GRADUATE SCHOOL APPLICATION ARCHIVE SYSTEM

(GSAAS)

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

By

Nazeer Ahmed Khan Fazal

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

Major Department:
Computer Science

May 2011

Fargo, North Dakota

North Dakota State University
Graduate School

Title

GRADUATE SCHOOL ARCHIVE

APPLICATION SYSTEM

By

NAZEER FAZAL

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

MASTER OF SCIENCE

North Dakota State University Libraries Addendum

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

ABSTRACT

Fazal, Nazeer Ahmed Khan, M.S., Department of Computer Science, College of Science and Mathematics, North Dakota State University, May 2011. Graduate School Application Archive System (GSAAS). Major Professor: Dr. Kendall Nygard.

The advent of using web applications has dramatically changed the way people access data and understand these concepts. Users have expressed problems involved with accessing the data in an easy and effective way with control for users. Therefore, many reviews agree that the issue warrants attention; however, consensus dissolves around how to respond to the problem. The literature review examines one approach to solving this problem. The purpose of this paper is to explain how to implement a comprehensive solution for a behavioral approach to the complex problem and also to explain how to build a web-application system from the requirements given by the user. The web application in this paper is called the Graduate School Application Archive System (GSAAS). The GSAAS is an archived, application-management software system that is implemented as a web application to manage archived applications. It is used to manage and control a large collection of web material (student application documents). The GSAAS facilitates user creation, access control, content control, editing and many essential web-maintenance functions. The way to implement and develop these concepts is clearly explained in this paper.

ACKNOWLEDGMENTS

I would, first of all, like to thank Dr. Kendall E. Nygard for his continued support, help, and direction. My sincere thanks go to Dr. Hyunsook Do, Dr. Saeed Salem, and Dr. Bakr Mourad Aly Ahmed for serving on the committee. I would like to thank my family and friends who supported and motivated me to accomplish this task.

I would like to express my deep appreciation and gratitude to my adviser, Dr. Kendall E. Nygard, for continued support, encouragement, teaching, and invaluable guidance throughout this research. Without his help, this work would not have been completed. I would like to thank my committee members, Dr. Saeed Salem, Dr. Hyunsook Do, and Dr. Bakr Mourad Aly Ahmed, for their suggestions and encouragement to improve this work.

I would like to express my deepest gratitude to my beloved parents, Mr. N. Fazal Ahmed Khan and Mrs. Shahida Begum, and to my lovable wife, Syeda Ambreen Fathima, for their support and encouragement. I am also thankful to all my friends for their consistent help and suggestions throughout this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1. INTRODUCTION

1.1. Background

The idea behind developing this application was to help the Graduate School access the archived student applications in the Embark Application System. The archived applications are from students applying to graduate school for various departments. The requirements of the Graduate School were an online application to effectively search the applications in the database and to present application information to the faculty when required. Faculty members will only be given access to their departments' applications. The web application will be hosted in such a way that it is available only through an intranet. The above requirements were the first step for initiating in designing and developing this paper.

The paper was fully designed and developed with the help of .Net technologies. The reason .Net technology was used is because of its ease of use and because it provides a new approach for software development. .Net is the first development platform designed from the ground up with the Internet in mind. Previously, Internet functionality was simply bolted on to pre-Internet operating systems such as UNIX and Windows. That setup required Internet software developers to understand a host of technologies and integration issues. .Net is designed and intended for highly distributed software, making Internet functionality and interoperability easier and more transparent to use with systems than ever before. .Net was first introduced in 2002 as .Net 1.0 and was intended to compete with Sun Java. .Net is very easy but the basics of the C language are required.

Using the ideas presented in this paper, a Graduate School Application Archive System (GSAAS) was designed to divide the interface into considerable frames and to visualize the information in such a way that users can easily navigate through and find the options more easily. The GSAAS software was custom developed for use with accessing archived student application forms.

1.2. Research Problem Statement

The Graduate School Application Archive System (GSAAS) is an interactive, web-based application and is widely used by all departments at North Dakota State University (NDSU) for the professors/lecturers to view archived applications for students in their department. This application was developed in the .NET technology, enabling the user to view the students' graduate application forms in a well-organized manner. GSAAS is an interactive, secured, and well-organized application to view the report. This paper further discusses how the application is implemented, starting with the design phase. Because this application is more pertinent with user interaction and user experience, equal importance is given to the user-interface design and information.

Human-computer interaction has been a very important aspect of the current design. All the user interfaces are required to follow the SDLC (Software Development Life Cycle) model, and they also require a certain amount of creativity in order to provide the user with the best experience while interacting with the software. Many studies have been conducted to improve the interfaces' usability (Greg). The main functions of GSAAS would be as follows:

- To view student application forms under their corresponding departments easily.

- To view student application details more easily and accurately.

The major advantage of using this application is not only viewing information about the student report, but the report can also be viewed at the department level. Therefore, this application could also be used as storage for a knowledge database of students' application details, and the information can be shared with different access levels for different grades of users.

The information provided in this application is more reliable because these student application forms are uploaded and maintained by an administered user (user who maintains the authority for the GSAAS). The repository is secured with the server authentication method. The rest of the chapters in the paper are organized as follows. Chapter 2 explains the high-level design model of NDSU's GSAAS using use case diagrams in addition to class and flow diagrams. This chapter mainly concentrates on the design of all navigation screens. Chapter 3 explains the architecture of the application. This chapter covers all the information related to interface design as well as database design. This chapter explains the 10 rules that need to be followed while designing an interface and it also discusses how these rules are achieved in GSAAS. Chapter 4 explains, in detail, the design of GSAAS and the usability of all modules with the help of screenshots. This chapter primarily focuses on illustrating the interfaces. Chapter 5 contains the Conclusions, Limitations, and Future Work. The chapter discusses possible future study as well as the limits of GSAAS.

CHAPTER 2. HIGH-LEVEL DESIGN

The Unified Modeling Language (UML) is becoming widely used for both databases and software modeling (Ambler). UML includes diagrams for use cases, static structures (class and object diagrams), behavior (state-chart, activity, flow, and collaboration diagrams), and implementation (component and deployment diagrams).

2.1. Use Case Diagram

The main purpose of use case diagrams is to provide different views of the system. The actions of particular objects are shown in an understandable view. Therefore, it is easy for users to understand the process without going into technical details. In addition, the privileges of the users and administrators can be clearly shown. The use case model shows system actions to the external interface and also offers high-level design for the application. Use case diagrams display a better view of the interaction between the system and humans. This model possesses a very effective concept of linking to scenarios, which keeps all the activity concrete (Bell).

Three of the most important objects for UML are, first, an object as a user. A user is external to a system. A user also interacts with the system, which may be a human user or another system. The goals and responsibilities of a user are to interact with the system and to use all of the functionalities. The second object in a use case diagram is the use case; they will identify functional requirements and describe the flow of steps. The main purpose of Use Cases is to describe actions performed by a system and to capture interactions between the system and the users. The third, and most important, object in use case diagrams is Relationships. Users are connected to the Use Cases by a line that

represents a relationship between the users and the Use Cases. Figure 2.1 shows the use case diagram for GSAAS. The user for this application could be a professor, a lecturer, or an administrator. The Use Cases, or the actions that could be performed in this system, are represented in the figure. The actions that user can perform are as follows:

- **Search:** The user has the authority to search for a student name under a certain department provided that the user is authorized to view student details for that department.
- **View:** The user has the authority to view student application forms for the certain department and that department should be authorized for the user.

The actions that admin can perform are as follows:

- **Add:** The admin has the authority to add a new user and to authorize him for a certain department based on the request.
- **View/Update:** The admin has the authority to update/view the existing user and to authorize him for a certain department based on the request.
- **Delete:** The admin has the authority to delete the existing user from the GSAAS for a certain department or for the entire database based on the request.

2.2. Flow Diagram

A flow diagram in Unified Modeling Language (UML) is an interactive diagram that shows the order of processes that are operating with each other (Drewry). This diagram is usually called a Message Flow Chart. Figure 2.2 shows the flow diagram for the search and view options in GSAAS. A flow diagram contains different processes and the

messages exchanged between them, all in the order in which they occur. Flow diagram allows the specification of simple runtime scenarios in a pictorial manner.

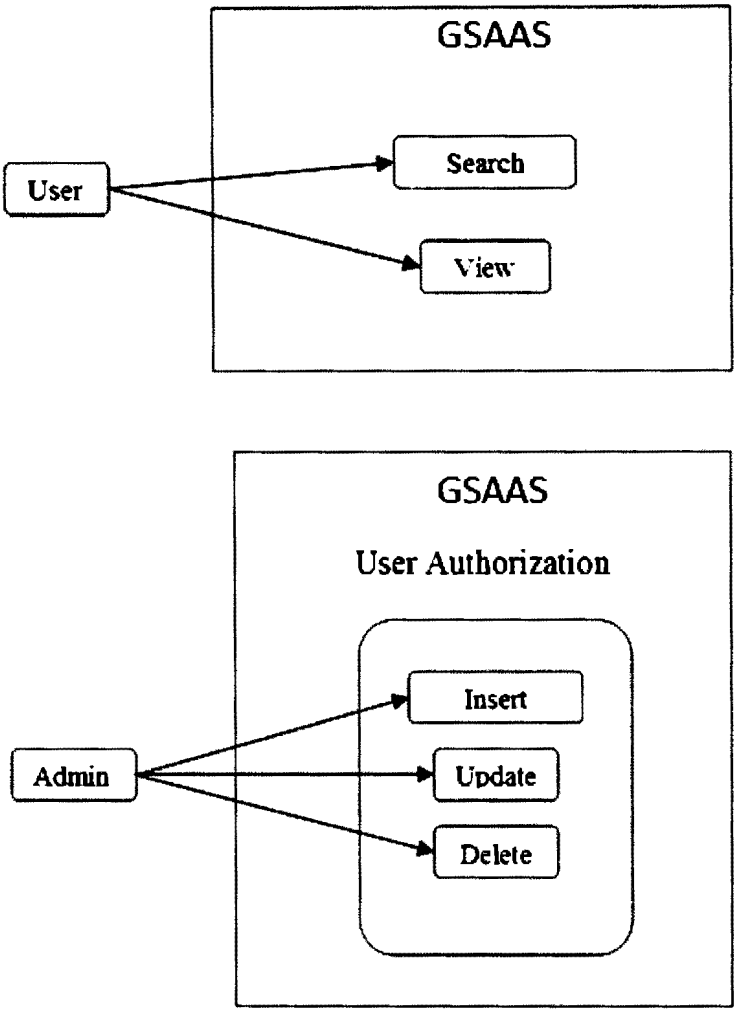

Figure 2.1 Use case diagrams.

