# Acknowledgements

I would like to extend special thanks to my parents for giving me the support and love in my whole college life. Without them, my education career would not be finished succesfully. I would like to thank Professor Tamisra Sanyal who took time from his busy schedule to help and answer my many questions about the project. Finally, I would like to thank my family and my friends for their constant support in my education.

Minh H Dang

# Table of Contents

# List of Figures

# Abstract

A Computerized Adaptive Testing System (CATS) is a desktop application for professors who want to manage a test and for students who want to take a test. This application estimates the knowledge level of tester based on the previous answers, retrieves an appropriate question from a question pool stored in a central database, and displays a question to the tester.

This application works in a Local Area Network (LAN) and requires a database server. It can operate in single-user environment as well as in multi-user network environment.

The CATS is developed using C# .NET with MySql database used as the centralized database. The system will use a dynamic link library from a third party to build connection between .Net framework to MySql database server.

# Introduction

A traditional paper-and-pencil exam has a fixed length and the same questions are presented to every examinee without consideration of individual ability. With this kind of testing, mass-administered testing is not easy to handle. The cost in time and money for administration is prohibitive for many examinees. In addition, the score from this type of test depends on the number of questions which the person answers correctly. The more knowledge a person has, the more questions he/she will answer correctly. The conventional examinations have a long and successful history. However, for any single person there are questions, which are too easy or too difficult. Answering easy questions correctly does not tell much about an examinee because most of the examinees can answer those questions correctly. It is the same situation with having a wrong answer for hard questions. The challenge is to be able to know precisely the ability of examinees.

One way to assess a candidate's talent more precisely is through an oral test. In this type of test, the examinee is asked a moderate question at the beginning. If he/she answers correctly, then he/she is presented with a harder question. If he/she answers incorrectly, then he/she is presented with an easier one. By proceeding to give the examinee questions based on his/her response to earlier questions, it is possible to estimate the examinee's proficiency in a short time. Using this strategy, asking many easy or hard questions, which do not help to determine the person's knowledge, is eliminated. This method focuses in on the level of difficulty of the questions which the examinee can answer.

In the 1970s, large testing programs used computers to score tests and process score reports. In the 1980s, computers have been used to administer exams (7). Tests

administered at computer terminals or on personal computers are known as computerized tests. Today computers have more power than before, so it is possible to develop a better way to estimate the examinee's level of competence. Computerized adaptive testing (CAT), the newest use of computerized testing, is a test like the oral test described previously. By knowing how well the person answered previous questions, the computer can choose a set of questions that is suitable for the examinee's ability. In CAT, the low ability examinee will see more easy questions, and the high ability examinee will see more difficult questions. They can both have the same percentage of correct answers but earn different scores.

CAT is similar to an oral exam in which the computer plays the role of a teacher and selects questions by estimating the ability of the examinee based on the previous questions. With CAT, time is saved by eliminating questions which are too easy or too hard. Mass-administered testing can be run with CAT. There are several advantages of CAT (6, pp. 8-9):

- *Administrative*. A CAT version of a test offers four administrative advantages over a traditional version of the same test: 1. Reduced test session length; 2. Flexibility of test session; 3. More standardization; 4. Simplified test revision.

- *Accurate Scoring*. A CAT system reduces errors that occur due to reliability problems with optical scanning equipment used to score a traditional paper-pencil test. In addition, there is a greater possibility for clerical error when tests are hand-scored.

- *Immediate Scoring and Feedback*. The most important benefit of CAT is the results of the testing are known immediately. Getting a score and a pass/fail decision right away for a certification candidate is important.

- *Unbiased Scoring*. Computers score everyone the same way.

- *Measurement Precision*. The measurement precision of the typical test is peaked around the average ability level of the target population. This means that most of the items cluster around medium difficulty with relatively few

easy or difficult items. This CAT strategy focuses on measurement of examinees at the ends of the ability distribution, since a CAT is designed to be appropriate for each examinee's ability level.

- *Test Security*. There are no test booklets to be stolen or marked. There is no way for examinee to copy another's answer because they rarely have the same questions and the order of questions is totally different. All of these allow the item pool to be effective for a longer period of time.

- *New Question Types* (point-and-click, drag and drop). New types of questions improve the ability to measure important skills.

- *Improved Performance-Based Testing*. With software simulations, for example, it is possible to require a certification candidate to demonstrate job skills directly on the test.

- *Reduction in Answering Errors*. Test takers make fewer extraneous errors answering computerized test questions than they do when filling out the small circles on answer forms for paper-and-pencil tests.

- *Enhanced Motivation*. While not well understood, it seem that taking a test on the computer is more interesting and less intimidating than taking the test on paper.

# Description and Intended Use

The Computerized Adaptive Testing System (CATS) is a software application. The CATS has been designed and developed to improve the facility for student to take a test and for professors to manage the test. It is able to infer the ability of a student therefore the high ability student will see more challenging questions.

There are three main elements of this application, the user interface, the main function and the centralized database. The student user interface retrieves questions from a database. Professors can enter and delete questions through their user interface. The main function of the system compares the answer, calculates a level of the next question, and requests the next question from the database. The database can permanently retain all the questions, information and score of students. The information and score of a student can be recalled for viewing and printing a report.

