Graduation Project Online Management System
ALHOSN University Case Study

ADEL KHELIFI, MURAD AL-RAJAB, SARA KAREM, LAMIA SUBHI
Software Engineering Department
ALHOSN University
Abu Dhabi
UNITED ARAB EMIRATES
a.khelifi@alhosnu.ae, m.alrajab@alhosnu.ae, skhmad01@students.alhosnu.ae,
lshussien01@students.alhosnu.ae

Abstract: - The graduation project is a cornerstone of the academic life, and professional career, of many students, and so whatever time is invested in such a project is truly worthwhile. Furthermore, students' extending themselves beyond their limitations in this endeavor is essential to enhancing the concepts they learned in previous courses. However, managing these projects poses real challenges for them, their supervisors, and the members of the graduation project committee. The purpose of this paper is to improve the ability of all these parties to manage the projects through the use of software that will allow students to organize their tasks and deadlines with minimum effort, enable supervisors to easily keep track of students and their progress, and ultimately permit graduation project committee members to monitor all transactions taking place to ensure that all is well.

Key-Words: - Graduation project, Online management, Software, Capstone project, Web application, Case study.

1 Introduction

Often, senior students struggle with their graduation projects throughout their college years, trying to manage their deliverables and their progress to meet deadlines. To ensure that they complete their projects, supervisors must become more involved, and what is needed to make this possible is an online application that would facilitate the whole process of creating and managing the project, which includes tasks, documentation, correspondence, and other related activities.

This paper describes all the system requirements of the Graduation Project Manager, and expands on the technical details of the application by presenting some of the designs. It also includes a detailed description of the system's high level design, along with implementation plans for that design.

The paper is organized as follows: Section 2 presents a literature review of graduation project online applications. Section 3 describes the methodology adopted to develop the online management system. Section 4 presents the requirements analysis for the application. The implementation steps for the application are given in section 5. Finally, section 6 provides our conclusion and suggestions for future work.

2 Literature Review

2.1 Definition of the Graduation

Graduation projects, also referred to as capstone projects, are papers that students must submit in order to graduate, and they are usually written in the students' senior year. The ideas for these projects may be provided by university faculty, or the students can develop their own. A graduation project must address a real-world problem, and students are expected to present logical and applicable solutions to that problem. The project can be viewed as the sum total of the student's education and skills gained throughout their years of schooling [1].

2.2. Significance of the Graduation project

Generally, prior to their senior year, students have not tackled projects of the magnitude of the graduation project. They are accustomed to short-term projects and assignments that do not involve deep analysis or the need to address real-world problems. Moreover, they typically aren't used to being limited by tight schedules, as previously they had worked with long-term deadlines and could start working on the assignments and projects quite close to those deadlines. So, why is the graduation project significant?

Perhaps for the first time, students are forced to break the habits they have acquired, and learn what it is like to be in an environment similar to a corporate one, and participate in group projects on a larger scale. As a result, they acquire new skills, and, if they perform well, they produce a result which has great potential and significance to businesses. So, it can be said that the graduation project can benefit students who work hard at

their projects by giving them foundational skills to help them transition to their professional careers [2, 3, 4].

3. Application Development Methodology

To the best of our knowledge, software for managing graduation projects is a relatively new idea. A great deal of research of an exploratory and descriptive nature is needed, and any research method chosen should reflect this.

3.1 Methodology adopted in this research

A combination of the case study and the personal interview is deemed appropriate for this research. The case study is an in-depth examination of a behavior, concept, or phenomenon. Complementary aspects of the case study are experiments and surveys. This research approach can be helpful in analyzing a real situation, and can serve as a strong basis for debate. However, since the information collected is about a real situation, the results may not be generic [5]. For this case study, the graduation projects undertaken in the faculty of engineering of ALHOSN University and at the College of Information Technology of Zayed University were examined. Although the results might not be entirely applicable to other faculties, the software will be flexible enough to be deployed in other environments.

The purpose of the personal interview is to encourage the interviewee to relate experiences and attitudes relevant to the research problem [6]. It is a flexible technique, in that the interviewer can probe into any interesting details that emerge during the interview, and focus on particular aspects. It should be noted that a spontaneous approach was deliberately allowed in the interview phase adopted in this study. Responses to certain questions can stimulate new awareness and interest in particular issues, which may then require additional probing [7].