Insert operation in GSAAS has options to insert a new user as a professor or lecturer and to authorize him for a certain department. Admin can perform two activities from this insert operation: he can insert a user and authorize several departments. Figure 2.3 shows the flow activity of actions that should be performed by the user to operate the insert option.

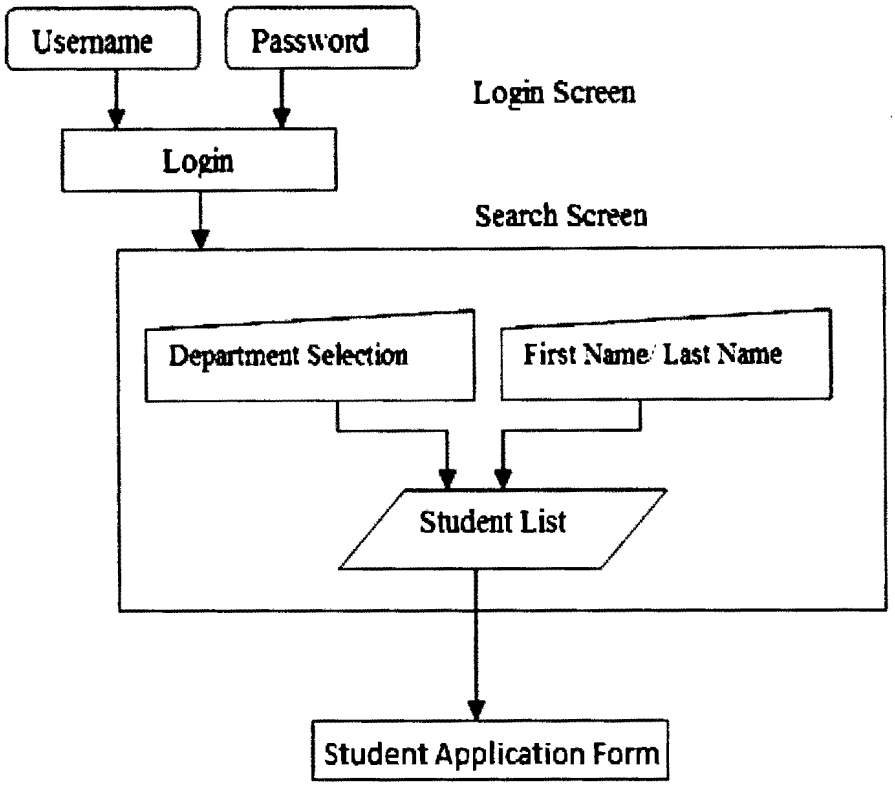

Figure 2.2 Flow diagram for search and view.

The delete operation in GSAAS can be performed to remove a user from the system: the user can be removed or unauthorized from the system by utilizing this operation. Figure 2.4 shows the flow activity of actions that should be performed to conduct the delete operation.

The view or update operation in GSAAS has various options. There are five fields that can be changed in this mode: the user can view or update the password, first name, last name, email, and user type. There is no required order to follow in this module. The administrator can view information and can update in any particular order. The most important rule in this module is that the user can only update while viewing the information. The reason for this rule is so that the user can have the option to actually see

the content in order to decide whether to make any change. This Update option is one of the reasons that GSAAS was designed with Windows forms because it will not allow the users to click the update button until he views the information. Figure 2.5 shows the flow of activities that the user can perform while utilizing the View or Update options.

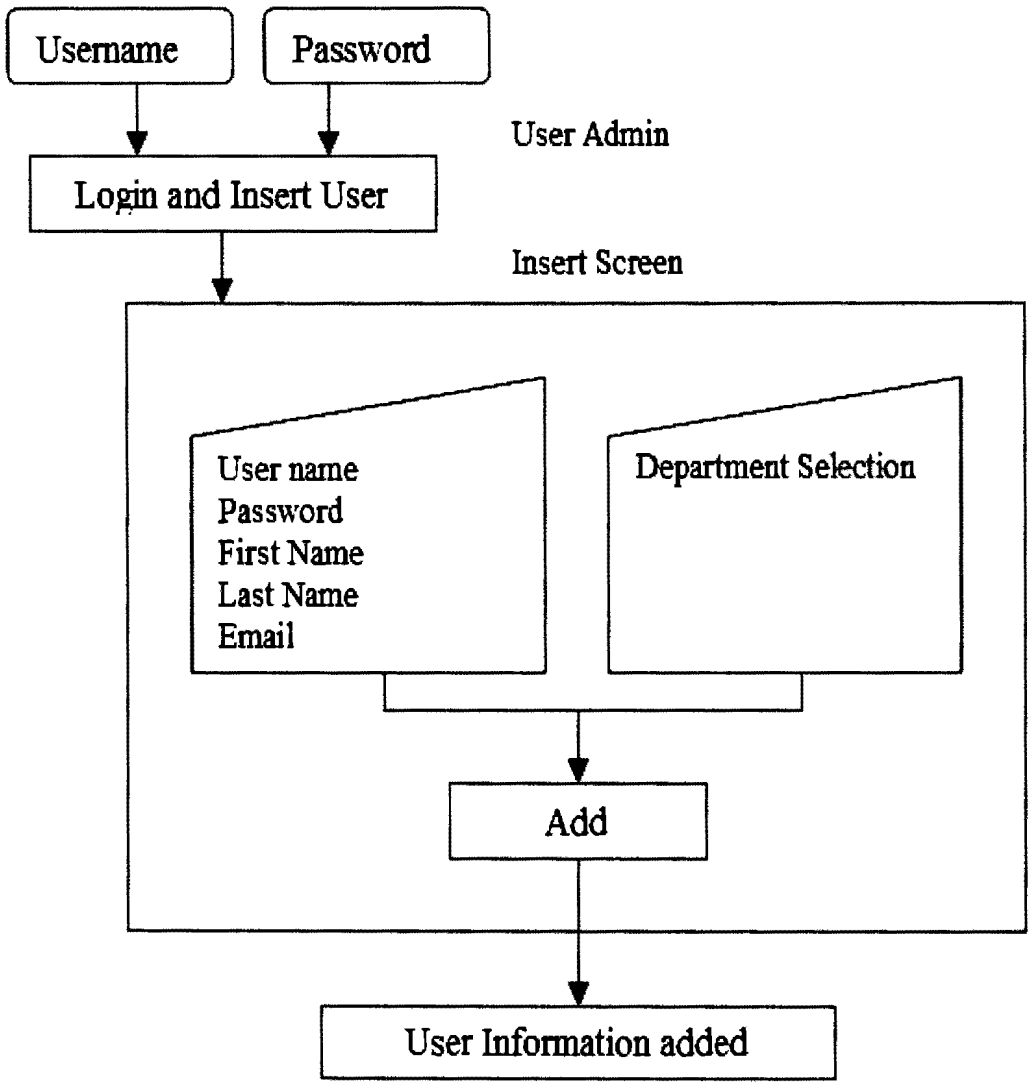

Figure 2.3 Flow diagram for add.

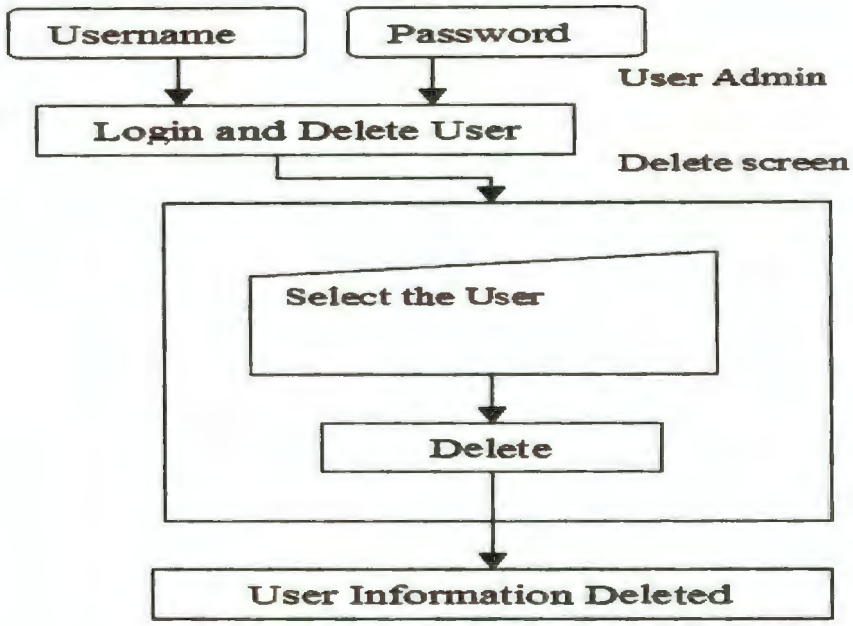

Figure 2.4 Flow diagram for delete.

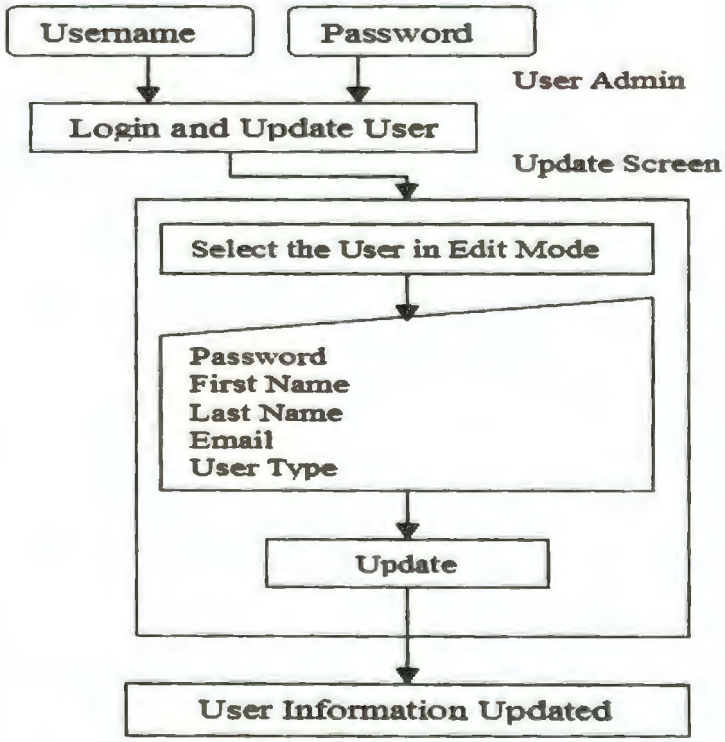

Figure 2.5 Flow diagram for update.

