Summer 2007

Educating Educators On Mastery Learning And Spiral Learning

Gao Lou Yang
Regis University

Regis University
College for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

Design of a Web-Based Project Tracking System

Robert F. Tschiemer

Regis University

School of Professional Studies

Master of Science in Computer Information Technology

Regis University

School for Professional Studies Graduate Programs

MSCIT Program

Graduate Programs Final Project/Thesis

Certification of Authorship of Professional Project Work

Print Student's Name **Robert F. Tschiemer**

Telephone **720-535-1063** Email **rtschiem@hotmail.com**

Date of Submission **23 June 2007** Degree Program **MSCIT**

Adviser/Faculty Name **Dr. Douglas Hart**

Certification of Authorship:

I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for the purpose of partial fulfillment of requirements for the Master of Science in Computer Information Technology Degree Program.

_____ 08/17/07

Student Signature *Date*

Regis University

School for Professional Studies Graduate Programs

MSCIT Program

Graduate Programs Final Project/Thesis

Authorization to Publish Student Work

I, **Robert F. Tschiemer**, the undersigned student, in the Master of Science in Computer Information Technology Degree Program hereby authorize Regis University to publish through a Regis University owned and maintained web server, the document described below ("Work"). I acknowledge and understand that the Work will be freely available to all users of the World Wide Web under the condition that it can only be used for legitimate, non-commercial academic research and study. I understand that this restriction on use will be contained in a header note on the Regis University website but will not be otherwise policed or enforced. I understand and acknowledge that under the Family Educational Rights and Privacy Act I have no obligation to release the Work to any party for any purpose. I am authorizing the release of the Work as a voluntary act without any coercion or restraint. On behalf of myself, my heirs, personal representatives and beneficiaries, I do hereby release Regis University, its officers, employees and agents from any claims, causes, causes of action, law suits, claims for injury, defamation, or other damage to me or my family arising out of or resulting from good faith compliance with the provisions of this authorization. This authorization shall be valid and in force until rescinded in writing.

Print Title of Document(s) to be published: _____

Design of a Web-Based Project Tracking System

_____ 08/17/07

Student Signature *Date*

Check if applicable:

____The Work contains private or proprietary information of the following parties and their attached permission is required as well: _____

Name of Organization and/or Authorized Personnel

Regis University

School for Professional Studies Graduate Programs

MSCIT Program

Graduate Programs Final Project/Thesis

Adviser/Professional Project Faculty Approval Form

Student's Name: Robert F. Tschiemer Program MSCIT

PLEASE PRINT

Professional Project Title: Design of a Web-Based Project Tracking System

PLEASE PRINT

Adviser Name: Dr. Douglas Hart

PLEASE PRINT

Project Faculty Name: Joseph Gerber

PLEASE PRINT

Adviser/Faculty Declaration:

I have advised this student through the Professional Project Process and approve of the final document as acceptable to be submitted as fulfillment of partial completion of requirements for the MSCIT Degree Program.

Project Adviser Approval:

_____ 08/17/07

Original Signature *Date*

Degree Chair Approval if:

The student has received project approval from Faculty and has followed due process in the completion of the project and subsequent documentation.

_____ _____

Original Degree Chair/Designee Signature *Date*

Abstract

This paper explores the design of a prototype Web-based database application that will support a professional project tracking system for students and Advisers. Specifically, this paper combines a relational database and web-based applications that are integrated into NET framework and PHP code technologies. The initial database-driven web application is designed for 200-300 student usage and is scalable to a 1500 student project usage. The methodology that governs the application design was determined by review of waterfall methodology approach. The chosen development methodology provides the application design and prototype construction. The actual database design and web page implementation will conclude the project.

Revision History

Table 1. Revision History

REVISION	DATE	DESCRIPTION
1-0	04/20/07	Rough draft submitted to Adviser
1-1	05/12/07	Draft submitted to instructor for initial critique
1-2	05/26/07	Draft submitted to instructor for updated draft changes
1-3	06/11/07	Draft submitted to proofreader
1-4	06/24/07	Draft submitted to instructor and Adviser for approval
1-5	08/17/07	Final Draft submitted to Adviser for approval

Table of Contents

List of Tables

List of Figures

Chapter One: Introduction and Executive Summary

1.1 Statement of the Problem

A web-based application for students and advisers for tracking professional projects is not currently available for the Software Engineering Department at Regis University. This researcher focused on the merger of database design technologies with web-based application technologies to produce an interactive student project tracking website.

Specifically, this design encompassed an Internet web-based application, which tracks student professional project development from the initial thesis through final project completion. The finished application provides both adviser and student a web based database to store, retrieve and manage project information.

1.2 Project goals

The goal of my project was to design a web-based database application that allows for both graduate students and their advisers to track a students' progress during the writing for their professional papers. Currently, computer information technology students do not have one place to manage their entire professional project papers. My application gives the student a way to integrate all of the project papers into one database, along with a website view that allows for quick access to all important papers and additional project works. In addition, advisers have a resource to check on student progress, and view completed project work. This prototype design presents a significant process challenge in developing a physical database that works with a web-based application because it incorporates both a database and a web-based application for adding and recovering project information. Further, research was done on this

intellection, but nothing quit fit the solution as building the design from scratch. However, this did not leave out using commercial software to aid in the design of the database and web applications because the main goal always remained building a final project that gave the user the most useful design. Shown below is the high level project application model. It is important because this model builds on the entire position of the relational database design.

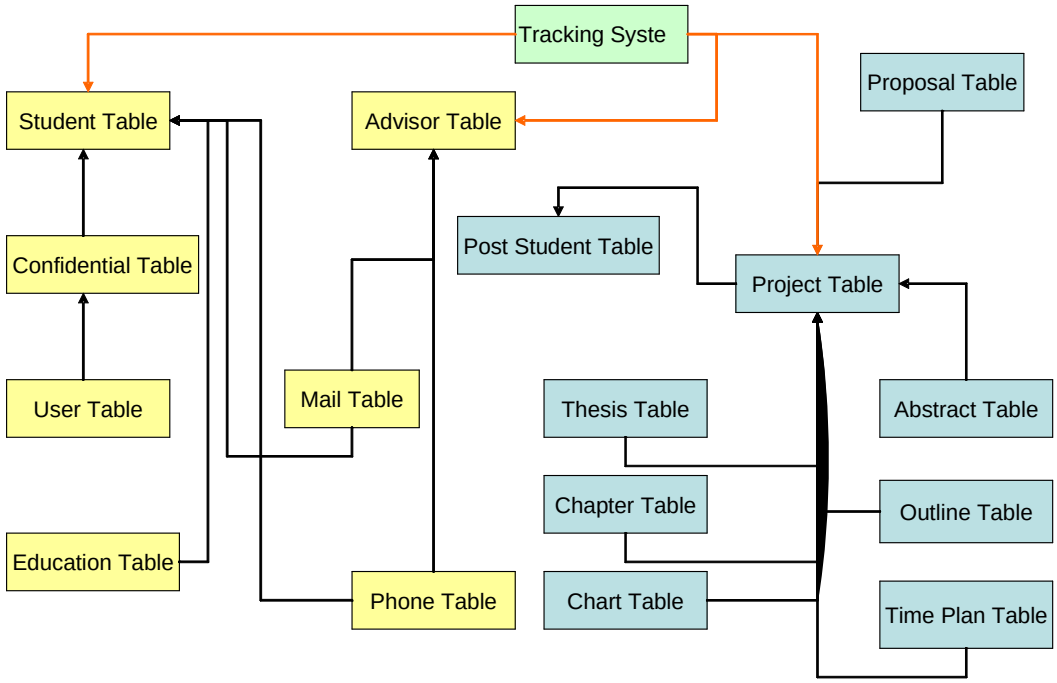

Figure 1. Logical Database Design model

The database model is divided into two functional groups. Group one stores information about the student and adviser and user login data while group two stores information about the student's project and student project completion data. The tracking system table combines both groups by acting as the application's linking table. An in-

depth relational project model design is discussed in chapter 3.

1.2 Why Develop the Project

At the onset of this project study, there was no application that gave students an application software tool that could track professional projects. Now this researchers' project has produced a prototype application that gives the student a logical way to track their professional projects from writing the abstract through the final project approve phase.

Design of this prototype application produced a web-based integrated database to store and retrieve student professional project papers. This design provides a physical place to colligate important aspects of tracking the student's professional project input such as Thesis Statement, Abstract, Proposal, Project Reports and Adviser information.

1.3 Barriers and Issues

The current project barriers consist of the following: (a) No commercial off-the-shelf (COTS) applications exist, which can be adapted to provide the same function as a newly developed program, (b) Initial application design interface may not be what the user needs in the final design, requiring the application to go through many iterations before it is ready for testing, (c) First step design work such as designing the database, database normalization, and database testing are all major barriers to application building, (d) Prototyping requires rapid develop, which sometimes leads to missing detailed underlying design steps such as forms design for the website application view.

1.4 Questions to Be Answered

The main questions that need answering were the following: First, the most important question was how the application would be used. Next, knowing what type of

data would be stored and retrieved from the database. The third most important question was asking what upkeep would be necessary to maintain the database and web-based application.

1.5 Limits and Scope of the Project

The overall scope of the project design provided a complete web-based database with easy access to a structured tracking system for all students and their advisers during the student's development for their professional project. However, the project is limited in design and support capacity because the prototype design supports only 200-300 students.

1.6 Project Organization

The project was divided into six development steps. The first step was the discovery phase. In this phase all the initial information about the project was collected and an outline of requirements was developed. The next step was the specification phase. This is the phase where deliverables were determined and any possible constraints on the project development are discussed. The next step was the design phase. This was the database modeling phase and discussed the modeling tools used to create the database. The fourth step was the implementation phase and consisted of the program coding of the overall design and checking for design faults. The fifth step was the integration phase or evaluation of the design. The demonstration phase was the final step in the project, which completed the application design.