In this research, a series of formal and informal interviews was conducted over a two-month period with students and supervisors from ALHOSN and Zayed Universities in the UAE. The interviews generally lasted one hour. Informal interviews were used to clarify and refine issues as they emerged [7]. Subsequently, the stakeholders in this research were identified, as were their impacts and responsibilities. Then, the system deployment environment was defined and the tools needed to develop the system were described.

3.2 Technology and tools used

In order to model the software requirements, the developers used the Unified Modeling Language (UML). UML is a language or notation specifying, visualizing, and documenting models of object oriented software systems. The following UML models were used:

Use Case Diagrams: show actors (people or other users of the system), use cases (the scenarios in which they use the system), and their relationships.

Class Diagrams: show classes and the relationships between them.

Sequence Diagrams: show objects and a sequence of method calls they make to other objects [8, 9].

Microsoft Office Project 2003 was used to manage the development phases of the software. It is a project management tool that enables project teams to be more productive. Microsoft Office Visio 2003 allowed the developers to visualize the requirements analysis and design of the software. RFFlow is software for presenting ideas with charts and diagrams, and is a tool for drawing and producing flowcharts. In this paper, RFFLOW was used to draw the entity relationship diagram, as this couldn't be done with Microsoft Visio.

Visual Paradigm for UML is a design tool that supports all UML diagrams and entity relationship diagrams [10]. In this paper, Visual Paradigm was used to draw the Database Schema model. In this research, the team also used Eclipse, which is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system [11]. Dreamweaver is Haverford's supported Web editor. It has features for both beginner and advanced Web page creators [12]. In this paper, Dreamweaver was used to create some parts of the Web page interfaces. Adobe Photoshop, or simply Photoshop, is a graphics editing program developed and published by Adobe Systems. It is commercial bitmap and image manipulation software [13]. In this research, Photoshop was used to draw the graphical components of the application's interface (such as the logo, the banner, etc.). The system, which is deployed on ALHOSN University servers, runs on a Windows 2003 server platform.

4. Requirements Analysis

The purpose of the analysis phase is to determine the needs of businesses and users. "The requirements for a system are the descriptions of the services provided by the system and its operational constraints" [14]. In other words, the functional requirements describe what the system must do.

4.1 Requirements Gathering Process

The following is a summary of the requirements that were gathered to implement the Graduation Project Manager software. They were collected using two methods.

First, some of the requirements were derived through an assessment of the general needs of ALHOSN University and Zayed University with regard to graduation projects. The assessment was based on the conclusions of a study of the graduation project process and how it is managed in the Faculty of Engineering at ALHOSN University and at the College of Information Technology at Zayed University.

Second, a set of questions was drawn up aimed at revealing the difficulties faced throughout the graduation process, and the needs of the professors to help the projects progress smoothly. Subsequently, meetings were held involving professors from different departments, such as Architectural Engineering, Software Engineering, and Urban Planning, of ALHOSN University and of the College of Information Technology at Zayed University. During the meetings, interviewees filled out a questionnaire and their responses were collected. Those responses expressed the needs and demands of the professors, and also highlighted the issues that had to be addressed.

An analysis of the responses revealed some of the software requirements. In addition, graduate students were surveyed to learn about the difficulties they had faced while working on their graduation projects.

4.2 System Requirements

The system requirements set out the system's functions, services, and operational constraints in detail, and they should be precisely expressed, and define exactly what is to be implemented. They are often classified as functional requirements and non-functional requirements.

