Nama : Leong Keat Poh

No. Matrik : WET020062

Kod Kursus : WXES3182

Tajuk Projek : Embedded System

Nama Penyelia : En. Ang Tan Fong

Nama Moderator : En. Por Lip Yee

ABSTRACT

Efficiency is a very important issue for a doctor because in the medical field, every minute even a second can save a persons life. But one of the major problems which the doctors are facing is that they are not being able to attend to all their patients because some of them are just too busy or the patient need to wait for a long time in order to meet the doctor. In order to solve this situation, Doctor's Personal Management System (DPMS) has been developed.

DPMS is a pocket PC application that helps doctors to plan their schedule, appointments, keep personal data and contacts of their friends and also patients.

With DPMS, hopefully the efficiency and productivity of the doctors will be increased and in the process, it will also satisfy the patients because they are able to meet the doctors on time and to get their treatment.

ACKNOWLEDGEMENT

The completion of WXES3181 would not be a success if not for a few individuals who have sacrificed their time and efforts to encourage and guide me throughout the past 3 months.

Firstly, I would like to thank my supervisor, Mr. Ang Tan Fong, who has given me the idea for developing Doctor's Personal Management System (DPMS). I would like to thank him for his continuous guidance and patience. He has helped me a lot on the choices of technologies to develop this system. He has also helped me to solve some of the system design while I was working on my documentation. Truly without his guidance and help, I might not be able to complete this report on time.

I would also like to thank Mr. Por Lip Yee for being my moderator. I appreciated all the constructive criticism and valuable suggestions to improve my system. My appreciation also goes to the faculty for keeping a copy of the seniors' report. It has been a really useful reference for me as I got many ideas and information from the seniors' reports.

Finally, I would like to thank my friends especially Dr.Quek who has been a great help in helping me to understand my logics in programming and also in helping me to debug my system. I would also like to thank my family members for their encouragement and support. Without them, I don't think I am able to complete this report successfully. Thank you everyone!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
~Introduction~

Chapter 1 – Introduction

1.1 Overview

Efficiency is a very important issue for a doctor because in the medical field, every minute even second can save a persons life. But one of the major problems which doctors are facing is that they are not being able to attend to all their patients because some of them are just too busy or the patient need to wait for a long time in order to meet the doctor. And sometimes because of this, there is some complains from the patient about the inefficiency of the hospitals and it might even cause some illness among patients to worsen.

With the help of technology advances today, we are able to help the doctors to manage their time more efficiently and wisely. And for that, we will need Doctor's Personal Management System (DPMS). DPMS is a pocket PC application that helps doctors to plan their schedule, appointments, keep personal data and contacts of their friends and patients.

DPMS enables the doctors to plan their time more systematically and also acts as a reminder for their appointments.

With DPMS, the efficiency and productivity of the doctors will be increased and in the process, it will also satisfy the patients because they are able to meet the doctors on time and to get their treatment.

1.2 Project Motivation

As we all know that time is indeed a very important factor in medical field. Therefore, the doctors need to be very efficient with their work and not to spend so much time trying to get an appointment with their patient. DPMS will make be able to help doctors to plan their time well so that they won't be wasting so much time to make appointment when the time wasted can be used to save more lives.

1.2.1 Current situation on making appointment

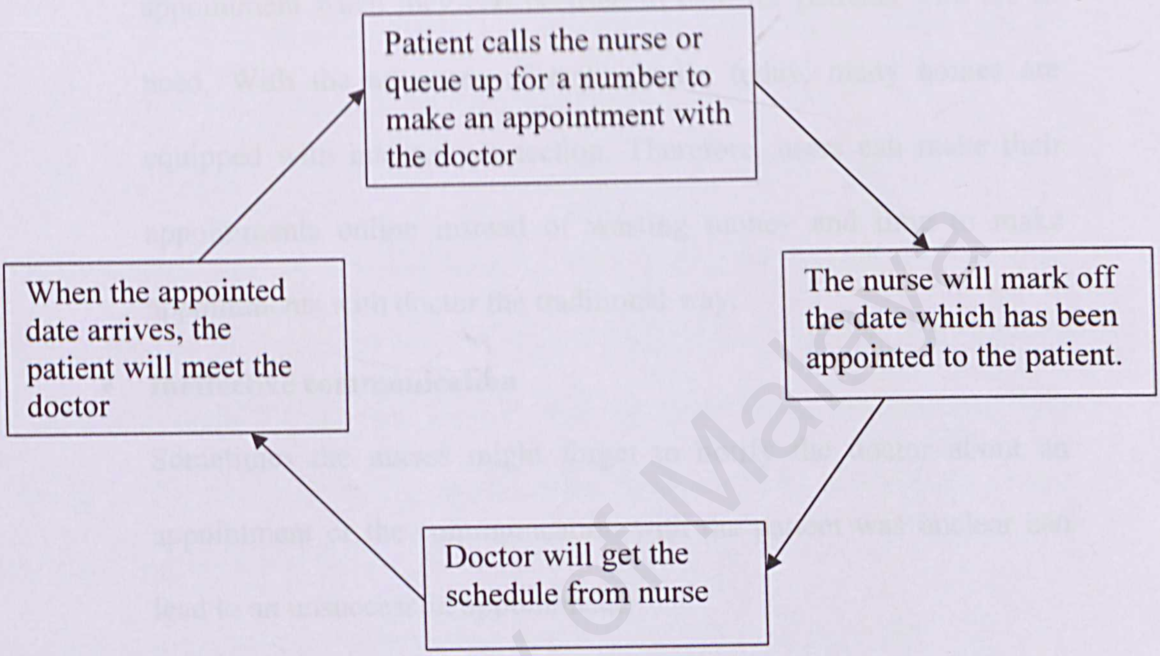

Patient calls the nurse or queue up for a number to make an appointment with the doctor
The nurse will mark off the date which has been appointed to the patient.
Doctor will get the schedule from nurse
When the appointed date arrives, the patient will meet the doctor

Figure 1-1: Traditional way of making an appointment with the doctor

1.2.2 Problems with current situation of making appointment

As we can see fro Figure 1-1, there are some weaknesses in the current method of making appointment with the doctor. Some of the weaknesses are:

- **Time consuming and inefficient**

  When making an appointment, a patient will have to either call or queue up to get a number. Sometimes, the patient will have to wait for hours before they can even get the number. This will waste the patient's time and also their traveling cost. Also sometimes, nurses might not be able to be there to attend to the phone call causing patient to have to keep calling in order to get an appointment. So the inefficiency can be quite frustrating for the patients at times.

- **Wastage of resources**

  Nurses are always needed to be at the phone to pick up calls for appointment when they can be used to care for patients who are in need. With the advances of technologies today, many homes are equipped with internet connection. Therefore, users can make their appointments online instead of wasting money and time to make appointments with doctor the traditional way.

- **Ineffective communication**

  Sometimes the nurses might forget to notify the doctor about an appointment or the communication with the patient was unclear can lead to an unsuccessful appointment.

1.2.3 Solutions for the current situation

An effective online system is needed to solve the current problems faced in making appointment with doctors. The system can allow patients to make appointment at anytime of the day and anywhere with internet connection. With the Pocket PC application also, doctors can download the latest updates for their schedule every morning before starting work so that they are able to get the latest schedule and appointments

Therefore with DPMS, doctors are able to get the latest appointments and manage their schedules on the go and patients are able to make appointments with the doctors more efficiently without any hassle.

1.3 Project Objectives

Core objectives of the project are as below:

- To enable doctors to know their appointments before hand through synchronization of their Pocket PC with the database.
- To provide patients with an alternative and more efficient way to make appointments with doctors through the website.

1.4 Project Scope

Generally, DPMS can be divided into 2 major parts, which are the Personal Management System for doctors and the Patient Appointment Section. The Personal Management System for doctors allows them to check their appointments and manage their schedules and Patient Appointment Section allows patient to check the doctor's availability and make appointments.

<u>The project scope of DPMS for **Personal Management System** includes:</u>

- Develop a Pocket PC application that includes:
  - Overview
  - Appointment
  - Contacts
  - Tasks
- Allows the doctor to download latest updates from the database

1.5 Expected Outcome

Expected outcome of DPMS for **Personal Management System** includes:

- Appointment – allows the doctors to key in their daily appointment with their patients.
- Overview – allows the doctors to view a bird's eye view of the appointments and tasks they have in order for them to make other plans.
- Contacts – doctors can insert the patient's contacts or personal contacts
- Tasks – enables the doctor to list down the tasks they have to complete.

1.6 Project Schedule

Project Stage	July				August				Sept				Oct				Nov				Dec				Jan			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Preliminary Study and Planning	■	■																										
Literature Study			■	■	■	■	■																					
System Analysis						■	■	■	■	■																		
System Design									■	■	■	■																
Prototype													■	■														
Development and Coding															■	■	■	■	■	■	■							
Unit Testing																				■	■	■	■	■				
System Testing																							■	■	■	■		
Documentation																										■	■	
Implementation and Maintenance																										■	■	

Figure 1-2: Project Schedule

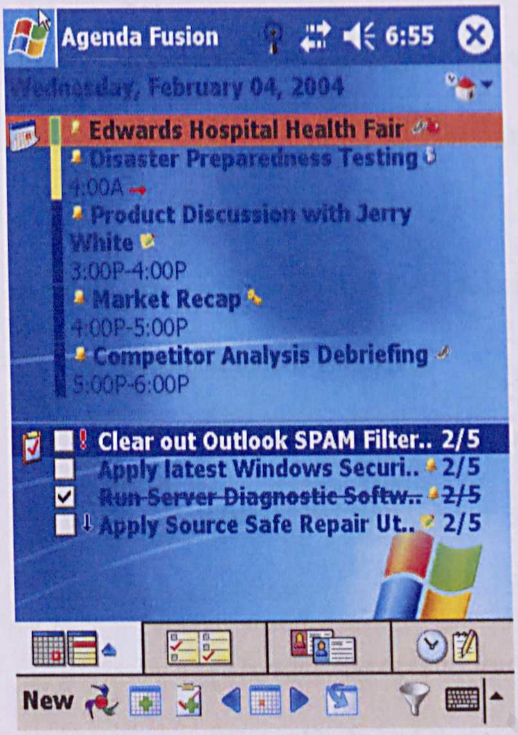
CHAPTER 2

~ Literature Review ~

Chapter 2 – Literature Review

2.1 Analysis Studies

2.1.1 Case Study 1 – Agenda Fusion 6.0

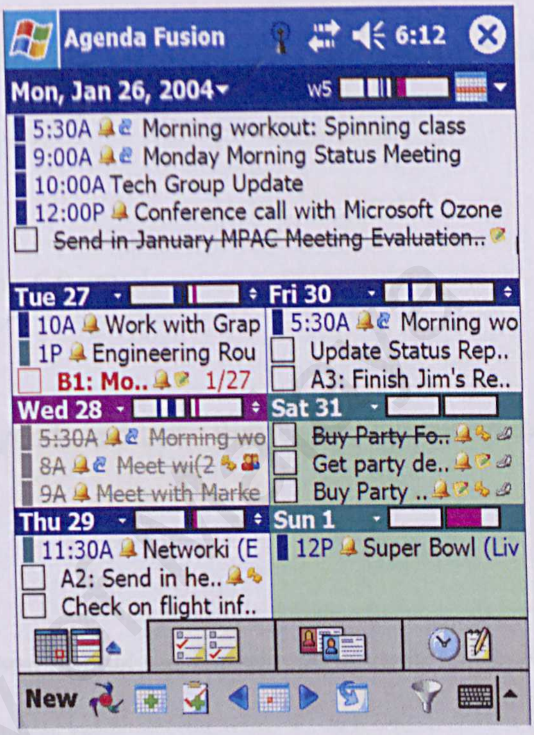


Wednesday, February 04, 2004

Edwards Hospital Health Fair
Disaster Preparedness Testing
4:00A
Product Discussion with Jerry White
3:00P-4:00P
Market Recap
4:00P-5:00P
Competitor Analysis Debriefing
5:00P-6:00P

□ Clear out Outlook SPAM Filter.. 2/5
□ Apply latest Windows Securi.. 2/5
☑ ~~Run Server Diagnostic Softw..~~ ~~2/5~~
□ Apply Source Safe Repair Ut.. 2/5

Figure 2-1: Today View



Mon, Jan 26, 2004

5:30A Morning workout: Spinning class
9:00A Monday Morning Status Meeting
10:00A Tech Group Update
12:00P Conference call with Microsoft Ozone
□ ~~Send in January MPAC Meeting Evaluation..~~

Tue 27
10A Work with Grap
1P Engineering Rou
□ B1: Mo.. 1/27

Fri 30
5:30A Morning wo
□ Update Status Rep..
□ A3: Finish Jim's Re..

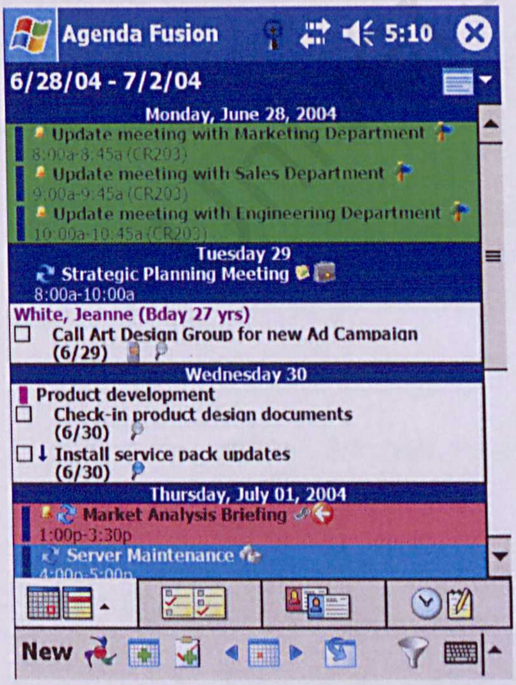
Wed 28
~~5:30A Morning wo~~
8A Meet wi(2
~~9A Meet with Marke~~

Sat 31
□ ~~Buy Party Fo..~~
□ Get party de..
□ Buy Party ..

Thu 29
11:30A Networki (E
□ A2: Send in he..
□ Check on flight inf..

Sun 1
12P Super Bowl (Liv

Figure 2-2: Weekly View



6/28/04 - 7/2/04

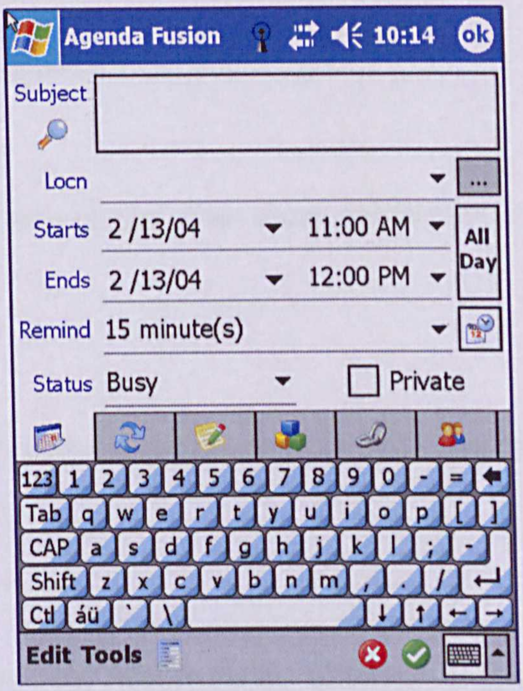
Monday, June 28, 2004
Update meeting with Marketing Department
8:00a-8:45a (CR203)
Update meeting with Sales Department
9:00a-9:45a (CR203)
Update meeting with Engineering Department
10:00a-10:45a (CR203)

Tuesday 29
Strategic Planning Meeting
8:00a-10:00a
White, Jeanne (Bday 27 yrs)
□ Call Art Design Group for new Ad Campaign (6/29)

Wednesday 30
Product development
□ Check-in product design documents (6/30)
□ Install service pack updates (6/30)

Thursday, July 01, 2004
Market Analysis Briefing
1:00p-3:30p
Server Maintenance
4:00p-5:00p

Figure 2-3: Agenda View


Agenda Fusion
10:14
ok
...
All Day
123 1 2 3 4 5 6 7 8 9 0 - =
Tab q w e r t y u i o p [ ]
CAP a s d f g h j k l ; '
Shift z x c v b n m , . /
Ctl áü ` \
Edit Tools

Subject
Locn
Starts 2 /13/04 11:00 AM
Ends 2 /13/04 12:00 PM
Remind 15 minute(s)
Status Busy □ Private

Figure 2-4: Appointment Editor

Agenda Fusion 6.0 delivers users with far beyond built-in Contacts, Calendar and Task applications included on the Pocket PC. It provides the user with a new feature which is the notes management view that allows the users to manage and organize users' Agenda Fusion alarm notes as well as other notes on the Pocket PC. Agenda Fusion delivers the most comprehensive set of time management features and yet it is easier to use than any other solution available.

Some of the main features of Agenda Fusion 6.0 are:-

a) Today View

- It allows the user to view agenda and tasks that are to be done on that particular day.

b) Weekly View

- Divides the screen into 7 components and shows the agenda and tasks for each day of that particular week.

c) Agenda View

- Allows the user to view the agenda for the whole week.

d) Appointment Editor

- Allows the user to enter the information of an event and also to set a reminder for the event.

e) Alarm Notes

- Allows the user to enter important events or tasks which needed reminders.

f) Categories Manager

- The user can customize their own colours for the different categories such as red for business, blue for holidays and etc.

h) Contacts View

- Shows the contacts to the user. It includes name of the person, a picture, address, position and phone no.
- It also shows the other contacts alphabetically.

i) Hourly View

- Hourly views are shown in either daily or weekly. Events or tasks are shown according to the hours so that users can clearly see their empty time slots allowing them plan more efficiently.

j) Tasks View

- Users can insert their tasks according to different categories and reminders can be set along with each task.

2.1.1.1 Result of study

Strengths:

- Allows the user to view events and task daily, weekly, monthly and by year making it easier for users to plan their schedules.
- Users are able to customize the screens.
- Comprehensive set of features and easy of use
- Compatible with most existing data on the pocket PC
- Available in 6 languages
- Users can get latest updates or patches from www.developerone.com

Weaknesses:

- The colour coordination is too complex. Each screen should only be limited to 3 colours or else it will strain the users' eyes.
- There is too many split screens for weekly views. It will make it hard for

users to see their appointments clearly because they are all cramped into a small screen.

- There isn't a legend to show users the colour representations.

2.1.2 Case Study 2 – Pocket Informant 5.0

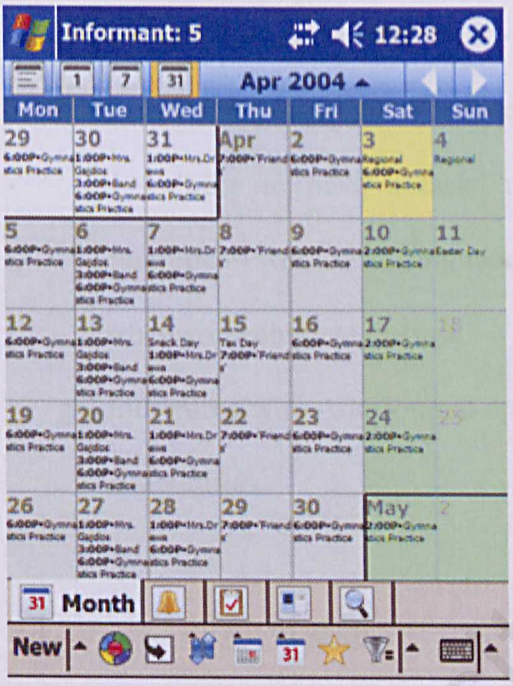
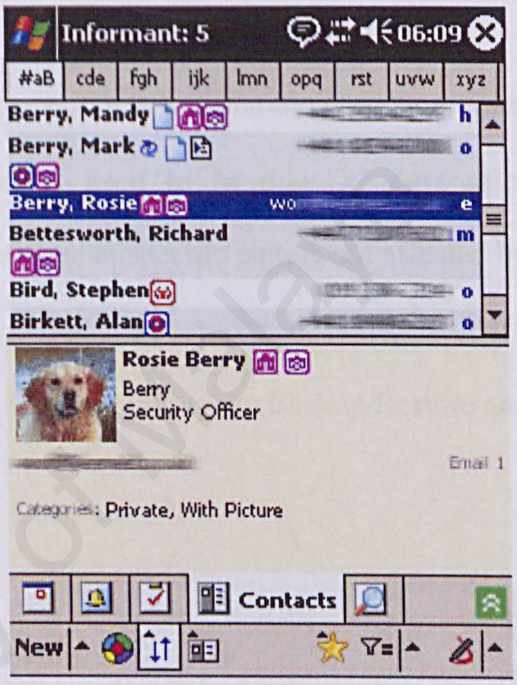


Figure 2-5: Monthly View



Berry, Mandy
Berry, Mark
Berry, Rosie
Bettesworth, Richard
Bird, Stephen
Birkett, Alan

Rosie Berry
Berry
Security Officer
Email 1
Categories: Private, With Picture

Figure 2-6: Contacts View

Pocket Informant 5.0 is designed to be very easy to use without being in the way. It can manage simple appointment and tasks and grow with users needs to handle even the most complex situations. Pocket Informant is designed with a turnkey solution – with no advance setup or steep learning curve.