1.7 Definitions of Terms

ASP.NET – Microsoft's active server pages. Used to create interactive HTML pages

CASE (Computer-Aided System Engineering) – design tools to assist in data analysis, conceptual data model and logical data model design, and to enable prototyping of applications and code generation

COMMERICAL OFF-THE-SHELF (COTS) – Pre-designed software of turnkey use

COMMON LANGURAGE RUNTIME (CLR) – Programming language built at runtime

RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS) – Term used to describe an entire suit of programs that both manage and communicate with relational database systems

DDL – Database script created when a new or updated database is generated

ENTERPRISE ARCHITECT by Sparx Systems – UML modeling tool

ENTITY RELATIONSHIP DIAGRAMS (ER) – Used to show relationships between database modules and their relationships

HYPERTEXT MARKUP LANGUAGE – HTML elements used to create form elements for web pages

HYPERTEXT PREPROCESSOR (PHP) – Web-coding language that works well with HTML and used to build websites

Integrated Development Environment (IDE) – Software development suites such as Microsoft Visual Studio

MSSQL – Microsoft SQL development tool

MySQL – Open source database development tool

.NET FRAMEWORK – Technology for supporting Windows development environment

Object–Oriented Programming (OOP) – Programming approach that uses modules of code and is more versatile than monolithic applications

ORM – Object-Role Modeling – Used with Microsoft Visio to build detailed ORM models

PHP RUNNER by XLine Software – PHP web page interface development tool and code generator

SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) – Model used in the development of software application design

STRUCTURED QUERY LANGUAGE (SQL or SEQUEL) – Computer language designed for the retrieval and management of data in relational database management systems

SQLYOG ENTERPRISE by Webyog Software – MySQL database modeling tool

UNIFIED MODELING LANGUAGE (UML) – Used to design UML class diagrams

VISIO for ENTERPRISE ARCHITECT – Microsoft development tool used to build databases, ORM models, and Entity Relationship Diagrams

VISUAL STUDIO.NET ENTERPRISE ARCHITECT – Microsoft modeling tool used as an all-in-one database, modeling, and data analysis suite

1.8 Summary

The aim of the project was to construct a web-based student project tracking system to support both students and advisers with writing thesis project papers. The project is not a commercially ready software program. However, the project is a sophisticated relational database web-based prototype application. The methodology used in the project was the waterfall method combined with the rapid prototyping method.

Chapter Two: Review of the Literature

2.1 Overview

This research focused on this development idea: a back-end database merged with a Web-based application that brings together both these technologies and processes. To accomplish this task, this researcher first looked to Microsoft's Visual Studio Enterprise Architect (Griffiths, Flandres, & Sell, 2003) After working with Visual Studio, it was determined that additional applications were needed to build a proper database without creating major problems in its design.

The primary program that bridged this development gap was Microsoft's Visio for Enterprise Architects (Halpin, 2003). These two programs set the initial start for building the project application. Creating interactive websites using PHP, MySQL, and Web Services create a powerful combination. However, in comparing PHP with ASP.NET (Kauffman, Spencer, & Willis, 2003) it was found that the two are not the same (Rosebrock, 2003). That is because unlike PHP, ASP does not always work well on different platforms and ASP is more proprietary in its design (Mercer et al., 2004). Because PHP web page and MySQL database are both open source application development tools, they were used in this project application development.

2.2 Relevant Literature and Research

Database modeling comes in many favors - from beginning to advance design. When building a database, objectives are the most important part of any project (Powell, 2006). What all this leads up to in the modeling world is information modeling (Halpin, 2003). The information modeling used in this project was Object-Role Modeling or ORM (Halpin, 2003). As Gordon Everest (Everest, 2001) professor of MIS and DBMS

at the University of Minnesota said “Data modeling is the foundation of information systems development. If you don’t get the database design right, then the systems you build will be like a house of cards, collapsing under the weight of inevitable future forces for revision, enhancement, integration, and quality improvement”. Other modeling tools, used in this project, are Entity-Relationship (ER) diagrams and Unified Modeling Language (UML) class diagrams.

2.3 Known and Unknown Factors and Their Effects

Known and unknown factors of this project application center on the methodology used and the derived design chosen from this methodology. For example, in the initial design research, the first several methodologies where studied for use with this project’s design. They include the waterfall model and the rapid prototype model. However, using just one single model would not cover all of the unique design features necessary for a solid application design. So, a combination of the waterfall and rapid prototype model was used to govern the final design concept. With this design methodology the unknown factors are what the final application design looks like and how the application is used? This is because of the rapid part of the design process – having the users interact with the current design and then get their feedback about how the application is fulfilling their expectations. The know factors are already covered by the initial design requirements and should product expected results. The combination of the two methods allows for modifications to be made while development is ongoing. So, what are the possible effects of combining both models together? The answer was found with the application design. As the application design advanced through the different stages – design, implementation, and testing this researcher found that if only one method

was used in this process such as the waterfall model, the user may not have gotten what their really needed, however, integrating both models – waterfall and rapid prototyping helped to determine the users real needs, and provide assistance with the application design by acting as a front end for guiding development and acting as a useful side effect.

2.4 Project Contribution to the Field of Study

The purpose of this study is to work with the various database and web service tools in the design of this researcher's project. In addition to using design tools, analysis was done on which database modeling technique was the best and how the starting database model was developed. Also, new design modeling techniques were looked at such as hybrid application development – using web technologies that allow users to interact with an application from their web browser from any location. All of these combined design processes where used in this project application design.

Chapter Three: Methodology

3.1 Research Methods Used

Different types of methodologies were investigated for use in this project. The waterfall and the rapid prototyping model were the methodologies picked for this project (Schach, 2002). The following figure depicts an example of the waterfall model:

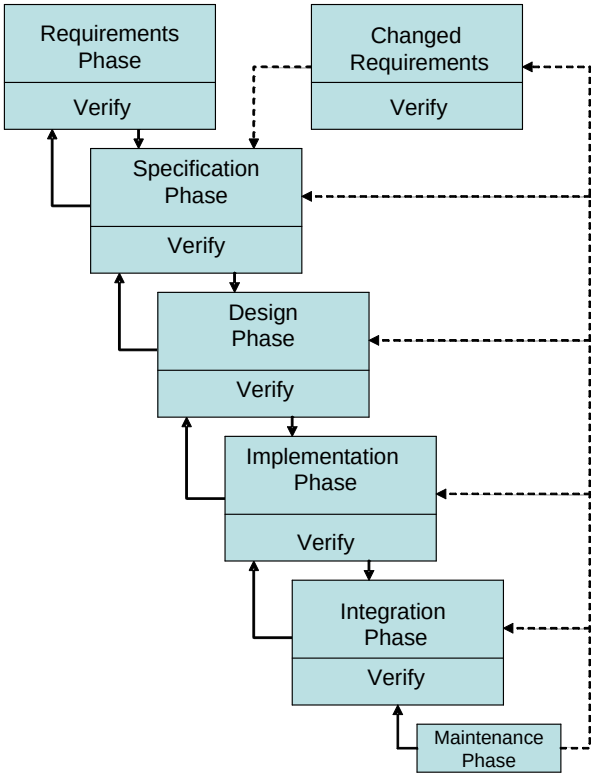

Figure 2. Tracking System Waterfall model

The waterfall model by itself is good for initial application design, however because the waterfall model can lead to conflicts between the original design specifications and what the user needs can limit the usefulness of the waterfall model. By combining both the waterfall model with the rapid prototype model the use gets a better

application design that closely follows what the user really needs.

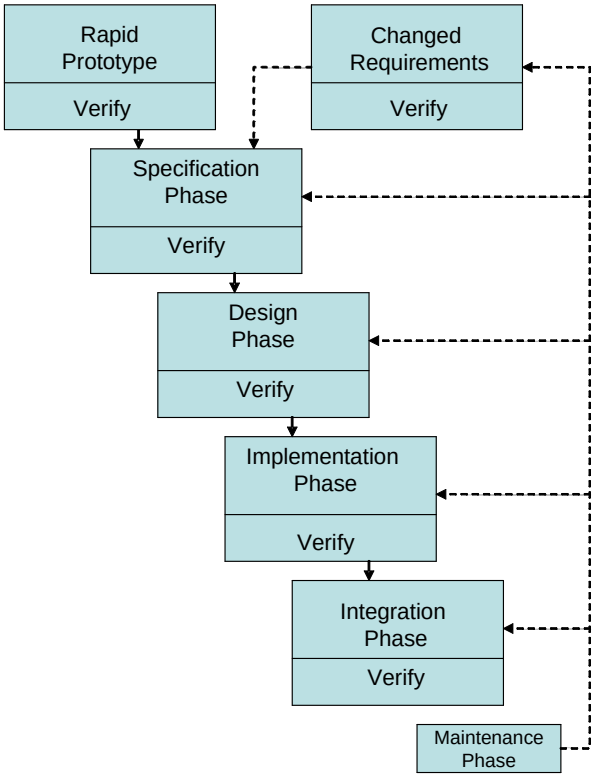

Figure 3. Tracking System Rapid Prototyping model

This combined methodology allows for the best design approach by first using the rapid prototyping approach to build the prototype to determine the user's real needs and then use this prototype to act as an input into the waterfall model. This approach minimizes development risks while adding user requirements to the overall design process.

3.2 Life-cycle Model

The following describes the Software Development Life Cycle (SDLC) used in this project:

This section contains the business rules for the application. This is the initial investigation or requirements phase, which includes gathering initial information about the selected project, developing an outline for project requirements, and determining project metrics such as: What is the scope of the project? Will the database met the needs of the user? The outcome from these questions is the formulation of the project development plan.

Specification phase: Determines the scope of the project on the conceptual level. The project developer concentrates on the facts that govern the application design, and use these facts to develop key deliverables.

Design phase: Building of the logical database model. This is also known as the design and build physical model phase. In this phase, the physical database is built for a specific RDBMS that answers questions such as: What type of information is needed to build a physical database that meets the needs of the user's requirements? All tables, rows, columns, and relationships are developed during the phase.

Integration phase: This was the evaluating phase. Here the application was checked for proper functionality while meeting the initial project specifications

Maintenance phase: Ongoing phase of the application life cycle. Requires upgrades and additional functions added to the project application as required by the user.