4.2.1 Functional Requirements

"These are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. In some cases, the functional requirements state what the system shouldn't do" [14]. The use case diagrams below (Fig. 1, 2, and 3) illustrate the system's functional requirements:
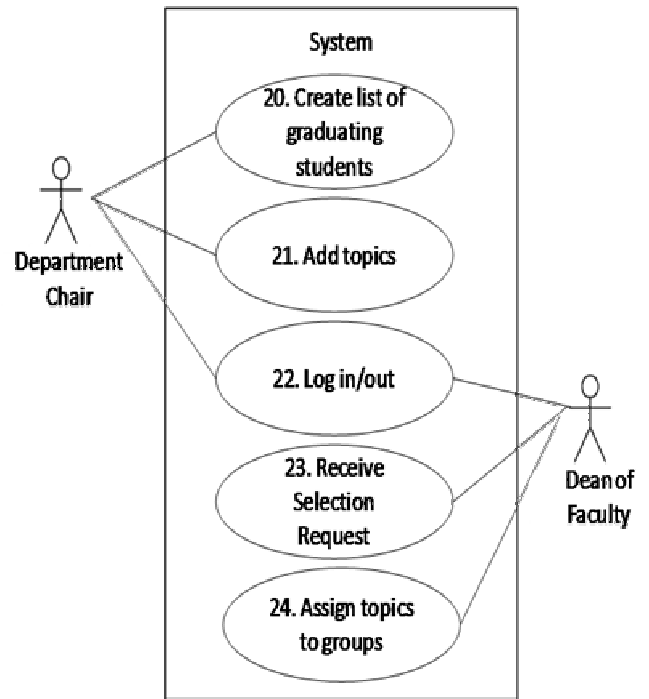

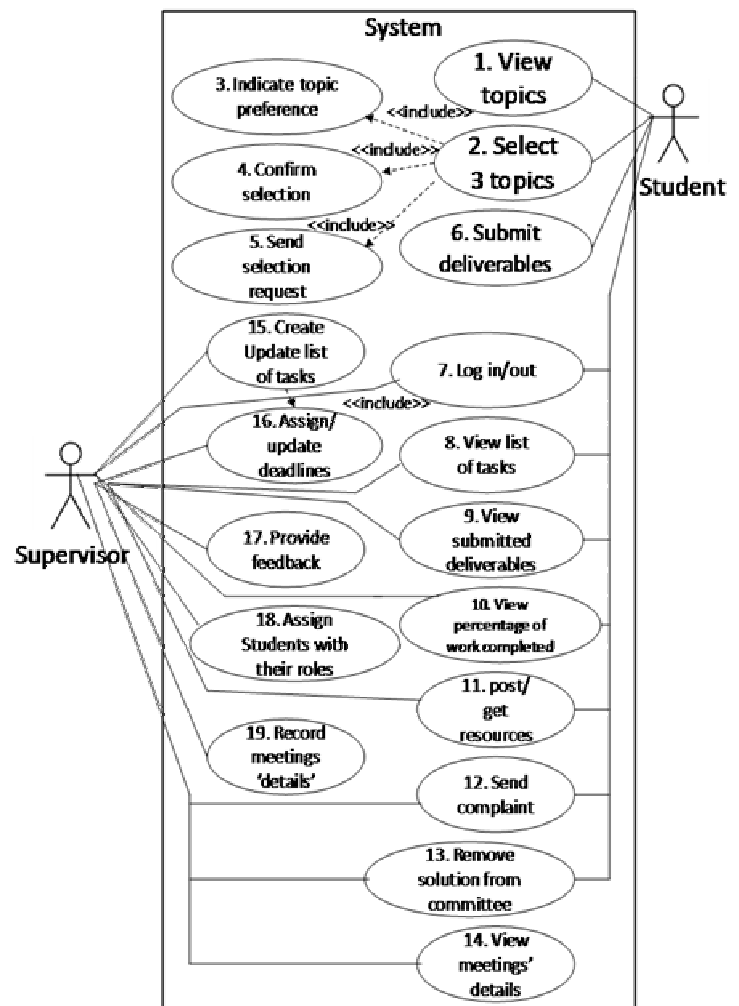Fig. 1: Use Case Diagram (Department chair / Dean of faculty)

