

INFORMATION SYSTEMS EDUCATION JOURNAL

In this issue:

- 4. Where do Student Outcomes Begin? Developing Professional and Personal Management Skills as a Strategy for Student Success in the First Computing Course and Beyond**
Sean Humpherys, West Texas A&M University
Jeffrey Babb, West Texas A&M University
Amjad Abdullat, West Texas A&M University
- 27. The Information Systems Core: A Study from the Perspective of IS Core Curricula in the U.S.**
Drew Hwang, Cal Poly Pomona
Zhongming Ma, Cal Poly Pomona
Ming Wang, California State University, Los Angeles
- 35. A Project Management Approach to Applying Best Practices to Online CS/MIS Experiential Learning Projects**
Dana Schwieger, Southeast Missouri State University
Ken Surendran, Southeast Missouri State University
- 43. Assessing Cyberbullying in Higher Education**
Ali Kamali, Missouri Western University
- 54. A Match in the Making: How Emergent Changes in the Marketing Discipline Present Opportunities for Information Systems Programs**
Maureen Carley, Virginia Commonwealth University
Jeffrey Babb, West Texas A&M University
- 68. Adding Value: Online Student Engagement**
Donna R. Everett, Morehead State University
- 77. A Systems Analysis and Design Case Study for a Business Modeling Learning Experience for a Capstone CIS/IS Systems Development Class**
Jack Russell, Northwestern State University
Barbara Russell, Northwestern State University
- 97. A Model for Establishing a Cybersecurity Center of Excellence**
Edward, J. Moskal, Saint Peter's University
- 109. Course Redesign Based On the Quality Matters Program: Examples of Before and After**
Jennifer Kreie, New Mexico State University
Susan Bussmann, New Mexico State University

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published by **EDSIG**, the Education Special Interest Group of AITP, the Association of Information Technology Professionals (Chicago, Illinois). Publishing frequency is six times per year. The first year of publication is 2003.

ISEDJ is published online (<http://isedj.org>). Our sister publication, the Proceedings of EDSIG (<http://www.edsigcon.org>) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 40%.

Information Systems Education Journal is pleased to be listed in the 1st Edition of Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org.

2015 AITP Education Special Interest Group (EDSIG) Board of Directors

Scott Hunsinger
Appalachian State Univ
President

Jeffry Babb
West Texas A&M
Vice President

Wendy Ceccucci
Quinnipiac University
President – 2013-2014

Eric Breimer
Siena College
Director

Nita Brooks
Middle Tennessee State Univ
Director

Tom Janicki
U North Carolina Wilmington
Director

Muhammed Miah
Southern Univ New Orleans
Director

James Pomykalski
Susquehanna University
Director

Anthony Serapiglia
St. Vincent College
Director

Leslie J. Waguespack Jr
Bentley University
Director

Peter Wu
Robert Morris University
Director

Lee Freeman
Univ. of Michigan - Dearborn
JISE Editor

Copyright © 2015 by the Education Special Interest Group (EDSIG) of the Association of Information Technology Professionals (AITP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to Nita Brooks, Editor, editor@isedj.org.

INFORMATION SYSTEMS EDUCATION JOURNAL

Editors

Nita Brooks
Senior Editor
Middle Tennessee State Univ

Thomas Janicki
Publisher
U of North Carolina Wilmington

Donald Colton
Emeritus Editor
Brigham Young University Hawaii

Jeffry Babb
Associate Editor
West Texas A&M University

Wendy Ceccucci
Associate Editor
Quinnipiac University

Melinda Korzaan
Associate Editor
Middle Tennessee State Univ

Guido Lang
Associate Editor
Quinnipiac University

George Nezek
Associate Editor
Univ of Wisconsin - Milwaukee

Samuel Sambasivam
Associate Editor
Azusa Pacific University

Anthony Serapiglia
Teaching Cases Co-Editor
St. Vincent College

Cameron Lawrence
Teaching Cases Co-Editor
The University of Montana

ISEDJ Editorial Board

Samuel Abraham
Siena Heights University

Mark Jones
Lock Haven University

Alan Peslak
Penn State University

Teko Jan Bekkering
Northeastern State University

James Lawler
Pace University

Doncho Petkov
Eastern Connecticut State Univ

Ulku Clark
U of North Carolina Wilmington

Paul Leidig
Grand Valley State University

James Pomykalski
Susquehanna University

Jamie Cotler
Siena College

Michelle Louch
Duquesne University

Franklyn Prescod
Ryerson University

Jeffrey Cummings
U of North Carolina Wilmington

Cynthia Martincic
Saint Vincent College

Bruce Saulnier
Quinnipiac University

Christopher Davis
U of South Florida St Petersburg

Fortune Mhlanga
Lipscomb University

Li-Jen Shannon
Sam Houston State University

Gerald DeHondt

Muhammed Miah
Southern Univ at New Orleans

Karthikeyan Umapathy
University of North Florida

Audrey Griffin
Chowan University

Edward Moskal
Saint Peter's University

Leslie Waguespack
Bentley University

Janet Helwig
Dominican University

Monica Parzinger
St. Mary's University

Bruce White
Quinnipiac University

Scott Hunsinger
Appalachian State University

Peter Y. Wu
Robert Morris University

A Systems Analysis and Design Case Study for a Business Modeling Learning Experience for a Capstone CIS/IS Systems Development Class

Jack Russell
jrussell@nsula.edu

Barbara Russell
brussell@nsula.edu

Computer Information Systems
Northwestern State University
Natchitoches, LA 71497, USA

Abstract

The goal is to provide a robust and challenging problem statement for a capstone, advanced systems analysis and design course for CIS/MIS/CS majors. In addition to the problem narrative, a representative solution for much of the business modeling deliverables is presented using the UML paradigm. A structured analysis deliverable will be the topic of a second paper on this subject. The authors teach the systems analysis and design course(s) or the systems development course(s) at their university. The CIS senior capstone course that the primary author teaches requires that the student complete one or more business modeling case studies. The authors have used some cases/problem statements from various systems analysis and design texts; however, the authors wanted a problem statement that would challenge a systems team at the senior undergraduate level to produce a complete static, functional and behavioral business model that could be designed and implemented. The authors believe that the narrative should include enough detail to enable the team to use either Unified Modeling Language (UML) or Structured Analysis.

Keywords: static model, structural model, functional model, behavioral model, project, tasks, teams, and team members.

1. INTRODUCTION

The goal of this paper is to develop a case study (i.e., problem statement) and a Unified Modeling Language (UML) solution for an automated system that will 1) create and track projects, 2) assign teams to projects, 3) assign tasks to teams, 4) assign the most qualified team members to the teams, and 5) address project costing. This business system for Premier Consulting, Inc. will be titled "Automated Project Management and Scheduling." The scenario of

this case study begins with the Chief Information Officer (CIO), which might be played by the instructor of the course, who approves and initiates a project and instructs the Project Director to create teams. This new case study is destined to be used in the Advanced Systems Analysis and Development capstone class. Either a multiple team approach or single team approach will be used in developing a solution depending upon the number of student in the class. Different teaming scenarios are presented in the conclusion to this paper.

Developing a collaborative learning environment is critical to the success of any IT/IS curriculum (Davis, Feinstein, Gorgone, Longenecker & Valacich, 2002). Also, an excellent study discusses the collaborative teaming approach in detail (Ewusi_Mensah, Seal, & Abraham, 2003). Developing a quality case study for this important capstone class drove the authors to develop the case study that follows.

2. THE CASE STUDY

Background

The new automated system is destined to replace the current, manual, error-prone process. The automation of this activity has been welcomed by management for quite some time with the anticipation of significantly reducing the company's expenses. It is hoped that this reduction will be accomplished by 1) creating more successful teams working with greater precision, 2) completing projects on time and within budget due to a better understanding of system requirements and tasks to be completed, and 3) starting projects on time as a result of the automated project scheduling system.