The above phases are rolled up into the following database life cycle model:

3.3 Resource Requirements

The project resource requirements included the following: (a) Build a well-structured relational database model, (b) Structure data integrity to ensure that no data is lost, and data is only destroyed when it should be, (c) Support plan queries and Ad hoc queries, (d) Support the business rules and objectives of the users, (e) Design tables in the database to ensure that then represent only a single subject, (f) Provide performance updates for any required changes to the database, (g) Provide for future growth, (h) Develop the database and web application using commercial design tools or CASE tools, (i) Implement and test the project application through a developmental website.

3.4 Project Analysis

3.4.1 Initial Development Phase

The preliminary investigation or planning phase was the starting point for all project development. This initial phase asked the question: is the project worth looking at? Once it was determined that the project was worth developing, application development started at the conceptual design level. The conceptual design level included: developing user roles, user constraints, and user interfaces. These project requirements became building blocks for the Use Case model. This project's Use Case model examined the external and internal processes that would take place in a student tracking application.

3.4.2 Project Model Development Phase

The design of the project grew from this preliminary planning phase, starting with the initial Use Case overview and ending with the maintenance and support phase. The Use Case model below represents the first step in the construction of the project application.

Figure 4. Initial Use Case model

The above Use Case model shows both students and advisers as actors. It also represents the initial external and internal processes taking place in the project. While students input their thesis data into the Tracking System application, advisers get their data through the Tracking System application. However, more processes are needed to complete the project. So, the next step in the Use Case level details students inputting project data: Abstract, Proposal, Thesis. It also demonstrates student projects being assigned, and advisers being assigned to students.

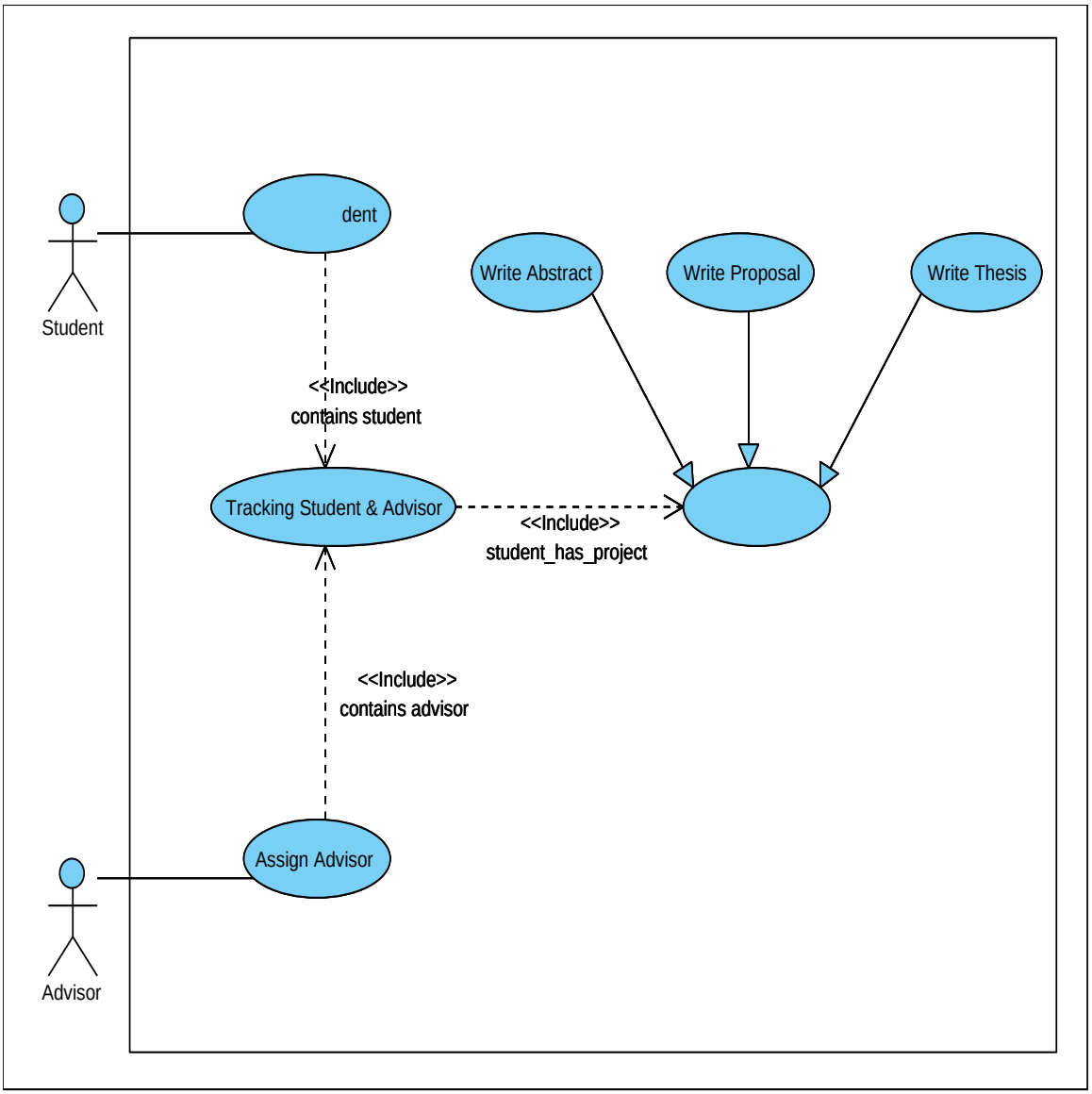

Figure 5. Use Case model showing the added student project data

Once each student is assigned an adviser and a project, the process is ready for the next step. In the Use Case view shown below, student, adviser, and project approval are all shown as actors. Both the student and adviser can input and retrieve data from the tracking system application. The project approval actor and final project approval actor are there to track initial project approval and final project completion. Once the student

gets adviser final project approval, the process completes, and the approved status completion date is stored in the Post Student table for future project status reports.

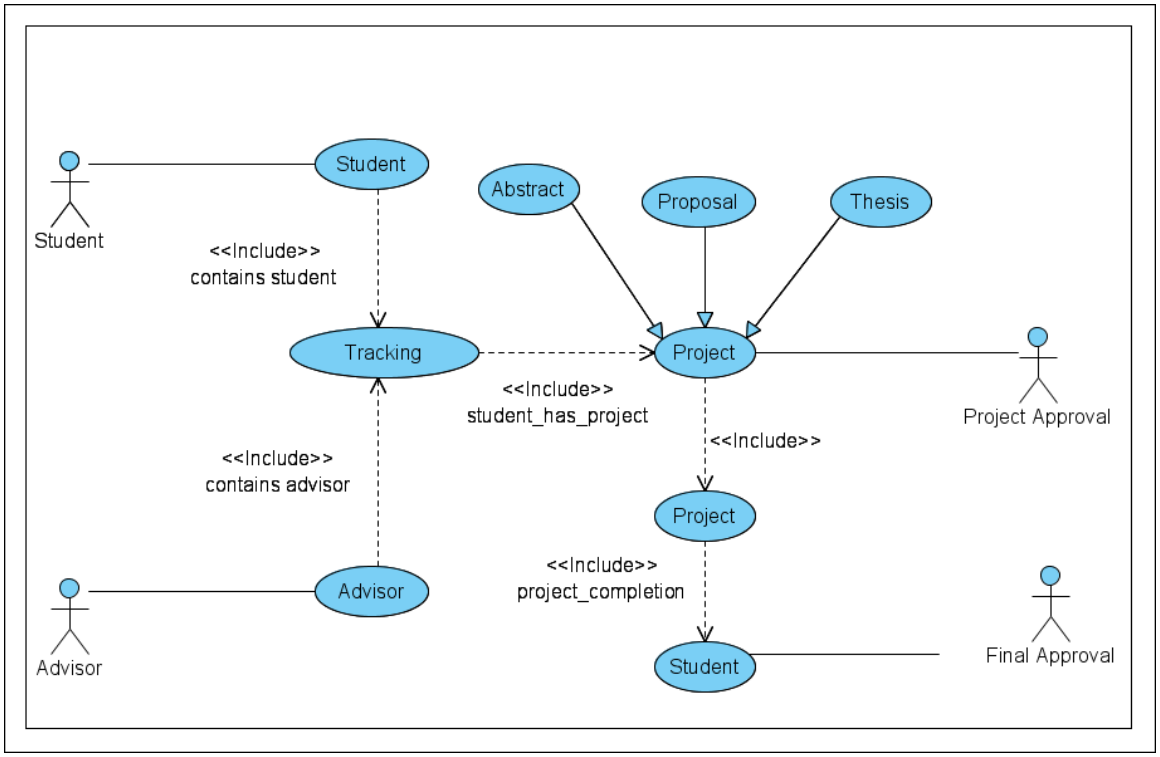

Figure 6. Use Case model showing the added student approval process

The final step is taking all of the above processes, and putting them into a communications diagram that steps through this approval process in more detail.

