A web-based system for course instruction and student advising

Praveen Danda
West Virginia University

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Recommended Citation
Danda, Praveen, "A web-based system for course instruction and student advising" (2009). Graduate Theses, Dissertations, and Problem Reports. 2066.
https://researchrepository.wvu.edu/etd/2066

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.
A WEB - BASED SYSTEM FOR COURSE INSTRUCTION AND STUDENT ADVISING

by

Praveen Danda

Thesis submitted to the College of Engineering and Mineral Resources at West Virginia University in partial Fulfillment of requirements for the Degree of

Master of Science
In
Industrial Engineering

Rashpal Singh Ahluwalia, Ph.D., (Chair)
Jack Byrd, Ph.D.
Majid Jaraiedi, Ph.D.

Department of Industrial and Management Systems Engineering
Morgantown, West Virginia
2009
ABSTRACT

A WEB-BASED SYSTEM FOR COURSE INSTRUCTION AND STUDENT ADVISING

Praveen Danda

Academic advising plays an important role in student success in college, freshman students who join the universities with certain goals may face difficulties in course selection, whom to contact, and where to get course information. Considering this, each student in the university is assigned an advisor from his/her department who can address student’s problems during their academic years in the college. The advisors spend considerable amount of time in analyzing student performance during the semester by viewing the student’s grades and transcripts. The rise in student enrollment in engineering majors at West Virginia University is making it increasingly difficult for the advisors to spend more time with their advisees’. There is a need for an automated system which can provide information fast, accurate and available on the web so that the advisor can address students’ problems more efficiently and spend more time discussing the issues and less time on clerical. The focus of this project is to develop a web based system for course instruction and student advising for Bachelors’ Program in Industrial Engineering (BSIE) at West Virginia University (WVU). This could lay the ground work for expanding the system to college wide programs.

The student advising sub system would help advisors in suggesting courses for next semester by automatically checking advisees’ completed courses and comparing them to the BSIE curriculum. The system would also help in verifying students’ eligibility for graduation by checking his/her completed academics to the graduation requirements.

The course instruction sub system could help users in many ways. The instructors can post class material such as notes, home works, projects, and take home tests online.
The instructor can post midterm grades online after every grade category in the semester. The user can provide access to the students registered for the class with a user id and password. The instructor can also keep track of past semesters’ grades. It has four different interfaces for four different user types namely, Department Administrator, Instructor, Teaching Assistant and Student.

The department administrator user would have access to all methods available in the entire website. She/he would be able to access student and faculty information online. The department administrator can Add/Block/Delete any user in the database. The administrator can manage course catalog and assign courses to faculty. The department administrator will be able to update courses’ schedule, room/building, department, and college information.

The faculty/instructor interface would allow the members to view student information and access course schedules information. It would provide specific information on advisee assigned to the faculty member. The faculty interface would also allow faculty members to update/upload course content to the web. They would also be able to manage class roster online.

The teaching assistant user would have access to some of the methods designed for the instructor. This user can upload/update course content to the website. She/he can provide user access for students to the website. The teaching assistant can also upload course grades after every homework/project/tests.

The student interface would provide the students with customized advice on which courses to take by referencing the most recent course information in the database. Students can view course material and grades online posted by the instructor. Students can get advice on courses to take next semester. It would provide information about the prerequisites to be taken for a specific course and also on selecting the approved number of General Education Curriculum (GEC) courses from each objective.
ACKNOWLEDGEMENT

I would like to thank my advisor Dr. Rashpal Singh Ahluwalia for his continuous support and guidance throughout my Masters and making this thesis a useful tool. I would also like to thank my committee members Dr. Jack Byrd and Dr. Majid Jaraiedi for their support and valuable suggestions in addressing more advising tasks. I would like to show my gratitude to the Department of Industrial and Management Systems Engineering (IMSE) and West Virginia University (WVU) for giving me an opportunity to pursue my Masters. Special thanks to my parents and friends for their constant support in all my success in life.
# TABLE OF CONTENTS

**ABSTRACT** .......................................................................................................................... ii

**ACKNOWLEDGEMENT** ........................................................................................................ iv

**TABLE OF CONTENTS** ..................................................................................................... v

**LIST OF FIGURES** .............................................................................................................. vii

**LIST OF TABLES** ................................................................................................................ ix

**CHAPTER 1: INTRODUCTION** .......................................................................................... 1

1.1 ACADEMIC ADVISING ....................................................................................................... 1
1.2 INDUSTRIAL ENGINEERING PROGRAM AT WVU ........................................................ 2
1.3 BSIE CURRICULUM ......................................................................................................... 4
1.4 DATA MINING .................................................................................................................. 6
1.5 DATA MINING TOOLS ...................................................................................................... 7
1.6 DATA MINING ALGORITHMS .......................................................................................... 7
1.7 PROBLEM STATEMENT ................................................................................................... 10

**CHAPTER 2: LITERATURE REVIEW** .................................................................................. 12

2.1 ACADEMIC ADVISING .................................................................................................... 12
2.2 WEB BASED STUDENT ADVISING .............................................................................. 14
2.3 DATA MINING ON STUDENT DATA ............................................................................. 16

**CHAPTER 3: DATABASE DESIGN** ....................................................................................... 18

3.1 DATABASE DEFINITION .................................................................................................. 18
3.2 DBMS ARCHITECTURE ................................................................................................... 18
3.3 CLIENT/SERVER ARCHITECTURE ............................................................................... 20
3.4 NORMALIZATION OF DATABASE ............................................................................... 21
3.5 DATABASE IMPLEMENTATION .................................................................................... 22
3.6 STORED PROCEDURES .................................................................................................. 34

**CHAPTER 4: SYSTEM ARCHITECTURE AND IMPLEMENTATION** .......................... 37

4.1 THREE TIER ARCHITECTURE ...................................................................................... 37
4.2 DATA MINING IMPLEMENTATION ............................................................................... 41
4.3 DATA MINING TASKS .................................................................................................... 42

**CHAPTER 5: SYSTEM VALIDATION** .................................................................................. 49

5.1 DEPARTMENT ADMINISTRATOR USER ...................................................................... 50
5.2 INSTRUCTOR USER ....................................................................................................... 58
5.3 STUDENT ADVISING .................................................................................................... 67
5.4 TEACHING ASSISTANT USER .................................................................................... 74
5.5 STUDENT USER ........................................................................................................... 76
LIST OF FIGURES

FIGURE 1-1 SIMPLE DECISION TREE ................................................................. 9
FIGURE 3-1: THREE-Schema Architecture [26] .............................................. 19
FIGURE 3-2: CLIENT/SERVER ARCHITECTURE [27] ......................................... 21
FIGURE 3-3 ENTITY RELATIONSHIP DIAGRAM ............................................. 34
FIGURE 4-1 THREE-TIER ARCHITECTURE [27] .............................................. 37
FIGURE 4-2 SYSTEM ARCHITECTURE .......................................................... 39
FIGURE 4-3 DATA MINING PROCESS DIAGRAM .......................................... 43
FIGURE 4-4 SCREEN SHOT OF BAR AND PIE CHART OF WVU GPA AND HIGH SCHOOL GPA FIELDS IN THE TABLE ................................................................. 45
FIGURE 4-5 SCREEN SHOT OF PIE CHARTS OF WVU GPA AND HIGH SCHOOL GPA FIELDS IN THE TABLE ................................................................................... 45
FIGURE 4-6 SCREEN SHOT OF THE MINING MODELS IN THE STRUCTURE AND THEIR ATTRIBUTES ................................................................. 46
FIGURE 4-7 MINING LEGEND FOR TEST DATA .............................................. 47
FIGURE 4-8 LIFT CHART ................................................................................... 48
FIGURE 5-1: HOME PAGE .................................................................................. 49
FIGURE 5-2 WEB PAGE AFTER SUCCESSFUL LOGIN BY ADMIN .................... 50
FIGURE 5-3 ADMIN HOME PAGE ..................................................................... 51
FIGURE 5-4 ADMIN ADD NEW COURSE WEB PAGE ....................................... 52
FIGURE 5-5 ADD NEW USER WEB PAGE ........................................................ 53
FIGURE 5-6 ADMIN MANAGE INSTRUCTORS WEB PAGE .............................. 54
FIGURE 5-7 ADMIN MANAGE STUDENT ADVISORS WEB PAGE WITH A STUDENT ACADEMIC PERFORMANCE .......................................................... 55
FIGURE 5-8 ADMIN MANAGE STUDENT ADVISORS .................................. 55
FIGURE 5-9 SCREEN SHOT OF MANAGE IE CURRICULUM ............................ 56
FIGURE 5-10 ADMIN UPDATE COURSE WEB PAGE ....................................... 57
FIGURE 5-11 ADMIN VIEW GRADES ............................................................... 58
FIGURE 5-12 OPTIONS FOR INSTRUCTOR AFTER SUCCESSFUL LOGIN ........ 59
LIST OF TABLES

TABLE 1-1: ADMISSIONS REQUIREMENTS TO CEMR........................................................... 3
TABLE 1-2: FRESHMAN YEAR COURSES .............................................................................. 4
TABLE 1-3: SOPHOMORE YEAR COURSES ............................................................................ 5
TABLE 1-4: JUNIOR YEAR COURSES ..................................................................................... 5
TABLE 1-5: SENIOR YEAR COURSES ..................................................................................... 6
TABLE 3-1 LOGIN TABLE FIELDS ....................................................................................... 24
TABLE 3-2 COURSES TABLE FIELDS................................................................................... 25
TABLE 3-3 COURSE SCHEDULE TABLE FIELDS..................................................................... 25
TABLE 3-4 COURSES PRE REQUISITES.................................................................................. 26
TABLE 3-5 COURSES CO REQUISITES TABLE FIELDS .......................................................... 26
TABLE 3-6 COURSE EQUIVALENT TABLE FIELDS ............................................................... 26
TABLE 3-7 GEC REQUIREMENTS TABLE FIELDS ............................................................... 27
TABLE 3-8 I REQUIRED COURSES TABLE ....................................................................... 28
TABLE 3-9 TECH ELECTIVE COURSES TABLE FIELDS ......................................................... 28
TABLE 3-10 COURSE INST TABLE FIELDS ......................................................................... 28
TABLE 3-11 COURSE STUDENTS TABLE FIELDS ................................................................. 29
TABLE 3-12 COURSE NOTES TABLE FIELDS ...................................................................... 29
TABLE 3-13 TRANSCRIPT TABLE FIELDS .......................................................................... 30
TABLE 3-14 STUDENT ADVISORS TABLE ........................................................................... 30
TABLE 3-15: FIELDS OF COLLEGE TABLE ......................................................................... 31
TABLE 3-16: FIELDS OF DEPARTMENT TABLE................................................................. 31
TABLE 3-17 USEFUL LINKS TABLE FIELDS ........................................................................ 31
TABLE 3-18 DATA FOR DATA MINING TABLE FIELDS......................................................... 32
1.1 Academic Advising

Proper academic advising plays an important role for student success in engineering programs. Essential to the success of any advising program are accuracy and consistency of information and customization of materials to meet individual student needs. Academic advising assists a student in selecting the courses in his/her major. In addition, the academic advisors help the students clarify their academic and career goals; assist with institutional policies and procedures to adapt students to college life. Academic advising is an important and time-consuming task and different tools and techniques are used to make it an effective and efficient process.

Academic advising as a part of the educational process of every university plays a critical role in providing better learning opportunities to students and encouraging them in achieving their goals. Academic advising as a part of educational process helps change the traditional advising process of providing information or inputs, to a new trend that focus on outcomes of student learning and success. This helps achieve certain key institutional goals such as providing support, feedback, faculty involvement in student learning and success [1].

Every student regardless of the age and level of education has certain goals. Having the facility of academic advising at every level helps them to step towards their success. Academic advising that facilitates student engagement and success is developmental and supports the social and intellectual growth that occurs during the college experience. Thus, the academic advising is considered and recognized as a key factor in educational mission of every institution as a form of teaching. This is the academic advising of the twenty-first century, and colleges and universities nationally and internationally are acknowledging and realizing the potential of this powerful educational strategy to engage and support student learning.