Some of the main features of Pocket Informant 5.0 are:

a) Task grouping by category, importance, priority, progress, completion and date.

b) User sharable colour scheme along with a built in colour category and icon manager.

c) Powerfully configurable - it offers the most powerful and intuitive way of customizing not only the look but the behavior to match users needs.

d) Contact/Company Pictures, Company Templates, and Soundex Searching for Contacts in the Search View

e) Lightning fast search view that for your calendar, tasks, contacts and notes with wildcard searching. Multi-select on the results of the search.

f) Task Regeneration and Task Recurrence options. Regenerating tasks provides the ability to have a task "regenerate" itself to be due "x" days after completion while recurring tasks regenerate at specific patters not affected by when you complete the task.

g) Journaling with AutoJournal options, Linking with Links Backup/Restore and new Editors.

2.1.2.1 Result of study

Strengths:

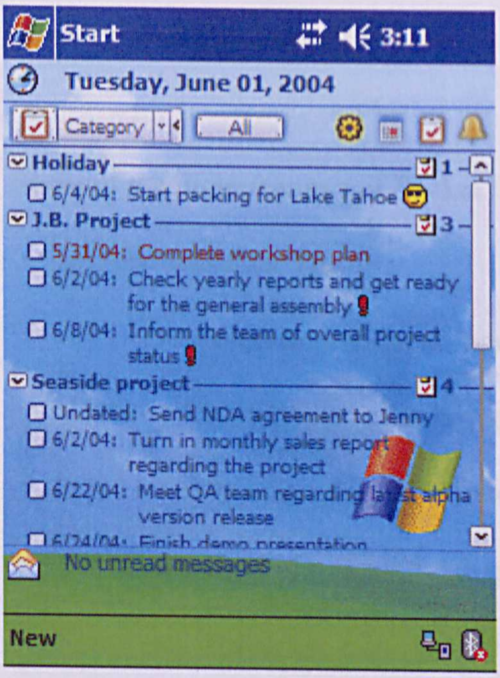
- Good colour coordination
- Customizable views
- 2/6/12 month picker complete with category colouring of users data straight from the data and category filtering
- Lightning fast search views
- Very simple to use and understand

Weaknesses:

- Limited repeat appointment function
- Slow to refresh

- Sometimes display the correct information in the Agenda but not in the weekly view

2.1.3 Case Study 3 – Pocket Breeze



Start 3:11

Tuesday, June 01, 2004

Category All

Holiday 1
□ 6/4/04: Start packing for Lake Tahoe

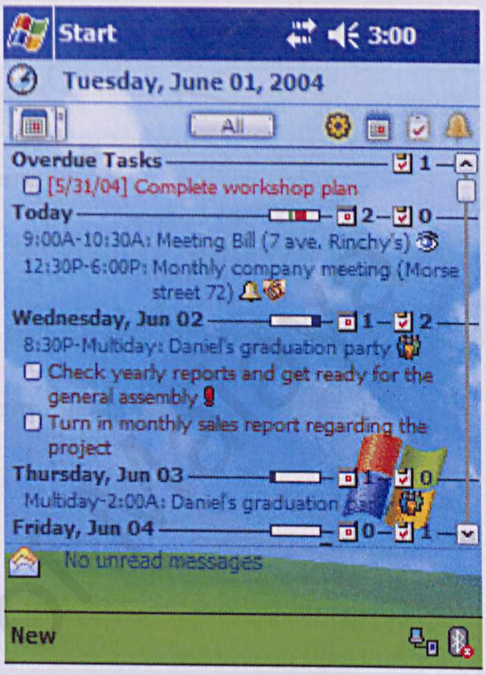
J.B. Project 3
□ 5/31/04: Complete workshop plan
□ 6/2/04: Check yearly reports and get ready for the general assembly
□ 6/8/04: Inform the team of overall project status

Seaside project 4
□ Undated: Send NDA agreement to Jenny
□ 6/2/04: Turn in monthly sales report regarding the project
□ 6/22/04: Meet QA team regarding latest alpha version release
□ 6/24/04: Finish demo presentation

No unread messages

New

Figure 2-7: Task by Category View



Start 3:00

Tuesday, June 01, 2004

All

Overdue Tasks 1
□ [5/31/04] Complete workshop plan

Today 2 0
9:00A-10:30A: Meeting Bill (7 ave. Rinchy's)
12:30P-6:00P: Monthly company meeting (Morse street 72)

Wednesday, Jun 02 1 2
8:30P-Multiday: Daniel's graduation party
□ Check yearly reports and get ready for the general assembly
□ Turn in monthly sales report regarding the project

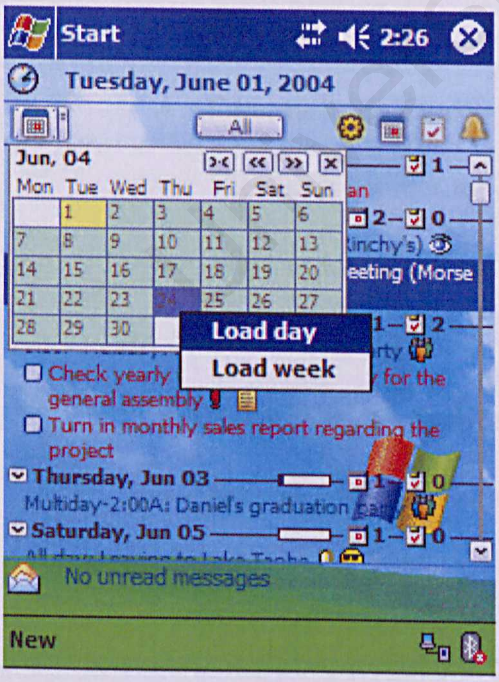
Thursday, Jun 03 1 0
Multiday-2:00A: Daniel's graduation par

Friday, Jun 04 0 1

No unread messages

New

Figure 2-8: Today View



Start 2:26

Tuesday, June 01, 2004

All

Jun, 04

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Load day
Load week

□ Check yearly reports and get ready for the general assembly
□ Turn in monthly sales report regarding the project

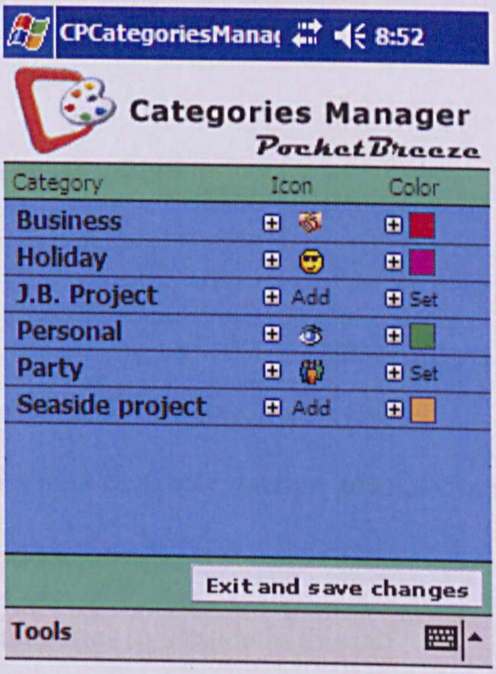
Thursday, Jun 03 1 0
Multiday-2:00A: Daniel's graduation par

Saturday, Jun 05 1 0

No unread messages

New

Figure 2-9: Calendar Tab



CPCategoriesMana 8:52

Categories Manager
PocketBreeze

Category	Icon	Color
Business	⊞	⊞ 🟥
Holiday	⊞	⊞ 🟪
J.B. Project	⊞ Add	⊞ Set
Personal	⊞	⊞ 🟩
Party	⊞	⊞ Set
Seaside project	⊞ Add	⊞ 🟧

Exit and save changes

Tools

Figure 2-10: Categories Manager

PocketBreeze allows the user to manage appointments from the Today screen. It views up to 30 days of appointments and tasks on the Today screen. It can integrate seamlessly with Agenda Fusion, Pocket Informant and the default Personal Information Management (PIM). Users can view tasks and appointment notes with scrolling pop-up windows.

Some of the main features of Pocket Breeze are:

a) The Calendar Tab

- View appointments and tasks in a single, tabbed display
- Include up to 30 days of upcoming agenda
- Pop-up calendar window
- Filter the Calendar display by category
- Complete tasks from the Today screen

b) The Tasks Tab

- Sort tasks
- Filter tasks by Category
- View task counter for each category grouping
- Mark tasks completed from Today screen

c) The Special Events Tab

- View contact's birthdays and anniversaries
- Special Events tab icon flashes as a reminder when a special event is near - users define how many days
- Define up to two additional categories to include in this tab

2.1.3.1 Result of study

Strengths:

- View appointments and tasks in a single, tabbed display
- Pop-up calendar window
- Special Events tab icon flashes as a reminder when a special event is near
- A year's worth of your contacts birthdays and anniversaries with flashing reminders

Weaknesses:

- Not very compatible with other Personal Information Management (PIM)
- Never show notes attached to the task or appointments
- Doesn't allow users to customize the screens

2.2 Software Architecture

There are a few software architectures available now: mainframe architecture, client-server architecture, two-tier architecture and three-tier architecture.

2.2.1 Mainframe Architecture

In mainframe system architecture, all operation is within the central host computer. User interacts with the host through a terminal that captures keystroke and sends that info to the host. Mainframe architecture is not tied to a hardware platform. User interaction can be cloned using PCs and UNIX workstations. A limitation of mainframe architecture is that it does not easily support graphical user interface or access to multiple databases from graphically dispersed sites.

2.2.2 Client-server Architecture

Client

Client is a networked information requester, usually a PC or workstation that can query database and/or other information from a server. Clients rely on servers for resources, such as files, devices, and even processing power.

Server

Server is a computer, usually a high-powered workstation, a minicomputer, or a mainframe, that houses information for manipulation by networked clients. Server is dedicated to managing disk drives (file servers), database (database servers), printers (print servers), or network traffic (network servers).

Client-server

Client-server is a network architecture in which each computer or process on the network is either a client or a server. Client-server architecture implies a cooperative processing of requests submitted by a client, or requester, to the server, which processes the requests and returns the results to the clients. The client manipulates the data and presents the result to the user.

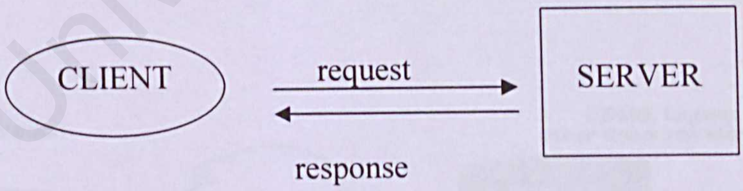


Figure 2-11: One-to-One Client Server

Client-server solutions can be in a many-to-one design that is more than one client typically makes requests of the server.

2.2.3 Two-Tier Architecture

2-tier architecture refers to client/server architectures in which the user interface runs on the client and the database is stored on the server. The actual application logic can run on either the client or the server. There are only the architecturally tiered data server and client.

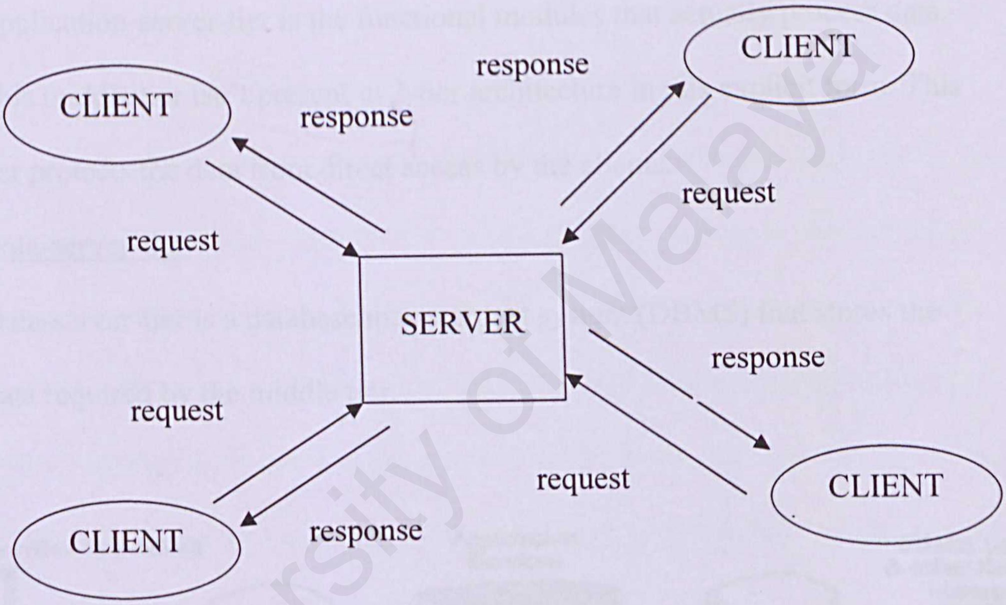


Figure 2-12: Many-to-One Client Server

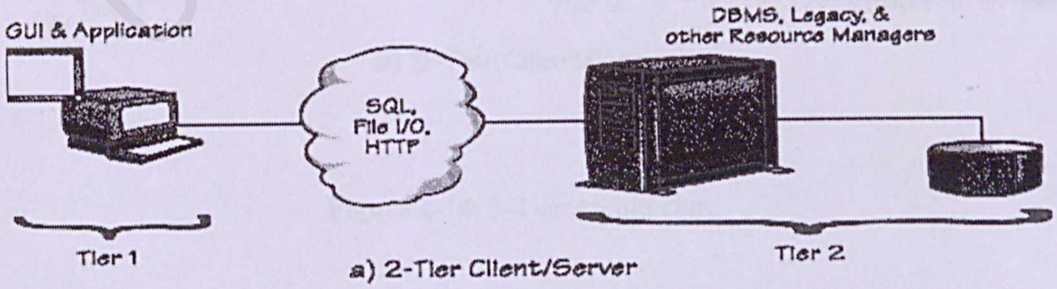


Figure 2-13: 2-Tier Architecture

2.2.4 Three-Tier Architecture

Three-tier architecture is a special type of client/server architecture consisting of three well-defined and separate processes, each running on a different platform:

The three tiers consist of:

1. Client-tier

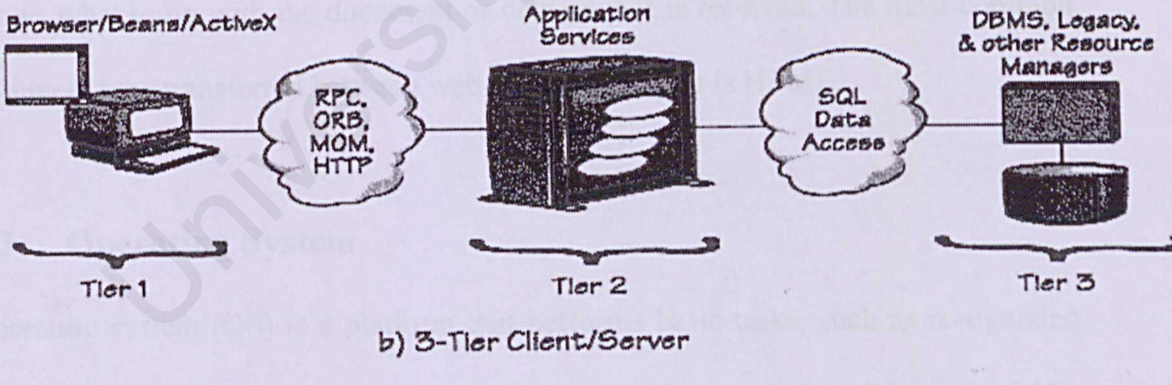
   Client-tier is the user interface, which runs on the user's computer.

2. Application-server-tier

   Application-server-tier is the functional modules that actually process data. This middle tier isn't present in 2-tier architecture in this explicit form. This tier protects the data from direct access by the clients.

3. Data-server-tier

   Data-server-tier is a database management system (DBMS) that stores the data required by the middle tier.



b) 3-Tier Client/Server

Figure 2-14: 3-Tier Architecture

2.2.5 Conclusion for Software Architecture

The three-tier architecture is chosen for this project because it is easier to implement and design. The three-tier design has many advantages over traditional two-tier or single-tier designs, the chief ones being:

- ❑ The added modularity makes it easier to modify or replace one tier without affecting the other tiers.
- ❑ Separating the application functions from the database functions makes it easier to implement load balancing.

In this project, the three tiers consists web browser as client-tier, web server as application-server-tier and database server as the data-server-tier. A web server can be defined as a computer program that receives HTTP requests from web browser for document. Web server will achieve and process the data from database server. Web server return both the document and the document type to the client so that the client knows what to do with the document or data once it is received. The most common document type transferred between web server and client is HTML.

2.3 Operating System

Operating system (OS) is a platform that performs basic tasks, such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk, and controlling peripheral devices such as disk drives and printers.

Besides that, the OS makes sure that different programs and users running at the same time do not interfere with each other. For security, OS ensures that unauthorized users do not access the system. OS provides a software platform to allow application programs run on it.

The most popular operating systems currently are UNIX, Linux, Windows 2000 and Windows XP.

2.3.1 UNIX

UNIX is a much older operating system that was created in the late 1960s. UNIX is designed to provide a multi-user, multitasking system for use by programmers. It began as an open source project that became widely used in Universities, scientific labs, and by the U.S. government. The philosophy behind the design of UNIX was to provide simple and powerful utilities that could be pieced together in a flexible manner to perform a wide variety of tasks. Over the years, hundreds of talented programmers contributed their own improvements to Unix making it extremely robust, stable, and fast.

However, UNIX is more difficult to learn and isn't as widely supported as Microsoft Windows 2000.

2.3.2 Linux

Linux has gradually become a popular operating system for Internet/ intranet serving purposes. With a host of performance enhancements that will benefit Web sites and Internet sites of all sizes, Linux is a stable and high-performance operating system for Internet usage.

Linux has made progress, primarily in functionality important to Internet infrastructure and Web server capabilities, including a greater selection of drivers, easier installation, GUI-based front ends for Web administration and window management.

2.3.3 Microsoft® Windows® 2000

Windows® 2000 is Microsoft's latest version of popular Windows® NT Operating System. Windows® 2000 Server has big improvement over Windows® NT 4.0. The changes, both fundamental and cosmetic, have made Windows® 2000 faster, more reliable, heavier-duty, and easier to use.

2.3.4 Microsoft® Windows® XP Professional

By building on the proven Windows® 2000 operating system, Windows® XP Professional delivers a reliable foundation you can count on to keep your computer up and running when you need it most. Not only is Windows® XP more reliable, it also helps you more easily recovered from system problems. Computers running Windows® XP Professional outperform Windows® 98 Second Edition, and meet the performance of Windows® 2000 on commercial benchmarks. Not only will Windows® XP launch applications faster, in most cases, your entire system will start up much more quickly.

Security features in Windows® XP Professional protect sensitive and confidential data both on your computer and while you are transmitting over a network or the Internet. With support for the latest security standards and enhanced virus protection, Windows® XP also protects you from some of the more common types of Internet attacks.

From its fresh, new look to its intuitive, task-based design, Windows® XP Professional makes it easier than ever to use the computer. Users will be able to get more done in less time, find items in a snap, and arrange files and folders in a way that makes sense to the users. Simply put, Windows® XP Professional helps users to work smarter.