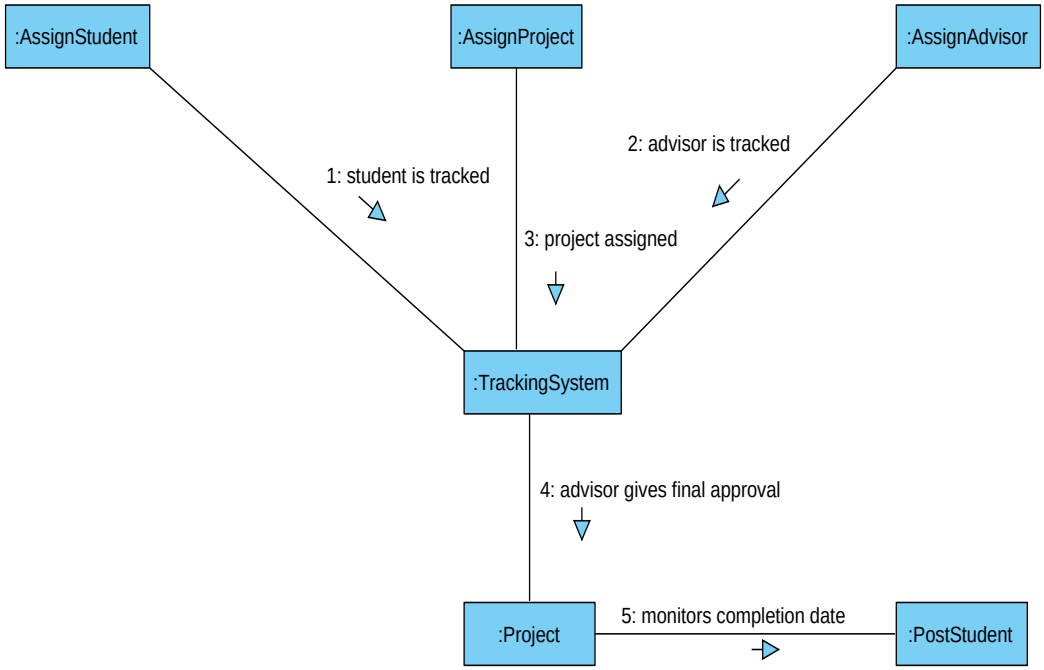

Figure 7. Communications diagram showing the tracking system process flow

In the above diagram each actor is tracked: student, adviser, and project. The student's project is tracked starting from being assigned to the tracking system through the post-student phase. Once the student's project is completed and receives final approval the student's project is tracked as a post project. This student post project is tracked by the completion date within the tracking system database at the logical level. All four levels of the system application combined: external, internal, conceptual, and logical build on the relational model, which sets the groundwork for the next step: building the relational database design.

3.4.3 Database Design Phase

The relational database design is derived from combining an UML model design with an ORM model design. See Appendix C, for a detailed UML and ORM diagram. The UML was initially used to guide the overall development of the database. The UML modeling approach was chosen so a clear picture of persistent relationships

between different database tables would show any problems in the original design concept (Naiburg & Maksimchuck, 2001). But the final database design could not be known when starting the initial design; the UML model tells what can go wrong if the database is not designed in the correct way (Date, 2005). After developing the UML model, the next step was integration of the UML model design into the ORM model design.

The ORM model set up the groundwork for the design pattern of the physical database model and is the most important step because is it correlates business rules with entity facts and objects in the ORM source model with data types and tables in the ER source model (Churcher, 2007). This association demonstrates how well the relationships between the ORM source model and the ER source model merged into the database diagram as a functional application. See Appendix C, for a detailed relational database model.

In the following ER model the design flow is divided into two distinct parts. The first part includes: Student, Confidential, User, Education, Phone, Mail, and Adviser tables. The second part includes: Project, Proposal, Abstract and Outline, Time Plan, Chart, Chapter, Thesis, and Post Student tables.

Although two separate parts of the ER model are shown, they are combined or joined through the linking table Tracking System found in part-1 of the ER model. This allows for the tables in the two parts to be independent from each another, while maintaining a link or association through the Tracking System table.

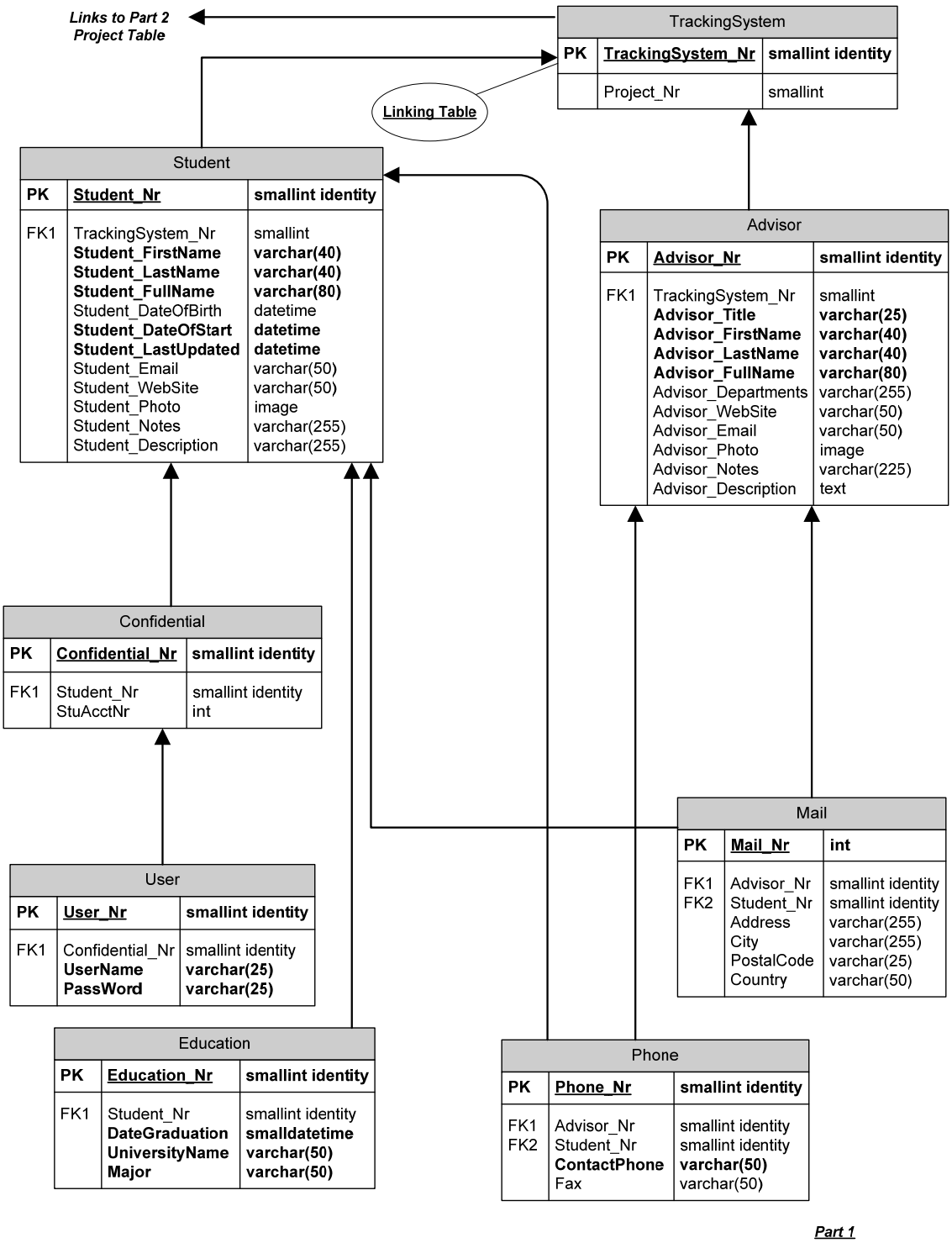

Figure 8. Relational Database view, showing Tracking System linking table

This model is the simplest view because it leaves out referential actions,

cardinality, and verb phases from the diagram. It also leaves out the objects, facts, and constraints that are found in the UML and ORM source model. Looking at this model in depth the question arises, at this point, about normalization. What is the correct database normalization? In this researcher's database, normalization does not go beyond the 3rd normal form (Hernandez, 2003). However, building a database from Visio for Enterprise Architects, the database is produced in the 5th normal form because of the build-in database rules. However, this researcher has found that this level of normalization was hard to verify with any confidence and the relational database model produced in the development of this project did not procedure a true 5th normal form (Allen & Terry, 2005). So what does this really mean? Examine the following definition: If the database "tables have a field that uniquely identifies each of its records, and each field in the table should describe the subject that the table represents" then the resulting table is in 3rd normal form. The final idea here is that if the database is properly designed and the methodology that is followed allows for no shortcuts, then the resulting design will already be in the proper normal form and thus should be transparent to the developer. See Figure 9 below for the second part of the relational database model. A detailed relational database model can be seen in Appendix C.