CHAPTER 3. DETAILED DESIGN

GSAAS is based on a three-tier architectural model that is generally used in many database applications. The application uses a Windows form for the user interface. The database stores C# (C sharp) class, Event Listener, Method, Image, and Feed information. Figure 3.1 displays the schematic representation of the GSAAS application architecture. Client-Server architecture describes the relationship between two computer programs in which one program (the client program) makes a service request to another program (the server program). Some examples for client-server architecture include e-mail exchange systems as well as web and database access. The most basic type of client-server architecture employs only two types of hosts: clients and servers. This type of architecture is sometimes referred to as a two-tier architectural model ("Client/Server Architecture"). This two-tier architecture means that the client acts as one tier while the server acts as another tier. GSAAS has a client program that interacts with the database whenever there is a data request from an end user. To make the client and server communication effective, the third tier, application layer, is included (Seguin).

3.1. The Application Architecture

GSAAS has three tiers in its design. The tiers are referred to as "layers" for the remainder of the paper to maintain consistency in terminology. The three layers in this architectural model are the database layer, the application layer, and the client layer. Each layer has a certain set of specific responsibilities.

- The database layer consists of the database management system. The data modification, as well as the addition and deletion of tasks, is performed inside this layer.

- The application layer contains the business or application logic, which connects the database layer and the client layer. With the help of the business logic, the data layer communicates with the client layer.
- The client layer is for the end users; from here, the user can interact with the system, modify the existing information, and obtain the requested details. The client layer is usually a graphical user interface.

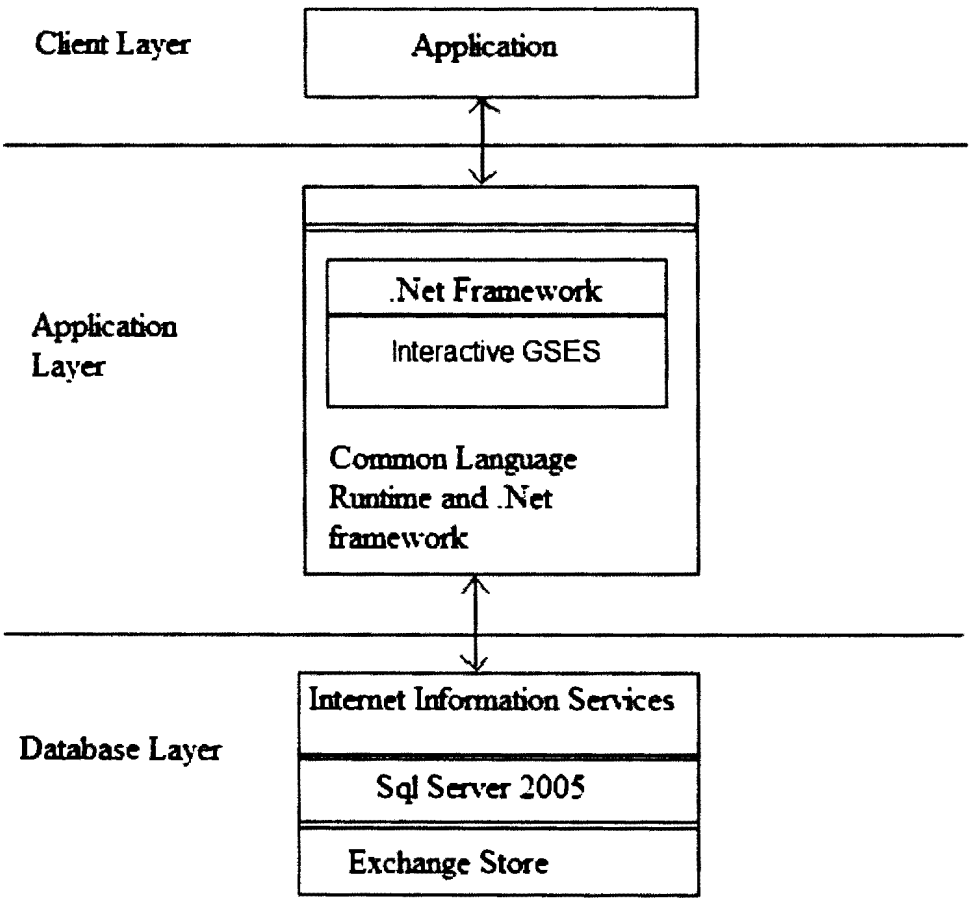

Figure 3.1 Three-tier architecture for GSAAS application.

3.1.1. Database Layer

The database layer of GSAAS is named the "Data Access Layer" (DAL). The entire script for data management (which typically includes storage and retrieval of data as

well as managing updates, and allowing simultaneous or concurrent access) is placed in this layer. The code snippet for accessing the database using C# is shown below.

```
Protected Void Login( Sender )
        // Establish a SqlConnection using new SqlConnection(Connection Object);
        //Create a SqlCommand using new SqlCommand();
        //Instantiate the SqlConnection;
        //Query the databse to insert a class;
        //Pass the parameters to the query using SqlCommand;
        //Open the Connection using open()
        //Execute the query using ExecuteReader();
        //Close the Connection using close();
```

The GSAAS database is not vulnerable to SQL (Structured Query Language) Injection. It is a technique to inject SQL query/command as an input in a web page. Many web pages take parameters from the user and make an SQL query to the database.

For instance, when a user logs in, the user name and password are sent as SQL parameters to run a query on the database to see if the user inputs are valid. With SQL injection, it is possible for a hacker to send a crafted user name and/or password field that will change the SQL query and, thus, grant the hacker access to the database. To handle these scenarios, GSAAS is built with a function to filter special characters or any crafted password to avoid such SQL injection.

3.1.2. Application Layer

The application layer contains the logic for performing user-invoked operations. When a user requests to change the data, this layer takes the instructions and, by using the

existing logic, performs the database modifications, giving the results back to the user. If the user wants to store some of the information, then the application layer accepts the data from client layer, communicates with the database layer, and stores the information.

Components of the application layer include the framework, which has been built into class libraries with the exception of handling. If the common business logic is used in different applications, then the application layer can be shared. The framework used in the "GSAAS" application is .Net framework 2.0. The C# language is the programming language used to write business logic in the application layer ("Visual C#").

3.1.3. Client Layer

In the case of a Windows application, the client is the User Interface. The Windows interface processes the requests, displays resources, issues user requests for resources, and processes responses. The following attributes are the basic set of features available in the client layer of GSAAS.

- Administrative users can view, modify, and add users in the database; they can also view and search in the database.
- Users can view and search in the database.
- Administrative users can delete users in the database.

3.2. Rules Followed for Front-End Design

Being a user-oriented, interactive software application, the Graduate School Application Archive System (GSAAS) required a significant amount of attention devoted to front-end design details. Before the application was created, numerous existing designs and interface design rules were considered (Shawn; Greg). It is a well-known fact that some unsuccessful (even highly functional) software proved that the user's interest in the

software mainly depends upon the user's experience with the interaction. Consequently, a recent decline in Microsoft operating systems sales shows that people preferred Windows XP to Windows Vista. Windows Vista contains a great deal of extra functionalities; however, this excessive functionality has made it difficult for users to understand the interface (McCraken). Microsoft Office 2007 was initially unpopular because its expanded menu options were displayed on the front end. Considering all these drawbacks, GSAAS was designed by following the "keep it simple" rule. Despite the fact that there is a significant amount of information displayed in the user interface, the application was designed to be very consistent.

3.2.1. The Ten Rules

Conducting a great deal of research, Raizlabs came up with ten "Interface Design and Development" rules. These rules are frequently considered for maintaining the consistency and also for including the necessary options ("Ten Rules of Interface Design"). This section explains about how each rule was adopted when designing the present application.

3.2.2. Probability

The number of options should be limited for any software or user interface. A greater number of options on a screen will increase the probability of distraction and then, the user may not be comfortable using the interface. During the initial stages of design, the screens were planned according to this rule so that the user can see only login, search user, and user admin. Under search, the options view and selections were included. Currently, the admin can see add, view/edit, an delete on user admin screen as well as logout on the

main interface. Even after this change in the design, the probability rule is strictly followed because the options available on the interface are all equally important.

3.2.3. Statistics

The rule of statistics states that a designer can satisfy only a certain percentage of users at a given point in time. Subsequently, the design has undergone various changes. The application was very basic, containing only the basic .Net framework information. To attain a higher degree of user satisfaction by making an adequate amount of data visible, the application was expanded to include graphics. User satisfaction was not directly considered because this application was not used or rated by another person.

3.2.4. Flow

The flow rule suggests a smooth transition between pages while accomplishing a task. This rule is followed by making all details available on the main screen. All exceptions are designed in such a way that the message will only appear for necessary actions, and it is very easy to navigate to the Login screen and start again.

3.2.5. Text

This rule is about the instructions used to guide the user in performing tasks. The instructions should be clear and precise, and the text used in the application should be common language and easy to read. When the user is attempting to perform any action, a precise message for unavailable data will be displayed on the screen.

3.2.6. Visual Design

This rule emphasizes that, along with the visual appearance, the interaction should also be given importance. The review is done after this step in order to verify that the options are visually effective as well as useful. The illustration of initial visual designs is

shown in Figure 3.2. On the login screen, the user can perform a single entry and login to enter GSAAS application.

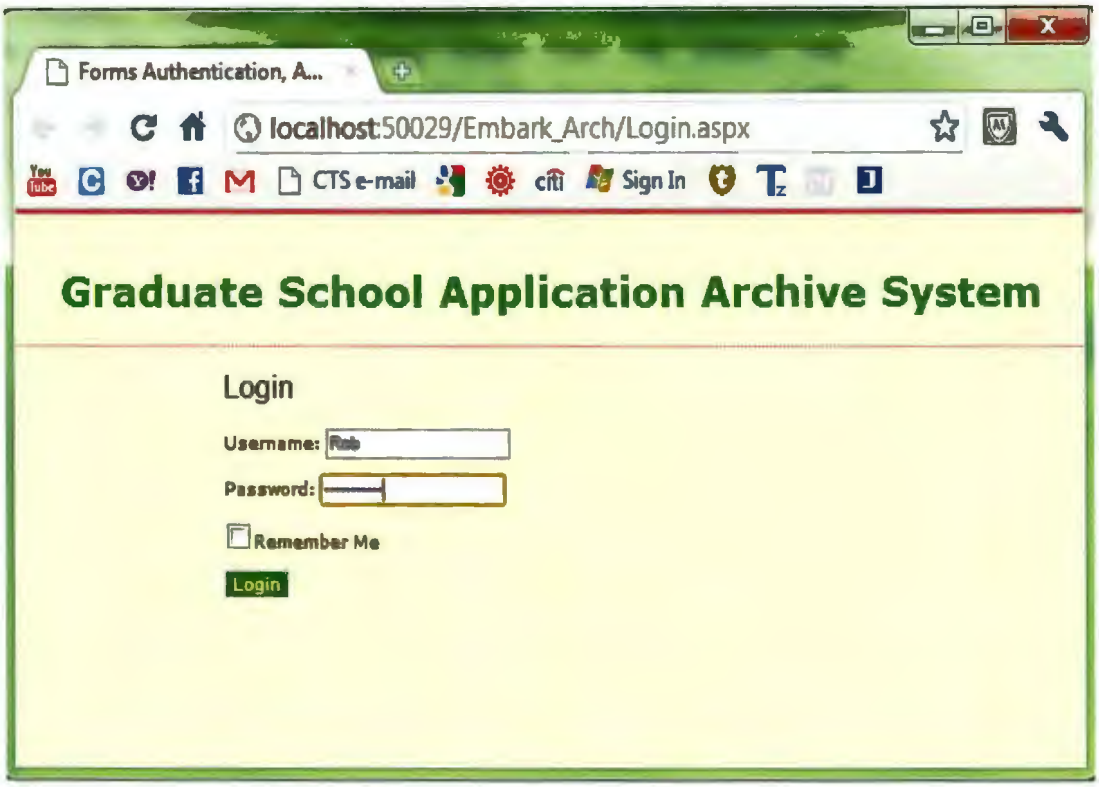

Figure 3.2 Illustration of visual design.