2.3.5 Conclusion on Operating System

The operating system has been chosen for this project is Microsoft® Windows® XP Professional because of the reliability and stability that it provides. Windows® XP Professional is also compatible with almost all of the software out in the market. Besides, it is also more user friendly and almost everyone owns a Windows® XP operating system.

2.4 Database Server

A database is a structured collection of data. To add, access, and process data stored in a computer database, a database server is needed. There are several database servers available currently: Oracle, MySQL and Microsoft SQL Server.

2.4.1 Oracle

Oracle is a multi-user database. It provides unprecedented ease-of-user and is pre-tuned and pre-configured for today's dynamic workgroup and line-of-bus environment.

Oracle includes a fully integrated set of easy-to-use management tools, full distribution, replication and web features. Oracle also provides the highest levels of availability through fast fail over, easier management, and zero data loss disaster protection, with Data Guard, the only complete data protection solution available on the market.

Oracle can runs on UNIX, Linux and Windows platform. However, it is expensive and separate licenses are required for each of its database engine.

2.4.2 MySQL

MySQL is a relational database management system. MySQL stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The tables are linked by defined relations making it possible to combine data from several tables on request.

MySQL is a small, compact, easy to use database server, ideal for small and medium sized applications. It is client/server implementation that consists of a server and many different client programs. It is available on a variety of UNIX platforms, Linux, Windows NT, Windows 95/98 and Windows 2000.

MySQL is Open Source Software. Open Source means that it is possible for anyone to use and modify. Anybody can download MySQL from the Internet and use it without paying anything. Anybody can study the source code and change it to fit their needs.

2.4.3 Microsoft SQL Server 2000

SQL Server 2000 provides the enterprise data management platform your organization needs to adapt quickly in a fast-changing environment.

Benchmarked for scalability, speed, and performance, SQL Server 2000 is a fully enterprise-class database product, providing core support for Extensible Markup Language (XML) and Internet queries.

Some of the features of Microsoft SQL Server 2000:

- **Easy-to-Use Business Intelligence (BI) Tools**

- Through rich data analysis and data mining capabilities that integrate with familiar applications such as Microsoft Office, SQL Server 2000 enables you to provide all of your employees with critical, timely business information tailored to their specific information needs.

- **Self-Tuning and Management Capabilities**
  - Revolutionary self-tuning and dynamic self-configuring features optimize database performance, while management tools automate standard activities. Graphical tools and wizards simplify setup, database design, and performance monitoring, allowing database administrators to focus on meeting strategic business needs.

- **Data Management Applications and Services**
  - SQL Server 2000 provides a powerful and comprehensive data management platform. Every software license includes extensive management and development tools, a powerful extraction, transformation, and loading (ETL) tool, business intelligence and analysis services, and new capabilities such as Notification Services.

2.5 Application Server

Application servers are strategic components for any IT infrastructure. They provide a run-time environment and back-end services that give applications the scalability, reliability and performance most enterprises require.

Today's application-server market and the e-commerce market are beginning to converge; a number of vendors are now offering a single platform that includes the underlying application server along with e-commerce and e-business functionality.

While some application-server vendors still offer their own development tools, many bundled in leading third-party tools in their solutions and can support applications that have been built in other development environments. This approach gives organizations new flexibility for building applications and lets them continue to use development tools that are already deployed and for which they already have expertise.

2.5.1 IBM WebSphere

IBM WebSphere Application Server Advanced Edition 3.5 is for large to midsize companies seeking an enterprise-level e-business solution and for those that want to develop in Java across a common platform. Its strengths include its reliability and integration capabilities, globalization, and platform support. However, the product lacks strong support for third-party management and EAI (enterprise application integration) products.

2.5.2 BEA WebLogic

BEA WebLogic Server is a fully featured, standards-based application server providing the foundation on which an enterprise can build its applications. For all of the crucial tasks of application development and deployment —from integrating enterprise systems and databases to delivering services and collaborating over the Internet —the starting place is BEA WebLogic Server. With its comprehensive set of features, compliance with open standards, multi-tiered architecture, and support for component-based development, Internet-savvy businesses are choosing WebLogic Server to develop and deploy best-of-breed applications.

2.5.3 ORACLE*9i* Application Server

Oracle*9i* Application Server (Oracle9iAS) offers the industry's fastest most complete and integrated J2EE-certified application server. Oracle9iAS has revolutionized the fast-growing application server market by being first to integrate all the technology required to build and deploy e-business portals, transactional applications, and Web services in a single product. Oracle9iAS supports all key Java, XML and emerging Web services industry standards.

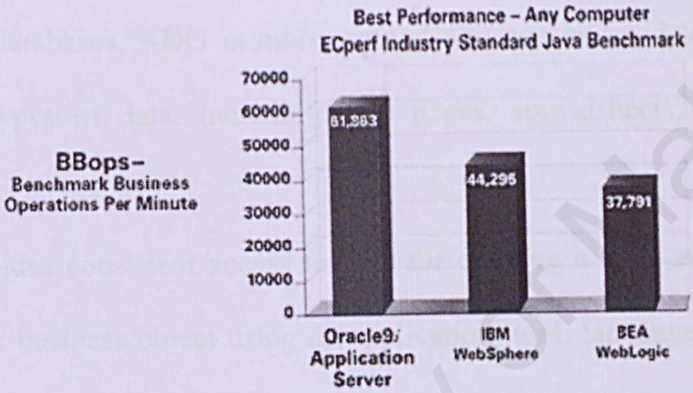


Figure 2-15: Application servers' comparison

2.6 Data Access Technology

DPMS will require data access technology to enable communication and access to its various databases. A few of the Microsoft Data access strategy and technology is reviewed and considered.

2.6.1 Universal Data Access (UDA)

UDA is a high-level specification developed by Microsoft for accessing data objects regardless of their structure. The strategy of Universal Data Access is to assure open, integrated, standards-based access to all types of data that is from SQL to non-SQL to even unstructured data across a wide variety of applications, from traditional client/server to the web. The main components of UDA are ADO, OLE DB and ODBC.

2.6.2 Active Data Object(ADO)

Active Data Object (ADO) is the Microsoft's newest high-level interface for data objects that most applications developers will use.

ADO is designed to eventually replace *Data Access Objects (DAO)* and *Remote Data Objects (RDO).* Unlike RDO and DAO, which are designed only for accessing relational databases, ADO is more general and can be used to access all sorts of different types of data, including web pages, spreadsheets, and other types of documents.

ADO provides consistent access to data for creating a front-end database client or middle-tier business object using an application, tool, language, or even an Internet browser. ADO is the single data interface for developers creating 1 to n-tier client/server and Web-based data-driven applications.

2.6.3 OLE DB

OLE DB Providers are the data access engines or services, as well as the business logic components that these applications can use in a highly interoperable, component-based environment.

OLE DB is a set of interfaces that are designed to provide data access to *all* data, regardless of type, format or location. It effectively "componentizes" database and related data processing functionality, breaking it up into interoperable components that can run as middleware on the client or server across a wide variety of applications. The OLE DB architecture provides for components such as direct data access interfaces, query engines, cursor engines, optimizers, business rules and transaction managers.

The concept of OLE DB is to explode the database into its basic parts. OLE DB delivers components, external to the database, that provide this typical database functionality in reusable component architecture. And these components, because they are not directly linked to the database itself, can be shared across multiple applications, systems and data stores to provide a higher level, universal interface.

2.6.4 Open Database Connectivity (ODBC)

ODBC is a standard database access method developed by Microsoft Corporation. The goal of ODBC is to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data. ODBC manages this by inserting a middle layer, called a database *driver*, between an application and the DBMS. The purpose of this layer is to translate the application's data queries into commands that the DBMS understands. For this to work, both the application and the DBMS must be *ODBC-compliant* -- that is, the application must be capable of issuing ODBC commands and the DBMS must be capable of responding to them. Since version 2.0, the standard supports SAG SQL.

2.6.5 Java Database Connectivity (JDBC)

JDBC technology is an API that lets you access virtually any tabular data source from the Java programming language. It provides cross-DBMS connectivity to a wide range of SQL databases, and now, with the new JDBC API, it also provides access to other tabular data sources, such as spreadsheets or flat files.

The JDBC API allows developers to take advantage of the Java platform's "Write Once, Run Anywhere" capabilities for industrial strength, cross-platform applications that require access to enterprise data. With a JDBC technology-enabled driver, a developer can easily connect all corporate data even in a heterogeneous environment.

2.6.6 ADO.NET

ADO.NET is an evolution of the ADO data access model that directly addresses user requirements for developing scalable applications. It was designed specifically for the web with scalability, statelessness and Extensible Markup Language (XML) in mind.

ADO.NET uses some ADO objects, such as the connection & command objects and also introduces new objects. Key new ADO.NET objects include DataSet, DataReader and DataAdapter.

The important distinction between this evolved stage of ADO.NET and previous data architectures is that there exists an object – the DataSet – that is separated and distinct from any data stores. Because of that, the DataSet functions as a standalone entity. You can think of the DataSet as an always disconnected recordset that knows nothing about the source or destination of the data it contains. Inside a DataSet, much like a database, there are tables, columns, relationships, constraints, views and so forth.

A DataAdapter is the object that connects to the database to fill the DataSet. Then, it connects back to the database to update the data there based on operations performed while the DataSet held the data. In the past, data processing has been primarily connection-based. Now in an effort to make multi-tiered application more efficient, data processing is turning to a message-based approach that revolves around chunks of information. At the center of this approach is the DataAdapter, which provides a bridge to retrieve and save data between a DataSet and its source data store. It accomplishes this by means of requests to the appropriate SQL commands made against the data store.

The XML-based DataSet object provides a consistent programming model that works with all models of data storage: flat, relational and hierarchical. It does this by having 'no knowledge' of the source of its data and by representing the data that it holds as collection and data types. Na matter what the source of the data within the DataSet is, it is manipulated through the same set of standard APIs exposed through the DataSet and its subordinate objects.

2.7 Programming Tool

2.7.1 Microsoft.NET

Microsoft.NET is software that connects information, people, systems and devices. It spans clients, servers and developer tools and consists of:

- The .NET framework programming model that enables developers to build Web-based applications, smart client applications and XML Web services applications which expose their functionality programmatically over the network using standard protocols such as SOAP and HTTP.
- Developer tools such as Microsoft Visual Studio .NET which provide a rapid application integrated development environment for programming with the .NET Framework.
- A set of servers, including Microsoft Windows 2000, Microsoft SQL Server and Microsoft BizTalk Server that integrates, runs, operates, and manages XML web services and applications

- show the .NET architecture
- Client software such as Windows XP, Windows CE, and Microsoft Office XP that helps developers deliver a deep and compelling user experience across a family of devices and existing products.

2.7.1.1 .NET Framework

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but internet distributed or executed remotely.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted 3rd party
- To provide a code-execution environment that minimizes software deployment and versioning conflicts
- To provide code-execution environment it eliminates the performance problems of scripted or interpreted environments
- To make the developer experience consistent across widely varying types of applications such as windows based applications and web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET framework can integrate with any other code.

The .NET framework has 2 main components: the common language runtime and the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management and remoting while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML web services. The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET framework not only provides several runtime hosts, but also supports the development of 3rd party runtime hosts. The following illustration shows the relationship of the common language runtime and the class library to your applications and to the overall system. It also shows how managed code operates within a larger architecture.

2.8 Programming Languages for Websites

2.8.1 ASP.NET

ASP.NET, Microsoft's way to build powerful web sites that go beyond regular HTML pages. ASP.NET is a robust, easy-to-use solution for developing complex web sites. Rather than building simple pages that are only for display, you'll create web pages that visitors can interact with in wonderful ways. ASP.NET is a server technology that brings together different pieces of the web to give web site developers more power than ever.

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web Services use Internet Information Services (IIS) and ASP.NET as the publishing mechanism for applications and both have a collection of supporting classes in the .NET Framework.

There's model for communicating between servers and clients, known as the event-driven model. The server waits around for something to happen on the client. Once it does, the server takes action and performs some pieces of functionality. If you type some text on the web page, the server responds to it because ASP.NET supplies information about what the client, a client-side script execute and sends into the server, just as submitting a form sends information to a server. The browser is simply an unknowing accomplice. It thinks it's just doing its job-displaying HTML. So, client-side scripts can't exactly interact with the server-side, but they can rely message via posts to the server. Thus ASP.NET ties together the server and the client, which allows developers to do things in web pages that weren't possible before. You don't have to focus on handling requests and responses but are free to concentrate on building the logic. You can react to user events immediately instead of waiting until forms are submitted. And you can know the structure of the User Interface (UI) and how to handle it ahead of time. ASP.NET truly makes developer's lives easier.

2.8.2 JavaServer Pages (JSP)

JavaServer Pages™ (JSP) is a web-scripting technology that can mix static HTML content with server-side scripting to produce dynamic output. By default, JSP uses Java as its scripting language; however, the specification allows other languages to be used, just as ASP can use other languages (such as JavaScript and VBScript). While JSP with Java will be more flexible and robust than scripting platforms based on simpler languages like JavaScript and VBScript.

JSP provides a number of server-side tags that allow developers to perform most dynamic content operations. So developers who are only familiar with scripting, or even those who are simply HTML designers, can use JSP tags for generating simple output. Advanced scripters or Java developers can also use the tags, or they can use the full Java language if they want to perform advanced operations in JSP pages.

2.8.3 Javascript

Javascript is a scripting language developed by Netscape to enable web authors to design interactive sites. Javascript is different from Java. Although it shares many of the features and structures of the full Java language, it was developed independently. Javascript can interact with HTML source code to enable web authors to spice up their sites with dynamic content. JavaScript is endorsed by a number of software companies and is an open language that anyone can use without purchasing a license. It is supported by recent browsers from Netscape and Microsoft, though Internet Explorer supports only a subset, which Microsoft calls Jscript.

2.9 Programming Languages for Pocket PC

2.9.1 Visual Basic.NET 2003 (VB.NET 2003)

Visual Basic .NET 2003 provides the easiest, most productive language and tool for rapidly building applications for Microsoft Windows® and the Web. Ideal for existing Visual Basic developers as well as new developers in the Microsoft .NET development environment, Visual Basic .NET 2003 delivers enhanced visual designers, increased application performance, and a powerful integrated development environment (IDE) to get you on the fast track to application development.

New support for mobile device application development enables you to use the skills you already have to build mobile Web applications and rich Pocket PC-based software. The XML Web service project template makes creating remote business logic components as easy as creating any class in Visual Basic. Optional object-oriented language constructs enable you to save time by reusing code and Windows Forms UI.

Visual Studio .NET 2003 Professional, simplifies smart device application development with new integrated support for the .NET Compact Framework. Visual Basic developers skilled in desktop development will find that they can easily use the same visual designers, code editor, debugger, and more importantly, the same programming model to create applications for the Pocket PC as well as other smart devices powered by the .NET Compact Framework. A powerful new integrated emulator also enables developers to build solutions without requiring hardware.

2.9.2 Microsoft Visual C#.NET

Microsoft Visual C# .NET 2003 is the comprehensive toolset for creating XML Web services and Microsoft .NET—connected applications for Microsoft Windows® and the Web.

This robust development package, which uses the component-oriented C# development language, offers beginning and intermediate developers with C++ or Java experience a modern language and environment for creating next-generation software. Visual C# .NET 2003 delivers superior functionality for streamlining business processes, including:

- Rapid design, development, and deployment support for creating and consuming Web services.
- Form designers and visual controls for creating rich Windows-based applications.
- Authoring tools and services for building powerful Microsoft .NET server-based solutions.
- Migration tools for converting Java-based projects to the Microsoft .NET development environment.

With Visual C# .NET 2003, developers can build solutions for the broadest range of clients, including Windows, the Web, and mobile or embedded devices. Using this elegant programming language and tool, developers can leverage their existing C++ and Java-language skills and knowledge to be successful in the .NET environment.

CHAPTER 3

~ Methodology ~

Chapter 3 - Methodology

3.1 Methodology

The system development methodology is a method to create a system with a series of steps or operations or can be defined as system life cycle model. Every system development process model (see Figure 3-1) includes system requirements (user, needs, resource) as input and a finished product as output.

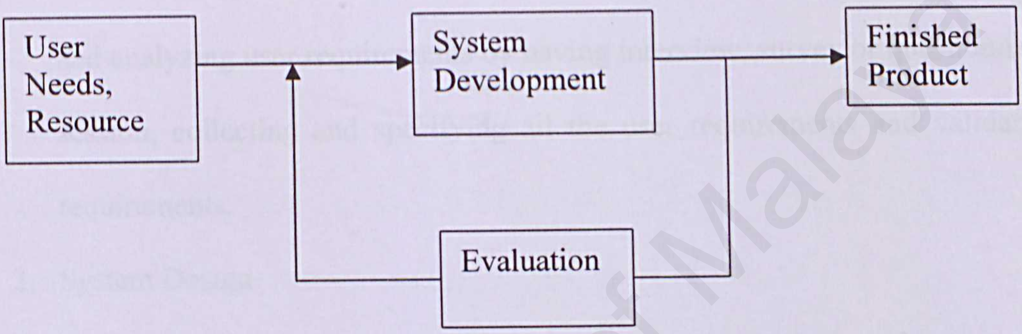


Figure 3-1: System Development Process Model

There are several process models in system development:

1. Waterfall Model
2. V Model
3. System Development Life Cycle (SDLC)
4. Spiral Model

Waterfall Model is chosen for DPMS because:

□ A good specification to begin with.

□ Easy to use

□ Systematic

□ Scope of project well understand

- Project risks have been accessed and are considered to be low.

3.1.1 Waterfall Model

Waterfall Model consists of eight stages that are depicted as cascading from one to another (see Figure 3-1). Each development stage should be completed before the next begins. The eight stages are:

1. Requirements Analysis

   Understanding and determining users need by having brainstorming, eliciting and analyzing user requirements by having interview, survey or questionnaire session, collecting and specifying all the user requirements and validating requirements.

2. System Design

   Outlining system functional by having feasibility studies or case studies on current system, determining and specifying hardware or software architecture and verifying system design.

3. Program Design

   Determining and specifying program design and database design and verifying program design.

4. Coding

   Involving programming, personal planning, tool acquisition, database development, component level documentation and programming management.

5. Unit and Integration Testing

   Test units separately and integrate the tested units. Then, testing on the integrated units.

6. System Testing

Combining all the integrated units into a system. Testing on the system.

Specifying, reviewing and updating of the system test and validating of system.

7. Acceptance Testing

   Testing on system completed. The system is delivered.

8. Operation and Maintenance

   Control and maintain the system. Revalidating of system.

The system has to be validated and verified during the stage of system testing. The verification is to make sure that the function in the DPMS works correctly and to check the quality of the implementation. The validation is to ensure that DPMS has implemented all the requirements in the specification.

