

# The Pedagogy of Utilizing Lengthy and Multifaceted Projects in Capstone Experiences

Shohreh Hashemi  
[Hashemis@uhd.edu](mailto:Hashemis@uhd.edu)

Gail Kellersberger  
[Kellersbergerg@uhd.edu](mailto:Kellersbergerg@uhd.edu)  
ELI Department, University of Houston-Downtown  
Houston, Texas 77002

## Abstract

In a capstone senior project course, the majority of project proposals approved by Computer Information Systems (CIS) faculty have a limited life span that does not exceed the period of a semester or in some case two consecutive semesters. There are excellent reasons for such choices, but the result leaves a large area of real-life experience unexplored by students who are about to step into the working world of IT -- and that is the experience of working with already established, complex systems that require maintenance, refinement, and troubleshooting. Success on the job in such a situation requires the ability to analyze and fully understand a complex system in all of its interrelationships. For students, such skill can only come through grappling with an established behemoth which is in need of tweaking. The students, charged with smaller projects within the larger system must comprehend all aspects of the system in order to create and implement, without negative effects, their project. Such experience is hard to come by, yet can be accommodated through capstone experience when an IS faculty can plan a longer time-spanned, complex project with a patient user and develop a pedagogy for such an unusual project. This paper presents a methodology through which the IS faculty can achieve this end. To illustrate the various aspects of the approach, an 8-year project involving the creation of a complex database is presented, with examples of how new students manipulated the project and interacted with previous students to maintain continuity and further the system to its final completion.

**Keywords:** Capstone Project, Information Systems Project, Multifaceted Projects, Computer Information Systems, Internships, Project Management, Senior Projects, Systems Analysis and Design

## 1. INTRODUCTION

Pedagogy can be defined as the underlying theory for and "strategies, techniques and approaches that teachers can use to facilitate" learning (Ohio State University, Commitment to Success Program, Glossary <http://ftad.osu.edu/CSP/glossary.html>). The theories that underlie the creation of capstone courses have been around for a while and have found general acceptance in uni-

versity doctrine. It is understood that senior level students benefit from an integrating, unifying experience that fosters bringing together the knowledge and skills learned from various university courses (Rebhun and Hashemi, 2005). There is also agreement that projects moving from the theoretical base into practical accomplishment are beneficial in preparing students for the workplace (Martincic, 2007). In addition, capstone courses are expected to enhance

skills consistently targeted by higher education such as critical thinking, student initiative, teamwork and collaboration. In fact, many universities evaluate proposed capstone courses based upon checklists that detail such goals (Miami University, Ohio Course Proposal Review Form, [www.units.muohio.edu/led/NewCourse/Review%20form-Capstone%20\(9-05\).pdf](http://www.units.muohio.edu/led/NewCourse/Review%20form-Capstone%20(9-05).pdf)).

The authors of this paper are in agreement with basic pedagogical theory for capstone courses, but see an unaddressed issue in the development of CIS capstone projects that move students from theory to practical applications (Kellersberger and Hashemi, 2006). Theoretically, students should become further prepared for the workplace through capstone projects. In reality, students are mainly exposed to short-term and fly-by systems that seldom reflect the tasks assigned to new graduate hire; i.e., refinement of existing complex applications. The lack of experience on large-scale integrated systems and the limited understanding of processes associated with total project management contribute to project failure (Gauld, 2007). Of course, the reason for limiting student exposure to short-term and small projects is understandable. When a Computer Information Systems (CIS) degree program allows for only one semester of capstone coursework (3-hour credit), the CIS faculty is faced with approving simple senior project proposals that can be completed within a long semester (15 weeks). Usually, the CIS faculty is unlikely to approve projects that require more than 150 man-hours, fearing that students may not have sufficient time to successfully complete the work within that semester period. Most CIS faculty, instead, request that a reduced project scope that is accomplishable with the 15-week period is submitted, thus achieving a complete systems development life cycle experience. Therefore, students are mainly exposed to short-term and fly-by systems such as small databases for small businesses or simple informational websites. While such projects provide an overall view of system design and development, they do not prepare the student for the typical systems analyst job given to new graduates, where the new hire must work with integrative and complex systems already in place that require appreciation and full comprehension of the existing system at hand, as well as its

integrated processes and data. Thus, this understandable mindset of faculty approving small capstone project proposals imposes limits.

Such limitations, however, can be avoided. Long-term projects on existing, multifaceted systems are possible and provide attractive experiences for capstone students. The goal of a capstone course is for students to use and enhance their previously learned skills and knowledge, an experience that will support them in later workplace environments (Kellersberger and Hashemi, 2006). Since project scopes vary, the outcome in student learning at the micro level is specific to the project that the student is engaged in – for example, an outcome might be that students learn more about small databases by discovering that the table where a particular piece of data is stored is corrupted. On the other hand, at the macro level, the learning outcome relates to increased knowledge and skills in such areas as project management, analysis and understanding of complex existing systems, teamwork (where applicable), interaction with user and project manager, troubleshooting and the myriad other abilities called upon in the workplace to bring work tasks to completion.

## 2. PEDAGOGY

In essence, the pedagogical thrust when using a complex, long-term project is providing students with access to and tasks associated with a complex system in order for students to gain experience in analyzing existing, multifaceted systems. The students must achieve a full understanding of the scope of the entire system even though they are working on a subset of a much larger and complex system. They must fully understand and appreciate the work that has been done before them by performing a complete analysis of the system to learn the intricate parts of the system, fully assimilate the work of their predecessors, and discover an effective way to implement and integrate their project modules into the overall system without impairing it (Malinowski and Noble, 2006).

Providing students with access to and tasks for a complex system poses problems that require pedagogical processes. Basically, the pedagogical construct needed is an approach which supports the CIS faculty in framing, approving, and managing lengthy,

complex projects for capstone experiences. The faculty member must have a complete understanding of the scope of the entire project to weigh its value and its learning opportunities for students against the increased involvement and time span required to make each student's interaction with the project a meaningful experience and for project completion. Such an approach could involve the spanning of time, manpower, resources, courses, and disciplines. It requires a user with a long view, patience, a non-immediate need, an interest in providing in-depth experience for students, and enough of a funds insufficiency to make waiting for the final result attractive. It requires additional involvement of the CIS faculty in order to be able to break down the project scope into manageable, semester-length work break-down structures (WBS), to assign them to the right group or individual, to work as a liaison with the user, and to direct the work break-down structures in order to extract the maximum teaching benefit from them. There is more opportunity for flexibility in assignment, talent to bear on the project, and ways of dividing the project, but the accompanying caveat is that there is more opportunity for error that can threaten the entire system and thus the overall project result. The CIS faculty must exert a tight review of system functionality and performance at each phase of the process – in other words, to be a deeply involved project manager.

The pedagogical model in this paper, then, sets up the faculty member as a hands-on project manager who performs two sets of related tasks – soliciting, compiling, evaluating, and choosing projects that meet the educational objectives of a capstone course (the full extent of system analysis and development [SAD] experience), and managing the long-term project. The CIS faculty performs the following functions prior to selecting projects:

- Analyzes project proposals for teaching benefits
- Defines overall scope of proposed projects
- Evaluates each proposed project's stakeholders' viability
- Determines time span of entire project and whether or not it can handle interruptions

- Selects a project that meets the full expectation of a senior project course.

