Enhancing the Value of the Capstone Experience Course

Dana Schwieger
dschwieger@semo.edu
Department of Accounting and MIS

Ken Surendran
ksurendran@semo.edu

Department of Computer Science
Southeast Missouri State University
Cape Girardeau, MO 63701, USA

Abstract

Capstone courses in MIS, CIS and CS programs are commonly incorporated into curriculum these days as they can be used effectively to assess several program objectives. In this paper, the authors discuss the usefulness of a combined Capstone Experience class consisting of CS/CIS and MIS students. The authors point out how this class provided considerable opportunity for collaborative learning, served as an instrument for assessing the respective program objectives relating to both technical and professional (soft) skills, and met the learning outcomes of the university's general educational requirements.

Keywords: Capstone experience, collaborative learning, CS/CIS and MIS programs, education strategies

1. INTRODUCTION

In recent years, programs in CS, CIS and MIS have all suffered from a drought in student enrollment. As a result, upper level courses have very small class sizes, often too small to meet the university's minimum class-size standard for offering a course. Last semester, such a situation forced two programs (CS/CIS and MIS) at the authors' institution, to combine capstone course enrollments.

2. BACKGROUND

A few years ago, the authors' university wanted to reduce the overall total number of program credit-hours for most of the University's programs by four credits. The purpose behind the credit hour adjustment was to assist undergraduate students in being able to complete a bachelor's degree within eight semesters. This issue forced the Computer Science Department at the authors' institution to offer the capstone course as a Capstone Experience course under the general educational category. In addition to meeting the capstone course requirements of the computer science discipline, this would also meet some of the nine general educational learning outcomes.

Offering the course under the general education umbrella required minimal adjustment since most of the learning outcomes were already addressed. The general education objectives (GEOs) include:

1. Demonstrate the ability to locate and gather information;
2. Demonstrate capabilities for critical thinking, reasoning, and analyzing;
3. Demonstrate effective communication skills;
4. Demonstrate an understanding of human experiences and the ability to relate them to the present;
5. Demonstrate an understanding of various cultures and their interrelationships;
6. Demonstrate the ability to integrate the breadth and diversity of knowledge and experience;
7. Demonstrate the ability to make informed, intelligent value decisions;
8. Demonstrate the ability to make informed, sensitive aesthetic responses; and
9. Demonstrate the ability to function responsibly in one's natural, social, and political environment (Southeast Missouri State University Studies Handbook, 2005-2006).

When the University's MIS program was faced with cancelling the MIS capstone course as a result of low enrollment numbers, combining the CS/CIS and MIS capstone course seemed to fit the general education's interdisciplinary approach.

In Spring 2009, such a Capstone Experience course (cataloged as a University Studies course) was offered in which ten CS/CIS students and three MIS students enrolled. The main prerequisites for this course included senior standing and knowledge of Software Engineering or Systems Analysis and Design.

In Section 4, the authors provide all of the details pertaining to this course, including course description and assessments. In Section 5, they discuss the usefulness of this course both from the student perspective and program assessment perspective. In the following section, they provide a brief literature review pertaining to the capstone experience and its use.

3. LITERATURE REVIEW

In this section, the authors examine previous research addressing the collaborative learning approach to the undergraduate capstone experience. Several authors note the value provided to students, future employers and faculty through experiential learning opportunities.

3.1 Technology Graduates

The value of an interdisciplinary approach to the capstone experience can be observed in a number of previous research contributions. Bullen et al. (2009) noted the importance of both non-technical and technical skill sets as well as real world experience in the IT graduates' background. In a 2005 article written by Beachboard and Beard, the authors argue that "…a majority of existing business-oriented MIS/IS curricula emphasize skills and knowledge associated with IS strategy and business application development and tend to overlook the skills and knowledge required to design, support, and manage IS infrastructure and operations" (p. 317). The authors examined requirements set forth by the IS 2002 model curricula and then described a capstone course at Idaho State University that emphasized enterprise IS operations from a managerial perspective. The course was oriented toward the management of IT and attempted to address the following objectives (Beachboard & Beard, 2005):

- Develop IT strategy aligned with business strategy;
- Evaluate common IS management practices;
- Examine IS governance and its relation to business strategy;
- Investigate IS management practices associated with IS infrastructure and service management;
- Explore the effects of culture and group dynamics; and
- Examine the challenges associated with potential future workplace ethical issues.