This application works in a Local Area Network (LAN), and it requires a database server. It can operate in single-user environment as well as in multi-user network environment.

# User Profiles

The CATS will have to accommodate a number of users. All users have to interact with the system to do their work such as take test, make test, manage account. Those people can be divied into three groups. These are students, professors, and administrators.

## Students

Students are users who take the test. They need to have a medium level of technical ability such as to know how to use mouse and keyboard. They will need a mouse to choose an answer. A keyboard is used when the students are required to give their credential to log on to the system.

## Professors

Professors have more capability than the students. They are the administrator of their testing database. They can add students to examinee list and change their password. They can create, delete and update a question in a specific pool which is stored in a database. They also have the right to create and to edit test information. In addition, they can monitor the score of students. Professors are expected to have a higher level of technical ability. They also have permission to go into the database, to see their testing database.

## Administrators

Administrators are the final user of the system. Their responsibilities are to manage user account of the system. They can view, create, delete, and edit information of any user belong to the system. All the sensitive information such as user's password,

question pools will be encrypted for security reason.  Even though, administrators have full permission with the database, they can not understand the content of questions and answers.

# Design Protocol

Since there are three types of users using a project and each one has some different features, I have decided to implement this project in different modules. Every module is responsible for providing functionalities for each type of user. This modular design allowed me to manage the complexity of the entire project.

Testing process

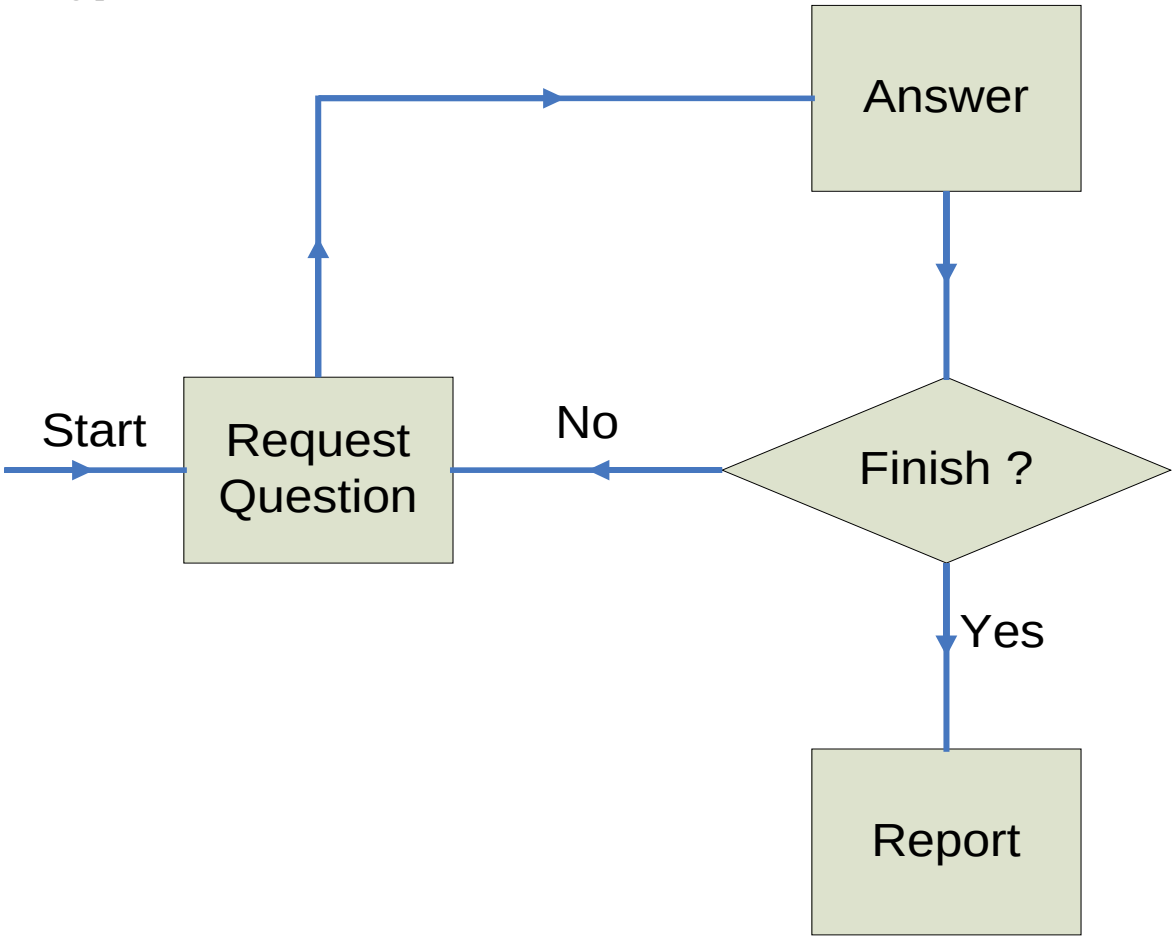


**Figure 1. Show how a testing process works**

A student is asked a moderate question at the beginning. If he/she answers correctly, then he/she is presented with a harder question. If he/she answers incorrectly, then he/she is presented with an easier one. By proceeding to give the student questions

based on his/her response to earlier questions, it is possible to estimate the student's proficiency in a short time. Using this strategy, asking many easy or hard questions, which do not help to determine the student's knowledge, is eliminated. This method focuses on the level of difficulty of the questions which the student can answer.