After high school, students entering the university become a member of the higher education community. Through academic advising students learn to think their roles and responsibilities as a member of higher education community and prepare to become educated citizens of a democratic society as they exit the institution. Academic advising
helps them realize the actual world beyond their own views, acknowledging their individual characteristics and values throughout their journey in the institution [1].

Academic advising should be recognized as a campus wide responsibility in which all departments, administrators, faculty, staff and students work together to promote student success. The internal and external factors of higher education should be viewed under microscope and identify strategic ways to achieve student learning, satisfaction and success. Academic advising takes place much more often and effectively when proper tools are in place to facilitate the more routine but essential tasks associated with basic course scheduling. Automation is required which supplements the conventional advising process with a web-based advising system to provide robustness of the advisory system. Most of the current web-based advising was typically a bulletin board with advising related announcements, a repository of official documents in a PDF file format or a collection of useful links that help students get official advising related information. They do not include any programs that process specific student information and produce customized advice for that particular student. Introduction of Web-Based Student Advising tools will produce several positive changes. These include improved satisfaction among faculty and students, and more timely and accurate updating of student records while simultaneously reducing the clerical workload.

1.2 Industrial Engineering Program at WVU

In the undergraduate education system, students follow a structured schedule which is set for a particular major. Each major has its unique curriculum and requirements, usually preset, but undergoing limited changes on a regular basis. The factors that have to be considered vary from school specific requirements, spring and fall course offerings and goals. Most majors typically require around eight semesters for completion and depending on the number of credits taken at a time the student is considered freshman, sophomore, junior, or senior. The courses in a curriculum typically deal with the major, general education courses, and technical electives. In addition, the courses have pre or co- requisites.

The Industrial Engineering (IE) major is part of the Industrial and Management Systems Engineering (IMSE) department. The IMSE program is part of the College of
Engineering and Mineral Resources at WVU. The IE program provides balanced background in the basic sciences, engineering sciences, social sciences, humanities, analysis, synthesis and evaluation thinking skills involved in design and decision making.

CEMR admits students as either an engineering student or a general engineering student. The objective of the program is to maximize the student’s learning ability in the freshman year. Admission to the program is based on a high school grade point average (un-weighted 4.0 scale) and math placement scores. The following table describes the admission requirements for each program.

**Table 1-1: Admissions Requirements to CEMR**

<table>
<thead>
<tr>
<th>Program</th>
<th>Residents</th>
<th>High School GPA</th>
<th>ACT Composite</th>
<th>ACT Math</th>
<th>SAT Total</th>
<th>SAT Math</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering</td>
<td>West Virginia</td>
<td>3</td>
<td>24</td>
<td>27</td>
<td>1110</td>
<td>620</td>
</tr>
<tr>
<td></td>
<td>Out-of-State</td>
<td>3</td>
<td>24</td>
<td>27</td>
<td>1110</td>
<td>620</td>
</tr>
<tr>
<td>General</td>
<td>West Virginia</td>
<td>2.5</td>
<td>22</td>
<td>23</td>
<td>1030</td>
<td>540</td>
</tr>
<tr>
<td>Engineering</td>
<td>Out-of-State</td>
<td>2.5</td>
<td>20</td>
<td>24</td>
<td>950</td>
<td>520</td>
</tr>
</tbody>
</table>

In addition to the above requirements, the students must meet the high school requirements of:

- Four units of English (including grammar, composition and literature)
- Three units of social studies (including U.S. history)
- Three units of college preparatory mathematics (algebra I and II and geometry)
- Two units of laboratory sciences (including physics, chemistry, biology, or other laboratory courses).
1.3 BSIE Curriculum

The General Engineering Program is meant for those students in the Freshman Year who had received either a math ACT score of less than 27 or a math SAT score of less than 620. These students are meant to take prerequisite courses for Calculus I (MATH 155) and Fundamentals of Chemistry (CHEM 115). Based on the standardized test scores, the student is required to take algebra, trigonometry and chemistry courses with a minimum grade of C to move to MATH 155 and CHEM 115. The first two years provide a well-rounded education in English, Mathematics, Chemistry, Physics, and in the beginning courses, which stress fundamental engineering concepts, skills and methodology. General Engineering Courses introduce different fields of engineering and help to develop logic and problem solving skills through class projects, computer systems and software, etc. In order to be admitted into Industrial Engineering program the student needs to take ENGR 101, ENGR 102, MATH 115, CHEM 115 and ENGL 101 and attain minimum GPA of at least 2.0. The IE curriculum too would consider the student to be a part of the general engineering curriculum until he/she meets the requirements by the end of the freshman year. Table 3.1 describes the list of required courses in the freshman year.

Table 1-2: Freshman Year Courses

<table>
<thead>
<tr>
<th>Semester 1</th>
<th>Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 126 and MATH 128 or MATH 129</td>
<td>6 or 4</td>
</tr>
<tr>
<td>CHEM 110 Intro to Chemistry</td>
<td>4</td>
</tr>
<tr>
<td>ENGR 199 Orientation to Engr</td>
<td>1</td>
</tr>
<tr>
<td>PHYS 110 Engineering Physics</td>
<td>3</td>
</tr>
<tr>
<td>ENGL 101 Comp and Rhetoric</td>
<td>3</td>
</tr>
<tr>
<td>GEC Elective</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>16 - 18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester 2</th>
<th>Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 155 Calculus I</td>
<td>4</td>
</tr>
<tr>
<td>CHEM 115 Fundamentals of Chemistry</td>
<td>4</td>
</tr>
<tr>
<td>ENGR 101 Engr Problem Solving</td>
<td>2</td>
</tr>
<tr>
<td>GEC Electives</td>
<td>6</td>
</tr>
<tr>
<td>Total</td>
<td>16</td>
</tr>
</tbody>
</table>

The BSIE program requires the student to attain a minimum GPA of 2.0 or better for all the industrial and management systems engineering courses attempted. The GPA
calculation considers the D/F repeat rule where a student can repeat a course when the student have not attempted more than 60 hours of credit when the course is taken. The initial grade will be dropped from GPA computation and new grade will be considered even if the grades may be worse than the original grade. This requirement assures that the student has demonstrated overall competence in their chosen major. The student must take approximately 15 credit hours per semester to complete the B.S degree program in four years. Table 3.2, Table 3.3 and Table 3.4 describe the courses for second year, third year and fourth year respectively.

Table 1-3: Sophomore Year Courses

<table>
<thead>
<tr>
<th>Semester 3</th>
<th>Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 251 Multivariable Calculus</td>
<td>4</td>
</tr>
<tr>
<td>CHEM 116 or PHYS 112</td>
<td>4</td>
</tr>
<tr>
<td>MAE 241 Statics</td>
<td>3</td>
</tr>
<tr>
<td>ENGL 102 Comp. &amp; Rhetoric</td>
<td>3</td>
</tr>
<tr>
<td>IENG 200 Fundamentals of IE</td>
<td>1</td>
</tr>
<tr>
<td>IENG 220 Re-Engineering</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>18</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester 4</th>
<th>Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 261 Elem. Differential Equat.</td>
<td>4</td>
</tr>
<tr>
<td>MAE 243 Mech. of Materials</td>
<td>3</td>
</tr>
<tr>
<td>IENG 213 Engineering Statistics</td>
<td>3</td>
</tr>
<tr>
<td>IENG 377 Engineering Economy</td>
<td>3</td>
</tr>
<tr>
<td>GEC Elective</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>16</strong></td>
</tr>
</tbody>
</table>

Table 1-4: Junior Year Courses

<table>
<thead>
<tr>
<th>Semester 5</th>
<th>Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECON 201 Microeconomics</td>
<td>3</td>
</tr>
<tr>
<td>IENG 304 Materials and Costing</td>
<td>3</td>
</tr>
<tr>
<td>IENG 350 Intro. Oper. Research</td>
<td>3</td>
</tr>
<tr>
<td>IENG 360 Human Factors Engr.</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>15</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester 6</th>
<th>Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECON 202 Macroeconomics</td>
<td>3</td>
</tr>
<tr>
<td>IENG 302 Mfg. Processes</td>
<td>2</td>
</tr>
<tr>
<td>IENG 303 Mfg. Processes Lab</td>
<td>1</td>
</tr>
<tr>
<td>IENG 316 Ind. Quality Cont.</td>
<td>3</td>
</tr>
<tr>
<td>IENG 331 Computer Appl. IE</td>
<td>3</td>
</tr>
<tr>
<td>IENG 343 Prod. Plan &amp; Design</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>15</strong></td>
</tr>
</tbody>
</table>
## Table 1-5: Senior Year Courses

<table>
<thead>
<tr>
<th>Semester 7</th>
<th>Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE 221 Basic Electric Eng.</td>
<td>3</td>
</tr>
<tr>
<td>EE 222 Basic Electric Lab</td>
<td>1</td>
</tr>
<tr>
<td>ENG Tech. Elective</td>
<td>3</td>
</tr>
<tr>
<td>IENG 455 Simula. Digital Meth.</td>
<td>3</td>
</tr>
<tr>
<td>IENG 471 Design Productive Sys.</td>
<td>3</td>
</tr>
<tr>
<td>GEC Elective</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>16</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester 8</th>
<th>Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>IENG 472 Design Prod. Systems</td>
<td>3</td>
</tr>
<tr>
<td>IENG Tech. Elective</td>
<td>3</td>
</tr>
<tr>
<td>IENG 446 Plant Layout/Mat'l Hand.</td>
<td>3</td>
</tr>
<tr>
<td>Select 2 of the following courses</td>
<td>6</td>
</tr>
<tr>
<td>IENG Tech. Elective</td>
<td></td>
</tr>
<tr>
<td>MAE 242 Dynamics</td>
<td></td>
</tr>
<tr>
<td>MAE 320 Thermodynamics</td>
<td></td>
</tr>
<tr>
<td>MAE 331 Fluid Mechanics</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>15</strong></td>
</tr>
</tbody>
</table>

The total number of credit hours account for 129. the required set of courses are CHEM 115, ECON 201, ECON 202, ENGR 101, ENGR 199, ENGR 213, ENGR 220, ENGR 302, IENG 303, IENG 314, IENG 316, IENG 360, IENG 446, IENG 455, IENG 471, MATH 155, MATH 156, MATH 261, MATH 271 and PHYS 111.

The technical electives are MAE 241, MAE 242, MAE 320, MAE 321, MAE 360, MAE 455 and set of Industrial Engineering courses [11].

### 1.4 Data Mining

Data mining is a process of analyzing data from different perspectives and developing some useful information. The data mining methods help extraction of hidden patterns and predictive information, which cannot be identified by naked eye, from the data provided which will help organizations to focus on their weakness and achieve their goals. It also allows the user to categorize data, summarize, and identify the relationships among different data fields.
Data mining consists of three stages namely, data, information and knowledge. Data is the raw input in the form of numbers, text or facts that can be processes by the data mining technique. In recent years all the organizations are focusing on saving data in different formats and databases. The recent developments in data storage software’s such as Oracle, Microsoft SQL Server, My SQL, Sybase etc, help organizations build their own database to store structured data. These include, operational data such as transactions, sales, inventory, payroll, accounting etc. non operational such as industry sales, forecast data, economic data etc. Meta data – data about the data itself such as logical data base design, definitions etc. The information is generated by the data mining tool provided the data. The information is the developed patterns, associations, or relationships among the data. This helps the user to visualize the statistical information out of the analyzed data. Knowledge is extracted from the information obtained from the data mining methods. The user can visualize the new patterns or trends from the analyzed data.

1.5 Data Mining Tools

There are many data mining tools available in the market and also as a free source. Some of these are Microsoft SQL Server Analysis Services, Clementine, SAS- Enterprise Miner, CART, S-Plus, Weka, COGNOS, PRW, JMP etc. All these tools use different algorithms and a few are in common and require prepared and preprocessed data as input. The applications of these data mining techniques include health care sector, Science and technology, national security, Ad revenue forecasting, credit risk analysis, fraud detection, quality control, education, and product and sales etc.

