

GRADUATE STUDENT PROGRESS TRACKING SYSTEM

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Pramodh Kumar Thalloji

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

May 2010

Fargo, North Dakota

North Dakota State University
Graduate School

Title

GRADUATE STUDENT PROGRESS

TRACKING SYSTEM

By

PRAMODH KUMAR THALLOJI

The Supervisory Committee certifies that this *disquisition* complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

North Dakota State University Libraries Addendum

To protect the privacy of individuals associated with the document, signatures have been removed from the digital version of this document.

ABSTRACT

Thalloji, Pramodh Kumar, M.S., Department of Computer Science, College of Science and Mathematics, North Dakota State University, May 2010. Graduate Student Progress Tracking System. Major Professor: Dr. Kenneth Magel.

In this paper, I propose and implement an application to track the graduate student progress in various Computer Science and Software Engineering graduate programs. This application will provide lists of students who in a given year or semester did not fulfill any of the numbered requirements of their programs. It will provide a list of students who passed the comprehensive or qualifier exams in that year or semester. It will also list the students who have taken longer than the time limit mandated by the Graduate School to complete their plan of study. This application was developed with the assumption that the database models the original PeopleSoft database. This application shall be modified to use the registrar's PeopleSoft database and get the information associated with students.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Kenneth Magel for helping me choose a paper topic. I convey my regards for his continuous help and support during my research. I would also like to thank Dr. Weiyi Zhang, Dr. Tariq King and Dr. Lisa M Montplaisir for serving on the graduate committee.

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGMENTS.....	iv
LIST OF TABLES.....	vii
LIST OF FIGURES.....	viii
LIST OF APPENDIX FIGURES.....	ix
CHAPTER 1. INTRODUCTION.....	1
1.1. Motivation	2
1.2. Approach	2
CHAPTER 2. STUDENT TRACKING SYSTEM	4
CHAPTER 3. RELATED WORK	6
3.1. Background.....	6
CHAPTER 4. TOOL DESIGN AND PLANNING	8
CHAPTER 5. TOOL IMPLEMENTATION	16
5.1. Architecture	16
5.2. Software.....	17
5.3. The Underlying Database System	19
CHAPTER 6. SCENARIO.....	26
6.1. Student Login	26
6.2. Advisor Login.....	29
6.3. Administrator Login	35
CHAPTER 7. CONCLUSIONS, LIMITATIONS AND FUTURE WORK.....	40
7.1. Results	40

7.2. Conclusions	41
7.3. Future Work.....	41
BIBLIOGRAPHY	43
APPENDIX A	44
APPENDIX B.....	48
APPENDIX C.....	59
APPENDIX D	61
APPENDIX E.....	65

LIST OF TABLES

<u>Table</u>	<u>Page</u>
5.1. Graduate student tracking system database table information.	19

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
4.1. Notes from brainstorming sessions.....	9
4.2. Notes from the brainstorming session continued.	9
4.3. Prototype using Access database-1	10
4.4. Prototype using Access database-2	11
4.5. Prototype using Access database-3	11
4.6. Hand charted application logic flow.....	12
5.1. Student tracking system architecture.....	16
5.2. Eclipse IDE with Tomcat integration.....	18
5.3. SQL Developer	18

LIST OF APPENDIX FIGURES

<u>Figure</u>	<u>Page</u>
A.1. Login page.....	44
A.2. Student invalid login.....	44
A.3. Student login homepage.....	45
A.4. Student status menu display page.....	45
A.5. Student alerts menu display page.....	46
A.6. Advisor info display page.....	46
A.7. Student general information page.....	47
A.8. Student login quick links page.....	47
B.1. Advisor login home page.....	48
B.2. Advisor <i>MyStudents</i> menu page.....	48
B.3. Advisee status display page.....	49
B.4. Advisor search menu page.....	49
B.5. Search by name results page.....	50
B.6. Search by student name course history page.....	50
B.7. Search by year page.....	51
B.8. Search by year results page.....	51
B.9. Search by year criteria plan of study approved listing.....	52
B.10. Search by year criteria credit shortage listing.....	52
B.11. Search by year, list of students who did not comply with graduate school mandated time limit for approval of plan of study.....	53
B.12. Advisor Login: Search by year, list of students who is yet to get approval for plan of study from graduate school.....	53

B.13. Search by semester display page.....	54
B.14. Alerts menu display page.....	54
B.15. Alerts indicating list of students who have not completed the core course requirements.....	55
B.16. Alerts indicating list of students who have not completed the total credits program requirement.....	55
B.17. Alerts indicating list of students who have not completed the total seminar credits requirement.....	56
B.18. Alerts indicating list of students who have not appeared for the comprehensive qualifier exam.....	56
B.19. Alerts indicating list of students who are yet to get the plan of study approved by the graduate school.....	57
B.20. Alerts indicating list of students who did not fulfill the graduate school mandated time limit for the approval of the plan of study.....	57
B.21. Alerts indicating list of students who did not defend their degree.....	58
B.22. Alerts indicating list of students who have default advisor.....	58
C.1. Administrator Login: Home page display.....	59
C.2. <i>Add/Modify</i> menu display.....	59
C.3. Addition of comprehensive qualifier information.....	60
C.4. Modification of comprehensive qualifier information.....	60

CHAPTER 1

INTRODUCTION

We must monitor the progress of the graduate student regularly. This should take place least once every semester by the respective advisor or the Dean of Academics. The time line and advisor-to-student ratio makes it almost impossible to track the progress of the students. Regular assessment of student's progress can identify and help solve many problems. These can be inadequate support or supervision, disenchantment with projects, personal issues or various aspects of a student's program. These problems interfere with the student's ability to work. A consistent indicator of a graduate program's quality is the timely completion of that program. The culture of timeliness allows programs to be productive and efficient.

Wide varieties of progress-tracking systems are in use today. I prescribe an automated tracking system. This system shall replace the traditional systems that are in use today.

The progress tracking systems fall into five broad categories:

- Annual meeting of student with Graduate Advisor/Director
- Annual progress report by student
- Annual progress report by supervisor
- Annual progress reports by both student and supervisor
- Tracking student supervisory committee meetings

1.1. Motivation

The current system for tracking the progress of a graduate student is predominantly paper based. This form of system leads to delays in information gathering. The manual analysis of the information gathered further delays process. A considerable amount of time is lost by the time the advisor and student meets. This will result in unavoidable circumstances such as student ending up taking a course that was not required. A very little can be done about such things. The intent of this paper is to provide with information and alerts that give a clear picture of a student's progress. These benefit both the student and the advisor.

We discussed different kinds of traditional tracking systems in the introduction. We need to replace them with new tracking systems. The new systems shall be capable of providing the course information, summarized alerts and program requirements at fingertips. This shall only prompt interest in monitoring student progress throughout their program but also guide them with alerts and suggestions. I strongly believe that such a system in place will improve communication between the student, advisor, faculty and department's administrative staff.

The motivation behind this project is the need for an effective method of communicating the alerts based on program requirements. I anticipate that this will benefit the department as a whole. The students and the advisors will be benefit from the use of this application.

1.2. Approach

The student tracking system presented in this paper will work on different stages of student progress.

Admission and Registration Stage:

The admission and registration stage involves the formal application, admission, first-time registration and enrollment. The application then starts tracking the student automatically. The application used the completed courses for analysis. The results of this analysis are the alerts and suggestions.

Enrollment and Course Work:

Students will enroll into different courses offered by the department. Our tracking system will evaluate the courses taken against the requirements for a given program.

Completion of the program requirements:

As a student nears completion of the program requirements, the proposed tracking system will evaluate the student against the requirements of a department's given program of study. Any student who does not fulfill these requirements will receive alerts. The instructor monitors the student's progress by viewing the same alerts. These may include completion of the required comprehensive qualifier or filing the plan of study etc.

Evaluation, Alerts and Communication:

The graduate student progress tracking system has built-in intelligence to track and evaluate the progress of the student and provide alerts. These alerts will help the student by tracking the needed requirements and providing the time limit specifications set by the school to complete them. The system advises advisors and advisees with specific alerts pertaining to the completion of a degree. This will prompt communication between both parties. This system assures that the requirements for completing the program are satisfied.

CHAPTER 2

STUDENT TRACKING SYSTEM

The aim of the student tracking system is to focus academic advising resources on students. Using this application helps especially students who need assistance. This process helps student graduate in the mandated time limit as per the program requirements by informing the student of the following:

- the core courses students need to take
- when these courses are to be taken
- timeline to complete their course work and graduate in a timely manner
- specifying if the student is complying with the requirements of the program
- comprehensive qualifier information
- providing the deadlines with respect to the requirements
- the graduate school's approval of the plan of study
- automatic monitoring and alert information

Students are encouraged to meet with their advisors during orientation and advised to choose a major. The student selects a major and receives a schedule that lists the coursework necessary to receive the degree. The student is encouraged to see their advisor every semester so that they can review the progress. However, keeping track of all the students is a daunting task for the advisors. Advisors can track a student's progress based on the annual reviews or meetings. This is more effective if universities have a manageable amount of students. Consider a scenario where a student is off track. The advisor is unaware of it unless the student has the annual review with the advisor. The advisor will come to know about this during the annual review. Say the

student completed couple of courses that do not account to the degree. The annual review reveals the same. Nevertheless, the time has already passed by. A very little can be done about this. The student tracking system avoids such a scenario. If such a system were in place there would have been an alert generated by the system to the advisor and the student as well. A corrective action will eliminate the unintended results such as student ending up taking courses not supposed to take. This system will provide access to student records instantly. Based on the guidelines of the program alerts are generated. By responding to these alerts students, advisors and administrative staff can work together instantly and avoid loss of valuable time. This will ensure corrective actions to bring the off-track students back on track. This will help student complete the program degree as planned.

The evaluation criteria for the existing systems at different universities are mainly the mandatory advisement meetings. Semester enrollment plans and program requirements are available online and as well as in hard copies. However, these methods are not effective mainly because most students lose track of the courses they have taken. They cannot evaluate their coursework. They are not able to abide by the mandatory requirements set by the programs as they lose focus due to lack of guidance and alerts.

The student tracking system presented in this paper will identify these gaps. It will provide the necessary information in order to keep students on track. This helps the student to graduate in the mandated time limit. Students who are off track will receive alerts. These alerts help the students, advisors and the administrative staff to determine what course of action is to be taken to get the student back on track.

CHAPTER 3

RELATED WORK

3.1. Background

In our current university system, the advisors and students follow the traditional methods for academic guidance. The student can access the information on the courses they have completed so far. The information presented makes it difficult for student to analyze it. Students remain unaware of if they are following the guidelines set by the department for completion of the program. The information available to the advisors is no different. The department administrators do not know the status of students with respect to the course completion.

I designed this application in an attempt to provide answers to all of these concerns. It tracks the graduate student progress in various Computer Science and Software Engineering graduate programs. This application shall provide lists of students who in a given year or semester did not fulfill any of the numbered requirements of their programs. It also provides the list of students who passed the comprehensive or qualifier in that year or semester, or students who have taken longer than the time mandated by the Graduate School to complete their plan of study. This application will also alert students as and when necessary. This application has the ability to provide the advisors with alerts and analysis of the course work accomplished by the students. The administrative staff of the department will be able to maintain the department level information such as the comprehensive qualifier results and attempts.

Anyone who enrolls in the courses offered by the university with a major from the Department of Computer Science is a student for this application. It stores basic information such as name, ID, contact information and the year and term the student

enrolled into the graduate program. Advisors are faculty members who review and provide feedback on the information available through the Graduate Student Progress Tracking System. The academic program will have specific guidelines about who should be an advisor. The advisor is associated with a student. This association grants access to the advisee academic records. The advisors will be able to view all of the student's information and will respond to the alerts and suggestion as the need arises. Administrators have access to all student academic information within the system. They have editing capabilities on certain information such as recording the comprehensive qualifier information.

CHAPTER 4

TOOL DESIGN AND PLANNING

I decided to develop an application to track the progress of graduate students in the Computer Science Department. I have organized the tool design and planning as follows:

Brainstorming:

I brainstormed to decide my approach. I researched for right tools. I charted the application control flow and application access levels. I decided on what data should be available to the chosen access levels. I learned how to organize and provide the alerts information. I decided on how to present the same. The common outcome for my brainstorming sessions were:

- making an outline for processes and data
- making diagrams for the flow of information
- putting the bits and pieces together and drawing a big picture

This helped me to visualize and put together my ideas. I put together drawings and pictures to see a big picture and acquire an organized outline list. The Figure 4.1 and Figure 4.2 represent the outcome of the brainstorming sessions. I put the information together to draw the big picture. This helps me with the design and better understanding of the requirements. It was easy for me to communicate with my advisor too. It keeps both the parties on the same page.

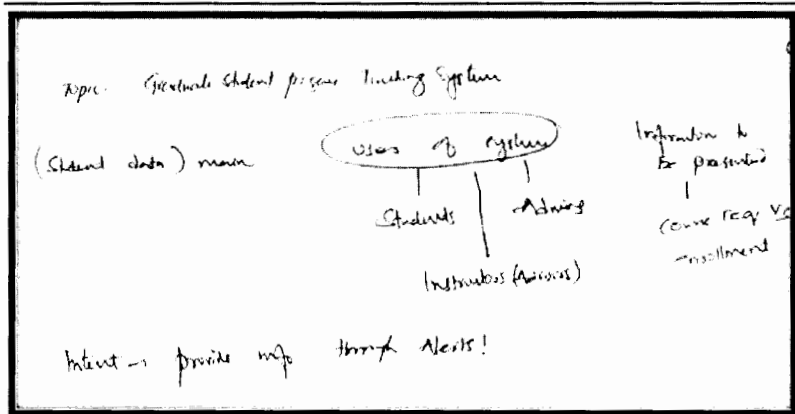


Figure 4.1. Notes from brainstorming sessions.

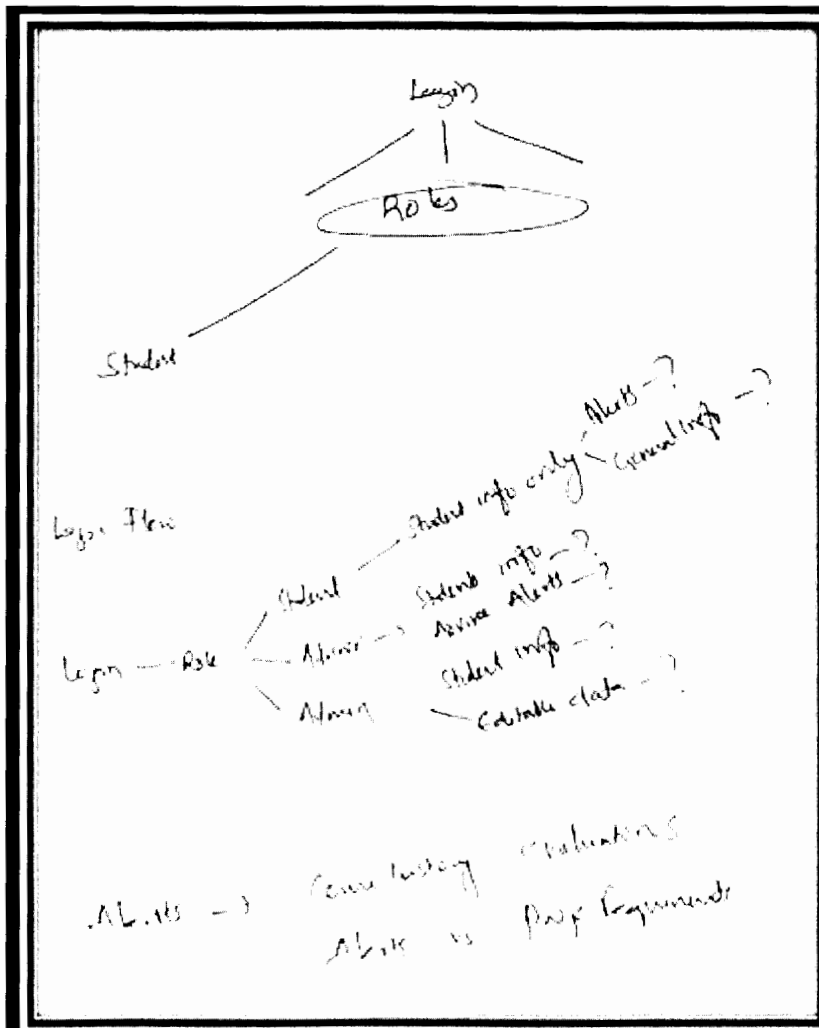


Figure 4.2. Notes from brainstorming session continued.

Background Study and Information Gathering:

I received insight into this topic through discussions with my advisor. I followed these discussions by researching related information on the internet. I studied the practices currently in use to find the loopholes that the new system would address. I searched knowledge bases such as ACM portals, which are journals for research addressing the flaws in existing practices.

Based on my brainstorming sessions and research I put my ideas together in the form of a prototype using Access database. I made a presentation to my advisor. The Figure 4.3, Figure 4.4 and Figure 4.5 illustrate the prototype using Access Database. I got feedback on the prototype.