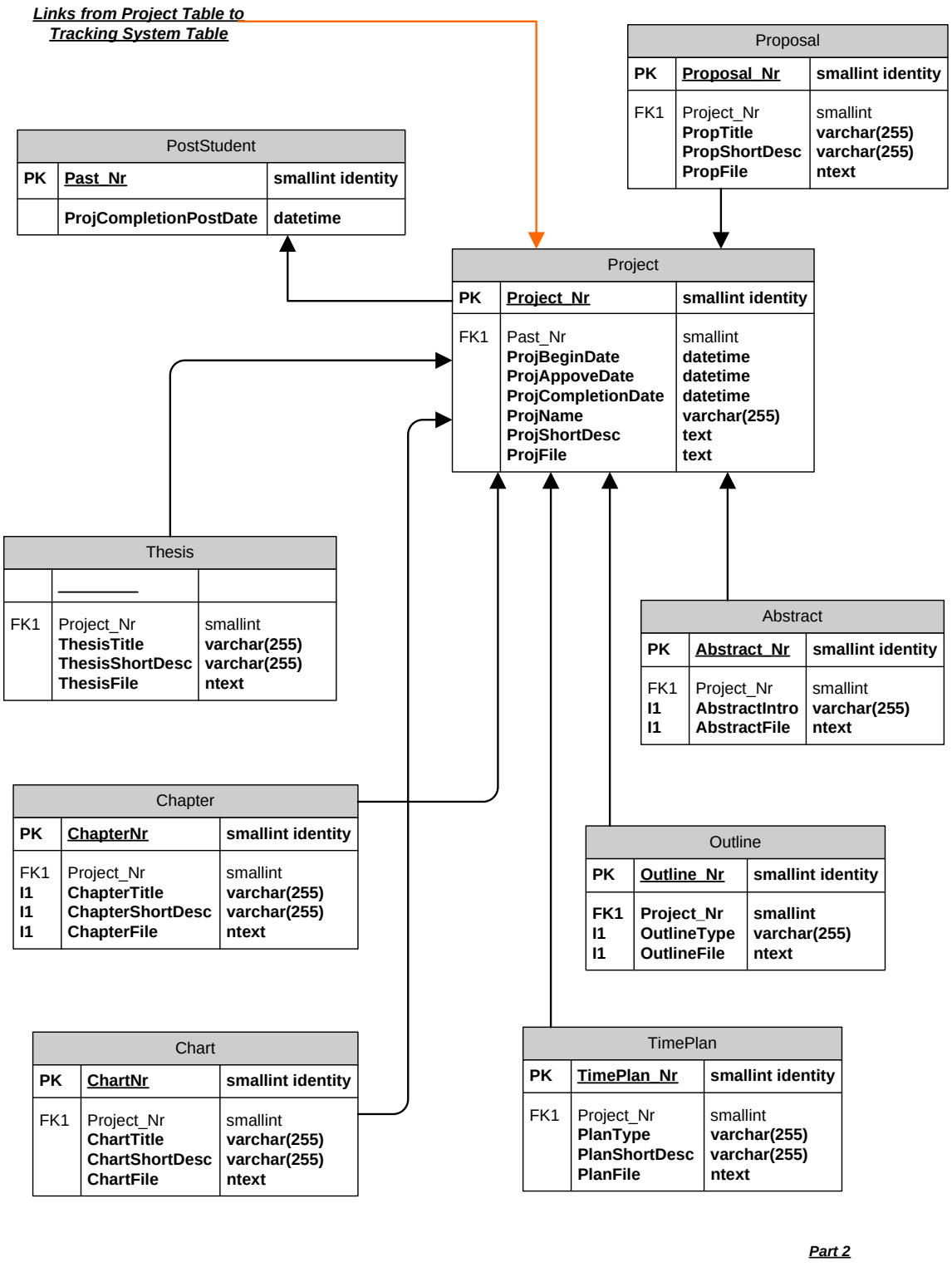

Figure 9. Relational Database view, showing project tables

This physical database was made from the ORM model, which was used to

produce the PHP code used in this project for the web-based database data entry. Now, the next step is the design is the forms for inputting and retrieving information from the database.

3.4.4 Web Pages Design Phase

Two approaches were examined. The first was designing with ASP.NET web pages and the second was designing with PHP web pages. Both APS and PHP create Web projects by using Hypertext Markup Language (HTML) and some programming language to produce source code for the creation of dynamic web pages (Valade, 2006). Both resulted in approximately the same web page design. So, programming languages such as C# (C-sharp) or Microsoft's Visual Basic.NET could be used. However, of the two possible forms design, this project's goal was to use ASP.NET, but found that the PHP coding design was easier to implement. Using an advance modeling tool called PHP Runner, this researcher was able to generate the complete starting page interface for the tracking system project as shown in Appendix C.

This main tracking system menu view represents the starting point for all student and adviser information data logging and data retrieval from the application. In addition, PHP was chosen as the final design approach for interacting with the student tracking database because it was the easiest to implement and is also an open source program (Williams & Lane, 2002).

3.4.5 Integration and Implementation Phase

The final project step was getting the whole project: database, forms, web page interface, and web server working together. To accomplish this task, this researcher used PHP Runner to build a connection to MySQL database through MySQL server, and then

make an additional connection through PHP Runner's demonstration web page link to test the application. Regression testing was done on this application, using test data for each web page application view and this project step was then determined a success.

3.4.6 Maintenance and Support Phase

The project application is supported through this researcher's website at bobtschiemer.net. Support for the project includes database and software maintenance to insure the database and user interface remain fully functional for all users.

3.5 Review of the Deliverables

The project deliverables were developed from the commercial application design tools, such as PHP Runner, Visio, Visual Studio, SQLog Enterprise, and Enterprise Architect. These deliverables included the relational database design and the web page forms, and were tested through a demonstration website.

The review of all project deliverables included the following: (a) Completed methodology that drove the design phase for this project, (b) Gantt chart schedules that showed the development process from beginning project research to completion of the project application, (c) Relational database models, (d) UML and ORM models, and (e) Lessons learned about the final application design.

3.6 Outcomes

Outcomes included the following: (a) Prototype web-based relational database application that allowed users to access stored student and adviser data through web-views, (b) Relational database in 3th normal form, (c) ORM models that were built demonstrating the use of interfaces between objects in the database, (d) Database was correctly designed, which reduced future changes to the overall design structure, (e) ER

models were built using user requirements and governed by business rules and a properly designed database.

3.7 Summary

The application started from the initial concept describing “what the project application would do” and “how the user would interact with the application and developed into a design methodology that became the building blocks for the relational model”. The logical model was derived from combining the waterfall and the rapid prototype methodologies. Additionally, these methodologies provided the requirements for the relational database design that also expanded on designs of the UML, ORM, and ER models. The end result for the application was successful because the correct relational model was used. This allowed for a proper database design, which led to a solid web page design. If less attention was given to the initial design, then the end result would not have been as successful as this design demonstrated.

Chapter Four: Project History

4.1 How the Project Began

The project began with discussions between me and my adviser. My adviser needed someone to explore the development of a software project that used Microsoft Visual Studio.NET software, combined with my current project idea for developing a process for graduate students to track their professional projects as they wrote them using PHP coding with a SEQUEL database. These two concepts formed the ground work for my project that developed into my project thesis.

To get this project started, a solid relational database model was necessary to provide a solid foundation for a web-driven database project that would support 200 - 300 students. Because developing the project required this researcher to be both the designer and developer, and develop the project through the use of modeling tools such as Microsoft .NET, PHP web page technologies, and other advanced database design technologies that used UML and ORM modeling (Bell, Johansen, Reynolds, Thangarathinam, & Whitlow, 2002).

The database and the Web-based interface to the database worked as expected. Overall, the project accomplished all the requirements that were originally planned.

4.2 How the Project Was Managed

This researcher is the owner of the project and was responsible for working as developer, analysis, and implementer, which included work on all of the developmental phases: requirements, specifications, designs, integrations, and implementations. A Gantt chart was developed to guide in the design process. In addition, key deliverables were identified and implemented into the final application build using the Gantt chart timeline

and design methodologies.

4.3 Significant Milestones in the Project

The most important milestones during the design and implementation of this project were the initial design choice that would be the foundation for the database design. Deciding on the proper methodologies put in place the building blocks for the actual building of the physical database and the implementation of the entire project. The combination of the waterfall and rapid prototype methodologies set the stage following the design process: analysis, specification, design, construction, and implementation. This process divided the database building into distinct phases following a design outline. Once these milestones were established, the next step was to build the database and connect the database to web-driven pages through a website. These development steps lead to the final milestone.

The final milestone was the implementation phase. Running the completed project on a development server was the first step. This included uploading all tables to a demonstration and testing website and logging into MySQL database engine. This was also successful and demonstrated that the project worked as originally planned. The following milestone table summarizes the overall project milestone development. A detailed view of the project application development milestones and overall project outline can be seen in Appendix A.

Table 2. Consolidated list of Project Application Milestones

Tasks	Begin Date	End Date
Analysis Phase		
Begin project research for professional project	29-Jul-05	29-Jul-06
Specification Phase		
Determine application specifications to software development, database and software design tools	01-Aug-06	14-Aug-06
Design and Construction Phase		
Physical application modeling, database, and web pages view	20-Aug-06	12-Oct-06
Integration and Testing Phase		
Demonstration of project application and website testing	15-Oct-06	18-Dec-06

4.4 Changes to the Project Plan

During the initial plan application design steps 1-6, the plan went through several important changes. First, all plan upgrades grew from the ORM model of the database and their relationships to that model. This model was the bidding glue behind the entire application design. The ORM design model went through several changes that included updating tables and adjusting their relationships to generate a proper 3rd normal form for a relational database model. The second, and no lesser important, change was to the database. The relational database produced by the design methodology was sound and without workarounds. In other words the final relational database design has not changed in its basic structure, which eliminates future overall changes to the design. However, this does not mean that additional requirements would not drive new table additions and

functions to the design.

Third, changes to the application were focused on changes to the application's user forms design. The use forms provide for inputting and retrieving information from the database. The fourth change was with testing the overall project design. Testing the application meant testing on a commercial server and not just testing on this researcher's development computer.

4.5 Project Evaluation

The evaluation of the project application would not be complete without examining the differences between the project's chosen relational model design and a combination for relational and object design models. The relational model was chosen because it is good at retrieving large amounts of data whereas the object model is good at retrieving single information data. However, the best of both worlds would be the combination of the relational and object model or the object-relational database model. This is understandability the possible best solution, but for this researcher's project the relational model is the best solution. Since the project application is designed to support a small population of users, management of smaller amounts of data is necessary.

Relational databases can be used to access unique or single data efficiently, similar to the object model, which allows for using a relational database instead of an object database to support the smaller project application. This brings up an additional point about the project relational database design. The project application was designed as a transactional database or a database that supports small transactions, whereas an object model is more suited for OLTP database design where possibly millions of users may access the database in one day. Because the project application was focused on an

in-house user-server database this makes the relational model the best solution to this requirement and meets the smaller support usage specification.

4.6 Discussion of Successes and Failures

The project goals were met, but not without some stumbling blocks. The first stumbling block was with the RDBMS. Working with entire suites of database software like Visual Studio and PHP Runner, designing from them required an understanding of how these software programs worked, and how to manger their communication with the relational database engine. The next issue was the project methodology, which focused on getting the initial information-gathering steps in place before proceeding with design steps. For example, during the design phase the model went through many versions before the right mix of tables, attributes, and their relationships were correct in design, meeting the normalization steps in setting up the proper relational design. The next stumbling block was with learning script coding in PHP. Finally, during the implementation step of the project, the successful working of the integrated database with application website demonstrated that the final project was working as expected and could be consider a success.

The design phase was the most difficult to accomplish because it involved the most detailed plan. Building the initial UML and ORM model required reworking both models to product the proper relationships between database tables. After this was accomplished the database was constructed successfully. During project development no project design steps lead to a total failure, a failure that could not be worked through to correct the program and get the proper solution.