1.6 Data Mining Algorithms

Data mining algorithm is a set of rules used to create a data mining model. Different types of algorithms include classification algorithms which are applied to the data set with one or more discrete variables, regression algorithms are applied to the dataset with continuous predictable variables, segmentation algorithms divide the data set into groups or clusters based on the similar properties of the data, and association algorithms find correlation between the different attributes in the dataset. The algorithms available in Microsoft Analysis Services are Microsoft Decision Trees algorithm,

The Data Mining sub part of this system is an introduction of data mining feature available to the department administrator user. Data mining algorithms were applied to the student data, available in the database. The user would have no other options to provide customized data, or apply various algorithms to the data. Microsoft Decision Trees algorithm, Naïve Bayes algorithm and Logistic Regression algorithm were applied to the data in the table “Data for Data Mining” in the database and the Mining Legend, which shows the accuracy of the applied mining model, is shown. Section 1.6.1 and 1.6.2 describes how a simple decision trees algorithm and Microsoft Decision Algorithm work.

1.6.1 Simple Decision Trees

A decision trees is a tree structure with branch nodes representing a choice of alternatives and nodes which represents a decision. Building a decision tree manually with large data is a tedious process. There are several algorithms and software’s available to construct a decision tree provided the data set. A decision tree starts with a root node which is usually the predictable variable or response variable. From the root node the algorithm splits each node recursively till all attributes are justified and finally a decision tree in which each branch represents a different possible scenario is constructed.

For example, in an exclusive-or problem [18], let us consider the following conditions:
If X=1 and Y=0 then class A
If X=0 and Y=1 then class A
If X=0 and Y=0 then class B
If X=1 and Y=1 then class B
A simple decision tree is shown in the figure below.
1.6.2 Microsoft Decision Trees

The Microsoft Decision Trees algorithm consists of different methods for creating a tree; it supports analytic tasks, such as regression, classification and association. It also supports modeling of both discrete and continuous variables. It uses different algorithms to give a best tree that fits the data. The method selected would depend on the task and the methods are linear regression, classification and association analysis. The tree can have multiple branches and the shape and depth of the tree depends on the scoring method and number of attributes and values present in the dataset.

This algorithm builds the tree by first calculating the correlation between all input variables and the predictable variable and selects the one which most cleanly separates the outcomes. Then the point of separation is calculating the information gain. The variable or attribute that has high information gain is used to divide the class into subsets which are then recursively analyzed by the same process until the tree cannot be split any more [19]. The equation to calculate information value or entropy for different cases within an attribute is shown below [18].

\[
\text{entropy}(P_1, P_2, ..., P_n) = -P_1 \log P_1 - P_2 \log P_2, ..., - P_n \log P_n
\]
Where, \( P_1, P_1, \ldots, P_n \) are arguments and expressed as fraction which add up to 1.

In the input dataset if the predictable column is discrete the algorithm automatically converts all the input values of continuous variables to discrete using discretization methods. If the predictable column is continuous, feature selection is applied to reduce the maximum number of outcomes. Microsoft Decision Trees uses Learning Bayesian Networks for dataset with discrete predictable columns and for continuous predictable columns it uses Auto Regressive Tree Models for Time-Series Analysis [19]. This method is applied to the student data and the results were discussed in Section 4.3.

### 1.7 Problem Statement

The objective of this research is to develop a web based system for course instruction and student advising. This research work will provide the students of IMSE Department at WVU an ideal web based student advising system. This web based system has Microsoft SQL Server as back end database will store all the information related to different users, courses, colleges and departments in well designed and normalized tables with proper entity relationships. This helps the advisor to pull the information of his/her advisee’s during course registration which helps in the analysis of his/her academic performance. The tool also short lists the courses to be taken for the next semester by checking the completed courses with a decent grade and also the GEC objectives to be fulfilled according to the BS program requirements. The advisor can check student’s graduation eligibility by verifying the information provided by the system such as, GEC requirements check table, Technical electives check table and GPA table which shows departmental GPA, Institutional GPA, number of credit hours completed including transferred credits from other institutions. The advisor can also add and update useful links that are posted on the website.

The course instruction part of the system will be a useful tool for the users of the system in different ways. The four different user types for the system are Department Administrator, Instructor/Advisor, Teaching Assistant and Student. The admin can manage the users, course catalog for each semester, instructors and their courses,
advisee’s and advisors in the system. This tool helps instructor manage online course material, post midterm grades online, keep track of past semesters class grades. The teaching assistant user has access to certain methods that an instructor have such as, manage students for the course, manage notes, and post mid term grades. The student user has access to the class notes for registered courses on the website and enrolled in the course. Students can view their grades online, class statistics, grading scheme, and useful links posted by the advisor. The student can also get advice such as, which courses to take next semester from the system which will check his/her transcript and compare to the BSIE curriculum and provides a list of courses he/she is eligible to take. The list is built by even checking the pre requisites satisfactory conditions for the courses.
CHAPTER 2: LITERATURE REVIEW

2.1 Academic Advising

A thesis entitled “The Changing Advising Needs of Undergraduate Students” was submitted by Mary E. Taylor to the Faculty of the Virginia Polytechnic Institute and State University. Her study was focused on determining if the advising topics between students and advisors vary by academic level, gender, or race. Conclusions were made after students and advisors were interviewed independently on their advising sessions, which suggest that the advisors spent little time on answering routine academic questions [5].

Based on the information gathered on student satisfaction, lack of faculty participation, adviser to advisee ratios, and course selection by an outside consultant at University of Wisconsin – Oshkosh made a commitment in 2002 to improve advising and other student academic support services and established a advisory council for academic advising to address the campus advising issues [6].

In his seminal work, Tinto [7] describes retention not as a goal but as a by-product of a successful and engaging college experience. The Association of American Colleges and Universities (AAC&U) also notes that in addition to retention and graduation rates, engagement of students in their own learning to be successful both professionally and civic life, are the key factors in student satisfaction and persistence toward graduation.

Indeed, in “Making the Most of College”, Richard Light [8] found that academic advising was perhaps one of the most “underestimated characteristics of a successful college experience”, adding further evidence that academic advising, when done well, adds value to the college experience.
Grand Valley State University has been the fastest growing university in the state of Michigan which led the student advising a challenging task. The rapid growth in student admissions has increased complexity in advising role of faculty. Thus, the university introduced database management systems and the web based tools. The introduction of IDMS-R provided the information such as, student transcripts, academic assessment progress reports, graduation requirements etc., at their desktop. This produced several advantages like student and faculty satisfaction, accurate updating of student information and timely graduation with less clerical work [2].

To provide the best possible guidance for undergraduate students, the department of computer science at University of Arkansas introduced a prototype expert system written in C-Prolog. The system uses a relational database model to store student’s records. Modules written to suggest free elective courses based on student interests and career goals [4]. The scheduling algorithm in the system provides the list of courses to be taken considering the courses recommended, course schedule and the BS program rules.

In 2007, an advising tool was developed in the IE department at WVU for Advising Undergraduate Students [9]. It assists students by giving comprehensive academic guidance in a consistent way and faculty in better utilizing the resources to keep a track on student’s academic progress. The administrator accesses, updates and manages the information in the database. However the tool was an idealistic model as the field names of the courses in the tool and those in the university database were not in coherence with each other. In addition the database did not meet the criteria of first normal form.

In 2008, A Software Tool for advising BSIE students was developed [30]. This tool supports three types of users, the Student, Instructor and Administrator. The student can successfully register with the pre requisites and co requisites specified for any given course. The Advisor can view detailed student information and the progress towards graduation. The administrator can successfully update and delete the data related to the students and instructors in the database. The administrator is given information about classes with room size less than class size which would appropriately allow the university
to avail the classroom with appropriate accommodation for all the students. However this system is a Windows-Based advising system which cannot be accessed through internet. The user needs to have the application on his local machine to have access to it.

The system being used at College of Engineering and Mineral Resources at WVU was built in 1992 in DOS environment. Its main objective is to provide timely information to students and advisors so that proper decisions can be made when registering for courses and planning academic progress toward a degree. Secondary objectives are to provide documentation for ABET inspections, to generate accurate reports concerning student status, class standing, and to develop a database which can be used for specific queries and statistical analyses of academic records for selected groups of students.

The COEARS consists of information stored in dBase files which are periodically updated by data transfer and by operator entry. Updates are done by electronic data transfer from WVU Admissions and Records (ARC) grade database, or from ARC-provided grades on diskettes. Codes are written in Microsoft PDS 7.1 BASIC to transfer data from file to file and perform calculations. Printed reports are generated for students and advisors [10].

The student and Technology Achieving Results (STAR) website is a resource that has been established by WVU for the registration of courses, accessing student accounts, viewing grades, and other information. The academic advising is restricted to providing only a PDF format file to inform the student with regards to registration of courses.

2.2 Web Based Student Advising

In 2004, University of California, Riverside developed a student data warehouse, an efficient report writing tool that enables users to access student data, create customized reports and send email to current students. Later in 2006, it released new version of it called Student Academic Information System (SAIS) which has a tool called Student Data Query System (SDQS), which enabled users to obtain data in a flexible, efficient and user friendly way [20]. Apart from this the UCR also have Student Academic
Advising System (SAAS) which is designed to provide information to instructors or advisors regarding individual students current enrollment, grades, earned units, and other academic information. The system also enables the advisors to run a degree audit against student’s major for graduation eligibility and record notes for future reference.

The Baylor University Waco, Texas has Unified Advising System (UAS) which provides information about their advisee’s. The information includes degree audit, student transcript, academic profile, advising history, and registration information. The university also has several other systems for students and faculty such as, BEARWEB-provides access to personal information such as addresses, phone numbers, emergency contacts, billing account summary, financial aid, transcripts, and registration for classes. Banner system utilizes SIS to provide information about students, prospects and applicants. ‘Chair SIS’ provides access to the department chairs and administrative assistants to get information about departments classes, schedules, students, grades etc. [21].

Course Planning Consultant (CPC) is a web based system developed for General engineering and Industrial Engineering students at University of Illinois, Urbana. This tool helps students in course selection for next semester by evaluating courses completed and prerequisite courses satisfactory condition. This helps reduce the risk of graduation delays [22].

IUCARE at Indiana University allows students to view their course history and compare it to the requirements of degree program. The students can also produce advising reports for their current majors or any other special programs. The students can see how in progress courses apply to their advising report [23].

The development of web based tool known as E-Coach which is a combination of three components, ED Doctor, Quality, and Quick helped the Texas Tech University provide student advising more objective and efficient. ED Doctor is used for education and professional development, Quick Advisor for smart advising, scheduling, and
registration, and Quality Assessment for assessment of educational processes. The project was developed by student summer interns and a graduate student all funded by Texas Higher Education Coordinating Board [24].

The Engineering Degree Monitoring System (EMS) at Michigan State University is a web based system to aid students in progress towards their degree. The system is accessed by students to view their current GPA, cumulative GPA, technical and required courses completed, and courses needed for graduation. The faculty and staff also use the system for making scheduled advising appointments, accessing student information, waivers, course lists, and graduation certification information [25].

2.3 Data Mining on Student Data

A data mining research was performed in the College of Engineering at Tennessee Technological University, Cookeville, Tennessee to identify the retention of engineering students through BS Degree program. Data mining was done to find the retention rate using ACT scores (Math, English and Composite) as the predictor of success. The information was obtained for over 1000 records includes; quarter and year entered, sex, residency status, ACT scores, final GPA, quarter and year leaving the university. Linear relationship and Polynomial relationship models were then applied to analyze the correlation between the two variables and the final result showed that 37.7 percent of retention rate was observed. The results further showed that ACT math, English and composite scores are good predictors of success in engineering [16].

