

INTELLIGENT HOSPITAL: PATIENT SMART CARD

By

FAIRUS BIN HARIRI

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

Copyright JUNE 2004

by

Fairus bin Hariri, 2004

CERTIFICATION OF APPROVAL

INTELLIGENT HOSPITAL: PATIENT SMART CARD

by

Fairus bin Hariri

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:



Prof. Dr. P.A. Venkatachalam
Professor
Electrical & Electronic Engineering
Academic Block No. 22
Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh, Perak Darul Ridzuan, MALAYSIA.

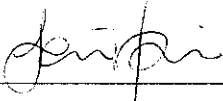
Professor Paruvachiammasai Venkatachalam,
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

JUNE 2004

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.



Fairus bin Hariri

ABSTRACT

The purpose of introducing smart card application in our lifestyle is to improve and develop the way of life to a more convenient style. This approach can be applied to various areas such like mobile communication, public transportation, finance, public sector, and even health care system. In this project, the main target is to design an application and basic system of interfacing between patient and medical institution, which is the Patient Smart Card. It will act as a synergy between clients or patients' personal information with the server or medical centre that complies with the system.

Patient Smart Card application development involves several stages. Defining the methodology, there will be four phases, which are analyzing, designing, coding and testing. The cores of the project are the database development, smart card application and its interface program. Data structure of the system is obtained through research and detailed assessment. Entirely, the back end of the system is concerning the source code and interface design. To achieve those with fine results, there are tools required during the whole process.

Thus, the result will be concluded based on the objectives set. Besides smart card and its reader, the system comprises of several forms which will act as the interface between database and client. The forms encompass the main menu, registration process, administration purpose, PIN number security system, patient particular information and related data regarding to mass patient medical records. All of them have the same focal purpose which is to ease and create simplicity for the current medical information record system applied at most of the health centers. Suggested works for further enhancement and realization are also stated.

ACKNOWLEDGEMENTS

Firstly, the author would like to express the deepest gratitude to God for giving the strength, wisdom and patience in order to complete this project as per time given. The author's admiration and gratification then goes to the supervisor, Professor Paruvachiammasai Venkatachalam for all the support and guidance within the period of completing Final Year Design Project (EEB 5034).

The author thanks Mr. Mohd Zuki Yusoff, lecturer of Universiti Teknologi PETRONAS (UTP) for coordinating the subject smoothly. Thank you also to UTP Department of Electrical and Electronics (E&E) Engineering Program Head, Associate Professor Dr. Mohd Noh Karsiti for his authorization and support within the period of completing this project.

The author thanks Mr. Hariri bin Sadiman and Mrs. Rukiah bt. Jamaluddin, the beloved parents for the morale support, love and encouragement.

The author thanks Mr. Marc Talbot, Smart Card expert from Axalto for the brief lecture, firm training session on the development of smart card and mostly for the supplement of Cyberflex Access Software Development Kit. The author thanks Mr. Ng Tze Yang, Axalto Smart Card Technical Consultant Access Card, and also to Mr. Dimitri Denamany, student of UTP for lending his thought, knowledge and experience regarding the smart card development.

The author thanks Mr. Arzumy Yusof, Ms. Nor Syarina Mohamad Nordin, Ms. Nora Azura, Mr. Saiful Adli Adam, all students of UTP, and Mr. Musa, E&E lab technicians for their severe involvement in the project and bestow their valuable understanding and acquaintance throughout the accomplishment of this project. On the whole, a single page acknowledgement is absolutely not enough to show author's sincere gratefulness, love and respect to all of the people who have helped.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF ABBREVIATIONS	ix
CHAPTER 1: INTRODUCTION	1
1.1 Background of Study	1
1.2 Problem Statement	2
1.2.1 Problem Identification	2
1.2.2 Significance of the Project	3
1.3 Objective and Scope of Study	3
1.3.1 The Relevancy of the Project	3
1.3.2 Feasibility of the Project within the Scope and Time Frame	4
CHAPTER 2: LITERATURE REVIEW	5
2.1 Why Patient Smart Card	5
2.2 Smart Card Technology	6
CHAPTER 3: METHODOLOGY	9
3.1 Procedure Identification	10
3.1.1 Analysis	10
3.1.2 Designing	11
3.1.3 Coding	12
3.1.4 Testing	28
3.1.5 Data Collection	28
3.2 Tools Required	28
3.2.1 Cyberflex Access E-gate Card SDK	30
3.2.2 Hardware Used	31
3.2.3 Software Used	32
CHAPTER 4: RESULTS AND DISCUSSIONS	33
4.1 Findings	33
4.1.1 Interface and Data Structure	33

4.1.2 Database	40
4.1.3 Bona Fide Existing Patients Data	41
4.2 Discussions	41
4.2.1 Overall Implementation	41
CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS	43
5.1 Relevancy to the Objectives	43
5.2 Expectation of the System	44
5.3 Problems Encountered	44
5.4 Suggested Future Work for Expansion and Continuation	45
5.4.1 Development of the System	45
5.4.2 Application-wise	45
LIST OF REFERENCES	47
LIST OF APPENDICES	48

LIST OF TABLES

Table 3.1	Data Type Particulars	18
Table 3.2	Cyberflex Access E-gate Card Specification	31
Table 4.1	Suggestion of the Preliminary Contents of the Smart Card	42

LIST OF FIGURES

Figure 1.1	Basic Layout of Patient Smart Card System	2
Figure 3.1	Software Engineering Waterfall Model	9
Figure 3.2	Java Card Architecture	11
Figure 3.3	Process Flow of Converting Source Code to Applet Instances	13
Figure 3.4	Program File Generator (PFG) Interface	22
Figure 3.5	Cyberflex Access Toolkit 4.4 Interface	25
Figure 3.6	Create Program Dialog Box	25
Figure 3.7	Create Instance Dialog Box	27
Figure 3.8	Block Diagram of Tools Required	29
Figure 4.1	System Main Menu	34
Figure 4.2	Administrator Menu	34
Figure 4.3	Patient New Registration	34
Figure 4.4	Existing User Menu	35
Figure 4.5	Personal Details Folder	36
Figure 4.6	Previous Medical Institution Folder	36
Figure 4.7	Doctors Preferences Folder	37
Figure 4.8	Medical History Folder	37
Figure 4.9	Medication Folder	38
Figure 4.10	Emergency Contact Folder	38
Figure 4.11	Medical History Folder	39
Figure 4.12	Immunization List Folder	39
Figure 4.13	List of Database Tables	40

LIST OF ABBREVIATIONS

APDU - Application protocol data unit
AID - Application identifier
API - Application programming interface
AUTH key - Authentication key
CAD - Card acceptance device
CAP - Converted applet
CA - Certificate authority
CLA - Class
COVE - Cryptographic Object Viewer and Editor
CryptoAPI - Cryptographic Application Programming Interface
DES - Data Encryption Standard
EEPROM - Electrically erasable programmable read-only memory
IJC - Interoperable java card
INS - Instruction
ISO 7816 - International Standardization Organization specification #7816
JCRE - Java card runtime environment
JVM - Java card virtual machine
KEK - Key encryption key
MAC - Message Authentication
PIN - Personal identification number
PKCS #11 - Public Key Cryptography Standard #11
PKI - Public Key Infrastructure
PVC - Poly Vinyl Carbon
RSA - Rivest, Shamir, and Adleman
SDK - Software development kit
SDLC - System development lifecycle
TPDU - Transmission protocol data unit

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 Background of Study

The health care organization nowadays should be very critical and effective. This is to ensure the quality of medical treatment, effectual management and cost reducing. Thus, patient smart card can be considered as elemental and very important for the next potential years. In general, this project will have an end product with a function of linking clients or patients' personal information with the server or medical centre that complies with the system (Figure 1.1). Apart from crucial security, the system should guarantee the integrity of all the records.

The three main components in this project would be the database development, smart card application and its interface program. The area of study is more towards information technology (IT) acquaintance. Besides human computer interaction (HCI) understanding, author must also get used to graphical user interface (GUI) designing method. These two subjects will have main function as an intermediary between human and machine. However, the foundation of this project would be Java Programming. Author has to familiarize with advanced programming language skills besides disseminating perceptive on the particular smart card architecture. Technically, author will encounter diverse new scope of study associated with IT knowledge.

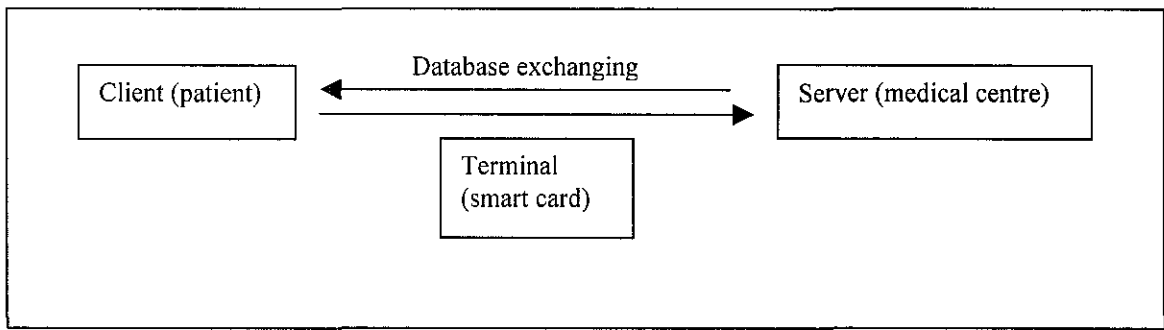


Figure 1.1 Basic layout of patient smart card system

1.2 Problem Statement

Hospitals or clinics are considered as one of the public areas. They are often visited by people most of the time, in our everyday lives. It is well known that public spot might be bottlenecking at times. By right, information storage might encounters problem besides demanding manpower during peak hours. Thus, author sees that the introduction of a smart card based health system is a must so that there will be a proper solution for efficient work ethics. This is a part of techniques to achieve paperless environment, efficient workforce and organized system.

1.2.1 Problem Identification

There are several problems, identified at the medical organizations which necessitate this project to be accomplished:

- a) Conventional way of registering new patient.
- b) Filing system using plain card can be frustrating for archive purpose.
- c) Patient information is not dynamically shared by other general practitioners (GP) for proper orientation.
- d) Patients' medical history is quickly unavailable for further reference.
- e) Patient medical information is not centralized between clinics and hospitals.

By right, as for as the author's concern, implementation of this project will lead to new problems such as:

- a) Possibility of illegal access.
- b) Hacking terminal database or smart card architecture.
- c) Client calamity of losing the smart card.
- d) Burden for client to carry numerous smart cards simultaneously.

1.2.2 Significance of the Project

The significance of this project to the problem is that it will act exactly as an intermediary between three parties, which are the medical centre personnel, smart card user and the interface of the system. Existing system such as My-Kad and Bankcard are part of the elements in prompt to develop this application. It will definitely create an effective confidential communication environment. This project also must be reliable enough for further enhancement, such as using more suitable interface and applying Oracle as a more stable database system. If this project is a complete success, it might also be applied in real industry, as long as it is stable and secured enough. It is going to play its part in the development of an intelligent hospital.

1.3 Objective and Scope of Study

1.3.1 The Relevancy of the Project

This project need less theory on pass course to be practiced, which mean author has the opportunity to go deep into IT knowledge. Yet, author must familiarize with the particular smart card architecture, for instance the electrical characteristics of the card. Upon completing this project, there will be two different approaches involved, which are divided into two respective semesters:

- a) Semester 1 - Design and develop preliminary interface layout and exact information to be in the system, and prepare tools for Java programming.
- b) Semester 2 - Develop and debug the prototype, its database and client terminal system for successful implementation.

For a long term plan, the main objective of this project is for enhancing the existing medical database system in our country. That is to create a smart, efficient, productive and efficient environment for the subject. However, it has to be realized partially, by applying a test run to a small clinic and particularly a single patient. Author had successfully achieved all the respective objectives during the first semester. Thus, to realize Patient Smart Card, for this particular 2nd semester, the objectives are:

- a) To complete the source code development for both Java card and interface software.
- b) To design the data structure and interface coordination appropriately.
- c) To design smart card that is able to store data.
- d) To gather information and perform case study on several existent patients to conjure up with the project.

1.3.2 Feasibility of the Project within the Scope and Time Frame

The product is a Patient Smart Card. Upon completing the whole prototype, the objectives of the 2nd semester are vital. The scheduled tasks and milestones for this semester of the final year project are summarized in the Gantt chart [Appendix A1]. It covers the entire schedule from doing the preliminary design and constructing data structure, coding, testing and also report writing. The scope of project seems to be feasible for author to complete on time with the possible outcomes, and the time allocated will be used efficiently for developing the whole end product.

CHAPTER 2

LITERATURE REVIEW AND THEORY

2. LITERATURE REVIEW

Deciding on what is good or not is very subjective. It depends on the user very own perspective of applying the subject. Patient Smart Card can be very offensive to some people and vice versa. Technologically, this application can as well be considered as evolution to the health care system. To satisfy a range of people would be impossible as some might judge it as a burden.

2.1 Why Patient Smart Card

The Health Card is a smart card incorporating a microprocessor and encoding functions. Patients authenticate themselves to the card by means of a PIN, and can permit the physician of their choice access to their personal data. The patient data is stored either on the card or by the background system. The Health Card is counterfeit-proof and the patient clearly identifiable by means of a photo and the pin. This avoids the risk of misuse connected with conventional health insurance cards, which in Germany has caused losses running into billions [12].

Doctors and paramedics in Bahrain will soon have instant access to patient medical records thanks to a smart card system being launched by Bahrain Defense Force (BDF) Hospital. BDF Hospital signed an agreement with the Central Informatics Organization, which will make it the first hospital to implement the system. Paramedics will be able to get the medical information of the patient's smart card and that will help them determine the best course of treatment, according to reports. The card could also inform the

hospital en route of the patient's condition electronically, by using a GSM device that will be installed in the ambulance, helping staff at the hospital to assess the condition more accurately. The new electronic system will also save time by eliminating the need for repeated laboratory reports, since doctors will be able to access up-dated information via the smart card. Medical information about the patients will be saved at a national data bank and medical personnel will be able to access it by using the smart card reader. The BDF Hospital will become the first in the Kingdom to implement the system [14].