## Select Question Process

I did some changes in binary search method in order to estimate examinee's skill more accurately. I assume there are only ten different levels, the lowest is 1 and the highest is 10. And the level of first question always is the 5 at the beginning. As I think in the real life, when the range of skill is bigger. For example, the range score in GRE test is 200 to 800 for each section. Therefore the lowest, the highest, and the middle skill can be changed to adapt correctly.

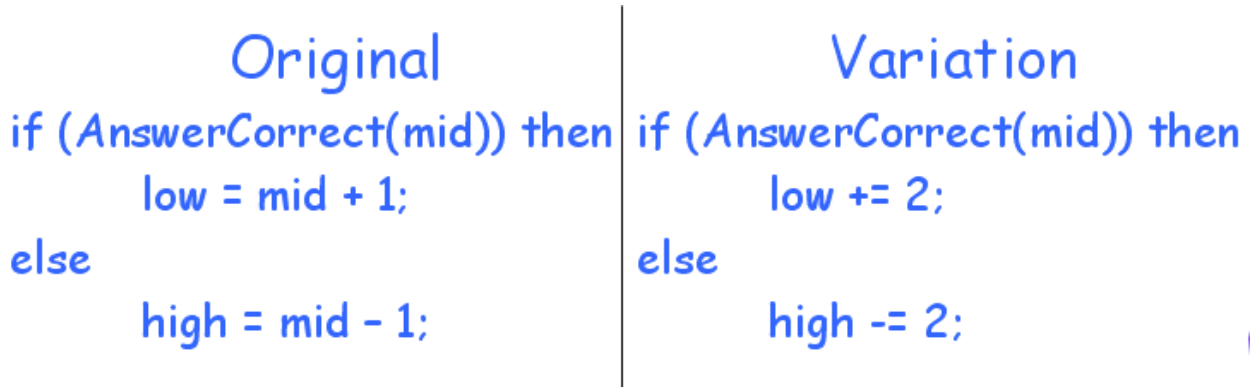


**Figure 2. Original and Variation Binary Search**

Let assume a skill of a student is 8 and he answer the first question incorrectly. If I use the standard binary search to estimate this student's skill, the range will be from the lowest 1 to the highest 4 (high = mid – 1 = 5 - 1). The next question will be asked at a level at the middle of the range. Which means the student can never be back at his actual

skill level. So I change a little in binary search method to estimate more accurately. In this case, the range was asked for a next question reduced from 1 to 8. Therefore a student still can come back to his actual level.

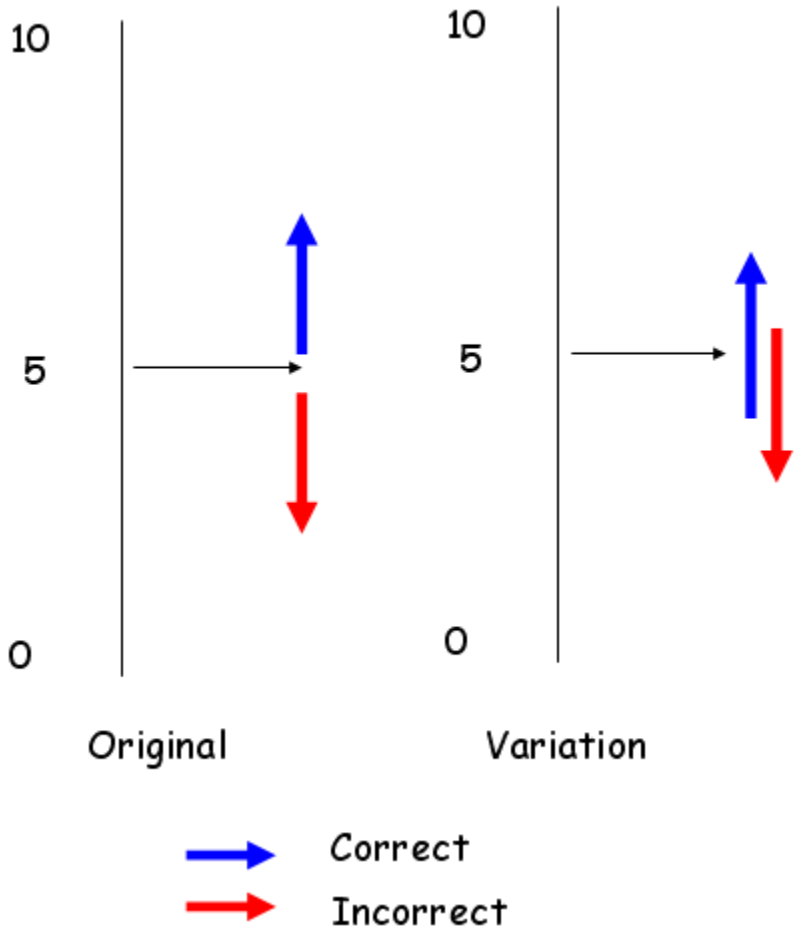


**Figure 3. Range Changing in Original and Variation Binary Search**

## Database Design

My database was designed using an open source software MySql. I created some tables to interact with the project. These tables are used to stored user information, pool information, test information, and record information. These tables are Users, Pools, Tests, StudentTest. In addition, there are many dynamic tables that will be created auomatically when using the project. Most of them are detail information of pools and

tests. These tables are neamed starting with P for pool, or T for test and following with distinct number to identify them.

