Designing Security System using MyKad

by

Mahfuza Mohamed Sirat

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

JUNE 2008

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Designing A Security System Using MyKad

by

Mahfuza Mohamed Sirat

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONICS ENGINEERING)

Approved by,

(Assoc. Prof. Dr Mohammad Awan)

Final Year Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

June 2008

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

MAHFUZA MOHAMED SIRAT

Acknowledgements

Non-stop gratitude, credits and thanks should be given to the following people that been mention below. Without the proper help and guidance of these people, the Final Year Project 1 might not have achieved its desired objectives.

Thank to **Assoc. Prof. Dr Mohammad Awan (Associate Professor, UTP)** for giving proper guidance and supervision throughout the whole Final Year Project, not forgetting in contributing towards development of personal values as a student.

I would like to thank **Ms. Salina Bt. Mohamad and coordinator members (Lecturers, UTP)** for contributing knowledge while in seminar/talk section. Special acknowledge to Electrical & Electronic Department for the support and Electrical & Electronic Technicians, Miss Siti Hawa for the guidance and advice.

Finally, thank to my beloved fellow colleagues for their help through the completion of this project.

ABSTRACT

This report describes the designing of a Security System based on MyKad for Malaysian boarding school use. The system can be used at school main gate that required high security in tracing people flow in and out. The current system that has been used by the Malaysian boarding school is 'Outing Card' and 'Guess Book'. The systems have several problems for a security staff to trace the previous data. This problem can be more critical if a student is not honest to apply for permission because anyone can sign the 'Outing Card'. The objective of this project is to develop a system that can detect and read MyKad and identify the status of the owner whether she/he can leave or enter the school area. This system can verify whether the MyKad owner is a student, a warden or a visitor. The system can be used as a registration record for visitor status. This project carried out using Visual Basic programming language and Borland C++ language. The system will login by reading the MyKad data, comparing the data with the stored database in server and displaying the status window.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ATR – Answer to Reset

APDU - Application Protocol Data Unit

CPU - Central Processing Unit

RAM - Random Access Memory

COS - Chip Operating System

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Security system is one of the important things in an organization, state or country. In primary or secondary school, student card or staff card is one of the security devices to identify staff, teachers or students. In a boarding school, 'Outing card' and 'Going home card' is a solution to define status of student with or without permission to leave the school area.

At a school main gate is one of the places that all students need to display their matrix cards or outing cards. What going to happen if some students try to duplicate warden signature and leave the school area without permission? MyKad security system is one of the solutions to improve the security system at school.

The common method that have been used by a organization such as boarding school, factory or personal building is by leaving their MyKad at the guard house while entering the premise. This method is not practical anymore because Malaysian MyKad contains other personal document such as driving license, passport information, health information, MEPS Cash, Touch 'n Go, ATM card and PKI. Using the proposed system, a visitor can enter without the need to leave MyKad at the guard house.

1.2 Problem Statement

Nowadays, the current system used by the boarding school is a piece of paper which contains student name, IC number and a warden permission. The security people will record manually the time of going out and in. For visitor records, a visitor needs to fill the log book before entering and after leaving the school area. This method causes a number of problems as follows:

- The data that been recorded can be less accurate and not secure.
- More time will be needed during the registration session at the guard house.
- Signing the outing card by the warden or teacher not a practical work if the number of student is large.
- Some students can imitate the warden signature without notice.

1.2 Objectives Of The Project

A software system is written to:

- Read the MyKad data and display the data for information.
- Create database of student and staff status.
- Incorporate MyKad data with the server database tables.
- Expose the author to smartcard and system application technologies.

1.4 Scope of Study

The scope of study for this project is to develop a software system that can recognize and read smartcard data and use it with the database. The system will be created using Visual Studio 2005 (Visual Basic). A database of students, visitor and staff were created to test the generation of the whole system.

CHAPTER 2

LITERATURE REVIEW AND THEORY

2.1 Smartcard

Smartcard are secure, compact and intelligent data carriers. It is widely regarded as specialized computers capable of processing, storing and safeguarding thousands of bytes of data. Similar in size and shape to plastics credits cards, smartcards with electrical contacts have a thin metallic plate just above centreline on one side of the card. Beneath this dime-sized plate is an integrated circuit (IC) chip containing a central processing unit (CPU), random access memory (RAM) and non-volatile data storage. Data stored in the smartcard's microchip can be accessed only through the chip operating system (COS), providing a high level of data security. This security takes the form of passwords that allow a user to access parts of the IC chip's memory or encryption/decryption measures, which translate the bytes stored in memory into information [1].

2.2 Current School Security System

The current school security system is using a piece outing card and also a guest book. Each of the school students need to have their own outing card. Any information of the student and their parent is stated inside that card. This card also contains the outing information each time student going out or in the school area.

On duty warden gives the outing permission by filling the outing card and signs the permission column. Figure 1 is the example of outing card that been use by school in Malaysia.

Figure 1: Outing card

For the guest registration, the guest book has been used. Figure 2 shows the example of guest book that have been used by Malaysian boarding school. The records from this book have a limitation of use. Sometime, due to the different types of hand written will make the security staff not able to trace the data.

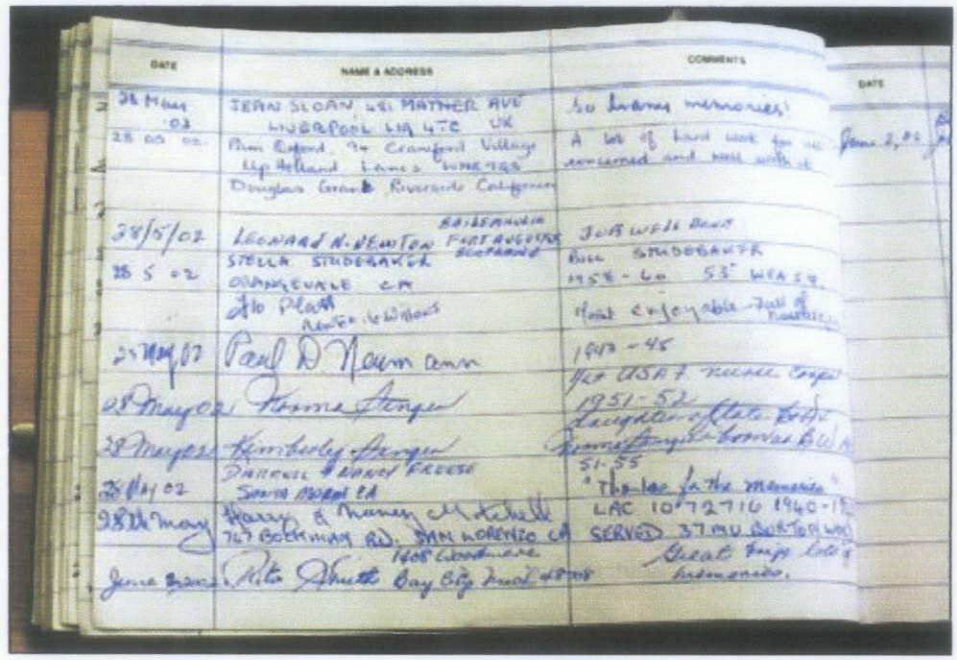

Figure 2: Guest book

2.3 MyKad

The MyKad is an Identity Card issued by the Malaysian Government that stores the user's information in a Smart Chip. This card was designed to be a multipurpose card that stores other user applications such as user's driver's license, health card and passport information [2].

2.4 Software Development Model

The software development model used in this project is the increment model. The incremental model is an intuitive approach to the waterfall model. Multiple development cycles take place here, making the life cycle a multi-waterfall cycle. Cycles are divided up into smaller, more easily managed iterations. Each iteration passes through the requirements, design, implementation and testing phases [7].

A working version of software is produced during the first iteration, so the working software will be produced at the early state during the software life cycle. Subsequent iterations build on the initial software produced during the first iteration.

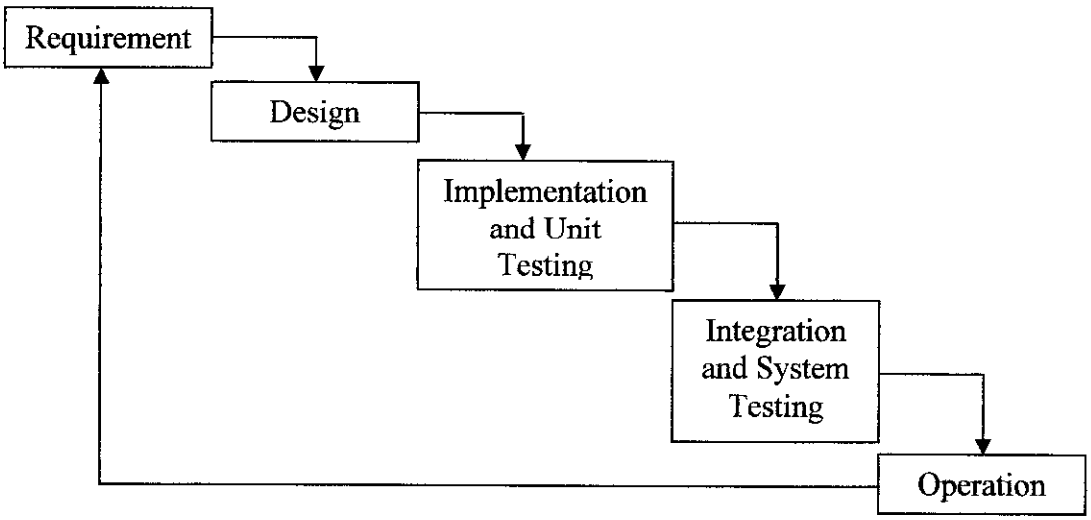

Figure 3: Increment Model

The advantages of this model are:

- This model can generates working software quickly and early during the software life cycle.
- Increment model is more flexible and less costly in order to change scope and requirements during the progress.
- With this model, the design is easier to test and debug during a smaller iteration.
- This model is easier to manage risk because risky pieces are identified and handled during its iteration.
- Each iterations of the model flow is an easily managed milestone.

2.5 Answer To Reset (ATR)

Every smartcard is required to give an answer when it is being reset. This reset occurs when the card is being powered-up by the smartcard terminal, usually upon insertion, but this reset can often also be done explicitly by giving a specific command to the reader. In the reset stage the card and the reader get introduced to one another and the foundation for the communication session is being made.

The structure of an ATR consists of several blocks, the Initial Character, Format Character, Interface Characters, Historical Bytes and the Check Character. The Initial Character, called TS, is used for defining the Bit-Coding, logical '1' is transmitted as Low (L) or High (H) and for recognition of the factor relevant for bit-length, used by the card.

Interface Characters and Historical Characters are optional, the Format Character (called T0) indicates, which characters are transmitted.

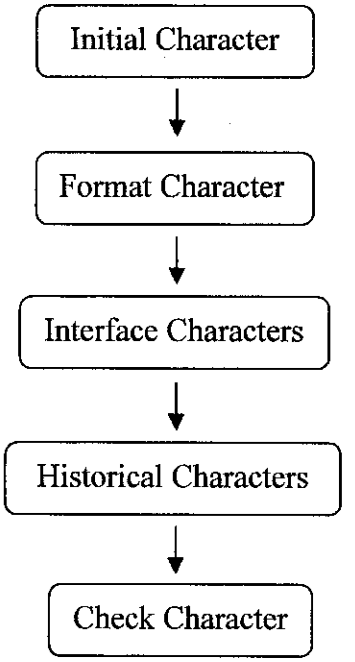

Figure 4: ATR schema

The ATR number for MyKad: 3B6700007320006C689000

2.6 Application Protocol Data Unit (APDU)

Application Protocol Data Unit is a packet of data exchanged between two application programs across a network [3]. This APDU has been specified in ISO 7816-4. There are two types of APDU: Command APDU, Response APDU.

The Command APDU is the command from the card reader or a device to the smartcard. There are mandatory header and optional body in the command APDU.

Table 1: Command Bytes

CLA	INS	P1	P2	Lc	DATA	Le

CLA = Class of Instruction

INS = Instruction

P1, P2 = Parameters 1 and 2

Lc = Length of the command data

DATA = Command data

Le = Length of the expected response

The Response APDU is the response of the smartcard to the device. Response APDU contain of optional body and mandatory trailer as shown in Table 2.

Table 2: Response byte

DATA	SW1	SW2

DATA = Response data

SW1, SW2 = Status Word 1 and 2

The APDU command and response can be send to the MyKad card by using the Rockey smartcard reader. The instruction, address and byte length for Mykad APDU are as shown in the Appendix A.

2.6.1 Reading Sections

Instead of viewing a particular command to read "Name", another command to read "IC no.", it is better to interpret a sequence of three commands (Set Length, Select Info, Read Info) as reading a section or reading the whole file. The fact is 'Name', 'IC no.' or 'Address' are stored in fixed-length fields and concatenated together to form files.

