

CERTIFICATION OF APPROVAL

Final Year Project Proposal Management System

By

Saiful Fakhri Bin Ahmad Zaim (3429)

*A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION TECHNOLOGY)*

Approved by,



(MR. HILMI HASAN)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

JANUARY 2006

t

GA

76.625

.5132

2006

1) Internet programming
2) Microsoft .NET Framework

CERTIFICATION OF ORIGINALITY

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



SAIFUL FAKHRI BIN AHMAD ZAIM

ABSTRACT

This report presents a proposed project for Information Technology Final Year Project (FYP) which will be carried out for two semesters. This report highlights objectives, problem statement, scope of study and methodology of the project. The project has three objectives which are *to develop a web based Final Year Project Proposal Management System, to improve information and communication system in FYP community and to perform a research and development of intelligent software agent for web based application.* The idea of the project derived from five problem statements which are *no formal and integrated management system for FYP proposal, lack of information and communication in FYP community, slow process of proposal approval and assignment of supervisor, usability of automated system to assist human work and the needs of features to manage FYP efficiently.* The scope of study for this project is the full development of web based system for FYP proposal management and research and development in intelligent software agent. The project used an incremental software development process as a methodology and required common web based development tools.

ACKNOWLEDGEMENT

First and foremost, thank God for this opportunity and seeing me through some truly difficult times especially through the journey of this Final Year Project. This project would not have been possible without the vital help of University of Technology Petronas (UTP) lecturers, students and those who involve either directly or indirectly for the conceptual contribution in the research and development of this project. First of all, I would like to express my deepest gratitude and appreciation to all parties who have contributed towards the success of my Final Year Project especially to my supportive supervisor, Mr. Hilmi Hasan towards his idea, guidance, moral support and teaching. Not to forget to my former supervisor, Mr. Shuib Basri for assisting me on my Final Year Project Part 1, the earliest stage of the project. Thank you to my parents; family that always has faith in me and supporting me in achieving my dreams. Without all of these parties, it would have been impossible for me to finish this dissertation. I appreciate very much the companion and help I got from past and present graduate students, researchers and UTP staffs that I had privilege to meet and work with. And last but not least, to my fellow colleagues for their companionships, helps, concerns and supports. It is a blessing to know these wonderful people.

TABLE OF CONTENT

CERTIFICATION		i
ABSTRACT		iii
ACKNOWLEDGEMENT		iv
LIST OF FIGURES		v
CHAPTER 1:	INTRODUCTION	6
	1.1 Background of Study	6
	1.2 Problem Statement	8
	1.3 Objective and Scope of Study	10
CHAPTER 2:	LITERATURE REVIEW	11
CHAPTER 3:	METHODOLOGY	13
	3.1 Process Identification	13
	3.2 Tools Required	18
CHAPTER 4:	RESULTS AND DISCUSSION	19
	4.1 Core Modules	19
	4.2 Additional Modules	30
CHAPTER 5:	CONCLUSION & RECOMMENDATION	31
REFERENCES		33
APPENDICES		34

LIST OF FIGURES

- Figure 3.1** Incremental development model
- Figure 3.2** Requirement identification phases
- Figure 3.3** System Architecture: General Overview
- Figure 3.4** System Architecture: Subsystem: Process Automation
- Figure 3.5** System Architecture: Subsystem: Information Based
- Figure 3.6** Testing Plan
- Figure 4.1** LOGIN module
- Figure 4.2** MAIN module: STUDENT
- Figure 4.3** HEADER menu
- Figure 4.4** MESSAGES box
- Figure 4.5** Status Information box
- Figure 4.6** PROPOSAL module: STUDENT
- Figure 4.7** REPORT module: STUDENT
- Figure 4.8** MESSAGES module: STUDENT
- Figure 4.9** MESSAGES module: Compose: STUDENT
- Figure 4.10** ACCOUNT SETTING module: STUDENT
- Figure 4.11** SCHEDULE module: ALL

ABBREVIATIONS AND NOMENCLATURES

FYP – Final Year Project

FYPMS – Final Year Project Proposal Management System

ASP – Active Server Pages

SQL – Sequel Query Language

IIS – Internet Information Services

SDLC – Software Development Lifecycle

DBMS – Database Management System

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Final Year Project (FYP) is a compulsory subject for all final year UTP students which is a design or research-based subject. It requires students to do research, design and/or development work in each discipline, especially on real-world problems. Student will work under the guidance of supervisor which has been assigned by the FYP coordinator or has been selected by the student. Basically a team comprising of FYP Committee, Coordinator, Supervisor, Advisor, Examiner and Students is formed for managing the final year project paper. Among the processes in FYP are:

- Submission of Titles and Project Synopsis
- Approval on Project Title, Synopsis and Assignment of Supervisor
- Selection and Prioritization of Project Titles
- Allocation of Approved Project Title
- Submission of Preliminary Report
- Approval for Purchase and Usage of Resources and Services
- Submission of Progress Report
- Submission of Interim/Final Draft

The full process flow of the project will be shown in the Appendix since in this report only a few processes will be highlighted as the related issues to be discussed.

In submission of titles and project synopsis, the students will propose their project title with a brief discussion about the proposal in written format to any lecturer

that usually related to area of research for the specific project. The selection of lecturer is made by the student which he or she is going to supervise the student throughout the completion of the project. One lecturer is subjected to supervise maximum of ten students concurrently. Thus, if the quota is fully filled, the student should propose the project to other available lecturers with the same area of specialization. Student may submit the proposal up to three different proposals. The next process is approval on project title, synopsis and assignment of supervisor. In this part, lecturers will discuss all proposals received with FYP Committee and they will decide whether to approve the proposals or not. During this time, the FYP committee confirms the assignment of supervisor to a group of students within the related areas. Once the project title and supervisor are confirmed, the students can start their planning for the specific project and schedule a meeting with supervisor.

The student is required to submit a preliminary report as an initial proposal for the project and the supervisor will assess the report and submit the marks to the FYP Coordinator. The progress of project will be observed by the supervisor and student is required to submit the progress in a log book format. If the supervisor wants to request on behalf of the student for purchase of consumables and equipment, technical support, related resources and visit for data collection, he or she is required to forward the request to FYP Coordinator and FYP coordinator will forward it to FYP Chairman for approval.

The student is required to submit a progress report with the guidance of supervisor. The supervisor will assess the report and submit the marks to FYP Coordinator. The rest of the processes are the submission of Interim Report, oral presentation, submission of bound copy of project dissertation and grading for the project which considered as final phases in FYP.

In this project, a web based system will be developed to form a system that can handle all processes mentioned. Generally, the system will consist of four sections

which are proposal section, data section, marks section and archive section. The proposal section will focus on proposal submission by student and classification of proposal in term of area of research. The data section will focus on assigning supervisor, grouping of students and supervisor, project timeline, meeting scheduling, progress tracking and submission of reports. The marks section will rule out the procedure of giving marks to the reports and for the whole project. The archive section will focus on archiving the currently completed projects and resources related to all past projects.

The research area in this project is the development of intelligent software agent. The system will be using an intelligent software agent to assist data processing and handling. It will be developed to work in web environment system and manage to process information precisely. The software agent is very important in this system to support decision making such as determining areas of research in proposal, assigning the specific lecturer based on proposal, grouping the student in specific research area, sorting mechanism and lots more. The software agent will be responsible for matchmaking and brokering the proposal at first and from there it will determine the other things like grouping and assigning to lecturer for supervision. The system is expected to work in intranet environment with the accessibility within the campus.

1.2 Problem Statement

Basically, there are a few problems initially that led to the idea of developing FYP Proposal management system. Here, the issues will be discussed one by one:

1. No formal and integrated management system for FYP

The current situation of the FYP proposal management is that it does not have any automation system to support all the processes in one single system. Basically, the

current system is more to manual work starting from the beginning of the FYP process which is the submission of proposal.

2. Lack of information and communication in FYP community

Since it does not have any formal portal for the subject itself, it is hard to deliver the information regarding the subject to all person involved in FYP. This problem contributes to the lack of communication among the FYP community.

3. Slow process of proposal approval and assignment of supervisor

This primary problem has led to the other matters such as a slow process of proposal approval and assigning supervisor to the student. The process of assigning topic to the related lecturer's area of specialization can be categorized as semi-technical work as it combines decision making with specific criteria which determined by human and logics.

4. Usability of automated system to assist human work

Most of technical work can be automated as long as it has the logic inside. But the concern is when it needs human intervention to make the process more accurate rather than too limited to the current configuration. To assign Final Year Project (FYP) topic to the related lecturer's area of specialization is going to involve two mechanism which is system logic based on specific input criteria and intelligent engine that acts as 'human' who determines the output of the system. It will be more than just assigning topic to the lecturers since it must concern the limit of number of students to be supervised and multiple proposals from each of the students.

5. The needs of features to manage FYP efficiently

In other issues, there are a few other functions need to be highlighted to clarify the significant of the new development of FYP system such as the proposal approval status, progress tracking, online report submission, coordinator/supervisor announcement method and coordinator decision support modules. These functions generally help the system to overcome the current problem of the FYP management for each of the specific process.

1.3 Objective and Scope of Study

The objectives of the project are:

1. To develop a web based FYP Proposal Management System
2. To improve information and communication system in FYP community.
3. To perform a research and development of intelligent software agent for web based application.

The scope of study for this project is a full development of intranet web based application which accessible within the campus by the authorized users. The system basically serves two major functions which is one as the automation system to manage FYP Proposal processes as mentioned earlier and the other one as the information based system that integrate and deliver information to the people involved in FYP.

