Follow this and additional works at: https://scholar.dsu.edu/theses
MATHIMATICS & NATURAL SCIENCES WEBSITE

A graduate project submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Master of Science

in

Information Systems

March, 2010

By

Nader Green

Project Committee:

Dr. Sreekanth Malladi
Dr. Steve Graham
Dr. Surendra Sarnikar

KARL E. MUNDT LIBRARY
Dakota State University
Madison, SD 57042-1799
Student Name: Nader Green

Master's Project Title: Mathematics and Natural Sciences Website

Signature Approval:

Faculty supervisor: M. Stroehart Date: 04/01/2010

Committee member: Sandra Simon  Date: 04/01/2010

Committee member:  Date: 04/11/2010

Original to Office of Graduate Studies and Research
Acid-free copies with written reports to library

4/5/2010
ABSTRACT

In an environment like a university seekers of information are many. Most of which are students looking for material regarding their courses. This project uses Visual studio 2005 in addition to SQL Server 2005 to enable those information seekers to obtain it, easily and efficiently.

Using Mathematics & Natural Sciences Department website will enable students to acquire information regarding core courses. In addition, each faculty member is able to modify his/her profile using database. This can be done using SQL Server to add, delete, modify own information on respective website.
DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.
I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

____________________________________
Nader Green
# TABLE OF CONTENTS

MATHMATICS & NATURAL SCIENCES WEBSITE .......................................................... 1
PROJECT APPROVAL FORM .................................................................................. II
ABSTRACT ........................................................................................................... III
DECLARATION ...................................................................................................... IV
TABLE OF CONTENTS ....................................................................................... V
LIST OF FIGURES ............................................................................................... VII

CHAPTER 1 INTRODUCTION ................................................................................... 1
  1.1 Problem Definition ..................................................................................... 1
  1.2 Problem Solution ...................................................................................... 2
  1.3 Project Description .................................................................................... 2
  1.4 Environment ............................................................................................. 5
  1.5 Deliverables ............................................................................................... 5
  1.6 Software and Hardware Requirements ..................................................... 5
      1.6.1 Software Requirements .................................................................. 5
      1.6.2 Hardware Requirements ................................................................ 5
  1.7 Success Criteria ......................................................................................... 5

CHAPTER 2 REQUIREMENT ANALYSIS ................................................................. 6
  2.1 Understanding Concept ............................................................................ 6
      2.1.1 Understanding Internet .................................................................. 6
      2.1.2 Why we developed internet application ....................................... 7
  2.2 Understanding ASP.NET ........................................................................... 7
  2.3 Various Modules ....................................................................................... 8

CHAPTER 3 DESIGN ............................................................................................... 11
  3.1 UML as Design Tool ................................................................................ 11
  3.2 Types of UML Diagrams used in this Project ........................................ 12
  3.3 Various UML Diagrams .......................................................................... 15
      3.3.1 Use case diagrams ...................................................................... 15
      3.3.2 Class Diagram ............................................................................ 18
      3.3.3 Activity Diagrams ...................................................................... 19

CHAPTER 4 IMPLEMENTATION ........................................................................... 22
  4.1 Creation of Tables .................................................................................... 22
      4.1.1 What is Database? ....................................................................... 22
      4.1.3 Tables in the Project .................................................................. 23
  4.2 Implementation of Coding ...................................................................... 24