University of Oklahoma strictly concerned about increasing the graduation rates and statistic research showed that only 51 percent of freshman in fall 1995 graduated within six years. The university considered this would cost the university in loss of revenue. This led to perform a data mining techniques like neural networks and Support Vector Machines (SVM) on the student data. The data was collected based on a survey with a set of questions which are attitudinal and demographic in nature. The obtained data was then classified, normalized and analysis was performed. The results show that one – third of the students did not obtain their degree within six years [17].
Cluster analysis and predictive analysis using Neural network, Logistic approach and decision trees was applied to the enrollment data at Sebha University, Libya [18]. The results has shown that higher percentage of female students graduated in the area of science, arts and medical and higher percentage of male students graduated from sport and law majors. The cluster analysis also showed that 66% of Law students were in university residence compared to 34% were outside campus.
3.1 Database Definition

A database is a structured collection of related data which is stored in a book, diskette, a computer or software such as DBASE, SQL Server, and ORACLE etc. The database is designed, built and populated with data for a specific purpose. The database technology plays a critical role in almost all areas where computers are used, including, business, engineering, military, medicine, law and education etc. The structure for the collected data is achieved by organizing the data according to a database model. A database relies on software known as Database Management System (DBMS) to organize the storage of data. Database Management Systems is a collection of programs that enables users to create and maintain a database and categorized according to the query languages that are available to access the database.

3.2 DBMS Architecture

The main characteristics of the database approach are, insulation of programs and data, support multiple users, and a schema to store the database description. This is called three-schema architecture. The three schema architecture consists of three levels, internal schema- describes the physical storage structure of the database, Conceptual schema- describes the structure of whole database for a community of users and also hides the information about the internal schema, External schema-describes a part of the database that a particular user group is interested and hides rest of the database. Figure 3-1 shows the hierarchy of the three schema architecture.
The DBMS are generally categorized as network, hierarchical, and object-oriented, and relational. The network model represents data as record types and a set type relationship. The hierarchical model represents data as hierarchical tree structures where each hierarchy represents number of related records. The object oriented model represents database in terms of objects, their properties and their operations. The objects with same behavior and structure belong to a class and these classes are organized into hierarchies. The relational data model represents database as a collection of tables, where each table can be stored as a separate file. The concept of relational database was first implemented by Dr. E. F. Codd, an employee of IBM Corporation, in 1970. The relational databases have a high level query languages and support limited form of user views.

The Desktop RDBMS are classified as shared-file systems, because they store their data in conventional files that multiple users can share on network. Earlier relational models like FoxPro, dBase, Clipper, and Paradox require a multitude of individual files to
store application and data objects. Today, almost every multi-user application is divided into a front-end file which contains application objects and links to a back-end file that holds the data.

Various implementations of the relational database model are available on RIM, RBASE 5000, PARADOX, OS/2 Database Manager, Dbase IV, CLIPPER, FOXBASE, and XDB. These were single user systems, ran under DOS, did not support SQL and called as Desktop RDBMS. These products later became multi-user versions such as Access (Microsoft, Inc.), SQL Server (Sybase, Inc) and SQL Server (Microsoft, Inc.) and called as Client/Server RDBMS.

3.3 Client/Server Architecture

The client-server architecture consists of a server machine which functions as network hub for other computers in the network called clients. The server computer is usually a large capacity computer with large amount of data storage and functionality. The client computers are small range computers, usually desktops and laptops within the network. The server computer will have the applications, software and connections to the database and clients are required to log on to the application to get access to the database or files or use any application on the server. For example, a web application hosted on a server which has SQL server as a back end database, the clients/users access the application through web browsers by logging in at the homepage. The client front end provides a graphical user interface (GUI) for data entry, display and reporting. Only the SQL statements and the specific data requested by the user pass over the network and the results are sent back through the network for display on the client computer. Figure 3-2 describes the general architecture of client/server architecture.
3.4 Normalization of Database

Designing tables for relational database follow a formalized procedure called normalization. The tables are normalized in a series of steps called normal forms. This helps in removing data redundancy, establishing proper relationships and improves performance of the database. Usually, the database should comply with the first normal form then the second normal form and then the third normal form. Normalizing all the tables in the database may not be necessary; in fact this would lead to unnecessary number of tables which in turn makes the database inefficient in its functioning [13].

3.4.1 First Normal Form

This requires tables to be flat and have no repeating or potentially repeating fields or groups of fields. A flat table is one in which every record has the same number of fields and a single field cannot contain multiple data values. All the tables in the student advising database have the same number of entities. The entities are grouped based on the user role, courses, colleges, departments, and miscellaneous. The fields related to student
are further grouped based on the category like educational, personal information etc. Similarly the courses are further grouped into pre-requisites and co-requisites.

### 3.4.2 Second Normal Form

This requires that data in all non key fields be fully dependent on the value of a primary key. This avoids data redundancy in the tables. In the student advising database each and every table has its own primary key and all the non key fields are dependent on the value of the primary key. For example, in the student table all the fields are related to student and are dependent on the primary key field ‘User ID’.

### 3.4.3 Third Normal Form

This requires that data in all non key fields of the table be fully dependent on the value of the primary key and describe only the object that the table represents. As discussed earlier, all the entities are grouped based on the user role, courses, colleges, and departments and miscellaneous, the database satisfies the third normal form.

### 3.4.4 Fourth Normal form

This requires that tables not contain fields for two or more independent, multi-valuated facts. This rule requires splitting tables that consist of lists of independent attributes.

### 3.4.5 Fifth Normal form

This involves further reducing redundancy by creating multiple two-field tables from tables that have more than two foreign keys.

The student advising database satisfies the first, second and third normal forms. A database is said to be well normalized if it satisfies the first, second and third normal forms.

### 3.5 Database Implementation

The database is implemented in SQL Server 2008. The tables satisfy the first three normal forms of the database design. The database is based on the requirement of the system which is to store and provide student information to different users of this
system. The data is stored in a structured way so that the users can apply different methods designed to meet their needs. Several stored procedures were written to meet specific needs and respond when called from the server side program to provide the required information.

### 3.5.1 Logical Schema of the Database

A logical schema is a collection of database objects such as tables and views, and defines a structure for the data fields having master levels (Primary key fields). This helps in building customized forms, reports and queries [14]. The parent table contains a primary key which acts as a foreign key in another table. The other table is known as a dependent table. Several new tables have been introduced for a more reliable structure and suitable for a web based student advising system. Table Users, Registered Courses, Transcripts, and User Info are the new tables introduced to reduce data redundancy and support multiple user roles like Administrator, Instructor/Advisor, Teaching Assistant and Student. Referential Integrity was enforced between every relationship between two tables i.e. when a row is deleted from the parent table, the database checks to see if there are any dependent rows matching foreign key constraints in the dependent tables and deletes the rows.

### 3.5.2 Table Users

The fields of Users table are shown in Table 3-1. This is a parent table which stores the user’s login information such as User ID, Password, User Type, Personal Information, Blocked, Department ID and Advisor Comments for students. The User ID is the primary key for the table. Every user of the system is given a unique User ID and all the information will be tracked with this field. The user will use this to login into the system. The User will also be given a default password while creating the account and it can be changed anytime later to his/her own word or phrase. The field ‘User Type’ stores whether the user is Administrator, Instructor, Teaching Assistant or a Student. After logging in, the system identifies the user by checking the ‘User Type” field and re-directs to his/her home page. The Blocked field describes whether the user has access to the website or blocked. The administrator has the capability to block or give access to the
website. The “AdvComments” field is used by the advisor to add comments during the advising session. The advisor comments could be the suggested courses, or any other information given to the advisee. The advisor can later pull the advisee record and check what information he had given earlier.

**Table 3-1 Login Table Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserID</td>
<td>nvarchar(50)</td>
<td>Unique ID for Users (PK)</td>
</tr>
<tr>
<td>Password</td>
<td>nvarchar(25)</td>
<td>Users Password</td>
</tr>
<tr>
<td>WVUID</td>
<td>int</td>
<td>Unique ID from WVU</td>
</tr>
<tr>
<td>UserType</td>
<td>nvarchar(25)</td>
<td>Admin, Instructor, Teaching Assistant, Student</td>
</tr>
<tr>
<td>FirstName</td>
<td>nvarchar(25)</td>
<td>Users First Name</td>
</tr>
<tr>
<td>LastName</td>
<td>nvarchar(25)</td>
<td>Users Last Name</td>
</tr>
<tr>
<td>Phone</td>
<td>nvarchar(25)</td>
<td>Users Phone Number</td>
</tr>
<tr>
<td>Blocked</td>
<td>bit</td>
<td>Blocked to use the system (true/false)</td>
</tr>
<tr>
<td>Email</td>
<td>text</td>
<td>Email address of User</td>
</tr>
<tr>
<td>DeptID</td>
<td>nchar(10)</td>
<td>DepartmentID of the users</td>
</tr>
<tr>
<td>AdvComments</td>
<td>nvarchar(MAX)</td>
<td>Comments added by advisors for their advisee's</td>
</tr>
</tbody>
</table>

3.5.3 Table Courses

The fields of Courses table are shown in Table 3-2. This table stores the information about courses available, number of credit hours. The blocked field describes whether the course is offered or not. The administrator user will have access to this table and he/she can edit the information in this table.
Table 3-2 Courses Table Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseID</td>
<td>nvarchar(25)</td>
<td>Combination of Subject and Number (PK)</td>
</tr>
<tr>
<td>CourseTitle</td>
<td>nvarchar(50)</td>
<td>Title of the course</td>
</tr>
<tr>
<td>Credits</td>
<td>int</td>
<td>No. of Credit Hours</td>
</tr>
<tr>
<td>Blocked</td>
<td>bit</td>
<td>Course offered (true/false)</td>
</tr>
</tbody>
</table>

3.5.4 Courses Schedule Table

The courses schedule table stores the information regarding the courses such as, instructor of the course, CRN number of the course, class timings, classroom details and the blocked field represents if the course is being offered for the current semester or not. The administrator user will have access to the table to edit any information in this table.

Table 3-3 Course Schedule Table Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseID</td>
<td>nvarchar(25)</td>
<td>Combination of Subject and Number (PK)</td>
</tr>
<tr>
<td>Term</td>
<td>nchar(10)</td>
<td>Terms the course is offered</td>
</tr>
<tr>
<td>UserID</td>
<td>nvarchar(50)</td>
<td>Instructor for the course</td>
</tr>
<tr>
<td>CRN</td>
<td>nchar(10)</td>
<td>CRN given by the University</td>
</tr>
<tr>
<td>SectionNo</td>
<td>nchar(10)</td>
<td>Section Number given by the University</td>
</tr>
<tr>
<td>Timings</td>
<td>nchar(255)</td>
<td>Class timings</td>
</tr>
<tr>
<td>Days</td>
<td>nchar(10)</td>
<td>Week Days of the class</td>
</tr>
<tr>
<td>BldgID</td>
<td>nchar(10)</td>
<td>BuildingID of the classroom</td>
</tr>
<tr>
<td>RoomID</td>
<td>nchar(10)</td>
<td>RoomI of the Classroom</td>
</tr>
<tr>
<td>DeptID</td>
<td>nchar(10)</td>
<td>DepartmentID the course offered by</td>
</tr>
<tr>
<td>Status</td>
<td>bit</td>
<td>Course offered (true/false)</td>
</tr>
</tbody>
</table>

3.5.5 Courses Pre Requisites Table

The fields of courses pre requisites table are shown in Table 3-4. This table stores the information related to the courses, course pre requisites, equivalent courses for the pre requisite, grade required in the prerequisite, academic year in which the student is eligible to take the course, and required number of credit hours completed for the eligibility to take the course. This information is useful to suggest courses to student for next semester.
### Table 3-4 Courses Pre Requisites

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>Unique ID for records (PK)</td>
</tr>
<tr>
<td>CourseID</td>
<td>nvarchar(25)</td>
<td>Combination of Subject and Number (FK)</td>
</tr>
<tr>
<td>PreReqCourseID</td>
<td>nvarchar(25)</td>
<td>Pre requisite for the course (FK)</td>
</tr>
<tr>
<td>PreReqEqui</td>
<td>nvarchar(25)</td>
<td>Equivalent course for pre requisite</td>
</tr>
<tr>
<td>Grade</td>
<td>nchar(10)</td>
<td>Grade criteria if any required</td>
</tr>
<tr>
<td>Standing</td>
<td>nchar(50)</td>
<td>Academic Year course can be taken</td>
</tr>
<tr>
<td>CreditHrs</td>
<td>nchar(10)</td>
<td>Credit hours for the course</td>
</tr>
</tbody>
</table>

### 3.5.6 Courses Co Requisites Table

The fields of the courses co requisites table are shown in Table 3-5. This table stores the information related to the co requisites for all courses in the Courses table.