During the development of the web based application, the research will be focused on intelligent software agent. Intelligent software will be developed and it will be integrated with the application to work in web environment system. It will serve a lot of functions such as matchmaking and brokering the proposal, determining area of research, sorting and searching mechanism, data processing and information gathering.

CHAPTER 2

LITERATURE REVIEW

As mentioned earlier in the Objective and Scope of Study section, the research element in this project is based on intelligent software agent. Since the impact of usability of this intelligent software agent for the whole development, it is a good thing to revise what is the intelligent software agent is all about. This is to ensure that the research focus achieves the objective of the project as it will be the biggest contribution for the whole development of FYP Proposal System.

Intelligent software agent is defined as a piece of software which performs a given task using information acquired from its environment to act in suitable manner so as to complete the task successfully. The software should be able to adapt itself based on changes occurring in its environment, so that a change in circumstances will still yield the intended result [1]. At minimum, an agent must run asynchronously and the degree of agency is enhanced if an agent represents a user in some way. A more advanced agent can interact with other entities such as data, applications or services. Further advanced agents can collaborate and negotiate with other agents. What exactly makes an agent “intelligent” is something that is hard to define. A workable definition in this case is the degree of reasoning and learned behavior which is the ability of the agent to accept the user’s statement of goals and carry out the task delegated to it. [2-3] At minimum, there can be some statement of preferences, perhaps in the form of rules with an inference engine or some other reasoning mechanism to act on these preferences. Higher levels of intelligence include a user model or some other form of understanding and reasoning about what a user wants done and planning the means to achieve this goal [4].

There is one good reason why a fairly concise definition of an agent that can meet with general needs of the FYP system such as matchmaking and brokering

information. So when considering just exactly what an agent is and which aspects of it are very important and which are less important, the objective of the system would not be overlooked.

CHAPTER 3

METHODOLOGY

In this section, I will discuss two important things in the methodology for this project which are process identification and tools required. Process identification will focus on the technique will be used to develop the software and tools required will explain the equipment needed for the development and operation.

3.1 Process Identification

In order to identify the process for the software development, a software process model has been used to find the correct technique to develop the software. A software process usually consists of four basic activities which are:

- Software specification
- Software design and implementation
- Software validation
- Software evolution

Software specification defines the functionality of the software and constrains on its operations. The defined specification will be produced in software design and implementation. In order to ensure that the design and implementation meet the specification, software validation will be performed. The changes in specification required the software to evolve to meet the changing requirement over the time.

Since this project may require change of requirement from time to time during the development, there is a need to support process iteration where parts of the processes are repeated as system requirements evolve. The system design and implementation work must be reworked to implement the changed requirements.

Thus, a hybrid model for software process is chosen to help the process iteration in this project. An incremental development approach is used to support the software process. The following figure shows the development model in this project using incremental development approach:

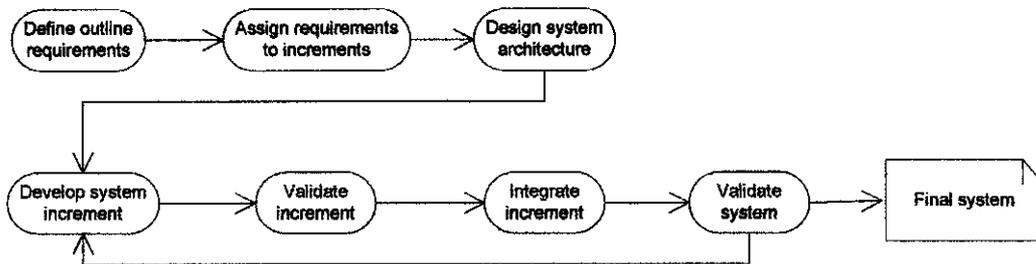


Figure 3.1

In software specification, there are three phases to be used for this project which are requirement elicitation and analysis, requirement specification and requirement validation. Requirement elicitation and analysis is the process of deriving the system requirements through observation of existing system, discussion with potential users, task analysis and so on. Requirement specification is the activity of translating the information gathered during the analysis into a set of requirements document. It is consist of user requirements and system requirements. The last phase, requirement validation is the activity to check the requirements for realism, consistency and completeness. The following figures illustrate the phases in requirement identification:

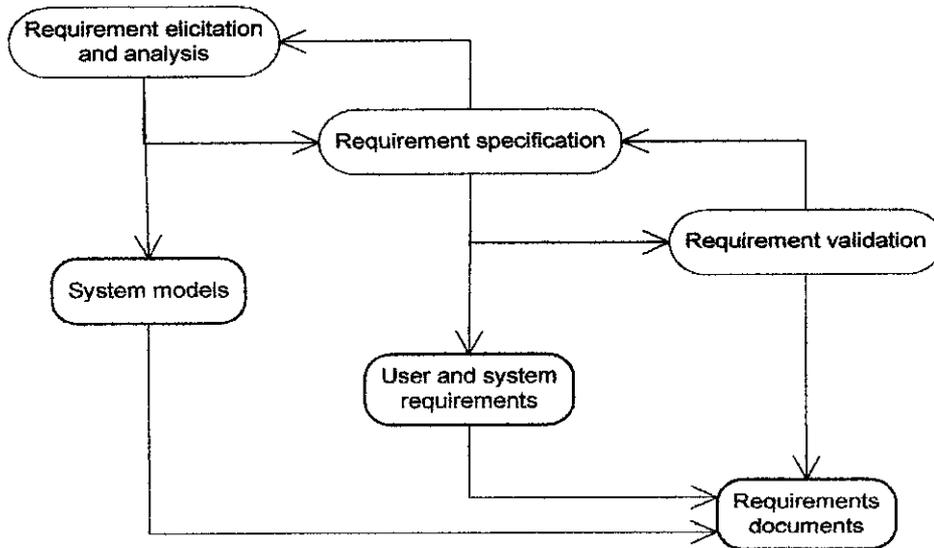


Figure 3.2

In software design and implementation, at the current phase, an architectural design has been developed to give the overview of the FYP Management System. The following figure illustrates the general overview of the system:

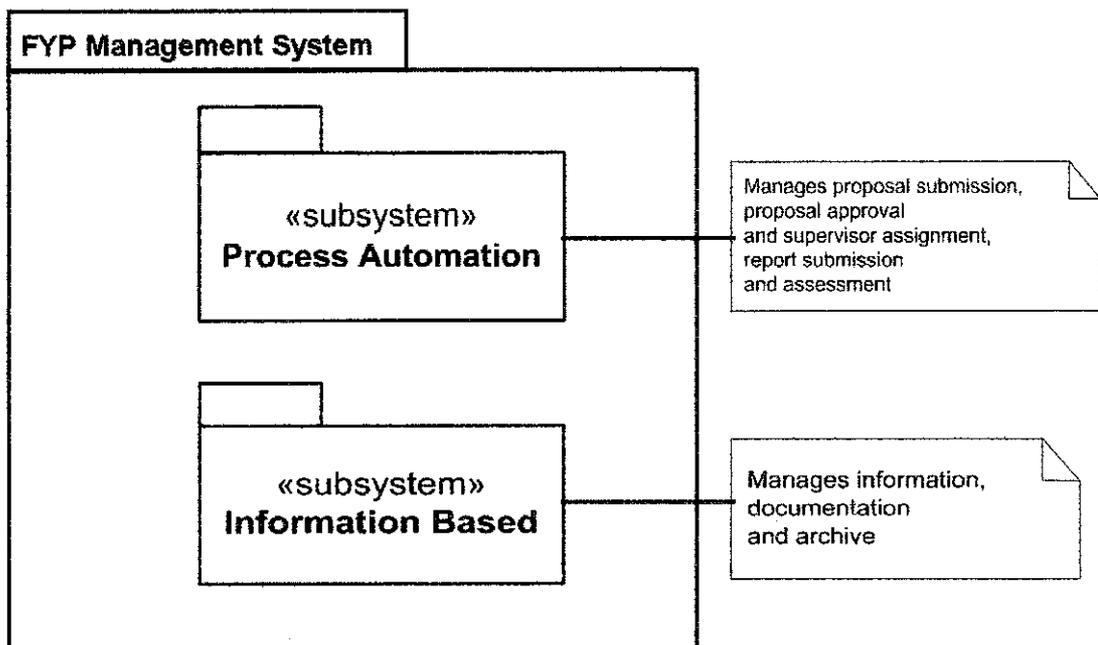


Figure 3.3

According to the illustration, the system consists of two subsystems which are Process Automation subsystem and Information Based subsystem. Process Automation is responsible to manage proposal submission of student, the proposal approval and supervisor assignment process, reports submission and assessment. Logically all of these processes are sequential which means one process is followed by the next process. The Information Based is responsible to manage information and documentation of Final Year Project and archive for the recently completed project. This subsystem will focus on information delivery about Final Year Project including all necessary documentation related to FYP and searching function for recent projects documentation.

