Faculty of Computer Science and
Information Technology

University of Malaya

Parking Summon System for Municipal Councils

By
Yee Hon Weng
WEK010311

Under the supervision of
Pn. Nazean Jomhari

Under the moderation of
Pn Siti Hafizah

2003/2004

WXES 3181/WXES 3182

Abstract

Parking Summon System for Municipal Councils (e-Summon System) is a complete system designed to ease the workload of officers in enforcing the parking laws. It aims to serve as a platform for local municipal councils to improve the efficiency of collecting compound and thus, improving the parking management as a whole.

e-Summon System will comprised of two main components, the PDA to issue compound notices and a main database to keep track of the issued notice. The system is devised to be employed by enforcing officers, data administrators and cashiers handling the collection of fines.

Acknowledgements

There are a few people whom I would like to thank for their great role in seeing the completion of this project.

I would like to express my sincerest gratitude to my supervisor, Pn Nazean Jomhari, for her patience and guidance. Next on my list will be my moderator, Pn Siti Hafizah, thank you for all your advice.

I would like to take this opportunity to extend my appreciation to Pn Nasriyah of MPPJ and En Arief Kamaruddin of DBKL, for being so helpful during the interview sessions.

Not forgotten are my parents, for all the support they gave during the course of this project and others who contributed to this project.

Thank you.

Contents

List of Table

List of Figures

Chapter One:

Introduction

1.0 Introduction

The advancement in human achievements especially in technology of computing has brought enormous benefit to the society. It enables certain task to be conducted more efficiently, which was previously hindered by many difficulties. Realizing this important fact, many organizations are beginning to reengineer their day-to-day process to exploit the power of information technology. It is for this exact reason that the system, "Parking Summon System for Municipal Councils" is being proposed.

The main objective of this system is to digitalize the business process for municipal councils in their task of issuing compound notices by using Personal Digital Assistant (PDA). The current practice of using pen-and-paper method has many loopholes and thus causes many problems, both in terms of the issuing of compound notices and the collecting of fines.

1.1 Project Overview

In order for municipal council officers to carry out their duty effectively, it is imperative that the devices or tools used is by nature, light-weight and provide a high degree of mobility. Therefore, one main component of the system will consist of a PDA. This device will be used to capture the details of parking violation and issue a compound notice.

The system consists also a main database, where the compound notices issued will be synchronized and kept. This main database will mainly be used by database administrators and cashiers handling the payment of the compound.

1.2 Project Motivation

The current systems employed by local municipal councils have a lot of drawbacks that is hindering the effectiveness and efficiency of parking management as a whole and in particular, in issuance of compound notices.

Most of the municipal councils used a minimal set of computerization in their day-to-day work and thus, creating problems such as shortage of personnel and draw complains from the public. For example, the usage of the traditional pen-and-paper in giving out compound warning requires the need of a data-entry clerk to update the database.

Local municipal councils also face difficulty in compound collection and due to the inadequacy of current system; they are finding it challenging to take court action against lawbreakers.

1.3 Project Objective

There are several aims which the author hopes to achieve through the completion of this project:

- To increase the efficiency of compound issuing system by applying information technology
- To minimize human error in issuance of compound notices
- To increase the effectiveness of enforcing the laws related to parking offences
- To ease the process of collecting compound for municipal councils
- To have a more user-friendly system to ease parking offenders in paying the compound
- To improve the overall parking management of municipal councils

1.4 Project Scope

Project scope identifies the core of the system as well as to define the system boundary.

The handheld device and the portable printer will only be used by enforcing officers to issue compound notices. It works as a standalone input tool, without direct communication to the main database, if it is not docked for synchronization purposes. Public cannot use the handheld as a mean to check the status of their outstanding compounds.

The main database is only accessible by authorized personnel of municipal councils. The available functions are again not directly used by the public. In inquiring or paying the compound, the public still need the assistance of a cashier.

1.5 Expected Project Outcome

The main deliverable of this project will be a complete system used to improve the overall enforcement of the parking laws by the local governments in Malaysia. This system will make up of two modules, the handheld device along with its portable printer and a database used to keep and track the entire compound notices issued.

1.5.1 PDA

This component is used as an inputting device, to be used by enforcing officers to issue parking compound notices. The usage of this device will be complemented with that of a portable printer, used to print the said notice.

1.5.2 Main Database

The main database consists of a database server, used to maintain data of parking violations gathered from the PDA. It is therefore that a method of synchronizing data between the two components must exist.

This database will be accessed by several key-users, namely the database administrators and the cashiers manning the payment operation. Therefore, different module must be implemented to limit access based on roles of the users.

1.6 Project Schedule

Project schedule is a form of planning used in project management to ensure the smooth running of the development process planned. The schedule will embody the scoping and planning of tasks in the process and the estimated time to be taken.

Figure 1.1 shows how the project is managed, schedule wise.

ID	Task Name	Duration
1	Project Definitions	3 wks
2	Literature Review	3 wks
3	Methodology	2 wks
4	Analysis	3 wks
5	Design	3 wks
6	Implementation	14 wks
7	Testing	4 wks
8	Documentation	28 wks

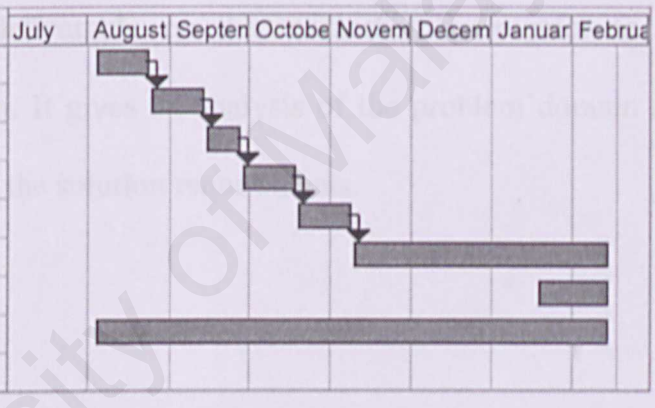

Figure 1.1 Scheduling of task for the project.

1.7 Chapter Summary

- Chapter One – Introduction

  This chapter brings an overall insight towards the project. Definition of the problem, project objective, scope and limitations as well as project planning will be presented in this chapter.

- Chapter Two – Literature Review

  This chapter is the main deliverable on the researches done prior to the implementation of the system. It gives an analysis of the problem domain and provides a general glimpse of the solution requirements.

- Chapter Three – Methodology

  This chapter discusses the approach used in system development as well as the techniques used for information elicitation. Included in this chapter will be functional and non-functional requirements of the system as well as technology used in the development.

- Chapter Four – System Analysis and Design

  Diagrams and interface design used in the system is being brought forward in this chapter. Included are elaborations on the design of different modules available in the system.

- Chapter Five – System Implementation

  This chapter focuses on the mapping of design and algorithm into computer codes executable in the programming language.

- Chapter Six – Testing

  Testing done in the duration of the project implementation will be documented here. It will feature reports on the functions of the system as compared to the requirement specified.

- Chapter Seven – System Evaluation and Conclusion

  The strengths, weaknesses and future enhancement of the system will be thoroughly examined here. This chapter also discusses the problems faced and the conclusion of the project.

Chapter Two:

Literature Review

2.0 Literature Review

2.1 Local Government

Local government has been described by the United Nations as "A political subdivision of a nation or (in a federal system) state, which is constituted by law and has substantial control of local affairs, including the power to impose taxes."

Local government in Malaysia is accordingly concerned with those authorities established as municipalities, district councils, local councils and town board. It is created by law and set apart from the central or state administration. Local government is differentiated from public corporation in terms of its areas of concern and jurisdiction, the range of services provided as well as its overall objective and purpose.

2.1.1 Case Study - Dewan Bandaraya Kuala Lumpur

The history of Dewan Bandaraya Kuala Lumpur (DBKL) can be traced back to the Sanitary Board, which was set up by the British Rule in the 1880s. It is only known by its current name in the year 1972, when Kuala Lumpur was bestowed the status of city by His Royal Highness Seri Paduka Baginda Yang di-Pertuan Agung, under the Kuala Lumpur Act 1971.

For its parking management, DBKL uses the meter system. The meter system works by allowing users to insert coins and to display the remaining time based on the amount put in. For its enforcement, officers will check these parking meters and issue compound notices when the time period has expired.

The compound notices are issued using pen-and-paper method. At the end of the day, the officer in-charge will submit the booklet, containing carbon-copy of the notices issued, to be updated into a main computer database. For this purpose, data has to be manually keyed-in by clerks.

When collecting the compound, user will have to present the compound notice and payment will be made to the cashier. The bottom portion of the notice will be printed-over and retained by user as proof of payment. The other portion of the compound notice will be used to update the main database, which again is done manually.

If the compound is not paid within the first three months of the offence, a reminder notice will be sent. A second reminder notice will be sent, a months from the first notice. By the third month, if the compound is not paid, DBKL has the right to transfer the case to the court of law, where the offender will be summoned.

To get vehicle ownership details, DBKL has to retrieve the data from the Road and Transport Department (RTD). This is done by querying the database on vehicles which does not have ownership details. The car registration number is then obtained

and batched to be manually sent to RTD office. Once the data is obtained, the details will be manually updated.

Disadvantage of the system employed by DBKL are:

- Prone to human-error

  Since most of the work-flow involve a high amount of manual work, the tendency for human-error to occur increases. This is especially so in the area of data-entry where the clerk has to deal with a large workload.

- Lag in updating the database

  DBKL are currently experiencing a lag in updating the main database. It has reported a two-day delay in updating records on new compound notices, while updating of payment have a backlog of three months.

- Lack of follow-up action on parking offences

  Due to the glitches in updating of offence and of compound payment, DBKL face difficulty in taking follow-up actions on parking violators.

2.1.2 Case Study - Majlis Perbandaran Petaling Jaya

Majlis Perbandaran Petaling Jaya (MPPJ) overlooks an area of 97.2 km and is one of the most well-known municipal council in the country due to its relentless effort in modernizing its administration.

For its parking management, MPPJ uses the pay-and-display system. To complement the pay-and-display system, MPPJ introduced the usage of handheld

and portable printer to issue compound notices. This solution is provided by Formis Network Services Sdn Bhd.

The main function of handheld is to record details on offences committed and to issue compound notices based on the data inputted. It is the equivalent of the compound-notice booklet of DBKL. Among the features of the handheld device is a keyboard and a built-in camera. The keyboard eases the enforcing officer in inputting data, while the camera is used to capture picture of the offence committed, whereby reducing cases of dispute. For printing of compound notice, a portable printer is used. The handheld device will be docked for data synchronization with the main database in MPPJ office at the end of the officer's shift.

To collect compound payment, cashier uses a different computer system from the main database. It is used to record compound paid and to issue receipt. Records from the cashier system will then be synchronized with main database to update the status of the compound.

MPPJ uses the same method as DBKL in obtaining the ownership details of vehicles which are issued compound notices.

The disadvantages of MPPJ's system are:

- Customised for Pay-and-Display

  Most of the functions in the handheld are specially tailor-made to suit the need of MPPJ's parking management. This will present a problem when an attempt is made to use the system in other municipality.

- Different system for cashier

  The cashier is not connected directly to the main database and this prevent the automation of data updating when a payment is made. This also prevents users from checking the status of their compound and necessitates the presence of the compound notice when making payment.

2.2 Operating System For PDA

Operating system is the main software in a particular computing device that powers its main function as well as to provide services for other application to operate.

The PDA market is currently witnessing the 'operating system war' between two companies, Palm Inc. with their PalmOS and Microsoft with Windows CE.

2.2.1 Windows CE

Windows CE is a compact, modular 32-bit operating system designed for use on devices with small memory requirements. Windows CE is very similar in design to its larger desktop cousin, Windows NT. Windows CE is a multitasking, multithreaded operating system like Windows NT. It includes most of the user interface features of Windows NT so that software developers can take advantage of most users' familiarity with Windows applications

Storage on Windows CE devices is a combination of random access memory (RAM) and read-only memory (ROM). Devices can also include expansion flash memory storage cards for additional storage space. PCMCIA cards can be added to many devices, and Windows CE provides full support for such cards.

In order to promote the usage of Windows CE, Microsoft has come out with development tools specifically for this platform, free for download from its website. The tools offered are Microsoft eMbedded Visual Tools 3.0, Software Development Kits (SDK) and Microsoft .NET Compact Framework.

2.2.2 PalmOS

PalmOS was first introduced as the operating system for Palm Pilot in 1996 and since then has grown from strength to strength. Today, PalmOS is in its version 5 and is used to power two out of three handhelds in the world. However, PalmOS has seen its share market being steadily eroded by the other operating system for handheld, Ms Windows CE.

PalmOS has been touted by users and reviewers alike as having the most user friendly interface for handhelds. Being in the market for quite some time proves to be positive for PalmOS as a wider range of hardware and software supports it.

Development environments are available for PalmOS in a number of different languages such as C, C++, Visual Basic and Java.

The most commonly used development environments use standard C as the programming language, and add-ons are available to allow you to use C++ frameworks as well. With C or C++ applications, the Palm OS provides much of the functionality that applications need for memory management, string manipulation, drawing to the screen, and so on.

The development environments for Visual Basic and Java require a runtime engine to be present. Some of these environments require the engine be present on the device, and others compile runtime elements into the application. In either case, the resulting code base is not as compact as with the C-based applications. However, the runtime engines are typically free to the user, and the development tool may very well add enough benefit to make the increase in code size worthwhile.

2.3 Wireless Technology

Wireless refers to the electrical or specifically computing device, that communicate among each other without the use of physical cabling or wiring. Along the years and breakthrough in scientific research has brought to us several choices of wireless technology, each with their own strength and weaknesses.

2.3.1 IR Wireless

IR wireless is the use of wireless technology in devices or systems that convey data through infrared (IR) radiation. Infrared is electromagnetic energy at a wavelength or wavelengths somewhat longer than those of red light. The shortest-wavelength IR borders visible red in the electromagnetic radiation spectrum.

The usage of infrared technology in computing is governed by an organization called the Infrared Data Association (IrDA). To achieve this purpose, an IrDA Standards was published and it is used to define the protocol suite used to support transmission of data between two devices using infrared technology.

Over the last three years, the members of the IrDA have been very successful at getting IrDA hardware deployed in a large number of new notebook computers. Microsoft Windows CE 1.0 was the first Windows operating system to provide built-in IrDA support. Windows 2000 and Windows 98 now also include support for the same IrDA programming APIs that have enabled file sharing applications and games on Windows CE.

The IrDA specification has allowed the infrared technology to be implemented at speeds between 9.6 kilobits per second and 4 megabits per second. However, IrDA has a limitation in that it is only applicable in short distance. Due to these features, IrDA is best suited for ad-hoc point-to-point networking.

It is also noted that most PDAs come equipped with IrDA as a standard feature.

2.3.2 Bluetooth

Bluetooth is a computing and telecommunications industry specification that describes how mobile phones, computers, and personal digital assistants (PDAs) can easily interconnect with each other and with home and business phones and computers using a short-range wireless connection.

Bluetooth requires that a low-cost transceiver chip be included in each device. The transceiver transmits and receives in a previously unused frequency band of 2.45 GHz that is available globally. In addition to data, up to three voice channels are available. Another advantage of Bluetooth is that there is no need for clear line of sight between devices and the transmission can go through despite having barriers such as wall.

Bluetooth allows data to be exchanged at the rate of 1 megabits per second.

Chapter Three:

Methodology

3.0 Methodology

3.1 Software Process

There are many ways to achieve a result and the development of software is no different. Software process is described by Sommerville as a set of activities and associated results which produce software product. Software development can be described as still in infancy, as compared to other engineering discipline. It is therefore a lot of research has been done on it and results in differing recommendations.

It is noted that most research agrees on these activities as the basis of software development:

- *Requirement analysis and definition*

  The system's services, constraints and goals are being identified in this stage. Output from this phase, which is called the specification document, serves as the input for the next.

- *System and software design*

  System and software design stage aims to establish the overall system architecture. It is often decomposed into several layers of abstraction and results in logical models of the solution system.

- *System implementation*

  It is in this stage that actual program coding for the system is done. The coding process is guided by models which were developed in the prior stage.

- *System testing and validation*

  Vigorous testing and validation is done on the system before being put into use. This is to minimize bugs in the system as well as to validate that the system fulfills the requirements.

3.1.1 The 'Waterfall' Software Process Model

This model of software development is the first model to be published and it is taken from other engineering discipline. This model maps the fundamental steps in development of software as a cascading processes, hence the name. Each phase in the model must be completed before the work on the next can begin.

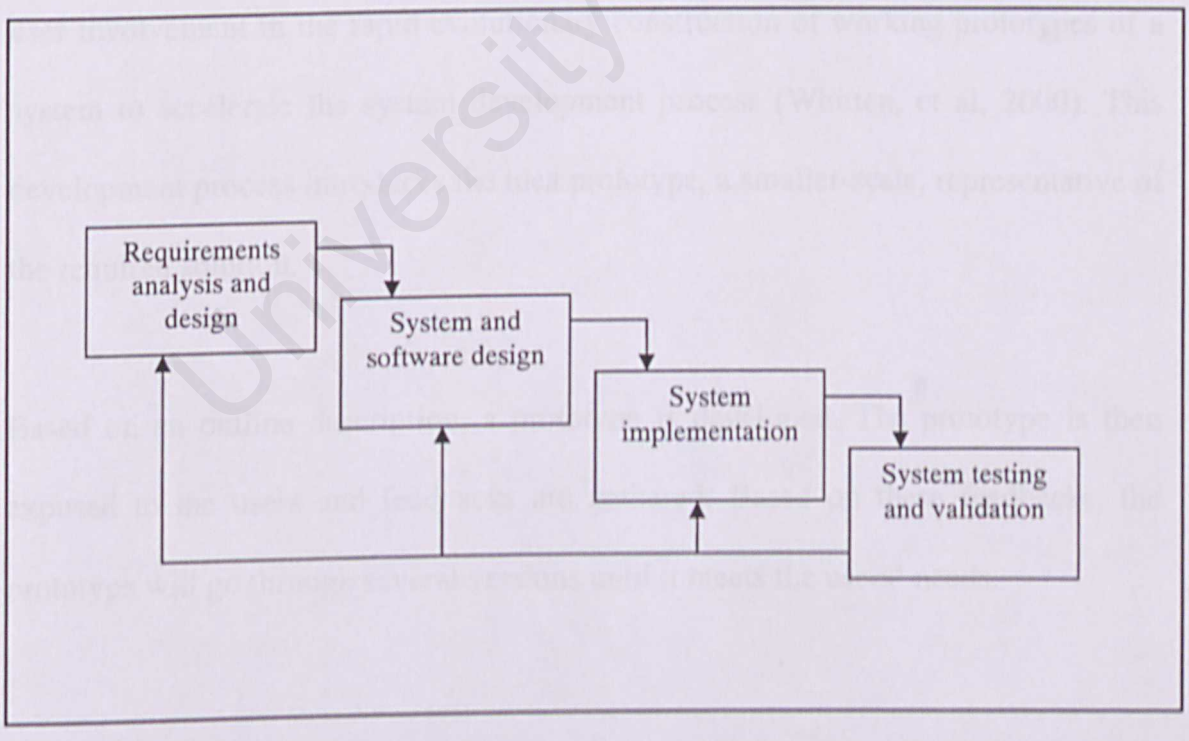

Figure 3.1 Waterfall model

Simplicity and ease to use are the main advantages of this model. However, on the flipside, the rigidity of this model inhibits its use as it is often costly to do iterations and significant rework of certain stages.

It is recommended that this model is used when there is a clear understanding of the system requirement.

Figure 3.1 shows the graphical abstraction waterfall model.

3.1.2 Rapid Application Development

Rapid Application Development (RAD) is a technique that emphasizes extensive user involvement in the rapid evolutionary construction of working prototypes of a system to accelerate the system development process (Whitten, et al, 2000). This development process introduces the idea prototype, a smaller-scale, representative of the required solution.

Based on an outline description, a prototype is developed. The prototype is then exposed to the users and feedbacks are gathered. Based on these feedbacks, the prototype will go through several versions until it meets the users' needs.

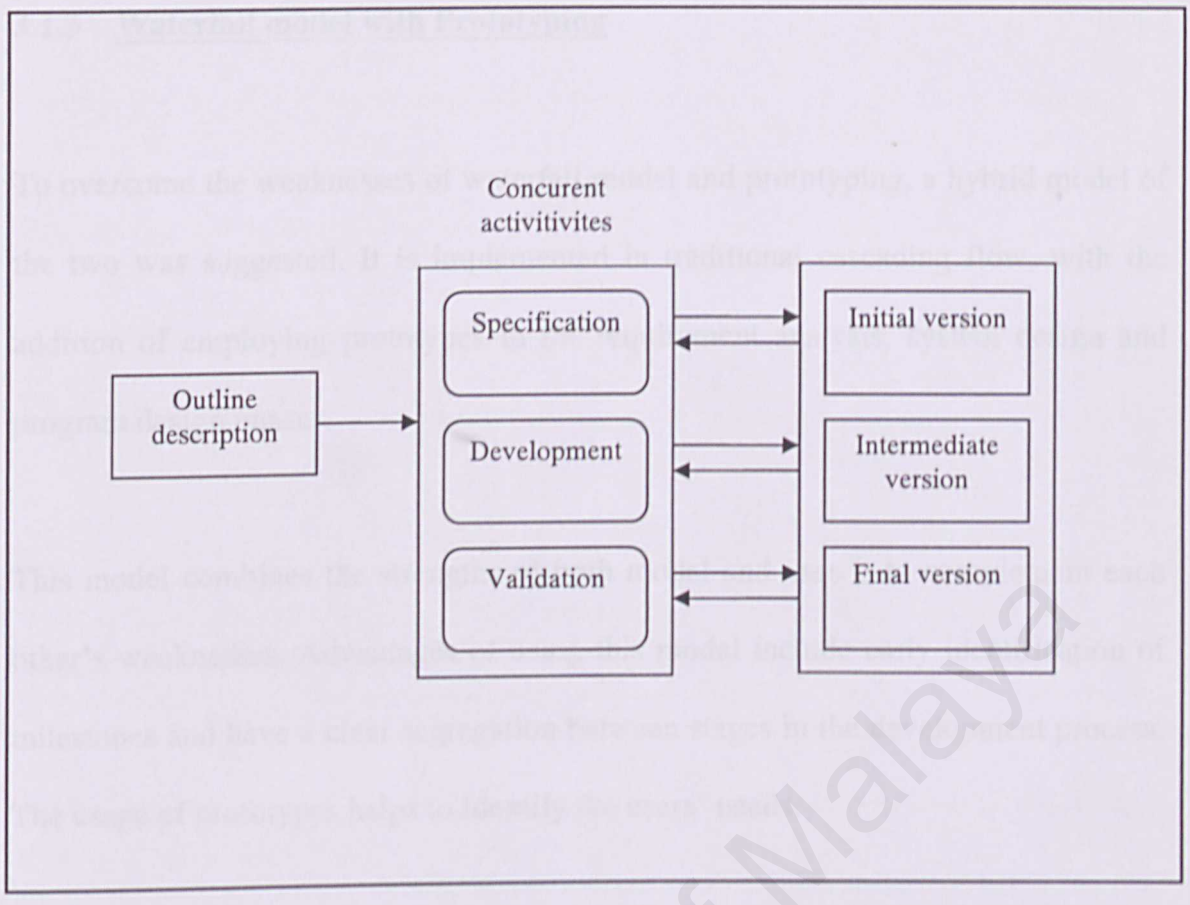

Figure 3.2 Rapid Application Development model

The activity of prototyping and gathering users' feedback can be seen as the combinations of several phases; requirements specification, system implementation and validation.

This approach is useful when the requirements are unclear and it reduces the risk of the system being buggy. However, this approach call for the use of special tools and the outcome system is often structured poorly. Furthermore, the high user involvement also makes this approach impractical.

Figure 3.2 shows the Rapid Application Development model.

3.1.3 Waterfall model with Prototyping

To overcome the weaknesses of waterfall model and prototyping, a hybrid model of the two was suggested. It is implemented in traditional cascading flow, with the addition of employing prototypes in the requirement analysis, system design and program design phases.

This model combines the strengths of both model and uses it to complement each other's weaknesses. Advantages of using this model include early identification of milestones and have a clear segregation between stages in the development process. The usage of prototypes helps to identify the users' needs.

