Enhancement on
UTP SAP Campus Management System

by

Amir Ismail Bin Abdul Kadar

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Business Information System)

JANUARY 2006

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Enhancement on UTP SAP Campus Management System

by

Amir Ismail Bin Abdul Kadar

A project dissertation submitted to the
Business Information System Programme
Universiti Teknologi PETRONAS
In partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(BUSINESS INFORMATION SYSTEM)

Approved by,

(KHAIRUL SHAFEE BIN KALID)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
January 2006

CERTIFICATE OF ORIGINALITY

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

(AMIR ISMAIL BIN ABDUL KADAR)

TABLE OF CONTENT

ABSTRACT

SAP Campus Management is a powerful solution for every constituent on campus life either to its executives, administration, faculty, or to its students. It also support the overall student life cycle from the registration process and up to the graduation.

Some of student academic processes in UTP are not supported by SAP Campus Management solution. From here, the work to improve the system is initiated to help increase the productivity and to lessen the burden of the admin staffs in maintaining the student academic cycle in UTP.

During the first part of the project, the two objectives have been completed which are to study and understand UTP Campus Management system's business processes and its framework and to identify for any potential area or business processes that can be improved and enhanced. Currently, for the second part of the project, the objective of the project is to design and develop the appropriate solutions or ABAP program in order to automate the audit graduation process.

This project will only focus its area of study on improving existing UTP Campus Management system. Specifically, the project development will focus on graduation process only.

The project has completed two phases from its four phases which is the planning phase and analysis phase. For this semester, design and implementation phases will be covered. Design phase is scheduled to end on end of February 2006 and implementation phase is scheduled to end on end of May 2006.

By the end of the project, a complete report will be given to project supervisor and a complete program to support the required business process(s) will be produced.

LIST OF ILLUSTRATION

List of Figure

List of Table

1. INTRODUCTION

1.1. Background

SAP has been providing business solutions to higher education and commercial customers around the world for 30 years. Hundreds of organizations use software from the SAP for Higher Education & Research (SAP for HE&R) set of solutions to integrate and manage their processes from finance and HR to facilities management and procurement to teaching and research. In use by its customers in the United States, the United Kingdom, Central Europe, Asia, and Latin America, the SAP® Campus Management application extends its portfolio to support business processes for student records, student financials, recruitment and admission, student advising, academic structure, and class scheduling as well as academic services for teaching and research.

The traditional administrative tool for managing student records, the student information system, has become a central element of any college or university business-application system. Changes in campus culture, especially the widespread use of the Internet, have raised student expectations of services provided by technology. At the same time, university leaders have become more accountable for the business aspects of campus life. As a result, the student information system has become an important management tool in the drive to become more effective and more competitive. Indeed, in many countries, this drive is encouraged by government initiatives to increase the education level of their citizens by modernizing universities and streamlining administrative processes. Today, a student information system is a crucial tool for supporting change in the global world of higher education and research.

Through a rigorous requirement-gathering process, SAP's architects have built a solution that supports the widely differing administrative processes as well as different educational models found around the world: a nonprogram-oriented model for commercial education offerings; a program-type model, which is most commonly used in the United States; and a program-oriented model, which is typical at many traditional universities in European countries.

But as education becomes global, many institutions face the challenge of supporting a variety of models. SAP Campus Management empowers any institutions to manage this with one solution. Furthermore, SAP Campus Management draws on best business practices not only from our higher education customers, but also from SAP experience in public services and the commercial world.

In Universiti Teknologi PETRONAS (UTP), Campus Management is being used widely throughout its business processes such as in finance department and in student administration such as student's recruitment, admission, registration, and graduation. In finance department for example, student fees such as study fee and accommodation fee are being managed by the department. Students result such as grades, GPAs, and CGPAs are being managed by examination and record unit. Like other Campus Managements, the Campus Management system in UTP follows the general student academic cycle starting from recruitment, admission, registration, examination and grading, pre-registration, graduation, and lastly deregistration. Each business process is connected with each other and will affect other processes. For example, if a student has not done his pre-registration during his previous semester, that student will not be recorded to be studied in UTP for that semester.