CHAPTER 5 SCREENS ........................................................................................... 27
  Home Page ....................................................................................................... 27
  Department Page ............................................................................................ 28
  Mathematics Page .......................................................................................... 29
  Physics Page .................................................................................................. 30
  Chemistry Page .............................................................................................. 31
  Courses Main Page ......................................................................................... 32
  Course Catalogue Page .................................................................................. 33
  Course Description Page ................................................................................ 34
  Login Page ...................................................................................................... 35
  Logged-in (Success) ....................................................................................... 36
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Components of Use Case Diagram</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>Elements of Class Diagram</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>Sample Action</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Activity Diagram</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>Case Diagram for Administrator</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>Case Diagram for Faculty/Staff Members</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>Case Diagram for Students/Parents</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>Class Diagram for team members</td>
<td>18</td>
</tr>
<tr>
<td>9</td>
<td>Activity Diagram for Login Module</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>Activity Diagram for Change Password module</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>Activity Diagram for Feedback module</td>
<td>21</td>
</tr>
<tr>
<td>12</td>
<td>Home Page</td>
<td>27</td>
</tr>
<tr>
<td>13</td>
<td>Department Page</td>
<td>28</td>
</tr>
<tr>
<td>14</td>
<td>Mathematics Page</td>
<td>29</td>
</tr>
<tr>
<td>15</td>
<td>Physics Page</td>
<td>30</td>
</tr>
<tr>
<td>16</td>
<td>Chemistry Page</td>
<td>31</td>
</tr>
<tr>
<td>17</td>
<td>Courses Main Page</td>
<td>32</td>
</tr>
<tr>
<td>18</td>
<td>Course Catalogue Page</td>
<td>33</td>
</tr>
<tr>
<td>19</td>
<td>Login Page</td>
<td>35</td>
</tr>
<tr>
<td>20</td>
<td>Login Success Page</td>
<td>36</td>
</tr>
<tr>
<td>21</td>
<td>Incorrect Log-in (Log-in Failed)</td>
<td>37</td>
</tr>
<tr>
<td>22</td>
<td>User Change Own Password</td>
<td>38</td>
</tr>
<tr>
<td>23</td>
<td>Choose File Upload Page</td>
<td>39</td>
</tr>
<tr>
<td>24</td>
<td>Upload File (Details) Page</td>
<td>40</td>
</tr>
<tr>
<td>25</td>
<td>User Create Account Page</td>
<td>41</td>
</tr>
<tr>
<td>26</td>
<td>Main Faculties’ Web Pages</td>
<td>42</td>
</tr>
<tr>
<td>27</td>
<td>Sample Instructor Welcome Page</td>
<td>43</td>
</tr>
<tr>
<td>28</td>
<td>Sample Instructor Office Hours Page</td>
<td>44</td>
</tr>
<tr>
<td>29</td>
<td>Sample Instructor CV Page</td>
<td>45</td>
</tr>
<tr>
<td>30</td>
<td>Sample Instructor Internet Links Page</td>
<td>46</td>
</tr>
<tr>
<td>31</td>
<td>Mathematica Page</td>
<td>47</td>
</tr>
<tr>
<td>32</td>
<td>Links and Contacts Page</td>
<td>48</td>
</tr>
<tr>
<td>33</td>
<td>Feedback (User Evaluation 1/3)</td>
<td>49</td>
</tr>
<tr>
<td>34</td>
<td>Feedback Comments (User Evaluation 2/3)</td>
<td>50</td>
</tr>
<tr>
<td>35</td>
<td>Feedback Comments (User Evaluation 3/3)</td>
<td>51</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Website is the gate to the outside world. The development of mathematics and natural sciences department website is aimed to enable interested parties to easily gain an access to the desired information. This website will enable exchanging information from/to faculties, students and information seekers. Hence, this website will effectively enhance the academic performance for faculties and students and increase their academic cooperation level.

Language: Visual Basic
Platform: .NET 2.0
Platform Technology: ASP.NET
Audience: Students, faculties, Parents, and teachers

1.1 Problem Definition

In the Present Days, handling information can be carried in a fast, safe, and reliable ways using computers and internet. Websites can help to carry information into the hands of its seekers. Student upon registration of their courses they seek courses which is required by their major in order to fulfill their plan for graduation through the help of the academic advisor. This creates a burden on the advisor head. In addition, time is being wasted by student to wait in lines until meeting their advisor.

In this type of situation, it is convenient for the user when he/she has a website that will provide information about their required courses in their major. As well as providing information about faculty.

The Mathematics & Natural Sciences Department doesn’t have a website to advertise and post information. For example, course information, resources, staff and faculty information. Hosting
such information is a vital and creates a competitive advantage with other Universities in the area.

Also, teachers do not possess an individual website and need a website for each. This will facilitate communication with their respective students. Instructor will be able to write a short welcome, Office hours, CV, internet link, and other related information.

There is a need to create one for the following reasons:
- Student will be able to identify courses needed for their major.
- Description of each course offered.
- Faculty member profile.

1.2 Problem Solution

Using our website design tools, we can solve the above problem by carrying out the work in the form of multiple of websites to show the desired information in an efficient manner.

Information related to courses, faculties, departments can be retrieved easily from the website. Teacher can upload files (assignment, presentation, practice quiz) to their website for student to download in their convenient time.

1.3 Project Description

Mathematics & Natural Sciences Department (MNSD) website is a tool used to manage and organize information needed in such a way that it is delivered to students when they require it within 24/7 as long as internet and computer is available. It also helps faculty to deliver their files to respected student at their convenient time and location.

The people who can use this tool are administrator, faculties, staff member, students, and parents.
Administrator
The Administrator is a person having administrative or managerial authority in MNSD website. His duty is to create, delete, and modify users to the database of the website.

Faculty
Faculty takes control of his/her profile posting his/her office hours, CV, internet link, email, and uploads their file,

Student
Student is able to access the website any time to display course information, download files, Teacher information, and department information.

Parents
Parents are able to get contact information of their respected son’s teacher. Also, gather information regarded their son’s major and courses in the track.

Modules
The basic modules involved in our project are listed below:
1- Department
2- Courses
3- Login
4- Create Account
5- Staff and Faculties
6- Mathematica
7- Link and contact
8- Feedback
Department
This module present descriptive information of three departments (Math, Physics, and Chemistry) and the courses provided.

Courses
In this module it provides course catalog which having list of courses plus the course description and description of each course.