4.7 Discussion of Project Variables

The discussion of project variables included the following: (a) Significant learning curve for database development tools, (b) Understanding and building a relational database model design was harder than expected.

4.4.1 Development Tools

The database design tools used in the development of this project proved a difficult task because each design tool has many features, which need time to get the proper understanding of how to use them. Of the many design tools used in this project development, three standout. The first two IDE tools are Visual Studio and Visio. This program was the 2003 Enterprise Architect version that allows for the integration of Visio for Enterprise Architect producing ER models that will forward engineer into relational databases. This is a nice combination between the two design tools, since newer versions of both Visio and Visual Studio are not able to accomplish this modeling capability. The final design tool was for producing PHP code and web page views. This program was easy to develop with and produced good overall web views, which had the added benefit for reducing the PHP code script to a minimum since much of the code was program generated.

4.4.2 Understanding the Relational Database Model Design

Building the proper relational database model took more development time than was originally planned. Understanding the complexities of the relational database model required a solid understanding of all aspects for the relational model as explained by Dr. Codd (Codd, 1990), and C.J. Date, who both published a number of papers and books on the subject of relational database design. So understanding the relational model is

important because, how it interacts with other development models such as object and object-relational models, and how to put this understanding to use when building the relational modeled database.

4.8 Project Findings

The completed project finished with a working and sound relational web-based database design. The completed design was tested on a commercial website and worked without any major problems.

There are two major recommendations found for the project. First, the project needs more advanced reporting capability. Now, the only reporting information available is the post-student status, which reports when a student has completed all project requirements. In addition, the project needs an expanded user interface view for inputting and retrieving information. The current user interface is adequate, but a more sophisticated GUI would make posting and retrieving data easier.

4.9 Summary

The project concluded with a working web-based database. The only remaining issue is with the implementation phase. This phase is contingent on the users' acceptance of the project. A demonstration to the user is necessary for final acceptance and for working out any additional details for final implementation of the project.

Chapter Five: Lessons Learned and Next Evolution of the Project

This project has given a greater insight into the steps needed to learn about and how to construct all the different interdependent development links that make up the whole working project. In completing my project, the project could have been developed for a large audience (supporting more students) and on a boarded level (email capabilities). The only draw back to the final project is that it is more static than dynamic that was originally thought and this would be the next focus on the project if more time for project development was allowed.

5.1 Lessons Learned

This project started with the goal of building a web-based database application that could be logged into from any web browser. This goal was accomplished, but the steps necessary to get there were extended beyond the initial methodology, which required the following changes: (a) Both the desktop and web applications required data management, (b) Prototype development leads to many changes, and (c) Using design tools are necessary.

Both the desktop and web application required data management, or the tracking of authorized student and advisers. This required a login tracking system for both groups that is secure and can be kept up-to-date, since proprietary information is being stored in the relational database. See sequence diagrams in Appendix C.

Prototype development leads to many changes cycle that can influence the outcome of the final application design. Sticking to the original specifications and requirements of the relational database design and still incorporate changes during

development is a balancing act between chosen methodologies and required changes to the design to produce a sound and stable relational database model. Building a conceptual model first insures that the physical database model will be solid in design and still remain flexible enough for future changes.

Using database and web application design tools is necessary. The all-in-one software suite specifically designed for this purpose is not always the end all product that gets the design results the developer needs. The design of this project, more-than-one design tool was necessary to accomplish the completed design. For example, for the initial conceptual relational model design, Enterprise Architect was used to generate an UML model that could be generated into a logical model. Getting to the physical relational model took the use of Visio Enterprise Architect, which uses build in database and verbalization rules to generate a proper physical database.

Producing a database that would work well with a PHP designed website required the correct database engine, which was MySQL. The final step was producing the web pages, using PHP and this was done with a design tool called PHP Runner. All of these tools were needed to accomplish the entire build and the all-in-one design tool would not have accomplished this task to the satisfaction of this developer.

5.2 What I would have done differently in the project

Despite the successful outcome, the following are three items that this researcher would change: (a) Get more help from outside experts in designing web applications, (b) Interaction with users was not possible, but for additional future application enchantments, this would be necessary to produce the correct design outcome, (c) Develop a testing scenario for application design, (d) use this scenario to follow-up on

software problems.

Get more help from outside experts in designing web applications. Designing web applications require a thorough understanding of HTML and PHP or some other type of web scripting language. Using a commercial design tools helps a great deal, but have the input from other experts with a board background in website and web application development can strengthen the web-based application development process.

Interacting closely with the users is important because they know what they want from the working application. However, since this project was developed as a prototype, then current users were not available for getting feedback about the application and how the users interact with it. Future design changes would require regression testing and would allow for problem reports to be generated from this testing. This would also allow proper changes to be made to the application.

5.3 Discussion of whether or not the project met initial project expectations

The project accomplished all initial requirements for the initial prototype design and followed the methodology outlined in the project. The primary goals met were the following: Sound relational database design that can handle 200-300 hundred student usage, and easy to use web page application interface that allows or data storage, data retrieval, and data management all through one application interface.

However, this may need to be increased as the database usage increases due to more students and advisers needing access to their data.

An easy to use, user web form is necessary for optimal performance with storing and retrieving data from the database. Web forms for accessing the data was designed for easy use, but will need additional changes to add more enchantments to the design as

future requirements become a necessity to keep up with new technologies and user requests for changes.

The final application provides for easy web application server interaction, the database can be easily updated and uploaded to the test server. This permits continued regression testing and software fixing and updating.

5.4 What the next stage of evolution for the project would be if it continued

The next phase for the project would be the development of object-relational database design. This is a more advance database modeling concept that can used with databases designed with Oracle, MySQL, and MSSQL engines. This concept also uses the power of object-oriented programming (OOP) and Windows forms applications, the .NET kind for project development. Using the .NET technologies allows for the developer to build dynamic websites. The .NET Framework allows developers to use languages such as C# to build dynamic websites that will run with any browser or any Windows applications. The building of the front end for the application and updated user interface are the next logical steps in the design process.

This would be the next level of development for this researcher' application, allowing for a dynamic web application giving the user a more elegant interface and page views for the developed application.

5.5 Conclusions and Recommendations

In conclusion, the project accomplished all initial requirements, which included the following: (a) Building a relational database for storing student and adviser data for 200-300 students, (b) Making the user interface easy to understand and use, and (c) Data management through one application interface.

In addition, recommendations for project improvement were the following: (a) Project needs an improved reporting capability, (b) The project needs a better user interface for inputting and getting data, (c) Get external professional help in developing the overall project design, and (d) Developed a regression testing model for future software upgrades.

5.6 Summary

Using this relational design rather than building from an object-relational design proved the best design choice for this project application. This database design integrated well with the PHP code and MySQL database engine. However, an MSQL database engine would work equally well for this design.

Developing from the UML and ORM model designs provided the best relational database model design for the application. The ORM model design build on the UML relationships between objects, which in turn build on the ORM model business rules at the conceptual level. This conceptual schema can be optimized for relational database design that created a design bridge between the logical and physical design model, the combination of the UML and ORM modeling methodologies go beyond the traditional attribute based ER modeling. However, all three models are use in the development of this project application, with the relational database model as the outcome for their designs.

Implementation of this relational model was the final step in the development process, building from the conceptual model to the physical model. This step was enhanced with the use of database design tools, although none of the design tools used in this project were all inclusive in their design output and needed the support of each others

modeling engines to get the desired result in the completed project application.

The final application was completed using server sided PHP coding, linked to MySQL relational database. This combination gave this developer an easy to update database and web page application. As usage requirements grow, more tables can easily be added without changing the underlying database structure. Additional web page views can also be added to accommodate database changes keeping the user interface current with table and attribute changes.

The relational database model is still rock solid in database design, being part of all future database RDBMS development for many years into the future (Date, 2005). The future of this researchers' project is sound in methodologies used in its design because, even with post-relational databases such as business logic servers (Date, 2005) as the next new generation of future general-purpose databases, the relational database, however, will still be a strong part of future database modeling architectures.

References

Allen, S., & Terry, E. (2005). *Beginning relational data modeling*. Berkeley, CA :; New York :: Apress ;; Distributed to the book trade in the U.S. by Springer-Verlag.

Bell, J., Johansen, B., Reynolds, M., Thangarathinam, T., & Whitlow, N. (2002). *Developing C# Windows Software*. Birmingham: Wrox Press.

Churcher, C. (2007). *Beginning database design*. Berkeley, CA :; New York :: Apress ;; Distributed to the book trade worldwide by Springer-Verlag New York.

Codd, E. F. (1990). *The Relational Model for Database Management: Version 2*.

Date, C. J. (2005). *Database In Depth Relational Theory for Practitioners* (Vol. 1). Sebastopol: O'Reilly Media, Inc.

Everest, G. (2001). Foreword. In *Information Modeling and Relational Databases* (pp. xx-xxi).

Griffiths, I., Flandres, J., & Sell, C. (2003). *Mastering Visual Studio.NET* (1 ed.). Boston: Morgan Kaufmann.

Halpin, T. A. (2003). *Database modeling with Microsoft Visio for Enterprise Architects*. San Francisco: Morgan Kaufmann Publishers.

Hernandez, M., J. (2003). *Database Design for Mere Mortals*: Addison-Wesley.

Kauffman, J., Spencer, K., & Willis, T. (2003). *Beginning ASP databases*. Berkeley, Calif. :: Apress.

Mercer, D. W., Allen, K., Steven D. Nowicki, Mercer, D., Dan Sqauier, & Choi, W. (2004). *Beginning PHP5* (1 ed.). Indianapolis: Wiley.

Naiburg, E. J., & Maksimchuck, R. A. (2001). *UML for database design*. Boston: Addison-Wesley.

Powell, G. (2006). *Beginning database design*. Indianpolis: Wiley.

Rosebrock, E. (2003). *Creating Interactive Websites with PHP and Web Services* (Vol. 1): SYBEX.

Schach, S. R. (2002). *Object Oriented and Classical Software* (5 ed.). New York: McGraw-Hill.