1.2. Problem Statement

Although SAP Campus Management generally has provided a complete solution for institution's student academic cycle or student life cycle, there are some processes that are not supported by the CM solution and some of these processes had to be manually done by the users and administrative staff of an institution. These weaknesses are meant to be as an opportunity for any volunteers to help improve the system and gain some experiences from it.

In all the processes in student academic cycle, graduation process is found to be able to be improved in its process of graduation audit. For each semester, the staffs need to manually do the graduation audit on the students, one by one. This consumes quite a time and it will consume more time if the number of students increase.

The problem with current system is there is no any program that can automatically check and compare students' enrolled courses with their own academic structures in order to keep track whether the students have completed their study or not in UTP. Thus, this will lead to some confusion to UTP admin staffs and to students in order to keep track students progress in UTP. Plus, the academic structures in UTP keep changing to ensure students competitiveness in business world.

1.3. Objectives and Scope of Study

In the first part of the project, which was done in previous semester, below objectives have been achieved:

1. To study and understand UTP Campus Management system's business processes and its framework.
2. To identify for any potential area or business processes that can be improved and enhanced.

In the second part of the project, below were the objectives that have been achieved by the end of the project:

1. To design and develop the appropriate solutions or ABAP program in order to automate the audit graduation process.

The scope of the study was focused on UTP Campus Management system. The work is focused on a single process of all the processes within a student academic cycle which is, in graduation process. Currently, based on the early analysis done, the graduation process was the most critical part that needs to be improved since current system cannot keeps track whether a student has graduated or not or has completed his/her courses based on the academic structure that he/she is in or has not.

2. LITERATURE REVIEW AND/OR THEORY

2.1. SAP Campus Management Overview

SAP Campus Management is designed to support the management of teaching and studies in general, and the progress of each individual student. The application is flexible to adapt to changing institutional needs and new technology, while supporting the traditional core processes of the student life cycle. Based on SAP Campus Management report ("SAP® Campus Management", 2005: 1), the current SAP Campus Management solution includes the following features and functions:

- **Academic structure** – The back-end hierarchical data structure can be used for a Web display of the academic offerings of your institution such as degree programs, specializations, and all courses offered by schools and faculties
- **Academic calendar** – Through here, users can set up academic years, sessions, class beginnings, and the duration of registration and exam periods. The processes of the student life cycle will follow this calendar.

