

APPOINTMENT ASSISTANT SYSTEM

MOHD NOORHAFIZ B. MOHD FUAT

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL: APPOINTMENT ASSISTANT SYSTEM

SESI PENGAJAIN: 2 08/09

Saya Mohd NoorHafiz bin Mohd Fuat

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (/)

 SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

 TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

 / TIDAK TERHAD

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat tetap: 5 Jalan P9G1/1, Precint 9

62250 Putrajaya

Nama Penyelia

Tarikh: _____

Tarikh: _____

APPOINTMENT ASSISTANT SYSTEM

MOHD NOORHAFIZ B. MOHD FUAT

This report is submitted in partial fulfillment of the requirements for the Bachelor of
Computer Science (Software Engineering)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2009

DECLARATION

I hereby declare this project entitled

APPOINTMENT ASSISTANT SYSTEM

Is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : _____ DATE : _____

(MOHD NOORHAFIZ B. MOHD FUAT)

SUPERVISOR: _____ DATE : _____

(CIK SITI MASTURA BT. BAHARUDIN)

DEDICATION

I would like to dedicate this report to my beloved parents and all my family members as an appreciation for their tremendous support to me. And also to all my friends who have helped me whether directly or indirectly in the progress of making this report. Not to forget, I also would like to dedicate this report for my PSM Supervisor and PSM Evaluator.

ACKNOWLEDGEMENTS

I would like to express my gratitude especially to my beloved family who always give me tremendous moral and financial support to me. Without their support and blessings, I'm not sure how am I going to finish and complete this report and project.

And a very special appreciation for Miss Siti Mastura bt Baharudin who is also my supervisor for her help, co-operation and also guidance during my Projek Sarjana Muda period. Without her hands on me, I am very sure the progress of this report would not be in smooth sailing.

I also would love to thank Dr.Choo Yun Huoy who is my PSM Evaluator as well in terms of facilitating me in order to complete this final report and regarding my project.

Finally to all my friends who have involved and helped me directly or indirectly with my PSM to achieve the requirements as an UTeM student as well, I would like to thank you all.

And for those who have not been mentioned above, I would love to put all of you in a pedestal of my heart.

ABSTRACT

The project that I built is going to run on a hand held device such as mobile phone or PDA desktop/laptop application. The application is called Appointment Assistant System. The application helps user to store, edit and delete their appointments. It can be said the application works pretty much like an organizer where the user can manage their schedule. There are a lot of similar systems available out there but most of them are web based system. Microsoft Windows also provide this system which they called Task Scheduler or Notes on windows mobile. In chapter one, introduction about the project is explained. Also stated in this chapter are the problem statement of the project, objective of the project, scope of the project, project significance, expected output and lastly the conclusion of this chapter. In chapter two, it discuss about the methodology and technique that applied in this project. A good planning will often lead to a good ending of a project. Thus, the planning was presented in this chapter. Some more, a schedule is created in this chapter in order to have a good time management so that system can be done in time. A milestone allows project management to much more accurately determine whether or not the project is on schedule. In chapter three, the system flow and the weaknesses of existing system are analyzed and the suggestion of new function to improve the system is recommended. The functional requirement, non-functional requirement and data requirement of new system was discussed in details. In chapter four, systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. This chapter defines and refines the design of the system. The designs of the system include system

architecture, user interface design, navigation design, input and output design and also the database design. The design stage is followed by implementation stage. In chapter five, implementation processes of the project are discussed. Also discussed in this chapter are about versioning control and the software configuration and environment setup. In chapter six, testing strategy process are carried out. This is to locate error of the system and then fix the error to ensure the system meet user's requirements and produce good quality product. The main purpose of software testing is to provide a systematic approach toward testing aspects of executing testing and monitoring results for testing. This chapter provides a detailed testing strategy and to document the procedures that will be performed by the test team. The testing strategy that will be used for this system is glass-box testing which is carried out by a programmer. The last chapter is chapter 7 which is concludes the conclusion of overall project. In this chapter, observation of strength and weaknesses of the system are observed and proposition of improvement are suggested for future improvement.