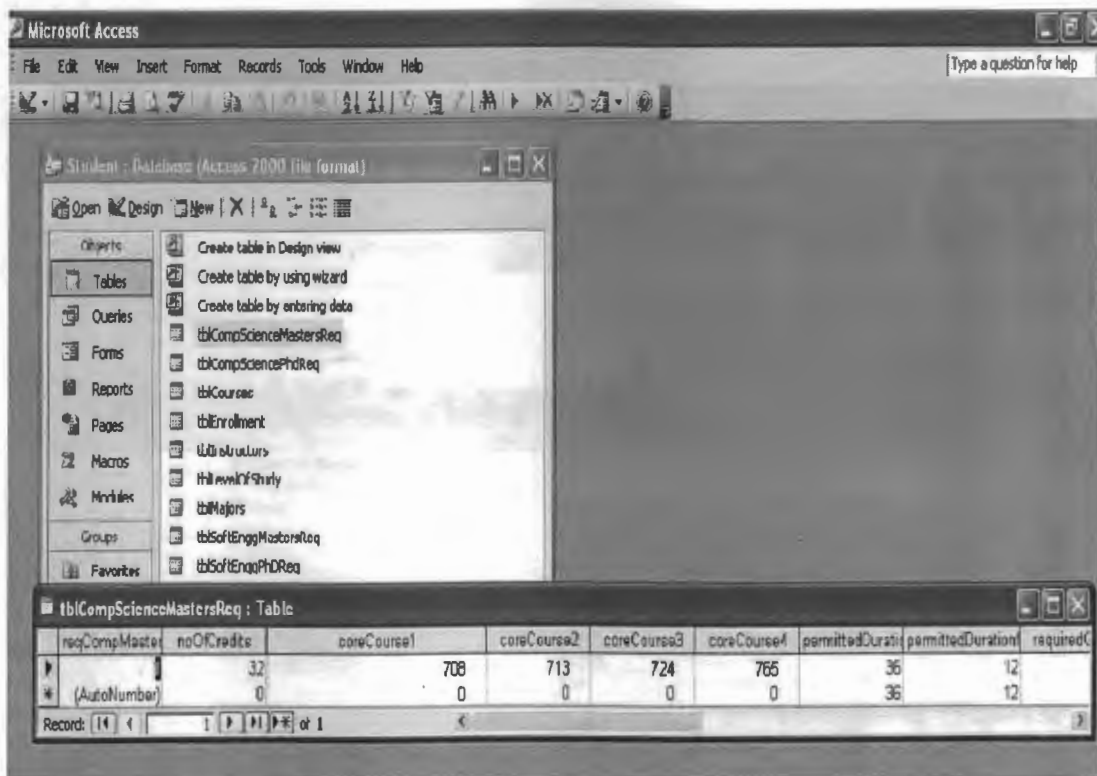


Figure 4.3. Prototype using Access database-1.

Microsoft Access

Student Database (Access 2000 file format)

tblCourses : Table

courseSNo	courseName	courseNumber	courseInstructor	credits
1	Foundations of Computer Networks	659	Xiaojiang (James) Du	3
2	Computer Forensics	676	Weiyei (Max) Zhang	3
3	Intro to Data Mining	679	William Perizo	3
4	Social Implications of Comp	689	Kendall Nygard	3
5	Software Design	716	Kenneth Magel	3
6	Software Construction	717	Dianxiang Xu	3
7	Software Testing and Debugging	718	Dianxiang Xu	3
8	Survey of AI	724	Yan Gu	3
9	Intro to Bioinformatics	732	Anne Denton	3
10	Database Systems Internals	786	William Perizo	3
11	Topics/Advanced Technology and Logistics	783	Kendall Nygard	3
12	Topics/Advanced Topics in Distributed Systems	783	Juan (Jen) Li	3
13	Topic/Information Retrieval and Web Search	783	Wei Jin	3
14	Topics/Wireless Networks/Mobile Compt	786	Weiyei (Max) Zhang	3
15	Sem/Educational Media	790	Brian M. Slator	1
16	Sem/Software Engineering	790	Kenneth Magel	1
17	Sem/Building Secure software	790	Dianxiang Xu	1
18	Sem/ Sensor Networks	790	Kendall Nygard	1
19	Sem/Wireless Security	790	Xiaojiang (James) Du	1
20	Master Paper	797	Staff	1
21	Master Paper Continued Registration	797R	Staff	1

Record: 14 of 79

Figure 4.4. Prototype using Access database-2.

Microsoft Access

Student Database (Access 2000 file format)

tblStudents : Table

studentSNo	firstName	middleName	lastName	emailAddress	studentID	enrollment
1	Pramodh Kurrar		Thalaji	Pramodh.Thalaji	0709740	Fall2006
2	Praveen Kumar		Pullagurula	Praveer.Pullagu	0600766	Spring2006
3	Senad		Cimic	Senad.Cimic@	0709345	Fall2006
4	Daniel		Acetuna	Daniel.Acetuna	0002346	Spring2000
5	Hamed		Ahmadi	Hamed.Ahmadi	0500234	Fall2005
6	Talukdar		Asgar	Talukdar.Asgar	0400456	Fall2003
7	Manu Kishore		Bhogadi	Manukishore.Bh	0823787	Fall2007
8	Boucheib		Falah	Boucheib.Falah	0723944	Spring2007
9	Karl Nils		Gurdensor	KarlNils.Gunder	0456735	Fall2004
10	Shanaka Nuradh		Herath	Shanaka.Herath	0534563	Spring2005
11	Ravi		Eda	Ravi.Eda@nde	0536633	Fall2005
12	Farzana		Jahan	Farzana.Jahar	0324456	Fall2003
13	Srikant		Kalki	Srikanth.Kalki	0321223	Spring2003
14	Lakshmi Mrudula		Kallam	Lakshmi.Kallar	0533234	Fall2005
15	Jerilyn Jawn		Kazack	Jerilyn.Kazack	0544444	Fall2005
16	Huma		Rizvi	Huma.Rizvi@rd	0769034	Spring2007
17	Mohammad		Gmadi	Mohammad.Gm	0402405	Spring2004

Record: 14 of 17

Figure 4.5. Prototype Using Access database-3.

Choosing the tools for implementation:

I continued with my brainstorming efforts. I revisited my initial understanding and compared it to the feedback I received. This helped me put together different ideas and figure out the next steps. The Figure 4.6 shows the outcome of the efforts in formulating the application logic flow. These sessions led me to some decision making. For example, I chose three-tier architecture for my application.

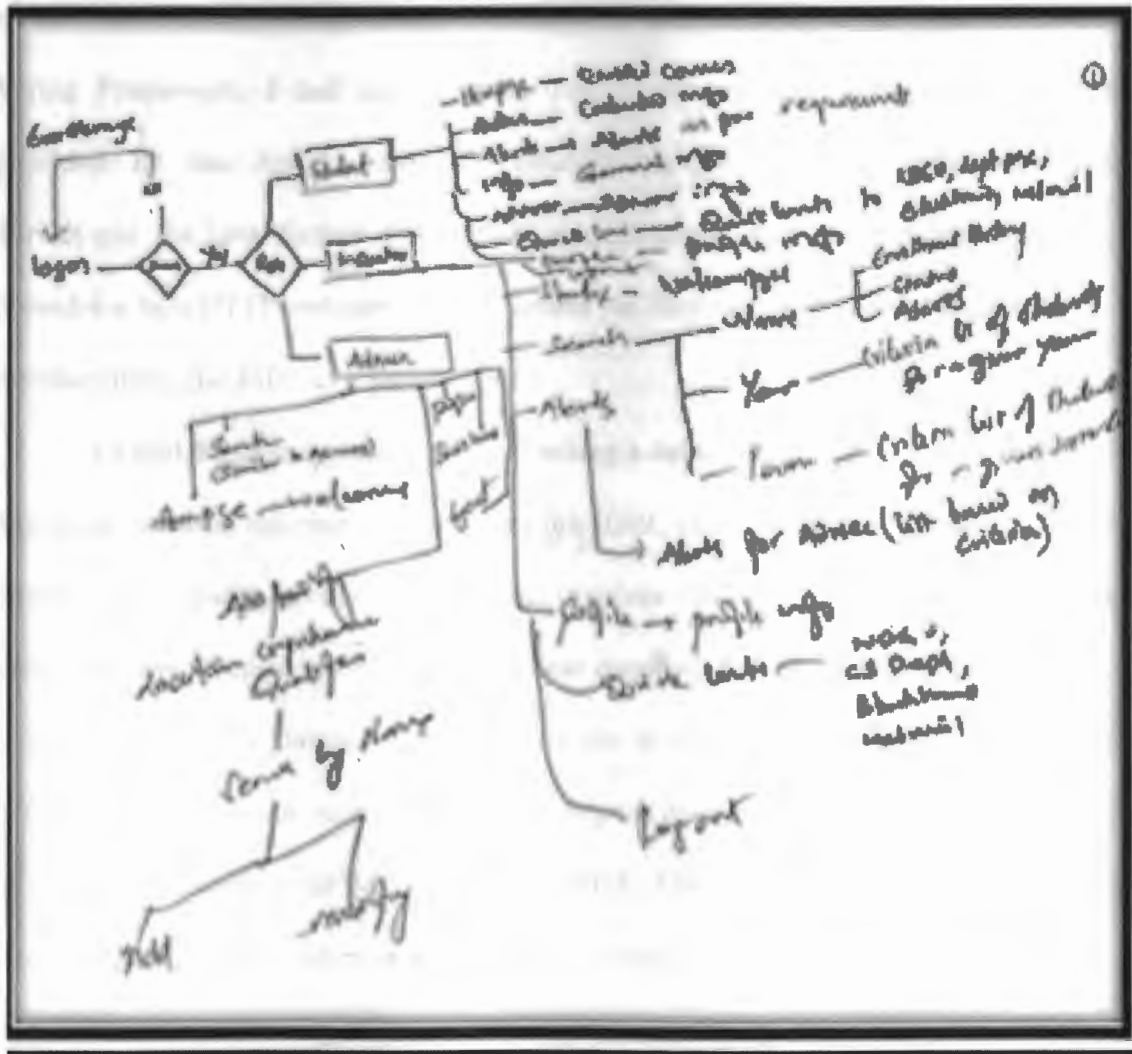


Figure 4.6. Hand charted application logic flow.

I decided to have a presentation layer (the graphical user interface), a middle tier that handled the logics and process flow and the backend database that houses all data for the application. The type of architecture I selected drove me to choose an integrated development environment. This kind of environment will let me manage my development, help me compile the code and deploy it simultaneously. Owing to these facts, I chose eclipse as Integrated Development Environment (IDE). The alternative was to use the Microsoft Visual Studio Developer. I was planning to use Java Struts with Spring Framework. I had used Apache Tomcat as my application server that was provided by the Apache Software Foundation (ASF). Tomcat implements Java Servlet and the Java Server Pages (JSP) specifications from Sun Microsystems and provides a Java HTTP web server environment for Java code to run. I developed the user interface using the JSP (Java Server Pages).

To find the best approach when accessing a database from Java is a daunting task. The most common solution is to program the JDBC (Java Database Connectivity) APIs directly. This results in complex code that involves calling the JDBC connection every time. We have to open and close the database connection each time database is called. I selected the IBATIS Database Layer to assist me in this area. IBATIS is a collection of open-source components that assist Java developers with data access. The most important of these components is the SQL Maps framework. This framework provides an efficient way to graph database values to Java objects by using XML configuration files.

I chose Oracle 10g XE as my database. I had the option of using MS Access or SQL XE. As part of my current job responsibilities, I work with the SQL database, so I took this opportunity to learn Oracle.

Based on the prototype review, requirement analysis, research analysis and brainstorming sessions I created a student database schema. This schema will hold the information required by the application. The relational database management system drives the schema. I ensured that the data was not redundant. I worked on dependencies and constraints. I included this information in the paper and constructed the schema. I have created the database tables and defined the constraints accordingly using the same schema.

I intend to manage the application server from eclipse IDE. To have this ability I had to set up eclipse IDE with the Tomcat application server. I downloaded the plug-in for managing Tomcat services through IDE. I used the Ant Build plug-in with eclipse. This helped me compile and execute the program to create a WAR file. I deployed the WAR file using Tomcat application server. Based on the process flow, control flow diagrams and hierarchy designed I have coded the logic for the control flow within the application. The actual screen design took place based on the mock screens. I compiled and deployed the application.

I have documented the case scenario for the application and presented it in the sections to follow. This helped me complete the project with a disciplined project management process. It encouraged frequent inspection and adaptation, self-organization and accountability, a set of engineering best practices intended to allow for the rapid delivery of high-quality software. I followed an approach that aligned the development in line with the requirements. I made certain assumptions and stated the conditions under which a student will be tracked using the graduate student progress tracking systems. These assumptions form the basis for the basic design as explained below.

Establishing the basic design:

The tracking system tracks the following students:

- a full time graduate student who is enrolled into a master's program offered by the Department of Computer Sciences
- only the course history of the courses offered by the Department of Computer Science
- only approved courses for coursework will be considered for the evaluation of degree completion
- the comprehensive qualifier data is dependent upon the entries made by the administrator, who shall be responsible for the data quality
- the tool implementation is solely for the demo purpose of serving as proof of the concept

Appendices D and E lists the requirements used for evaluation of the student's progress using this tool.

CHAPTER 5

TOOL IMPLEMENTATION

5.1. Architecture

The application uses a three-tier architecture model. There are three or more interacting tiers in three-tier architecture or multi-tier architecture. Each of these tiers has its own specific responsibilities. The Figure 5.1 represents the student tracking system architecture.

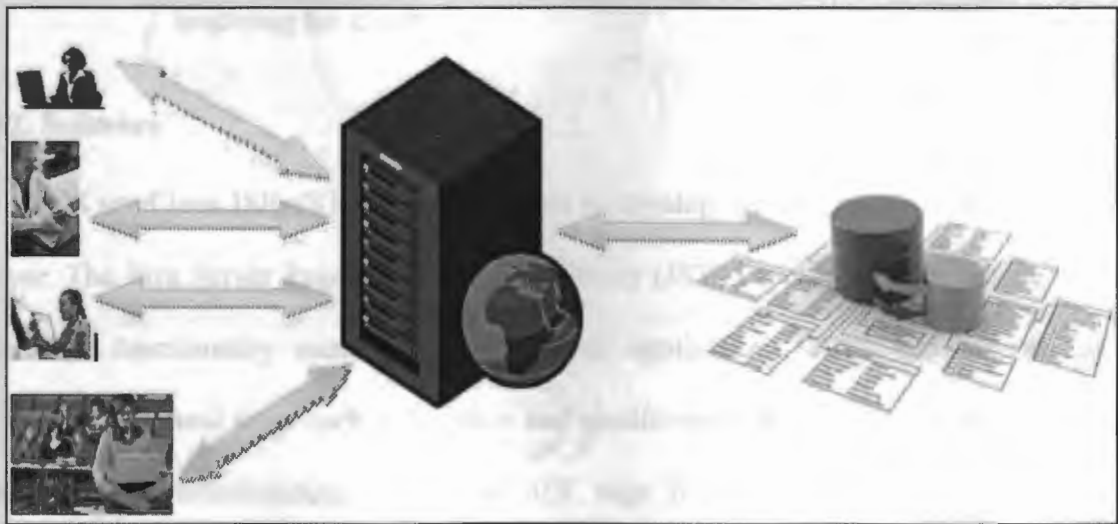


Figure 5.1. Student tracking system architecture.

5.1.1. The presentation layer or the client layer

The client contains the presentation logic. It includes simple controls and user input validations. This application layer is also known as a thin client.

5.1.2. The application server or the middle layer

The middle tier is also known as the application server, provides the business processes logic and the data access.

5.1.3. The Database layer or the tier3 layer

This is the data server that provides the business data.

Some of the advantages of three-tier architecture include:

- It is easier to modify or replace any tier without affecting the other tiers.
- Separating the application and database functionality means better load balancing.
- Adequate security policies are enforced within the server tiers without hindering the clients.

5.2. Software

I used Java JSP, JSTL and Java scripts to develop the user interface or the client layer. The Java Server Pages Standard Tag Library (JSTL) encapsulates as simple tags the core functionality common to many Web applications. JSTL has support for common structural tasks such as iteration and conditionals, tags for manipulating XML documents, internationalization tags and SQL tags. It also provides a framework for integrating existing custom tags with JSTL tags. I used struts with springs and have used ant build to deploy the application; both integrated into the eclipse. The Apache Tomcat application server is my web application server. I added it to the eclipse using available addins. I could manage all the services from one place.