Systems Requirements Statement

- The Project Director creates a project and a "project profile" for each project. The creation of the project profile will require the determination of project employee costs, the assignment of tasks to the project, and the assignment of a project manager. Examples of various tasks are: 1) "prepare cost-benefit analysis report", 2) "produce entity relationship diagram", 3) "produce class diagram", etc.). Once created, the project profile will consist of project id, project personnel cost, a list of tasks assigned, and the project manager.
- The Project Director also creates the teams for a given project, assigns employees to the teams, and assigns a team leader.
- The Project Manager is responsible for assigning tasks to the various teams working on the projects(s).
- The Team Leader assigns tasks to the team members. For all intents and purposes, the various team leaders in the class will mimic this because they are also responsible for assigning tasks to their team members working on this case (automating these activities).

Figure 1 (in the Appendix) illustrates this hierarchy.

Additional functionality includes:

- Retrieve and update information about various software projects.
- Retrieve and update information about various project teams.
- Provide information about specific team member assignments.
- Provide information that allows for better assignment of team members to specific teams.
- Provide information about the location of projects, teams assigned, and members assigned to the various teams, and what the team member skill sets are.
- Perform function point analysis computations to determine the personnel cost of the project to be created. This activity would be part of the process of creating the project.
- Provide project costing information to Accounting so that the client can be invoiced upon completion of project phases. The costing information will be determined from the function point analysis study alluded to above.
- Retrieve projects and indicate which tasks have been completed and what tasks that are currently being worked on by each project. This is expanded upon later.

The Static/Structural Model

Premier Consulting Incorporated is a global organization. A branch office can be either a foreign or domestic office and has a branch manager. Both the foreign and domestic branch offices will oversee their respective regional offices and each regional office oversees district offices. Each regional office will have a manager. Each district office will have a manager. A district office may sometimes oversee departments but not always. The departments may include areas such as recruiting, training, research, and development. Each department will have a manager. The automated system needs to be able to report on information for the branches, regions, and districts. This information would include data about their branch, region and district; identification number, name, address, phone, and manager name.

A department may have one or more employees. An employee may be assigned to

one or more teams; a team may consist of several employees. However, it is possible for a team to consist of a single employee. A team will never have zero employees; thus requiring that at least one team member be assigned upon team creation. An employee can be classified as either regular staff or a consultant. Regular staff will draw a straight monthly check. A consultant is paid by the hour and receives time and half pay for hours over 40. Regular staff receives health benefits and pay social security tax (FICA). Consultants do not receive health benefits nor do they pay FICA.

Premier Consulting Incorporated seeks to improve the quality of the productive work performed by teams; therefore, quality assurance (QA) teams are used to assess the performance of teams on projects completed. The QA teams may consist of one or more regular staff and one or more consultants. Saying it another way, some regular staff and some of the consultants are parts of the QA Team. An exception is that the consultant or staff chosen for the QA Team cannot be on one of the project teams being studied. In other words, he or she cannot be studying or evaluating him/herself.

A project may have zero teams assigned to it initially. The project manager may choose to assign teams to the project at a later date, or choose to assign a team or teams to the project upon project creation. But, the Project Director creates the project profile for the project. A team may be assigned to one or several projects at a time. Typically, a project may involve a multiplicity of teams working at various stages of the project. In other words, the data analysis team could be working on the data-modeling phase of the project while the process analysis team is completing the activity-modeling phase of the project.

Figure 2 (in the Appendix) describes the project, the team, the team member(s), and task relationship. An example of a project is defined here as a project to create an order entry process, with teams assigned, and employees assigned to the teams. A project will consist of one or more tasks, and a task may be assigned to one or more projects. Figure 3 below illustrates this relationship.

The process of assigning tasks to a project is to assist in the creation of a project profile. The project profile data is needed in order to create

the teams and to assign the appropriate team members with the correct skill sets to the aforementioned teams. A specific team will be assigned a specific task or tasks, and a task may be assigned to one or more teams. A task is often categorized as programming, systems analysis, systems design, database design, data mining, e-commerce / web-enablement, telecommunications, or hardware design and will include a task difficulty coefficient/rating (1 to 10 – where 10 is the most difficult). Examples of more detailed tasks may include “prepare cost benefit analysis report,” “draw ERD,” “prepare class diagram”, “draw activity diagram for Accept Orders Process”, “code client/server web-enabled order system”, etc. To summarize, the project is created with a project profile and a project manager. The project profile consists of the project costs and assigned tasks. Projects will be assigned teams. The assigned teams will, in turn, be assigned tasks from the task list associated with the project. Figure 4 (in the Appendix) illustrates this association.

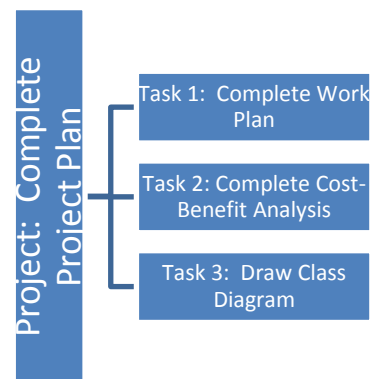


Figure 3: Relationship between Projects and Tasks Assigned.

The company needs to be able to identify what kind of task or tasks a specific team is working on at a specific point in time. In some cases, company management needs to be able to assign a team based on team category (systems analysis, programming, database design, etc.) to a specific task within a project phase and to make sure that the team’s expertise rating is high enough to complete a given task’s difficulty rating (determined by its difficulty coefficient). In other words, the management wants to make sure talent is matched with responsibility (assigning the most experienced and talented people to the specific task).

A team is limited to seven (7) members, and the team will be assigned specific tasks. The team leader will be responsible for assigning these tasks to specific team members that have been assigned to the team. Once all tasks are completed for that team, the team is dissolved or assigned to another project.

A team will complete a given project, and once the team completes a project, the team is either dissolved or is assigned to another project. A team member is often assigned to another team at that point, but it is possible for a consultant to not be immediately assigned to a team. During this period, the consultant is said to be "on the beach." A staff member cannot be "on the beach." Staff will have regular duties to work on while waiting on another team assignment. The consultant will often attend workshops to sharpen his or her skills in a particular area during this waiting period. The system needs to have the functionality to remove a team from a project and reassign the team to a different project or delete the team altogether. Also, a given task can be reassigned to a different team when management feels that it would promote project completion.

A project will often consist of project phase categories. General information about a phase is discussed below. Project phase categories may include categories such as planning, analysis, design, implementation, testing, and production. It may be important to keep table data about these phases since all projects will fit into one of these categories, and a user may want to print information about the phase such as phase number, phase title, and phase description/comments. A specific phase category may be included on many different projects; and, on the other hand, a project may involve many different project phase categories. A project phase will have a start date and stop date and will include information about the title of the phase (Examples: "Requirements Analysis" or "Object Modeling.")

As previously mentioned, the branch office can be categorized as either domestic or foreign. A domestic branch will also include information about the state code, state subdivision (county/parish), and so forth. A foreign branch will also include information about country code, foreign country excise tax, and so forth.

Functional Model

How the Over-All System Interacts with External Entities/Actors. A narrative follows:

"John Reynolds, Project Manager, who works for the Personnel Department and is responsible for providing the right information to the Project Scheduling system. First, John will send a folder of available employees to the Project Scheduling System. Mary Raeger, who also works for Personnel, will provide the system with an official and approved request to create a project. Of course, the client would have sent a system request to the company that would have to be approved first. The business system will produce various reports that are sent to management. One report that is sent to management is a roster of various projects along with the assigned teams and assigned employees. A job report that tracks the number of hours an assigned employee (by team) worked on any given project is sent to accounting for costing purposes. This job report will also be sent to personnel so that a history log can be prepared by the personnel department and sent to the employee on a quarterly basis. An integral part of the system is the actual assignment of the teams to the projects and the assignment of employees to the teams. This is an intricate operation that will calculate project complexity estimates, and provide a project complexity report to the information systems department for review. The validated project complexity estimates will eventually be used as input by the system to actually calculate function points. These estimates will include details about the number and complexity of inputs, outputs, database tables, interfaces, and queries. The system will perform a number of complex correlations and calculations and produce a project assignment report that lists the project, the teams assigned, and assigned employees/consultants for the specific team (s). This report is sent to management and to Information Systems to be distributed to the appropriate project managers."