With the cost and demand of healthcare increasing worldwide, this industry needs to make use of the latest technology to give people the most adequate and efficient care. The key is to provide a link via smart cards that allow systems to exchange information. Two questions must be answered regarding smartcards: Who are the key people using them and how does it benefit them? More over, patients have mobile, secure and up-to-date information, doctors save time doing paperwork for referrals and insurance claims, insurers eliminate fraud and redundant costs, and pharmacies will save time filling prescriptions electronically once they are added to this system. "The smart card is simply the old paper medical record with a college education"[8].

These statements clarify the needs of smart card technology. The usage of smart card in medical centre institution creates an easier way of communicating and arranging patients' records.

2.2 Smart Card Technology

“Plastic cards are a part of the way of life in the most industrialized countries. We can use them to identify ourselves, to travel, to gain access to buildings, to obtain cash from our bank, and to pay for services. We are regularly offered new types of cards; many people collect every card they are offered, while others feel that their lives are already excessively controlled by anonymous pieces of plastic” [5, pg 1].

Axalto provides leading-edge technology to enable innovative smart card and terminal applications that enhance the security and convenience of businesses and communities of all kinds. The company provides cards, terminals, development tools and support in open configurations for operators, developers, integrators and distributors worldwide [9].

There will be 6 billion money cards within 5 years. The distribution will be 40% in Asia, 35% in Europe, and 25% in North America. This shows optimism for the Asian market despite its current economic slump. The figures last year indicate 76% of all smart cards were in Europe, 14% in Asia, and 10% in America. 62% of people surveyed said deployment of an infrastructure to support Smart cards is crucial for its success. 32% believe consumer acceptance is barrier to Smart Card success. Multi application cards will be used for Healthcare (45%) and loyalty programs (32%). International vertical markets include retail (33%), telecommunications (28%) and banking (22%). Smart cards bring in new applications that require more than just adding a smart card reader [15].

Smart cards provide computing and business systems the enormous benefit of portable and secure storage of data and value. At the same time, the integration of smart cards into your system introduces its own security management issues, as people access card data far and wide in a variety of applications. In terms of smart card security system, it provides data integrity, authentication, non-repudiation and authorization where the processes of allowing access to specific data within a system and the utilization of a third party to manage and certify each of the users of your system are performed. Cryptography is the method of converting data from a human readable form to a modified form, and then back to its original readable form, to make unauthorized access difficult. The original data may be in a human-readable form, such as a text file, or it may be in a computer-readable form, such as a database, spreadsheet or graphics file. The original data is called unencrypted data or plain text. The modified data is called encrypted data or cipher text. The process of converting the unencrypted data is called encryption. The process of converting encrypted data to unencrypted data is called

decryption [10].

From author point of view, the application of smart card in every single area in our life is now becoming a necessity. People are now living in a fast pace and demanding high lifestyle. People will insist for a firm and quick response in every aspect of life. This is where the smart card applications provide solution with guaranteed security.

CHAPTER 3

METHODOLOGY

3. METHDOLOGY

The process of completing this project involves several components and distinct stages. Mainly, the 3 parts of the system that are the interface, client terminal and database server are completed separately.

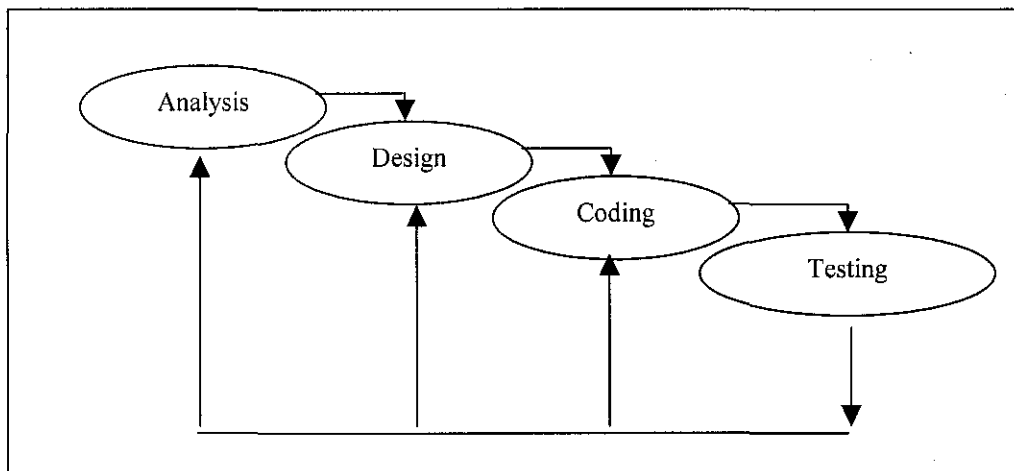


Figure 3.1 Software Engineering Waterfall Model

Applying software engineering knowledge, the classic waterfall model is applied to design the system. These stages are referred to as system development lifecycle (SDLC). This methodology was chosen because it provides systematic approach in solving any encountered problem. Cycle in SDLC refers to the natural tendency for the system to cycle through these activities, which can be viewed in Figure 3.1.

3.1 Procedure Identification

3.1.1 Analysis

This phase is for the study of the problems, causes and effects. Next, there will be problem identification and a list of requirements in providing the solution. There are two elements in the preliminary analysis, which are the system comprehensive data and smart card architecture. These two provide ample information and detail in proceeding to the next designing stage.

a. System Comprehensive Data

Several methods of gathering useful data are used throughout this preliminary period. There had been interview, literature reviews, Internet research and also questionnaire to clarify the problem statement. The data compiled will be used for creating comprehensive coverage of patient database and interface of the system. In short, every aspect of this project is covered expect for the smart card terminal programming and its architecture.

b. Smart card architecture

This is done by performing literature reviews and doing research on the Internet. The information obtained is not directly involves with the project, but it provides more understanding of the smart card interior system.

For this project, smart card programmed in Java programming language is used. The latest Java Card technology smart card is from Schlumberger called Cyberflex Access e-gate 32K card. The Cyberflex Access e-gate 32K card adds byte code verification (Code shield feature) for applets downloaded to the smart card, supports USB communication protocol, and has some differences in the implementation of the PutKey command.

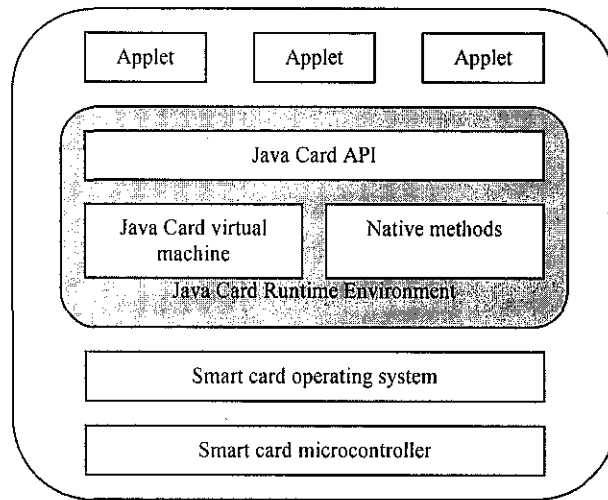


Figure 3.2 Java card architecture

The Java Card architecture is illustrated in Figure 3.2. As can be seen, it looks very similar to traditional Java. The smart card operating system (OS) is layered on top of a smart card microcontroller and is aimed at providing common services like file and data management, communication, and command execution. From the communication point of view, Java Card is fully compliant with ISO/IEC 7816. In particular, Java Card supports communication protocols and commands in accordance with ISO/IEC 7816-3 and ISO/IEC 7816-4, respectively.

A new Cyberflex Access card contains the following elements:

- a) Card Manager Application
- b) Default key set for the Card Manager
- c) Card Production Life Cycle (CPLC) data
- d) Global Platform compatibility for Java Card Virtual Machine v2.1

3.1.2 Designing

The designing of system layout and GUI for the application needs a good HCI knowledge. Therefore, during this stage, the three main factor of HCI are the study of:

- a) Anthropology - science of the origin and development of human.

- b) Psychology - science of the mind and human nature.
- c) Sociology - science of organization of human society.

The knowledge of system analysis is applied. At this step, the designing started with Unified Modelling Language (UML) diagram. Object-oriented analysis was performed using UML. Object-oriented design is a design strategy where system designers think in terms of 'things' instead of operations or functions.

The execution system is made up of interacting objects that maintain their own local state and provide operations on that state information. An object-oriented design process involves designing the object classes and the relationships between these classes. When the design is realized as an executing program, the required objects are created dynamically using the class definitions. UML is an integration of the different notations for describing object-oriented designs. There are several diagrams produced using UML [2].

3.1.3 Coding

Upon the completion of analysing and designing stages, there is the coding of the whole source code for the system. The main source codes to be developed are for the Java Applet smart card programs for Open Platform Cyberflex Access cards. The other one would be for the interface using Microsoft Visual Basic 6.0. This stage consumes most of the time because the main constraint is to learn all the programming language which is relatively different from author's main course. Coding stage also will mostly necessitate author to be well literate in computer programming software.

During this stage, the process of developing the source code of this particular Open Platform smart card is important. There are stages and exact practices to be performed such as its requirements, library path setting and Java applet instances. Figure 3.3 shows exactly on how to prepare Java source code for use on an Open Platform Cyberflex Access.

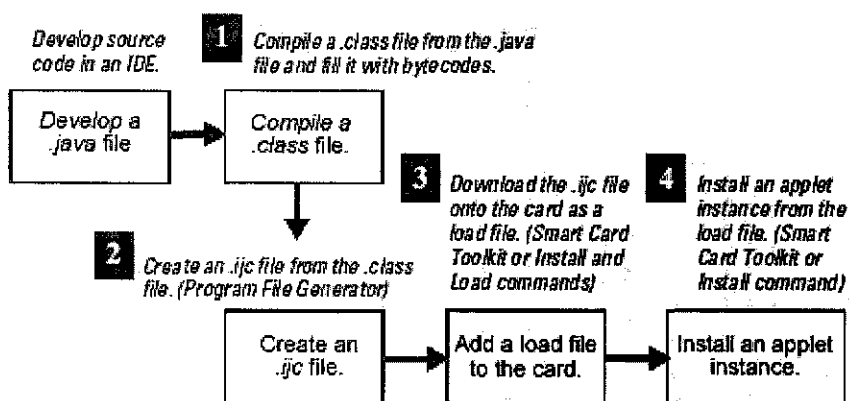


Figure 3.3 Process flow of converting source code to applet instances [7]

This part will explain exactly on how to prepare Java source code for use on an Open Platform Cyberflex Access smart card.

Firstly, Java source file is converted into a class file. The card class file contains the program information converted to a stream of Java byte codes. Then, the class file is converted to an applet file. This can be performed by using Program File Generator dialog (PFG) box. In this case, the Sun Java Card Runtime Environment (JCRE) must be available when Program File Generator is running. Next the program file data is added to the card as a load file. This is done by using the smart card toolkit software provided. However, it also can be done by using Application Protocol Data Unit (APDU) commands. Finally, the applet instance is installed on the card by also using the software provided or APDU commands [7].

a. Terminology

There are certainly different terminologies or new jargons that are frequently to be used. It is necessary to explain term by term before going further to the details. The descriptions of all the terms are as below [7].

- a) APDU - application protocol data unit. It is a sequence of hexadecimal values that match to the low-level format for data exchanged. It performs interconnection between the host application and the card (through a card reader or terminal). All the command and response for APDU formats are defined by the ISO 7816-4 specification.
- b) API - application programming interface. It is actually software that defines the calling conventions to gain access to lower-level services performed by any other application.
- c) AUTH key - authentication key. To make use an Open Platform Cyberflex Access card, AUTH key is the first key that will be asked in a key set. It is a static key used to generate a session key for encryption and authentication.
- d) Authentication - In the case of smart card cryptography, authentication give means of the ability of the sender and receiver to confirm each other's identity.
- e) CAD - card acceptance device. This is the term for the terminal or card reader used to communicate with a smart card.
- f) CAP file - converted applet file. The file contains an executable binary representation of the classes in a Java package and specially formatted to be compatible with the Java Card platform. However, for this project, the CAP file is replaced by the IJC file type.
- g) Card Manager – It represents the application on the card which actually handles incoming commands. It also controls which applications can be loaded onto the card after it is issued.
- h) Certificate – It contains a user's public key, digitally signed by a certificate authority (CA) to assure the recipient that the key belongs to the user. Digital certificates may be kept in registries, which authenticated users search to find other users' public keys.
- i) Certificate authority (CA) - A trusted body or unit that maintains information about a client user's identities and issues a digital certificate to each user. A CA may be an independent enterprise or sub company of the user's company. The convenience of the endorsement depends on the certificate issuer's status of a trusted authority.

- j) CLA - class. This is the first byte of an APDU and it identifies the command class. The class and instruction (INS) bytes exclusively will recognize the command type.
- k) COVE - Cryptographic Object Viewer and Editor. This software is included in the Software Development Kit (SDK) provided. The function is personalize cards and setting up keys and key files on the card.
- l) CryptoAPI - Cryptographic Application Programming Interface. It is a Personal Computer/Smart Card (PC/SC) application and support high-level programming environment that routes high-level function calls to a cryptographic service provider (CSP) as an interface to card-based services. CryptoAPI is included in all 32-bit Windows software, and provides the cardholder with an easy-to-use interface for identity checks.
- m) Cryptography – It is the discipline of information security where all plaintext or ordinary text is encrypted into cipher text and then decrypted.
- n) DES - Data Encryption Standard. A symmetric algorithm which uses a single key for both encryption and decryption. It is suitable to be used when keys provided are distributed and stored dependably and securely. Thus, DES main function is to protect data from being spied during transmission.
- o) Diversified keys – There are three types of keys in order to establish a secure channel on an Open Platform Cyberflex Access card. They are AUTH, MAC, and KEK keys. They are used after being generated with a Master Key set. By applying this, each card will of course have different keys. Therefore, user can only access the card by knowing not only the Master Keys but also the cryptographic algorithm and the data used. In short, it gives a card greater security.
- p) EEPROM - Electrically erasable programmable read-only memory. It is the type of microprocessor memory that does not lose or erased when power is removed and it only can be erased and reprogrammed repeatedly with applied electrical voltage.
- q) E-gate - commercial name of the smart card provided by the respective company. It has the capability of interfacing with a standard ISO 7816 reader or with the Universal Serial Bus (USB) port of a PC.
- r) Global PIN - global personal identification number. An alphanumeric string used as a password to establish person-to-card authentication.