### Table 3-5 Courses Co Requisites Table fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>Unique ID for records (PK)</td>
</tr>
<tr>
<td>CourseID</td>
<td>nvarchar(25)</td>
<td>Combination of Subject and Number (FK)</td>
</tr>
<tr>
<td>CoReqCourseID</td>
<td>nchar(10)</td>
<td>Pre requisite for the course (FK)</td>
</tr>
</tbody>
</table>

### 3.5.7 Course Equivalent Table

The fields of Course Equivalent table are shown in Table 3-6. This table stores the information of all the equivalent courses for courses in IE required courses. This information will be used while checking the student transcript against the IE required courses.

### Table 3-6 Course Equivalent Table Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>Unique ID for records (PK)</td>
</tr>
<tr>
<td>CourseID</td>
<td>nchar(10)</td>
<td>Combination of Subject and Number (FK)</td>
</tr>
<tr>
<td>EquivalentCourseID</td>
<td>nchar(10)</td>
<td>Equivalent for the course (FK)</td>
</tr>
</tbody>
</table>
3.5.8 GEC Requirements Table

The fields of GEC requirements table are shown in Table 3-7. This table stores the information related to the GEC required courses in the WVU University. This table information will be used to check the student transcript against the IE program and Institutional GEC objective satisfaction criteria.

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>Unique ID for records (PK)</td>
</tr>
<tr>
<td>CourseID</td>
<td>nvarchar(255)</td>
<td>Combination of Subject and Number (FK)</td>
</tr>
<tr>
<td>Title</td>
<td>nvarchar(255)</td>
<td>Title of the course</td>
</tr>
<tr>
<td>Obj1</td>
<td>nvarchar(255)</td>
<td>Objective number and 'X' if the course satisfies</td>
</tr>
<tr>
<td>Obj2</td>
<td>nvarchar(255)</td>
<td>Objective number and 'X' if the course satisfies</td>
</tr>
<tr>
<td>Obj3</td>
<td>nvarchar(255)</td>
<td>Objective number and 'X' if the course satisfies</td>
</tr>
<tr>
<td>Obj4</td>
<td>nvarchar(255)</td>
<td>Objective number and 'X' if the course satisfies</td>
</tr>
<tr>
<td>Obj5</td>
<td>nvarchar(255)</td>
<td>Objective number and 'X' if the course satisfies</td>
</tr>
<tr>
<td>Obj6</td>
<td>nvarchar(255)</td>
<td>Objective number and 'X' if the course satisfies</td>
</tr>
<tr>
<td>Obj7</td>
<td>nvarchar(255)</td>
<td>Objective number and 'X' if the course satisfies</td>
</tr>
<tr>
<td>Obj8</td>
<td>nvarchar(255)</td>
<td>Objective number and 'X' if the course satisfies</td>
</tr>
<tr>
<td>Obj9</td>
<td>nvarchar(255)</td>
<td>Objective number and 'X' if the course satisfies</td>
</tr>
<tr>
<td>Blocked</td>
<td>bit</td>
<td>True/False</td>
</tr>
<tr>
<td>Comments</td>
<td>nvarchar(255)</td>
<td>Comments</td>
</tr>
</tbody>
</table>

3.5.9 IE Required Courses Table

The IE Required courses table stores the information related to the required courses in the BSIE curriculum. This helps in comparing the students completed courses
to the IE required courses and suggest courses for next semester and check for graduation eligibility.

### Table 3-8 Required Courses Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseID</td>
<td>nvarchar(25)</td>
<td>Combination of Subject and Number (PK)</td>
</tr>
<tr>
<td>EquiValentCourseID</td>
<td>nvarchar(25)</td>
<td>Equivalent for the course (FK)</td>
</tr>
</tbody>
</table>

### 3.5.10 Tech Elective Courses Table

This table stores the information related to the technical elective courses in the BSIE curriculum. The course list in this table can be modified by the administrator.

### Table 3-9 Tech Elective Courses Table Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseID</td>
<td>nvarchar(25)</td>
<td>Combination of Subject and Number (PK)</td>
</tr>
</tbody>
</table>

### 3.5.11 Course Inst Table

This table contains the information related to the course instructors. The User ID field in the table represents the unique ID of the instructor in the system. Instructors can have multiple courses. The administrator user will have access to this table and change any information. The user interface of the administrator allows him to assign instructors to courses for a semester or block access to the course if he/she is no more an instructor for that course.

### Table 3-10 Course Inst Table Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>Unique ID for records (PK)</td>
</tr>
<tr>
<td>User ID</td>
<td>nvarchar(50)</td>
<td>User ID of Instructor (FK)</td>
</tr>
<tr>
<td>CourseID</td>
<td>nvarchar(25)</td>
<td>Combination of Subject and Number (FK)</td>
</tr>
<tr>
<td>TermID</td>
<td>nvarchar(25)</td>
<td>Current Term</td>
</tr>
<tr>
<td>Blocked</td>
<td>bit</td>
<td>Course blocked for the user (true/false)</td>
</tr>
</tbody>
</table>
3.5.12 Course Students Table

This table stores the information about the currently registered courses by students which are in the system. The administrator and the instructor user will have access to this table. The student’s User ID, Course ID and Term ID are stored in this table when instructor uploads class roster to the system for a particular course to create user accounts for registered students. The instructor can block the student from accessing the course material by checking the block field any time during the semester. He/She can also delete the student from the list if the student dropped the course but, the student is not deleted from the system.

Table 3-11 Course Students Table Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>Unique ID for records (PK)</td>
</tr>
<tr>
<td>UserID</td>
<td>nvarchar(50)</td>
<td>UserID of Student (FK)</td>
</tr>
<tr>
<td>CourseID</td>
<td>nvarchar(25)</td>
<td>Combination of Subject and Number (FK)</td>
</tr>
<tr>
<td>TermID</td>
<td>nvarchar(25)</td>
<td>Current Term</td>
</tr>
<tr>
<td>Blocked</td>
<td>bit</td>
<td>Course blocked for the user (true/false)</td>
</tr>
</tbody>
</table>

3.5.13 Course Notes Table

This table stores the information related to the course material uploaded by the instructor for a particular course. The “NotesFileName” field contains the file name and its physical path to the file on the server machine. The notes uploaded by the instructor can be distinguished by the term, and the instructor can also block/release the notes from being listed in the view notes page.

Table 3-12 Course Notes Tale Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>Unique ID for records(PK)</td>
</tr>
<tr>
<td>NotesFileName</td>
<td>nvarchar(255)</td>
<td>Filename and Location Path</td>
</tr>
<tr>
<td>CourseID</td>
<td>nvarchar(25)</td>
<td>Combination of Subject and Number (FK)</td>
</tr>
<tr>
<td>TermID</td>
<td>nvarchar(25)</td>
<td>Current Term</td>
</tr>
<tr>
<td>Blocked</td>
<td>bit</td>
<td>Notes blocked (true/false)</td>
</tr>
</tbody>
</table>
3.5.14 Transcript Table

The fields of Transcript table are shown in Table 3-13. This table stores the students completed courses, grades, term course taken, credits hours of the course, and quality points for the course. This table also stores the information of the transferred courses from other institution. If a course is repeated the Repeat field in the table stores information to which course grade to include and which grade to exclude while calculating the GPA.

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>Unique ID for records (PK)</td>
</tr>
<tr>
<td>UserID</td>
<td>nvarchar(50)</td>
<td>UserID of Student (FK)</td>
</tr>
<tr>
<td>TermID</td>
<td>nchar(10)</td>
<td>Current Term</td>
</tr>
<tr>
<td>CourseID</td>
<td>nvarchar(25)</td>
<td>Combination of Subject and Number (FK)</td>
</tr>
<tr>
<td>Credits</td>
<td>int</td>
<td>Credit hours for the course</td>
</tr>
<tr>
<td>Grade</td>
<td>nchar(10)</td>
<td>Grade Received</td>
</tr>
<tr>
<td>Qpts</td>
<td></td>
<td>Grade Points</td>
</tr>
<tr>
<td>Repeat</td>
<td>nchar(10)</td>
<td>Course Repeated (I/E)</td>
</tr>
</tbody>
</table>

3.5.15 Student Advisors Table

The fields of student advisors table are shown in Table 3-14. This table stores the information about the students and their advisors. The administrator will have access to this table. The user interface of the administrator allows him to assign advisors to the students. After successful login of an instructor/advisor, the list of his/her advisees WVU ID and Student Name are available to select to redirect to advising home page.

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>Unique ID for records (PK)</td>
</tr>
<tr>
<td>UserID</td>
<td>nvarchar(50)</td>
<td>UserID of Student (FK)</td>
</tr>
<tr>
<td>AdvisorUserID</td>
<td>nvarchar(50)</td>
<td>UserID of Instructor (FK)</td>
</tr>
</tbody>
</table>
### 3.5.16 College Table

The fields of college table are shown in Table 3-18. This table stores the information of different colleges in West Virginia University.

#### Table 3-15: Fields of College Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CollegeID</td>
<td>nchar(10)</td>
<td>Primary Key for College</td>
</tr>
<tr>
<td>CollName</td>
<td>nchar(30)</td>
<td>College Name</td>
</tr>
<tr>
<td>CollAddress</td>
<td>nchar(50)</td>
<td>Mailing Address of the College</td>
</tr>
<tr>
<td>CollPhone</td>
<td>nchar(20)</td>
<td>Phone Number</td>
</tr>
</tbody>
</table>

### 3.5.17 Department Table

The fields of department table are shown in Table 3-19. This table stores the information of all the departments of a college in the West Virginia University.

#### Table 3-16: Fields of Department Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeptID</td>
<td>nchar(10)</td>
<td>Primary Key for Department</td>
</tr>
<tr>
<td>CollegeID</td>
<td>nchar(10)</td>
<td>College to which Department belongs</td>
</tr>
<tr>
<td>DeptName</td>
<td>nchar(30)</td>
<td>Name of the Department</td>
</tr>
<tr>
<td>DeptAddress</td>
<td>nchar(50)</td>
<td>Mailing address of the Department</td>
</tr>
<tr>
<td>DeptPhone</td>
<td>nchar(20)</td>
<td>Phone Number</td>
</tr>
</tbody>
</table>

### 3.5.18 Useful Links Table

The fields of useful Links table are shown in Table 3-17. This table stores the information regarding the useful links posted by the administrator and advisor. These links can be updated and deleted by the user.

#### Table 3-17 Useful Links Table Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>Unique ID for records (PK)</td>
</tr>
<tr>
<td>Notes</td>
<td>nvarchar(50)</td>
<td>Description for the Link</td>
</tr>
<tr>
<td>LinkURL</td>
<td>nvarchar(MAX)</td>
<td>URL for the link</td>
</tr>
</tbody>
</table>
3.5.19 Data for Data Mining Table

This table is an independent table in the database. This table contains student’s academic information such as pre admission data ACT scores, high school GPA and grades obtained in few courses in IE department. This information is only used for data mining purpose in the system which is available only for the administrator user. The information in this table is not stored as actual grades but are categorized into good, average, bad depending on for the requirement that the data should be balanced for applying data mining methods.