Because all of the data will be drawn from a database, there must be a reliable database to give the demanding application access to the questions and the user's process data. MySql is one of the trusted databases for the desktop application. The database will contain information about the user's login credentials, personal profiles, pools, and tests. The administrator will be the only one who can carry out the changes to the database.

Users table – Contains all information of users.

| Name | Type | Length | Decimals | Allow Null | |
|---|---|---|---|---|---|
| SSN | char | 9 | 0 | ☐ | 🔑 |
| FirstName | varchar | 20 | 0 | ☐ | |
| LastName | varchar | 20 | 0 | ☐ | |
| Type | tinyint | 4 | 0 | ☐ | |
| Password | varchar | 100 | 0 | ☐ | |

User can not have the same SSN since it is unique for each user. Type of user can be 0 if a user if a student, 1 if a user is a professor, and 2 if a user is an administrator. A login process requires a matching of SSN and Password in Users table.

Pools table – Contains all information of pools. From this table, a professor can only access to his pools.

| Name | Type | Length | Decimals | Allow Null | |
|---|---|---|---|---|---|
| PoolID | char | 5 | 0 | ☐ | 🔑 |
| SSN | char | 10 | 0 | ☐ | |
| Name | varchar | 50 | 0 | ☐ | |

When new pool is created, its information is stored in the Pools table.

Tests table – Contains all information of tests. From this table, a professor can only access to his tests.

| Name | Type | Length | Decimals | Allow Null |
|---|---|---|---|---|
| TestID | char | 5 | 0 | ☐ |
| SSN | char | 10 | 0 | ☐ |
| Name | varchar | 50 | 0 | ☐ |

When new test is created, its information is stored in the Tests table.

StudentTest table – Contains all records of all tests of all students. From this table, a student will be known if a test is already taken and his score of the specific test.

| Name | Type | Length | Decimals | Allow Null |
|---|---|---|---|---|
| RecordID | int | 100 | 0 | ☐ |
| SSN | char | 9 | 0 | ☐ |
| TestID | char | 6 | 0 | ☐ |
| TestName | varchar | 50 | 0 | ☐ |
| Score | varchar | 5 | 0 | ☐ |
| Result | varchar | 65000 | 0 | ☑ |

After taking the test, an application will update Score and Result field.

Pool detail tables – These tables are created when a professor needs a new question pool for his intended use. However all of them have the same structure.

| Name | Type | Length | Decimals | Allow Null |
|---|---|---|---|---|
| QuestionID | int | 10 | 0 | ☐ |
| Question | varchar | 10000 | 0 | ☐ |
| Ans1 | varchar | 10000 | 0 | ☐ |
| Ans2 | varchar | 10000 | 0 | ☐ |
| Ans3 | varchar | 10000 | 0 | ☐ |
| Ans4 | varchar | 10000 | 0 | ☐ |
| Correct | char | 1 | 0 | ☐ |
| Level | varchar | 4 | 0 | ☐ |

Test detail tables – These tables are created when a professor needs a new test for his intended use. All of them have the same structure.

| Name | Type | Length | Decimals | Allow Null |
|---|---|---|---|---|
| PoolID | char | 6 | 0 | ☐ |
| NumberOfQuestion | varchar | 10 | 0 | ☐ |
| Time | varchar | 5 | 0 | ☑ |

It contains information about a test. The number of question is the maximum question will be presented for a student. PoolID shows the pool which question set must be retrieve to ask student.

# Development

## Timeline

*Senior Design I - The Proposal (Fall 2004)*

| **Task** | **Days** | **From** | **To** |
|---|---|---|---|
| Research for area interest | 5 | Sep 30 | Oct 4 |
| Research for a system | 16 | Nov 4 | Nov 23 |
| More research about CAT | 7 | Nov 4 | Nov 12 |
| Final Proposal | 23 | Nov 8 | Nov 30 |
| Power Point Presentation | 2 | Nov 27 | Nov 28 |
| Oral Presentation | 1 | Dec 2 | Dec 2 |

**Table 1:** *Fall Schedule Weekly Details*

*Senior Design II - The Design Freeze(Summer 2005)*

| **Task** | **Days** | **From** | **To** |
|---|---|---|---|
| More research for a system | 7 | June 20 | June 27 |
| Improve C# and SQL Server skill | 7 | June 27 | July 04 |
| Database Design Start | 14 | July 04 | July 18 |

| | | | |
|---|---|---|---|
| User Interface Design | 14 | July 04 | July 18 |
| Final Document | 22 | August 01 | August 22 |
| Power Point Presentation | 2 | August 14 | August 15 |
| Oral Presentation | 1 | August 15 | August 15 |