Login
The user can get access to this tool with the help of this module

Create Account
Using this module faculty can create an account.

Staff and Faculties
Provide information about staff and faculty. For example, CV, Office hours, image, interesting internet links.

Mathematica
This link will lead to the website of Mathematica. The Mathematica website will list the objectives behind its use and benefits. Mathematica aids the students not only for computation, but for modeling, simulation, visualization, development, documentation, and deployment.

Link and contact
To facilitate the communication between students and staff member or faculty.

Feedback
This feedback provides analysis and evaluation of the website. The user will have the ability to write plain text into the textbox. As well as the user select a value between 1 and 5 to rate the site.
1.4 Environment

The environment to run this project is systems with windows XP/windows 2000 operating system, which are connected to internet.

1.5 Deliverables

This tool is basically an intranet application.

1.6 Software and Hardware Requirements

1.6.1 Software Requirements
   Programming tool: Visual Studio 2005
   Database: Microsoft SQL Server 2005
   Web Server: IIS (Internet Information Services)
   Operating System: Windows XP

1.6.2 Hardware Requirements
   Processor: Pentium 3, Pentium 4, Celeron
   RAM: 512MB or higher

1.7 Success Criteria

This project can be successfully used by organizations for effective task delegation and monitoring of projects.
CHAPTER 2

REQUIREMENT ANALYSIS

2.1 Understanding Concept

2.1.1 Understanding Internet

The Internet is a worldwide system of interconnected computer networks which uses Transfer Control Protocol / Internet Protocol (TCP/IP) to serve huge number of users worldwide.

It is a network of networks that consists of millions of private and none private networks of local to worldwide scope that link networks by different types of wires (copper wires, fiber-optic cables, wireless connections) and other technologies.

The Internet carries enormous amount of information (resources and services), especially the inter-linked hypertext documents of the World Wide Web (WWW). This is in addition to the support of electronic mail, online chat, file transfer and file sharing, games, commerce, social networking, publishing, video on demand, and teleconferencing, telecommunications, and Voice over Internet Protocol (VoIP) applications allow person-to-person communication via voice and video. (Anonymous, 2009)

Why we use internet

1. It is available 24 hours and 7 days a week.
2. Allow electronic mail and online chat.
3. Provide file transfer and file sharing.
4. Offer gaming, commerce, social networking, and publishing.
5. Supply video on demand, teleconferencing, and telecommunications.
6. Make Voice over Internet Protocol (VoIP) applications available.
2.1.2 Why we developed internet application

The Internet is a global data communications system. Using ASP.net application allows real-time controls, reports and queries, and charting. These services can be accessible all the time from everywhere.

2.2 Understanding ASP.NET

ASP.NET is a programming framework used to create enterprise-class Web Applications. These applications are accessible on a global basis leading to efficient information management. (Startvbdotnet, 2009)

The advantages of using ASP.NET are listed below:

1. Reduces code required to build large applications
2. development is simpler and easier to maintain with server-side programming model
3. Easier to write and maintain ASP.Net pages because the source code and HTML are together.
4. There is a lot of page flexibility and power because source code is executed on the server.
5. The source code is executed on the server. The pages have lots of power and flexibility by this approach
6. the user information entries are being validated without writing a single line of code

Internet Information Services

Internet Information Server (IIS) is a group of Internet-based services to use with Microsoft Windows. Internet Information Server is considered the second server after Apache HTTP Server which comes first. The services provide FTP, FTPS, SMTP, NNTP, and HTTP/HTTPS. In the testing environment ASP.NET application will use developer computer as a web server. While implementing ASP.NET application on World Wide Web (WWW), we will run it on a host web server. When testing those pages, actually we are accessing the pages through IIS (internet information server) and retrieve the final HTML through an HTTP transfer.
2.3 Various Modules

We are going to describe the modules involved in the project in some details. The project consist of the following modules

1. Welcome Page
2. Departments
3. Courses
4. Log in
5. Create account
6. Staff and Faculties
7. Mathematics
8. Links and contact
9. Feedback
10. News

Welcome Page

The Welcome screen is the default ASP.Net page. It is considered the first page for user and it provides links to all other modules within the project. It includes a word of welcome from department chairman.

Departments

This web page describes Mathematics and Natural Sciences and will guide user to the content of each study area major courses. There are sub-sections under this department.

- Preparatory program
- Mathematics
- Physics
- Chemistry

Courses

The course web page brief users of major courses required under Mathematics and Natural Sciences. The sub-sections are course catalogue and course description. Course catalogue list all courses provided by the department. Course description gives a short explanation of the course objectives and pre-requisite.
Log in

Log in web page is used by faculty and staff members to access their own respective web page.

Create account

Users, staff, faculties, students can create an account using this page to access, maintain resources. When the user tries to create an account a few information is required to be filled in. User name, password, confirm password, e-mail, and security question security answer are required to create the account. This information is being stores in the database for authentication.