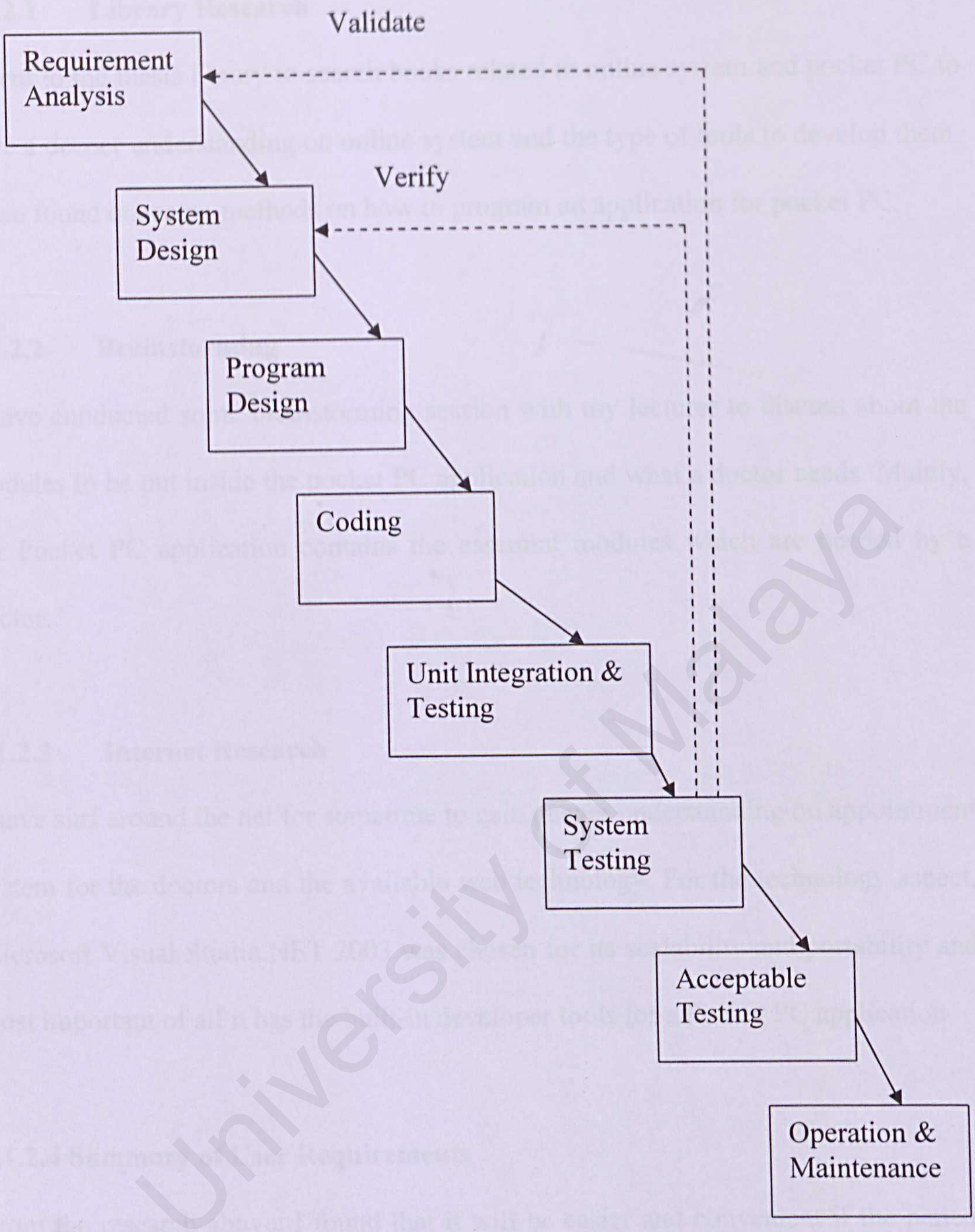


Figure 3-2: Waterfall Model

3.1.2 Techniques Used To Define Requirements

Effective and appropriate techniques must be used to define and elicit users' requirements. Research methods that usually used are library research, brainstorming and Internet research.

3.1.2.1 Library Research

I went to the thesis library to search books related to online system and pocket PC to have a deeper understanding on online system and the type of tools to develop them. I also found out some methods on how to program an application for pocket PC.

3.1.2.2 Brainstorming

I have conducted some brainstorming session with my lecturer to discuss about the modules to be put inside the pocket PC application and what a doctor needs. Mainly, the Pocket PC application contains the essential modules which are needed by a doctor.

3.1.2.3 Internet Research

I have surf around the net for sometime to gain deeper understanding on appointment system for the doctors and the available web technology. For the technology aspect, Microsoft Visual Studio.NET 2003 was chosen for its scalability and portability and most important of all it has the built-in developer tools for a Pocket PC application.

3.1.2.4 Summary of User Requirements

From the research above, I found that it will be easier and convenient if the patient can make their appointments online because it is faster and the doctors can get them instantly. The doctors are also able to plan out their schedule better because their patient can make appointments with them much earlier.

CHAPTER 4

~ System Requirements Analysis ~

Chapter 4 - System Requirements Analysis

4.1 Functional Requirements

Functional requirement is a statement of the service or functions that a system should provide, how the system reacts to particular inputs, and how the system should behave in particular situations. **[Sommerwille, 1998]**

The functional requirement for DPMS consists of two main parts: Doctor Management Section which is an application for the Pocket PC and Patient Section which is an online website for patients. For Doctor Management Section, functional requirement consists of Appointment module, Tasks module, Schedule module and Contacts module. For Patient Section, functional requirement consists of Registration module, Booking module and Doctor's schedule module.

4.1.1 Pocket PC Modules

4.1.1.1 Overview Module

This module will allow the doctors to view their tasks and appointments in a bird's eye view. Having this module, the doctors are able to see all their tasks and appointments so that it will help them to plan their schedule with more efficiency and without any clashes.

4.1.1.2 Tasks Module

Doctors can insert their important task which they need to complete in this module. There is also a setting for the alarm to sound according to the time and date specified by the doctor to remind them of the tasks. The tasks can also be prioritized according to their level of importance.

4.1.2.3 Contacts Module

This module allows the doctors to keep all of his or her patients' contacts including their addresses and personal pictures. This will enable the doctors to be able to keep in touch with their patients.

4.1.2.4 Appointment Module

This module allows the doctors to key in their daily appointments. The appointment module also allow doctors to have easy access to the patient's contact as there is a button for the doctor to click to get the patient's mobile number.

4.2 Non-Functional Requirements Analysis

Non-functional specifications are the constraints under which a system must operate and the standards which must be met by the delivered system **[Sommerwille, 1995]**. The new Doctor's Personal Management System website must ensures certain web application qualities like user-friendliness, correctness, functionality, reliability, flexibility, efficiency as well as maintainability.

4.2.1 User-Friendliness

User interfaces design creates an effective communication medium between a human and a computer. Therefore, it is very important to make sure that the interfaces fulfill user-friendliness so that it would not cause trouble to users. The Golden Rules **[Mandel, 1997]** coins three rules:

- Place the user in control

  This will define interaction modes in a way that does not force a user into unnecessary or undesired actions. Besides, it also provides flexible interaction for different users for instance via mouse movement and keyboard commands.

- Reduce the user's memory load

  One of the principles that enable an interface to reduce the user's memory load is by reducing demand on short-term memory. The interface should be designed to reduce the requirements to remember past actions and results.

- Make the interface consistent

  The interface design should apply to consistent fashion where all visual information must be organized according to a design standard that is maintained throughout all screen displays. Apart from that, input mechanisms are constrained to a limited set that are used consistently throughout the application. Lastly, mechanisms for navigating from task to task are consistently defined and implemented.

4.2.2 Correctness

A program or system must operate correctly or it provides little value to its users. Correctness is the degree to which the software performs its required function. To ensure this application quality, lots of testing and trial-and-errors will be carried out.

4.2.3 Functionality

The functionalities stressed here are the searching and retrieving capability, which is very important in any web applications that deal with data retrieval from existing database. Besides, navigation and browsing features as well as application domain-related features will be taken into account.

4.2.4 Reliability

Reliability is the extent to which a program can be expected to perform its intended function with required precision **[Pressman, 2001]**. It is closely related to correct link processing, error recovery and user input validation and recovery. This quality is essential as it indicates how far users will be confident in the implementation of the new computerized system in getting daily minutes processing done.

4.2.5 Flexibility

For the DPMS, flexibility of the system is stress on the Java-based system, which is able to receive user request from multi-platform. In other words, whether a user makes request from Windows-platform computer or Linux-platform computer, the user is able to retrieve the appropriate output. This is indeed the strength of the technology used in developing the DPMS.

4.2.6 Efficiency

Undeniable, efficiency is the main key for implementing the new meetings management system. Efficiency is understood as the ability of a process procedure to be called or accessed unlimitedly to produce similar performance outcomes at an acceptable or credible speed **[Sommerwille, 1995]**. Efficiency is measured base on response time performance, page generation speed and graphics generation speed.

4.2.7 Maintainability

System maintenance accounts would require more effort if the system is not designed according to good programming practices. Maintainability is the ease with which a program can be corrected if an error is encountered, adapted if its environment changes, or enhanced if the customer desires a change in requirements **[Pressman, 2001]**.

4.2.8 Security

The proposed system has also security measures to minimize the risk of data exposure to unauthorized people.

4.3 Chosen Platform, Application Server, Database and Tools

4.3.1 Chosen Operating System

For the DPMS, Windows is chosen as the development platform. Microsoft's Windows XP Professional is built to work with a series of microprocessors from the Intel Corporation that share the same or similar sets of instructions.

The main reason for choosing Microsoft's Windows XP Professional as the development operating system is because Windows XP Professional is widely used in many industrials and fields. Therefore, the implementation of the new system can be done easily and effectively.

4.3.2 Chosen Database Management System

After considering some DBMS in the market, Microsoft SQL Server 2000 has been chosen as the DBMS for DPMS.

Microsoft SQL Server 2000 is chosen because it provides the enterprise data management platform your organization needs to adapt quickly in a fast-changing environment.

Some of the features of Microsoft SQL Server 2000:

- **Easy-to-Use Business Intelligence (BI) Tools**
    - Through rich data analysis and data mining capabilities that integrate with familiar applications such as Microsoft Office, SQL Server 2000 enables you to provide all of your employees with

critical, timely business information tailored to their specific information needs.

- **Self-Tuning and Management Capabilities**
    - Revolutionary self-tuning and dynamic self-configuring features optimize database performance, while management tools automate standard activities. Graphical tools and wizards simplify setup, database design, and performance monitoring, allowing database administrators to focus on meeting strategic business needs.
- **Data Management Applications and Services**
    - SQL Server 2000 provides a powerful and comprehensive data management platform. Every software license includes extensive management and development tools, a powerful extraction, transformation, and loading (ETL) tool, business intelligence and analysis services, and new capabilities such as Notification Services.

4.3.3 Chosen Development Data Access Technology

For DPMS, ADO.NET has been chosen as the Development Data Access Technology.

ADO.NET is a set of classes that expose data access services to the .NET programmer.

ADO.NET provides functionality to developers writing managed code similar to the functionality provided to native COM developers by ADO. ADO.NET provides consistent access to data sources such as Microsoft® SQL Server™, as well as data sources exposed through OLE DB and XML. Data-sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, manipulate, and update data.

ADO.NET provides a rich set of components for creating distributed, data-sharing applications. It is an integral part of the .NET Framework, providing access to relational data, XML, and application data. ADO.NET supports a variety of development needs, including the creation of front-end database clients and middle-tier business objects used by applications, tools, languages, or Internet browsers.

4.3.4 Chosen Development Application Server

The chosen Development Application Server is Internet information Server (IIS). Internet Information Services is a complete Web server available in all versions of Windows Server 2003. Designed for intranets, the Internet, and extranets, IIS makes it possible for organizations of all sizes to quickly and easily deploy powerful Web sites and applications. In addition, IIS provides a high-performance platform for applications built using the Microsoft .NET Framework.

4.3.5 Chosen Web Development Language

The chosen Web Development Tool will be ASP.NET. ASP.NET is chosen because of some of the reasons below:-

- Easy Programming Model
    - ASP.NET makes building real world Web applications dramatically easier. ASP.NET server controls enable an HTML-like style of declarative programming that let you build great pages with far less code than with classic ASP.

- Flexible Language Options
    - ASP.NET lets you leverage your current programming language skills. Unlike classic ASP, which supports only interpreted VBScript and JScript, ASP.NET now supports more than 25 .NET languages.
- Rich output caching
    - ASP.NET output caching can dramatically improve the performance and scalability of your application. When output caching is enabled on a page, ASP.NET executes the page just once, and saves the result in memory in addition to sending it to the user. When another user requests the same page, ASP.NET serves the cached result from memory without re-executing the page.
- Memory Leak, DeadLock and Crash Protection
    - ASP.NET automatically detects and recovers from errors like deadlocks and memory leaks to ensure your application is always available to your users.
- Mobile Web Device Support
    - ASP.NET Mobile Controls let you easily target cell phones, PDAs -- over 80 mobile Web devices -- using ASP.NET. You write your application just once, and the mobile controls automatically generate WAP/WML, HTML, or iMode as required by the requesting device.

4.3.6 Chosen Development Language for Pocket PC

The chosen development tool for pocket PC is Visual Basic.NET 2003 or better known as VB.NET 2003. Visual Basic .NET 2003 provides the easiest, most productive language and tool for rapidly building applications for Microsoft Windows® and the Web. Ideal for existing Visual Basic developers as well as new developers in the Microsoft .NET development environment, Visual Basic .NET 2003 delivers enhanced visual designers, increased application performance, and a powerful integrated development environment (IDE) to get you on the fast track to application development.

Some of the features of VB.NET 2003 are as follows:-

- Build Robust Windows-based Applications
    - With new Windows Forms, developers using Visual Basic .NET 2003 can build Windows-based applications that make use of the rich user interface features available in the Windows operating system.
- Resolve Deployment and Versioning Issues
    - With Visual Basic .NET 2003, issues with Component Object Model (COM) registration and DLL overwrites are eliminated. Side-by-side versioning enables multiple versions of the same components to exist on the same machine, allowing applications to use the specific version of a component with which they were tested.
- Easily Create Web Applications
    - Using the new Web Forms Designer in Visual Basic .NET 2003, Visual Basic developers can apply the skills they have today to build true thin-client Web-based applications. Drag-and-drop Web Form creation delivers "Visual Basic for the Web" while code behind forms enables developers to replace limited scripting capabilities of the past with the full power and speed of the compiled Visual Basic .NET language.

- Visual Basic Skills to Program Smart Devices
    - The version of Visual Basic .NET 2003 for professional developers, Visual Studio .NET 2003 Professional, simplifies smart device application development with new integrated support for the .NET Compact Framework.

CHAPTER 5

~ System Design ~

Chapter 5 – System Design

5.1 Introduction

System Design is a phase of the waterfall that the entire requirements for the system are translated into system characteristics. The requirements for system are regarding to the analysis that had been discussed in the previous chapter. System design includes the following issues:

- System Architecture Design
- System Functionality Design
- User Interface Design
- Database Design

5.2 Overview of System Architecture

5.2.1 Architecture design for web-based

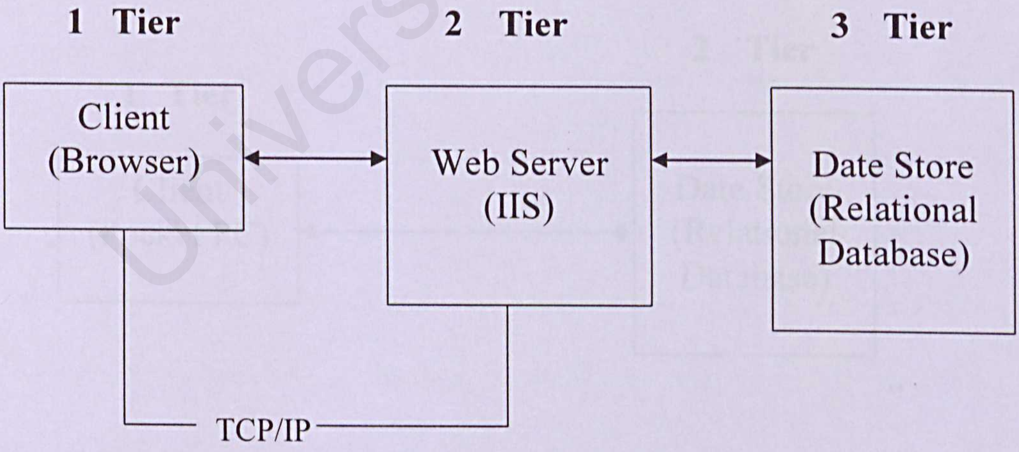


Figure 5-1: 3-Tier Architecture of Web-based

The conceptual architecture of the three-tier application applies when we split an application across three tiers are split into three logical components of the application:

user interface, computational logic and data storage. In reality, the three-tier Web applications generally consist of a Web browser for the user interface, a Web server connected to a "middle tier" application, and a persistent store that is frequently a relational database. (Refer to Figure 5-1)

DPMS will be using Visual Studio.NET as the web development tool. After the middle tier receives client's requests, in the same tier, ADO.NET will communicate with database to decide ways to process the data in order to generate desired output to users.

The main purpose of having three-tier architecture is to assign main functionality to each tier to ensure no function overlapped. Different people could handle each tier using different languages. Therefore, whenever there is error or system fault occurs, the problems can be detected and fixed easily without interfering with other tier.

5.2.2 Architecture design for Pocket PC

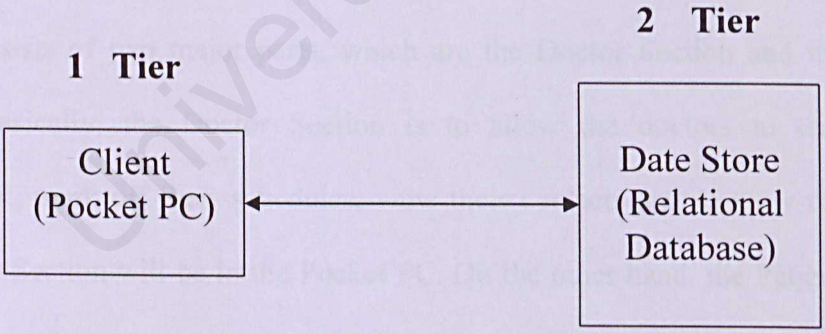


Figure 5-2: 2-Tier Architecture of Pocket PC

Visual Studio.NET will be used to as the Pocket PC development tool. Visual Basic.NET (VB.NET) will be used as the language to develop the interfaces. All of the client's request will be communicated to the SQL CE, which is a database for

Pocket PC and the database will decide ways to process the data in order to generate desired output to users.

5.3 System Functionality Design

5.3.1 System Structure Charts

The objective of system structure chart is to show how the modules in DPMS are related to each other.

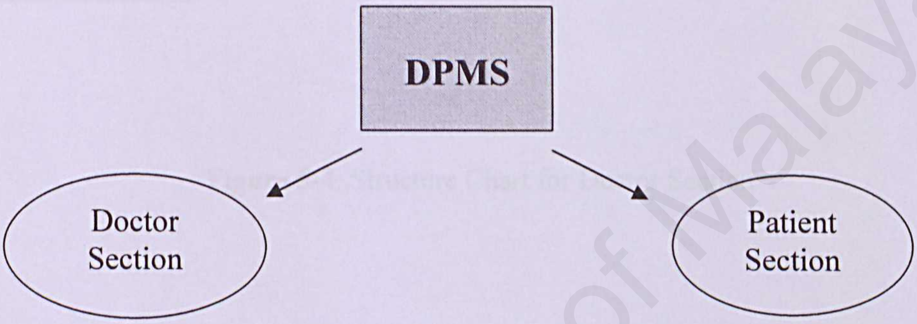


Figure 5-3: Structure Chart for DPMS

DPMS consists of two major parts, which are the Doctor Section and the Patient Section. Basically, the Doctor Section is to allow the doctors to check their appointment, mark off their schedules, view their contacts and to view their tasks. The Doctor Section will be in the Pocket PC. On the other hand, the Patient Section allows the patient to check the availability of the desired doctors and to book them for an appointment. The Patient Section will be web-based.