Table 3-18 Data for Data Mining Table Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>float</td>
<td>Unique ID for records (PK)</td>
</tr>
<tr>
<td>ACT_Comp_C</td>
<td>float</td>
<td>ACT Composite Score</td>
</tr>
<tr>
<td>ACT_Math_C</td>
<td>float</td>
<td>ACT Math Score</td>
</tr>
<tr>
<td>HS_GPA</td>
<td>float</td>
<td>High school GPA</td>
</tr>
<tr>
<td>CHEM115</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>MATH155</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>ENGR101</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>MATH156</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>Engr102</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>Phys111</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>MATH251</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>MAE241</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>CHEM116</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>IENG200</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>IENG220</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>MATH261</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>MAE243</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>IENG213</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>IENG377</td>
<td>nvarchar(255)</td>
<td>Grade category (Good,Avg,Bad,N/A)</td>
</tr>
<tr>
<td>WVU_GPA_C</td>
<td>nvarchar(255)</td>
<td>WVU GPA</td>
</tr>
</tbody>
</table>

3.5.20 Course Grades Table

This is a table which can be created by the instructor at runtime to post mid term grades of a particular course during the semester. The structure for the table can be
defined by the user however, in order to have this table related to the schema of the database the table should include a field similar to the User ID which is a primary key field in users table. The instructor can just provide the excel file with any number of grade categories such as test, projects, home works etc. the system will set data types and creates the table with a name “coursetable” followed by the Course ID and Term ID. Whenever the instructor uploads grades with new grade categories the old table will be automatically deleted and creates a new table with new structure provided in the excel file. The student users from their interface can only see their grades with no access to edit the grades.

3.5.21 Course Grade Description Table

This table is also created at runtime while the course grades table is created. This table contains same fields as the course grades table, but the data stored would be the weights and out off values for each grade description provided in the excel file. The instructor will have access to the table to edit the grading scheme. This weights and out off values will be used to compute the final grade of the student. The name of the table will be “coursegradedesc” followed by the Course ID and Term ID. This table will be overwritten whenever the course grades table is recreated.

3.5.22 Entity Relationship Diagram

The relationship diagram below describes the relationship between various tables in the database. The relation between different entities can be one-to-one or one-to-many. The key symbol represents the field as the primary key and the infinity symbol represents the one-to-many relationship. As referential integrity between tables has been established, it ensures to check the values of the foreign or dependent values in tables when the values of the field in parent table are being altered.
3.6 Stored Procedures

The stored procedures are the Transact SQL statements which are simplified and stored as manageable block in the database. SQL server executes the stored procedure and saves the execution plan and reuses if the same procedure is recalled which helps in increased performance of database. The stored procedures reduces long SQL queries to single line codes an can be used by multiple clients and servers which helps in reduced client/server traffic and efficient reuse of code and programming abstraction. Stored procedures are very secured and only accessed by the users with granted permission. The
following are the description on each stored procedure developed for the system, which are very common T-SQL statements for multiple users.

3.6.1 SP Login

This stored procedure is called from the login page when a user selects “Login” button after providing his/her User ID and Password. The stored procedure has User ID and password as input parameters and returns a string value of User Type if the user exists and “User Not Found” if the user is not registered in the database. The program then re directs the user to his/her home page reading the value returned by the stored procedure.

3.6.2 SP Course Inst

The course instructor stored procedure is called after the user is successfully logged in. This procedure has User ID as input variable and returns the list of courses assigned to the instructor by the department administrator. The dropdown on the login page is populated by the list returned by the stored procedure. When the user selects a course from the list the user is re directed to his/her home page where he/she have access to the methods an instructor user can perform for a course in the system.

3.6.3 SP View Notes

The view notes stored procedure is called when an instructor, student or a teaching assistant user selects “View Notes” option from their home page. This procedure has Course ID and Term ID as input variables and returns the list of files uploaded and not blocked by the instructor. The dropdown available on the view notes page is populated by the list returned by the stored procedure and when the user selects an item in the list the program pulls the file from the server and opens it on the web page.
3.6.4 SP Inst View Grades

This stored procedure is called when an instructor user selects the option “View Grades” from the instructor home page. The input variables for this stored procedure are Course ID and Term ID, which returns course grades uploaded by the instructor for the particular course and term. The returned data is in the form of a data table which is in the same structure as the course grades table in the database and can be attached to a grid view for better view. The program allows the instructor user to edit/delete the grades.

3.6.5 SP Student View Grades

This stored procedure is called when a student user selects the option view grades from the student’s home page. This stored procedure has Course ID, Term ID, and User ID as input variables and returns the course grades uploaded and released by the course instructor. The procedure returns only the grades of the particular User ID and has no access to edit the grades. The values for the input variables are provided by the program from session variables. The returned results are bound to a grid view on the view grades page.

3.6.6 SP View Grade Descriptions for a Course

This stored procedure is called when a instructor, student, or teaching assistant user selects the option “View Grade Description” available on view grades page. The stored procedure has Course ID and Term ID as input variables and returns the information available in the particular course grade description table from the database. The returned results are in the form of a data table which can be bound to a grid view on the web page.
CHAPTER 4: SYSTEM ARCHITECTURE AND IMPLEMENTATION

4.1 Three Tier Architecture

This tool will be designed using three-tier design methodology. A 3-tier application is a program which is organized into three major disjunctive tiers [9]. These tiers are:

- Tier -1: User Interface (Front End)
- Tier - 2: Data Access (Server Programming)
- Tier - 3: Data Base (Backend).

Each tier can be deployed in geographically separated computers in a network. The tiers can be deployed on physically separated machines. The characteristic of the tier communication is that the tiers will communicate only to their adjacent neighbors. For example, The User Interface Tier will interact directly with the Data Access Tier and not directly with Data base or Data Tiers.

![Figure 4-1 Three-Tier Architecture](image)

Figure 4-1 Three-Tier Architecture [27]

One of the key elements of any application design is the system architecture. The system architecture defines how pieces of the application interact with each other, and
what functionality each piece is responsible for performing. There are styles of application architecture with each style being characterized through the number of layers between the user and the data. Each layer generally runs on a different system or in a different process space on the same system. With three-tier applications, the business rules are removed from the client and are executed on a system in between the user interface and the data storage system. The client application provides user interface for the system. The business rules server ensures that all of the business processing is done correctly. It serves as an intermediary between the client and the data storage. In this type of application, the client does not access the data storage system directly. This type of system allows for any part of the system to be modified without having to change the other two parts. Since the parts of the application communicate through interfaces, then as long as the interface remains the same, the internal workings can be changed without affecting the rest of the system [28].

A web based approach will enable the users to utilize the database from anywhere and at anytime. ASP.NET is Microsoft’s latest technology for building web-based applications and services, a successor to Active Server Pages (ASP) that draws on the power of the .NET Framework development platform and the Visual Studio.NET developer toolset [12]. ASP.NET provides an efficient approach to developing web applications because of the simplistic, effective approach in designing and creating user friendly interface while being able to write server programs within the design. The created applications can directly be implemented on the World Wide Web (WWW) server or any web servers available. The ability to support multiple languages and recover from memory leaks and errors helps it more efficient and reliable.

The web based system for course instruction and student advising is built in a three tier architecture in which, the user interface for the users of the system which are HTML Pages built using Visual Studio 2008, the data access tier which is server side programming written in Visual Basic scripting language and kept in separate file for each web page built for users, the database with tables, relationships, and stored procedures were built in Microsoft SQL Server 2008.
The user interface tier of the system has four different interfaces built for four user types: Department Administrator, Instructor/Advisor, Teaching Assistant, and Student. Different Master Page’s were built for four user types which contains the navigation menu to the web pages to perform particular methods which are discussed in Chapter 5. The data access tier has the server side program written in visual basic contains sub procedures and functions which are called on a button click event, from tier-1, to perform particular operation. For example, when an advisor called the method “Academic Performance” on a student, the program calls sub procedure “Get Student Transcript” which first interacts with database tier (tier-3) gets the student transcript and applies the logic and rules written in “Compute GPA” sub procedure and displays the results back on the user interface (tier-1). The data access tier in the system has various tables to store the student data and several stored procedures which are discussed in Chapter 3. The system architecture for the system is shown in Figure 4-2.

![Figure 4-2 System Architecture](image-url)
4.1.1 Department Administrator

Department Administrator Home page is gateway for this user. The department administrator is responsible for meeting the administrative, managerial, and technological needs of the student advising in his/her department. This user performs methods like Add/Update/Delete courses, and program requirements. He/she will perform the database administration. The administrator users will be able to Add/Update/Delete faculty, courses, and program requirements. They will be able to assign courses to faculty and maintain course information current. Considering the possibility that administrator user could also be an instructor, the methods on the administrator home page contains the option to view his/her courses which re-directs to the instructor home page where he can perform all the methods as an instructor.

4.1.2 Instructor

The Instructor home page will be the gateway for the instructor/faculty/advisor. The Instructor users will be able to set up accounts for students registered in their classes. They will be able to upload lecture notes, homework problems, and grades for various class activities. He/she can view the courses completed by his/her advisee’s to analyze their academic performance and check their graduation eligibility. The tool will also provide information on the courses completed and courses to be taken to ensure timely graduation referring to the BSIE curriculum. The advisor can post links to forms or other information on the web for student reference.

4.1.3 Teaching Assistant

The Teaching Assistant home page will be the gateway for these users. The teaching assistant user will have access to some of the methods designed for the instructor. This user can upload/update course content to the website. She/he can provide user access for students to the website. The teaching assistant can also upload course
grades after every homework/project/tests but he/she will not have access to edit any of the grades.

4.1.4 Students

The Student home page will be the gateway for student users. The student users will be able to view notes and grades for current semester courses. They will also be able to view final grades of courses completed. They can also query the system to suggest possible courses for next semester and run IE program audit which compares the completed courses to the BSIE curriculum and displays the courses left to do. The system also tells the user calculated departmental and institutional GPA, number of credit hours done and left to do. The methods include links to the BSIE curriculum for reference and other links posted by the advisor or the administrator.

The detailed description of each method available to different users is discussed in Chapter 5: System Validation which would give a walkthrough the web application with screen shots.

4.2 Data Mining Implementation

This sub system will introduce the data mining feature to the system. The system is implemented using Microsoft SQL Server Analysis Services (SSAS) which provides data mining functionality with a combination of server and client applications. It facilitates objects for development and deployment of business intelligence applications. It also provides a set of data mining algorithms for the users to mine their data to view patterns and trends in the data. SSAS is also extensible so that user can add custom algorithms to support particular mining needs. For the student data, which is mostly discrete data, the Naïve Bayes Algorithm gives better results compared to Microsoft Decision Trees.
4.2.1 Student Data

Applying data mining algorithms to the student data help predict the performance, or classify students based on their performance. At West Virginia University, any data related to student is stored in the STAR System. The data mining is applied only to the Industrial Engineering student’s data available in the STAR system. The data was obtained from the problem report submitted by Mr. Abhinav Gaddam [29]. His work was focused on classifying undergraduate students in Industrial Engineering department into high risk, medium risk and low risk categories based on their academic performance.

The data is stored in a table “Data for Data Mining” in the database, which is used in creating the data mining structure in Analysis Services and Microsoft Decision Trees algorithm was applied.

4.3 Data Mining Tasks

Microsoft SQL Server Analysis Services has built in data mining methods which can be applied to any application domain. Applying data mining includes several steps such as; defining the problem, preparing data, exploring data, building models, exploring models, and deploying models. Figure 4-3 shows the data mining process diagram. However, in this system only the tasks discussed below were performed.
4.3.1 Defining the Problem

This step includes defining the scope of the problem. This could be done by answering the general questions like; the purpose of data mining, what type of trends we are looking for, what kind of predictions we are looking for, what kind of data we have discrete or continuous, and how the attributes are related to the predicted variable. The scope of this system is to apply data mining algorithms to the student data of industrial engineering students analyze accuracy of algorithms applied on the student data.

4.3.2 Preparing the Data

This step includes collection of data for the data mining process. The data can be obtained in various formats like, excel files, from SQL queries, text files, Arff files etc. The main task in collection of data is identifying appropriate data for the problem defined. The data should not include inconsistent data, or no missing values, and all the parameters should be in correlation to the predictable variable. Data preprocessing is
performed on the raw data to prepare for data mining. This transforms the data into a format that will be more easily and effectively used by data mining techniques. The preprocessing of data includes, selecting a subset from a large population of data, manipulate the data to produce a single input, remove noise from data, organize the data for more efficient access and pull out specific data that is significant to a particular context.

In this system the data was obtained from [29] and was imported to a table in the database. Exploratory Data Analysis (EDA) was done on the student data in order to get a better understanding of the data and discover basic relationships or patterns. The description on the table “Data for Data Mining” was discussed in Section 3.5.19.