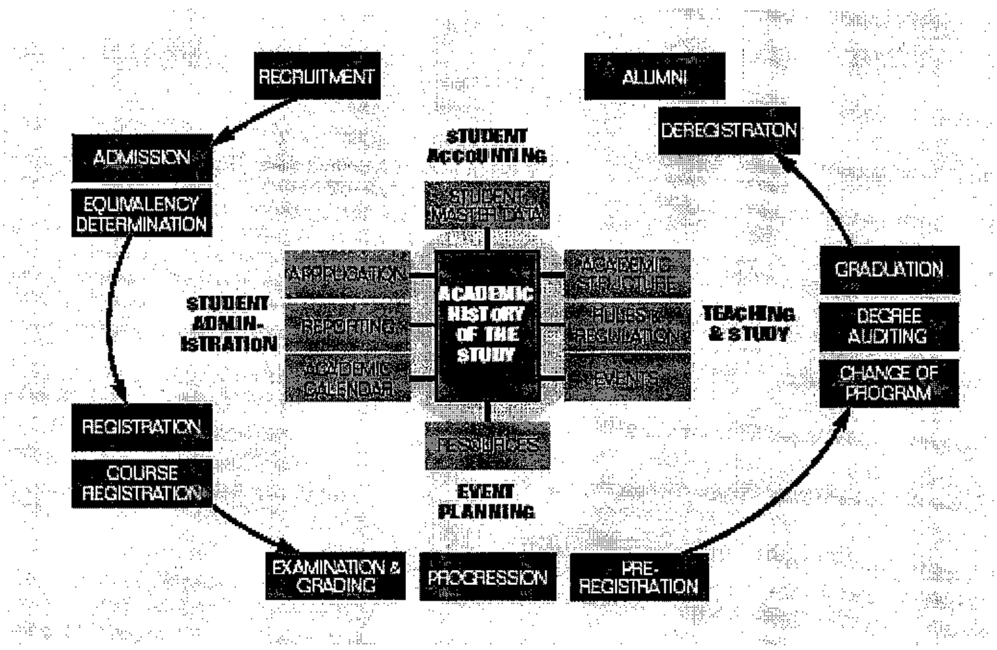


Figure 2.1: Student academic cycle

Above figure shows the general student academic cycle or student life cycle in Campus Management solution that is provided by SAP. All institutions who implement SAP Campus Management solution within their organizations generally have the same academic cycle as above figure. Above figure is taken from one of Campus Management reports (“SAP® Campus Management”, 2005: 2).

- **Course scheduling** – This functionality allows customers to track and schedule a range of resources including teachers and technical support.
- **Equivalency determination** – In SAP Campus Management, users can define transfer agreements with other higher-education institutions. Once defined, the application will assess the student’s external achievements and suggest internal equivalencies, transfer credits, and grades.

- **Admissions** – Easy-to-use online applications are included in the core processes of student administration to promote quick responses and efficient processing. The Internet applications are routed to the responsible admission officers. Application processing can be accelerated with customized workflows. Subprocesses such as rule checking or creation of student master records and account data from applications can be triggered in the background for automated processing. Finally, the system can send e-mail or letters to prospects, informing them of the final status of their applications. In the meantime, students can always check the current status of their applications on the Web.
- **Enrollment and registration** – A student can register for courses or for a degree program, and by doing so attain "active" student status for an academic year or an academic year and session. Financial holds can be set if fees have not been paid on time. Web-based priority registrations can be supported by "time windows" defined in the academic calendar. Windows can be used to manage workload, to reduce performance risks during a registration period, or to give selected students first access to limited seats in classes. It is possible to have multiple sessions open for registration. During the registration procedure, the system can automatically check time conflicts in a student's schedule, custom-defined registration rules, capacity limits of events, holds, and so on. Senior staff can be given the authority to override specific rules if necessary. SAP Campus Management also supports processes such as leave of absence, change of program, withdrawal, and dismissal.
- **Student file** – It is the core repository for the most important student data and the central starting point of all student-related processes. In the student file, users will find personnel data such as addresses, visa information, and information on related persons or the particular student's account. Users can add and change individual notes, remove or set financial or academic holds, and check all kinds of status information. The student file provides access to the academic history of the student, to current course registration, and progression results. Access to data is protected by authorizations and confidentiality warnings per the Family Educational Rights and Privacy Act (FERPA).

- **Online account statements and bill payments** – Students can check the status of their student accounts at any time – online. They can pay open items either online, using functionality in the SAP Biller Direct application, or in person at the bursar's office using cash-desk functionality. SAP Biller Direct and cash desk functionality support online payment-card verification and are fully integrated with the student receivable system. Depending on the university's policy, a student can select open items to be paid or just indicate a payment amount, which will be applied to open items based on clearing rules defined by the institution. The software supports partial payment and overpayment as well as installment plans.
- **Student receivables and student billing** – The application provides a highly flexible fee calculation including refunding, due dates, and account assignments for controlling and funds management. This functionality allows for complete receivable processing including invoicing, payments, clearing, and dunning, and can easily handle high data volumes. The application supports financial-aid disbursements including loans and third-party sponsorship. Aid in excess of fees can be transferred electronically to the student's bank account or paid out by check. Where required, SAP Campus Management can issue invoices to sponsors including lists of those students who were sponsored.
- **Grading** – Authorized faculty and staff can access class rosters online. Examiners can enter results with grading scales that vary according to course level or individual students. When an examiner has entered and checked all results, he or she can complete the grading process by setting the "final" grading status. Only then can students see their grades in the student self-service area: Only then can follow-up processes like progression and degree audit make use of the results.