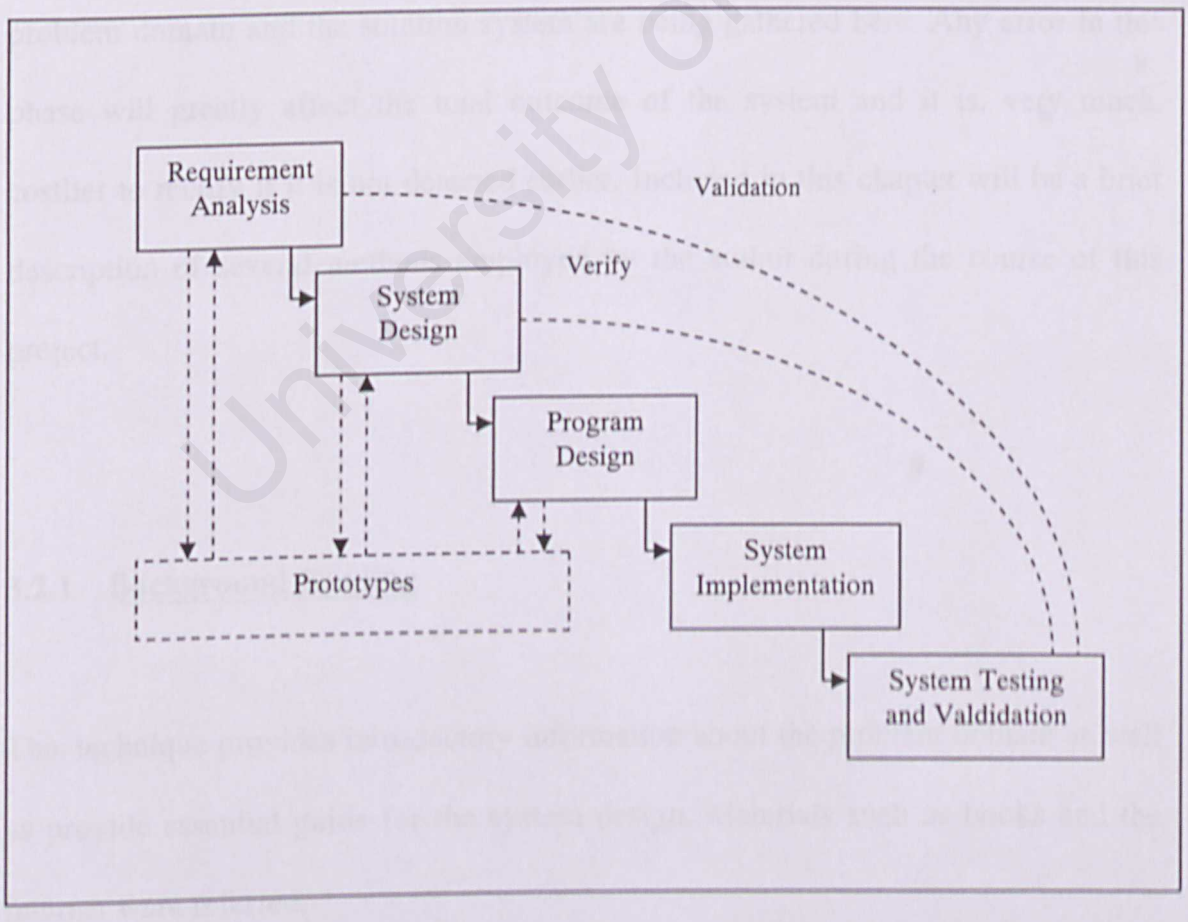

Figure 3.3 Waterfall model with prototyping

This model also creates a better visibility of the process as compared to using prototyping in the development alone. Furthermore, the model also takes into consideration the need of iterations in the development process.

Figure 3.3 shows the waterfall model with prototyping which is employed as the development process in the project.

3.2 Information Elicitation

Elicitation, also known as information gathering, concerns with the acquisition of information. It is an important phase in the software process, for details of the problem domain and the solution system are being gathered here. Any error in this phase will greatly affect the total outcome of the system and it is, very much, costlier to rectify if it is not detected earlier. Included in this chapter will be a brief description of several methods employed by the author during the course of this project.

3.2.1 Background Reading

This technique provides introductory information about the problem domain as well as provide essential guide for the system design. Materials such as books and the internet were referred.

3.2.2 Interview

Interview is a technique that elicits information that resides within people's heads and it is one of the most commonly used elicitation method.

There exist several types of approach in interviewing. The structured interview calls for a set of pre-planned questions and is rigid, while at the extreme end, the unstructured requires no prior planning. A semi-structured interview approach, where a reasonable but not excessive planning was done, was practiced by the author.

In the course of this project, two interviewing session was organized by the author; one with the officials at Majlis Perbandaran Petaling Jaya and the other at Dewan Bandaraya Kuala Lumpur.

For questions of the interview, please refer to Appendix A.

3.2.3 Task Observation

Task observation is a fact-finding technique wherein the systems analyst either participates in or watches a person perform activities to learn about the system. This method is largely used to look at how users interact with the precursor system.

Observation is used to complement other elicitation techniques especially where description of a complex task is involved. However, it is noted that data gathered from this process needs to be validated to improve its accuracy.

3.3 System Requirements

System requirements are the description of the services and constraints of a particular system. It can also be described as effects which the user wishes to be brought about in the problem domain. System requirement can be further classified into functional and non-functional requirements.

3.3.1 Functional Requirements

The functional requirements, also known as behavior requirements, describe the functionality or services that the system is expected to provide.

For E-summon, functional requirement will be further divided into its component, the handheld and main database.

3.3.1.1 PDA

- Consist of a palmtop computer (handheld) to input details of parking violations and another (printer) to print the compound notice
- The PDA must provide security features
    - Require identification and authentication process
    - Identity will be determined using officer identification number, to be assigned later
    - Provide a module for security administration purpose
- Must automatically generate serial number
    - Serial number consists of eight digits; first two unique to the device (as device identification) and the last six digits are running number.
    - Serial number will be used as the primary key for each compound notice
- Data such as time and date must be automatically provided for each compound notice
- Provide two methods of inputting; character recognition and touch-screen keyboard
- Data input
    - Field 'Nombor Kenderaan' will be direct input
    - Field 'Nombor Cukai Jalan' will be direct input
    - Field 'Jenama Kenderaan' will be code number and drop-down menu
    - Field 'Warna Kenderaan' will be code number and drop-down menu
    - Field 'Tempat Kesalahan' will be direct input
    - Field 'Peruntukkan Undang-undang' will be drop-down menu

- Field 'Seksyen / Kaedah' will reflect the input of 'Peruntukkan Undang-undang'; provided through drop-down menu

- Provide a mean to synchronize data with main database

3.3.1.2 Main database

The functional requirements for the main database will be reviewed in terms of modules available.

- User Account Setup

  This module provides functions for system administrator to setup user accounts. Functions available in this module are

- Data Administration

  This module provides access to only authorized clerks to update details of vehicles which were issued compound notice. However, only tables Pemilik, Notis_Peringatan_1, Notis_Peringatan_2 and Tindakan_Mahkamah are modifiable under this module.

Create new user	o Create a new profile for new user. o Setup authorization level o Provide user name and password to login to database
Modify user	o Change user profile
Delete user	o Delete a user profile
Create new officer	o Create a profile for enforcing officers o Provide officer id and password to login to PDA system
Modify officer	o Change officer profile
Delete officer	o Delete officer profile

Table 3.1 Functions of User Account Setup

Modify data	o Modify the data in tables mentioned
Search record	o Search for specific record
Print record	o Print a specific record
Report Generation	o Able to generate reports and print them

Table 3.2 Functions of Data Administration module

- Cashier

  Only cashier have access to this module. Cashiers have only limited access to the database and is only able to modify the payment status and mod_bayaran field in the Kompaun field. A receipt, which serial number will be automatically generated, will be issued. The Tarikh_bayar field will be automatically updated with the date of payment.

Search record	o Search for a specific record
Pay compound	o Modify the status, mod_bayaran field o A receipt must be issued

Table 3.3 Functions of Cashier module

3.3.2 Non-Functional Requirements

Non-functional requirements define the attributes of the system that do not directly concern with the specific functions delivered by the system. It can also be defined as performance parameter expected of from the system. Most non-functional requirements affect the system as a whole and therefore can be viewed as more critical than individual functional requirements. The following describes the non-functional requirements in detail.

3.3.2.1 Modularity and Maintainability

The system must be implemented in a modular architecture. This is important to ensure high maintainability as well as to help user understand the system. Furthermore, codes and design that are executed modularly are easier to enhance and upgrade.

The practice of modularizing the system also promotes reuse for common procedures and functions as well as to prevent code redundancy.

3.3.2.2 Consistency

Consistency can be described as locating the info in the same area each time a new screen is accessed. It groups information that logically belong together.

Consistency is imperative in promoting system simplicity and to lessen the users' learning curve.

3.3.2.3 Reliability

System that does not produce dangerous or costly failure when used in a reasonable method implies reliability. Reliability also means able to perform required functions and operations without resulting in unnecessary and unplanned down-time.

3.3.2.4 Security

The system should provide security to data which it accesses. Methods of identifying and authenticating user must be implemented. The system must also ensure that each user have proper authorization and can only access data pertinent to him or her.

In terms of functions, a user may only execute actions that are allowed in his or her authority level.

3.3.2.5 Response Time

The response time for the system must be within the users' level of acceptance. To improve response time, input validation must be implemented in the client's side. This is to prevent time wastage in sending the input data to server and pass the validation error result.

3.4 Chosen Technology

There exist a wide variety of selections in terms of technology for the development of the system. This part of the chapter describes the technologies chosen to build the system.

3.4.1 Operating System

The system involves three platforms, namely the server, client and PDA. Therefore, consideration must be made in order to choose the most suitable operating system.

3.4.1.1 Windows 2000 Server

Windows 2000 Server is the multipurpose network operating system offering from Microsoft. It is suitable to use by businesses of all sizes. Windows 2000 Server promises a reliable and secure sharing of resources across the network and is compatible with most business applications.

Among features available in Windows 2000 server is its flexibility and high scalability as it supports two methods of scaling; scale-up and scale-out. Scale-up refers to running a single application or image on a single server, and having the ability to incrementally add system hardware resources to increase overall system performance. Scale-out has been defined as distributing the computing workload among multiple servers by clustering or load balancing, with the ability to add or subtract servers to increase or decrease capacity.

Windows 2000 Server also offers high manageability. With the use of Configure Your Server Wizard, one is able to reduce the time it takes to build a server and at the same time, reduces the likelihood of error. Additionally, Windows 2000 Server

allows you to configure your network more easily. It provides support for Plug and Play network adapters, significantly reducing device configuration time.

Furthermore, Windows 2000 Server offers a centralized management services that supports group policy and provides a management console that presents a common user interface presentation tool.

Windows 2000 Server will be employed as the operating system for the server.

3.4.1.2 Windows XP Professional

Windows XP is the latest offering from Microsoft from its famed Windows-family of operating systems.

Its main architecture is derived from the Windows 2000 platform, which is 32-bit based and offers fully protected memory model, therefore, guaranteeing its reliability. It also brings forth the other features from previous predecessors such as innovative support services and Plug and Play.

Windows XP also provides improved code protection by only allowing its critical kernel data to be read-only. In terms of security, this OS provides administrators a policy driven mechanism to identify software running in their environment and control its ability to execute. Another main draw towards Windows XP is its simplified user interface and high usability as compared to other OS.

For the project, Windows XP will be used to power client PCs accessing the server.

3.4.1.3 Windows CE

Windows CE was chosen over PalmOS as the operating system for the PDA is based on several reasons.

The advantage of using Windows CE is that application programmers get a huge productivity boost because Windows CE is based on the Win32 API. This means that programmers who are familiar with programming for traditional Windows platforms like Windows NT can begin programming Windows CE applications with very little additional training. Certainly there are features that are unique to Windows CE. But understanding traditional Windows programming is a big advantage when moving to the Windows CE operating system.

3.4.2 Database Management System

For database management system, Microsoft SQL Server 2000 will be used. . It is a relational database management and analysis system for e-commerce, line-of-business, and data warehousing solutions. It is a database management system that provides high performance and availability features to partition load and ensure uptime, and advanced management and tuning functionality to automate routine tasks and lower total cost of ownership.

SQL Server 2000 employs graphical tools and setup wizards to help in the design of databases. It also optimizes database performance by having self-tuning and

dynamic self-configuring features. This further simplifies the process of building up a database.

In terms of replication, SQL Server 2000 implements merge, transactional, and snapshot replication with heterogeneous systems. Another benefit of this product is its high integration ability with other Microsoft line such as SQL Server CE and Microsoft Office suite.

3.4.3 Data Access

3.4.3.1 ADO .NET

Microsoft ActiveX Data Object (ADO) .NET is the latest data access technology by Microsoft that is entirely built based on the .NET Framework. ADO .NET offers a few benefits over its predecessors in that it has disconnected data architecture, presents tight integration with XML, have a common data representation with the ability to combine data from multiple and varied data sources as well as being an optimized facilities for interacting with a database.

ADO .NET supports most of today's development models. However it remains as similar as ADO to give current ADO programmers a head start in converting to the new data access technology. Another plus point for current batch of ADO programmers is that ADO .NET is mostly backward compatible with ADO.

3.4.3.2 ADOCE

Microsoft ActiveX Data Object for Windows CE (ADOCE) is the subset of the desktop ADO especially designed to work in the Windows CE environment. Among the functionality provided by ADOCE is the presence of an internal database provider that enables users to access database stored locally in a Windows CE device, as opposed to working remotely with database stored on a desktop.

With the ADOCE control, you can develop applications using most of the same ADO syntax you use for programming on a desktop computer. ADOCE requires minimal work to port existing ADO-based applications. Whenever possible, ADOCE returns the same error values and error strings as ADO, which maintains the separation of errors between OLE DB errors and native ADO errors.

3.4.4 IR Wireless

Despite appearing as inferior to the Bluetooth technology of wireless communication, IR wireless has a few strengths that make it fitting to be utilized in the project.

Even though Bluetooth is able to transmit at a further distance, it applies a broadcast method of communicating. This can present several problems especially when there are a few Bluetooth enabled device in the vicinity. The user has to manually select which device it wishes to communicate with. This is in contrast with IR wireless

where a point-to-point clear line of sight must be available between the devices and therefore diminishes the above said problem.

Another argument will be in terms of hardware cost. Most PDAs come ready for IR and so is most of the portable printers. In the case where a device is Bluetooth compliant, it has a tendency to be higher priced than devices with IR.

3.4.5 Visual Basic .NET

Visual Basic .NET (VB .NET) 2000 provides the easiest and most productive tool for building applications. Its ease of use compiled together with a user-friendly integrated development environment, makes it one of the best tool to support rapid application development.

As an approach to be backward compatible with Visual Basic 6.0, VB .NET provides an upgrade wizard for programmers to convert their old coding to that of the new standard. Together with ADO .NET, VB .NET provides a highly flexible and scalable data access. Some other features of VB .NET include the infusion of application with built-in security, provides a direct access to the .NET Framework and the capability to target an extensive array of mobile devices.

The main reason VB .NET is chosen is due to its simplicity in use and its support for prototyping, which is an essential process in the project.

3.4.6 eMbedded Visual Basic 3.0

eMbedded Visual Basic (eVB) is a cross between browser-based VBScript and Visual Basic for the desktop, designed specifically to run in Windows CE platform. It forms a part of the eMbedded Visual Toolkit 3.0 and is distributed free by Microsoft.

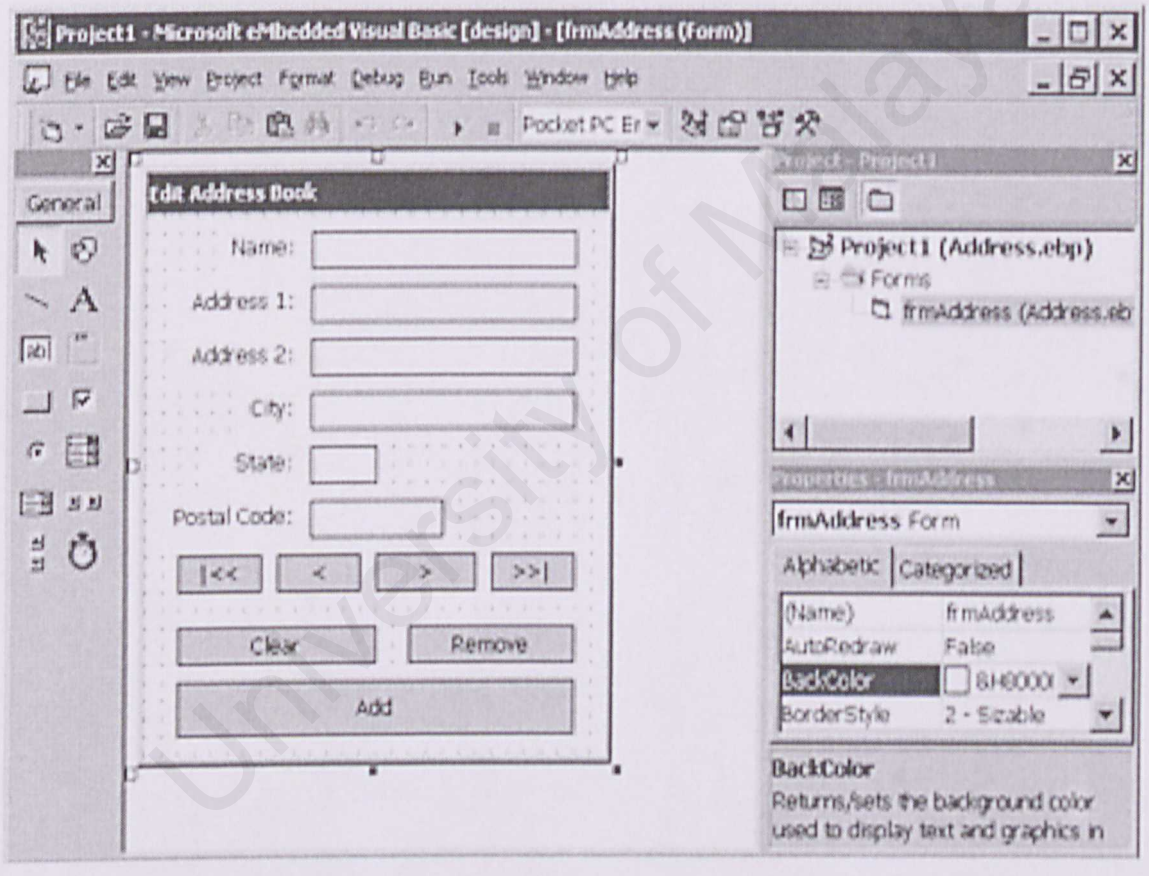

Figure 3.4 A screenshot of eMbedded Visual Basic IDE

eVB offers a integrated development environment which is very similar to Visual Basic 6.0. This is to lower the steep in learning curve as developers familiar with Visual Basic 6.0 can make the switch easily. This is shown in Figure 3.4..

Other features of eVB include the support for a set of remote tools such as Remote Registry Editor, File Viewer and Process Viewer. All these tools are useful when to administrate the development of a Windows CE-based device remotely. Also included in the eVB is an emulator to test the application.

Chapter Four:

System Analysis & Design

4.0 System Analysis and Design

4.1 System Architecture - Two-tiered Client/Server Architecture

A client/server system is a solution in which the presentation, presentation logic, application logic, data manipulation and data layers are distributed between client PCs and one or more server (Whitten, et al., 2002).

As the name suggests, it must consist of clients and at least a database server. In the two-tiered client/server computing, a server is used to host shared databases and execute all commands and services for the database, while clients are used as a front to access the data and the commands.

This architecture is chosen due to its advantage in

- Generating less traffic

  This architecture requires less network bandwidth as only database requests are sent and the record that are needed are actually transported over to and from the client

- Ensures database integrity

  The two-tiered architecture offers ease in maintaining database integrity as only the records used by a client is locked. This further allows other users to simultaneously work on other records in the same table or database.

4.2 Use Case

A use case specifies the behaviour of a system and is a description of a set of sequences of actions, including variants that a system performs to yield an observable result to an actor.

It is used to capture the intended behaviour of the system that is to be developed. It also provides a common understanding between the developers and users as well as serving to validate the architecture of the system.

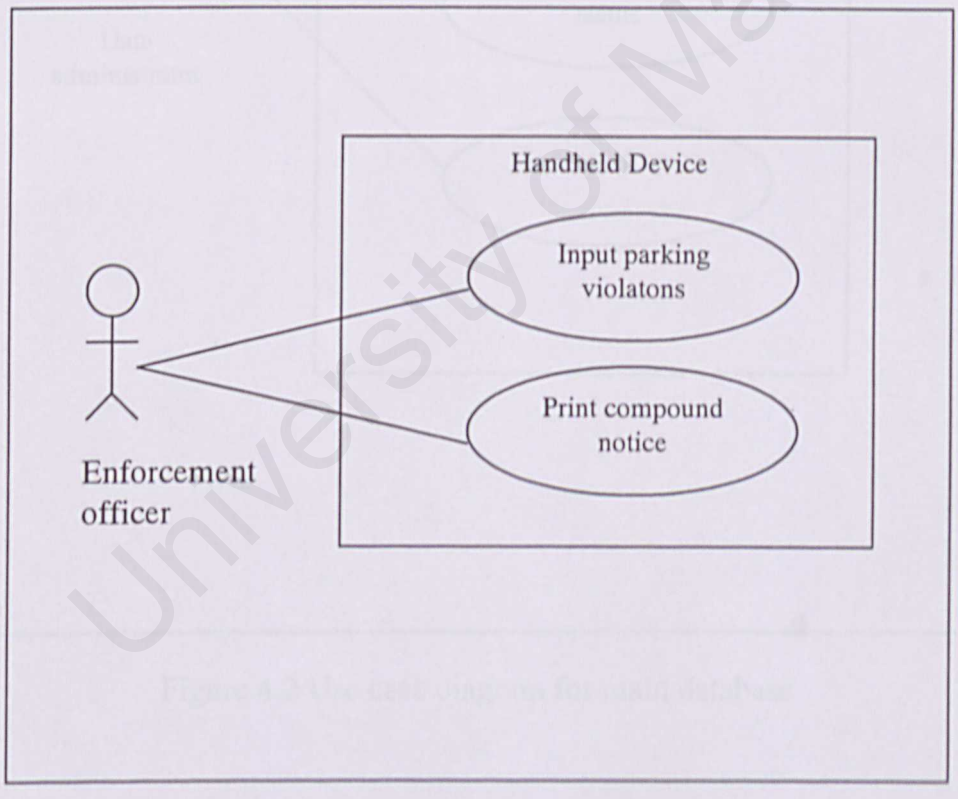

Figure 4.1 Use case diagram for handheld

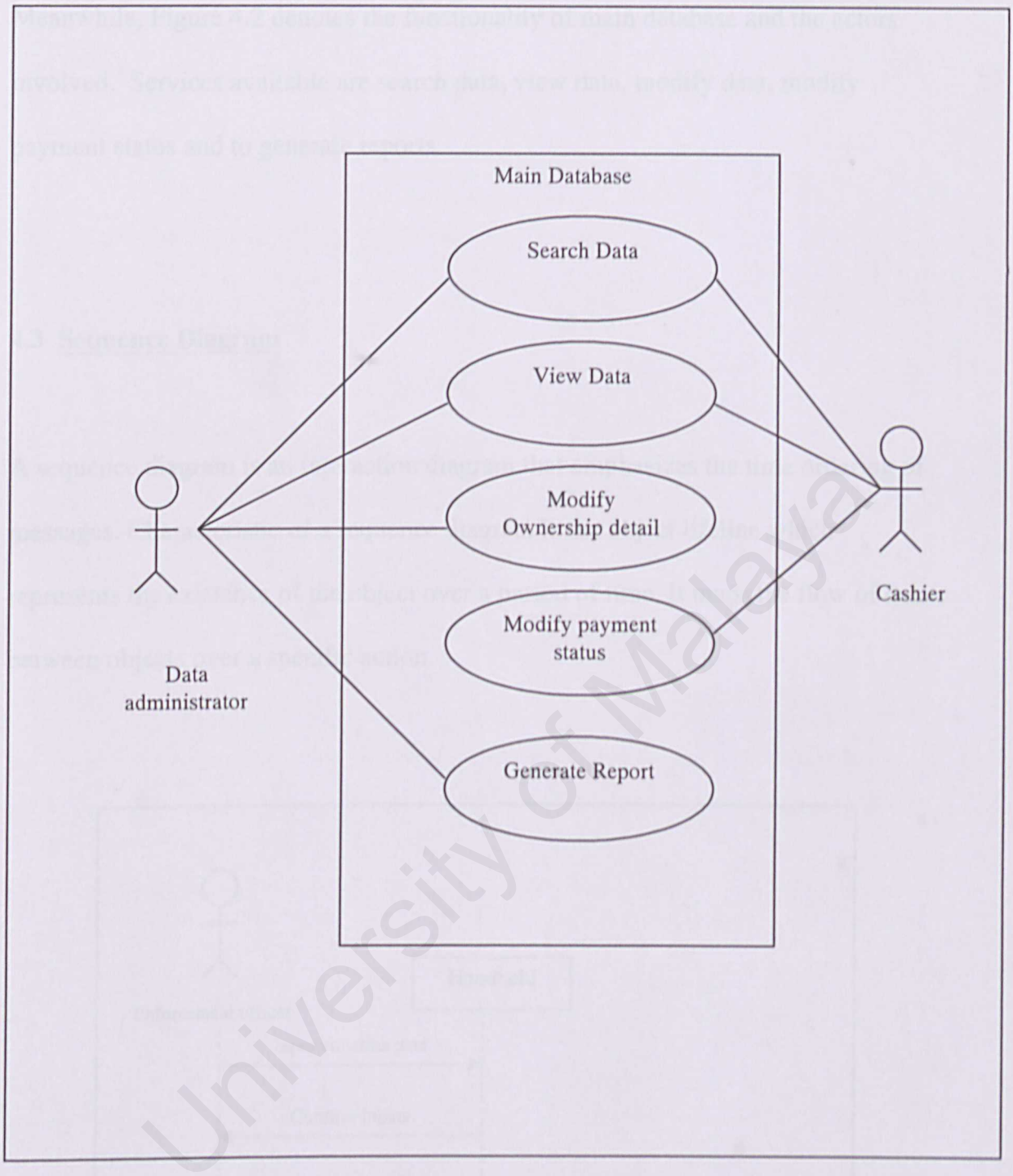

Figure 4.2 Use case diagram for main database

Figure 4.1 shows the services which are provided by the handheld to the enforcement officers. It should provide the two basic functionality, to input the details of offence and to print a compound notice.