3.2.7. Controls

The controls should be placed appropriately. To show control buttons like login, search, and user admin, a picture box is used for viewing the information.

3.2.8. A Square Wheel

A "Square Wheel" rule follows well-practiced methods rather than adhering to completely new methods. This rule suggests following the existing patterns to write logic and to design the interface. Therefore, GSAAS has followed an abstract factory of creational patterns (The creational pattern is one of the design patterns.). By following this abstract factory, it will attempt to catch event and is used to throw meaningful exceptions.

3.2.9. Design for Mistakes

This process provides options for the users to correct mistakes. All information entered by the user is editable. Nothing is final until the user clicks the event (i.e., login, search, and user admin) button.

3.2.10. Talk to Your Users

This rule suggests getting opinions from all classes of users before and after creating the application. GSAAS application has attempted to follow well-accepted designs; however, the user's opinion is not taken into consideration because of various limitations. Intensive testing is done to ensure that all of controls are working correctly and that they are in the appropriate places.

3.2.11. Iterate

The design should be reviewed constantly to find any hidden faults. The design of this tool involved a great deal of iterations, and changes were made to include all of possible details.

The interface design in the first iteration consists of the Login screen, and the total application is separated into three parts. The first screen displays the Student Application search, consisting of the tree hierarchy of the department and student name under each professor/lecturer. The second screen displays the User Admin and consists of all users (i.e., professors/lecturer) who use this GSAAS application. The third screen consists of managing all the user details.

When the user enters the login screen, a search screen (Figure 3.3) will appear in GSAAS. The intial design expected the user to enter the student name alone for the search criteria while department was optional, but the design was modified to let the user select

the department as the primary key for the search and then proceed with the student name where the name is an optional field for the next level of view.

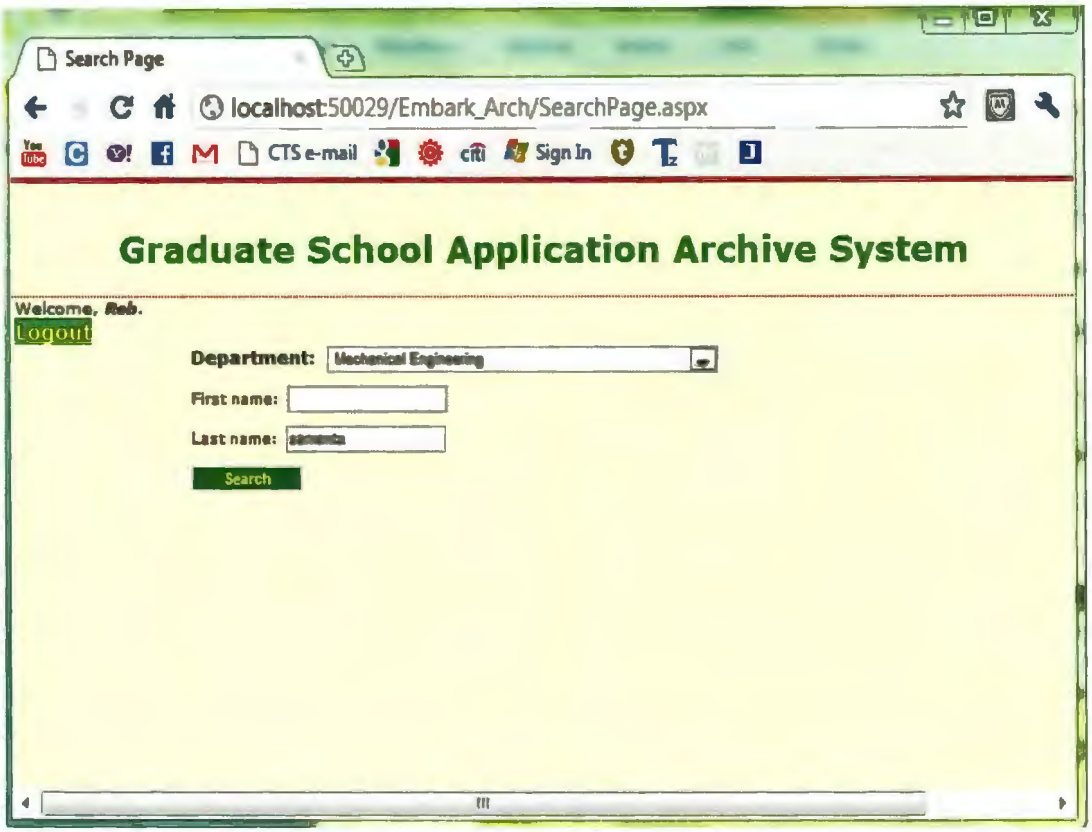

Figure 3.3 Search screen.

Department is the first level of the search, and name is optional because the user can restrict the student details pertaining to a particular department for which he or she has authorization. On a blank selection of department, no error message was displayed in the intial design, but the department was made mandatory in a later design to enhance the user's search criteria. Figure 3.4 shows the error message, as displayed to the user, when the department field is left blank.

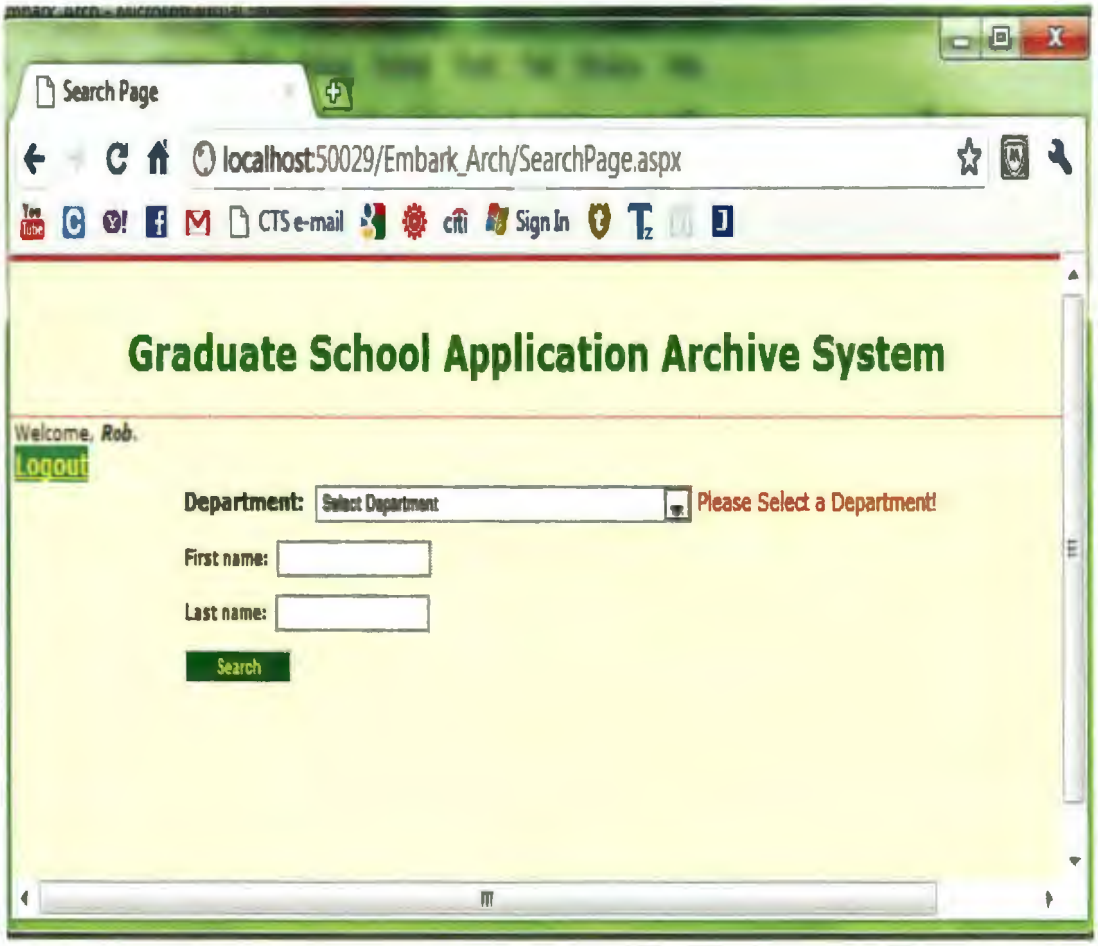

Figure 3.4 Validation of department on search screen.

3.3. Back-End Design

The back-end design in GSAAS is achieved using C# scripting and structured query language through the SQL client. The back end is very important for supporting the application and also for providing quick connectivity. The back end of GSAAS is comprised of main components like Database Management Systems (DBMS) and Scripting.

3.3.1. Database Management Systems (DBMS)

This section discusses the reasons for using the Microsoft SQL Server database in comparision to other commercially available DBMSs. The SQL Server is a large-scale

DBMS with the features and ability to manage very large quantities of data. Its design is ideally suited for managing databases that are typical of many Windows database applications (Samuel). The Microsoft SQL Server provides a data platform with many advantages and business benefits. These benefits include better security, performance, and scalability; developer productivity; and Business Intelligence tools (BI).

The database management system used in GSAAS manages the relationships in the database. The database used here is a relational database consisting of several data tables connected to each other. Each data table includes several records, and each record is a collection of fields containing related information. The DBMS is a set of components designed by administrators for managing the database in an effective manner. A DBMS is a set of components used for defining, constructing, and manipulating a database. The relational databases designed for GSAAS store and manage relationships between the data.

3.3.2. GSAAS Database Structure

The SQL Server management edition database (free edition for students) is used as the database for GSAAS. This tool displays information about the professor/lecturer, student application form, and department details.