Major Processes

The system consists of five (5) major processes. The project director must 1) be able to create a project along with its profile, 2) The project director also creates teams for the various projects along with its team leader, 3) The project manager will assign tasks for each team, 4) the team leader will assign specific tasks to a team member, and 5) the system will also

generate special management reports to the CIO.

As previously discussed, the Project Director is responsible for creating the projects, assigning tasks to projects, assigning a project manager, creating and assigning teams to projects, assigning members to teams, and assigning team leaders for each team. One of the first jobs for the Project Manager is to assign tasks to the teams. The Team Leader is responsible for handling the assignment of tasks to individual members on the team and overseeing their completion. While this has been illustrated previously, it's worth mentioning again to highlight the specific processes as they relate to the Project Director, Project Manager, and Team Leader.

As previously alluded to, but with more detail, the process of creating projects will unconditionally require the process of determining a project profile. This process will involve determining the personnel cost of the project and the tasks needed to complete the project.

The process of creating a team will always necessitate the assignment of at least one team member. Assigning team member(s) is a separate process, but will always execute alongside the process of creating the team. In other words, it unconditionally executes.

Behavioral Model

The behavioral aspect of the scheduling system will work as follows: the project director logs on to the Scheduling System with a user id and password. The system will check the authenticity of this information. If correct, the system will present the director with a window/web page. The window/webpage will prompt the project director with a main menu. This main menu should enable the project director to 1) create a new project along with its project profile, 2) create a new team for the project, 3) assign team(s) to a project, and (4) assign members to a team, and 5) Generate Reports.

The project manager logs on the Scheduling System with user id and password.

The system will check authenticity of this information. The manager will be prompted with a menu that will allow him or her to assign tasks to a team. The team leader logs on to the

Scheduling System with user id and password. The system will check authenticity of this information. The team leader will assign team tasks to specific team members with associated due dates.

The system should provide an interface to enable the assignment of specific tasks to be assigned to that project. In addition, there should be an option to list all the projects so that a manager can observe sub-form information listing existing teams assigned to the project along with the team members for each team assigned.

Information Steps and Flow

The automated system should enable the Project Director to create a new project by entering a project ID, project description, start date, stop date, and Project Manager. Next the Project Director should be provided a list box to select an existing team to be assigned to the project or a textbox prompt to create a new team for this project. This would include Team ID and Team Description. Subsequently, after creating a new team, the system will display this team information with a prompt to select available employees from a list box. As the Project Director selects an available member, the system will check the member profile for matching skills and availability. If this selection is satisfactory then the member will be placed into a "members selected" list box (essentially a shopping cart.) This process will repeat until the manager is finished selecting potential team members.

This scenario is intentionally left incomplete since a system prototype is required. Some details are left to your discretion and creativity!

Creating the Project

In creating a project, as mentioned above, the system must first determine the "project profile." Next, if the project profile successfully completes, then the system will record the project, and then display the project information to the Project Director.

Next, the system will prompt the Project Director (previously alluded to) with a list of the various project phases for the Project Director to analyze. Then the Project Director will repeatedly select needed phases from the list he or she wishes to be associated with this project.

The system will then take the selected phase as input, and place the phase into a phases assignment list (basically a shopping cart). This will repeat until the Project Director is finished selecting phases for the project. When the system detects that there are no more phases to be selected, the system will list the project name, project duration, and the list of phases required for that project. The system prompts the user that his activity is complete and will then stop. Figure 5 below helps conceptualize this process.

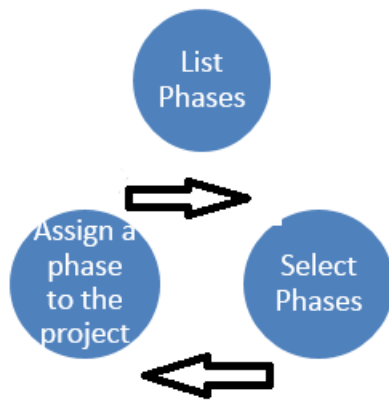


Figure 5: The Activity of Assigning Phases to a Project

The process of determining a project profile requires that the system prompt the Project Director to enter details related to the number and assigned weight of inputs, outputs, tables/files, interfaces, and Queries/Reports, and project complexity points (i.e., how complex is this project relative to other projects?). Once this data has been entered, the automated system will compute a Total Adjusted Function Point (TAFP) amount. If the function points are less than 400 then the system will determine the lines of code by multiplying the TAFP by the lines of code per function point for a particular language or other development framework (located in an array). If the function points are greater than or equal to 400 then the automated system prompts the CIO to complete a request to the Vice President of Finance for additional funding for the project (the project is too big). This may entail the CIO break this project into two separate projects if the VP of Finance grants the funding request. The chart below (Figure 6) illustrates the corresponding lines of code for a particular language.

Using the total lines of code as input, the system will determine the person-months (PM) needed for this project ($1.4 \times (\text{Lines}/1000)$). The system will then determine the MONTHS need for the project ($3.0 \times \text{PM}$)^{1/3}. The system will then determine the number of PERSONS to assign to the team (MONTHS divided by PM). The system will then compute the personnel COST of the system by multiplying PERSONS by \$250 per hour. This does not include materials, computers, servers, and network costs, and the determination of these costs are beyond the scope of this project and not be addressed by the system model. The project director and CIO will manually determine these costs.

| Language | Lines of Code / FP |
|-----------------|--------------------|
| C# | 65 |
| JAVA | 75 |
| Visual Basic | 50 |
| COBOL | 70 |
| HTML/CSS/JS/PHP | 70 |

Figure 6: Corresponding Lines of Code Per Function Point for a Designated Language

Assigning Tasks to the Project

Next, the system will repeat a similar dialog with the project director to accept tasks to be performed by the project. This requires the system to display a tasks list, and prompt the project director to select the tasks repeatedly until all tasks have been selected.

Creating the Team

The process of creating teams could involve similar dialog between the Project Director and the system as discussed above (dialog between the Project Director and creating the project). A difference here is that when a team is created it will be required to assign a member to the team. At least one member must be assigned. At least one task must be assigned to the team, but is likely that several tasks will be assigned to the team.

Assigning Tasks to Members

The team leader handles this activity. This process involves a similar scenario as the activity of assigning members to teams, except in this case, the team leader will start by selecting a team member chosen for his/her team; and, subsequently, will choose tasks from a list box and assign those selected tasks to the

selected team member. This process would be repeated until all team members have been assigned the tasks for which they are responsible. This will be communicated to the project manager so that the project manager can assemble a complete Gantt chart and PERT chart for the business system project.

You are encouraged to use your own creativity in augmenting any additional functionality as long as you specify your assumptions.

Deliverables

The student teams will submit:

- A UML functional model and structural model (use case diagram, use case descriptions, activity diagrams for each use case, and a class diagram).
- A UML behavioral model that includes a sequence diagram for each use case and a state transition/machine diagram for at least one complex class for the business rules provided.
- A system design specification that includes an interface/navigation design, database design, and program design.
- The business model must provide sufficient detail to support a systems design effort.
- A prototype that represents the functionality for 1) creating projects, 2) determining the project profile, 3) assigning tasks to the various projects, 4) creating teams, 5) assigning tasks to teams, and 6) assigning tasks to individual, specific members on the team.
- A program module that executes successfully for at least two of the use cases.
- Assumptions may be documented and included with the work.