Valade, J. (2006). *PHP & MySQL : your visual blueprint for creating dynamic, database-driven Websites*. Hoboken, NJ: John Wiley.

Williams, H. E., & Lane, D. (2002). *Web database applications with PHP & MySQL* (1st ed.). Beijing ; Sebastopol, CA: O'Reilly.

Appendix A

Project Task Plan

Figure 10. Project development Gantt chart

Appendix B

Project Screen Shots

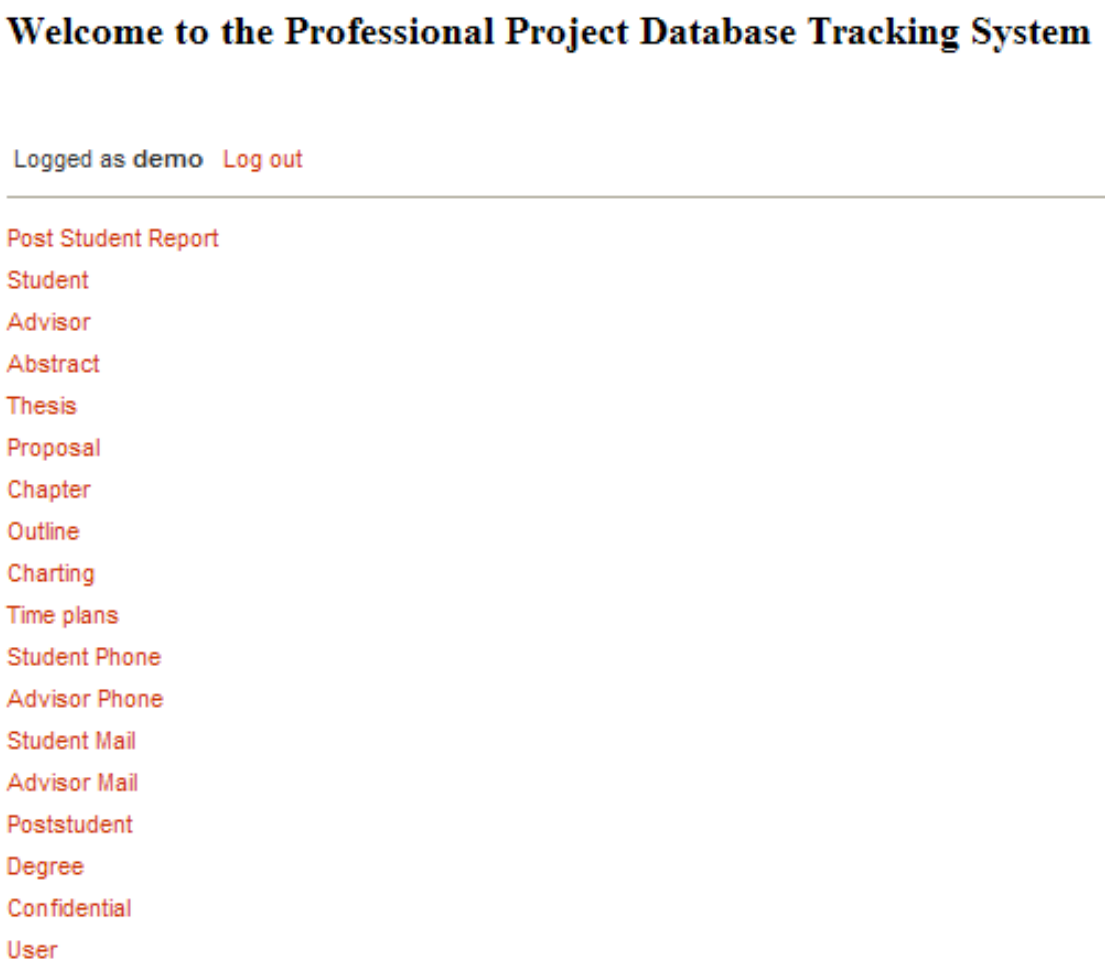

Figure 11. Open Screen of the Tracking System view

Appendix C

Project Diagrams

Figure 12. Relational ERD model part – 1

Figure 13. Relational ERD model part – 2

Student/Advisor setup accounts and Student/Advisor input personal information. Additionally, Student inputs project paper work, while Advisor uses data to track student's progress. The setup account and get/input data scenarios are also shown.