- s) Global Platform specifications - Open and have cross-industry standards for card program management and terminal communications. Moreover, it supports a security-oriented approach to card application development and operation. It is the based operating reference for the Open Platform Smart Card.
- t) IJC - interoperable java card CAP file. A file on the smart card that contains an executable binary representation of the classes in a Java package. It is then formatted to be compatible with the Java Card platform. It also contains extra information needed for on-card byte code compare to the older CAP file.
- u) INS - instruction. This is the second byte of an APDU which identifies the command instruction. The class (CLA), as describe earlier and instruction bytes uniquely identify the command type.
- v) ISO 7816 - International Standardization Organization specification #7816. The organization has provided a set of standards for smart card physical attributes, data elements, commands, and security architecture.
- w) Java card applet - Smart card applet which is written in the Java language. By right, the applet can be downloaded to a Java-enabled smart card as a load file, and then be instantiated and executed it in the Java Card Runtime Environment (JCRE).
- x) JCRE - Java card runtime environment. This term is covering the concept of a whole system; consist of the Java card virtual machine (JVM), Java smart card libraries, and supporting files. Together, they comprise an environment for executing smart card applet instances.
- y) JVM - Java card virtual machine. In simple word, it is actually a converter that runs on the host system and also an interpreter that runs on the card. In other words, these applications will act as a translator that converts an application's instructions into a format the smart card can understand. The JVM loads and executes Java class files and finally makes applications transferable.
- z) KEK key - key encryption key. On the smart card, a KEK key is the third key in a key set. The KEK key is a static key used to generate a session key for key encryption. It is one of the three keys included in the diversified keys.

- aa) Load file - A file downloaded to the smart card from a program file, which is then being instantiated so that it can be used as a Java applet instance that runs on the card.
- bb) MAC - Message Authentication Code. A variation on a one-way hash function, which uses a DES or 3DES key to compute the hash digest. This feature makes a MAC useful for confirming data integrity, as well as authenticating the identities of the message originator and recipient. RSA key pairs can also be used with one-way hash functions to produce a digital signature.
- cc) Open Platform smart card - A smart card whose operating system is based on the Global Platform specification not really on a file system.
- dd) PIN - personal identification number. An alphanumeric string which can be used as a password to set up person-to-card verification.
- ee) PKCS #11 - Public Key Cryptography Standard #11. The founder Rivest, Shamir, and Adleman (RSA) Laboratory-sponsored standard protocols developed to assist secure information exchange. Cyberflex Access series cards support PKCS #11-compliant card programs by supplying a library of functions that offer cryptographic and security services to the PKCS #11 interface.
- ff) PKI - Public Key Infrastructure. It is an infrastructure model for exchanging data and money through the use of a public and private key pair. The keys can be obtained and shared through a trusted authority. It also provides for digital certificates that can identify individuals and organizations.
- gg) Program File Generator - One of the software provided in the SDK. An application for the host system which can convert class files in a Java package into an IJC file and an export file for the converted package.
- hh) RSA - A set of asymmetric key system. An extensively used system known by the initials of its originators, whom are Rivest, Shamir, and Adleman (RSA). In this guide, RSA stands for the 512-bit, 768-bit, or 1024-bit encryption algorithm used by Cyberflex Accessories smart cards. RSA uses a public and private *key pair*, with the *public key* published openly, while the *private key* remains secret. It is done that way to enhance security.

- ii) Security domain - It is an environment that is already established by the developer for the smart card itself to protect data, such as an applet. The environment will also enable applications from multiple providers to inhabit together on a smart card without compromising each other's security. Typically, a security domain has a domain-specific key set for security use.
- jj) TPDU - transmission protocol data unit. This last terminology is actually a basic element of the lower-level protocol for exchanging APDU data between the host application and the card. As defined by the ISO 7816-3 specification, the T=0 protocol is an asynchronous, byte-oriented, half-duplex transmission protocol in which a byte is the smallest transmissible data unit. While the T=1 protocol is an asynchronous, block-oriented, half-duplex transmission protocol.

b. Java Card Development Requirement

There are a few requirements to be fulfilled during the process of developing the applet instances. Process method is created and derived from the applet class itself. Besides, data types used must be only Boolean, byte, and short as the card does not support other data types. Byte and short data types are *signed*, and must be within the ranges as in table 3.1 below [7].

Table 3.1 Data type particulars

Data Type	Length	Maximum Value	Minimum Value
Byte	8 bits	+127	-128
Short	16 bits	+32767	-32768

c. Methods to be included

An applet must include the install, select, and process methods. All these methods were to be declared in the applet's primary class. The purpose of install method is to create the class objects needed by the applet. The select method is to prepare the applet to

receive and process Application Protocol Data Unit (APDU) commands. Finally, the function of process method is to handle APDU commands.

d. Library Path

To ensure source code can be debugged, the host system's environment variable path must be set to the class library directory and also to the Cyberflex Access class files in author's development environment. In this case, the path to the class library in Microsoft Visual J++ 6.0 must be set properly.

e. Instructions

Since the Java card used has limitation on the Electrically Erasable Programmable Read Only Memory (EEPROM), there are several orders to make the program as efficient as possible to minimize the requirements for the Java operand stack and also to reduce the overall EEPROM the applet instance needs. Each instantiation of an object or array declared with a local scope allocates memory. Each call to the local method allocates memory until all of the Java heap resources are exhausted. For variables, the static and final modifiers are used in the declaration statements so that the variables act as constants. This practice minimizes the applet instance's size and improves performance. Each new variable introduced consumes additional card resources. So, it is better to reuse variables.

Local variables also should not be used frequently since they occupy space in the Java operand stack. Class hierarchy also must be as simple as it can since calling classes within classes takes up program space. It can be done by using class hierarchies in the prototype program, and the hierarchy is compressed before converting the code to an applet. Arguments also were to be used as few as possible. Reviewing of source code is also important since it helps a lot in editing process. Unused variable definitions and operations must be removed. In terms of data types, shorts is more preferred rather than

bytes wherever possible because shorts consume the same amount of card memory as bytes do but offer a wider range of values [7].

f. Creating applet

The main part of this project would be developing the applet. In a simple word, applet is the source code for the Java card program. There are several stages to be taken into account while creating this applet. In the latter, there will be explanation on some of the basic commands which is used in the programming.

The main thing to do of all the Java programming is to import the package classes. By using the import keyword, it will gives access to the methods and definitions of the package's class files [Appendix A2-1]. Then applet is defined as instances of a class extended from java card framework [Appendix A2-2].

The next stage is to define the first constant which is the code of the class (CLA) byte in the command APDU reader. The class byte is actually identifying the applet. The codes for all the instruction (INS) byte needed for the Patient Smart Card program are then declared in the command APDU header. These are also called as instance variables [Appendix A2-3]. Then, the buffer is defined. The main function of buffer is to contain the string data on the card and also holds APDU's for further processing [Appendix A2-4].

At this phase, the private constructor is declared. It is actually the instance of the application class file and it is instantiated by its install method as explain earlier. In the process of developing source code, it is preferable to allocate all the memory that an applet needs during its lifetime inside the constructor [Appendix A2-5].

Those are some of the parts of the constructor. Technically, at the end of constructor part, the instance of the applet is created. The JCRE will then summon the install method as the last step in the applet installation process [Appendix A2-6].

Next, the select method is called by JCRE to inform the card that this applet is selected and to perform the initialization needed to process the following APDU messages. It is very understandable that the two function, install and select are vital to the successful of this source code. In short, the install and select method works together [Appendix A2-7]. Precisely, JCRE will successfully select the applet. After that, it has to dispatch the incoming APDUs to the process method. The JCRE owns and maintains the APDU object, which encapsulates details of the underlying transmission protocol, as specified for TPDU in ISO 7816-3. The APDU object provides a common interface. The process method will dispatch messages to the class methods [Appendix A2-8].

The function is to verify whether the applet can accept this APDU message. If an applet does not catch an exception, the JCRE catches it. The main purpose of the process method is to perform the action the APDU specifies and return an appropriate response to the terminal. The INS byte specifies the type of action to be performed. The basis and core of this project would be storing and reading a string from smart card. This is the concept of the whole project. The vital point is the command function for storing string on the card and also for calling string from the card, respectively [Appendix A2-9].

On the whole, the elements of the system source code and also probably other application using this Open Platform Smart Card are divided uniformly. This is one of the advantages of Java programming. The elements are *import class*, *define applet*, *define constant*, *define buffer*, *develop constructor*, *install method*, *select method*, and *process method*. The constructor part is actually the class or specific function to be designed such as calling and storing string and also PIN verification. It depends on the application itself.

g. Converting Source Code to a Card Application

Upon the completion of developing source code, it has to be converted to a particular file so that it can be used for the card application. In order to compile the source code, any standard Java compiler can be used for compiling and debugging. However, the compiler's class path has to be verified that it is set to the Cyberflex Access class library directory on the host system. The path to the Cyberflex Access library files must be set correctly in the development environment [7].

The source code will then be converted as a class file, with condition there are no syntax error. The next step is to convert the class file into a program file. The program file which puts the load file in a form the JVM can understand. As explained earlier, the SDK provided has included software called Program File Generator (PFG), which it can be used to convert the class file to a program file. Upon the successful of this stage, the file can be downloaded to the card as a load file and finally instantiate the load file before running the application in real time environment.

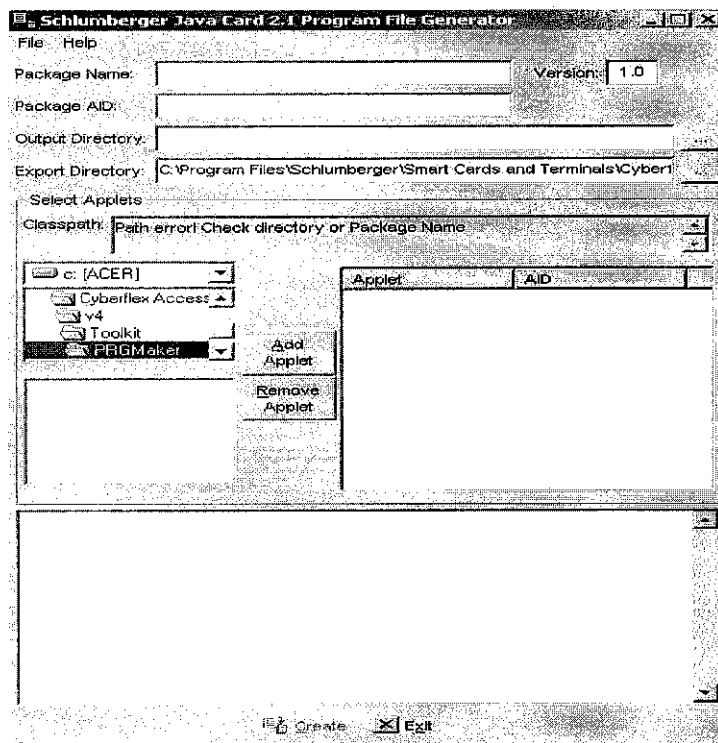


Figure 3.4 Program File Generator (PFG) interface

The Program File Generator dialog box (Figure 3.4) is a graphical user interface for the Sun Converter from Sun Microsystems, Inc. It is added to the host system when you install the Cyberflex Access SDK. There are a few particulars that must be filled in before converting class file into an IJC file.

- a) Package Name – It is the name for the Java package contents and a group of related classes and interfaces within a single directory. The name used for the output file name is the last entry of the Package Name field. This column must be filled in before selecting the directory in which the applet class file is located.
- b) Package Application Identifier (AID) – This column is for the AID needed to be specified for a load file that can be downloaded from the program file. The package AID must be a hexadecimal value 5–16 bytes long. COVE will automatically fill in the package AID from an IJC or CAP file when creating the program file to download the applet to the card. The first 5 bytes of the package AID and applet AID must be identical.
- c) Output Directory – This is where identification of the destination for the output file (the IJC file) must be done.
- d) Export Directory – This directory contains the Java export files the Cyberflex Access SDK installation program adds to the host system. The location for the installed export files will appear automatically in the Export Directory box.

After filling correctly all the information, a class file is selected to add to the program file. This can be done at the Select Applets column. The Add Applet button is clicked right after the desired class file is found. The Applet AID dialog box appears and it must be filled in correctly. As explained earlier, the first 5 bytes of the package AID and the first 5 bytes of the applet AID must be identical. After completed the process, the AID and class file name should appear in the list next to the class navigation pane.

The conversion will begin and continue except if the operation is interrupted. The status of the ongoing operation will appear in the status pane in the PFG dialog box right at the bottom as can be seen in Figure 3.4. Upon the completion of the conversion process, there will be several types of information such as below.

- a) Sun converter version - Version number of the converter included in the Cyberflex Access SDK. This converter produces a program file that can be downloaded to the smart card.
- b) Program file components - Components of the program file with the number of bytes each component occupies.
- c) Number of class files or packages - Number of class files or applets included in the program file and number of packages imported.
- d) Destination of the output file - Complete path to the program file.
- e) Some of errors statement, if occurred. The log files are like JCANormal.txt, JCAError.txt, IJCNormal.txt, and IJCError.txt.