To address these needs in the classroom, the authors incorporated "real world" examples, encouraged students to share their personal experiences related to topics covered, assigned practitioner-oriented readings and utilized group projects involving the practitioner community (Beachboard & Beard, 2005).

Wei, Siow, and Burley (2007) and Denton and Spangler (2001) also noted the value of experiential learning projects in an IS education. Wei, Slow, and Burley (2007) described their own experiences and suggested a comprehensive approach to implementing experiential learning experiences in information systems and technology management courses. They provided a conceptual model that outlined guidelines for faculty, student teams, clients and assessment (Wei, Slow, & Bruley, 2007). Denton and Spangler (2003) examined the use of a pre-capstone experience to identify areas of educational weakness in their MIS students. They also examined how the multiple skill levels of the student teams increased the richness of the learning experience.

In the capstone IS course offered at Ohio University, the program attempted to provide a comprehensive experience for the students addressing soft skills, experiential learning, conceptual elements as well as career readiness (McGann & Cahill, 2005). The authors noted the overall success that the integrative approach had garnered. Not only was that course able to pull together, and practically apply, the focus of their IS program, but the students also left the course with an overall feeling of preparedness (McGann & Cahill, 2005). Similar observations were made in the Towson University's masters of Applied Information Technology program (Lazar, 2003). Students applied their Web site development skills to enhance web accessibility awareness among local community organizations (Lazar, 2003). The opportunity provided students with a platform for applying their skills and knowledge.

Alexander (2001) noted the benefits and success factors that were experienced from experiential learning projects assigned in an Accounting Information Systems course at the University of Wisconsin-La Crosse. The author noted that, to facilitate success, faculty should make sure that projects have clear specifications, have realistic expectations of the students' capabilities, encourage the students to communicate both orally and through written proposals, safeguard the client's data, make sure students back-up work, and keep in touch with the client after the project had ended to help with minor system problems (Alexander, 2001).

Saulnier (2003) advocated the importance of experiential learning opportunities in the systems analysis and design course. He noted that, "We can no more teach Analysis & Design without having our students do Analysis & Design than we can teach our students swimming without having them get in the water" (p. 4). One of the benefits of experiential learning projects that Salunier found was that students learned about the process of learning which could help to make them better lifelong learners (2003).

3.2 Success Factors

Wilcoxon and Zigurs (2003) described a systematic approach to addressing experiential learning projects in IS classes to encourage a successful experience. The authors noted that the stakeholders provided a significant factor to the success of the experience.

Another important aspect of any educational project is a good assessment tool. Goode and Teh (2005) offer a peer-oriented evaluation instrument that they have used successfully, over several revisions. With this tool, students evaluate their team members by examining quality of work performed, timeliness of submissions, level of effort expended, willingness to work in the group and level of contribution broken down into labor, intellectual input, and overall contribution (Goode and Teh, 2005).

Guthrie and Navarrete (2004) examined students' attitudes regarding experiential learning opportunities. They created a survey instrument and distributed it to students both before and after an experiential learning project. The authors found that the students' opinions changed, both positively and negatively, as a result of the project. Overall, the students surveyed felt that their soft and technical skills improved after the experience. The students' opinion towards drawing an analogy between the project and future work also increased (Guthrie & Navarrete, 2004). Experiential learning opportunities provide opportunities for students to apply and build upon the knowledge that they have gained in the classroom setting.

