



**LEONG KEE FOO  
WET 98057**

**Session 2000/2001**

**WXET 3182**

**Projek Ilmiah Tahap Akhir II**

*Supervisor:*  
**Mr. Ling Teck Chaw**

*Moderator:*  
**Cik Nor Aniza Abdullah**

## ABSTRACT

The technology of graphics is moving to the era of three dimensions (3D). In 1970s, people were using the text-based computer system. When the technique of 2D image processing was invented, people changed their system into 2D representation because of its ability to explain more information in only one image. Around 1990s, 3D graphics have been pushed out to the market. Soon, people accepted 3D as an advance information representation medium.

The most important achievement of 3D technology is the making of virtual reality, which provides human unlimited space and interactivity. The *v-Faculty* applies the concept of virtual reality by creating a 3D-like virtual faculty on the web. It emphasizes on user interactivity where people can communicate with the objects inside the virtual world. In the other word, a realistic-looking faculty is presented on the computer screen.

Besides that, the virtual faculty provides a new online course registration system because the current system is imperfect. The current system is using a centralized processing system. It encounters the problems of data storage management and processing time. Therefore, the new system could improve these weaknesses by distributing data to the database in every faculty. In addition, the new system also provides the students with the facilities of personal data checking and examination result checking as well as courses taken checking.

The research incorporates all features mentioned above as one system, which is named *v-Faculty*. The detail about this research is covered in this report.

## ACKNOWLEDGEMENTS

A thesis report and system like this never just "happens". Without the innovative thinking and support of many people, this thesis would not exist. My good fortune has enabled me to work with a partner of true professional in both the academic and system development. Nevertheless, my supervisor, Mr. Ling Teck Chaw, has provided me sufficient knowledge and skills to finish this thesis.

The most pleasant task I have is expressing my gratitude to my supervisor and my thesis partner, Mr. Hong Wye Kean. I appreciate the support and encouragement I have received from faculty and university. I could not have done without the advice and help of Cik Asiah, system analyst of University of Malaya, in Bahagian Perkhidmatan Komputer (BPK). A special thanks goes to Mr. Ang Tan Foong, my senior, for his suggestions and helps. I also owe my deepest gratitude to Cik Nor Aniza who has recommended techniques to learn and to use the VRML code.

I would also like to acknowledge several other course mates who played important roles in system development. Mr. Wong Wing Hong, Mr. The Eng Kiong, Miss Goh Sze Eng, Miss Chan Yim Khim, Mr. Leong Soon Hing and Mr. Che Choong Lee provided valuable assistance in system development.

Finally, my heartfelt thanks go to my parents who always support my decisions. They provided me the most valuable thing in my life, education.

**TABLE OF CONTENTS**

<b>[CHAPTER 1]</b> .....	1
--------------------------	---

<b>INTRODUCTION</b> .....	1
---------------------------	---

1.1 Introduction.....	1
1.2 Project Overview.....	2
1.3 Project Objectives .....	2
1.3.1 The objectives of the Virtual Faculty subsystem:.....	2
1.3.2 The objectives of student academic: .....	3
1.4 Project Scope.....	4
1.4.1 Introduction Page Subsystem (IPSS) .....	4
1.4.2 Virtual Tour Sub-System (VTSS).....	5
1.4.3 Students Academic Sub-System (SASS) .....	5
1.5 Target Users .....	6
1.5.1 Administrator .....	6
1.5.2 Academic Advisor.....	7
1.5.3 Student.....	7
1.5.4 Public.....	7
1.6 Project Schedule.....	8
1.7 Report Organization.....	9

<b>[CHAPTER 2]</b> .....	10
--------------------------	----

<b>LITERATURE REVIEW</b> .....	10
--------------------------------	----

2.1 Introduction.....	10
2.2 Previous Projects.....	10
2.2.1 Previous Projects Done Using VRML .....	10
2.2.2 Previous Projects Done for Course Registration System.....	16
2.3 Development Tools.....	18
2.3.1 The Tools for VTSS Development .....	18
2.3.2 The Tools for SASS Development.....	27
2.3.3 Platform.....	30
2.4 Conclusion.....	31

---

[CHAPTER 3].....	32
------------------	----

<b>SYSTEM ANALYSIS AND REQUIREMENTS</b> .....	32
---	----

3.1 Introduction.....	32
3.2 Techniques Used .....	32
3.3 Tools and Software Analysis .....	33
3.4 Functional Requirements .....	34
3.4.1 IPSS.....	34
3.4.2 VTSS.....	34
3.4.3 SASS .....	35
3.5 Non-Functional Requirements .....	36
3.5.1 Reliability.....	36
3.5.2 Availability and Manageability.....	36
3.5.3 Security .....	36
3.5.4 Usability .....	37
3.5.5 Flexibility .....	37
3.5.6 Portability .....	37
3.6 System Model.....	37
3.7 Conclusion.....	38

[CHAPTER 4].....	40
------------------	----

<b>SYSTEM DESIGN</b> .....	40
----------------------------	----

4.1 Introduction.....	40
4.2 System Architecture .....	40
Figure 9: Data Flow Diagram for SASS.....	43
Conversely, each faculty will enter the out-faculty course into Central Database to let those students from other faculty to register it. Therefore, a student can register all courses at once. ....	44
4.3 System Flow.....	44
4.3 Database Design.....	50
4.3.1 Local Database.....	50
3) VFeedBack.....	51
4.3.2 Faculty Database .....	51
4.3.3 Central Database .....	53
4.4 Conclusion.....	53

---

[CHAPTER 5].....	54
------------------	----

<b>SYSTEM IMPLEMENTATION</b> .....	54
------------------------------------	----

5.1 Feedback Module .....	54
5.2 VTSS Module.....	57
5.3 Login Module.....	57
5.4 Add User Module (for Administrator and Staff).....	60
5.5 Approve Registration Module (for Academic Supervisor).....	61
5.6 Add/Remove Course Module (for Staff).....	62
5.7 Register Courses Module (for Student).....	62
5.8 Profile Checking Module .....	63
5.9 Result Checking Module.....	63
5.10 Database Connection Module .....	64
5.11 Conclusion.....	67

[CHAPTER 6].....	68
------------------	----

<b>SYSTEM TESTING</b> .....	68
-----------------------------	----

6.1 Unit Testing.....	68
6.2 Module Testing .....	69
6.2.1 Database Module Testing.....	69
6.2.2 Feedback Module Testing.....	69
6.2.3 VTSS Module Testing.....	70
6.2.4 Login Module Testing.....	70
6.2.5 Add User Module Testing.....	70
6.2.6 Approve Registration Module Testing.....	71
6.2.7 Add/Remove Course Module Testing.....	71
6.2.8 Register Courses Module Testing .....	72
6.2.9 Profile Checking Module Testing .....	72
6.2.10 Result Checking Module Testing.....	72
6.3 Integration Testing .....	73
6.4 System Testing.....	74
6.5 Conclusion.....	74

[CHAPTER 7].....	75
------------------	----

<b>CONCLUSION</b> .....	75
-------------------------	----

7.1 System Evaluation.....	75
7.2 Strengths.....	75

---

## CHAPTER II

7.3	Limitations .....	77
7.4	Future Enhancement.....	78

**LIST OF FIGURES**

<i>Figure 1: The architecture of v-Faculty.</i> .....	4
<i>Figure 2: Target users and the way they use to enter v-Faculty system.</i> .....	6
<i>Figure 3: Schedule for v-Faculty development.</i> .....	8
<i>Figure 4: DFD of current course registration system.</i> .....	16
<i>Figure 5: The architecture of current course registration system</i> .....	17
<i>Figure 6: System Development Model</i> .....	38
<i>Figure 7: Zero Diagram for v-Faculty</i> .....	40
<i>Figure 8: Data Flow Diagram for IPSS</i> .....	41
<i>Figure 9: Data Flow Diagram for SASS</i> .....	43
<i>Figure 10: Main System Flow</i> .....	45
<i>Figure 11: Flow Chart of IPSS Feedback</i> .....	46
<i>Figure 12: SASS Administrator Access Control Flowchart</i> .....	47
<i>Figure 13: SASS Administrator Feedback Flowchart</i> .....	47
<i>Figure 14: SASS View Exam Results Flowchart</i> .....	48
<i>Figure 15: SASS View Student Profile Flowchart</i> .....	48
<i>Figure 16: SASS Change Password Flowchart</i> .....	49
<i>Figure 17: SASS View Courses Taken</i> .....	49
<i>Figure 18: Interface for Add User Module</i> .....	60
<i>Figure 19: View the Feedback</i> .....	69
<i>Figure 20: Add user</i> .....	70
<i>Figure 21: Register course</i> .....	71
<i>Figure 22: Result of examination</i> .....	72

## [CHAPTER 1]

# INTRODUCTION

### 1.1 Introduction

The extensive growth of graphics technology brings human into a new era. The war between static 2D images and dynamic 3D virtual space seems like inevitable. Obviously, the success takes the side of 3D. The use of 3D in representing information is far better than 2D. For example, 3D can exactly point out the structure of human DNA but 2D cannot do the same thing. In the 3D virtual space, people could communicate and interact with the virtual world. This is the contribution of the Internet. Besides that, the Internet facilitates people in doing their daily transactions, which imply the idea of online. Therefore, this project incorporates the concept of virtual reality along with online system as a new idea, which is *v-Faculty*.

The *v-Faculty* takes this opportunity to develop a virtual space where provides different approach for communication. This virtual space is actually a faculty. People who visit this faculty can move around the faculty and interact with the object in it. Traditional HTML pages are not able to provide these features. Since VRML has higher interactivity, the virtual faculty could include more services. Those services are online course registration, student profile checking, student examination result checking, and student courses taken checking. These services enhance the capability of *v-Faculty* in providing sufficient services for the students.

## 1.2 Project Overview

Conventional 2D mapping could not show as much information as 3D mapping. This project applies virtual reality concept, which derived from 3D technology, in developing Faculty of Computer Science and Information Technology. It offers visitor an opportunity to explore the virtual faculty. During the exploration, visitor could get interactive information about objects and places in the virtual faculty. This could make the virtual faculty more comprehensive.

The virtual faculty could attract those who are new in this faculty. It facilitates them with the manual tour, auto tour and semi-auto tour. The visitor can follow any one of these tours to explore the virtual faculty. The student who visits the v-faculty can use another services of this system. The services are online course registration, student profile checking, student examination result checking, and student courses taken checking.

## 1.3 Project Objectives

The objectives of this project have been categorized into two parts.

### 1.3.1 The objectives of the Virtual Faculty subsystem:

1. Create a model for a faculty and put it on the Web. This model could provide realistic view of the faculty to the public.

2. Provide user with auto tour and manual tour. The auto tour will bring user from a specific starting point to the intended location while the manual tour lets user decide where to go.
3. Enable the model respond to user action. Extra information about object or places in the virtual faculty will be shown when user clicking on it.
4. Develop avatar (humanoid) in the virtual faculty. This avatar can provide extra information for user.

### **1.3.2 The objectives of student academic:**

1. Let students view their academic profile. This profile includes examination result, courses taken and total credit hours taken.
2. Enable student to change or update his/her personal data such as address and contact number.
3. Allow student to register courses. The courses are group by core subjects and non-core subjects. The staff in every faculty has to provide the information about the subjects. The student can register after the staff has entered the subject detail.
4. Provide security control over the communication process when students are accessing their personal data and academic profile.

## 1.4 Project Scope

The project consists of three main modules. These three modules are Introduction Page Subsystem, Virtual Tour Subsystem and Student Academic Subsystem as shown in *figure 1*.

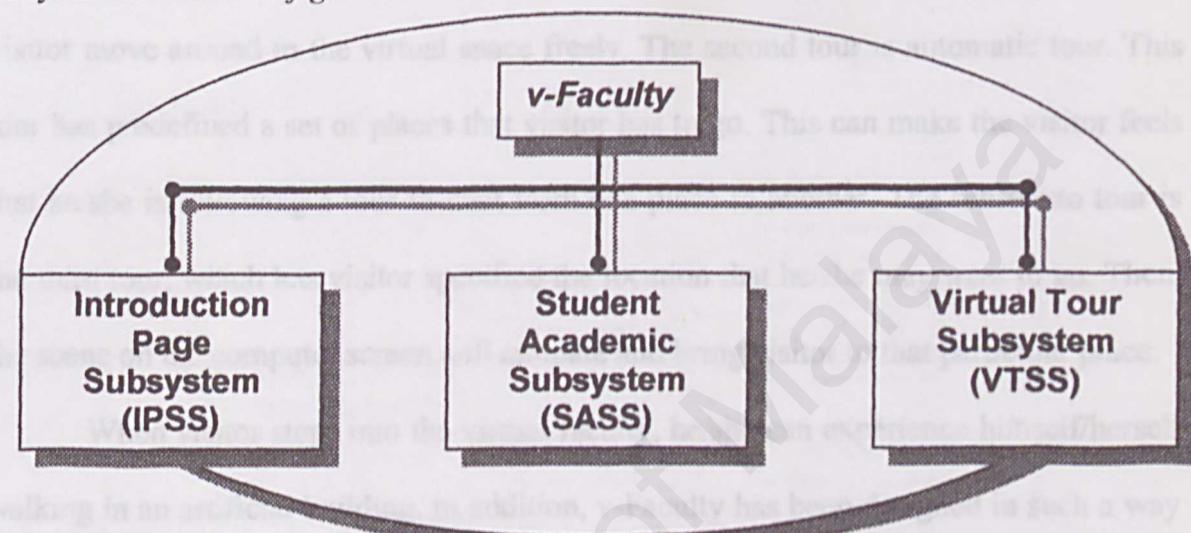


Figure 1: The architecture of v-Faculty.

### 1.4.1 Introduction Page Subsystem (IPSS)

The v-Faculty starts with IPSS before entering SASS or VTSS. In the main page of IPSS, visitors could view the information that introduces this project. Then, IPSS provides some links to interesting sites, VTSS and SASS. Furthermore, there is a feedback corner to collect comments and enquiries from the visitors. System stores these feedback in a database for future references such as introducing the Frequent Asked Questions (FAQ).

and online course registration.

Among these transactions, the main concern of SASS is online course registration. This module implements paperless concept by removing the paper forms in

#### 1.4.2 Virtual Tour Sub-System (VTSS)

The visitors could access VTSS pages through IPSS. The VTSS displays the virtual model of a faculty, Faculty of Computer Science and Information Technology. It facilitates the visitor with three tours. The first tour is manual tour, which simply lets visitor move around in the virtual space freely. The second tour is automatic tour. This tour has predefined a set of places that visitor has to go. This can make the visitor feels that he/she is following a tour to visit from one place to another. The semi-auto tour is the third tour, which lets visitor specified the location that he/she may want to go. Then, the scene on the computer screen will animate and bring visitor to that particular place.

When visitor steps into the virtual faculty, he/she can experience himself/herself walking in an artificial building. In addition, v-Faculty has been designed in such a way that the virtual space is identical to the real environment. Every object in the virtual space contains the characteristic of the actual object in terms of location, color, shape and size. When visitor clicks on the object, correlated information will be prompted out.

#### 1.4.3 Students Academic Sub-System (SASS)

Students have to login from the IPSS to SASS. This module is mainly developed for students to do some of their common transactions on-line. These transactions are student examination result checking, courses taken checking, student profile checking, and online course registration.

Among these transactions, the main concern of SASS is online course registration. This module implements paperless concept by removing the paper forms in

traditional course registration system. It also applies the architecture of distributed network system to reduce the impact of centralized processing system. Therefore, the data management and processing time could be improved enormously. Nevertheless, this system provides a high flexibility in facilitating students to complete their registration.