After a project is selected, the CIS faculty performs the following functions:

- Establishes and secures budget for project
- Develops high-level work breakdown structures (WBSs) for each stage of the project that support both project completion and teaching benefits
- Sets up each capstone course experience to accommodate full SAD experience and system understanding, student workload, and system documentation in the way of project deliverables
- Selects student participants at each new stage of the project
- Maintains communication between user, faculty and student
- Serves as a liaison between students and users over the course of the project
- Evaluates individual projects at the end of each of the four SAD phases and assesses final deliverables

These functions comprise the basic activities of instruction and serve as the pedagogical model for a long-term CIS capstone project. A discussion of each function follows.

#### **a. Evaluate user's viability and serve as liaison over the course of the project**

A project manager must look at the viability of the major stakeholders. The signature authority of the user's department must fully support the project. Otherwise, the faculty project manager may encounter hurdles along the way, anything from lack of funds for necessary items to the pulling of essential personnel from the project at critical times. The end-user's lack of sufficient resources may motivate them to wait patiently through the long permutations of the project before receiving the desired end result. Such users are unlikely to be found in the business environment, where systems are needed now and budgets are set aside to support completion of the project. However, in the university environment, it is fairly easy to find entities who are using a system that meets their essential needs but does not offer the niceties that could make their functioning more elegant and targeted. Funding is often not available for such extras, but those same refinements can be the

target of a long-term project for capstone experiences.

**b. Define scope of project and determine if it can handle interruptions**

The faculty member must have a very clear understanding of the project in all of its permutations. The user is first interviewed to get an in-depth understanding of the needs of the department or business in relationship to the project and to note any challenges such as timing, interruption, personnel and other factors that may impact the project over time. Others who may be involved in the testing, information provision, and management of the final deliverable are interviewed to determine their viability in working with students and with a prolonged project where repetition of actions may be required. Any existing materials, software, programming, and related data are analyzed to help the faculty understand breadth and scope of the project.

**c. Determine time span of entire project**

The faculty member now identifies within the high-level project scope integral, interrelated subsystems. Each subsystem is now divided into WBSs. This determines which WBS must be completed before the next one can commence. These segments must be arranged so that each can be completed within a semester's time. Some small WBSs may require a single capstone student to effect completion in a semester (100-150 man hours), while other larger and/or more complex WBSs may require multiples of 100-150 man hours and groups of students in order to reach completion within a semester. Based upon an understanding of how much progress can be accomplished by each student or group of students in the length of one capstone course, the faculty determines how much time it will take to deliver the final project to the user. This also tells the faculty how many capstone experiences the project will support.

**d. Establish and secure budget for project**

Budgeting is a significant factor in determining the viability of taking on a long-term complex system as a project for capstone experiences. Issues involving time by salaried persons must be agreed upon at the

project's onset. Issues involving the purchase of any requisite hardware or software must be planned and budgeted for prior to committing the first group of students to the project. Budgetary support for supplies and other required resources must be planned and agreed upon. In some cases, new hardware must be purchased, installed, and maintained. Additional needed software must be purchased, installed, and tested. Man-hours dedicated to the project by the user have to be supported and planned by the manager of the unit, which means that the project must have the full support of the unit manager (major project stakeholder).

**e. Analyze project for teaching benefits**

Once the segments are determined, the faculty member considers the work required in each segment and extracts the possible teaching benefits in the work. That is where the emphasis of the course will lie. Some of the benefits may be obvious and applicable to most segments, such as critical thinking and in-depth analysis enhancement. Others may derive from the type of work required in that segment, such as user interface development skills. Since the goal of capstone CIS courses is to provide students an in-depth system analysis and design (SAD) experience, each student is required to perform all tasks associated with full SAD (analyze, design, code, test, implement, etc.)

**f. Develop WBSs for each stage of the project that support both project completion and teaching benefits**

As previously noted, once project scope is well-defined and time implications are clear to the faculty, then the faculty must create WBSs that have clear deliverables to assign to individual students or groups of students. The clarity of the WBS is a strong factor in the success of the project. The faculty must be certain that each WBS can be achieved within the semester's time and that each student is fully engaged in the systems development life cycle (SDLC). The time required to complete the WBS and the length of the course determines what WBS is selected for student(s) to complete. If the next step of the project requires a specific WBS which does not meet the course time allotment, a semester's rest may be signaled. Therefore, the faculty must choose projects which can accommodate interruptions in the timeline because, at times, set-

ting time constraints could impede student learning or jeopardize the integrity of the project.

**g. Set up each capstone experience to accommodate system understanding, student workload, and documentation in the way of deliverables**

Because the learning curve is always steep in a project where students must first analyze the complex system before implementing new parts, about one-third of a semester should be set aside for student understanding of the system in place and what is required of the student, another third for new components design, implementation, testing, and integration into the existing system, and the last third for documentation. Students are required to work with state-of-the-art technology available at the university. Of course, just as in any real-world system, a major change in technology during the long-term span of the project could force a complete redesign of the system (Allen, 1977). Retirement of the base software or hardware would be a good example of the type of major change that could require abandoning the project and constructing a new one. In our case, this would have required something as major as Microsoft going bankrupt, Access being retired or PCs becoming obsolete. A clear and tangible description of deliverables is necessary and should reflect the timing of each capstone assignment. The project deliverables, which include organization charts, Gantt charts, network diagrams, context diagrams, entity relationship diagrams (ERD), Data Flow diagrams (DFD), Program Evaluation Review Technique (PERT) charts, diaries, technical environment diagrams, a user manual, installation manual, and a back-up and recovery manual as well as full documentation of the system (both print and electronic copies), are compiled in the project notebook which later serves as the user's technical diagram and is used for referencing, especially for trouble shooting and system enhancements (See appendix 4).

**h. Select student participants at each new stage of the project**

Manpower (student) selection is a factor of a successful project. The faculty must determine which parts of the project can be handled by groups and which parts by individuals so that the project can be assigned ac-

ording to class composition and WBS complexity. The faculty must also, especially in the later stages of the project, be more cautious when evaluating whether a student's abilities are right for that stage of the project. A student's lack of familiarity with the system as a whole or its intricate, inter-related parts could lead to design and implementation problems that could so profoundly affect the system's integrity as to make recovery more time-consuming than the senior project course could accommodate in one semester. By then, the student would not only have lost the in-depth exposure that makes the capstone course such a powerful learning experience, but would also face the possibility of an incomplete grade at the critical time of an approaching graduation. For example, students lacking a background in database development may never understand the complexities of a database-driven system in enough depth to add refinements to it (Lenox, L., et al., 2004). Thus, there are semesters when no students should be assigned to the project because the necessary abilities or needs are not present. There are semesters where a group might work on the project, and others where an individual will be assigned – all of these depending upon the faculty's analysis of what is needed for the project and what is needed by the student(s).

**i. Maintain communication between user, faculty and student**

Communication is an essential skill in long-term project capstone experiences. As the overall project depends upon completion of scheduled WBSs, it is important for the faculty member to keep close tabs on student progress. For this reason, it is advisable to schedule several stages of oversight of deliverables and to maintain weekly meetings with students. One way to handle this is through review and evaluation of reports, charts, diagrams, and codes at the end of each SDLC phase. Communication between students and user is an area vulnerable to misunderstanding, especially when the user is not schooled in the language or processes of CIS functions. Therefore, three-way communication (via email, meetings, and conference calls) among students, user and faculty member are suggested to avoid concern or misunderstandings that impede the project. Communication between student team members can also be spotty, and a

faculty member who is using a group of students for a set of WBSs should establish formal group communication requirements.