Meanwhile, Figure 4.2 denotes the functionality of main database and the actors involved. Services available are search data, view data, modify data, modify payment status and to generate reports.

4.3 Sequence Diagram

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. Characteristic of a sequence diagram is the object lifeline which represents the existence of the object over a period of time. It maps the flow of data between objects over a specific action.

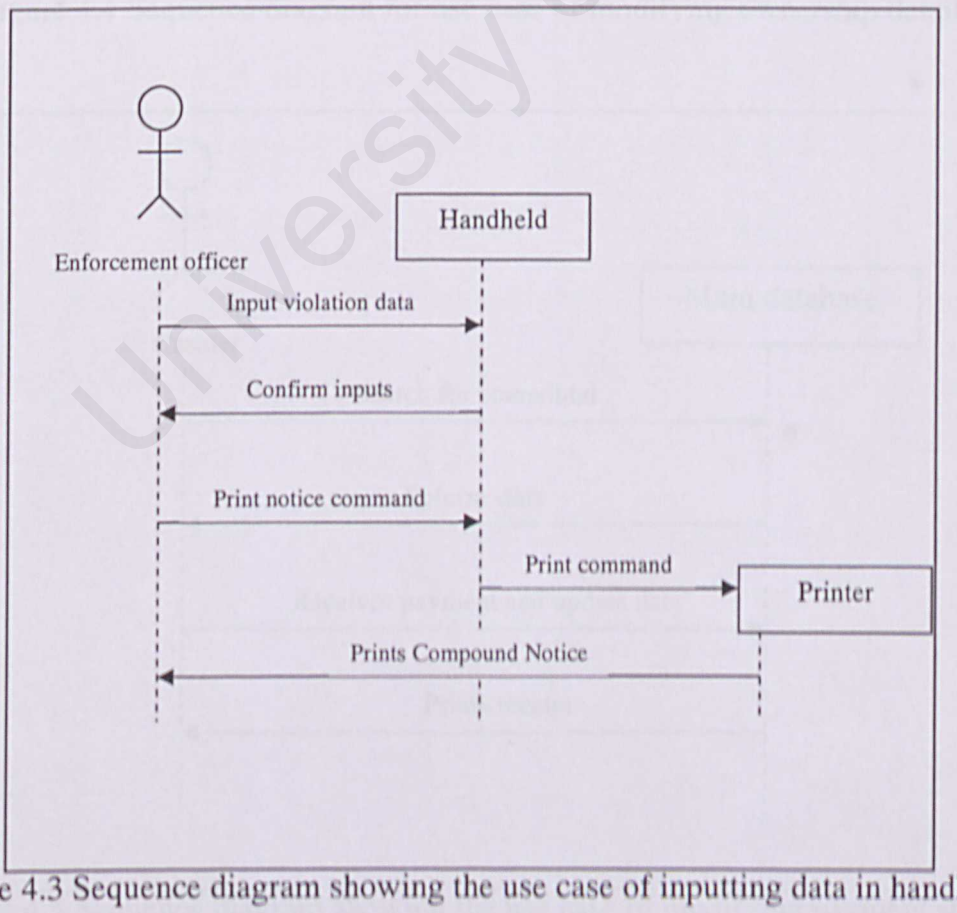

Figure 4.3 Sequence diagram showing the use case of inputting data in handheld

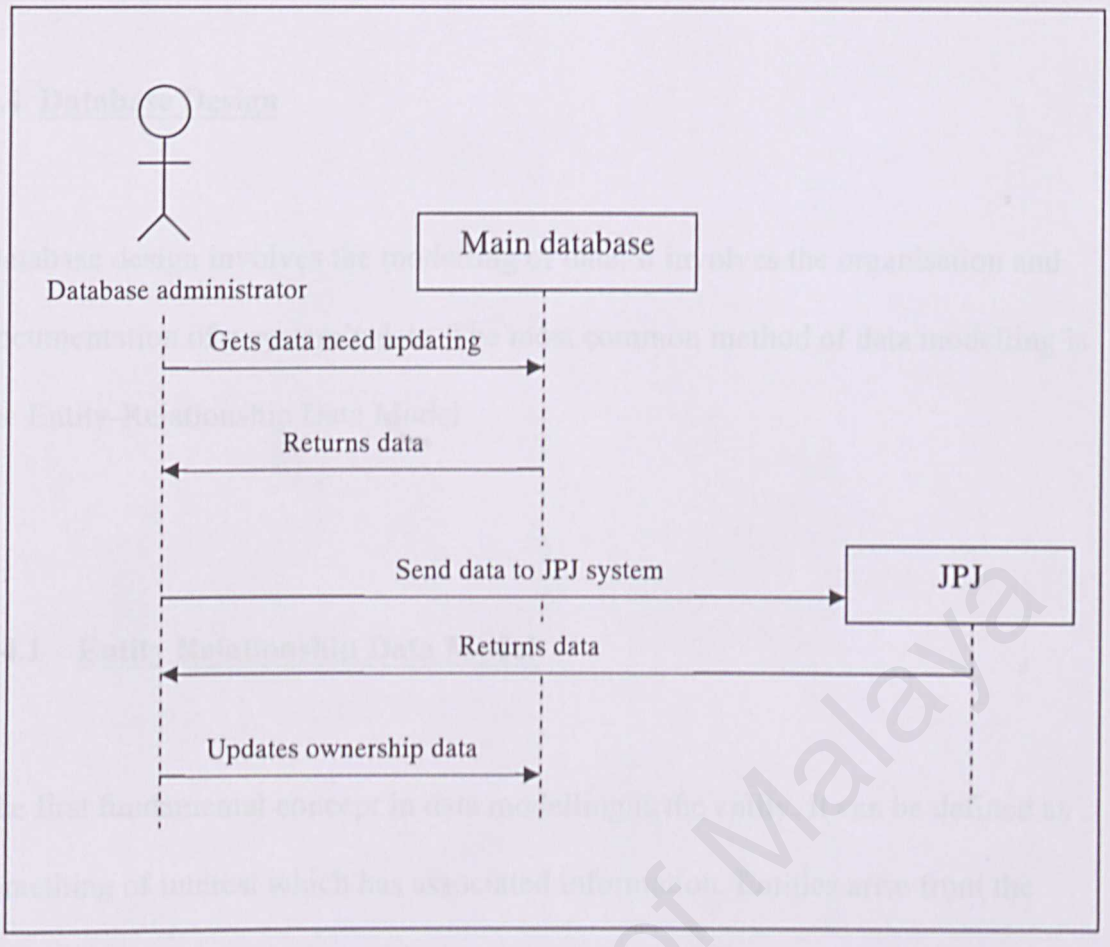

Figure 4.4 Sequence diagram for use case of modifying ownership details

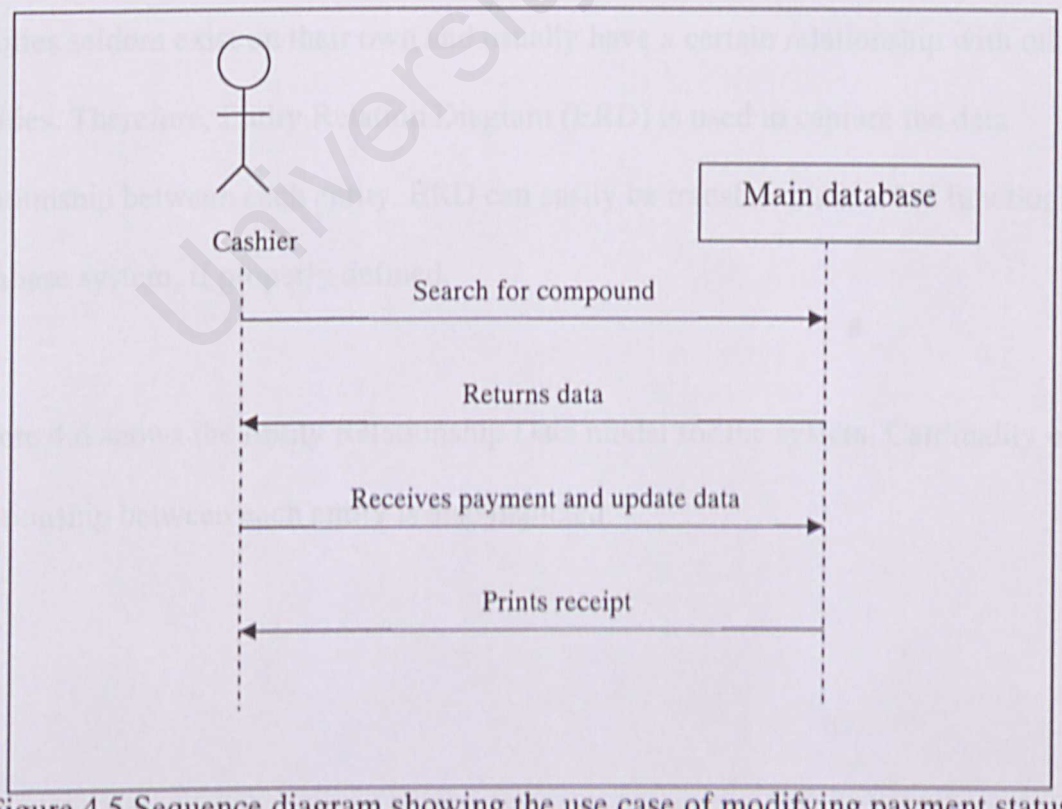

Figure 4.5 Sequence diagram showing the use case of modifying payment status

4.4 Database Design

Database design involves the modelling of data. It involves the organisation and documentation of a system's data. The most common method of data modelling is the Entity-Relationship Data Model

4.4.1 Entity Relationship Data Model

The first fundamental concept in data modelling is the entity. It can be defined as something of interest which has associated information. Entities arise from the problem domain and corresponds to both tangible and abstract item.

Entities seldom exist on their own and usually have a certain relationship with other entities. Therefore, Entity Relation Diagram (ERD) is used to capture the data relationship between each entity. ERD can easily be translated into a full functioning database system, if properly defined.

Figure 4.6 shows the Entity Relationship Data model for the system. Cardinality of relationship between each entity is also depicted.

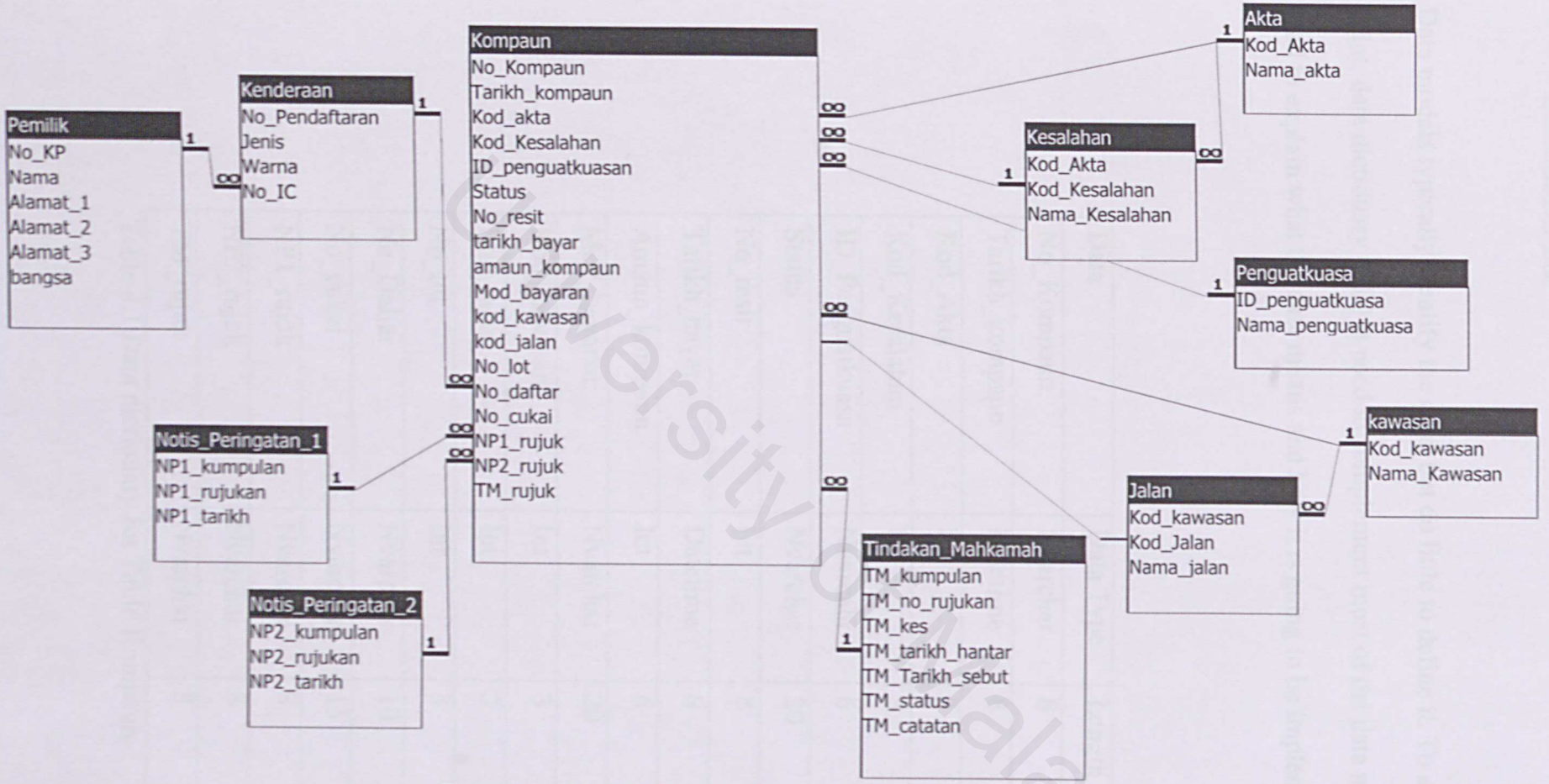

Figure 4.6 Entity Relationship diagram

4.4.2 Data Dictionary

Data models typically identify the data, but do little to define it. To address this point, data dictionary (DD) is used to complement most of the data models. DD is used to explain what the data means and how it is going to be implemented.

Data	Data Type	Length
No_Kompaun	Nvarchar	8
Tarikh_kompaun	Datetime	8
Kod_Akta	Int	3
Kod_Kesalahan	Int	3
ID_Penguatkuasa	Nvarcahr	6
Status	Nvarchar	20
No_resit	Int	8
Tarikh_bayar	Datetime	8
Amaun_kompaun	Int	6
Mod_bayaran	Nvarchar	20
Kod_kawasan	Int	3
Kod_jalan	Int	3
No_lot	Int	3
No_Daftar	Nvarchar	10
No_cukai	Nvarchar	15
NP1_rujuk	Nvarchar	8
NP2_rujuk	Nvarchar	8
TM_rujuk	Nvarchar	8

Table 4.1 Data dictionary for Table Kompaun

Data	Data Type	Length
No_KP	Nvarchar	14
Nama	Nvarchar	25
Alamat_1	Nvarchar	15
Alamat_2	Nvarchar	15
Alamat_3	Nvarchar	15
Bangsa	Nvarchar	10

Table 4.2 Data dictionary for Table Pemilik

Data	Data Type	Length
No_Pendaftaran	Nvarchar	10
Jenis	Nvarchar	25
Warna	Nvarchar	15
No_KP	Nvarchar	14

Table 4.3 Data dictionary for Table Kenderaan

Data	Data Type	Length
Kod_akta	Int	3
Nama_akta	Nvarchar	25

Table 4.4 Data dictionary for Table Akta

Data	Data Type	Length
Kod_Kesalahan	Int	3
Nama_Kesalahan	Nvarchar	25
Kod_Akta	Int	3

Table 4.5 Data dictionary for Table Kesalahan

Data	Data Type	Length
Kod_Kawasan	Int	3
Nama_Kawasan	Nvarchar	25

Table 4.6 Data dictionary for Table Kawasan

Data	Data Type	Length
Kod_Jalan	Int	3
Nama_Jalan	Nvarchar	25
Kod_Kawasan	Int	3

Table 4.7 Data dictionary for Table Jalan

Data	Data Type	Length
NP1_Kumpulan	Int	6
NP1_rujukan	Nvarchar	8
NP1_Tarikh	Datetime	8

Table 4.8 Data dictionary for Table Notis Peringatan 1

Data	Data Type	Length
NP2_Kumpulan	Int	6
NP2_rujukan	Nvarchar	8
NP2_Tarikh	Datetime	8

Table 4.9 Data dictionary for Table Notis Peringatan 2

Data	Data Type	Length
TM_Kumpulan	Int	6
TM_no_rujukan	Nvarchar	8
TM_Kes	Nvarchar	12
TM_Tarikh_hantar	Datetime	8
TM_Tarikh_sebut	Datetime	8
TM_Status	Nvarchar	15
TM_catatan	Nvarchar	40

Table 4.10 Data dictionary for Table Tindakan Mahkamah

Data	Data Type	Length
ID_Penguatkuasa	Nvarchar	6
Nama_Penguatkuasa	Nvarchar	25

Table 4.11 Data dictionary for Table Penguatkuasa

Data administrator have access to the following table

Table	Read	Write
Akta	X	X
Jalan	X	x
Kawasan	X	x
Kenderaan	x	
Kesalahan	x	
Kompaun	x	
Pemilik	x	x
Penguatkuasa	x	x

4.6 Graphical User Interface Designs

Most of the time, a system graphical user interface (GUI) is used as a metric for the system quality. This is because GUI can affect the user and the usage of the system tremendously.

A good GUI should provide the following advantages

- Increase effectiveness and efficiency

  User must be able to access the system in a congruent manner to their individual tasks and needs.

- Increase productivity

  GUI must be able to boost user productivity and not hinder it. This can be achieved by employing principle of ergonomics in designing the user interface and workspace.

- Minimise learning curve

  In introducing new work process using GUI, it is important that the interface retain most of its previous counterpart but at the same time employs automation to achieve its goal.

  Appendix B shows sample of forms currently used to issue compound notices.

To gain the benefits of these advantages, several rule-of-thumbs in interface design is practised.

- Consistency

  A system should act, look and operate the same throughout. It is the cardinal rule of all design activity and interface is no different. It can help to reduce requirements for human learning by allowing skills learned in one situation to be transferred to another similar to it.

- Error prevention and simple error handling

  The user interface must be able to perform simple error checking, significantly in data input. Suitable prompts must be provided to inform user of the incompatibility between the field and inputs.

- Simplicity

  Simplicity is gained by reducing the number of choices at any point in human-computer interaction. Where choices are obvious, drop-down menu or similar mechanism should be employed. Simplicity also implies the logical grouping of objects that belongs together.

Figure 4.7, 4.8 and 4.9 shows the interface design for e-Summon System's desktop application while Figure 4.10 shows the interface design for PDA application