For example:

"Name" stored in file jpn-1-1, offset 0x00E9, length 0x28

"IC no." stored in file jpn-1-1, offset 0x0111, length 0x0D

To read name data of the owner, file jpn-1-1 need to be read with the offset 0x00E9 and length 0x28. The same procedure can be applied to read the 'IC no.', but now with different offset and also length (offset 0x0111, length 0x0D). To simplify the coding, both data can be read in one command like read jpn-1-1, with offset 0x00E9 and length 0x35.

The example below shows the command to read the data and response given by the smartcard. The last response given by the smartcard is the 'Name' and 'IC. No' data in ASCII code. APPENDIX B shows the standard ASCII code.

Example: read jpn-1-1, offset 0x00E9, length 0x28.
Reader: C8 32 00 00 05 08 00 00 28 00
Card: 91 08
Reader: CC 00 00 00 08 01 00 01 00 E9 00 28 00
Card: 94 28
Reader: CC 06 00 00 28
Card: 4D 59 20 4E 41 4D 45 20 20 20 20... 90 00

2.6.2 Application

The first two commands, 'Select Application' and 'Get Response', are used to select petai JPN, JPJ or IMM application. After the smartcard has been reset, select application is used to define the application that need to be read.
Example:
Reader: 00 A4 04 00 0A A0 00 00 00 74 4A 50 4E 00 10
Card: 61 05
Reader: 00 C0 00 00 05
Card: 6F 03 82 01 38 90 00

"00 A4 04 00 0A" is the 'CLA INS P1 P2 P3' for 'Select Application'. The data part of the APDU consists of 10 bytes: "A0 00 00 00 74 4A 50 4E 00 10". The "A0 00 00 00 74" and "00 10" parts are constant. "4A 50 4E" represents "JPN". Change to "JPJ" or "IMM" for those applications. "00 C0 00 00 05" is the "CLA INS P1 P2 P3" for "Get Response".

2.7 MySQL Server Database

MySQL server been managed by the MySQL Administrator tool. This tool can be used to manage users, change server configuration, manage server databases, and monitor server status [4].

Adding a new user by MySQL Administrator is shown in Figure 5 and the username and password need to be specified. Both information will be used in writing the system codes.

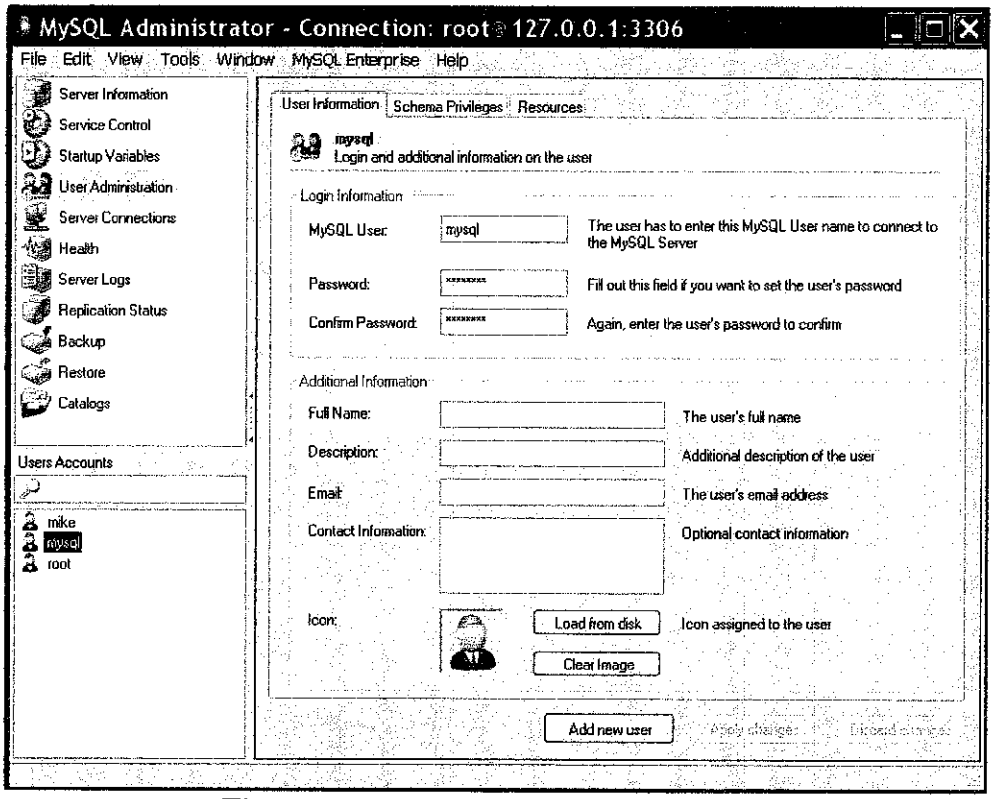

Figure 5: MySQL Administrator

2.8 MySQL Connector/ (Open Database Connectivity) ODBC

MySQL Connector allows the MySQL database server to be connected using the ODBC database API on all platforms. Connector/NET is written in C++ and it is completely managed code. The setup for the ODBC is using the Data Sources (ODBC) Administrator. Figure 6 shows the Data Sources (ODBC) Administrator window and Figure 7 shows the system connector setting.

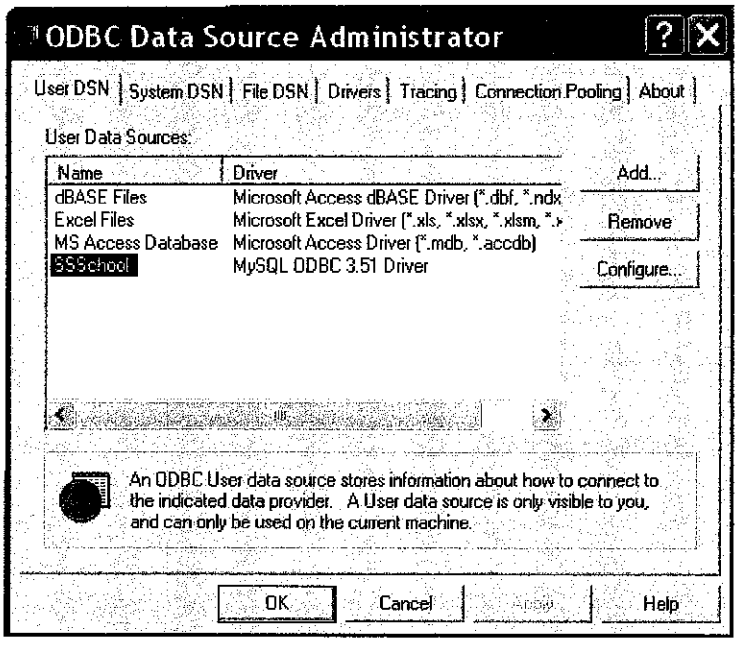

Figure 6: Data Sources (ODBC) Administrator window

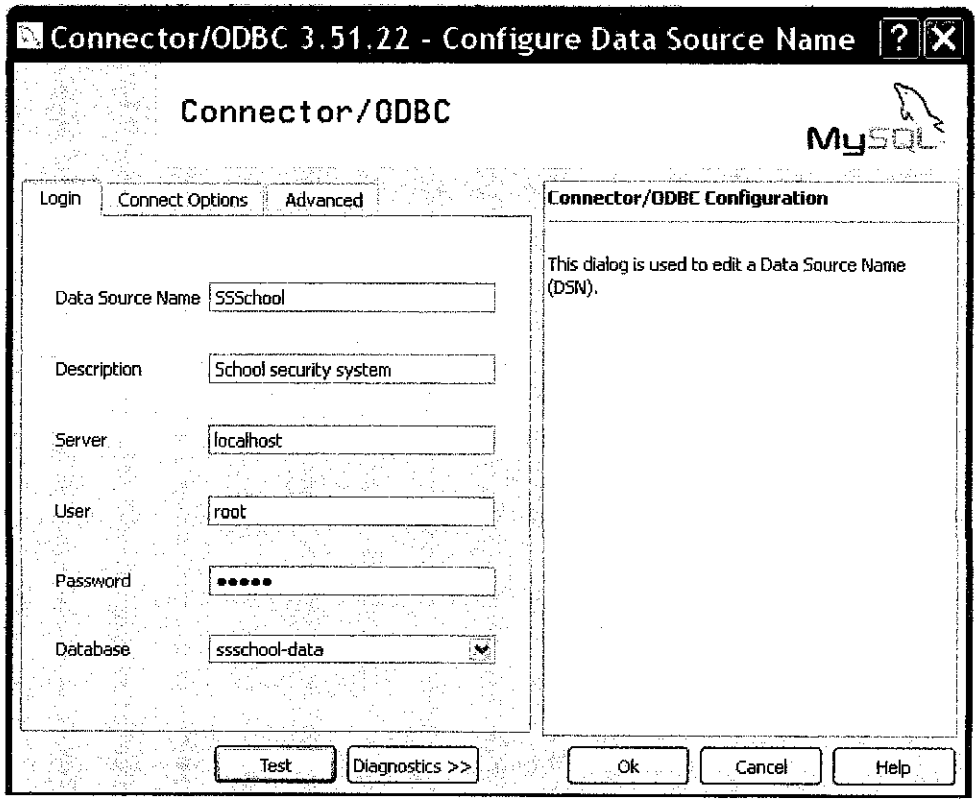

Figure 7: System connection setting

2.9 DataSet Objects

In order to call the data in the database, dataset Objects has been used. A dataset object holds a copy of the data in memory, disconnected from the data source. The dataset object can be filled with data from many sources, including SQL Server database, mane other database formats such as Access, a simple text file, a spreadsheet and also an array or a collection [7].

A DataSet object can hold one or more DataTable objects. A table can be viewed like a spreadsheet in other words with rows and columns. Each row in the table represents the data for one item, person or transaction. Each column in a table is used to store a different of data, such as an account number, a name, an address or ID number in the author case.

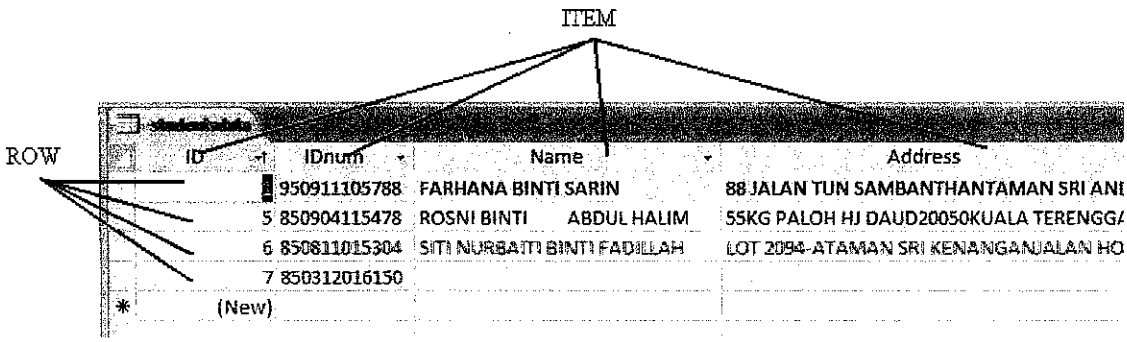

Figure 8: Example of row and column in the Dataset

2.10 American Standard Code for Information Interchange (ASCII)

ASCII is a standard character-coding scheme used by most computers to display letters, digits and special characters [5]. The ASCII tables are included in the Appendix B.

In ASCII, every letter, number, and punctuation symbol has a corresponding number, or ASCII code. For example, the character for the number 1 has the code 49, capital letter A has the code 65, and a blank space has the code 32. This encoding system not only lets a computer store a document as a series of numbers, but also lets it share such documents with other computers that use the ASCII system.

CHAPTER 3

METHODOLOGY

3.1 Design Flow

Flowchart for the methodology process is shown in Figure 9. The flow is started with MyKad detection by the MyKad SDK. Next, the connection between smartcard with security system is done by ATR transaction code. Then, MyKad data can be read and the data will be used to compare with the database table.

Only MyKad name and IC number have been read before the comparison process. The software system will checked the MyKad IC number with the database data. Then, the software system identified the status of the MyKad owner. There are three different types of status that the software system can identify. There are warden, student and visitor status. Each status can access different application after the identified stage.

