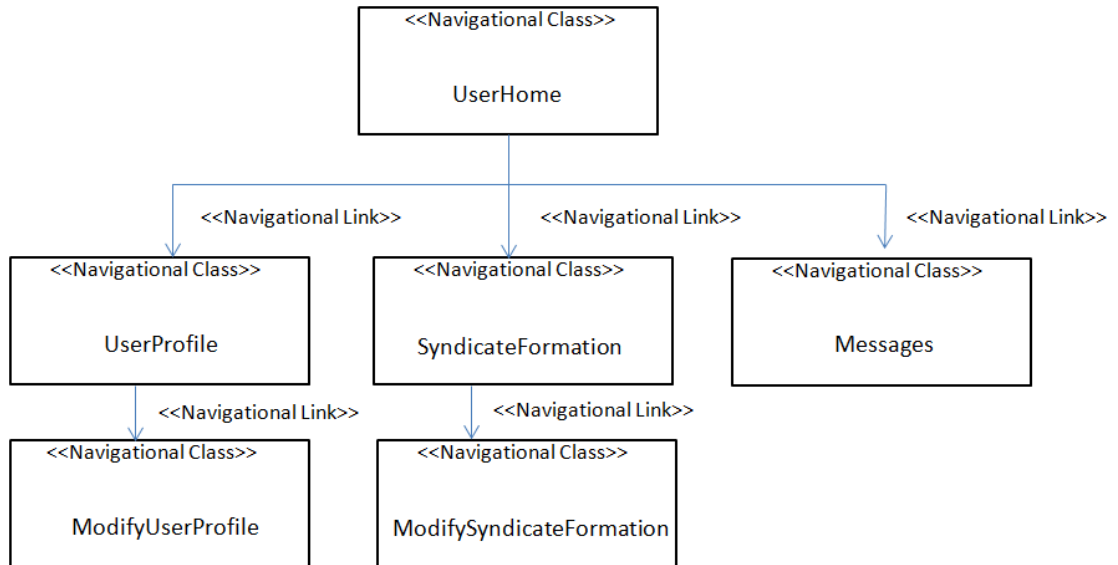


Figure 4-15 Course Advisor's View Hypertext Model

HoD's View

A Head of Department can access pages of his/her user profile, approve syndicates, project proposals, and read & send messages.

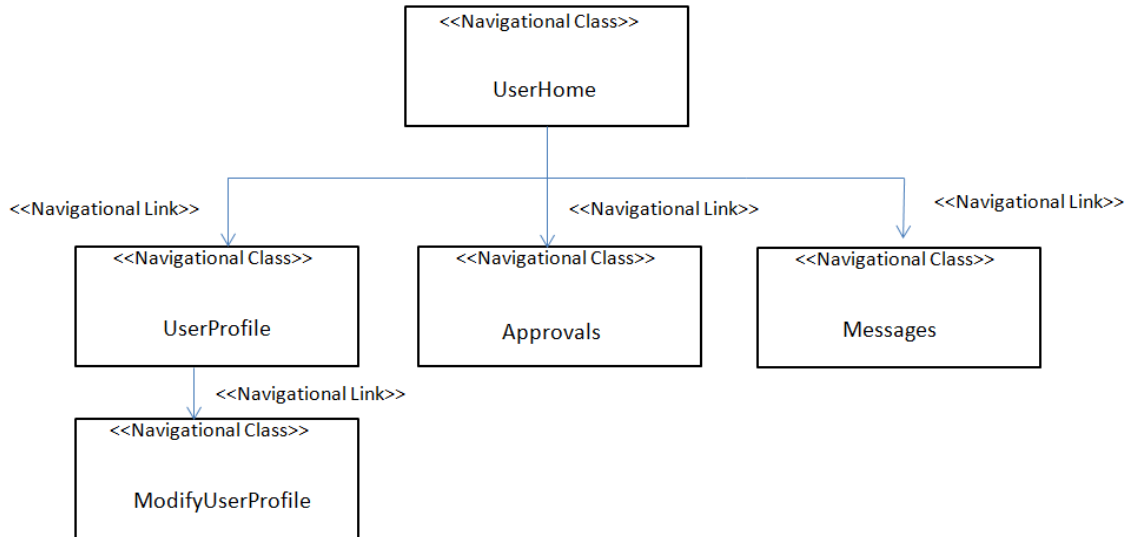


Figure 4-16: HoD's View Hypertext Model

Project Coordinator's View

A project coordinator can access pages of his/her user profile, creation or modification of project activities, adding or modification of project plan, adding or editing of news and events, and read and send messages.

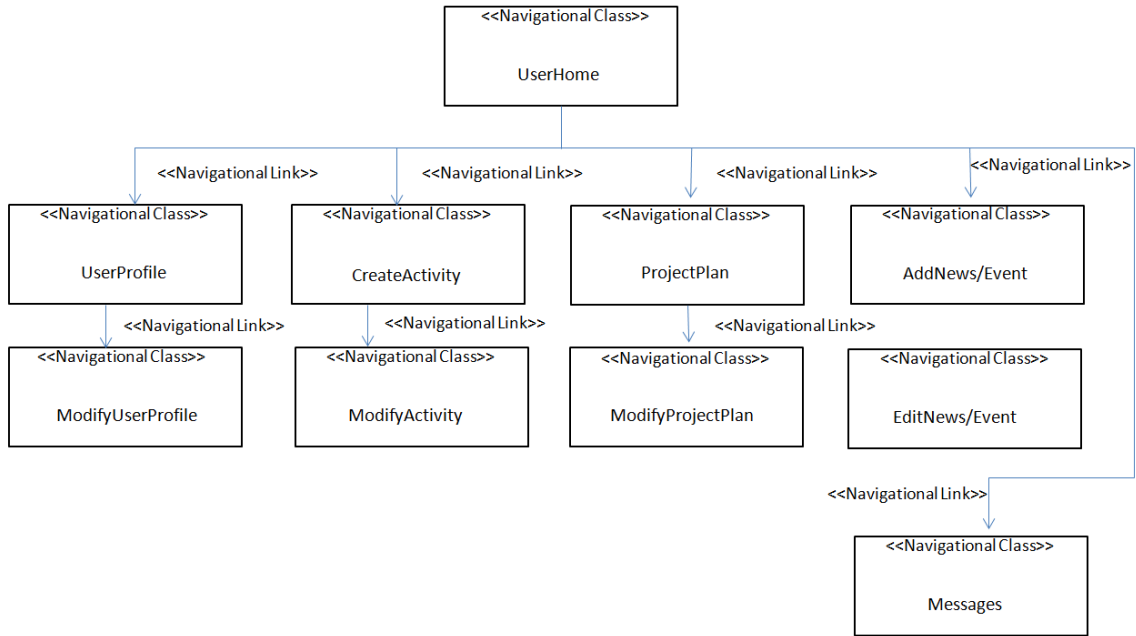


Figure 4-17: Project Coordinator's View Hypertext Model

Student's View

A student can access pages of his/her user profile, choosing project, upload deliverables, and read and send messages.