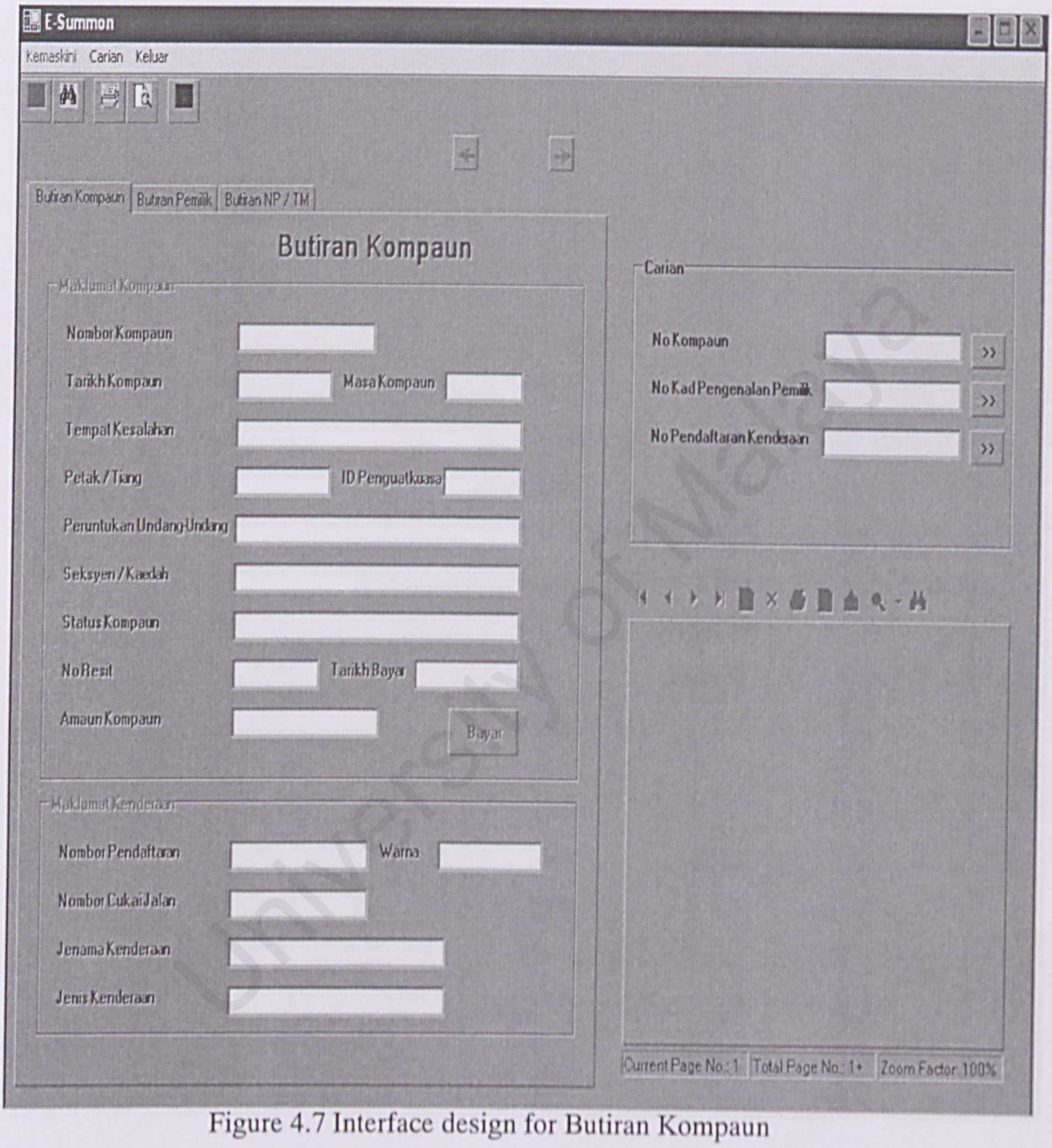

Figure 4.7 Interface design for Butiran Kompaun

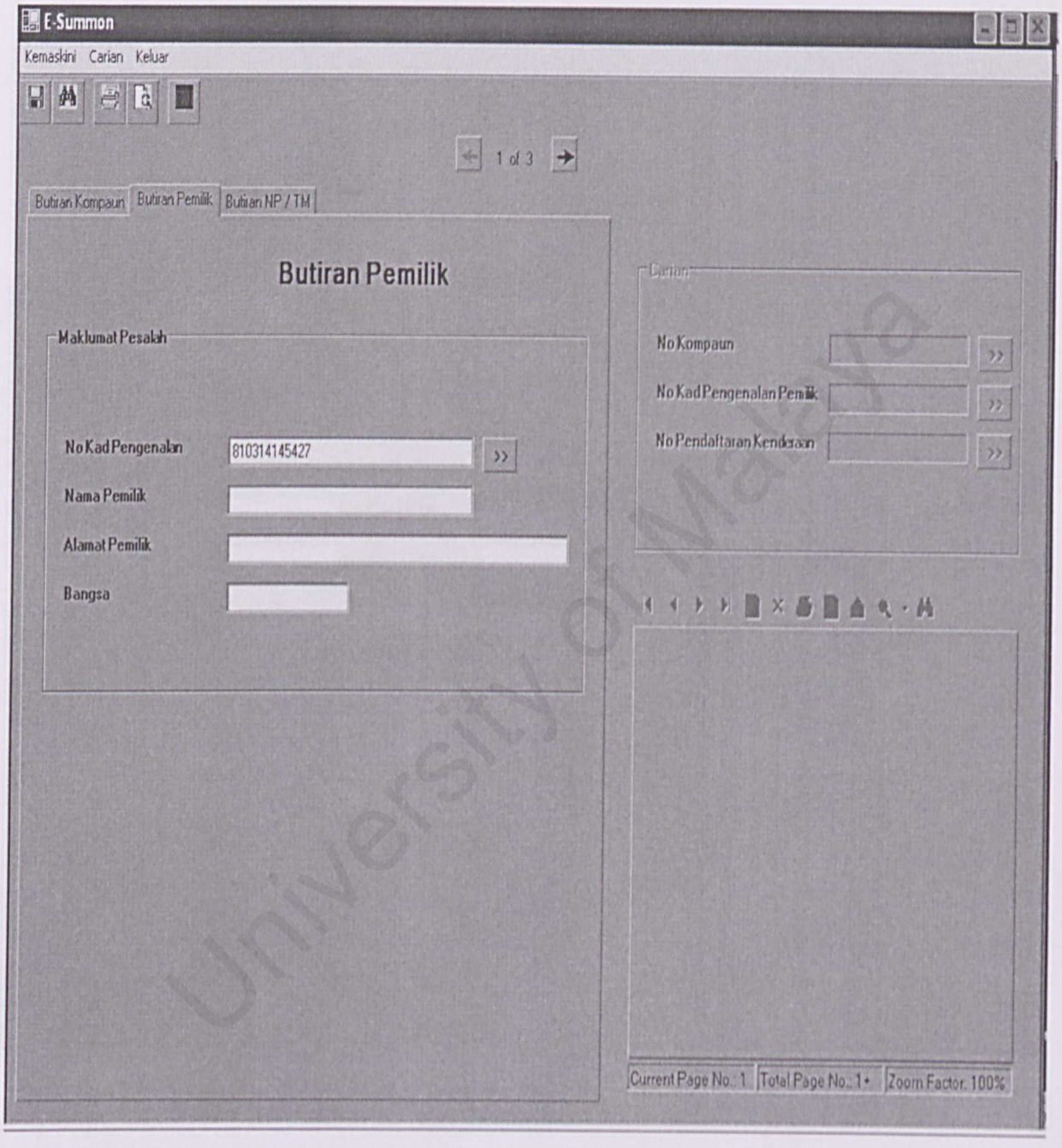

Figure 4.8 Interface design for Butiran Pesalah

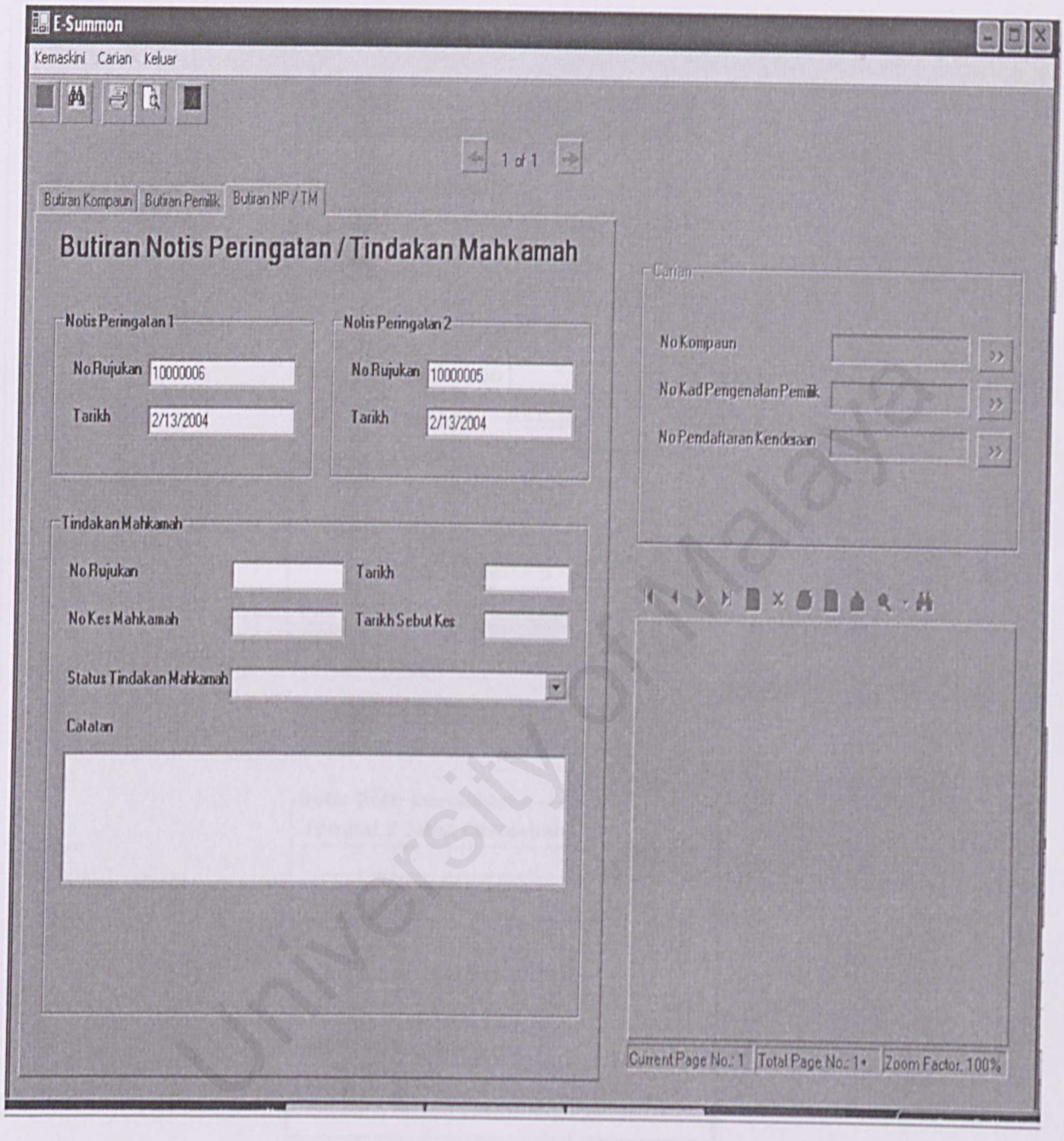

Figure 4.9 Interface design for Butiran Notis Peringatan / Tindakan Mahkamah

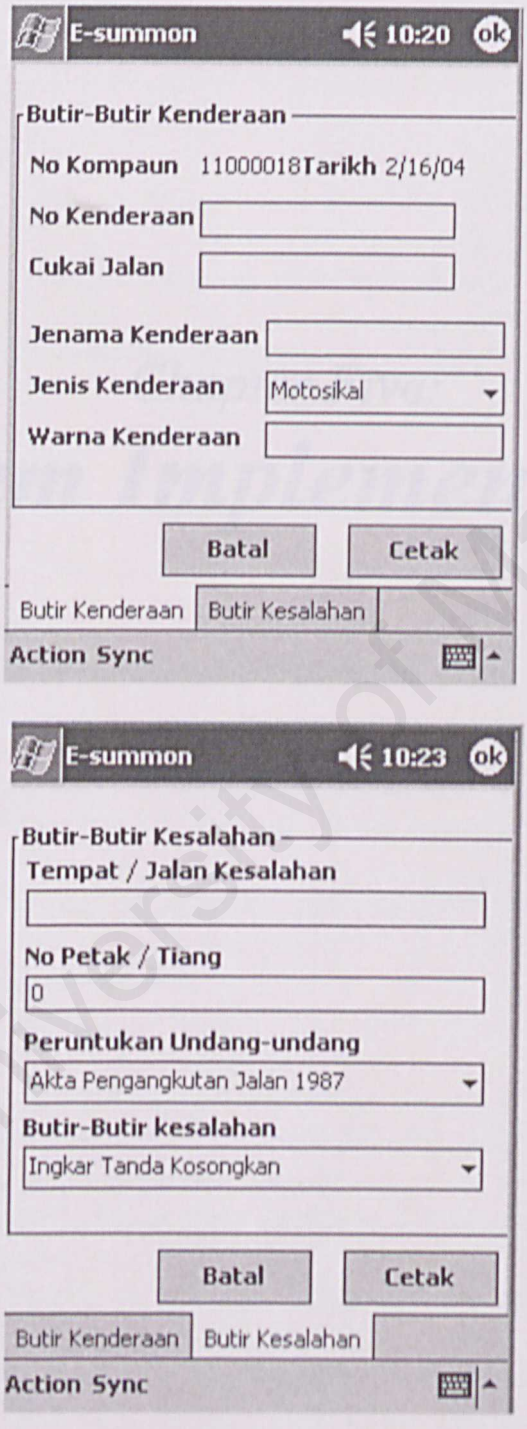

Figure 4.10 Interface design for handheld

Chapter Five:

System Implementation

5.0 System Implementation

System implementation in software development is a process to convert system requirement into program codes.

The initial stage of system implementation involves setting up the development environment. This includes setting up development tools to facilitate system implementation. The development environment is suited according to different development process.

5.1 Implementing Database - Microsoft SQL Server 2000

The first activity being carried out in implementation phase is database programming for the desktop application. This involves working with Microsoft SQL Server 2000 with service pack 3.

Database can be created using several methods. For the purpose of implementation, Enterprise Manager is used to create the e-Summon System database. The reason for this undertaking is that Enterprise Manager offers the best option in terms of ease-of-use. Tables are defined through a user-friendly interface. It also offers tools such as diagramming to help in defining the relationship between tables. This feature is important in order to preserve the integrity of the data in the tables.

The database was created based on the definition of the system requirement.

5.2 Implementing Database - Microsoft SQL Server CE 2.0

Microsoft SQL CE 2.0 (SQL CE) is used to power database support for PDA. Several adjustments of settings to the working computer system must be done to enable the replication of databases from SQL CE to SQL Desktop.

5.2.1 Microsoft SQL Server 2000

Microsoft SQL Server 2000 (SQL Server) must be installed in the desktop. For successful installation of SQL CE, the installer chosen must correspond to the service pack used in SQL Server (in this case Service Pack 3).

To enable a database from SQL Server for replication:

a) In **Enterprise Manager**, choose from menu **Tools** → **Replication** → **Create and Manage Publications ...**

b) Choose **Create Publication ...**

c) At the dialog box, choose the database for replication and click Next

d) Choose **Merge Publication** and click **Next**

e) Check the option of **Devices running SQL Server CE** and click **Next**

f) Choose the tables needed for replication and click **Next**

g) Provide a name for the publication. By default it will take the name of the database. Click **Next**

h) Choose the option of **Create the publication as specified** and click **Next.**

i) Click **Finish** to start the publication.

5.2.2 Installing Microsoft SQL Server CE 2.0

a) Run the SQL CE installer. Ensure that the installer corresponds to the service pack of Microsoft SQL Server 2000.

b) To install SQL CE into PDA, manually copy the files of **ssce20.dll**, **sscesa20.dll**, **ssceerror20en.dll** and **dllregister.exe** from the correct processor folder into the PDA by using Microsoft ActiveSync.

c) Run **dllregister.exe** from PDA to register the .dll files

5.2.3 Microsoft Loopback Adapter

a) Install **Microsoft Loopback Adapter**

b) In **Network Connection**, configure the **Network Bridge**'s **TCP/IP** to have an IP address (for example, 10.10.10.10) and **subnet mask** of **255.255.255.0**

Loopback Adapter works by creating a bridge in the local network card. Configuration is important to assign a IP address to the desktop machine.

Note: Pocket PC 2002 emulator will only work after the Loopback Adapter has been installed and configured.

5.2.4 Configuration of Internet Information Service (IIS)

a) To check if **IIS** is running, right-click on **My Computer** and choose **Manage**. Look for IIS under the folder of **Services and Application**. If IIS is not running, install it from the Windows Installation CD.

b) Once IIS is installed, create a new folder in **C:\inetpub\wwwroot** called '**ssce**'. Copy the file of **sscesa20.dll** from **C:\Program Files\Microsoft SQL Server CE 2.0\Server** into the new folder (**C:\inetpub\wwwroot\ssce**).

c) To configure IIS for SQL CE, use the **SQL Server CE Connectivity Management tool**.

d) Create a **new virtual directory** to point to the **sscesa20.dll** file in **C:\inetpub\wwwroot\ssce**.

e) Set the **HTTP Authentication** using the option of **Basic Authentication** and **Integrated Windows authentication**.

f) In **NTFS permission**, only allow **SQL Server CE Server Agent** to **'Read and Execute'**

g) To check for successful configuring of IIS, from the desktop machine and PDA, use Internet Explorer to access the virtual directory (http://<machine_name>/ssce/sscesa20.dll)

Internet Information Service (IIS) is an essential component in order for SQL CE to work. This is because SQL CE communicates and replicates with SQL Server by using the TCP/IP protocol. Thus, the machine hosting SQL Server must have its IIS enabled.

By using the Basic Authentication with Windows Authentication options, a user login into the virtual directory will be prompted for username and password. Username and password entered is the same as the one used to access Windows from the IIS machine.

5.2.5 Replication of data

To replicate data, use the Sync menu option as described in the User Manual (Appendix I)

5.3 Coding Implementation

Two development tools were used in coding e-Summon System. Microsoft eMBedded Visual Basic (eVB) is used to implement the PDA codes while Microsoft Visual Basic .NET (VB .NET) is used for desktop implementation. Both use the language of Visual Basic with eVB having some of its functionality stripped to suit the PDA environment.

5.3.1 eVB Implementation

A total of 5 forms and 4 modules were used in the implementation of e-Summon System PDA application. The forms along with a brief description of the usage are presented in Table 5.1, along with a list of main sub classes involved.

Figure 5.1 and Figure 5.2 show the source code used in initCmbUndang and initCmbSeksyen. This two source code shows how the combo box cmbSeksyen reacts to the changes in combo box cmbUndang.

Also listed are source code for replication of database between SQL Server CE and SQL Server 2000 (Figure 5.3).

Form Name	File Name	Description	Main Sub Classes
Login Refer to Appendix C	login.ebf	The first form accessed by user when application starts. Requires user to input id and password to check for authentication	6 ExitApp 7 InitMenuBar 8 LoginConnect 9 LoginDisconnect 10 wrongLogin
frmMain Refer to Appendix D	frmMain.ebf	The main form for user to issue compound notices. Consist of two tab pages, Butir Kenderaan and Butir Kesalahan.	• GetLastNoKompaun • InitMainApplication • InitMenuBar • Simpan • initCmbUndang • initSeksyen
frmReplAddSubscri ption Refer to Appendix E	frmReplAddSubscrip tion.ebf	One of four forms used for replication. This form is used to add file subscription	
frmReplInternetURL Refer to Appendix F	frmReplInternetURL. ebf	Another form used for replication. Requires user to input the internet URL to the host machine for replication	
frmReplSynchronize Pub Refer to Appendix G	frmReplSynchronize Pub.ebf	Form used in replication for user to input publication details such as name of server and database to replicate	
frmReplSynchronize Sub Refer to Appendix H	frmReplSynchronize Sub.ebf	Form for user to input details on subscriber, for the purpose of replication	

Table 5.1 Description of forms and main sub classes for e-Summon System PDA application.

```
Private Sub initCmbUndang()

    On Error Resume Next
    Dim oRS As ADOCE.Recordset
    Dim sSQL As String

    sSQL = "SELECT Nama_akta,kod_akta FROM akta order by kod_akta"

    'Connect if not connected

    If Not goADOcn Is Nothing Then
       localConnect
    End If

    'Execute the command

    Set oRS = goADOcn.Execute(sSQL)

    If oRS Is Nothing Then
       MsgBox "No recordset returned", vbOKOnly, "InitKenderaan
Error"
       Exit Sub
    End If

    'Loop through each record addint the Undang name to the
    'Undang combo box

    Do While Not oRS.EOF
       cmbUndang.AddItem oRS.Fields.Item("nama_akta").Value
       oRS.MoveNext
    Loop

    'Disconnect
    localDisConnect
    Set oRS = Nothing

    If Err.Number <> 0 Then
       Call ADOErrRoutine
       Exit Sub
    End If

    On Error GoTo 0

End Sub
```

Figure 5.2 Source code for sub class of initCmbUndang

```
Private Sub initseksyen()

    cmbSeksyen.Clear

    On Error Resume Next
    Dim oRS As ADOCE.Recordset
    Dim sSQL As String

    Dim clicked As Integer

    'takes the value of cmbUndang +1
    clicked = cmbUndang.ListIndex + 1
```

Figure 5.2 Source cod for sub class initseksyen (part 1 of 2)

```
    sSQL = "SELECT kod_akta, kod_kesalahan, nama_kesalahan FROM
    kesalahan WHERE kod_akta = " & clicked & " ORDER BY
    kod_kesalahan"

    'Connect if not connected

    If Not goADOcn Is Nothing Then
       LocalConnect
    End If

    'Execute the command

    Set oRS = goADOcn.Execute(sSQL)

    If oRS Is Nothing Then
       MsgBox "No recordset returned", vbOKOnly, "CmbUndang_click
Error"
       Exit Sub
    End If

    Do While Not oRS.EOF
       cmbSeksyen.AddItem oRS.Fields.Item("nama_kesalahan").Value
       oRS.MoveNext
    Loop

    'Disconnect
    localDisConnect

    If Err.Number <> 0 Then
       Call ADOErrRoutine
       Exit Sub
    End If

    On Error GoTo 0

    End Sub
```

Figure 5.2 Source cod for sub class initseksyen (part 2 of 2)

```
Private Sub cmdOK_Click()

    'Dim CEMerge As SSCE.Replication declared in Global Module

    Dim str As String

    'Set Subscriber Properties
    'value taken from frmReplSynchronizeSub
    CEMerge.Subscriber = txtSubscriber.Text
    CEMerge.SubscriberConnectionString =
       txtSubscriberConnectionString.Text

    'Call the Initialize, Run Terminate methods to synchronize the
    'subscription
    On Error Resume Next

    'show busy mouse icon
     ShowWaitCursor

   CEMerge.Initialize
```

Figure 5.3 Source code for SQL Server Ce replication (part 1 of 2)

```
    If CEMerge.ErrorRecords.Count > 0 Then
       HideWaitCursor
       ShowErrors CEMerge.ErrorRecords, "Initialization Failed"
    Else

    On Error Resume Next

    CEMerge.Run
    HideWaitCursor

    If CEMerge.ErrorRecords.Count > 0 Then
       ShowErrors CEMerge.ErrorRecords, "Synchronization Failed"
    Else
       str = "Synchronization Complete" & vbCrLf
       str = str & "Publisher Changes = " & CEMerge.PublisherChanges
          & vbCrLf
       str = str & "Publisher Conflicts = " &
          CEMerge.PublisherConflicts & vbCrLf
       str    =    str    &    "Subscriber    Changes    =    "    &
          CEMerge.SubscriberChanges & vbCrLf
       MsgBox str, vbOKOnly, " S Y N C H R O N I Z E "
    End If

    CEMerge.Terminate

    End If

    Me.Hide
    frmMainMenu.Show

End Sub
```

Figure 5.3 Source code for SQL Server Ce replication (part 2 of 2)

5.3.2 VB .NET Implementation

VB .NET is used to develop the e-Summon System desktop application. 3 forms and a global module was coded. The program also utilizes 2 Crystal Report forms. Table 5.2 list a brief description of each form and its relevant major sub classes.

Form	File Name	Description	Main Sub Classes
Login	login.vb	The first form accessed by user when application starts. Requires user to input id and password to check for authentication. This form also serves as a filter of access level and determines the form that the user will access.	• success • loginProb
mainEntry	mainEntry.vb	This form is access by two user roles; data clerk and cashier. The difference between the roles is the functionality accessed.	• refreshDisplay • kiraAmaunDibayar • cmdNextDB • cmdPrevDB • clearAll • cmdSearchKP • cmdSearchDaftar • NP1 • NP2 • TM • cmdBayar
frmAdmin	frmAdmin.vb	This form is accessed only by user with administrator clearance. Used to manage user IDs.	• simpan • tambah • display • carian

Table 5.2 Description of forms and main sub classes for e-Summon System desktop application.

5.3.2.1 VB .NET and Ms SQL Server 2000

In implementing SQL Server into VB .NET program, several settings are needed to enable the connection.

a) In VB .NET, right-click at **Data Connection** node in the **Server Explorer** window and click **Add Connection**

b) In the **Provider** tab of the window that appears, choose "**Microsoft OLE DB Provider for SQL Server**", which is the access driver for SQL Server databases.

c) In **Connection** tab, click the drop down button in the server name text box. Choose the name of the server from the drop down menu. If the SQL server name is not available, check that the server is running.

d) For information to log on to server, key in the user name and password (if applicable).

e) Select the appropriate database from the drop down menu and click **OK**. Now the database is listed as a connection in the Server Explorer.

f) Drag the database node into the Windows Form. This creates a **SQLConnection** to the source, which the Windows Form designer shows as **SQLConnection1**.

g) From the **Toolbox**'s **Data** subheading, drag **SQLDataAdapter** onto the designer form. This displays the **Data Adapter Configuration Wizard**, which configures the **SQLDataAdapter** instance with a custom query for populating a dataset.

h) Click **Next** to display a drop-down list of possible connections. Select the connection created in the previous step from the drop-down list and click **Next**

i) Use the default option of **Use SQL statement** in defining method for SQLDataAdapter to access the database. Click **Next**

j) Click the **Query Builder** button and select the necessary tables. Click **Add** to add the chosen tables. Click **OK** then **Finish**.

k) Drag **DataSet** from **Data** tab in the **ToolBox** into the form. This displays the **Add DataSet** window

l) Choose the **Untype DataSet (no schema)** to create a dataset with no schema. Click **OK**.

The above steps will cause VB .NET to auto-generate codes related to connecting to the SQL Server.

5.3.2.2 VB .NET and Crystal Report

e-Summon System utilizes Crystal Report by generating a ADO.NET database object and connecting it to the report. This is done by using the WriteXMLSchema function available in DataSet which generate a .xsd file. Refer to Figure 5.4 for a sample of source code as implemented in e-Summon System. By running the code, an XML schema will be generated. This method is necessary if the SQL statement used is complex

```
Dim ssql As String

'SQL Select Statement

ssql = "SELECT kompaun.*, kenderaan.*, pemilik.*,kesalahan.*,akta.*"
ssql = ssql & "FROM (akta INNER JOIN kesalahan ON akta.kod_akta =
  kesalahan.kod_akta) "

ssql = ssql & "INNER JOIN ((pemilik INNER JOIN kenderaan ON
  pemilik.no_kp_pesalah=kenderaan.no_kp)"

ssql = ssql & "INNER JOIN kompaun ON
   kompaun.no_pendaftaran=kenderaan.no_pendaftaran )"
ssql = ssql & "ON (kompaun.kod_kesalahan= kesalahan.kod_kesalahan)"

SqlDataAdapter1.SelectCommand.CommandText = ssql

SqlDataAdapter1.Fill(DataSet6)
DataSet6.WriteXmlSchema("c:\XMLSchema.xsd")
```

Figure 5.4 Source code for generating ADO.NET object for complex SQL statement

Below steps were taken in order to connect Crystal Report to the generated ADO.NET object

a) In the Visual Studio .NET **Solution Explorer**, right-click the project to display the shortcut menu.

b) Point to **Add** and click **Add New Item**.

c) In the Add New Item dialog box, select **Crystal Report** from the Templates area. Click **Open**.

d) In **Crystal Report Gallery**, choose **As a Blank Report** and click **OK**.

e) On **File** menu, click **Save** to save the report.

f) Right click in the Report Designer, point to **Database**, and click **Database Expert ...**

g) You'll be presented with **Database Expert** wizard.

h) In the Database Expert wizard, expand the **Create New Connection** folder, and double click the **ADO.NET (XML)** folder

i) In **XML File Path**, browse to the XML schema file generated earlier (XMLSchema.xsd) and click **Open**. Click **Finish**.

j) Add the XML file into **Selected** Tables.

k) Now in the **Database Fields** node of **Field Explorer** will show the table and all its fields

l) Drag and drop the fields onto the report and format them as required.