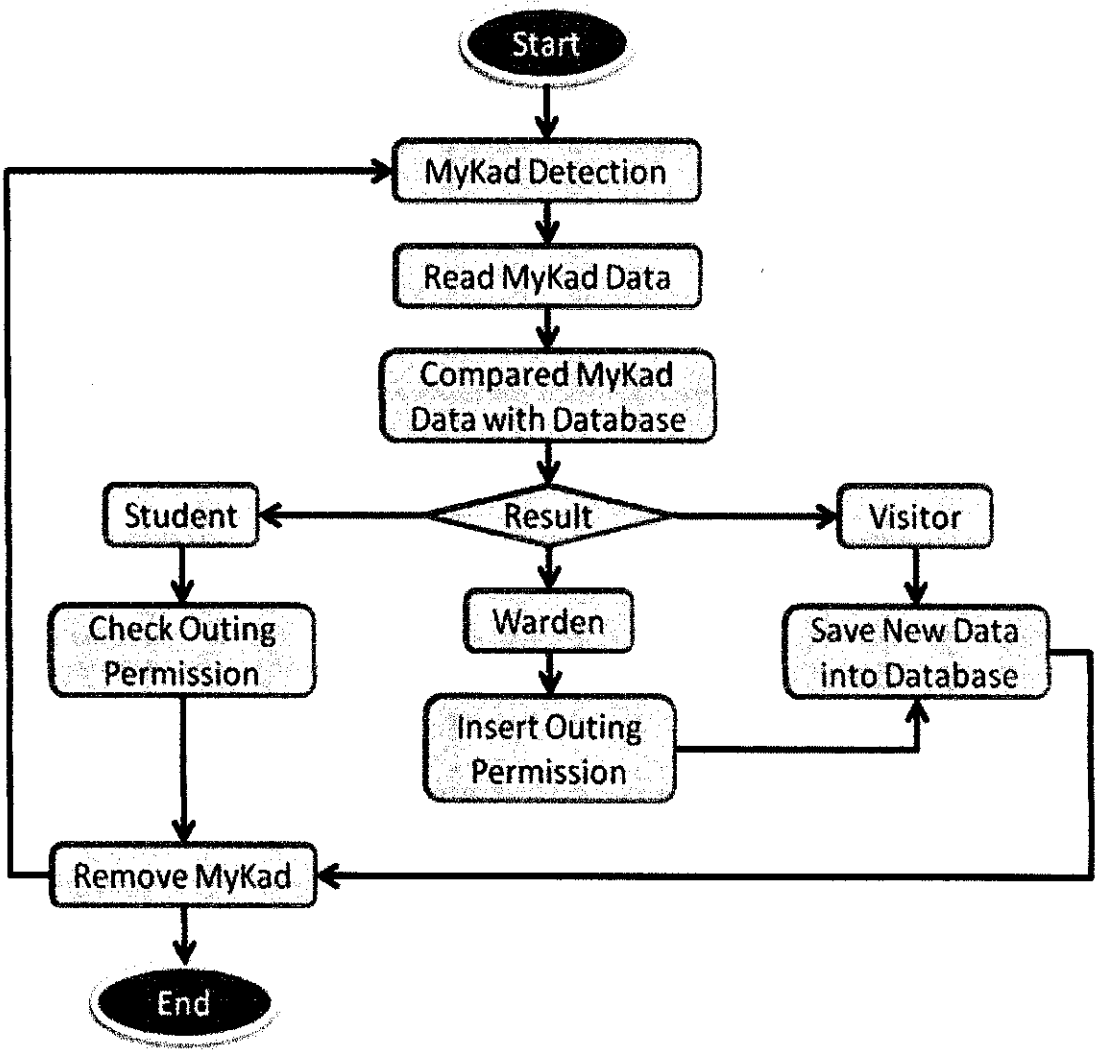

Figure 9: Flowchart for methodology flow

3.2 APDU command

The flow that been used in order to read the smartcard is shown in Figure 10. The figure shows the flow of APDU command before getting the data (read the data).

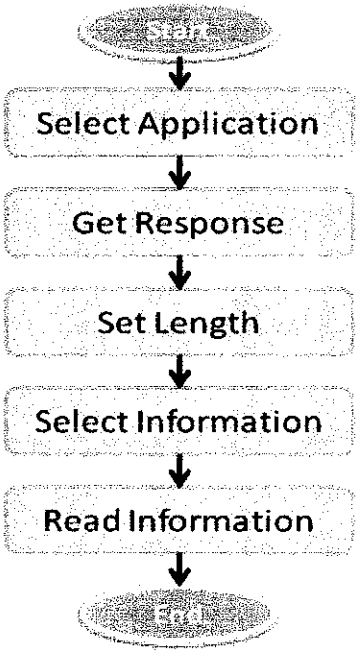

Figure 10: Flow diagram for APDU command

The APDU command has been tested using the WatchData CradTools2.9. This software will send the APDU command to the smartcard and display the smartcard response in standard ASCII code. Figure 11 shows the WatchData CradTools2.9 window and Figure 12 shows the smartcard response after WatchData gave the APDU command.

For School Security System development, the APDU command has been send by the C++ coding and the data being called by the Visual basic programming.

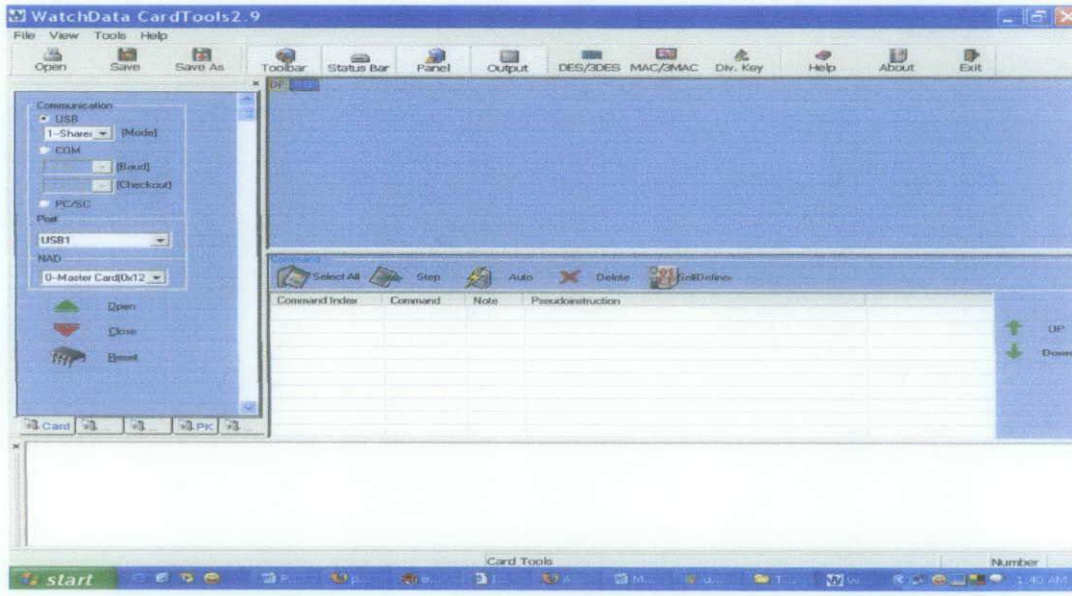

Figure 11: WatchData window

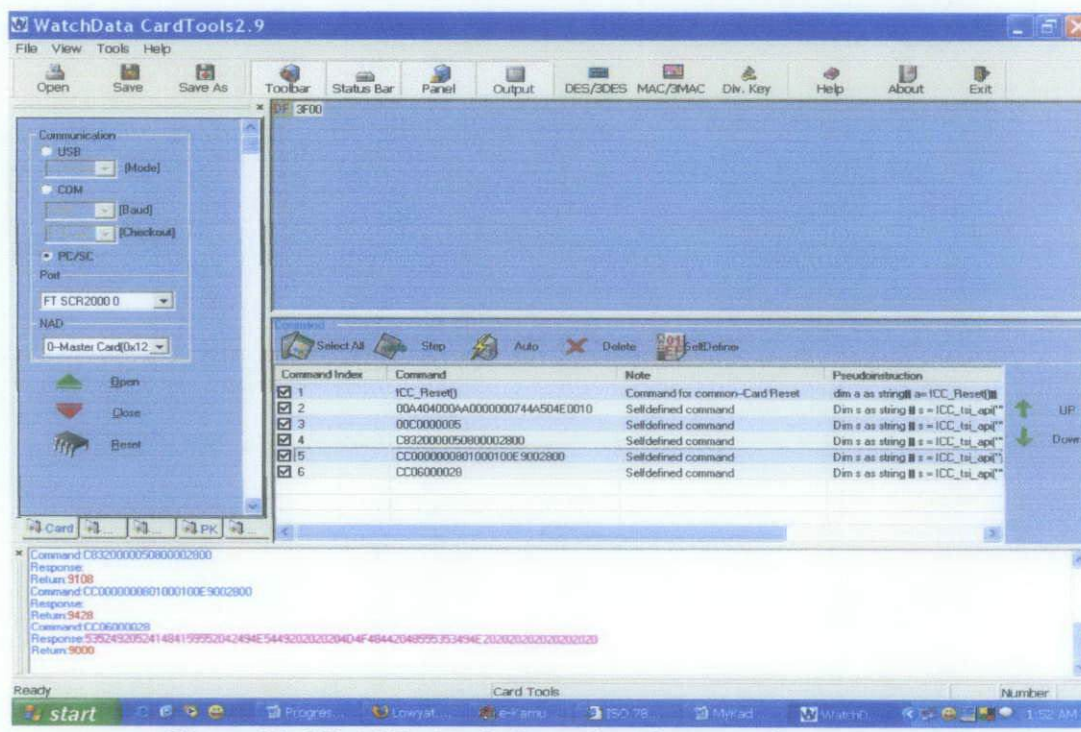

Figure 12: WatchData window after the operation

The response given by the smartcard will be discussed in the Result and Discussion in Chapter 4.

3.3 MyKad.vb (Automatic Detection)

From the flow diagram shown in Figure 13, it show that the coding begin by checking whether the smartcard has been inserted into the device or not. If the smartcard is not inserted in the reader, the message box will display "No Card Inserted".

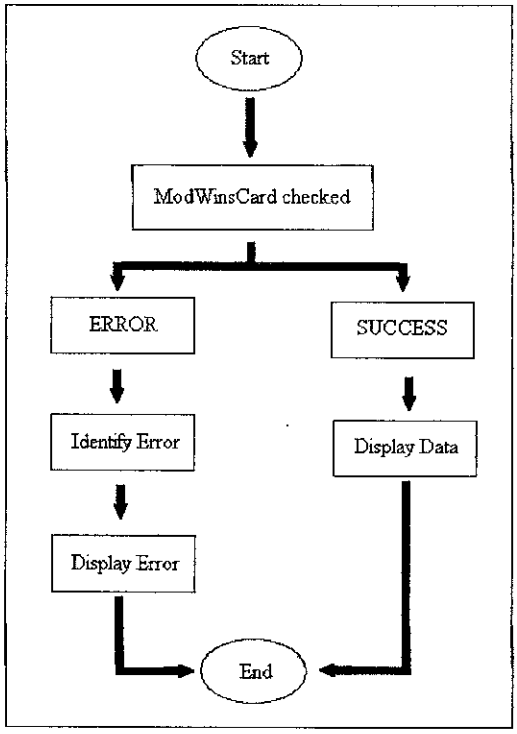

Figure 13: Flow diagram for automatic detection coding

After the card been inserted into the card reader, ModWinsCard.dll or the library file will check the condition of the smartcard. If the smartcard has an error, this library will identify the error and it will be displayed on the message box. Full coding for the automatic detection is given in Appendix C. This code is written in Visual Basic programming language.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Successful Read MyKad Data And Display The Data

This system can read the name, I/C number, address and gender of the card owner. After the reading part is completed, the system will check whether the data read are already stored in the database or not. Then the system will display the status of that card owner. If the data in the card is not in the student database, the system will store this new data into the visitor database. Figure 14 shows the window after reading the ID number and ID name.

Figure 14: Successful reading the MyKad data

4.2 APDU Command

After giving several commands to the MyKad, the response is shown in Table 2. The last response shown in Figure 15 is the name of the MyKad owner.

Table 3: Response by the smartcard

53	52	49	20	52	41	48	41	59	55	20		
S	R	I		R	A	H	A	Y	U			
42	49	4E	54	49	20	20	20	20	4D	4F	48	44
B	I	N	T	I					M	O	H	D
20	48	55	53	53	49	4E	20	20	20	20	20	20
	H	U	S	S	I	N						

```
Response:3B6700007320006C689000
Return:9000
Command:00A404000AA0000000744A504E0010
Response:
Return:6105
Command:00C0000005
Response:6F03820138
Return:9000
Command:C832000005080000 2800
Response:
Return:9108
Command:CC0000000801000100E9002800
Response:
Return:9428
Command:CC06000028
Response:535249205241484159552042494E5449202020204D4F484420485553534E2020202020202020
020
Return:9000
```

Figure 15: WatchData Result

For the C Language compilation, the result is shown in Figure 16 and Figure 17. A user needs to press 'enter' key in order to allow the application instruction to start reading the smartcard data. Then the application window will display the smartcard data.