The UML Solution

The UML solution consists of a number of UML diagrams covering the functional, structural and behavioral models. The authors employed two main textbooks on the subject (Kendall & Kendall, 2012) and (Dennis, Wixom, & Tegarden, 2012). The solution consists of:

1. Normalized class diagram (Figure 7, in the Appendix)
2. Use case diagram (Figure 8, in the Appendix)
3. The use case description for assigning tasks to the team member (Figure 9, in the Appendix)
4. Activity diagrams for:

- a. creating a project (Figure 10, in the Appendix)
- b. activity diagram for creating the project profile (Figure 11, in the Appendix)
- c. activity diagram for creating a team (Figure 12, in the Appendix)
- d. activity diagram for assigning tasks to a team member (Figure 13, in the Appendix)

5. Sequence diagram for creating a project (Figure 14, in the Appendix) and a sequence diagram for each of the other use cases (Figure 8, in the Appendix).

A part of the goal of this paper is to encourage instructors of UML to review and provide the authors with either corrections or suggestions on how to improve the case study and solution. Furthermore, the authors are interested in knowing how others would assign this to classes.

3. RECOMMENDATIONS

Two elements should be considered in a capstone class using this case study: 1) Team formation and 2) Team Deliverables.

Student Team Formation

A discussion of how the instructors form the student teams (Russell, Russell, & Tastle, 2005) follows:

- 1) The instructor creates the teams.
- 2) Students from the class submit their anonymous resumes.
- 3) The students in the class are required to read other student resumes.
- 4) From having read the student resumes, students will nominate team leaders.
- 5) From the nominations, the class will choose team leaders.
- 6) From the pool of anonymous student resumes that remain, the team leaders choose their team members in a "round robin" fashion. This is repeated in until all students are assigned to a team.

While there are real flaws with the approach, at least this approach enables the seniors to grasp the difficulty of choosing people for a team when little is known about them other than what is observed from a resume. Soon the seniors will graduate and become IT professionals having to read resumes and decide who to interview for either an IT position with the company or to

decide if they have the skills for a specific project team.

Team Formation Scenarios

One scenario we are considering is to create a Functional Modeling Team, Structural Modeling Team, Behavioral Modeling Team, and a Software Development Team. Students are assigned to the teams based on their backgrounds ascertained from their resumes. This approach is being considered since it mimics the case on which they are working. This would add an element of realism, but would only work with a small class of approximately 16 to 20 students (i.e., 4 or 5 per team). With larger class sizes the second scenario below might be preferred over this first scenario presented.

A second scenario is to require each team to deliver the entire system. We have used this approach by simply allowing fewer deliverables from each phase. This scenario works better with larger class sizes. The negative aspect is that there is a tremendous duplicity of effort by the teams with the chance of plagiarism between the teams.

If the first scenario is used then selected members from each team will participate in the Software Development Team to implement at least two non-trivial applications. In the second scenario, each team will designate one or two team members to develop the code.

Semester Deliverables and Completion Time Line

Figure 15 (in the Appendix) illustrates the project phase, deliverables and a time line to complete the deliverables. The deliverables are common for a systems project (Dennis, et al.)

Russell, Russell, and Tastle discuss the content of the capstone class in their paper (Russell et al., 2005). Students present the system proposal in Week 8 in front of the class with required PowerPoint slides along with a hard copy that is submitted to the instructor at that time. In Week 16, the system specification is presented in front of the class with a complete PowerPoint presentation. The system design specification presentation involves also demonstrating program functionality using a software development platform.

The student evaluation (Russell et al., 2005) requires a peer-evaluation from each student member on his/her team that is worth 10% of

the student's final grade. Team leaders are enticed into becoming a team leader by enabling the team leader to earn 5 bonus points for the semester. A team leader evaluation (Likert Scale) of 5 adds 5 points. A grade of 4 adds 3 points. A grade of 3 adds zero points. A grade of 2 deducts 3 points. A grade of 1 deducts 5 points. This is a sliding scale; therefore, a student must think twice before they choose to be a poor team leader as it can cost him or her ½ letter grade for the semester.

Future Research

The authors plan to use the case study in the capstone class this spring 2015 semester and collect both attitudinal and cognitive data from the students taking the class. We are looking to see how well this new case study is perceived by our seniors, and what impact the case study has on the learning process in our classroom. Variables such as examination score performance and peer evaluation scores will be analyzed and compared to previous semesters. We want to determine the difficulty of the case study compared to a few cases that we have used in the past. Previous semester grades and spring 2015 grades will be collected in this regard. We want to find out if the case study improved specific skill performance in producing various UML diagrams. We are interested in measuring performance differences on exam questions (variables): 1) Drawing Use Case Diagram, 2) Drawing Activity Diagram, 3) Drawing Class Diagram. The overall goal is to continuously improve the IS capstone experience at our university.

4. REFERENCES

- Davis, Feinstein, Gorgone, Longenecker and Valacich, (2002), *IS 2002 Information Systems Model Curriculum*, College of Business and Economics, Washington State University, Pullman, WA.
- Dennis, A. Wixom B., Tegarden D., (2012), *Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach*, 4th edition; Chapters 4, 5, and 6 (pages 153 - 247).
- Ewusi_Mensah, K, K C Seal, and DM Abraham, (2003), "Developing a Collaborative Learning Facility to Support Advanced Information Systems Courses", *ISECON 2002 Proceedings*, v 20 (San Diego), ISSN: 1542-

7382. Also appears in the *ISEDJ* Journal 2:
(13), ISSN: 1545-679X.

Kendall, K. & Kendall, J., (2006), *Systems
Analysis and Design*, 8th edition, Chapter 10
(pages 281 - 327).

Russell, J., Russell, B., and Tastle, W., (2004),
"Teaching Soft Skills in a Systems
Development Capstone Class", *Proceedings
of ISECON 2004*. Also appears in the *ISEFJ*
Journal 3: (19), August 2005.