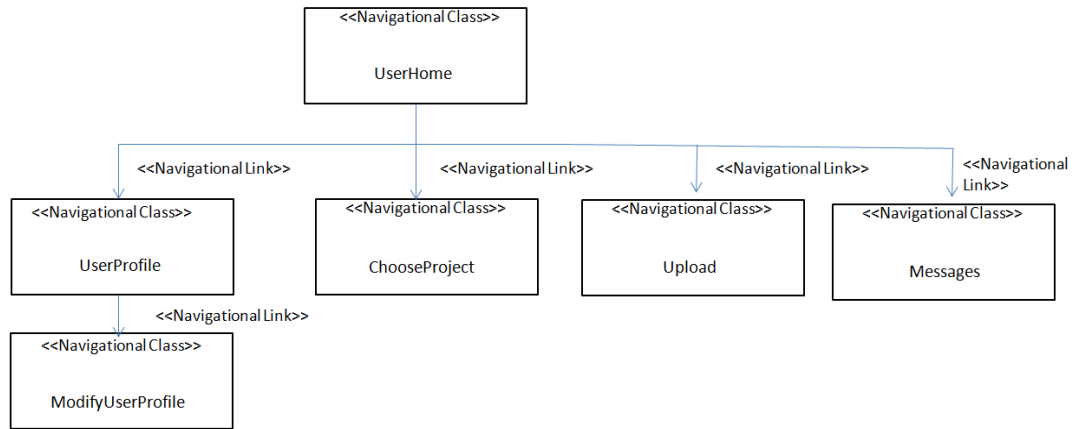


Figure 4-18: Student's View Hypertext Model

4.3.3.3 Access Model

Supervisor's View

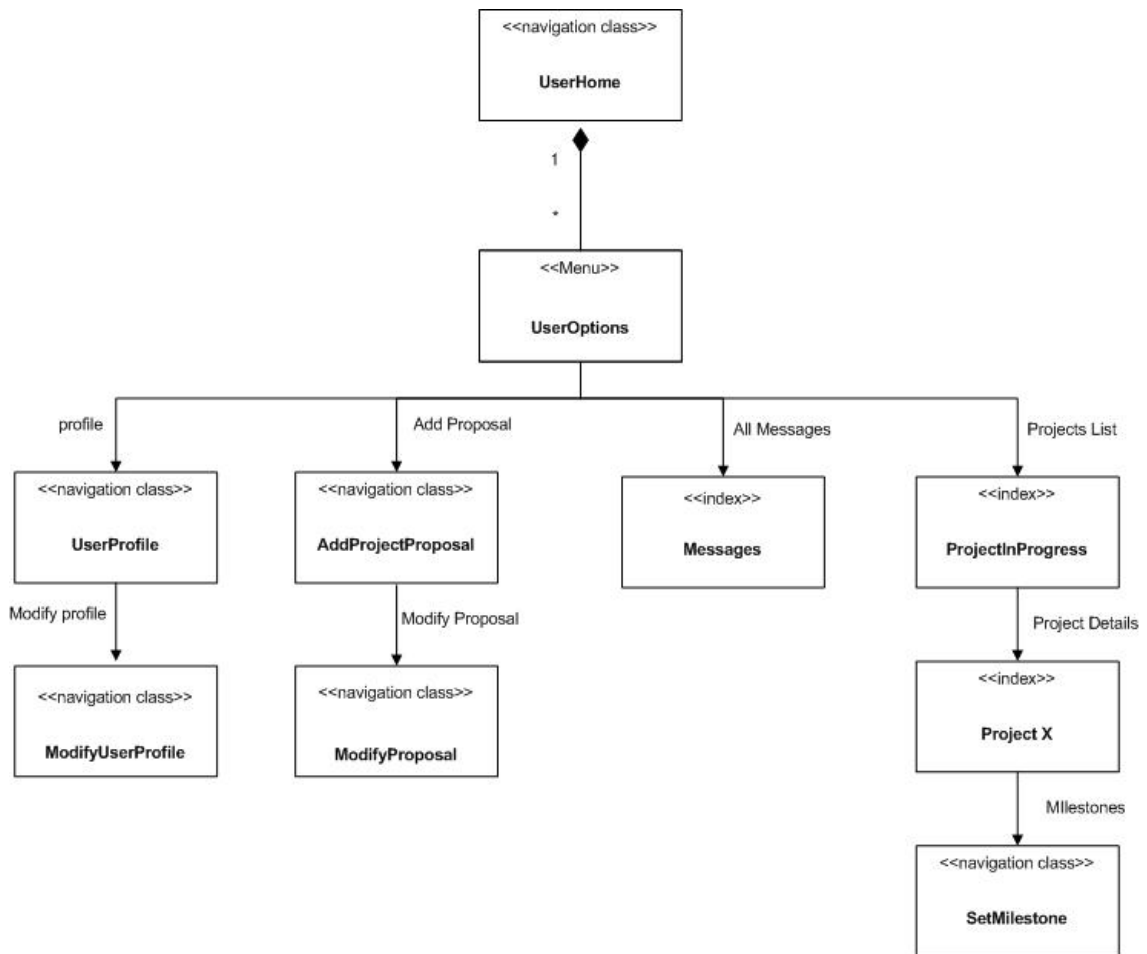


Figure 4-19: Supervisor's View Access Model

Course Advisor's View



Figure 4-20: Course Advisor's View Access Model

HoD's View

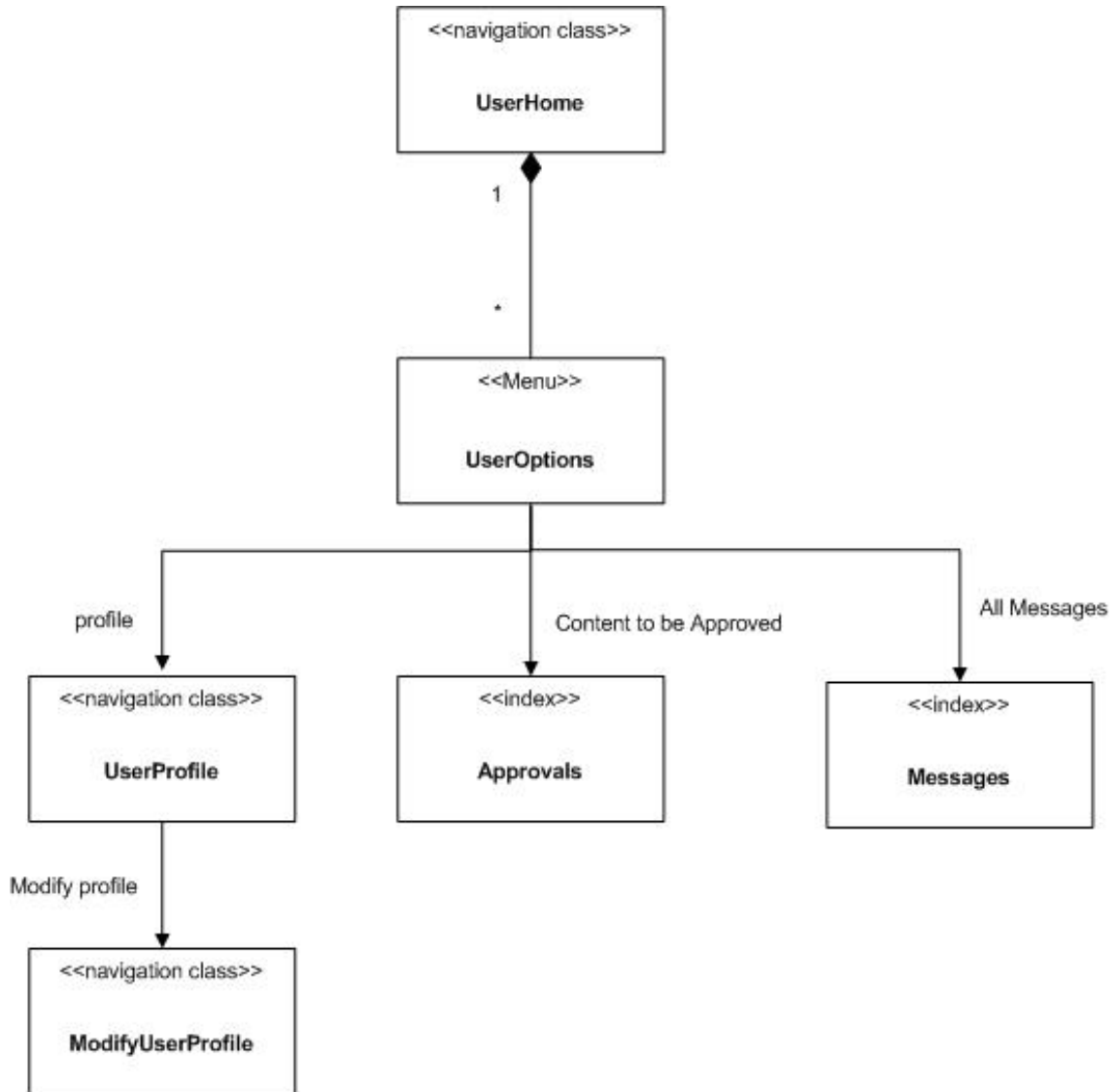


Figure 4-21: HoD's View Access Model

Project Coordinator's View

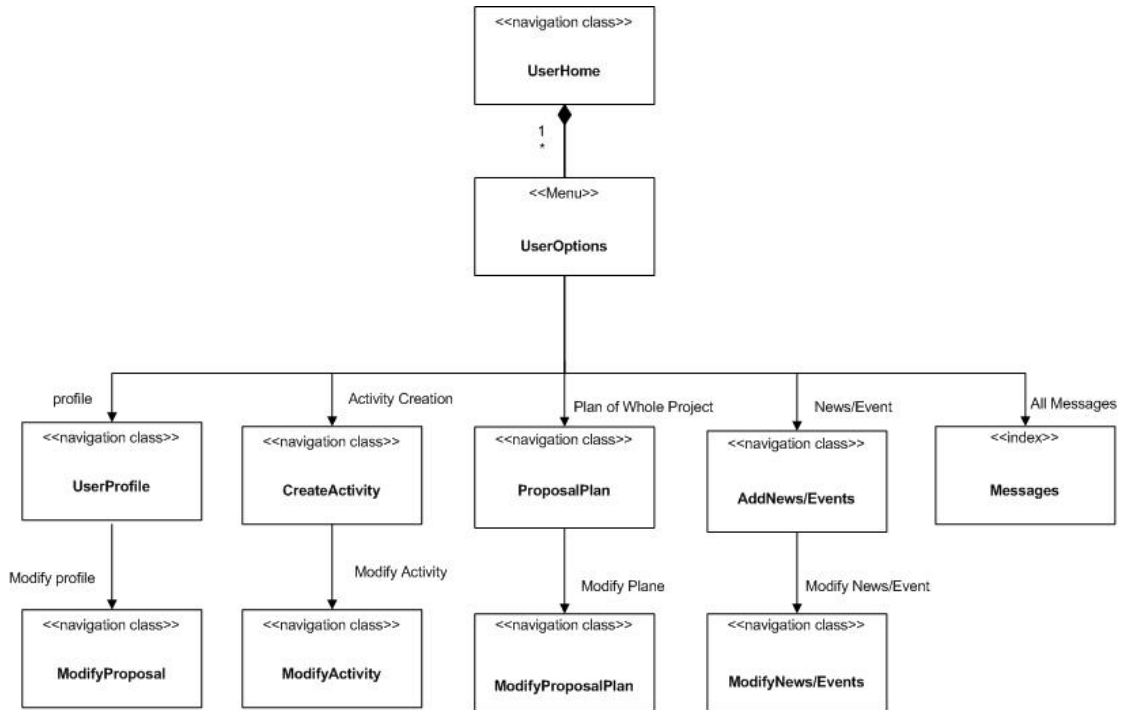


Figure 4-22: Project Coordinator's View Access Model

Student's View

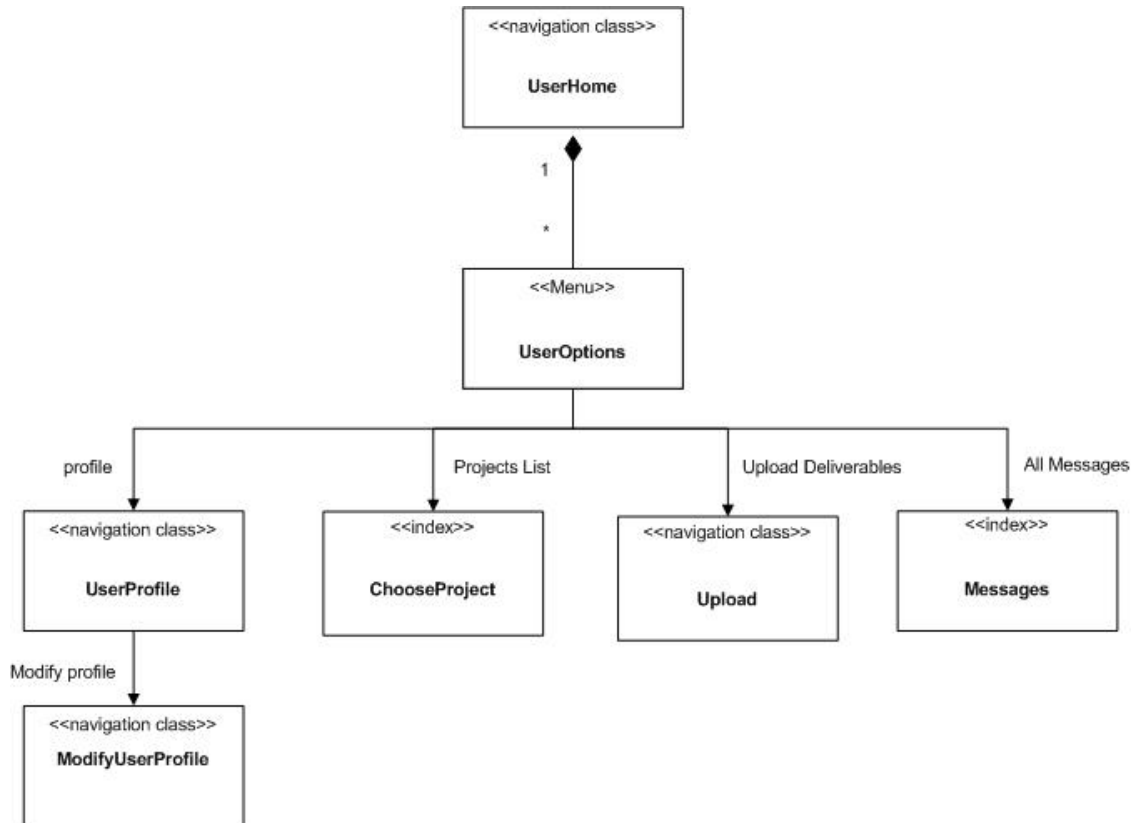


Figure 4-23: Student's View Access Model

4.3.4 Presentation Model

Presentation is the orthogonal task of defining the look and feel of pages in a site view. WebML does not include a specific model for expressing presentation at the conceptual level, but leverages standard approaches, more familiar to graphic and communication experts. This model uses the stereotype classes <<page>> and <<presentation unit>> to depict presentation pages and presentation units. The presentation models for some pages of our project are given below:

Home Page:

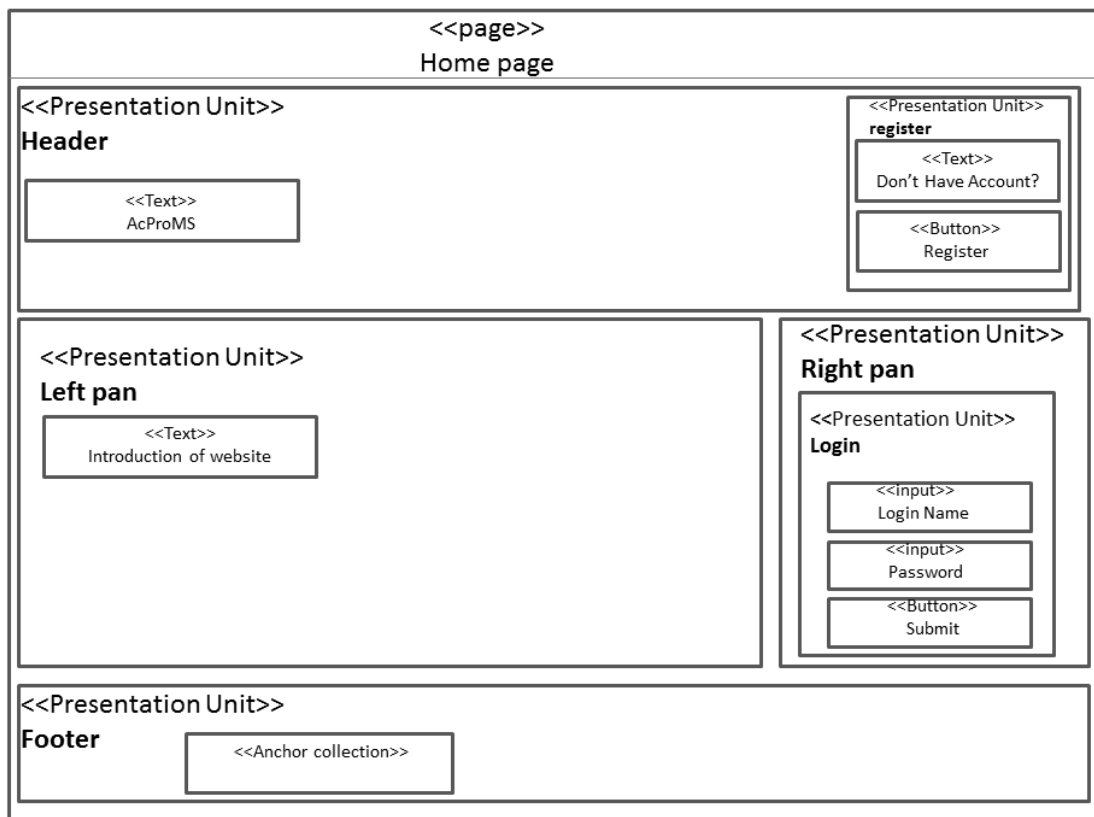


Figure 4-24: Home Page Presentation Model

The <<page>> “Home Page” has four presentation units, namely, header, left and right pan and footer. The header and footer have anchor collection that’s means that there may be any number of links according to our need at time of implementation.