The final step would be to download the Program File as a Load File. The load file contains the byte codes for the class or classes in the program file. The process can be done by either using APDU commands or using Smart Card Toolkit. The Toolkit provided interface can be seen in Figure 3.5 below.

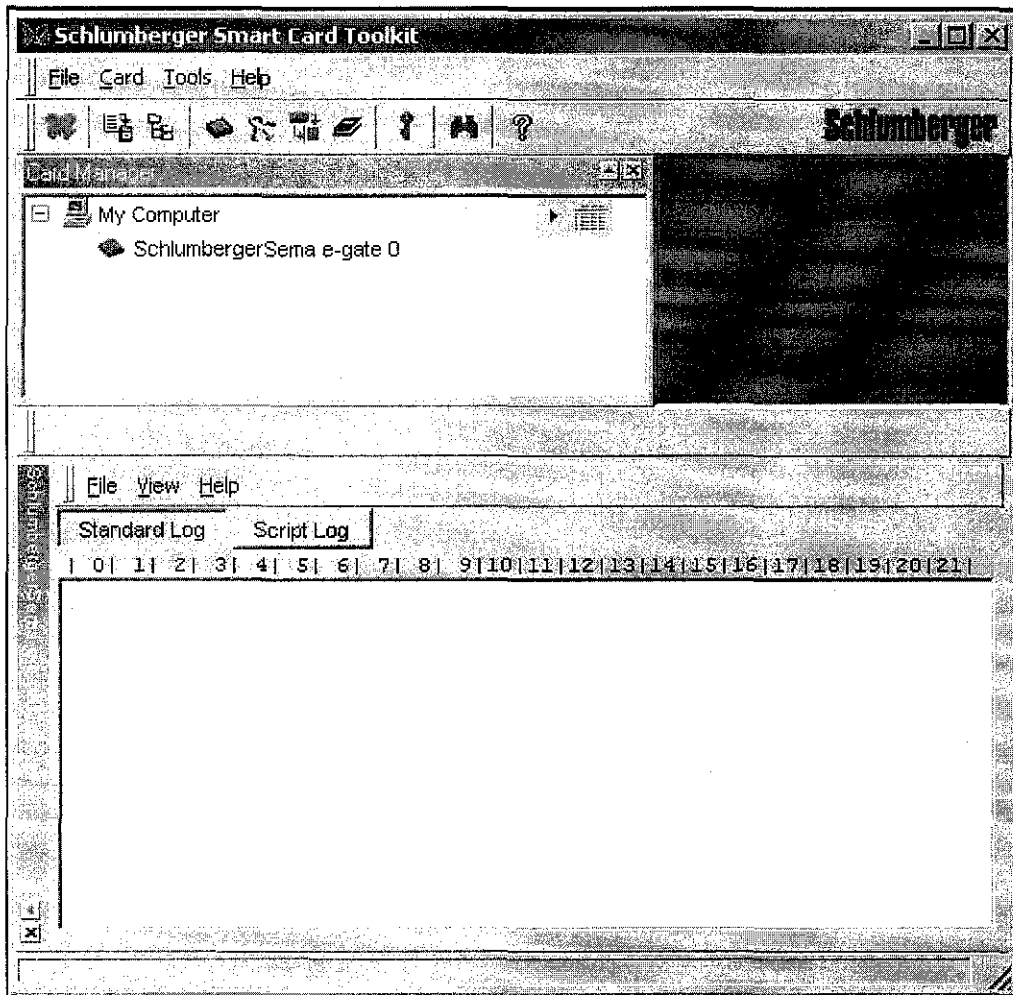


Figure 3.5 Cyberflex Access Toolkit 4.4 interface

In this case it is more comfortable in using the Toolkit provided. The program file can be selected from the Load Files and Libraries folder icon in the Card Manager window. Next, the Create Program dialog box appears, as shown in Figure 3.6.

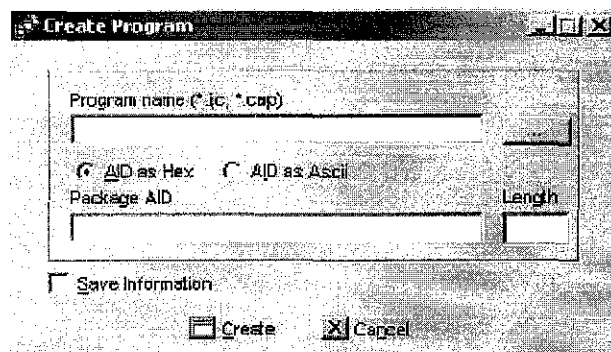


Figure 3.6 Create Program dialog box

In this dialog box, the information for the IJC files that need is desired must be entered.

- a) Program name (*ijc, *cap) box - The IJC file needed for creating the load file is specified. Program files with the IJC extension can be downloaded to all Open Platform Cyberflex Access cards. This is the advantage compared to the CAP extension files.
- b) Package AID box - The Package AID box will automatically display the AID for the load file, an exclusive hexadecimal value 5-16 bytes long. This AID number must match the value entered in the Package AID box of the PFG dialog as explained earlier in this report.
- c) The Length box is just to display the number of AID bytes.
- d) AID as Hex radio button - This button must be checked, which means the AID entered must be in hexadecimal format.
- e) Save Information check box – This box is used to save the information that appears in the Create Program dialog box. The saved information will then appears automatically whenever the dialog box is redisplayed, even in a future card session with a different card inserted. The information remains in the host system database as long as new user overwrites it by again clicking the Create button with the Save information check box selected.

To instantiate the applet, APDU commands can be used or simply by using Smart Card Toolkit. The load file can be instantiated in the Card Manager pane of the main window by applying right click practice. It is as similar as creating the program file. Pop-up menu will then appear as Figure 3.7.

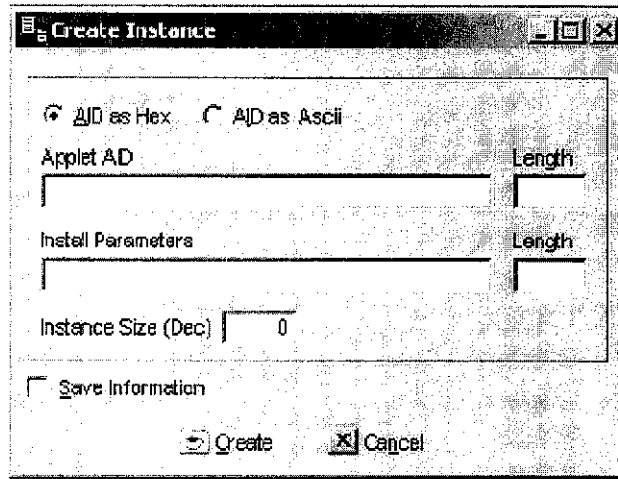


Figure 3.7 Create Instance dialog box

Similarly, the information for the instance needs to be specified.

- a) Applet AID box - The AID must be a unique value 5-16 bytes long and it must match the class AID that was specified in the program file.
- b) The Length box automatically displays the number of AID bytes when the AID is entered.
- c) AID as Hex / AID as ASCII radio buttons - These options is used to specify the format of the AID value (hexadecimal or ASCII). However, this project only use Hex format.
- d) Install Parameters box – This is used to enter any application parameters for the Install command to pass to the applet instance.
- e) Instance Directory Size (Dec) box – This box is used to enter the amount of EEPROM needed for the applet instance and its associated files and headers. The size must be in bytes, expressed in decimal format.

It is vital to ensure the applet designed must have an install method, and the install method must call the register method. If these methods are absent, the card registry has no record of the applet instance, so the card cannot locate the instance to select it or send it commands.

3.1.4 Testing

Testing would be the final stage of the whole process of this project. Basically testing is a process of after compiling and debugging the source code. There are some logical errors, which any compiling machine will not be able to detect it, apart from the syntax error. This is an evaluation period of the project development where run time error testing will be executed to manipulate real time situation.

3.1.5 Data Collection

To ensure the project can adapt with real time purpose, bona fide data and information from several patients are required. Therefore, the feedback with several medical institutions and officials is vital. This is performed in normal procedure where author had to follow all formality, protocols and established contact with all significant personnel. Several hospitals in Ipoh and Batu Gajah were selected to obtain the data required.

3.2 Tools Required

This project should have a reliable prototype and it does not cost high budget. Since it is a smart card application project, therefore it requires some additional hardware such as the card itself and the reader. For the next coding and testing phases, Microsoft Visual Basic 6.0 and Microsoft Access is used as the platform to build the system.

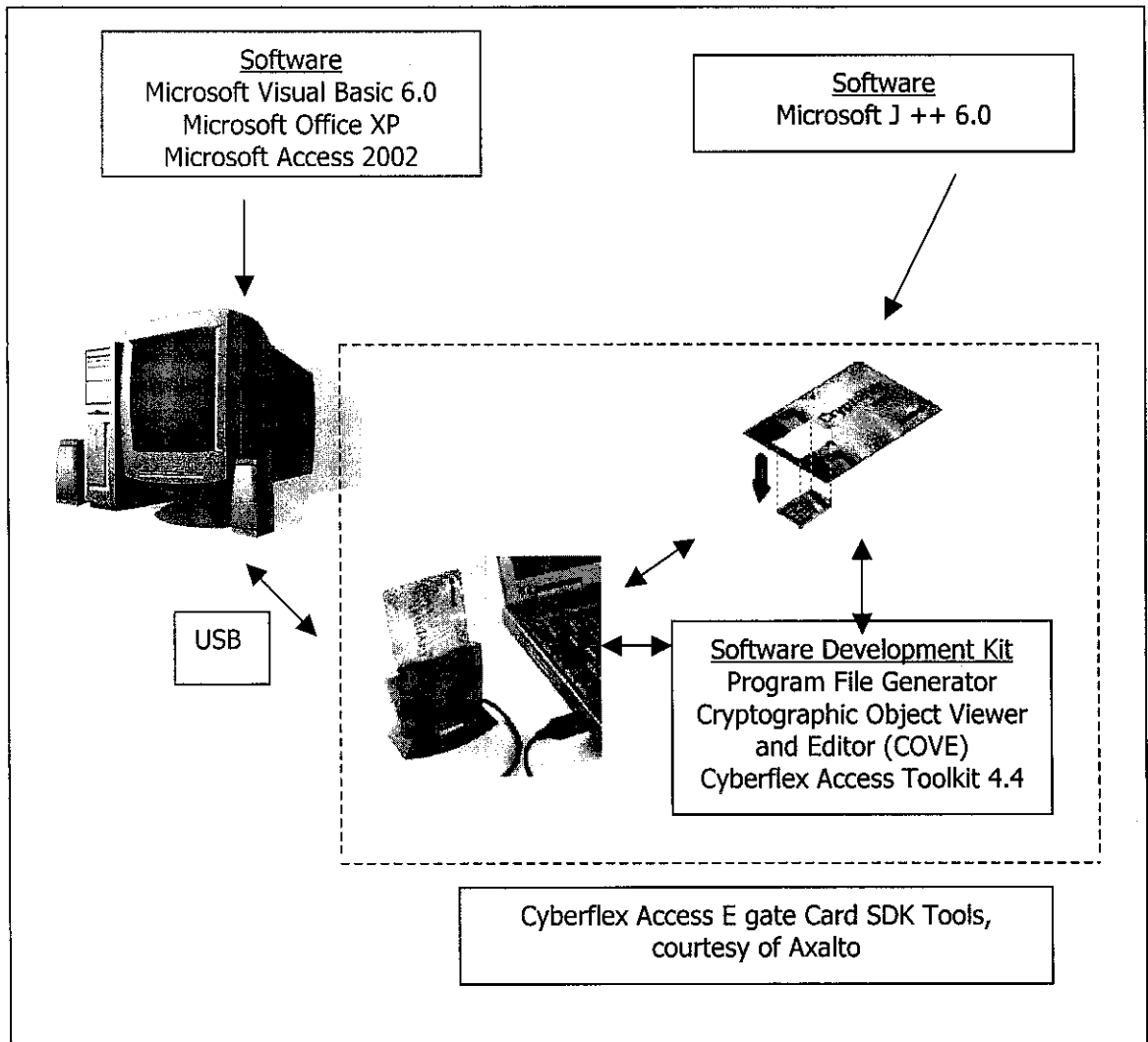


Figure 3.8 Block Diagram of Tools Required

The tools required for developing the system prototype are as Figure 3.8. The software listed is used during the whole process of completing the project especially for most of the source code writing and compiling phase. Cyberflex Access E gate Card Software Development Kit is the core component in this project.

3.2.1 Cyberflex Access E-gate Card Software Development Kit

The implementation of the prototype uses Cyberflex Access E-gate Card Software Development Kit. The kit comprises of the smart card, reader and complete software. Cyberflex Access cards support card programs, or card applets, written in compliance with Java Card 2.1.1 or higher specifications.

Axalto's Cyberflex Access Software Development Kit (SDK) enables user to build smart card-enabled PC applications for multi-function Java-programmable and cryptographic smart cards. It has the ability to perform various applications.

- a) Configuring cards for cryptographic services.
- b) Developing and dynamically loading Java applets.
- c) Integrating custom PC applications.
- d) Smart card enabled security application - digital signature generation and verification, authentication and encryption/decryption.

a. Specifications

Cyberflex Access E-gate Card complies with the ISO 7816 standard. It also has the certification from Entrust, a company with virtual certification business. It has the main features like multi application and very capable Electrically Erasable Programmable Read Only Memory (EEPROM) with 32 Kilobyte storage. It also secured with Cyberflex Java based engine. The body is built with Poly Vinyl Chloride (PVC) and it is tamper-resistant. Below are the complete specifications of the card [11].