Staff and Faculties

All members, staff and faculties, are being listed in this web page. Links are provided for each faculty members. After clicking any link of a faculty it direct you to the selected own web page which contains his/her office hours, CV, and internet links.

Mathematica

Accessing information regarding Mathematica can be pulled out from this page. Mathematica is a computational software program used in scientific, engineering, and mathematical fields and other areas of technical computing.

Links and contact

Important Links and contacts are being posted in this page. Enrolment, events, facility support, finance, financial aid, human resources, public relation, marketing and advertising information technology, info, library, lifelong learning, reactors office, and web mater are all examples of the links provided by this page.
Feedback

Users can write and send their suggestions and notices using this web page. There are 4 steps to complete the feedback. The steps are:

1. Contact info.
2. Comments
3. Summary
4. Complete

News

News is being displayed on Welcome page for users of the website.
CHAPTER 3

DESIGN

3.1 Unified Modeling Language (UML)

UML is a standardized general-purpose modeling language in the field of software engineering. The standard is managed, and was created by, the Object Management Group.

UML includes a set of graphical notation techniques to create visual models of software-intensive systems.

The Unified Modeling Language is fast becoming a required skill for virtually anybody involved in a software project. Requirements analysts, software developers, architects, UI designers, database professionals, testers and project managers are increasingly being asked to create and consume specifications written in UML.

Components

The unified modeling Language (UML) is a standard language. It’s mainly used for specification, visualization, construction, and documentation the structure of the software systems. The UML consists of a collection of practices, set by engineers. It shows a significant success in modelling large and complex systems. The UML is an important element for developing objects oriented software and software development process. The UML uses graphical diagrams to express software design. The use of UML helps to improve communication among project team members, validate the designed architecture of the software and explore potential designs.
3.2 Types of UML diagrams used in this project

Each UML diagram is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. The following are types of UML diagrams:

Use Case Diagram
A use case diagram in the Unified Modeling Language (UML) is a descriptive diagram. Its purpose is to display a graphical overview of the system functionality in terms of actors, their goals (use cases), and any dependencies between those use cases.

![Diagram](image)

Figure 1. Components of Use Case Diagram
Figure 2. Sample Action

An actor represents a user or another system that will interact with the system you are modelling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

Class Diagram:

Class diagram models class structure and contents shows us the relationship properties such as containment, inheritance, associations and others. Classes are composed of three things: a name, attributes, and operations. Below is an example of a class.

Figure 3. Elements of Class Diagram
Activity Diagram

Activity diagram is a special state diagram where it shows the process flow control. It displays workflows of stepwise activities and actions. As well as showing the conditional or parallel activities.

![Activity Diagram](image)

Figure 4. Activity Diagram
3.3 Various UML Diagrams

3.3.1 Use case diagrams

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>administrator</td>
</tr>
<tr>
<td>2</td>
<td>faculties</td>
</tr>
<tr>
<td>3</td>
<td>staff member</td>
</tr>
<tr>
<td>4</td>
<td>students</td>
</tr>
<tr>
<td>5</td>
<td>parents</td>
</tr>
</tbody>
</table>

Use Case Diagram for administrator:

Figure 5. Case Diagram for Administrator
Use Case Diagram for faculty/staff members:

Figure 6. Case Diagram for Faculty/Staff Members
Use Case Diagram for student/parent:

Figure 7. Case Diagram for Students/Parents
3.3.2 Class Diagram

Class Diagram for team member

Administrator
- Username
- Password
- Add
- Delete
- View
- Modify

Faculty
- Username
- Password
- View
- Modify

Staff
- Username
- Password
- Add
- View

Student
- Username
- Password
- View

Parents
- Username
- Password
- View

Figure 8. Class Diagram for team members
3.3.3 Activity Diagrams

Activity Diagram for Login module

Accept Login Details

Validate Username and Password

Reject

Accept

Show Respective Details

Figure 9. Activity Diagram for Login Module
Activity Diagram for Change Password module

Figure 10. Activity Diagram for Change Password module
Activity Diagram for Feedback module

Supply username & E-mail

Confirm correct format of e-mail address

Accept

Enter Comments

Confirm comments entered 300 words and confirm rating between 1 and 5

Reject

Accept

Show message that feedback is completed

Figure 11. Activity Diagram for Feedback module
CHAPTER 4

Implementation

4.1 Creation of tables

4.1.1 What is Database?

A database is a collection of data. The term database include data and according to Beccerra-Fernandez “Data comprises facts, observations, or perceptions“ and that “Data represents raw numbers or assertions” (Becerra-Fernandez, 2004). Data can take many shapes and formats. Sample of data, but not limited to, the following: words, text, images, pictures, symbols, signs and numbers. The computer maintains the data for later access.

Database Table

A database table is a set of data elements (values) that is organized using a model of vertical columns and horizontal rows. A table has a specified number of columns, but can have any number of rows. Each row is identified by the values appearing in a particular column subset which has been identified as a Primary key.