I designed the database using software Oracle 10Gi Express edition. Oracle Database 10g Express Edition (Oracle Database XE) is an entry-level, small-footprint database. Based on the Oracle Database 10g Release 2 code base, it is free to develop, deploy and distribute. It is fast to download and simple to administer. I used the eclipse

IDE to develop, debug, compile and deploy my code. The Figure 5.2 illustrates the use of Eclipse tool. I have used the SQL developer as a client to connect to the Oracle 10Gi database server. The same tool has been illustrated in the Figure 5.3

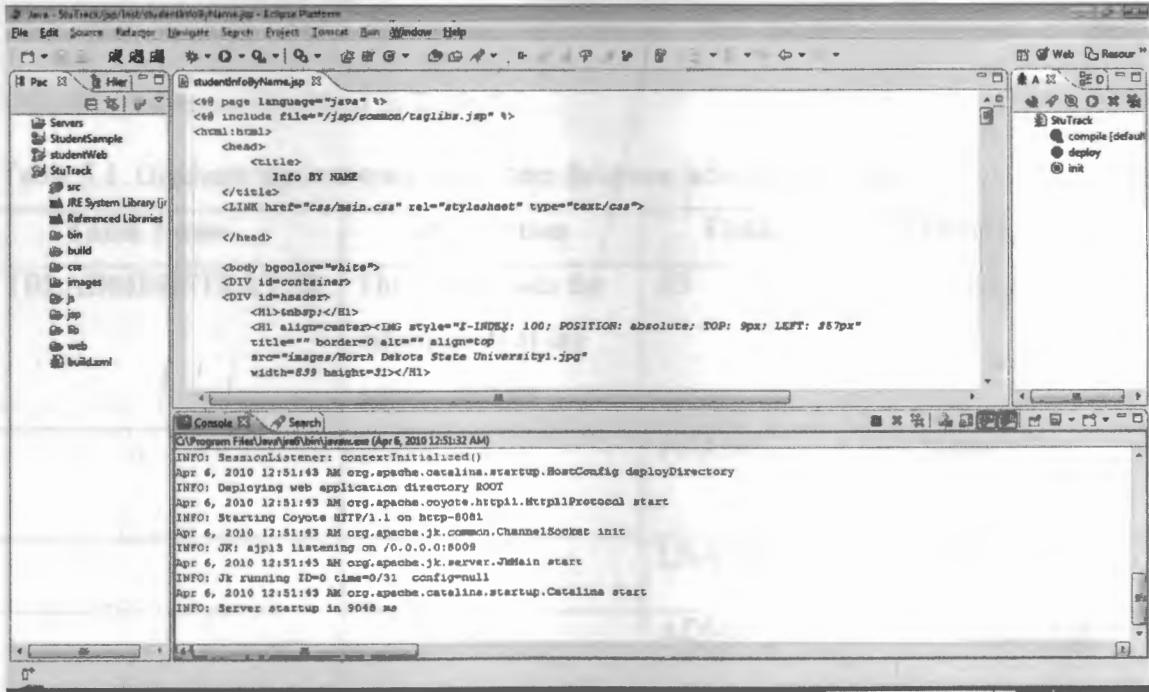


Figure 5.2. Eclipse IDE with Tomcat integration.

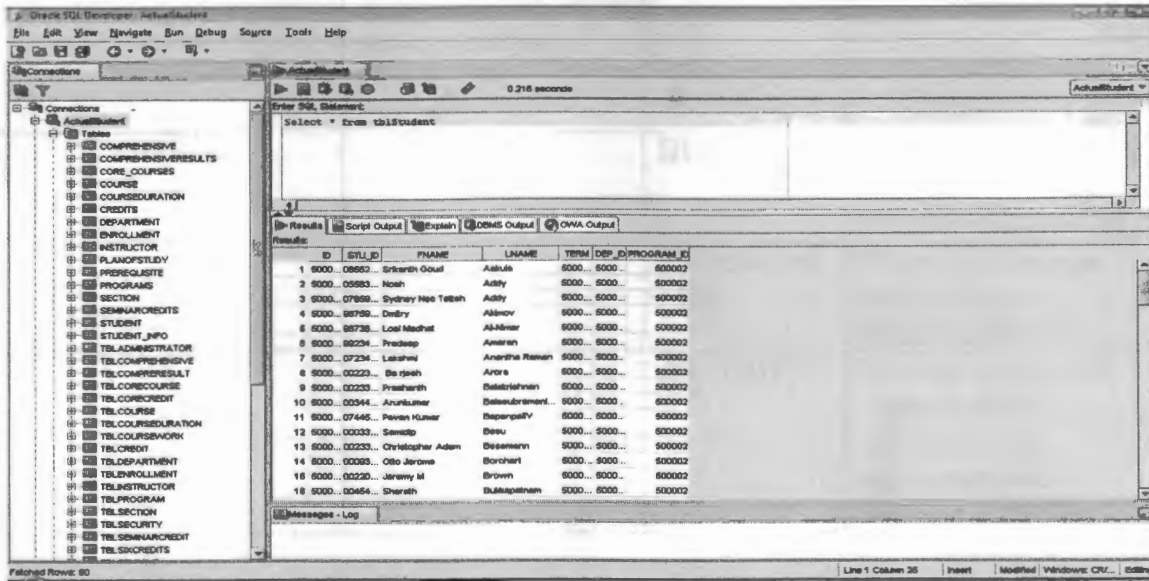


Figure 5.3. SQL Developer.

5.3. The Underlying Database System

As discussed above the project includes several components. This includes the actual database, the management application, associated help files, data files and image files. The database stores the actual information. The database comprised of the following tables as described in Table 5.1.

Table 5.1. Graduate student tracking system database table information

Table Name	Description	Fields	Description
TBLADMINISTRATOR	This table holds the information of the administrators	ID	Primary Key (ID)
		FNAME	First Name of the Admin
		LNAME	Last Name of the Admin
		ADMIN_ID	NDSU assigned ID
TBLCOMPREHENSIVE	This table holds the information of the number of attempts allowed by the department for clearing the comprehensive	ID	Primary Key (ID)
		DEP_ID	This is the foreign key from the TBLDEPARTMENT
		PROGRAM_ID	This is the foreign key from the TBLPROGRAM
		ATTEMPTS	The max number of comprehensive attempts as per department and program guidelines

Table 5.1. (continued)

Table Name	Description	Fields	Description
TBLCOMPRERESULT	This table holds the information of the comprehensive results along with the passing criteria	ID	Primary Key (ID)
		STU_ID	This is the foreign key from the TBLSTUDENT
		COMPRE_TERM	This is the foreign key from the TBLTERM
		ATTEMPT_NO	This indicates the comprehensive attempt number for a given student
		RESULT	The result of the comprehensive qualifier
		CRITERIA	The criteria for passing is entered here
		CUTOFF	The cut off for the comprehensive qualifier
TBLCORECOURSE	This table holds the core course information as mandated by the department	ID	Primary Key (ID)
		DEP_ID	This is the foreign key from the TBLDEPARTMENT
		PROGRAM_ID	This is the foreign key from the TBLPROGRAM
		REQUIRED	Indicates if the course is optional core course of required
		CRS_ID	This is the foreign key from the TBLCOURSE

Table 5.1. (continued)

Table Name	Description	Fields	Description
TBLCOURSE	This table holds the course information	ID	This table holds the course information
		CRS_CODE	The code assigned by the department for the course offered
		CRS_TITLE	The Title for the course offered by the department
		CRS_DESC	The description of the course offered by the department
		CRS_CREDITS	The credits associated with the course offered by the department
		DEP_ID	This is the foreign key from the TBLDEPARTMENT
TBLCOURSEDURATION	This table holds the course duration information	ID	Primary Key (ID)
		DEP_ID	This is the foreign key from the TBLDEPARTMENT
		PROGRAM_ID	This is the foreign key from the TBLPROGRAM
		DURATION	The maximum allowed time duration for completing a given degree program (in Months)
TBLCOURSEWORK	This table holds the plan of study, dissertation and thesis/paper information	ID	Primary Key (ID)
		STU_ID	This is the foreign key from the TBLSTUDENT
		PLAN_OF_STUDY_TERM_ID	This is the foreign key from the TBLTERM

Table 5.1. (continued)

Table Name	Description	Fields	Description
		PAPER_THESIS	This will indicate if a student will present a thesis or paper for the completion of degree
		DISSERTATION_TERM_ID	This is the foreign key from the TBLTERM
		ADVISOR	This is the foreign key from the TBLINSTRUCTOR. This will indicate the name of the advisor for the student
TBLCREDIT	This table holds the credit requirements for the programs offered by the departments	ID	Primary Key (ID)
		DEP_ID	This is the foreign key from the TBLDEPARTMENT
		PROGRAM_ID	This is the foreign key from the TBLPROGRAM
		CREDITS	The total credits required by a given program is stored here
TBLDEPARTMENT	This table holds the department information	ID	Primary Key (ID)
		DEP_CODE	Department codes say 'CSCI' for computer science etc.
		DEP_DESC	The description of the departments
TBLENROLLMENT	This table holds the student enrollment information	ID	Primary Key (ID)
		STU_ID	This is the foreign key from the TBLSTUDENT
		SEC_ID	This is the foreign key from the TBLSECTION

Table 5.1. (continued)

Table Name	Description	Fields	Description
TBLINSTRUCTOR	This table holds the instructors information	ID	Primary Key (ID)
		INS_ID	This is the unique id assigned by NDSU
		INS_FNAME	Instructor First Name
		INS_LNAME	Instructor Last Name
		DEP_ID	This is the foreign key from the TBLDEPARTMENT
TBLPROGRAM	This table holds the information about the programs offered by departments	ID	Primary Key (ID)
		PROGRAM_CODE	The program code say MSE for masters in software engineering
		PROGRAM_DESC	The description of the program
		DEP_ID	This is the foreign key from the TBLDEPARTMENT
TBLSECTION	This table holds the sections information which combines the courses offered with the term and instructor information	ID	Primary Key (ID)
		SEC_BLDG	The building location say IACC etc
		SEC_ROOM	The class room number where the instructor led sessions will take place
		SEC_TIME	The time during which a course is offered
		CRS_ID	This is the foreign key from the TBLCOURSE

Table 5.1. (continued)

Table Name	Description	Fields	Description
		INS_ID	This is the foreign key from the TBLINSTRUCTOR
		TERM_ID	This is the foreign key from the TBLTERM
TBLSECURITY	This table holds the login credentials	ID	Primary Key (ID)
		PASSWORD	Holds the password supplied by the user
TBLSEMINARCREDIT	This table holds the seminar credit information	ID	This table holds the seminar credit information
		DEP_ID	This is the foreign key from the TBLDEPARTMENT
		PROGRAM_ID	This is the foreign key from the TBLPROGRAM
		CREDITS	The required number of seminar credits as per the program guidelines formulated by the department
TBLSIXCREDITS	This table holds the course information where a software engineering graduate student shall complete minimum six credit requirement	ID	Primary Key (ID)
		CRS_ID	This is the foreign key from the TBLCOURSE
TBLSTUDENT	This table holds the basic information about the students	ID	Primary Key (ID)
		STU_ID	This is the unique id assigned to a student by NDSU
		FNAME	Student First Name
		LNAME	Student Last Name

Table 5.1. (continued)

Table Name	Description	Fields	Description
		TERM	Student Term
		DEP_ID	This is the foreign key from the TBLDEPARTMENT. This will hold the department information associated with the student.
		PROGRAM_ID	This is the foreign key from the TBLPROGRAM. This will hold the program into which a student is enrolled.
TBLTERM	This table hold the term information	ID	Primary Key (ID)
		TERM	The term description

CHAPTER 6

SCENARIO

The application of student tracking system has three different groups based on their roles:

1. Student
2. Advisor
3. Administrator

Based on the above-mentioned groups the users will have different levels of access to data. Security rights are granted based on the login credentials that are associated with the groups.

In the following section, I will discuss the case scenario that will demonstrate the use of the tracking system.

6.1. Student Login

The student who will access the student tracking system using the URL provided. Each eligible student has a valid login id and password. This typically would be the student id assigned by the university. A successful login will let the student access the information within the application. An unsuccessful login will ask for the student's credentials again. We call this an invalid login. The page is redirected to the login screen and a relevant message associated with invalid login appears. There will be helpful links to reset the page. The links to *forgot password* will give some hints on password. If this does not remind the user the password then they have to contact the administrator of the tool.

6.1.1. Student Home Page

The successful login will bring the student to the home page. This page will list the courses in which the student has enrolled. The application home page has several menu options available for navigation. These menu links will guide the student through the application. These menu options are discussed below in detail.

6.1.2. Student –Status Menu

The status menu will provide the student with the summary of the course requirements he/she has completed so far. These typically include:

- a. Core Courses Completed
- b. Total Credits Completed
- c. Seminar Credits Completed
- d. Comprehensive Results/attempts/criteria/cutoff
- e. Other Program specific Requirements
- f. Plan of Study Status
- g. Defense Status
- h. Advisor Status/Information

6.1.3. Student - Alerts Menu

The alerts menu will guide the student toward the program requirements into which he/she is enrolled. These alerts are the result of comparing the courses completed against the program requirements. These typically include:

- a. Total left over credits
- b. Total seminar credits left over
- c. Core courses to be completed
- d. Comprehensive, if not attempted or passed

- e. Other program specific requirements
- f. Plan of study, if not approved by the Graduate School
- g. Defense status, if degree is not defended
- h. Advisor, if one is not chosen

6.1.4. Student Advisor Info

The student menu has a menu item called *Advisor*. This will display the advisor information to the student. This will provide the student with information and contact details that will allow student to communicate with advisor. This will prompt timely communication between the advisor and student.

A student is assigned a default advisor if he does not pick one. The assignment of the default advisor depends on the program into which a student is enrolled. If a student is unaware of the default advisor assignment, they can click on the alerts menu and find the information.

6.1.5. Student – Profile Menu: Student Profile Display

This menu option will display the student's personal information. The information available on this page will reflect the student's program, the term he/she enrolled into the program and the student's name and contact details. This is the student's personal information and student will only have read access. These pages are for information purposes only.

6.1.6. Student - Quick Links

This menu section will provide the student with the opportunity to navigate to the frequently used applications and URL's. These include the following:

- a. NDSU Home Page : This link will navigate to the NDSU home page

- b. Department Home Page : This link will navigate to the Computer Science Department homepage
- c. Web Mail : This link will navigate to the NDSU Web Email login page
- d. Black Board : This link will navigate to the NDSU Black Board login page

6.1.7. Student – Logout

This menu section will end the student login session and will log him/her out of the application.

6.2. Advisor Login

The login page for the student tracking system for an advisor is no different from the student page. The login credentials will decide the access to different data elements and menu options. The advisor is provided with the login credentials. The successful login will take the advisor to home page. The home page will host the information about the student tracking system and its use. There are different menu options available on this page. These menu items will guide the advisor with this application.

6.2.1. Advisor MyStudents Menu

The *MyStudents* menu will display the list of names of the advisees for whom the logged-in instructor is the advisor. The application will show a summarized view of the status of the student when the advisor clicks on the advisee name:

- Student List
- Hyperlink to Student Status page

The student list displayed is based on the association of the advisor with the student. Advisor can only view the list of his advisees. This information includes the first

name, last name and the student id of the student. The list is a hyper link to the summarized view of the student status.

6.2.2. Advisor Search Menu

This menu option will provide the instructor with search capabilities in three different criteria: viz. by student, by year and by semester term and year. The intention of providing these options is to let advisor have access to the student information organized by different criteria. This criterion summarizes the information at the department level. As said earlier, the instructor will have three different search options with which they can access the information:

1. Search by Student Name
2. Search by Year
3. Search by Term and Year

6.2.2.1. Search by student name

The instructor will have with the option of searching for a student by name. The results of the search criteria will display all matching names. The instructor will have the capability to pick the student that he/she would like to track. This kind of search criteria will let the advisor focus on one student at a given time.

The instructor will type in the student name to search. The application will return all matches. In case of no match, the application will return a message saying that there was no match. The student's list displayed is a hyperlink. There will be three categories for information retrieval:

- the list of all courses the student has completed
- the student's summarized information with respect to the program requirements

- the student alerts information with respect to the program requirements

Based on the sensitivity and impact of the alerts, the advisor can select a suitable mode of communication with the student. This can be through email, a meeting, or a phone call. This reduces the time taken for communicating the corrective actions. This will render the much-needed guidance to the student at the right time. The aim is to get the student back on track

6.2.2.2. *Year*

The search by year will benefit the advisors by providing the sliced information on a yearly basis. The slicing is based off the year parameter. The advisor chooses the year from the drop down. This kind of search will enable the advisors to filter the students based on a given criteria.

The criteria can typically be the list of students who took the comprehensive qualifier exam in a given year. The criteria could be the list of the students who did not meet the credits requirement in course work as a whole. The advisor might be interested in the list of students who in a given year did not fulfill the seminar credit requirement as specified by the program. The criteria can also include the list of students who did not attend the comprehensive qualifier exam or those who did not get approval from the Graduate School for the plan of study as per department guidelines. Similarly, the instructor can click on a link for the list of students who in a given year have appeared for the comprehensive qualifier exam. The instructor or the advisor will have access to a link that will display the list of all students who have defended their degree in a given year. The results of these criteria will assist advisors through the department level information available at their fingertips. The details of the display include the student names, IDs, term and year. The rest of the information pertains to the criteria selected.

Based on the associated program the credit requirements are typically checked for the following categories:

- Complete Credits for Course Work (Computer Science Graduate Program)
- Complete Credits for Course Work (Software Engineering Program)
- Seminar Credit Requirements for Computer Science Graduate Program
- Seminar Credit Requirements for Software Engineering Program

The advisors will have the query criteria for finding out the students who have their plan of study approved by the Graduate School, but who took more than the mandated time limit. The list will be limited to students who completed filing their plan of study during the given year. Similarly, the instructor has a list of students who are yet to get approval for the plan of study during a given year.

6.2.2.3. Tab: Term Cum Year

The instructor search option provides the ability to search by the term cum year. The semester information is presented as a drop down option. The user will typically select the term from the options provided. The selection of the term will submit the selection to navigate the page to a screen where advisors are presented with the links to different criteria.

Different queries organize the data in different dimensions. They form various grouping based on the semester parameter chosen by the advisor. The query option includes the following:

- the list of students who took the comprehensive qualifier exam in a given term

- the list of the students who did not meet the credits requirements in course work the list of the students who did not meet the seminar requirements as specified by the program
- the list of students who did not attend the comprehensive qualifier exam
- list of the students who did not get approval for the plan of study as per program guidelines
- the list of students who have defended their degree in a given semester
- the list of students who defended a paper or thesis

This will give the instructors a report of the student's progress at the department level for a given semester. The details of the display include the student names, IDs, the term. The student list for credit requirements will display when an instructor clicks to find out the list of people who have not fulfilled the total credit requirements for computer science program in a given term. There is a similar option provided for the software engineering students.