**j. Evaluate individual projects at pre-established stages and assess final deliverables**

If a firm, tangible set of requirements for deliverables which follow a timed schedule throughout the capstone course has been established, the faculty member is much better able to judge the progress of the project and intervene if necessary to get things back on an even keel. The final assessment for the course is based on the entire project notebook containing all required deliverables. As these deliverables were derived from WBSs that were established to capitalize on the teaching benefits identified in the project, successful completion of the deliverables indicates a successful capstone experience based upon a long-term project (and also reflects upon the success of the pedagogical modal used to develop the capstone course). The faculty member can be confident that the student has gained valuable skill and has had an integrative and practical experience that directly reflects the workplace environment into which the student will soon enter.

To support a full understanding of the pedagogical model suggested in this paper, a complex, long-term project – the ELI Student Information System project – accomplished through CIS senior projects and CIS directed studies spaced over an eight-year period of time is presented as an example as a multifaceted capstone project.

### **3. BACKGROUND AND PROJECT SCOPE**

#### **The ELI Need**

This example project was completed for a continuing education department at the University of Houston-Downtown (UHD <http://www.uhd.edu>) – the English Language Institute (ELI <http://www.uhd.edu/eli>). The ELI trains internationals in English language skills and offers a wide variety of educational programs, everything from a classic intensive English program to contractual work with companies both local and in other countries. Because of the range of programs offered, the ELI crosses the lines of both academic structures and continuing education structures.

In 2000, the ELI had an unsophisticated student database built by an enterprising faculty member with Filemaker Pro. As might be expected from a system designer with little programming and system design background, the database was limited from the outset in what it could do. Later, the university implemented new technology which forced the migration of the student database to a new platform. This caused instability in the database that eventually became fatal. However, several years before the database crashed, the ELI director recognized the downward trend of the database and began searching for a replacement which would be more sophisticated and robust than the initial database.

#### **Project Purpose**

The purpose of the project was to build an elaborate Student Information System (SIS) for the ELI that performed many of the tasks required by the department which its then-current automated system could not perform. Several factors converged to make this project needed and ideal for the capstone experience of CIS students.

#### **Project Background**

The student information system available to the rest of the university was Banner, a computerized student information system used by many universities including UHD. Although Banner offers many functionalities, it would have required major customization to meet ELI's basic intensive program needs, and it could not have addressed both the academically structured programs and the continuing education structured programs within the same Banner system. The ELI would have had to operate two separately functioning Banner systems with different user interfaces and processes. In addition, both systems would have had to have been set up to ensure independent functioning from the regular university Banner system. This was due to the many differences between the ELI infrastructure and that of the university. For example, the ELI has a markedly different calendar from the university. Fee structure and collection methods for each internal program follow a separate logic. The nature of the ELI student requires different identifiers due to tracking requirements from the federal government, issues with students far from home, emergency functions, and other factors. Contractual

elements and recruiting elements require additional system modules that are not existent in Banner. ELI educational programs have differing credit status and transcript needs, and the entire department's soft-money status adds an entirely new layer of needs. In short, the department is a complex interweaving of educational program types which made the needs of a student information system more complex than those of a straight academic program. University IT personnel were unable to dedicate the manpower required to make the appropriate changes and additions to Banner for it to meet the needs of the ELI department.

Furthermore, commercial databases such as Peopleware were expensive, very generic, and had limitations in desired functions because they were solely dedicated to a continuing education infrastructure. Since many elements of the ELI's programs were more like academic coursework than continuing education coursework, these software packages were not robust and comprehensive enough to meet all of the ELI's requirements of a student information system.

Since no viable solution presented itself, the ELI director approached the CIS faculty to inquire if the existing system could be fixed or if a customized student information system could be built by the CIS senior project students.

### **CIS Need**

Meanwhile, the CIS faculty had been looking for a long-term, complex project that could be monitored closely to assign to students to provide them with a different type of capstone experience.

### **Collaboration, CIS and ELI**

Thus, a preliminary period of investigation was begun by the CIS faculty to determine if the project was suitable and if the ELI was a viable user (*evaluate user's viability and serve as liaison over the course of the project*). The ELI director provided for the CIS faculty a description of the program specifics (see Appendix 1), handling such things as term of study, levels of study, various programs, methods of promotion, financial data (tuitions, fees, full payments, partial payments, insurance, materials, discounts), student data, Homeland Security data, and a host of other items. The ELI

director also provided a list of the types of reports desired of the student information system (see Appendix 2).

Upon discussion, it became clear to the ELI director that fixing the existing database would be a major undertaking and would still result in a limited system. After a thorough analysis, the CIS faculty suggested the possibility of an entirely new Student Information System (SIS) which would address all of the ELI department's needs in keeping track of student and financial records (*define scope of project and determine if it can handle interruptions*). It was recognized that this would be a complex and long-term project that would have to be worked on by a number of students over many semesters and that would require a significant investment of time and patience from the ELI staff as the system needed to keep track of six separate areas – demographics, immigration data, student performance data, financial data, registration tracking, and recruitment data – and generate around 60 separate queries and reports (*determine time span of entire project*). The project would also have some budgetary implications. The computer that would house the project for ELI data testing was old and had to be replaced. The ELI secretary would have to devote a fair amount of time to working with CIS students and testing their results. Several flash drives were required over the length of the project, and, as it turned out, time had to be spent on developing and processing interim data reporting (excel sheets) when the ELI database crashed. These budgetary requirements needed to be covered by the ELI (*establish and secure budget for project*). The ELI director accepted the suggestion and the budgetary responsibility, and so the project began in earnest during the summer of 2000.

The CIS faculty and the end-users of this project worked with all involved students over the eight-year period. Much of the work was retraced in that new sets of students had to analyze and understand the system – this was a basic course objective – but each student or group of students furthered the project and felt a sense of ownership for the concrete accomplishment of that stage in a complex project. In addition, students remained connected to the project after course completion by sharing their project notebook and their understanding of



both the project and their portion of it with new students coming into the project with new objectives. All in all, the project was considered a success by the CIS faculty and the end users.

#### 4. Implementing the Multifaceted Project

The project, from outset to present configuration, involved many steps. The CIS faculty began, as Project Manager, to analyze the project, develop the WBS, determine teaching and learning objectives associated with WBS, and the process for delivering instruction to achieve these objectives (*analyze project for teaching benefits*). Understanding the magnitude of the project, the CIS faculty divided it into stages – understanding of the existing systems (both automated and manual), documenting the existing systems, diagramming the various involved departmental processes and their dependencies, documenting problems associated with the outdated automated database, compiling, categorizing, and cataloging forms and reports in use at the time, conducting JAD and RAD sessions to design, develop, and implement a preliminary prototype of the SIS, further understanding of the intricate parts of the department's internal processes, refining the prototype, adding functionality and GUI components to the prototype, testing the enhanced prototype, and repeating the process with each new group of students until the system was ready to be fully implemented and delivered to the ELI department (*develop work breakdown structures for each stage of the project that support both project completion and teaching benefits*).

A sample of WBSs developed for the ELI project follow.