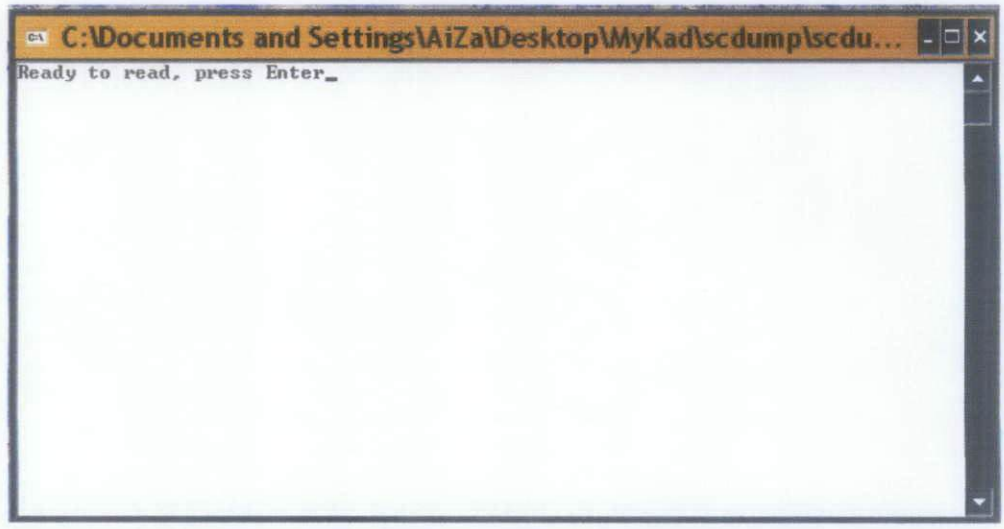

Figure 16: Compilation window

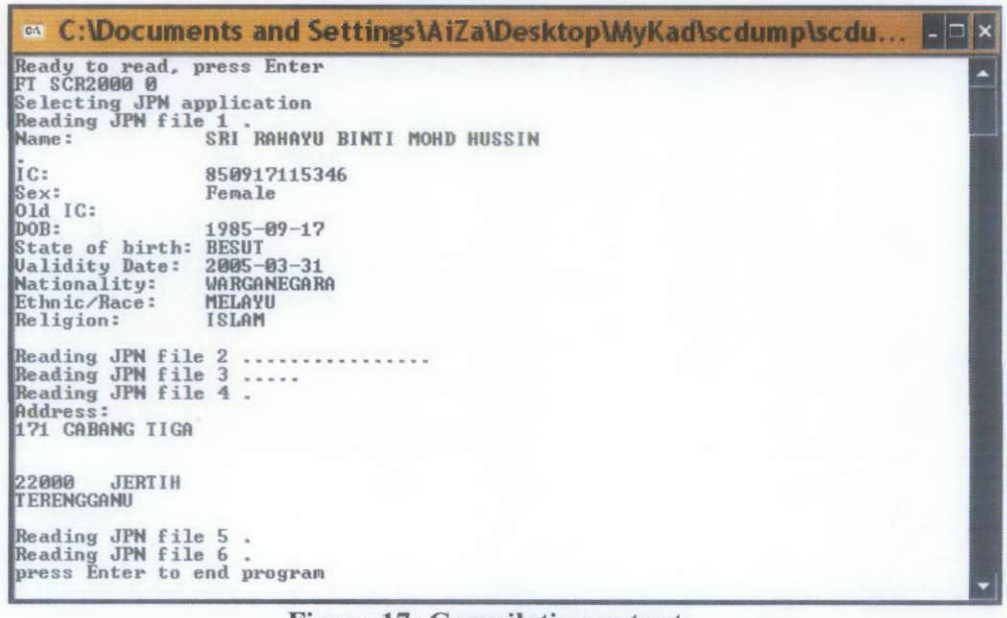

Figure 17: Compilation output

4.3 Visitor Status

Figure 18 shows the visitor status window. The system will automatically display a visitor data such as name, ID number, address and time during the in and out flow. Figure 19 shows the database table for visitor. From the table, it shows that the visitor has come to the school for nineteen times.

Figure 18: Visitor status

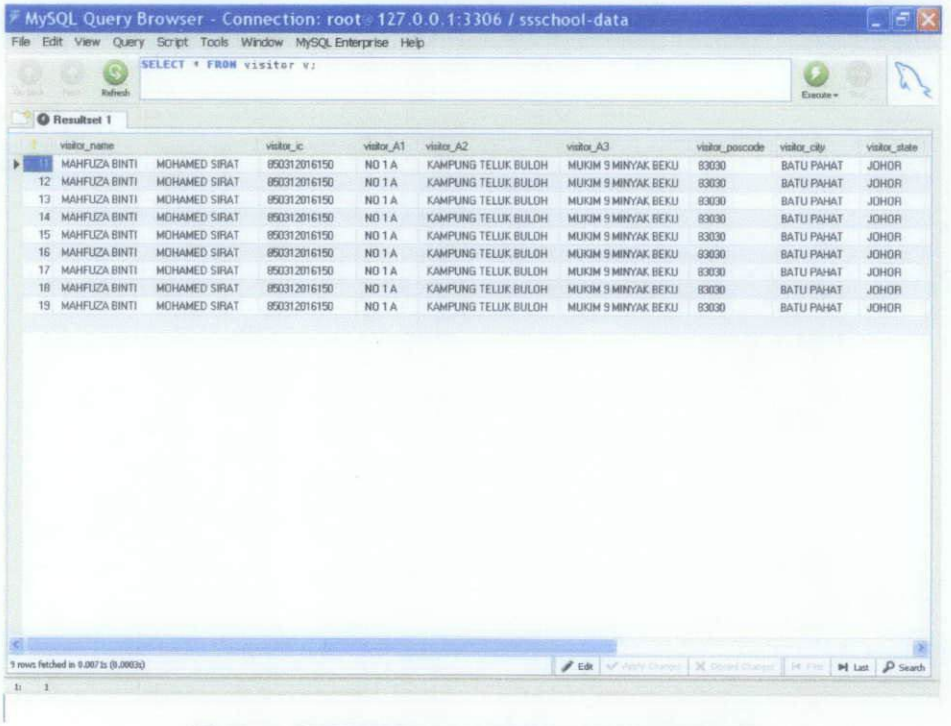

Figure 19: Database table for visitor record

4.4 Warden Status

Figure 20 shows the warden status window and the application window that can be accessed only by warden Mykad. From the warden option window, the warden needs to fill in the student IC number and verify whether the student ID number is correct or not. Then, warden can fill in the outing information or outing permission to the student. The new outing information will be saved in the student status database table.

Figure 20: Warden Status

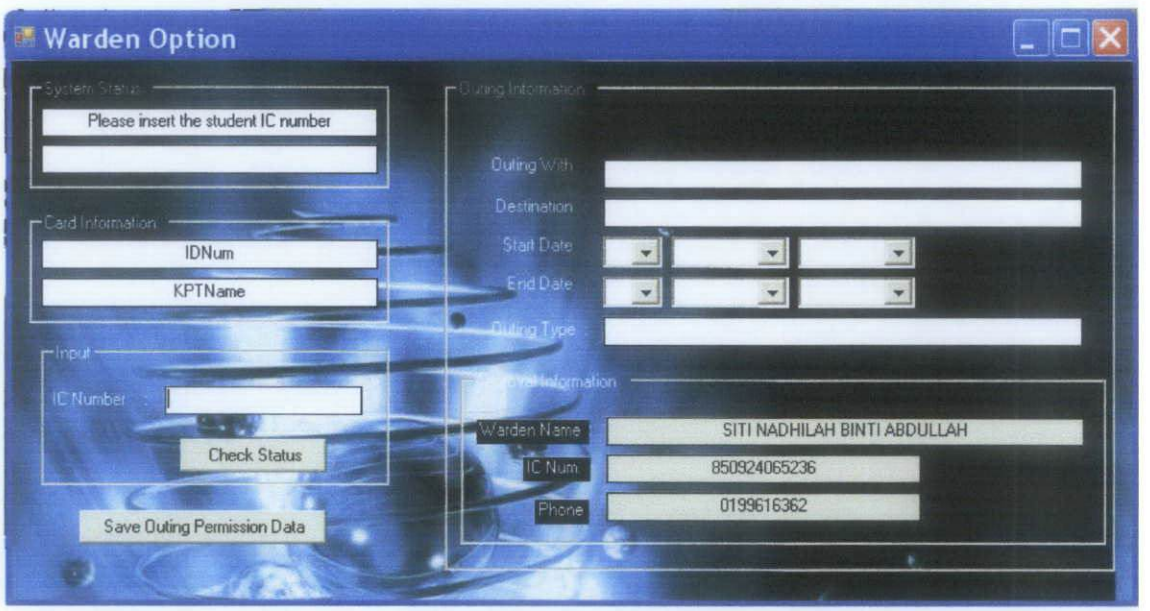

Figure 21: Warden Option window

4.5 Student Status

Figure 22 shows the student status window and Figure 23 shows the status of the student. In this example, this student does not get the outing permission. The system will display "Please get the outing permission" sentence.

Figure 22: Student Status

Figure 23: Status window without outing permission

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

As described in Chapter 4, the data of smartcard are successfully read and displayed on the system. Although this smartcard contain more than five different data, but this system only use several of the data such as identification number, owner name, address and also gender. For a visitor status, the picture of the owner is important in order to make sure the person is the owner of the card.

The matching of the smartcard data with the database runs smoothly. The system can differentiate the status of the owner. In this system, there are three different statuses, which is the student status, followed by the warden and the visitor status. The status id displayed after the card has been connected to the reader and the system performs the different operation depending on its status.

For the visitor status, the data of the MyKad owner will be saved in the visitor database table. As the warden status, there is opportunity to access other application window for giving permission for outing. This window will only pop up when the warden MyKad is inserted into the reader.

5.2 Lesson Learnt And Future Work

The project basically had improved the author's knowledge on the smart card application technology and using the Visual Studio. Even though there were hard times in implementing the software system, the project has also exposed the author on how to manage a project, to keep up with it and many more valuable lessons were learned.

As a suggestion for the future work, this system can be improved by implementing the system using new technology of MyKad. New technology is using Radio Frequency identification (RFID) technology that can reduce the operating time of the system.

REFERENCES

[1] Smart Card Technology and Applications
http://members.aol.com/pjsmart/page5.htm, 6 Sept 2007

[2] MyKad
http://www.vericardsys.com/index.php, 15 August 2007

[3] Application Protocol Data Unit
http://burks.bton.ac.uk/burks/foldoc/64/6.htm, 16 April 2008

[4] MySQL Tutorial
http://www.vbmysql.com/articles/vbnet-mysql-tutorials/the-vbnet-mysql-tutorial-part-3, 4 March 2008

[5] Answer to Reset Explained
http://www.cozmanova.com, 7 Sept 2007

[6] American Standard Code for Information Interchange
http://webmaster.lycos.co.uk/glossary/, 20 March 2008

[7] Government Multipurpose Card
http://www.jpn.gov.my/gmpc/GMPC.htm, 25 August 2007

[8] Software Development Model
http://www.codebetter.com, 11 August 2007

[9] J. C. Bradley & A. C. Millspugh, Advanced Programming Using Visual Basic.NET (2002), McGraw-Hill Companies.