- **Progression** – SAP Campus Management can measures the progress of students in different ways. One option is to determine the academic standing (good standing, on probation, and so on) and class (freshman, sophomore, junior, or senior) based on grade point averages (GPA) and achieved credits of all undergraduate or graduate work of a student. Another option is to evaluate a student's progress against the requirements of a given stage of the student's degree program.
- **Degree and stage auditing** – Degree and stage auditing is an Internet-based tool for various users to check the current status of a student's progress toward a degree. A student can use it to check his or her current status in progressing toward a degree. Faculty and staff can use it to run a degree audit during student advising or as part of the examination and graduation process.
- **Graduation** – SAP Campus Management can track applications for graduation and can evaluate results of degree audits to support the graduation process. A background report can automatically de-register students after their graduation.
- **Correspondence** – SAP Smart Forms provides an easy-to-use graphical user interface (GUI) for creating and adjusting forms for mass correspondence without any programming skills. SAP Campus Management provides many predefined attributes such as a student's name and ID, the status of an admission application, or data regarding a student's academic history, for automated processing of your correspondence. Additionally, correspondence document templates based on Microsoft® Word or Microsoft Excel are available with SAP Workflow to facilitate individual correspondence with prospects or students, for example, during an admission process.
- **Multidimensional reporting** – Set up of several data dimensions and structures is standard, as is the option to generate key figures for strategic reporting. Web enhancements enable you to publish online reports via the Web.

- **Security and authorization** – The authorization process in SAP Campus Management includes data security on the transaction, process, and activity level. In addition, a structural authorization method enables users to set up restrictions depending on the organizational structure of your institution. For example, limiting the authorization of a user so that he or she can only change degree programs and courses offered by a specific faculty or school while he or she can still display all programs and courses offered by the university. An audit trail mechanism in SAP Campus Management tracks critical data changes in activity documents, access to which is usually highly restricted.

From the report, it says, within one year (2002–2003), seven SAP customers went live with SAP Campus Management and further implementation projects are in progress. Live implementations are proceeding on four continents: Europe, North America, Latin America, and Asia.

The following are some highlights of SAP Campus Management projects at four very different institutions with similar goals. The In each case, these SAP customers were looking for a low-cost, high-return extension of their existing SAP environment with integrated workflow and seamless processes. Teams of university subject experts worked for several months with external consultants in process reengineering, customizing, and go-live. After thorough training, each customer went successfully into production. User acceptance of the new technology has been high among both students and staff. On the next pages, a quick overview about SAP Campus Management implementations for each of the university are highlighted which are University of Basel (Switzerland), University of Montevideo (Uruguay), University of Mississippi (United States), and Katholieke Universiteit Leuven (Belgium). The following overviews on the next pages were derived from SAP Campus Management report ("SAP® Campus Management", 2005: 1)

2.1.1 University of Basel, Switzerland

Implemented functions for:

- Organizational structure
- Academic structure
- Student administration
- Admissions
- Re-registration
- Student accounting
- Integration with SAP Human Resources and mySAP Financials

2.1.2 University of Montevideo, Uruguay

Implemented functions for:

- Student administration
- Academic rules
- Course registration
- Grading
- Student self-service
- Student accounting
- Integration with mySAP Financials and SAP Human Resources

2.1.3 The University of Mississippi, United States

Implemented functions for:

- Academic structure (including publication on the Web)
- Student administration
- Admission via the Web with ISR applications
- Course registration (including self-services via the Web)
- Grading self-service for examiners on the Web
- Reports are now delivered as spreadsheets
- Student accounting
- Online student account statements
- Full integration with SAP Human Resources and mySAP Financials including controlling and funds management