APPENDIX

Figure 1: Hierarchy of Project Management and their Functions

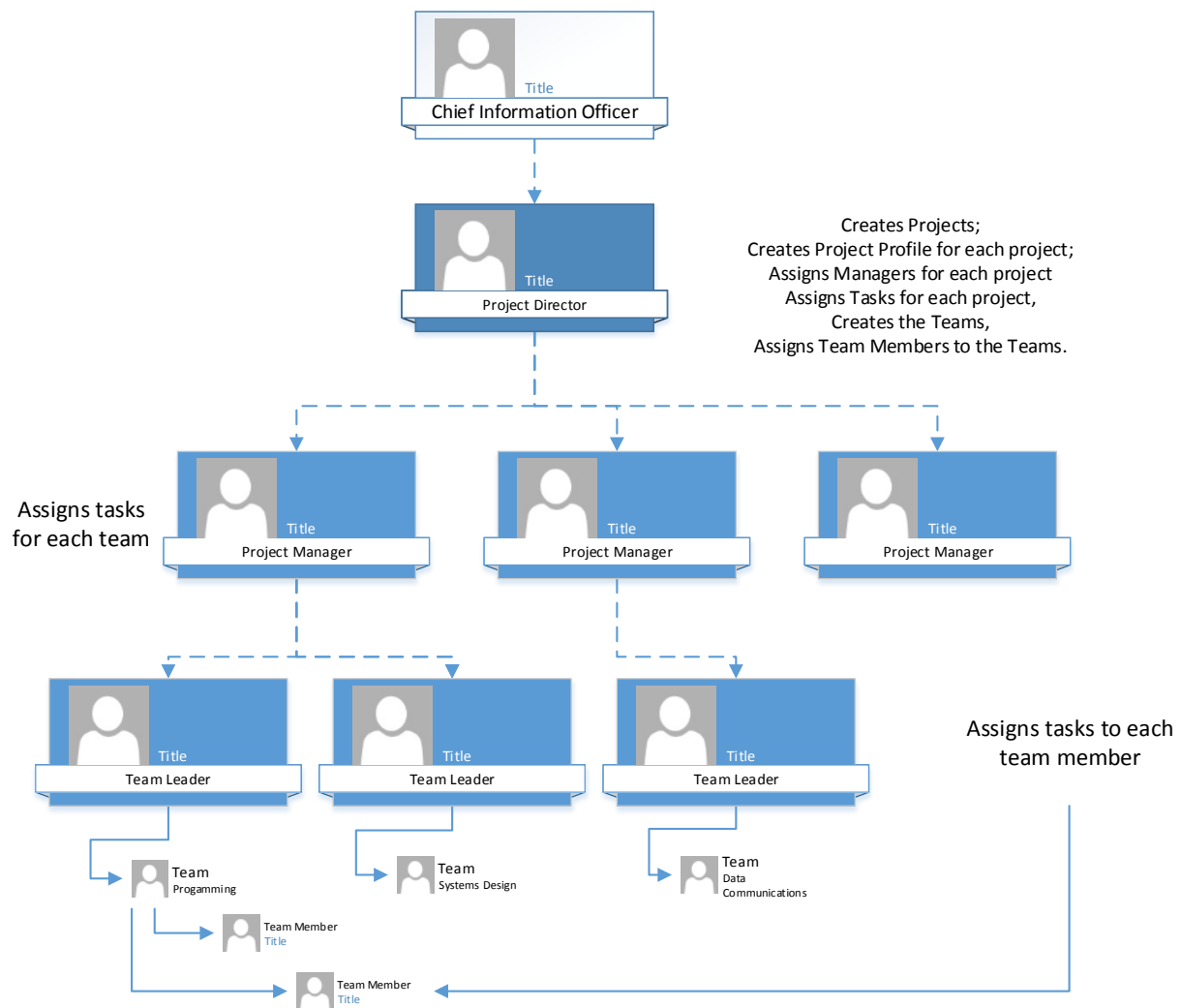


Figure 2: The Relationship Between Projects, Teams, and Team Members

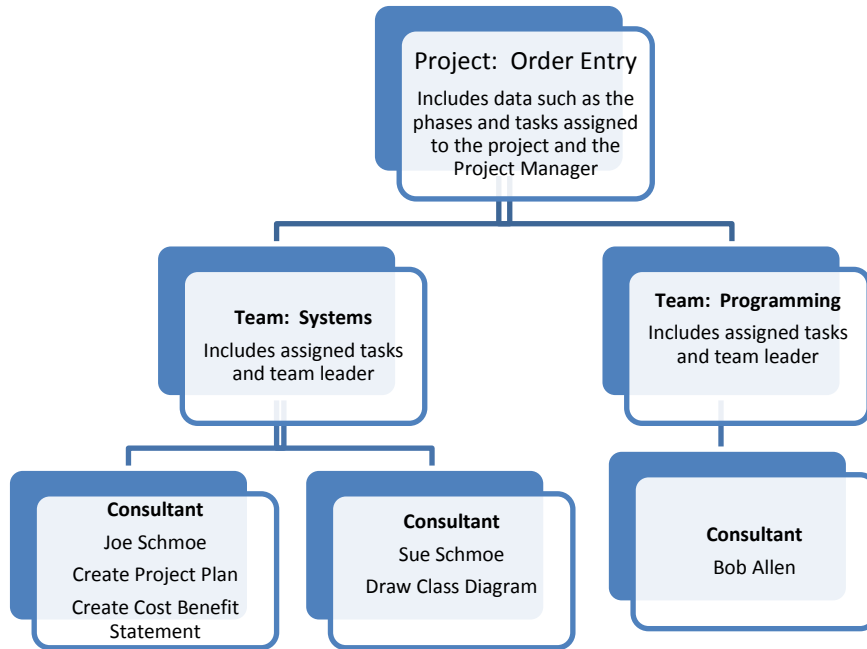


Figure 4: Project Relationship: Project Profile and Teams Assigned

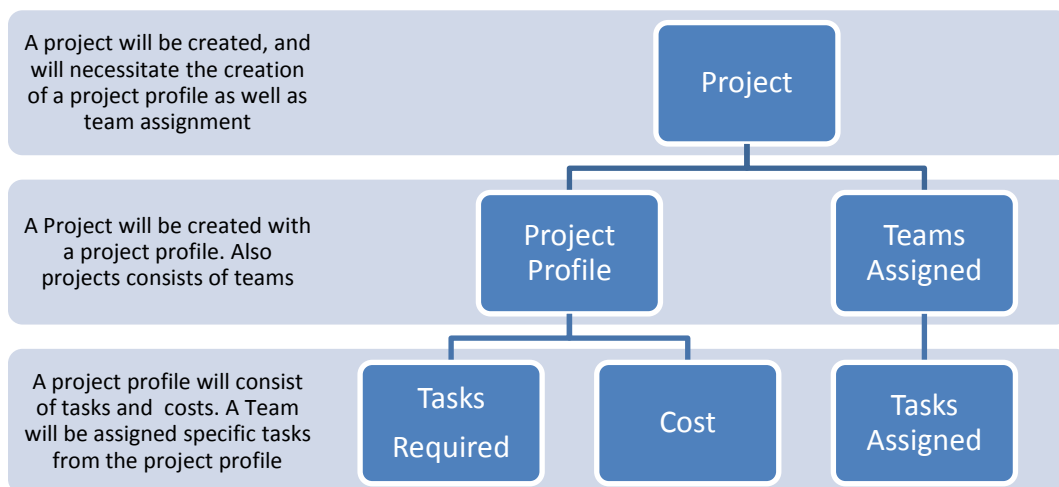


Figure 7: Class Diagram for Premier Consulting

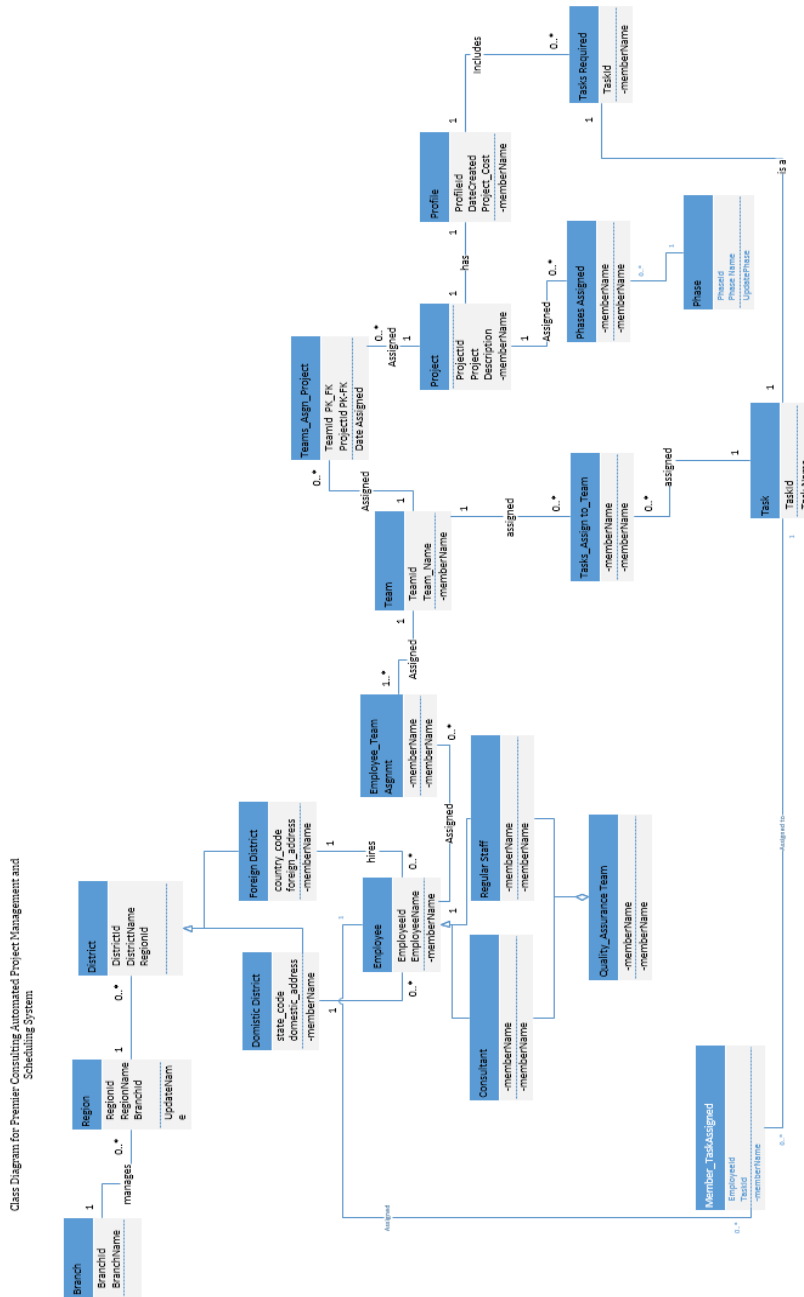


Figure 8: Use Case Diagram for Premier Consulting

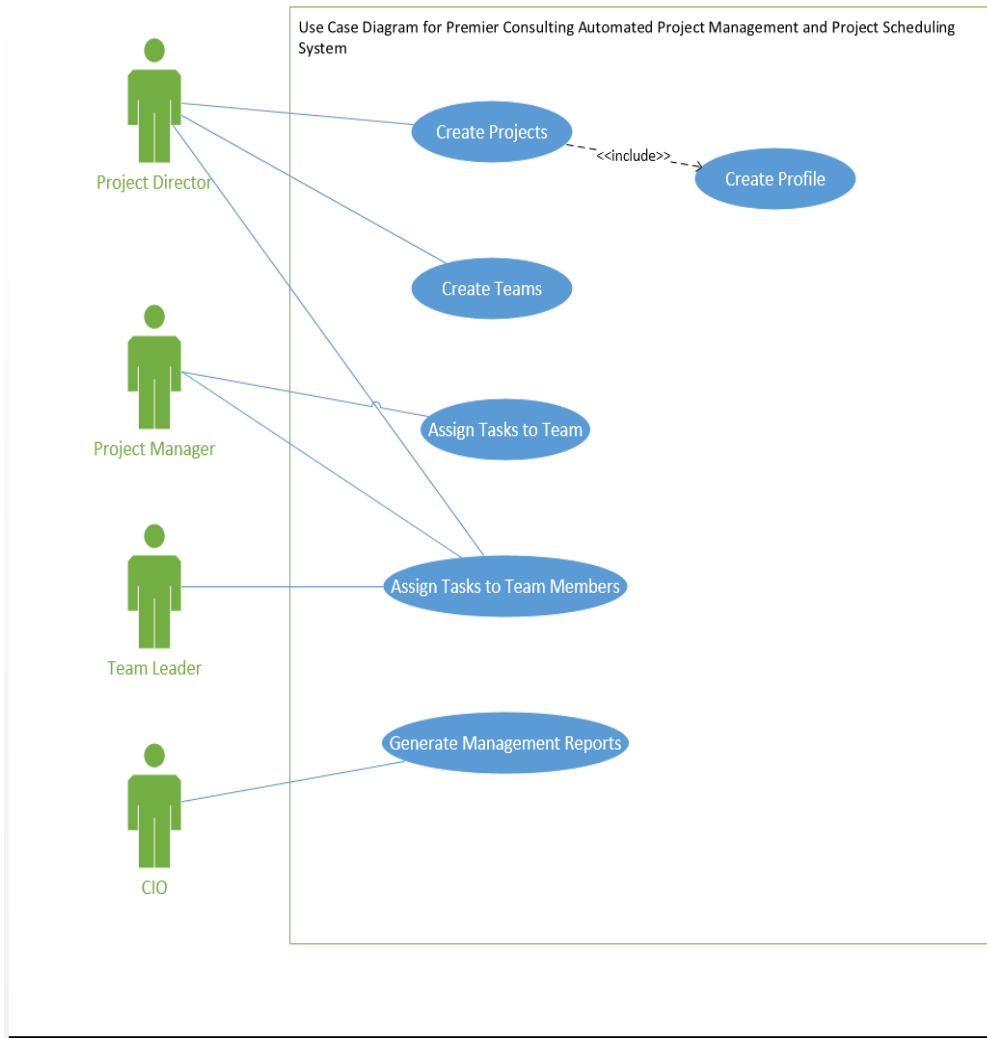


Figure 9: Use Case Description for Assigning Tasks to Team Member

| | | | | | |
|-----------------------------|--|--|------------------------|---|---|
| Use Case Name | Assign Tasks to Team Members | | | | |
| Scenario | A responsibility of the Team Leader is to assign tasks to the individual team members on his/her team. | | | | |
| Triggering Event | Notice from Project Manager to assign tasks to team members. | | | | |
| Brief Description | The Project Manager notifies the Team Leader that the Project has been created, teams assigned to the project by the Director, and that the Project Manager has assigned the tasks to the team that the team leader is in charge of. The leader is notified that he or she is to assign the various tasks to the individual members on the team. | | | | |
| Actors | Project Director, Project Manager, Team Leader | | | | |
| Related Use Cases | Create Project, Assign Tasks to Project, Create Team, Assign Tasks to Team | | | | |
| Stakeholders | CIO | | | | |
| Preconditions | | | | | |
| Postconditions | | | | | |