[10] W. E. Burrows & J. D. Langford, Learning Programming Using Visual Basic.Net (2003), McGraw-Hill Irwin

[11] R. J. Oberg, P. Thorsteinson & D. L. Wyatt, Application Development using Visual Basic and .Net (2003), Pearson education. Inc

APPENDIX A

JPN APDU ADDRESS

jpn-1-1

Offset	Length (Hex)	Length (Dec)	SDK Function Name	Description
0000	03	3		01 04 24
0003	96	150	JPN_OrgName	original name
0099	50	30+30+20	JPN_GMPCName	GMPC name
00E9	28	20+20	JPN_KPTName	KPT name
0111	0D	13	JPN_IDNum	ID number
011E	01	1	JPN_Gender	gender
011F	08	8	JPN_OldIDNum	old ID number
0127	04	4	JPN_BirthDate	date of birth
012B	19	25	JPN_BirthPlace	place of birth
0144	04	4	JPN_DateIssued	date issued
0148	12	18	JPN_Citizenship	citizenship
015A	19	25	JPN_Race	race
0173	0B	11	JPN_Religion	religion
017E	01	1	JPN_EastMalaysian	East Malaysian
017F	02	2	JPN_RJ	RJ?
0181	02	2	JPN_KT	KT?
0183	0B	11	JPN_OtherID	other ID
018E	01	1	JPN_Category	category
018F	01	1	JPN_CardVer	card version
0190	04	4	JPN_GreenCardExpiry	green card expiry date
0194	14	20	JPN_GreenCardNationality	green card nationality
01A8	23	35		All 00

jpn-1-2

0000	03	3		01 40 03
0003	FA0	4000	JPN_Photo	JPEG photo
0FA3	08	8		All 00

jpn-1-3

0000	03	3		01 12 03
0003	14	20		"R1L1",0,0...
0017	256	598	JPN_Thumb1	right thumb
026D	256	598	JPN_Thumb2	left thumb
04C3	08	8		All 00

jpn-1-4

0000	03	3		01 01 52
0003	1E	30	JPN_Address1	address line 1
0021	1E	30	JPN_Address2	address line 2
003F	1E	30	JPN_Address3	address line 3
005D	03	3	JPN_Postcode	postcode
0060	19	25	JPN_City	city
0079	1E	30	JPN_State	state
0097	14	20		FF 00 00...

jpn-1-5

0000	03	3		01 12 00
0003	09	9	JPN_SocsoNum	socso number
000C	1F	31		All 00

jpn-1-6

0000	03	3		01 17 00
0003	0A	10	JPN_Locality	locality
000D	1E	30		All 00

APPENDIX B

ASCII STANDARD

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	NUL	32	20		64	40	@	96	60	`
1	01	SOH	33	21	!	65	41	A	97	61	a
2	02	STX	34	22	"	66	42	B	98	62	b
3	03	ETX	35	23	#	67	43	C	99	63	c
4	04	EOT	36	24	$	68	44	D	100	64	d
5	05	ENQ	37	25	%	69	45	E	101	65	e
6	06	ACK	38	26	&	70	46	F	102	66	f
7	07	BEL	39	27	'	71	47	G	103	67	g
8	08	BS	40	28	(	72	48	H	104	68	h
9	09	TAB	41	29	)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x

25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

APPENDIX C
MODWINSCARD CODE

```
Imports System
Imports Microsoft.VisualBasic
Imports System.Runtime.InteropServices

<StructLayout(LayoutKind.Sequential)> _
Public Structure SCARD_IO_REQUEST

    Public dwProtocol As Integer
    Public cbPciLength As Integer

End Structure

<StructLayout(LayoutKind.Sequential)> _
Public Structure APDURec

    Public bCLA As Byte
    Public bINS As Byte
    Public bP1 As Byte
    Public bP2 As Byte
    Public bP3 As Byte
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=256)> _
    Public DataIn As Byte()
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=256)> _
    Public DataOut As Byte()
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=3)> _
    Public SW As Byte()
    Public IsSend As Boolean

End Structure

<StructLayout(LayoutKind.Sequential)> _
Public Structure SCARD_READERSTATE

    Public RdrName As String
    Public UserData As Integer
    Public RdrCurrState As Integer
    Public RdrEventState As Integer
    Public ATRLength As Integer
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=37)> _
    Public ATRValue As Byte()

End Structure

Public Class ModWinsCard

    Public Const SCARD_S_SUCCESS = 0
    Public Const SCARD_ATR_LENGTH = 33


    '=============================================================
    ' Memory Card type constants
    '=============================================================
    Public Const CT_MCU = &H0              'MCU
    Public Const CT_IIC_Auto = &H1         'IIC (Auto Detect Memory Size)
    Public Const CT_IIC_1K = &H2           'IIC (1K)
    Public Const CT_IIC_2K = &H3           'IIC (2K)
    Public Const CT_IIC_4K = &H4           'IIC (4K)
    Public Const CT_IIC_8K = &H5           'IIC (8K)
    Public Const CT_IIC_16K = &H6          'IIC (16K)
    Public Const CT_IIC_32K = &H7          'IIC (32K)
    Public Const CT_IIC_64K = &H8          'IIC (64K)
    Public Const CT_IIC_128K = &H9         'IIC (128K)
```

```
Public Const CT_IIC_256K = &HA          ' IIC (256K)
Public Const CT_IIC_512K = &HB          ' IIC (512K)
Public Const CT_IIC_1024K = &HC         ' IIC (1024K)
Public Const CT_AT88SC153 = &HD          ' AT88SC153
Public Const CT_AT88SC1608 = &HE         ' AT88SC1608
Public Const CT_SLE4418 = &HF           ' SLE4418
Public Const CT_SLE4428 = &H10          ' SLE4428
Public Const CT_SLE4432 = &H11          ' SLE4432
Public Const CT_SLE4442 = &H12          ' SLE4442
Public Const CT_SLE4406 = &H13          ' SLE4406
Public Const CT_SLE4436 = &H14          ' SLE4436
Public Const CT_SLE5536 = &H15          ' SLE5536
Public Const CT_MCUT0 = &H16            ' MCU T=0
Public Const CT_MCUT1 = &H17            ' MCU T=1
Public Const CT_MCU_Auto = &H18          ' MCU Autodetect

'=============================================================
' CONTEXT SCOPE
'=============================================================
Public Const SCARD_SCOPE_USER = 0
'=============================================================
Public Const SCARD_SCOPE_TERMINAL = 1
'=============================================================
Public Const SCARD_SCOPE_SYSTEM = 2
'=============================================================
Public Const SCARD_STATE_UNAWARE = &H0
'=============================================================
Public Const SCARD_STATE_IGNORE = &H1
'=============================================================
Public Const SCARD_STATE_CHANGED = &H2
'=============================================================
Public Const SCARD_STATE_UNKNOWN = &H4
'=============================================================
Public Const SCARD_STATE_UNAVAILABLE = &H8
'=============================================================
Public Const SCARD_STATE_EMPTY = &H10
'=============================================================
Public Const SCARD_STATE_PRESENT = &H20
'=============================================================
Public Const SCARD_STATE_ATRMATCH = &H40
'=============================================================
Public Const SCARD_STATE_EXCLUSIVE = &H80
'=============================================================
Public Const SCARD_STATE_INUSE = &H100
'=============================================================
Public Const SCARD_STATE_MUTE = &H200
'=============================================================
Public Const SCARD_STATE_UNPOWERED = &H400
'=============================================================
Public Const SCARD_SHARE_EXCLUSIVE = 1
'=============================================================
Public Const SCARD_SHARE_SHARED = 2
'=============================================================
Public Const SCARD_SHARE_DIRECT = 3
'=============================================================
Public Const SCARD_LEAVE_CARD = 0   ' Don't do anything special on close
Public Const SCARD_RESET_CARD = 1   ' Reset the card on close
Public Const SCARD_UNPOWER_CARD = 2 ' Power down the card on close
Public Const SCARD_EJECT_CARD = 3   ' Eject the card on close
'=============================================================
Public Const FILE_DEVICE_SMARTCARD As Long = &H310000

' Reader action IOCTLs
Public Const IOCTL_SMARTCARD_DIRECT As Long = FILE_DEVICE_SMARTCARD + 2050 * 4
Public Const IOCTL_SMARTCARD_SELECT_SLOT As Long = FILE_DEVICE_SMARTCARD + 2051 * 4
Public Const IOCTL_SMARTCARD_DRAW_LCDBMP As Long = FILE_DEVICE_SMARTCARD + 2052 * 4
Public Const IOCTL_SMARTCARD_DISPLAY_LCD As Long = FILE_DEVICE_SMARTCARD + 2053 * 4
Public Const IOCTL_SMARTCARD_CLR_LCD As Long = FILE_DEVICE_SMARTCARD + 2054 * 4
Public Const IOCTL_SMARTCARD_READ_KEYPAD As Long = FILE_DEVICE_SMARTCARD + 2055 * 4
Public Const IOCTL_SMARTCARD_READ_RTC As Long = FILE_DEVICE_SMARTCARD + 2057 * 4
```

```
Public Const IOCTL_SMARTCARD_SET_RTC As Long = FILE_DEVICE_SMARTCARD + 2058 * 4
Public Const IOCTL_SMARTCARD_SET_OPTION As Long = FILE_DEVICE_SMARTCARD + 2059 * 4
Public Const IOCTL_SMARTCARD_SET_LED As Long = FILE_DEVICE_SMARTCARD + 2060 * 4
Public Const IOCTL_SMARTCARD_LOAD_KEY As Long = FILE_DEVICE_SMARTCARD + 2062 * 4
Public Const IOCTL_SMARTCARD_READ_EEPROM As Long = FILE_DEVICE_SMARTCARD + 2065 * 4
Public Const IOCTL_SMARTCARD_WRITE_EEPROM As Long = FILE_DEVICE_SMARTCARD + 2066 * 4
Public Const IOCTL_SMARTCARD_GET_VERSION As Long = FILE_DEVICE_SMARTCARD + 2067 * 4
Public Const IOCTL_SMARTCARD_GET_READER_INFO As Long = FILE_DEVICE_SMARTCARD + 2051 * 4
Public Const IOCTL_SMARTCARD_SET_CARD_TYPE As Long = FILE_DEVICE_SMARTCARD + 2060 * 4

'=============================================================
' Error Codes
'=============================================================
Public Const SCARD_F_INTERNAL_ERROR = &H80100001
Public Const SCARD_E_CANCELLED = &H80100002
Public Const SCARD_E_INVALID_HANDLE = &H80100003
Public Const SCARD_E_INVALID_PARAMETER = &H80100004
Public Const SCARD_E_INVALID_TARGET = &H80100005
Public Const SCARD_E_NO_MEMORY = &H80100006
Public Const SCARD_F_WAITED_TOO_Integer = &H80100007
Public Const SCARD_E_INSUFFICIENT_BUFFER = &H80100008
Public Const SCARD_E_UNKNOWN_READER = &H80100009
Public Const SCARD_E_TIMEOUT = &H8010000A
Public Const SCARD_E_SHARING_VIOLATION = &H8010000B
Public Const SCARD_E_NO_SMARTCARD = &H8010000C
Public Const SCARD_E_UNKNOWN_CARD = &H8010000D
Public Const SCARD_E_CANT_DISPOSE = &H8010000E
Public Const SCARD_E_PROTO_MISMATCH = &H8010000F
Public Const SCARD_E_NOT_READY = &H80100010
Public Const SCARD_E_INVALID_VALUE = &H80100011
Public Const SCARD_E_SYSTEM_CANCELLED = &H80100012
Public Const SCARD_F_COMM_ERROR = &H80100013
Public Const SCARD_F_UNKNOWN_ERROR = &H80100014
Public Const SCARD_E_INVALID_ATR = &H80100015
Public Const SCARD_E_NOT_TRANSACTED = &H80100016
Public Const SCARD_E_READER_UNAVAILABLE = &H80100017
Public Const SCARD_P_SHUTDOWN = &H80100018
Public Const SCARD_E_PCI_TOO_SMALL = &H80100019
Public Const SCARD_E_READER_UNSUPPORTED = &H8010001A
Public Const SCARD_E_DUPLICATE_READER = &H8010001B
Public Const SCARD_E_CARD_UNSUPPORTED = &H8010001C
Public Const SCARD_E_NO_SERVICE = &H8010001D
Public Const SCARD_E_SERVICE_STOPPED = &H8010001E
Public Const SCARD_W_UNSUPPORTED_CARD = &H80100065
Public Const SCARD_W_UNRESPONSIVE_CARD = &H80100066
Public Const SCARD_W_UNPOWERED_CARD = &H80100067
Public Const SCARD_W_RESET_CARD = &H80100068
Public Const SCARD_W_REMOVED_CARD = &H80100069
'=============================================================
' PROTOCOL
'=============================================================
Public Const SCARD_PROTOCOL_UNDEFINED = &H0         ' There is no active protocol.
Public Const SCARD_PROTOCOL_T0 = &H1                ' T=0 is the active protocol.
Public Const SCARD_PROTOCOL_T1 = &H2                ' T=1 is the active protocol.
Public Const SCARD_PROTOCOL_RAW = &H10000           ' Raw is the active protocol.
Public Const SCARD_PROTOCOL_DEFAULT = &H80000000    ' Use implicit PTS.
'=============================================================
' READER STATE
'=============================================================
Public Const SCARD_UNKNOWN = 0
'=============================================================
Public Const SCARD_ABSENT = 1
'=============================================================
Public Const SCARD_PRESENT = 2
'=============================================================
Public Const SCARD_SWALLOWED = 3
'=============================================================
Public Const SCARD_POWERED = 4
'=============================================================
Public Const SCARD_NEGOTIABLE = 5
```

```
'=============================================================
Public Const SCARD_SPECIFIC = 6

Public Declare Function SCardEstablishContext Lib "Winscard.dll" (ByVal dwScope As Integer, _
                                    ByVal pvReserved1 As Integer, _
                                    ByVal pvReserved2 As Integer, _
                                    ByRef phContext As Integer) As Integer


Public Declare Function SCardReleaseContext Lib "Winscard.dll" (ByVal hContext As Integer) As Integer

Public Declare Function SCardConnect Lib "Winscard.dll" Alias "SCardConnectA" (ByVal hContext As Integer, _
                                    ByVal szReaderName As String, _
                                    ByVal dwShareMode As Integer, _
                                    ByVal dwPrefProtocol As Integer, _
                                    ByRef hCard As Integer, _
                                    ByRef ActiveProtocol As Integer) As Integer

Public Declare Function SCardDisconnect Lib "Winscard.dll" (ByVal hCard As Integer, _
                                    ByVal Disposistion As Integer) As Integer

Public Declare Function SCardBeginTransaction Lib "Winscard.dll" (ByVal hCard As Integer) As Integer

Public Declare Function SCardEndTransaction Lib "Winscard.dll" (ByVal hCard As Integer, _
                                    ByVal Disposition As Integer) As Integer

Public Declare Function SCardState Lib "Winscard.dll" (ByVal hCard As Integer, _
                                    ByRef State As Integer, _
                                    ByRef Protocol As Integer, _
                                    ByRef ATR As Byte, _
                                    ByRef ATRLen As Integer) As Integer

Public Declare Function SCardStatus Lib "Winscard.dll" Alias "SCardStatusA" (ByVal hCard As Integer, _
                                    ByVal szReaderName As String, _
                                    ByRef pcchReaderLen As Integer, _
                                    ByRef State As Integer, _
                                    ByRef Protocol As Integer, _
                                    ByRef ATR As Byte, _
                                    ByRef ATRLen As Integer) As Integer

Public Declare Function SCardTransmit Lib "Winscard.dll" (ByVal hCard As Integer, _
                                    ByRef pioSendRequest As SCARD_IO_REQUEST, _
                                    ByRef SendBuff As Byte, _
                                    ByVal SendBuffLen As Integer, _
                                    ByRef pioRecvRequest As SCARD_IO_REQUEST, _
                                    ByRef RecvBuff As Byte, _
                                    ByRef RecvBuffLen As Integer) As Integer

Public Declare Function SCardListReaders Lib "Winscard.dll" Alias "SCardListReadersA" (ByVal hContext As Integer, _
                                    ByVal mzGroup As String, _
                                    ByVal ReaderList As String, _
                                    ByRef pcchReaders As Integer) As Integer

Public Declare Function SCardGetStatusChange Lib "Winscard.dll" Alias "SCardGetStatusChangeA" (ByVal hContext As Integer, _
                                    ByVal TimeOut As Integer, _
                                    ByRef ReaderState As SCARD_READERSTATE, _
                                    ByVal ReaderCount As Integer) As Integer
End Class
```