Based on the program the student is enrolled in, the credit requirements are typically checked for the following categories:

- Complete Credits for Course Work (Computer Science Graduate Program)
- Complete Credits for Course Work (Software Engineering Program)
- Seminar Credit Requirements for Computer Science Graduate Program
- Seminar Credit Requirements for Software Engineering Program

The advisors will have the query criteria for finding out the students who got their plan of study approved, but took more than the mandated time limit. The list will be limited to the students who filed their plan of study during the given term. Similarly, the

instructor can access the list of students who have had their plan of study approved during a given term.

6.2.3. *Advisor – Alerts menu*

The advisor also has an alerts menu. The alerts menu for the advisor is associated with the data pertaining to his/her advisee only. The information presented in this menu section will provide summarized alerts for advisees at their fingertips. Based on the sensitivity of the alert, the advisor can work out a plan for the student and bring them back on track towards the program completion.

The menu *Alerts* will lead to a set of query options similar to the search options discussed above. For a given advisor, there are listed links that lead to a list of advisees who have not completed the core course requirements. This information is very critical. Analyzing the information based on these alerts, the advisor can decide to communicate with the concerned student. This kind of timely action will avert any disastrous or damage causing scenarios.

The advisor can check on the students who have not completed the total number of required credits toward their degrees. They get alerts on students who have not fulfilled the seminar requirements. The advisor has the ability to find the list of advisees who are yet to take the comprehensive qualifier exam. The advisors or instructors will have the query criteria for accessing the list of students who have received approval for their plan of study from the Graduate School, but took more than the mandated time limit. This list will be limited to the list of advisees who had their plan of study approved. The instructor can access the list of students who have approval for their plan of study from Graduate School.

The student is assigned a default advisor until he/she chooses one. If the logged in advisor is the default advisor for a set of students, he/she will be able to see the same list. The student remains under the default advisor's guidance until the student picks his/her advisor. This will ensure that the default advisor has access to all the data pertaining to the student who needs the guidance.

6.2.4. Advisor Profile

This menu option will display the advisor's personal and contact information.

6.2.5. Quick Links

This menu section will provide the advisor with the opportunity to navigate to frequently used applications and URLs. These include the following:

- a. NDSU Home Page : This link will navigate to the NDSU home page
- b. Department Home Page : This link will navigate to the Computer Science Department homepage
- c. Web Mail : This link will navigate to the NDSU Web Email login page
- d. Black Board : This link will navigate to the NDSU blackboard login page

6.3. Administrator Login

The administrator login page is the same as that of the advisor and student. Upon successful login, the administrator will see the home page. The home page will have the information about the tool and the intended use of the application.

6.3.1. Admin Search Menu

This menu option will provide the administrator with search capabilities based on three different criteria: viz. by student, by year, or by year and by semester term. The intention of this option is to allow administrators access to student information across the

department. As said earlier, the administrator will be given three different search criteria by which they can access the information:

1. Search by Student Name
2. Search by Year
3. Search by Term and Year

The search capabilities are no different from the ones discussed in the advisor section.

6.3.1.1. Search by student

The search by student will provide the following information about the student:

- Student Course History
- Student Course Summary
- Student Alerts

6.3.1.2. Search by Year

This search will provide an option to select a year and pool the following information:

- All students who completed their plan of study in a given year
- All students who appeared for the comprehensive in a given year
- All students who defended their degree in a given year
- All students who fall short of the total credit requirements in a given year
- All students who fall short of the seminar credits in a given year
- All students who got their plan of study approved but took more than the mandated time limit in a given year

6.3.1.3. Search by Term

This search will allow for the selection of term cum year from the dropdown menu and provide the following information:

- All students who completed their plan of study in a given term and year
- All students who appeared for the comprehensive exam in a given term and year
- All students who defended their degree in a given term and year
- All students who fell short of the total credit requirements in a given term and year
- All students who fell short of the seminar credits in a given term and year
- All students who received approval of their plan of study, but took more than the mandated time limit in a given term and year

6.3.2. Add/Modify/Maintain Comprehensive Qualifier Information

The administrators have access to ***add/modify*** menu item. The intent is that they maintain the comprehensive qualifier information for the Computer Science Department. They will be responsible for maintaining the comprehensive qualifier information. The administrators will have access to the student records for maintaining their qualifier information. Clicking the menu item for ***add/modify*** will display the search option for a student. The administrator will have the ability to search for a student whose record needs to be either modified or added to the comprehensive qualifier data. The search will retrieve all the matching records. If no match is found, the relevant message is returned.

The search will retrieve the names of the students that match the search key. The list will include the student name and ID. A clickable name will lead to either an update page or an add page. This is dependent on the criteria into which the selected record falls. A student will fall under the ***add*** criteria if he/she never took the comprehensive qualifier exam. Such a scenario will occur when a record should be added to the comprehensive data table. A student will fall into the ***modify*** criteria when he/she already exists in the

comprehensive data table. This is a scenario where the student record needs to be updated.

ADD: The admin clicks on a student who has appeared in the comprehensive qualifier for the first time. The link will lead to the page where admin can register the comprehensive result. The admin is directed to click **ADD** on the screen. Clicking on the **ADD** link will lead to a page where the administrator will add the comprehensive information for the student. The admin will click on add to add this record to the comprehensive data table. This concludes the addition of new records to the comprehensive data table.

MODIFY: The admin clicks on a student who has already appeared in the comprehensive qualifier. The link will lead to the modify page. The admin will be directed to click **MODIFY** on the screen. A click on the student name will result in the student record with the existing information about the qualifier exam. There will be an option to **MODIFY**. The admin will be able to update the qualifier information relevant to the student record accessed.

Both **ADD** and **MODIFY** will have the drop down menu for selecting the term and the result of pass or fail for a given record. After filling in the information about the qualifier exam, either in the *add* or *modify* option, the admin will commit the action. This concludes the **ADD** and **MODIFY** for admin.

6.3.3. Admin Profile

This menu option will display the administrator personal and contact information.

6.3.4. *Quick Links*

This menu section will provide the administrator with the opportunity to navigate to the frequently used applications and URL. These include the following:

- NDSU Home Page: This link will navigate to the NDSU home page
- Department Home Page: This link will navigate to the Computer Science Department homepage
- Web Mail: This link will navigate to the NDSU Web email login page
- Black Board : This link will navigate to the NDSU Black Board login page

CHAPTER 7

CONCLUSIONS, LIMITATIONS AND FUTURE WORK

7.1. Results

It is important to recall that the student tracking system has been proposed, designed, developed and implemented. The aim of this implementation is to accomplish the task of keeping the student and advisor informed about a student's progress towards a degree program. We achieve this by drawing the alerts from the course work being completed by the student. These alerts are based off the program requirements specified by the department.

We have seen that the student is able to track the progress using this tool. The tool has the built-in intelligence to provide summarized information. The tool has the alerts section that will keep the student informed of the corrective actions to be taken to satisfy the program requirements. The additional information such as the program requirements will help the student gain useful information in one place. This will benefit the student's progress toward program completion. Utilizing this tracking system the student can gain more focus on the program. There by fulfill the program requirements.

The student tracking system helps advisors track their advisees with real time academic information. This tool collects data in different dimensions of and puts them together to draw intelligent alerts. These alerts will assist the advisor in tracing the students who need guidance and in what areas of concern. Thus eliminating the consequences of delayed corrective measures to put the advisee back on track.

The intended results from the students who use this system shall be encouraging. The walk-in advising visits shall increase, communication shall increase and the scheduling of regular advisor/advisee appointments shall increase. It helps the students and advisors with the information and alerts. The data administration for administrators who maintain department level information pertaining to the graduate students is simplified.

7.2. Conclusions

I proposed, developed and implemented a tool for tracking the progress of the graduate students in the Computer Science Department. When used, this tool will provide lists of students who in a given year or semester did not fulfill any of the numbered requirements of their programs, who passed the comprehensive or qualifier exams in that year or semester, or who have taken longer than the Graduate School's mandated time limit to complete their plan of study. Such data will benefit the students, advisors and administrators by monitoring and guiding the progress of the student through the fulfillment of his/her program requirements.

A system of this nature should significantly improve the timing and quality of contact between advisors and students. This will change the way an advisor can connect and work out a plan with the student in a timely manner in order to keep challenges from becoming disasters.

7.3. Future Work

When integrated with PeopleSoft, the application will show the real data. This will help advisors and students with information and alerts. It will also significantly change the way a student's progress is tracked. The application is scalable and can be

extended to include all the programs offered by the Computer Science Department. Upon successful implementation of this student progress tracking system, it can be extended to all other departments. I anticipate that this program will have a significant impact on the way students and advisors interact. It will help students work towards their goal of graduation. It will increase the number of students who graduate in the mandated time limit by meeting all the program requirements. A consistent indicator of a graduate program's quality is the timely completion of that program. The culture of timeliness allows programs to be productive and efficient.

BIBLIOGRAPHY

- [1] *NDSU Academic Catalog*, North Dakota State University, Fargo, ND, 58105
- [2] *Department of Computer Science*, North Dakota State University, Fargo, ND, 58105
- [3] Date, C. J., and H. Darwen, *A Guide to the SQL Standard*, Addison-Wesley, Reading, MA, 1994.
- [4] Newman, W. M., and M. G. Lamming, *Interactive System Design*, Addison-Wesley, Workingham, England, 1995.
- [5] Silberschatz, A., H. Korth, and S. Sudarshan, *Database System Concepts*, McGraw-Hill, New York, 1997
- [6] OPAAGA, *Student Tracking Systems Can Be Used to Enhance Graduation and Retention Rates*, Report No. 06-48, May 2006
- [7] Coffey, J. C.; Palmer, J. "Implementing Student Tracking Systems at Community Colleges." Washington, D.C.: American Association of Community and Junior Colleges, forthcoming 1990.
- [8] "Computers and Student Flow/Tracking Systems," by Judith W. Leslie
- [9] *ConnectND, Peoplesoft*, North Dakota State University, Fargo, ND, 58105
- [10] [Ibatis] <http://ibatis.apache.org/java.cgi>
- [11] [Eclipse] <http://www.eclipse.org/>
- [12] [Oracle] <http://www.oracle.com/index.html>

APPENDIX A

Figure A.1. Login page.



Figure A.2. Student invalid login.

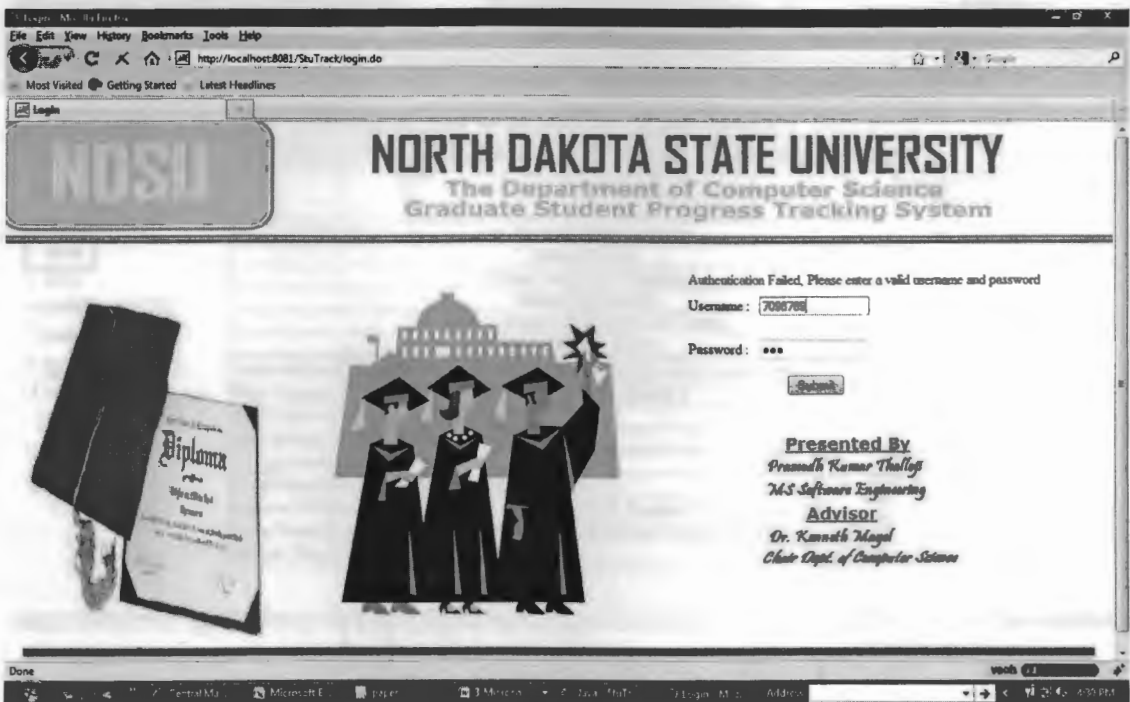


Figure A.3. Student login homepage.

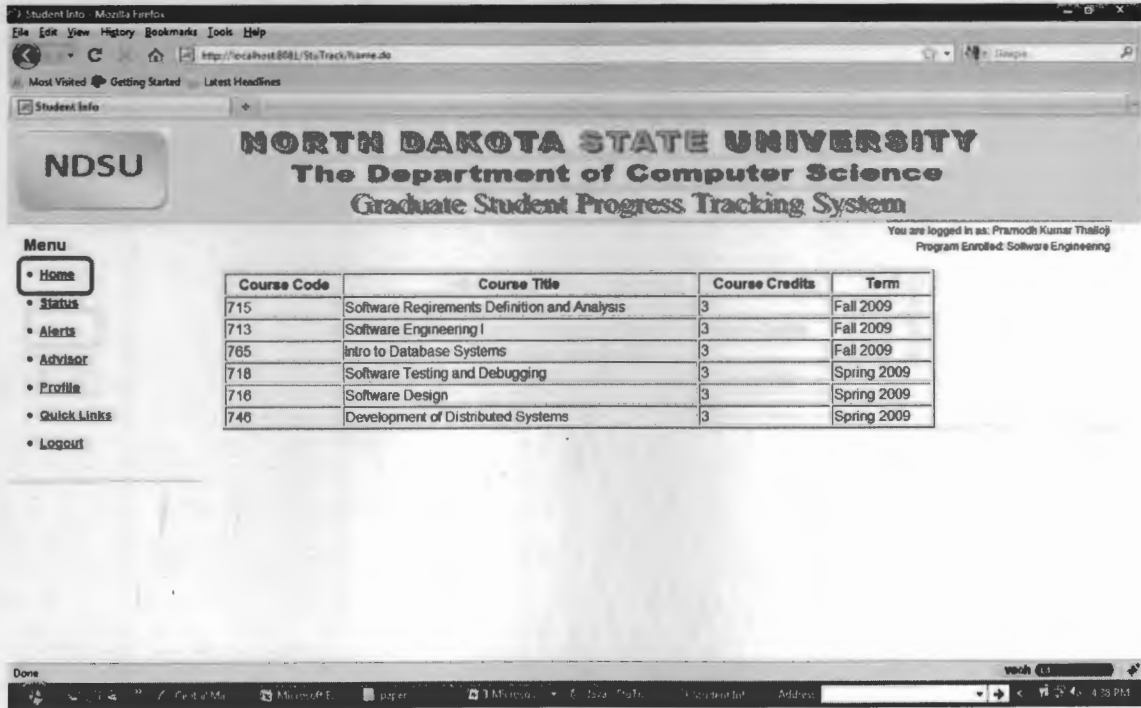


Figure A.4. Student status menu display page.

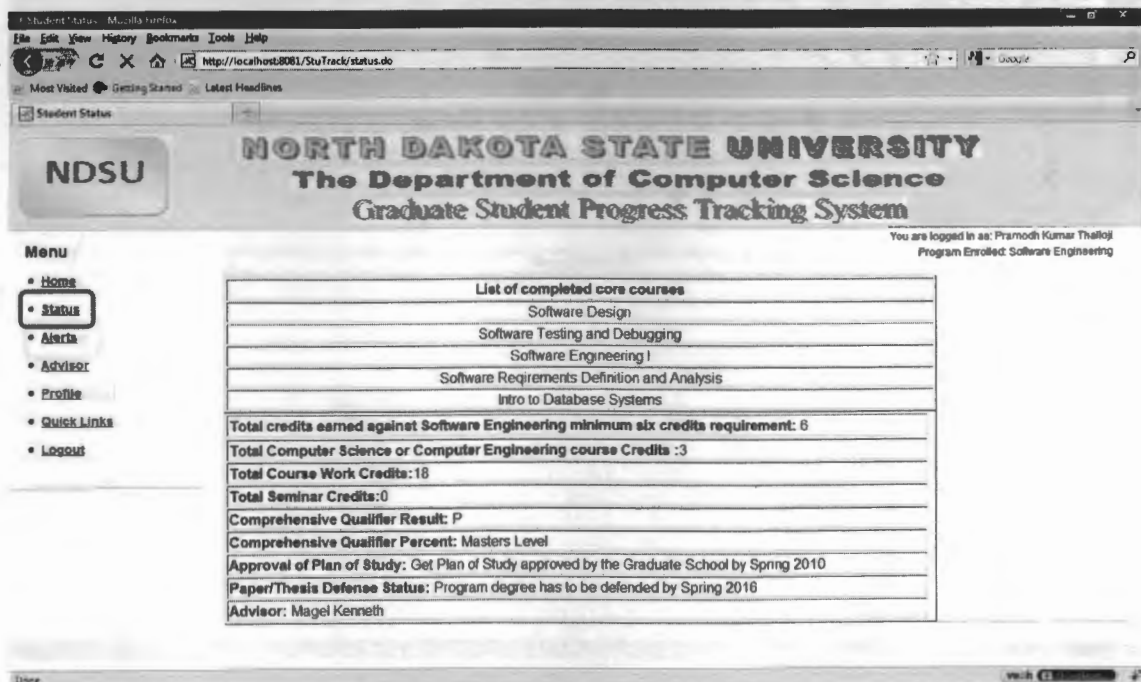


Figure A.5. Student alerts menu display page.