| Flow of Events | | Project Director | Project Manager | Team Leader | System |
| | 1 | | 1 | 1 Enters Logon Information | 1 Accepts Logon Information from Team Leader and prompts Team Leader to List Team Members |
| | 2 | | 2 | 2 Lists the Team Members and selects Team Member to be assigned. | 2 Accepts the selected Team Member and Lists the Team Tasks. |
| | 3 | | 3 | 3 The Team Leader selects a Team Task to be assigned to the Team Member | 3 Assigns the selected task to the Team Member (Team Member Task Assignment) Step 3repeats until no more tasks are to be assigned. |
| | 4 | | 4 | | 4 Once all tasks are assigned to the Team Membersthe system notifies the Project Manager to assemble the Gantt Chart and Pert Chart |
| | 5 | | 5 | 5 Logons on to the system; inputs tasks assigned, and generates a Gantt Chart and a PERT chart. The Project Manager notifies the system that the Gantt and PERT are created successfully. | 5 The system confirms that the charts are prepared. The system prompts for approval. |
| | 6 | | 6 | 6 The Project Manager responds with an approval message. | 6 If the Project Manager approves the charts then the system communicates this to the Project Director, and lists the Gantt and PERT information to the Director. The system will prompt the Project Director for an approval decision. |
| | 7 | 7 Project Director responds to the system prompt with his/her approval decision. | 7 | | 7 The system accepts the Project Director's response with a message indicating whether the charts were approved or not approved. |
| Exception Conditions | | | | | |