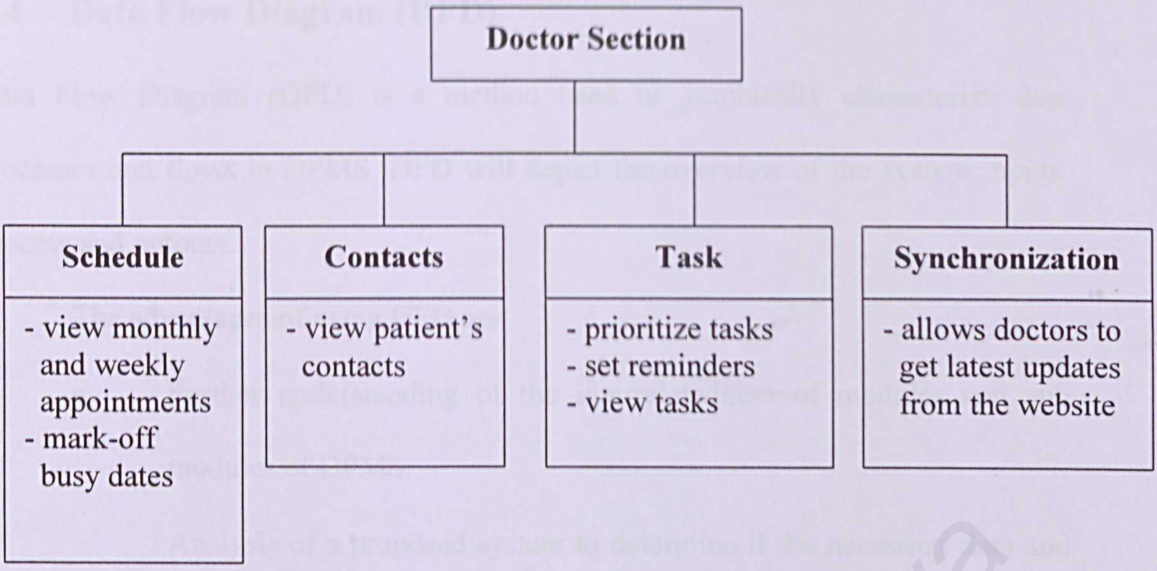

Doctor Section
Schedule
- view monthly and weekly appointments
- mark-off busy dates
Contacts
- view patient's contacts
Task
- prioritize tasks
- set reminders
- view tasks
Synchronization
- allows doctors to get latest updates from the website

Figure 5-4: Structure Chart for Doctor Section

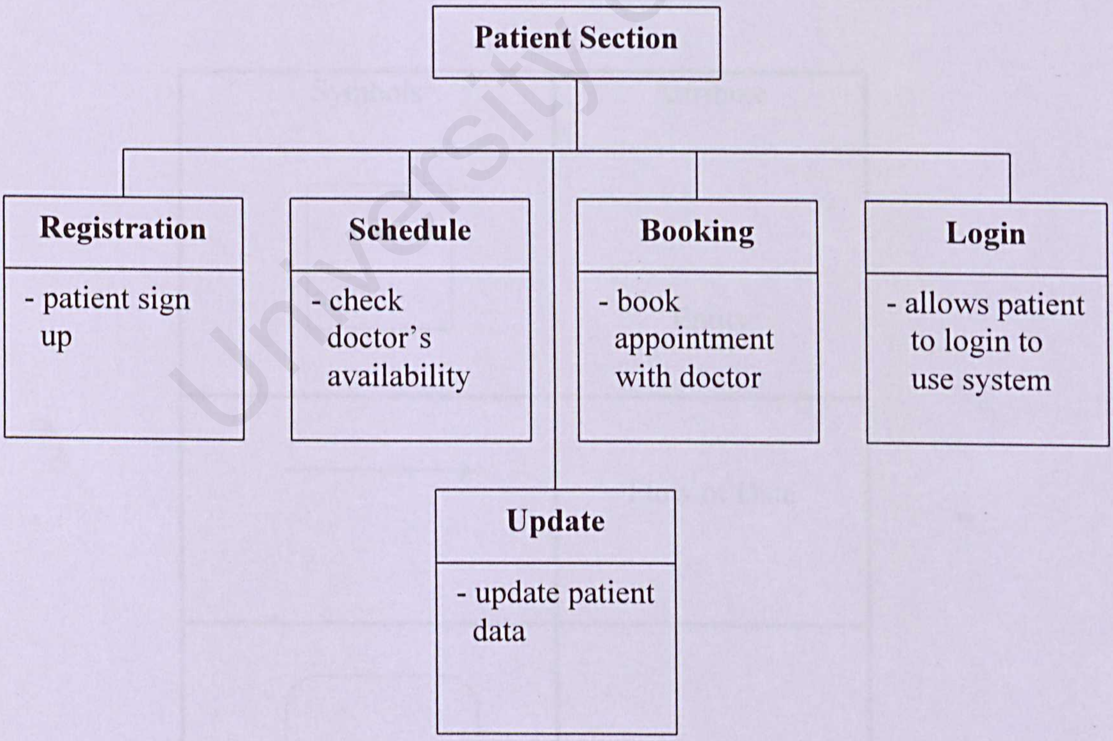

Patient Section
Registration
- patient sign up
Schedule
- check doctor's availability
Booking
- book appointment with doctor
Login
- allows patient to login to use system
Update
- update patient data

Figure 5-5: Structure Chart for Patient Section

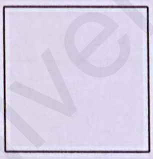
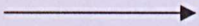
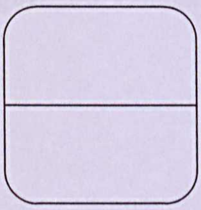
5.4 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) is a method used to graphically characterize data processes and flows in DPMS. DFD will depict the overview of the system inputs, process and outputs.

The advantages of using DFD are:

- Further understanding of the interrelatedness of modules and sub modules of DPMS.
- Analysis of a proposed system to determine if the necessary data and processes have been defined.

DFD is easy to be understood as it has symbols that specify the physical aspects of implementation. There four basic symbols in DFD: entity, flow of data, process and data stores (see Table 5-1).

Symbols	Attribute
	Entity
	Flow of Data
	Process

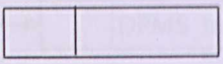
	Data Store

Table 5-1: DFD Symbols

The convention, which is used to design DFD are based on the work by C.Gane and T.Sarson. The data flow is conceptualized with a top-down perspective. So, the Context Level Diagram will be drawn, followed by the Diagram 0. Diagram 0 is an overview process of all the major modules in DPMS that includes all the data stores, entities and process involved.

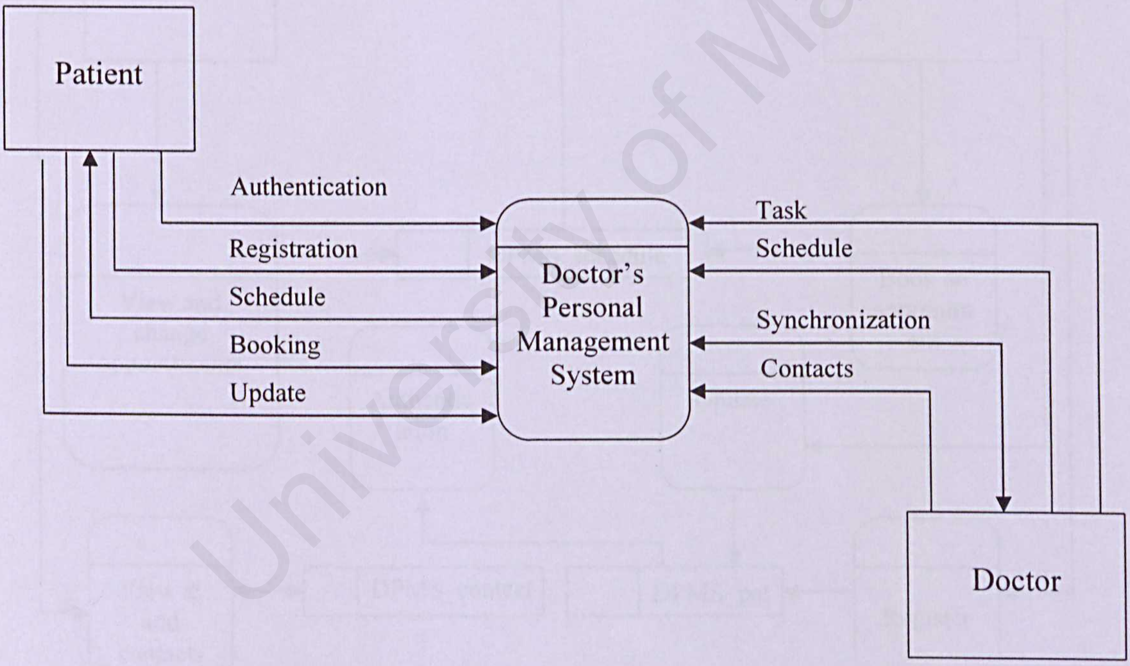


Figure 5-6: Context Level Diagram for DPMS

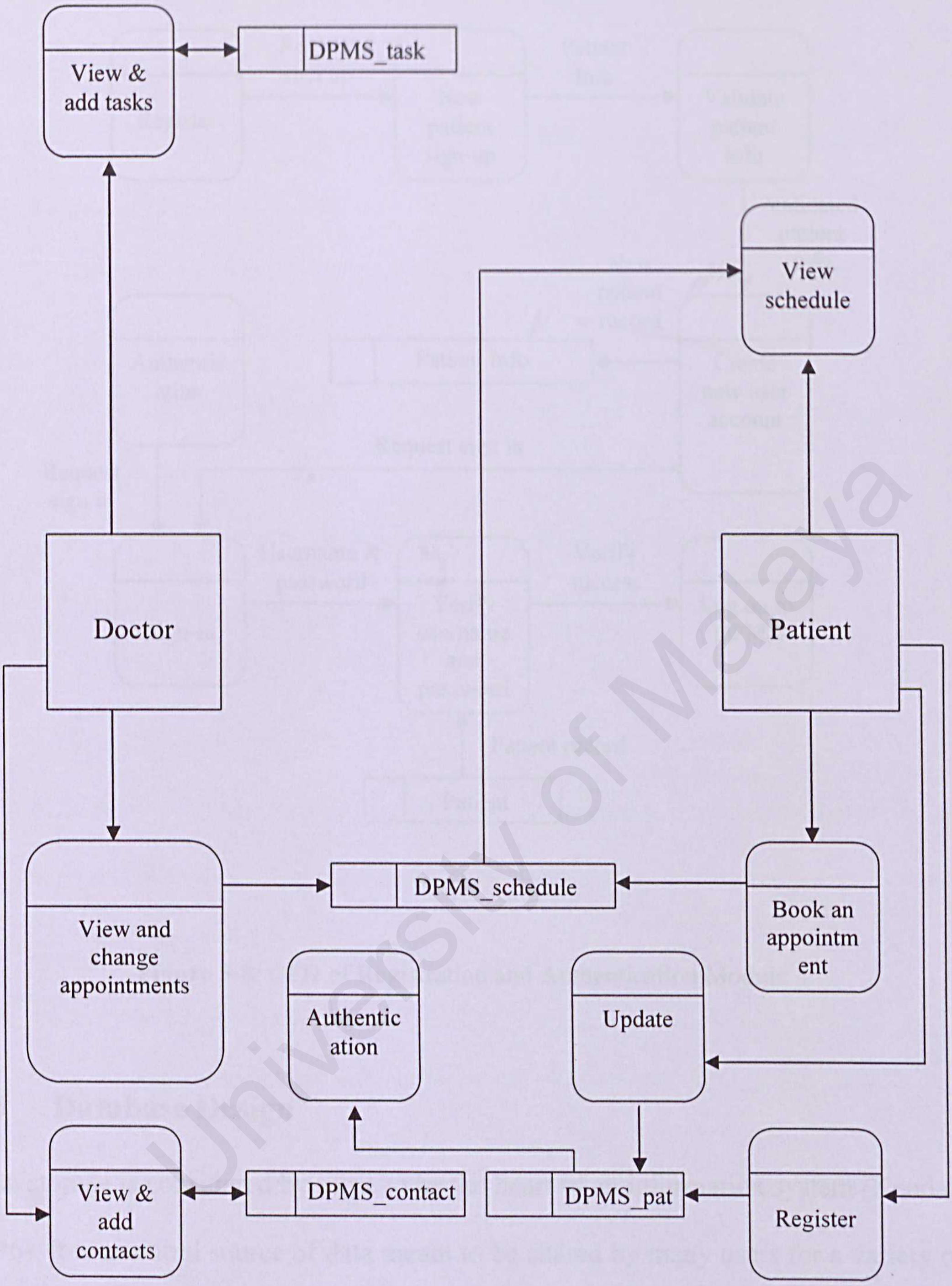


Figure 5-7: Diagram 0 for DPMS

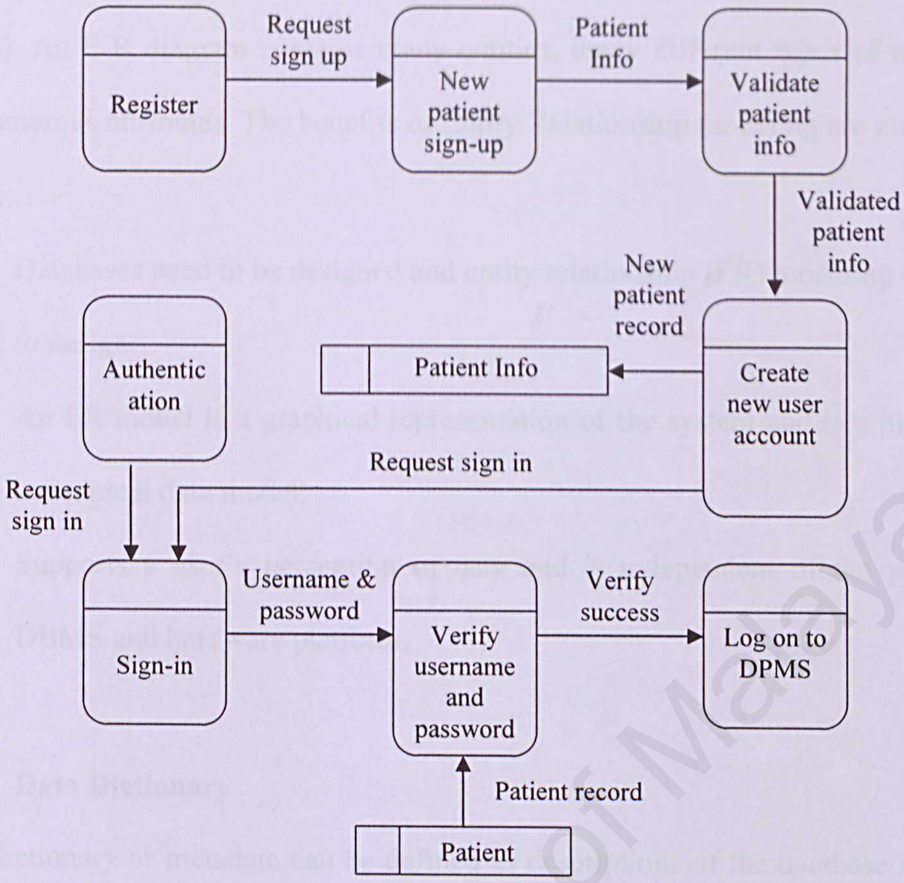


Figure 5-8: DFD of Registration and Authentication Module

5.5 Database Design

Data storage is considered by some to be the heart of an information system (Kendall, 1996). It is a central source of data meant to be shared by many users for a variety of applications. The heart of a database is the DBMS (database management system), which allows the creation, modification and updating of the database; the retrieval of data; and the generation of reports. The main objective of database design is to make sure that data is available when the user wants to use it. Apart from that, the accuracy, consistency and integrity of data must be assured from time to time, to provide efficient data storage as well as efficient updating and retrieval.

In 1976, Peter Chen had introduced the use of the entity-relationship model (E-R Model). An E-R diagram contains many entities, many different types of relations, and numerous attributes. The benefits of Entity Relationship modeling are mentioned below:

i. Databases need to be designed and entity relationship (ER) modeling is an aid to design.

ii. An ER model is a graphical representation of the system and is a high-level conceptual data model.

iii. Supports a user's perception of data and is independent of the particular DBMS and hardware platform.

5.5.1 Data Dictionary

Data dictionary or metadata can be defined as descriptions of the database structure and contents. Data dictionary defines the field, field type and descriptions of each table.

In DPMS, one database had been defined namely DPMS and contained 5 tables, which are DPMS_task, DPMS_schedule, DPMS_appoint, DPMS_contact and DPMS_pat.

Database Name: DPMS

Table Name: MyTask

Field Name	Data Type	Length	Notes
subject	ntext	255	Task id
priority	ntext	255	
status	ntext	255	Status of tasks

startDate	datetime	10	
endDate	datetime	10	

Table 5-2: Table of MyTask

Table Name: MyApp

Field Name	Data Type	Length	Notes
subject	nText	255	Schedule id
name	nText	255	
date	datetime	10	
time	nText	255	
reminder	nText	255	

Table 5-3: Table of MyApp

Table Name: MyContacts

Field Name	Data Type	Length	Notes
name	nText	10	
jobTitle	nText	40	
phoneh	nText	255	
phonem	nText	255	
phoneo	nText	255	
home_add	nText	255	
home_city	nText	255	
home_state	nText	255	

Home_postcode	nText	255	
office_add	nText	255	
office_city	nText	255	
office_state	nText	255	
office_postcode	nText	255	

Table 5-4: Table of MyContacts

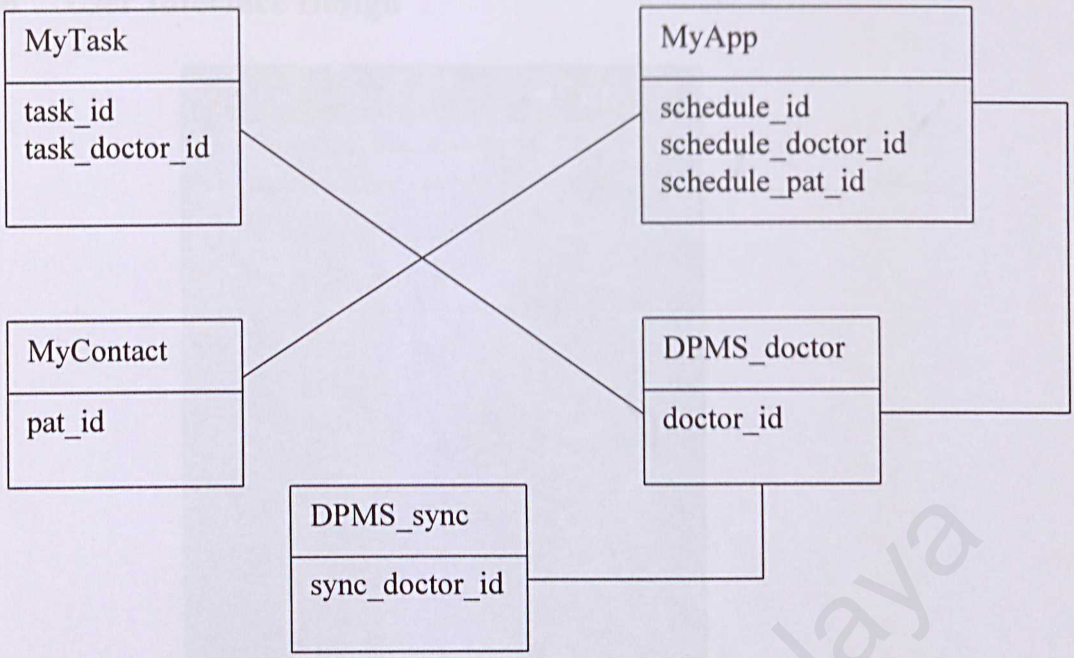
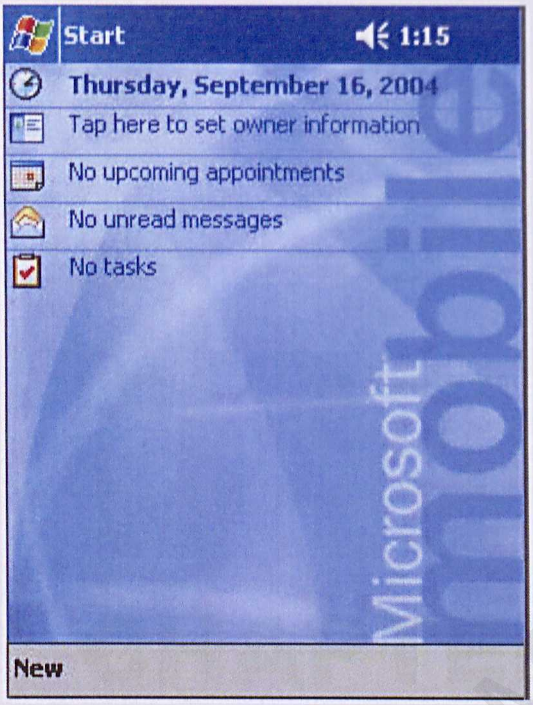


Figure 5-9: Class diagram for DPMS system

5.6 User Interface Design



Thursday, September 16, 2004

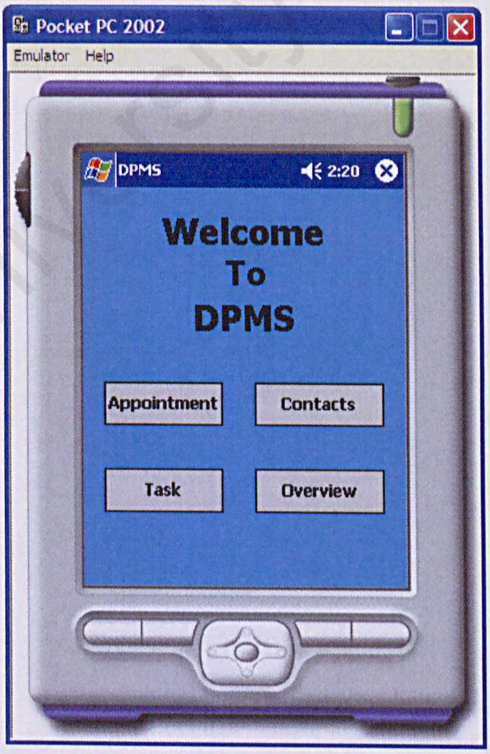
Tap here to set owner information

No upcoming appointments

No unread messages

No tasks

Figure 5-10: User Interface for Pocket PC



Welcome To DPMS

Figure 5-11: Proposed User Interface for DPMS

CHAPTER 6

~ System Implementation ~

Chapter 6 - Implementation

6.1 Introduction

The subsequent phase following the completing of the system design stage in the system development life cycle is the actual implementation of the system. Even though careful planning during the system design phase will bring about easier implementation of the system, this phase should still be carried out meticulously and cautiously.