## 1.5 Target Users

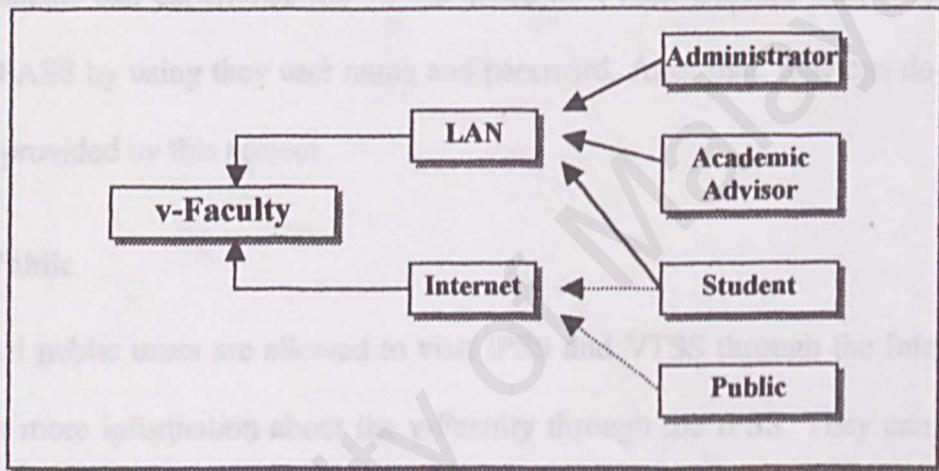


Figure 2: Target users and the way they use to enter v-Faculty system.

### 1.5.1 Administrator

The administrator is the authorized person chosen from each faculty. He needs to maintain the user list and check the availability of lecturers' room. He can remove the user who is no longer using this system from the list. Then, when a lecturer resigned or shifted, he has to update the database of the room numbers. Besides, he is responsible to reply all enquiries from the visitors of IPSS.

### 1.5.2 Academic Advisor

In SASS, the job of academic supervisor is to approve the course registration of his/her students. Therefore, SASS would not update the database without the approval of academic supervisor.

### 1.5.3 Student

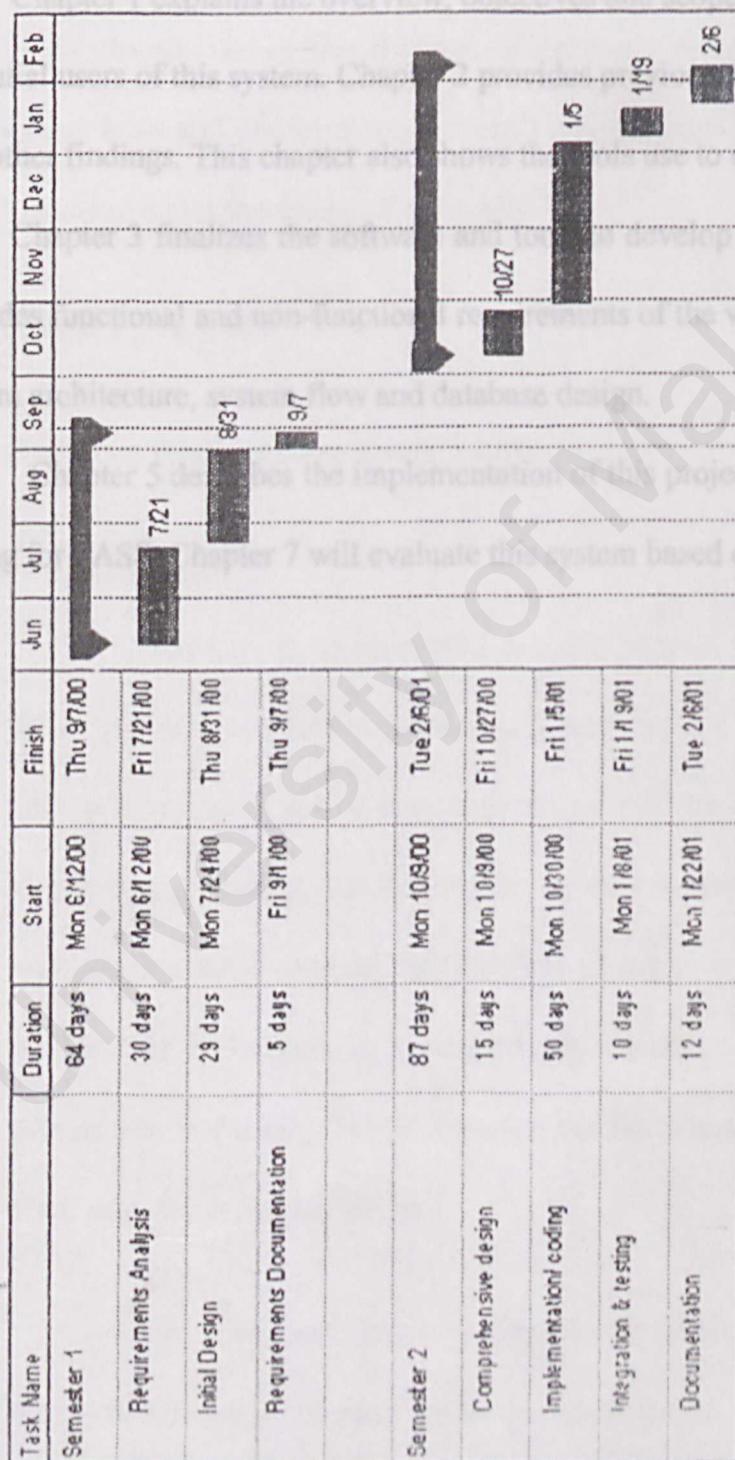
Students can experience the virtual space in VTSS. Besides that, they can login into the SASS by using they user name and password. After that, they can do the online services provided by this project.

### 1.5.4 Public

All public users are allowed to visit IPSS and VTSS through the Internet. They could get more information about the v-Faculty through the IPSS. They can send their comments to the feedback corner in order to express their opinion.

## 1.6 Project Schedule

*Figure 3* shows the time schedule when we were developing our system.



*Figure 3: Schedule for w-Faculty development.*

## 1.7 Report Organization

This report contains the following chapters:

Chapter 1 explains the overview, objectives and scope of this project as well as the potential users of this system. Chapter 2 provides previous projects done by other people and other findings. This chapter also shows the tools use to develop the v-Faculty.

Chapter 3 finalizes the software and tools to develop v-Faculty. This chapter also includes functional and non-functional requirements of the v-Faculty. Chapter 4 includes system architecture, system flow and database design.

Chapter 5 describes the implementation of this project while chapter 6 covers the testing for SASS. Chapter 7 will evaluate this system based on the achievement.

Smart City is an integrated decision support system & data manager for urban planning and landscaping. Smart City is a computer visualization tool that will represent urban environments. It will be designed to be capable of displaying, exploring, and sharing the urban environment easily and quickly. It will provide three-dimensional graphics of urban environment driven by data. Within this computer generated city environment, the user can view the ground level of a city, "walk" between the buildings, and view large city areas from user-defined aerial height.

Smart City will preserve the characteristics of urban environmental elements that are of importance to the investigation. Smart City, like Stewart's

## [CHAPTER 2]

### LITERATURE REVIEW

#### 2.1 Introduction

This chapter shows the findings of previous projects done by other people, development tools and platform description. These findings will provide the knowledge and information about the design of v-Faculty.

#### 2.2 Previous Projects

##### 2.2.1 Previous Projects Done Using VRML

###### 2.2.1.1 Smart City

Smart City is an integrated decision support system & data manager for urban planning and landscape design. Smart City is a computer visualization tool that will represent urban environments. It will be designed to be capable of displaying, exploring, and altering the urban environment easily and quickly. It will provide the 3-dimensional graphics of urban environment driven by data. Within that 3-D computer generated city environment, the user can view the ground level of a city, "walk" between the buildings, and view large city areas from user-defined aerial height.

Smart City will preserve the characteristics of urban environmental elements that are of importance to the investigation. Smart City, like Stewart's

definition of the visual surrogate will offer the planner or decision maker the ability to show more scenes or views, and to conduct the visual assessment under more controlled conditions. Moreover, Smart City will offer the planner or decision-makers the opportunity to add or subtract elements in a landscape, allowing them the ability to manipulate representations of existing environments or create representations of completely novel environments. Consequently, Smart City would be the bridge for better communication between the urban environment and the planners, decision-makers, or even ordinary people. For this project, as an initial implementation, ESLARP (East St. Louis Action Research Project) has completed surveys of data about housing conditions (such as good, dilapidated, demolished, and etc.), land uses, and infrastructure condition (such as sidewalk, street trees, manhole, and etc.) at a level of spatial resolution sufficient for Smart City to use to construct images of the city.

#### Development Tool: Arcview 3D

Advantage(s)of tool: The models can convert into VRML format.

#### 2.2.1.2 CyberTown ([Url: http://www.cybertown.com/main\\_iframes.html](http://www.cybertown.com/main_iframes.html))

Cybertown is a free, clean, safe community on the Internet. It is a great place for people to communicate, explore and share in the magic of online 3D. Cybertown is known as the "Civilization for the Virtual Age" - a futuristic, immersive society accessible via the Internet. Cybertown citizens use personalized 3D avatars to represent themselves and they can own free personal

3D homes with virtual pets, hold jobs, form clubs, shop in the virtual mall, dance in the nightclub, play games in the Casino and Arcade, get free Cybertown e-mail addresses, attend live celebrity and author chat events and more. Citizens are actively involved in the social structure of the community and with each other, making Cybertown a true community that is created by its own residents.

You earn virtual "city cash" by visiting Cybertown everyday. Your 3D house can be bought and upgraded. There is a whole social and economic structure in Cybertown and you can participate in it with other people from all over the world. You can do this across multiple Cybertown Colonies, all with their own special and unique identity.

Cybertown features technological advances like "shared-state" environments where your actions with objects will be seen simultaneously by people in the same environment from anywhere in the world. For example: when you move the sofa in your living room, your guests will see where you put it and know where to sit. You can secure your house so that only your friends can visit. The identity established by your avatar - which can be completely customized - is also maintained from community to community.

Citizens can create their own customized objects either for their own use or for trade. Citizens can create custom avatars with the new blaxxun Avatar Studio software available for sale at: <http://www.blaxxun.com>. Soon citizens will

be able to create, sell and trade customhouses or other living abodes as well as custom avatars. Some of the merchants selling "real" goods in the community will give discounts for earned community credits so your credits can go towards purchasing real-world goods and services as well.

The purpose of Cybertown is the harmonious interchange of ideas and information between community members and between Colonies and to give you something worthwhile to create. So, make Cybertown your community and be a pioneer in the next big step in global communication and interactivity!

#### 2.2.1.3 PhoNet ([Url: http://www.co.umist.ac.uk/~xtian/gd97/paper/PhoNet.html](http://www.co.umist.ac.uk/~xtian/gd97/paper/PhoNet.html))

*PhoNet* is a tool for the interactive exploration of a telephone call database in three dimensions. It has been implemented as a Java applet and uses VRML to draw the 3D scene. Therefore, it can be easily embedded in an HTML document. The information displayed comes from a real telephone call database of the United States and contains 452 telephone numbers and 768 calls between them. The most effective way of displaying a network of telephone calls is by means of a graph, in order to clearly show the relationships between different telephone numbers. A phone number is a node and a set of calls between two phone numbers is a link. Every node is represented as a cube and every link is represented as a cylinder of different size and color, accordingly to the number of calls in the set. The application computes the graph layout, builds the VRML scene and reacts to user actions. These have the effect of moving the viewpoint,

altering the graph structure, modifying the 3D scene and showing the information of interest to the user. The 3D scene is coupled with a 2D interface with labels and buttons that help in performing those tasks that could not be otherwise achieved in a user-friendly manner. A 3D approach was chosen to investigate the use of 3D interfaces when dealing with large amounts of interconnected data, in the hope that an added dimension would make the visualization clearer. The choice of Java and VRML would allow platform independence and Internet distribution.

The application of 3D graphics to the visualization and exploration of a database of highly interconnected data has proven to be not only feasible but valid for easing the understanding of the relationships between the objects and for discovering aggregations. Implementing ad-hoc navigational aids can greatly reduce the difficulties encountered when exploring a 3D world with 2D devices like the mouse and the computer screen. The use of VRML and Java has allowed access to the application from any computer connected to the Internet without the need for specialized equipment. PhoNet is one of the first examples of 3D information visualization on the Web. The few technical problems that still remain will be addressed if the non-standard VRML extensions will be accepted by the VRML community and implemented in more browsers, making platform independence a reality. A video showing PhoNet in use won the joint first prize in the video section of the Graph Drawing Competition held in conjunction with the 5th Symposium on Graph Drawing, Rome, September 1997.

The PhoNet applet can be found at the following URL: <http://bounce.to/xres>

#### 2.2.1.4 Planet 9 Studios (Url: <http://www.planet9.com>)

Planet 9 Studios is a 3D content company focused on providing real business solutions for the Internet. They have produced over 200 virtual worlds for a variety of applications such as marketing, advertising, product visualization, training, architectural simulation, military visualization and entertainment. Their animation is used for TV, commercials and real estate and is constantly incubating new software products for companies and helping them to reach market.

Over the past eight years, Planet 9 Studios has grown from a computer animation firm to the leading provider of 3D content on the Internet. The opportunity to build in cyberspace came in March of 1995, before 3D browsers were even available on the Internet. Planet 9 Studios created a low polygon model of the South of Market area of San Francisco. When 3D Internet browsers such as Intervista's Worldview became available in August of 1995, they launched their first virtual world, VirtualSOMA.

Planet 9 Studios built the first commercial VRML worlds on the Internet. Our clients include Intel, Microsoft, IBM, Softbank, Toshiba, Ziff

Davis, Nippon Telephone and Telegraph, and many other clients looking to

expand their Internet and Intranet capabilities.

Known as a "cyber architect" by the writer and VRML evangelist,

Mark Pesce, David Colleen (founder of Planet 9 Studios) continues to set

new precedents in building virtual worlds and virtual cities. Planet 9

Studios works closely with technology developers such as IBM, Fujitsu,

Platinum, Animation Science, and Noegenesis to assure online users of a

great 3D experience.

## 2.2.2 Previous Projects Done for Course Registration System

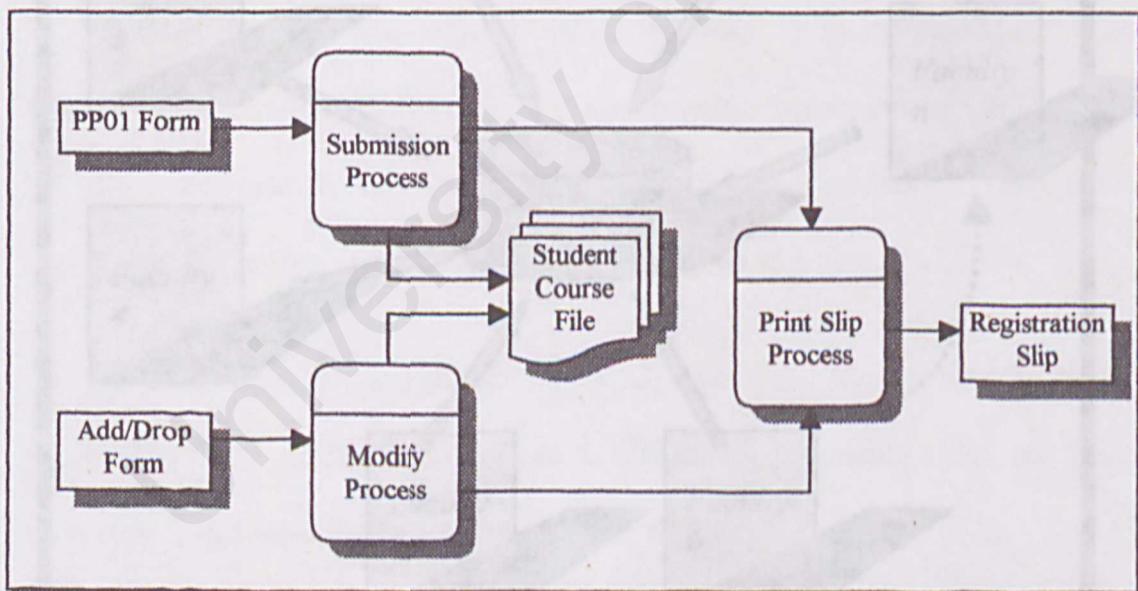


Figure 4: DFD of current course registration system

This system needs the students to get the PP01 Form from their office. Then, they have to fill up the name and code of the courses they want to register. After that, they have to get the approval from the academic supervisor. Next, they submit the PP01

to the office. The staff in the office will enter all data into computer and update it into a central database. Last, the staff prints the registration slip for the students. If the student wants to register another course, he has to fill up the Add/Drop Form. The staff will print another registration slip for the student as an up-to-date copy. The process is shown in figure 4.