The following figures illustrate the components for both subsystems Process Automation and Information Based in their own subsystem architecture:

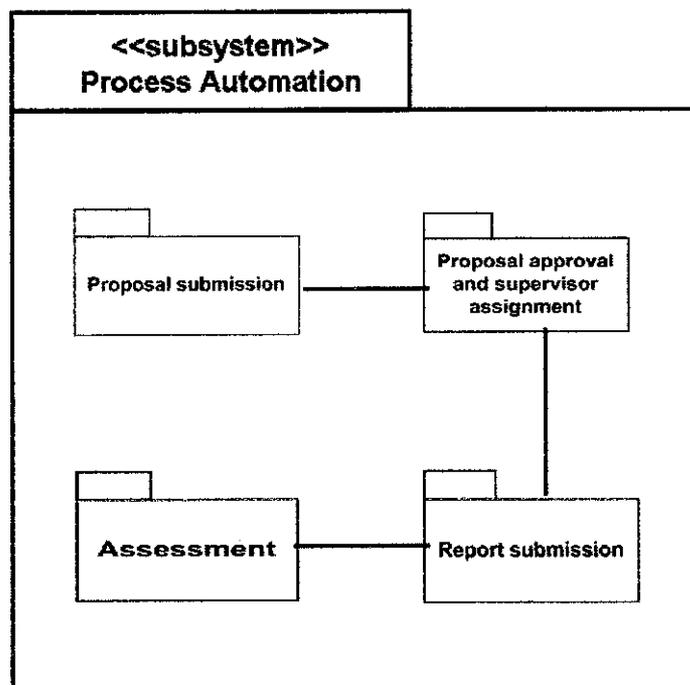


Figure 3.4

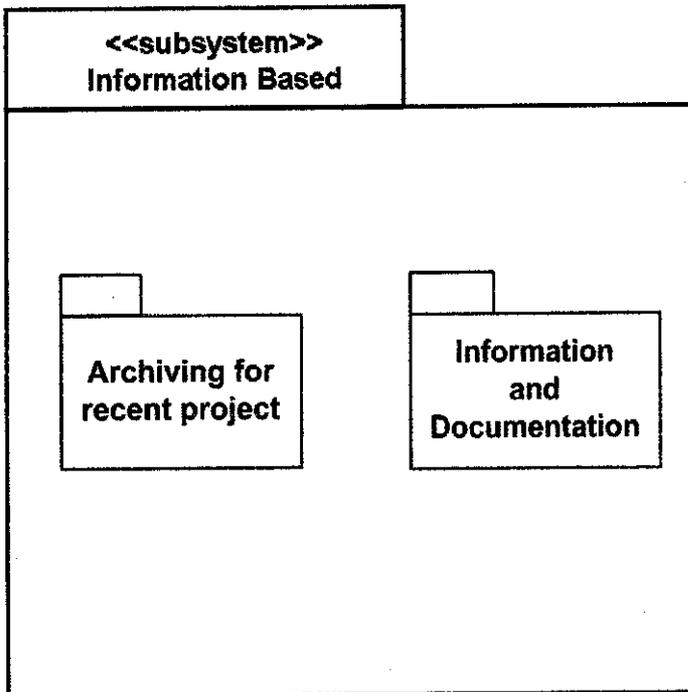


Figure 3.5

In *software validation*, the process is intended to show that this system conforms to its specification and that the system meets the expectations of the potential users. It involves checking processes such as inspections and reviews at each stage of the software process from user requirement definition to program development. The FYP Proposal Management System is considered as large systems built out of sub-systems which are built out of modules which are composed of procedures and functions. Thus, the testing process should proceed in stages where testing is carried out incrementally in conjunction with system implementation. For this project, there will be five stages of in the testing process which are unit testing, module testing, sub-system testing, system testing and acceptance testing. The following figure illustrates this testing process in conjunction with software process for this project.

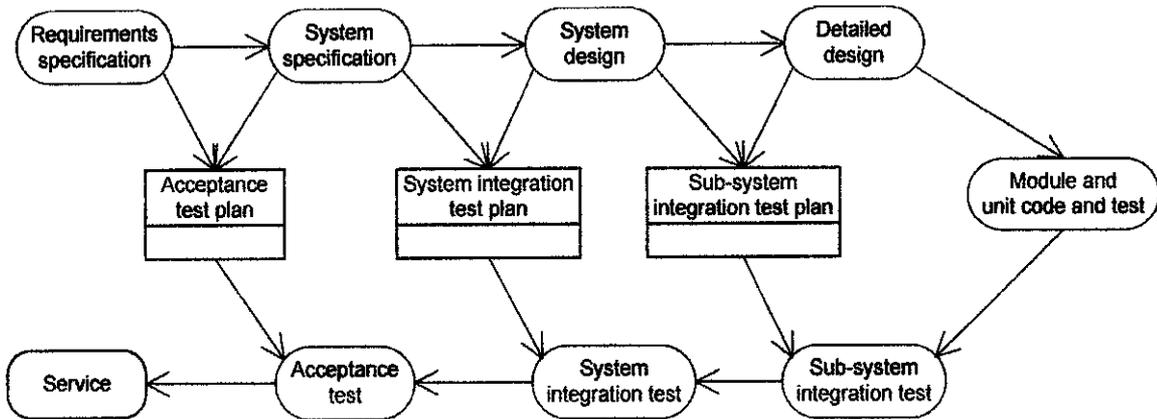


Figure 3.6

3.2 Tools Required

The development of FYP Proposal (FYP) Management System is going to require a few tools and equipment during the development and implementation. The tools are usually common in web based system development and they are:

- Web platform – Active Server Pages (ASP)
- Web programming tools – Microsoft Development Environment
- Web server – Internet Information Services (IIS)
- Operating system – Microsoft Windows 2000 Server
- Database Management System – Microsoft SQL 2000

CHAPTER 4

RESULTS AND DISCUSSION

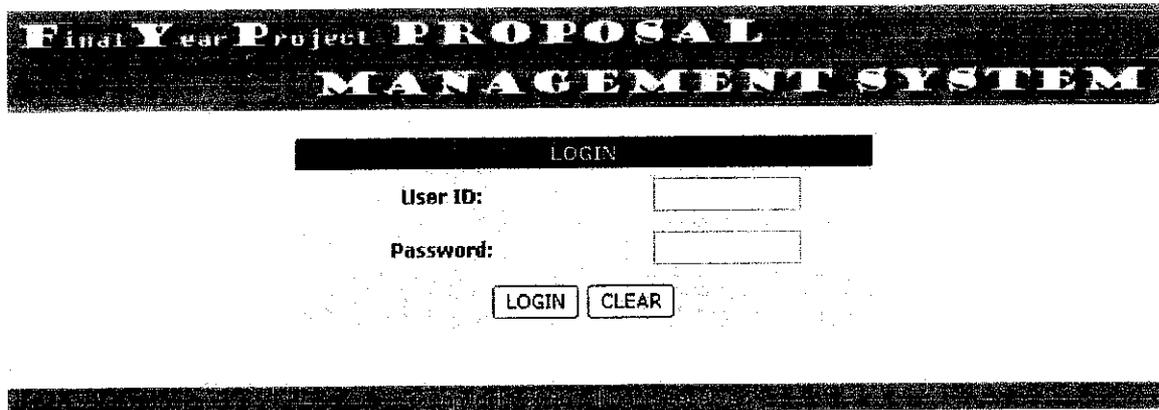
The main content of this section is the discussion of progress for the project until the date of this report written. The discussion will be carried out according to the modules developed in the system. Some of the modules are still in development which would not be discussed in this report.

Basically, the system serves 3 types of users for different purposes. The 3 types of users as mentioned earlier are Student, Supervisor and Coordinator. All users will be able to access 3 main modules; *Proposals, Messages and Account Setting*. By the way, each module has different functions for each type of user. For instance, Proposals module allows Student to submit proposal while for Supervisor the function is to approve the proposal and for Coordinator one of the function is to manage the system how to handle proposals submitted by students. Each module will be discussed in detail in the next part.

4.1 Core Modules

Seven important modules have been completed for this project until now. These seven modules serves are developed for Student and Supervisor, the most crucial users in the system. All modules for coordinator are still in development and the discussion will be reported in the final dissertation.

LOGIN module



The screenshot shows the login interface for the 'Final Year Project PROPOSAL MANAGEMENT SYSTEM'. At the top, the system name is displayed in a stylized font. Below this, a black header bar contains the word 'LOGIN' in white. The main form area has a light background and contains two input fields: 'User ID:' and 'Password:'. Below the 'Password:' field are two buttons: 'LOGIN' and 'CLEAR'.

Figure 4.1

As far as the login concept is concerned, the purpose of the LOGIN module (Figure 4.1) is to provide user authentication for the user of this system. Once the user has provided the correct login information, the system will validate the user from the database to determine the user's type, which is needed in order to bring the user to the correct modules in the system. Since the system serves for three different types of users, it provides 3 different interfaces for each user to differentiate the usage of the system itself. The different interfaces will be seen much clearer in the next modules discussed.

The user ID used in this system is based on student/staff ID. The coordinator will be responsible to import the list of students' IDs and lecturers' IDs qualified for Final Year Project to the system. The coordinator will be provided necessary functions to setup the account for all users. For instance, users will be provided initial password for the first time login and they are required to change the password later on.