APPENDIX D
MAIN CODE

```
Imports System
Imports System.IO
Imports System.Text
Imports System.Data
Imports MySql.Data.MySqlClient

Public Class SSSchool
    Inherits System.Windows.Forms.Form
    Public Shared IDnum As String
    Public Shared ID_name As String
    Public Shared warden_phone As String
    Public Shared ID_A1 As String
    Public Shared ID_A2 As String
    Public Shared ID_A3 As String
    Public Shared ID_poscode As String
    Public Shared ID_city As String
    Public Shared ID_state As String
    Public Shared ID_gender As String
    Public Shared fs1 As System.IO.FileStream
    Friend WithEvents status_disp As System.Windows.Forms.TextBox
    Friend WithEvents picMyKad As System.Windows.Forms.PictureBox

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        InitializeComponent()

    End Sub

    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
    Friend WithEvents GroupBox2 As System.Windows.Forms.GroupBox
    Friend WithEvents txtIDNum As System.Windows.Forms.TextBox
    Friend WithEvents txtKPTName As System.Windows.Forms.TextBox
    Friend WithEvents TextBox123 As System.Windows.Forms.TextBox
    Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox
    Friend WithEvents Timer1 As System.Windows.Forms.Timer
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.components = New System.ComponentModel.Container
        Dim resources As System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(SSSchool))
        Me.TextBox1 = New System.Windows.Forms.TextBox
        Me.GroupBox2 = New System.Windows.Forms.GroupBox
        Me.TextBox123 = New System.Windows.Forms.TextBox
        Me.Timer1 = New System.Windows.Forms.Timer(Me.components)
        Me.txtIDNum = New System.Windows.Forms.TextBox
        Me.txtKPTName = New System.Windows.Forms.TextBox
        Me.GroupBox1 = New System.Windows.Forms.GroupBox
```

```
Me.picMyKad = New System.Windows.Forms.PictureBox
Me.status_disp = New System.Windows.Forms.TextBox
Me.GroupBox2.SuspendLayout()
Me.GroupBox1.SuspendLayout()
CType(Me.picMyKad, System.ComponentModel.ISupportInitialize).BeginInit()
Me.SuspendLayout()
'TextBox1
Me.TextBox1.BackColor = System.Drawing.Color.White
Me.TextBox1.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
Me.TextBox1.Location = New System.Drawing.Point(8, 20)
Me.TextBox1.Name = "TextBox1"
Me.TextBox1.Size = New System.Drawing.Size(232, 20)
Me.TextBox1.TabIndex = 5
Me.TextBox1.TextAlign = System.Windows.Forms.HorizontalAlignment.Center
'GroupBox2
Me.GroupBox2.BackColor = System.Drawing.Color.Transparent
Me.GroupBox2.Controls.Add(Me.TextBox123)
Me.GroupBox2.Controls.Add(Me.TextBox1)
Me.GroupBox2.ForeColor = System.Drawing.SystemColors.ControlLightLight
Me.GroupBox2.Location = New System.Drawing.Point(8, 12)
Me.GroupBox2.Name = "GroupBox2"
Me.GroupBox2.Size = New System.Drawing.Size(248, 82)
Me.GroupBox2.TabIndex = 6
Me.GroupBox2.TabStop = False
Me.GroupBox2.Text = "System Status"
'TextBox123
Me.TextBox123.BackColor = System.Drawing.Color.White
Me.TextBox123.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
Me.TextBox123.Location = New System.Drawing.Point(8, 45)
Me.TextBox123.Name = "TextBox123"
Me.TextBox123.Size = New System.Drawing.Size(232, 20)
Me.TextBox123.TabIndex = 6
Me.TextBox123.TextAlign = System.Windows.Forms.HorizontalAlignment.Center
'Timer1
'txtIDNum
Me.txtIDNum.BackColor = System.Drawing.Color.White
Me.txtIDNum.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
Me.txtIDNum.Location = New System.Drawing.Point(8, 18)
Me.txtIDNum.Name = "txtIDNum"
Me.txtIDNum.ReadOnly = True
Me.txtIDNum.Size = New System.Drawing.Size(232, 20)
Me.txtIDNum.TabIndex = 51
Me.txtIDNum.Text = "IDNum"
Me.txtIDNum.TextAlign = System.Windows.Forms.HorizontalAlignment.Center
'txtKPTName
Me.txtKPTName.BackColor = System.Drawing.Color.White
Me.txtKPTName.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
Me.txtKPTName.Location = New System.Drawing.Point(8, 44)
Me.txtKPTName.Name = "txtKPTName"
Me.txtKPTName.ReadOnly = True
Me.txtKPTName.Size = New System.Drawing.Size(232, 20)
Me.txtKPTName.TabIndex = 52
Me.txtKPTName.Text = "KPTName"
Me.txtKPTName.TextAlign = System.Windows.Forms.HorizontalAlignment.Center
'GroupBox1
Me.GroupBox1.BackColor = System.Drawing.Color.Transparent
Me.GroupBox1.Controls.Add(Me.txtKPTName)
Me.GroupBox1.Controls.Add(Me.txtIDNum)
Me.GroupBox1.ForeColor = System.Drawing.SystemColors.ControlLightLight
Me.GroupBox1.Location = New System.Drawing.Point(8, 115)
Me.GroupBox1.Name = "GroupBox1"
Me.GroupBox1.Size = New System.Drawing.Size(248, 75)
Me.GroupBox1.TabIndex = 53
Me.GroupBox1.TabStop = False
Me.GroupBox1.Text = "Card Information"
'picMyKad
Me.picMyKad.BackColor = System.Drawing.Color.Transparent
Me.picMyKad.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
Me.picMyKad.Location = New System.Drawing.Point(56, 196)
Me.picMyKad.Name = "picMyKad"
```

```
        Me.picMyKad.Size = New System.Drawing.Size(150, 200)
        Me.picMyKad.TabIndex = 59
        Me.picMyKad.TabStop = False
        'status_disp
        Me.status_disp.BackColor = System.Drawing.Color.White
        Me.status_disp.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
        Me.status_disp.Font = New System.Drawing.Font("Microsoft Sans Serif", 12.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.status_disp.Location = New System.Drawing.Point(16, 402)
        Me.status_disp.Multiline = True
        Me.status_disp.Name = "status_disp"
        Me.status_disp.ReadOnly = True
        Me.status_disp.Size = New System.Drawing.Size(232, 46)
        Me.status_disp.TabIndex = 60
        Me.status_disp.TextAlign = System.Windows.Forms.HorizontalAlignment.Center
        'SSSchool
        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.BackgroundImage = Global.MyKad.My.Resources.Resources._12
        Me.ClientSize = New System.Drawing.Size(269, 460)
        Me.Controls.Add(Me.status_disp)
        Me.Controls.Add(Me.picMyKad)
        Me.Controls.Add(Me.GroupBox2)
        Me.Controls.Add(Me.GroupBox1)
        Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog
        Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)
        Me.MaximizeBox = False
        Me.MinimizeBox = False
        Me.Name = "SSSchool"
        Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
        Me.Text = "SSSchool"
        Me.GroupBox2.ResumeLayout(False)
        Me.GroupBox2.PerformLayout()
        Me.GroupBox1.ResumeLayout(False)
        Me.GroupBox1.PerformLayout()
        CType(Me.picMyKad, System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)
        Me.PerformLayout()

    End Sub
#End Region

    Dim cmd, cmd1, cmd3, cmd22, cmd33 As OdbcCommand
    Dim read As OdbcDataReader
    Dim retcode, hContext, hCard As Integer
    Dim ReaderState As SCARD_READERSTATE

    Dim Mysql As String
    Dim dad As OdbcDataAdapter
    Dim dad1 As OdbcDataAdapter
    Dim dst As DataSet
    Dim dst1 As DataSet
    Dim Count As Integer = 0
    Dim CountConn As Integer = 0
    Dim conn As MySqlConnection

    Dim inc, inc1 As Integer
    Dim MaxRows As Integer
    Dim warden_IC As String
    Dim student_IC As String
    Dim strConn As String = "Driver={MySQL ODBC 3.51 Driver};Server=127.0.0.1;Database=ssschool-
iata;User=mysql;Password=mysql;Option=3;"
    Dim MyConn As New OdbcConnection(strConn)
    Dim SQL As String
    Dim SQL1 As String
    Dim omyKAD As New myKADPro.myKadSDK
    Dim strResult As String

    Private Sub Outing_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

        Dim sReaderList As String
```

```
    Dim ctr As Integer
    Dim sReaderGroup As String
    Dim ReaderCount As Integer

    Dim fsno1 As System.IO.FileStream

    fsno1 = New System.IO.FileStream("C:\MyKad\img\nophoto.jpg", IO.FileMode.Open, IO.FileAccess.Read)
    picMyKad.Image = System.Drawing.Image.FromStream(fsno1)
    fsno1.Close()
    fsno1.Dispose()

    For ctr = 0 To 2
      sReaderList = sReaderList + vbNullChar
    Next
    ReaderCount = 2
    retcode = ModWinsCard.SCardEstablishContext(ModWinsCard.SCARD_SCOPE_USER, 0, 0, hContext)
    If retcode <> ModWinsCard.SCARD_S_SUCCESS Then
      TextBox1.Text = "Established Reader Failed!"
    End If
    retcode = ModWinsCard.SCardListReaders(hContext, sReaderGroup, sReaderList, ReaderCount)
    txtIDNum.Text = "-"
    txtKPTName.Text = "-"
    TextBox123.Text = "-"
    Timer1.Enabled = True
End Sub

'Routines for Error Codes

Public Function GetScardErrMsg(ByVal ReturnCode As Integer) As String
    Dim omyKAD As New myKADPro.myKadSDK
    Select Case ReturnCode
      Case ModWinsCard.SCARD_E_CANCELLED
        GetScardErrMsg = "The action was canceled by an SCardCancel request."
      Case ModWinsCard.SCARD_E_CANT_DISPOSE
        GetScardErrMsg = "The system could not dispose of the media in the requested manner."
      Case ModWinsCard.SCARD_E_CARD_UNSUPPORTED
        GetScardErrMsg = "The smart card does not meet minimal requirements for support."
      Case ModWinsCard.SCARD_E_DUPLICATE_READER
        GetScardErrMsg = "The reader driver didn't produce a unique reader name."
      Case ModWinsCard.SCARD_E_INSUFFICIENT_BUFFER
        GetScardErrMsg = "The data buffer for returned data is too small for the returned data."
      Case ModWinsCard.SCARD_E_INVALID_ATR
        GetScardErrMsg = "An ATR string obtained from the registry is not a valid ATR string."
      Case ModWinsCard.SCARD_E_INVALID_HANDLE
        GetScardErrMsg = "The supplied handle was invalid."
      Case ModWinsCard.SCARD_E_INVALID_PARAMETER
        GetScardErrMsg = "One or more of the supplied parameters could not be properly interpreted."
      Case ModWinsCard.SCARD_E_INVALID_TARGET
        GetScardErrMsg = "Registry startup information is missing or invalid."
      Case ModWinsCard.SCARD_E_INVALID_VALUE
        GetScardErrMsg = "One or more of the supplied parameter values could not be properly interpreted."
      Case ModWinsCard.SCARD_E_NOT_READY
        GetScardErrMsg = "The reader or card is not ready to accept commands."
      Case ModWinsCard.SCARD_E_NOT_TRANSACTED
        GetScardErrMsg = "An attempt was made to end a non-existent transaction."
      Case ModWinsCard.SCARD_E_NO_MEMORY
        GetScardErrMsg = "Not enough memory available to complete this command."
      Case ModWinsCard.SCARD_E_NO_SERVICE
        GetScardErrMsg = "The smart card resource manager is not running."
      Case ModWinsCard.SCARD_E_NO_SMARTCARD
        GetScardErrMsg = "The operation requires a smart card, but no smart card is currently in the device."
      Case ModWinsCard.SCARD_E_PCI_TOO_SMALL
        GetScardErrMsg = "The PCI receive buffer was too small."
      Case ModWinsCard.SCARD_E_PROTO_MISMATCH
        GetScardErrMsg = "The requested protocols are incompatible with the protocol currently in use with the card."
      Case ModWinsCard.SCARD_E_READER_UNAVAILABLE
        GetScardErrMsg = "The specified reader is not currently available for use."
      Case ModWinsCard.SCARD_E_READER_UNSUPPORTED
        GetScardErrMsg = "The reader driver does not meet minimal requirements for support."
      Case ModWinsCard.SCARD_E_SERVICE_STOPPED
```

```
        GetScardErrMsg = "The smart card resource manager has shut down."
      Case ModWinsCard.SCARD_E_SHARING_VIOLATION
        GetScardErrMsg = "The smart card cannot be accessed because of other outstanding connections."
      Case ModWinsCard.SCARD_E_SYSTEM_CANCELLED
        GetScardErrMsg = "The action was canceled by the system, presumably to log off or shut down."
      Case ModWinsCard.SCARD_E_TIMEOUT
        GetScardErrMsg = "The user-specified timeout value has expired."
      Case ModWinsCard.SCARD_E_UNKNOWN_CARD
        GetScardErrMsg = "The specified smart card name is not recognized."
      Case ModWinsCard.SCARD_E_UNKNOWN_READER
        GetScardErrMsg = "The specified reader name is not recognized."
      Case ModWinsCard.SCARD_F_COMM_ERROR
        GetScardErrMsg = "An internal communications error has been detected."
      Case ModWinsCard.SCARD_F_INTERNAL_ERROR
        GetScardErrMsg = "An internal consistency check failed."
      Case ModWinsCard.SCARD_F_UNKNOWN_ERROR
        GetScardErrMsg = "An internal error has been detected, but the source is unknown."
      Case ModWinsCard.SCARD_F_WAITED_TOO_Integer
        GetScardErrMsg = "An internal consistency timer has expired."
      Case ModWinsCard.SCARD_S_SUCCESS
        GetScardErrMsg = "No error was encountered."
      Case ModWinsCard.SCARD_W_REMOVED_CARD
        GetScardErrMsg = "The smart card has been removed, so that further communication is not possible."
      Case ModWinsCard.SCARD_W_RESET_CARD
        GetScardErrMsg = "The smart card has been reset, so any shared state information is invalid."
      Case ModWinsCard.SCARD_W_UNPOWERED_CARD
        GetScardErrMsg = "Power has been removed from the smart card, so that further communication is not possible."
      Case ModWinsCard.SCARD_W_UNRESPONSIVE_CARD
        GetScardErrMsg = "The smart card is not responding to a reset."
      Case ModWinsCard.SCARD_W_UNSUPPORTED_CARD
        GetScardErrMsg = "The reader cannot communicate with the card, due to ATR string configuration conflicts."
      Case Else
        GetScardErrMsg = "?"
    End Select
End Function

Public Sub LoadListToControl(ByVal Ctrl As ComboBox, ByVal ReaderList As String)
    Dim omyKAD As New myKADPro.myKadSDK
    Dim sTemp As String
    Dim index As Integer
    index = 1
    sTemp = ""
    Ctrl.Items.Clear()
    While (Mid(ReaderList, index, 1) <> vbNullChar)
      While (Mid(ReaderList, index, 1) <> vbNullChar)
        sTemp = sTemp + Mid(ReaderList, index, 1)
        index = index + 1
      End While
      index = index + 1
      Ctrl.Items.Add(sTemp)
      sTemp = ""
    End While
End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    Dim omyKAD As New myKADPro.myKadSDK
    Dim CardStatus As Boolean
  ReaderState.RdrName = "FT SCR2000 0"
    retcode = ModWinsCard.SCardGetStatusChange(hContext, 0, ReaderState, 1)
    If retcode <> ModWinsCard.SCARD_S_SUCCESS Then
      TextBox1.Text = (GetScardErrMsg(retcode))
      Exit Sub
    Else
      If (Int(ReaderState.RdrEventState / 32) Mod 2) Then
        CardStatus = True
      Else
        CardStatus = False
      End If
    End If
```

```
If retcode <> ModWinsCard.SCARD_S_SUCCESS Then
   Exit Sub
Else

   If CardStatus Then
      If Count = 0 Then
         TextBox1.Text = "Card Inserted."
         TextBox123.Text = "Please Wait..."
         strResult = omyKAD.BeginTransaction("FT SCR2000 0")
         If strResult = "0" Then
            txtIDNum.Text = omyKAD.getIDNum
            IDnum = omyKAD.getIDNum
            txtKPTName.Text = omyKAD.getKPTName
            ID_name = omyKAD.getKPTName
            ID_A1 = omyKAD.getAddress1
            ID_A2 = omyKAD.getAddress2
            ID_A3 = omyKAD.getAddress3
            ID_poscode = omyKAD.getPostcode
            ID_city = omyKAD.getCity
            ID_state = omyKAD.getState
            ID_gender = omyKAD.getGender
            strResult = omyKAD.getPhoto
            picMyKad.SizeMode = PictureBoxSizeMode.StretchImage
            Dim filename, currFilename As String

            ' Specify a valid picture file path on computer.
            currFilename = "myphoto2.jpg"

            If currFilename = "myphoto.jpg" Then
               filename = "c:\myphoto2.jpg"
               If File.Exists("C:\myphoto.jpg") Then
                  File.Delete("C:\myphoto.jpg")
               End If
               currFilename = "myphoto2.jpg"
            Else
               filename = "c:\myphoto.jpg"
               If File.Exists("C:\myphoto2.jpg") Then
                  File.Delete("C:\myphoto2.jpg")
               End If
               currFilename = "myphoto.jpg"
            End If

            On Error Resume Next
            fs1 = New System.IO.FileStream(filename, IO.FileMode.Open, IO.FileAccess.Read)
            currFilename = fs1.Name
            picMyKad.Image = System.Drawing.Image.FromStream(fs1)
            openDatabase()
            fs1.Close()
            fs1.Dispose()
            strResult = omyKAD.EndTransaction()

         Else

            txtIDNum.Text = "-"
            txtKPTName.Text = "-"
            TextBox1.Text = "Error!!!"
            TextBox123.Text = "Please Remove Card And Try Again."

            omyKAD.Dispose()

            strResult = omyKAD.EndTransaction()
            omyKAD.stopReader()

            Timer1.Enabled = True
            Call Outing_Load(sender, e)

         End If

         omyKAD.stopReader()
      End If
```

```
            Exit Sub

        Else
            omyKAD.EndTransaction()
            omyKAD.stopReader()
            TextBox1.Text = "No Card Inserted."
            txtIDNum.Text = "-"
            txtKPTName.Text = "-"
            TextBox123.Text = "Please Insert Card."
            Count = 0
            Call Outing_Load(sender, e)
        End If
    End If
End Sub
Private Sub openDatabase()
    Dim myConnString As String
    conn = New MySqlConnection()
    myConnString = "server=" & "127.0.0.1" & ";" _
     & "username=" & "mysql" & ";" _
     & "password=" & "mysql" & ";" _
     & "database=ssschool-data"
    conn.ConnectionString = myConnString
    Try
        conn.Open()
        check_warden()
        conn.Close()
    Catch myerror As MySqlException
        MessageBox.Show("Error Connecting to Database: " & myerror.Message)
    Finally
        conn.Dispose()
    End Try
End Sub

Private Sub check_warden()
    MyConn.Open()
    SQL = "SELECT * FROM warden w"
    cmd = New OdbcCommand(SQL, MyConn)
    dad = New OdbcDataAdapter(cmd)
    dst = New DataSet
    dad.Fill(dst, "ssschool-data")
    MyConn.Close()
    MaxRows = dst.Tables("ssschool-data").Rows.Count
    inc = -1
    If IDnum <> warden_IC Then
        Do While inc < MaxRows - 1 And IDnum <> warden_IC
            inc = inc + 1
            w_update()
        Loop
        If IDnum <> warden_IC Then
            check_student()
        Else
            status_disp.Text = "WARDEN"
            MsgBox("Warden")
            Dim mainForm As New frmwarden
            mainForm.Show()
        End If
    End If
End Sub

Private Sub w_update()
    warden_IC = dst.Tables("ssschool-data").Rows(inc).Item(2)
    warden_phone = dst.Tables("ssschool-data").Rows(inc).Item(9)
End Sub

Private Sub check_student()
    MyConn.Open()
    SQL1 = "SELECT * FROM student s"
    cmd1 = New OdbcCommand(SQL1, MyConn)
    dad1 = New OdbcDataAdapter(cmd1)
    dst1 = New DataSet
```

```
        dad1.Fill(dst1, "ssschool-data")
        MyConn.Close()
        MaxRows = dst1.Tables("ssschool-data").Rows.Count
        inc1 = -1

        If IDnum <> student_IC Then
            Do While inc1 < MaxRows - 1 And IDnum <> student_IC
                inc1 = inc1 + 1
                s_update()
            Loop
            If IDnum <> student_IC Then
                status_disp.Text = "VISITOR"
                MsgBox("Visitor", MsgBoxStyle.OkOnly)
                visitor()
            Else
                status_disp.Text = "STUDENT"
                MsgBox("Student status")
                Dim mainForm1 As New frmstudent
                mainForm1.Show()
            End If
        End If
    End Sub

    Private Sub s_update()
        student_IC = dst1.Tables("ssschool-data").Rows(inc1).Item(2)
    End Sub
    Private Sub visitor()
        Dim mainForm2 As New frmvisitor
        mainForm2.Show()
    End Sub
End Class
```