### 2.3 Development Tools

#### 2.3.1.3 The Architecture of Current System

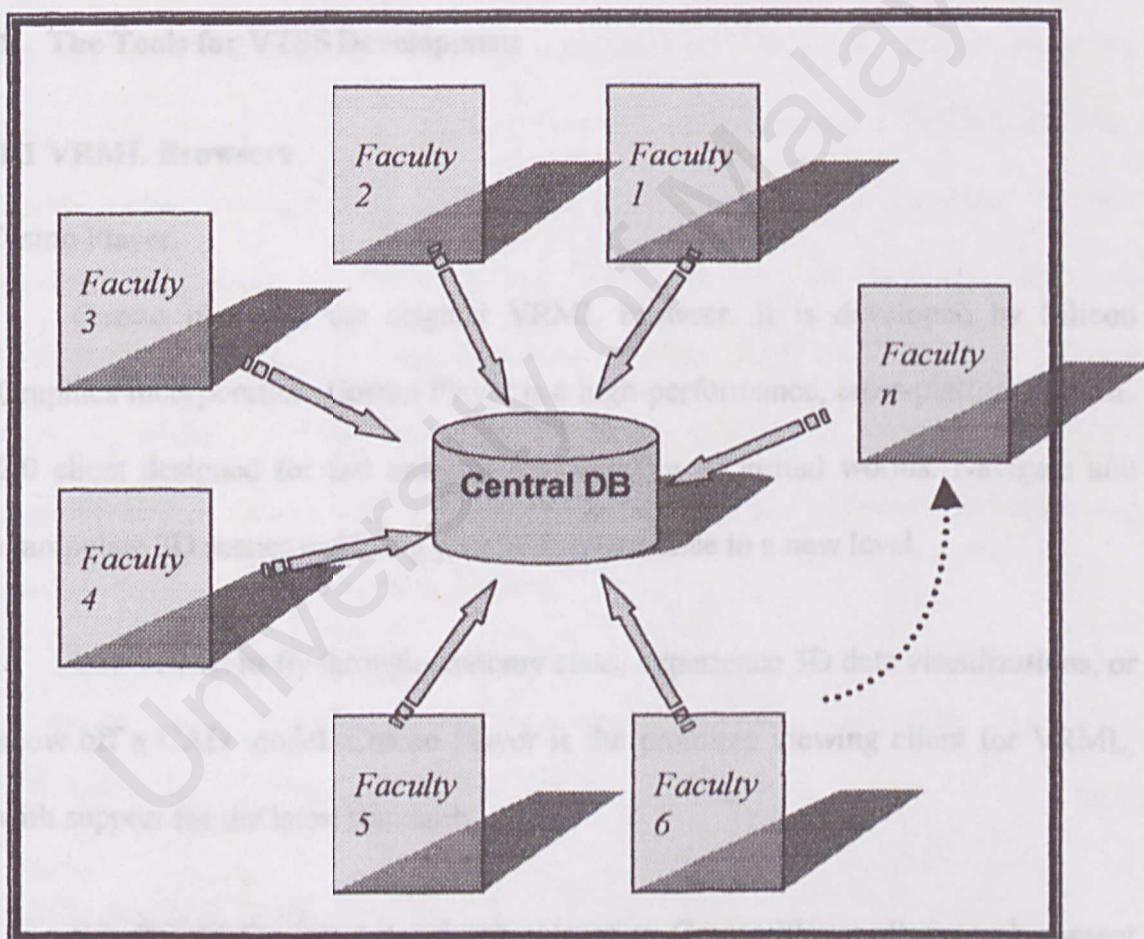


Figure 5: The architecture of current course registration system

Figure 5 shows that each faculty uses its own application to establish the connection to the database. If there are ten faculties, ten connections will be established. After that, the staff in the office can create, modify or delete the students' record. This

mechanism does not involve complex database processing. It implements the centralized processing system, which all data is stored in one place. The use of this system is limited to local area network. Student does not have the right to access this system. Therefore, it is very secure. However, the central database must have very good ability in data management. Otherwise, the database will be corrupted by the data.

## 2.3 Development Tools

### 2.3.1 The Tools for VTSS Development

#### 2.3.1.1 VRML Browsers

##### 1) Cosmo Player

Cosmo player is the original VRML browser. It is developed by Silicon Graphics Incorporated. Cosmo Player is a high-performance, cross-platform VRML 2.0 client designed for fast and efficient viewing of virtual worlds. Navigate and manipulate 3D scenes and bring your Web experience to a new level.

Use VRML to fly through anatomy class, experience 3D data visualizations, or show off a CAD model. Cosmo Player is the premiere viewing client for VRML, with support for the latest standards.

Whether on the Internet or in an enterprise, Cosmo Player allows web content creators and applications developers to add visual and multimedia elements to their work.

## 2) Cortona VRML Client

Parallel Graphics Incorporation develops Cortona VRML Client. Cortona is a free add-on for Microsoft Internet Explorer and Netscape Navigator, capable to browse virtual worlds written in VRML.

Highly optimized for complex VRML rendering, performance and ease of use, Cortona is the smallest and fastest available plug-in browser, which allows users to view immediate changes in 3D scenes. It provides full VRML 97 support, including Java language and JavaScript language support. It also provides ParallelGraphics' VRML extensions support: NURBS, Spline and keyboard sensor. The other supports provided by Cortona are Realvideo and Realaudio, Macromedia Flash animation, Drag and Drop support, External Authoring Interface (EAI) and 2<sup>nd</sup> EAI for Internet Explorer, VRML Automation Interface (Automation Interface for managing VRML scene) based on the ActiveX Automation Technology, and software renderings and hardware accelerators DirectX and OpenGL.

It provides additional opportunities for the detection and resolution of object collisions in 3D scenes. It also provides Automatic component installation technology. It is optimized for Intel Pentium III processor.

### 2.3.1.2 3D graphics Tools

#### 1) 3D Image Cube 2.0

3D ImageCube 2.0 makes it easy to create electronic picture cubes from digital photos. The easy-to-use interface guides user through the process of selecting a set

of digital images and placing them on the faces of a cube or carousel. The resulting interactive cube can be manipulated using the mouse or with a set of simple controls. 3D Image Cube 2.0 provides three sizes for cubes and carousels. It is able to link each face to a separate URL. It provides visual interface for creating cubes and carousels. It is a simple yet elegant image editor. It provides auto processing of images for bandwidth efficiency and mini 3D controls. It targets windows and frames support for hyperlinks. Yet, it is an easy FTP feature for publishing results.

## 2) 3D ImageScene

3D ImageScene lets user choose several .gif and .jpg graphics to create an interactive 3D scene. It provides easy-to-follow wizard for user to create the scene. There are several options to choose colors, spacing, and captions. The results are saved in Web-standard HTML and VRML formats, so you can immediately see and interact with your scene using your Web browser and a VRML 2.0 viewer. (A free VRML viewer is available at the developer's Website.) User can revisit the process at any time to make additions and modifications to the new scenes. Six templates (highway, browsing room, PhotoGoRound, photo album, art gallery, and simple landscape) make it even easier to create interactive, 3D scenes based on the digital photos -- for Web publishing or just for fun. User can run this evaluation edition five times.

### 3) 3DEM

3DEM for Win95/98 and WinNT will produce three dimensional terrain scenes and flyby animations from a wide variety of freely available data sources including:

- USGS Digital Elevation Model (DEM) files
- USGS Spatial Data Transfer Standard (SDTS DEM) files
- USGS Global 30 Arc Second Elevation Data Set (GTOPO30 DEM) files
- NOAA Global Land One-km Base Elevation (GLOBE DEM) files
- NASA Mars Digital Topographic Map (DTM) files
- Any topographic data file organized by rows and columns of elevation data
- Internally generated Mandelbrot fractals

3DEM can merge multiple DEMs to provide high-resolution overhead maps and 3D projections of large surface areas, limited only by the computer's memory. Geographic coordinates (latitude and longitude) are shown on all overhead map displays. Both Lat-Lon and UTM coordinates are supported, allowing display and measurement of position to high accuracy. Global Positioning System (GPS) receiver waypoints, routes, and tracks can be read via serial interface and displayed on 3D images and flybys of the terrain, allowing visualization of the path of a trek through the wilderness.

3DEM uses the SGI/Microsoft OpenGL libraries for high speed 3D rendering. 3DEM will render 16 or 24 bit color three dimensional projections, red-blue projections requiring red-blue 3D glasses for viewing, or color 3D projections

requiring Liquid Crystal Shutter (LCS) electronic shutter glasses for viewing. 3DEM scenes can be saved in the following formats.

- Windows Bitmap (\*.bmp, \*.dib)
- Joint Photographic Experts Group (\*.jpg, \*.jpeg)
- Zsoft Paintbrush (\*.pcx)
- Portable Network Graphics (\*.png)
- Portable Image (\*.pnm, \*.ppm, \*.pgm)
- Sun Raster Image (\*.ras, \*.sun)
- Silicon Graphics RGB (\*.rgb, \*.rgba, \*.bw, \*.sgi)
- Truevision Targa (\*.tga)
- Tagged Image File (\*.tif, \*.tiff)
- X-Pixmap (\*.xpm)
- VRML world (\*.wrl)
- Binary terrain matrix (\*.bin)
- Terragen terrain (\*.ter)

3DEM allows low-resolution real-time flyby of DEM landscapes using OpenGL.

Keyboard controls are used to climb, dive, and turn in space above the DEM surface.

The path through space is recorded in memory during flight, allowing subsequent creation of a full resolution mpeg animation along the flight path. Real-time flyby animations can be created in the following formats

- Flyby animation AVI (\*.avi)
- Flyby animation MPEG (\*.mpg, \*.mpeg)

3DEM provides an intuitive user interface, high reliability, and the very finest 3D images and flyby animations created from freely available terrain data.

#### 4) Dome 4.6

DOME is a freeware DOS utility for generating the coordinates of a geodesic dome or sphere. It uses command-line input. It can generates data files in the following output formats:

- *Drawing Exchange Format* (DXF) for import into CAD packages
- *Persistence of Vision Raytracer* (POV) format for creating photo-realistic images.
- *Virtual Reality Mark-up Language* (WRL) files for interactive Web viewing.
- *ASCII Report* (DAT) format including chord, axial and face angle data.
- *ASCII Import Format* (PRN) for use with spreadsheets, etc.

It supports Tetrahedron, Octahedron & Icosahedron Polyhedron types. It could generates Symmetry Triangles and Full Spheres for Class I & II Structures. It also provides "Buckyball" Structuring (now includes all three polyhedron types) and Elliptical Geodesics. It is defines and a make file for Non-Borland C compilation (i.e. Unix, Linux, etc.). It provides the functions for calculating Point, Edge & Face Coordinates. It also provides DXFtoPOV utility.

## 5) 3D Studio Max

3D studio max 3 supports the largest developer community of any 3D application, with a huge selection of third-party plug-ins. A wide range of tools build on this flexible architecture, including advanced character building with Discreet's own popular character studio. The powerful new renderer in 3d Studio Max 3 maintains high speed while delivering superior results for photorealistic images, scenes, and special effects. While enhancing the built-in renderer, 3d Studio Max 3 also provides a connection to other advanced renderers like mental ray for additional rendering capabilities and production integration. Modeling flexibility includes interactive

creation about any point, in any context, on any face, at any frame and no limits on geometry/scene size or number of objects, cameras, lights, materials, maps, modeling history, or rendering effects. Its MeshSmooth NURMS provides intuitive subdivision surface modeling with vertex and edge weighting, and interactive control of mesh levels. Modifiers can also include gizmos for direct manipulation on parameters for fast and efficient workflow. Powerful spline-based patch modeling allows users to create complex characters from a network of splines or to convert primitives to patches for direct manipulation. Its image file format support AVI, BMP, CIN, EPS, FLC, GIF\*, JPG, PNG, RGB, RLA, RPF, TGA, TIF, YUV\*, Photoshop PSD\*, and QuickTime MOV. Its geometry file format support IGES\*, PRJ, SHP, STL, VRML, 3d studio max 3DS, 3D ASCII Scene, Adobe Illustrator AI, AutoCAD DWG and DXF, Adobe Type1\* and TrueType\* fonts (\*= import only). Furthermore image and geometry file support are both extensible. However, 3D Studio Max 3 only runs on NT platform.

## 6) Nendo

Built on Winged Edge Technologies' years of expertise developing tools for 3D artists, Nendo is the first 3D modeling and 3D painting package that is both simple enough for the novice and powerful enough for the professional. With Nendo, user can create compelling 3D objects for use in model libraries, games, print material, web sites, visual simulation, and architectural applications. Powerful

operations for modeling and 3D painting are incorporated into an interface that is simple and intuitive to use.

Model in a "digital clay" environment always works within an interface based on traditional sculpting. Use a variety of powerful modeling functions such as extrude, scale, rotate, connect, bevel, inset, mirror, and smooth to create organic as well as hard-edged objects with ease. A smart workflow incorporates context-sensitive menus that help guide users to the relevant operations for any selected element, making Nendo simple to learn, and allowing user to model quickly.

Then, with the click of a button, paint in an interactive 3D environment using familiar painting tools, quickly accessing brush options such as size, opacity, and softness, and selecting color from an expandable color picker. When done, export the creation to a variety of file formats for use in many other applications.

## 7) Arcview 3D

The ArcView 3D Analyst extension enables users to create, analyze, and display surface data. This generic surface modeling package is ideal for both the novice and the advanced user, its functionality answering the needs of those performing tasks related to surface analysis and display. Unique features of ArcView 3D Analyst include support for triangulated irregular networks (TINs) and simple three-dimensional vector geometry, as well as interactive perspective viewing.

In ArcView 3D Analyst, the most commonly used functions are accessible from pull down menus and tool buttons that are added to the ArcView GIS interface when the extension is installed.

### 2.3.1.3 Text Editors

#### 1) MED

MED is a powerful shareware text editor for programmers, currently available for the platforms Windows 9x/NT4.0/2000 and OS/2 PM.

MED's functionality is designed to be suitable for any text editing job. In addition MED provides many features that support writing and compiling source code (C/C++, Java, HTML, VRML, REXX, Ada, Modula2, Pascal, Fortran, X86/MC68k Assembler, Perl, Clipper, TeX, Tcl/Tk, Matlab, etc.), that make MED a versatile IDE for any programming language.