Table 3.2 Cyberflex Access E-gate Card Specification

Standard Compliance	Certifications	Main Features	Security	Card Body
ISO 7816 Java Card 2.1.1 Open Platform 2.0.1 PKCS #11 Crypto API	FIPS 140-1 level 2 VOP & Java Card Entrust	Bi-protocol support: T=0 and T=1 Global PIN capability and PIN sharing by application booklet Interoperability thanks to Java Card 2.1.1 Single Power supply from 2.7 V to 5.5 V Temperature range from -25 to 75 degrees C	Extended crypto support: DES, 3-DES, RSA, SHA-1 Encryption and signature support State-of-the-art built-in security features Highly secured Cyberflex Java based engine Powerful applet fire-walling	Embossable with overlay Magnetic Stripe, hologram, and signature panel options available MiFare and HID options available Reader less technology offered with e-gate smart cards USB-V2 compliant

3.2.2 Hardware Used

- a) Processor - Pentium 3 or higher
- b) RAM – 64 Megabyte or higher
- c) Disk space - 1 Gigabyte or higher
- d) Monitor - 14 inch with 1024 x 768 resolution
- e) Graphic card - any standard
- f) Mouse/Keyboard - any standard
- g) Axalto Cyberflex Access Software Development Kit Version 4.4
 - i. Cyberflex Access e-gate 32K card
 - ii. E-gate USB reader

3.2.3 Software Used

- a) Operating System - Windows 98 or higher
- b) Axalto Smart Cards and Terminals complete package
 - i. Cyberflex Access Toolkit V 4.4
 - ii. COVE User and Admin Tool V 4.4
 - iii. Program File Generator
- c) Microsoft Office XP
- d) Microsoft Visual Basic 6.0
- e) Microsoft Access 2002
- f) Microsoft Visual J++ 6.0

CHAPTER 4

RESULTS AND DISCUSSIONS

4. RESULTS AND DISCUSSIONS

This is the most critical part in the project. Research output is presented in this section. The results must meet the requirement of the target audience to fulfil their expectation of a system. Methodology applied within the timeframe of this project should get accurate result. Findings and results during the coding stage since that is the key of the project are discussed.

4.1 Findings

The output for the end product is the system requirements and system functionalities. It mainly engaged with the interfaces, data structure and database tables.

4.1.1 Interface and Data Structure

The interface is designed and developed by using Microsoft Visual Basic 6.0. There are 2 main components in the interface design which author declares them as *system menu* and *data modules*.

a. System Menu

The system starts with the main menu which will lead to four other sub menus, which are *administrator*, *new registration*, *existing use* and *terms and policy*. Each of them has different functionality and interface design.

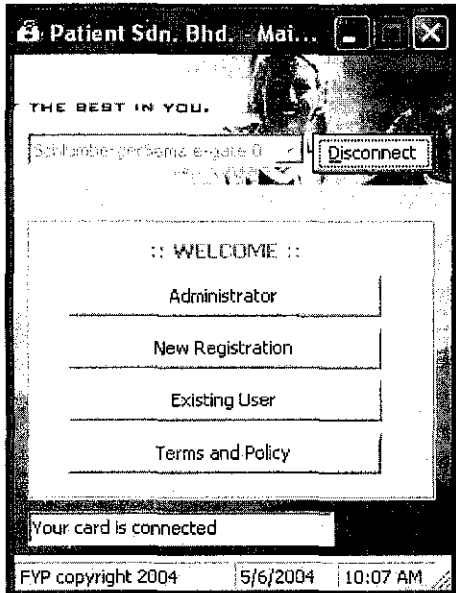


Figure 4.1 System Main Menu

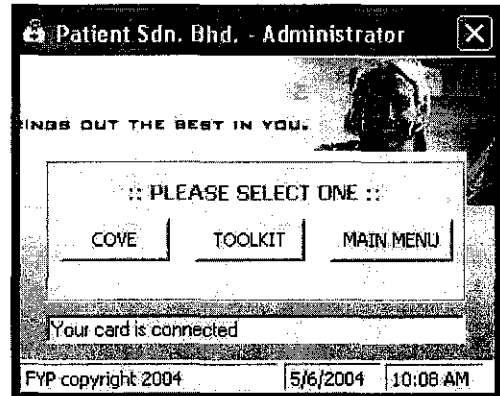


Figure 4.2 Administrator Menu

The menu also contains the connectivity to the smart card and shows the card status. There are also date and time features. The source code is vital in associating the reader, interface and the card itself [Appendix A3-1]. Administrator menu contains the link button for administration personnel to go to smart card internal data contents and to change the core password. There is also button to go back to main menu.

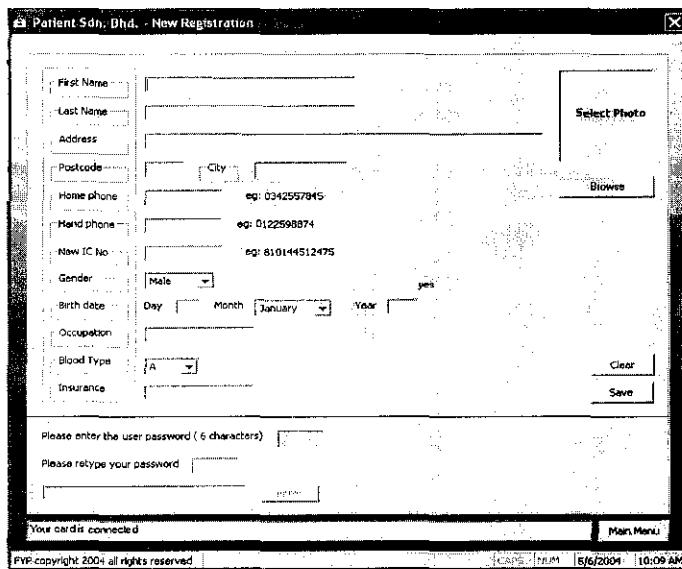


Figure 4.3 Patient New Registration

The form is for a first time user or patient to register at the respective clinic or hospital. It contains all personal details particular information. User also creates password for his or her smart card at the end of registering process.

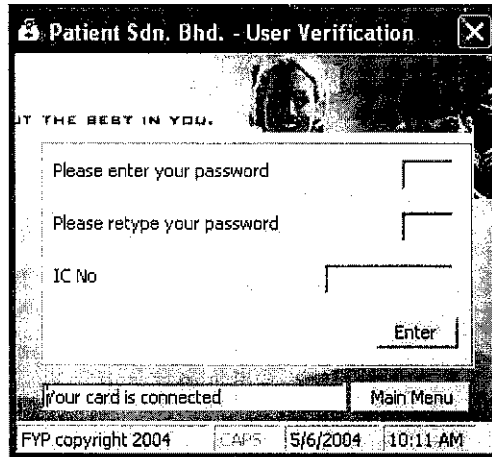


Figure 4.4 Existing User Menu

Existing user can only pass through to the main data system by entering his or her password in this menu. For security reason, the user must also enter the IC number as a verification for both smart card and database. Password can be tried three times else the system will discontinue. If IC number and password entered mismatch, the system will also halt [Appendix A3-2].

b. Data Modules

All data for the system is placed in only one form. By applying Microsoft Tabbed Control 6.0 for this form, data structure is arranged in a proper way besides having it looks more real in sense of application wise. Moreover, it reduces the possibility of forms disordered form.

There are eight folders to differentiate and classify all patients' data. The type of information selected is totally based on author's last semester research on system comprehensive data. Thus, for this semester, it is just a matter of designing the

interface and segregating all information using Microsoft Visual Basic 6.0 [Appendix A3-3].

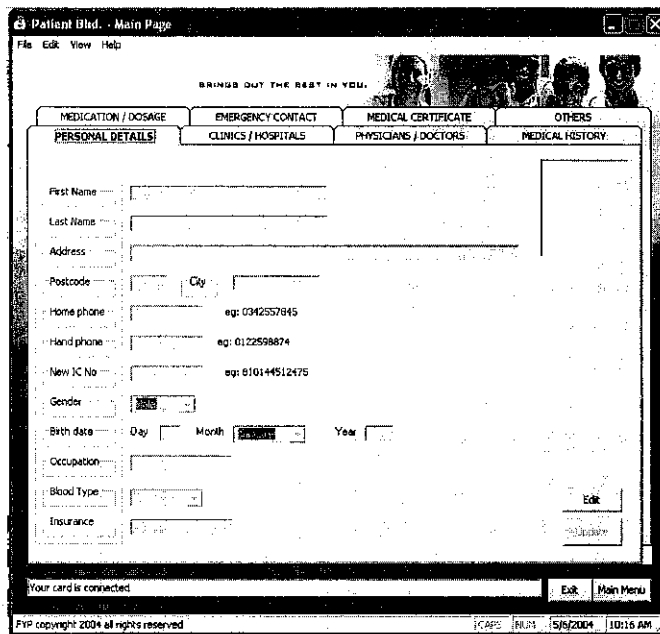


Figure 4.5 Personal Details Folder

Personal details folder contains complete information and bio data about patient. There is also basic information on patient’s emergency contact person for any urgency reason.

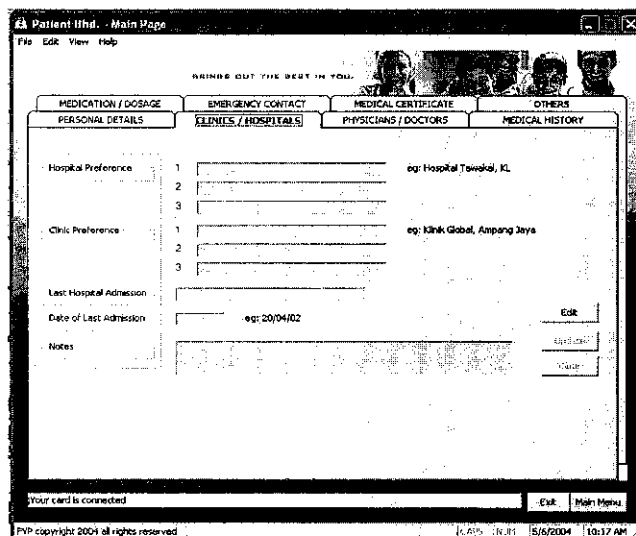


Figure 4.6 Previous Medical Institution Folder

This folder contains list of hospitals and clinics preferred and previously visited by the respective patient, location and date of last admission to any medical institution.

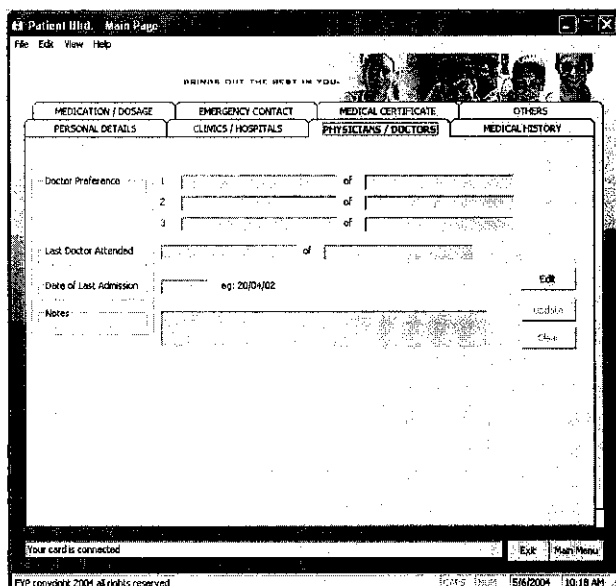


Figure 4.7 Doctors Preferences Folder

This folder contains the list of doctors preferred by the respective patient, last doctor attended the patient and the date of the occasion.

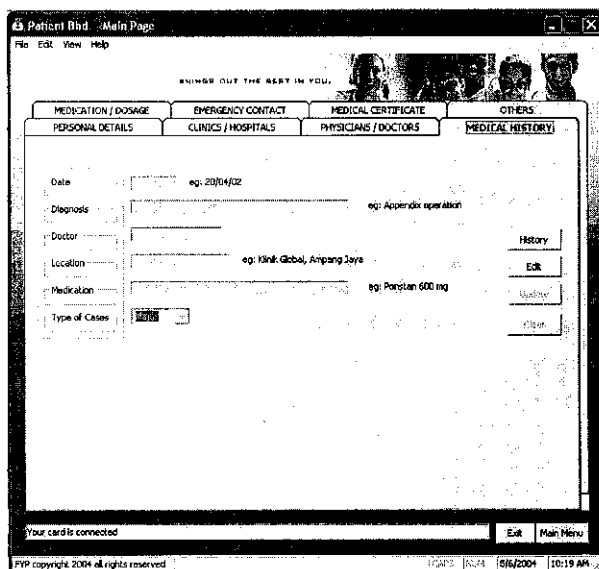


Figure 4.8 Medical History Folder

This folder contains the detail about all patient major or minor medical cases in the past. It includes the diagnosis, date of cases, medication given and location of treatment. There is a history button to see the archive.

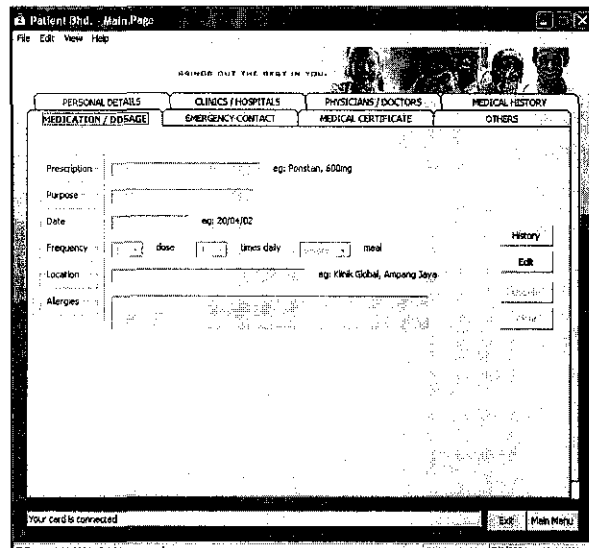


Figure 4.9 Medication Folder

This folder contains all previous medicine or drugs prescription for the patient. The details are like the regularity of the medicine, purpose, location and date of the treatment. There is also a column for patient's allergies declaration. There is a history button to see the archive.