Student Profile Page:

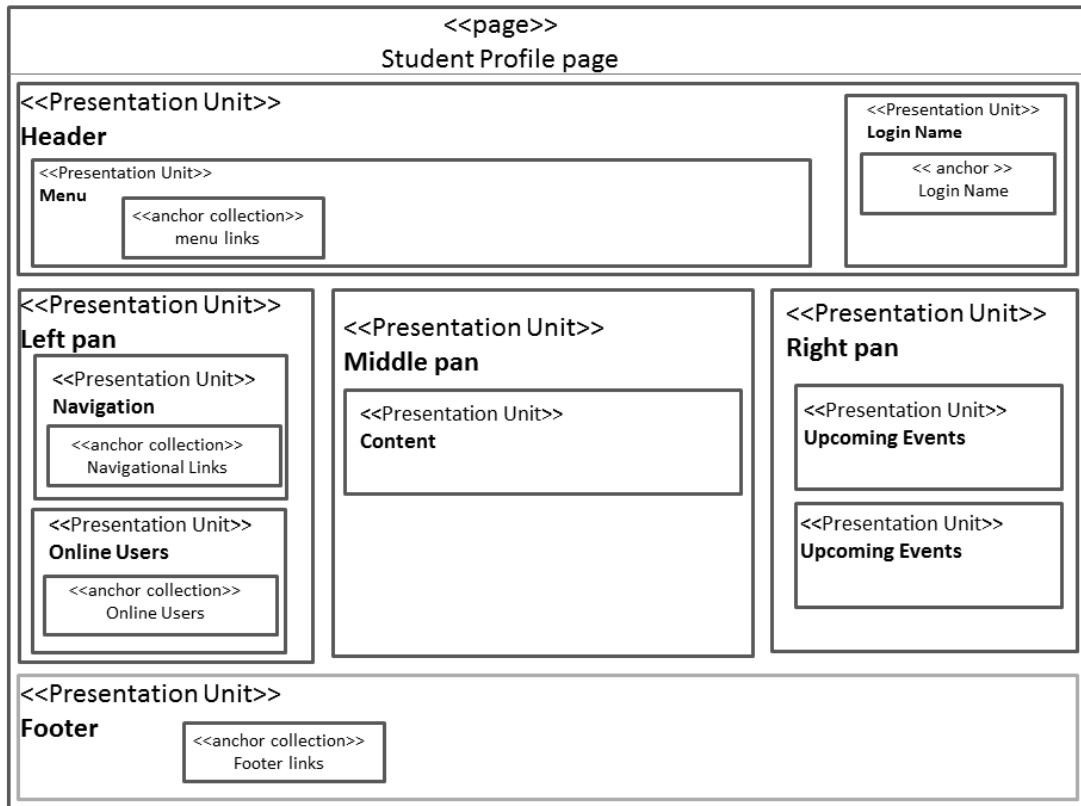


Figure 4-25: Student Profile Page Presentation Model

The <<page>> “Student Profile Page” has five presentation units, namely, header, left, middle and right pan and footer. The middle pan is our content area and the left and right pans are consist with many other pages of student. Content pan can contain text and images.

Submit Deliverable Page:

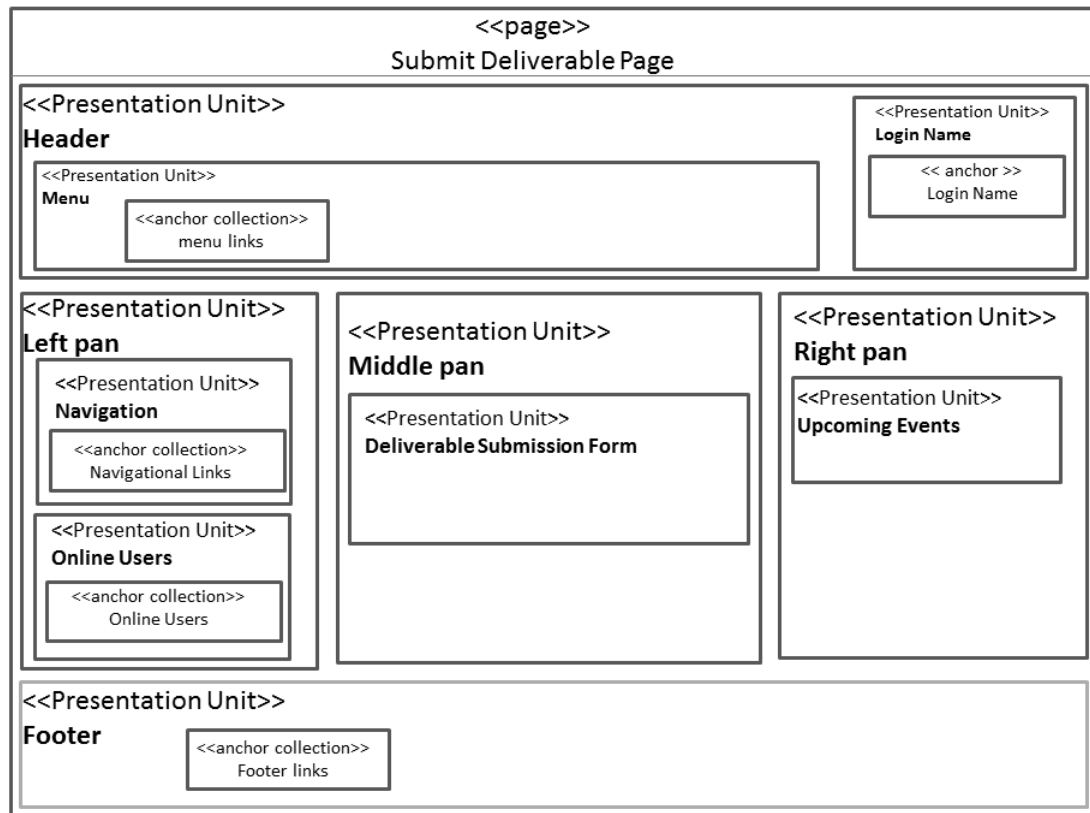


Figure 4-26: Submit Deliverable Page Presentation Model

The `<<page>>` "Submit Deliverable Page" has five presentation units, namely, header, left, middle and right pan and footer. Deliverable submission form contains fields necessary to gather information required for a document.

Create Activity Page:

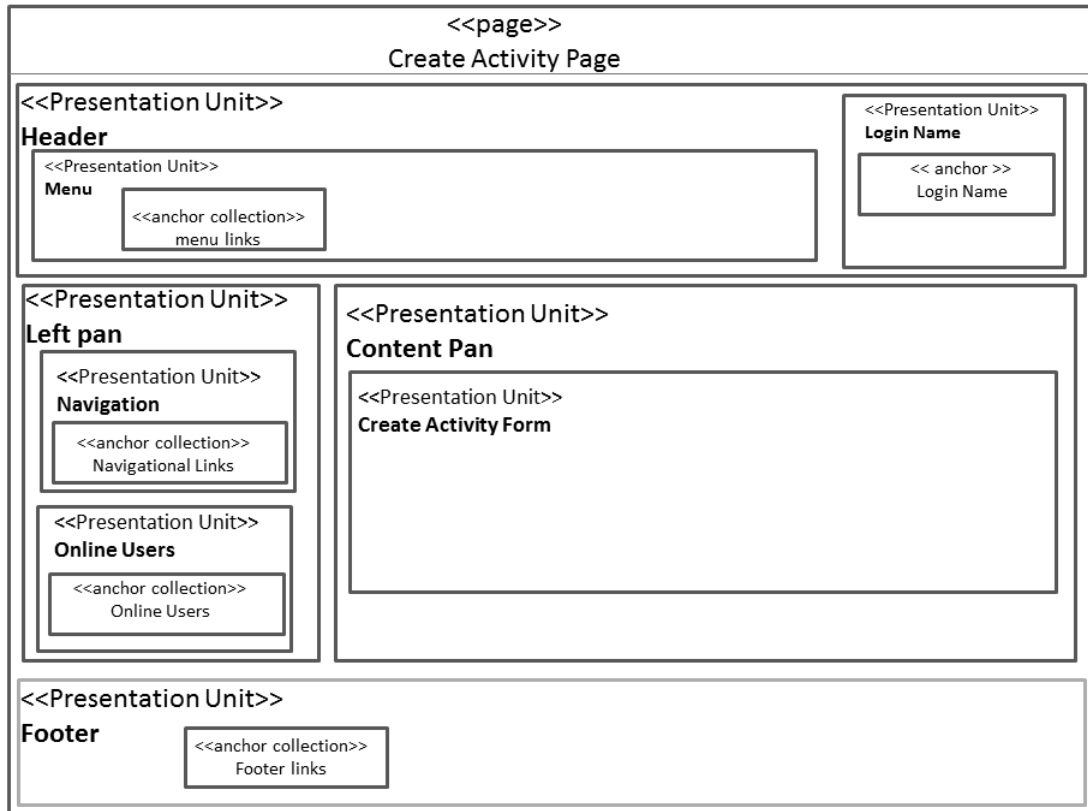


Figure 4-27: Create Activity Page Presentation Model

The `<<page>>` "Create Activity Page" has four presentation units, namely, header, left, content and footer. The creativity pan is also customizable so that the user can add or remove fields to his/her requirement. Activity pan also have calendar to choose start and end date of activity.

Add Proposal Page:

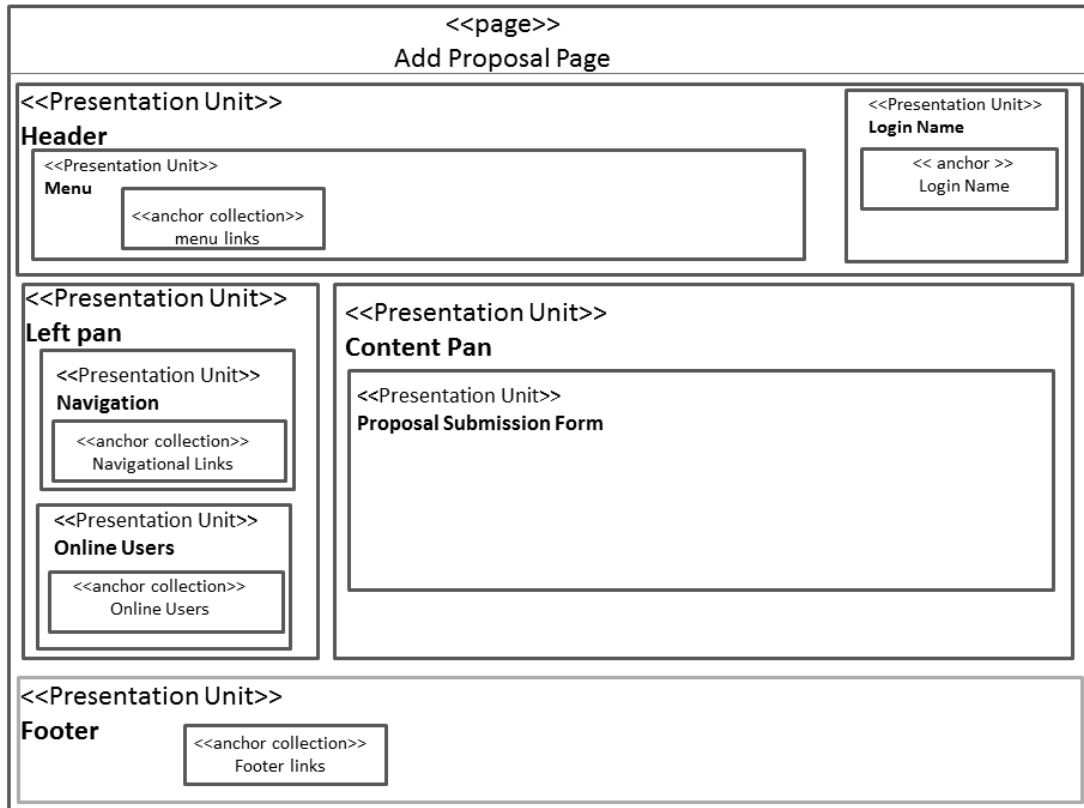


Figure 4-28: Add Proposal Page Presentation Model

The <<page>> “All Proposal Page” has four presentation units, namely, header, left, content and footer. Here middle pan contain proposal submission form. The form can have any number of fields according to the proposal format of organization.