#### 2) VRMLPad 1.2

VRMLPad 1.2 supports automation and scripting. For example, when you are typing one character, the related keyword list will prompt out to let user choose the keyword he wants. It is a material editor, which means that it follows the VRML 2.0's rules when we are writing the codes. At the same time, we can see a tree list, which fold all the nodes using in our codes. It also provides the keyboard shortcut. User could use the inline command easily. Nevertheless, it also allows user to insert anchors and external prototypes. It provides Smart AutoComplete function and

dynamic error detection. It can do the syntax highlighting and advanced find and replace. It provides visual support for the scene tree operations. It runs the operations on resources. It is able to preview the VRML scene. It contains a publishing wizard.

### 2.3.2 The Tools for SASS Development

#### 1) Oracle 8

Oracle8i, the database for Internet computing, changes the way information is managed and accessed to meet the demands of the Internet age, while providing significant new features for traditional applications. Using solutions developed and deployed with Oracle8i, any organization can exploit the unlimited opportunities the Internet provides. Oracle8i provides advanced tools to manage all data found in today's most popular web sites — and the performance and scalability needed to support these sites and other applications. Oracle8i delivers unprecedented ease-of-use, power, and price/performance for workgroup or department-level applications. From single-server environments for small businesses to highly distributed, branch environments, Oracle8i includes all the facilities necessary to build business-critical applications. In today's fast-paced business climate, application developers must produce advanced applications quickly to meet ever-changing needs.

Oracle8i introduces additional support for Java, today's most popular and productive programming language — with a robust, integrated, and scalable Java Virtual Machine (VM) within the server. This expands Oracle's support for Java into all tiers of

applications, allowing Java programs to be deployed where they perform best —in the client, server or middle tier — without recompiling or modifying the Java code. Oracle*8i* also includes Oracle WebDB, a browser-based application development, deployment, and management toolset that make it easy and inexpensive to web-enable a business. Oracle*8i* is much more than just a simple relational data store. Having invented the relational database, Oracle continues to lead and revolutionize the database industry by introducing Oracle iFS and Oracle *interMedia* with Oracle*8i*. Oracle iFS, the Internet file system, combines the easy-to-use functionality of a file system with the advantages of all data in a single server — data integration with the cost savings of a single server. Oracle*8i* also introduces Oracle *interMedia*, which allows businesses to manage and access multi-media data, including image, text, audio/video, and spatial data.

Deployment of applications is just as critical as their functionality, with performance and manageability the key to successful implementation. Oracle*8i* provides many alternatives to the client/server implementation architecture. Oracle reduces the complexity of application management by providing an architecture that allows for thin-clients that are easy to support and maintain, and by consolidating data and logic in a single or replicated server environment. Data can be replicated to servers that are closer to users and have only the data those users need, providing better performance.

Finally, Oracle*8i* includes Oracle Enterprise Manager, the comprehensive management framework for managing the Oracle database and application environment. Oracle Enterprise Manager includes an easy-to-use centralized console, a rich set of management tools, and the extensibility to detect and solve problems. Oracle Enterprise

Manager also includes several administrative applications for performing day-to-day tasks for databases and applications, such as scheduling backup routines. Based on the world's leading database technology, Oracle8i is designed for efficient application development, superior content management, and successful application deployment on the Internet.

## 2) MS SQL Server 7.0

The release of the new database product from Microsoft promises several new exciting features. The online analytical processing (OLAP) component of Microsoft® SQL Server™ 7.0 provides fast, efficient analysis of complex information essential to reporting, data analysis, decision support, and data modeling. The Dynamic Row-Level Locking automatically chooses the optimal level of lock (row, page, multiple page, and table) for all database operations. The query processor has been redesigned to better support the large databases and complex queries found in decision support, data warehousing, and OLAP applications.

SQL Server version 7.0 Enterprise Edition supports memory addressing greater than 4 GB, in conjunction with Windows NT Server 5.0, Alpha processor-based systems and other techniques. A database can automatically grow and shrink within configurable limits, minimizing the need for DBA intervention. It is no longer necessary to pre-allocate space and manage data structures. Concurrency, scalability and reliability are improved with simplified data structures and algorithms. Run-time checks of critical data structures make the database much more robust, minimizing

Manager also includes several administrative applications for performing day-to-day tasks for databases and applications, such as scheduling backup routines. Based on the world's leading database technology, Oracle8i is designed for efficient application development, superior content management, and successful application deployment on the Internet.

## 2) MS SQL Server 7.0

The release of the new database product from Microsoft promises several new exciting features. The online analytical processing (OLAP) component of Microsoft® SQL Server™ 7.0 provides fast, efficient analysis of complex information essential to reporting, data analysis, decision support, and data modeling. The Dynamic Row-Level Locking automatically chooses the optimal level of lock (row, page, multiple page, and table) for all database operations. The query processor has been redesigned to better support the large databases and complex queries found in decision support, data warehousing, and OLAP applications.

SQL Server version 7.0 Enterprise Edition supports memory addressing greater than 4 GB, in conjunction with Windows NT Server 5.0, Alpha processor-based systems and other techniques. A database can automatically grow and shrink within configurable limits, minimizing the need for DBA intervention. It is no longer necessary to pre-allocate space and manage data structures. Concurrency, scalability and reliability are improved with simplified data structures and algorithms. Run-time checks of critical data structures make the database much more robust, minimizing

the need for consistency checks.

### 2.3.3 Platform

#### Microsoft Windows NT

The most significant new feature in Windows 2000 may be Active Directory, Microsoft's replacement for the old Windows NT domain-controller system. For small networks, which used only a single domain under Windows NT, Active Directory might not be worth the effort to learn and configure. For larger networks that are (or should have been) organized in multiple domains, Active Directory represents a definite improvement. Unlike flat domains, Active Directory organizes the network into a hierarchical "forest," with logical units--"trees"--identified by a fully qualified DNS address. Thus, it's necessary to have a working DNS server on the LAN. Security and reliability of the operating system has also greatly improved compared to the NT version. Windows 2000 protects itself in a passive-aggressive way by allowing the overwrite, then replacing the original file. This feature protects the operating system, but might wreak havoc on third-party providers wishing to replace Windows DLLs with their own enhanced versions. That's not the only reason why Windows 2000 seems more stable; more effort appears to have been made to protect the kernel memory from overwrites or leaks. Another cornerstone of Windows 2000, and a leap forward, is device support. Both server and workstation versions have Windows 9x-style Plug and Play support for new hardware, making adding new peripherals much easier.

## 2.4 Conclusion

This chapter has shown some of the projects done by other people. From these projects, the v-Faculty may adopt some of the techniques they used to develop the VTSS and SASS. *Chapter 3 System Analysis and Requirements* shows the result of tool analysis in order to choose the most appropriate tool for the v-Faculty.

The chapter analyses the requirements of v-Faculty. The analysis will discover the requirements of end-user in order to study the feasibility of this system. The acceptability of the system after it has been delivered depends on how well it meets the user's needs and support the work to be automated.

## 3.2 Techniques Used

Several techniques have been used to collect the information needed. These techniques include brainstorming, Internet surfing, book reading and interview. At the initial stage, a brainstorming session is held to gather information about v-Faculty based on personal experience and knowledge. The first idea come out was from the name of this project, which based on the 3D representation of faculty. Then, many refined ideas are emerged. These ideas have defined the basic functions that v-Faculty should perform.

Besides that, the Internet provides many other sources about this project like electronic books, tutorials, news group, mailing list, similar projects and other related information. The knowledge from the books and tutorials is used to develop personal skills in different fields. The news group and mailing list is used to exchange information.

## [CHAPTER 3]

# SYSTEM ANALYSIS AND REQUIREMENTS

### 3.1 Introduction

This chapter analyses the requirements of v-Faculty. The analysis will discover the requirements of end-user in order to study the feasibility of this system. The acceptability of the system after it has been delivered depends on how well it meets the user's needs and support the work to be automated.

### 3.2 Techniques Used

Several techniques have been used to collect the information needed. These techniques include brainstorming, Internet surfing, book reading and interview.

At the initial stage, a brainstorming session is held to gather information about v-Faculty based on personal experience and knowledge. The first idea came out was from the name of this project, which carried out the 3D representation of faculty. Then, many related ideas are gathered. These ideas have defined the basic functions that v-Faculty should perform.

Besides that, the Internet provides many other sources about this project like electronic books, tutorials, news group, mailing list, similar projects and other related information. The knowledge from the books and tutorials is used to develop personal skills in different fields. The news group and mailing list is used to exchange information.

The library is another source for the information. In order to obtain the skills of system development, the traditional reading strategy helped a lot. The books borrowed from the library illustrate many techniques used in virtual world development.

After a few interviews have been conducted, the interviewees explained the expected flow of this system. These interviewees are system user, system administrator, and system developer. They have provided a lot of information about the user requirements, which can improve the performance of v-Faculty.

### 3.3 Tools and Software Analysis

This analysis concerns about the usability and familiarity of the tools.

First, this system will use Windows NT as the operating system for the whole project development.

Second, it needs to be put on the Web. Current 3D technology that supports this function is Virtual Reality Modeling Language (VRML). The VRML version 1.0 is obsolete because lack of capabilities for communication and interactivity. Therefore, this system will use VRML 2.0 as the backbone of the virtual faculty. User needs to use a VRML browser to view the VRML code. The browser that gives the best support for all VRML 2.0 codes is Cosmo Player.

Third, the system needs a 3D graphics tool to develop model. The model must be able to export into VRML 2.0 format. Furthermore, it must be compatible with NT platform. Therefore, 3D Studio Max is chosen for modeling purposes.

2. Forth, another stage of system development involves code editing. VRMLPad is the user-friendliest editor to modify the exported codes. The exported VRML codes will be enhanced by adding sound effect, texture, animation and sensors.

Last, this system needs a database for data storing. Oracle 8i with high capabilities in data management would be the first choice.

### 3.4 Functional Requirements

1. The following are the functional requirements for each of the sub-systems:

#### 3.4.1 IPSS

1. Provide users with information regarding our thesis topics.

2. Receive feedback from visitors.

2.1 Submit comments or questions by pressing a button

2.2 Store all comments and questions in a text file

2.3 Update and refresh the feedback page to let visitor view their comments or questions immediately

2.4 Provide a link to go back to IPSS main page

3. Provide links to the VTSS and SASS.

3.1 For the SASS, user has to login by using their user ID and password

4. Provide links to other VRML and faculty web pages.

#### 3.4.2 VTSS

1. Provide options for users to choose the tour mode preferred that is manual or auto.

2. Provide additional information upon users' action (click or drag).
3. Provide a link to let user go back to IPSS main page.
4. Provide an avatar to answer user's question for the location of places in the faculty.

4.1 User will be given a list of answer to choose

4.2 The system must react to the choice made by user

### 3.4.3 SASS

1. Only authorized user can enter this page
2. User can make their choice to check result, check personal profile, check courses taken or register new courses.
3. System should also allow users, if authenticated, to update or change their personal data.
4. Provide the registration form for users to register their courses.
  - 4.1 System have to retrieve data from database to check the courses taken first
  - 4.2 User can add or modify the courses they want
  - 4.3 User can update the modified information
  - 4.4 User can know the status of his/her registration
5. An administrator/super user option must be available for the administrators to maintain the system.
6. Provide a corner for administrator to maintain the user list.
7. Provide the function to let administrator change the lecturers' name according to room numbers.

8. System must provide a button to let user logout.
  - 8.1 After user logout, bring user to IPSS main page
9. System should log every user login or logout the system and keep it in a text file for security purposes.
10. Academic advisor could use SASS to approve the registration of the students.
  - 10.1 Update the data to the real database

### **3.5 Non-Functional Requirements**

#### **3.5.1 Reliability**

This system should design in such a way that process errors are avoided or trapped before they result in output error. It shall not cause any unnecessary actions of the overall environment.

#### **3.5.2 Availability and Manageability**

This system shall be available to the users anywhere and any time to ensure that the operation and services are running smoothly. It shall be capable to let user manage and operate it easily.

#### **3.5.3 Security**

This system should ensure the use of the information is legal. Only authorized user can modify the system. The authenticated users shall have the access right to view or modify the data in the database. This system shall include the login procedure to

identify the user who is using it. Communication with the system needs to be established with validation control to ensure authenticity of the data transfer.

#### 3.5.4 Usability

This system must provide documentation or guideline to teach user how to use it. At least, intuitive interface shall present in such a way that makes user feel easy to use.

#### 3.5.5 Flexibility

This system shall be capable for future expansion. System should allow the integration with other systems and new technologies.

#### 3.5.6 Portability

This system must enable its application to operate on various platforms regardless of manufacturer or operating system.

### 3.6 System Model

The figure 6 shows that this project adopted the waterfall and prototyping model for system design. At the beginning stage, the requirements of the system are identified. It included techniques used in collecting the information. After that, these requirements went through a proper analysis. This analysis included the tools that need for system development and user requirements.

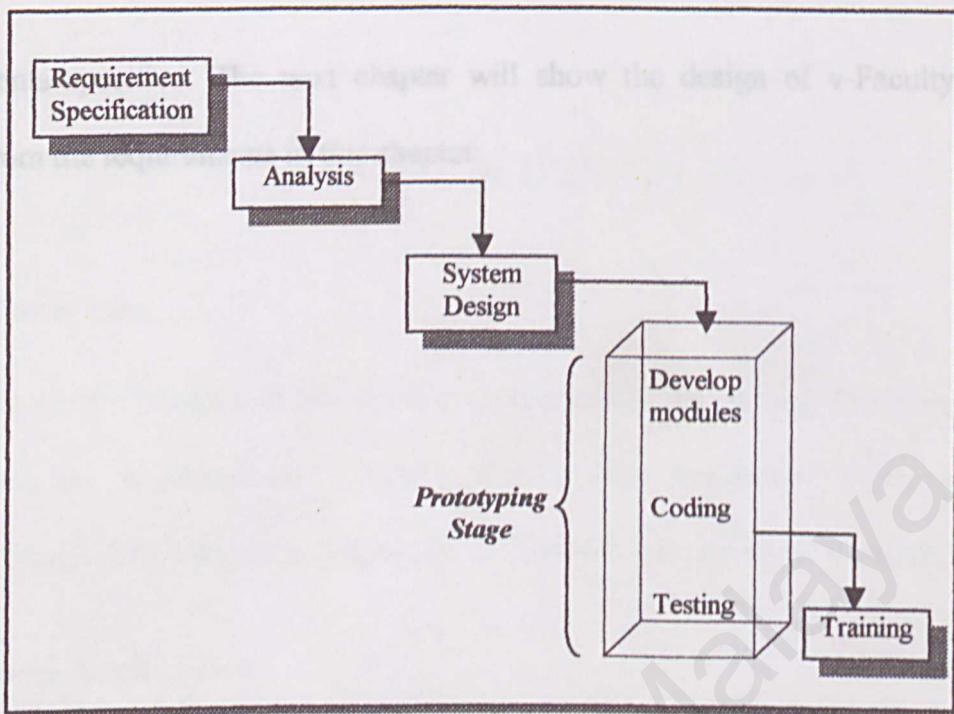


Figure 6: System Development Model

The prototyping method is used when the development reaches the stage of application or module design. It focuses on creating the prototypes for each proposed module that is specified in the scope of the project. The prototypes will be tested iteratively in order to discover incorrectness of the model and to rectify it.

Finally, the project provides user training for the v-Faculty user. The training program ensures the users have necessary skills to operate the system.

### 3.7 Conclusion

This chapter finalized the tools and software to be used in this project. In second part, it described the functional and non-functional requirements that required by system user. Based on the data gathered, the system will be developed according to the

requirements specified. The next chapter will show the design of v-Faculty, which derives from the requirements in this chapter.

#### 4.1 Introduction

The system design will reveal the completeness of the system. The completeness of a system can be determined by looking at the system architecture, user interface and database design. This chapter will describe all three topics above.