4. CAPSTONE EXPERIENCE COURSE

The Capstone Experience course offered at the authors' institution to senior level students, has a broad theme - *building group-based solutions for real-world problems.* This theme can be achieved by having the students, working in teams, complete client sponsored system development projects. In this course, the students apply the principles, techniques / heuristics, processes and tools for developing a product such as quality software with all of the documentation for a client. The students learn these principles and techniques in an earlier systems development concept course so that they can apply those concepts in the Capstone Experience course.

Both the CS/CIS and MIS students take a systems development concept course in which they learn systems analysis and design before taking the Capstone Experience course. They do not, however, take the same concepts course. The CS/CIS students learn the Object Oriented approach (under the course title Software Engineering) and the MIS students learn the procedure centric systems analysis and design approach. The MIS students do get a brief (3-week) introduction to the object oriented approach. Thus, all the three majors have a background in the primary workflows (requirements analysis, design specification, coding and testing) of the system development process before taking the Capstone Experience course.

Following this concept course, the CS/CIS students take the Capstone Experience course. In 2009, the MIS students also took the Capstone Experience course offered by the Computer Science Department (they previously had taken the System Implementation and Practice course offered in the MIS program).

4.1 Aim and Learning Outcomes

The aim of the course is to apply the concepts and techniques learned in the Software Engineering / Systems Analysis and Design course for developing quality software products and to examine the supportive workflows such as project management, quality management, and configuration management. To achieve this, the participants, working in groups on a client-sponsored project, apply the techniques and standard tools, object-oriented or procedural, to develop a software product together with all the documentation and other system artifacts. They interact with the client to understand the application domain. In addition, they are exposed to recent developments in the field of system development (software engineering) through a survey of literature and classroom presentations.

There are two sets of learning outcomes for this course: General Education and Program (computing curricula). Of the nine general education outcomes (GEOs) stated in the introduction section, the first three and the sixth outcome are emphasized in this course. In regards to the computing curricula, the following outcomes are emphasized in this course:

1. Discuss project management and communications management issues in software development.
2. Discuss the various testing concepts for establishing quality assurance
3. Apply the knowledge from their major discipline to develop a software product for a client together with all the documentation and other system artifacts through teamwork.
4. Orally present the intermediate system artifacts (generated during analysis and design) for review and evaluation.
5. Create analysis and design documentation pertaining to the system they are developing

The System Implementation and Practice course offered to MIS majors (which was substituted by the CS/CIS Capstone Experience course in Spring 2009) has the following learning outcomes:

1. Obtain practical experience with working on an information systems development project in a team environment;
2. Be able to apply concepts and techniques for developing quality software products;

3. Be familiar with supportive workflows such as project management, quality management, communication management and configuration management;
4. Be able to create functional user documentation and other system artifacts;
5. Be familiar with recent developments in the field of software development.

In the above list of outcomes, the team work is explicit. Other than that, the first four outcomes map fairly well with those of the Capstone Experience course, even though the wordings are different. The last outcome, though not specifically listed in the Capstone Experience, has an assessment to address this requirement explicitly.

4.2 Assessments

In order to demonstrate the achievements of the above learning outcomes, a variety of assessments are used in this course. The first two computing curricula outcomes are verified through two closed-book exams. The third outcome is measured through the following evidences (assessments):

E1. Orally present the project highlights to the members of academic board and faculty.

E2. Demonstrate the final product to the clients, academic advisory board members and faculty members in a show-tell fashion

E3. Provide a project report delivered in three stages. (The details of the deliverables are in Appendix-A.)

E4. Prepare a final project report that has both system details and user notes. The project report includes individual logs and reflections as well (details in Appendix-A.)

Item-E4 in the above listing includes analysis and design documents that are used for evaluating and demonstrating the accomplishment of the next (E5) evidence. Three in-class presentations are used for verifying the achievements in the fourth (E4) outcome: (The instruments used for evaluating these presentations are provided in Appendix-B.)