**Table 2:** *Summer Schedule Weekly Details*

*Senior Design III - The Final Report (Autumn 2006)*

| **Task** | **Days** | **From** | **To** |
|---|---|---|---|
| Building a system | 50 | April 10 | May 30 |
| Testing a system | 50 | April 10 | May 30 |
| Final Document | 23 | May 17 | June 10 |
| Power Point Presentation | 2 | May 30 | June 01 |
| Oral Presentation | 1 | June 01 | June 01 |

**Table 3:** *Autumn Schedule Weekly Details*

Budget

| **Item** | **Actual Cost** | **Project Cost** |
|---|---|---|
| *Software:* | | |
| Microsoft Visual Studio C# .NET | $ 109.00 | Own |
| MySql | 0.00 | Free |
| Adobe Photoshop 7.0 | 163.00 | IT Lab |
| *Hardware:* | | |
| Pentium IV 2.66 Ghz, 512 MB RAM, 60 GB | 1,700.00 | Own |
| **Total** | **$1,972.00** | |

# Deliverables

This project will meet the following deliverables when it is finished.

- MySql database will be used as the centralized database. A database will consist of the tables and features described above. It will be created and tested for reliability of administration. The system will use a dynamic link library from a third party to built connection between .Net framework to MySql database server.
- The system will have the ability to estimate students level while taking a test. It depends on previous answers of students to select the next question.
- User authentication to log on the system
- User can change their credential
- A student user can take a test on a client side.
- A student will receive a score and a report about the test
- A professor create and modify questions on a client
- A professor can also create a student account.
- A system can report class's performances.
- The security of a system is realy important. Only permitted users can logon to the system. The question pools will be protected as well as user's credential. The hash function will be used to encrypt user's password in order to store in the database. The secret key will be used to encrypt question pools.

# Testing Plan

- User’s credential and question pools. The database will be checked to guarantee that all security criteria are met.
- Compatibility of a system with MySql database.
- Testing algorithm to ensure a system precisely of students’ level. Classmates and professors will be testing users of a system. Feedbacks from them will be used to improve the algorithm.

# Proof of Design

SharingMethod class – This class contains many methods which is used in many different places of the system. Most of methods in this class are static and they are called whenever the system need. This class helped me to reduce a coding and testing time.

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Text;
using MySql.Data.MySqlClient;

namespace CATS
{
	/// <summary>
	/// Summary description for SharingMethods.
	/// </summary>
	public class SharingMethods
	{
		private static MySqlConnection conn;

		public SharingMethods()
		{

		}

		public static string encodePassword(string
originalPassword)
		{
			//Declarations
			Byte[] originalBytes;
			Byte[] encodedBytes;
			MD5 md5;

			//Instantiate MD5CryptoServiceProvider, get bytes for
original password and compute hash (encoded password)
			md5 = new MD5CryptoServiceProvider();
			originalBytes =
System.Text.Encoding.Default.GetBytes(originalPassword);
			encodedBytes = md5.ComputeHash(originalBytes);

			//Convert encoded bytes back to a 'readable' string
			return BitConverter.ToString(encodedBytes);
		}

		public static int getTypeUser(string SSN, string password)
		{
			createConnection();
```

```
                string strSelect = "SELECT Type FROM Users WHERE SSN
= '" + SSN + "' AND password = '" + password + "'";

                MySqlCommand cmd = new MySqlCommand(strSelect, conn);
                MySqlDataReader rdr = cmd.ExecuteReader();


                int type = -1;
                if (rdr.HasRows)
                {
                      rdr.Read();
                      type = int.Parse(rdr[0] + "");

                }

                rdr.Close();
                closeConnection();
                return type;
            }

            private static void createConnection()
            {
                string DataSource = "                    ";
                string Database = "      ";
                string UserID = "        ";
                string Password = "       ";
                string strConn = "Data Source= " + DataSource +
";Database=" + Database + ";User ID=" + UserID + ";Password=" +
Password;
                conn = new MySqlConnection(strConn);
                conn.Open();
            }

        public static MySqlConnection getConnection()
        {
            string DataSource = "                    ";
            string Database = "      ";
            string UserID = "        ";
            string Password = "       ";
            string strConn = "Data Source= " + DataSource +
";Database=" + Database + ";User ID=" + UserID + ";Password=" +
Password;
            conn = new MySqlConnection(strConn);
            return conn;
        }

            private static void closeConnection()
            {
                conn.Close();
                conn.Dispose();
            }

            public static string[] searchUser(string SSN)
            {
                createConnection();
                string strSearch = "SELECT * FROM Users WHERE SSN =
'" + SSN + "'";
```

```
                List<string> ret = new List<string>();