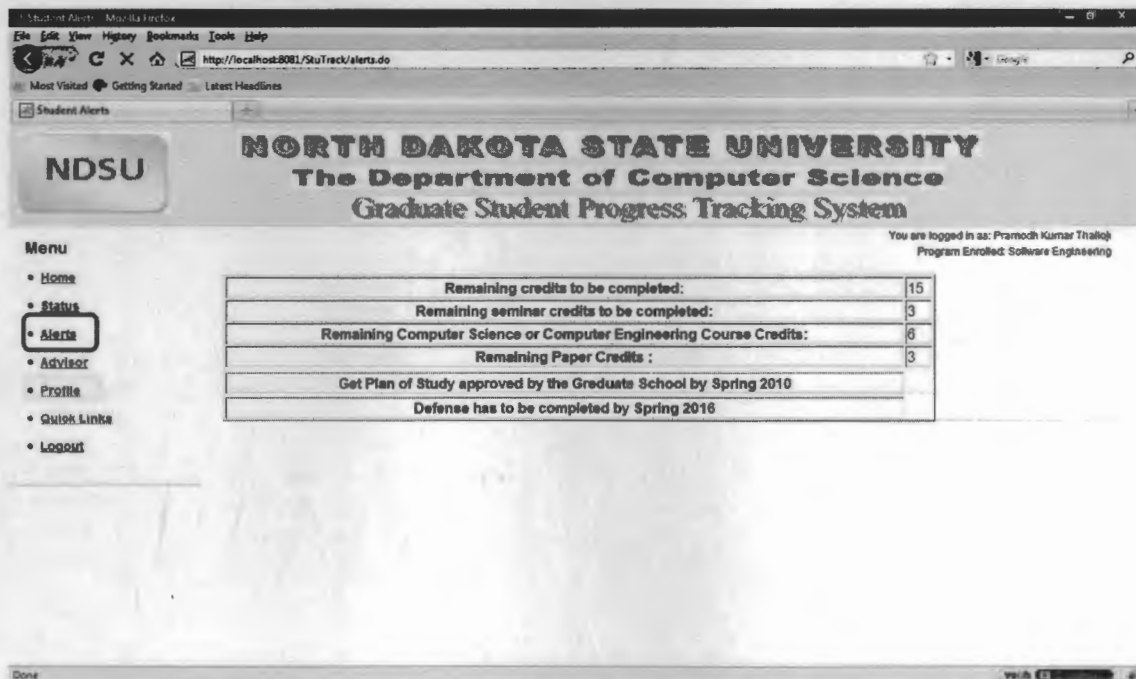


Figure A.6. Advisor info display page.

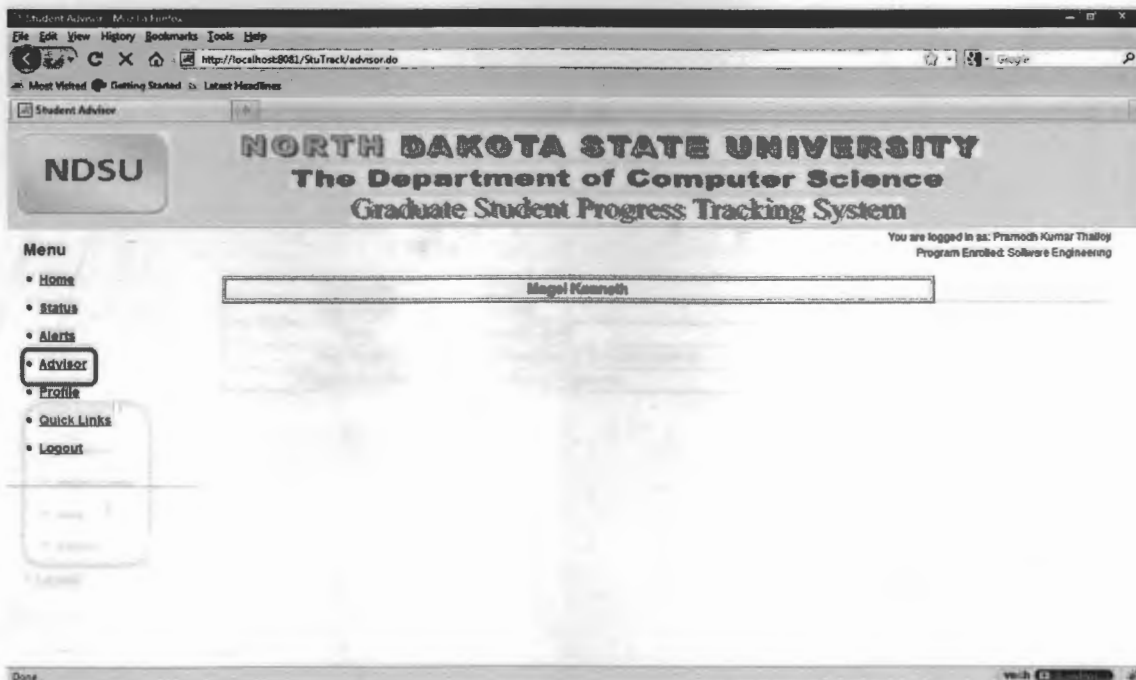


Figure A.7. Student general information page.

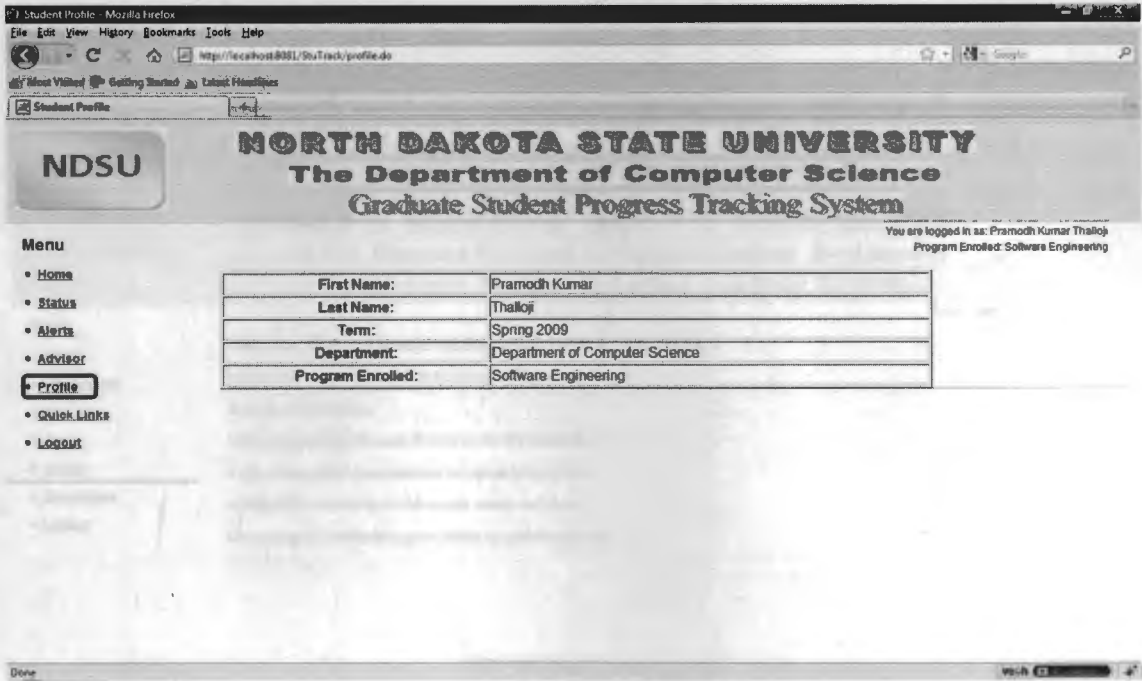
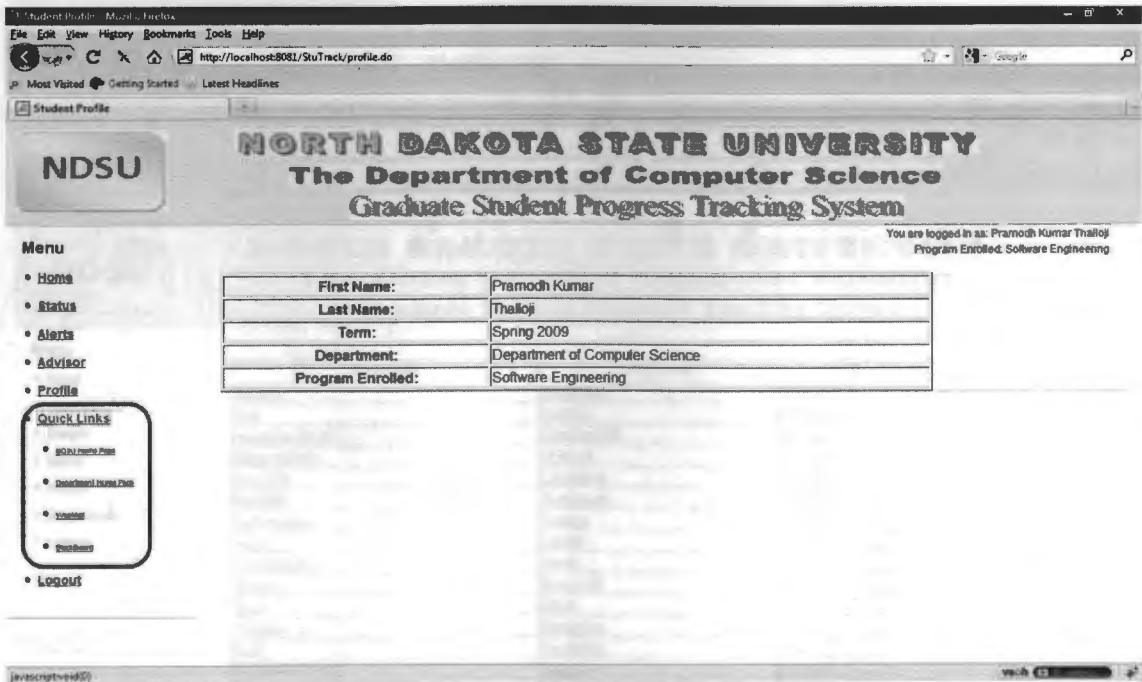


Figure A.8. Student login quick links page.



APPENDIX B

Figure B.1. Advisor login home page.

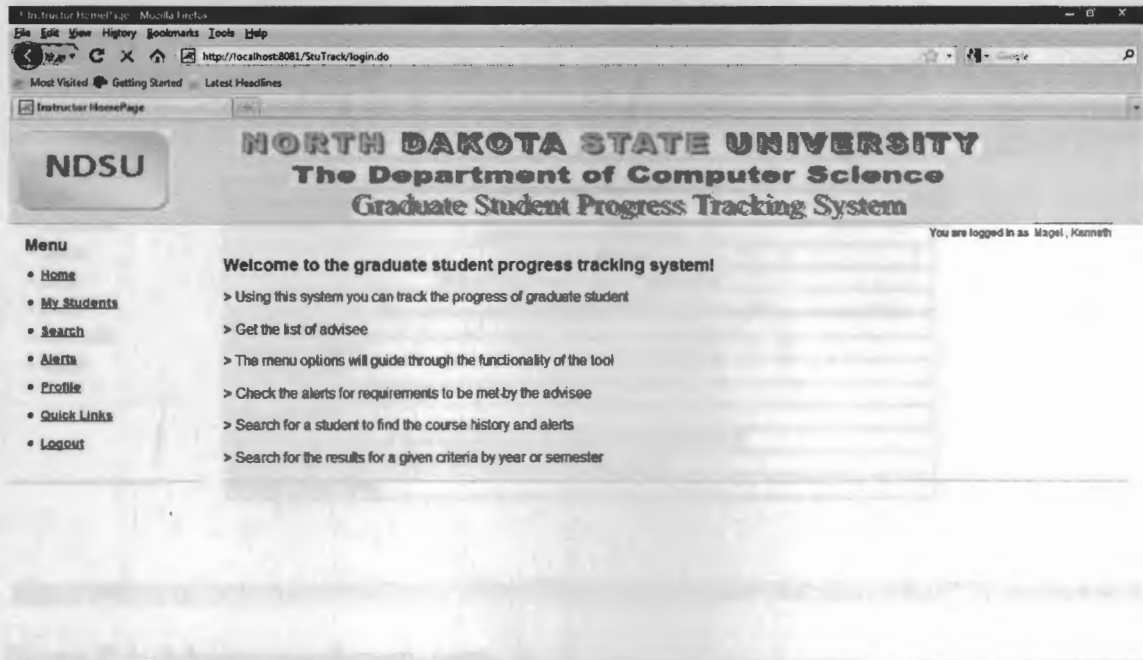


Figure B.2. Advisor *MyStudents* menu page.

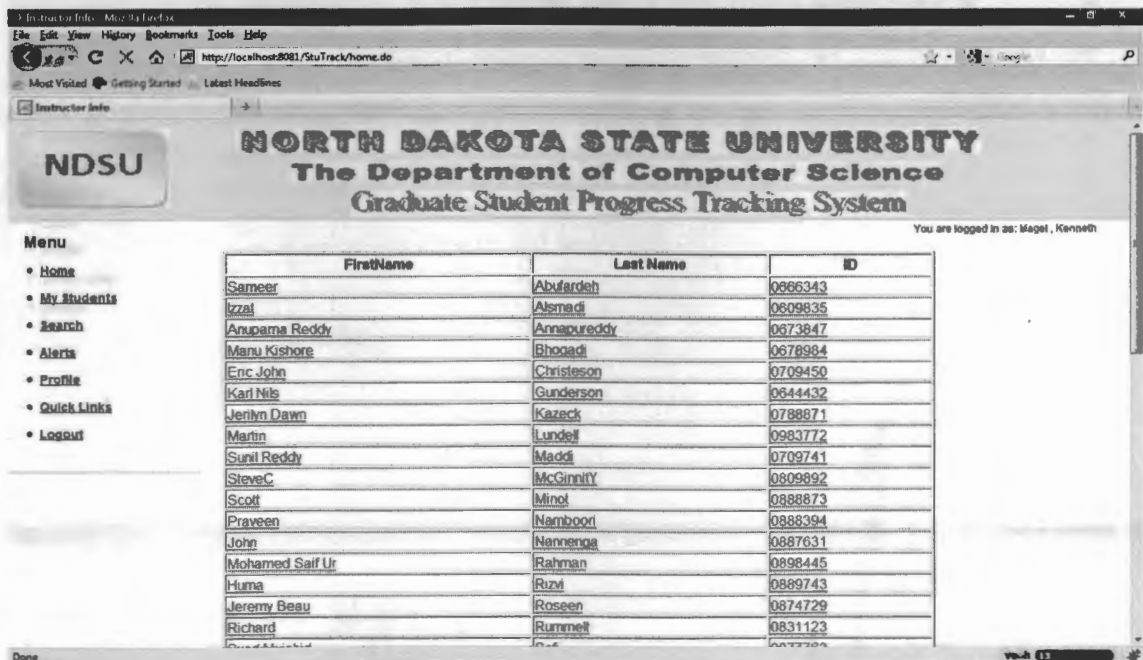


Figure B.3. Advisee status display page.

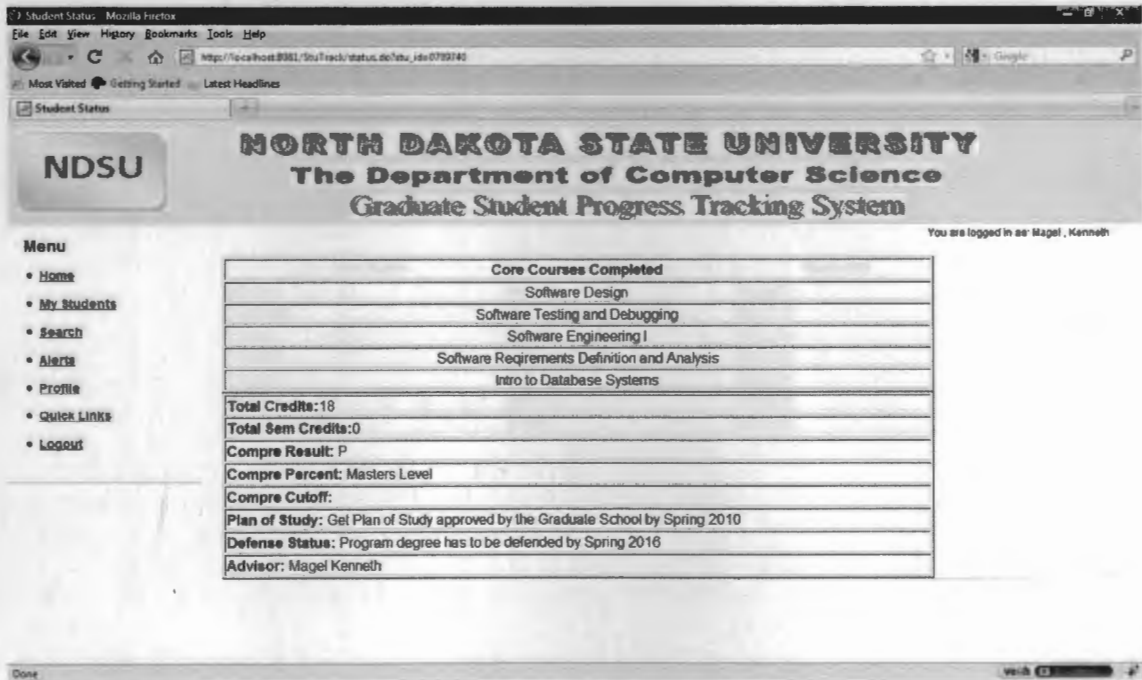


Figure B.4. Advisor search menu page.