There are five database tables designed and related to accommodate all this information in an organized way. For example, the "USER_TB" table is designed to store information about the user login credential and role. New users can be added, and existing user information can be modified or deleted. The "USER_DEPT" table is used for storing details about the user related to the department. The "USER_NAME" field in the table relates to "USER_TB" to map the user and his corresponding department. This table is updated for the addition of a new user.

The "DEPT_TB" table contains details about the Department ID and Department Description. The "STUDENT_TB" table contains complete details about the students with their names and their corresponding application form paths. The "DEPARTMENT_PROGRAM" field in this table relates to "DEPT_TB" for the "DEPARTMENT_ID" field, is mapped to the corresponding user (professor/lecturer) by the "USER_DEPT" table, and displays student details for the mapped User. The "LOG_TB" table is used to store the log details of user activity in GSAAS. All five tables used in the database are shown below.

Table 1. Schema for "USER_TB" table

COLUMN NAME	DATA TYPE	PRIMARY KEY	AUTO-INCREMENT
User_Name	nvarchar(50)	Y	N
Password	nvarchar(50)	N	N
First_Name	nvarchar(50)	N	N
Last_Name	nvarchar(50)	N	N
Email	nvarchar(50)	N	N
User_Type	nvarchar(50)	N	N

Table 2. Schema for "USER_DEPT" table

COLUMN NAME	DATA TYPE	PRIMARY KEY	AUTO-INCREMENT
User_Name	nvarchar(50)	N	N
Dept_Id	nvarchar(50)	N	N

Table 3. Schema for "DEPT_TB" table

COLUMN NAME	DATA TYPE	PRIMARY KEY	AUTO-INCREMENT
Dept_ID	nvarchar(50)	Y	N
Dept_Name	nvarchar(50)	N	N

Table 4. Schema for "STUDENT_TB" table

COLUMN NAME	DATA TYPE	PRIMARY KEY	AUTO-INCREMENT
Student_Id	nvarchar(50)	N	N
Last_Name	nvarchar(50)	N	N
First_Name	nvarchar(50)	N	N
Gender	nvarchar(50)	N	N
Birthdate	nvarchar(50)	N	N
Email	nvarchar(50)	N	N
Year_Of_Entry	nvarchar(50)	N	N
Term_Of_Entry	nvarchar(50)	N	N
Date_Applied	nvarchar(50)	N	N
Department_Program	nvarchar(50)	N	N
Program_Type	nvarchar(50)	N	N
Citizenshipcountry	nvarchar(50)	N	N
Citizenship_Status	nvarchar(50)	N	N
Grad_School_Decision	nvarchar(50)	N	N
Path	nvarchar(50)	N	N

Table 5. Schema for "LOG_TB" table

COLUMN NAME	DATA TYPE	PRIMARY KEY	AUTO-INCREMENT
Username	nvarchar(50)	N	N
Page	nvarchar(50)	N	N
Log_Message	nvarchar(150)	N	N
DateTime	Datetime	N	N

3.3.3. Challenging Issues and Solutions

In this section, the challenges faced while designing the application are discussed. The primary objective is to make this information more user friendly. After a significant amount of research on user interfaces, it was decided to integrate a web-based application in a Windows-based application (Seguin). Therefore, two kinds of applications can be displayed in GSAAS: one is the student application search by the user, and the other is managing the user in the same application dynamically. To make this application user friendly was a more difficult task because there was no user opinion available. I analyzed and brought this application more interactive to the user by a complete test on all search criteria.

The next challenging issue was to give extra privileges to users in the GSAAS application. Some programming changes were made to enable users to not only view the necessary information, but also to add more information to the existing application. This extra feature is the user administration part where the user can include; modify the existing information; and delete user information, such as the user profile and personal details, from the existing databases.

CHAPTER 4. APPLICATION AND USABILITY

This chapter explains all options available in the application. Screenshots are used to locate the options on the user interface.

4.1. Home Interface

Figure 4.1 shows the main interface for GSAAS. There are many features that follow the standard rules for designing interfaces included in this application. GSAAS displays proper user friendly messages if there is a wrong entry made by the user.

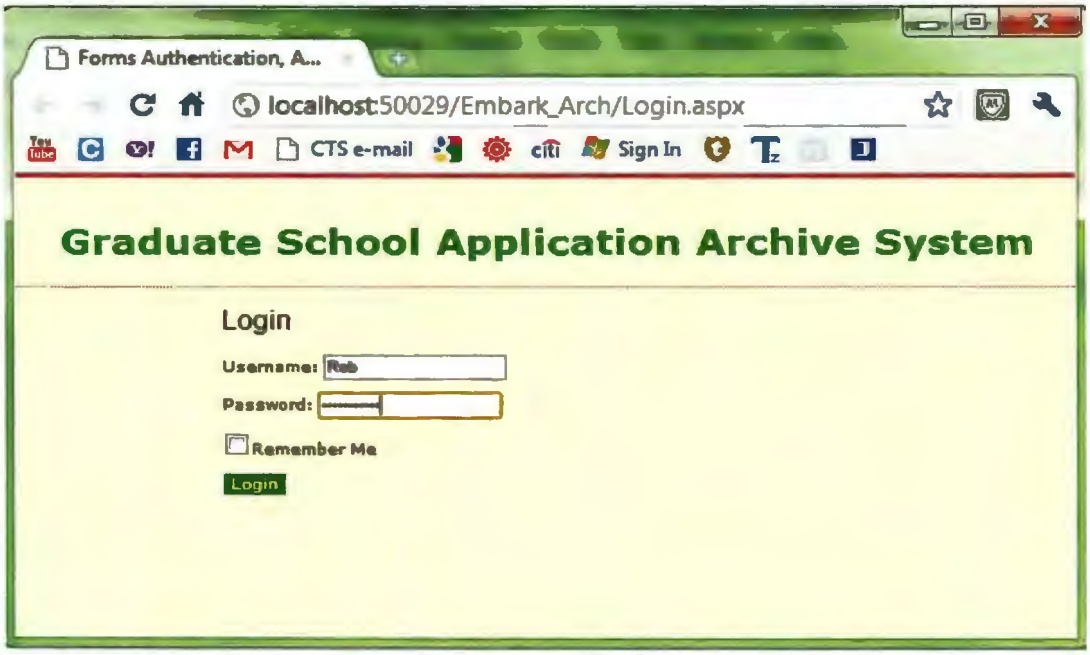

Figure 4.1 Login interface for GSAAS.

4.2. Search Screen

Figure 4.2 shows the interface screen when an administrator logins into GSAAS. When the administrator logins he will have access to the search screen and user admin button which is used to add, update or delete any user information. Figure 4.3 shows the search screen when a user logs inn without any administrator privileges. The user admin button is hidden so that the user cannot change or update other user information.

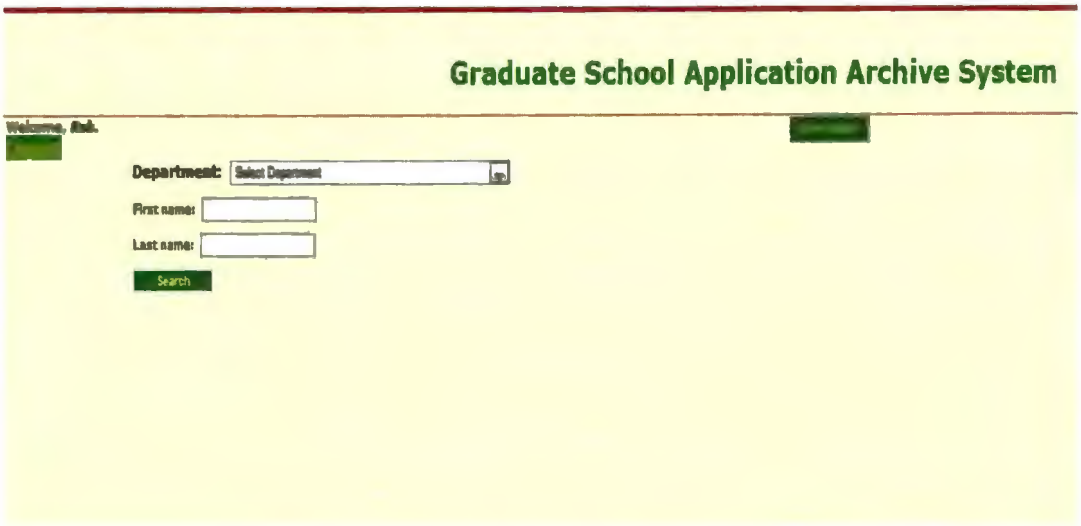

Figure 4.2 Interface when a user logs into the system as user admin.

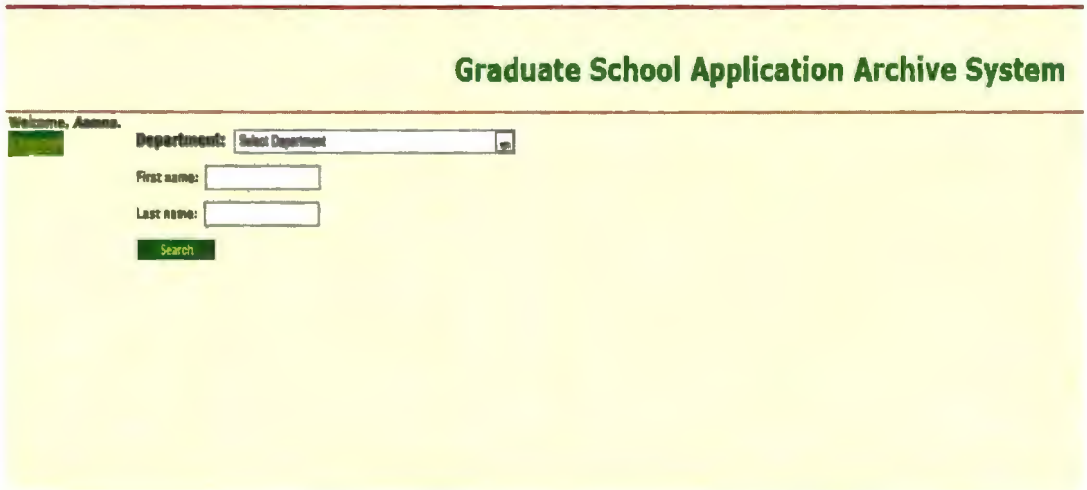

Figure 4.3 Student search screen for GSAAS.

4.2.1. Select Student by Department

Figure 4.4 shows the search screen of GSAAS, when only the department is selected and the name fields are left blank. GSAAS retrieves all student information from the department selected. The user has access to departments which are set by administrator and the administrator will have access to all the departments in the university.