#### 4.2 System Architecture



Figure 7: Zoo Diagram for v-Faculty

Figure 7 shows the basic structure of the system. It consists of three parts: User and SASS.

##### 4.2.1 Introduction Page SASS (SSO)

User who enters the system will first visit the Home page that introduces the system for the first time. It is designed to look like a landing page, which created by using HTML code within the ASPX language link to VIE, and

## [CHAPTER 4]

# SYSTEM DESIGN

### 4.1 Introduction

the system design will reveal the completeness of the system. The completeness of a system can be determined by looking at the system architecture, system flow and database design. This chapter will describe all these three topics about v-Faculty.

### 4.2 System Architecture

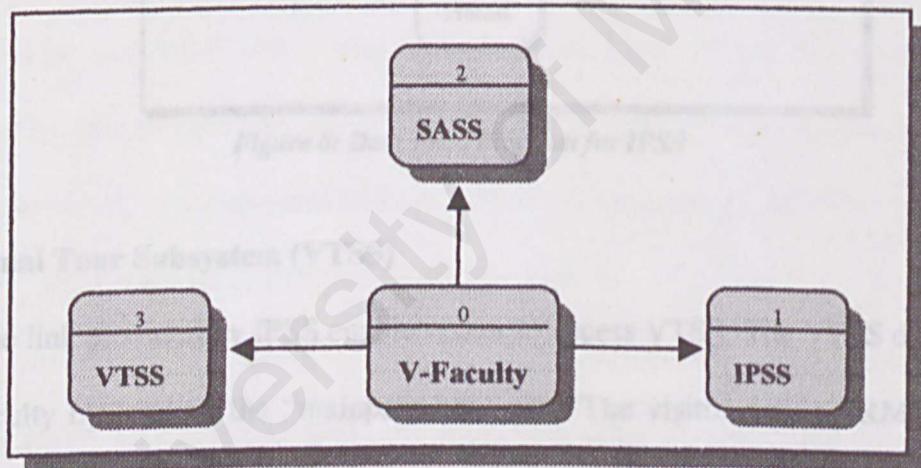


Figure 7: Zero Diagram for v-Faculty

Figure 7 shows the basic structure of the v-faculty. It consists of IPSS, VTSS and SASS.

#### 4.2.1 Introduction Page Subsystem (IPSS)

User who enters the system will come to IPSS. He/she can view the information that introduces the system in the first page. This page is called the introduction page, which created by plain HTML code. Besides that, IPSS provides links to VTSS and

SASS. There is a feedback corner embedded into IPSS to collect comments and enquiries from the user. All feedbacks will store in database. The Active Server Pages (ASP) will do the process of data storing. After an user entered comments, the feedback page will refresh to include his/her comments into the page. The process is shown in figure 8.

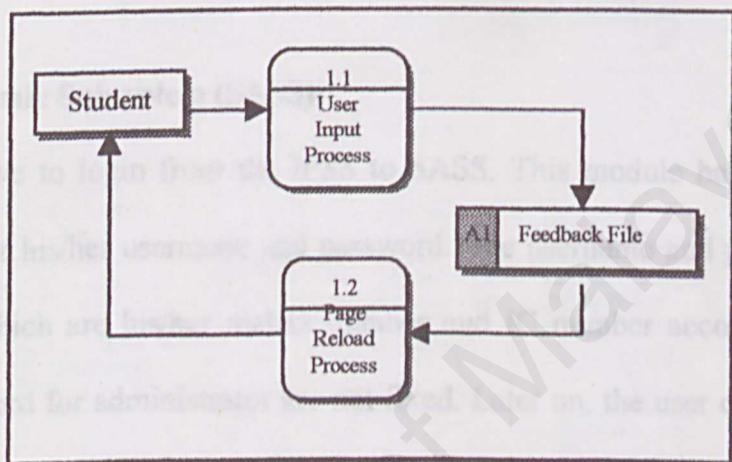


Figure 8: Data Flow Diagram for IPSS

#### 4.2.2 Virtual Tour Subsystem (VTSS)

The link provided by IPSS enable visitor to access VTSS. The VTSS displays the virtual faculty by loading the “mainpage.htm” file. The visitor needs VRML browser (Cosmo Player) to view this page.

The system uses a 3D graphics tool (3D Studio Max) to model out the whole building. Then, the model will export into VRML codes. Next, these codes will be edited by text editor (VRMLPad) to provide sound effects, sensors, textures and animations. After that, the edited VRML codes will embed into HTML page. When visitor request for this page, system will display the page.

By default, VTSS is in manual tour. There are 3 buttons for visitor to switch between manual, auto and semi-auto tour. When visitor is maneuvering the virtual world, system will show information related to the location. This information is stored in the database. Therefore, the system needs an administrator to maintain data.

#### 4.2.3 Student Academic Subsystem (SASS)

The users have to login from the IPSS to SASS. This module has provided a login to let user enter his/her username and password. The username and password for student are fixed, which are his/her matrix number and IC number accordingly. The username and password for administrator are not fixed. Later on, the user could change the password by entering the Change Password Process. This process needs user to enter his/her old password, new password and confirmed new password. All these passwords and username will be stored in a special local database.

If a student enters SASS, he/she could do the checking of personal data, examination result and courses taken. At the same time, he/she could also register courses. ASP and Oracle will do all these actions. ASP serves as an agent to communicate between user and database.

In SASS, there are two main databases for course registration system, which are Faculty Database and Central Database. Every faculty could maintain their own Faculty Database while there is only one Central Database in central. Every faculty will enter the in-faculty course into the Faculty Database to let its students register it.

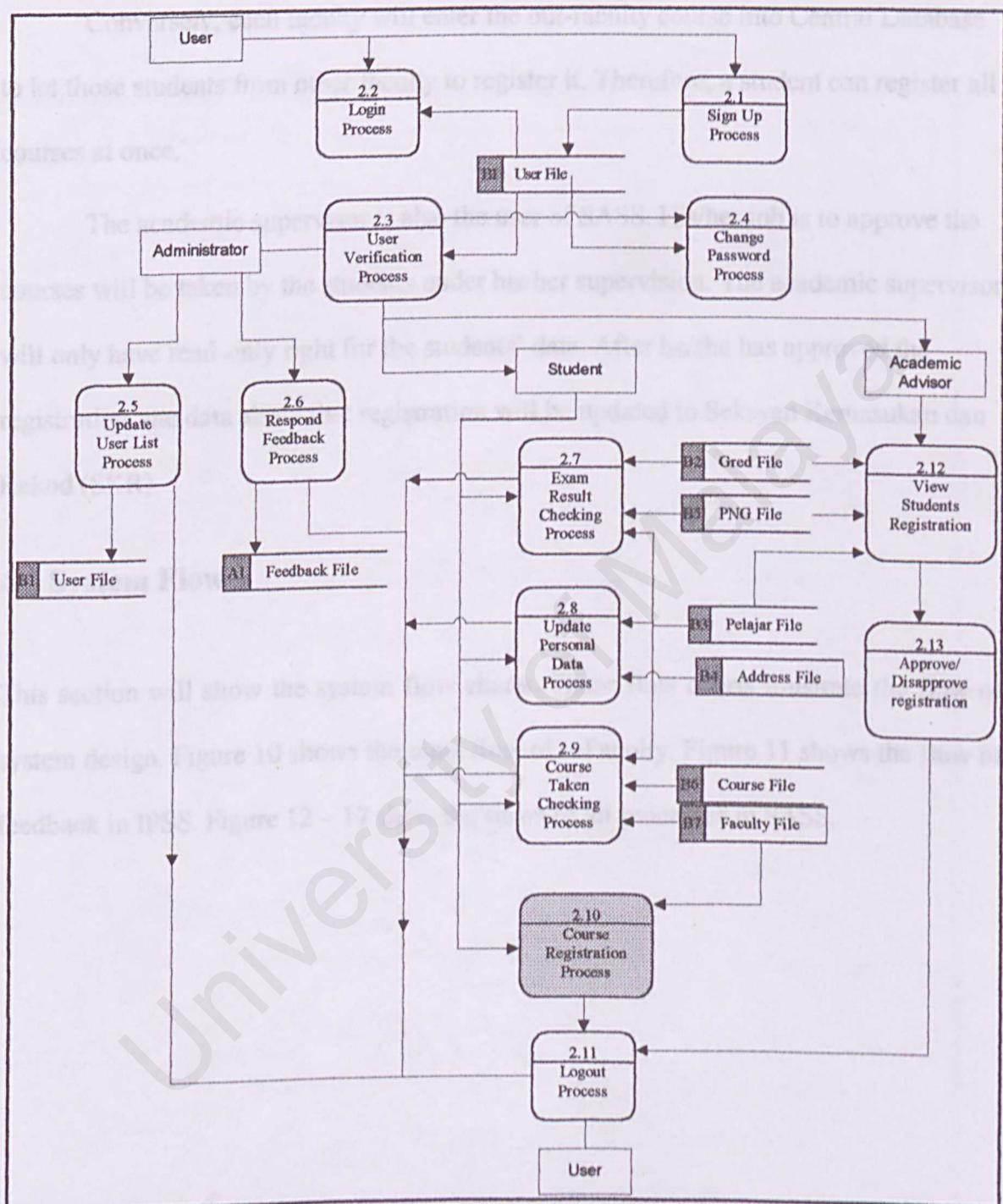


Figure 9: Data Flow Diagram for SASS

Conversely, each faculty will enter the out-faculty course into Central Database to let those students from other faculty to register it. Therefore, a student can register all courses at once.

The academic supervisor is also the user of SASS. His/her job is to approve the courses will be taken by the students under his/her supervision. The academic supervisor will only have read-only right for the students' data. After he/she has approved the registration, the data about that registration will be updated to Seksyen Kemasukan dan Rekod (SKR).

#### 4.3 System Flow

This section will show the system flow charts. These flow charts illustrate the flow of system design. Figure 10 shows the main flow of v-Faculty. Figure 11 shows the flow of feedback in IPSS. Figure 12 – 17 show the flows of all processes in SASS.