TABLE OF CONTENTS

CHAPTER	SUBJECT	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENTS	iv
	ABSTRACT	v
	TABLE OF CONTENTS	vii
	LIST OF FIGURES	x
	LIST OF TABLES	xii
CHAPTER 1	INTRODUCTION	
	1.1 Project Background	1
	1.2 Problem Statement	2
	1.3 Objective	2
	1.4 Scope	3
	1.5 Project Significance	3
	1.6 Expected Output	4
	1.7 Conclusion	4
CHAPTER 2	LITERATURE REVIEW AND PROJECT METHODOLOGY	5
	2.1 Introduction	
	2.2 Facts and Findings	10
	2.2.1 Domain	10

2.2.2 Existing System	10
2.2.3 Techniques	13
2.3 Project Methodology	13
2.4 Project Requirements	15
2.4.1 Software Requirements	15
2.4.2 Hardware Requirements	15
2.5 Project Schedules and Milestones	16
2.6 Conclusion	17
CHAPTER 3 ANALYSIS	
3.1 Introduction	18
3.2 Problem Analysis	19
3.3 Requirement Analysis	
3.3.1 Data Requirement	24
3.3.2 Functional Requirement	25
3.3.3 Non-Functional Requirement	
3.3.3.1 Performance and Throughput	28
3.3.3.2 Integrity	28
3.3.3.3 Reliability & Availability	29
3.3.3.4 Security	29
3.3.3.5 Manageability	29
3.3.4 Others Requirement	30
3.4 Conclusion	31
CHAPTER 4 DESIGN	
4.1 Introduction	32
4.2 High Level Design	
4.2.1 System Architecture	33
4.2.2 System Interface	
4.2.2.1 User Interface	35
4.4.4.2 Navigation Design	39
4.4.4.3 Input Design	40

4.2.3 Database Design	
4.2.3.1 Conceptual and Logical Database Design	41
4.2.4 Data Dictionary	42
4.3 Detailed Design	
4.3.1 Software Design	43
4.4 Conclusion	53
CHAPTER 5 IMPLEMENTATION	
5.1 Introduction	54
5.2 Software Development Environment Setup	56
5.3 Software Configuration Management	56
5.3.1 Configuration Environment Setup	57
5.3.2 Version Control Procedure	58
5.4 Implementation Status	59
5.5 Conclusion	60
CHAPTER 6 TESTING	
6.1 Introduction	61
6.2 Test plan	62
6.2.1 Test Organization	62
6.2.2 Test Environment	62
6.2.3 Test Schedule	63
6.3 Test Strategies	64
6.3.1 Classes of test	64
6.4 Test Design	65
6.4.1 Test Description	65

6.4.2 Test Data	75
6.5 Test Results And Analysis	76
6.6 Conclusion	81
CHAPTER 7 Project Conclusion	
7.1 Observation On Weakness And Strengths	82
7.2 Propositions For Improvement	83
7.3 Contributions	83
7.4 Conclusion	84

LIST OF FIGURES

DIAGRAM	TITLE	PAGE
2.1	A stand – alone application architecture	6
2.2	A comparison of mobile application language	7
2.3	A mobile application architecture	8
2.4	An example mobile application architecture	8
2.5	Mobile Organizer by Symbian.Org	11
2.6	Windows Mobile Organizer	11
2.7	Rational Unified Process	14
2.8	Gant Chart	16
3.1	Business Flow Diagram of Existing System	19
3.2	Task Sequence Diagram	20
3.3	Event Sequence Diagram	21
3.4	Address Book Sequence Diagram	22
3.5	Notes Sequence Diagram	23
3.6	AAS Entity Relationship Diagram	24
3.7	AAS Use Case Diagram	27
4.1	System architecture	34
4.2	Welcome screen	35
4.3	New password screen	35
4.4	Change Password screen	36
4.5	View Appointment screen	36
4.6	Add Appointment screen	37
4.7	Update and Delete Appointment screen	37
4.8	Login screen	38
4.9	Help screen	38

4.10	Navigation design	39
4.11	Entity Relationship Diagram of Appointment Assistant System	41
4.12	Login class Diagram	43
4.13	Change password class diagram	43
4.14	Add new information class diagram	44
4.15	Help class diagram	44
5.1	Deployment Diagram of Appointment Assistant System	56
5.2	Subversion tool's concept	58
6.1	Test Organization for AAS	62
6.2	Path Coverage Technique for Login	65
6.3	Path Coverage Technique for Add Appointment	66
6.4	Path Coverage Technique for Edit Appointment	67
6.5	Path Coverage Technique for Delete Appointment	68
6.6	Path Coverage Technique for Change Password	69
6.7	Path Coverage Technique for View Appointment	70
6.8	Path Coverage Technique for Logout	71
6.9	Path Coverage Technique for Reminder	72
6.10	Path Coverage Technique for Help	73
6.11	Path Coverage Technique for Forgot Password	74