Implementation should be carried out by following the system design rigorously as the design is a direct result of thorough research on the topic of the project, not forgetting the goals, objectives and taking into consideration other issues such as the targeted system users and the system's operating environment.

However, during the actual implementation of the system, it might happen that certain parts of the system design are found to be unsuitable. This is inevitable as during the design phase, it is easy to get over-enthusiastic and overlook important aspects of the system. Furthermore, it is hard to predict what will happen during the actual implementation.

Certain attributes of the system may have to be implemented differently from the original plan due to hardware or software limitations. As long as the system fulfills critical requirements such as the core functions of the system and do not violate the pre-determined business rules, modifications to the system design is allowable.

In this chapter, a thorough description will be given on the entire implementation process and the approaches used to implement the system.

6.2 Development Environment

The development environment depicted here are the settings, conditions and surroundings where the actual system development takes place. It is important to take note of the environment in which the development was carried out because it may have effect on the speed and effectiveness of the system's implementation.

6.2.1 Actual Hardware Requirement

The actual hardware specifications utilized for the entire implementation phase are as described below. It was found to be suitable and satisfactory for the entire period of the development stage.

Category	*Description*
Processor	Intel Centrino 1.3GHz
Hard Disk Size	30 GB
RAM Size	512MB
Combo Drive	24x10x12x/8x Combo Drive
VGA Card	Intel Extreme Graphics

Table 6-1: Hardware Specification for the Implementation Process

6.2.2 Actual Software Requirement

Table 6-2 shows the software specifications that were needed during the system's implementation. They are essential towards the development of the system, e.g. IIS 6.0 must be installed as the web server because the system utilized a web application as the graphical user interface. Furthermore, since the development is carried out in a .NET environment, thus the Microsoft .NET Framework must be installed before system implementation can be carried out.

Category	*Description*
Operating System	Microsoft Windows XP Professional
Web Server	Microsoft Internet Information Services 6.0
Special Requirements	Microsoft .NET Framework 1.1

Table 6-2: Software Specification for the Implementation Process

6.2.2.1 Software Tools for Development

Table 6-3 describes the software tools needed to develop the system. These development tools have been pre-determined earlier on (during the system design phase) as the selected tools for the system implementation. Among these development tools, the core system functions are fully developed with the Microsoft Visual Studio .NET 2003 suite.

Category	*Description*
Development Tools	Microsoft Visual Studio.NET 2003
Database	Microsoft SQL Server 2000 Microsoft SQL Server Compact Edition

Table 6-3: Software Tools for System Development

6.3 Approaches to the Development of the System

There are a few things that need to be considered before starting the real implementation of the system. These include determining the "rules and regulations" when it comes to converting the system design into actual coding. The following sections show a review of the coding methodologies, convention and best practices that are taken into consideration for developing the system.

6.3.1 Coding Methodology

Before diving straight into the system's implementation, a study of the different approaches that can be used to develop the system was carried out. Basically, there are two types of methodology which is used to develop a system:

1) top-down approach

2) bottom-up approach

6.3.1.1 Top-Down Coding Methodology

Top-down programming refers to a style of programming where an application is constructed starting with a high-level description of what it is supposed to do, and breaking the specification down into simpler pieces, until a level has been reached that corresponds to the primitives of the programming language to be used.

Top-down programming tends to generate modules that are based on functionality, usually in the form of functions or procedures. Typically, the high-level specification of the system states functionality. This high-level description is then refined to be a sequence or a loop of simpler functions or procedures, that are then themselves refined, etc.

Among the advantages of top-down implementations are:

i. Avoiding chaos of attempting to design the system all at once. Planning and implementing can be incredibly complex. Attempting to get all subsystems in place and running at once is basically planning to fail.

ii. Enables separate development works to be done in parallel on different but necessary subsystems that can save a great deal of time.

iii. Prevents the developer from getting mired in details that might cause him/her to lose sight of why the system is developed in the first place.

However, this top-down approach is not without its disadvantages. Among them are:

i. All decisions made from the start of the project depend directly or indirectly on the high-level specification of the application. However, specification tends to change over time. When that happens, there is a great risk that large parts of the application need to be rewritten.

ii. There is a chance that the system will be wrongly divided into unsuitable subsystems. Attention must be paid to overlapping needs and sharing of resources so that the partitioning of subsystems makes sense for the system.

iii. Modules generated are very specific to the application that is being written, thus not very reusable.

6.3.1.2 Bottom-Up Coding Methodology

Bottom-up programming refers to a style of programming where an application is constructed starting with existing primitives of the programming language, and constructing gradually more and more complicated features, until all of the application has been written.

In a language such as C or Java, bottom-up programming takes the form of constructing abstract data types from primitives of the language or from existing abstract data types. In Common Lisp, in addition to constructing abstract data types, it is common to build *functions* bottom-up from simpler functions, and to use macros to construct new *special forms* from simpler ones.

The advantages of bottom-up implementations are:

i. Testing is simplified since no stubs are needed. While it might be necessary to write test functions, these are simpler to write than stubs, and sometimes not necessary at all.

ii. Pieces of programs written bottom-up tend to be more general, and thus more reusable than pieces of programs written top-down.

iii. Critical functions can be coded first to test their efficiency.

At the other end, bottom-up implementation also has its weaknesses, as described below:

i. Development maybe somewhat directionless because overall program organisation and higher level design is postponed.

ii. Developers may get too deep into the specific details of the different modules of the system and end up missing the deadline.

iii. The sub-modules of the system will still be eventually integrated.

6.3.1.3 Chosen Coding Methodology

After reviewing the two different approaches toward system implementation and comparing it against the characteristics of the system, it has been decided that the top-down approach would be used for the implementation of the system.

This decision is made partly due to the characteristics of the chosen programming language for development: VB.NET which advocates the usage of object oriented programming practice. Development in an object oriented environment practically envisions an abstract object at the top level before going down to the specifics.

Bottom-up coding convention is more suitable for function driven programming languages like C, C++ and Common Lisp.

6.3.2 Segregation of the Development Process

Basically, the system's implementation will be carried out in layered paradigms, which are as the following:

1. User Interface Layer
2. Data Layer
3. Transaction Layer

The benefits of carrying out the implementation process in a layered style are:

i. *Maintainability*

Code is organized in a recognized manner. Task oriented code is centrally located.

ii. *Reusability*

Task oriented code is easily developed for reuse, specifically for tasks that cross application boundaries.

iii. *Testability*

Modular testing can be carried out easily. Modularization breaks up the code average task into smaller, manageable units.

iv. *Speed*

Modular code can be optimized without affecting the calling procedures.

6.3.2.1 User Interface Layer

Basically the user interface layer is the portion of the system which enables the users to interact with the system minus all the unnecessary codes which works invisibly beneath the system. A good user interface should shield all the exceptions and bugs of the system by providing user-friendly error messages. Operations at this layer include the following:

i. Displaying all application data or information via a web interface.
ii. Responding to the changing states of the web interface.
iii. Initiating all user requests.

6.3.2.2 Data Layer

The data layer lies invisible from the eye of the user but actually performs a lot of operations related to data transactions. These include:

i. Formatting of data in the structure through validating data from input fields
ii. Formatting of data for display in the user interface layer
iii. Formatting of data for use in the transaction layer
iv. Sorting the data in arrays or structures
v. Sorting the data in arrays or structures.

6.3.2.3 Transaction Layer

The transaction layer uses the data layer as its application data repository and transfers or manipulates all data to and from an eternal data source. The transaction layer includes operations such as the following:

i. Initiating or building of all queries

ii. Handling violation of external database access rules.

iii. Controlling of all transactional logic

6.4 Coding Implementation

After determining the approaches used to code the system and segregating the implementation into three various layers, it is almost time to get started with the coding process. Before that, it is important to come up with a set of rules which should be abided by during the coding process, as described in the following sections.

6.4.1 Coding Convention

Coding convention is an important attribute of source codes and it determines the intelligibility of a program. The element of coding conventions include internal (source codes level) documentation, methods of data declarations and approaches to statement constructions.

In order to ensure system consistency, maintainability and readability, several programming conventions have been employed in the coding process. These programming conventions are as follows:

i. Use meaningful variable names, constant names, procedures names and parameter variable names to self document a program without excessive use of comments.

ii. Plan the layout of the program source code to improve its readability. Each sentence is started on a new line; statements following control structures are indented; white space is used to set off related blocks of codes, etc. These may seem insignificant but will help reduce the time needed to understand the program flow in the case where other programmers refer the code.

iii. All variables are declared at the beginning of procedures and declarations are separated from executables statements with a blank line to improve program readability.

iv. Insert comments to document the program and improve program understandability.

v. Group related types of codes together

6.4.2 Coding Documentation

Coding documentation begins with the selection of identifier names, continues with the composition of connecting all separate components and ends with the re-organization of the program. Blank lines and indentations are used so that comments can be readily distinguished from the source codes.

Elements taken to consideration in coding for an easy to maintain and enhanced system are internal documentation, standard naming conventions and standard graphical user interfaces.

Internal documentation is achieved using comments in coding, providing a clear guide to programmers for future enhancements. Statements of purpose indicating the functions of modules and descriptive comments are embedded into source codes to describe the processing functions.

Standard naming convention and also a standard usage of graphical user interface components is employed in developing the system. Standard naming conventions provides the programmers with easy identification of variables. A standard usage of graphical user interface components provides users an environment that will not generate much surprise to them.

6.4.3 Classifications of Program Code

For this system, program code is basically divided into a few sections depending on the functions performed:

i. *Database operations*

This includes populating ListView in VB.NET with values, displaying data from database and basic data operations such as creating, updating and deleting records.

ii. *Core functions*

For this system, its core functions include detecting the dates and time and getting user details and information.

6.4.3.1 Database Operations

(i) Importing the required Class

Operations related to database for this system utilized the ADO.NET data access technology. Therefore, all database related operations must have the following statement:

```
import System.Data.SqlClient;
```

(ii) Connecting to the Database

The database connection string.

```
Dim connStr As String = "Data Source = DPMS.sdf; Password
= 'keat'"
    Dim engine As New SqlCeEngine(connStr)
```

Therefore, any reference to the connection string is done like this:

```
Dim conn As New SqlCeConnection(connStr)
  If Not File.Exists("DPMS.sdf") Then
      engine.CreateDatabase()
      engine.Dispose()
      Try
        Dim cmd As SqlCeCommand = conn.CreateCommand()
        conn.Open()
        cmd.CommandText = "CREATE TABLE myApp (subject
nText, location nText, date DateTime, time nText,
reminder nText)"
        cmd.ExecuteNonQuery()
Catch
      Finally
        conn.Close()
      End Try
  End If
```

(iii) Database Operations

A standard database operation can utilize any of the ADO.NET components, including the Data Set, Data Adapter, Data Table, Data View, Data Column and Data Row. Below is a basic operation of populating a server control with data from the database:

```
Dim cmd As SqlCeCommand = conn.CreateCommand()
    conn.Open()
    cmd.CommandText = "SELECT * FROM myTask "
    Dim myReader As SqlCeDataReader =
cmd.ExecuteReader(CommandBehavior.CloseConnection)
While myReader.Read()
    ItemTmp = New
ListViewItem(myReader.Item("subject").ToString)

ItemTmp.SubItems.Add(FormatDateTime(myReader.Item("startDa
te"), DateFormat.ShortDate).ToString)
End While
```

6.4.3.2 Core Functions

The core functions of the system utilized traditional Windows API. Before it can be used, it needs to be initialized, as shown below:

```
Private Sub mmuExit_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles mmuExit.Click
```

6.5 Program Optimization

Certain coding conventions need to be taken into consideration when programming of the system takes place. This is needed to optimize the performance of the system and to make sure it is not bogged down by unnecessary and unwanted delays because of carelessness when coding the system. Any overlooked aspect can be the source of a system's underperformance in the operating environment.

6.5.1 Increase Execution Speed

The system should perform at its average speed (if not at its maximum speed). Several steps can be taken to ensure this:

i. *Avoid using variant data types*

This is due to the fact that variant data types require additional internal program standards to identify the information being stored. This is especially important when utilizing JavaScript in web applications.

ii. *Minimize the amount of program initialization*

The controls within the web application windows are forced to appear before the startup code is executed. This makes the user perceive that the program is running faster.

iii. *Use image control* whenever displaying a bitmap image in the program.

6.5.2 Decreasing Program Size

Downsizing the program volume can help in increasing execution speed and makes sure the system does not use up too much of the computer's resources. Steps are taken to make sure program size is smaller than needed:

i. Reviewing codes for unused variables, constants and "dead code" and removes it from the program code.

ii. Assigning the string variable to a zero length string if it is no longer needed.

6.6 Chapter Summary

This chapter discusses the techniques that guarantee a better implementation of the system and the actual development of the system itself. It is no less important compared to the other stages in the software development life cycle. The next chapter will be an elaboration on system testing and how the testing was carried out.

CHAPTER 7

~ Testing ~

Chapter 7 - Testing

7.1 Introduction

System testing is the testing of a complete system prior to delivery. The purpose of system testing is to identify defects that will only surface when a complete system is assembled. That is, defects that cannot be attributed to individual components or the interaction between two components. System testing includes testing of performance, security, configuration sensitivity, start-up and recovery from failure modes.

Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

7.2 Objectives of System Testing

Before we determine the types of testing that needs to be carried out on the system, we must first decide the objectives of carrying out the testing in the first place. This can help in pin-pointing the goals that the testing process needs to achieve. The objectives of the testing process have been determined as below:

i. *Software Reliability*

Making sure that the system performs critical tasks or core functions as pre-determined in the system design phase. Furthermore, specific user requests and business rules must be fulfilled.

ii. *Software Quality*

Software quality is characterized by the correctness of program logic and implementation. It begins with testing the software during development. Each module must be tested to make sure that it functions correctly at the time it is written or modified. Test values and boundary conditions must both be verified. Next, the module should undergo interface testing to check for functional errors.

iii. *System Assurance*

The main purpose of system assurance is to deliver a quality product. Conformance to requirements increases the organization's confidence in the system.

iv. *Optimum Performance and Capacity Utilization*

Another purpose of testing is to ensure optimum performance and capacity utilization of e-commerce system components. The purpose of stress or capacity testing/planning is to make sure that the system is able to perform acceptably at peak usage.

v. *Price of non-conformance*

The main purpose of testing is to detect errors and error-prone areas in a system. Testing must be thorough and well planned. A partially tested system is as bad as an untested system. And the price of an untested and under-tested system is high.

7.3 Approaches used in Testing of the System

As with the previous System Implementation phase, there are a lot of theories, methodologies and best practices associated with testing phase. A review is done on all the possible approaches that can be employed in the testing process before deciding on the most suitable style for the testing of the system.

7.3.1 Best Practices for the Testing Process

While testing a system, the steps should be followed whenever possible:

i. Prepare comprehensive test plan specifications and test cases for each level of testing. Supplement these with the test data and test logs. Test plans for system testing may involve operators and test plans because acceptance testing involves customers.

ii. Design the test cases to test system restrictions, such as file and database size (stress testing).

iii. Develop the data to test specific cases. Copies of live files must not be used except for Acceptance testing.

iv. Perform Regression testing on each component of the system. This ensures that no anomalies have crept into the system because of the changes made to the system.

v. Make sure to document and set up the test environment for each level in advance of testing. Test environments specify the preconditions required to perform the tests.

vi. Specify the intended test coverage as part of the test plan. Test coverage is the degree to which specific test cases address all specified requirements for a specific system or component.

7.3.2 Testing Techniques

The following section separates the suitable sections for different testing techniques:

White Box Testing	*Black Box Testing*
Complete Path Testing	Equivalence Partitioning
Branch or Decision	Boundary Value Analysis
Condition Testing	Cause Effect Graphing
Data Flow Testing	Syntax Testing
Loop Testing	

Table 7-1 Testing Techniques

7.3.3 Types of Testing

The testing process for the system can be divided similarly to the layered paradigm used during the system implementation process. It is described as following:

Type	*Description*
1) User Interface	- Content Testing - Browser Compatibility
2) Transaction Operations	- Performance Testing - Stress Testing
3) Data Operations	- Data Integrity - Data validity

Table 7-2 Layered Testing

Besides the types of testing carried in the layered approach above, specific tests that must be carried out across all the layers include:

i. Security

- In this system, security is an important aspect of the system because the data displayed in the application is confidential and sensitive. Therefore, testing should be carried out to block unauthorized access and attempts to change database content.

ii. Regression Testing

- Regression testing refers to retesting previously tested components/functionality of the system to ensure that they function properly even after a change has been made to parts of the system. As defects are discovered in a component, modifications should be made to correct them. This may require other components in the testing process to be retested.

iii. User Acceptance Testing

- Acceptance testing is performed on a collection of business functions in a production environment. This is the final stage in the testing process before the system is accepted for operational use. It involves testing the system with data supplied by the customer or the site visitor rather than the simulated data developed as part of the testing process.

7.3.3.1 User Interface Testing

User Interface is basically the part of the system which the user interacts with, thus it should be placed with the utmost importance as the other critical operations such as correctness of database transactions.

7.3.3.1.1 Content Testing

(i) Visual appeal

The visual appearance of a pocket pc application is important. Regardless of the developer's choice for color, font, or graphics, the tester needs to test for the appearance of the site and bring out problem areas. Tests required to check the visual appeal of a site are described below in the table below:

What to Test	*Environment*	*Tools / Techniques*
Fonts	User Environment	GUI Testing
Intensity of Colors	User Environment	GUI Testing
Graphics	Development Environment	GUI Testing
Grammar and Spelling	User/Development Environment	GUI/Proofreading/Spell check
Facts and Figures	User Environment	GUI Testing

Table 7-3 Layered Testing

(ii) Browse for Font Style

- There are a number a lot of choice for the font styles nowadays. A font style that can be clear and easy for the user to read will be a good choice of font.

(iii) Consistency of font size

- Test for consistency of font size throughout the pocket pc application. A body text font size of 10 to 14 and a heading font size of 18 to 24 are the norm.

(iii) Colors

- Consider the combinations of foreground and background colors throughout the application. For example, it may be difficult to read yellow text on a white background.

(iv) Graphics

- Fewer graphics on a pocket pc application aid in faster downloads. Developers must test for download time of graphics-intensive pages.

7.3.3.1.2 Overall Graphical User Interface Testing

The following test cases have been carried out on the system and the output is as expected:

Test Description	*Result*
Test each toolbar and menu item for navigation using the mouse and keyboard.	Passed
Test window navigation using the mouse and keyboard.	Passed
Test to make sure that proper format masks are used. For example, all drop-down boxes should be properly sorted. The date entry should also be properly formatted.	Passed
Test that the colors, fonts, and font widths are to standard for the field prompts and displayed text.	Passed
Make sure that vertical scroll bars or horizontal scroll bars do not appear unless required.	Passed

Test that the various controls on the window are aligned correctly.	Passed
Check the spellings of all the text displayed in the window, such as the window caption, status bar options, field prompts, pop-up text, and error messages.	Passed
Check for the display of defaults if there are any.	Passed
Check for the tab order. It should be from top left to bottom right. Also, the read-only/disabled fields should be avoided in the TAB sequence.	Passed
Check that the cursor is positioned on the first input field when the window is opened. Make sure if any default button is specified, it should work properly.	Passed
Validate the behavior of each control, such as push button, radio button, list box, and so on	Passed

Table 7-4 Overall User Interface Testing

7.3.3.2 Transaction Operations Testing

Testing on this layer deals with testing the correctness of the business logic, business rule and the performance of the system.

(i) Performance Testing

Software performance testing aims to ensure that the software performs in accordance to the operational specifications for response time, processing costs, storage use, and printed output.