- Create a prototype of the system
- Identify data elements
- Identify and create tables and relationships
- Document processes
- Catalog forms and reports
- Design a series of preliminary user interfaces with background processes
- Refine the prototype of the system.
- Conduct a detailed analysis
- Create additional tables
- Design, develop, implement and test additional GUIs

- Identify additional processes, and sub-processes, and data elements
- Add components and processes
- Test the prototype
- Document and diagram
- Analyze, design and develop the SIS
- Conduct JAD and RAD sessions with user
- Identify possible problems and document possible future enhancements
- Repeat the process, building and refining the SIS, looking for inconsistencies and
- Add functionality to the prototype
- Analyze the system and study documentation prepared by the previous group
- Study the preliminary design
- Meet with users on a regular basis to discuss the preliminary design
- Create more attractive GUI interfaces
- Understand the various processes and outputs
- Identify any unknowns
- Resolve any unaddressed issues
- Add new functionalities

An on-going task of the CIS faculty was to ensure that the project was protected and that the learning objectives could be assimilated by the chosen participants (*select student participants at each new stage of the project*). Approximately a week before each semester began, senior project students' GPAs were checked and the ELI system was assigned to those familiar with Access and Crystal Reports (needed for system development) and those with the highest GPAs. This was done primarily because, as time passed, the ELI system became more and more complex and demanded a higher learning curve for understanding the many data elements, tables, relationships, and other various intricate elements. It was assumed that students familiar with Access and Crystal Reports and with higher GPAs would have a better understanding and appreciation for the complex data base.

As each capstone course was planned, a timeline was established that permitted a preliminary understanding of the project and system as well as an in-depth analysis of what had been accomplished by previous students (*set up each capstone experience to accommodate system understanding, student workload, and documentation in the way of deliverables*). WBSs were assigned



that did not require more than one third of the course time and that could be accomplished through the knowledge gained by the student in previous coursework. Deliverables were defined and documentation requirements outlined. Most capstone courses included a final presentation of deliverables to students and faculty along with written documentation in a project notebook.

Actual implementation of the project began during the first four weeks of the nine-week summer term in 2000. The first of the 14 students who worked on the project was assigned to meet on a regular basis with the users to gain enough understanding to create a prototype of the system. Follow-up calls and visits by the faculty member were made to the user to ensure that there were no misunderstandings (*maintain communication between user, faculty and student*). Thereafter, and for the last five weeks of the semester, the student was required to design a series of preliminary user interfaces along with background processes. Using JAD and RAD, the student delivered a prototype of the system at the end of the nine-week period which was assessed in accordance with defined requirements (*evaluate individual projects at pre-established stages and assess final deliverables*). This semester's work provided the student with a tremendous amount of analysis, some system design and experience in implementing a minimum system with unsophisticated GUIs and background processes, always with the understanding that this was not expected to be the final system (sample of teaching benefits). This first experience also initiated the users in working with CIS students who used computer jargon and notations, and who were driven by a set of steps and deadlines imposed upon the work by the course structure. The users were also forced to develop an ever-increasing understanding of the department processes in order to explain and delineate fine points.

The following semester, capstone students with the needed expertise were not available to the CIS faculty, causing concern about losing momentum and possibly the user. While allowances had been discussed with the user about the possibility of "rest" semesters, this seemed too soon in the life of the project. Therefore, the CIS faculty recruited two directed-studies students with the needed expertise to continue work on the

project with the goal of repeating the process followed by the first students in order to produce an even more refined prototype. These students were given the summer student's project notebook containing all system-related documentations and were asked to meet with the users on a regular basis to further understand, identify processes within, refine and test the new prototype, all the while, documenting and diagramming all. At the end of that semester, the infrastructure for the new SIS started to take shape. Once again, the two students had performed extensive systems analysis and design, updated an existing system (the prototype), and developed, implemented and tested new system components. The users learned to document enough about the process of a semester so that they were able to remember where and how to proceed with the next phase.

By the following semester, enough information had been gathered and sufficient preliminary work had been done on the prototype so that serious work toward the SIS could now begin. The CIS faculty had carefully sifted the project for its teachable elements, established a schedule of WBSs for each student or group that followed, and categorized each WBS to accommodate system understanding, student work, and documentation. At this juncture, a group of five senior project students were assigned to the project. This number would result in 700 man hours, a boost to move the project from its prototype stage to an actual system. A group was formed, a group leader was chosen, all previous system documentation was provided to the group, a preliminary meeting with the user was scheduled, and the group began the process once again of analyzing, designing, and developing the SIS. The group performed a series of Joint Application Development (JAD) and Rapid Application Development (RAD) sessions with the user to depict the preliminary system design. At the end of that semester, a viable, partially working SIS was delivered to the CIS faculty, who saw this as a first concrete expression of the true SIS. The use of groups was found to be very important, as group discussions helped individual students with deeper understanding of the system, as well as generated questions and clarified ambiguities.

The following semester, another group of senior project students were assigned to the project. This group was asked to study the system documentation prepared by the previous group; study the preliminary design; meet with users on a regular basis to discuss the preliminary design, especially the GUI interfaces; understand the various processes and outputs; and identify any unknowns. By the end of that semester, the ELI system façade was completed, additional processes were added to the skeleton system built by the previous group, and a system prototype was implemented. The following semester, the project was handed over to a student who was asked to repeat the last semester's activities, looking for any inconsistency and adding functionality to the prototype. By the end of that semester, the ELI system was shaping up and ready for further development. Meanwhile, the existing ELI database crashed irrevocably, and – faced with no database but one in the making – the user chose to rely upon excel spreadsheets of data rather than ditch the effort which was now in its fourth year and promising.

For the next six years, other CIS students worked on the project to resolve previously unaddressed issues and to expand the system with more functionality while the ELI limped along, gathering its statistics by hand. In 2008, the new ELI SIS came into its last stage of refinement.

A total of 16 students spent 1600 hours (an average of 100 man-hours by each student per semester) to devise and implement this complex SIS system. At each interval, a more attractive set of GUIs were developed and additional functionality implemented. At the time of this paper, the system is almost ready for full implementation (see Appendix 3).

## 5. CONCLUSION

Complex, long-term projects can be successful senior projects and directed studies topics, and can provide full SAD experience usually unavailable to students soon to graduate from CIS programs. The disadvantages of using long-term projects for capstone experiences lie in the delicate balancing act performed by the instructor and the length of time of user involvement before a completed project can be delivered.

Exquisite up-front and on-going planning is required from the instructor.

Patience (and good humor) is required from the user.

Some semesters may be as much of a step backward as a step forward if capstone students succeed in damaging one part of the project in order to forward another part.

There can be frustrating downtimes (especially to the user) while waiting for appropriately skilled capstone students to continue on the project.

The advantages of using long-term projects for capstone experiences lie in the students' immediate application of resulting skill achievement to jobs typically filled by new graduates (system analyst positions).

First and foremost, students experience working with a large, complex, in-place system put together and documented by others who may or may not think and work in the same patterns as does the student

Students learn how to gain a global understanding of a large, complex, in-place system

Students gain experience in mapping and working with components that have critical interrelationships

Students experience working within a complex system where an ambiguous command or line of code can do anything from affecting a function of the system to crashing it

Students, because of the nature of the project, refine their collaboration and teamworking skills by working with users, instructors and colleagues

Students learn where the satisfactions lie in contributing a small portion to a complex project as opposed to authoring an entire project. (Student satisfaction and ownership of the project is fostered through continuity between phases and thus is not a drawback in undertaking a complex, long-term project.)