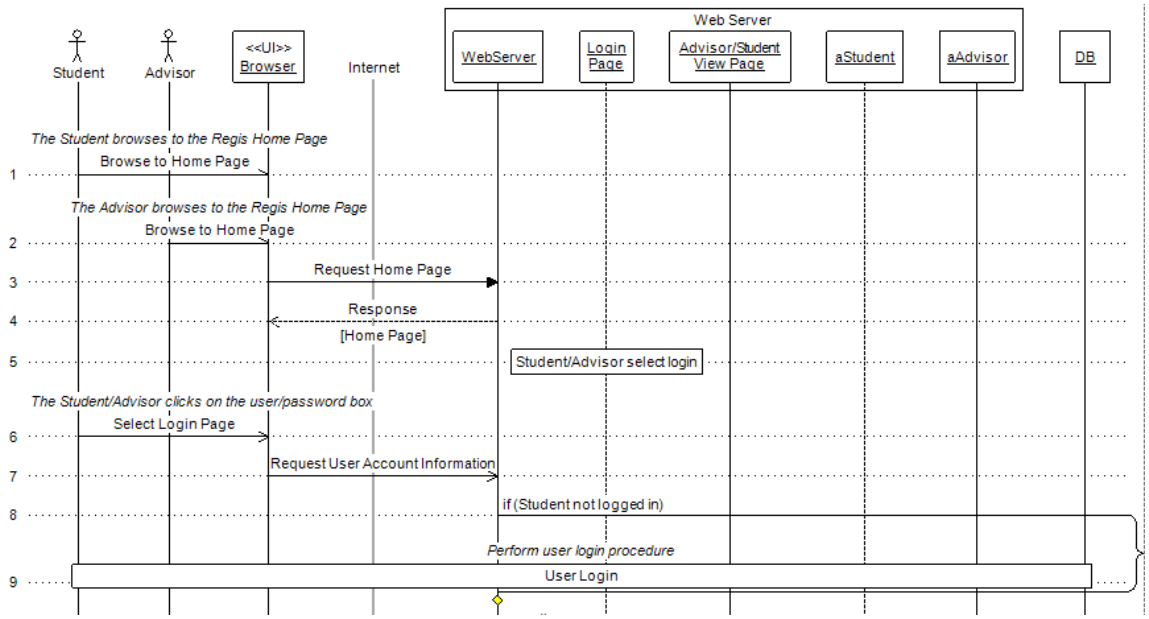

Figure 14. Sequence Diagram part – 1

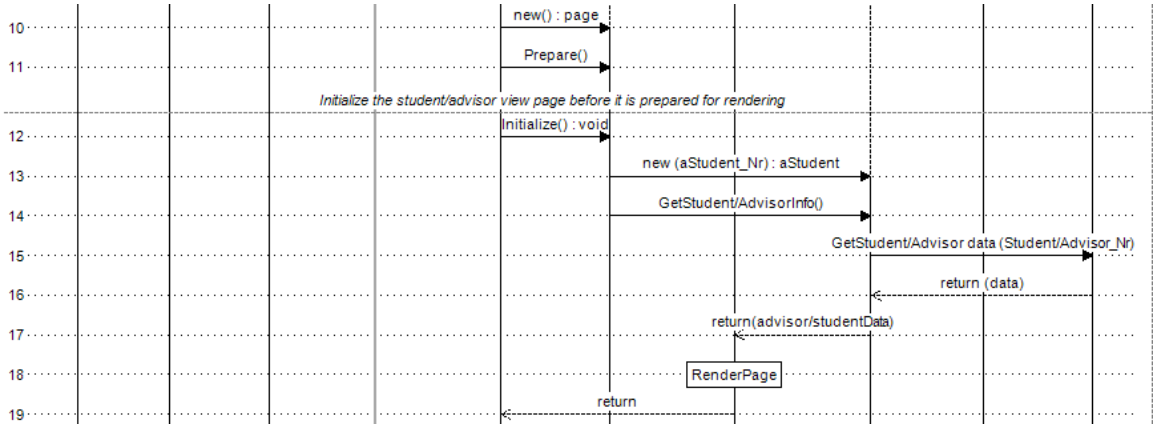

Figure 15. Sequence Diagram part – 2

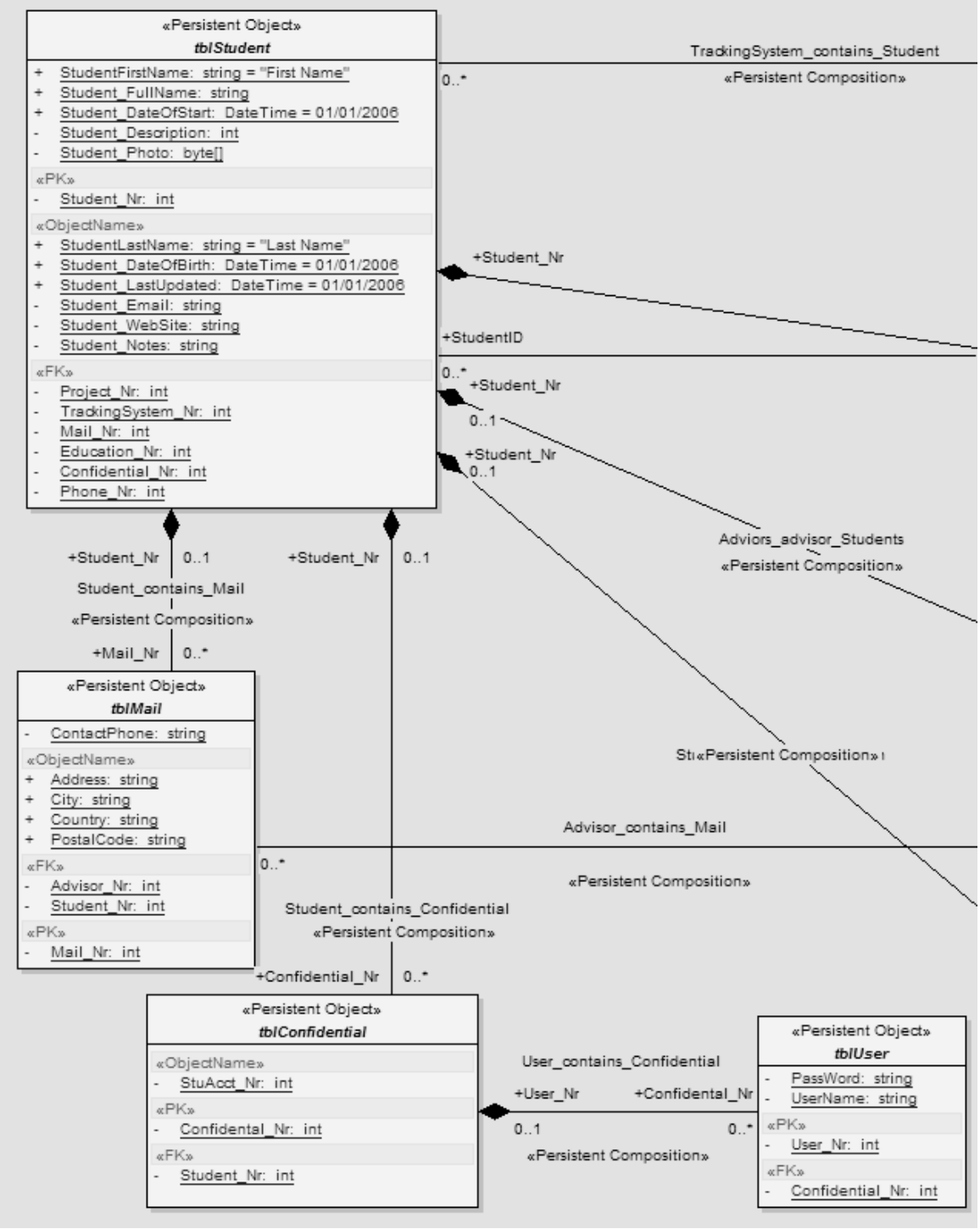

Figure 16. Regis Tracking System Object Model part – 1

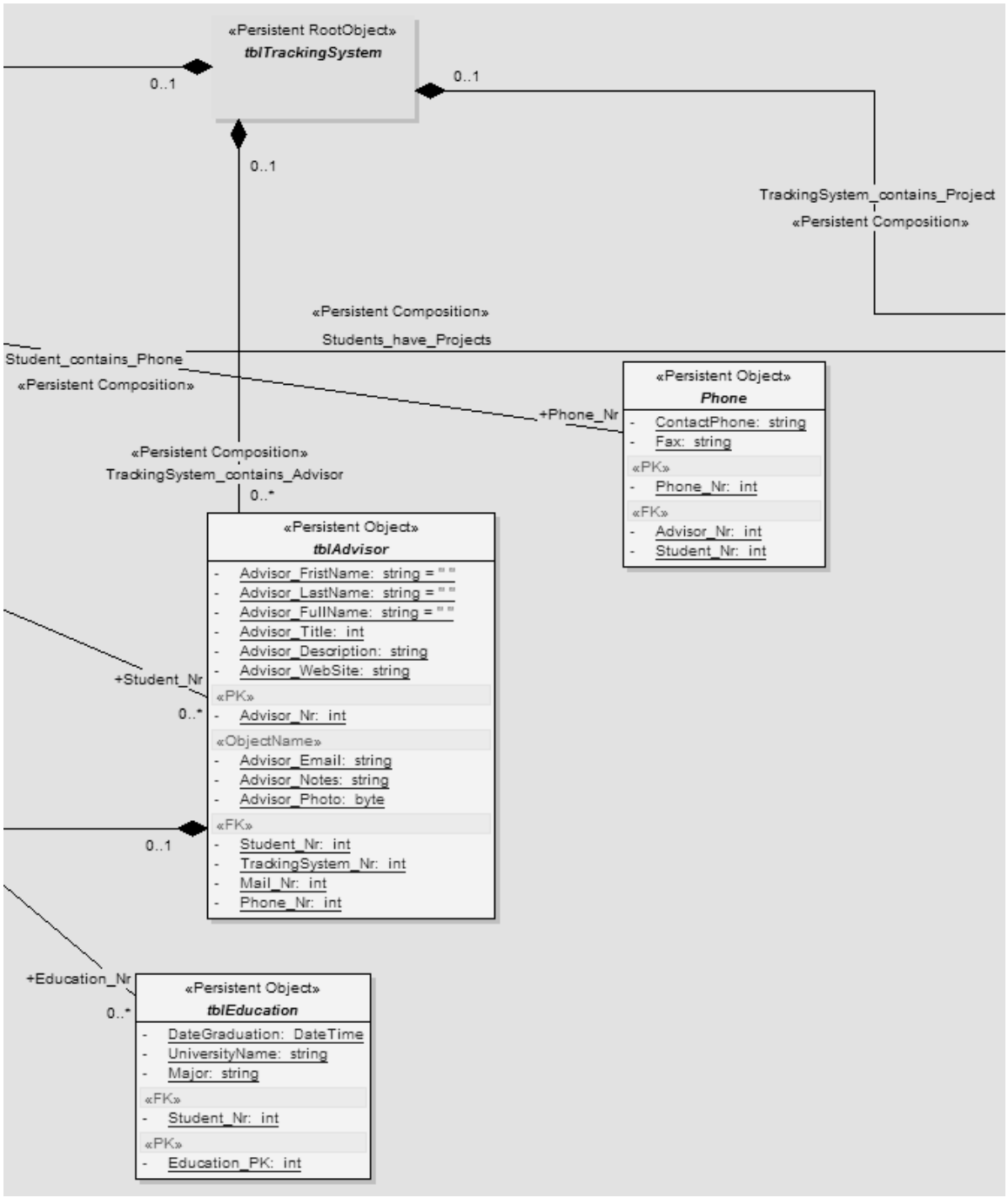

Figure 17. Regis Tracking System Object Model part – 2

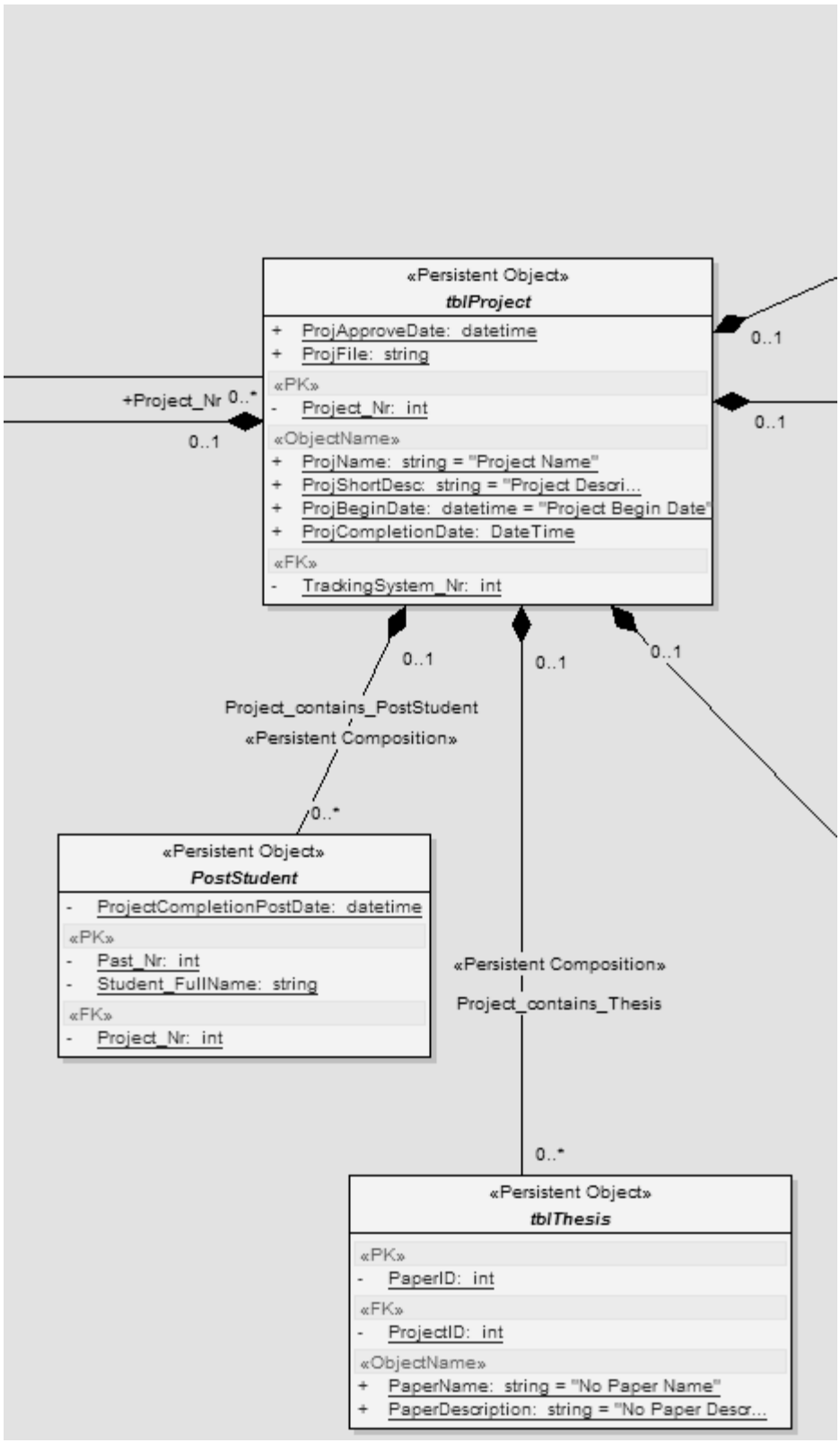

Figure 18. Regis Tracking System Object Model part – 3

Figure 19. Regis Tracking System Object Model part – 4