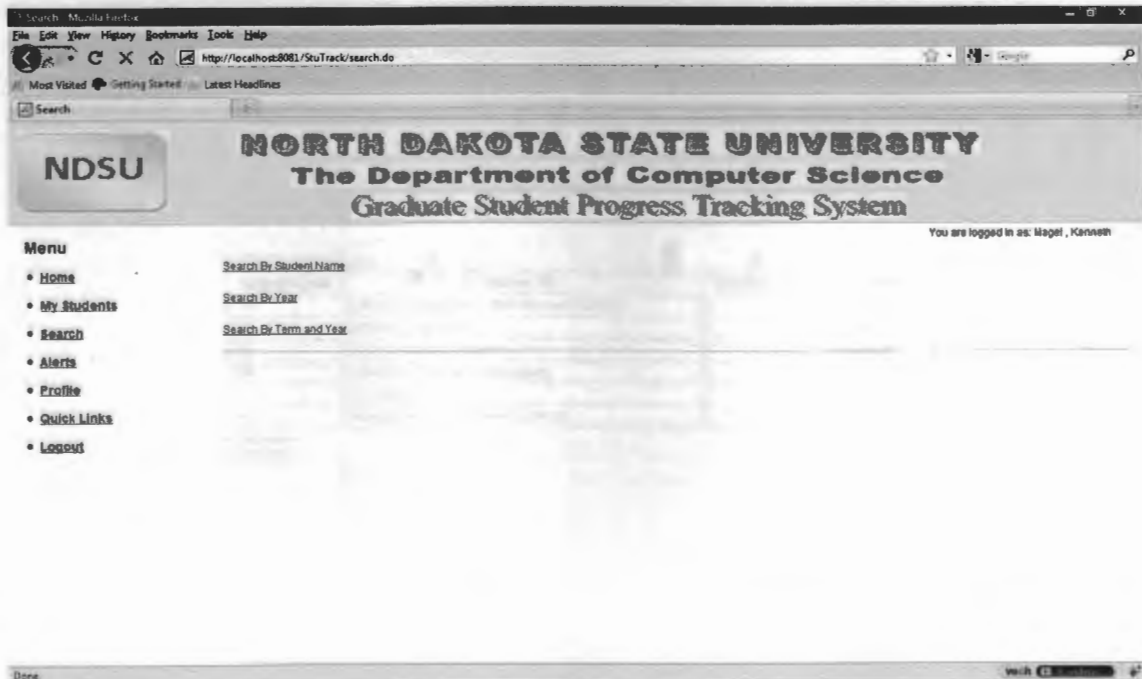


Figure B.5. Search by name results page.

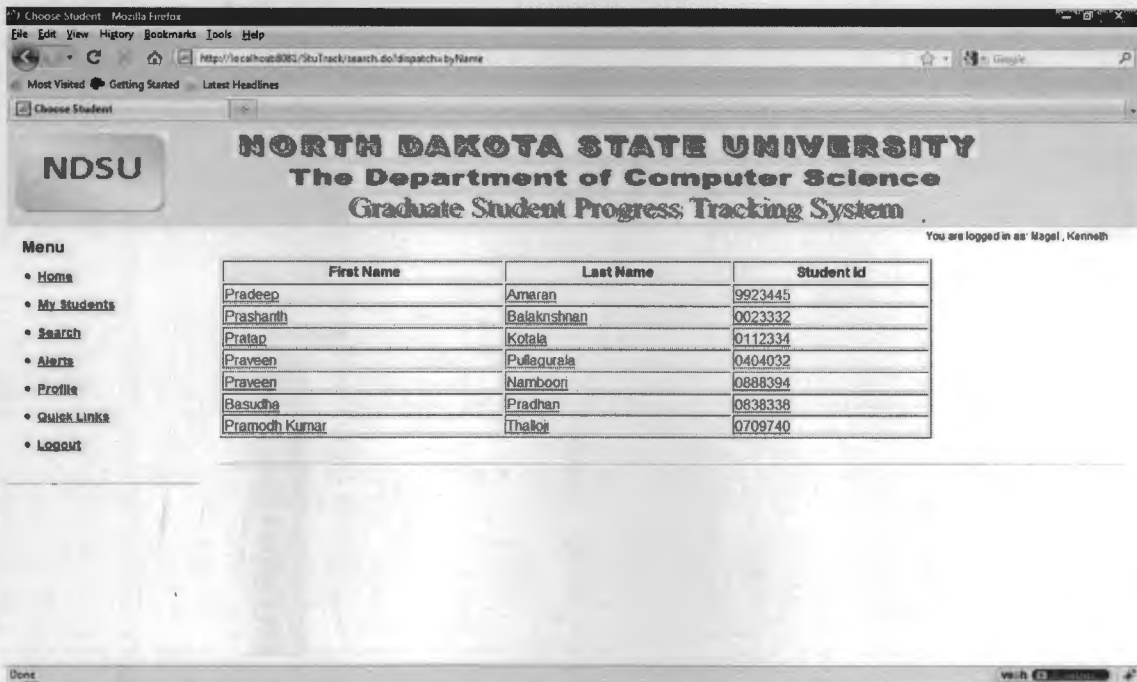


Figure B.6. Search by student name course history page.

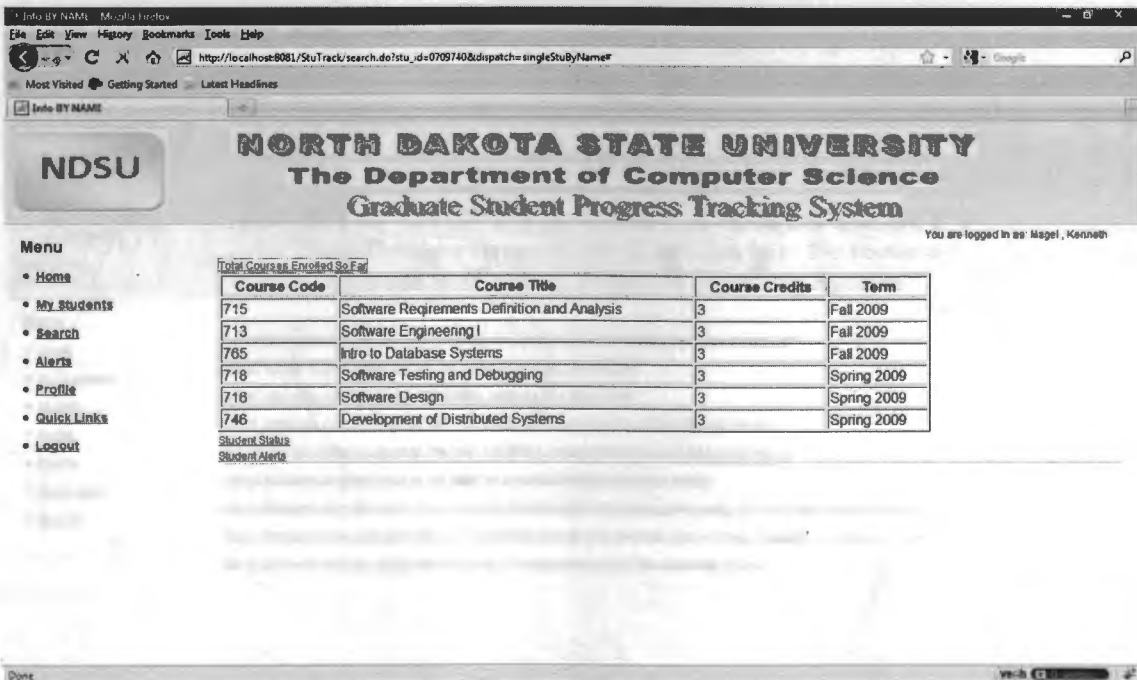


Figure B.7. Search by year page.

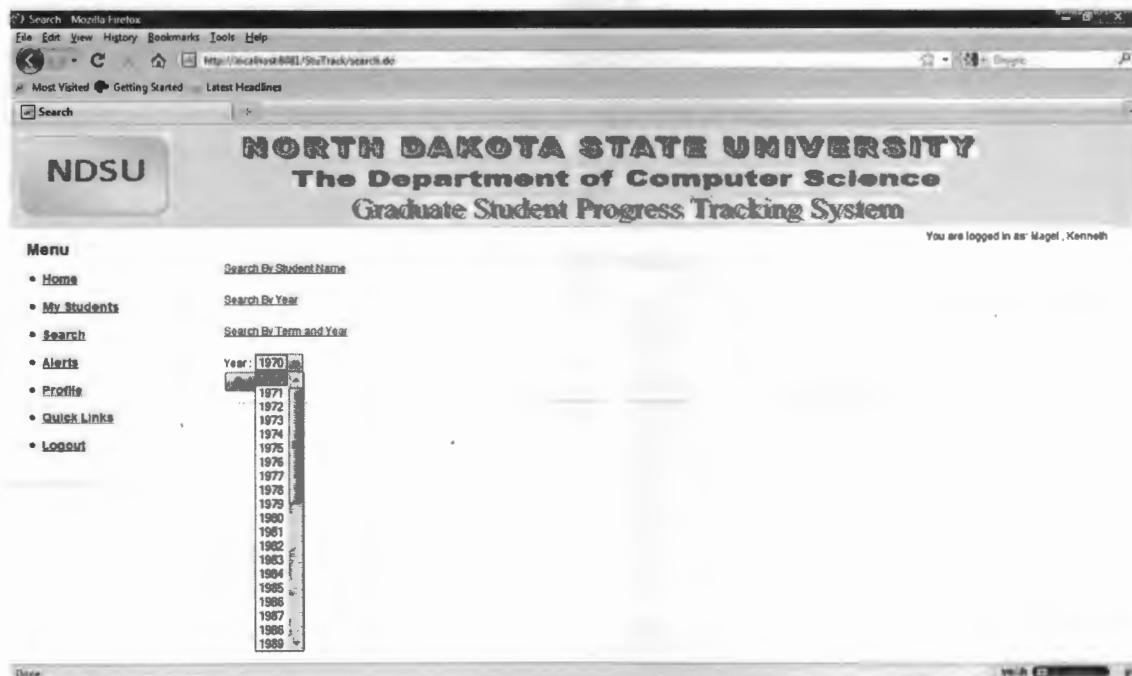


Figure B.8. Search by year results page.

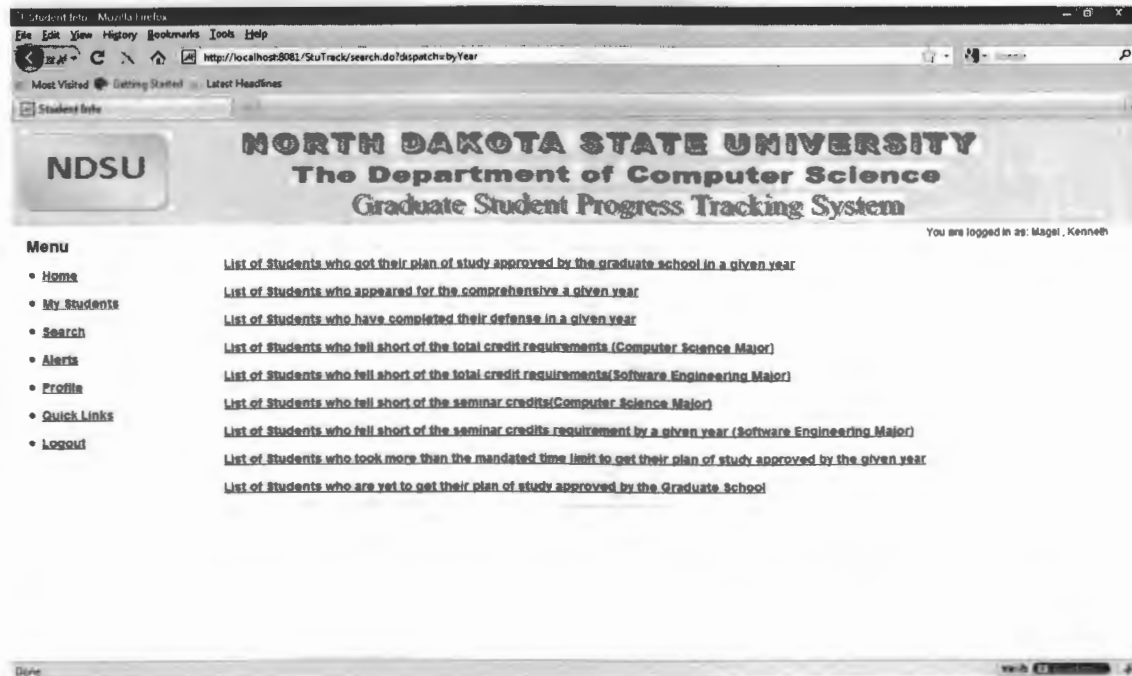


Figure B.9. Search by year criteria plan of study approved listing.

The screenshot shows a web browser window displaying the NDSU Graduate Student Progress Tracking System. The page title is "NORTH DAKOTA STATE UNIVERSITY The Department of Computer Science Graduate Student Progress Tracking System". The user is logged in as "Magel, Kenneth". A menu on the left includes Home, My Students, Search, Alerts, Profile, Quick Links, and Logout. The main content area displays a table of students whose plan of study was approved by the graduate school in a given year.

First Name	Last Name	Student Id	Term	Program Desc
Sameer	Abufardeh	0666343	Spring 2009	Software Engineering
Surjeet Singh	Banga	0667885	Fall 2009	Software Engineering
Manu Kishore	Bhogadi	0678984	Spring 2009	Software Engineering
Dibakar	Bhowmick	0665435	Spring 2009	Software Engineering
Gregory	Boyko	0796843	Fall 2009	Software Engineering
Ryan	Carlson	0777364	Fall 2009	Software Engineering
Anuj Kumar Singh	Chauhan	0709987	Summer 2009	Software Engineering
Eric John	Christeson	0709450	Spring 2009	Software Engineering
Senad	Cmic	0763842	Spring 2009	Software Engineering
Daniel George	Debit	0788893	Summer 2009	Software Engineering
Rav	Eda	0709443	Spring 2009	Software Engineering
Bouchaib	Falah	0798993	Spring 2009	Software Engineering
Karl Nils	Gunderson	0644432	Fall 2009	Software Engineering
Shanaka Nuradh	Herath	0709744	Fall 2009	Software Engineering
Farzana	Jahan	0777763	Fall 2009	Software Engineering
Bryce Ronald	Johnson	0766653	Summer 2009	Software Engineering

Figure B.10. Search by year criteria credit shortage listing.

The screenshot shows the same NDSU Graduate Student Progress Tracking System interface. The user is logged in as "Magel, Kenneth". The main content area displays a table of students who fell short of the total credit requirements for the Computer Science Major.

First Name	Last Name	Student Id	Total Credits
Virginia	Jefferson	0777598	30
Patti	Spangler	0678010	9
Rebecca	Peel	0909332	9
Mark	Scrimshire	0789105	29
Barbara	Miller	0984006	26
Deborah	Forville	0838572	18
Roddy	Fletcher	0689977	9
Anitha	Chakraborti	0637582	12
Kevin	Yang	0788628	26
Ralph	Traulwein	0765357	26
Amanda	Wright	0600055	27
Cathy	Martin	0766887	21

Figure B.11. Search by year, list of students who did not comply with graduate school mandated time limit for approval of plan of study.

The screenshot shows a web browser window with the URL <http://localhost:8081/StuTrack/search.do?dispatch=byYear#>. The page header includes the NDSU logo and the text "NORTH DAKOTA STATE UNIVERSITY The Department of Computer Science Graduate Student Progress Tracking System". A user is logged in as "Magel, Kenneth".

Menu

- Home
- My Students
- Search
- Alerts
- Profile
- Quick Links
- Logout

Navigation links:

- List of Students who got their plan of study approved by the graduate school in a given year
- List of Students who appeared for the comprehensive a given year
- List of Students who have completed their defense in a given year
- List of Students who fall short of the total credit requirements (Computer Science Major)
- List of Students who fall short of the total credit requirements (Software Engineering Major)
- List of Students who fall short of the seminar credits (Computer Science Major)
- List of Students who fall short of the seminar credits requirement by a given year (Software Engineering Major)
- List of Students who took more than the mandated time limit to get their plan of study approved by the given year
- List of Students who are yet to get their plan of study approved by the Graduate School

First Name	Last Name	Student Id	Term	Program Desc
Sameer	Abufardeh	0666343	Spring 2009	Software Engineering
Surjeet Singh	Banga	0667885	Fall 2009	Software Engineering
Gregory	Boyko	0796843	Fall 2009	Software Engineering

Figure B.12. Advisor Login: Search by year, list of students who is yet to get approval for plan of study from graduate school.

The screenshot shows a web browser window with the URL <http://localhost:8081/StuTrack/search.do?dispatch=byYear#>. The page header includes the NDSU logo and the text "NORTH DAKOTA STATE UNIVERSITY The Department of Computer Science Graduate Student Progress Tracking System". A user is logged in as "Magel, Kenneth".