Fig.2: Use Case Diagram of Graduation Project Manager, continued (Student / Supervisor)
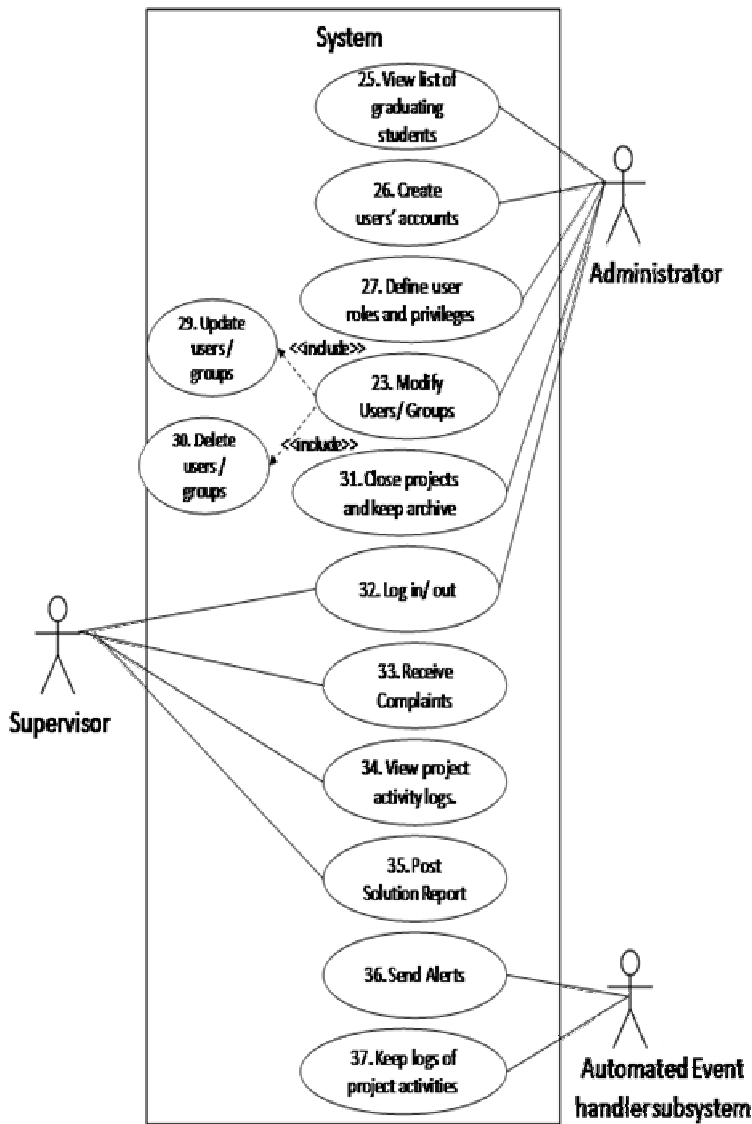
Fig. 3: Use Case diagram, continued (Administrator/ Head of Graduation Committee / Handler subsystem)

4.2.2 Non-Functional Requirements

"These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process and standards. Non-functional requirements often apply to the system as a whole. They do not usually apply to individual system features and services" [14].

4.2.3 Specification of Non-Functional requirements

The following non-functional requirements are applied for the Graduation Project Manager:

- **Usability requirements:** concerned with the ease of use of a system. Since the Graduation Project Manager will be used by students with different educational backgrounds and levels of computer knowledge, the system needs to provide users with a simple and easy interface to allow for this variation.

- **Privacy and security requirements:** concerned with the need for data confidentiality and integrity. The Graduation Project Manager is required to keep all the data, transactions, and communications within each group away from the public and hidden from other groups, and ensure that only the direct supervisor of each group has access to this information. The system also needs to ensure that only authorized users can gain access to group pages and prohibit any unauthorized access (system needs to check for authenticity).

- **Reliability requirements:** constraints on the run-time behavior of the system. They usually relate to the amount of time the system has been running without failures.

  Students will use the Graduation Project Manager very frequently to communicate with their supervisor and with each other within their groups. They also expect the system to be available serve their needs at any time.

  This means that there is a demand for the system to be continuously available, and have a minimum failure rate, as that might affect the students' productivity and progress.

- **Portability requirements:** a requirement which states that the system being developed should be capable of running on different platforms. A Graduation Project Manager must be portable, because it is an online application that will be used by many users with different machines/browsers. This can be achieved through constant testing for portability of the application relative to the most common browsers, such as Firefox, Internet Explorer, and Safari.

- **Flexibility:** "The ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed" [15]. This means that flexibility allows the system to be tailored according to the needs of different users, rather than being specific to a particular user group. Faculty members from different departments might be interested in using the system. So the system needs to be flexible to allow new features to be readily added [16].