4.3.3 Building Models

After we have the data ready for data mining the next step would be creating a mining structure which consists of the data for applying mining models. Processing the mining structure, analysis services generates statistical information on the data that can be used for further analysis. A mining structure can contain any number of mining models. Bar charts and pie charts can also be produced in the mining structure. Figure 4-4 shows the screen shot of the Bar chart and Pie chart for the data in the table. The X, Y, and Z in the Bar chart of WVU GPA indicate different categories of the data which was obtained after Exploratory Data Analysis (EDA). X, Y, and Z indicate High, Medium, and Low respectively. Figure 4-5 shows the screen shot of the Pie Charts for the data in the table.
Figure 4-4 Screen shot of Bar and Pie Chart of WVU GPA and High School GPA Fields in the Table

Figure 4-5 Screen shot of Pie Charts of WVU GPA and High School GPA Fields in the Table
A mining model is an algorithm to which you specify the input columns, predictable columns, and parameters that tell the algorithm how to process the data etc. Processing the model is called training the model which applies the algorithm to the data in the mining structure to extract patterns. Figure 4-6 shows the screen shot of the mining models in the mining structure and their attributes.

Figure 4-6 Screen shot of the Mining Models in the Structure and their Attributes

4.3.4 Deploying Models

After building the models, exploring the model and validating helps to test their effectiveness to determine which model gives the accurate results. This can be done by providing the training set and testing set to the model and checking the accuracy of the predicted values. Comparison charts can also be produced in the SQL server analysis service. The models can be deployed and stored to perform many tasks such as predictions by providing testing data, building queries to retrieve statistical information from the model, embedding the mining model to an application with user interface, and
creating reports. The algorithms applied to the data were Microsoft Decision Trees, Microsoft Naïve Bayes, and Microsoft Logistic Regression.

After the successful deployment of the mining models, an accuracy test was performed on the models by providing a test data which is stored in a table named “Testing Data 1” in the database. The test data contains approximately 20 records with no missing values in the same structure as the training dataset. Microsoft Naïve Bayes and Logistic Regression algorithms were also added to the mining structure to compare the prediction accuracy results among the models. The results show that the prediction probability of Microsoft Naïve Bayes algorithm was 88.61%, Microsoft Decision Trees was 59.59%, and Microsoft Logistic Regression was 63.41% for the test data provided. The Mining Legend after running the accuracy test is shown in Figure 4-7 and Figure 4-8 shows the screen shot of the Lift Chart comparing the mining models.

![Mining Legend for Test Data](image.png)
This work has only provided the initiation to the data mining on student data using Microsoft Analysis Services. Further more detailed research can be done on the attributes in the dataset and apply different data mining algorithms and identify the most influencing variables on the student performance. Creating interface to provide customized data as testing dataset and an option to select the available mining algorithms would make the feature handy for student advisors.
CHAPTER 5: SYSTEM VALIDATION

The default web page for all the users in the system is the Login Page. The user (System Admin/Admin/Instructor/Student) provides his/her user id and password to have access to the system. The web page interacts with the database to verify and identify the user role to re-direct to his/her home page. The four different user types have four different home pages which were designed based on the tasks they perform on the website. Each page has the navigation bar with the method name created for the user to perform and re-directs to the respective web pages. Figure 5-1 shows the screen shot of the application home page.

Figure 5-1: Home Page
5.1 Department Administrator User

The role of the department administrator was described in Chapter 4. Upon validation of the user as administrator, the three different options will be available for the user to namely, select a course, select an advisee and “go to admin page”. If the admin user is also an instructor he/she will select a course from the list provided in a dropdown which re-directs to the instructor home page. Thus, the user will have access to all the methods available for the instructor. The admin can also select an advisee from the list of his advisees in the dropdown which re-directs to the student advising home page. The user can also select the option “go to admin home”, where the user will have access to all the methods available to perform as an admin. Figure 5-2 shows the screen shot of the web page for system administrator after successful login.

Figure 5-2 Web Page after successful login by admin
There are specific methods which the system administrator can perform on the website and are placed as a navigation menu on the home page. Figure 5-3 is a screen shot of the admin home page after selecting the option, “continue to admin home page”. The methods for system administrator are described as follows.

![Figure 5-3 Admin Home Page](image)

5.1.1 Home

This option re-directs the user to the web page where the user will have all three options available after successful login. The user can select this option from any web page he/she is working on. This allows the user to navigate through the system from admin page to the course instruction or to student advising part of the system.

5.1.2 Add New Course

This method helps the user to add a new course to the course catalog for the semester. The user will have to enter details of the course such as, Course Number, Title, Credit hours to successfully register the course in the catalog. The user will see a
“success” message if the course is successful added or an error defining why the course was not added to the catalog. The errors could be, if the course already exists in the catalog, not providing sufficient information etc. Figure 5-4 shows the admin add new course web page.

![Admin Add New Course Web Page](Image)

Figure 5-4 Admin add new course web page

### 5.1.3 Add New User

This method helps the admin to add a new user to the system. The administrator can add an instructor, teaching assistant, student, or another administrator individually. The admin user will have to provide all the information of the new user to successfully add to the system. The administrator will provide a user id and password to the new user. However, the password can later be changed by the user. Figure 5-5 is a screen shot of admin add user web page.
5.1.4 Manage Instructors

This method is used to manage instructors and their courses. The administrator assigns courses to the instructors every semester. This method shows the list of instructors in a dropdown list, all the courses in the catalog in a dropdown list and the courses already assigned to the instructor. The admin user will select a course from the course dropdown list and hit the add button to add it to the instructors course list. Thus, when the instructor user is logged in he/she will see the courses he/she was assigned. The admin can also block or delete certain courses which are already on the instructors’ course list. Figure 5-6 is a screen shot of the admin manage instructor’s web page.
5.1.5 **Manage Student Advisors**

This method is used to assign advisors to the existing or new students in the IE department. At this page the admin will see a whole list of students in the system and the advisors of the students. Like an advisor the administrator user can review any IE students’ academic performance by selecting the student from the dropdown list. This helps in taking a decision on assigning advisors to the students. To assign advisors, the admin user would have to select “Edit” button next to student User ID, which pulls all the available instructors in the system into a dropdown list from which the admin can pick an instructor and update the record assigning an instructor as an advisor to that particular student. Figure 5-7 is a screen shot of admin manage student advisors web page with a student’s academic performance report. Figure 5-8 is the admin manage student advisors web page.
Figure 5-7 Admin manage student advisors web page with a student academic performance

Figure 5-8 Admin manage student advisors
5.1.6 Manage IE Curriculum

This method is used to update the courses in the IE curriculum. The required courses, technical electives and general elective courses were stored in three separate tables. When the user selects the option manage IE curriculum, all the course list from the three tables were displayed in three different grid view’s with capability to edit and delete. The admin user can add new course to the curriculum by selecting a course from the dropdown list with all the courses available in the courses table in database. Figure 5-9 is a screen shot of Manage IE Curriculum web page.

![Figure 5-9 Screen shot of Manage IE Curriculum](image)

5.1.7 Update Course

This method is used to update course information in the course catalog. Only the administrator will have the access to perform this method. The admin user can change the course id, title, number of credit hours and block the course if it not being offered for any particular semester or even delete the course if it no longer offered. Figure 5-10 is a screen shot of admin update course web page.
5.1.8 Add Term

This method is only an administrative task which adds a new term to the database. These terms are available in the form of a dropdown list in several methods used by the students, instructors, and administrator. The web page displays a dropdown of years up to 10 from the current year, and terms namely, spring, summer, and fall. The admin user will select the year and term and hit add term which will be stored in the database.

5.1.9 View Grades

This method will allow the admin user to see the grades of students in the system. The user can select a course and term to view the grades. These grades will be pulled from the transcript table which contains student unique user id, course id, letter grade, credit hours, quality points achieved and if the course has been repeated or not. Figure 5-11 is a screenshot of admin view grads web page.
5.1.10 Change Password

This feature is available for all the user types in the system. The user can change their password any time once logged in. The user will have to enter his/her old password for verification and his/her new password to successfully change the old password.

5.2 Instructor User

After the user is authorized as an instructor from the Login Page the user will see the options select a course and select an advisee. The user will see the list of courses for which they are the instructor. He/She selects one course which then re-directs to the course instruction home page where he/she can perform several tasks. All the actions performed in the course home page are limited to the course the user selected before. Figure 5-12 is a screenshot of the web home page after successful login. If the instructor selects a course the user will be re-directed to the course instruction home page shown in Figure 5-13.
Figure 5-12 Options for instructor after successful login

Figure 5-13 Course Instruction Home Page
The methods available for the instructor in the course home page are described as follows.

5.2.1 View Notes

This method re-directs the user to the Notes page where the user can see the uploaded notes as tree nodes on the side bar of the page. For example, if the user selects a course IENG331 from the list, in the view notes page he/she can only see the notes uploaded for that particular course and having the status released. Figure 5-14 is the screenshot of view notes page.

![Figure 5-14 View Notes Page](image)

5.2.2 View Grades

This method is used to view the grades of the students currently enrolled in the course and also from the previous terms. The instructor can edit the grades by selecting edit option available next to each student user id. The instructor can release or block the grades from being viewed by the student. The user can also view the Histogram for each grade categories in the course grades table. The dropdown on the web page lists the different grade categories in the course grades table, the user would have to select any of
the listed categories and select “View Bar Chart” button which will generate the bar chart.

5.2.3 View Final Grades

This option allows the instructor to see the final grades for the course in selected term. The page displays a dropdown with list of courses the instructor is teaching and another dropdown with terms. The user can select course and term to see final grades. The instructor user does have an option to edit the grades. Figure 5-16 is a screen shot of the instructor view final grades web page. The method called finalize grade which is available in the instructor page should always precede this method. Because, the course grades table is a temporary table created at run time during the semester and at the end of the semester after assigning the final letter grades the instructor has to apply the method finalize grades which will copy the letter grades from the course grades table and save to transcript table.
5.2.4 Upload Notes

The user can upload notes for the course by browsing for the file on the local hard disk and hit upload. The system copies the file and saves it to the course folder and saves the file path in the database. All the information about the file is stored in the course notes table in the database which can be edited by the instructor. The user can upload all file types like “.mdb”, “.doc”, “.pdf”, “.zip” etc. And the user can upload and block the files to release later.

5.2.5 Update Notes

The user has an option to release/block notes from being viewed by students. This helps when the instructor uploads tests he can block them and only release at the time of the test. Figure 5-17 is a screen shot of the update notes page.
5.2.6 Upload Grades

This method is used to upload students grades currently enrolled for the course. These grades are stored in a temporary table in the database. The user can upload grades after every homework, project or test. He/she can upload grades from an excel sheet which is efficient and fast. The excel sheet has to be in a particular format with Student ID field necessary in order to connect to the schema of the database. The excel sheet format is available on the upload grades page as a download link for reference. The user just have to provide the excel file with all grade categories and data, the program automatically will copy the data from excel file and creates a course grades table in the data base called “course grades’ followed by the course id and term id. The program will also create a grade description table called “course grade desc” followed by course id and term id. The course grades table contains all the student records with grades where as the course grade description contains the weights and out of values for each grade category. This helps in computing the total score and letter grade for the course. These tables are however temporary tables which will be over written every time the grades are uploaded with a new grading scheme or different grade categories. The instructor will have an
option to finalize grades so that at the end of semester the user can save the final letter grades to transcript table. Figure 5-18 shows the screen shot of upload grades web page.

![Figure 5-18 Upload grades web page](image)

5.2.7 Add Class Roster

This method is used to add new students to the class. In an every new term the instructor can give access to the students registered for the course by entering their information into the database. This can be done by selecting the excel sheet with all the students information such as Student ID, Name, and other personal information. This method automatically creates the User ID and Password for the student. The User ID will be their Student ID and a default Password. The students can change their password anytime after login. If the student is already in the system, the student is just added to the course students table where the student’s courses are saved and he/she can use their existing User ID and Password to access the course. If the is new to the system the method will then create a User ID and Password. Before uploading the class roster the
instruction have an option to see the complete list which is being uploaded to the system. The instructor can later edit, block, or delete students from the roster.