MAIN module

Final Year Project PROPOSAL MANAGEMENT SYSTEM

PROPOSALS MESSAGES ACCOUNT SETTING ARCHIVE LOGOUT

Logged in user: Md Hakim Saleh (Student)

ANNOUNCEMENT

Deadline for proposal submission

The following is the deadline for the proposal submission: 21 July 2006 Friday

Posted by: Nordin Zakaria
Date: 6/19/2006 9:49:14 AM

PROPOSAL STATUS				
Proposal	Group	Supervisor	Status	Action
Software for Bookstore	Software Engineering	Shuib Basri	Approved	<input type="button" value="Confirm"/>
System for Timetable	Software Engineering		Pending	

Figure 4.2a Main Module – User: Student

Final Year Project PROPOSAL MANAGEMENT SYSTEM

PROPOSALS MESSAGES ACCOUNT SETTING ARCHIVE LOGOUT

Logged in user: Shuib Basri (Supervisor)

Deadline for proposal submission

The following is the deadline for the proposal submission: 21 July 2006 Friday

Posted by: Nordin Zakaria
Date: 6/19/2006 9:49:14 AM

STATUS

PROPOSALS

No proposal pending for approval

Software for Bookstore

No proposal has been rejected

SUPERVISEES

Saiful Fakhri Ahmad Zaim - INDUSTRIAL INTERNSHIP MANAGEMENT SYSTEM

Figure 4.2b Main Module – User: Supervisor

Final Year Project PROPOSAL MANAGEMENT SYSTEM

PROPOSALS MESSAGES ACCOUNT SETTING ARCHIVE LOGOUT

Logged in user: Nordin Zakaria (Administrator)

ANNOUNCEMENT

Deadline for proposal submission

The following is the deadline for the proposal submission: 21 July 2006 Friday

Posted by: Nordin Zakaria
Date: 6/19/2006 9:49:14 AM

PROGRESS OVERVIEW

PROPOSALS

Pending for approval

Student	Title	Supervisor	Group
Salmiah Sulong	Student Information Aggregator Tools using RSS	Shakirah Mohd Taib	E-Commerce
Md Hakim Saleh	System for Timetable	Rozana Kasbon	Software Engineering
Nurulaziah Zaki	Text Summarization in Semantic Way	Helmi Md Rais	Knowledge Management

Approved

Student	Title	Supervisor	Group
Nurulaziah Zaki	Students Activities and Co-O Management System	Saipunidzam Mahamad	Software Engineering
Md Hakim Saleh	Software for Bookstore	Shuib Basri	Software Engineering

Rejected

No proposal has been rejected

Confirmed

Student	Title	Supervisor	Group
---------	-------	------------	-------

Figure 4.2c Main Module – User: Administrator

At a glance, the above figures (Figure 4.2) show the main page when the user has successfully logged in. The system has detected that the user's type and thus bring the user to this main page and all the modules accessible for Student, Supervisor and Coordinator have been preset.

As an introduction for all modules in this system from the main page, we will start with the header menu (Figure 4.3). The header menu contains five main modules; PROPOSALS/REPORTS, MESSAGES, ACCOUNT SETTING, SCHEDULE and ARCHIVE. These menus provide access to the specific page of each module.



Figure 4.3

Right below the menu bar is the logged in user's name followed by announcement box and the posted information (Figure 4.4). The announcement is displayed every time the user logged in and the announcement that will be displayed is the latest one while the recent announcement can be found in MESSAGES module.

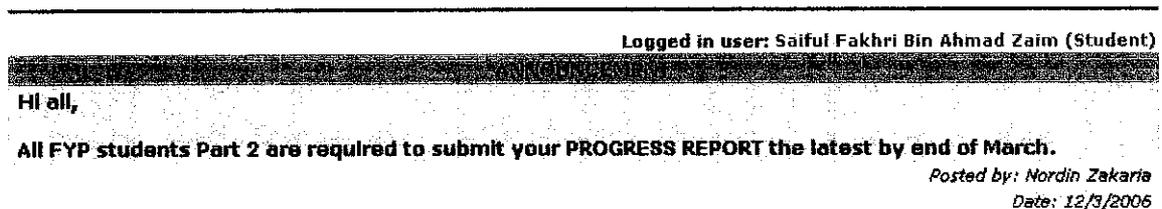


Figure 4.4

The next table displayed below the announcement box is the Status information for the user (Figure 4.5). In this case it provides general information about the proposal status, group that the user belongs to and reports submission status. Proposal status will indicate how many proposals have been submitted and the approval status for each proposal submitted. Group status will indicate the group of research the student belongs to and his or her supervisor's name. Technically the group information is available only when the proposal has been approved by the respective supervisor. The reports status will indicate the submission status for each type of reports required in Final Year Project. For example if the progress report has been submitted, the status will be indicated as *Submitted* next to the report's type.

STATUS		
Proposal	Group	Reports
GHG Research Assistant Systems - Approved - Visual Remote Vehicle Tracking and Monitoring System - Discard -	E-Commerce Supervisor: Mazlina Mahat	Preliminary report - Submitted - Interim report - Submitted - Weekly report - Week 4 - Progress report - Pending - Final report - Pending -

Figure 4.5

PROPOSAL module (USER: STUDENT)

Final Year Project PROPOSAL MANAGEMENT SYSTEM

[PROPOSALS](#) [MESSAGES](#) [ACCOUNT SETTING](#) [ARCHIVE](#) [LOGOUT](#)

Status Logged in user: Md Hakim Saleh (Student)

Submit Proposal

You have submitted 2 proposal(s). You can submit 1 more proposal(s)

SUBMIT NEW PROPOSAL

Project Title:

Problem Statement:

Objectives:

Research Group: Software Engineering ▼

Figure 4.6a – Submit Proposal – User: Student

Final Year Project PROPOSAL MANAGEMENT SYSTEM

PROPOSALS MESSAGES ACCOUNT SETTING ARCHIVE LOGOUT

Logged in user: Shuib Basri (Supervisor)

<< [Back to Proposal page](#)

Proposer's name: Saiful Fakhri Ahmad Zaim	ID: 3429
Email: saiful.fakhri@gmail.com	Contact No.: 012-4079810
Project Title: INDUSTRIAL INTERNSHIP MANAGEMENT SYSTEM	
Problem Statement:	
The student industrial internship programme needs to be managed more efficient since out of other programmes offered for student, industrial internship is the most complicated programme to handle by the management since it involves outside people. Thus, UTP should be equipped with a system	
Objectives:	
1. To provide a web based, extranet system to manage the industrial internship program from the very beginning of the process to the final process in industrial internship management.	
2. To emphasize the research on intelligent engine for	
Summary:	
The primary focus for this research project is to provide a web based system using extranet environment for the industrial internship management. The target users for this	

Figure 4.6b – Submit Proposal – User: Supervisor

Accessible from menu bar under the PROPOSALS, this module is purposely for the students to submit their proposals up to 3 different proposals. At the top of the page under the menu bar is the notification of how many proposals have been submitted and how many more can be submitted. Below the notice box is the proposal template that will be used to send a proposal. Just like the current manual process, the users are required to fill in the information in the text areas given and hit the button SUBMIT to send the proposal through the system. At the back end of the system, the proposal will be analyzed using the intelligent agent to identify which group the proposal is belongs to. Once the group has been identified, the proposal will be sent to the available lecturer in that particular group. The availability of lecturer is determined by the number of supervisees currently assigned to him or her. When the lecturer received the proposal, then the next step is the approval process from his or her point of view which will be discussed in detail in the Supervisor modules section.

Logged in user: Nordin Zakaria (Administrator)

RECENT ANNOUNCEMENT		
6/19/2006 9:49:14 AM	Deadline for proposal submission	Delete
6/19/2006 9:48:31 AM	Information about supervisor under networking	Delete
6/19/2006 9:46:52 AM	Attention: Please confirm your proposal	Delete
6/16/2006 5:59:16 PM	Proposal Submission	Delete

COMPOSE ANNOUNCEMENT

SUBJECT:

MESSAGE:

Figure 4.8b Message (announcement) – User: Administrator

Next to PROPOSALS menu (from the bar menu) is the MESSAGES menu which will redirect the user to MESSAGES module page (Figure 4.8). The functions of this module are to display recent announcement, messages received (Inbox) and to provide message composer. The Recent Announcement section in Figure 4.8 listed all recent announcement have been posted by the coordinator or administrator. Immediately below the announcement section is Inbox section. This section serves similarly to Inbox function in any email program. In this system, the Inbox stores all the messages received either from supervisors or coordinator/administrator.

Below the Inbox section is Compose Message (shown separately in Figure 4.9) that provides the user with message composer function. In this case, the user can send the message to his or her supervisor, lecturers in the same research group and administrator. This group of people is listed in the composer and the user can simply choose from the list and send the message.

COMPOSE MESSAGE

TO: Dr Ahmad Kamil Mahmood

SUBJECT:

MESSAGE:

SEND RESET

Figure 4.9

ACCOUNT SETTING module (USER: STUDENT)

Final Year Project **PROPOSAL**
MANAGEMENT SYSTEM

PROPOSALS MESSAGES ACCOUNT SETTING ARCHIVE LOGOUT

Logged in user: Saiful Fakhri Ahmad Zaim (Student)

ACCOUNT SETTING

Profile	Change Password
Name: Saiful Fakhri Ahmad Zaim	Original password: <input type="text"/>
ID: 3429	New password: <input type="text"/>
Programme: Information Technology	Confirm password: <input type="text"/>
Email: saiful.fakhri@gmail.com Edit	Save Reset
Contact no: 012-4079810 Edit	

Figure 4.10a Setting – User: Student

Final Year Project PROPOSAL MANAGEMENT SYSTEM

PROPOSALS MESSAGES ACCOUNT SETTING ARCHIVE LOGOUT

Logged in user: Nordin Zakaria (Administrator)

STUDENT ACCOUNT SETTING					
User ID	Name	Program		Reset	Password
3310	Md Hakim Saleh	Information Technology		Edit	Reset
2347	Md Shahrul Naim	Information System		Edit	Reset
3378	Nurulaziah Zaki	Information Technology		Edit	Reset
3429	Saiful Fakhri Ahmad Zaim	Information Technology		Edit	Reset
3475	Salmiah Sulong	Information System		Edit	Reset
9000	The Juggernut	Information Technology		Edit	Reset

NEW STUDENT

User ID	<input type="text"/>	Name	<input type="text"/>	Program	<input type="text"/>	<input type="button" value="Save"/>
---------	----------------------	------	----------------------	---------	----------------------	-------------------------------------