Figure 10: Activity Diagram for Create Project Use Case

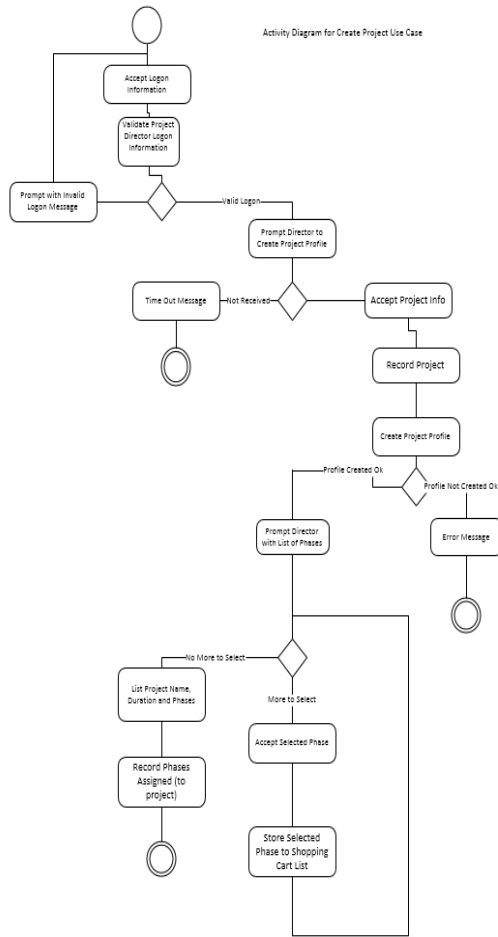


Figure 11: Activity Diagram for Create Project Profile

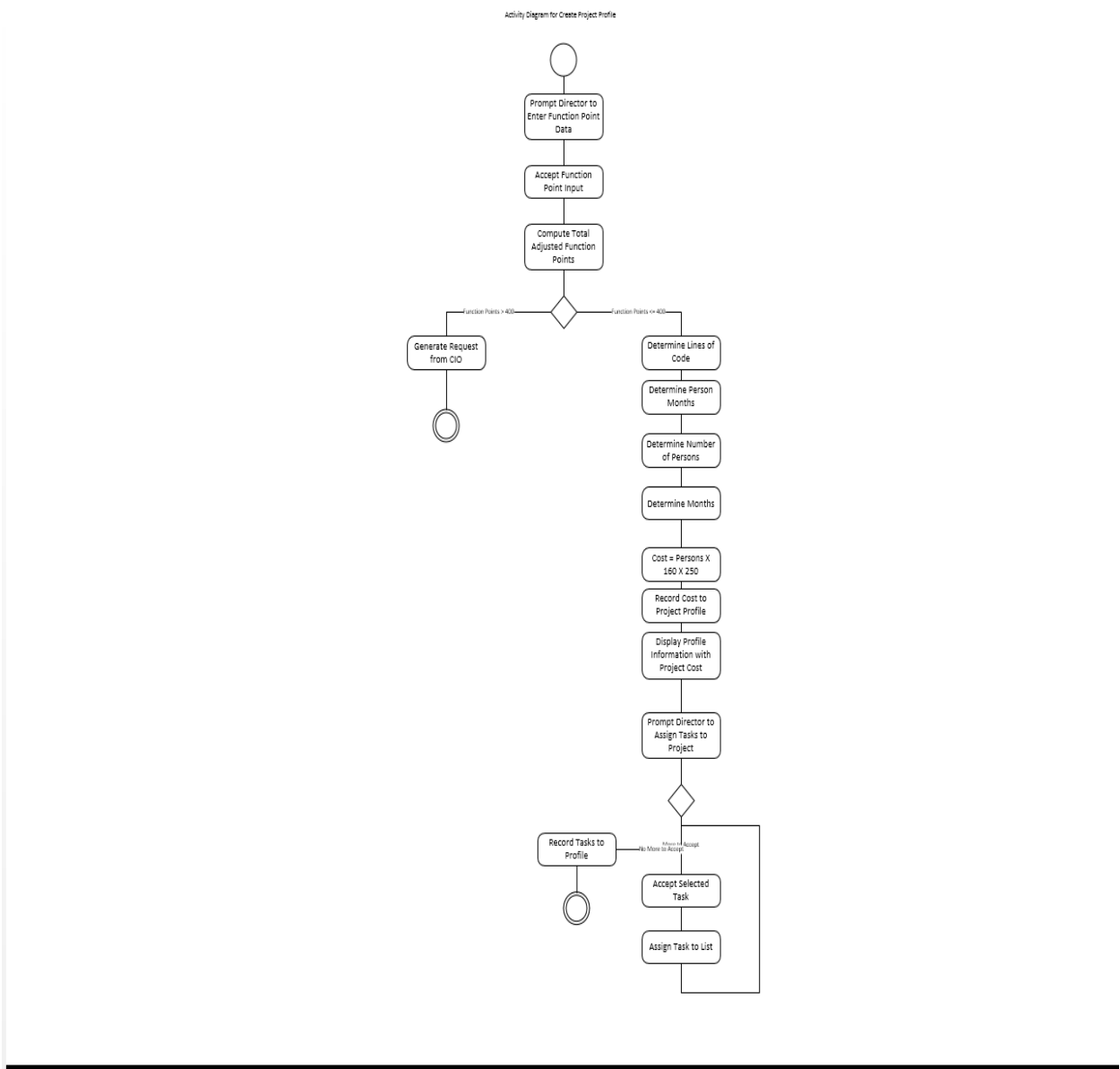


Figure 12: Activity Diagram for Create Team Use Case

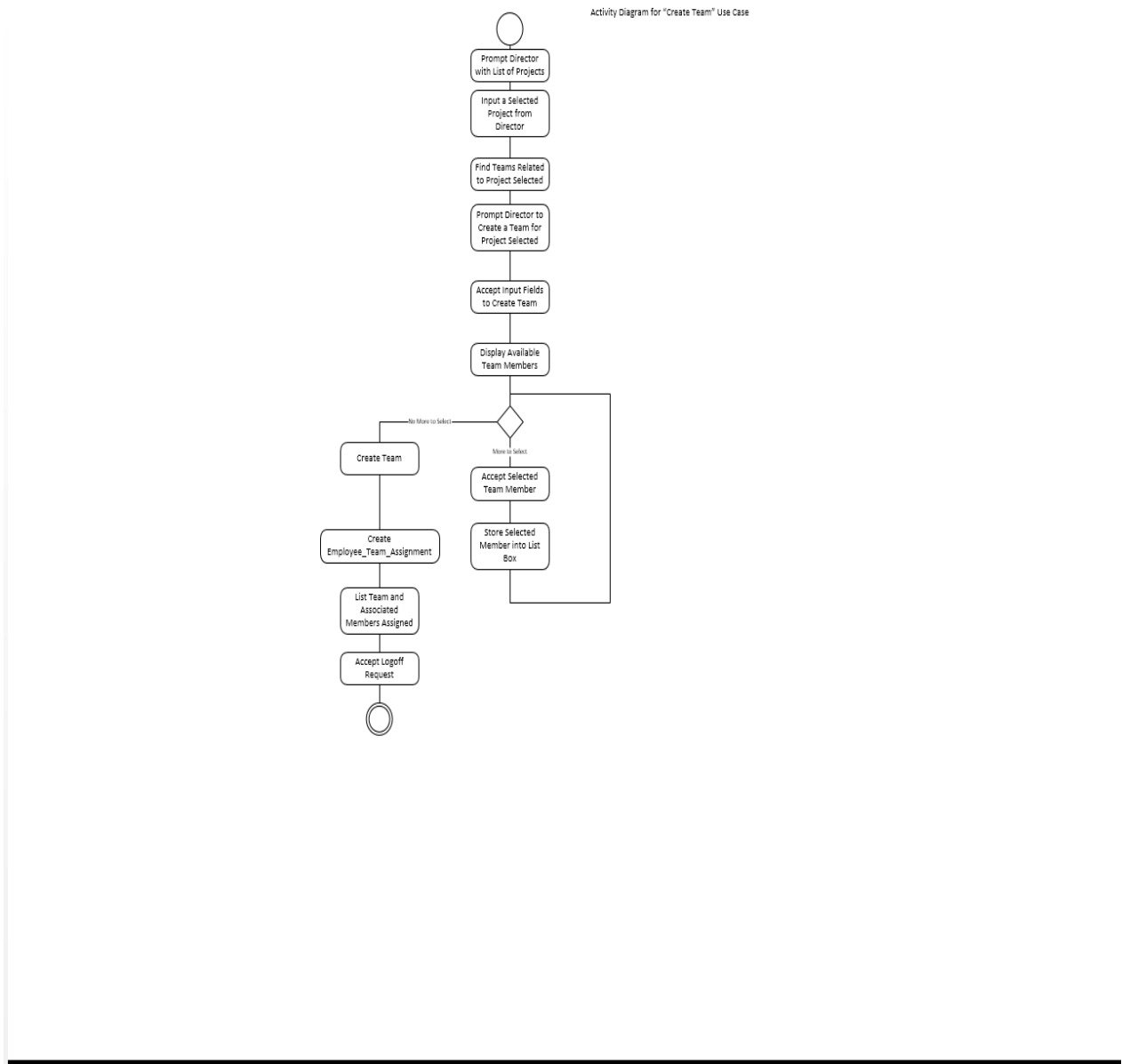


Figure 13: Activity Diagram for Assign Tasks to Team Members

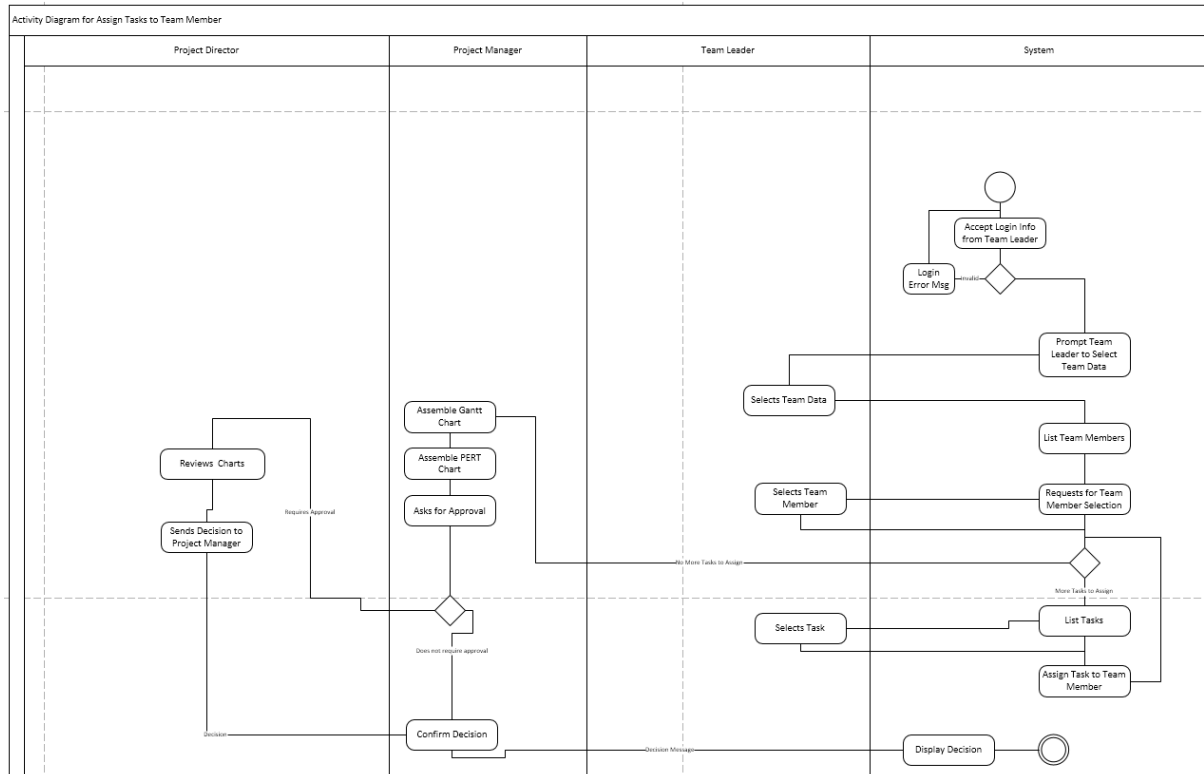


Figure 14: Sequence Diagram for Create Project Use Case

Sequence Diagram for Create Project Use Case

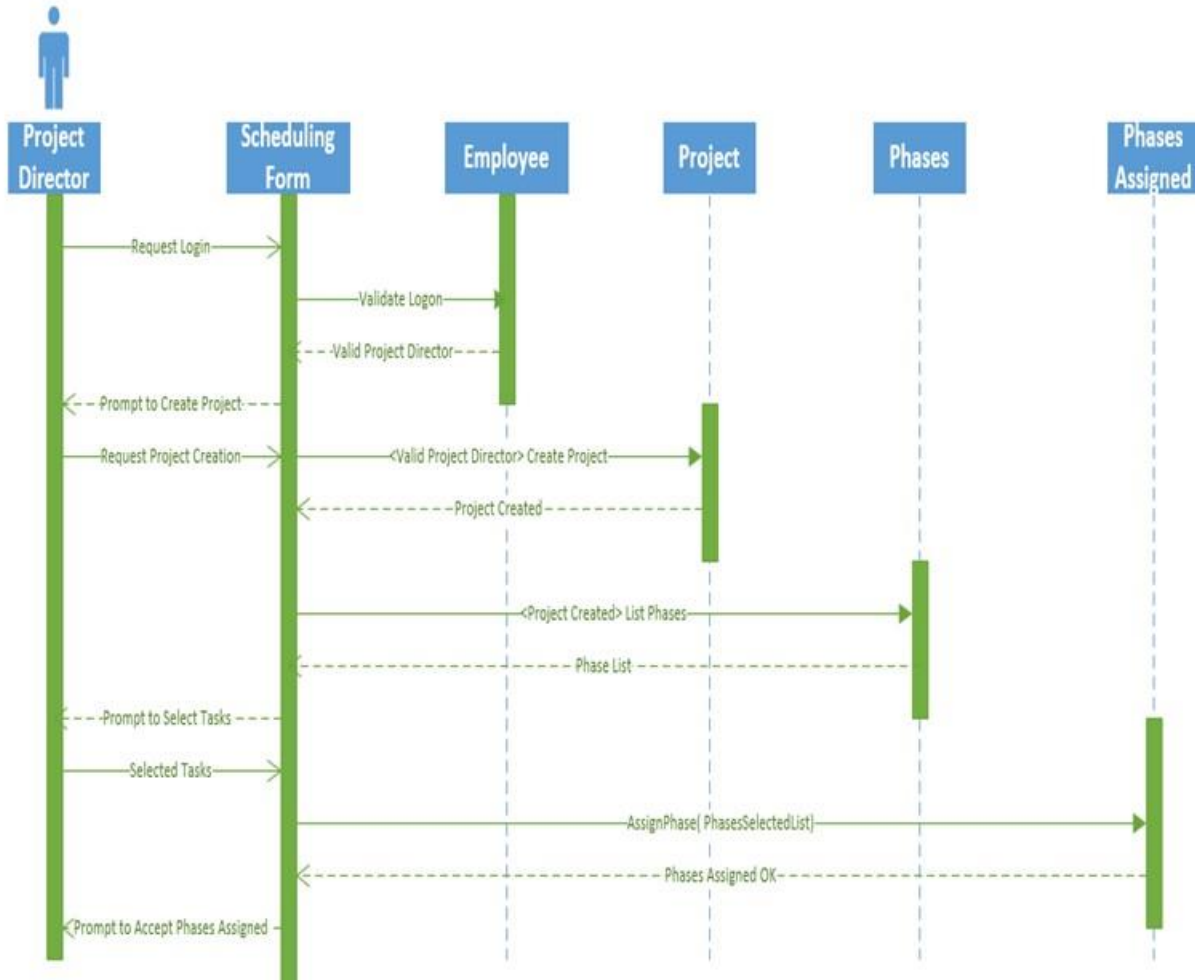


Figure 15: Capstone Class Semester Deliverables and Completion Time Line

| Project Phase | Deliverable | Time Line |
|-----------------------|---|------------------|
| Planning | Project Plan, Gantt Chart and PERT Chart | Week 4 |
| | Function Point Analysis Report | Week 5 |
| | Cost-Benefit and Break-even Analysis Report | Week 6 |
| Analysis | Functional Business Model (Use Case Diagram, Use Case Descriptions, and Activity Diagrams) | Week 7 |
| | Structural Model (Class Diagram) | Week 8 |
| | Systems Proposal Report and Presentation | Week 9 |
| Design | Interface and Navigation Design | Week 10 |
| | Behavioral Model (Sequence Diagrams and State Machine/Transition Diagrams) | Week 11 |
| | Database Design and Program Design | Week 12 |
| | Structured Walkthrough | Week 13 |
| Implementation | Program Development | Week 14 |
| | Program Testing | Week 15 |
| Final | Systems Specification Report | Week 16 |
| | Systems Specification Presentation by Teams | |