Menu

- Home
- My Students
- Search
- Alerts
- Profile
- Quick Links
- Logout

Navigation links:

- List of Students who got their plan of study approved by the graduate school in a given year
- List of Students who appeared for the comprehensive a given year
- List of Students who have completed their defense in a given year
- List of Students who fall short of the total credit requirements (Computer Science Major)
- List of Students who fall short of the total credit requirements (Software Engineering Major)
- List of Students who fall short of the seminar credits (Computer Science Major)
- List of Students who fall short of the seminar credits requirement by a given year (Software Engineering Major)
- List of Students who took more than the mandated time limit to get their plan of study approved by the given year
- List of Students who are yet to get their plan of study approved by the Graduate School

First Name	Last Name	Student Id	Program Desc
Anupama Reddy	Annasureddy	0673847	Software Engineering
Martin	Lundell	0983772	Software Engineering
SteveC	McGinnily	0809892	Software Engineering
Scott	Minot	0888873	Software Engineering
Praveen	Namboon	0888394	Software Engineering
John	Nannenga	0887631	Software Engineering

Figure B.13. Search by semester display page.

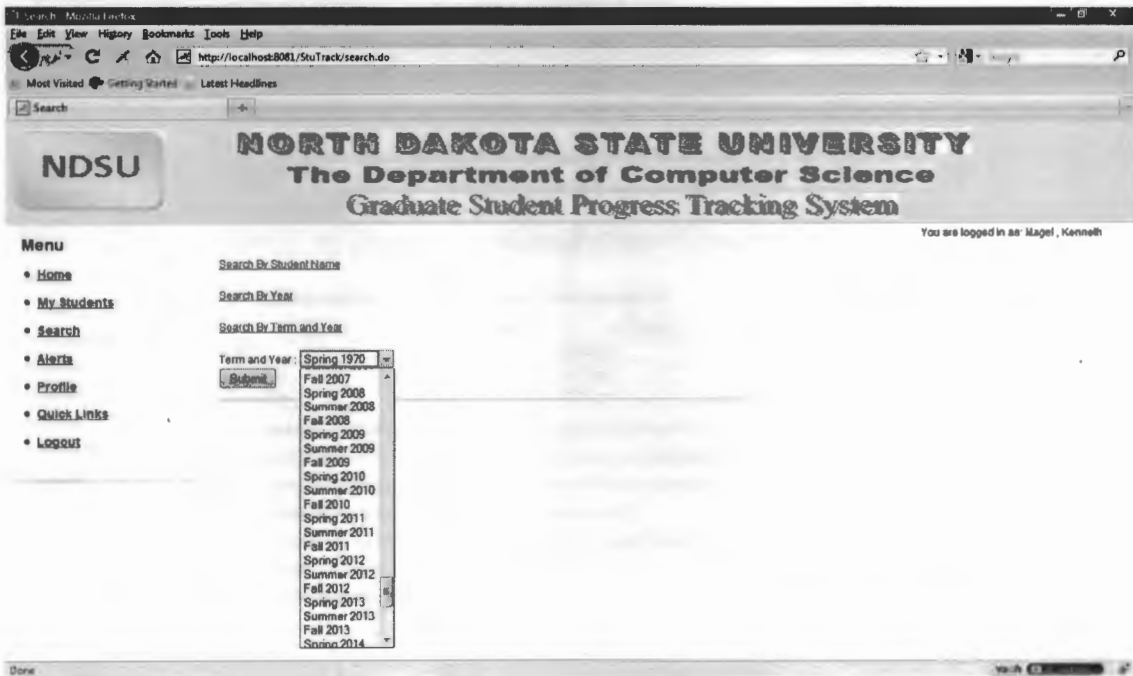


Figure B.14. Alerts menu display page.

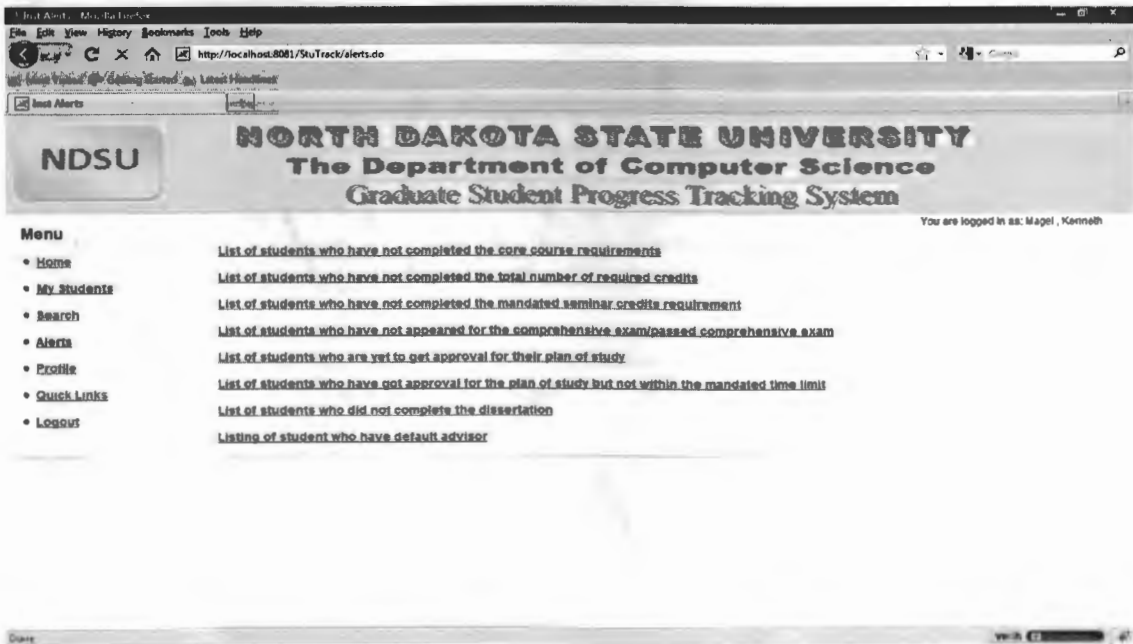


Figure B.15. Alerts indicating list of students who have not completed the core course requirements.

NDSU NORTH DAKOTA STATE UNIVERSITY
The Department of Computer Science
Graduate Student Progress Tracking System

You are logged in as: Magel, Kenneth

Menu

- Home
- My Students
- Search
- Alerts
- Profile
- Quick Links
- Logout

List of students who have not completed the core course requirements

First Name	Last Name	Student Id	Total Credits
Arun	Srivastava	0909098	3
Eric John	Christeson	0709450	6
Scott Allen	Vorachek	0900403	6

List of students who have not completed the total number of required credits

List of students who have not completed the mandated seminar credits requirement

List of students who have not appeared for the comprehensive exam/passed comprehensive exam

List of students who are yet to get approval for their plan of study

List of students who have got approval for the plan of study but not within the mandated time limit

List of students who did not complete the dissertation

Listing of student who have default advisor

Figure B.16. Alerts indicating list of students who have not completed the total credits program requirement.

NDSU NORTH DAKOTA STATE UNIVERSITY
The Department of Computer Science
Graduate Student Progress Tracking System

You are logged in as: Magel, Kenneth

Menu

- Home
- My Students
- Search
- Alerts
- Profile
- Quick Links
- Logout

List of students who have not completed the core course requirements

List of students who have not completed the total number of required credits

First Name	Last Name	Student Id	Total Credits
Arun	Srivastava	0909098	18
Bickrant	Shrestha	0800031	27
Eric John	Christeson	0709450	32
Huma	Rizvi	0889743	27
Jeremy Beau	Rosen	0874729	27
Jerilyn Dawn	Kazack	0788871	32
John	Nannenga	0887831	30
Martin	Lundell	0983772	32
Mehdi	Satter	0843322	27
Mohamed Saif Ur	Rahman	0898445	30
Mohammad	Smadi	0809091	27
Mridula	Sarker	0837348	27
Pramodh Kumar	Thalaji	0709740	18
Praveen	Nambooni	0888394	30
Rajat	Upadhyay	0903340	18

Figure B.17. Alerts indicating list of students who have not completed the total seminar credits requirement.

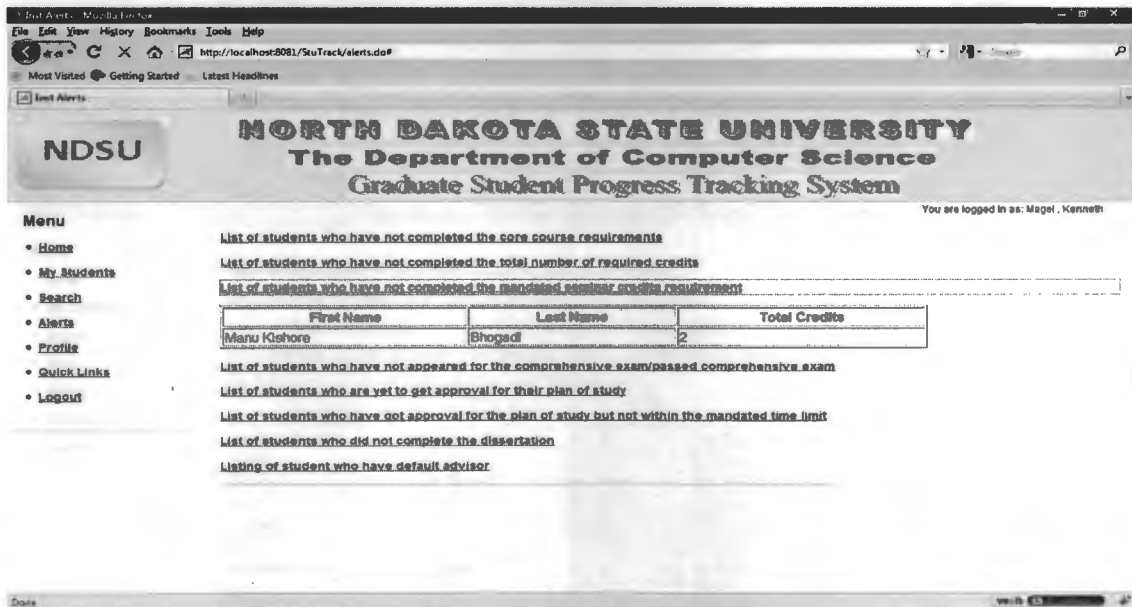


Figure B.18. Alerts indicating list of students who have not appeared for the comprehensive qualifier exam.

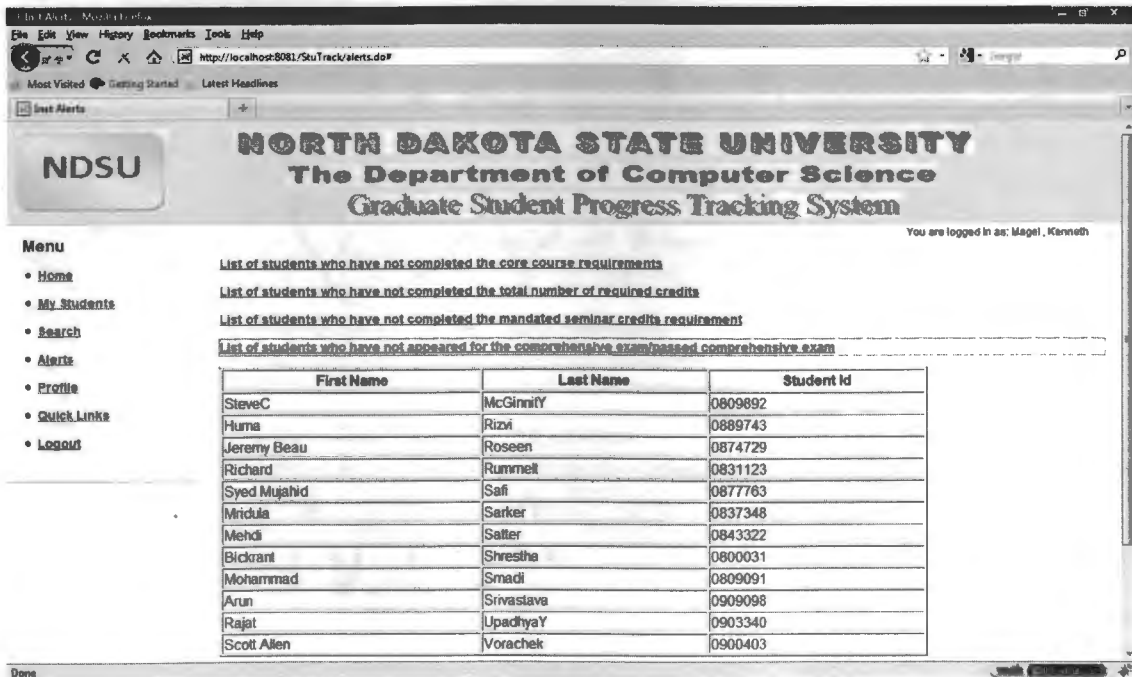


Figure B.19. Alerts indicating list of students who are yet to get the plan of study approved by the graduate school.

NORTH DAKOTA STATE UNIVERSITY
The Department of Computer Science
Graduate Student Progress Tracking System

You are logged in as: Magel, Kenneth

Menu

- Home
- My Students
- Search
- Alerts
- Profile
- Quick Links
- Logout

[List of students who have not completed the core course requirements](#)

[List of students who have not completed the total number of required credits](#)

[List of students who have not completed the mandated seminar credits requirement](#)

[List of students who have not appeared for the comprehensive exam/passed comprehensive exam](#)

[List of students who are yet to get approval for their plan of study](#)

First Name	Last Name	Student Id	Program Desc
Anupama Reddy	Annasureddy	0673847	Software Engineering
Martin	Lundell	0983772	Software Engineering
SteveC	McGinnTY	0809892	Software Engineering
Scott	Minot	0888873	Software Engineering
Praveen	Namboori	0888394	Software Engineering
John	Nannenga	0887631	Software Engineering
Mohamed Saif Ur	Rahman	0898445	Software Engineering
Huma	Rizvi	0889743	Software Engineering
Jeremy Beau	Roseen	0874729	Software Engineering
Richard	Rummelt	0831123	Software Engineering
Mehdi	Satter	0843322	Software Engineering

Figure B.20. Alerts indicating list of students who did not fulfill the graduate school mandated time limit for the approval of the plan of study.

NORTH DAKOTA STATE UNIVERSITY
The Department of Computer Science
Graduate Student Progress Tracking System

You are logged in as: Magel, Kenneth

Menu

- Home
- My Students
- Search
- Alerts
- Profile
- Quick Links
- Logout

[List of students who have not completed the core course requirements](#)

[List of students who have not completed the total number of required credits](#)

[List of students who have not completed the mandated seminar credits requirement](#)

[List of students who have not appeared for the comprehensive exam/passed comprehensive exam](#)

[List of students who are yet to get approval for their plan of study](#)

[List of students who have not approval for the plan of study but not within the mandated time limit](#)

First Name	Last Name	Student Id	Term	Program Desc
Sameer	Abufardeh	0886343	Spring 2009	Software Engineering

[List of students who did not complete the dissertation](#)

[Listing of student who have default advisor](#)

Figure B.21. Alerts indicating list of students who did not defend their degree.

The screenshot shows a web browser displaying the NDSU Graduate Student Progress Tracking System. The page title is "NORTH DAKOTA STATE UNIVERSITY The Department of Computer Science Graduate Student Progress Tracking System". The user is logged in as "Magel, Kenneth". A menu on the left includes Home, My Students, Search, Alerts, Profile, Quick Links, and Logout. The main content area lists several alert categories, with the selected one being "List of students who did not complete the dissertation". Below this is a table with the following data:

First Name	Last Name	Student Id	Program Desc
Anupama Reddy	Annapureddy	0673847	Software Engineering
Ryan	Carlson	0777364	Software Engineering
Anuj Kumar Singh	Chauhan	0709987	Software Engineering
Eric John	Chrsteson	0709450	Software Engineering
Senad	Cimic	0763842	Software Engineering
Daniel George	Debill	0788893	Software Engineering
Ravi	Eda	0709443	Software Engineering
Bouchaib	Falah	0798993	Software Engineering
...

Figure B.22. Alerts indicating list of students who have default advisor.

The screenshot shows the same NDSU Graduate Student Progress Tracking System interface. The selected alert is "Listing of student who have default advisor". Below this is a table with the following data:

First Name	Last Name	Student Id	Program Desc	Advisor First Name	Advisor Last Name
Anupama Reddy	Annapureddy	0673847	Software Engineering	Magel	Kenneth
Martin	Lundell	0983772	Software Engineering	Magel	Kenneth
SteveC	McGinnity	0809892	Software Engineering	Magel	Kenneth
Scott	Minot	0888873	Software Engineering	Magel	Kenneth
Praveen	Namboon	0888394	Software Engineering	Magel	Kenneth
John	Nannenga	0887631	Software Engineering	Magel	Kenneth
Mohamed Saif Ur	Rahman	0898445	Software Engineering	Magel	Kenneth

APPENDIX C

Figure C.1. Administrator Login: Home page display.