There are two types of keys, Primary key and Foreign key. The primary key is used to identify rows in the table but foreign key is used to create the relation between two tables.
4.1.3 Tables in the Project

The following is a sample database tables.

**Faculty table:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Char</td>
<td>10</td>
</tr>
<tr>
<td>Title</td>
<td>nchar</td>
<td>5</td>
</tr>
<tr>
<td>First</td>
<td>nchar</td>
<td>15</td>
</tr>
<tr>
<td>Last</td>
<td>nchar</td>
<td>15</td>
</tr>
<tr>
<td>Gender</td>
<td>nchar</td>
<td>10</td>
</tr>
<tr>
<td>CV</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>OfficeNo</td>
<td>nchar</td>
<td>10</td>
</tr>
<tr>
<td>OfficeHours</td>
<td>nchar</td>
<td>10</td>
</tr>
<tr>
<td>Image</td>
<td>image</td>
<td></td>
</tr>
</tbody>
</table>

Primary Key: id

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>char(10)</td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>nchar(5)</td>
<td></td>
</tr>
<tr>
<td>First</td>
<td>nchar(15)</td>
<td></td>
</tr>
<tr>
<td>Last</td>
<td>nchar(15)</td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td>nchar(10)</td>
<td></td>
</tr>
<tr>
<td>CV</td>
<td>text</td>
<td></td>
</tr>
<tr>
<td>OfficeNo</td>
<td>nchar(10)</td>
<td></td>
</tr>
<tr>
<td>OfficeHours</td>
<td>nchar(10)</td>
<td></td>
</tr>
<tr>
<td>Image</td>
<td>image</td>
<td></td>
</tr>
</tbody>
</table>

Whenever a new faculty is added then a unique id is given and is stored in the above table.
4.2 Implementation of Coding

ASP.NET website (Microsoft, 2009) in Visual Basic language was used to code the project.

Sample code

'This is a feedback code

Partial Class feedback_feedback
    Inherits System.Web.UI.Page

    Protected Sub CustomValidator1_ServerValidate( ByVal source As Object, ByVal args As System.Web.UI.WebControls.ServerValidateEventArgs) Handles CustomValidator1.ServerValidate
        If TextBox3.Text.Length > 300 Then
            args.IsValid = False
        Else
            args.IsValid = True
        End If
    End Sub

    Protected Sub Wizard1_FinishButtonClick(ByVal sender As Object, ByVal e As System.Web.UI.WebControlsWizardNavigationEventArgs) Handles Wizard1.FinishButtonClick
        SendMail(TextBox2.Text, TextBox3.Text)
    End Sub

    Private Sub SendMail(ByVal from As String, ByVal body As String)
        Dim mailServerName As String = "SMTP.MyDomain.com"
        Dim message As MailMessage = New MailMessage(from, "ngreen@pmu.edu.sa", "feedback", body)
        Dim mailClient As SmtpClient = New SmtpClient

        mailClient.Host = mailServerName
        mailClient.Send(message)
        message.Dispose()
    End Sub
End Class

'Upload file code

Partial Class _Default
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        Dim UpPath As String
        Dim UpName As String
        UpPath = "C:\UploadedUserFiles"
        UpName = Dir(UpPath, vbDirectory)
        If UpName = "" Then
            MkDir("C:\UploadedUserFiles")
        End If
    End Sub
End Sub

Protected Sub UploadButton_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles UploadButton.Click
    FileName.InnerText = FileField.PostedFile.FileName
    FileContent.InnerText = FileField.PostedFile.ContentType
    FileSize.InnerText = FileField.PostedFile.ContentLength
    UploadDetails.Visible = True

    Dim strFileName As String
    strFileName = FileField.PostedFile.FileName
    Dim c As String = System.IO.Path.GetFileName(strFileName) ' We don't need the path, just the name.

    Try
        FileField.PostedFile.SaveAs("C:\UploadedUserFiles\" + c)
        Span1.InnerText = "File Uploaded Successfully."
    Catch Exp As Exception
        Span1.InnerText = "Some Error occurred."
        UploadDetails.Visible = False
    End Try

End Sub
End Class

Partial Class MasterPageSecure_uploadafilebest
    Inherits System.Web.UI.Page

    Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
        filename.InnerText = FileField.PostedFile.FileName
        filecontent.InnerText = FileField.PostedFile.ContentType
        filesize.InnerText = FileField.PostedFile.ContentLength
        UploadDetails.Visible = True

        Dim myfilename As String
        myfilename = FileField.PostedFile.FileName
        Dim c As String = System.IO.Path.GetFileName(myfilename)

        Try
            FileField.PostedFile.SaveAs("c:\nader\uploadusersfiles\" + c)
            span1.InnerText = "File Uploaded Successfully"
        Catch ex As Exception
            span1.InnerText = "File Uploaded Failed"
            UploadDetails.Visible = False
        End Try