APPENDIX E
WARDEN STATUS CODE

```
Imports System
Imports System.IO
Imports System.Text
Imports System.Data
Imports MySql.Data.MySqlClient

Public Class frmwarden
    Dim conn As MySqlConnection
    Dim strConn As String = "Driver={MySQL ODBC 3.51 Driver};Server=127.0.0.1;Database=ssschool-
data;User=mysql;Password=mysql;Option=3;"
    Dim MyConn As New OdbcConnection(strConn)
    Dim SQL, SQL1 As String
    Dim cmd1 As OdbcCommand
    Dim dad, dad1 As OdbcDataAdapter
    Dim dst, dst1 As DataSet
    Dim inc, inc1 As Integer
    Dim MaxRows, MaxRows1 As Integer
    Dim student_IC As String
    Dim student_name As String
    Dim ReaderState As SCARD_READERSTATE
    Dim retcode, hContext, hCard As Integer
    Dim Count As Integer = 0
    Dim CountConn As Integer = 0
    Dim omyKAD As New myKADPro.myKadSDK

    Private Sub frmwarden_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        txtWardenName.Text = SSSchool.ID_name
        txtWardenICNo.Text = SSSchool.IDnum
        txtWardenPhone.Text = SSSchool.warden_phone
        TextBox1.Text = "Please insert the student IC number"

    End Sub

    Private Sub btnmanual_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnmanual.Click
        txtIDNum.Text = txtSubmit.Text
        findName()
    End Sub

    Private Sub btnsave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click
        opendatabase()
        MyConn.Open()
        SQL = "SELECT * FROM student_status s;"
        dad = New OdbcDataAdapter(SQL, MyConn)
        dst = New DataSet
        dad.Fill(dst, "ssschool-data")
        MyConn.Close()

        MaxRows = dst.Tables("ssschool-data").Rows.Count
        inc = -1

        Dim cb As New Odbc.OdbcCommandBuilder(dad)
        Dim dsNewRow As DataRow

        dsNewRow = dst.Tables("ssschool-data").NewRow()
        dsNewRow.Item("student_ID") = txtIDNum.Text
        dsNewRow.Item("student_name") = txtKPTName.Text
        dsNewRow.Item("date_apply") = Now.Date & " " & Now.Hour & ":" & Now.Minute
        dsNewRow.Item("outing_with") = txtOutingWith.Text
        dsNewRow.Item("destination") = txtDestination.Text
        dsNewRow.Item("start_date") = ComboBox3.Text & ":" & ComboBox5.Text & ":" & ComboBox13.Text
        dsNewRow.Item("end_date") = ComboBox1.Text & ComboBox2.Text & ComboBox4.Text
        dsNewRow.Item("type") = txtOutingType.Text
        dsNewRow.Item("warden_name") = txtWardenName.Text
        dsNewRow.Item("warden_ic") = txtWardenICNo.Text
```

```
    dsNewRow.Item("warden_phone") = txtWardenPhone.Text
    dsNewRow.Item("status") = "Got permission"
    dst.Tables("ssschool-data").Rows.Add(dsNewRow)
    dad.Update(dst, "ssschool-data")

    MsgBox("New Record added to the Database")

  End Sub
  Private Sub opendatabase()
    Dim myConnString As String

    conn = New MySqlConnection()
    myConnString = "server=" & "127.0.0.1" & ";" _
     & "username=" & "mysql" & ";" _
     & "password=" & "mysql" & ";" _
     & "database=ssschool-data"
    conn.ConnectionString = myConnString

    Try
      conn.Open()
      conn.Close()

    Catch myerror As MySqlException
      MessageBox.Show("Error Connecting to Database: " & myerror.Message)
    Finally
      conn.Dispose()
    End Try

  End Sub
  Private Sub findName()
    opendatabase()
    MyConn.Open()
    SQL1 = "SELECT * FROM student s"
    cmd1 = New OdbcCommand(SQL1, MyConn)
    dad1 = New OdbcDataAdapter(cmd1)
    dst1 = New DataSet
    dad1.Fill(dst1, "ssschool-data")
    MyConn.Close()
    MaxRows = dst1.Tables("ssschool-data").Rows.Count
    inc1 = -1

    If txtIDNum.Text <> student_IC Then
      Do While inc1 < MaxRows - 1 And txtIDNum.Text <> student_IC
        inc1 = inc1 + 1
        s_update()
      Loop
      If txtIDNum.Text <> student_IC Then
        MsgBox("Owner is not student")
      Else
        txtKPTName.Text = student_name
      End If
    End If

  End Sub
  Private Sub s_update()
    student_IC = dst1.Tables("ssschool-data").Rows(inc1).Item(2)
    student_name = dst1.Tables("ssschool-data").Rows(inc1).Item(1)
  End Sub
End Class
```