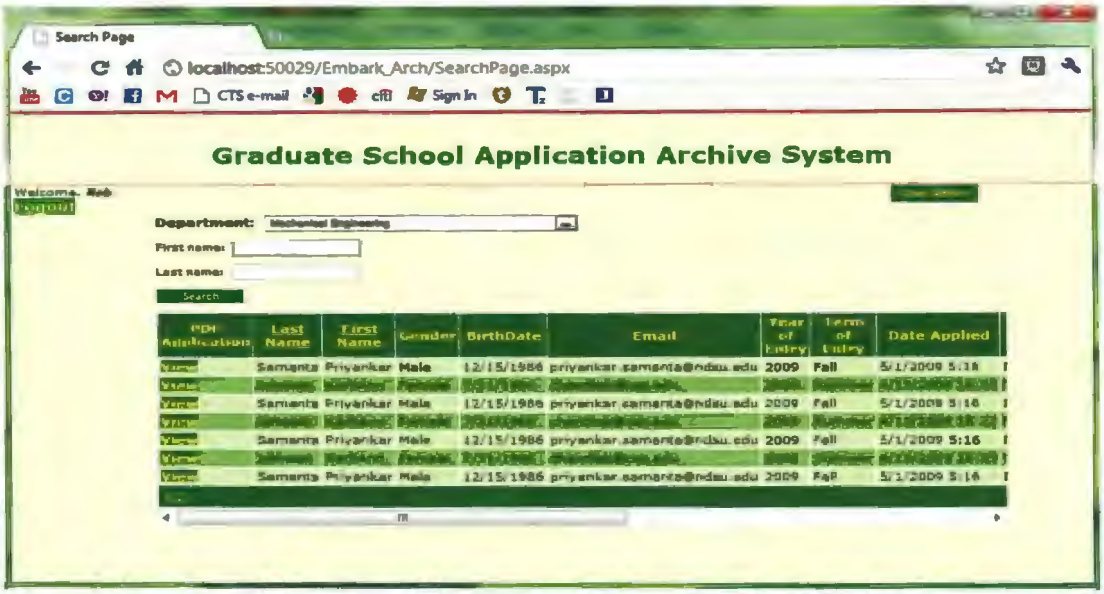

Figure 4.4 Student search screen for GSAAS by department.

4.2.2. Select Student by Department and First Name

Figure 4.5 shows the search screen of GSAAS, when the department and first name is filled. GSAAS retrieves all the student information from the department selected and filters the students list on first name entered.

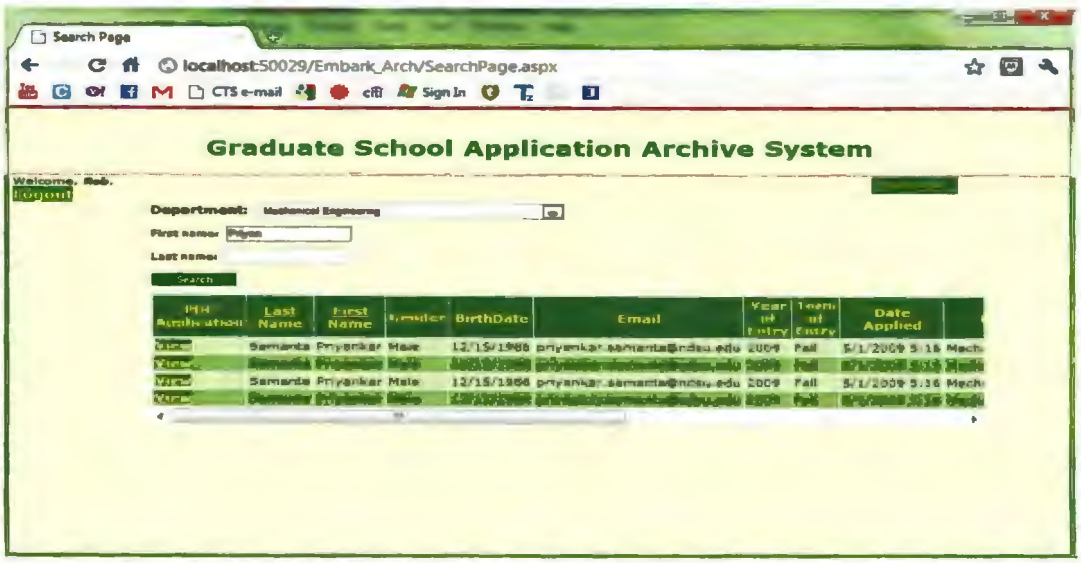

Figure 4.5 Student search screen for GSAAS by department and first name.

4.2.3. Select Student by Department and Last Name

Figure 4.6 shows the search screen of GSAAS, when the department and last name is filled. GSAAS retrieves all the student information from the department selected and filters the students list on last name entered.

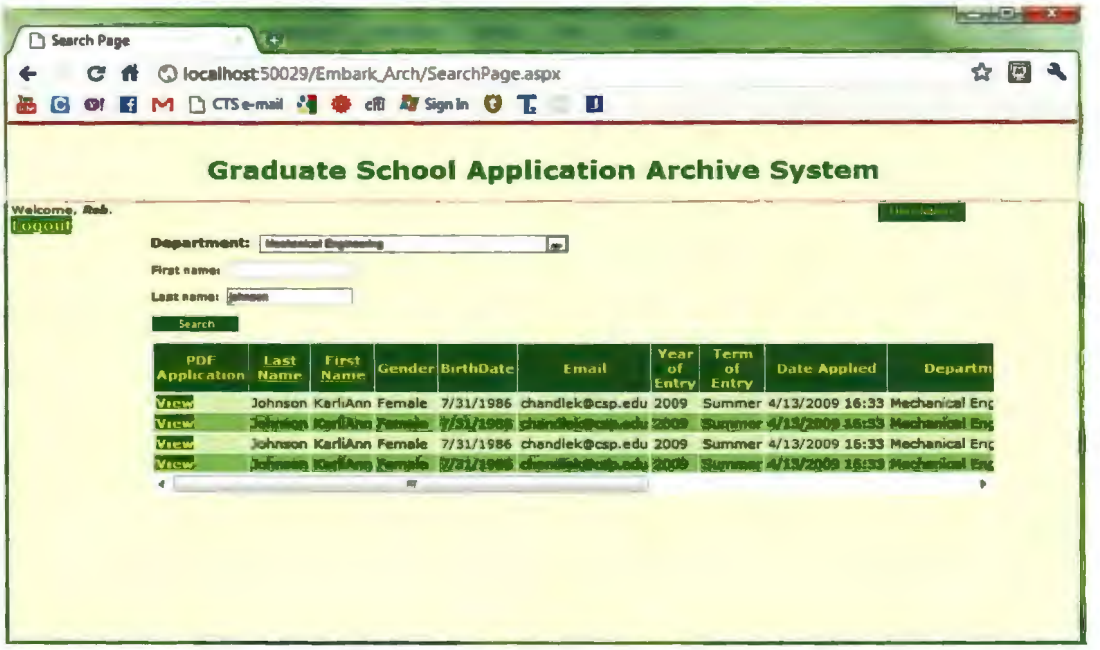

Figure 4.6 Student search screen for GSAAS by department and last name.

4.2.4. Student Application Form on Selection

Figure 4.7 shows the student application form when viewed from the search screen of GSAAS, to view the application the user needs to click on the view link corresponding to the student in the result grid of search screen.

4.2.5. Error Message for Department Selection

When the user has not selected a department, GSAAS will give an error message, "Please Select Department," which is shown in Figure 4.8. Similarly, when the user selects a department and there is no student in that department, then it will show an error message, "There are no data records to display," which is shown in Figure 4.9.

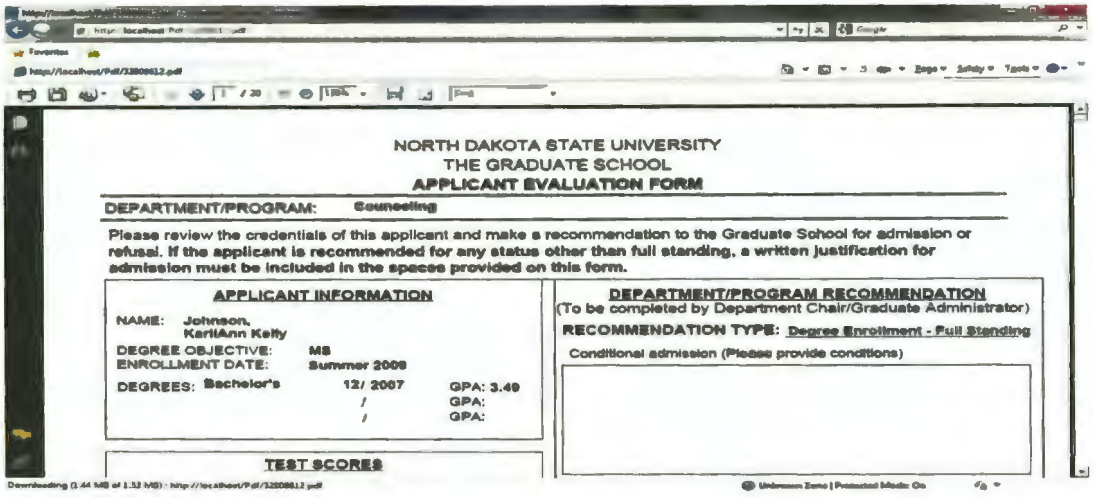

Figure 4.7 Student application form when selected to view.

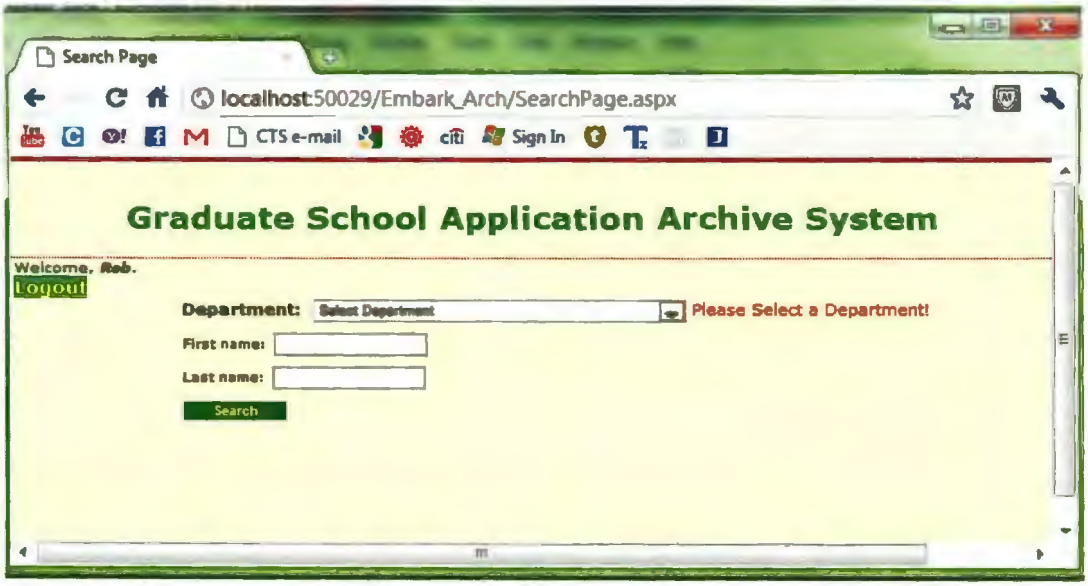

Figure 4.8 Exception message for department selection.