Though student learning is apparent and answers well to the type of experiences many new graduates will face on-the-job, following certain pedagogical guidelines can add to the efficacy of such

projects. As demonstrated, careful project management is necessary. The end user must understand the time and patience that will be required before the end product is achieved, and be willing to participate. Students must have WBS assignments which can be achieved in the semester's timeframe, knowing that an in-depth understanding of the SIS will be the first major undertaking. Faculty must segment the overall project so that each achievable WBS allows 33% of the semester's time for analysis, 33% for design, and 33% for documentation. Single students and groups of students can be effective, depending upon the stage of the project, and selection of student expertise and experience in the later stages ensures a better experience for the student. Also, the instructor can structure both capstone classes and directed studies classes on the long-term project over its span. In conclusion, the authors encourage reconsideration of the limit typically placed on capstone course projects – that the entire project be completed within the 15-week semester period. In those cases where experience with a complex, in-place system is desirable, a long-term project off of which a number of sequential capstone courses are drawn can provide an excellent opportunity for student learning. Future efforts that would measure and compare learning outcomes of students working on short-term versus lengthy and multifaceted projects can further quantify the effectiveness of using lengthy and multifaceted projects for Capstone experiences.

## 6. REFERENCES

- Allen, T.J. (1977). *Managing the Flow of Technology*. Cambridge, MA: MIT Press.
- Gauld, R. (2007) "Public sector information system project failures: Lessons from a New Zealand hospital organization." *Government Information Quarterly*, 24(1), pp. 102-114.
- Kellersberger, G. and Hashemi S., (2006) "Building an International Resource Center Web Site -- A Capstone Senior Project Course Experience." The Proceedings of ISECON 2006, v 23 (Dallas): §2714. (Revised in *Information Systems Education Journal* 5(12).)
- Lenox, L., Woratschek, C., and Woratschek T. (2004) "The Pros and Cons of Using a Comprehensive Final Project in a Database Management Systems Course: Marvin's Magnificent Magazine Publishing House." Proceeding of ISECON 2004.
- Malinowski, C and Noble, R.E. (2006) "Capstone Projects: Putting the Pieces Together." Proceedings of ISECON 2006, v 23 (Dallas): §2154.
- Martincic, C.J. (2007) "Combining Real-World Internships With Software Development Courses." Proceedings of ISECON 2007, v 24 (Pittsburgh): §1544.
- Miami University, Ohio Course Proposal Review Form, [www.units.muohio.edu/led/NewCourse/Review%20form-Capstone%20\(9-05\).pdf](http://www.units.muohio.edu/led/NewCourse/Review%20form-Capstone%20(9-05).pdf).
- Ohio State University, Commitment to Success Program, Glossary <http://ftad.osu.edu/CSP/glossary.html>
- Rebhun, H. and Hashemi, S. (2005) "Systems Development Projects – Gaining Practical Experience while Meeting Community Needs: A Win-Win State of Affairs." Proceedings of ISECON 2005, v 22 (Columbus OH): §3554.

**Appendix 1**  
**ELI Program**

**Instructional Specifications:**

1. ELI organizes its instruction by terms -- Two 6.5 week terms = one semester.
2. Students may enter one of four programs in the ELI, but majority enter the Intensive English program.
3. ELI instructional programs are Intensive English, Conversational English, Business English, and Computers and English.
4. Students may take short-term afternoon classes in addition to the regular program.
5. Intensive English (IE) students take four different classes every term.
6. IE students are placed into one of seven levels.
7. Students promote or fail each term.
8. Students receive grade reports each term.
9. Students may graduate from the ELI each term.
10. Students receive grades of P (pass), IP (in progress but remaining in the same level), NP (fail), or NPA (fail but may re-take the placement test and be placed accordingly).

**Registration Specifications:**

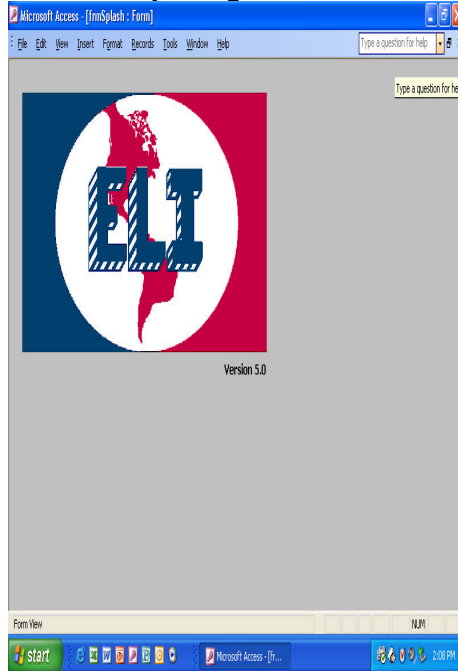
1. Students can register per term or per semester.
2. Students can be residents (green card holders or citizens) or internationals (students on F-1 visas, B visas, or other specialized visas), and tuitions are different for the two groups.
3. Students can be sent by recruiting agencies in other countries.
4. Recruiting agencies may have contracts with us, thus receiving a commission for registered students they send.
5. All students may purchase medical insurance, and F-1 students must have medical insurance.

**Appendix 2**  
**Sample Reports**

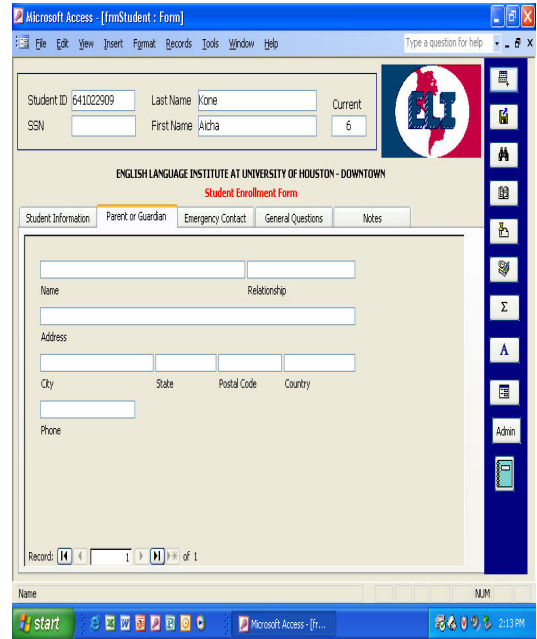
1. Number of students per level
2. Number of international-designated and F-1 visa holding students in program
3. Student gender and Birthday
4. Country of origin and region of origin
5. Class list
6. Financial totals and Number of refunds
7. Payment alert
8. Visitor Visa alert and IP alert
9. Student grade report and Honors list
10. Number of new and former students
11. Number of brochures sent by mail
12. Number of e-mail brochure/applications sent
13. A specific student's teachers
14. A specific level's teachers
15. Number of level failures
16. Total tuition paid by international students, less refunds
17. Total tuition paid by resident students, less refunds
18. Total tuition held over to apply to next semester
19. Number of walk-in brochures distributed
20. Number of applications received
21. List of recommended students
22. Exit test scores by student
23. Michigan test scores by student, by term, by level
24. Scholarship winners
25. Reason for leaving (from pull-down list)
26. Intended major (from pull-down list)
27. Class evaluation results per class, per level, per term
28. Program evaluation results per level, per term
29. Average length of stay
30. Student transcript
31. % of cohort that remains to graduation
32. % level failures in cohort
33. Average number of terms retention for cohort
34. Retention by region and by country
35. Failures by region, country, and skill
36. End of term/semester/year report
37. Number of information requests sent through recruiters by agency
38. Number of registered students sent through recruiters by agency
39. Number of students who have interrupted study
40. Recruiter list and contact list
41. Students by term sent by recruiter on contract list