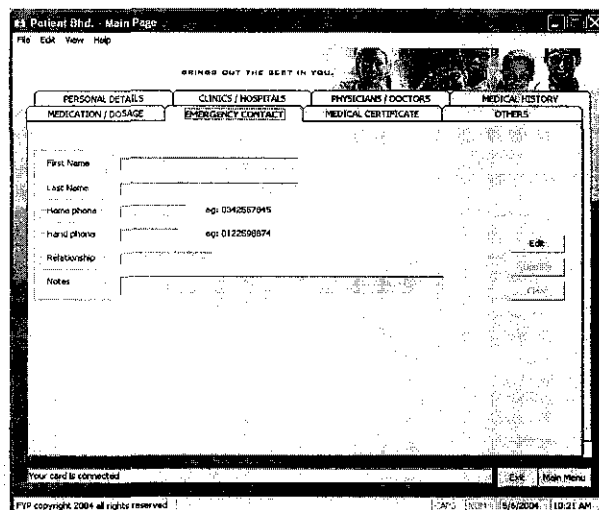


Figure 4.10 Emergency Contact Folder

This folder contains the basic information of patient's emergency contact person in case of emergency.

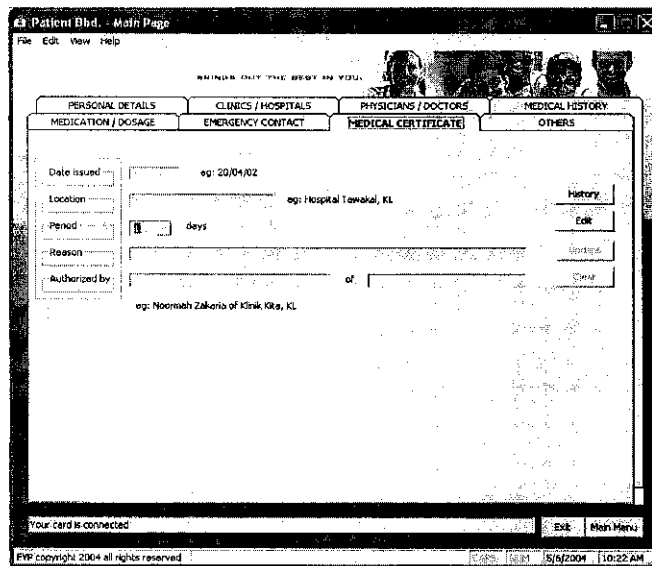


Figure 4.11 Medical Certificate Folder

This folder contains the patient's history of all medical certificate (MC) applied at the respective medical institution. There is a history button to see the archive.

Period	Types of Vaccine	Suitable Period	Status
Infants	B.C.G	at birth	<input type="checkbox"/>
	1st Hepatitis B Vaccine	at birth	<input type="checkbox"/>
	2nd Hepatitis B Vaccine	1 month	<input type="checkbox"/>
	3rd Hepatitis B Vaccine	6th month	<input type="checkbox"/>
	1st Triple Antigen, Polio + Hib	3rd month	<input type="checkbox"/>
	2nd Triple Antigen, Polio + Hib	4th month	<input type="checkbox"/>
Children	3rd Triple Antigen, Polio + Hib	6th month	<input type="checkbox"/>
	Measles MMR	6th month	<input type="checkbox"/>
	1st Booster, Polio	6th month	<input type="checkbox"/>
School Children	2nd Booster, Polio	6th month	<input type="checkbox"/>
	2nd Booster Dose Triple Antigen, Polio B.C.G (if no scars)	7 yrs	<input type="checkbox"/>
	B.C.G	12 yrs	<input type="checkbox"/>
	Tetanus Toxoid	15 yrs	<input type="checkbox"/>
Pregnant Mother	Rubella (female only)	13 yrs - 17 yrs	<input type="checkbox"/>
	1st Dose Tetanus Toxoid	24 - 28 weeks of pregnancy	<input type="checkbox"/>
	2nd Dose Tetanus Toxoid	6 - 8 weeks after first dose	<input type="checkbox"/>
	1 Dose Tetanus Toxoid	at every following pregnancy	<input type="checkbox"/>

Figure 4.12 Immunization List Folder

This folder contains complete list of immunization that should be taken in Malaysia for both gender at a particular stage of age and also for pregnant mother. The list is as stated by Ministry of Health Department.

Other forms in the system are the archives of the related folder and information provider folder, which contains the names of authorized personnel who can modify or view the patient's data. It is for security reason and further references.

4.1.2 Database

The database is designed by using Microsoft Access 2002. There are ten tables responding to the data modules interface designed (Figure 4.13). The database function is simply to store and retrieve patient's data depending on the task performed on the interface. The database is linked to the interface by applying Microsoft ADO Data Control 6.0 (OLEDB).

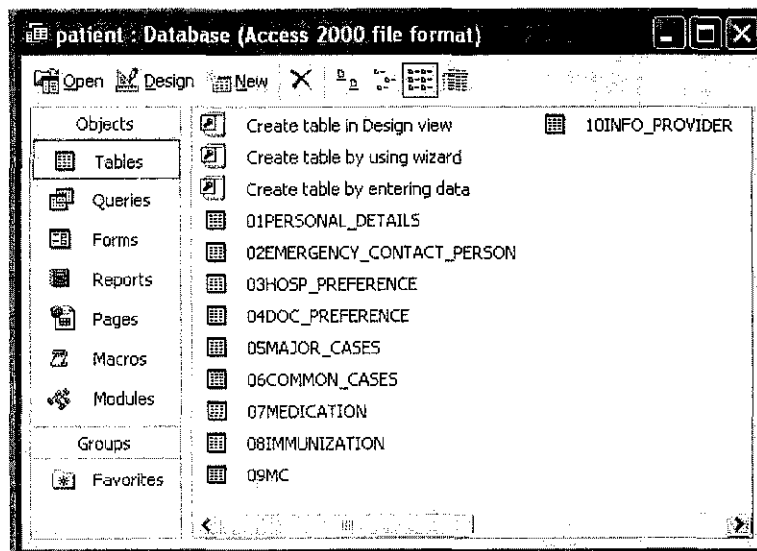


Figure 4.13 List of Database Tables

There will be three tables of database included inside the interface for the purpose of viewing and monitoring by the authorized personnel. They are for the Medical History, Medication and Medical Certificate. These three particulars are linked

using project component Microsoft Datagrid Control 6.0 (OLEDB). The particulars are selected because they are vital information and in need of quick and firm response in case of emergency. Thus, authorized personnel don't have to access the database system in a whole and finding the data needed would be very time consuming.

4.1.3 Bona Fide Existing Patients Data

The next stage after coding would be testing. In order to perform this phase, there must be a real data on patient's information. The purpose is to execute author's system with the mock data to fulfil the demand of task during real time application.

Therefore, author collected the data desired by establishing contact with several medical institutions (Appendix A4). Author's objective was to collect a number of random patient information to be analysed and inserted in the database by using the designed system. There are protocols and regulations in order to obtain those data. Below is the list of medical institution which author had encountered.

- a) Jabatan Kesihatan Negeri Perak.
- b) Hospital Besar Batu Gajah.
- c) Hospital Changkat Melintang Bota.
- d) Hospital Besar Ipoh

4.2 Discussions

4.2.1 Overall Implementation

In term of data structure, author progress is very much in line with last semester proposal [2] and suggested content as in Table 4.1 below. The contents of Patient Smart Card is divided into five main areas which are the past medical centre or pharmacy attended, history of doctors whom attended the patient, pervious major

treatment given to patient, current and history of medication and dosage and patient's insurance or financial background.

Table 4.1 Suggestion of the Preliminary Contents of the Smart Card [2]

<i>Data/ Contents</i>	Past Pharmacy or Medical Centre	History of Physicians or Doctors	Previous Major Treatment	Medication and Dosage	Financial and Insurance Background
Transfer	E-admission and sign in	Reports/ recommendation	Related results and history	E-prescription	-
Constant	Privileges	-	-	Allergies	Validity
Dynamic	General reports and recommendations by certified personnel				
Personal files	Self-preference	Self-preference or private doctors	Diagnostic findings	-	To be updated and verified

However, the financial and insurance background is omitted from a single form. This is because the priority of this project is to concentrate on patient's data on medical information. Furthermore, author did not focus on the type of data, which means the whole system information is at the same priority level.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5. CONCLUSIONS AND RECOMMENDATIONS

As a conclusion, this project is a success since it achieves the four objectives stated earlier. Gradually, author can now make the project as significant as it can be and the relevance of the matter towards engineering prospect are becoming understandable and apparent.

5.1 Relevancy to the Objectives

The first objective is to complete the source code development for both Java card and interface software. This is completed by doing some modification and revision from the source provided in SDK provided. The accomplishment leads to the testing phase besides establishing the connectivity with databases of the system.

The second objective is to design the data structure and interface coordination appropriately. This is the most crucial part since this act as a synergy between user and the system to interact and it is successfully achieved.

The third objective is to design smart card that is able to store data. This is in concurrence with the first objective but the key is for the card to store PIN number and basic information for the system to retrieve.

The fourth objective is to gather information and perform case study on several existent patients to conjure up with the project. Upon this succession, it leads to the

testing process which can detect run time error. This is vital consecutively to manipulate real time application.

5.2 Expectation of the System

The system should provides data that allow justification of medical decisions, high level of trust in data precision and short time of data interpretation. Moreover, Patient Smart Card ensures the basic functions to act in emergency cases. In addition, the card provides mutual information among the doctors who are implied in the long-term treatment of the patient.

Consequently, author expects this project to perform the above tasks. Upon the completion of the project, author also has set certain long term target for the smart card.

- a) The card must be able to issue all personal data registration and store more data.
- b) Able to do data modification and updating or eventually to the cancellation of card.
- c) The needs of high-level of data security are a priority besides maintaining its confidentiality and credibility.

5.3 Problems Encountered

Throughout the process of developing this system, author encountered some minor problems which affecting the flow of the progress. Nevertheless, these setbacks managed to be minimized without much trouble. The first problem is the minimum number of technical expertise in Open Platform smart card using Java programming in author's university. However, author has been able to overcome it by learning process and travel to places to gain knowledge. The learning process took a period of time since author background is not very much of IT literate.

Communication also is a problem during the process of gathering real patients' data. This is because author has to go to many levels of authorities before getting the exact information wanted. The protocols and level of management seems to be the main factor of this drawback. However, author coped with the situation well and obtained the data as expected but in a longer period.

From the technical aspect, author encountered a problem on the card memory size. There is no identification for the exact sufficiency of the card. There is an infinite value of the card ability to save string bytes. Thus, it is assumed that it can't deal with the system since this project involves with many databases. For this reason, author has to experiment to determine the total amount of EEPROM the program requires.

5.4 Suggested Future Work for Expansion and Continuation

The author hopes that this project will be a major success, useful and practical to be used in the future. In the meanwhile, the author focuses on to cater the user for a user-friendly system and also the design constraints.

5.4.1 Development of the System

- a) Improvisation in terms of its interface to ease user interaction.
- b) Usage of a more stable database to adapt with real time environment.
- c) Clarification of the definite memory size of the card.

5.4.2 Application-wise

- a) Doctors also will be provided with special cards which give the write access on the patient card, in the area reserved for the specialist doctors.
- b) Security is guaranteed with more identifier such as biometrics and image scanning during patient authentication.
- c) Detailed insurance coverage that can be used directly during medical operation.

- d) The application is standardized for all institutions and it has the continuity with tele-medicine system and other up-to-the-minute gadgets.
- e) The smart card has the ability to store all data in turn to be responsive to other medical institution so that it can be shared dynamically throughout the country.
- f) The application has a portable reader complete with the interface software to be more practical during emergency cases.

As a conclusion, the author hopes that this project meets the expectation and successful in achieving the long terms objective of the project. The exposure and experience gained throughout the project will be a very good base for working environment. Certainly, the knowledge and benefits gained will be implied in daily life.

REFERENCES

Books

1. **C. Thomas Wu** 2001, *Introduction to Object Oriented Programming with Java 2nd Edition*, McGraw Hill.
2. **Fairus bin Hariri** 2003, *Final Year Project I Final Report*, Universiti Teknologi PETRONAS.
3. **Jeffrey Whitten** 2001, *System Analysis and Design Methods 5th Edition*, McGraw Hill.
4. **John Smiley** 2002, *Learn to Program with Java*, McGraw-Hill/Osborne.
5. **Mike Hendry** 1997, *Smart Card Security and Applications*, Arctech House, Inc.
6. **Uwe Hansmann** 1999, *Smart Card Application Development Using Java*, Springer.
7. **SchlumbergerSema**, Cyberflex Access Software Development Kit 4.4, Programmer's Guide.
8. **Bit magazine** June 1998.