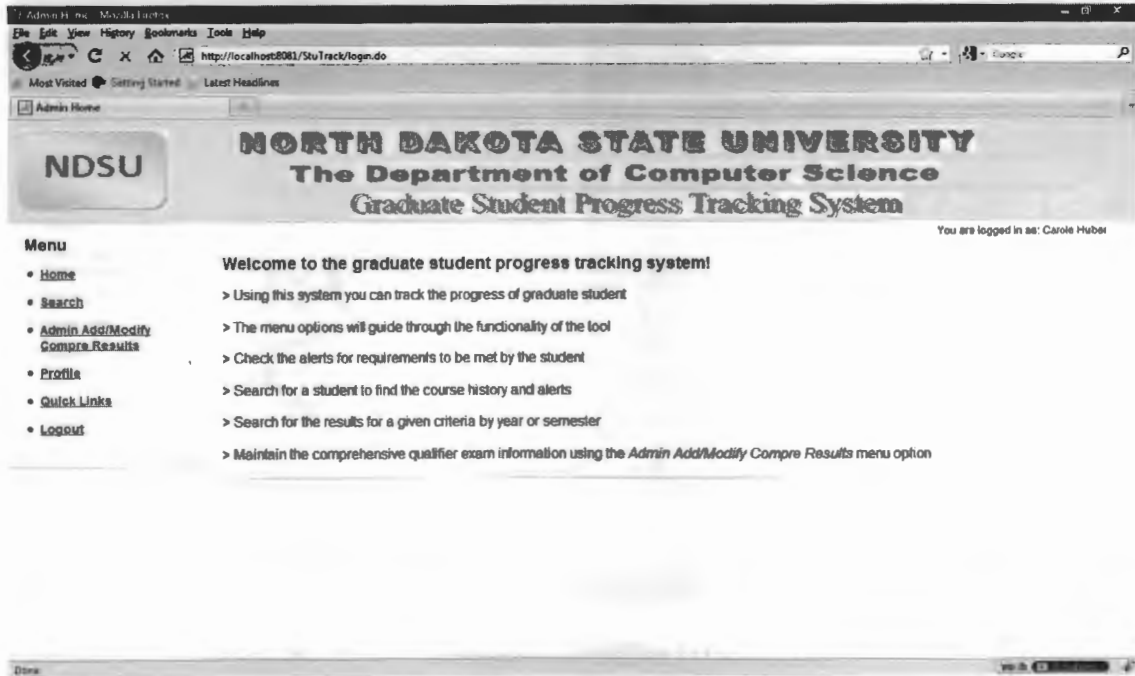


Figure C.2. Add/Modify menu display.

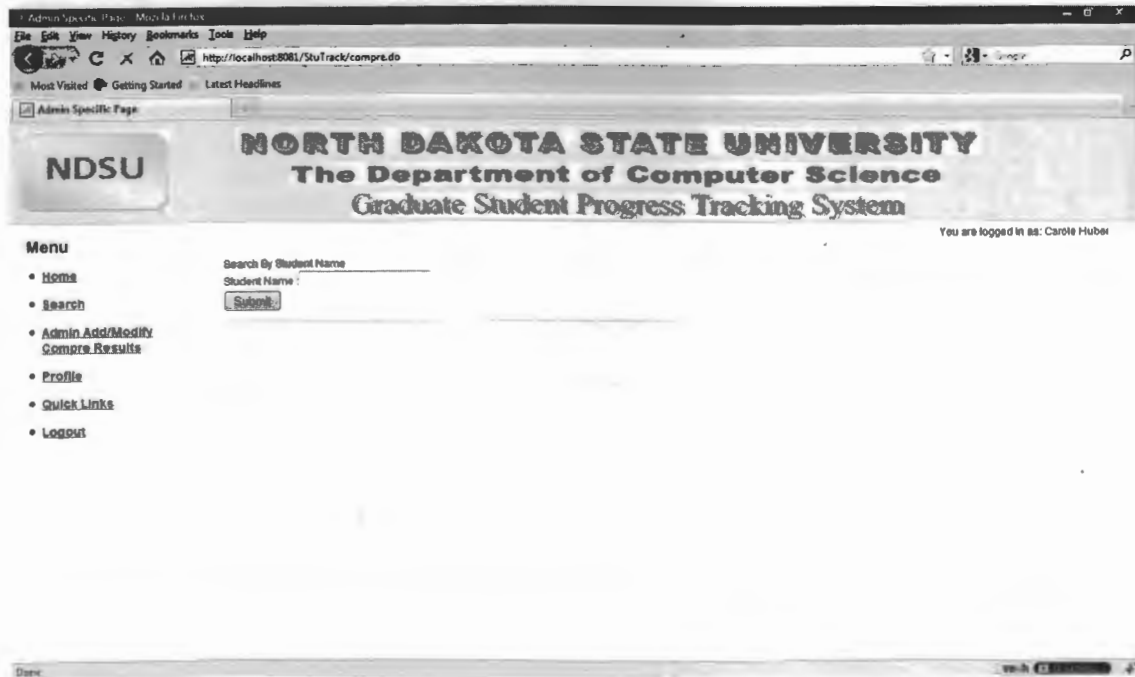


Figure C.3. Addition of comprehensive qualifier information.

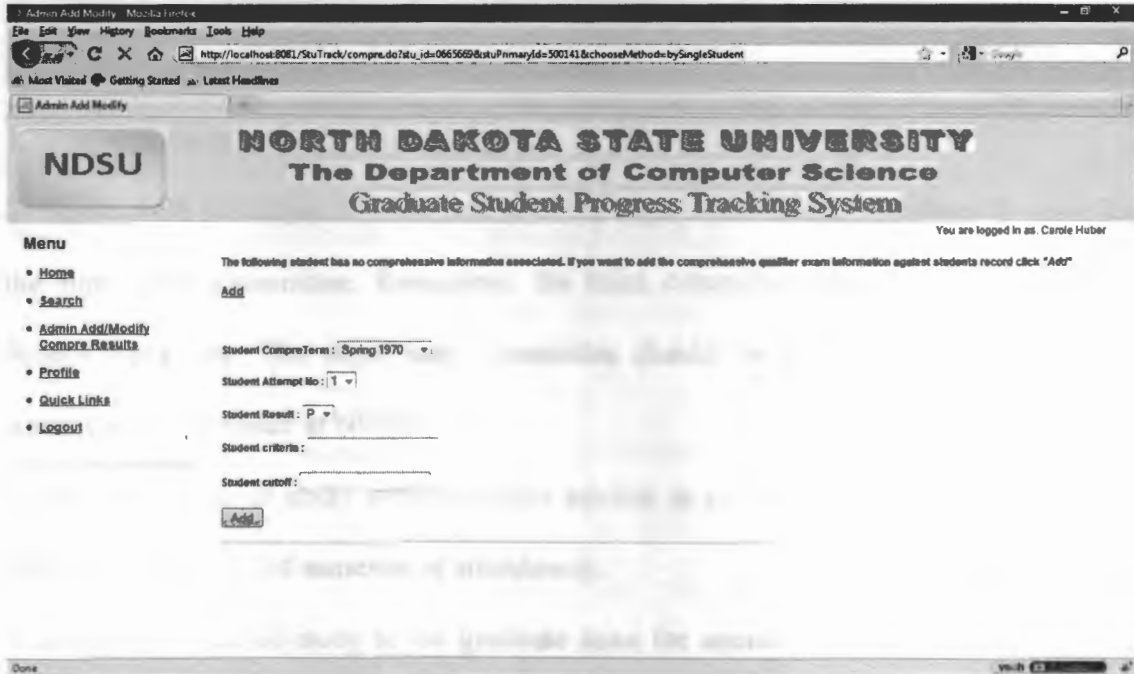
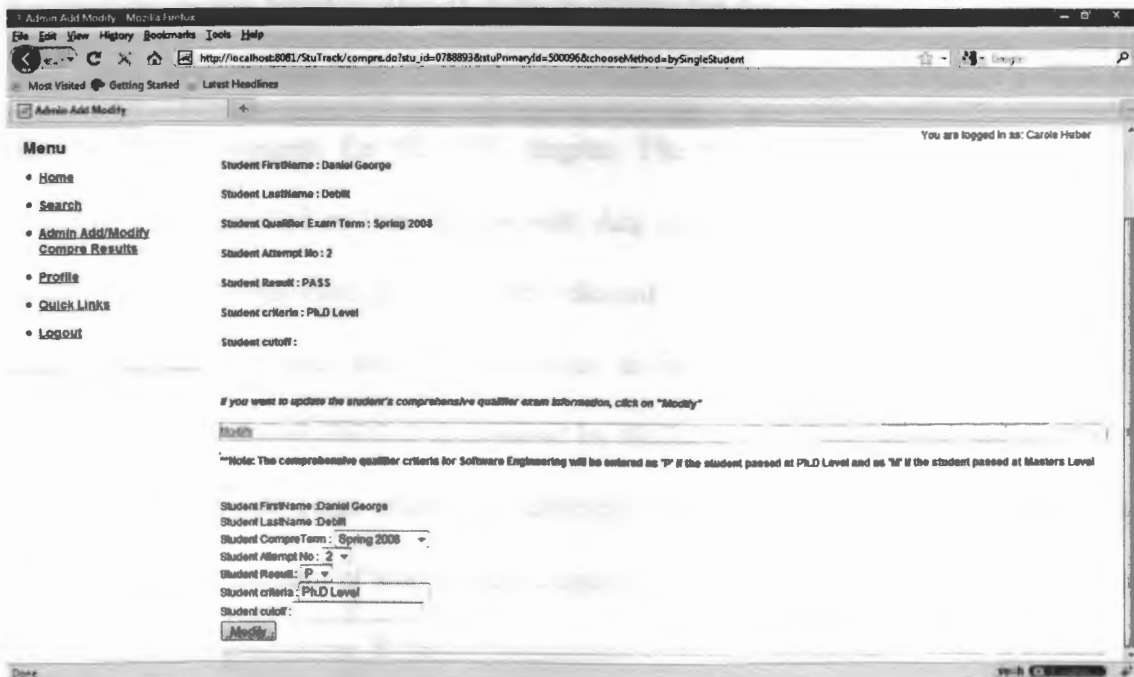


Figure C.4. Modification of comprehensive qualifier information.



APPENDIX D

Masters in Computer Science Evaluation Criteria:

1. Gain admission to the Graduate School.
2. Select, with the major advisor, one or more department faculty members to serve on the supervisory committee. Remember, the third committee member is the Graduate School Appointee. The supervisory committee should be selected by the end of the student's third semester at NDSU.
3. Develop a plan of study with the major advisor in consultation with the supervisory committee (usually 3rd semester of attendance).
5. Submit the plan of study to the graduate dean for approval by the third semester of attendance.
6. Complete courses listed on plan of study, including the disquisition.

The plan of study lists all the courses a student has taken and intends to take to satisfy the requirements for the M.S. degree. The names of the student's advisory committee are identified on this form as well. Any courses the student wishes to transfer to NDSU from another institution must be indicated on the plan.

The plan of study will be appropriate to meet the interests and needs of the student in his/her chosen field as determined by the supervisory committee and approved by the department/program chair, the academic dean, and the dean of the Graduate School. The master's plan of study must be approved by the Graduate School at least one semester prior to graduation. It should include the specific courses the student is expected to complete and all other requirements of the particular master's degree that the student is seeking. Computer Science requires a minimum of 32 semester credits, including two

credits of seminar (CS 790). Operations Research requires 32 semester credits, including two credits of graduate seminar (CS 790). At least two and no more than nine of the credits may be in CS 797 or 798. Computer Science permits a maximum of nine hours of independent study. All 600 and 700 level Computer Science courses may be used, except in cases where the department has determined that there is too much overlap between a 600-level course and a 700-level course in the same area. In such a case, students should take the 700-level course.

The plan of study may be revised as advisable and necessary. Revisions must be approved by the student, supervisory committee and dean of the Graduate School. The graduate dean will officially notify the student, supervisory committee, department/program chair, and registrar of all approved changes.

7. Course Work

At least 32 approved credits, including:

- (1) The core courses: CS 708, CS 713, CS 737, CS 653, and one of the two courses CS 724 or CS 765.
- (2) Two courses chosen from CS 654, CS 751, CS 752, CS 760 or CS 761.
- (3) The following courses or their equivalent, if not included in undergraduate studies: Probability and Statistics (Stat 467, 468), *Linear Algebra (Math 260 or Math 327), *FORTRAN programming (CS 160)
- (4) At least two credits of research seminars (CS 790). No more than two credits of seminars may be applied to the 32-credit requirement.

NOTE: Courses below the 600-level may not be used as part of the 32 minimum required credits

8. Comprehensive Qualifier

This is a three-hour written examination offered in both semesters, usually around the twelfth or thirteenth week. A passing score is required (usually between 60 and 70). The Graduate Coordinator should be consulted if there are any questions or concerns about this examination.

Every M.S. student must pass the comprehensive examination within the student's first two years in the program or by the first time it is offered after 24 semester hours are completed, whichever comes later. Students admitted with conditions will have three years to complete the comprehensive examination, since the conditions must be completed before they take the exam. The reading list for the exam consists of the current texts for the four core courses. Mastery of the contents of these books will allow the student to answer all questions satisfactorily. The test is made up of four parts, of approximately equal length, corresponding to the four courses. Normally questions covering the topics in a specific book are prepared and graded by one or more faculty members who have taught that course recently; this does not mean, however, that questions are restricted to topics covered in that offering of the course. In situations where two instructors have taught a course from different texts and it is not possible to select questions from a common body of material, it is sometimes necessary to include alternate versions of one part, or possibly to allow a student to choose from a set of questions. A student will be allowed to take the comprehensive exam at most twice.

9. Thesis and Comprehensive Study Options

For the thesis option, the research credits (CS 798) must be between six and ten credits. For the Comprehensive Study option, the research credits (CS 797) must be between two and four credits.

10. Time Limit

It is a Graduate School requirement that all requirements for the masters degree be completed within seven years of beginning the program. At the time of graduation, any requirements, especially courses, which were completed more than seven years earlier will not be credited. If they appear on the plan of study, they must be replaced with more recent course work.

APPENDIX E

Masters in Software Engineering Evaluation Criteria:

1. Software Engineering core (12 semester hours):

- \ a. CSCI 713: Software Development Processes
- \ b. CSCI 765: Introduction to Database Systems
- \ c. CSCI 716: Software Design
- d. Either CSCI 715: Software Requirements Definition, or CSCI 718:

Software Testing and Debugging. Each student selects one of these two courses. Students must complete the core within five semesters of their entering the program.

2. The Software Engineering Comprehensive Examination. This examination shall include integrative questions on the four courses which make up the software engineering core (see 1 above). The exam must be passed within the first seven semesters of their program. Each student is allowed a maximum of two attempts to pass this examination.

Students are encouraged to complete the comprehensive examination early in their program.

3. Six credits (not part of the core) from:

- \ a. CSCI 714: Software Project Planning and Estimation
- b. CSCI 715: Software Requirements Definition
- c. CSCI 717: Software Construction
- d. CSCI 718: Software Testing and Debugging
- e. CSCI 747: Software Complexity Metrics
- f. CSCI 745: Formal Methods for Software Development
- g. CSCI 746: Development of Distributed Applications

4. Nine credits of other Computer Science or Computer Engineering courses selected with and approved by the student's graduate advisory committee.
5. Three CSCI 790 graduate seminars in software engineering areas (1 credit each). These seminars must be approved in advance by the student's graduate advisor (a form is provided for this purpose).
6. A comprehensive study option (3 credits) paper based on a significant software development project undertaken by the student, perhaps as a member of a team, either at the University or as part of a job. This project will require design, implementation, and testing of a significant piece of computer software.
7. A Final Oral Examination on the paper and course work. This examination shall include questions on design choices, implementation methods, and testing choices for the student project.

Select, with the major advisor, one or more department faculty members to serve on the supervisory committee. Remember, the third committee member is the Graduate School Appointee. The supervisory committee should be selected by the end of the student's third semester at NDSU.

8. Develop a plan of study with the major advisor in consultation with the supervisory committee (usually 3rd semester of attendance).
9. Submit the plan of study to the graduate dean for approval by the third semester of attendance.
10. Complete courses listed on plan of study, including the disquisition.

The plan of study lists all the courses a student has taken and intends to take to satisfy the requirements for the M.S. degree. The names of the student's advisory

committee are identified on this form as well. Any courses the student wishes to transfer to NDSU from another institution must be indicated on the plan.

The plan of study should be appropriate to meet the interests and needs of the student in his/her chosen field as determined by the supervisory committee and approved by the department/program chair, the academic dean, and the dean of the Graduate School. The master's plan of study must be approved by the Graduate School at least one semester prior to graduation.

11. Comprehensive Qualifier

This is a three-hour written examination offered in both semesters, usually around the twelfth or thirteenth week. A passing score is required (at Ph.D level or MS level). The Graduate Coordinator should be consulted if there are any questions or concerns about this examination.

Every M.S. student must pass the comprehensive examination within the student's first two years in the program or by the first time it is offered after 24 semester hours are completed, whichever comes later. Students admitted with conditions will have three years to complete the comprehensive examination, since the conditions must be completed before they take the exam.

The reading list for the exam consists of the current texts for the four core courses. Mastery of the contents of these books will allow the student to answer all questions satisfactorily.

The test is made up of four parts, of approximately equal length, corresponding to the four courses. Normally questions covering the topics in a specific book are prepared and graded by one or more faculty members who have taught the course recently; this

does not mean, however, that questions are restricted to topics covered in that offering of the course. In situations where two instructors have taught a course from different texts and it is not possible to select questions from a common body of material, it is sometimes necessary to include alternate versions of one part, or possibly to allow a student to choose from a set of questions.

A student will be allowed to take the comprehensive exam at most twice.

12. Time Limit

It is a Graduate School requirement that all requirements for the masters degree be completed within seven years of beginning the program. At the time of graduation, any requirements, especially courses, which were completed more than seven years earlier will not be credited. If they appear on the plan of study, they must be replaced with more recent course work.