APPENDIX F
STUDENT STATUS CODE

```
Imports System
Imports System.IO
Imports System.Text
Imports System.Data
Imports MySql.Data.MySqlClient
Public Class frmstudent

  Dim conn As MySqlConnection
  Dim strConn As String = "Driver={MySQL ODBC 3.51 Driver};Server=127.0.0.1;Database=ssschool-
data;User=mysql;Password=mysql;Option=3;"
  Dim MyConn As New OdbcConnection(strConn)
  Dim SQL, SQL1 As String
  Dim cmd1 As OdbcCommand
  Dim dad, dad1 As OdbcDataAdapter
  Dim dst, dst1 As DataSet
  Dim inc, inc1 As Integer
  Dim MaxRows, MaxRows1 As Integer
  Dim space As String
  Dim status As String
  Dim date_apply As String
  Dim outing_with As String
  Dim destination As String
  Dim start_date As String
  Dim end_date As String
  Dim outing_type As String
  Dim warden_name As String
  Dim warden_ic As String
  Dim warden_phone As String
  Dim student_name As String
  Dim student_id As String

  Private Sub frmstudent_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    opendatabase()
    check_student()
    picMyKad.SizeMode = PictureBoxSizeMode.StretchImage
    picMyKad.Image = System.Drawing.Image.FromStream(SSSchool.fs1)

  End Sub
  Private Sub opendatabase()
    Dim myConnString As String

    conn = New MySqlConnection()
    myConnString = "server=" & "127.0.0.1" & ";" _
     & "username=" & "mysql" & ";" _
     & "password=" & "mysql" & ";" _
     & "database=ssschool-data"
    conn.ConnectionString = myConnString

    Try
      conn.Open()
      conn.Close()

    Catch myerror As MySqlException
      MessageBox.Show("Error Connecting to Database: " & myerror.Message)
    Finally
      conn.Dispose()
    End Try

  End Sub
  Private Sub check_student()

    MyConn.Open()
    SQL1 = "SELECT * FROM student_status s"
    cmd1 = New OdbcCommand(SQL1, MyConn)
```

```
    dad1 = New OdbcDataAdapter(cmd1)
    dst1 = New DataSet
    dad1.Fill(dst1, "ssschool-data")
    MyConn.Close()
    MaxRows = dst1.Tables("ssschool-data").Rows.Count
    inc1 = -1
    space = "Got permission"

    If space <> status Then
      Do While inc1 < MaxRows - 1 And space <> status And SSSchool.IDnum <> student_id
        inc1 = inc1 + 1
        s_update()
      Loop
      If space <> status Then
        Me.Close()
        SSSchool.status_disp.Text = "Please get the outing permission"
      Else
        TextBox1.Text = "Got Permission For Outing"
        txtDateTimeApply.Text = date_apply
        txtKPTName.Text = SSSchool.ID_name
        txtIDNum.Text = SSSchool.IDnum
        txtDateTimeApply.Text = date_apply
        txtOutingWith.Text = outing_with
        txtDestination.Text = destination
        txtStartDate.Text = start_date
        txtEndDate.Text = end_date
        txtOutingType.Text = outing_type
        txtWardenName.Text = warden_name
        txtWardenICNo.Text = warden_ic
        txtWardenPhone.Text = warden_phone
        out()

      End If
    End If

  End Sub
  Private Sub s_update()
    student_id = dst1.Tables("ssschool-data").Rows(inc1).Item(1)
    date_apply = dst1.Tables("ssschool-data").Rows(inc1).Item(3)
    outing_with = dst1.Tables("ssschool-data").Rows(inc1).Item(4)
    destination = dst1.Tables("ssschool-data").Rows(inc1).Item(5)
    start_date = dst1.Tables("ssschool-data").Rows(inc1).Item(6)
    end_date = dst1.Tables("ssschool-data").Rows(inc1).Item(7)
    outing_type = dst1.Tables("ssschool-data").Rows(inc1).Item(8)
    warden_name = dst1.Tables("ssschool-data").Rows(inc1).Item(9)
    warden_ic = dst1.Tables("ssschool-data").Rows(inc1).Item(10)
    warden_phone = dst1.Tables("ssschool-data").Rows(inc1).Item(11)
    status = dst1.Tables("ssschool-data").Rows(inc1).Item(12)
  End Sub
  Private Sub out()
    Dim cb As New Odbc.OdbcCommandBuilder(dad1)
    dst1.Tables("ssschool-data").Rows(inc1).Item(12) = "out"
    dad1.Update(dst1, "ssschool-data")

  End Sub
End Class
```

APPENDIX G
VISITOR STATUS CODE

```
Imports System
Imports System.IO
Imports System.Text
Imports System.Data
Imports MySql.Data.MySqlClient

Public Class frmvisitor
    Dim conn As MySqlConnection
    Dim strConn As String = "Driver={MySQL ODBC 3.51 Driver};Server=127.0.0.1;Database=ssschool-
data;User=mysql;Password=mysql;Option=3;"
    Dim MyConn As New OdbcConnection(strConn)
    Dim SQL1 As String
    Dim cmd1 As OdbcCommand
    Dim dad1 As OdbcDataAdapter
    Dim dst1 As DataSet
    Dim inc1 As Integer
    Dim MaxRows1 As Integer
    Dim visitor_ic As String
    Dim visitor_name As String
    Dim visitor_A1 As String
    Dim visitor_A2 As String
    Dim visitor_A3 As String
    Dim visitor_poscode As String
    Dim visitor_city As String
    Dim visitor_state As String
    Dim visitor_gender As String
    Dim time_in As String
    Dim time_out As String
    Dim status As String
    Dim status_out As String
    Dim visitor_status As String

    Private Sub frmvisitor_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

        opendatabase()
        check_visitor()
        picMyKad.SizeMode = PictureBoxSizeMode.StretchImage
        picMyKad.Image = System.Drawing.Image.FromStream(SSSchool.fs1)
        txtIDNum.Text = SSSchool.IDnum
        txtKPTName.Text = SSSchool.ID_name
    End Sub
    Private Sub opendatabase()
        Dim myConnString As String

        conn = New MySqlConnection()
        myConnString = "server=" & "127.0.0.1" & ";" _
         & "username=" & "mysql" & ";" _
         & "password=" & "mysql" & ";" _
         & "database=ssschool-data"

        conn.ConnectionString = myConnString

        Try
            conn.Open()
            conn.Close()

        Catch myerror As MySqlException
            MessageBox.Show("Error Connecting to Database: " & myerror.Message)
        Finally
            conn.Dispose()
        End Try

    End Sub
```

```
Private Sub check_visitor()

    MyConn.Open()
    SQL1 = "SELECT * FROM visitor v"
    cmd1 = New OdbcCommand(SQL1, MyConn)
    dad1 = New OdbcDataAdapter(cmd1)
    dst1 = New DataSet
    dad1.Fill(dst1, "ssschool-data")
    MyConn.Close()
    MaxRows1 = dst1.Tables("ssschool-data").Rows.Count
    inc1 = -1
    status = "IN"
    status_out = "OUT"

    If status <> visitor_status And status_out <> visitor_status Then
      Do While inc1 < MaxRows1 - 1 And status <> visitor_status And SSSchool.IDnum <> visitor_ic
        inc1 = inc1 + 1
        v_update()
      Loop
      If status <> visitor_status And status_out <> visitor_status Then
        out()
        Me.Close()
      Else
        save_visitor()
        MsgBox("New Record added to the Database")
      End If
    End If

  End Sub
  Private Sub v_update()
    visitor_name = dst1.Tables("ssschool-data").Rows(inc1).Item(1)
    visitor_ic = dst1.Tables("ssschool-data").Rows(inc1).Item(2)
    visitor_A1 = dst1.Tables("ssschool-data").Rows(inc1).Item(3)
    visitor_A2 = dst1.Tables("ssschool-data").Rows(inc1).Item(4)
    visitor_A3 = dst1.Tables("ssschool-data").Rows(inc1).Item(5)
    visitor_poscode = dst1.Tables("ssschool-data").Rows(inc1).Item(6)
    visitor_city = dst1.Tables("ssschool-data").Rows(inc1).Item(7)
    visitor_state = dst1.Tables("ssschool-data").Rows(inc1).Item(8)
    visitor_gender = dst1.Tables("ssschool-data").Rows(inc1).Item(9)
    time_in = dst1.Tables("ssschool-data").Rows(inc1).Item(10)
    visitor_status = dst1.Tables("ssschool-data").Rows(inc1).Item(11)

  End Sub
  Private Sub save_visitor()
    Dim dsNewRow As DataRow
    Dim cb As New Odbc.OdbcCommandBuilder(dad1)
    dsNewRow = dst1.Tables("ssschool-data").NewRow()

    dsNewRow.Item("visitor_name") = SSSchool.ID_name
    dsNewRow.Item("visitor_ic") = SSSchool.IDnum
    dsNewRow.Item("visitor_A1") = SSSchool.ID_A1
    dsNewRow.Item("visitor_A2") = SSSchool.ID_A2
    dsNewRow.Item("visitor_A3") = SSSchool.ID_A3
    dsNewRow.Item("visitor_poscode") = SSSchool.ID_poscode
    dsNewRow.Item("visitor_city") = SSSchool.ID_city
    dsNewRow.Item("visitor_state") = SSSchool.ID_state
    dsNewRow.Item("visitor_gender") = SSSchool.ID_gender
    dsNewRow.Item("time_in") = Now.Date & " " & Now.Hour & ":" & Now.Minute
    dsNewRow.Item("status") = "IN"
    dst1.Tables("ssschool-data").Rows.Add(dsNewRow)
    dad1.Update(dst1, "ssschool-data")

  End Sub
  Private Sub out()
    Dim cb As New Odbc.OdbcCommandBuilder(dad1)
    dst1.Tables("ssschool-data").Rows(inc1).Item(11) = "OUT"
    dad1.Update(dst1, "ssschool-data")
  End Sub
End Class
```