All interfaces are fully tested. This includes verifying the facilities and equipment, and checking to make sure that the communication lines are performing satisfactorily.

What to Test	*Environment*	*Tools/Technique*
Correct data capture	Development environment	Testing ASP, CGI scripts by Black Box and White Box technique, Boundary value analysis, and Equivalence partitioning.
Transactions completion	User/System test environment	Functional testing by simulating user data.

Table 7-5 Software Performance Testing

(ii) *Stress Testing*

Running the system in a high-stress mode creates high demands on resources and stress tests the system. Some systems are designed to handle a specified volume of load. For example, a bank transaction processing system may be designed to process up to 100 transactions per second; an operating system may be designed to handle up to 200 separate terminals. Tests must be designed to ensure that the system can process expected load. This usually involves planning a series of tests where the load is gradually increased to reflect the expected usage pattern.

Stress tests steadily increase the load on the system beyond the maximum design load until the system fails. This type of testing has a dual function:

i. It tests the failure behavior of the system. Circumstances may arise through an unexpected combination of events where the load placed on the system exceeds the maximum anticipated load. Stress testing determines if overloading the system results in loss of data or user service.

ii. It stresses the system and may cause certain defects to come to light, which may not normally manifest the errors.

7.3.3.3 Data Operations Testing

Data operations testing should be carried down to the most painstaking detail because it contains the most sensitive operations of the entire system. For example, data saved wrongly could jeopardize the reputation of an employee. Table 7.8 shows the testing planned for data operations:

What to Test	*Environment*	*Tools/Technique*
Data integrity	Development environment	White Box Testing
Data validity	Development environment	White Box Testing

Table 7-6 Data Operations Testing

(i) Data Integrity Testing

Important data stored in the database include the contacts information, task and appointment information. Testing must verify the correctness of the stored data. Therefore, testing should be performed on a regular basis because data changes over time.

A test case for each module, which deals specifically with data integrity testing, has been specially formulated, as shown in Table 7-7, Table 7-8, Table 7-9 and table 7-10:

Test Description	*Input*	*Expected Output*	*Actual Output*
A new task is created	Task Details	A new record is inserted into the database. Notification given upon success.	As expected
An existing task is updated	Updated Task Details	Existing task is updated.	As expected
An existing task is deleted	Task Details	Existing task is deleted.	As expected
Date values are converted correctly	Date Field	If provided date field does not conform to the standard calendar, a default value is saved. Notification given for incorrect date fields.	As expected

Table 7-7 Task Module Test Case for Data Integrity

Test Description	*Input*	*Expected Output*	*Actual Output*
A new appointment is created	Appointment Details	A new appointment is inserted into the database.	As expected
An existing appointment is updated	Updated Appointment Details	Existing appointment is updated.	As expected
An existing Appointment is deleted	Appointment Details	Existing appointment is deleted.	As expected

Date values are converted correctly	Date Field	If provided date field does not conform to the standard calendar, a default value is saved.	As expected
Values are left blank	Tracking Details	Default value: “N/A” is saved into the database.	As expected
There is an existing Name field	Tracking Details	Name of contact will appear in the combo box and when it is clicked, the mobile phone number will be populated into the phone(m) text box.	As expected

Table 7-8 Appointment Module Test Case for Data Integrity

Test Description	*Input*	*Expected Output*	*Actual Output*
A new contact is created	Contact Details	A new contact is inserted into the database.	As expected
An existing contact is updated	Updated Contact Details	Existing contact is updated.	As expected
An existing Contact is deleted	Contact Details	Existing contact is deleted.	As expected

Table 7-9 Contacts Module Test Case for Data Integrity

Test Description	*Input*	*Expected Output*	*Actual Output*
An existing Task is shown in the Task table	Task Details	Existing task is shown.	As expected
An existing Appointment is shown in the Appointment table	Appointment Details	Existing appointment is shown.	As expected

Table 7-10 Overview Module Test Case for Data Integrity

(iii) Data Validity Testing

Errors caused due to incorrect data entry, called data validity errors, are probably the most common data-related errors. These errors are also the most difficult to detect in the system. These errors are typically caused when a large volume of data is entered in a short time frame. For example, $67 can be entered as $76 by mistake. The data entered is therefore invalid.

For this system, the most sensitive data is the date provided in the task module. The End Date in the task module must either be the same or a later date compared to the Start Date or else n error message will appear.

7.4 System Testing Considerations

i. *The Event List*

All the possible triggers must be exercised and the expected results compared with the actual results. Every function should be tested by one or more events in the event list.

ii. *Specific Scenarios*

The entire set of possible scenarios or user profiles should be specified for a given application. This can be done by defining meaningful user profiles and restrict the testing to them.

iii. *Testing Transactions*

A list of possible transactions, either extracted from the scenarios or from the event list, is tracked through the software system to ascertain that they function correctly from input to output.

iv. *Documentation Testing*

All examples used in user manual must be tested for correctness and for whether or not the manual give the exact answers users will obtain when they run the examples. All the functionality should be available through both parts of the manual and through effective, accurate index. Further, all accessible terminology in the manual should be understood by end-users.

7.5 Chapter Summary

System Testing is a critical phase in analyzing logical errors in the system and to test the system's liabilities. Several objectives of this chapter have been identified to achieve the testing goals. The test cases are not formulated by using the conventional way of arranging by different modules. A new approach is used by abiding to data integrity and data validity rules and these rules encapsulate all modules. With the completion of the testing phase, basically the system is ready to be deployed. The next chapter ends this report by rounding up the system's limitations, strengths, weaknesses and suggestions on further enhancements.

CHAPTER 8

~ System Evaluation and Conclusion ~

Chapter 8 – System Evaluation and Conclusion

8.1 Introduction

At the end of the System Development Life Cycle, after the product has been delivered and deployed, the system developer is left with one final task which is to evaluate the system, the processes involved to develop the system and most importantly lessons learnt from the entire lifetime of the system development.

This is important as it can help us identify best practices which have been applied and remember to use it again in the future. Besides that, we must also take note of the less than perfect events that happened and remember NOT to repeat the same mistake again.

Every system should go through the system evaluation phase, for the benefit of the system and also the system developer. It is only through experience that system developers can learn and enhance their skills. Therefore, a thorough evaluation should always be carried out at the very end of every system's life cycle.

Throughout this chapter, the system will be evaluated to reveal its strengths, weaknesses, limitations and possible future enhancements which may be added on to the existing system. Besides that, a brief explanation of the personal problems encountered during the development of the system will be given.

8.2 Problems Encountered and Solutions to Overcome Them

The development of the system was definitely not problem-free. With the time constraint impended on the system, it is important to overcome problems as soon as possible and with the best available solutions. The project schedule should be followed stringently amid problems to ensure the system will be delivered on time.

8.2.1 Difficulty in Choosing Development Tools

With the introduction of .NET technology, it was a blessing in some ways but a huge limitation in others. .NET provides unimaginable possibilities with its cross-language feature, managed code concept and powerful language capabilities. However, once you decide to plan the system on top of a .NET platform, there is no turning back.

There is no choice of development tools as currently there are only one .NET compliant development tools, and that is the Visual Studio .NET 2003. Whether you like it or not, you will be stuck with it for the entire lifetime of the system. There is no freedom in choosing the development tools you like, e.g. there are a myriad of development tools for the Java language: JCreator, Sun Suite or even as simple as the notepad.

As the problem with .NET is inevitable, I tried to work around the dilemma by doing extensive research on the Visual Studio.NET suite to make sure I fully understand the features embedded inside and make full use of it. If I can't fight the problem, I might as well try to work with it.

8.2.2 Difficulty in Choosing the Programming Language

After deciding on the .NET technology, I was faced with the problem of choosing the programming language for my system. In order to satisfy everybody and cater for a large audience of programmers, Visual Studio .NET 2003 encompasses a variety of programming languages, among them: Visual Basic.NET, C#.NET, Visual C++, ASP.NET and J#.

With limited knowledge on the respective programming languages, it was understandable that I was worried I will end up choosing the wrong language for my system. How different is Visual Basic.NET from C#.NET? Is there a clear cut difference between the two?

It was impossible to try out all the programming language and decide which is more suitable as there was not enough time. Therefore, I decided to seek advice from professional programmers who were proficient in all the languages. I did this by joining online forums on programming languages and asking the opinions from different people to avoid biased judgments.

As a result, I got more than enough help from the online forums. The users there have been extremely helpful and provided many valuable insights on the characteristics of each programming language.

8.2.3 Lack of Knowledge in the Chosen Programming Language

I did not have any experience before in the chosen programming language: VB.NET. Again, there was no sufficient time to learn everything entirely from scratch. Thus, I had to pick up pieces from time to time as I moved on through the development stages of the system. Due to lack of full understanding on the concept of the language (plus the fact that is a relatively new language), I encountered some unavoidable problems during the coding of the system.

Again, I turned to the Internet for the possible solutions to my problems. I limited myself to not spending more than three days on the same problem. If the same problem persists, I will look for alternative solutions. It is just not feasible to hang on to the same problem, even if I might eventually manage to solve it three weeks later. There was a deadline to consider.

8.2.4 Lack of Knowledge on the Chosen Technology

.NET Technology is still in its infancy. We have never been exposed to it formally in lectures and there is no special courses dedicated specially to it yet. Therefore, a lot of research has to be done on our own.

.NET is very famous for its own "managed library" where a huge collection of classes and methods have been created specially for the .NET technology. This means that all the .NET programming languages can utilize it and eventually any cross usage between the programming languages is supported also.

From the beginning, I had identified a few classes from the .NET managed library which are essential for the development of my system. However, as time passes by, I realized that the classes that have been chosen were unable fulfill all the needs of my system. Due to lack of understanding of the .NET framework, I was faced with the problem of having an incomplete system on my hands.

Fortunately, there was still the traditional API library to turn to. It was found to be compatible with the .NET framework and the .NET languages. In the end, the classes from the API library were used instead of the .NET managed library.

8.2.5 Lack of Knowledge on the Chosen Database Technology

For this system, Microsoft SQL Server Compact Edition (CE) 2.0 has been chosen as the database technology. This software is still in its infancy. I have never been exposed to this software and therefore a lot of time has been put in to do research on this software.

After my research, I've found that Microsoft SQL Server CE will be the best database software to be used together with VB.NET to develop the embedded system despite some of its restrictions. One of the main restrictions was that it does not support Join Tables which mean that I am not able to relate the tables as freely as Microsoft SQL Server. Thus, I had to spend even more time to find out alternative ways to relate the tables.

Fortunately, with the help of a friend I was able to find the alternative way to relate the tables and made the relationships work between my tables.

8.3 System Strengths

The system has a few characteristics which denote the strength of the system. There are as described below:

i. *Contact Name and Mobile number extract function*

- In the appointment module, users are able to extract contact names from the available combo box. When the name has been selected, the mobile number will be shown to allow the user to contact the particular person in case of any changes.

ii. *An integrated application for doctors to key in their personal information*

- DPMS integrates the Task module, Appointment module, Contacts module and the Overview module to allow easy access for doctors. Besides easy access

iii. *Overview of Tasks and Appointments*

- The Overview module provides a bird's eye view of the tasks and appointments which the doctor has in order to allow the doctor to plan their schedule with even more efficiency and without any collision with other appointments or tasks.

iv. *User Friendly Graphical Interface*

- The graphical user interface design is tailored to the average user capabilities and promises easiness of usage. Even a novice user will have no problem navigating through the web application.

v. *Reusability*

- Source codes are organized into reusable classes in an object oriented environment. Interface codes are separated from the logic processing clearly. This advocates for easy reuse of the different components of the system or while adding future enhancements to the source codes.

8.4 System Limitations

The finished system is not without its limitations. There are described as below:

i. *No Reminder function for Appointment*

- In the appointment module, there isn't a reminder (eg. sound or change in color for the appointment that is coming) to remind the doctors of their appointment.

ii. *Overview should be in the form of a calendar*

- The Overview module should be in the form of a calendar so that it will be easier for the doctors to view their tasks and appointments and will allow them to be able to recognize their busy and free dates much easier.

iii. *A help button should be included into the system*

- The presence of a Help button will be very useful as it will provide the user with guidance in case they are lost or do not know how to use the system.

8.5 Future Enhancements

There are a few suggestions for future enhancements to be made to the system to rectify its weakness. These suggestions are described as below:

i. *Reminder function to be added*

- The reminder function to be added into the Appointment and Task module so that doctors will be able to set reminders to remind them of their appointments and tasks before hand so that they can prepare for it and will not forget about it.

ii. *A shortcut button should be created on the Today screen*

- A shortcut button for the system should be created on the Today screen to allow easy access for the doctors. This will reduce the time taken for the doctors to search for the system in the pocket pc.

iii. *More features should be added into the system*

- The system should allow for more powerful features: ie. Money management to manage the doctor's expense or password management to keep the doctor's password for things like email, bank account and others.

8.6 Chapter Summary

The development of the system is finally completed. The system evaluation denotes the final stage of the system development life cycle. This project has given me many insights on the true nature of developing a system, from its initial stage of determining the user needs, targeted users, coming up with a design of the system and actually implementing the system itself. It has been a wonderful and unforgettable experience for me.

REFERENCE

REFERENCES

[1] Sommerwille, I. (1995). *Software Engineering*. 5th edition. Reading: Addison-Wesley Ltd.

[2] Stephen R. Schach. (2002). *Object-Oriented and Classical Software Engineering*. 6th edition. McGraw-Hill

[3] http://www.handango.com

[4] http://www.developerone.com

[5] http://www.microsoft.com

[6] http://www.java.com

[7] Difference between ASP and JSP. Available at: http://java.sun.com/products/jsp/jsp-asp.html

[8] Christopher T., Tim B. (2002) *eMbedded Visual Basic. Windows CE and Pocket PC Mobile Applications.* Sams Publishing

[9] http://www.codeproject.com

[10] http://www.gotdotnet.com

[11] http://msdn.microsoft.com

[12] Duncan M., Kent S. (2002). *Visual Basic.NET in 21 Days*. Sams Publishing

[13] http://www.dotnet247.com

APPENDIX
~User Manual~

Appendix – User Manual

1.1 Introduction

Doctor Personal Management System (DPMS) is a pocket pc application that will allow doctors to keep track of their task and appointment more efficiently. The objective of this user manual is to help the doctors to understand the usage of the features and also to guide them to get use to the application.

This manual will teach the doctors how to install the application into their pocket pc and also how to use it.

1.2 Installation and setup

Before running DPMS on the computer, the user must have the following software installed into the computer. The software needed are:

- Microsoft Windows XP Professional with Service Pack 2
- Microsoft SQL Server 2000 with Service Pack 3a
- Microsoft Visual Studio.NET 2003
- .NET Framework Version 1.1
- Microsoft SQL Server Compact Edition (CE) 2.0 with the latest update
- Internet Information Service (IIS)

1.2.1 Installing and Configuring SQL Server CE on the Server

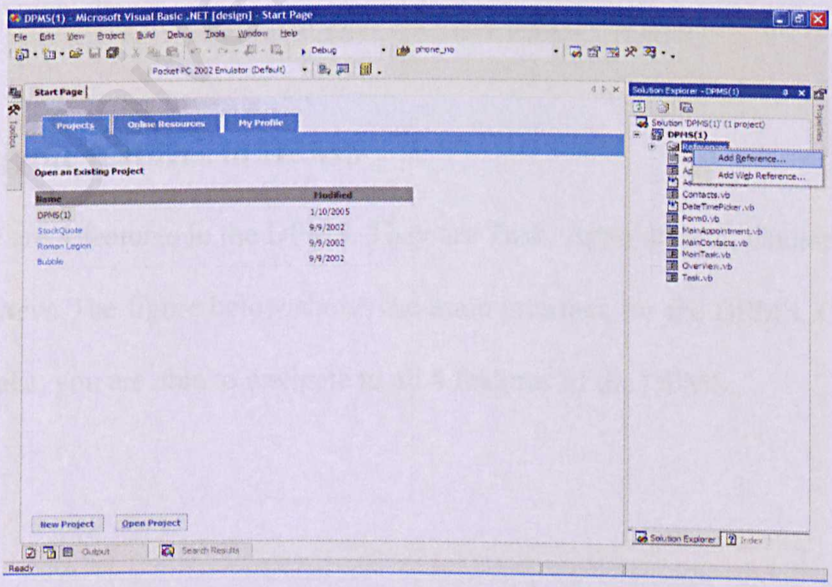
SQL Server CE uses Internet Information Services (IIS) to communicate with enterprise SQL Server. To install SQL Server CE on the server, use the SQL Server CE installation package which can be obtain from Microsoft's Web site at www.microsoft.com/sql/downloads. Once downloaded, run the installation on the server that will provide the SQL Server CE services. You will also need to run the installation on the development machine and choose the Development Tools option during installation.

When installation is complete, you need to register the two server-side DLLs, sscerp20.dll and sscesa20.dll by using the RegSvr32.exe. In a typical installation, these DLLs are installed to C:\Program Files\Microsoft SQL Server CE\Server, but can be moved or copied before registering. RegSvr32.exe is located in the System32 directory.

1.2.2 Installing and Configuring SQL Server CE on the Device or Emulator

Although there is no installation package to install SQL Server CE onto a Device or Emulator, you need to pay special attention when running SQL Server CE for the first time.

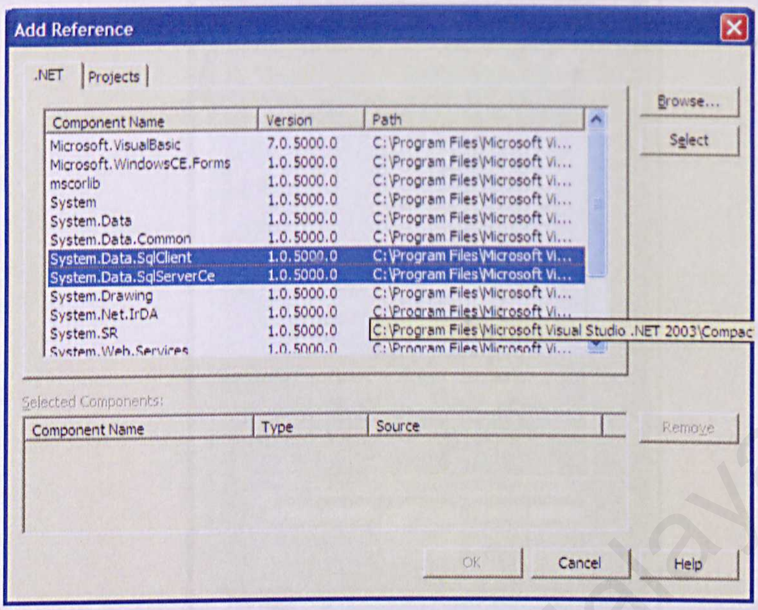
To install SQL Server CE on the Device or Emulator, right click on the Reference in the Solution Explorer and click on Add Reference as shown below.



Open an Existing Project

Name	Modified
DPMS(1)	1/10/2005
StockQuote	9/9/2002
Roman Legion	9/9/2002
Bubble	9/9/2002

In the Add Reference Window, select System.Data.SqlClient and System.Data.SqlServerCe. After selecting both of the references, click OK.



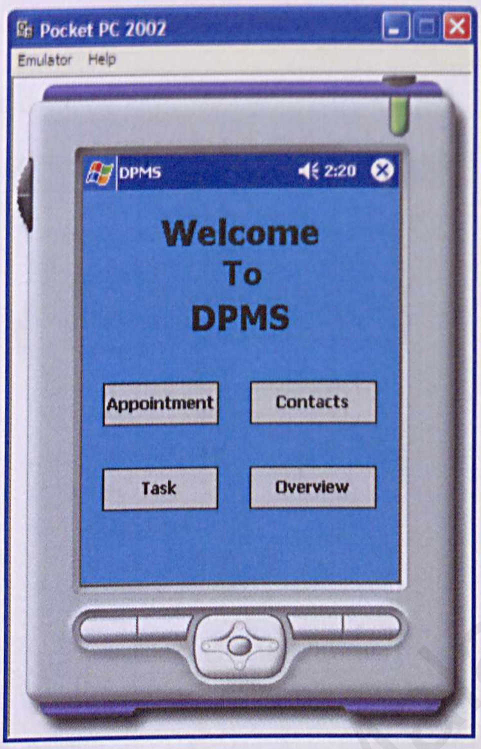
Component Name	Version	Path
Microsoft.VisualBasic	7.0.5000.0	C:\Program Files\Microsoft Vi...
Microsoft.WindowsCE.Forms	1.0.5000.0	C:\Program Files\Microsoft Vi...
mscorlib	1.0.5000.0	C:\Program Files\Microsoft Vi...
System	1.0.5000.0	C:\Program Files\Microsoft Vi...
System.Data	1.0.5000.0	C:\Program Files\Microsoft Vi...
System.Data.Common	1.0.5000.0	C:\Program Files\Microsoft Vi...
System.Data.SqlClient	1.0.5000.0	C:\Program Files\Microsoft Vi...
System.Data.SqlServerCe	1.0.5000.0	C:\Program Files\Microsoft Vi...
System.Drawing	1.0.5000.0	C:\Program Files\Microsoft Vi...
System.Net.IrDA	1.0.5000.0	C:\Program Files\Microsoft Vi...
System.SR	1.0.5000.0	C:\Program Files\Microsoft Visual Studio .NET 2003\Compac
System.Web.Services	1.0.5000.0	C:\Program Files\Microsoft Vi...