Figure 4.10b Setting – User: Admin

SUPERVISOR ACCOUNT SETTING				
User ID	Name	Group	Rank	Edit
1111	Baharum Baharudin	Artificial Intelligence	1	Edit
1118	Norshuhani Zamin	Artificial Intelligence	2	Edit
1090	Jack Bauer	Artificial Intelligence	3	Edit
1127	Anang Hudaya Muhamad Amin	E-Commerce	3	Edit
1113	Shakirah Mohd Taib	E-Commerce	1	Edit
1120	Mazeyanti Mohd Ariffin	E-Commerce	2	Edit
1124	Syarifah Bahiyah Rahayu Syed Mansoor	GIS	2	Edit
1117	Wan Fatimah Wan Ahmad	GIS	1	Edit
1131	Utbah Zainuddin	GIS	3	Edit
1130	Helmi Md Rais	Knowledge Management	3	Edit
1116	Halabi Hasbullah	Knowledge Management	1	Edit
1123	Suhaimi Abdul Rahman	Knowledge Management	2	Edit
1125	Ahmad Kamil Mahmood	Knowledge Management	4	Edit
1122	Abas Md Said	Multimedia/Graphics	2	Edit
1115	Jafreezal Jaafar	Multimedia/Graphics	1	Edit
1129	Vivian Yong Suet Peng	Multimedia/Graphics	3	Edit
1128	Dayang Rohana Awang Rambli	Networking	3	Edit
1114	Khairul Shafee Khalid	Networking	1	Edit
1121	Mohd Fadzil Hassan	Networking	2	Edit
1119	Rozana Kasbon	Software Engineering	2	Edit
1126	Saipunidzam Mahamad	Software Engineering	3	Edit
1112	Shuib Basri	Software Engineering	1	Edit

NEW SUPERVISOR

User ID	<input type="text"/>	Name	<input type="text"/>	Group	Software Engineering	<input type="button" value="Save"/>
---------	----------------------	------	----------------------	-------	----------------------	-------------------------------------

Figure 4.10c Setting – User: Admin

Accessible from the menu bar, ACCOUNT SETTING menu will redirect the user to ACCOUNT SETTING module page (Figure 4.10). The function of this module is to allow the user to view, edit the profile and to change the password. For user type Student, the editable detail is only for Email and Contact Number. On the same page, the user can change the password for the login purpose.

CHAPTER 5

CONCLUSION

As a conclusion for this report, here I would like to highlight some advantages of this project. The system will be well designed and developed according to the detail process identification in the software process. Thus, the development of the system will be right on track by following the planning for the whole system. Based on the recent analytical study, this project is relevant to the current situation of existing system of FYP Proposal management. It is due to the method of resolving the current problems in FYP Proposal management by the development of this web based system. The system is expected to solve the problems by providing a lot of automation functions to assist the processes in FYP which are carried out manually for the current time. With the capability of managing and delivering information via the web system, the target of the system is to provide effective communication among the FYP community itself and better solution for information look-out. This can be achieved with the development of subsystem specifically for managing information and documentation of FYP and archiving for the recently completed project. All of these advantages will be achieved through a good planning for the entire development process of this project.

RECOMMENDATION

Since the system has been developed for a very specific purpose with and with limited functionalities, a few recommendations have been made for further improvement of this system.

- Archive for all proposals for reference
 - This function is expected to be added in order to assist student and lecturer to browse through all recent proposals. The reason is, student might want to know what kind of projects already carried out in FYP. This will help the student to come out with a new idea and avoid repeating submission of the same topic unintentionally.

- Time constrain functions
 - The system currently is designed to support the submission of proposal at any time which means there is no due date or deadline for the submission. The improved system later is expected to have a function that can set a deadline for the submission as well as restrict the submission later than the date. This is correlated to the policy to improve time effectiveness for the whole FYP progress.

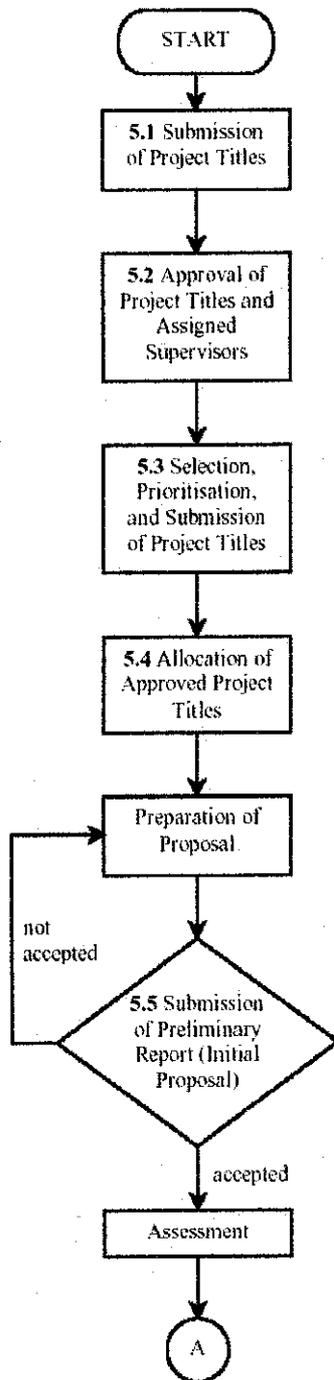
- Editable submitted proposal
 - All proposals that have been submitted through this system are not allowed to be edited in term of its content. The improved system is expected to have a function that can allow the proposal to be edited by the submitter under a controlled condition that might be necessary. This is to improve the flexibility of the system to be manipulated at certain cases like for example the proposal has been approved but the content needs to be changed for some reasons.

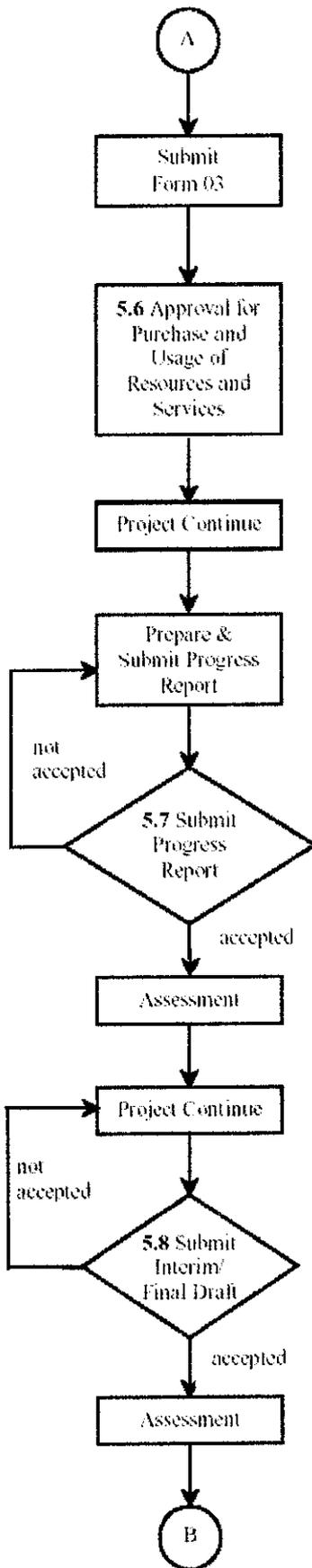
REFERENCES

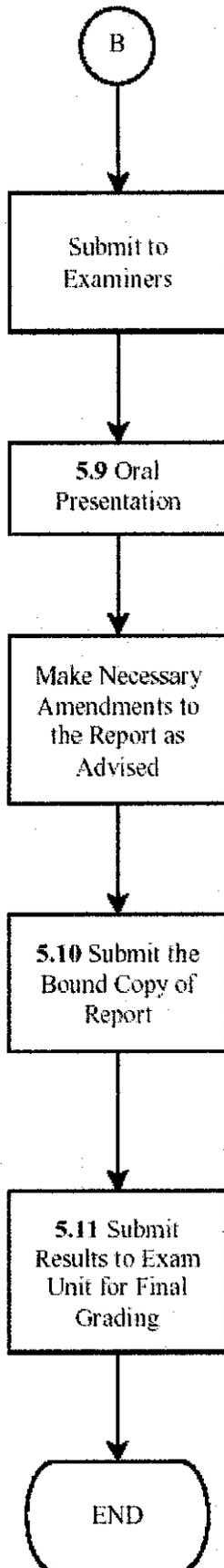
- [1] Etzioni, O. and Weld, D.S. Intelligent Agents on the Internet - Fact, Fiction, and Forecast. *IEEE Expert - Intelligent Internet Services*, number 4, page 44-49. August 1999.
- [2] Gilbert, Aparicio, et al. The Role of Intelligent Agents in the Information Infrastructure. IBM, United States, 2000.
<http://activist.gpl.ibm.com:81/WhitePaper/ptc2.htm>
- [3] Janca, Peter. Pragmatic Application of Information Agents. BIS Strategic Decisions, Norwell, United States, May 2001.
- [4] Nwana, H.S. Software Agents: An Overview. Intelligent Systems Research AA&T, BT Laboratories, Ipswich, United Kingdom, 2002.
<http://www.cs.umbc.edu/agents/papers/ao.ps>
- [5] Wooldridge, M., and Jennings, N.R. Intelligent Agents: Theory and Practice. January 2001. <http://www.doc.mmu.ac.uk/STAFF/mike/ker95/ker95-html.html>
- [6] IITA, Information Infrastructure Technology and Applications. Report of the IITA Task Group High Performance Computing, Communications and Information Technology Subcommittee, 2003.
<http://www.hpcc.gov/reports/reports-nc0/index.iita.html>

APPENDIX I:

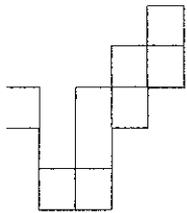
FYP Process Flow

Final Year Project Process Flow





APPENDIX II: FYP Final Presentation Slides

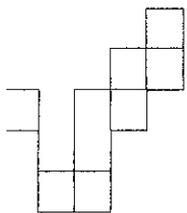


FYP Proposal Management System

By
Saiful Fakhri Bin Ahmad Zaim
3429
Information Technology

Presentation Outline

- Introduction
 - Background of Project
 - Problem Statement
 - Objectives and Scope of Project
- Methodology
 - Process Identification
 - Tools Required
- Discussion
- Recommendation & Conclusion
- Product Demonstration
- References



Introduction

Background of Project

- FYP proposal as a kick starter to carry out the project through out the year.
- Student submits proposal to any lecturer usually related to area of research for that project.
- Web based system to manage proposal through the process of submission, grouping, approval and confirmation.
- Research in intelligent software agent

Problem Statement

- Mismatch between the project and the supervision
- Unequal number of students supervised
- Slow process of proposal approval
- Lack of information and communication

Objectives

- To develop a web based system for proposal management
- To solve the problem of
 - Mismatch between the project and the supervision
 - Unequal number of students supervised
 - Slow process of proposal approval
 - Lack of information and communication

Scope of Project

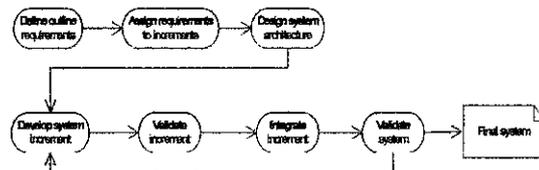
- Full development of intranet web based application
- Research and development of intelligent software agent

Methodology

Process Identification

- Incremental Development Approach
 - Software specification
 - Software design and implementation
 - Software validation

Process Identification...cont



Tools Required

- Web platform – Active Server Pages (ASP)
- Web programming tools – Microsoft Development Environment
- Web server – Internet Information Services (IIS)
- Operating system for server – Microsoft Windows 2000 Server
- Database Management System – Microsoft SQL Server 2000

Discussion

Users Requirement

- Student
 - Submit proposal
 - Confirm proposal
- Supervisor
 - Approve or reject proposal
- Coordinator
 - Observe the progress
 - Manage users account

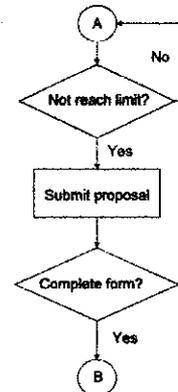
System Requirement

- Proposal module
 - Match the proposal with the supervisor based on defined procedure
 - Indicate the status of submitted proposal
- Message module
 - Perform simple messaging function (send, receive, archive)
- Account Setting module
 - Password and particulars management

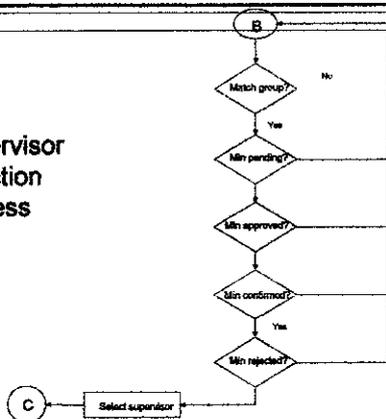
System Requirement... cont

- Proposal module
 - Process flow (next slide)

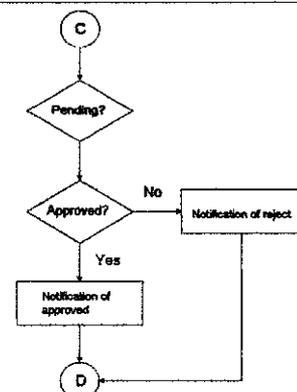
Proposal submission process



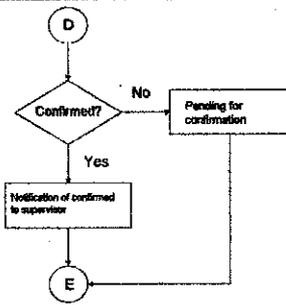
Supervisor selection process



Proposal approval process

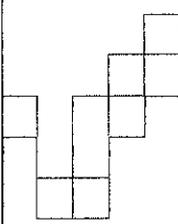


Proposal confirmation process



Limitations

- Student can't choose supervisor
- Digital format of proposal – no print out
- Limited function of messaging
- Submitted proposal cannot be edited using the system



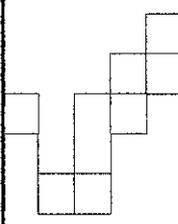
Recommendation & Conclusion

Recommendation

- Archive for all proposals for reference
- Time constrain functions
- Editable submitted proposal
- File attachment function

Conclusion

- Relevant to the needs of current situation
- Feasible – function and purpose
- Capable to solve the addressed problem
- Accurate in managing proposal
- Reduce the time taken for approval process



Product Demonstration



References

- Nwana, H.S. *Software Agents: An Overview*. Intelligent Systems Research AA&T, BT Laboratories, Ipswich, United Kingdom, 2002.
<http://www.cs.umbc.edu/agents/papers/eo.ps>
- Gilbert, Aparicio, et al. *The Role of Intelligent Agents in the Information Infrastructure*. IBM, United States, 2000.
<http://activist.lpl.ibm.com:81/WhitePaper/ptc2.htm>
- Sommerland, B. *Software Engineering Practice 3rd Edition*, Apprenice Hall Publications, 2005.

APPENDIX III: Implementation Codes

```
language=VBScript%>
```

```
Main module (User:student) -->
```

```
<!--#INCLUDE FILE='_library\libFunctions.asp'-->
<!--#INCLUDE FILE='_library\ADOVBS.INC'-->
<!--#INCLUDE FILE='_library\error.asp'-->
```

```
conn, rsViewProposal, sqlViewProposal, sqlMySV, rsMySV, strSUPV_NAME
rsViewAnnouncement, sqlViewAnnouncement
```

```
conn = server.CreateObject("ADODB.Connection")
conn.Open = "Provider=SQLOLEDB.1;Password=*****;User ID=sa;Initial Catalog=fypms;Data
Source=CHEEP" ✓
```

```
rsViewProposal = server.CreateObject("ADODB.RecordSet")
rsMySV = server.CreateObject("ADODB.RecordSet")
rsViewAnnouncement = server.CreateObject("ADODB.RecordSet")
```

```
sqlViewAnnouncement = "SELECT SUBJECT, MESSAGE, DATEPOST FROM dbo.NOTICE ORDER BY
DATEPOST DESC" ✓
```

```
rsViewAnnouncement.Open sqlViewAnnouncement, conn
```

```
-----BEGIN TOP OF PAGE-----
```

```
'Create procedures to generate HTML
htmlHeader( "Final Year Project Management System - Status" )
htmlDetail( "" )
htmlFooter()
```

```
*** FUNCTIONS *****'
```

```
function to read the values from the table & display in the fixed format
function htmlDetail( strTitle)
Response.write("<BODY Link='RED' TopMargin='0' Link='BLUE' leftMargin=0 topMargin=0
rightMargin=0 MARGINHEIGHT=0 MARGINWIDTH=0>" ) & VBCRLF
Response.write("<SCRIPT language=JavaScript src=""_library/menu_array.js""></SCRIPT>
") & VBCRLF
Response.write("<SCRIPT language=JavaScript src=""_library/mmenu.js"" type=text/
javascript></SCRIPT>") & VBCRLF ✓
```

```
table align=center border=0 width="780" cellpadding="0" cellspacing="0">
<tr>
<td>
```

```
<table width="780" border="0" cellpadding="0" cellspacing="0">
<tr>
<td valign=top></td>
```

```

    </tr>
</table>
<br><br>
<table width="780" border="0" cellpadding="2" cellspacing="0" ID="Table1">
<tr>
<td align=right style="font-size:8pt"><b>Logged in user: <a style="color:blue"><%=
session("NAME")%> (<%=session("USERTYPE")%>)</a></b></td>
</tr>
</table>
<table width="780" border="0" cellpadding="2" cellspacing="0">
<tr>
<td align =center style="font-size:10pt" bgcolor=#ff9933><b>ANNOUNCEMENT</b></td>
</tr>
<tr>
<td style="font-size:10pt" bgcolor=#d4d0c8>&nbsp;<b><%=rsViewAnnouncement("SUBJECT")
%><br><br>&nbsp;<%=rsViewAnnouncement("MESSAGE")%></b></td>
</tr>
<tr>
<td align =right style="font-size:8pt"><i>Posted by: Nordin Zakaria</i></td>
</tr>
<tr>
<td align =right style="font-size:8pt"><i>Date: <%=rsViewAnnouncement("DATEPOST")%></i>
</td>
</tr>
</table>

<table width="780" border="0" cellpadding="2" cellspacing="0" ID="Table2">
<tr>
<td colspan=3 align=center style="font-size:10pt" bgcolor=#000000 style="color:white">
<b>PROPOSAL STATUS</b></td>
</tr>
</table>
<table width="780" border="1" cellpadding="1" cellspacing="0" ID="Table3">
<tr style="font-size:9pt" bgcolor=Gainsboro><td width=40% align=center>&nbsp;<b>
Proposal</b></td><td width=20% align=center>&nbsp;<b>Group</b></td><td width=20%
align=center>&nbsp;<b>Supervisor</b></td><td width=10% align=center>&nbsp;<b>Status</b>
</td><td width=10% align=center>&nbsp;<b>Action</b></td></tr>
<%

sqlViewProposal = "SELECT dbo.USER_ACCOUNT.USERID, dbo.PROPOSAL_STUDENT.TITLE, dbo.
PROPOSAL_STUDENT.RESEARCH_AREA, dbo.PROPOSAL_STUDENT.APPROVAL_STATUS, dbo.
PROPOSAL_STUDENT.SUPV_ID FROM dbo.USER_ACCOUNT INNER JOIN dbo.PROPOSAL_STUDENT ON dbo.
USER_ACCOUNT.USERID = dbo.PROPOSAL_STUDENT.USERID WHERE (dbo.PROPOSAL_STUDENT.USERID
= '& session("USERID") &')"

rsViewProposal.Open sqlViewProposal, conn

if rsViewProposal.EOF then%>

<tr style="font-size:9pt" bgcolor=Gainsboro><td colspan =5 align=center><br>No
proposal submitted<br>&nbsp;</td></tr>

<%

else

    Do While NOT rsViewProposal.EOF%>

        <%if rsViewProposal("APPROVAL_STATUS")= "P" then%>

            <tr style="font-size:9pt" bgcolor=Gainsboro><td align=center>&nbsp;<i><%=
rsViewProposal("TITLE")%></i></td><td align=center><%=rsViewProposal("RESEARCH AREA")
%></td><td align=center>&nbsp;</td><td align=center>&nbsp;<a style="color:red">
Pending</a></td><td align=center>&nbsp;</td></tr>