LIST OF TABLES

TABLE	TITLE	PAGE
2.1	Comparison between three mobile appointment application	12
3.1	Data Dictionary for AAS	24
3.2	AAS Functional Requirement	25
3.3	Performance and Throughput Systemic Qualities	28
3.4	Integrity Systemic Qualities	28
3.5	Reliability and Availability Systemic Qualities	29
3.6	Security Systemic Qualities	29
3.7	Manageability Systemic Qualities	29
4.1	Input Design	40
4.2	Data dictionary of Appointment Assistant System	42
5.1	Implementation Status of AAS	59
6.1	Test Schedule for AAS	63
6.2	Unit Testing for Login	66
6.3	Unit Testing for Add Appointment	67
6.4	Unit Testing for Edit Appointment	68
6.5	Unit Testing for Delete Appointment	69

6.6	Unit Testing for Change Password	70
6.7	Unit Testing for View Appointment	71
6.8	Unit Testing for Logout	71
6.9	Unit Testing for Reminder	72
6.10	Unit Testing for Help	73
6.11	Unit Testing for Forgot Password	74
6.12	Test Data for AAS	75
6.13	User Acceptance Test for Login Module	76
6.14	User Acceptance Test for Add Appointment Module	76
6.15	User Acceptance Test for Edit Appointment Module	77
6.16	User Acceptance Test for Change Password Module	77
6.17	User Acceptance Test for View Appointment Module	78
6.18	User Acceptance Test for Logout Module	78
6.19	User Acceptance Test for Reminder Module	79
6.20	User Acceptance Test for Delete Appointment Module	79
6.21	User Acceptance Test for Help Module	80
6.22	User Acceptance Test for Forgot Password Module	80

ATTACHMENT

ATTACHMENT	TITLE	PAGE
I	LOG BOOK	85
II	PROPOSAL FORM	90
III	USER MANUAL	96
IV	REFERENCES	101

Chapter I

INTRODUCTION

1.1 Project Background

The project that I am going to build is going to run on a hand held device such as mobile phone or PDA desktop/laptop application. There are a lot of similar systems available out there but most of them are web based system. Microsoft Windows also provide this system which they called Task Scheduler or Notes on windows mobile.

The system is very useful for someone who is always busy and interact or dealing with a lot of people. The old conventional way for managing appointment is by taking down a note in a piece of paper or just remembering it inside head. But we are human and our human brain cannot remember as much as a computer can. Human tends to forget information they stored inside brain in a while if the information is beyond the human brain capabilities.

If user write down the appointment on a piece of paper, they often misplaced it and most important thing is they might missed the appointment because nobody going to remind them.

So the system that I proposed is intended to overcome all those stated problems and provide the security and ease of use functionality for user.

1.2 Problem Statement

There are some problems faced by users if they use the old conventional way such as:

- Usage of paper thus more wastage
- User might misplaced or lost it
- Not able to remind
- User still have to memorize everything

1.3 Objective

The objectives of this system are :-

- Provide reminding function so user would not miss their appointment
- Develop a very easy to use system although for non IT literate person
- Eliminate the use of paper
- Connectionless system. No internet connection needed.
- Cross platform. Can be executed both on desktop or pda

1.4 Scope

The target user for this system is basically students regardless they are secondary students or university/college students. Normally students are very busy with their own task. They are given homework or assignment to do and submit at certain times. They also might have to attend some classes or courses or meeting someone. This system will help them remembering those entire datelines.

1.5 Project Significance

The significance of this project is students who use this system can gain full advantage of the system. They can also save costs because there is no need to use paper to take notes about appointment or dateline to meet. They will never late of miss the dateline again with the help of this system. Because the system will be build on mobile device, a student with a windows based mobile phone can access it anytime anywhere as long as they have the application inside their device thus making their life easier and more efficient.

1.6 Expected Output

The system is expected to fully function meaning that user can add an appointment, delete the appointment or edit it and the system is able to remind them when the dateline is on time. The system is also expected to run cross platform, whether in desktop/laptop or PDA. And last but not least this system is expected can be fully developed and 100% working and also to meet all the objective requirements.

1.7 Conclusion

As for the conclusion, the system will help students to manage their appointment or dateline for assignment. Besides, they can take advantage of using this system and coping with this modern IT lifestyle. In addition, students also can save their money and time by using this system. And last but not least, the methodologies of the project will be discussed in next chapter.

CHAPTER II

LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

In this chapter, the literature review of the system and the methodologies which are using in developing the system will be discussed. This chapter includes the current relevant research and also delivers summaries of previous research or work regarding on the same particular research to this project case study. In order to support the finding, some existing system will be analyzed to get more understanding of the new project which will be delivered later.

First of all, stand alone application/desktop application will be discussed as the system will implement this concept later. A standalone is an entity that has no dependencies to anything. It can stand alone by itself. A standalone program is any program that is run in a standalone PC.