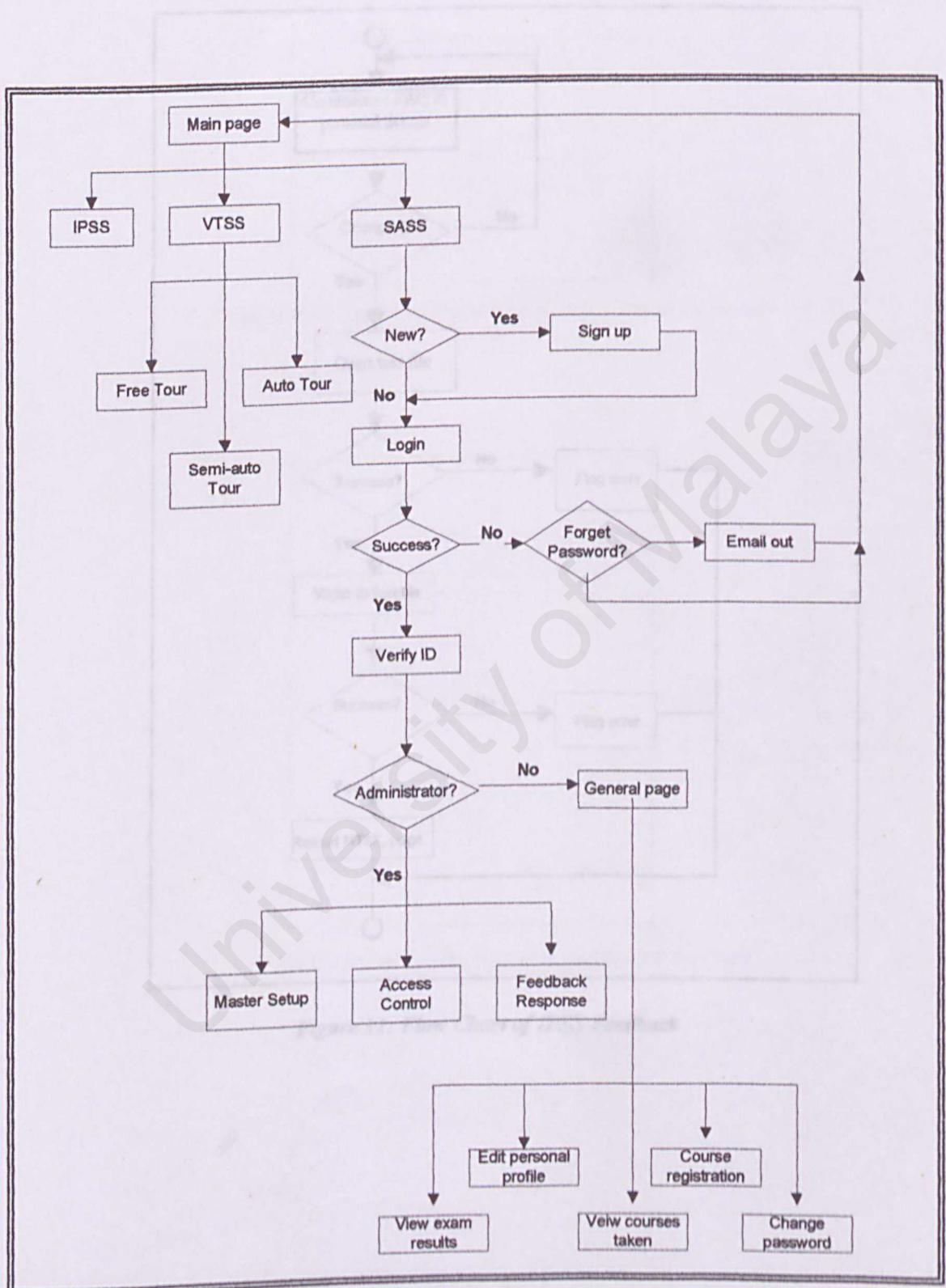


Figure 10: Main System Flow

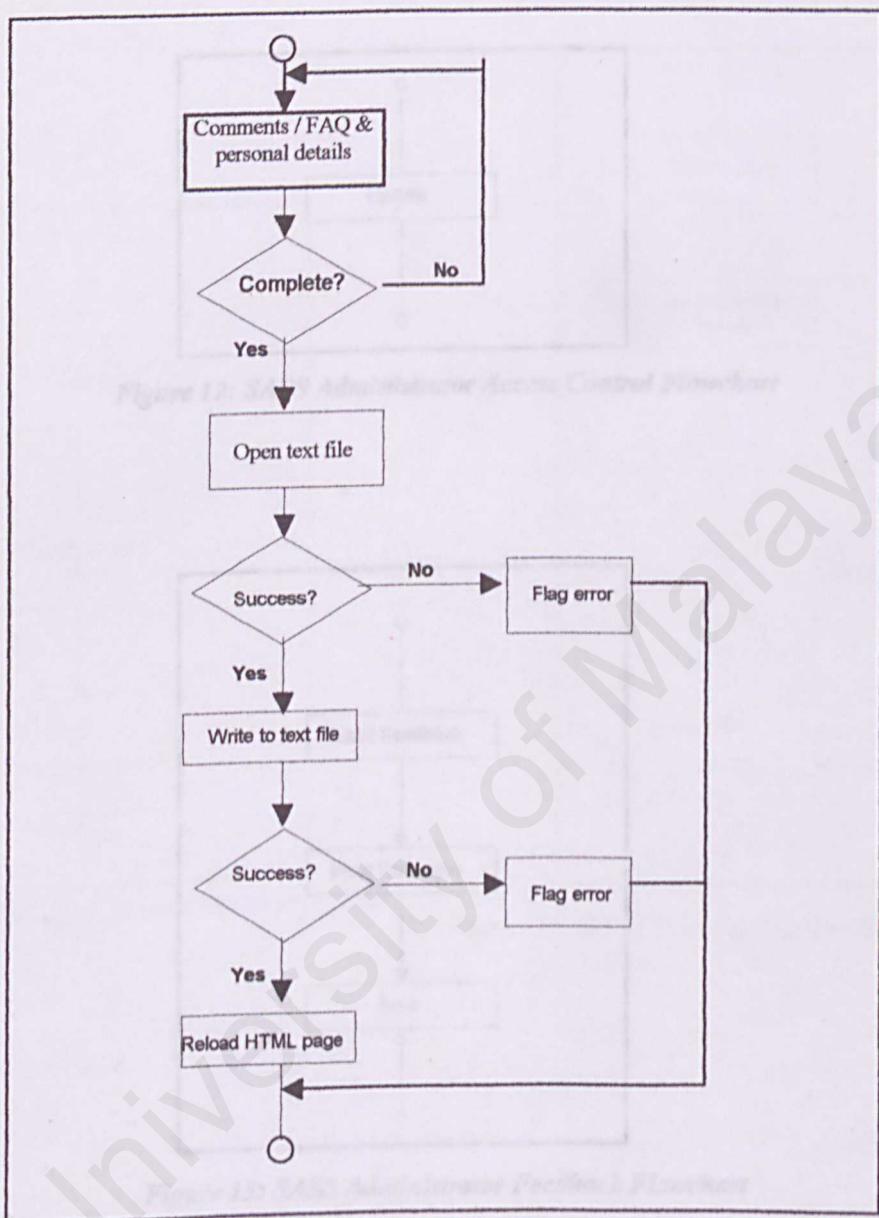


Figure 11: Flow Chart of IPSS Feedback

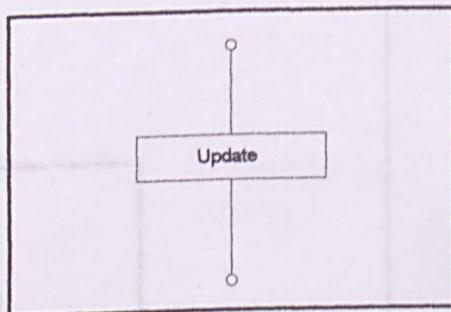


Figure 12: SASS Administrator Access Control Flowchart

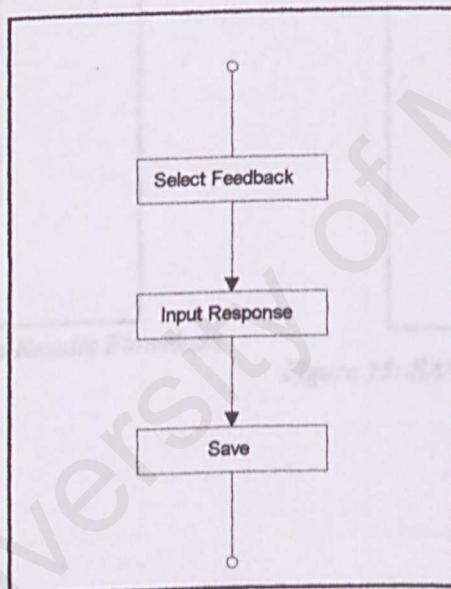


Figure 13: SASS Administrator Feedback Flowchart

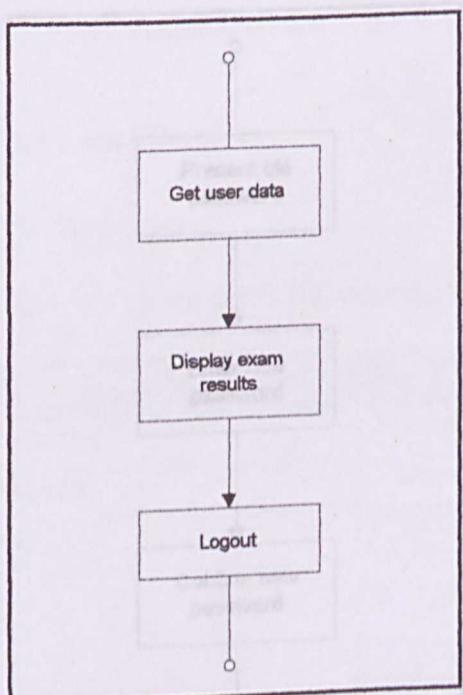


Figure 14: SASS View Exam Results Flowchart

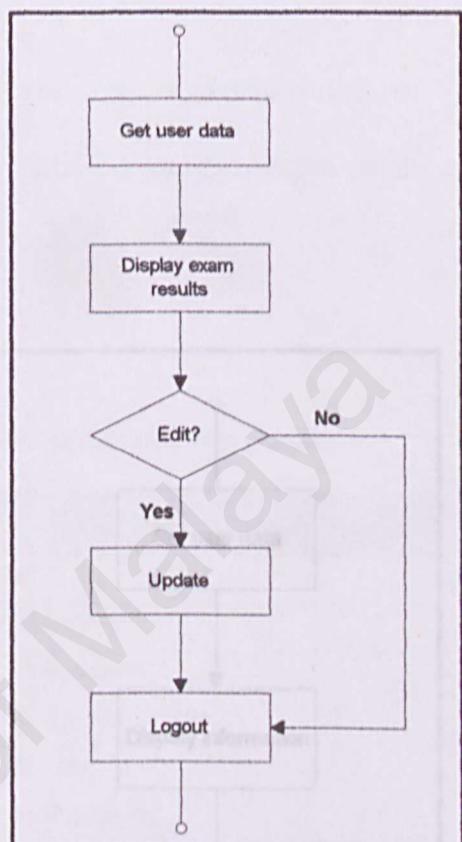


Figure 15: SASS View Student Profile Flowchart



Figure 16: SASS Change Student Profile Flowchart

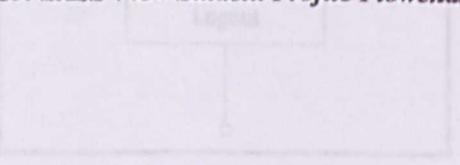


Figure 17: SASS View Courses Taken

### 4.3 Database Design

There are four main tables used in this project. The tables are located in three different local databases: student database, faculty database and the central database.

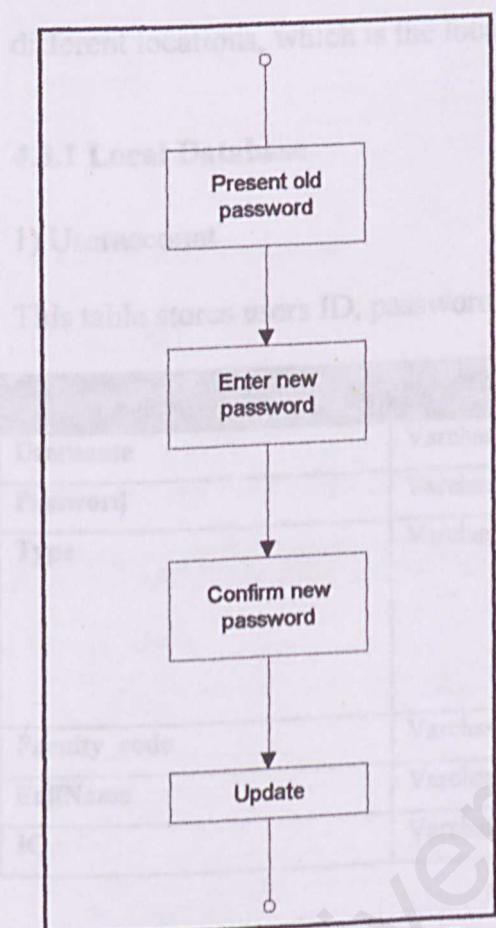


Figure 16: SASS Change Password Flowchart

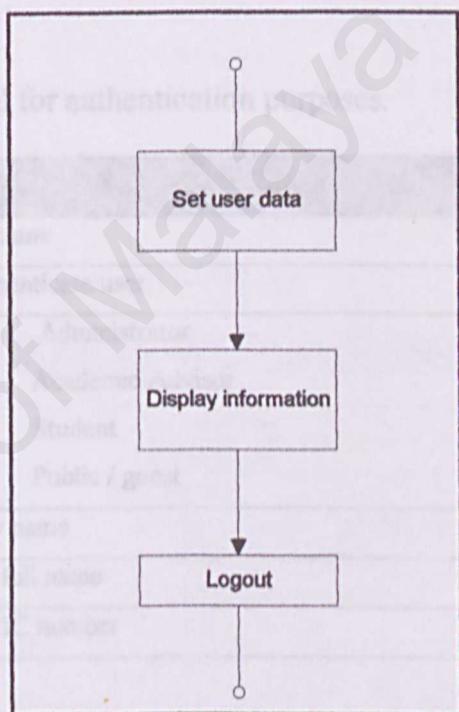


Figure 17: SASS View Courses Taken

This table is used by administrator to identify lecture rooms in the faculty. It will be used in the VTRS module to give information about locations in the faculty.

Room ID	V-001 (50)	Lecturer name
Name	V-002 (50)	Brief description about the room
Description	V-003 (50)	Lecturer's email address
Capacity	V-004 (50)	ID of the room
Room ID	V-005 (50)	Lecturer's photo

## 4.3 Database Design

There are four main tables used in this project. The tables are located in three different locations, which is the local database, faculty database and the central database.

### 4.3.1 Local Database

#### 1) Useraccount

This table stores users ID, password and access level for authentication purposes.

Field Name	Data Type	Description
Username	Varchar2 (20)	Login name
Password	Varchar2 (15)	To authenticate user
Type	Varchar2 (30)	1 Administrator 2 Academic Advisor 3 Student 4 Public / guest
Faculty_code	Varchar2 (150)	Faculty name
FullName	Varchar2 (50)	User's full name
IC	Varchar2 (15)	User's IC number

#### 2) Building\_Info

This table is used by administrator to identify lecturer rooms in the faculty. It will be used in the VTSS module to give information about locations in the faculty.

Field Name	Data Type	Description
Name	Varchar2 (50)	Lecturer name
Description	Varchar2 (255)	Brief description about the room
Email	Varchar2 (30)	Lecturer's email address
RoomID		ID of the room
Pic_path	VarChar2 (100)	Lecturer's picture

### 3) VFeedBack

This table will store the feedback from the visitor.

Field Name	Data Type	Description
Name	Varchar2 (30)	Nick name of the visitor
Comments	Varchar2 (255)	Message
Email	Varchar2 (40)	Email address
Fdate	Date	Date submit

### 4.3.2 Faculty Database

#### 1) Courses

This table will store courses in the faculty and its availability for registration in a particular semester. This table is to be administered by the staff in faculty.

Field Name	Data Type	Description
ID	Number	Primary key
Course_name	Varchar2 (50)	Course Name
Course_code	Varchar2 (10)	Course Code
Pre_requisite	Varchar2 (10)	Pra-syarat
Max	Number	Maximum number of student
InFac	Number	Available for in-faculty
OutFac	Number	Available for out-faculty
CurInFac	Number	Current in-faculty
CurOutFac	Number	Current out-faculty
Course_Type	Varchar2 (2)	Type of course (ex. Core subject, elective etc.)
Credit	Varchar2 (10)	Credit hour for the subject

#### 2) Student\_Contact

Field Name	Data Type	Description
IC	Varchar2 (14)	IC number
Street_1	Varchar2 (50)	First part in the address
Street_2	Varchar2 (50)	Second part in the address (optional)
Postcode	Varchar2 (10)	Postcode
State	Varchar2 (50)	State
Country	Varchar2 (30)	Country
Phone	Varchar2 (12)	Phone number
Type	Varchar2 (1)	0 for permanent address 1 for temporary address

### 3) Student\_Course

Field Name	Data Type	Description
IC	Varchar2 (14)	IC number
Reg_date	Varchar2 (50)	Registration date
Course_code	Varchar2 (10)	Course code ex. WXET1001
Course_name	Varchar2 (30)	Course name
Type	Varchar2 (2)	Core subject, elective or others
Credit	Varchar2 (2)	Credit hour for this course
Gred	Varchar2 (2)	Grade in examination
Status	Varchar2 (2)	Approved or not approve
Course_type	Varchar2 (2)	Core subject, elective or others
Course_credit	Varchar2 (2)	Credit hour for this course
Point	Varchar2 (10)	Point = gred x credit

### 4) Student\_Profile

Field Name	Data Type	Description
IC	Varchar2 (14)	Student's IC number
Name	Varchar2 (50)	Student name
Matrix	Varchar2 (10)	Matrix number
Photo	Varchar2 (50)	Path for the location of image file
Sex	Varchar2 (10)	Male or female
Race	Varchar2 (20)	Race
Birthdate	Varchar2 (10)	Date of birth
Marital_Status	Varchar2 (10)	Marital status
Nation	Varchar2 (20)	Nation
Religion	Varchar2 (20)	Religion
Program_code	Varchar2 (10)	Program code, ex. WET
Cur_level	Varchar2 (1)	Year 1, 2 or 3
Major	Varchar2 (30)	
Minor	Varchar2 (30)	
Qualification	Varchar2 (15)	STPM, Matriculation
CGPA	Varchar2 (5)	
Faculty_code	Varchar2 (30)	
Program	Varchar2 (50)	Ex. Sarjana Muda
Academic_advisor	Varchar2 (50)	Academic supervisor's name
Remark	Varchar2 (255)	Academic supervisor can put note for this student
ID	Number (4)	Primary key
Reg_status	Number (0)	Status of course registration
Faculty	Varchar2 (30)	

In this chapter, it has showed the system architecture, flow chart and database design. The architecture of system is the main basic structure of this system. This project will be developed according to the structure designed in this chapter. The next chapter will discuss on the evaluation of the system.

**[CHAPTER 3]****SYSTEM IMPLEMENTATION****4.3.3 Central Database****1) Courses**

This table stores courses offered as outside faculty courses and university courses.

<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>
<b>ID</b>	Number	Primary key
<b>Course name</b>	Varchar2 (50)	Course Name
<b>Course code</b>	Varchar2 (10)	Course Code
<b>Pre requisite</b>	Varchar2 (10)	Pra-syarat
<b>Max</b>	Number	Maximum number of student
<b>InFac</b>	Number	Available for in-faculty
<b>OutFac</b>	Number	Available for out-faculty
<b>CurInFac</b>	Number	Current in-faculty
<b>CurOutFac</b>	Number	Current out-faculty
<b>Course Type</b>	Varchar2 (2)	Type of course (ex. Core subject, elective etc.)
<b>Credit</b>	Varchar2 (10)	Credit hour for the subject

**2) Fac\_Details**

<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>
<b>Faculty code</b>	Varchar2 (30)	Faculty's code
<b>DSN</b>	Varchar2 (50)	Database server name, user id and password

**3) Semester**

<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>
<b>Cur Session</b>	Varchar2 (10)	Current Session
<b>Cur Sem</b>	Varchar2 (1)	Current semester

**4.4 Conclusion**

In this chapter, it has showed the system architecture, flow charts and database design. The architecture of system is the most basic structure of this system. This project will be developed according to the structure designed in this chapter. The next chapter will describe the implementation of the system.

## [CHAPTER 5]

## SYSTEM IMPLEMENTATION

The system is running under the platform of Windows 2000 Server combining with Internet Information Server (IIS) 5.0 and Oracle 8i Enterprise. This combination is expected to optimize system performance. In this chapter, the modules that are running within the system will be discussed. This can show the system flow during the implementation.

### 5.1 Feedback Module

The feedback module consists of two components. The first one is to let user send feedback while another one is to let user view feedback. Basically, user can only view the feedback after they have sent it.