4.4 Summary

Major design specifications needed to be adopted in order to implement the core functionalities of **AcProMS** have been highlighted in this chapter. High and low level architecture provided the basic structure and approach to implement functions in accordance with the design requirements. Furthermore this chapter has focused on

Sequential development and implementation of each component to achieve the top priority functionality of system first.

CHAPTER 5

System Implementation

System Implementation

5.1 Introduction

In this chapter a summary is given of strategies adopted to develop AcProMS. All the strategies varied in efficiency and response but as a whole were useful. The system was mainly divided five modules, User Interface, User Management Module, Project Management Module, Workflow Management Module, and Syndicate Management Module. The coding for the server side has been done in the Asp.net using c# language in Microsoft Visual Studio 2010, while the client end has been done using HTML, CSS and JQuery. This chapter presents the implementation details of the project.

5.2 Implementation

System implementation has been highlighted in the Figure 5-1 showing the overall architecture and design implementation of **AcProMS**. The client sends data to presentation layer which is implemented in HTML, CSS and JQuery. Presentation layer communicates with the business layer, implemented in ASP.NET and C#, to initiate required functionality.

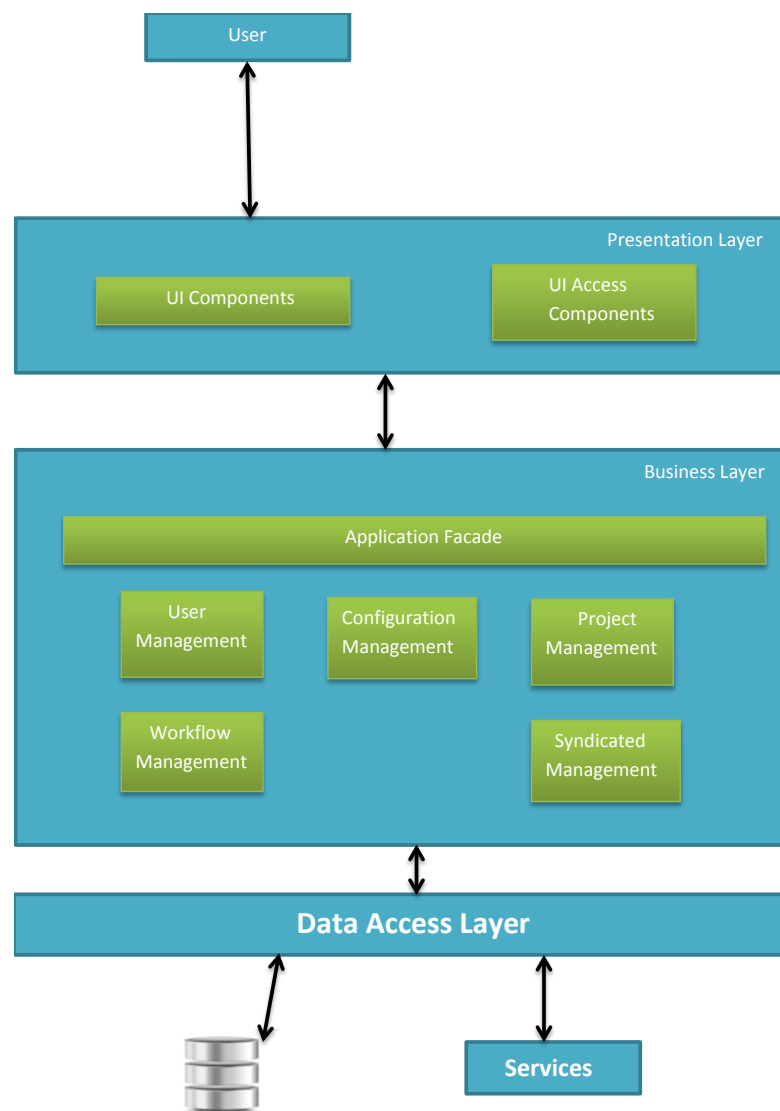


Figure 5-1:Implementation structure

5.3 Implementation Language

The implementation language which has been chosen for the Central Server is ASP.NET using C#. There are many advantages of ASP.NET:

Ease of development:

Development is always easy with the available server controls in ASP.NET apart from the user controls. This makes data management very much efficient and reduces the coding effort. These controllers are inbuilt available with Visual Studio and no need of finding them as external plugins. For example, a table with editing, deleting and insertion abilities is just a drag and drop of a server control named grid view.

Master Pages to Improve Efficiency:

Master pages provide the facility to reduce code weight of your application as well as speedup the application execution when a same content part is available for different pages.

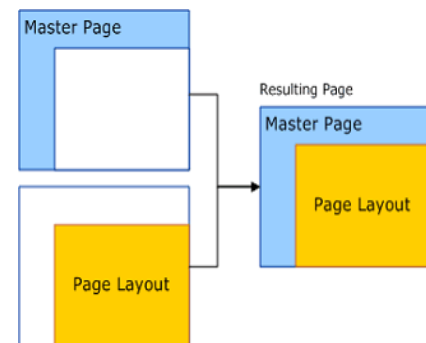


Figure 5-2: Master pages

Got Your Source with You:

In other languages such as PHP, the only available option for hosting the application is copying the files of the project to the server. ASP.NET makes the same functionality available with hosting only the register dlls in the server, which keeps your source code with you. This improves the security of the code for certain extend. Form the developer's

perspective; it saves your copyrights with you. Someday your source code will worth you more than the developed application.

C# has been preferred over other languages because:

- 1 C# is an object oriented language.
- 2 It is compiled to an intermediate language (CIL) independently of the Language it was developed or the target machine architecture and operating System
- 3 Automatic garbage collection
- 4 Pointers no longer needed (but optional)
- 5 Reflection capabilities
- 6 Definition of classes and functions can be done in any order
- 7 Declaration of functions and classes not needed
- 8 Non-existing circular dependencies
- 9 Classes can be defined within classes
- 10 There are no global functions or variables, everything belongs to a class
- 11 All the variables are initialized to their default values before being used
- 12 (This is automatic by default but can be done manually using static constructors)
- 13 You can't use non-Boolean variables (integers, floats...) as conditions. This indicates it is much cleaner and less error prone language.
- 14 Applications can be executed within a restricted sandbox

5.4 ASP.NET Page Life Cycle

In general terms, the page goes through the stages outlined in the following. In addition to the page life-cycle stages, there are application stages that occur before and after a

request but are not specific to a page. Some parts of the life cycle occur only when a page is processed as a postback. For postbacks, the page life cycle is the same during a partial-page postback (as when you use an UpdatePanel control) as it is during a full-page postback.

Stage	Description
Page request	The page request occurs before the page life cycle begins. When the page is requested by a user, ASP.NET determines whether the page needs to be parsed and compiled (therefore beginning the life of a page), or whether a cached version of the page can be sent in response without running the page.
Start	In the start stage, page properties such as Request and Response are set. At this stage, the page also determines whether the request is a postback or a new request and sets the IsPostBack property. The page also sets the UICulture property.
Initialization	During page initialization, controls on the page are available and each control's UniqueID property is set. A master page and themes are also applied to the page if applicable. If the current request is a postback, the postback data has not yet been loaded and control property values have not been restored to the values from view state.
Load	During load, if the current request is a postback, control properties are loaded with information recovered from view state and control state.
Postback event handling	If the request is a postback, control event handlers are called. After that, the Validate method of all validator controls is called, which sets the IsValid property of individual validator controls and of the page. (There is an exception to this sequence: the handler for the event that caused validation is called after validation.)
Rendering	Before rendering, view state is saved for the page and all controls. During the rendering stage, the page calls the Render method for each control, providing a text writer that writes its output to the OutputStream object of the page's Response property.
Unload	The Unload event is raised after the page has been fully rendered, sent to the client, and is ready to be discarded. At this point, page properties such as Response and Request are unloaded and cleanup is performed.

Individual ASP.NET server controls have their own life cycle that is similar to the page life cycle. For example, a control's Init and Load events occur during the corresponding page events.

Although both Init and Load recursively occur on each control, they happen in reverse order. The Init event (and also the Unload event) for each child control occur before the corresponding event is raised for its container (bottom-up). However the Load event for a container occurs before the Load events for its child controls (top-down). Master pages behave like child controls on a page: the master page Init event occurs before the page Init and Load events, and the master page Load event occurs after the page Init and Load events.

When a class is created that inherits from the Page class, in addition to handling events raised by the page, methods can be override from the page's base class. Note that when an event handler is created using the Page_event syntax, the base implementation is implicitly called and therefore no need to call it in user defined method. For example, the base page class's OnLoad method is always called, whether aPage_Load method is called or not. However, if pageOnLoad method is overridden with the override keyword it must explicitly call the base method. For example, if OnLoad method is override on the page, you must call base.Load in order for the base implementation to be run.

The following illustration shows some of the most important methods of the Page class that can be override in order to add code that executes at specific points in the page life cycle. The illustration also shows how these methods relate to page events and to control

events. The sequence of methods and events in the illustration is from top to bottom, and within each row from left to right.