2.1.4 The Katholieke Universiteit Leuven, Belgium

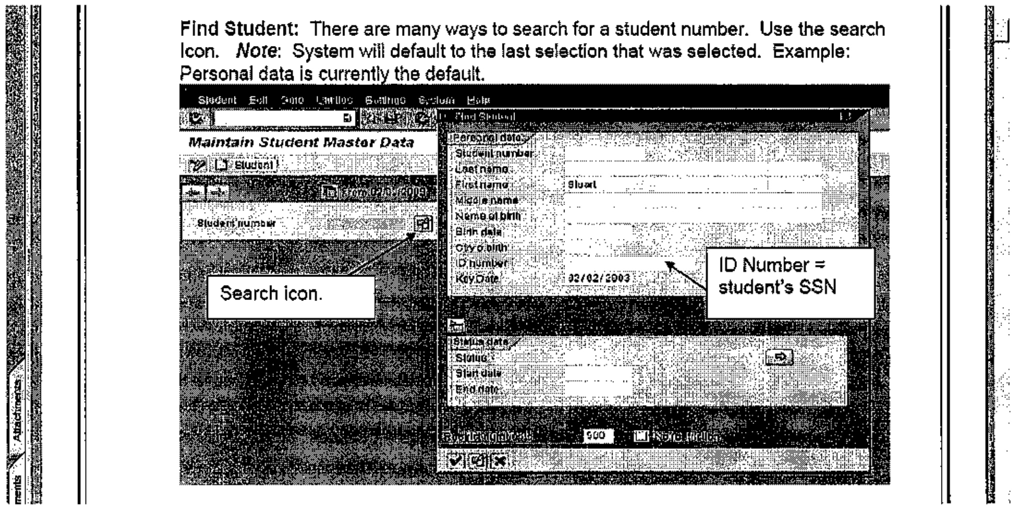
Implemented functions for:

- Academic structure (including publication on the Web)
- Planning of the academic year (including publication of schedules on the Web)
- Student administration (including registration and re-registration via the Web)
- Student accounting
- Reporting using SAP Business Information Warehouse
- Full integration to SAP Human Resources, mySAP Financials including Real Estate

2.2. SAP CM example screenshots

Based on the screenshots below (see Figure2.2 and 2.3) which is derived from report ("Student Master Data – SAP Screenshot", 2004: 3) from University of Mississippi, a part of the database design was structured and was planned out.

2.2.1. Student Master Table – Find function



Find Student: There are many ways to search for a student number. Use the search Icon. *Note*: System will default to the last selection that was selected. Example: Personal data is currently the default.

Maintain Student Master Data

Find Student

Personal data

Student number

Last name

First name: Stuart

Middle name

Name at birth

Birth date

City o birth

ID number

Key Date: 02/02/2003

Status data

Status

Start date

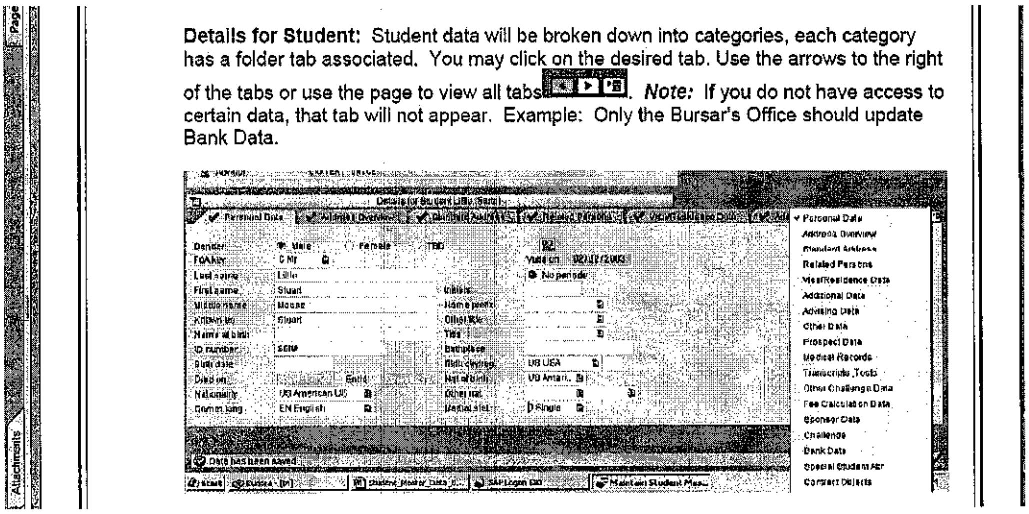
End date

Search icon.