                MySqlCommand cmd = new MySqlCommand(strSearch, conn);
                MySqlDataReader rdr = cmd.ExecuteReader();

            if (rdr.HasRows)
            {
                rdr.Read();
                ret.Add(rdr[0].ToString());
                ret.Add(rdr[1].ToString());
                ret.Add(rdr[2].ToString());
                ret.Add(rdr[3].ToString());
                ret.Add(rdr[4].ToString());
            }

                closeConnection();

            return ret.ToArray();
            }

        public static string addUser(string[] infor)
        {
            createConnection();
            string strAdd = "INSERT INTO Users(SSN, FirstName,
LastName, Type, Password) VALUES ('" + infor[0] + "','" + infor[1] +
"','" + infor[2] + "'," + infor[3] + ",'" + infor[4] + "')";

            string ret = "";
            try
            {
                MySqlCommand cmd = new MySqlCommand(strAdd, conn);
                cmd.ExecuteReader();
                ret = "Add successfully";
            }
            catch
            {
                ret = "User is already exist!";
            }
            closeConnection();

            return ret;
        }

        public static string deleteUser(string SSN)
        {
            createConnection();
            string strDelete = "DELETE FROM Users WHERE SSN ='" + SSN +
"'";

            string ret = "";
            try
            {
                MySqlCommand cmd = new MySqlCommand(strDelete, conn);
                cmd.ExecuteReader();
                ret = "Delete successfully";
            }
```

```
            catch
            {
                ret = "Can not delete!";
            }
            closeConnection();

            return ret;
        }

        public static string updateUser(string[] infor)
        {
            createConnection();
            string strUpdate = "UPDATE Users SET FirstName = '" +
infor[1] + "', LastName = '" + infor[2] + "', Type = '" + infor[3] +
"', Password = '" + infor[4] + "' WHERE SSN = '" + infor[0] + "'";

            string ret = "";
            try
            {
                MySqlCommand cmd = new MySqlCommand(strUpdate, conn);
                cmd.ExecuteReader();
                ret = "Update successfully";
            }
            catch
            {
                ret = "Update failed!";
            }
            closeConnection();

            return ret;
        }
    }
}
```

## Login form


**Figure 4. Login Screen**

This is a first screen when an application is launched. After login successfully, a user will be presented to different screen defending on his type of user.

```
private void btnOk_Click(object sender, System.EventArgs e)
{
	string SSN = txtSSN.Text;
	string password = txtPassword.Text;
	if (SSN != "" && password != "")
	{
		password = SharingMethods.encodePassword(password);


		int r = SharingMethods.getTypeUser(SSN, password);
		txtSSN.Clear();
		txtPassword.Clear();

		switch (r)
		{
			case -1: //SSN or password does not match
			MessageBox.Show("Invalid SSN or password.\nPlease try
again.");
			count++;
			if (count == 3) this.Close();
			break;
			case 0: //User is a student
			Student(SSN);
			break;
			case 1: //User is a professor
			Professor(SSN);
			break;
			case 2: //User is a system admin
			Admin();
			break;
		}
	}
}

private void Student(string SSN)
{
	frmStudent studentForm = new frmStudent(SSN);
	this.Visible = false;
	if (studentForm.ShowDialog() == DialogResult.Cancel)
	{
		studentForm.Dispose();
		this.Close();
	}
}

private void Professor(string SSN)
{
	frmProfessor professorForm = new frmProfessor(SSN);
	this.Visible = false;
	if (professorForm.ShowDialog() == DialogResult.Cancel)
	{
		professorForm.Dispose();
		this.Close();
	}
}