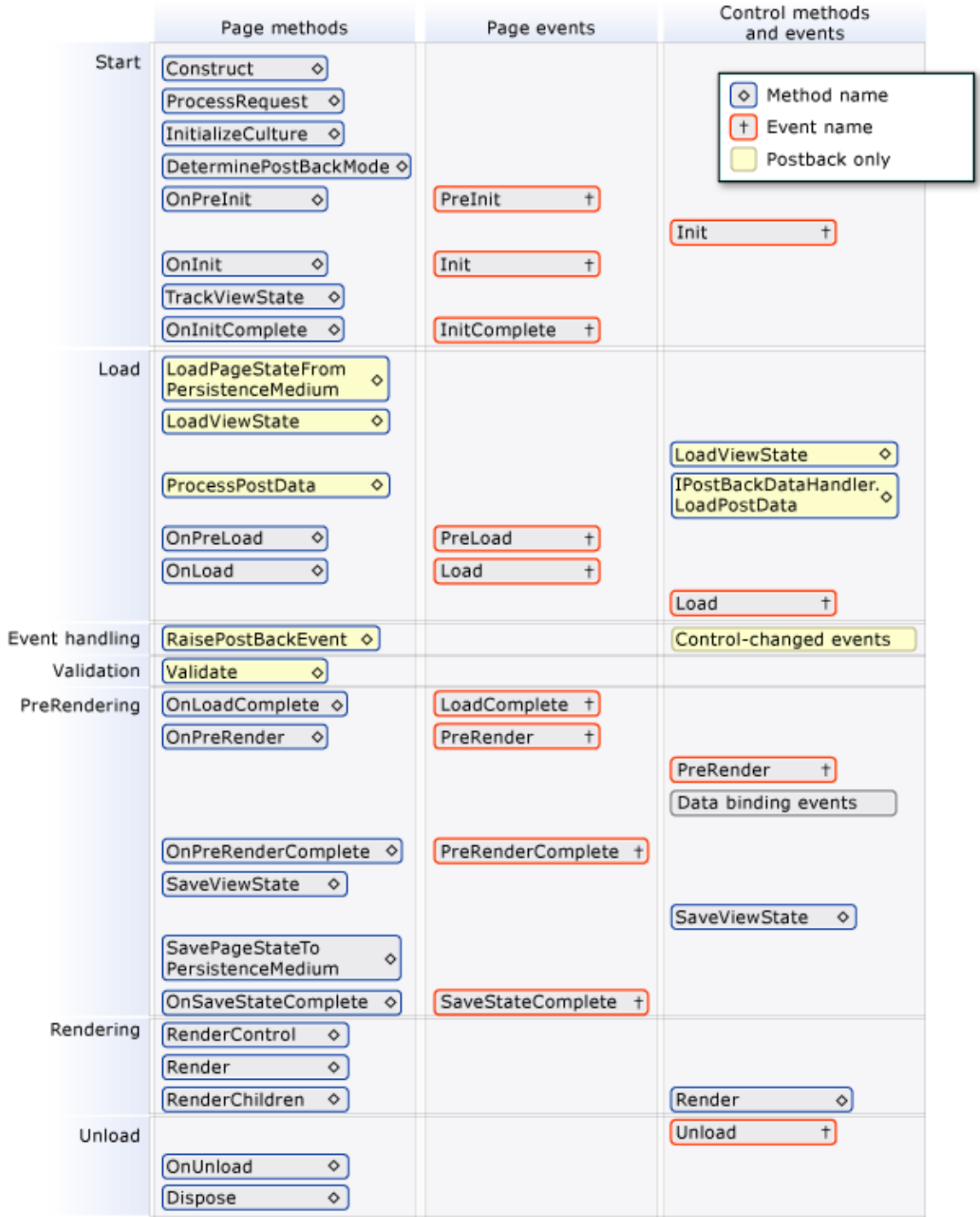


Figure 5-3: Page Life Cycle

5.5 Distribution of Modules

The main Modules of the application are:

5.5.1 AcProMS User Management Module

This module is responsible for user registration and authentication. Authentication is the process of obtaining identification credentials such as name and password from a user and validating those credentials against some authority. If the credentials are valid, the entity that submitted the credentials is considered an authenticated identity. Once an identity has been authenticated, the authorization process determines whether that identity has access to a given resource.

ASP.NET implements authentication through authentication providers, the code modules that contain the code necessary to authenticate the requestor's credentials.

Windows Authentication: Windows Authentication treats the user identity supplied by Microsoft Internet Information Services (IIS) as the authenticated user in an ASP.NET application. IIS provides a number of authentication mechanisms to verify user identity, including anonymous authentication, Windows integrated (NTLM) authentication, Windows integrated (Kerberos) authentication, Basic (base64 encoded) authentication, Digest authentication, and authentication based on client certificates. Windows Authentication is implemented in ASP.NET using the `WindowsAuthenticationModule` module. The module constructs a `WindowsIdentity` based on the credentials supplied by IIS and sets the identity as the current `User` property value for the application.

Windows Authentication is the default authentication mechanism for ASP.NET applications and is identified as the authentication mode for an application using the authentication configuration element, as shown in the following code example.

```
<system.web>
<authentication mode="Windows"/>
</system.web>
```

Form Authentication:Forms authentication enables you to authenticate the user name and password of your users using a login form that you create. Unauthenticated requests are redirected to a login page, where the user provides credentials and submits the form. If the application authenticates the request, the system issues a ticket that contains a key for reestablishing the identity for subsequent requests.

```
<system.web>
<authentication mode="Forms">
</authentication>
</system.web>
```

5.5.2 Project Management Module

This module provides the functionality to supervisor to add/modify the project proposal.

A syndicate should be able to choose one or more project of different supervisors. Module should allow the allocation of a project based on 1) First come first served 2) Approved by the supervisor.

Project Management Module must ensure that student can't remove project once allocated.

5.5.3 Syndicate Management

This module is responsible for the creation of syndicates (groups) by the authorized users.

This module assures, with the help of user management module, the authenticity of user.

Authorized user can change syndicates a later stage using functionality provided by this module.

5.5.4 WorkflowManagement Module

This module must provide functionality to project coordinator to set and modify timeline for different project activities.

System may provide functionality to create a news event and send an email to group leaders if due date of any submission is coming up.

5.5.5 User Interface Module

Since AcProMS is a web application so user interface was a highly important part of it and was treated as a separate module. The issues were to design interface in such a way that no functionality is hidden from user and data should be presented in a clear way to end user so nothing is missed by him.

The interface was developed using ASP, JQuery and AJAX. The interface was kept separate from business layer.

CSS 3.0 was used so that most old browsers are compatible with the web application but still a few features won't show up correctly on browsers older than Internet Explorer 6.0.

Jquery is an open source JavaScript library used for client side scripting (Writing the program of web application which will run on client side). JQuery was used for form validation in user interface, so when a user is creating a new project proposal, registering into the system or using any other form for entering data he shouldn't enter a wrong value or type. This will ensure data is in its correct format.

Ajax stands for asynchronous JavaScript and XML. It is also used to write client side code of a web application and main purpose is to make web applications interactive. Ajax was used in development of user interface for AcProMS. The reason to use Ajax was it sends

data to and retrieves from server asynchronously without affecting the web page or redirecting user to a new page.

5.6 Distribution of Classes with respect to Modules

All important functionality which is provided by classes are listed below.

5.6.1 Login Class

The user login class is a Windows Form which allows the administrator to login. It gives a link for the users according to their role to user other functionalities of system. Upon successful login the user is redirected to their home page.

5.6.2 Add User Class

The add user class allows the administrator to add new users to the **AcProMS** system. Before adding the user to the database it makes sure that the user has entered valid values and not left compulsory fields as empty. In case a compulsory field has been left empty it asks the administrator to correctly enter.

5.6.3 Update User Class

The Update user class allows the administrator to update the information about a user.

5.6.4 Add Project Proposal Class

The add proposal class allows the supervisor to add new project proposals for approval. Once projects are approved then it will be available for selection. Before adding the proposals to the database it makes sure that the user has entered valid values and not left compulsory fields as empty. In case a compulsory field has been left empty it asks the user to correctly enter.

```

#region " save Command"
if (btnAddSynopsis.CommandArgument.Length == 0)
{
using (FYPDataContext dc = newFYPDataContext())
{
PS master = (PS)this.Master;
tblSynop synopsis = newtblSynop();
synopsis.sTitle = txtTitle.Text.Trim();
synopsis.sExtendedTitle = txtExTitle.Text.Trim();
synopsis.sBriefDescription = eDescription.Text.Trim();
synopsis.sScope = eScop.Text.Trim();
synopsis.sAcademicObjectives = eObjective.Text.Trim();
synopsis.sEndGoal = eEndGoal.Text.Trim();
synopsis.sResources = eResources.Text.Trim();
synopsis.sStudents = txtStudents.Text.Trim();
synopsis.sWorkDone = eWorkDone.Text.Trim();
synopsis.sSpecialSkills = eSkills.Text.Trim();
synopsis.lSupervisorID = master.CurrentUser;
synopsis.bStatus = false;
synopsis.dtDate = DateTime.Now.Date;
dc.tblSynops.InsertOnSubmit(synopsis);
dc.SubmitChanges();
}
}
#endregion
#region " Update Command"
elseif (btnAddSynopsis.CommandArgument.Length > 0)
{
using (FYPDataContext dc = newFYPDataContext())
{
tblSynop synopsis = dc.tblSynops.FirstOrDefault(
x =>x.lSynopsisID == long.Parse(
btnAddSynopsis.CommandArgument));
synopsis.sTitle = txtTitle.Text.Trim();
synopsis.sExtendedTitle = txtExTitle.Text.Trim();
synopsis.sBriefDescription = eDescription.Text.Trim();
synopsis.sScope = eScop.Text.Trim();
synopsis.sAcademicObjectives = eObjective.Text.Trim();
synopsis.sEndGoal = eEndGoal.Text.Trim();
synopsis.sResources = eResources.Text.Trim();
synopsis.sStudents = txtStudents.Text.Trim();
synopsis.sWorkDone = eWorkDone.Text.Trim();
synopsis.sSpecialSkills = eSkills.Text.Trim();
dc.SubmitChanges();
}
}
#endregion

```

5.6.5 Choose project Class

This class is responsible for fetching all available projects and presents them to student. After that it provides student an interface through which student can select project. This class ensures that project can only be selected by group leader.

```
long? lSynopsisId = long.Parse(Request.QueryString["pid"]);
using (FYPDataContext dc = new FYPDataContext())
{
    var proposals = (from pro in dc.tblSynops
                     where (pro.lSynopsisID == lSynopsisId)
                     select new
                        {
                            pro.lSynopsisID
                        }).FirstOrDefault();
    Student master = (Student)this.Master;
    var group = (from grp in dc.tblFYPSyndicates
                 where grp.lStudentID == master.CurrentUser
                 select new { grp.lGroupID }).FirstOrDefault();
    tblChosenProject prjct = new tblChosenProject();
    prjct.lprojectID = proposals.lSynopsisID;
    prjct.lGroupID = group.lGroupID;
    prjct.dtDate = DateTime.Now.Date;
    dc.tblChosenProjects.InsertOnSubmit(prjct);
    dc.SubmitChanges();
}
```

5.7 Summary

Details of the classes implemented have been discussed in this chapter. ASP.NET and Visual C#.Net has been used as the programming language for the project due to its object-oriented and platform independent nature.

CHAPTER 6

Software Testing

Software Testing

6.1 Introduction

In this section of the document, system testing and load balancing test has been performed using different test cases. Software testing can also be stated as the process of validating and verifying that a software program/application/product 1) meets the business and technical requirements that guided its design and development 2) works as expected; and 3) Can be implemented with the same characteristics.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the Info-Tech Research Group adopted.

6.2 Functional Testing

“Software development organizations with an effective functional testing practice have a fast and objective way to determine whether each functional requirement is actually implemented in the code. With functional testing, the team translates functional requirements into executable test cases that confirm how well the code satisfies the requirements at any given time. It provides unprecedented objective insight into requirement status and prevents the missing or incorrect functionality implementations that can lead to countless rewrites (and then budget overruns and missed deadlines), user dissatisfaction, and project failure.”

(Functional Testing By Adam Kolawa, Co-Founder of Parasoft)

Functional testing typically involves five steps:

1. The identification of functions that the software is expected to perform.
2. The creation of input data based on the function's specifications.
3. The determination of output based on the function's specifications.
4. The execution of the test case.
5. The comparison of actual and expected outputs.

Team Member Assignments:

The roles each will play in the testing phase.

Name	High-Level Testing Assignments
Daud Azim Khan	Test case writing and execution for features ID: 1-15.
Haider Ali	Test case writing and execution for features ID: 1-15.
Muhammad Ali	Test case writing and execution for features ID: 1-15.

Scope: This section details the features that will be included in the functional testing phase(s) and those that will be excluded.

Features Included in Testing:

Feature ID	Description
1	Home Page Login and Password Fields
2	Add/Edit Course

3	Add Department
4	Add News
5	Add New User
6	Add/Remove member
7	Set Group Leader
8	Add New Group
9	Create User Through CSV
10	Create User Through Form
11	Add Activity
12	Approve Project
13	Choose Project
14	Add Project
15	Work Space

Functional Test Cases & Their Execution: The following test cases will be created and executed against the application.