```

```

    <%elseif rsViewProposal("APPROVAL_STATUS")= "A" Then
        sqlMySv = "SELECT * FROM USER_ACCOUNT WHERE USERID = '"& rsViewProposal(
"SUPV_ID") &"'"
        rsMySV.Open sqlMySV, conn

        strSUPV_NAME = rsMySV("NAME")

        rsMySV.Close

    %>

    <tr style="font-size:9pt" bgcolor=Gainsboro><td align=center>&nbsp;<i><%=
rsViewProposal("TITLE")%></i></td><td align=center><%=rsViewProposal("RESEARCH_AREA")
%></td><td align=center>&nbsp;<%=strSUPV_NAME%></td><td align=center>&nbsp;<a style=
"color:blue">Approved</a></td><td align=center><form method =post name =frmConfirm
action ="main_1_confirm.asp?title=<%=rsViewProposal("TITLE")%>&id=<%=session("USERID"
)%>&supv_id=<%=rsViewProposal("SUPV_ID")%>"><br><input type =submit name = "confirm"
value="Confirm"></form></td></tr>

    <%elseif rsViewProposal("APPROVAL_STATUS")= "C" Then
        sqlMySv = "SELECT * FROM USER_ACCOUNT WHERE USERID = '"& rsViewProposal(
"SUPV_ID") &"'"
        rsMySV.Open sqlMySV, conn

        strSUPV_NAME = rsMySV("NAME")

        rsMySV.Close

    %>

    <tr style="font-size:9pt" bgcolor=Gainsboro><td align=center>&nbsp;<i><%=
rsViewProposal("TITLE")%></i></td><td align=center><%=rsViewProposal("RESEARCH_AREA")
%></td><td align=center>&nbsp;<%=strSUPV_NAME%></td><td align=center>&nbsp;<a style=
"color:blue">Confirmed</a></td><td align=center>&nbsp;</td></tr>

    <%elseif rsViewProposal("APPROVAL_STATUS")= "R" then%>
    <tr style="font-size:9pt" bgcolor=Gainsboro><td align=center>&nbsp;<i><%=
rsViewProposal("TITLE")%></i></td><td align=center><%=rsViewProposal("RESEARCH_AREA")
%></td><td align=center>&nbsp;<%=strSUPV_NAME%></td><td align=center>&nbsp;<a style=
"color:red">Rejected</a></td><td align=center><form method =post name =frmDelete
action ="main_1_delete.asp?title=<%=rsViewProposal("TITLE")%>&id=<%=session("USERID"
)%>"><br><input type =submit name = "delete" value="Delete"></form></td></tr>

    <%elseif rsViewProposal("APPROVAL_STATUS")= "D" then%>
    <tr style="font-size:9pt" bgcolor=Gainsboro><td align=center>&nbsp;<i><%=
rsViewProposal("TITLE")%></i></td><td align=center><%=rsViewProposal("RESEARCH_AREA")
%></td><td align=center>&nbsp;</td><td align=center>&nbsp;<a style="color:red">
Dropped</a></td><td align=center>&nbsp;</td></tr>

    <%else%>

    <%end if%>

<%
rsViewProposal.MoveNext
Loop

```

```
    %>

<%end if%>

</table>

<%rsViewProposal.Close%>

    <br>
</td>
</tr>

able>

End function

'Free the resources that the recordset variable & Data Connection was using
Function htmlFooter()
    Response.Write "</body>" & VBCRLF
End function

<input language="JavaScript" type="button" value="New Proposal" onmouseover="checkStatus()" onmouseout="checkSubmitProposal()" onmousedown="checkSubmitReport()" onmouseup="checkAnnouncement()" onkeypress="checkInbox()" onkeydown="checkSetting()">
or USER

Function checkStatus()
{
    document.location = 'main_1.asp';
}

Function checkSubmitProposal()
{
    document.location = 'newproposal_1.asp';
}

Function checkSubmitReport()
{
    document.location = 'report_1.asp';
}

Function checkAnnouncement()
{
    document.location = 'messages_1.asp';
}

Function checkInbox()
{
    document.location = 'messages_1_inbox.asp';
}

Function checkSetting()
{
```

```
document.location = 'setting_1.asp';
}

function checkSchedule()
{
    document.location = 'schedule_1.asp';
}

function BarPanel()
{
    <% if Session("ACCESS_ID") = "3" or Session("ACCESS_ID") = "2" then %>
        document.location = 'BarLst.asp' ;
    <% else %>
        alert("You don't have access for this Module.");
    <% end if %>
}

function underConstruct()
{
    alert("Under Construction.");
}
-->
</SCRIPT>
nl>
```

language=VBScript%>

Submit Proposal module -->

```

<!--#INCLUDE FILE='_library\libFunctions.asp'-->
<!--#INCLUDE FILE='_library\ADOVBS.INC'-->
<!--#INCLUDE FILE='_library\error.asp'-->

```

```

conn, rsAdd, sqlAdd, rsFindSupv, sqlFindSupv, rsFindMinPending, sqlFindMinPending,
rsFindMinConfirmed, sqlFindMinConfirmed, rsFindMinApproved, sqlFindMinApproved,
rsFindMinRejected, sqlFindMinRejected, rsFindMinRank, sqlFindMinRank, rsUpdatePending,
sqlUpdatePending
strUserID, strTitle, strProbStat, strObj, strArea, strSummary, strTools, strGroup,
strSUPV_ID
intMinPending, intMinRank, intMinApproved, intMinConfirmed, intMinRejected

```

```

conn = server.CreateObject("ADODB.Connection")
.Open = "Provider=SQLOLEDB.1;Password=*****;User ID=sa;Initial Catalog=fypms;Data
Source=CHEEP"

```

```

rsFindSupv = server.CreateObject("ADODB.RecordSet")
rsFindMinPending = server.CreateObject("ADODB.RecordSet")
rsFindMinConfirmed = server.CreateObject("ADODB.RecordSet")
rsFindMinApproved = server.CreateObject("ADODB.RecordSet")
rsFindMinRejected = server.CreateObject("ADODB.RecordSet")
rsFindMinRank = server.CreateObject("ADODB.RecordSet")
rsAdd = server.CreateObject("ADODB.RecordSet")
rsUpdatePending = server.CreateObject("ADODB.RecordSet")

```

```

itle = Request.Form("txtTitle")
robStat = Request.Form("txtProbStat")
bj = Request.Form("txtObj")
rea = Request.Form("lstGroup")
ummary = Request.Form("txtSummary")
ools = Request.Form("txtTools")

```

```

ion("strGroup") = strArea

```

```

onse.Write strArea

```

```

nd minimum pending

```

```

FindMinPending = "SELECT MIN(PENDING) as MIN_PENDING FROM PROPOSAL_SETTING where
RESEARCH_GROUP = '" & strArea & "'"
ndMinPending.Open sqlFindMinPending, conn

```

```

finPending = rsFindMinPending("MIN_PENDING")

```

```

onse.Write intMinPending & "(P)"

```

```

nd minimum approved out of minimum pending

```

```

FindMinApproved = "SELECT MIN(APPROVED) as MIN_APPROVED FROM PROPOSAL_SETTING where
RESEARCH_GROUP = '" & strArea & "' and PENDING = '" & intMinPending & "'"
ndMinApproved.Open sqlFindMinApproved, conn

```

```

MinApproved = rsFindMinApproved("MIN_APPROVED")

```

```

onse.Write intMinApproved & "(A)"

```

```

nd minimum confirmed out of min.pending and min.approved

```

```

FindMinConfirmed = "SELECT MIN(CONFIRMED) as MIN_CONFIRMED FROM PROPOSAL_SETTING where