**Appendix 3**  
**ELI Student Information System**  
**Selected User Interfaces**

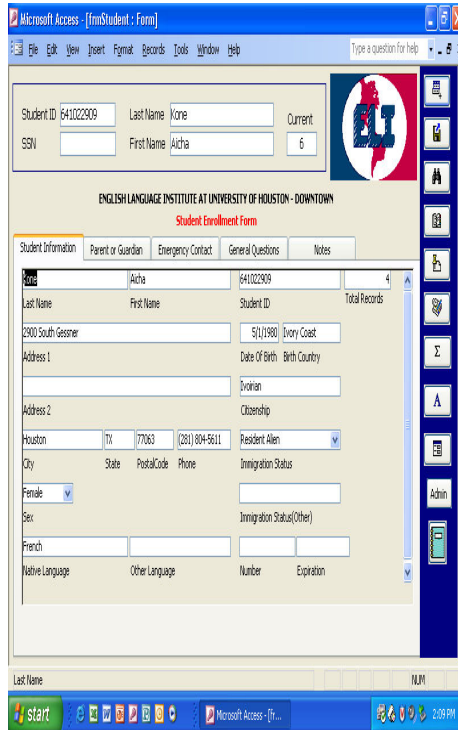
**Opening Screen**



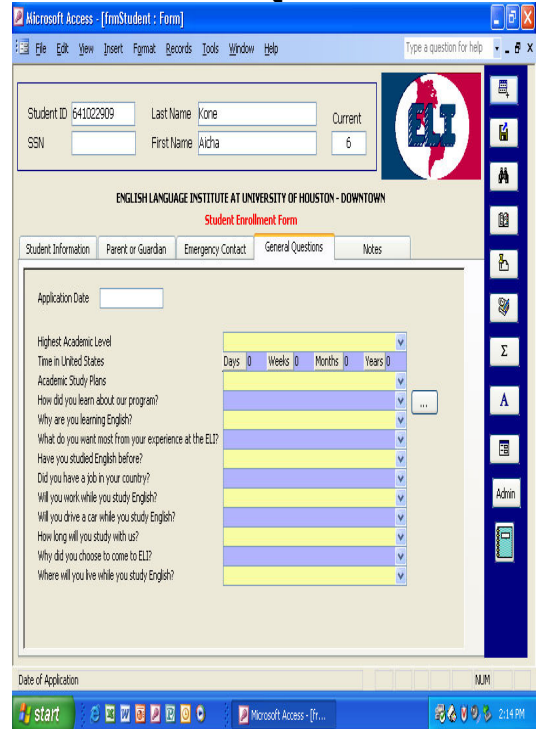
**Student Enrollment form  
Parent or Guardian**



**Student Enrollment  
Student Information**



**Student Enrollment  
General Questions**



### Student Placement Record

Microsoft Access - [frmPlacement : Form]

Student ID  Last Name  Current   
 SSN  First Name

ENGLISH LANGUAGE INSTITUTE AT THE UNIVERSITY OF HOUSTON - DOWNTOWN

**Student Placement Records**

Year  Semester  Term  Level

**Enter Placement Results**

MP1  Writing Test  0  
 MP2  Michigan Total  0

Suggested Level    
 Level    
 Next Level

**Student Placement Records**

Sem	Term	Year	Writ	Mich	Sugg Level	Place Level

Michigan Score Part1 NUM

### Course Listing

frmAdmin : Form

Registration Information Course Fees Course Listing Recruiters

**Course Listing**

ClassCode	ClassName
BE	Business English
IEP	Intensive English Program
SPEC	Special Course

You can add, delete, or modify the courses offered by the English Language Institute by using the Data Grid to the right.

### Student Grade History

Microsoft Access - [frmGrade : Form]

Student ID  Last Name  Current   
 SSN  First Name

ENGLISH LANGUAGE INSTITUTE AT UNIVERSITY OF HOUSTON - DOWNTOWN

**Student Grade History**

Year  Semester  Session  Level  Class

AE  Conversation   
 Reading  Bus. Eng. Lab   
 LS  Bus. Eng.   
 Writing  Gatekeeper Test Score

Final Grade   
 Next Level

Year	Semester	Term	Level	AE	Conversation	Reading	LS

Record: 1 of 1

Form View NUM

### Recruitment Information

Microsoft Access - [frmRecruitment : Form]