4.3. User Admin Selection

When the user has clicked on Select will display the department accessible for the particular user as shown in Figure 4.10. When the user has clicked on Edit, it will allow the user Admin to change the user credential Figure 4.11. After selecting the delete option, the user is removed from GSAAS.

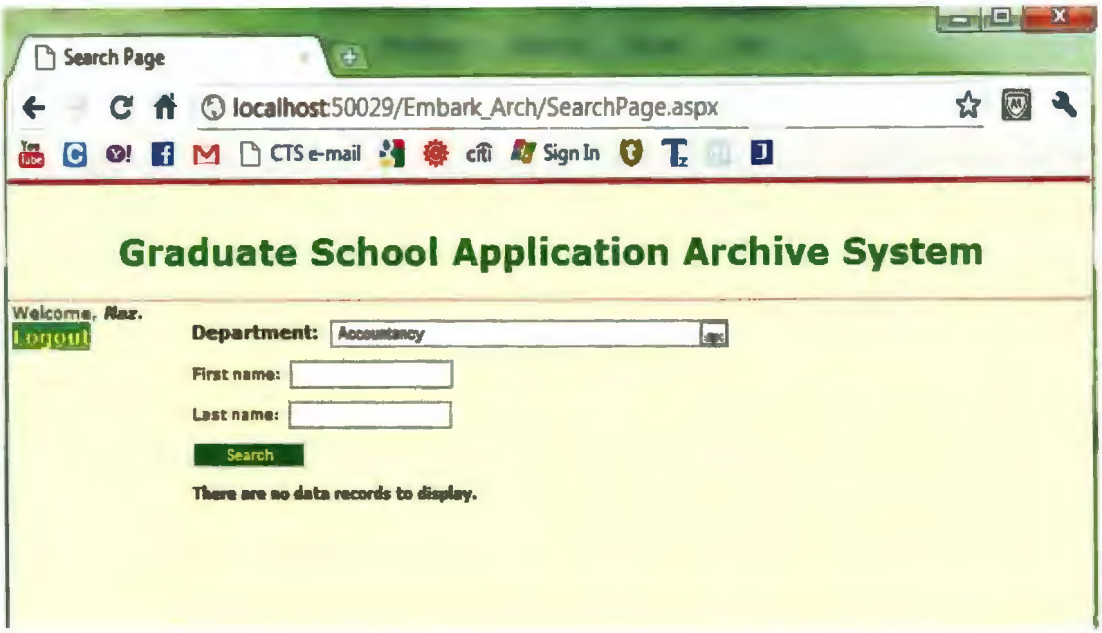

Figure 4.9 Exception message for no record found.

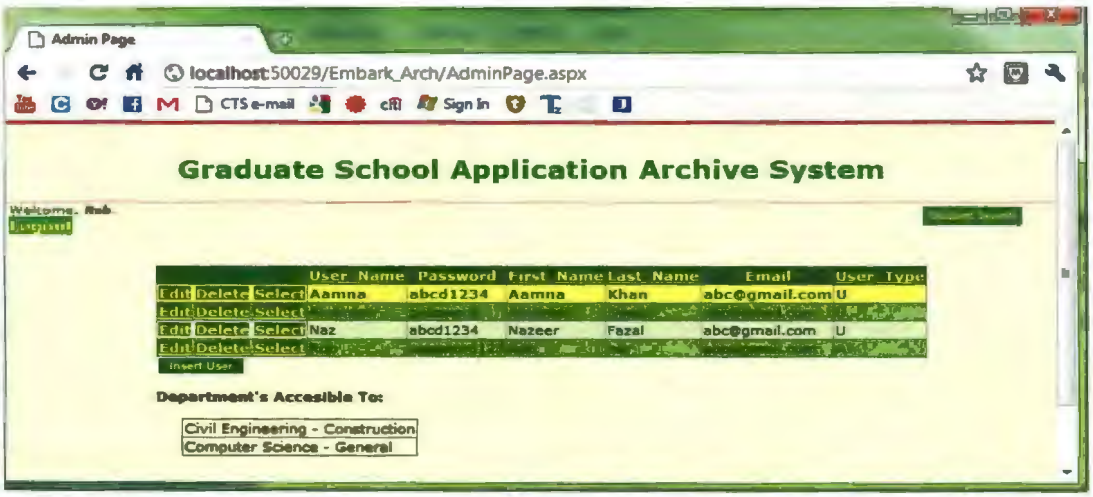

Figure 4.10 User admin selections on user

4.4. Adding a New User

Figure 4.12 shows the "Insert User" button to add a new user to GSAAS. When using this option, GSAAS allows creating a user in the system. Figure 4.13 shows the screen used to add a new user with all his credentials. Figure 4.14 shows the newly created user details and his department authorization.

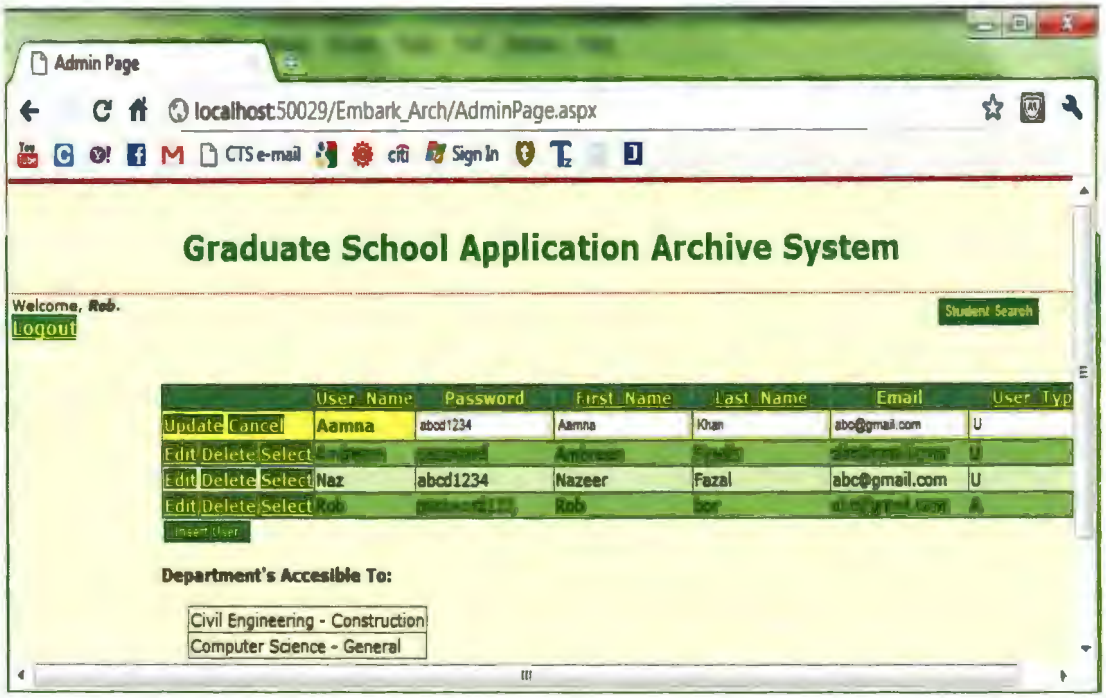

Figure 4.11 User admin editing user details.

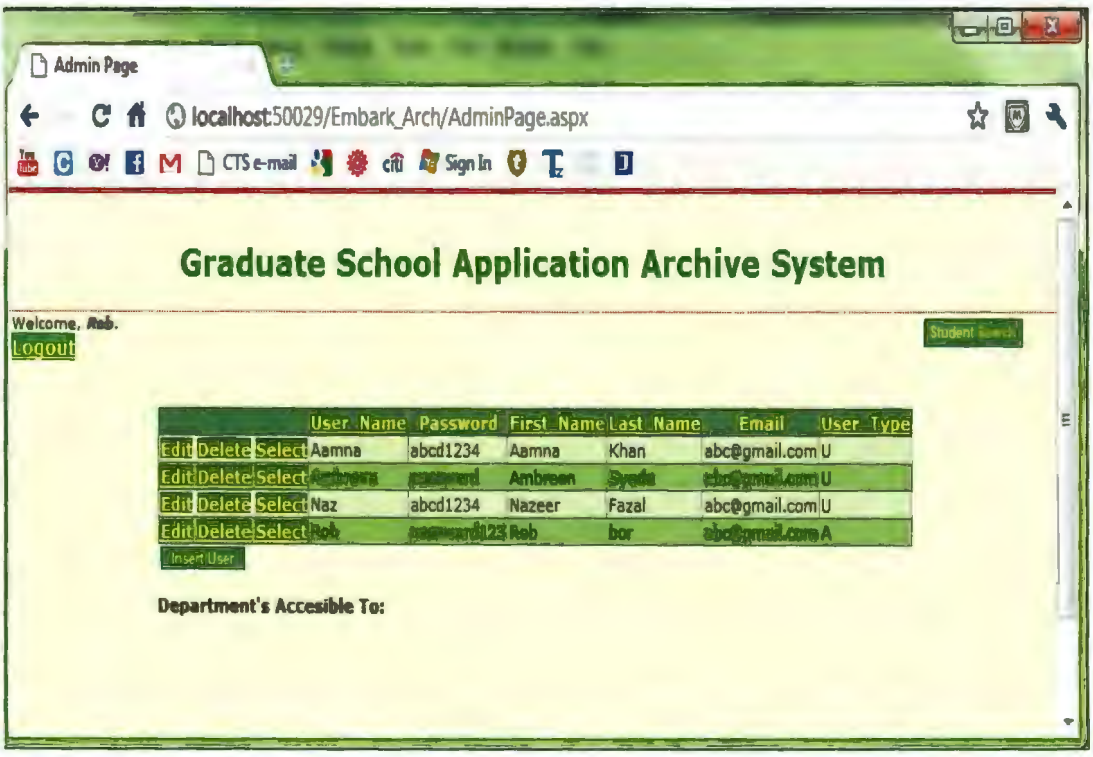

Figure 4.12 User admin selections on insert user.