E5. Presentation of Use Case Diagram or Context and Level-0 Data Flow diagrams

E6. Presentation of Use Case Model or DEF and ERD

E7. Presentation of User Interfaces

Many of these assessments automatically address some of the general education outcomes listed in Section 1 as well. For instance, achievements in GEO-#3 can be demonstrated by E1 through E7. The requirements specification and design specifications (including design decisions) help measure achievements on capabilities for critical thinking, reasoning and analyzing (GEO-# 2). In order to demonstrate GEO-# 1 (locate and gather information) a research presentation is included:

E8. Research on any one of the following topics and make a presentation to the class.

The purpose of this assignment is to help students acquire a better understanding of the topics that were not considered in detail during the lecture sessions. It recognizes the fact that system development is an evolving area and that several of the new developments are not considered in the concept course. Hence, a list of topics is provided from which each student team will select one. In this presentation, each member contributes one or two aspects. Example topics for the presentations include: Aspect Oriented Programming, Agile methodology - eXtreme Programming, CoCoMo – Cost Models, Service Oriented Architecture, and recent developments in UML. This assessment also serves as an instrument for the last outcome listed for the MIS System Implementation and Practice course.

4.3 Plan and Schedule

A course of this nature requires considerable planning. To begin with, the projects from clients are solicited during the fall semester. Ideal projects are those that are not time critical, that were put on ice for some reason (may be resource constraints), or that were left behind as alternatives but require a proof of concept (may be with a proto-type). Project allocation is completed before the end of the fall semester (this gives the teams time to learn new products that may be used in the project). The schedule for topics and assessments are shown in Appendix-C. Broadly stating, the students work on the analysis and design of the system during the first eight weeks of the semester and on implementation in the second eight weeks. During the first half, they also learn and take exams on the support process, present their results orally for the other student teams to review, and prepare intermediary project reports. During the second half of the course, the students make the research presentation, complete the project, make the project presentation, demonstrate the system to evaluators, and prepare the final project report.

5. BENEFITS OF THE COURSE

The Capstone Experience course serves several purposes. Students are given the opportunity to integrate various knowledge elements learned from different courses and apply them to meeting real-world system requirements and offering evidence to assess program objectives. Team projects offer opportunities to gain some real-world experience, to develop soft-skills, and to get into the habit of life-long learning in a cooperative fashion. These benefits are discussed in this section.

5.1 Knowledge Integration

In most of their earlier courses, students have been carrying out simple assignments addressing specific knowledge elements. For the most part, these courses address specific topic areas in relative isolation. Through the Capstone Experience, students see the value of the different courses come together such as programming, database, networking, and technical writing through the development of a system solution for meeting a practical problem. They get to interact with a client, hone data collection and interview skills, learn from others, take on responsibilities through team work-share, and to face their personal areas of weakness.

5.2 Co-operative Learning

It is not possible for any computing program to provide all of the required knowledge elements for carrying out a system development project. Sometimes there are obvious holes in a program. For instance the CS students do not have a course on User Interface; however, the CIS students learn user interfaces in their Visual Basic programming courses. The CS students can therefore learn a few things about UI from the CIS student(s) in their team. Similarly the CIS students can learn about the usefulness of algorithms for improving their system performance. The MIS students did not learn enough about Object Orientation do get an opportunity to see how classes are used. They can also consult the CS/CIS students to resolve their questions during implementation. The MIS students serve as good reviewers of requirements specification produced by other teams. Similarly the technical weaknesses in the designs by MIS students can be critiqued by the CS/CIS students during the review process.

5.3 Soft-skill Development

Invariably, students in a team have to take on different roles during the project period such as project leader, programmer, tester, communication coordinator, documentation specialist, system analyst, system designer, and database designer. Each role comes with unique opportunities for practicing soft skills. Two such roles are the project leader and the communication coordinator. The project leader has to develop the plan, delegate work to other team members and ensure their commitment to the project and their assignment. The communication coordinator manages all of the communications (organizing meetings, preparation of agenda and minutes and ensuring quality of documents) with the client and within team members. Since these roles are rotated during the period, each student usually takes on at least three different roles during the course of the project.