In sending the feedback, users need to enter a nickname, email address and messages. They have to click on the submit button in order to send the data through Internet to v-faculty's database server. Before that, system uses client script to check whether all text fields are filled. If there were any blank field, system would not allow user to proceed. Otherwise, the user will be redirected by the system to the page that let them view the updated feedback page.

In viewing the feedback, users can just click on the drop-down menu. System will bring them to the view feedback page.

```
sub showViewFeedback
    dim cnn, rs, sqlstr
    set cnn = Server.CreateObject ("ADODB.Connection")
```

```
set rs = Server.CreateObject ("ADODB.Recordset")
cnn.Open "dsn=mainDB;uid=sys;pwd=system"
sqlstr = "select * from sys.feedback"
set rs = cnn.Execute(sqlstr)

Response.Write "<table border=0 cellspacing=2 cellpadding=2 width='70%'>"
Response.Write "<TR><TD colspan=3><FONT color=darkorchid><STRONG>Feedback
From"
Response.Write "Visitors</STRONG></FONT></TD></TR>" 
Response.Write "<TR><TD colspan=3 ><P>&ampnbsp</P></TD></TR>" 
Response.Write "<TR style='FONT-SIZE: x-small; FONT-WEIGHT: bold;'><TD
bgcolor=#a8a8ff>Nick Name:</TD><TD bgcolor=#C8C8FF>Email:</TD><TD
bgcolor=#e8e8ff>Message(s):</TD></TR>" 
while not rs.EOF
    Response.Write "<tr style='FONT-SIZE: x-small'>" 
    Response.Write "<td bgcolor=#a8a8ff width='15%'>" 
    Response.Write rs.Fields(0) 
    Response.Write "</td><td bgcolor=#C8C8FF width='20%'>" 
    Response.Write rs.Fields(2) 
    Response.Write "</td><td bgcolor=#e8e8ff>" 
    Response.Write rs.Fields(1) & "<br>(" & rs.Fields(3) & ")" 
    Response.Write "</td></tr>" 
    rs.MoveNext
wend
Response.Write "<TR><TD colspan=3 ><P>&ampnbsp</P></TD></TR>" 
Response.Write "<TR><TD colspan=4 align=right><INPUT type=button
value='Send Feedback' id=sendFeedback name=sendFeedback
onclick=javascript:window.location.href='result.asp?id=f0' style='BACKGROUND-
COLOR: #ffffdd'></TD></TR>" 
Response.Write "<TR><TD colspan=4><center><font size=-2><a
href='http://www.fsktm.um.edu.my' target='new'>" 
Response.Write "<@ 2000 Faculty of Computer Science and Information
Technology University of Malaya</a></font>" 
Response.Write "<br><font size=-2 color=darkorchid>Last Updated
&ampnbsp" 
Response.Write "<script language=javascript>document.write
(document.fileModifiedDate);</script>" 
Response.Write "<br>All rights reserved.</FON
```

```
</center></TD></TR></TABLE>"
```

```
rs.Close
```

```

cnn.Close
end sub

sub showSendFeedback
  Response.Write "<FORM ACTION='../../asp/handleFeedback.asp' METHOD=POST"
  name=Feedback>

  Response.Write "<table border=0 cellspacing=2 cellpadding=2 width='70%'>"
  Response.Write "<TR><TD colspan=4><FONT color=darkorchid><STRONG>Feedback
From"
  Response.Write "Visitors</STRONG></FONT></TD></TR>"
  Response.Write "<TR><TD colspan=4><P>&ampnbsp</P></TD></TR>"

  Response.Write "<TR><TD><FONT size=2><B><I>Nick Name:</I></B></FONT></TD>"
  Response.Write "<TD><INPUT id=txtName name=txtName style='BACKGROUND-COLOR:
#eeeeff'></TD><TD><TD></TR>"

  Response.Write "<TR><TD><FONT size=2><B><I>Email Add:</I></B></FONT></TD>"
  Response.Write "<TD><INPUT id=txtEmail name=txtEmail style='BACKGROUND-
COLOR: #eeeeff'></TD><td></td><TD></TD></TR>"

  Response.Write "<TR><TD colspan=4><P>&ampnbsp</P></TD></TR>"

  Response.Write "<TR bgcolor=#ddffff><TD colspan=4><STRONG><EM><FONT
size=2>Message / Comment:</FONT></EM>&ampnbsp&ampnbsp</STRONG> <FONT size=2
style='COLOR: navy'>(max 255 characters)</FONT></TD></TR>"

  Response.Write "<TR><TD colspan=4><TEXTAREA id=txtMessage style='FONT-SIZE:
x-small; BACKGROUND-IMAGE: url(../../image/txtBG.jpg); FONT-STYLE: italic; FONT-
FAMILY: fantasy;' name=txtMessage rows=10 cols=85></TEXTAREA></TD></TR>"

  Response.Write "<TR><TD colspan=2><P>&ampnbsp</P></TD><TD align=right><INPUT
id=submit type=submit value=Submit name=submit style='BACKGROUND-COLOR:
#ddffff'></TD>"

  Response.Write "<TD align=right><INPUT id=reset type=reset value=Reset
name=reset style='BACKGROUND-COLOR: #ddFF'></TD></TR>"

  Response.Write "<TR><TD colspan=4><P>&ampnbsp</P></TD></TR>"

  Response.Write "<TR><TD colspan=3><center><font size=-2><a
href='http://www.fsktm.um.edu.my' target='new'>"
```

```
Response.Write "<hr></font><font size=-2 color=darkorchid>Last Updated  
&nbsp;"  
Response.Write "<script language=javascript>document.write  
(document.fileModifiedDate);</script>"  
Response.Write "<br>All rights  
reserved.</FONT></center></TD></TR></TABLE></FORM>"  
end sub
```

## 5.2 VTSS Module

When user clicks on the menu, the main page for VTSS will be shown. In the main page, user can choose either manual tour, semi-auto tour or auto tour. By clicking on the function point at bottom left corner, user will be redirected to the page chosen.

## 5.3 Login Module

Users must enter user name and password before they enter the SASS. In login module, system can recognize different type of users. Then, it should redirect different user type to different page accordingly. For example, administrator should see the admin page after logged in.

At the beginning, this module will get the user name and password from the client-side user. After that, server will find matched user name and password from the database. Appropriate message will be displayed while login failure.

```
sub showLogin
```

```

Response.Write "<style type='text/css'>"  

Response.Write ".help"  

Response.Write "{background-color: #eeeeff;"  

Response.Write "width: 200px;"  

Response.Write "visibility: hidden;}"  

Response.Write "</style>"  

Response.Write "<SCRIPT language=javascript>"  

Response.Write "var isShow=false;"  

Response.Write "function toggle() {"  

Response.Write " if (document.layers) {"  

Response.Write "   if (isShow) {"  

Response.Write "     help1.style.visibility = 'hide'; isShow=false; }"  

Response.Write "   else { "  

Response.Write "     help1.style.visibility = 'show'; isShow=true; } )"  

Response.Write " else if (document.all) {"  

Response.Write "   if (isShow) {"  

Response.Write "     help1.style.visibility = 'hidden'; isShow=false; }"  

Response.Write "   else { "  

Response.Write "     help1.style.visibility = 'visible'; isShow=true; }  

}"  

Response.Write ""  

Response.Write ")"  

Response.Write "</script>"  

Response.Write "<FORM ACTION='../../asp/user.asp' METHOD=POST name=Login>"  

Response.Write "<table border=0 cellspacing=2 cellpadding=2 width='90%'>"  

Response.Write "<TR><TD colspan=4><FONT color=darkorchid><STRONG>User  
Authentication</STRONG></FONT></TD></TR>"  

Response.Write "<TR><TD colspan=4><P>&ampnbsp</P></TD></TR>"  

Response.Write "<TR><TD colspan=4>Visitor needs to enter personal username  
and "</TD>  

Response.Write "password before proceed to the requested page.</TD></TR>"  

Response.Write "<TR><TD align=right><STRONG><EM style='COLOR:  
#6666ff'>username :</EM></TD>"  

Response.Write "<TD><INPUT id=txtUserName name=txtUserName style='HEIGHT:  
22px; WIDTH: 120px'></TD></STRONG></TD><TD colspan=2></TD>"

```

Response.Write "<TD rowspan=3 align=left><DIV class=help id=help1><h2><U><I><FONT color=#000080 size=3>HELP</FONT></I></U></h2>"  
Response.Write "<FONT size=2>If you are a student of University of "  
Response.Write "Malaya, your personal user name will be your matrix number."  
  
Response.Write "<BR>Your password will be your identity card number.</FONT>"  
Response.Write "</DIV><TD></TR>"  
  
Response.Write "<TR><TD align=right><EM style='COLOR:#6666ff'><STRONG>password :</TD>"  
Response.Write "<TD><INPUT id=txtPassword name=txtPassword type=password style='HEIGHT: 22px; LEFT: 231px; TOP: 59px; WIDTH: 120px'></STRONG></EM></TD><TD colspan=2></TD></TR>"  
Response.Write "<TR><TD><TD><INPUT id=login name=login type=submit value=Login>"  
Response.Write "<INPUT id=reset name=reset type=reset value=Reset>"  
Response.Write "</STRONG></EM><I><A HREF='#' onclick='javascript:toggle(); return false'>help</A></I></TD><TD></TD></TR>"  
  
Response.Write "<TR><TD colspan=4><P>&ampnbsp</P></TD></TR>"  
Response.Write "<TR><TD colspan=3><center><font size=-2><a href='http://www.fsktm.um.edu.my' target='new'>"  
Response.Write "<@ 2000 Faculty of Computer Science and Information Technology University of Malaya</a></font>"  
Response.Write "<br><font size=-2 color=darkorchid>Last Updated &nbsp;"  
Response.Write "<script language=javascript>document.write (document.fileModifiedDate);</script>"  
Response.Write "<br>All rights reserved.</font></center></TD></TR></TABLE></FORM>"  
end sub  
  
dim ans, res, result  
  
CollName = Request("CollName")  
User = Request("username")  
Pass = Request("password")  
Email = Request("Email")  
ID = Request("PersonID")

## 5.4 Add User Module (for Administrator and Staff)

Only administrator and staff can add user to the database. The administrator can add all types of users (administrator, staff, academic supervisor and student) while the staff can only add student to the database. User of this module has to enter user name, password, full name, user type, and faculty of the people who wants to join this system.

After that, these data will be stored in the database table called *useraccount*. Later on, the login process will compare the user name and password from here.

Add New User	
Full Name:	Leong Kee Foo
New User Name:	WET98057
IC Number:	770705-14-5463
Password:	1234567890
Confirm Password:	1234567890
User Type:	Student
Faculty:	FSKTM
<input type="button" value="Add"/> <input type="button" value="Reset"/> <input type="button" value="Logout"/> <input type="button" value="Main Page"/>	

Figure 18: Interface for Add User Module

&lt;%

```
dim usr, psw, utype, ic, fullname
dim cnn, rs, sqlstr
```

```
fullname = Request("fullname")
usr = Request("newName")
psw = Request("NewPwd")
utype = Request("usertype")
ic = Request.Form.Item("ICNO")
```

```
</script>
```

Codes above show the function to ensure that student is not clicking on the Add button without fill in the course code.

coming semester. In this module, the staff enter the details of a course follow by click the save button. The record will be stored in the database. To delete a record, staff may

## 5.8 Profile Checking Module

A student who has logged in to this system can view or edit his/her personal profile. Therefore, the upper part in this form is non-changeable value like name, IC number, program code, race, birth date and so on. At the lower part of this form, there are permanent address and temporary address. Student can change the addresses by editing the value in the text boxes follow by clicking the save button.

point, system will do some checking on it. The system will check for duplicated code for that student. This will ensure that a student would not register a course twice within the

## 5.9 Result Checking Module

This module is used to let student know their result in all previous examinations. When student logged in to this system, system will get their IC number as a key to find all the data related to him/her. Therefore, system could get their result out from the database by using this IC number. After that, the result will be displayed in a table like codes below:

```
Sub printTable
    Response.Write "<TABLE border=0 cellPadding=5 cellSpacing=5 width='100%'>"
    Response.Write "<TR style='FONT-SIZE: x-small' bgColor='#e0e0ff'>"
    Response.Write "    <TD><STRONG>No.</STRONG></TD>"
    Response.Write "    <TD><STRONG>Subject Code</STRONG> </TD>"
    Response.Write "    <TD><STRONG>Subject Name</STRONG> </TD>"
    Response.Write "    <TD><STRONG>Credit Hour</STRONG> </TD>"
    Response.Write "    <TD><STRONG>Gred</STRONG></TD>"
```

## 5.6 Add/Remove Course Module (for Staff)

The staff must enter a list of courses in order to let student register it during the coming semester. In this module, the staff enter the details of a course follow by click the save button. The record will be stored in the database. To delete a record, staff may click on the delete button.

## 5.7 Register Courses Module (for Student)

The students can register their courses in this module. First, the student has to enter the code for the course. After that, he/she has to click on the add button. At this point, system will do some checking on it. The system will check for duplicated code for that student. This will ensure that a student would not register a course twice within the same semester. Then, system will find the code out from the central database in order to determine that the code is under the out-faculty category. After that, the system will do checking for in-faculty category. If the code is not duplicated and is either falls in in-faculty category or out-faculty category, it will be added to the database. The latest registration status will be shown after this.

```
<script>
function checkEmpty() {
    if (document.course.code.value == "") {
        alert("Please enter the code first !");
        return false;
    }
    else
        return true;
}
```

```
faculty = Request("faculty")

set rs = Server.CreateObject ("ADODB.Recordset")
cnn = "dsn=mainDB;uid=system;pwd=manager"
sqlstr = "select * from system.useraccount where username=''" & usr & "'"
rs.Open sqlstr, cnn, adOpenDynamic, adLockOptimistic, adcmdText

'Response.Write usr & " " & psw & " " & utype
if not rs.EOF then
    Response.Write "This user name is being used by somebody. Please choose
another one."
else
    rs.AddNew
    rs.Fields("fullname") = fullname
    rs.Fields("username") = usr
    rs.Fields("password") = psw
    rs.Fields("type") = utype
    rs.Fields("IC") = ic
    rs.Fields("faculty_code") = faculty
    rs.Update
end if
```

## **5.5 Approve Registration Module (for Academic Supervisor)**

The job of the academic supervisor is to approve the registration of his/her students. The academic supervisor can just tick the checkbox under the *reg\_status* column to approve the registration.

```

Response.Write "<TD><STRONG>Points</STRONG></TD></TR>"
```

```

while not rs.EOF
    Response.Write "<TR style='FONT-SIZE: x-small' bgColor='#e0e0e0'>"
    Response.Write "<TD><STRONG>" & rowCounter & ".</STRONG></TD>"
```

```

    Response.Write "<TD><STRONG>" & rs.Fields("course_code") &
"</STRONG></TD>"
```

```

    Response.Write "<TD><STRONG>" & rs.Fields("course_name") &
"</STRONG></TD>"
```

```

    Response.Write "<TD align=center><STRONG>" & rs.Fields("credit") &
"</STRONG></TD>"
```

```

    Response.Write "<TD align=center><STRONG>" & rs.Fields("gred") &
"</STRONG></TD>"
```

```

    Response.Write "<TD align=right><STRONG>" & rs.Fields("point")
'mid(formatcurrency(rs.Fields("point"),1),2,len(formatcurrency(rs.Fields("point"
"),1))) & "</STRONG></TD></TR>"
```

```

    credits = rs.Fields("credit") + credits
    points = rs.Fields("point") + points
    rs.MoveNext
```

```

    rowCounter = rowCounter + 1
wend
```

```

Response.Write "<TR bgcolor='#e0e0ff' style='FONT-SIZE: x-small'><TD
colspan=3 align=right><strong>TOTAL</strong></TD><TD align=center><strong>" &
credits & "</strong></TD>"
```

```

    Response.Write "<TD align=center>-</TD><TD align=right><strong>" & points &
"</strong></TD></TR>"
```

```

    Response.Write "<TR style='FONT-SIZE: x-small' bgColor='#ffe0ff'><TD
colspan=5 align=right><strong>GPA</strong></TD><TD align=right><strong>"
```

```

    Response.Write (points/credits) & "</strong></TD></TR>"
```

```

    Response.Write "</TABLE><HR>"
```

end sub

## 5.10 Database Connection Module

In almost every \*.asp files, there are a few standard ways to connect to the database. First, the way of include header file for ADODB connection. It can be like this:

```
<!-- include c:\program files\common files\system\ado\msado15.dll -->
```

or

```
<!-- #include file = "..\adovbs.inc" -->
```

This file is important when developer wants to use the properties of a recordset like addnew, update and so on. The result of the testing showed that the second way is a bit faster than the first way. Therefore, all \*.asp files in v-Faculty system are using the second one.