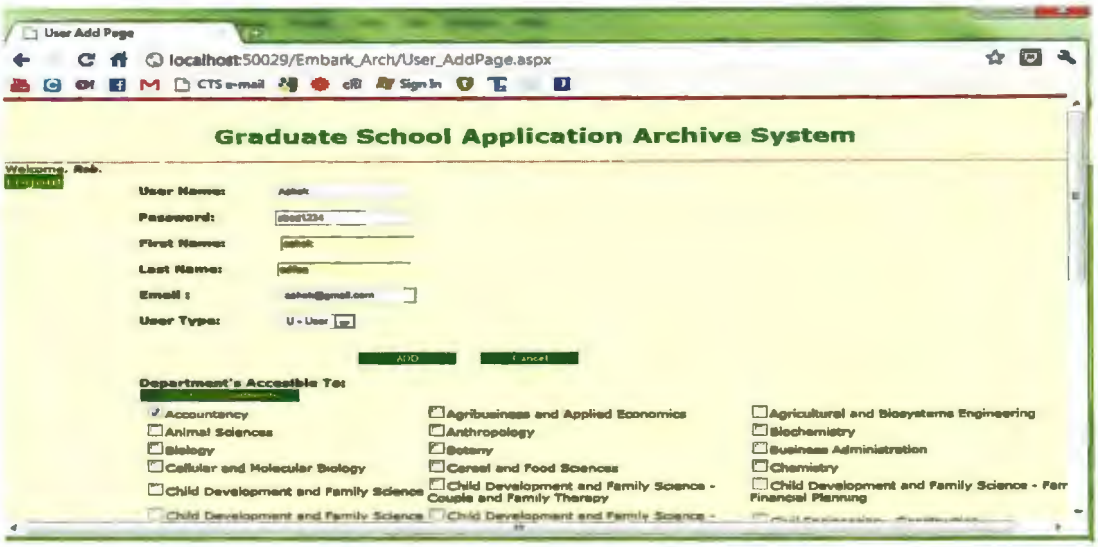

Figure 4.13 User admin new user creation screen for GSAAS.

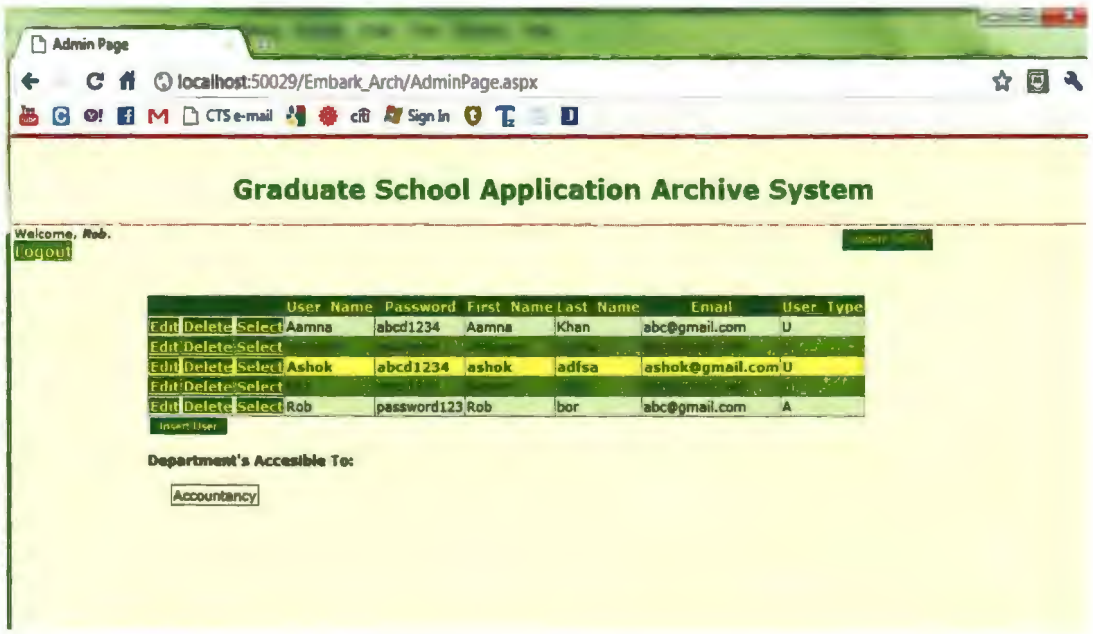

Figure 4.14 User admin's new user details.

4.5. User Application Access Log

Figure 4.15 shows user access log activity which is logged in our database for security and auditing purpose.

UserName	Page	Log_Message	DateTime
Rob	Login	User Logged in Successfully	14-04-2010 01:23:50
Rob	SearchPage	Search -Civil Engineering - Construction||	14-04-2010 01:23:57
	Logout	User Logged Out Successfully	14-04-2010 01:23:58
Rob	Login	User Logged in Successfully	14-04-2010 01:25:07
Rob	SearchPage	Search -Anthropology||	14-04-2010 01:25:10
	Logout	User Logged Out Successfully	14-04-2010 01:25:12
Rob	Login	User Logged in Successfully	14-04-2010 01:31:48
Rob	User_AddPage	AddUser -Ambreen	14-04-2010 01:32:40
	Logout	User Logged Out Successfully	14-04-2010 01:32:42
Rob	Login	User Logged in Successfully	14-04-2010 01:57:04
	Logout	User Logged Out Successfully	14-04-2010 01:57:40
Rob	Login	User Logged in Successfully	28-12-2010 20:31:47
Rob	SearchPage	Search -Electrical and Computer Engineering|Samanta|	28-12-2010 20:33:25
Rob	SearchPage	Search -Mechanical Engineering|Samanta|	28-12-2010 20:33:39
Rob	SearchPage	Search -Mechanical Engineering||Samanta	28-12-2010 20:33:52
	Logout	User Logged Out Successfully	28-12-2010 20:34:20
	Logout	User Logged Out Successfully	28-12-2010 20:34:21
Rob	Login	User Logged in Successfully	28-12-2010 22:20:34
Naz	SearchPage	Search -Accountancy||AURA|	28-12-2010 22:49:49
Naz	SearchPage	Search -Accountancy|Laura|	28-12-2010 22:49:53
Rob	SearchPage	Search -Software Engineering||	28-12-2010 22:51:37
	Logout	User Logged Out Successfully	28-12-2010 22:53:59
	Logout	User Logged Out Successfully	28-12-2010 22:54:06

Figure 4.15 User access log maintained internally for security reasons.

4.6. GSAAS Exit Screen

Figure 4.16 shows the user exit screen which is displayed when the user log's out from the GSAAS application by clicking the logout button on the application.

Figure 4.16 User exit screen

CHAPTER 5. CONCLUSIONS AND FUTURE RECOMMENDATIONS

5.1. Conclusions

The "GSAAS" application is a simple-to-use, Windows-based solution to make the student application form easier to be viewed by user. This application can be used to understand students' specialization and their special interest about a particular department, which helps the user assess the student. This application can create an optimal way in positioning the student in the correct department by using all the student credentials. This application does not have any impact on the number of active users due to the fact that the database can be distributed to every user individually. This application also has the capacity to store an enormous amount of data.

5.2. Limitations

The limitation and the challenge of GSAAS are to provide a user interface screen for a specific user to add student details (i.e., student application form). It should be noted that this application currently provides a broad structure upon which several other modules can easily be developed and attached.

5.3. Future Work

Numerous enhancements, which could increase the scope of GSAAS features, can be performed in addition to the existing applications. A number of possibilities for future work and enhancements are discussed below.

The present GSAAS does not provide the ability to upload the Student application form through user entry screen. Another application where the student can directly download an online detail entry form and these data will be automatically uploaded in the GSAAS database instead of the manual back-end process of the current system. The

present interactive tool for GSAAS does not provide the ability to help users accessing the system on World Wide Web. Therefore, the feature could be improved by allowing access to the application from online sources as well.

REFERENCES

Ambler, Scott W. "UML 2 Use Case Diagrams." *Agile Modeling: Effective Practices for Modeling and Documentation*. 2003. Web. Apr. 2011. <http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>.

Ambler, Scott W. "User Interface Design Tips, Techniques, and Principles." *Ambysoft*. 2005. Web. Aug. 2011. <http://www.ambysoft.com/essays/userInterfaceDesign.html>.

Bell, Donald. "UML Basics: The Class Diagram." *IBM - United States*. DeveloperWorks, 15 Sept. 2004. Web. June 2011. <http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/>.

"Client/Server Architecture." *Microsoft TechNet: Resources for IT Professionals*. Microsoft Corporation, Nov. 2010. Web. Feb. 2011. <http://technet.microsoft.com/en-us/library/cc917543.aspx>.

Drewry, Tony. "Data Flow Diagrams." *UWE-CEMS*. Oct. 2005. Web. June 2011. <http://www.cems.uwe.ac.uk/~tdrewry/dfds.htm>.

Greg. "Ten Rules of Interface Design." *UI - User Interface*. Tribe, 3 Apr. 2004. Web. May 2011. <http://ui.tribe.net/thread/17ddc2e7-deb6-4cda-b114-9ba28b2e4ebc>.

J, Samuel. "Introduction to SQL Server." *Samueldotj.com*. Web. Sept. 2011. <http://www.samueldotj.com/Articles/Introduction%20to%20SQL%20Server.pdf>.

Seguin, Karl. "Foundations of Programming - Building Better Software." *Openmymind.net*. CodeBetter.Com, 2007. Web. Sept. 2011. <http://openmymind.net/FoundationsOfProgramming.pdf>.

Shanker, Greg. "Microsoft SQL Server and Oracle Database: A Comparative Study on Total Cost of Administration (TCA)." Alinean, Inc., May 2006. Web. Jan. 2011. <http://www.alinean.com/PDFs/Alinean-MicrosoftAndOracleTCAStudy.pdf>.

Shawn. "8 Golden Rules of Interface Design." *Devirtuoso*. 28 May 2009. Web. Feb. 2011. <http://www.devirtuoso.com/2009/05/8-golden-rules-of-interface-design/>.

"The Ten Rules of Interface Design - Quick UI Tips for Designing Software." *Raizlabs - Photo Software, Interfaces Design*. 16 Apr. 2009. Web. Mar. 2011. <http://www.raizlabs.com/interface/top10.asp>.

"UML Use Case Diagrams: Guidelines." *MSDN - Microsoft Developer Network*. Microsoft, 2010. Web. Aug. 2011. <http://msdn.microsoft.com/en-us/library/dd409432.aspx>.

UML Use Case Diagrams. Rep. Object Mentor, Nov. 1998. Web. Mar. 2011. <http://www.objectmentor.com/resources/articles/usecases.pdf>.

"Visual C#." *MSDN - Microsoft Developer Network*. Microsoft, 2010. Web. May 2011. <http://msdn.microsoft.com/en-us/library/kx37x362.aspx>.

McCracken. "Windows XP vs. Vista: An Explosion of Opinion." 20 Mar. 2008. <http://www.pcworld.com/article/143414/windows_xp_vs_vista_an_explosion_of_opinion.html>.