Websites

9. **Axalto** 2003, <http://www.axalto.com>
10. **Card Logix, Towitoko and Fargo Corporation** 1999, *Smart Card Basics*, <http://www.smartcardbasics.com>
11. **Cyberflex Access Home Page** 2003, <http://www.cyberflex.com>
12. **Giesecke & Devrient** 2003, <http://www.gi-dde.com/eng/main/home/index.php4>
13. **Java Technology** 2003, <http://www.java.sun.com>
14. **Middleeast medical** 2003, <http://www.middleeastmedical.com/News.asp?Article=249>
15. **Pace Integration** 2003, <http://www.paceintegration.com>

APPENDICES

A1. Gantt chart

A2. Java Applet Source Code

A3. Microsoft Visual Basic 6.0 – Interface Source Code

A3-1: Main Menu

A3-2: User Verification

A3-3: Data Module

A4. Cover Letters

FINAL YEAR PROJECT (EEB 5034) GANTT CHART

APPENDIX A1

ACTIVITIES	WEEKLY REVIEW																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
<i>Preliminary design</i>																		
Theoretical Learning (VB, Database)																		
Data structure																		
<i>Coding</i>																		
Creating Database																		
Link between interface and database																		
Java applet																		
<i>testing</i>																		
run time error																		
<i>Report writing</i>																		
Progress report 1																		
Progress report 2																		
Draft Report																		
Final Report (soft cover)																		
Extended Abstracts																		
Final Report (hard cover)																		
<i>Presentation</i>																		

Student's Name: Fairus bin Hariri

Student's id #: 1533

APPENDIX A2 - Java Applet Source Code

A2-1

```
import javacard.framework.*;
import javacard.security.*;
```

A2-2

```
public class Patient extends javacard.framework.Applet {
```

A2-3

```
final static byte Patient_CLA = (byte)0x85;
```

A2-4

```
byte TheBuffer[];
```

A2-5

```
private Patient(byte buffer[],short offset,byte length) {
    TheBuffer = new byte[100];
    if (buffer[offset] == (byte)0) {
        register();
    }
    else {
        register(buffer, (short)(offset+1), (byte)(buffer[offset]));
    }
}
```

A2-6

```
public static void install(byte buffer[],short offset,byte length) {
    new Patient(buffer, offset, length);
}
```

A2-7

```
public boolean select() {
    return true;
}
```

A2-8

```
public void process(APDU apdu) throws ISOException{
    byte buffer[] = apdu.getBuffer();
    if (selectingApplet()) {
        ISOException.throwIt(ISO7816.SW_NO_ERROR);
    }
    if (buffer[ISO7816.OFFSET_CLA] != SimpleString_CLA)
        ISOException.throwIt(ISO7816.SW_CLA_NOT_SUPPORTED);
    byte ins = buffer[ISO7816.OFFSET_INS];
    switch (ins) {
        case SET:
            SetString(apdu);
    }
}
```

```

break;
case GET:
GetString(apdu);
break;
default:
ISOException.throwIt(ISO7816.SW_INS_NOT_SUPPORTED);
}
}

```

A2-9

```

private void SetString(APDU apdu) {
byte buffer[] = apdu.getBuffer();
byte size = (byte)(apdu.setIncomingAndReceive());
byte index;

private void GetString(APDU apdu) {
byte buffer[] = apdu.getBuffer();
byte numBytes = buffer[ISO7816.OFFSET_LC];
if (numBytes == (byte)0) {
ISOException.throwIt((short)(0x6200 + TheBuffer[0]));
}
apdu.setOutgoing();
apdu.setOutgoingLength(numBytes);
byte index;
for (index = 0; index <= numBytes; index++)
buffer[index] = TheBuffer[(byte)(index + 1)];
apdu.sendBytes((short)0, (short)numBytes);
return;
}
}

```

APPENDIX A3 - Microsoft Visual Basic 6.0 – Interface Source Code

A3-1: Main menu

Option Explicit

```
Private WithEvents m_iopSys As IOPCOM3Lib.IOPSystem3
Private m_iopCard As IOPCOM3Lib.IOPCard3
Private Sub cmdAdmin_Click()
```

```
frmMainmenu.Hide
frmAdmin.Show
```

```
End Sub
```

```
Private Sub cmdBack_Click()
```

```
Me.Hide
```

```
Private Sub m_iopSys_CardRemoved(ByVal ReaderName As String)
If cmdConnect.Caption <> "Connect" Then
    cmbReaders.Enabled = True

    cmdConnect.Caption = "Connect"
    cmdConnect.ToolTipText = "Click to connect to selected reader"

    txtStatus.Text = "Your card has been disconnected"

    Set m_iopCard = Nothing
End If
End Sub
```

A3-2: User Verification

```
' Get string from card callback
If txtPass.Text = txtPassche.Text Then
```

```
Dim CLA As Byte
Dim INS As Byte
Dim P1 As Byte
Dim P2 As Byte
Dim byteArray() As Byte
Dim emptyarray() As Byte
Dim iStatus As Integer
Dim bResult As Boolean
```

```
Dim byteLength As Byte
Dim sString As String
```

```
CLA = &H85
INS = &H20
P1 = 0
P2 = 0
```

```
'If IsNumeric(txtLength.Text) And Val(txtLength.Text) > 0 Then
'If Len(Trim(txtPass.Text)) <> 0 Then
```

```
If Len(Trim(txtPass.Text)) = 6 Then
```

```
byteLength = 6 * Len(Trim(txtPass.Text)) * txtLength.Text
bResult = m_iopCard.SendAPDU(CLA, INS, P1, P2, emptyarray(), byteLength, byteArray(),
iStatus)
```

```
If iStatus = &H9000 Then
sString = StrConv(byteArray, vbUnicode)
txtGet.Text = sString
If sString = Trim(txtPass.Text) Then
txtStatus.Text = "Thank you" & "Returned string from card"
```

```
Adoper.Recordset.MoveFirst 'make sure row position at the beginning
Do While Not Adoper.Recordset.EOF 'loop all rows
If txtChic.Text = Adoper.Recordset.Fields("ic no").Value Then
```

```
frmAll.Adocli.Recordset.MoveFirst 'make sure row position at the beginning
Do While Not frmAll.Adocli.Recordset.EOF 'loop all rows
If txtChic.Text = frmAll.Adocli.Recordset.Fields("ic no").Value Then
Exit Do
End If
frmAll.Adocli.Recordset.MoveNext
Loop
```

```
frmAll.Show
frmAll.Command1.Enabled = False
frmAll.Command3.Enabled = False
frmAll.Command5.Enabled = False
frmAll.Command8.Enabled = False
frmAll.Command24.Enabled = False
frmAll.Command13.Enabled = False
frmProvider.Command10.Enabled = False
frmImmune.Command10.Enabled = False
Unload Me
```

```
Exit Sub
End If
Adoper.Recordset.MoveNext
Loop
txtStatus.Text = "Please enter your IC number."
```

```
MsgBox "Sorry, data not found", vbCritical, "Error message"  
txtChic.Text = ""
```

```
Else
```

```
    intc = intc + 1
```

```
    If intc = 3 Then
```

```
        intc = 0
```

```
        cmdConnect_Click
```

```
    End If
```

```
    txtStatus.Text = "Sorry, please try again"
```

```
    txtPassche.Text = ""
```

```
    txtPass.Text = ""
```

```
End If
```

```
Else
```

```
    txtStatus.Text = "SendAPDU " + m_iopCard.LastError.Description
```

```
End If
```

```
Else
```

```
    txtStatus.Text = "Please enter 6 characters password."
```

```
    txtPassche.Text = ""
```

```
    txtPass.Text = ""
```

```
End If
```

```
Else
```

```
    txtStatus.Text = "Sorry, the password mismatch"
```

```
    txtPassche.Text = ""
```

```
    txtPass.Text = ""
```

```
End If
```

```
End Sub
```

A3-3: Data Module

```
Private Sub Command10_Click()
```

```
    Adocli.Recordset.MoveFirst 'make sure row position at the beginning
```

```
    Do While Not Adoper.Recordset.EOF 'loop all rows
```

```
    If txtIc2.Text = Adocli.Recordset.Fields("ic no").Value Then
```

```
        Adocli.Recordset.Fields("hospital1").Value = Trim(txtHos1.Text)
```

```
        Adocli.Recordset.Fields("hospital2").Value = Trim(txtHos2.Text)
```

```
        Adocli.Recordset.Fields("hospital3").Value = Trim(txtHos3.Text)
```

```
        Adocli.Recordset.Fields("clinic1").Value = Trim(txtCli1.Text)
```

```
        Adocli.Recordset.Fields("clinic2").Value = Trim(txtCli2.Text)
```

```
        Adocli.Recordset.Fields("clinic3").Value = Trim(txtCli3.Text)
```

```
        Adocli.Recordset.Fields("last admission place").Value = Trim(txtLasth.Text)
```

```
        Adocli.Recordset.Fields("last admission date").Value = Trim(txtLastd.Text)
```

```
        Adocli.Recordset.Fields("notes").Value = Trim(txtNotes1.Text)
```



```
Adocli.Recordset.Save
```

```
MsgBox "Data updated successfully", vbOKOnly, "Confirmation"
```

```
cmdEdit2.Enabled = False
```

```
Exit Do
```

```
Else
```

```
Adocli.Recordset.MoveNext
```

```
End If
```

```
Loop
```

```
End Sub
```

```
Private Sub Command11_Click()
```

```
'If Len(Trim(txtDoc1.Text)) = 0 Then
```

```
'MsgBox "Please enter at least 1 doctor's name", vbCritical, "Error Message"
```

```
'Else
```

```
Adophy.Recordset.MoveFirst 'make sure row position at the beginning
```

```
Do While Not Adophy.Recordset.EOF 'loop all rows
```

```
    If txtIc3.Text = Adophy.Recordset.Fields("ic no").Value Then
```

```
        Adophy.Recordset.Fields("doctor1").Value = Trim(txtDoc1.Text)
```

```
        Adophy.Recordset.Fields("doc1 place").Value = Trim(txtDoc2.Text)
```

```
        Adophy.Recordset.Fields("doctor2").Value = Trim(txtDoc3.Text)
```

```
        Adophy.Recordset.Fields("doc2 place").Value = Trim(txtDocp1.Text)
```

```
        Adophy.Recordset.Fields("doctor3").Value = Trim(txtDocp2.Text)
```

```
        Adophy.Recordset.Fields("doc3 place").Value = Trim(txtDocp3.Text)
```

```
        Adophy.Recordset.Fields("last doctor attended").Value = Trim(txtLastdoc.Text)
```

```
        Adophy.Recordset.Fields("last doctor place").Value = Trim(txtLastdocp.Text)
```

```
        Adophy.Recordset.Fields("date").Value = Trim(txtLastdate.Text)
```

```
        Adophy.Recordset.Fields("notes").Value = Trim(txtNotes2.Text)
```

```
        Adophy.Recordset.Save
```

```
    MsgBox "Data updated successfully", vbOKOnly, "Confirmation"
```

```
    cmdEdit3.Enabled = False
```

```
    Exit Do
```

```
    Else
```

```
        Adophy.Recordset.MoveNext
```

```
    End If
```

```
    Loop
```

```
Loop
```

```
End Sub
```

```
Private Sub Command13_Click()
```

```
txtDatemc.Text = ""
```

```
txtLocmc.Text = ""
```

```
txtReasonmc.Text = ""
```

```
txtAuthormc.Text = ""  
txtAuthormcloc.Text = ""
```

```
End Sub
```

```
Private Sub Command14_Click()
```

```
frmHistory.Show
```

```
End Sub
```

```
Private Sub Command15_Click()
```

```
'Adoper.Recordset.MoveFirst 'make sure row position at the beginning  
' Do While Not Adoper.Recordset.EOF 'loop all rows
```

```
,
```

```
'If txtIc.Text = Adoper.Recordset.Fields("ic no").Value Then
```

```
,
```

```
'Adoper.Recordset.Fields("first name").Value = Trim(txtFirst.Text)
```

```
'Adoper.Recordset.Fields("last name").Value = Trim(txtLast.Text)
```

```
'Adoper.Recordset.Fields("address").Value = Trim(txtAdd.Text)
```

```
'Adoper.Recordset.Fields("post code").Value = Val(Trim(txtCode.Text))
```

```
'Adoper.Recordset.Fields("city").Value = Trim(txtCity.Text)
```

```
'Adoper.Recordset.Fields("home phone").Value = Trim(txtHome.Text)
```

```
'Adoper.Recordset.Fields("hand phone").Value = Trim(txtHand.Text)
```

```
'Adoper.Recordset.Fields("ic no").Value = Trim(txtIc.Text)
```

```
'Adoper.Recordset.Fields("gender").Value = Trim(cmbGender.Text)
```

```
'Adoper.Recordset.Fields("birth day").Value = Val(Trim(txtDay.Text))
```

```
'Adoper.Recordset.Fields("birth month").Value = Trim(cmbMonth.Text)
```

```
'Adoper.Recordset.Fields("birth year").Value = Val(Trim(txtYear.Text))
```

```
'Adoper.Recordset.Fields("occupation").Value = Trim(txtOcc.Text)
```

```
'Adoper.Recordset.Fields("blood type").Value = Trim(cmbBlood.Text)
```

```
'Adoper.Recordset.Fields("insurance").Value = Trim(txtInsure.Text)
```

```
'Adoper.Recordset.Update
```

```
'MsgBox "Data updated successfully", vbOKOnly, "Confirmation"
```

```
'cmdEdit1.Enabled = False
```

```
'Exit Do
```

```
'Else
```

```
'Adoper.Recordset.MoveNext
```

```
'End If
```

```
'Loop
```

```
If Len(Trim(txtDatemed.Text)) = 0 Or Len(Trim(txtLocmed.Text)) Then
```

```
MsgBox "Please fill in all particular", vbCritical, "Error Message"
```

```
End If
```

```
If cmbTypec.Text = "major" Then
```

```
Adomaj.Recordset.AddNew
```

```
Adomaj.Recordset.Fields("date").Value = Trim(txtDatemed.Text)
```

```
Adomaj.Recordset.Fields("diagnosis").Value = Trim(txtDiag.Text)
```

```
Adomaj.Recordset.Fields("doctor").Value = Trim(txtDocmed.Text)
```

```
Adomaj.Recordset.Fields("location").Value = Trim(txtLocmed.Text)
```

```
Adomaj.Recordset.Fields("medication").Value = Trim(txtMedet.Text)
```

```
Adomaj.Recordset.Save
```

```
Else
```

```
Adocom.Recordset.AddNew
```

```
Adocom.Recordset.Fields("date").Value = Trim(txtDatemed.Text)
```

```
Adocom.Recordset.Fields("diagnosis").Value = Trim(txtDiag.Text)
```

```
Adocom.Recordset.Fields("doctor").Value = Trim(txtDocmed.Text)
```

```
Adocom.Recordset.Fields("location").Value = Trim(txtLocmed.Text)
```

```
Adocom.Recordset.Fields("medication").Value = Trim(txtMedet.Text)
```

```
Adocom.Recordset.Save
```

```
End If
```

```
End Sub
```

```
Private Sub Command16_Click()
```

```
'Adoper.Recordset.MoveFirst 'make sure row position at the beginning
```

```
' Do While Not Adoper.Recordset.EOF 'loop all rows
```

```
,
```

```
'If txtIc.Text = Adoper.Recordset.Fields("ic no").Value Then
```

```
,
```

```
'Adoper.Recordset.Fields("first name").Value = Trim(txtFirst.Text)
```

```
'Adoper.Recordset.Fields("last name").Value = Trim(txtLast.Text)
```

```
'Adoper.Recordset.Fields("address").Value = Trim(txtAdd.Text)
```

```
'Adoper.Recordset.Fields("post code").Value = Val(Trim(txtCode.Text))
```

```
'Adoper.Recordset.Fields("city").Value = Trim(txtCity.Text)
```

```
'Adoper.Recordset.Fields("home phone").Value = Trim(txtHome.Text)
```

```
'Adoper.Recordset.Fields("hand phone").Value = Trim(txtHand.Text)
```

```
'Adoper.Recordset.Fields("ic no").Value = Trim(txtIc.Text)
```

```
'Adoper.Recordset.Fields("gender").Value = Trim(cmbGender.Text)
```

```
'Adoper.Recordset.Fields("birth day").Value = Val(Trim(txtDay.Text))
```

```
'Adoper.Recordset.Fields("birth month").Value = Trim(cmbMonth.Text)
```

```
'Adoper.Recordset.Fields("birth year").Value = Val(Trim(txtYear.Text))
```

```
'Adoper.Recordset.Fields("occupation").Value = Trim(txtOcc.Text)
```

```
'Adoper.Recordset.Fields("blood type").Value = Trim(cmbBlood.Text)
```

```
'Adoper.Recordset.Fields("insurance").Value = Trim(txtInsure.Text)
```

```
'Adoper.Recordset.Update
```

```
'MsgBox "Data updated successfully", vbOKOnly, "Confirmation"
```

```
'cmdEdit1.Enabled = False
```

```
'Exit Do
```

```
'Else
```

```
'Adoper.Recordset.MoveNext
```

```
'End If
```

```
'Loop
```

```
Adomed.Recordset.MoveFirst 'make sure row position at the beginning
```

```
Do While Not Adomed.Recordset.EOF 'loop all rows
```

```
If txtIc.Text = Adomed.Recordset.Fields("ic no").Value Then
```

```
    If Len(Trim(txtPrescrip.Text)) = 0 Or Len(Trim(txtDatedos.Text)) = 0 Then
```

```
        MsgBox "Please fill in all particular", vbCritical, "Error Message"
```

```
    Else
```

```
Adomed.Recordset.Fields("prescription").Value = Trim(txtPrescrip.Text)
Adomed.Recordset.Fields("purpose").Value = Trim(txtPurpose.Text)
Adomed.Recordset.Fields("date").Value = Trim(txtDatedos.Text)
Adomed.Recordset.Fields("frequency1").Value = Trim(cmbFreqd.Text)
Adomed.Recordset.Fields("frequency2").Value = Trim(cmbFregt.Text)
Adomed.Recordset.Fields("frequency3").Value = Trim(cmbFreqm.Text)
Adomed.Recordset.Fields("location").Value = Trim(txtLocdos.Text)
Adomed.Recordset.Fields("allergies").Value = Trim(txtAller.Text)
Adomed.Recordset.Save
```

```
End If
End Sub
```

```
Private Sub Command17_Click()
```

```
If Len(Trim(txtDatemc.Text)) = 0 Or Len(Trim(txtLocmc.Text)) = 0 Then
MsgBox "Please fill in all particulars", vbCritical, "Error Message"
Else
```

```
Adomc.Recordset.AddNew
Adomc.Recordset.Fields("date").Value = Trim(txtDatemc.Text)
Adomc.Recordset.Fields("location").Value = Trim(txtLocmc.Text)
Adomc.Recordset.Fields("period").Value = Trim(cmbPeriodmc.Text)
Adomc.Recordset.Fields("reason").Value = Trim(txtReasonmc.Text)
Adomc.Recordset.Fields("authorized name").Value = Trim(txtAuthormc.Text)
Adomc.Recordset.Fields("authorized place").Value = Trim(txtAuthormcloc.Text)
Adomc.Recordset.Save
```

```
End If
```

```
End Sub
```

```
Private Sub Command22_Click()
```

```
frmHistorymc.Show
```

```
End Sub
```

```
Private Sub Command23_Click()
```

```
If Len(Trim(txtFirsteme.Text)) = 0 Or Len(Trim(txtRel.Text)) = 0 Then
MsgBox "Sorry, data is not complete", vbCritical, "Error Message"
Else
```

```
Adoeme.Recordset.AddNew
Adoeme.Recordset.Fields("first name").Value = Trim(txtFirsteme.Text)
Adoeme.Recordset.Fields("last name").Value = Trim(txtLasteme.Text)
Adoeme.Recordset.Fields("home phone").Value = Trim(txtHomeeme.Text)
Adoeme.Recordset.Fields("hand phone").Value = Val(Trim(txtHandeme.Text))
Adoeme.Recordset.Fields("relationship").Value = Trim(txtRel.Text)
Adoeme.Recordset.Fields("notes").Value = Trim(txtNoteseme.Text)
Adoper.Recordset.Save
```