Second, creating connection and recordset to the database may have two alternatives:

i) 

```
dim sql, rst, cnn
```

```
set rst = new Server.CreateObject("ADODB.Recordset")
```

```
cnn = "DSN=mainDB;uid=system;pwd=manager"
```

```
sql = "SELECT * FROM useraccount"
```

```
rst.open sql, cnn, adOpenDynamic, adLockOptimistic, adCmdText
```

or

ii) 

```
dim sql, rst, cnn
```

```
set cnn = new Server.CreateObject("ADODB.Connection")
```

```
cnn.open "DSN=mainDB;uid=system;pwd=manager"
```

```
sql = "SELECT * FROM useraccount"
```

```
set rst = cnn.Execute(sql)
```

Both alternatives have the same effect, where two of them pull data out from the database and store it into the variable called `rst`. However, each of them may have different usage. Code i) allows user to add new data into the database while code ii) only allows user view the data. For the performance wise, code i) is faster than code ii) in terms of getting data out from the database. The reason is open a connection is slower than open a recordset. Code i) opens the recordset straight away. The system will establish a simple connection for it. While, code ii) request a full connection, which involves more complex procedure than code i).

### 5.11 Conclusion

However, code ii) can easily update and delete a record in the database. For example,

```
sql = "UPDATE useraccount SET name='Ali'"
```

```
cnn.Execute(sql)
```

As conclusion, this testing shows that developer may use either way according to their needs. In this system, both techniques are used.

Third, checking for empty recordset. In ASP, when this error occurred, the IIS will terminate the whole system. During testing, the codes for checking has been added like below:

If `rst.EOF` then

*Response.Write "The record requested is empty."*

Else

*'show the process*

*'it may print out data in table form*

**[CHAPTER 5]**

.....  
..... } codes  
.....

This system is tested based on certain criterions. First, the unit testing has tested individual modules for every subsystem. Second, the module testing has tested the

*End If*

When there is no data in the database, the message “*The record requested is unknown.*” will be shown and SASS as well. After that, the system used an integration testing which called system testing, tested out the full functionalities of the system.

## 5.11 Conclusion

This system has covered all the modules that discussed in previous chapters. During the implementation, some of the bugs may be discovered from time to time. Therefore, next chapter will cover the testing for this system.

When student entered empty code and pressed the “Add Topic”, the message showed during testing

## [CHAPTER 6]

### SYSTEM TESTING

#### 6.2.1 Database Module Testing

This system is tested based on certain criterions. First, the unit testing has tested individual units in the every subsystem. Second, the module testing has tested the workable of IPSS, VTSS and SASS as well. After that, the system used an integration testing when these three subsystems combine through the web pages. Finally, an overall testing, which called system testing, tested out the full functionalities of the system.

- 1) When user has filled up everything in the course form, the latest message is showed.

### 6.1 Unit Testing

#### Check Empty Code

```
<script>
function checkEmpty() {
    if (document.course.code.value == "") {
        alert("Please enter the code first !");
        return false;
    }
    else
        return true;
}

</script>
```

When student entered empty code and pressed the Add button, the message showed during testing.

- 1) If user did not fill up all the fields, system will prompt error.

## 6.2 Module Testing

### 6.2.1 Database Module Testing

When there is no record in the database, system will show appropriate message instead of terminate the implementation of the system.

### 6.2.2 Feedback Module Testing

- After user has filled up everything in the feedback form, the latest message is showed.

Feedback From Visitors		
Nick Name:	Email:	Message(s):
Leo	leo@um.edu.my	Hello, i m kee foo here. i want to show my message out. thank you! (1/12/2001 4:24:19 PM)
WWH	winghong@hotmail.com	Very good must add some more...graphics must be more advance connection too slow must research on speed (1/15/2001 3:34:19 PM)
comguide	james@yahoo.com	hw come must P111, Pentium11 cannot ah? wat the point of the system? (1/15/2001 3:35:58 PM)
Teh	tekiong@yahoo.com	What a rubbish page ! Should something better than this. What lah, can't even go into the VRML site. Download plugin also can not. Should improve !!!!!!! (1/18/2001 12:41:14 PM)

**Send Feedback**

*Figure 19: View the Feedback*

- If user did not fill up all the fields, system will prompt error.

### 6.2.3 VTSS Module Testing

- i) When user clicked on the menu, the main page for VTSS must come out.
- ii) When user clicked on the function point at the bottom left corner, page load according to user selection.
- iii) When user clicked on object inside the virtual world, appropriate information must show in another frame in the same window.

### 6.2.4 Login Module Testing

- i) When user entered the user name and password, system will validate it.
- ii) If error occurred, error message will prompt out.
- iii) After login, user must be redirected to correct page.

### 6.2.5 Add User Module Testing

The screenshot shows a web-based application titled 'Add New User'. The form contains the following fields:

Full Name:	Leo
New User Name:	Leo
IC Number:	770705-14-5463
Password:	password
Confirm Password:	password
User Type:	Student
Faculty:	FSKTM

At the bottom of the form are four buttons: 'Add', 'Reset', '< Logout >', and 'Main Page'.

Figure 20: Add user

- i) User must fill in all data in the text fields. If not system will prompt error message.
- ii) After added a user, this user can login using the new username and password.

### 6.2.6 Approve Registration Module Testing

- i) After academic supervisor checked the reg\_status, the status in the registration form for the student must change to OK.

### 6.2.7 Add/Remove Course Module Testing

- i) When user entered all data and pressed the save button, the data can be updated to the database.
- ii) If data missing, system will prompt error.

Course Registration							
<b>CAUTION:</b> 1. Students have to make sure their data shown below are correct. 2. If any data is incorrect, please contact SKR to correct it as soon as possible. 3. Students should think carefully before add or drop any course. 4. Students can check the list of courses available by clicking <a href="#">here</a>							
STUDENT NAME	IC NUMBER	MATRIX NUMBER	FACULTY	MAJOR	SESSION	SEMESTER	
keefon	770705-14-8463	wt098057	FSKTM	mgt	2000/2001	2	
No.	Course Code	Course Name		Credit	Status	Type	Action
1.	WXET201E	IT Subject		3	OK	CO	<a href="#">Drop</a>
2.	WXET1001	IT SUBJECT 001		3	NR	CO	<a href="#">Drop</a>
3.	WXET1003	IT SUBJECT 003		3	NR	CO	<a href="#">Drop</a>
4.	CBEB1001	BUSINESS SUBJECT 001		3	NR	CO	<a href="#">Drop</a>
5.	WXET1002	IT SUBJECT 002		3	NR	CO	<a href="#">Drop</a>
							<a href="#">Add</a>
<a href="#">Logout</a>   <a href="#">&lt;&lt;&lt; Back</a>							

Figure 21: Register course

### 6.2.8 Register Courses Module Testing

- When user has entered the course code and pressed the add button, the system should do the following checking:
  - if the code already registered, system should prompt error message;
  - if the code is in-faculty course, the course type is CO;
  - if the code is out-faculty course, the course type is either CC or UN.

### 6.2.9 Profile Checking Module Testing

- When user pressed the profile-checking menu, his/her personal detail must be shown.
- If user made changes to the address ad save it, the latest data must appear in database.

### 6.2.10 Result Checking Module Testing

When user clicked on the result-checking menu, the result must show.

The Results of Examination						
*** NOTE: if any figure is wrong, please check with Seksyen Kemasukan Dan Rekod (SKR).						
No.	Subject Code	Subject Name	Credit Hour	Grade	Points	
1.	WXET2016	IT Subject	3	A	12	

Figure 22: Result of examination

### 6.3 Integration Testing

Sometimes, when passing a value from a module to the other may not work properly. In the integration testing, we have to make sure that these values are correctly passed from one form to another. Normally, when one form wants to pass a value to another form, there are a few alternatives. Two of them are as below:

i) `<FORM NAME=form1 ID=form1 ACTION="doSomething.asp"`

`METHOD=post>`

`<INPUT TYPE=text NAME=text1 ID=text1>`

`<INPUT TYPE=submit NAME=submit1 ID=submit1>`

`</FORM>`

when user click on the submit button, the data will send to the doSomething.asp form.

ii) `<A HREF=# onclick="doSomething.asp?id=value1">Send Me</A>`

when user click on the link, system will send value1 to the form called DoSomething.asp.

In the doSomething.asp form, there is a function called Request.QueryString().

Here, the form will assign a variable to store this value like this:

`Dim value`

`value = Request.QueryString(id)`

For testing purposes, an extra code has been written in order to make sure that the value is passed from the source form.

*Response. Write value & " has passed to here."*

The testing for this is to pass string "abc" from facultylist.asp to delfaculty.asp.

The result of "abc" can be printed out in the delfaculty.asp.

## 7.1 System Testing

### 6.4 System Testing

In the system testing, first, an administrator login to the system and create a user called staffA, a person who hold the priority of a staff. After that, staffA adds a course called WXET1000 to the database. When a student login from another PC, he/she wanted to register the course WXET1000. When he press the add button, the code added. If he/she misspells the WXET1000 as WXET1001, the system prompted him/her an error message. After that, the administrator creates an academic supervisor in the database. The academic supervisor approved the registration of the student. Then, the student just now can see that the status of the course WXET1000 has changed to OK.

### 6.5 Conclusion

Testing has two main functions; one is to establish the presence of defects in a program while another one is to help judge whether or not the program is usable in practice. This concludes that a testing can only demonstrate the presence of errors. It cannot show that there are no errors in the system. In next chapter, this report will evaluate the overall performance by stating the strengths and limitations of this system.

## [CHAPTER 7]

### CONCLUSION

#### 7.1 System Evaluation

After the system implementation and testing, we found that most of the functionalities mentioned in the objectives are covered. There is a part that has been taken out from the system, which is the avatar. The avatar is a very new idea in the Internet. There are too much effort put into it but it did not show a good result. As conclusion, this project has covered all the objectives mentioned in front.

#### 7.2 Strengths

##### 7.2.1 VTSS

The VTSS is an online virtual world that provides 3D interactive space on the Internet. With the functions provided by VRML, user can create a simple and beautiful 3D space. Some of the 3D tools in the market may not be able to support this feature. Although some of them can do the same thing but it has to worry about the size of the file. A \*.wrl file is very small. In the market, there are many tools can simplify the VRML codes. Therefore, the file size is no longer a constraint for this system. Further more, the \*.wrl file can be zipped into smaller form by using the VRML Pad. A 100 MB \*.wrl file can be simplified until 2 MB. These features indirectly produce a high quality product on the Internet.

### 7.1.2 SASS

The SASS is an online transactions system, which allows student register course, check profile and check result. This is a very important system in any educational institution. The strength of this system is: it can work on the Internet. No other similar system can provide this functionality.

The second advantage of this system is its ability to distribute all the data to every faculty. Instead of keep all data in one central database, we can use a few database to store all the data. This implies another advantage, which is disaster prevention. If anything happens in any database, this system still can run.

Since SASS applied the concept of online, availability is another advantage of this system. The server may operate 24 hours a day, then the student can access it any time. If a comparison is made between current system and SASS, it is very obvious that the availability of SASS is higher.

SASS emphasizes user friendliness. A simple and easy-to-understand interface makes people feel easy to use. All the functional links are labelled in order to show the actual usage.

By implementing SASS, the university can save a lot of resources. It included money, time and human resource. The first reason is university does not need to assign a permanent staff for this system. The second reason is this system does not involve a lot of paper work.

## 7.3 Limitations

### 7.3.1 VTSS

The browsers in the market do not provide the plug in for the VRML files. User needs to download the Cosmo Player or Cortona from the Internet. Although the Internet Explorer has its own plug in for VRML, it does not support all codes written in VRML 2.0. Consequently, the effect of VRML becomes lesser.

There is a gap between hardware technology and software technology. The developer for software engineering has invented the way to put 3D graphics on the Internet while most of the hardware in the market still lack of capability to support it. This can be explained in terms of transfer rate and graphics processing power.

The VRML 2.0 is not capable to interact with database for the time being. This feature of VRML is still under research in the lab. Until year 2000, the capability of VRML 2.0 is only limited to display 3D graphics, animations, sounds and movies, and sensors. It cannot receive input from user. At the same time, it cannot store a shared variable for the javascript. Therefore, the VRML 3.0 is expected to add in these features.

### 7.3.2 SASS

The SASS does not have enough information about the students. There are a lot of data that could not get from university due to confidentiality and

### 3. A virtual office of a faculty on the Internet

privacy. In addition, some of the departments do not allow others to access their database. Therefore, this subsystem may not provide sufficient information to the users.

SASS is an incomplete system because it needs to integrate with other system to perform well. For example, the student registration system may provide other personal information about a student. The examination system may show details about the papers taken by a student.

## 7.4 Future Enhancement

In future, the v-Faculty may do a few enhancements in order to make it as complete as it can.

1. Integrate it with other student related system

This system is very suitable to integrate with other systems. Its functionality will increase by providing student more information. The most suitable system to integrate with SASS is student registration system, student examination system, student financial system and student society system. This integration can keep full record of any student in this university.

2. The VTSS can integrate with VRML's new features in order to create a full VRML page including manage user input, connecting to database, and share variable with HTML pages.

### 3. A virtual office of a faculty on the Internet

REF: Student, staff and everyone do not need to come to the university. They can

- [1] Har... do all transactions anywhere as long as they have a PC on hand.

Addison-Wesley

- [2] Ames, A. L., Mueller, J. C., and, J. L. (1995). *The Internet 2.0 Sourcebook*.

- [3] Lemay, L., M., and, S., (1997). *Learn! Learning Workshop 3D*.

Graphics and VRML for Learning

- [4] Petrie, M. (1997). *Virtual Reality and Building Computer Models*.

- [5] Marin, C. (1997). *Introduction to VRML*. Zycad Corporation Academic

Computer Publishing

- [6] Rael, B. (1997). *Modeling Objects 2.0 with Java*. Academic Computer Publishing

## REFERENCES

- [1] Hartman, J., Wernecke, J. and Carey, R. (1996). The VRML 2.0 Handbook.  
Addison-Wesley
- [2] Ames, A. L., Nadeau, D. R. and Moreland, J. L. (1996). The VRML 2.0 Sourcebook.
- [3] Lemay, L., Murdock, K. and Couch, J. (1997). Laura Lemay's Web Workshop: 3D  
Graphics and VRML 2. Sams Publishing
- [4] Pesce, M. (1995). VRML Browsing and Building Cyberspace.
- [5] Marrin, C. (1997). Sams Teach Yourself VRML 2 in 21 Days. Sams, Macmillan  
Computer Publishing.
- [6] Roel, B. (1997). Late Night VRML 2.0 with Java. Que, Macmillan Computer  
Publishing.