5. Implementation

Prior to embarking on the implementation phase, the developers accorded high significance to the design phase, which is aimed at modeling the system requirements for the users in a simple way so that they

can easily understand the system's functions. The Entity relationship and the UML diagrams (Use case and Sequence diagrams) are examples of modeling standards which are used to refine the requirements and present them in a readable style and with more detail than those described at the requirements analysis phase. The designs are created to aid the developers in the transition from the requirements analysis phase to the implementation phase, while keeping the user involved [17].

The goal of the implementation phase is to implement a system correctly on a particular set or range of computers using particular tools and programming languages. This stage is primarily environmental, and works with the realities of the particular machines, systems, language compilers, tools, developers, and clients necessary to translate a design into working code. Just as the design phase may include some analysis effort approached from a computational standpoint, the implementation phase always includes design effort. Implementation-level design is a reconciliation activity, where, in principle, executable models, implementation languages and tools, performance requirements, and delivery schedules must ultimately be combined, while at the same time maintaining correctness, reliability, extensibility, maintainability, and related criteria.

Software developers take the design documents, plans, and development tools, such as editors, compilers, and debuggers, and then start writing software. This is usually the longest phase in the product life cycle. So far, in this implementation, version 1.0 of the system has been completed. This version incorporates the core functions of the Graduation Project Manager, and is the result of the first iteration in the software development cycle followed in this paper. However, the complete system design has been created, which means that all the requirements and functions have been identified, and not just the designs for the current release. Therefore, the later iterations in the software's life can be commenced as soon as a release is completed [18].

5.1 Graduation Project Manager Architecture

An architectural design is a plan for how the system will be distributed across the computers (network environment), and what hardware and software will be used for each computer. There are three principle application architectures in use today: server-based architectures, client-based architectures, and client-server architectures. The client-server architecture attempts to balance the processing or workloads between the client and the server. This is a popular option, because it involves lower overhead and easier maintenance. Web-based systems usually follow this architecture, with the Web browser (the client) performing presentation and only minimal application logic using programming

languages, such as JavaScript, while the server handles the application logic, the data access logic, and storage. The benefits of a client-server architecture are threefold. First, it allows scalability, which means that it is easy to increase or decrease the server's storage and processing capabilities. If one server becomes overloaded, another server is simply added, so that many servers can be used to perform the application logic, data access logic, or data storage. Second, it can support many different types of clients and servers, which means that it is possible to connect computers that use different operating systems. Third, for a thin client server architecture (one containing a small portion of the application logic) that uses Internet standards, it is a simple matter to clearly separate the presentation logic, application logic, and data access logic, and design each to be somewhat independent. This means that the interface can be changed without affecting the application logic, and vice versa. For these reasons, the concept of the client-server architecture was implemented as the Graduation Project Manager's architecture, as shown in Figure 4 below.
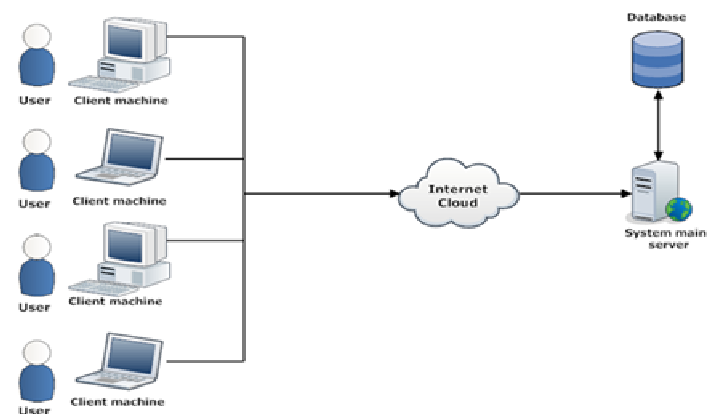

Fig. 4: CPM Main Architecture

5.2 Model View Controller (MVC)

In this Web application, the MVC architecture is used for subdividing the system logic, where JSP is used as the view, servlets as the controller, and JavaBeans as the model [19].

The MVC, as shown in Figure 5, is a software architectural pattern used in software engineering, commonly with Java-based Web systems. It is the product of a desire to separate the code that creates and manipulates the data from the code that presents the data.