For Feature ID # 1:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Access Home page and login with correct values	Login	Login	Pass
2		Access Home page and login with incorrect values	Not Login	Not Login	Pass

For Feature ID # 2:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Add Course	Course created successfully	Course created successfully	Pass
2		Edit Course	Course edited successfully	Course edited successfully	Pass

For Feature ID # 3:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Add Department	Department Added	Department Added	Pass

For Feature ID # 4:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Add News	News Added	News Added	Pass

For Feature ID # 5:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Add New User	User Added	User Added	Pass

For Feature ID # 6:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Add New Member	Member Added	Member Added	Pass
2		Delete Member	Member Deleted	Member Deleted	Pass

For Feature ID # 7:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Set Group Leader	Group Leader set	Group Leader set	Pass

For Feature ID # 8:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Add New Group	Group Added	Group Added	Pass

For Feature ID # 9:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Create User through CVS file	User Created	User Created	Pass

For Feature ID # 10:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Create User through form	User Created	User Created	Pass

For Feature ID # 11:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Add Activity	Activity Added	Activity Added	Pass

For Feature ID # 12:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Approve Project	Project Approved	Project Approved	Pass

For Feature ID # 13:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Choose Project	Project Chosen	Project Chosen	Pass

For Feature ID # 14:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Add Project	Project Added	Project Added	Pass

For Feature ID # 15:

S#	Steps	Input	Expected O/P	Output	Remarks
1		Add Workspace	Workspace Added	Workspace added	Pass

6.3 Summary

Testing not only maintains the software quality but also improves over all usability of the project. At different stages of development suitable testing techniques were used to ensure product works accurately and efficiently. All errors detected during testing were removed.

CHAPTER 7

Conclusion and Future Enhancements

Conclusion and Future Enhancements

7.1 Introduction

This chapter describes the possible future enhancements in the project. Suggestions have been presented in further sections for improvement of features and adding new features to AcProMS. AcProMS has been developed in such a way to make it easy to expand and reuse code. This will help in extension of AcProMS to cover further disciplines of project management and also how web services can be used to solve project management problems in an effective way. There are a lot of enhancements that can be done in order to make the system more effective and efficient in the face of providing flexibility and ease of inter-organization communication.

7.2 Conclusion

In our modern age computer has replaced human in nearly every field. The tasks that were accomplished manually are now being handled by computerized systems to minimize the human flaws and time consumption. In recent years web applications are on rise and have emerged to hold ground in the every field of information especially business and academic sector. We have developed our graduation project keeping this thing in mind. The activities of final year projects are handled manually in all of the NUST campuses. AcProMS is designed to facilitate the final year project and thesis management system with minimum human intervention.

Our system is an efficient, user-friendly, self-explanatory web-based system that manages final year projects and thesis in a secure way. The system can be modified to use in any academic organization as well as in non-academic organization. Overall **AcProMS** serves as the robust and dynamic web application for the educational organizations to meet their needs of project management.

7.3 Future Enhancements of AcProMS Functionalities

The project can be extended to manage the activities of projects in any organization. Here in our application we have different roles. In organization these roles can be replaced by project manager, project leader, project team members. AcProMS is already giving functionality of adding new roles. So doing some modification and enhancement this project can be used as any project management system.

As this project handles information flow and is developed for educational activities so this can be extended to develop a learning management system for universities and colleges. AcProMS provides the functionality of news posting and other interaction methods such as chatting so by modifying its functionalities it can be used as a learning management system. Currently we have text chatting between different users. This chatting functionality can be enhanced to video chat.

A mobile application can be developed and integrated with the current system to facilitate the user to access features of the system through their mobile phones.

AcProMS can be extended to be a web service so that other developers or organizations can use the functionalities of the system using it as a web service.

7.4 Summary

Web services can be used conveniently to help us in real-life problems like academic projects management. There is a wide range of options in web services to allow organizations to collaborate and work towards the common goal of project management. This will not only help in better management of project activities but also prevent projects from failing because better management leads to project success.

APPENDIX A

Bibliography

References

- [1] <http://pmstudent.com/what-is-a-project/>
- [2] Dimitrijević, Tanja. "Managing European Projects with the Aid of Web Applications". Diss. UniverzanaPrimorskemFakultetaza Management, Koper, 2010
- [3] <http://www.projecttimes.com/project-management-software/536-4pm-for-project-management/view-details.html>
- [4] Ilijaš, Kopal 2009. "Making informed decisions based on your project's output". International Business Conference on Business Process Management, Ljubljana
- [5] <http://www.smashingapps.com/2010/02/15/10-free-tools-for-effective-project-management.html>
- [6] <http://training.dotproject.net/mod/resource/view.php?id=120>
- [7] http://docs.mantisbt.org/master/en/administration_guide.txt
- [8] http://docs.mantisbt.org/master/en/administration_guide.html#AEN2392
- [9] http://docs.mantisbt.org/master/en/administration_guide.html#AEN2365
- [10] <http://www.brothersoft.com/mantis-bug-tracking-jumpbox-131829.html>
- [11] http://www.scripts.com/script_list
- [12] <http://www.websitesource.com/software/php-projekt.php>
- [13] The ITG Evangelist, Steve Romero: Cloud Computing and Governance. July 7, 2010.
- [14] www.ipedr.com/vol10/110-S10096.pdf
- [15] <http://producingoss.com/en/vc.html>, and Karl Fogel, Producing open source software: how to run a successful free software project
- [16] <http://www.developer.com/open/article.php/3902371/Subversion-vs-Git-Choosing-the-Right-Open-Source-Version-Control-System.htm>

[17] <http://thinkvitamin.com/code/why-you-should-switch-from-subversion-to-git/>

[18] <http://marc.info/?l=git&m=131682109207335&w=2>

APPENDIX A-1
USER MANUAL

AcProMS

TABLE OF CONTENTS

1. Reading Instructions.....	
2. Installations.....	
3. How to use the system.....	
4. Procedures.....	
4.1 Student Workspace.....	
4.2 Supervisor Workspace.....	
4.3 Deploy Web Application.....	

1. Reading Instructions

This Manual is a guide to the system “AcProMS”. It contains essential instructions for setup and operations. The system provides a user friendly interface which allows you to directly interact and monitor the system.

This Manual should be read in the order given.

2. Installation

Here are the basic tasks that you have to do in order to deploy to a third-party hosting company:

1. Set up an account with a hosting company.
2. Create a publish profile. A publish profile specifies the server you are deploying to, credentials for logging on to the server, databases to deploy, and other options. The hosting company sends you the settings you need for the publish profile, either in an email or in a file that you can import into Visual Studio.
3. Start the deployment process by clicking the one-click publish button. The Publish button is in the Publish Web wizard that you use to create publish profiles. It is also available in the Web One Click Publish toolbar that you can choose from the View–Toolbars menu in Visual Studio.

3. How to use the system

Operation of AcProMS comprises of following steps:

1. Login to Application
2. There is a menu according to each role. Select appropriate option to perform that process.

These steps are in order and they encapsulate the application logic. User need not go into the detail. If detail analysis is required user should read the technical thesis document.

The Applications provided by the system are:

- Add Activity
- Add Projects
- Select Project
- Add Role
- Add Deliverables
- Assessment

4. Procedure

The underlying procedures are specific for each application.

4.1 Add Projects

This Feature provides the ability to Supervisor publish his/her project idea and this synopsis will be approved by respective authorities. After that these synopsis will be available to students to choose.

4.2 Select Project

This feature will facilitate the students to choose a project of their interest. Students can reserve more than one project at a time of different supervisors.

4.3 Create Activity

This feature will facilitate the Project Coordinator to create an activity of specific event. He can add start and end time of that activity and he can also modify the activity according to circumstances.

AcProMS

HOME PROJECTS PROFILE

Awais Majeed Log Out

- Add Proposals
- Proposal Status
- Selected Projects
- Projects in Progress
- News And Activities
- 2012 Project Schedule

June 2012						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Academic Project Management System

Group Information

Name	Course	GPA
NC Hamza Javed	BESE-15	3.100
NC Manahil	BESE-15	2.100

Upload Resources

Title:

Attach File: Browse...

Upload

Assign Task

Title:

Deadline:

Resource Person:

Deliverable: Yes No

Add

Title	Complete(%)	File	Submitted By
Final Thesis	100	Assignments Final Thesis compete78.docx	NC Hamza Javed

Contacts (0)

AcProMS

HOME PROJECT WORKSPACE PROFILE

NC Muhammad Umair Log Out

- Project Proposals
- Chosen Projects
- Project Activities
- 2012 Project Schedule

June 2012						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Target Recognition in Autonomous Tank

Your Group Members

Name	Course	GPA
Capt. Adnan Hussain	BESE-15	2.550

Upload Resources

Title:

Attach File: Browse...

Upload

Assigned Tasks

Task Title:

Complete: %

File: Browse...

Submit

Uploaded Resources

[Research Paper on Target Recognition in Autonomous Tank Uploaded By Nayeed Rao](#)

[Help on Object Recognition Uploaded By Capt. Adnan Hussain](#)

Contacts (0)

The Installation/Deployment of Web Application

The preceding tutorial looked at how to deploy a simple ASP.NET web application to a web host provider. Specifically, the tutorial showed how to use an FTP client like FileZilla to transfer the necessary files from the development environment to the production environment. Visual Studio also offers built-in tools to facilitate deployment to a web host provider. This tutorial examines two of these tools: the Copy [Web Site](#) tool, where you can move files to and from a remote web server using FTP or FrontPage Server Extensions; and the Publish tool, which copies the entire website to a specified location.

Note: Other deployment-related tools offered by Visual Studio include [Web Setup Projects](#) and [Web Deployment Projects](#) Add-In. Web Setup Projects package a website's contents and configuration information into a single MSI file. This option is most useful for websites that are deployed within an intranet or for companies that sell a pre-packaged web application that customers install on their own web servers. The Web Deployment Projects Add-In is a Visual Studio Add-In that facilitates specifying configuration differences between builds for development environments and production environments.

Deploying Your Site Using the Copy Web Site Tool

Visual Studio's Copy Web Site tool is similar in functionality to a stand-alone FTP client. In a nutshell, the Copy Web Site tool allows you to connect to a remote web site through FTP or FrontPage Server Extensions. Similar to FileZilla's user interface, the Copy Web Site user interface consists of two panes: the left pane lists the local files while the right pane lists those files on the destination server.

Note: The Copy Web Site tool is only available for Web Site Projects. Visual Studio does offer this tool when you are working with a Web Application Project.

Let's take a look at using the Copy Web Site tool to publish the Book Review application to production. Because the Copy Web Site tool only works with projects that use the Web Site Project model we can only examine using this tool with the BookReviewsWSP project. Open that project.

Launch the Copy Web Site tool project by clicking the Copy Web Site icon in the Solution Explorer (this icon is circled in Figure 1); alternatively, you can select the Copy Web Site option from the Website menu. Either approach launches the Copy Web Site user interface shown in Figure 1; only the left pane in Figure 1 is populated because we have yet to connect to a remote server.

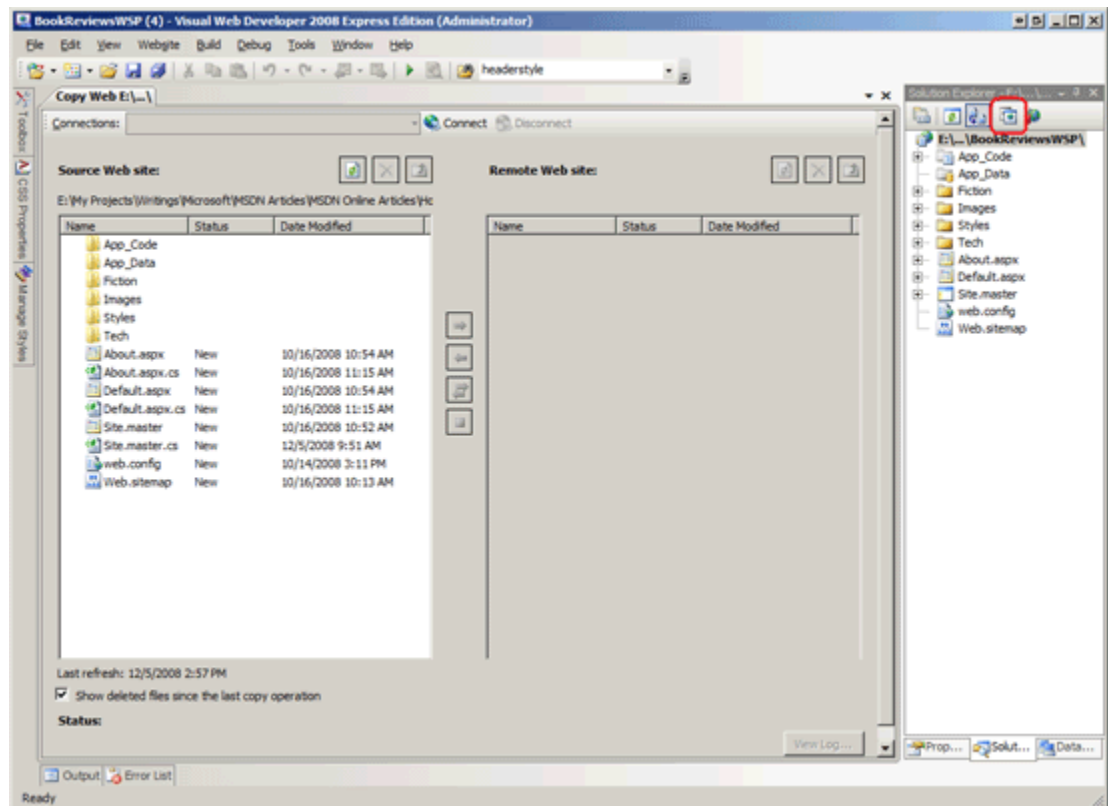


Figure 1: The Copy Web Site Tool's User Interface is Divided Into Two Panes

In order to deploy our site we need to first connect to the web host provider. Click the Connect button at the top of the Copy Web Site user interface. This displays the Open Web Site dialog box shown in Figure 2.