5.4 Program Objectives

The CS/CIS program has seven objectives. This capstone course provides evidence for assessing the following five of the seven objectives:

1. Demonstrate an understanding of computer science fundamentals.
2. Demonstrate fundamental software engineering skills on a non-trivial project to the satisfaction of a client.
3. Be prepared to enter the workforce as an entry level applied computer scientist.
4. Demonstrate the ability to communicate effectively.
5. Demonstrate critical thinking skills.

Since the projects are evaluated by the advisory board members, the instrument used for assessing the student deliverables can be used for program assessment as well. The criteria used in this evaluation are:

1. Product functionality (does the system do what it is supposed to do)
2. Product interfaces (Clarity, Organization, Navigation)
3. Use of Software Engineering Process (A systematic approach)
4. Quality of documentation
5. Overall quality of the product

An example of the summary evaluation results for four projects assigned during the Spring 2009 semester are provided in Table 1. The first three projects were developed by CS/CIS students and the last one was carried out by MIS students. True collaborative leaning could have occurred had the three MIS students joined the CS/CIS teams, rather than working by themselves on separate projects. However, integration did take place as the teams shared knowledge and skills in the overall development process.

The five characteristics listed above were used to examine each of the projects by ten evaluators. The characteristics were evaluated on a five-point scale with "1" being the lowest and "5" being the highest.

A week before the actual project presentations, the students were asked to review the products of other teams. This was used as a formative assessment with the feedback used to make adjustments to the final products.

Table 1 - Project Evaluation Summary

Item	Proj-1	Proj-2	Proj-3	MIS
1	4.9	4.9	4.8	4.4
2	4.3	4.4	4.6	4.8
3	4.4	4.8	4.5	4.2
4	4.8	4.6	4.5	4.0
5	4.8	4.8	4.8	4.4

Overall, the evaluators' satisfaction with each of the projects was rather high. As was expected, the MIS students did a better job on user interfaces.

6. CONCLUSION

Experiential learning opportunities provide students with an opportunity to apply all of the knowledge and skills that they have gained over the course of their educational program into an integrative, real world experience. At the authors' institution, the capstone experience attempts to address learning objectives established by the University, program and course to provide students with a well rounded experience applying both technical and soft skills. In Spring 2009, when the MIS capstone course was cut due to low enrollments, what started off as a solution of convenience turned into integrative learning opportunity for both the MIS and CS/CIS students alike. Students had the added value of working with users and developers of multiple levels and skills to create an acceptable client solution.

The addition of the MIS team was not initially planned. As a result, not enough changes were made to the course to reflect the nature of the MIS program. Also, the MIS students were given a project from within the university instead of having an external client. However, we intend to continue with the combined Capstone Experience class for a few years. We will plan to assign projects sponsored by external client to MIS students as well. We also plan to introduce more rigorous review processes for the analysis and design specifications by peers in order to enhance learning from each other. Further, the concept of integrated teams will be explored in the future.

The final project document contains all aspects of the project. It includes the students' reflections, their specific contributions and a letter from the client regarding the acceptance of the final product. The students are allowed to borrow this document for showing it to their future employers when they go for employment interviews. Even though the instructor has to spend a significant amount of time in planning the projects, it is worth the trouble since real world projects add value to the overall quality of the students' educational programs.

7. SOURCES

Alexander, R. A. (2001) “Need to update your Information technology? Try service learning.” Nonprofit World, Vol. 19, No. 5, pp. 27-29.

Beachboard, J. C., and D. V. Beard. (2005) "Innovation in Information Systems Education-II Enterprise IS Management: A Capstone Course for Undergraduate IS Majors." Communications of AIS 2005, Vol. 15, No. 18, pp. 315-330.

Bullen, C. V., T. Abraham, K. Gallagher, J. C. Simon, and P. Zwieg (2009) “IT Workforce Trends: Implications for Curriculum and Hiring.” Communications of the Association for Information Systems, Vol. 24, No. 9, pp. 129-140.

Denton, J. W., and W. E. Spangler (2001) “Effectiveness of an Integrated Precapstone Project in Learning Information Systems Concepts.” Journal of Information Systems Education, Vol. 12, No. 149, pp. 149-156.