6.0 Testing

Several principles were applied during the test cycle of e-Summon System. The principles are:

- There should be a proper and thorough planning involved before the actual testing is done. This principal disregards the type of testing being carried out, be it unit or integration test
- All tests should be traceable to the user requirements itself. This is done by validating the system against user requirement
- Pareto principle: 80% of all undetected errors are traceable to 20% of all modules.

The text cycle for e-Summon System is done in three levels, unit testing, integration testing and system testing. Unit testing involves tests on individual modules while integration testing involves testing the modules as they work together. Integration testing is largely divided into two major parts to reflect the different applications for desktop and PDA. The system testing is done by ensuring that the whole system is working as required, from replicating data across platform to printing out the necessary reports.

6.1 Unit Testing of e-Summon System

Unit testing involves the testing of individual modules or small cluster of modules. The objective of unit testing is to absolutely exercise individual classes to ensure superiority in their design and implementation. Unit testing is used to find errors that often do not appear in integration testing. This is due to the smaller subset of input and decision paths are used at this level. The three main types of unit testing performed are ad-hoc testing and black-box testing

6.1.1 Ad-hoc testing

Ad-hoc testing is a method mainly used in the implementation phase of the system. This testing is unorganized and depends largely on the programmer. This unstructured method is ideal to detect problems while coding as it offers a high degree of flexibility. This can prevent errors from being carried forward into other part of the system while saving the cost due to early detection.

The disadvantage of ad-hoc testing is in the lack of tracking in terms of testing done. Therefore, ad-hoc testing is not as effective as a standalone tool and must be complimented with other form of testing.

6.1.2 Black-box Testing

Black-box testing is carried out by making an assumption that the inner working or logic structure of the code is unknown. The types of black-box testing used are:

- Error Guessing

  This approach is closely related to ad-hoc testing but is somehow more structured and organized. It involves writing test cases for parts of functions which tends to be erroneous.

- Boundary value Analysis

  This test is designed to check on errors involving boundary values of equivalent classes.

- Module Interface Testing

  This test is used to check if the values along the interface are correct as they relate to modules which call them.

6.2 Integration Testing

Integration testing for e-Summon System can largely be divided into two parts, the desktop application and the PDA application. Similar method of testing is used to test the two parts

- Top-down integration

  Due to the modularized nature of system implementation, the top-down approach is best suited to test the integration. This approach necessitates the test to begin from the highest-level of the main program and to gradually add stubs until the bottom is reached. Stubs are characterized as modules which are lesser in complexity. This method is useful to catch bugs that will only be revealed as the integration reaches the bottom.

6.3 System Testing

This testing is done to ensure the smooth running of the system as a whole. Several considerations were taken into account when performing system testing

- Specific scenario

  Several sets of scenario in using the system were designed for the purpose of testing. These scenarios depict the typical ways of using the system. These scenarios also include a complete sequence of functional task for each user profile. It involves work flow and basic set of functions the user needs to follow to produce a work result.

- Documentation Testing

  All examples in the user manual were tested to ensure that it is correct and gives the exact answers users will obtain when they run the examples.

Chapter Seven:

System Evaluation & Conclusion

7.0 System Evaluation and Conclusion

7.1 Problems encountered

Several setbacks were encountered in completing the e-Summon System project. All these problems were either solved or a workaround was achieved, which ultimately, serves as an invaluable experience for the author.

- Inexperience in embedded programming

  One of the major problems faced was inexperience in programming with PDA and Windows CE. This situation is worsened by the fact that there is a lack of local expertise in this area. One of the ways that the author took to solve this obstacle is by joining an embedded programming forum in the internet, http://groups.yahoo.com/embeddeddevelopers

- SQL Server CE and SQL Server 2000 replication

  The system development also hit a snag when getting the two SQL servers to replicate. Most of the guides available in the internet are sketchy at best, often assuming that readers of the material are advanced programmers.

- Time constraint

  Despite earlier project management planning being done, the system implementation phase took longer than anticipated. This is due some unforeseen circumstances that delayed the project

7.2 System Strength

e-Summon System offers several advantages to municipal councils who are its targeted users:

- Increase of efficiency

  The usage of e-Summon System is able to increase the efficiency of its user by cutting off several human-resource intensive activities such as data entry and paperwork. Its cashier component also helps in speeding up of payment-making and the record updating process.

- Minimization of human error in issuance of compound notices

  As discussed previously, the system is able to reduce some human-intensive activities, thereby minimizes the chance of human-error being involve. Furthermore the system also provides several automated data checking in ensuring that the details inputted are correct.

- Increase of effectiveness in enforcing the laws

  By having the database continuously up-to-date, users can easily identify and haul the traffic offenders to court for failing to pay the compound notices. This is facilitated by the features of updating the compound reminder notices and court action.

7.3 Limitations

There are still some limitations and constraints in the usage of the system.

- Period between synchronizing the data

  Despite reducing the need of human intervention in updating the compound notices, the system will still have an outdated database if the PDA component is not synchronized with the desktop. This is due to the need for PDA to be docked and replicated through physical network. A likely scenario that will cause problem is when a traffic offender tries to pay his compound before the data is synchronized from the PDA.

- Compound Notice Appeal

  The current system does not provide facilities for the traffic offenders to appeal their cases.

- Troublesome for public to check compound status

  In order to check the status of compound, the public will need to access the system through the municipal councils. This can be quite troublesome as the public needs to make their way to the municipal council office to do so.

- Manually updating of vehicle ownership details

  e-Summon System still requires the aid of a data clerk to input vehicle ownership detail. This is because such records are kept by the Road Transport Department.

7.4 Future Enhancements

Following is some suggested future enhancements to be undertaken in order to improve the system

- Wireless connectivity

  It is proposed that the system includes wireless connectivity for the PDA component in synchronizing the data with the desktop. This wireless connectivity can be achieved using, for example, either the WAP or GPRS services. With this feature, the database will have the data as soon as it is being entered in the PDA and will thus solve the current limitation. It is reckoned that this feature is possible with the advancement and the lessening of cost in wireless networking technology.

- Compound Notices Appeal module

  Another module can be added to handle appeal cases. This module with necessitate the system to create another role, the legal officer, to look into the appeal cases and make a decision based on details given by traffic offenders.

- Online/SMS checking of compound status

  The database can be made available in the World Wide Web to enable public to check the status of their compound. Another innovation will be to enable the checking of compound status through Short Messaging Services (SMS). However, considerations must be given on the issue of security.

- Interface with Road Transport Department (RTD)system

  To solve the need of data clerk in manually updating vehicle ownership details, it is proposed that future enhancements of e-Summon System include an interface to the RTD system. This will enable e-Summon System to have the latest information on the vehicles without having to tax its own database because the data will be maintained over at the RTD's side.

7.5 Conclusion

e-Summon System is a complete system designed for use by municipal councils in managing traffic offences. It encompasses every element of the current work process and improves it by harnessing the power of information technology. This system is beneficial in improving the management of traffic compound notices and ensures that appropriate legal actions are taken towards traffic offenders.

On a lesser note, by having effective law enforcement, it will also helps in creating a more law abiding citizens.

References

8.0 References

- Yahoo! Group embbeddeddevelopers

  http://groups.yahoo.com/embeddeddevelopers.

- CEWindows.NET

  http://www.cewindows.net

- Windows Mobile-Based Pocket PC Home Page

  http://www.microsoft.com/windowsmobile/products/pocketpc/default.mspx

- Windows CE.NET Homepage

  http://www.microsoft.com/windows/embedded/ce/default.asp

- Microsoft SQL Server : SQL Server CE Homepage

  http://www.microsoft.com/sql/ce/default.asp

- DevBuzz : SQL Server CE RDA Replication Configuration

  http://www.devbuzz.com/content/zinc_ss_ce_configuration_pg1.asp

- *MSDN Homepage*

  http://msdn.microsoft.com

- The Code Project

  http://codeproject.com/netcf/netcfgetstarted.asp

- vbCity.com – Developing Application for Windows CE – Addresio

  http://www.vbcity.com/pubs/article.asp?alias=addresio

- *Google.Com*

  http://www.google.com.my

- Deitel et al., (2002), Visual Basic .NET How to Program, 2nd, Prentice Hall

- Rob, P., & Coronel (1995), Database System Design, Implementation and Management, Boyd and Fraser

- Abdullah Embong (2000), Sistem Pangkalan Data : Konsep Asas, Rekabentuk dan Pelaksanaan, Tradisi Ilmu Sdn Bhd

- Pfleeger, S.L., (2001), Software Engineering : Theory and Practice, 2nd, Prentice Hall

- Whitten J.L, et al, (2001), System Analysis and Design Methods, 5th, McGraw-Hill

- Booch, G. et al, (2001), The Unified Modeling Language User Guide, Addison-Wesley

- Galitz W.O., (1996), Essential Guide to User Interface Design, Wiley Computer Publishing

- Preece, J., (1994), Human Computer Interaction, Addison-Wesley

- Olson, D.L., (2001), Introduction to Information Systems Project Management, McGraw-Hill International Editions

Appendix A

Interview Questions

Interview Questions

1) Bolehkah encik menjelaskan proses mengeluarkan saman terhadap kesalahan meletak kereta?
2) Apakah maklumat yang akan diambil semasa pengeluaran kompaun? Bolehkah encik menunjukkan satu contoh resit kompaun yang dikeluarkan?
3) Bagaimanakah nombor siri pada buku resit kompaun ditentukan?
4) Apakah kelemahan pada sistem yang ada sekarang?
5) Bagaimanakah dengan sistem pangkalan data yang ada pada majlis perbandaran? Bolehkah saya melihat dokumentasian yang berkenaaan dengan pangkalan data tersebut?
6) Berapakah bayaran yang dikenakan terhadap setiap saman yang dikeluarkan?
7) Apakah yang akan terjadi sekiranya kenderaan yang sama disaman dua kali dalam sehari?
8) Secara purata, berapakah kenderaan yang disaman dalam sehari?
9) Secara purata, berapakah kenderaan yang disaman oleh setiap pegawai?
10) Apakah cadangan encik dalam pembangunan sistem yang baru ini?

Appendix B

Samples of DBKL Compound Notice

ASAL

DEWAN BANDARAYA KUALA LUMPUR
DIREKTORAT PENGUATKUASAAN

DBKL/DP.4

No. F 000000

NOTIS KESALAHAN SERTA TAWARAN KOMPAUN

(1) No. Kenderaan :

(2) No. Cukai Jalan

(3) Jenama Kenderaan :.....

(4) Tempat / Jalan :.....

(5) No. Petak / Tiang :.....(Zon :.....)

(6) Tarikh kesalahan :.....Waktu (7) ..... Pg Ptg Mlm

Kepada Pemandu/Pemilik kenderaan bernombor seperti tersebut diatas. Dengan ini adalah diberitahu bahawa satu aduan untuk tindakan mahkamah akan dibuat terhadap tuan kerana kesalahan yang dilakukan seperti yang dinyatakan dibawah :

(8) **Peruntukan Undang-Undang.**

- ☐ **Akta Pengangkutan Jalan 1987**
- ☐ **Kaedah -Kaedah Lalulintas Jalan 1959**
- ☐ **Perintah Lalulintas Jalan [Letak Kereta] 1969/1984**
- ☐ **Akta Jalan, Parit Dan Bangunan 1974**
- ☐ **Undang-Undang Kecil Taman (W.P.K.L) 1981**
- ☐ .....

(9) Seksyen/Kaedah :.....

(10) No/Butir-Butir Kesalahan :.....

.....

(Rujuk senarai peruntukan undang-undang dan kesalahan dimuka sebelah belakang notis ini)

(11) ..... Tandatangan Pengadu/Warden Lalulintas

(12) ..... No. Anggota

(13) ..... Tarikh Notis dikeluarkan

TAWARAN KOMPAUN

Saya bersedia mengkompaunkan kesalahan yang ditandakan diatas dengan bayaran yang ditetapkan mengikut jadual dibawah ini.Kegagalan menjelaskan bayaran kompaun mengakibatkan tindakan mahkamah akan di ambil terhadap tuan/puan mengenai kesalahan yang dilakukan.

Jadual Pembayaran	Bayaran dalam Tempoh 14 hari	Bayaran selepas Tempoh 14 hari	Bayaran selepas 28 hari dan sebelum tindakan undang-undang
☐ Kesalahan tidak membayar caj meter	RM 10.00	RM 20.00	RM 25.00
☐ Motosikal	RM 10.00	RM 20.00	RM 25.00
☐ Kereta	RM 30.00	RM 50.00	RM 60.00
☐ Van/Lori Kecil/4WD	RM 40.00	RM 60.00	RM 70.00
☐ Lori Besar/Bas	RM 50.00	RM 75.00	RM 85.00

.....
(ROHAYAH BINTI KARIM)
Jabatan Hal Ehwal Undang-Undang
b.p Datuk Bandar Kuala Lumpur

(14)

[Diterima Jumlah Yang Dinyatakan Dengan Angka Disebelah]

DEWAN BANDARAYA KUALA LUMPUR
DIREKTORAT PENGUATKUASAAN

KERATAN UNTUK CATITAN BAYARAN KOMPAUN

Sila simpan keratan ini untuk rekod tuan jika kompaun telah dibayar.

Terima Kasih.

No. F 000000

PENERANGAN

SENARAI PERUNTUKAN UNDANG-UNDANG DAN KESALAHAN-KESALAHAN

AKTA PENGANGKUTAN JALAN 1987.

1.	Seksyen 79(2)	Ingkar Tanda Kosongkan
2.	Seksyen 79(2)	Ingkar Tanda Larangan Letak Kereta
3.	Seksyen 79(2)	Ingkar Tanda Larangan Berhenti

KAEDAH-KAEDAH LALULINTAS JALAN 1959.

4.	Kaedah 12(1)	Kurang 30 kaki dari simpang / selekoh.(L.N.166/59)
5.	Kaedah 12(2)	Kurang 10 kaki dari Pili Bomba. (L.N.166/59)
6.	Kaedah 12(3)	Kurang 30 kaki dari Perhentian Bas.(L.N.166/59)
7.	Kaedah 12(4)/16(1)	Menyebabkan Halangan.(L.N.166/59)
8.	Kaedah 16(2)	Tidak rapat dengan tepi jalan.(L.N.166/59)
9.	Kaedah 44	Letak kenderaan di siar kaki.(L.N.166/59)
10.	Kaedah 12(3)(c)	Letak dijalan yang ada dua garisan selari. (L.N.167/59)
11.	Kaedah 12(3A)	Letak kenderaan di garisan kuning.(L.N.167/59)

PERINTAH LALULINTAS JALAN [LETAK KERETA] 1969 / 1984.

12.	Perintah -/ 8(1)	Letak tanpa bayar caj meter / melebehi masa.
13.	Perintah 3(2) / 6	Letak diluar tempat letak kereta.
14.	Perintah 6(1) / 7(1)	Letak melintangi garisan petak letak kereta.
15.	Perintah 6(2) / 7(2)	Ingkar tanda & isyarat ditempat letak kereta
16.	Perintah 17(1) /15(1)	Menghalang petak letak kereta.

AKTA JALAN , PARIT DAN BANGUNAN 1974

17.	Seksyen 46	Memberhentikan kenderaan disiar kaki.

UNDANG-UNDANG KECIL TAMAN 1981

18.	UUK 3(i)	Merosakan tumbuhan-tumbuhan.
19.	UUK 7	Letak diluar tempat dikhaskan untuk letak kereta.

TEMPAT & WAKTU PEMBAYARAN

Bangunan DBKL,
Jalan Raja Laut,
50350 Kuala Lumpur.

Isnin-Jumaat : 7.45 pagi - 4.30 ptg
Sabtu : 7.45 pagi - 12.45 tgh

Direktorat Penguatkuasaan,
Dewan Bandaraya Kuala Lumpur,
Jalan Tun Razak, 50400 Kuala Lumpur.

Isnin-Khamis : 7.45 pagi - 4.30 ptg
Jumaat : 7.45 pagi -12.15 tgh
: 2.45 ptg - 4.30 ptg
Sabtu : 7.45 pagi -12.45 tgh

Semua Pejabat Cawangan,
Dewan Bandaraya Kuala Lumpur.

Isnin- Jumaat : 7.45 pagi - 4.30 ptg
Sabtu : 7.45 pagi -12.45 tgh

BAYARAN MELALUI POS

Bayaran dengan cara pos hendalah dibuat dalam bentuk Kiriman Wang Pos, Wang Pos, Pesanan Juruwang atau Draf Bank kepada Datuk Bandar Kuala Lumpur, Peti Surat 11022, Kuala Lumpur. Notis kesalahan ini hendaklah disertakan apabila membuat pembayaran.

SEKSYEN 115 AKTA PENGANGKUTAN JALAN 1987

Permintaan butir-butir mengenai pemandu / pemilik kenderaan motor pada masa kesalahan dilakukan.
Tuan / Puan dikehendaki memenuhi maklumat-maklumat yang diminta seperti dibawah:

Nama Pemilik / Pemandu :.....

No. Kad Pengenalan :.....

Alamat Kediaman :.....

.....Poskod :.....

Tarikh Pemilikan kenderaan : Dari ..... Hingga .....

No. Lesen Pemandu :..... Tarikh Tamat .....

No. Telefon :.....

.....

Tandatangan Pemilik/Pemandu Kenderaan

Appendix C

Source Code for login.ebf

```
Option Explicit On
Dim flg1 As Boolean, flg2 As Boolean
'MainFlag As String


Private Sub Form_Load()

    InitMenuBar()

    txtIDPenguatkuasa.Text = ""
    txtKatalaluan.Text = ""
    cmdTeruskan.Enabled = False


    'If gbLoginDBExists = False Then
    '  ShowWaitCursor
    ' InitFunc
    'HideWaitCursor
    'End If
End Sub
Private Sub InitMenuBar()
    mnuNew = MenuBar3.Controls.AddMenu("Action", "mnuNew")
    mnuNew.Items.Add(1, "mnuNewExit", "Exit")
End Sub

Private Sub MenuBar3_MenuClick(ByVal Item As MenuBarLib.Item)
    Select Case Item.Key
        Case "mnuNewExit"
            ExitApp()
    End Select
End Sub
Private Sub txtIDPenguatkuasa_Change()

    If txtIDPenguatkuasa.Text <> "" Then
        flg1 = True
    End If

    If flg1 = True Then
        If flg2 = True Then
            cmdTeruskan.Enabled = True
        End If
    End If


End Sub


Private Sub txtKatalaluan_Change()
    If txtKatalaluan.Text <> "" Then
        flg2 = True
    End If


    If flg1 = True Then
        If flg2 = True Then
            cmdTeruskan.Enabled = True
        End If
    End If
End Sub
Sub LoginConnect()
    On Error Resume Next
    goADOcn.Close()
    Err.Clear()
    goADOcn.Open()
    If Err.Number <> 0 Then
        Call ADOErrRoutine()
    End If
    On Error GoTo 0
```

```
End Sub

Sub LoginDisConnect()
    On Error Resume Next
    goADOcn.Close()
    If Err.Number <> 0 Then
        Call ADOErrRoutine()
    End If
    On Error GoTo 0
End Sub
Private Sub cmdTeruskan_Click()
    Dim name As String
    Dim pwd As String
    Dim lSQL As String

    Dim lRS As ADODB.Recordset

    name = txtIDPenguatkuasa.Text
    pwd = txtKatalaluan.Text

    SetADOConnectionsTest()

    LoginConnect()

    lSQL = "SELECT katalaluan FROM penguatkuasa WHERE
    ID_Penguatkuasa LIKE '" & name & "' "

    'MsgBox (lSQL)


    On Error Resume Next
    lRS = goADOcn.Execute(lSQL)


    If lRS Is Nothing Then
        HideWaitCursor()

        MsgBox("Login Gagal", vbOKOnly, "Login Gagal")
        Exit Sub
    Else
        Dim sTemp As String
        sTemp = lRS.Fields.Item("Katalaluan").Value
        'MsgBox (sTemp)

        It pwd = sTemp Then
            ReleaseADOConnectionsTest()
            lRS = Nothing
            LoginDisConnect()
            pubID = name

            frmMain.Show()

            Me.Hide()


        Else
            wrongLogin()
        End If



    End If

    If Err.Number <> 0 Then
        HideWaitCursor()
        Call ADOErrRoutine()
        Exit Sub
    End If

```

```
138     On Error GoTo 0
139
140
141
142
143     'ReleaseADOConnectionsTest
144     'lRS.Close
145
146
147     LoginDisConnect()
148
149     'frmMain.Show
150     'Me.Hide
151
152
153 End Sub
154 Private Sub wrongLogin()
155     MsgBox("ID Penguatkuasa atau Katalaluan yang salah")
156     txtKatalaluan.Text = ""
157
158 End Sub
159 Private Sub ExitApp()
160     'Clean up the application
161     On Error Resume Next
162     goADOcn.Close()
163     goADOcn = Nothing
164     goADOrs = Nothing
165     goADOXcat = Nothing
166     App.End()
167 End Sub
168
```