        'Dim myfilename As String = filefield.PostedFile.ContentLength

    End Sub
Protected Sub Button1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Load
    Dim upPath As String
    Dim upName As String
    upPath = "C:\nader\uploadusersfiles"
    upName = Dir(upPath, vbDirectory)
    If (upName = ") Then
        MkDir("C:\nader\uploadusersfiles")
    End If
End Sub
End Class

<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0">

    <siteMapNode url="Default.aspx" title="Home" description="" />

    <siteMapNode url="Department.aspx" title="Department" description="" />
        <siteMapNode url="Preparatory.aspx" title="Preparatory Program" description="" />
        <siteMapNode url="Mathematics.aspx" title="Mathematics" description="" />
        <siteMapNode url="Physics.aspx" title="Physics" description="" />
        <siteMapNode url="Chemistry.aspx" title="Chemistry" description="" />
    </siteMapNode>

    <siteMapNode url="Courses.aspx" title="Courses" description="" />
        <siteMapNode url="CourseCatalog.aspx" title="Course Catalog" description="" />
        <siteMapNode url="CourseDescription.aspx" title="Course Description" description="" />
    </siteMapNode>

    <siteMapNode url="Login.aspx" title="Login" description="" />
    <siteMapNode url="CreateAccount.aspx" title="Create Account" description="" />
    <siteMapNode url="MasterTeachers/facultiesList.aspx" title="Staff and Faculties" description="" />
    <siteMapNode url="Mathematica/Mathematica.aspx" title="Mathematica" description="" />
    <siteMapNode url="Links_Contacts/Links_Contacts.aspx" title="Links and Contacts" description="" />
    <siteMapNode url="feedback/feedback.aspx" title="Feedback" description="" />
</siteMap>
CHAPTER 5

SCREENS

Home Page

Welcome from Dr. Rahim Karimpour, Chairman of the Mathematics and Natural Sciences. It is pleasure to welcome you to the Mathematics and Natural Sciences Department...

(Under Constructions)

Figure 12. Home Page
Department Page

Thursday, March 25

WELCOME

DEPARTMENT

1. **Mathematics (Preparatory)**

Mathematics provides an approach to problem solving through logic and reasoning. It is used to identify, analyze, generalize, and communicate quantitative relationships.

Learning outcomes for students completing the Mathematics requirement are:

- Knowing the fundamental notation and rules of a mathematical system.
- Recognizing problems to which mathematics can be applied.
- Translating problems into mathematical form.
- Being able to construct and interpret visual representations of mathematical relationships.
- Constructing logical and valid mathematical arguments.
- Determining mathematical relationships and solutions to problems.
- Communicating mathematical relationships and solutions.

2. **Natural and Physical Sciences (Preparatory)**

The goal of the natural and physical sciences is to better understand nature. The natural and physical sciences systematically study natural phenomena. They do so by observing nature, by collecting and analyzing data, by formulating, testing, and revising hypotheses, and by developing theories.

Learning outcomes for students completing the Natural and Physical Sciences requirements are:

- Understanding what the realm of science is, and why science is important to their lives.
- Comprehending current principles and theories used to explain natural phenomena and understanding the role of theory in science.
- Seeing science as a process of conducting systematic observation, formulating and testing hypotheses, collecting and evaluating data, recognizing sources of error and uncertainty in experimental methods, and disseminating results.
- Developing an understanding of how human activity affects the natural environment.
- Making informed judgments about science-related topics and policies.

Click on a link below:

- Mathematics (core)

---

Figure 13. Department Page
Mathematics Page

Mathematics

Mathematics provides an approach to problem solving through logic and reasoning. It is used to identify, analyze, generalize, and communicate quantitative relationships. Learning outcomes for students completing the Mathematics requirement are:

- Knowing the fundamental notation and rules of a mathematical system.
- Translating problems to which mathematics can be applied.
- Constructing and using visual representations of mathematical relationships.
- Constructing logical and valid mathematical arguments.
- Communicating mathematical relationships and solutions.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credit Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 1311</td>
<td>Finite Mathematics for Students of Business</td>
<td>3</td>
</tr>
<tr>
<td>MATH 1313</td>
<td>Calculus for Students of Business</td>
<td>3</td>
</tr>
<tr>
<td>STAT 1113</td>
<td>Statistical Methods</td>
<td>3</td>
</tr>
<tr>
<td>MATH 1422</td>
<td>Calculus I</td>
<td>4</td>
</tr>
<tr>
<td>MATH 1433</td>
<td>Calculus II</td>
<td>4</td>
</tr>
</tbody>
</table>

Figure 14. Mathematics Page
Physics Page

Physics Page

The goal of the Physics Component of the PMU Program is to enhance students' understanding of physics.