Model: the domain-specific representation of data upon which the application operates. The MVC doesn't specifically mention the data access layer, because it is understood to be underneath the model, or encapsulated by it.

View: what renders the model in a form suitable for interaction, typically a User Interface element. Multiple

views can exist for a single model for different purposes. Controller: the component that receives input and initiates a response by making calls on the model object. The goal of the MVC model is to reduce the complexity of the architectural design and increase the flexibility and maintainability of the code by decoupling models and views. It is also responsible for invoking the business logic and data access code.
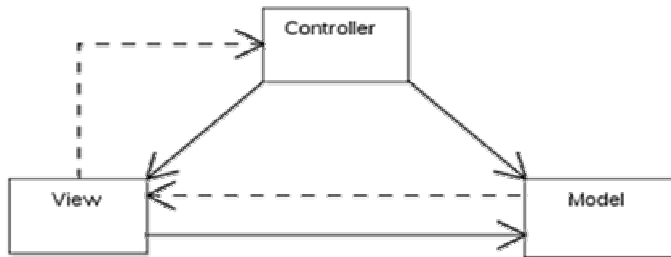

Fig. 5: MVC Architectural Pattern

5.3 Java Technology and JSP Selection

Java technology is used because it is highly portable, as Java-based applications can be run on any machine regardless of the underlying operating system. Java Web technology (JSP and Java servlets) is very compatible with different browsers, as opposed to asp.net, for instance, which is usually error-prone on browsers other than Internet Explorer. Sun developed JSP technology using the Java Community Process (Open Source). The JSP application programming interface (API) has undoubtedly benefited from the input of this extensive community, and will likely continue to do so.
Java Web technology is used in many of today's popular Web applications, such as PayPal, eBay, Yahoo, and YouTube [19].

5.4 Interface design

Interface design is the process of defining how the system will interact with the external entities (system users or other systems). It describes the layout of the pages and the flow of events, and is also concerned with where and how data are represented on the pages.
However, in a User Interface Design, the needs, experience, and capabilities of the system users must be taken into account. In addition, the designers should be aware of users' physical and mental limitations (e.g. limited short-term memory) and should recognize that people make mistakes [14]. Figures (6-7) below illustrate some of the system's graphical interfaces.

Fig. 6: Heading of the Capstone Committee's Home Page

Fig. 7: Administrator's Create User page

6. Conclusion and Future Work

The concept on which the Graduation Project Manager is built has noticeable potential, and the opportunity to grow and improve. Below are examples of future improvements that can be made to the Graduation Project Manager software:

- Have a tips section for students, where professors can post hints and tips over time (these could be accessed by all students, and not just a specific group);

- Allow students to exchange deliverables before they submit (this could provide a backup facility for all their files, and provide accessibility to them as required);
- Allow supervisors to create one list of tasks, which could be used as a template for many different projects;
- Implement more authentication controls, for example by hashing the password before sending it off to the server, or by applying SSL to enhance the security level of the system;
- Have a news bar, where the Head of the Graduation Committee could, for example, post updates regarding new regulations, or changes to the project timeline, etc.;
- Introduce new requirements through conducting more meetings with stakeholders, which we believe would be possible;
- Provide students with a calendar for memos, reminders, appointments, etc.;
- Allow the Head of Graduation Committee to post progress reports for students and supervisors to fill out and send back over the system;
- Implement the automated event handler subsystem.

In summary, the graduation project is a cornerstone of students' academic learning, and of their future professional career as well. This paper brought a solution to improve the organization of graduation project tasks, for everyone involved, through software that allows students to manage their tasks and deadlines with minimum effort, enables supervisors to easily keep track of students' work and progress, and ultimately permits graduation project committee members to monitor all transactions taking place to ensure that all is well.

In order to analyze the requirements of this research, a combination of the case study and the personal interview was deemed an appropriate approach to follow in this paper. In addition, a series of formal and informal interviews was conducted over a two-month period with students and supervisors from ALHOSN University and Zayed University. The developed software enables students to select topics, obtain resources from their supervisor, upload deliverables to their supervisor, fill in a calendar with a to-do list, and view their project plan and the percentage of work completed. Moreover, supervisors can post resources for their students, create a list of tasks along with their deadlines, obtain deliverables, provide feedback by posting remarks or uploading updated documents, and view the percentage of work completed. The web-based application is implemented using JSP and Java Servlets, as part of the J2EE platform to dynamically generate Web pages. JSP and Java Servlets are run on an Apache Tomcat server. The system utilizes a relational Oracle Database.