private void Admin()
```

```
{
	frmAdmin adminForm = new frmAdmin();
	this.Visible = false;
	if (adminForm.ShowDialog() == DialogResult.Cancel)
	{
		adminForm.Dispose();
		this.Close();
	}
}
```

Administrator form

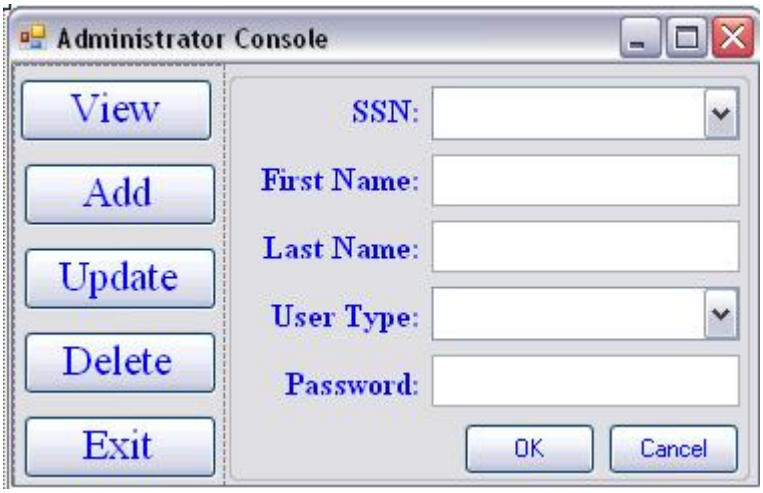


**Figure 5. Administrator Screen**

This is a screen when an admin successfully login. From here, he/she can manage an user account such as View, Add, Update, Delete by clicking correctsponding button.

Professor form

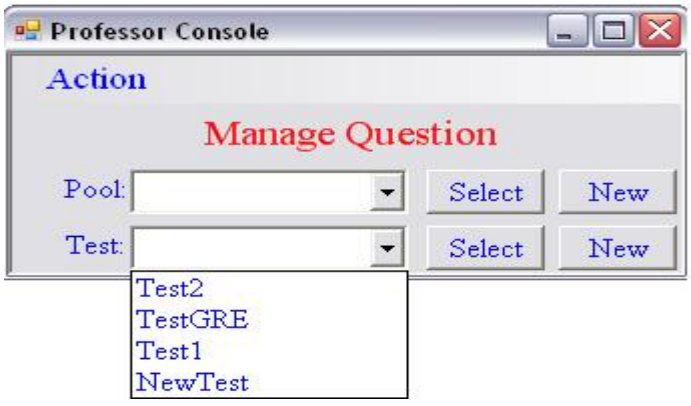


**Figure 6. Professor Screen**

This is a screen when a professor successfully login. From here, he/she can manage question pools or tests such as Edit, Add new by clicking correctsponding button. He/she can alos see the result of a test from Action menu.

## Student form

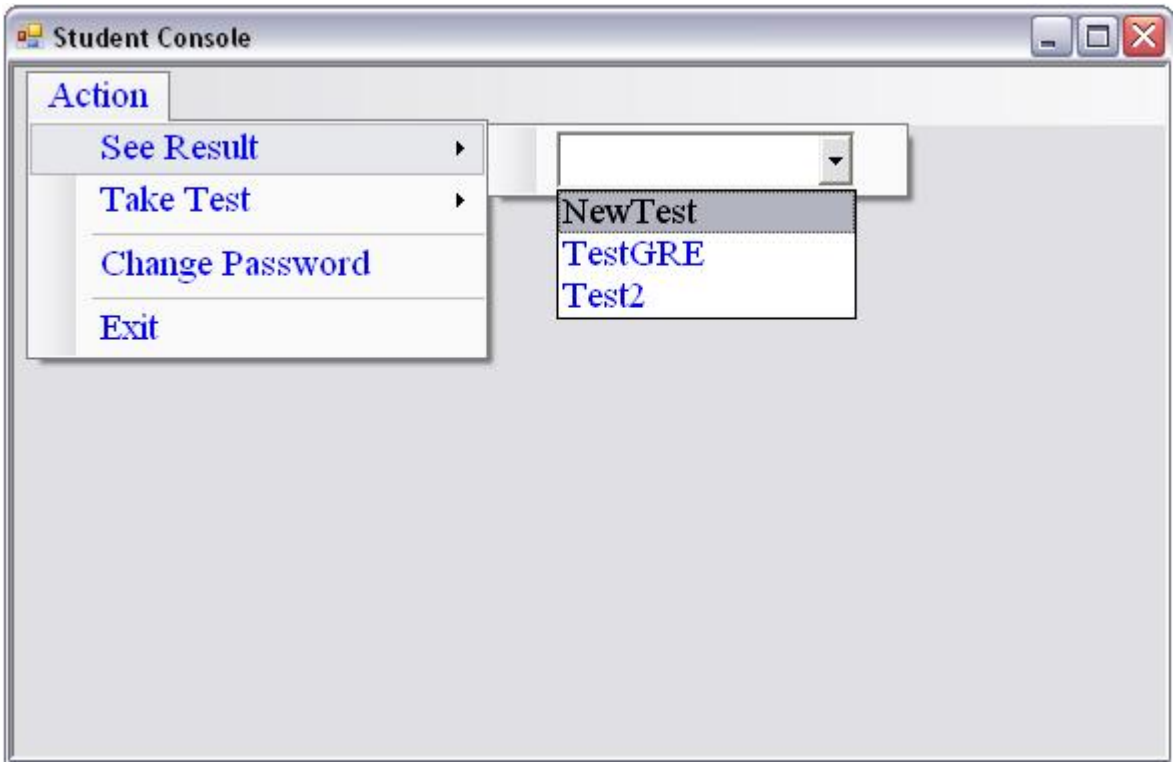


**Figure 7. Student Screen**

This is a screen when a student successfully login. From here, he/she can take tests which he/she are assigned or see result of the taken tests Action menu. Once student decide to take test, he/she will be presented the following screen.

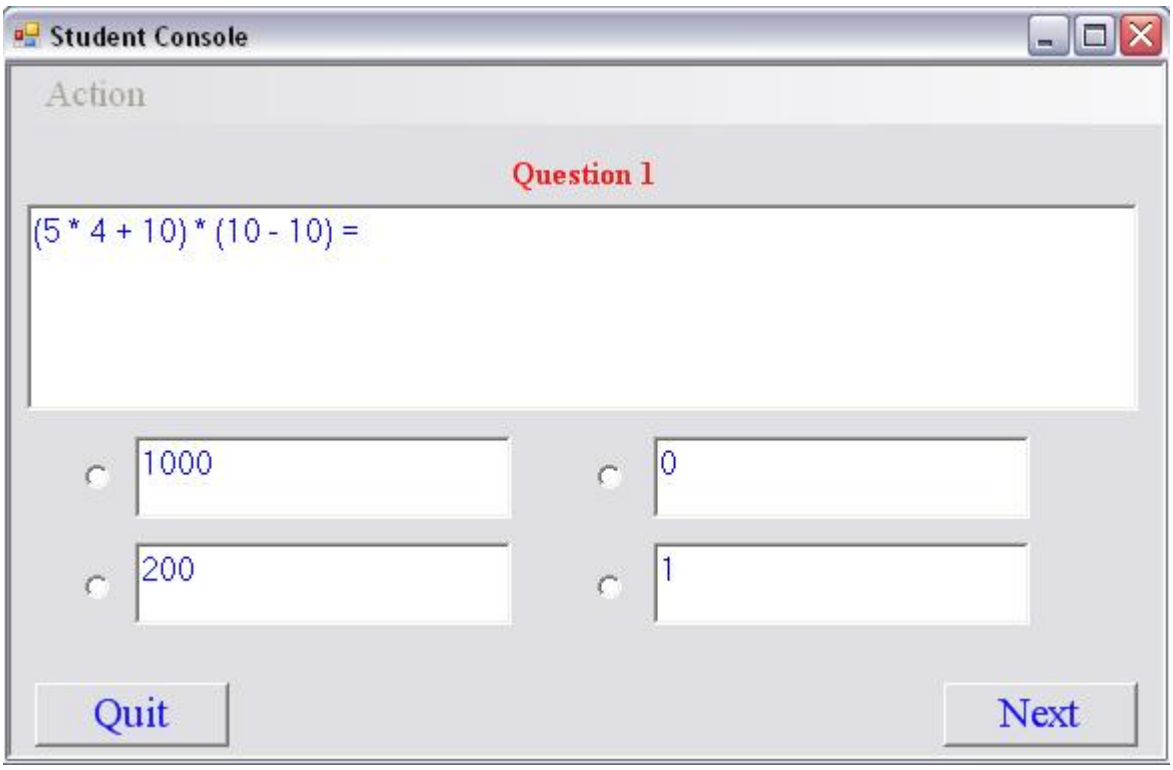


**Figure 8. Student Testing Screen**

The question will be presented in the first textbox then followed by four answer options. Base on the answer of a stundent, a computer tries to estimate the student skill and then selects the next appropriate question.