Appendix D

Source Code for frmMain.ebf

```
1
2 Option Explicit On
3
4
5
6 Private Sub cmbUndang_Click()
7
8     initseksyen()
9
10    cmbSeksyen.ListIndex = 0
11
12
13 End Sub
14
15 Private Sub initseksyen()
16
17    cmbSeksyen.Clear()
18
19    On Error Resume Next
20    Dim oRS As ADOCE.Recordset
21    Dim sSQL As String
22
23    Dim clicked As Integer
24
25    'Dim temp As String
26
27
28
29    clicked = cmbUndang.ListIndex + 1
30
31
32
33    sSQL = "SELECT kod_akta, kod_kesalahan, nama_kesalahan FROM      ✔
      kesalahan WHERE kod_akta = " & clicked & " ORDER BY              ✔
      kod_kesalahan"
34
35
36    'MsgBox (temp)
37
38    'Nama_akta,kod_akta FROM akta order by kod_akta"
39
40    'Connect if not connected
41
42    If Not goADOcn Is Nothing Then
43       localConnect()
44    End If
45
46    'Execute the command
47    oRS = goADOcn.Execute(sSQL)
48    If oRS Is Nothing Then
49       MsgBox("No recordset returned", vbOKOnly, "CmbUndang_click   ✔
      Error")
50       Exit Sub
51    End If
52
53    'Loop through each record addint the customer name to the
54    'Customer combo box
55    Do While Not oRS.EOF
56       cmbSeksyen.AddItem(oRS.Fields.Item("nama_kesalahan").Value)
57       oRS.MoveNext()
58    Loop
59    'Disconnect
60    localDisConnect()
61    If Err.Number <> 0 Then
62       Call ADOErrRoutine()
63       Exit Sub
64    End If
65    On Error GoTo 0
66
```

```
End Sub

Private Sub cmdBatal1_Click()
    InitKenderaan()
End Sub

Private Sub cmdBatal2_Click()
    InitKenderaan()
End Sub

Private Sub cmdCetak1_Click()
    check()

End Sub

Private Sub cmdCetak2_Click()
    check()
End Sub
'This is the code for the Main portion of the application
Private Sub Form_Load()
    frmMain.Visible = False
    'StartupOnlyInit

    'Ensure all frames are correct size
    Dim itop As Integer
    Dim ileft As Integer
    Dim iwidth As Integer
    Dim iheight As Integer
    itop = TabStrip1.ClientTop
    ileft = TabStrip1.ClientLeft
    iwidth = TabStrip1.ClientWidth
    iheight = TabStrip1.ClientHeight
    fraKenderaan.Top = itop
    fraKenderaan.Left = ileft
    fraKenderaan.Width = iwidth
    fraKenderaan.Height = iheight
    fraKesalahan.Top = itop
    fraKesalahan.Left = ileft
    fraKesalahan.Width = iwidth
    fraKesalahan.Height = iheight
    'fraLogin.Top = itop
    'fraLogin.Left = ileft
    'fraLogin.Width = iwidth
    'fraLogin.Height = iheight

    InitReplRDA()


    'Set correct zorder for frames
    fraKenderaan.ZOrder(vbBringToFront)

    'Initialize the MenuBar
    InitMenuBar()

    'Then initialize the application
    gbFirstTimeStartUp = False
    InitMainApplication()

End Sub
Private Sub disableTab()
    TabStrip1.Enabled = False
End Sub
Private Sub Form_OKClick()
    ExitApp()
End Sub

Private Sub lblDate_Click()

End Sub
```

```
136
137 Private Sub TabStrip1_Click()
138     'Reposition the correct Frame
139     Select Case TabStrip1.SelectedItem.Key
140         Case "Kenderaan"
141             fraKenderaan.ZOrder(vbBringToFront)
142         Case "Kesalahan"
143             fraKesalahan.ZOrder(vbBringToFront)
144     End Select
145 End Sub
146
147 'This initializes the menu bar
148 Private Sub InitMenuBar()
149     'Set this at design time
150
151     mnuNew = MenuBar1.Controls.AddMenu("Action", "mnuNew")
152     'mnuNew.Items.Add 1, "mnuLogOut", "Log Out"
153     mnuNew.Items.Add(1, "mnuNewExit", "Exit")
154
155
156     mnuSynchronize = MenuBar1.Controls.AddMenu("Sync", "
        mnuSynchronize")
157     mnuSynchronize.Items.Add(1, "mnuSynchronizeAddSubscription", "
        Add Subscription")
158     mnuSynchronize.Items.Add(2, "mnuSynchronizeReplicateData", "
        Replicate Data")
159 End Sub
160
161 'This responses to the menubar selections
162 Private Sub MenuBar1_MenuClick(ByVal Item As MenuBarLib.Item)
163
164     Select Case Item.Key
165         Case "mnuNewExit"
166             ExitApp()
167         Case "mnuLogOut"
168             Me.Hide()
169             Login.Show()
170
171         Case "mnuSynchronizeAddSubscription"
172             frmReplAddSubscription.Show()
173         Case "mnuSynchronizeReplicateData"
174             frmReplInternetURL.Show()
175     End Select
176
177 End Sub
178 Private Sub ExitApp()
179     'Clean up the application
180     On Error Resume Next
181     goADOcn.Close()
182     goADOcn = Nothing
183     goADOrs = Nothing
184     goADOXcat = Nothing
185     App.End()
186 End Sub
187 Public Sub InitMainApplication()
188     'Get the current OrderID
189
190
191     giCurrentNoKompaun = GetLastNoKompaun() + 1
192
193     'Initialize the Kenderaan info
194     InitKenderaan()
195
196
197
198 End Sub
199
200 Private Sub InitKenderaan()
201     lblNoKompaun.Caption = CStr(giCurrentNoKompaun)
```

```
202
203     lblTarikh.Caption = FormatDateTime(Now, vbShortDate)
204     txtNoKend.Text = ""
205     txtNoCukai.Text = ""
206     txtJenamaKend.Text = ""
207     cmbJenisKenderaan.Text = ""
208     txtWarnaKend.Text = ""
209
210     InitKesalahan()
211
212
213     If cmbUndang.ListCount = 0 Then
214
215         initcmbundang()
216
217     End If
218
219     If cmbJenisKenderaan.ListCount = 0 Then
220         initcmbJenisKenderaan()
221     End If
222
223
224
225
226     cmbUndang.ListIndex = 0
227     cmbUndang_Click()
228
229     cmbJenisKenderaan.ListIndex = 0
230
231
232
233 End Sub
234 Private Sub initcmbJenisKenderaan()
235
236     On Error Resume Next
237     Dim oRS As ADOCE.Recordset
238     Dim sSQL As String
239
240     sSQL = "SELECT nama_kenderaan, kod_jK FROM jKenderaan ORDER BY   ↙
        kod_jk"
241
242     'Connect if not connected
243
244     If Not goADOcn Is Nothing Then
245         localConnect()
246     End If
247
248     'Execute the command
249     oRS = goADOcn.Execute(sSQL)
250     If oRS Is Nothing Then
251         MsgBox("No recordset returned", vbOKOnly, "                  ↙
        InitcmbJenisKenderaan Error")
252         Exit Sub
253     End If
254
255     'Loop through each record addint the customer name to the
256     'Customer combo box
257     Do While Not oRS.EOF
258         cmbJenisKenderaan.AddItem(oRS.Fields.Item("nama_Kenderaan"). ↙
        Value)
259         oRS.MoveNext()
260     Loop
261     'Disconnect
262     localDisConnect()
263     oRS = Nothing
264
265     If Err.Number <> 0 Then
266         Call ADOErrRoutine()
267         Exit Sub
```

```
268     End If
269     On Error GoTo 0
270
271
272 End Sub
273 Private Sub initcmbseksyen()
274
275     On Error Resume Next
276     Dim oRS As ADOCE.Recordset
277     Dim sSQL As String
278
279     sSQL = "SELECT Nama_akta,kod_akta FROM akta order by kod_akta"
280
281     'Connect if not connected
282
283     If Not goADOcn Is Nothing Then
284         localConnect()
285     End If
286
287     'Execute the command
288     oRS = goADOcn.Execute(sSQL)
289     If oRS Is Nothing Then
290         MsgBox("No recordset returned", vbOKOnly, "InitKenderaan
        Error")
291         Exit Sub
292     End If
293
294     'Loop through each record addint the customer name to the
295     'Customer combo box
296     Do While Not oRS.EOF
297         cmbUndang.AddItem(oRS.Fields.Item("nama_akta").Value)
298         oRS.MoveNext()
299     Loop
300     'Disconnect
301     localDisConnect()
302     oRS = Nothing
303
304     If Err.Number <> 0 Then
305         Call ADOErrRoutine()
306         Exit Sub
307     End If
308     On Error GoTo 0
309
310
311 End Sub
312 Private Sub initcmbundang()
313
314     On Error Resume Next
315     Dim oRS As ADOCE.Recordset
316     Dim sSQL As String
317
318     sSQL = "SELECT Nama_akta,kod_akta FROM akta order by kod_akta"
319
320     'Connect if not connected
321
322     If Not goADOcn Is Nothing Then
323         localConnect()
324     End If
325
326     'Execute the command
327     oRS = goADOcn.Execute(sSQL)
328     If oRS Is Nothing Then
329         MsgBox("No recordset returned", vbOKOnly, "InitKenderaan
        Error")
330         Exit Sub
331     End If
332
333     'Loop through each record addint the customer name to the
334     'Customer combo box
```

```
335     Do While Not oRS.EOF
336         cmbUndang.AddItem(oRS.Fields.Item("nama_akta").Value)
337         oRS.MoveNext()
338     Loop
339     'Disconnect
340     localDisConnect()
341     oRS = Nothing
342
343     If Err.Number <> 0 Then
344         Call ADOErrRoutine()
345         Exit Sub
346     End If
347     On Error GoTo 0
348
349 End Sub
350
351 Private Sub InitKesalahan()
352
353     txtTempatKesalahan.Text = ""
354     txtPetakTiang.Text = "0"
355     cmbUndang.Text = ""
356     cmbSeksyen.Text = ""
357
358
359
360 End Sub
361
362 Public Function GetLastKompaunID() As Long
363     'Note: This function relies on the fact that the noKompaun's are ↙
        sequential and
364     'there are no holes in the sequential list of them. It also      ↙
        assumes that the
365     'MAX(no_kompaun) will be the next sequential OrderID for the     ↙
        local database
366
367     Dim iTemp As Long
368     GetLastKompaunID = 0
369
370     localConnect()
371
372     Dim oRS As ADOCE.Recordset
373     Dim sSQL As String
374
375     sSQL = "SELECT MAX(no_Kompaun) FROM noKompaun"
376
377     On Error Resume Next
378     oRS = goADOcn.Execute(sSQL)
379
380     'Catch errors
381
382     If Err.Number <> 0 Then
383         localDisConnect()
384         Call ADOErrRoutine()
385         Exit Function
386     End If
387
388     On Error GoTo 0
389     If oRS Is Nothing Then
390         MsgBox("No Recordset", vbOKOnly, "GetLastKompaunNo")
391     ElseIf IsNull(oRS.Fields.Item(0).Value) Then
392         MsgBox("Kompaun No error", vbOKOnly, "GetLastKompaunNo")
393     Else
394         iTemp = CLng(oRS.Fields.Item(0).Value)
395         GetLastKompaunID = iTemp
396     End If
397     oRS = Nothing
398     localDisConnect()
399 End Function
400
```

```
401 ' ADOCE error object handling routine
402 Private Sub ADOErrRoutine()
403     Dim ierror As Integer
404     Dim iparam As Integer
405     Dim ADOErr As ADOCE.Error
406     Dim sTemp As String
407     Dim strErr As String
408
409     If goADOcn.Errors.Count > 0 Then 'There was an ADO Error
410         For ierror = 0 To goADOcn.Errors.Count - 1
411             ADOErr = goADOcn.Errors(ierror)
412             strErr = "Desc: " & ADOErr.Description & vbCrLf
413             strErr = strErr & "Number: " & Hex(ADOErr.Number) &
    vbCrLf
414             strErr = strErr & "NativeErr: " & ADOErr.NativeError &
    vbCrLf
415             strErr = strErr & SSCEErrorDescription(ADOErr.
    NativeError, sTemp) & vbCrLf
416             strErr = strErr & sTemp & vbCrLf
417             '         For iparam = 0 To ADOErr.ErrorParameters.Count
     - 1
418             '             strErr = strErr & "param " & iparam & " = "
     & ADOErr.ErrorParameters(iparam) & vbCrLf
419             '         Next parm
420             strErr = strErr & "source = " & ADOErr.Source
421             MsgBox(strErr)
422         Next ierror
423         Err.Number = 0
424         Err.Clear()
425     Else 'There was a Windows CE Error
426         MsgBox("Err Number: " & Err.Number & vbCrLf & "Description:
    " & Err.Description, vbOKOnly, "CE Error")
427         Err.Clear()
428     End If
429
430 End Sub
431
432 Private Sub check()
433
434     If txtNoKend.Text = "" Then
435         swKend()
436         MsgBox("No Kenderaan tiada maklumat")
437         txtNoKend.SetFocus()
438     Else
439         If txtNoCukai.Text = "" Then
440             swKend()
441             MsgBox("No Cukai tiada maklumat")
442             txtNoCukai.SetFocus()
443         Else
444             If txtJenamaKend.Text = "" Then
445                 swKend()
446                 MsgBox("Jenama Kenderaan tiada maklumat")
447                 txtJenamaKend.SetFocus()
448
449             Else
450
451                 If cmbJenisKenderaan.Text = "" Then
452                     swKend()
453                     MsgBox("Jenis Kenderaan tiada maklumat")
454                     cmbJenisKenderaan.SetFocus()
455                 Else
456
457
458                     If txtWarnaKend.Text = "" Then
459                         swKend()
460                         MsgBox("Warna Kenderaan tiada maklumat")
461                         txtWarnaKend.SetFocus()
462                     Else
463                         If txtTempatKesalahan.Text = "" Then
```

```
464                             swKesalahan()
465                             MsgBox("Tempat / Jalan Kesalahan tiada   ↙
    maklumat")
466                             txtTempatKesalahan.SetFocus()
467                         Else
468                             If IsNumeric(txtPetakTiang.Text) = False ↙
     Then
469                                 swKesalahan()
470                                 MsgBox("Masukkan nombor petak /      ↙
    tiang yang sah")
471                                 txtPetakTiang.SetFocus()
472                             Else
473
474                                 If cmbUndang.Text = "" Then
475                                     swKesalahan()
476                                     MsgBox("Peruntukan Undang-undang ↙
     tiada maklumat")
477                                     cmbUndang.SetFocus()
478                                 Else
479
480                                     If cmbSeksyen.Text = "" Then
481
482                                         'If txtPetakTiang.Text = "0" ↙
     Then
483                                         swKesalahan()
484                                         MsgBox("Butiran Kesalahan    ↙
    tiada maklumat")
485                                         cmbSeksyen.SetFocus()
486
487                                         'End If
488
489
490                                     Else
491
492
493                                         If vbNo = MsgBox("Arahan ini ↙
     akan menyimpan rekod dan mencetak kompaun. Teruskan?", vbYesNo, ↙
     "Cetak") Then
    Exit Sub
495
496                                         Else : simpan()
497
498                                         End If
499
500                                     End If
501                                 End If
502
503                             End If
504                         End If
505                     End If
506                 End If
507             End If
508         End If
509     End If
510
511
512 End Sub
513
514 Private Sub swKend()
515
516     TabStrip1.SelectedItem = TabStrip1.Tabs.Item("Kenderaan")
517     'Move to the Kenderaan frame
518     fraKenderaan.ZOrder(vbBringToFront)
519     fraKenderaan.Refresh()
520
521 End Sub
522
523 Private Sub swKesalahan()
524
```

```
525     TabStrip1.SelectedItem = TabStrip1.Tabs.Item("Kesalahan")
526     'Move to the Kesalahan frame
527     fraKesalahan.ZOrder(vbBringToFront)
528     fraKesalahan.Refresh()
529 End Sub
530
531 Private Sub simpan()
532
533     'ShowWaitCursor
534     Dim sSQL As String
535
536     'variables used in SQL
537
538     Dim iCurrentOrderID As Long
539
540     Dim tarikhKompaun As Date
541     Dim masaKompaun As Date
542
543
544     Dim noKenderaan As String
545     Dim noCukai As String
546     Dim jenamaKenderaan As String
547
548     Dim warnaKenderaan As String
549
550     Dim tempatKesalahan As String
551     Dim undang As Integer
552
553
554
555     Dim seksyen As Integer
556     Dim jenisKenderaan As Integer
557     Dim petakTiang As Integer
558
559
560
561     Dim noKompaun As Long
562
563     'points to relevant fields
564
565     tarikhKompaun = FormatDateTime(Now, vbShortDate)
566
567
568     masaKompaun = FormatDateTime(Now, vbShortTime)
569
570
571
572     noKenderaan = txtNoKend
573     noCukai = txtNoCukai
574     jenamaKenderaan = txtJenamaKend
575     warnaKenderaan = txtWarnaKend
576     tempatKesalahan = txtTempatKesalahan
577
578     undang = cmbUndang.ListIndex + 1
579
580     seksyen = cmbSeksyen.ListIndex + 1
581
582     jenisKenderaan = cmbJenisKenderaan.ListIndex + 1
583
584
585     petakTiang = CInt(txtPetakTiang.Text)
586
587
588
589     noKompaun = lblNoKompaun.Caption
590     idPenguatkuasa = pubID
591     SetADOConnections()
592
593     localConnect()
```

```
594
595     Dim oRS As ADOCE.Recordset
596
597     sSQL = "SELECT * FROM kenderaan WHERE no_pendaftaran = '" &
        noKenderaan & "' "
598
599     On Error Resume Next
600     oRS = goADOcn.Execute(sSQL)
601     'MsgBox ("OK")
602
603     If oRS Is Nothing Then
604         'HideWaitCursor
605         MsgBox("No Recordset", vbOKOnly, " ")
606
607
608     ElseIf IsNull(oRS.Fields.Item(0).Value) Then
609
610         sSQL = "INSERT INTO kenderaan (no_pendaftaran,Jenama,warna,
        jenisKenderaan,no_Cukai) VALUES ( '" & noKenderaan & "', '" &
        jenamaKenderaan & "', '" & warnaKenderaan & "', '" &
        jenisKenderaan & "','" & noCukai & "' )"
611         On Error Resume Next
612         goADOcn.Execute(sSQL)
613
614         If Err.Number <> 0 Then
615             Call ADOErrRoutine()
616             Exit Sub
617         End If
618
619         On Error GoTo 0
620     Else
621
622         'MsgBox "OrderID Identity error", vbOKOnly, "GetLastOrderID"
623         sSQL = "UPDATE kenderaan SET Jenama = '" & jenamaKenderaan &
         "', warna = '" & warnaKenderaan & "', no_cukai =  '" & noCukai
        & "', jenisKenderaan =  '" & jenisKenderaan & "' WHERE
        no_pendaftaran = '" & noKenderaan & "' "
624         'MsgBox (sSQL)
625
626         On Error Resume Next
627         goADOcn.Execute(sSQL)
628
629         If Err.Number <> 0 Then
630             Call ADOErrRoutine()
631             Exit Sub
632
633
634         End If
635
636         On Error GoTo 0
637     End If
638     'MsgBox (tarikhKompaun)
639
640
641     sSQL = "INSERT INTO kompaun (no_kompaun, tarikh_kompaun,
        masa_kompaun, kod_akta, kod_kesalahan, petakTiang, kod_status,
        id_penguatkuasa, nama_kawasan, no_pendaftaran) "
642     sSQL = sSQL & "VALUES ('" & noKompaun & "', '" & tarikhKompaun &
         "', '1899-12-30 " & masaKompaun & "', " & undang & " , " &
        seksyen & ", " & petakTiang & " ,1, '" & idPenguatkuasa & "' , "
643     sSQL = sSQL & "'" & tempatKesalahan & "', '" & noKenderaan & "')
         "
644
645
646     'MsgBox (sSQL)
647
648     On Error Resume Next
649     goADOcn.Execute(sSQL)
650
```

```
651     If Err.Number <> 0 Then
652         Call ADOErrRoutine()
653         Exit Sub
654     End If
655
656     On Error GoTo 0
657
658     'goADOrs.Open "Kenderaan", goADOcn, adopenkeyset,
        adlockoptimistic, adcmdtabledirect
659
660
661     'goADOrs.AddNew
662     'goADOrs.Fields("no_pendaftaran") = noKenderaan
663     'goADOrs.Fields("jenis") = jenamaKenderaan
664     'goADOrs.Fields("warna") = warnaKenderaan
665
666     'picfile.Close
667
668     'goADOrs.Update
669
670     'rs.Close
671     'cn.Close
672     'Set rs = Nothing
673     'Set cn = Nothing
674
675     localDisConnect()
676
677     'HideWaitCursor
678
679     MsgBox("Simpanan ke Pangkalan Data Berjaya")
680     oRS = Nothing
681
682     ReleaseADOConnectionsTest()
683
684
685     InitMainApplication()
686     swKend()
687
688     'localConnect
689
690     'sSQL = "INSERT INTO no_pendaftaran " & _
691     '"(no_pendaftaran, jenis, warna) " & _
692     '"VALUES " & _
693     '"'" & noKenderaan & "'," & _
694     '"'" & noCukai & "'," & _
695     '"'" & warnaKenderaan & "'" & _
696     '")"
697
698     'MsgBox (sSQL)
699
700     '   "'" & ShipName & "'," & _
701     '   "'" & ShipAddress & "'," & _
702     '   "'" & ShipCity & "'," & _
703     '   "'" & ShipRegion & "'," & _
704     '   "'" & ShipPostalCode & "'," & _
705     '   "'" & ShipCountry & "', " & _
706     '   " " & Freight & " " & _
707     '    ")"
708
709
710     'On Error Resume Next
711     'goADOcn.Execute (sSQL)
712     'If Err.Number <> 0 Then
713     ' Call ADOErrRoutine
714
715     'Exit Sub
716     'End If
717     'On Error GoTo 0
718     'localDisConnect
```

```
719
720
721 End Sub
722
723 Public Sub SetADOConnections()
724     'Setup ADO objects.
725     goADOcn = CreateObject("ADOCE.connection.3.1")
726     goADOrs = CreateObject("ADOCE.recordset.3.1")
727     goADOXcat = CreateObject("ADOXCE.catalog.3.1")
728     'Setup ADOCE to connect to local SSCE DB.
729     goADOcn.ConnectionString = gcstrLocalConnect
730 End Sub
731
732 Public Function GetLastNoKompaun() As Long
733
734     Dim iTemp As Long
735
736     GetLastNoKompaun = 0
737
738     SetADOConnections()
739
740     localConnect()
741
742     Dim tRS As ADOCE.Recordset
743     Dim sSQL As String
744
745     sSQL = "SELECT MAX(no_kompaun) FROM Kompaun"
746
747
748
749
750     'Get the highest OrderID in the local database
751
752
753
754
755
756     On Error Resume Next
757     tRS = goADOcn.Execute(sSQL)
758     'MsgBox ("ok2")
759     'Catch Errors
760     If Err.Number <> 0 Then
761         localDisConnect()
762         Call ADOErrRoutine()
763         Exit Function
764     End If
765
766     On Error GoTo 0
767
768
769
770     If tRS Is Nothing Then
771         MsgBox("No Recordset", vbOKOnly, "GetLastNoKompaun")
772
773     ElseIf IsNull(tRS.Fields.Item(0).Value) Then
774         MsgBox("NoKompaun Identity error", vbOKOnly, "
        GetLastNoKompaun")
775     Else
776         iTemp = CLng(tRS.Fields.Item(0).Value)
777         GetLastNoKompaun = iTemp
778
779     End If
780     tRS = Nothing
781     localDisConnect()
782
783
784
785 End Function
786 Private Sub txtIDPenguatkuasa_Change()
```

```
787
788     If txtIDPenguatkuasa.Text <> "" Then
789         flg1 = True
790     End If
791
792     If flg1 = True Then
793         If flg2 = True Then
794             cmdTeruskan.Enabled = True
795         End If
796     End If
797
798
799 End Sub
800
801
802 Private Sub txtKatalaluan_Change()
803     If txtKatalaluan.Text <> "" Then
804         flg2 = True
805     End If
806
807
808     If flg1 = True Then
809         If flg2 = True Then
810             cmdTeruskan.Enabled = True
811         End If
812     End If
813 End Sub
814
815
```

Appendix E

Source Code for frmReplAddSubscription.ebf

```
1  Option Explicit On
2
3  Private Sub Form_Activate()
4      txtSubscriberConnectionString.Text = "data source=my documents\
       esummon.sdf"
5  End Sub
6
7  Private Sub cmdOK_Click()
8
9      CEMerge.SubscriberConnectionString =
       txtSubscriberConnectionString.Text
10
11     On Error Resume Next
12     If optCreateDatabase.Value = True Then
13         CEMerge.AddSubscription(CREATE_DATABASE)
14     Else
15         CEMerge.AddSubscription(EXISTING_DATABASE)
16     End If
17
18     If CEMerge.ErrorRecords.Count > 0 Then
19         ShowErrors(CEMerge.ErrorRecords, "Add Subscription Failed")
20     Else
21         MsgBox("Subscription Added", vbOKOnly, " A d d  S u b s c r
       i p t i o n ")
22     End If
23
24     Me.Hide()
25     frmMainMenu.Show()
26 End Sub
27
28 Private Sub cmdCancel_Click()
29     Me.Hide()
30     frmMainMenu.Show()
31 End Sub
32
33 Private Sub Form_Load()
34
35 End Sub
36
```

Appendix F

Source Code for frmReplInternetURL.ebf

```
1  Option Explicit On
2
3  Private Sub Form_Activate()
4
5      'Set Internet Properties
6      txtInternetURL.Text = CEMerge.InternetURL
7      txtInternetLogin.Text = CEMerge.InternetLogin
8      txtInternetPassword.Text = CEMerge.InternetPassword
9
10 End Sub
11
12 Private Sub cmdOK_Click()
13
14     Dim str As String
15
16     'Set Internet Properties
17     CEMerge.InternetURL = txtInternetURL.Text
18     CEMerge.InternetLogin = txtInternetLogin.Text
19     CEMerge.InternetPassword = txtInternetPassword.Text
20
21     'Go to next form in sequence
22     Me.Hide()
23     'If mnuItem = "mnuReplSynchronize" Then
24     frmReplSynchronizePub.Show()
25     'Else
26
27     'End If
28
29 End Sub
30
31 Private Sub cmdCancel_Click()
32     Me.Hide()
33     frmMainMenu.Show()
34 End Sub
35
36
```