You can connect to the destination website by selecting one of the four options from the left:

- **File System** - select this to deploy your site to a folder or network share accessible from your computer.
- **Local IIS** - use this option to deploy the site to the IIS web server installed on your computer.
- **FTP Site** - connect to a remote web site using FTP.
- **Remote Site** - connect to a remote website using FrontPage Server Extensions.

Most [web host](#) providers support FTP, but fewer offer FrontPage Server Extension support. For that reason, I've selected the FTP Site option and then entered the connection information as shown in Figure 2.

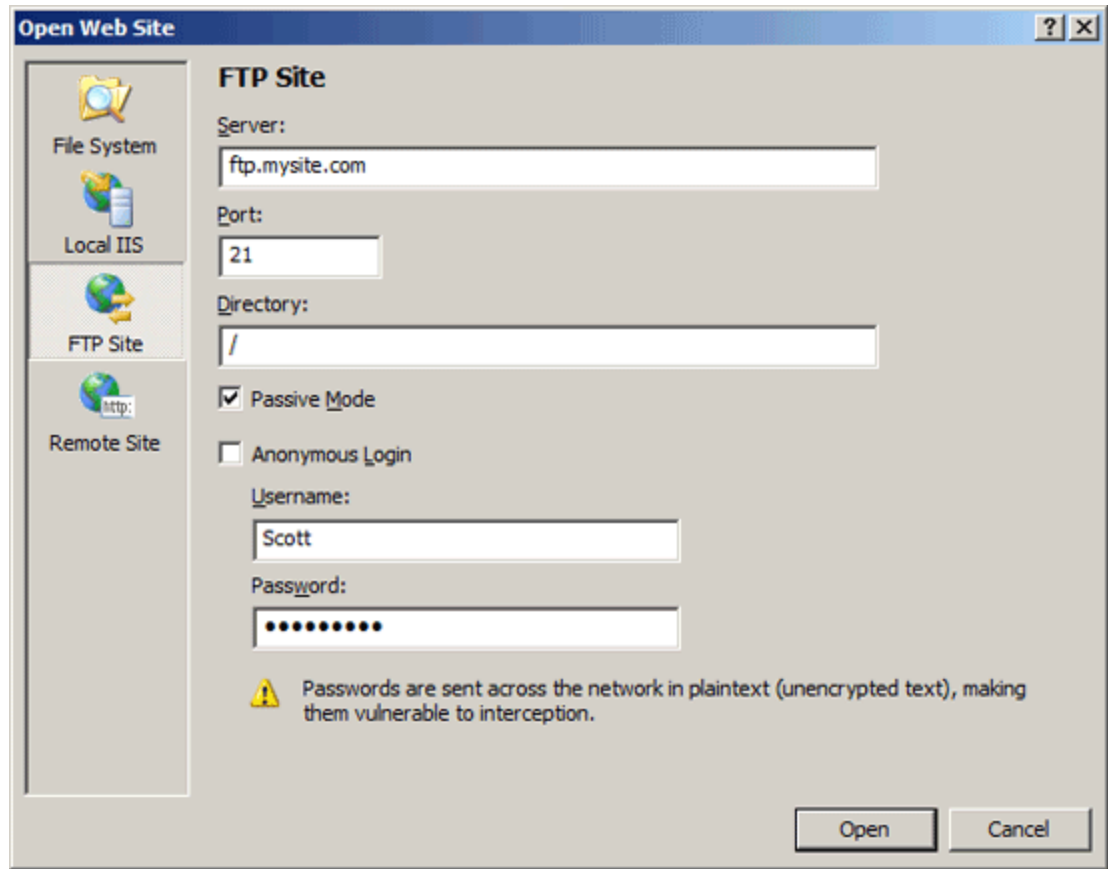


Figure 2: Specify the Destination Website

After you connect, the Copy Web Site tool loads the files at the remote site in the right pane and indicates the status of each file: New, Deleted, Changed, or Unchanged. You can copy a file from the local site to the remote site, or vice-versa.

Let's add a new page to the BookReviewsWSP project and then deploy it so that we can see the Copy Web Site tool in action. Create a new ASP.NET page in Visual Studio in the root directory named Privacy.aspx. Have the page use the master page Site.master and add your site's privacy policy to this page. Figure 3 shows Visual Studio after this page has been created.

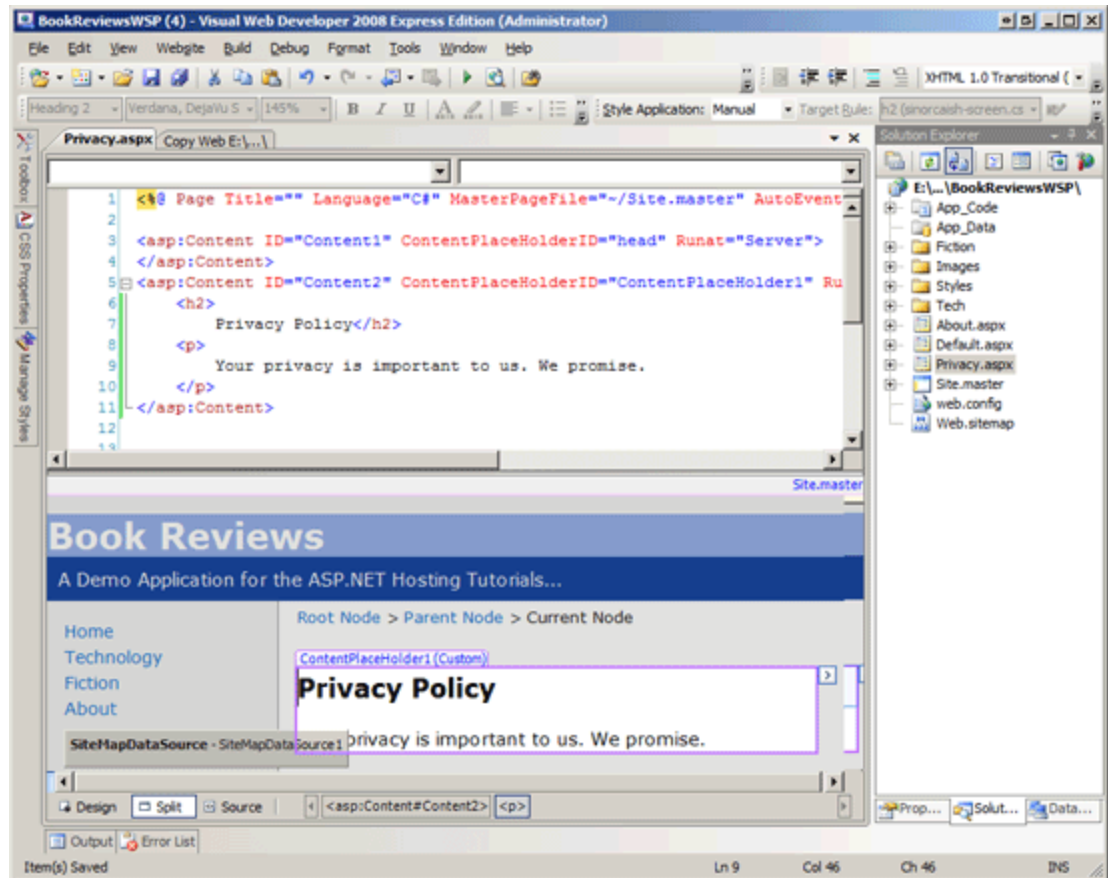


Figure 3: Add a New Page Named Privacy.aspx to the Website's Root Folder

Next, return to the Copy Web Site user interface. As Figure 4 shows, the left pane now includes the new files - Policy.aspx and Policy.aspx.cs. What's more, these files are marked with an arrow icon and a Status of New, indicating that they exist on the local site but not on the remote site.

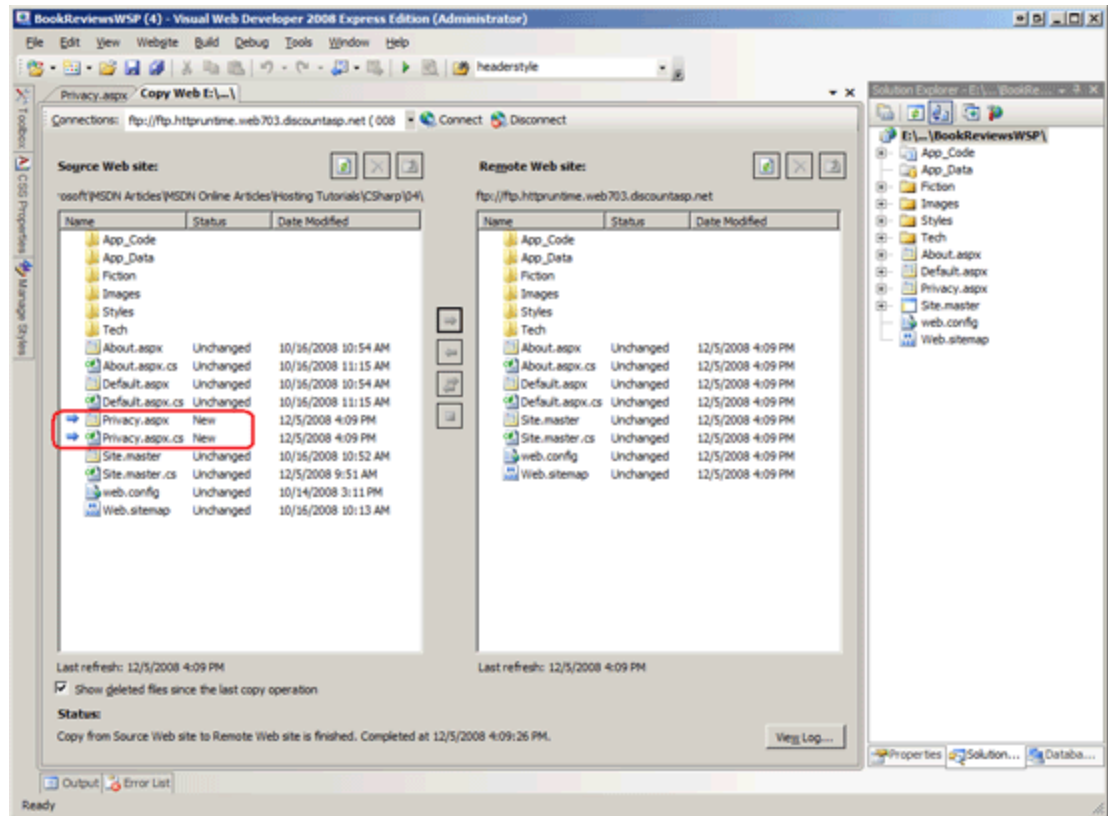


Figure 4: The Copy Web Site Tool Includes the New Privacy.aspx Page in its Left Pane

To deploy the new files select them and then click the arrow icon to transfer them to the remote site. After the transfer completes the Policy.aspx and Policy.aspx.cs files exist on both the local and remote sites with the status Unchanged.