Goode, S. and S. T. Teh (2005) “Peer Review to Support Student Assessment in Teams.” Journal of Information Systems Education, Vol. 16, No. 4, pp. 379-383.

Guthrie, R. A. and C. J. Navarrete (2004) “Service-Learning Impact on IS Students in a Web Development Course.” Information Systems Education Journal, Vol. 2, No. 12, http://isedj.org/2/12/.

Lazar, J. (2003) “Improving Web Accessibility Through Service-Learning Partnerships.” Information Systems Education Journal, Vol. 1, No. 33, http://isedj.org/1/33/.

McGann, S. T. and M. A. Cahill (2005) “Pulling it all Together: An IS Capstone Course for the 21st Century Emphasizing Experiential & Conceptual Aspects, Soft Skills and Career Readiness.” Issues in Information Systems, Vol. 6, No. 1, pp. 391-397.

Saulnier, B. M. (2003) “Creating Significant Learning Experiences in Systems Analysis and Design: Towards a Service Learning Paradigm.” Information Systems Education Journal, Vol. 1, No. 4, http://isedj.org/1/4/.

Southeast Missouri State University (2005-2006) University Studies Handbook, 2005-2006. Southeast University Press, Vol. 128, No. 1, p. 63.

Wei, K., J. Siow and D. L. Burley, (2007) “Implementing Service-learning to the Information Systems and Technology Management Program: A study of an Undergraduate Capstone Course.” Journal of Information Systems Education, Vol. 18, No. 1, pp. 125-136.

Wilcox, E. and I. Zigurs (2003) “A Method for Enhancing the Success of Service-Learning Projects.” Information Systems Curricula. Information Systems Education Journal, Vol. 1, No. 17. http://isedj.org/1/17/.

APPENDIX –A: DETAILS OF DELIVERABLES

1. Inception phase: Requirements Specification.: Use Case Model (UCM) documentation. Contents of the Requirements Specification: (Get this document reviewed by the client)

1. Introduction (Scope of your project; brief description of the existing system or manual process)
2. Use Case Model (use case diagrams, actors, use cases, use case descriptions in the standard format)
3. Glossary
4. Supplementary Specification
5. Project Plan This plan should address (broadly) the remaining phases

The MIS students provide a context diagram and zero-level Data flow Diagram.

Process Description of the System (using context and DFDs) and Data Description of the System (using ERDs).

2. Elaboration phase: Iteration-1: Analysis Spec & draft Design spec.: Use Case Analysis (UCA), Architectural design, UI Designs. Contents of Analysis and Draft Design Spec.:

1 Introduction (Rationale for the use cases considered in the baseline design; attach the modified use case diagram as well, summary of non-functional requirements)

2 Use Case Analysis: Packages and main class diagram showing relationships, attributes and responsibilities / operations. (Include a list of identified classes along with a brief description for each analysis class; and interaction diagrams; revise / unify classes considering the various use cases; class-mechanism mapping (to cater for non-functional mapping)

3 System Design (chosen system architecture for the application; decisions concerning database, chosen language for development, use of frameworks or COTS – along with brief descriptions)

4 Preliminary user interface design (forms and reports)

The MIS students provide higher level Data Flow Diagrams in place of class diagrams.

3. Elaboration & Construction: Baseline System and Final Design spec.: Iteration-2: System Design. Contents of Baseline System and Final Design Spec.:

1 Revised system design (chosen software architecture; decisions concerning database, language, use of frameworks or COTS – along with brief descriptions).

2 Final user interface design (forms, displays and reports).

3 Database design (normalized and optimized ERD- schema)

4 Class (object) design (refine attributes, operations and apply inheritance; where applicable; provide state diagrams for the dynamic objects)

5 Binary (soft copy of code) for a working baseline system, implementing the selected use cases (code incorporating system to provide functionalities, user interfaces and database)

6 Brief user notes (for the chosen use cases; these will be later used in the user manual).

The MIS students provide structure chart in place of analysis class diagrams.