ID Number = student's SSN

Figure 2.2: CM screenshot – Find function

2.2.2. Student Master Table – Student details



Details for Student: Student data will be broken down into categories, each category has a folder tab associated. You may click on the desired tab. Use the arrows to the right of the tabs or use the page to view all tabs◀ ▶ ▤. *Note:* If you do not have access to certain data, that tab will not appear. Example: Only the Bursar's Office should update Bank Data.

Gender: ● Male ○ Female ○ TBD

FOAKey: C Mr

Last name: Little

First name: Stuart

Middle name: Mouse

Known as: Stuart

Name at birth

ID number: SSN

Birth date

Died on

Nationality: US American US

Comm.lang.: EN English

Valid on: 02/02/2003

● No periods

Initials

Name prefix

Other title

Title

Birthplace

Birth country: US USA

Nat.at birth: US Ameri..

Other nat.

Marital stat.: 1 Single

Data has been saved

Figure 2.3: CM screenshot – Student details

3. METHODOLOGY/PROJECT WORK

The following development phases were followed during the development of the project: planning, analysis, design, and implementation. However, the project was divided into two parts which is conducted within two different semesters. Planning and analysis phases were conducted within the first part of the project or project part A and design and implementation were conducted within the second part of the project or project part B. Below was the timeline for the project (see Table 3.1 and Table 3.2).

	Time				
	August'05	September'05	October'05	November'05	Disember'05
Planning	■	■			
Analysis			■	■	■

Table 3.1: Project part A's timeline

	Time				
	January'06	February'06	March'06	April'06	May'06
Design	■	■			
Implementation			■	■	■

Table 3.2: Project part B's timeline

In the following pages, the details about the four phases will be elaborated.

Below (see Figure 3.1) is the diagram of methodology used within the project. It starts with planning, analysis, design, and lastly implementation. Within planning, research and user interview were done. In analysis, use case diagram and user requirement were produced. During design phase, system architecture, database design, and user interface were designed. At the last phase, implementation phase, the codes were written, the program was tested, and a report was sent to the project supervisor. In the following pages, the details about the methodology are discussed.