Along with listing new files, the Copy Web Site tool highlights any files that differ between the local and remote sites. To see this in action, return to the Privacy.aspx page and add a few more words to the privacy policy. Save the page and then return to the Copy Web Site tool. As Figure 5 shows, the Privacy.aspx page in the left pane has a status of Changed indicating that it is out of sync with the remote site.

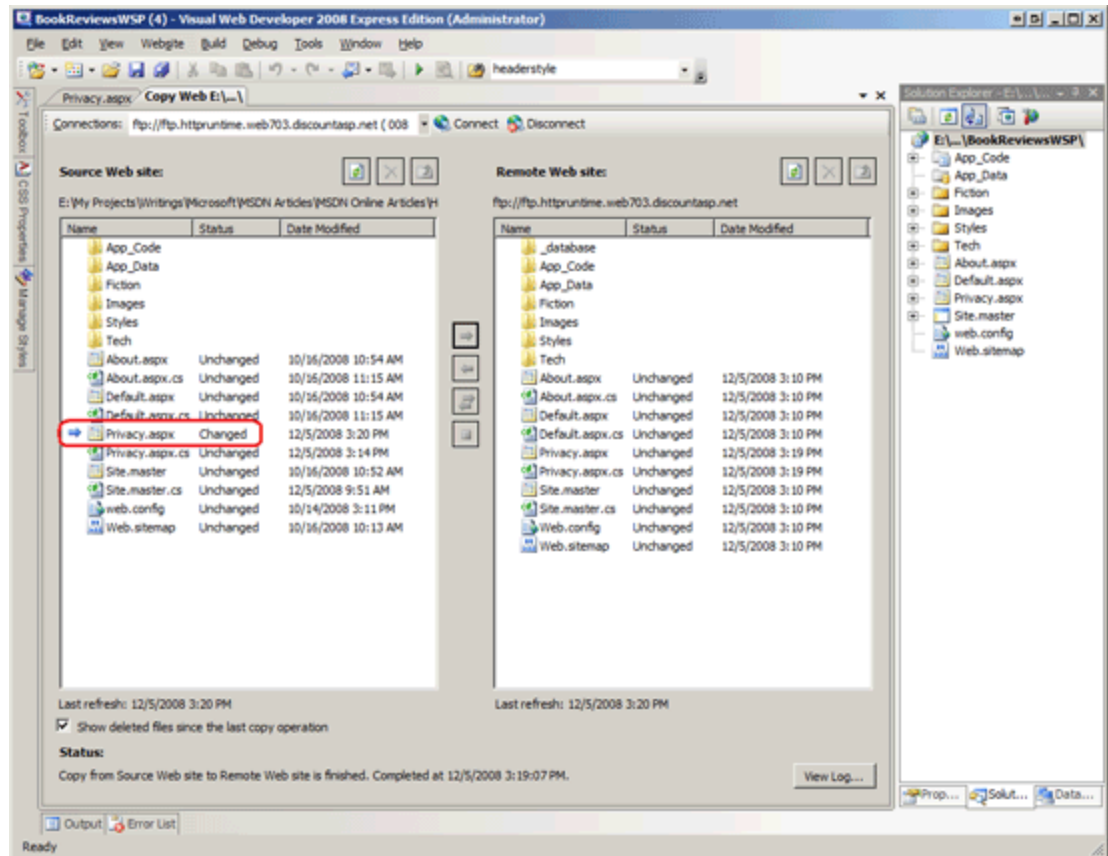


Figure 5: The Copy Web Site Tool Indicates that the Privacy.aspx Page has been Changed

The Copy Web Site tool also indicates if a file has been deleted since the last copy operation. Delete the Privacy.aspx from the local project and refresh the Copy Web Site tool. The Privacy.aspx and Privacy.aspx.cs files remain listed in the left pane, but have a Deleted status indicating that they have been removed since the last copy operation.

Publishing a Web Application

Another way to deploy your web application from within Visual Studio is to use the Publish option, which is accessible via the Build menu. The Publish option explicitly compiles the application and then copies all of the necessary files up to

the specified remote site. As we'll see shortly, the Publish option is more blunt than the Copy Web Site tool. Whereas the Copy Web Site tool lets you examine the files on the local and remote sites and permits you to upload or download individual files as needed, the Publish option deploys the entire web application.

In addition to copying all of the needed files to the specified remote site, the Publish option also explicitly compiles the application. Given that Web Application Projects need to be explicitly compiled it should come as no surprise that the Publish option is available for Web Application Projects. What may be a bit surprising is that the Publish option is also available for Web Site Projects. As noted in the [Determining What Files Need to Be Deployed](#) tutorial, Web Site Projects can be explicitly compiled through a process referred to as *pre-compilation*. This tutorial focuses on using the Publish option with Web Application Projects; a future tutorial will explore pre-compilation, at which point we'll return to look at using the Publish option with Web Site Projects.

Note: While the Publish option is available in Visual Studio for both Web Site Projects and Web Application Projects, Visual Web Developer only offers the Publish option for Web Application Projects.

Let's look at deploying the Book Reviews application using the Publish option. Start by opening BookReviewsWAP (the Web Application Project) in Visual Studio. From the Publish menu choose the Build BookReviewsWAP project. This brings up a dialog box that prompts for the target location, among other configuration options (see Figure 6). Much like with the Copy Web Site tool you can enter a location that points to a local folder, a local website on IIS, a remote

website that supports FrontPage Server Extensions, or an FTP server address. You can choose whether to replace the files on the remote web server with the deployed files or to delete all of the content on the remote site before publishing. You can also specify whether to copy:

- Only the files in the project needed to run the application, which omits the unneeded source code and project-related files.
- All project files, which includes the source code files and Visual Studio project files like the Solution file.
- All files in the source project folder, which copies all files in the source project folder regardless of whether they're included in the project.

There's also an option to upload the contents of the App_Data folder.

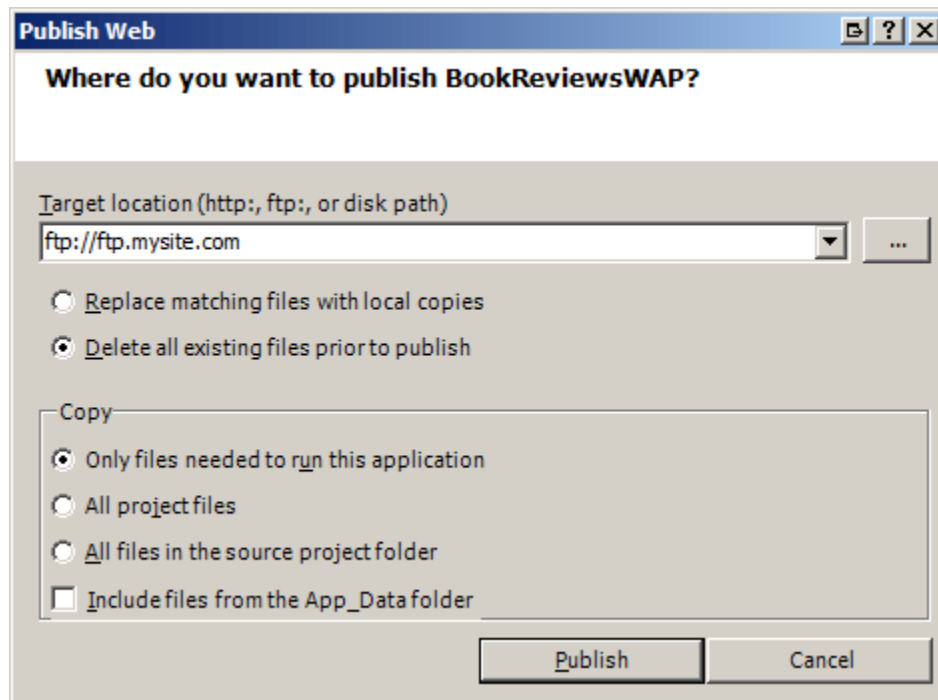


Figure 6: Specify the Destination Website

For the Book Review application the remote site contains the files deployed when copying the BookReviewsWSP project via the Copy Web Site tool. Therefore, let's have the Publish option start by deleting all existing content. Also, let's just copy the necessary files rather than cluttering the production environment with unneeded source code and project files. After specifying these options, click the Publish button. Over the next several seconds Visual Studio will deploy the necessary files to the destination site, displaying its progress in the Output window.

Figure 7 shows the files on the FTP site after the Publish operation has completed. Note that only the markup pages and the necessary sever- and client-side support files have been uploaded.

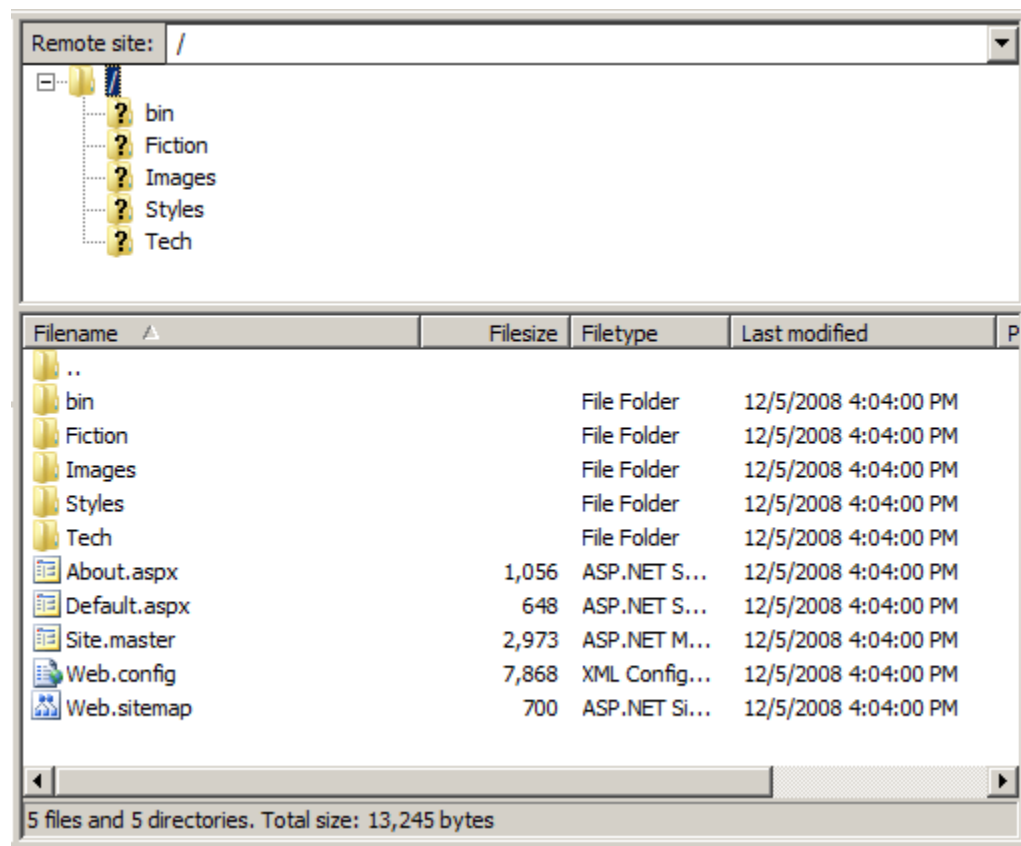


Figure 7: Only the Needed Files Were Published to the Production Environment

The Publish option is a less nuanced tool than the Copy Web Site tool. Whereas the Copy Web Site tool allows you to inspect the files on the local and remote sites and see how they differ, the Publish option provides no such interface. Moreover, the Copy Web Site tool enables you to make one-off changes, uploading or deleting individual files. The Publish option does not allow such fine-grained control; instead, it publishes the *entire* application. This behavior has its pros and cons. On the plus side, you know when using the Publish option you won't be forgetting to upload an important file. But consider what happens if you have made a small change to a very large website - with the Publish option you cannot update that page or two that has been modified, but instead you must wait while Visual Studio deploys the entire site.

It's not uncommon for there to be certain files whose content differs between the production and development environments. A key example is the application's configuration file, `Web.config`. Because the Publish option blindly copies the web application files it overwrites the production environment's customized configuration files with the version in the development environment. The subsequent tutorial explores this topic further and offers tips for deploying a web application when such differences exist.