Appendix G

Source Code for frmReplSynchronizePub.ebf

```
Option Explicit On

Private Sub Form_Activate()

    'Set Publisher and Distributor Properties
    txtPublisher.Text = CEMerge.Publisher
    txtPublisherDatabase.Text = CEMerge.PublisherDatabase
    txtPublication.Text = CEMerge.Publication
    If CEMerge.PublisherSecurityMode = DB_AUTHENTICATION Then
        optDBAuthentication.Value = True
    Else
        optDBAuthentication.Value = False
    End If
    txtPublisherLogin.Text = CEMerge.PublisherLogin
    txtPublisherPassword.Text = CEMerge.PublisherPassword

End Sub

Private Sub optDBAuthentication_Click()
    If optDBAuthentication.Value = True Then
        txtPublisherLogin.Enabled = True
        txtPublisherPassword.Enabled = True
    Else
        txtPublisherLogin.Enabled = False
        txtPublisherPassword.Enabled = False
    End If
End Sub


Private Sub optNTAuthentication_Click()

    If optDBAuthentication.Value = True Then
        txtPublisherLogin.Enabled = True
        txtPublisherPassword.Enabled = True
    Else
        txtPublisherLogin.Enabled = False
        txtPublisherPassword.Enabled = False
    End If
End Sub


Private Sub cmdOK_Click()

    Dim str As String

    'Set Publisher and Distributor Properties
    CEMerge.Publication = txtPublication.Text
    CEMerge.PublisherDatabase = txtPublisherDatabase.Text
    CEMerge.Publisher = txtPublisher.Text
    CEMerge.Distributor = txtPublisher.Text
    If optDBAuthentication.Value = True Then
        CEMerge.PublisherSecurityMode = DB_AUTHENTICATION
        CEMerge.DistributorSecurityMode = DB_AUTHENTICATION
    Else
        CEMerge.PublisherSecurityMode = NT_AUTHENTICATION
        CEMerge.DistributorSecurityMode = NT_AUTHENTICATION
    End If
    CEMerge.PublisherLogin = txtPublisherLogin.Text
    CEMerge.DistributorLogin = txtPublisherLogin.Text
    CEMerge.PublisherPassword = txtPublisherPassword.Text
    CEMerge.DistributorPassword = txtPublisherPassword.Text

    'Go to next form in sequence
    Me.Hide()
    'If mnuItem = "mnuReplSynchronize" Then
    frmReplSynchronizeSub.Show()
    'Else
    '   frmMainMenu.Show
    ' End If
```

```
70
71 End Sub
72
73 Private Sub cmdCancel_Click()
74     Me.Hide()
75     frmMain.Show()
76 End Sub
77
```

Appendix H

Source Code for frmReplSynchronizeSub.ebf

```
1  Option Explicit On
2
3  Private Sub Form_Activate()
4
5      'Set Subscriber Properties
6      txtSubscriber.Text = CEMerge.Subscriber
7      txtSubscriberConnectionString.Text = CEMerge.
       SubscriberConnectionString
8  End Sub
9
10 Private Sub cmdOK_Click()
11
12     Dim str As String
13
14     'Set Subscriber Properties
15     CEMerge.Subscriber = txtSubscriber.Text
16     CEMerge.SubscriberConnectionString =
       txtSubscriberConnectionString.Text
17
18     'Call the Initailize, Run Terminate methods to synchronize the
       subscription
19     On Error Resume Next
20     ShowWaitCursor()
21
22     CEMerge.Initialize()
23
24     If CEMerge.ErrorRecords.Count > 0 Then
25         HideWaitCursor()
26         ShowErrors(CEMerge.ErrorRecords, "Initialization Failed")
27     Else
28         On Error Resume Next
29         CEMerge.Run()
30         HideWaitCursor()
31         If CEMerge.ErrorRecords.Count > 0 Then
32             ShowErrors(CEMerge.ErrorRecords, "Synchronization Failed"
       )
33         Else
34             str = "Synchronization Complete" & vbCrLf
35             str = str & "Publisher Changes = " & CEMerge.
       PublisherChanges & vbCrLf
36             str = str & "Publisher Conflicts = " & CEMerge.
       PublisherConflicts & vbCrLf
37             str = str & "Subscriber Changes = " & CEMerge.
       SubscriberChanges & vbCrLf
38             MsgBox(str, vbOKOnly, " S Y N C H R O N I Z E ")
39         End If
40         CEMerge.Terminate()
41     End If
42
43     Me.Hide()
44     frmMainMenu.Show()
45 End Sub
46
47 Private Sub cmdCancel_Click()
48     Me.Hide()
49     frmMain.Show()
50 End Sub
51
52
```

Appendix I

User Manual

e-Summon System

User Manual

Contents

Introduction

e-Summon System is an application developed with the assistance of Kuala Lumpur City Hall (DBKL) to ease the work-burden of traffic summons management. This system consists of two main components; the PDA and desktop, each designed with specific users in mind.

The PDA will be used by enforcing officers while doing their rounds in checking for traffic offences. PDA is utilized to suit the high-mobility needs of the officers. The desktop component will be stationed at the main office. It will be further modularized into different parts to suit the users, the administrator, the data clerk and cashier.

This manual serves as guidance for users to maximize the usage of the overall system.

Introduction

e-Summon System is an application developed with the assistance of Kuala Lumpur City Hall (DBKL) to ease the work-burden of traffic summons management. This system consists of two main components; the PDA and desktop, each designed with specific users in mind.

The PDA will be used by enforcing officers while doing their rounds in checking for traffic offences. PDA is utilized to suit the high-mobility needs of the officers. The desktop component will be stationed at the main office. It will be further modularized into different parts to suit the users, the administrator, the data clerk and cashier.

This manual serves as guidance for users to maximize the usage of the overall system.

e-Summon System – PDA component

Login Page

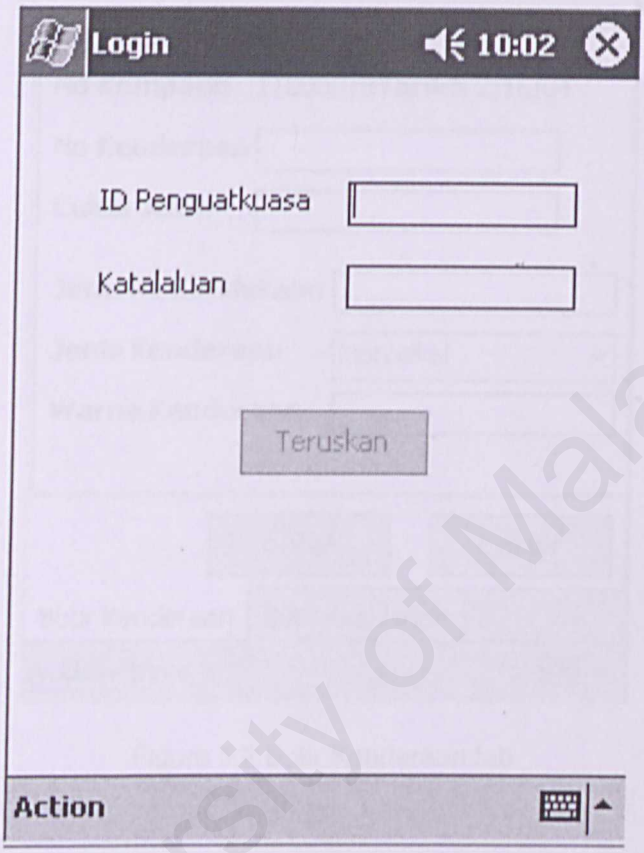

Figure 1.1 Login Page

1) When using the PDA application, user will be greeted with the screen prompting for enforcing officer ID and their password (Figure 1.1).
2) Fill in the enforcing officer ID in '**ID Penguatkuasa**'
3) Fill in the password in '**Katalaluan**'
4) Click '**Teruskan**'
5) If login fail, please contact the system administrator for the correct id name and password.

Main Page

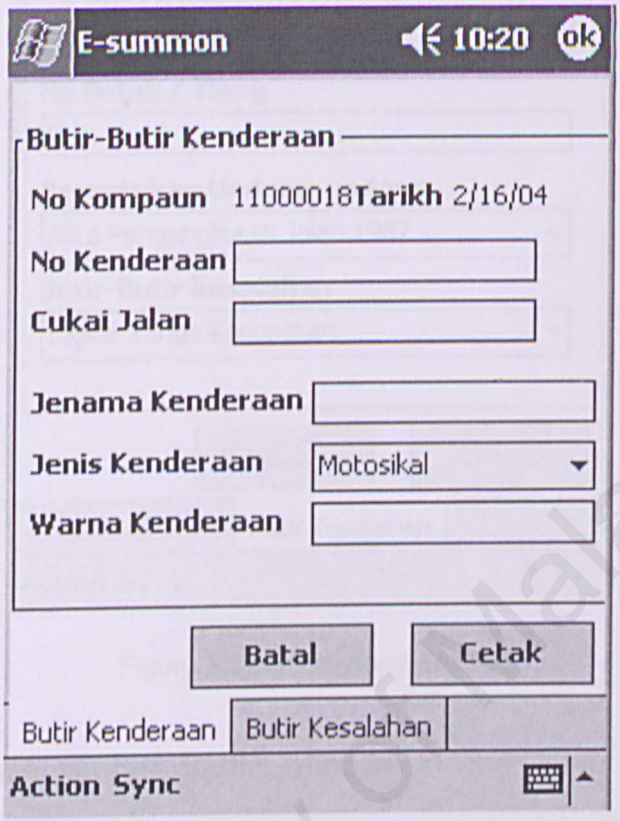

Figure 1.2 Butir Kenderaan tab

1) Fill in the vehicle registration number in '**No Kenderaan**' field (Figure 1.2)
2) Enter the road tax number in '**Cukai Jalan**'
3) Input in the brand of car in '**Jenama Kenderaan**' field
4) Choose the type of vehicle from '**Jenis Kenderaan**' combo box. Available choices are '**Motosikal**', '**Kereta**', '**Van/Lori/4WD**' and '**Lori Besar/Bas**'.
5) Input the color of the vehicle in the field of '**Warna Kenderaan**'

Note: The field of '**No Kompaun**' is automated to take the next number of compound notice. The field of '**Tarikh**' is also automated to take the date of the issuance of compound notice

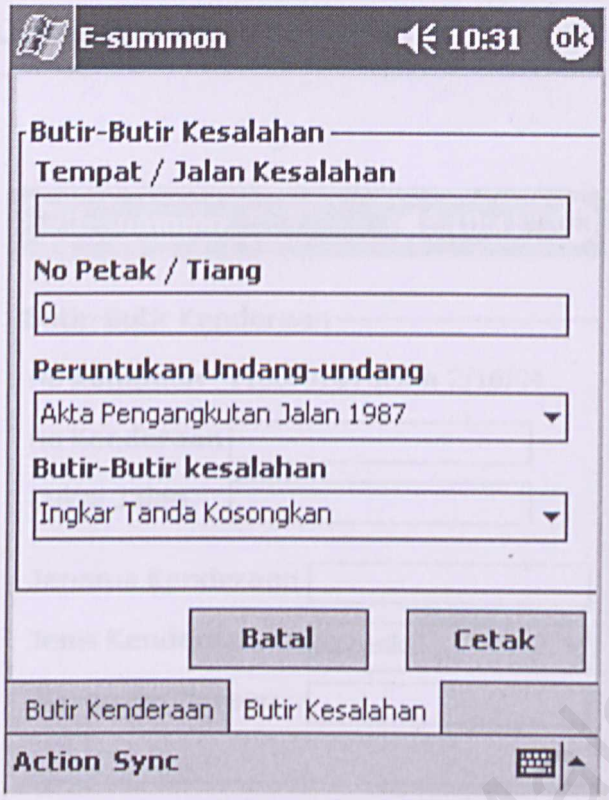

Figure 1.3 Butir Kesalahan tab

6) Fill in the place of the traffic offence in the filed '**Tempat / Jalan Kesalahan**' (Figure 1.3).
7) The field of '**No Petak/Tiang**' has been defaulted to the value of '**0**'. If applicable, fill in with the number of the parking space or payment machine.
8) Choose the law which the offence contravene from field '**Peruntukan Undang-undang**'
9) Choose the description of offence from '**Butir-butir Kesalahan**'
10) Click '**Cetak**' to save and print the compound notice.
11) To reset the fields, click '**Batal**'
12) To exit from e-Summon System, click menu **Action → Exit**

Note: Field of 'Butir-butir Kesalahan' is dependant on 'Peruntukan Undang-undang'. Any changes to the latter will be reflected in the former field.

Sync menu option

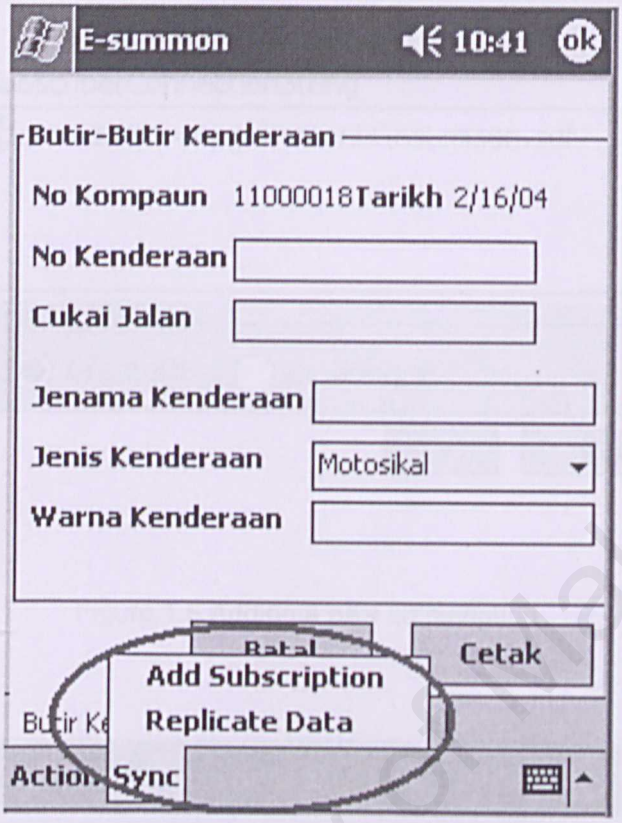

Figure 1.4 Sync menu option

1) To add subscription, at menu click at **Sync** → **Add Subscription** (Figure 1.4)
2) Edit the data source to point to new connection string (Figure 1.5).
3) Choose **CreateDB** to create new database or **ExistingDB** to add subscription to existing database.
4) Click **OK** to continue or **Cancel** to exit.

Note : Add Subscription is used to add new local file subscription into the system for the purpose of replication. The data source defaulted to the 'my

documents\esummon.sdf', the filename of the database needed in the system.

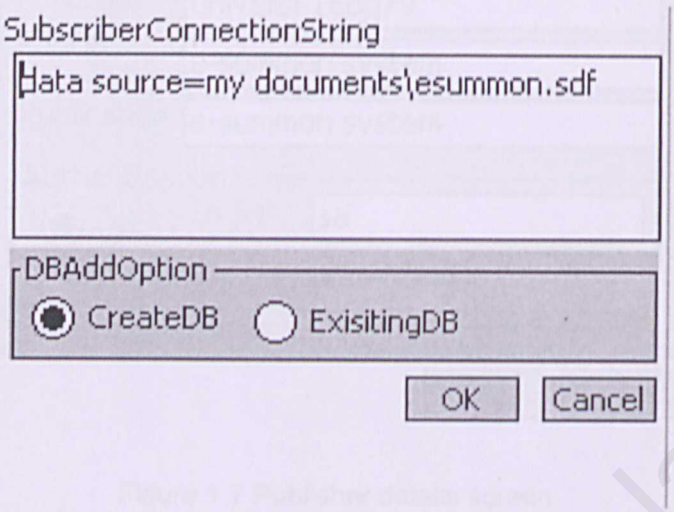

Figure 1.5 Adding a new subscription

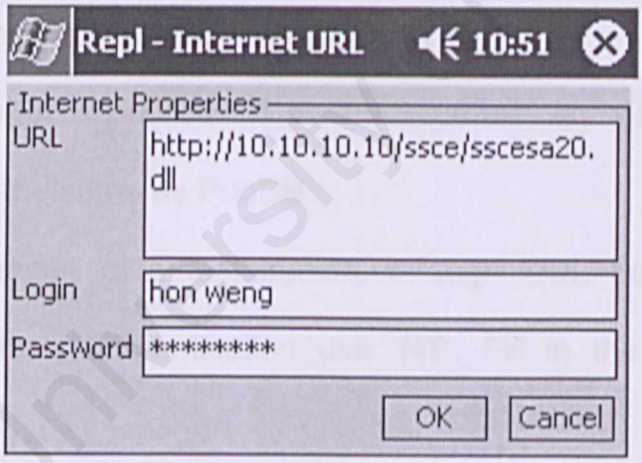

Figure 1.6 Repl-Internet URL screen

5) To synchronize file, click menu Sync → Replicate Data

6) Edit the URL (Figure 1.6)

7) Enter the login name and password

8) Click **OK**

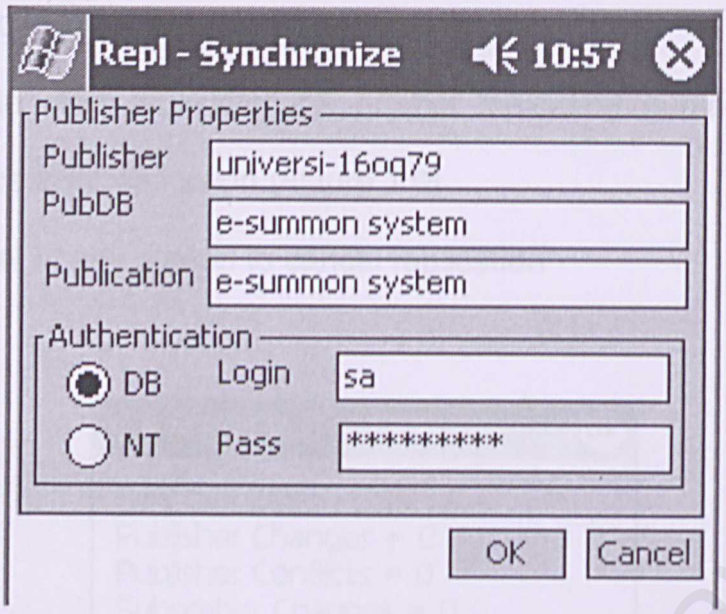

Figure 1.7 Publisher details screen

9) Enter the name of the SQL Server in field "**Publisher**' (Figure 1.7)

10) Enter the name of database in '**PubDB**'

11) Enter the name of publication in '**Publication**'. By default, the field of Publication is the same as PubDB

12) Choose the mode of authentication, through SQL Server use '**DB**' or through Windows authentication use '**NT**'. Fill in the login name and password

13) Click **OK**

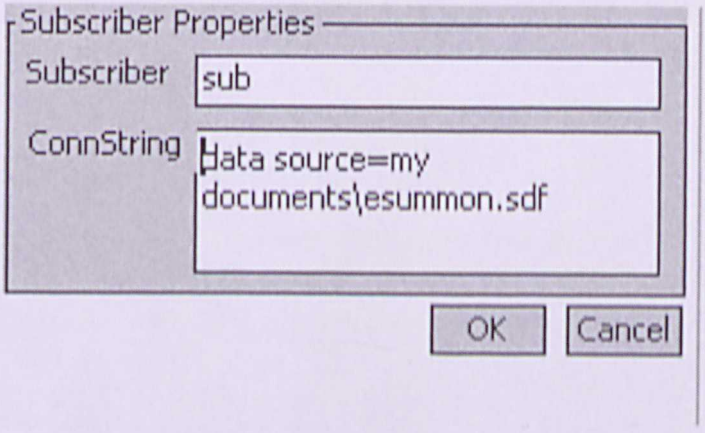

Figure 1.8 Subscriber details screen

14) Enter the name of subscriber in '**Sub**' (Figure 1.8)

15) Enter the connection string in '**ConnString**'

16) Click **OK** to start replication. A prompt indicating number of changes signify success in replication (Figure 1.9)

17) Click **Cancel** at any screen to cancel replication

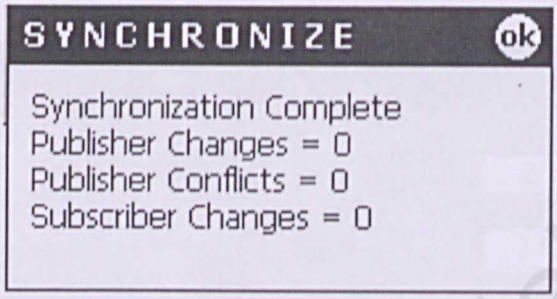

Figure 1.9 Prompt indicating numbers of changes between publisher and subscribers

e-Summon System – Desktop Component

Desktop component consist of three modules based on usage by three different roles; namely Administrator, Data Clerk and Cashier All three modules are accessed based on the access level permitted by their login id.

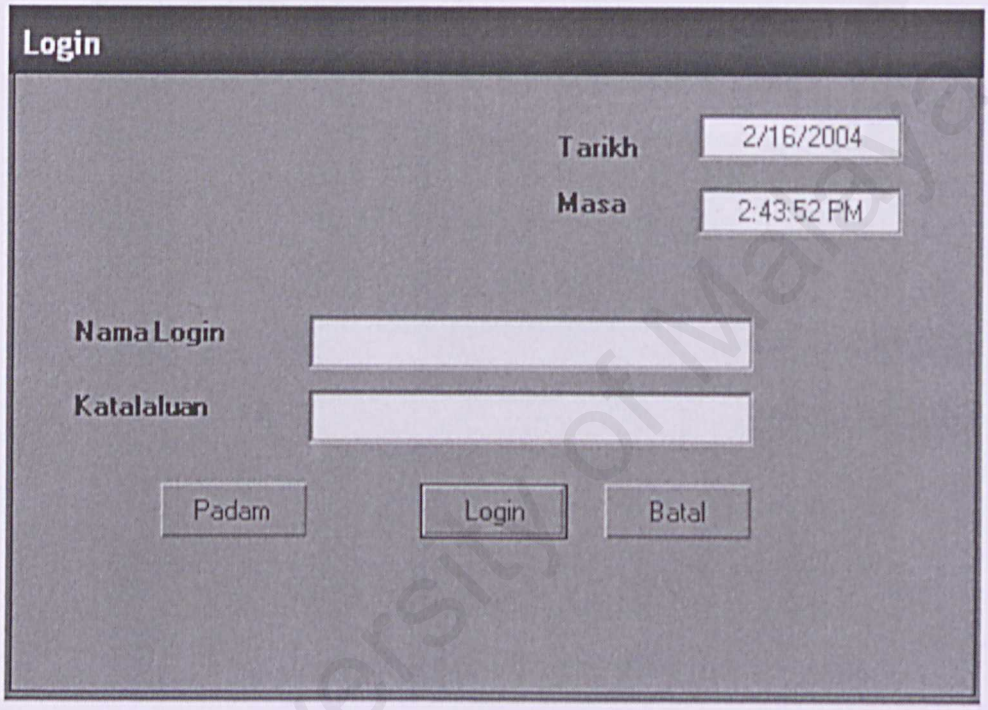

Figure 2.1 Login page

1) Enter the login name and password (Figure 2.1)
2) Click **Login** to continue or **Batal** to cancel
3) To reset the fields, click **Padam**

Figure 2.1 displays the login page for e-Summon System desktop component. The login will differentiate each users role and will thus display different settings

Administrator

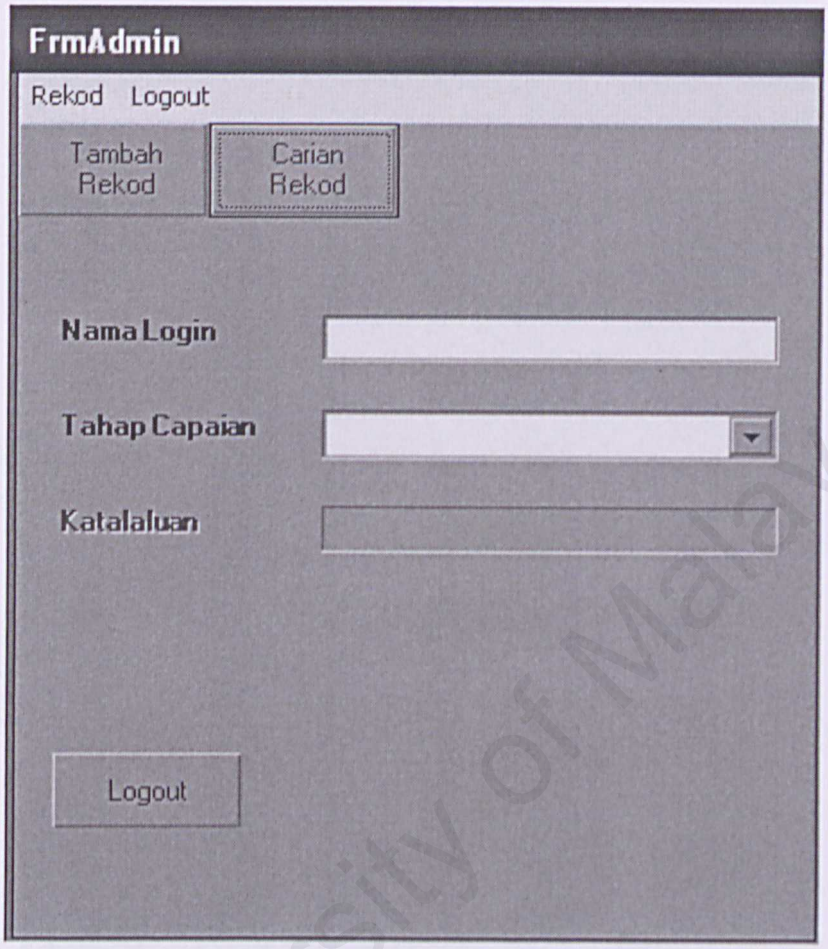

Figure 2.2 Administrator screen

1) Three options are available for administrators, 'Tambah Rekod', 'Carian Rekod' and Logout (Figure 2.2)

Adding a new record

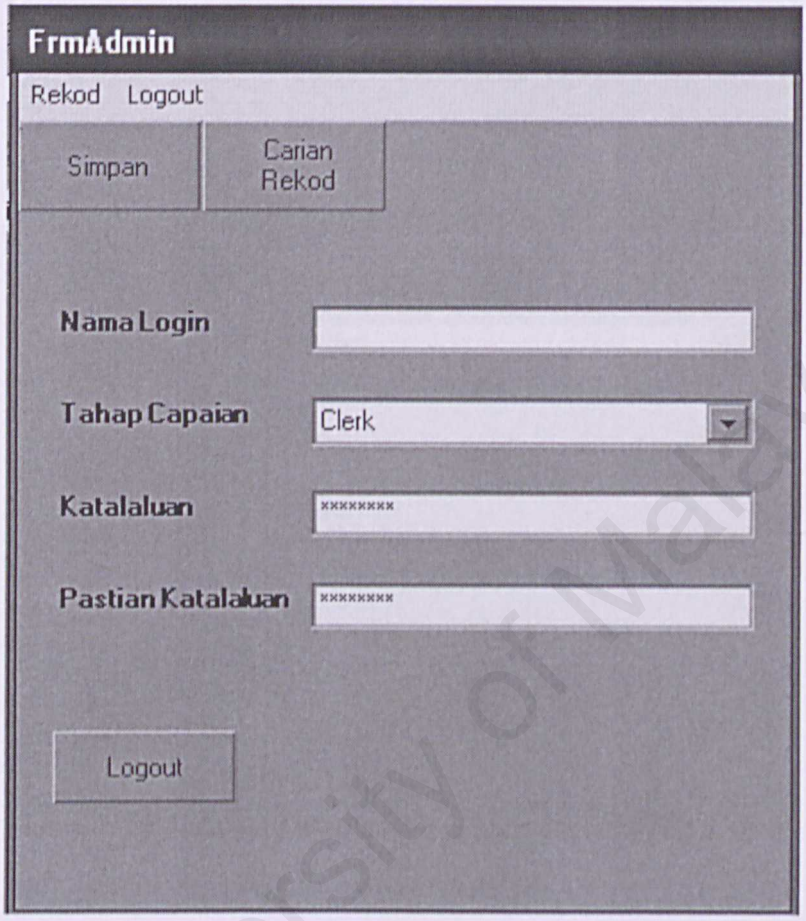

Figure 2.3 Adding a new record

1) Click on '**Tambah Rekod**'. Field 'Katalaluan' and 'Pastian Katalaluan' will be enabled. 'Tambah Rekod' button will be changed to 'Simpan' (Figure 2.3
2) Enter login name in field '**Nama Login**'
3) Choose the relevant access level in combo box '**Tahap Capaian**'. Available options are Administrator, Clerk and Cashier.
4) Enter the initial password to be used in '**Katalaluan**'. Reconfirm the password in '**Pastian Katalaluan**'.