Figure 5-19 Instructor upload class roster web page

5.2.8 Update Students

This method is used to edit or delete the existing student users. The instructor can only edit or delete the information of the students registered for the particular course he/she selected on the instructor home page. The instructor can also block the individual user from having access to the course, rather than deleting the user completely. Figure 5-20 shows the instructor update students web page.
5.2.9 Add Student

This method is used to add a user individually. The instructor will have to provide all the student information to successfully add the user. The method automatically assigns Student ID as User ID and a default password. This method helps when a student is registered after the classes started and needs access to the class material. Figure 5-21 is a screen shot of add student web page.
5.2.10 Delete Students by Term

This method helps the instructor remove old records from the database. The instructor can only delete the students and their information for the particular course selected initially. The method deletes the course grades table and course grade description tables of the selected previous semesters. This helps maintain the database and increases the performance. The students deleted from this method are not deleted from the system; they are only deleted from the course inst table and can no longer access the course material for this course. The interface will just have a dropdown list of all available terms and the instructor user will just select the term and hit delete.

5.3 Student Advising

The system is built such a way that the instructor or the admin can be an advisor. There is no need for another user type because, generally students interact mostly with their instructors and department chair every day and are always at the same campus. The department also assigns instructors as student advisors for many reasons. The administrator user and the instructor users have access to the student advising part of the
They can navigate to this system by selecting an advisee from the list provided as dropdown after successful login which re-directs to the student advising home page. Figure 5-22 is a screen shot of student advising home page.

![Student Advising Home Page]

**Figure 5-22 Student advising home page**

### 5.3.1 Academic Progress

This method can be referred to as program audit. The method will pull the advisees’ all completed courses from the transcript table and compare it to the BSIE program requirements. According to the BSIE curriculum the student is supposed to have done all the IE required courses which are listed in the IE requirements table in the database, supposed to do at least four technical electives from the available list, and complete general elective courses which satisfy several objectives of the department and the institute. The user can select which information to be displayed by selecting the checkbox on the side bar of the web page. Figure 5-23 shows the academic progress showing IE requirements and Technical Elective courses done by a student.
Figure 5-23 Academic progress showing IE requirements and Technical Elective courses done by a student

The list of GEC courses and the information related to the courses satisfying the objectives is stored in the GEC requirements table in the database. The method will compare each course to the GEC requirements table and creates a list for the advisor showing the objectives satisfied by the student. The GEC table displayed on the page works like a scratch book where the dark green “tick” marks represents the objective for which the GEC course is now counted for, where as the light green colored “tick” marks represents the GEC course also satisfies the respective objective. The advisor can click on the “tick” marks to change the objective to the course counts for. These changes are saved in the transcript table in the database.

The method will also compute the departmental and institutional GPA. The departmental GPA considers only the courses taken from the IE department which help the advisor to see how the student is performing in the IE program. The institutional GPA considers the courses the student has taken from the day he/she entered the institute. The
GPA calculation considers the D/F repeat rule where a student can repeat a course when the student have not attempted more than 60 hours of credit when the course is taken. The initial grade will be dropped from GPA computation and new grade will be considered even though the grades worse than the original grade. The method will consider the credit hours transferred from other institutions for IE program audit, but will not include the grade in institutional or departmental GPA calculations. In computation of IE GPA, if a course is repeated, only the better grade is considered. Figure 5-24 shows the academic progress web page with GEC objectives done by students and GPA computations.

Figure 5-24 Academic progress web page with GEC objectives done by students and GPA computations

5.3.2 Suggest Courses

This method helps provide the list of courses to the advisee by running an IE program audit. The method will pull the courses completed till the current semester and compares it to the IE required table, technical electives table and GEC requirements table
and lists the courses that are yet to be done. The method will also check if the student is eligible to take the course by checking if the advisee has done the pre requisites of each course and met all the special requirements like minimum grade required in the pre requisites, academic year standing, and number of credit hours completed. The interface also provides the option for an advisor to add comments after every time he/she suggests the courses to the advisee and can retrieve the comments later for reference. The advisor can edit or add comments in the textbox provided and will be saved to the database.

Figure 5-25 is a screen shot of advisor suggest courses web page.

Figure 5-25 Advisee suggest courses

The advisor can also click on the courses displayed to see the pre requisites, co requisites and the schedule of the course, all displayed in a little popup window. Figure 5-26 shows the screen shot of the popup window with course information.
5.3.3 Manage Transcript

This method allows the advisor to manage the advisee’s transcript in the system. This system is not in coherence with the admissions and records of the university all the transcripts from this system are unofficial. If the student takes any course from outside the department it is not saved to the transcript table in this database. The advisor will have access to the transcript table and he/she can add courses taken by the advisee to the transcript table. The interface displays all the courses taken by the advisee so far in the system, and the advisor can enter more course id’s, term the course was taken, credit hours of the course, and grade if available. Figure 5-27 shows the screen shot of advisor manage transcript web page.
5.3.4 Useful Links

This method is used to post the web links to certain information available on the university website to any files which will be useful to the students of IE program. There is information available for students who enter the IE or any other program at WVU but some students may not know where it is available. This method helps the advisors to post the links, add description, delete if the link is outdated. Figure 5-28 is a screen shot of useful links web page.
5.4 Teaching Assistant User

The teaching assistant user will have access to some of the methods designed for the instructor. This user can upload/update course content to the website. He/she can provide user access for students to the website. The teaching assistant can also upload course grades after every homework/project/tests but he/she will not have access to edit any of the grades. The other methods available to teaching assistant user are viewing notes, viewing grades, uploading and update notes, updating students, and adding new students. After successful login of the user the option to select a course will appear and as the user selects the course he/she is re-directed to the course instruction home page. These users are created and courses are assigned by the administrator user. Figure 5-29 shows the login page after teaching assistant successful login and Figure 5-30 is a screen shot of the teaching assistant home page.
Figure 5-29 Teaching assistant after successful login

Figure 5-30 Teaching assistant home page
5.5 Student User

The student users are created by administrator or the instructors. All the students who have user id, password and are not blocked by admin have access to the student advising system, where the student can check completed courses, run IE program audit, get suggestion on courses to take next semester and also have access to the useful links posted by the admin or the advisor. The student user will have access to the course instruction part of the system only if he/she is registered to the course and the instructor adds the user to the system and is not blocked for any reason. The students after successful login will see an option to select the course to navigate to course instruction part. Figure 5-31 shows the login page after student user’s successful login.

![Figure 5-31 Student login page after successful login](image)

From the login page after the student user selects a course from the list, he/she is re-directed to the student home page. Figure 5-32 is the screenshot of the student course home page.
The methods available for the student users are described as follows.

5.5.1 View Notes

This method re-directs to the view notes page. The view notes page is common for both instructor and the student. The user will get a list of the notes uploaded by the instructor with status released. The user selects the notes from the dropdown list which pulls the file from the server and displays in the body frame of the web page.

5.5.2 View Grades

This method displays the grades for the current semester for the course the user selected after logging in. The grades page displays only the user’s grades from the course grades table. The user has no access to the grades of other students in the class. The user also has an option to see the class statistics such as class average, minimum, maximum, and standard deviation for each grade category. The student user could also view the histogram for each grade category from the grades uploaded by the instructor. The user
would have to select a category from the dropdown list and select “View Bar Chart” which will generate the histogram and displayed on the web page. There is also an option to see the grading scheme which describes the weight of each grade category and out of value. Figure 5-33 shows the screen shot of students view grades page.

Figure 5-33 Student view grades page

Figure 5-34 shows the student grades with class statistics and Figure 5-35 shows the student grades with grading scheme.
Figure 5-34 Grades with class statistics

Figure 5-35 Grades with grading scheme
5.5.3 Advising

This method re-directs the student user to the student advising home page. The student will have access to the methods that an advisor has but with some limitations. Figure 5-36 is a screen shot of students advising home page.

![Student advising home page](image)

**Figure 5-36 Student advising home page**

5.5.4 Academic Progress

This method is the same as the academic advising method of advisor. This provides all the information that an advisor gets from his method. This helps student run IE program audit and avoids having an appointment with his/her advisor which would save time for both the users. The user can select which information to display by selecting the options available on the side bar of the web page. Figure 5-37 shows a sample academic progress web page of a student.
5.5.5 Suggest Courses

This method is also similar to the suggest courses method of an advisor but the student user will not have access to the advisor comments. The student user can only get a list of courses to be completed. The method also checks the eligibility to take the course by checking satisfactory conditions like course pre requisites, academic year standing, credit hours done etc. Figure 5-38 is a screen shot of student suggest courses web page.
5.5.6 Useful Links

This method will display the useful links and their description. The student users have no access to add new links, edit or delete the posted links. As the student user selects the link the web page related to the link is opened in a new browser window. Figure 5-39 shows the screen shot of the student’s useful links web page.

Figure 5-38 Student suggest courses web page
5.5.7 Change Password

This feature is available for all user types. The user can change his/her password after successful login. He/she has to provide the old password and new password to verify and successfully change the password.
CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Conclusion

The objective of this thesis was to develop a web-based system for course instruction and student advising, in Visual Studio 2008 and SQL Server as backend database. This tool helps in including broader goals which minimize the amount of time and energy spent by the advisor and helps minimize a lot of physical work for an advisor who has to get the data from an office secretary or look up or refer the University Manual. This tool helps the advisor keep a check on his advisee’s consistent academic record. This tool is not a replacement for an advisor but it definitely provides enormous assistance to an advisor. It especially helps new advisors who have just joined the university. The course instruction part of the system helps the instructor in class tasks such as posting online material, class grades etc. The data mining feature has been introduced in the system where the admin user can apply several data mining methods on the student’s pre admission data and few course grades to see which data mining algorithm has high accuracy in predicting the student final GPA. The key features of the system are:

- Online course material
- Post grades online
- Four user types namely administrator, instructor, teaching assistant, and student
- Uploading class roster, grades from excel file
- Customized course grades structure and grading scheme
- Keep track of advisee’s academic record
- Perform IE program audit on advisee’s transcript
- Check for graduation eligibility
- Computing departmental and institutional GPA considering D/F repeat policy and other rules
- Suggesting courses for next semester by verifying completed courses and checking pre requisite conditions to be eligible to take the course.
• Post useful links
• Manage users of the system online

6.2 Future Work

The system can be expanded to other departments and colleges. The student advising part of the system can be improved by addressing more advising tasks which would help advisors and students. The system could have an ability to store multiple curriculum requirements. It can provide as much advising information on the website to reduce meetings with advisors. The system could also have a user interface for the instructor to create grading schemes for a course as an alternative for uploading grades from an excel file. The students in the system could be organized under different categories such as honors, athletes, GPA < 2.0 and Reserve Office Training Corps (ROTC). Develop a feature for the advisor to assign a temporary advisor to his/her advisees. The data mining part of the system can be developed to a greater extent by providing the user interface for the user to build query from the available student data and apply various data mining methods, deploy and view results, and graphs online.
REFERENCES

3. J. D. Beatty, the National Academic Advising Association, Iowa State University.
6. Freeman, Lynn C, Establishing Effective Advising Practices to Influence Student Learning and Success Peer Review, winter 2008
17. Kash Barker, Theodore Trafalis and Teri Reed Rhoads “Learning from Student Data” School of Industrial Engineering, University of Oklahoma, 2002
23. IUCARE Advising System at Indiana University, website link, www.wcet.info/resources/publications/gude1003/guide/%7B17A69B48-C9B311D3-9309-005004AD2ACC%7D_1669.htm (accessed on September 19, 2009)
32. Raul Mihali, Tarek Sobh and Damir Vamoser, SKED: A Course Scheduling and Advising Software, University of Bridgeport.
33. Serpil Tin, Ismet Erkmen, A Software Tool for Academic Advisors for Student Performance Follow-up, Middle East Technical University.
34. Jeffrey L. McClellan, The Advisor as Servant: The Theoretical and Philosophical Relevance of Servant Leadership to Academic Advising, Utah Valley State College
36. findarticles.com/p/articles/mi_qa4115/is_200801/ai_n25502246, web reference accessed Date: September 3, 2008