```
private void btnAnswer_Click(object sender, EventArgs e)
{
    if (radioA.Checked || radioB.Checked || radioC.Checked ||
radioD.Checked)
    {
                string right = "";
                if (radioA.Checked) right = "1";
                if (radioB.Checked) right = "2";
                if (radioC.Checked) right = "3";
                if (radioD.Checked) right = "4";
                string ok = "";
                if (!right.Equals(correct))
                {
                    ok = "W";
                    hi -= 2;
                }
                else
                {
                    ok = "R";
                    lo += 2;
```

```
                    min = Math.Min(min, score);
                    max = Math.Max(max, score);
                }

                set += questionID + "," + right + "," + ok + " ";

                getQuestion();
            }
            else
            {
                MessageBox.Show("You must select an answer before going
to the next question");
            }
        }
```

# Conclusion and Recommendations

I choose to do a Computerized Adaptive Testing System based on the need I saw in different testing softwares. The project helped me develop necessary skills in the field of Information Engineering Technology. I also developed time management skill that is necessary in any career.

Even though this project is complete and fully functional, it still require much more work on user interface and algorithm for selecting question. It also need to work in any kind of database such as MsSql, Microsoft Access. For example, a user want to practice at home and he has only Microsoft Access.

Overall, this project was challenging and make me realize the importance of communication with other people such as my advisor, my friends who helped and commented the project. The technical aspect of creating this project also challenged me. The knowledge that I gained thoughout this project will help me a lot in my future career goals.

# References

1. Dunkel, Patricia A. “Considerations in Developing and Using Computer-Adaptive Tests to Assess Second Language Proficiency”. http://www.cal.org/resources/digest/cat.html. October, 1999.

2. Fairtest, “Computerized Testing: More Questions Than Answers. Fair Test Fact Sheet”. http://www.fairtest.org/facts/computer.htm. 1998

3. Grist, Susan. “Computerized Adaptive Tests. ERIC Digest No. 107”. http://www.ericdigests.org/pre-9213/tests.htm. 2000

4. Linden, Wim J. van der and Cees A. W. Glas. *Computerized adaptive testing: theory and practice***,** Dordrecht, Boston: Kluwer Academic, 2000

5. Rudner, Lawrence M. “An On-line, Interactive, Computer Adaptive Testing Tutorial”. http://edres.org/scripts/cat/catdemo.htm. October, 1998

6. Sands, A William and Brian K. Waters and James R. McBride. *Computerized adaptive testing: from inquiry to operation.* Washington, DC: American Psychological Association, 1997

7. Wainer, Howard. *Computerized adaptive testing: a primer.* Mahwah, N.J: Lawrence Erlbaum Associates, 2000