Searching for Record

Figure 2.4 Search screen

1) To search for a specifi record, click on the button of '**Carian Rekod**' (Figure 2.4)
2) Fill in the '**Nama Login**' field with id to manage and click on the '**Carian Rekod**' again.
3) Click Logout anytime to end the application.

Data Clerk

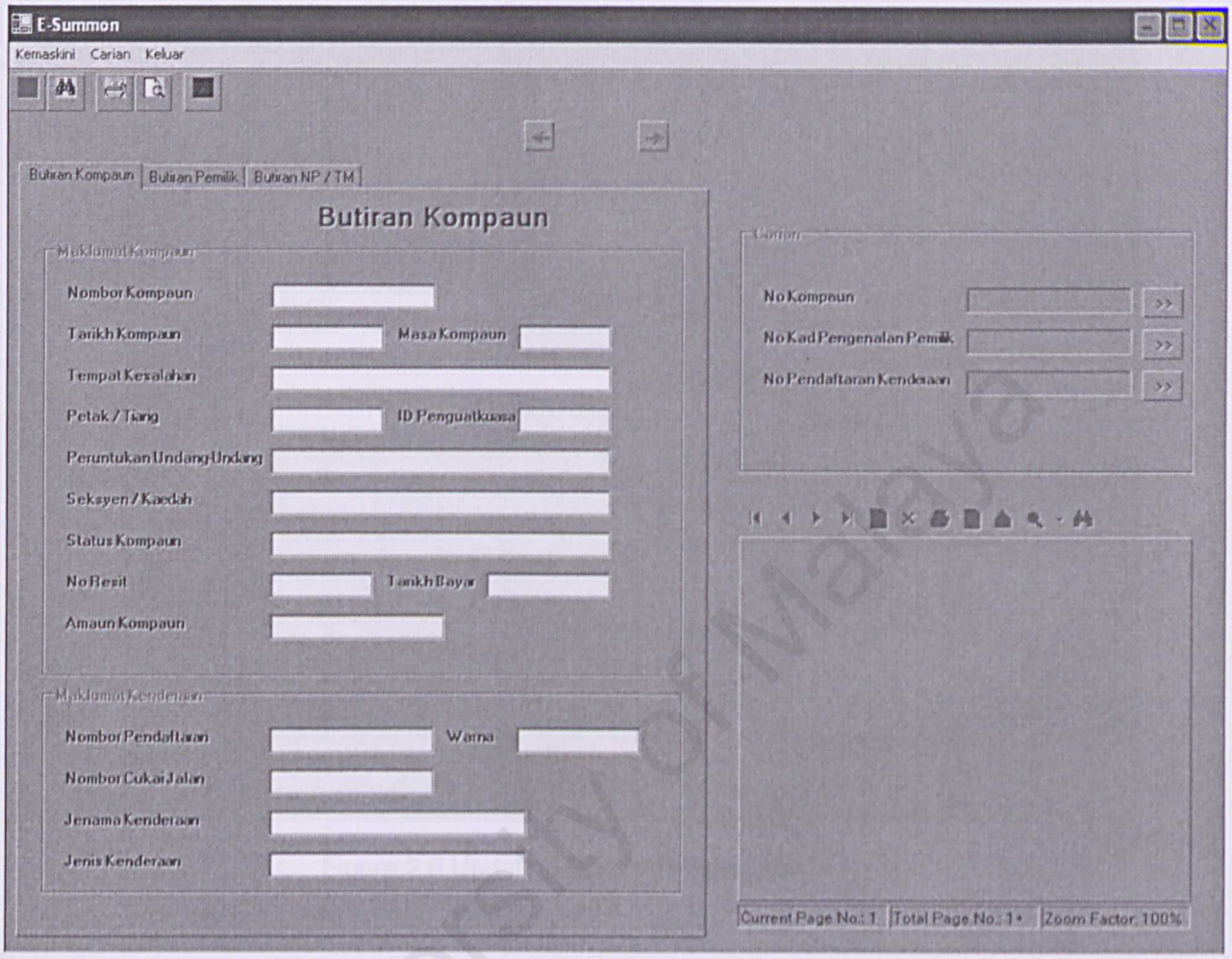

Figure 2.5 Clerk screen

Data clerk module consists of three main areas, the tab control, the search area and print preview. Tab control is further made up of the pages 'Butiran Kompaun', 'Butiran Pemilik' and 'Butiran Notis Peringatan / Tindakan Mahkamah' Menu offers three options, 'Kemaskini', 'Carian' and 'Keluar' Kemaskini is used to update compound notice database.

Updating Car Ownership

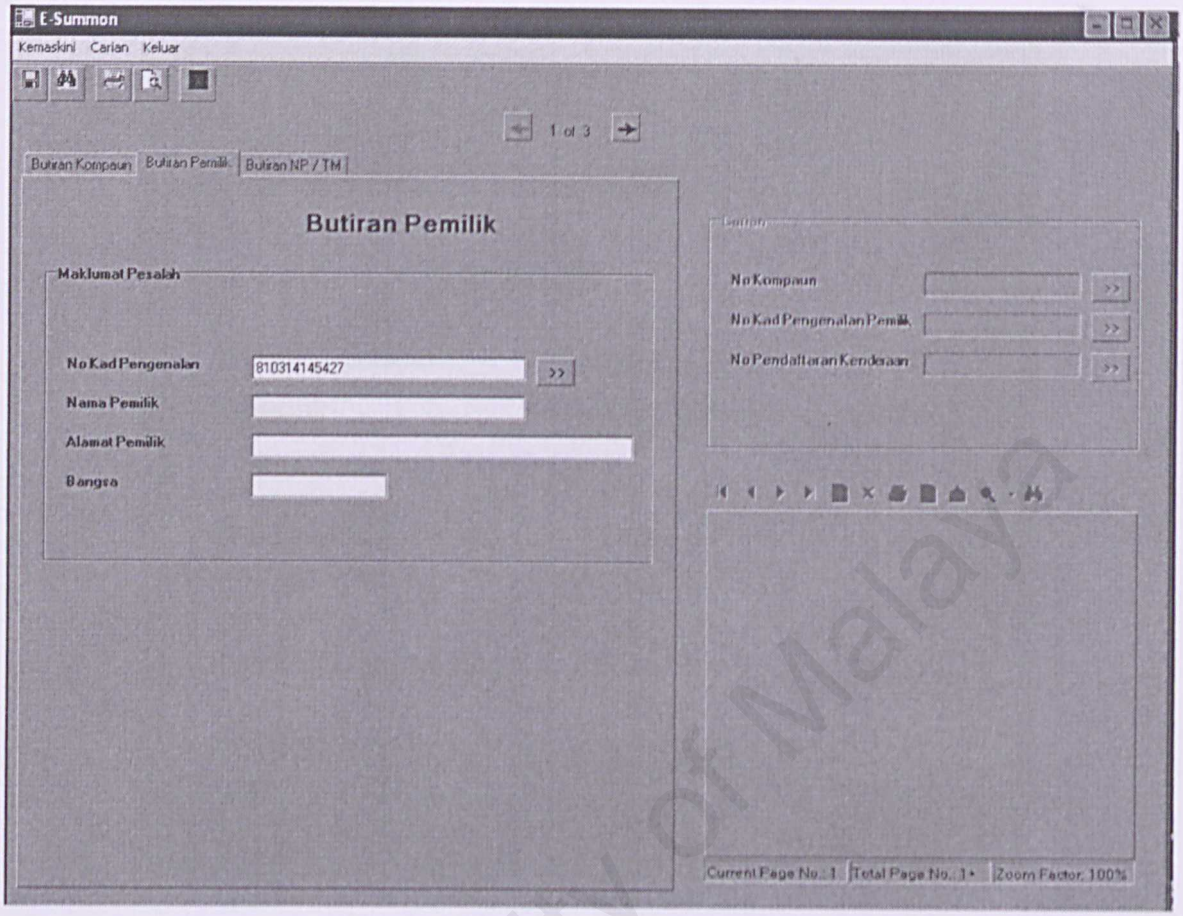

Figure 2.6 Updating Car Ownership details

1) Click on **Kemaskini→Maklumat Pemilik**, to update compound notices with no owner information. (Figure 2.6).
2) Input the identity card numner in the field '**No Kad Pengenalan**'. Click on button '>>' to check if previous record for the specific owner exist. If it exists, then the relevant fields will be filled.
3) After all the relevant fields has been filled, click **Kemaskini → Simpan** or click on the save icon.

Note: The navigational arrows on the top of the screen indicate the number of records to be updated.

Reminder Notices

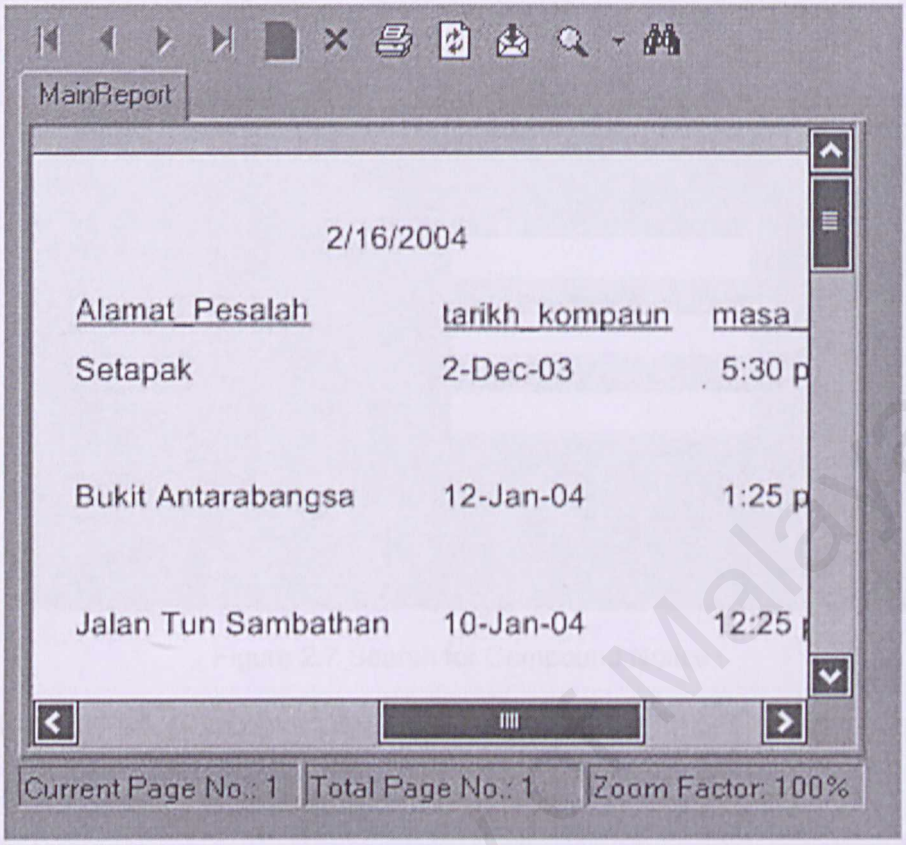

Figure 2.7 Preview of Reminder Notice

1) To update reminder notices to offenders, from menu choose **'Kemaskini'→'Notis Peringatan'→'Pertama'** for first notice. A list of all the vehicles which will be served with the reminder notice is displayed in the preview area. (Figure 2.7)
2) Click **Print** icon to print.
3) To print the second reminder notice, click **'Kemaskini'→'Notis Peringatan'→'Kedua'** from the menu

Searching for Compound Notices

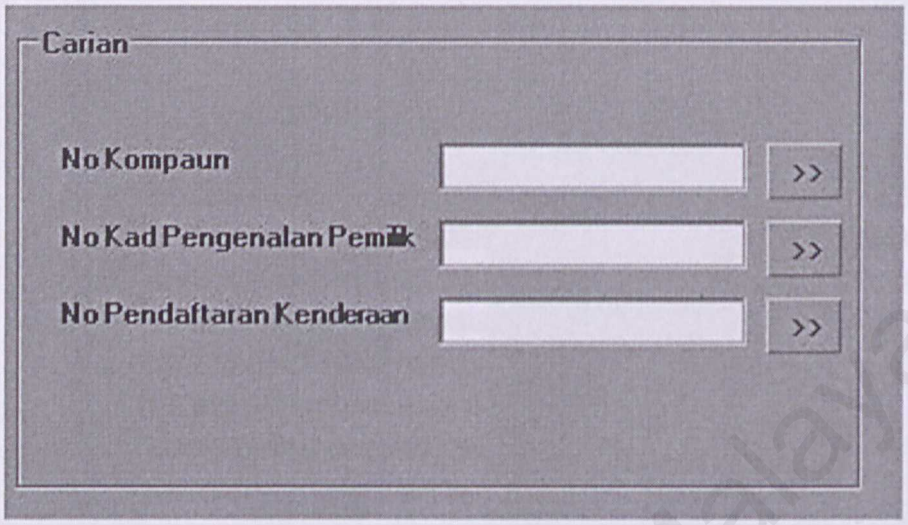

Figure 2.7 Search for Compound Notice

1) Search can be performed by either clicking on the '**Carian**' menu or the search icon. It will activate the 'Carian' area.
2) Search can be performed using either the compound number, identity card number of offender or car registration number.
3) The specific field must be filled in and click on the '>>' button adjacent to the field
4) The tab control area will be filled with relevant details if search is successful.

Cashier

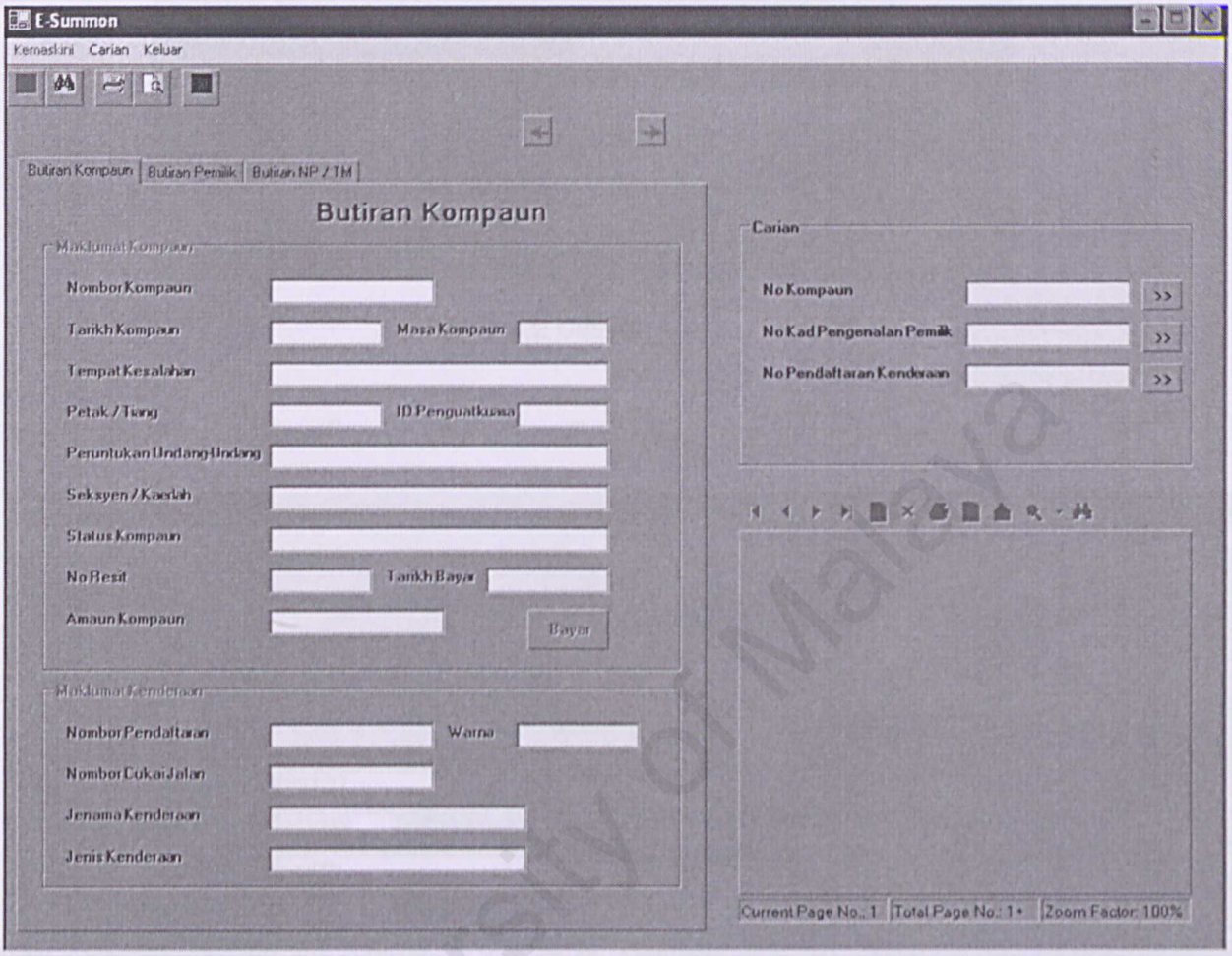

Figure 4.1 Cashier Module

The cashier module consists of a tab control interface, a search area and print preview (Figure 4.1).

Figure 4.2 Cashier search

1) Search for a specific compound notice using the search area (Figure 4.2)

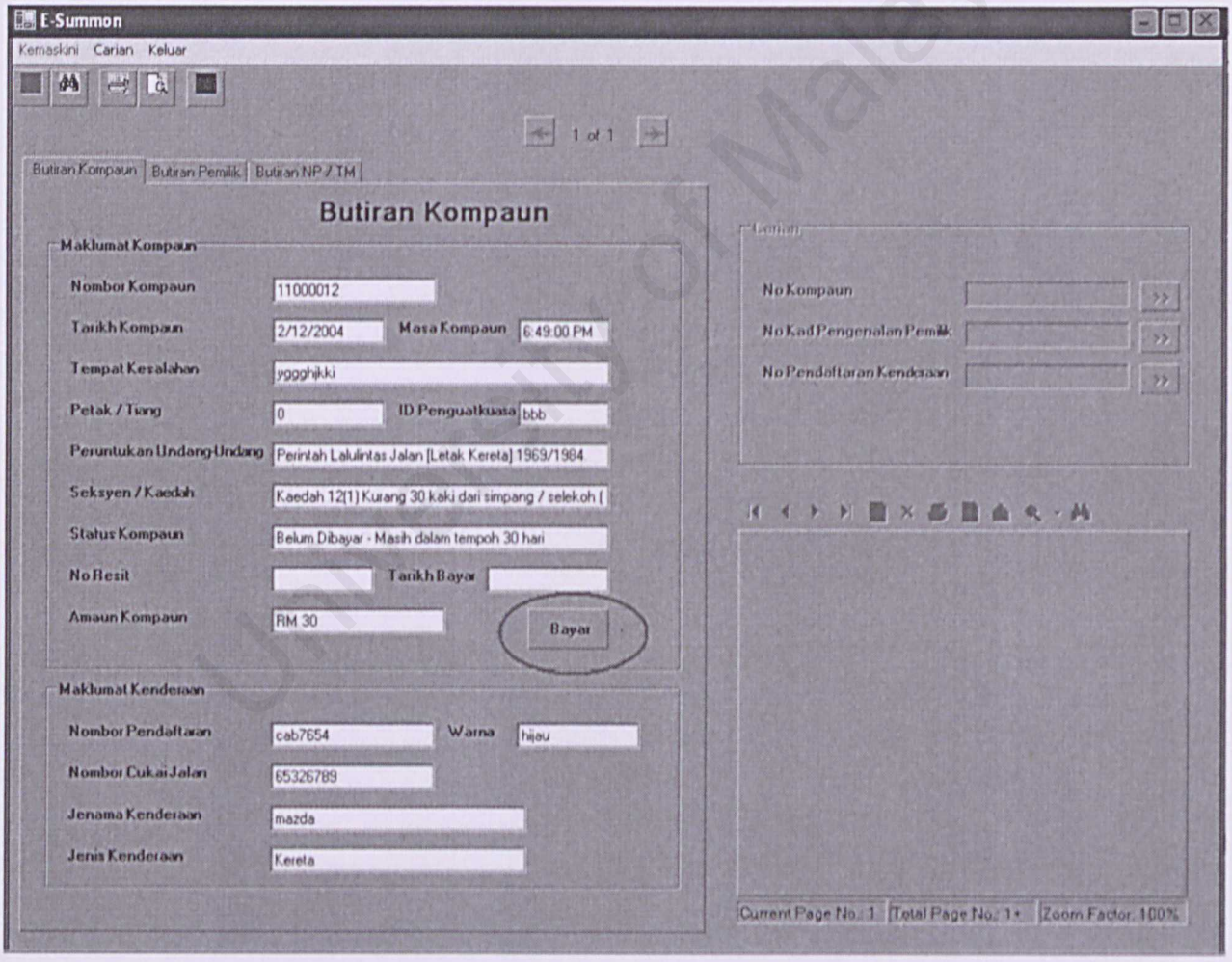

Figure 4.3 Paying a compound notice

2) Click on the 'Bayar' button to pay a compound notice (Figure 4.3)

3) A receipt will be printed.

Chapter Six:

Testing