ENGLISH LANGUAGE INSTITUTE AT UNIVERSITY OF HOUSTON - DOWNTOWN

**Recruitment Information**

Recruit  Total Record Count  0

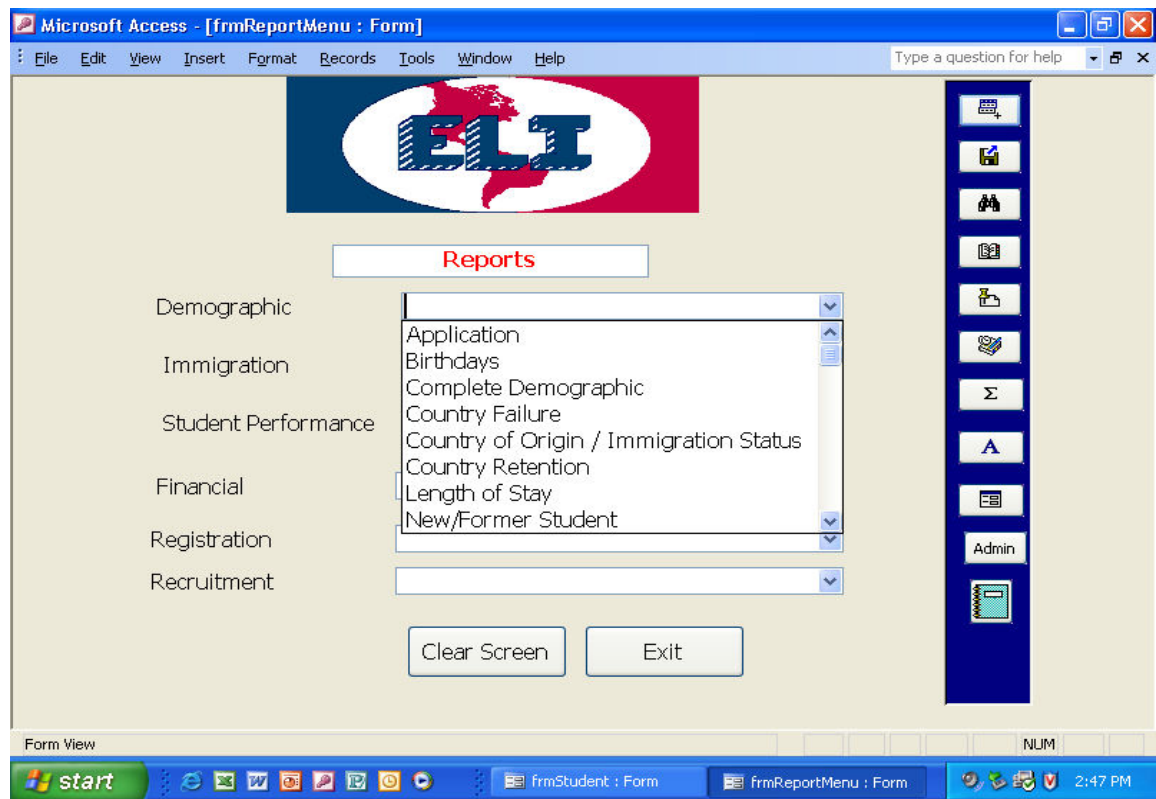
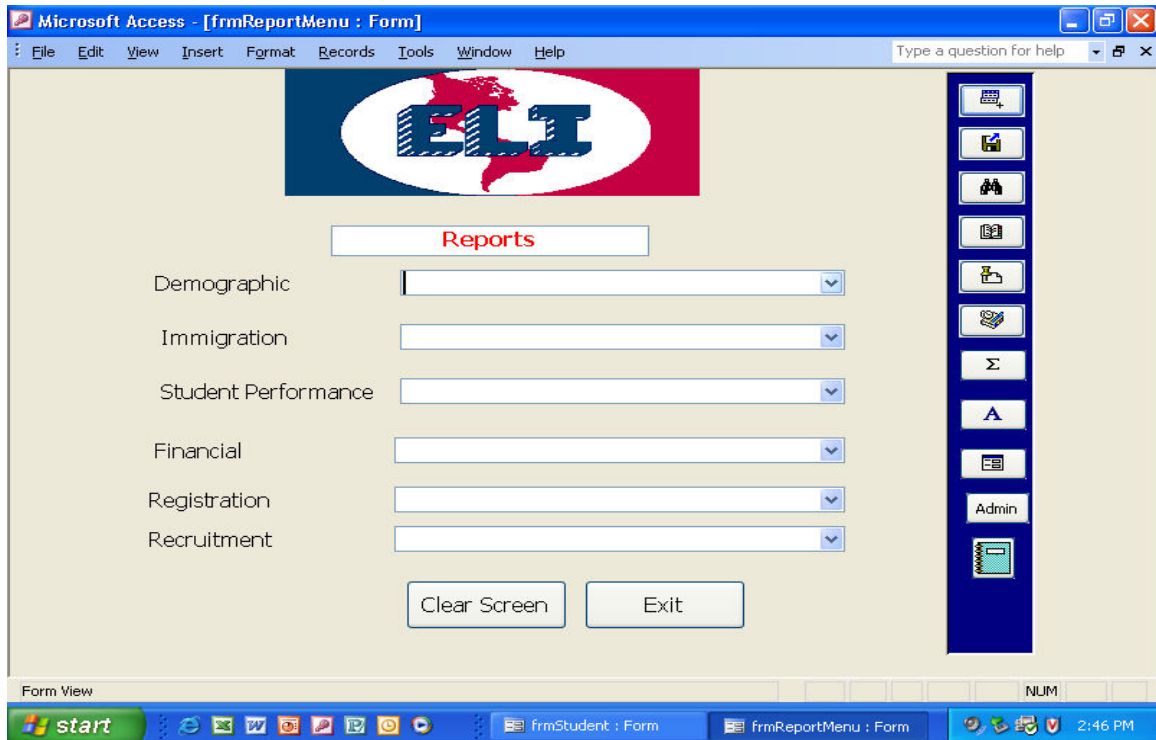
Last Name  First Name  Application Date  10/19/2006

Address 1  Phone  Fax   
 Address 2  E-Mail   
 City  State  Zip  Home Country  Nationality

Agency  Contact Info    
 Recruitment Agency  Contact Method    
 Recruitment Semester  Current Semester  Year  Contact    
 Recruitment Term  Current Term  Date of Contract  Date of 1st Letter

Form View NUM

### Reports





**Appendix 4**  
**Selected Project Deliverables**

**Phase I Report**

**Phase I Report - Table of Contents**

Company Information .....	<b>Error! Bookmark not defined.</b>
Present System Information: .....	<b>Error! Bookmark not defined.</b>
Proposed System Information .....	<b>Error! Bookmark not defined.</b>
Department Organizational Chart .....	<b>Error! Bookmark not defined.</b>
High-Level Context DFD .....	1
Technical Environment Diagram .....	<b>Error! Bookmark not defined.</b>
Journal.....	<b>Error! Bookmark not defined.</b>
Gantt Chart .....	<b>Error! Bookmark not defined.</b>
User Approval Form.....	<b>Error! Bookmark not defined.</b>
Resume .....	<b>Error! Bookmark not defined.</b>

**Phase II Report**

**Phase II Report - Table of Contents**

Allocation of Functions .....	<b>Error! Bookmark not defined.</b>
Manual Task Definition .....	<b>Error! Bookmark not defined.</b>
Equipment Identification .....	<b>Error! Bookmark not defined.</b>
Equipment Specification .....	<b>Error! Bookmark not defined.</b>
Database.....	<b>Error! Bookmark not defined.</b>
Computer Program Identification.....	<b>Error! Bookmark not defined.</b>
System Test Requirements Definition.....	<b>Error! Bookmark not defined.</b>
System Flowchart.....	<b>Error! Bookmark not defined.</b>
Data Flow Diagram.....	<b>Error! Bookmark not defined.</b>
System Façade .....	<b>Error! Bookmark not defined.</b>
Journal.....	<b>Error! Bookmark not defined.</b>
User Approval Form.....	18

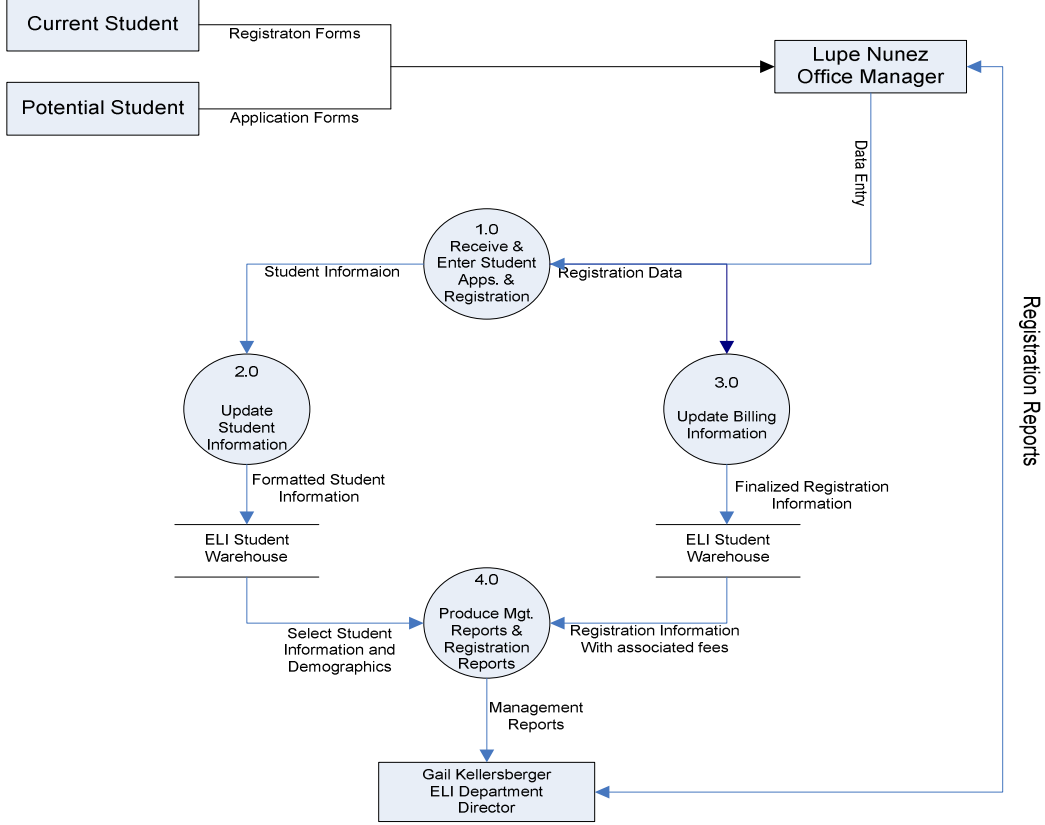
**Phase III Report****Phase III Report - Table of Contents**