Component Name	Type	Source

After adding the references, you can choose your deployment device which is the Pocket PC 2002 Emulator which is the default device or a Pocket PC Device. After choosing the deployment device, you can click Start to run the application and SQL Server CE will automatically be installed into the deployment device.

1.3 Using the features in DPMS

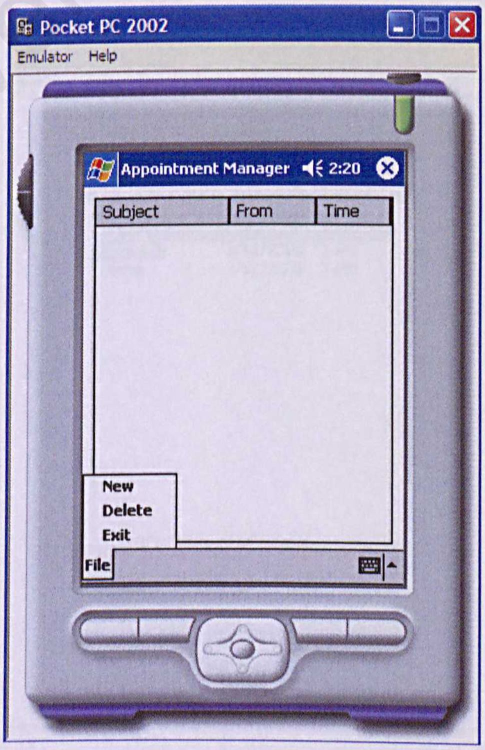
There are 4 features in the DPMS. They are Task, Appointment, Contacts and Overview. The figure below shows the main interface for the DPMS. On this interface, you are able to navigate to all 4 features of the DPMS.



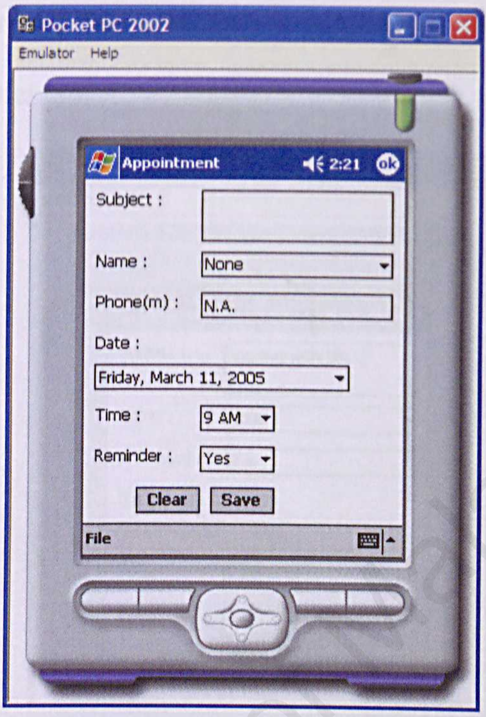
1.3.1 Using the Appointment

1.3.1.1 Creating a new Appointment

To create a new Appointment, click on the File icon and then click New.



You will enter into the New Appointment screen. Click Save when you have finished entering the data.



Subject :

Name : None

Phone(m) : N.A.

Date :

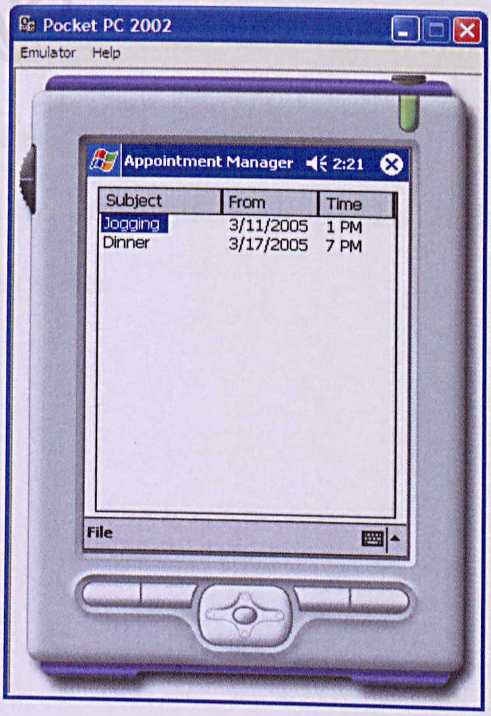
Friday, March 11, 2005

Time : 9 AM

Reminder : Yes

1.3.1.2 Deleting an Appointment

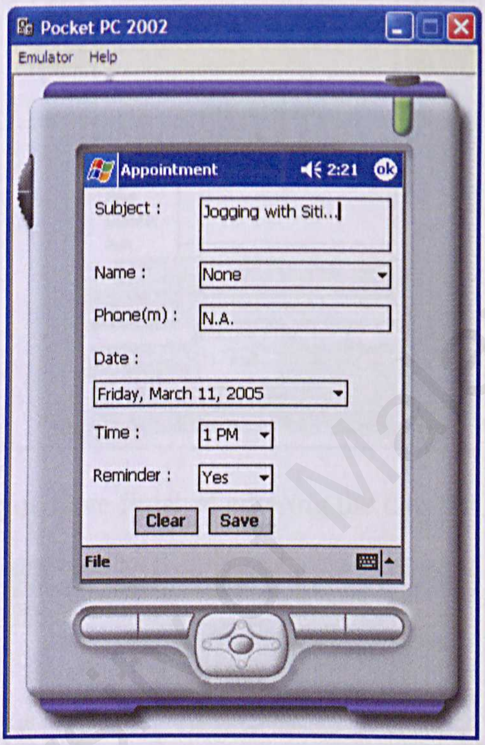
Click on the appropriate appointment and click File, then New and click Delete to delete an appointment.



Subject	From	Time
Jogging	3/11/2005	1 PM
Dinner	3/17/2005	7 PM

1.3.1.3 Updating an Appointment

Double click on the appropriate appointment and you will enter into the appointment screen where you can update your appointment data. Once you are done, click Save.



Subject : Jogging with Siti...

Name : None

Phone(m) : N.A.

Date : Friday, March 11, 2005

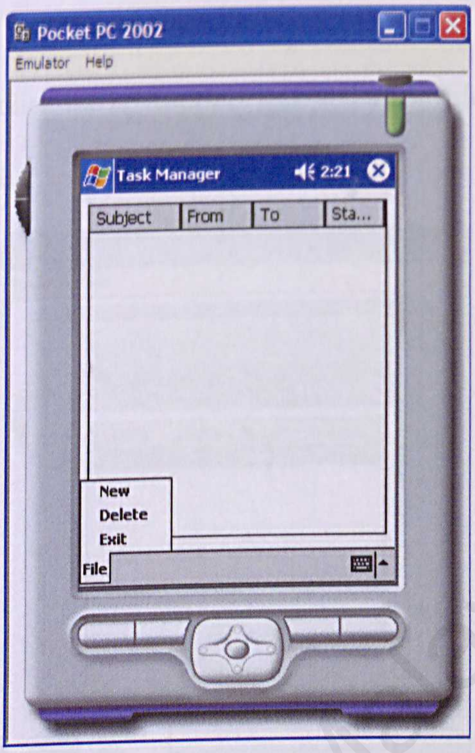
Time : 1 PM

Reminder : Yes

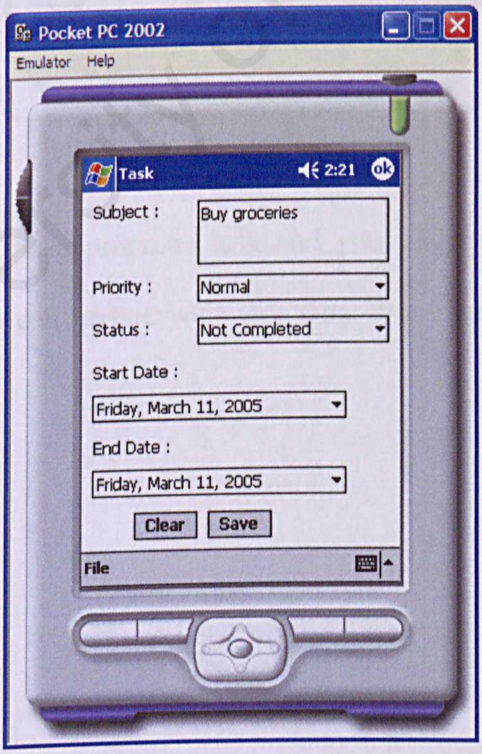
1.3.2 Using the Task

1.3.2.1 Creating a new Task

To create a new Task, click on the File icon and click New. You will enter the New Task screen.



Click Save when you have finished entering the data for the new task.



Subject : Buy groceries

Priority : Normal

Status : Not Completed

Start Date :

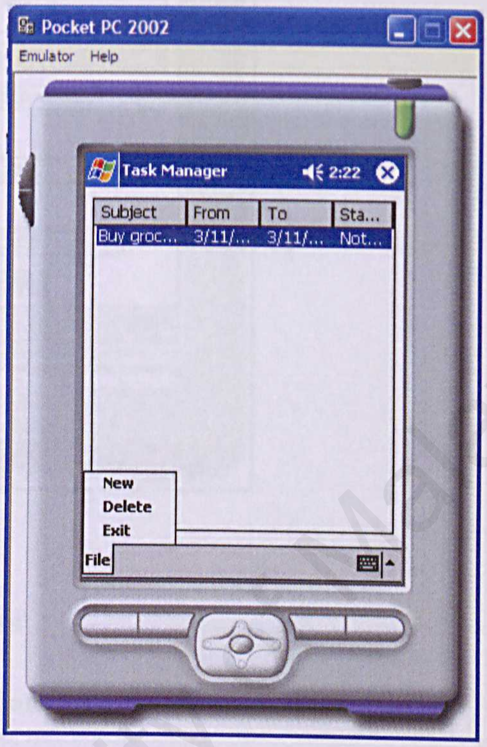
Friday, March 11, 2005

End Date :

Friday, March 11, 2005

1.3.2.2 Deleting a Task

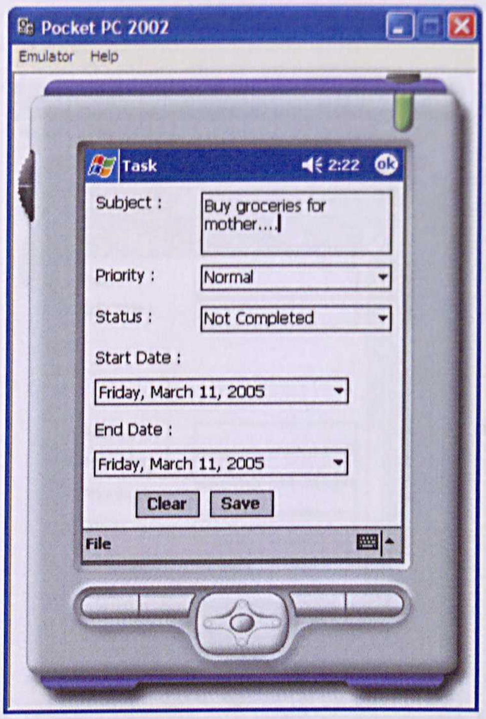
To delete a Task, click the appropriate task and click File then New followed by Delete.



Subject	From	To	Sta...
Buy groc...	3/11/...	3/11/...	Not...

1.3.2.3 Updating a Task

Double click on the appropriate task and you will enter into the task screen where you can update your task data. Once you are done, click Save.



Subject : Buy groceries for mother....

Priority : Normal

Status : Not Completed

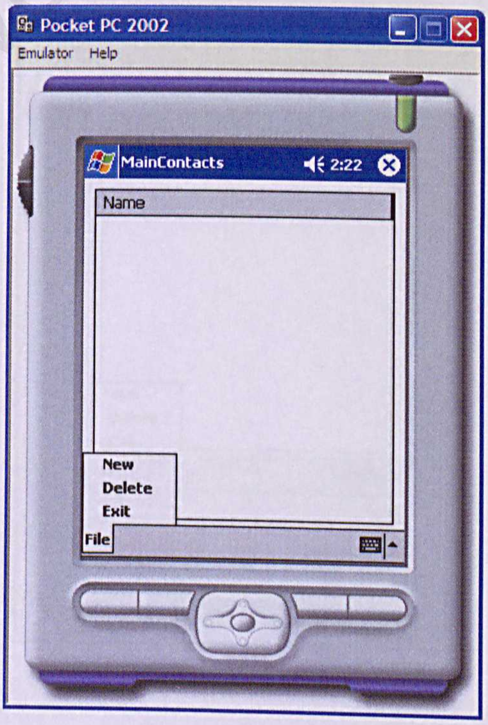
Start Date : Friday, March 11, 2005

End Date : Friday, March 11, 2005

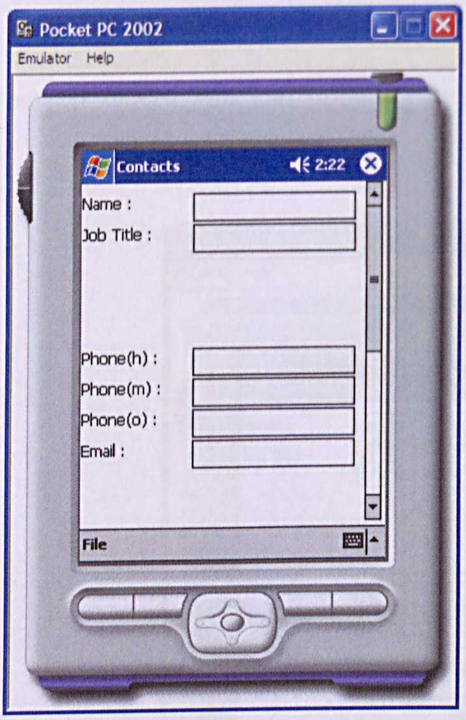
1.3.3 Using the Contacts

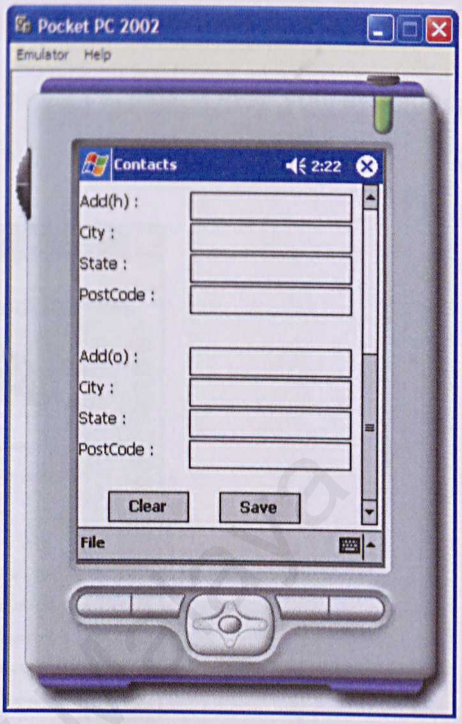
1.3.3.1 Creating a new Contact

To create a new Contact, click the File icon and click New and you will enter into the New Contact window.



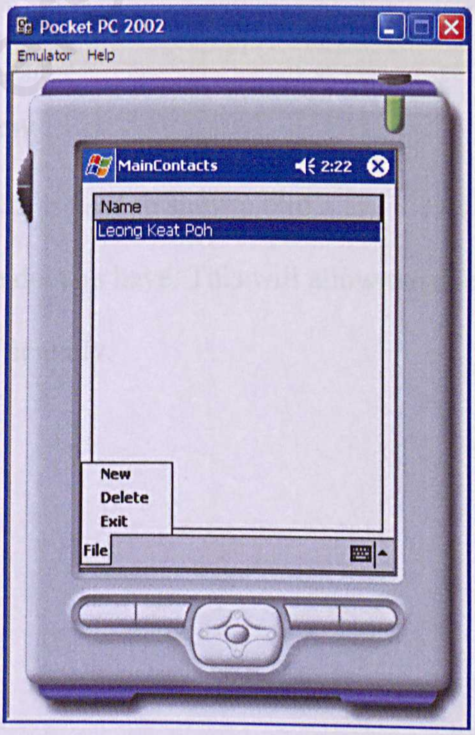
Click Save when you have finished entering all the new contact's data.





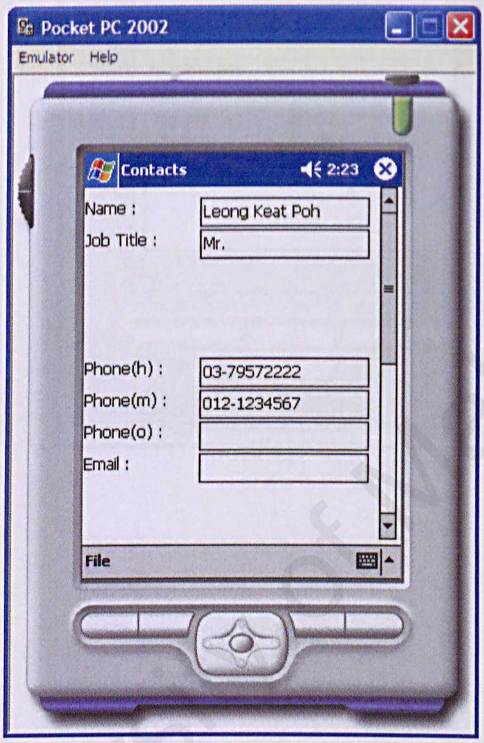
1.3.3.2 Deleting a Contact

To Delete a Contact, click on the name and click on File, followed by Delete.



1.3.3.3 Updating a Contact

Double click on the appropriate name and you will enter into the contact screen where you can update your contact's data. Once you are done, click Save.



Name : Leong Keat Poh

Job Title : Mr.

Phone(h) : 03-79572222

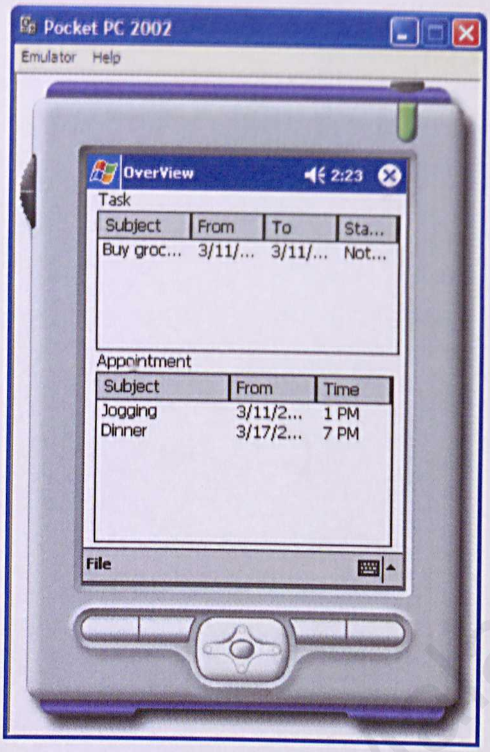
Phone(m) : 012-1234567

Phone(o) :

Email :

1.3.4 Using the Overview

The Overview in DPMS is used to show a bird's eye view on the Tasks and Appointments that the doctors have. This will allow the doctors to adjust their schedule easier and efficiently.



Task

Subject	From	To	Sta...
Buy groc...	3/11/...	3/11/...	Not...

Appointment

Subject	From	Time
Jogging	3/11/2...	1 PM
Dinner	3/17/2...	7 PM