```

```

RESEARCH_GROUP = '' & strArea & '' and PENDING = '' & intMinPending & '' and
APPROVED = '' & intMinApproved & ''
ndMinConfirmed.Open sqlFindMinConfirmed, conn

inConfirmed = rsFindMinConfirmed("MIN_CONFIRMED")

onse.Write intMinConfirmed & "(C)"

d minimum rejected out of min.pending and min.approved and min.confirmed

indMinRejected = "SELECT MIN(REJECTED) as MIN_REJECTED FROM PROPOSAL_SETTING where
RESEARCH_GROUP = '' & strArea & '' and PENDING = '' & intMinPending & '' and APPROVED
= '' & intMinApproved & '' and CONFIRMED = '' & intMinConfirmed & ''
ndMinRejected.Open sqlFindMinRejected, conn

linRejected = rsFindMinRejected("MIN_REJECTED")

onse.Write intMinRejected & "(R)"

d minimum rank out of min.pending and min.approved and min.confirmed and min.rejected

indMinRank = "SELECT MIN(RANK) as MIN_RANK FROM PROPOSAL_SETTING where RESEARCH_GROUP
= '' & strArea & '' and PENDING = '' & intMinPending & '' and APPROVED = '' &
intMinApproved & '' and CONFIRMED = '' & intMinConfirmed & '' and REJECTED = '' &
intMinRejected & ''
ndMinRank.Open sqlFindMinRank, conn

linRank = rsFindMinRank("MIN_RANK")

onse.Write intMinRank & "(R)"

d the sv using above min. value

indSupv = "SELECT * FROM PROPOSAL_SETTING where RESEARCH_GROUP = '' & strArea & ''
and PENDING = '' & intMinPending & '' and APPROVED = '' & intMinApproved & '' and
CONFIRMED = '' & intMinConfirmed & '' and REJECTED = '' & intMinRejected & '' and
RANK = '' & intMinRank & ''
ndSupv.Open sqlFindSupv,conn

rsFindSupv.EOF then

sv found

found

SUPV_ID = rsFindSupv("SUPV_ID")

onse.Write strSUPV_ID

add = "INSERT INTO PROPOSAL_STUDENT values ('& session("USERID") &', '& strTitle &',
'& strProbStat &', '& strObj &', '& strArea &', '& strSummary &', '& strTools &',
'& strSUPV_ID &', 'P')"

id.Open sqlAdd,conn

updatePending = "UPDATE PROPOSAL_SETTING SET PENDING = PENDING + 1 where SUPV_ID = ''
& strSUPV_ID & ''
updatePending.Open sqlUpdatePending,conn

onse.Redirect("newproposal_1_submit_success.asp")

if

```

ndSupv.Close

-----BEGIN TOP OF PAGE-----

```
'Create procedures to generate HTML
htmlHeader( "Final Year Project Management System - Proposal" )
htmlDetail( "" )
htmlFooter()
```

**** FUNCTIONS ****'

action to read the values from the table & display in the fixed format

```
function htmlDetail( strTitle)
  Response.write("<BODY Link='RED' TopMargin='0' Link='BLUE' leftMargin=0 topMargin=0
rightMargin=0 MARGINHEIGHT=0 MARGINWIDTH=0>" ) & VBCRLF
  Response.write("<SCRIPT language=JavaScript src=""_library/menu_array.js""></SCRIPT>"
) & VBCRLF
  Response.write("<SCRIPT language=JavaScript src=""_library/mmenu.js"" type=text/
javascript></SCRIPT>") & VBCRLF
```

End function

```
'Free the resources that the recordset variable & Data Connection was using
Function htmlFooter()
  Response.Write "</body>" & VBCRLF
End function
```

```
ript language="JavaScript">
For USER
```

```
ction checkStatus()
{
  document.location = 'main_1.asp';
}
```

```
ction checkSubmitProposal()
{
  document.location = 'newproposal_1.asp';
}
```

```
tion checkSubmitReport()
{
    document.location = 'report_1.asp';
}

tion checkMessages()
{
    document.location = 'messages_1.asp';
}

tion checkSetting()
{
    document.location = 'setting_1.asp';
}

tion checkSchedule()
{
    document.location = 'schedule_1.asp';
}

tion BarPanel()
{
    <% if Session("ACCESS_ID") = "3" or Session("ACCESS_ID") = "2" then %>
        document.location = 'BarLst.asp' ;
    <% else %>
        alert("You don't have access for this Module.");
    <% end if %>
}

tion underConstruct()
{
    alert("Under Construction.");
}
-->
</SCRIPT>
ml>
```

```
language=VBScript%>
```

```
Process of approval module -->
```

```
!--#INCLUDE FILE='_library\libFunctions.asp'-->  
!--#INCLUDE FILE='_library\ADOVBS.INC'-->  
!--#INCLUDE FILE='_library\error.asp'-->
```

```
conn, rsViewProposal, sqlViewProposal, rsViewAllProposal, rsUpdateStatus,  
sqlUpdateStatus, sqlAddSV, rsAddSV, rsUpdateSetting, sqlUpdateSetting
```

```
conn = server.CreateObject("ADODB.Connection")  
Open = "Provider=SQLOLEDB.1;Password=*****;User ID=sa;Initial Catalog=fypms;Data  
source=CHEEP"
```

```
rsViewProposal = server.CreateObject("ADODB.RecordSet")  
rsViewAllProposal = server.CreateObject("ADODB.RecordSet")  
rsUpdateStatus = server.CreateObject("ADODB.RecordSet")  
rsAddSV = server.CreateObject("ADODB.RecordSet")  
rsUpdateSetting = server.CreateObject("ADODB.RecordSet")
```

```
rsViewAllProposal = "SELECT dbo.PROPOSAL_STUDENT.USERID, dbo.PROPOSAL_STUDENT.TITLE, dbo.  
PROPOSAL_STUDENT.PROBSTAT, dbo.PROPOSAL_STUDENT.OBJ, dbo.PROPOSAL_STUDENT.SUMMARY,  
dbo.PROPOSAL_STUDENT.TOOLS, dbo.USER_ACCOUNT.NAME, dbo.USER_ACCOUNT.EMAIL, dbo.  
USER_ACCOUNT.CONTACTNO FROM dbo.PROPOSAL_STUDENT INNER JOIN dbo.USER_ACCOUNT ON dbo.  
PROPOSAL_STUDENT.USERID = dbo.USER_ACCOUNT.USERID WHERE (dbo.PROPOSAL_STUDENT.USERID  
'& session("strUserID") & "') AND (dbo.PROPOSAL_STUDENT.TITLE = '& session(  
strTitle") & "')"
```

```
rsViewAllProposal.Open sqlViewAllProposal, conn
```

```
if request.Form("tbc") <> "" then
```

```
dateStatus = "UPDATE PROPOSAL_STUDENT SET APPROVAL_STATUS = 'P' WHERE USERID = '&  
session("strUserID") & "' and TITLE = '& session("strTitle") & "'"
```

```
dateStatus.Open sqlUpdateStatus, conn
```

```
response.Redirect "proposal_2.asp"
```

```
endif
```

```
if request.Form("approve") <> "" then
```

```
dateStatus = "UPDATE PROPOSAL_STUDENT SET APPROVAL_STATUS = 'A' WHERE USERID = '&  
session("strUserID") & "' and TITLE = '& session("strTitle") & "'"
```

```
dateStatus.Open sqlUpdateStatus, conn
```

```
dateSetting = "UPDATE PROPOSAL_SETTING SET PENDING = PENDING - 1, APPROVED =  
APPROVED + 1 where SUPV_ID = '& session("USERID") & "'"  
dateSetting.Open sqlUpdateSetting, conn
```

```
response.Redirect "proposal_2.asp"
```

```
endif
```

```
if request.Form("reject") <> "" then
```

```

updateStatus = "UPDATE PROPOSAL_STUDENT SET APPROVAL_STATUS = 'R' WHERE USERID = '" &
session("strUserID") &"' and TITLE = '" & session("strTitle") &"'"
lateStatus.Open sqlUpdateStatus, conn

updateSetting = "UPDATE PROPOSAL_SETTING SET PENDING = PENDING - 1, REJECTED =
REJECTED + 1 where SUPV_ID = '" & session("USERID") &"'"
lateSetting.Open sqlUpdateSetting, conn

onse.Redirect "proposal_2.asp"
.f

```

-----BEGIN TOP OF PAGE-----

```

'Create procedures to generate HTML
htmlHeader( "Final Year Project Management System - Proposal" )
htmlDetail( "" )
htmlFooter()

**** FUNCTIONS ****

function to read the values from the table & display in the fixed format
function htmlDetail( strTitle)
Response.write("<BODY Link='RED' TopMargin='0' Link='BLUE' leftMargin=0 topMargin=0
rightMargin=0 MARGINHEIGHT=0 MARGINWIDTH=0>" ) & VBCRLF
Response.write("<SCRIPT language=JavaScript src=""_library/menu_array.js""></SCRIPT>"
') & VBCRLF
Response.write("<SCRIPT language=JavaScript src=""_library/mmenu.js"" type=text/
javascript></SCRIPT>") & VBCRLF

```

End function

```

'Free the resources that the recordset variable & Data Connection was using
function htmlFooter()
Response.Write "</body>" & VBCRLF
End function

```

```

ipt language="JavaScript">
or USER

```

```

tion checkStatus()
{
document.location = 'main_1.asp';

```

```
function checkSubmitProposal()
{
    document.location = 'newproposal_1.asp';
}

function checkSubmitReport()
{
    document.location = 'report_1.asp';
}

function checkMessages()
{
    document.location = 'messages_1.asp';
}

function checkSetting()
{
    document.location = 'setting_1.asp';
}

function checkSchedule()
{
    document.location = 'schedule_1.asp';
}

function BarPanel()
{
    <% if Session("ACCESS_ID") = "3" or Session("ACCESS_ID") = "2" then %>
        document.location = 'BarLst.asp' ;
    <% else %>
        alert("You don't have access for this Module.");
    <% end if %>
}

function underConstruct()
{
    alert("Under Construction.");
}

-->
</SCRIPT>
al>
```