<table>
<thead>
<tr>
<th>Key</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHYS 111</td>
<td>Introduction to Physical Science</td>
</tr>
<tr>
<td>PHYS 1411</td>
<td>Physics I</td>
</tr>
<tr>
<td>PHYS 1421</td>
<td>Physics for Engineers II</td>
</tr>
</tbody>
</table>

8 semester hours (two 4-hour courses) must be taken. Colleges designate specific required courses, if any.

Figure 15. Physics Page
The goal of the chemistry component of the PSSU Program is to enhance students' understanding of chemistry.

Key

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOL 1411</td>
<td>Introductory Biology</td>
</tr>
<tr>
<td>CHEN 1412</td>
<td>Introductory Chemistry</td>
</tr>
<tr>
<td>CHEN 1422</td>
<td>Chemistry for Engineers II</td>
</tr>
</tbody>
</table>

5 semester hours (two 4-hour courses) must be taken. Colleges designate specific required courses, if any.

Figure 16. Chemistry Page
Courses Main Page

Figure 17. Courses Main Page
### Course Catalogue Page

**Welcome**

<table>
<thead>
<tr>
<th>Home</th>
<th>University Program</th>
<th>Mathematics</th>
<th>Key</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Preparatory Program</td>
<td>MATH 1311</td>
<td>Finite Mathematics for Students of Business</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MATH 1312</td>
<td>Calculus for Students of Business</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Courses</td>
<td>MATH 1313</td>
<td>STATISTICAL METHODS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MATH 1311</td>
<td>Pre-Calculus Mathematics</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MATH 1221</td>
<td>Calculus I</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MATH 1221</td>
<td>Calculus II</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MATH 1331</td>
<td>Calculus III</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MATH 2201</td>
<td>Linear Algebra</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MATH 2202</td>
<td>Differential Equations</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MATH 2203</td>
<td>Introduction to Discrete Mathematics</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHEM 1411</td>
<td>Introduction to Chemistry</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHEM 1412</td>
<td>Chemistry for Engineers I</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHEM 1413</td>
<td>Chemistry for Engineers II</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHEM 1421</td>
<td>Chemistry for Engineers III</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHEM 1422</td>
<td>Chemistry for Engineers IV</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PHYS 1421</td>
<td>Physics for Engineers I</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PHYS 1422</td>
<td>Physics for Engineers II</td>
<td></td>
</tr>
</tbody>
</table>

**News**

- Updated News Here!
Course Description Page
Login Page

Figure 19. Login Page
Logged-in (Success)

Figure 20. Login Success Page
Incorrect Log-in (Log-in Failed)

Figure 21. Incorrect Log-in (Log-in Failed)
User Change Own Password

Figure 22. User Change Own Password
Choose file Upload Page

Figure 23. Choose File Upload Page
Figure 24. Upload File (Details) Page
User Create Account Page

Figure 25. User Create Account Page
Main Faculties’ Web Pages

Figure 26. Main Faculties’ Web Pages
Welcome to the website of Nader Green

Welcome

My name is Nader. I am forty-five years old. I am a teacher in Prince Mohammad University, KSA, and I love my job so much. But, in fact, teaching is a hard job. I live in a big two-floor villa with my beautiful family, near the beach. I have two boys and one daughter. My wife is also a mathematics teacher and in the inner university.

I like reading especially English novels and also love traveling. Because through our trips we will be able to become more about the other culture. In addition, I can create new friendships with many different people from all over the world.

Return to:
- Location List
- Math & Natural Sciences
- Biotechnology

Figure 27. Sample Instructor Welcome Page
Sample Instructor Office Hours Page

Welcome

- Office Hours
- CV
- Instructor Links

Office Hours:
- Monday: 12:00 pm - 2:00 pm
- Friday: 12:00 pm - 2:00 pm
- Wednesday: 1:00 pm - 4:00 pm
- Office: B 172

Figure 28. Sample Instructor Office Hours Page
Sample Instructor CV Page

Figure 29. Sample Instructor CV Page
Sample Instructor Internet Links Page

Welcome to the web site of Nader Green

- Welcome
- Office Hours
- CV
- Internet Links

Interesting Links:
http://www.statcsयस्ती.com/
http://www.math.com/

Return to:
Course Links
Math & General Interest
Home Page

Figure 30. Sample Instructor Internet Links Page
Mathematica Page

Figure 31. Mathematica Page
Figure 32. Links and Contacts Page
Feedback (User Evaluation 1/3)

Figure 33. Feedback (User Evaluation 1/3)
Feedback Comments (User Evaluation 2/3)

Figure 34. Feedback Comments (User Evaluation 2/3)
Figure 35. Feedback Comments (User Evaluation 3/3)
CHAPTER 6

TEST CASES

6.1 Test Case for Administrator:
The Administrator login is validated and logged in to the website. When the Administrator creates an account for an Employee, he will provide name, password, and security question and answer. After, that will be notified by the account.

6.2 Test Case for Faculty and Staff:
The faculty/staff login page is created and validated. Then, the faculty/staff can login and view the stored information in the Database through his interface and he can as well add/Modify information to the Database. The Database is created using SQL Server and it is tested.