4. Software product with all documentation. Contents of final project report:

1 Cover page (project title; authors) and Contents page (include those on softcopy)

2 Project Overview (Project sponsor and project team, Project scope, Summary of requirements along with the main use case diagram, Project plan – Gantt chart on a single page, System acceptance letter from Client) – about 6 pages in all

3 System Documentation. (Compilation of the following: use case descriptions, interaction diagrams, final packages with main class diagrams, table of analysis and design mechanisms, application system architecture (in terms of subsystems/packages), class design, user interface designs, database designs, state diagrams – if any, state important design and implementation decisions You may put the diagrams in appendices and make reference to them in the main body. About 5 pages narration plus diagrams and use case descriptions.

4 Application Test (test plan, test cases and test results – where possible put the details in appendix) – about 2 pages plus details on appendix.

5 User Documentation (user guide by actor / facility)

6 Conclusion (what was achieved, Major problems encountered and solutions, reflections on what you have learnt from the project – team as a whole; breakdown of effort - one page per person - indicating the time spent by each on the project activities; what could be done further) – about 5 pages excluding the individual log.

7 Provide a softcopy of all the model diagrams, user interfaces, user notes and code.

The MIS students provide Context and Data Flow Diagrams and also Structure Charts.

Appendix –B: Evaluation Instruments

1. Use Case Presentation (Definitely disagree to definitely agree):

The presentation helped me understand the functional requirements very clearly	1	2	3	4	5
The presentation helped understand the scope of the system very precisely.	1	2	3	4	5
The use case diagram is professionally done and is up to acceptable standard*	1	2	3	4	5
The actors and use cases make a lot sense and match with the project aim*	1	2	3	4	5
From the presentation, it is obvious the team is on the right track	1	2	3	4	5
* In some cases other diagrams may be used as needed					

2. Use case Model

The presentation helped me understand the scope and size of the project very clearly	1	2	3	4	5
The presentation helped me understand the flows in the two use cases very precisely	1	2	3	4	5
The presentation helped me understand the non-functional requirements very clearly.	1	2	3	4	5
The presentation helped me learn the important key words (terms) used in the project's domain area.	1	2	3	4	5

3. User Interface Presentation

The presentation helped me understand the high-level architecture of the system very clearly	1	2	3	4	5
The presentation helped me understand the major input forms, displays, and the system interfaces clearly.	1	2	3	4	5
The interfaces are done professionally and are up to acceptable standard	1	2	3	4	5
The presentation addressed the form design / interface design clearly	1	2	3	4	5
On the whole I like the user / system interfaces and they are appropriate (can be used as they are)	1	2	3	4	5

4. Research Presentation:

A. Content (Criteria) Use applicable factors listed below (Maximum 25 points)

1. Scope of research stated clearly
2. Relevance to System Development established
3. Demonstrated inquiry / research skill
4. Stated the underlying principles
5. Had logically organized the material
6. Helped in enhancing your learning
7. Completeness (achieved the set objectives)
8. Provided references
9. Gave practical tips

B. Presentation: Consider the following quality issues (Maximum 15 Points)

1. Quality (to suit purpose) of the presentation
2. Organization of the presentation
3. Cogency
4. Clarity
5. Focus
6. Responsiveness to the audience

Appendix –C: Topics and Assessment Schedule

Week	Topics and Assessments
1	Course overview, Review methodologies
2	Testing [Use Case and Project Plan presentation]
3	Project Management; [Exam –1]
4	Communications Management [UCM presentation]
5	Quality & Configuration Management [Requirements Spec Due]
6	Work on Project. [Exam –2]
7	Work on Project
8	Work on Project [UI presentation & Analysis Spec]
9	Research Presentations
10	Work on Project [Baseline system & final design]
11	Work on Project; Weekly review
12	Work on Project; Weekly review
13	Work on Project; Weekly review
14	[Project Presentation & demo of final system to client & faculty]
15	[Project demo of final system to public]
16	[Project documentation along with system binary].