In conclusion, we believe that this Graduation Project Manager software can significantly enhance graduation project management for all the parties concerned.

References:

[1] EssayTown. (2009). *Term Paper Help*. Retrieved Dec. 4, 2009, from http://www.essaytown.com: http://www.essaytown.com/writing/capstone-projects

[2] Spolsky, J. (2009, 10 26). Retrieved Dec. 7, 2009, from http://www.joelonsoftware.com: http://www.joelonsoftware.com/items/2009/10/26.html

[3] keeai. (2009, 7 23). Retrieved Dec. 5, 2009, from http://keeai.wordpress.com: http://keeai.wordpress.com/2009/07/23/the-meaning-of-a-capstone-project/

[4] Duke University School of Law. (2009). *Ad Hoc Seminars, Capstone Projects & Independent Studies*. Retrieved Dec. 4, 2009, from Duke Law: http://www.law.duke.edu/curriculum/capstone/

[5] Fitzgerald, B. &Kenny, T. (2003). *http://www.netproject.com/online.html*. Retrieved from http://www.netproject.com: http://www.netproject.com/docs/Beaumont.pdf

[6] Walker, R. (1988). *Applied Qualitative Research.* Hampshire: Gower.

[7] Eisenhardt, K. (1989). Building theory from case study research. Academy of Management Review 14, 4, pp. 532-550.

[8] Wikipedia. (2009). Unified Modeling Language. Retrieved Dec. 9, 2009, from Wikipedia: http://en.wikipedia.org/wiki/Unified_Modeling_Language

[9] Webmaster, K. (2009). *Welcome*. Retrieved Dec. 8, 2009, from KDE Documentation: http://docs.kde.org/stable/en/kdesdk/umbrello/uml-basics.html

[10] Visual Paradigm. (2010, 4 18). *http://www.visual-paradigm.com/product/vpuml/*. Retrieved from http://www.visual-paradigm.com: http://www.visual-paradigm.com/product/vpuml/

[11] Wikipedia. (2010, 4 18). *http://en.wikipedia.org/wiki/Eclipse_(software)*. Retrieved from http://en.wikipedia.org: http://en.wikipedia.org/wiki/Eclipse_(software)

[12] Center Haverford, C. a. (2010, 4 19). *http://www.haverford.edu/acc/docs/software/dreamweaver/welcome.htm*. Retrieved from http://www.haverford.edu: http://www.haverford.edu/acc/docs/software/dreamweaver/welcome.htm

[13]   Wikipedia.   (2010,   4   19). http://en.wikipedia.org/wiki/Adobe_Photoshop. Retrieved   from   http://en.wikipedia.org: http://en.wikipedia.org/wiki/Adobe_Photoshop

[14] Sommerville, I. (2007). Software Engineering (8th Edition ed.). London, United Kingdom: Person Education.

[15] IEEE. (1999). IEEE Standard Glossary of Software Engineering Terminology 610.12-1990. IEEE Standards Software Engineering, 1999 Edition, Volume One: Customer and Terminology Standards.

[16] Eden, A. H. & Mens, T. M. (June 2006). Measuring Software Flexibility. IEEE Software, vol. 153, no. 3, 113-126.

[17] Dennis, Alan, Wixom, Barbara Haley, Roth, Roberta M. (2006). Systems Analysis & Design (3rd Edition ed.). Hoboken, NJ, USA: John Wiley & Sons, Inc.

[18]   Essays   Example.   (n.d.). *http://www.exampleessays.com/viewpaper/8491.html* Retrieved   4   25,   2010,   from http://www.exampleessays.com: http://www.exampleessays.com/viewpaper/8491.htm l

[19] Hall, Marty & Brown, Larry. *Core Servlets and Javaserver Pages: Core Technologies, Vol. 1.* (2nd, Ed.) Prentice Hall and Sun Microsystems Press.