Computer Equipment Acquisition, Installation, and Testing	<b>Error! Bookmark not defined.</b>
Computer Program Coding, Debugging, and Testing .....	<b>Error! Bookmark not defined.</b>
System Testing .....	<b>Error! Bookmark not defined.</b>
Changeover Plan Preparation .....	<b>Error! Bookmark not defined.</b>
System Installation Preparation.....	<b>Error! Bookmark not defined.</b>
System Installation Manual Preparation.....	<b>Error! Bookmark not defined.</b>
System Implementation Planning .....	<b>Error! Bookmark not defined.</b>
User Reference Manual Preparation .....	<b>Error! Bookmark not defined.</b>
User Training .....	<b>Error! Bookmark not defined.</b>
ELI Database Reports Created .....	<b>Error! Bookmark not defined.</b>
Written Codes .....	<b>Error! Bookmark not defined.</b>
User Interfaces .....	<b>Error! Bookmark not defined.</b>
Real Input/Output Data.....	<b>Error! Bookmark not defined.</b>
Gantt Chart .....	<b>Error! Bookmark not defined.</b>
PERT Chart.....	1
Journal.....	<b>Error! Bookmark not defined.</b>
User Approval Form.....	<b>Error! Bookmark not defined.</b>

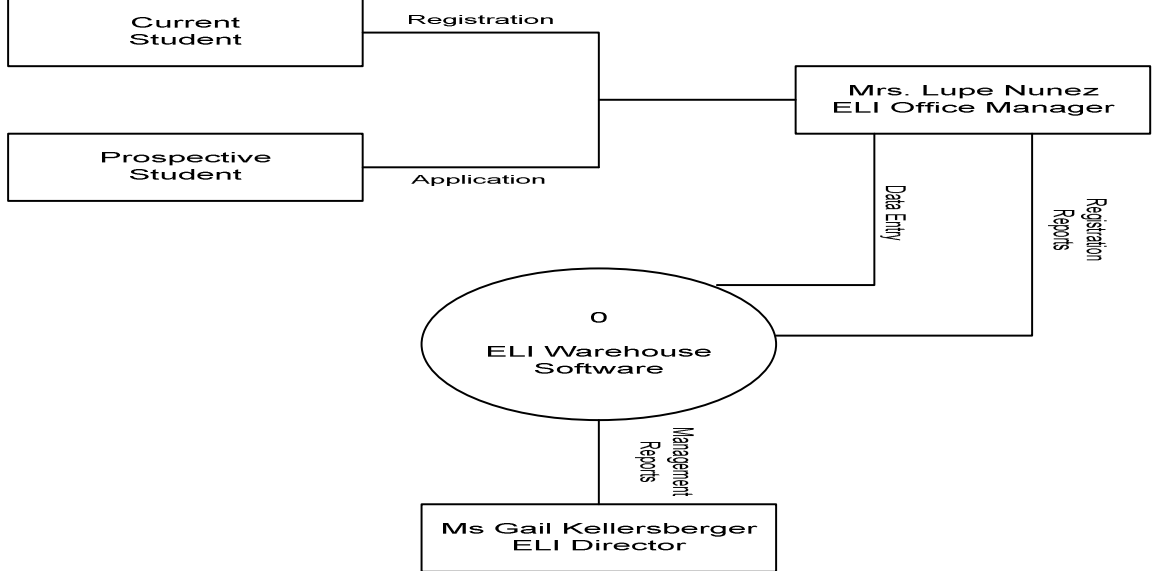
**Phase IV Report****Phase IV Report - Table of Contents**

System Changeover .....	<b>Error! Bookmark not defined.</b>
System Implementation .....	<b>Error! Bookmark not defined.</b>
System Testing .....	<b>Error! Bookmark not defined.</b>
User Training .....	<b>Error! Bookmark not defined.</b>
User Reference Manual .....	<b>Error! Bookmark not defined.</b>
System Installation and Backup Manual .....	<b>Error! Bookmark not defined.</b>
Technical Reference Manual.....	<b>Error! Bookmark not defined.</b>
Journal.....	<b>Error! Bookmark not defined.</b>
User Approval Form.....	<b>Error! Bookmark not defined.</b>

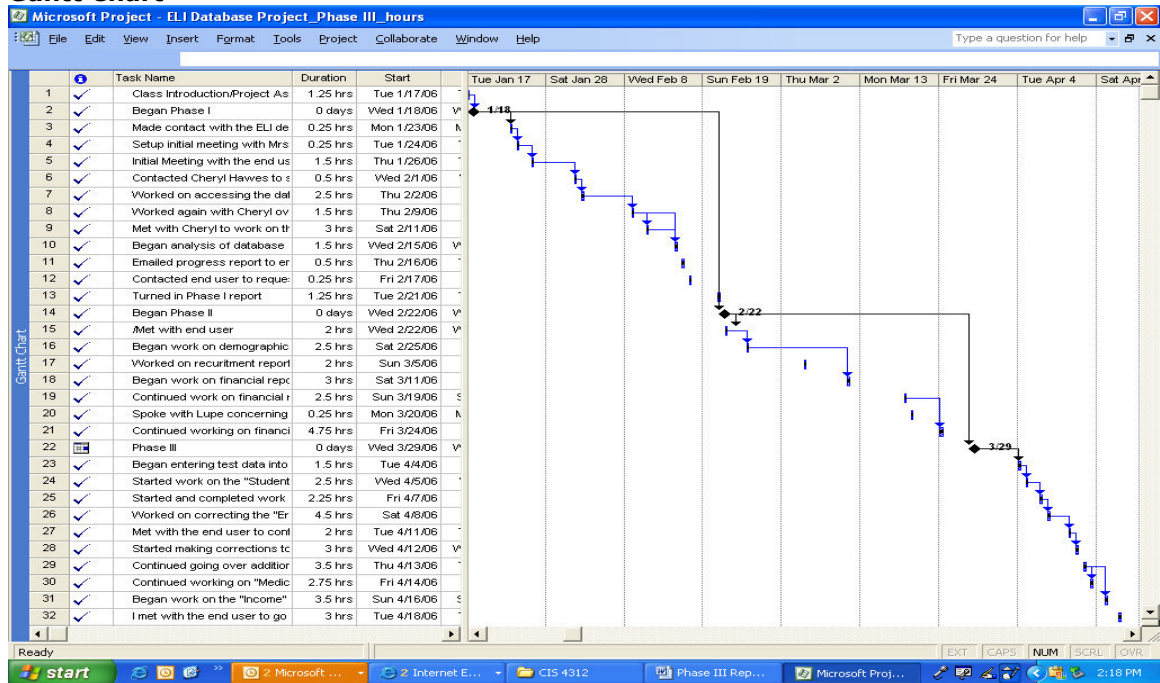
**High-Level Context Diagram**



**Data Flow Diagram**



### Gantt Chart



### PERT Chart