6.3 Test Case for Student/Parent:
The Student/Parent view Faculty pages created and validated. Then, the Student/Parent can view his/her faculty webpage information in the Database through his interface. The Database is created using SQL Server and it is tested.
CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion:

This website can be used by any educational institute to allow students and parents to access and monitor Instructor’s and course information. Faculty will be able to modify information and upload files related to their courses for each academic year taught.

Future Enhancements:

1. Student can submit his/her own homework and assignments.
2. Evaluation of course and instructor.
3. Search engine for the site or world wide
4. Listing sections offered per semester for registration purpose.
Appendices

APPENDIX-1

Dot Net Framework

The .NET Framework is an integral Windows component that supports building and running the next generation of applications and XML Web services. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.

- To provide a code-execution environment that minimizes software deployment and versioning conflicts.

- To provide a code-execution environment that promotes safe execution of code, including code created by an unknown or semi-trusted third party.

- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.

- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.

- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that promote security and robustness. In fact, the concept of code management is a fundamental principle of the runtime.
Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime enables you to embed managed components or Windows Forms controls in HTML documents. Hosting the runtime in this way makes managed mobile code (similar to Microsoft® ActiveX® controls) possible, but with significant improvements that only managed code can offer, such as semi-trusted execution and isolated file storage.

The following illustration shows the relationship of the common language runtime and the class library to your applications and to the overall system. The illustration also shows how managed code operates within a larger architecture.

.NET Framework in context
The following sections describe the main components and features of the .NET Framework in greater detail.

Features of the Common Language Runtime

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally feature rich.

The runtime also enforces code robustness by implementing a strict type-and-code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

In addition, the managed environment of the runtime eliminates many common software issues. For example, the runtime automatically handles object layout and manages references to objects, releasing them when they are no longer being used. This automatic memory management resolves the two most common application errors, memory leaks and invalid memory references.

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers. Any compiler vendor who chooses to target the runtime can do so. Language compilers that target the .NET
Framework make the features of the .NET Framework available to existing code written in that language, greatly easing the migration process for existing applications.

While the runtime is designed for the software of the future, it also supports software of today and yesterday. Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs.

The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance.

Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft® SQL Server™ and Internet Information Services (IIS). This infrastructure enables you to use managed code to write your business logic, while still enjoying the superior performance of the industry's best enterprise servers that support runtime hosting.

.NET Framework Class Library

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string
management, data collection, database connectivity, and file access. In addition to these common
tasks, the class library includes types that support a variety of specialized development scenarios.
For example, you can use the .NET Framework to develop the following types of applications
and services:

- Console applications.
- Windows GUI applications (Windows Forms).
- Windows Presentation Foundation (WPF) applications.
- ASP.NET applications.
- Web services.
- Windows services.

For example, the Windows Forms classes are a comprehensive set of reusable types that vastly
simplify Windows GUI development. If you write an ASP.NET Web Form application, you can
use the Web Forms classes. (Anonymous, 2010)

ASP.NET is a platform for developing a web based application. It lets you create sophisticated
web application that can interact with users. For example, ASP.NET can use data entry controls
(such as textboxes and buttons to accept input data from user process, retrieve or update database
data and send results of these operations back to the user. (Lowe, Cogswell, & Cox, 2006)
Appendix 2

Microsoft SQL Server

SQL Server 2005 (codenamed Yukon), released in October 2005, is the successor to SQL Server 2000. It included native support for managing XML data, in addition to relational data. For this purpose, it defined an xml data type that could be used either as a data type in database columns or as literals in queries. XML columns can be associated with XSD schemas; XML data being stored is verified against the schema. XML is converted to an internal binary data type before being stored in the database. Specialized indexing methods were made available for XML data. XML data is queried using XQuery; SQL Server 2005 added some extensions to the T-SQL language to allow embedding XQuery queries in T-SQL. In addition, it also defines a new extension to XQuery, called XML DML, that allows query-based modifications to XML data. SQL Server 2005 also allows a database server to be exposed over web services using TDS packets encapsulated within SOAP (protocol) requests. When the data is accessed over web services, results are returned as XML.

For relational data, T-SQL has been augmented with error handling features (try/catch) and support for recursive queries (Common Table Expressions). SQL Server 2005 has also been enhanced with new indexing algorithms and better error recovery systems. Data pages are checksummed for better error resiliency, and optimistic concurrency support has been added for better performance. Permissions and access control have been made more granular and the query processor handles concurrent execution of queries in a more efficient way. Partitions on tables and indexes are supported natively, so scaling out a database onto a cluster is easier. SQL CLR was introduced with SQL Server 2005 to let it integrate with the .NET Framework.

SQL Server 2005 introduced "MARS" (Multiple Active Results Sets), a method of allowing usage of database connections for multiple purposes.
Reference