It may be an application program or any system program. However this distinction does not really true, since the computer usually need to be running some program that prepares (and sometimes initially processes) the stand-alone program to begin with (e.g. .NET Framework, Java Runtime Environment etc). Hence all programs are launched or prepared/processed by other programs (with the sole exception of the bootstrap loader), and no code can really be said to stand alone.

If stand-alone is more widely defined as a program not needing the services of other programs once it is running, then most operating systems can also justifiably be called stand-alone, since they need the bootstrap loader only for starting up. In some embedded systems, the one and only application that they run is truly stand-alone, in the sense that it includes all the code that's ever going to be seen by the CPU of those systems. This kind of application has to contain its own operating system.

A distinction might be drawn between programs that run as operating system processes and those that are loaded as an add-on in the context of an existing process for example a plug-in. A less ambiguous terminology refers to application and non-application code, since application seems to be a commonly-accepted cross-platform term for a program that runs as a separate process and is invoked as a direct result of some user action. The non-application code then has to be loaded and run as part of the process context of the application code.

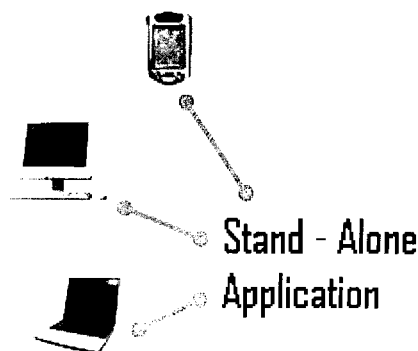


Figure 2.1: A stand – alone application architecture

The second method that will be used to develop the system is mobile application method. Mobile platform development often described as applications for handheld devices such as personal digital assistants (PDA), enterprise digital assistants or mobile phones. Mobile applications can be written in many languages such as Java, Microsoft Visual Basic.Net, SymbianOS and etc. Below are the comparisons of languages that can be used in writing a mobile application.

Overview	
Java ME	Ideal for a portable solution, if the Java ME platform provides the needed functionality. Good for vertical applications that must be portable. Device-specific libraries exist for many devices and are commonly used for games, making them non-portable.
Symbian	Very powerful for general purpose development. The Symbian based S60 platform is strongly supported by Nokia with some support from other device manufacturers. In Japan NTT DoCoMo's Symbian based MOAP platform is also well supported by a number of manufacturers (Fujitsu, Sony Ericsson Japan, Mitsubishi and Sharp amongst others). It should be noted, however, that MOAP is not an open development platform. Another Symbian based platform, UIQ, is less well supported (principally by Sony Ericsson and Motorola). Currently large device deployments in Europe and Japan, with little penetration in the US market.
Android	Recently announced by the Open Handset Alliance, whose 34 members include Google, HTC, Motorola, Qualcomm, and T-Mobile, Android is a new Linux-based platform currently available only as a developer pre-release. Although it does not yet have any fielded implementations, its support by 34 major software, hardware and telecoms companies makes it likely that it will be rapidly adopted from 2008. The Linux kernel is used as a hardware abstraction layer (HAL). Application programming is exclusively done in Java. You need the Android specific Java SDK. Besides the Android Java Libraries it is possible to use normal Java IDEs.
iPhone	The iPhone and iPod Touch SDK uses Objective C, based on the C programming language. Currently, is only available on Mac OS X 10.5 and is the only way to write an iPhone application. All applications must be cleared by Apple before being hosted on the AppStore, the sole distribution channel for iPhone and iPod touch applications.
C/C++	Ideal for prototyping and quickly developing database powered applications. Also useful for porting Object Pascal software to mobiles. Can access the native APIs when translated headers are available.
Python	Ideal for initial prototyping and concept testing when functionality falls outside Java ME
NET Compact Framework	Ideal for deployment on homogeneous Pocket-PC devices. However not cross platform and limited to Microsoft Windows Mobile devices.
BREW	Ideal for deploying applications for deployment on CDMA-based networks (also supports GPRS/GSM models) with a deployed Brew Content Platform especially if OTA app deployment is desired. Little penetration in Europe.
Pocket PC and Microsoft Smartphone	Ideal for enterprise applications with an existing PC infrastructure and options for significant development investment. However not cross platform and limited to Microsoft devices.
Palm OS	Significant player with strong enterprise following in the important US market

Figure 2.2: A comparison of mobile application language. From “www.wikipedia.org/mobileapplicationdevelopment.html”

Mobile application is often defined as an application which can be operate on mobile devices and often is also independent. There are typically three major components to a mobile architecture:-

- An existing system
- A middleware application
- A handheld application

The reason a middleware application is usually needed is to provide data transformation, apply business logic, and be a central point of communication for the devices. If a new system is being developed or rewritten then no middleware may be