End If

End Sub

Private Sub Command24_Click()

```
txtFirsteme.Text = ""  
txtLasteme.Text = ""  
txtHomeeme.Text = ""  
txtHandeme.Text = ""  
txtRel.Text = ""  
txtNoteseme.Text = ""
```

End Sub

Private Sub Command27_Click()

```
frmHistorymed.Show
```

End Sub

Private Sub Command3_Click()

```
txtDoc1.Text = ""  
txtDoc2.Text = ""  
txtDoc3.Text = ""  
txtDocp1.Text = ""  
txtDocp2.Text = ""  
txtDocp3.Text = ""  
txtLastdoc.Text = ""  
txtLastdocp.Text = ""  
txtLastdate.Text = ""  
txtNotes2.Text = ""
```

End Sub

Private Sub Command5_Click()

```
txtDatemed.Text = ""  
txtDiag.Text = ""  
txtDocmed.Text = ""  
txtLocmed.Text = ""  
txtMedet.Text = ""
```

End Sub

Private Sub Command7_Click()

```
frmProvider.Show
```

End Sub

Private Sub Command8_Click()

```
txtPrescrip.Text = ""  
txtPurpose.Text = ""  
txtDatedos.Text = ""  
txtLocdos.Text = ""  
txtAller.Text = ""
```

End Sub

Private Function callload()

```
Adoper.Recordset.MoveFirst  
Do While Not Adoper.Recordset.EOF  
If Adoper.Recordset.Fields("ic no").Value = frmVerification.txtChic.Text Then  
txtFirst.Text = Adoper.Recordset.Fields("first name").Value  
txtLast.Text = Adoper.Recordset.Fields("last name").Value  
txtAdd.Text = Adoper.Recordset.Fields("address").Value  
txtCode.Text = Adoper.Recordset.Fields("post code").Value  
txtCity.Text = Adoper.Recordset.Fields("city").Value  
txtHome.Text = Adoper.Recordset.Fields("home phone").Value  
txtHand.Text = Adoper.Recordset.Fields("hand phone").Value  
txtIc.Text = Adoper.Recordset.Fields("ic no").Value  
cmbGender.Text = Adoper.Recordset.Fields("gender").Value  
txtDay.Text = Adoper.Recordset.Fields("birth day").Value  
cmbMonth.Text = Adoper.Recordset.Fields("birth month").Value  
txtYear.Text = Adoper.Recordset.Fields("birth year").Value  
txtOcc.Text = Adoper.Recordset.Fields("occupation").Value  
imgRet.Picture = LoadPicture(Adoper.Recordset.Fields("photo").Value)  
txtIc2.Text = Adocli.Recordset.Fields("ic no").Value  
txtIc3.Text = Adophy.Recordset.Fields("ic no").Value
```

```
End If  
Adoper.Recordset.MoveNext  
' Adocli.Recordset.MoveNext  
' Adophy.Recordset.MoveNext  
Loop  
End Function
```

APPENDIX A4 - Cover Letters

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100



JABATAN KESIHATAN NEGERI,
PERAK DARUL RIDZUAN,
(PERAK STATE HEALTH DEPARTMENT),
JALAN PANGLIMA BUKIT GANTANG WAHAB,
30590 IPOH.

Telefon: 05-253 3489/253 3012
05-253 3612
No. Fax: 05-243 3420
E-Mail: jknpk@po.jaring.my

Ruj. Kami : Bil (12) dlm.JKN.PK.Latihan 2004
Tarikh : 10 April 2004

Pengarah Hospital
Hospital **BATU GAJAH/CHANGKAT MELINTANG**

Tuan/Puan,

ENCIK FAIRUS BIN HARIRI
NO KAD PENGENALAN : 810424-14-5727

Dengan hormatnya saya diarah merujuk kepada perkara diatas

2. Sukacita dimaklumkan bahawa penama di atas adalah seorang pelajar tahun akhir jurusan Kejuruteraan Elektrikal dan Elektronik dari Universiti Teknologi Petronas, Tronoh Perak.
3. Beliau sedang menjalankan satu kaji selidik ke atas Projek Tahun Akhir yang di namakan "**PATIENT SMART CARD**".
4. Pelajar ini akan menemui pihak tuan/puan untuk mengumpul data-data atau sokongan teknikal yang berkaitan dengan projek beliau. Umumnya, pelajar ini akan menemuramah di antara 30 – 50 pesakit. Antara data-data yang bakal dikumpulkan adalah seperti keterangan peribadi, pilihan hospital dan doktor, sejarah perubatan, ubat-ubatan, immunisasi dan sijil sakit (MC).
5. Sukacita dimaklumkan pihak kami tiada halangan untuk pelajar tersebut melakukan kaji selidik di hospital tuan/puan.

Sekian, terima kasih.

**"PENYAYANG, BEKERJA BERPASUKAN DAN PROFESIONALISMA
ADALAH BUDAYA KERJA KITA"**

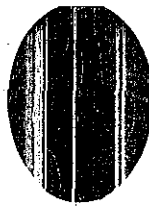
Saya yang menurut perintah,

(**WAN BADARIAH BTE. DATO' SERI WAN MOHD**)
Ketua Penolong Pengarah (Pengurusan)
b.p. Pengarah Kesihatan Negeri
Jabatan Kesihatan Negeri
PERAK DARUL RIDZUAN

s.k

Fail Timbul

amir@ozza



UNIVERSITI
TEKNOLOGI
PETRONAS

Our Ref

UTP-ACAD/EE/7202-02-009-36

Your Ref

29 March 2004

Pengarah
Jabatan Kesihatan Negeri Perak
Jalan Panglima Bukit Gantang Wahab
32100 Ipoh
PERAK DARUL RIDZUAN

Dears Sir / Mdm;

To Whom It May Concern:

It is hereby certified that Mr. Fairus bin Hariri (I/C No: 810424-14-5727), final year student of Electrical and Electronics Engineering Program at University of Technology Petronas is working on his Final Year Research Project on PATIENT SMART CARD.

The student may meet you to collect some data or technical support related to his project. Particularly, the student will perform a case study on the range of 30 – 50 patients. Data collected will be regarding their *personal details, hospitals and doctors preferences, medical history, medication, immunization and medical certificate.*

There would be absolutely no problem in following the medical legal and official authorization. The research problem assigned to him will benefit the general public, if he successfully implements it.

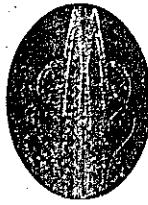
Any help extended to the student will be highly appreciated.

Thanks and regards,

.....
Assoc. Professor Dr. Mohd Noh Karsiti
Program Head
Department of Electrical and Electronics Engineering
University of Technology PETRONAS

UNIVERSITI TEKNOLOGI PETRONAS

INSTITUTE OF TECHNOLOGY PETRONAS SDN. BHD.
(Company No: 352875U) Wholly-owned subsidiary of PETRONAS
Bandar Seri Iskandar, 31750 Tronoh, Perak Darul Ridzuan, Malaysia



UNIVERSITI
TEKNOLOGI
PETRONAS

Our Ref : UTP-ACAD/EE/7202-02-009-30

Your Ref :

5 March 2004

Pengarah Hospital
Hospital Batu Gajah,
Jalan Changkat,
Peti Surat 40, 31000 Batu Gajah,
Perak Darul Ridzuan

Dears Sir / Mdm;

To Whom It May Concern:

It is hereby certified that Mr. Fairus bin Hariri (I/C No: 810424-14-5727), final year student of Electrical and Electronics Engineering Program at University of Technology Petronas is working on his Final Year Research Project on PATIENT SMART CARD.

The student may meet you to collect some data or technical support related to his project. Particularly, the student will perform a case study on the range of 30 – 50 patients. Data collected will be regarding their *personal details, hospitals and doctors preferences, medical history, medication, immunization and medical certificate.*

There would be absolutely no problem in following the medical legal and official authorization. The research problem assigned to him will benefit the general public, if he successfully implements it.

Any help extended to the student will be highly appreciated.

Thanks and regards,

.....
Assoc. Professor Dr. Mohd Noh Karsiti
Program Head
Department of Electrical and Electronics Engineering
University of Technology PETRONAS

UNIVERSITI TEKNOLOGI PETRONAS

INSTITUTE OF TECHNOLOGY PETRONAS SDN. BHD.

(Company No: 352875U) Wholly-owned subsidiary of PETRONAS

Bandar Seri Iskandar, 31750 Tronoh, Perak Darul Ridzuan, Malaysia

Tel: 605-3721100 Fax: 605-3667691 D. Lines: HRM 605-3721345 Finance 605-3721359 Library 605-3721330

Public Affairs 605-3721317 Student Services 605-3721272 Academic 605-3721440 Security 605-3721319

Fax: HRM 605-3654075 Finance 605-3654087 Library 605-3667672 Student Services 605-3721286 Academic 605-3654082 Website <http://www.utp.edu.my>