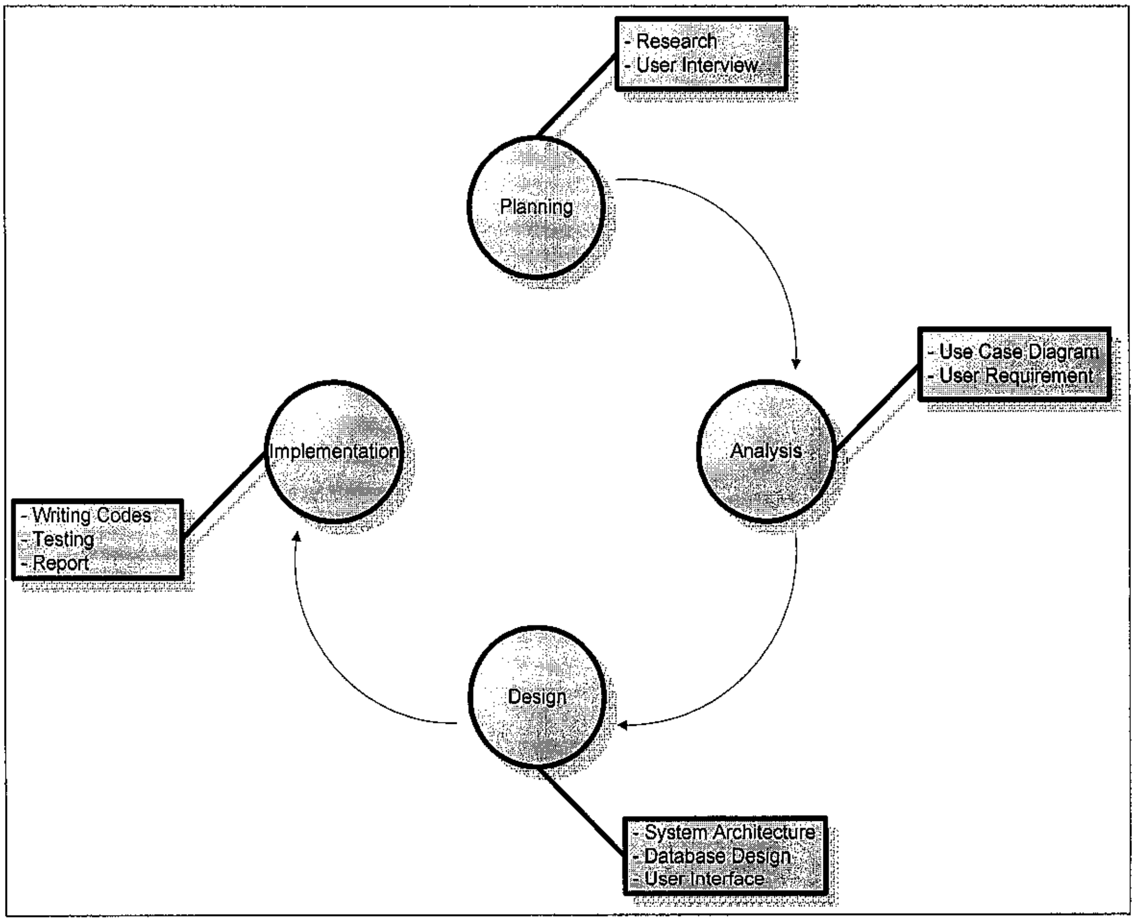


Figure 3.1: Project methodology

3.1. Planning

3.1.1. Research

In order to understand the Campus Management features and functions, necessary information about the application was gathered through any documentation that exists whether from Internet or library about SAP Campus Management application. Through here, any flaws in UTP Campus Management implementation can be detected by comparing it with other universities Campus Management application.

3.1.2. User Interviews

Interview sessions were conducted on one of executive from UTP Examination and Records unit, Mr. Azlan Zahuri. System overview and the business processes that involve in the system are part of the subjects in the system. Other than that, user requirements were came from him also.

3.2. Analysis

3.2.1. Use Case Diagram

Based on the briefing with the Examination and Records Executive, Mr. Azlan Zahuri, analysis was done to understand the graduation processes involved within the system and a diagram was produced to illustrate the process. Refer to Figure 4.1 at page 20 under Results and Discussion section for more details.

3.2.2. User Requirements

User requirements gathering were conducted during analysis phase and the requirements comes from Mr. Azlan Zahuri. Additional requirements were gathered from time to time throughout the development of the project. Details on user requirements can be found at page 22 under Results and Discussion section.

3.3. **Design**

3.3.1. System Architecture

System architecture was the first thing that was drafted during the design phase of the project. See Figure 4.3 at page 23 under Results and Discussion section for more details.

3.3.2. Database Design

Next, ERD and database design were planned out. Both of it can be found at Figure 4.4 and Figure 4.5 at page 24 and page 25 under Results and Discussion section.

3.3.3. User Interface

Next, before the implementation was started, the user interface of the program such as screens was planned out. The interfaces can be found from Figure 4.6, Figure 4.7, and Figure 4.8 at page 26, 27, and 28 respectively.

3.4. Implementation

3.4.1. Writing Codes

The implementation phase was started by writing the codes for the program especially on the screens and functions. The codes for the program can be found at Appendix under Developer Manual section starting at page 32 to page 57.

3.4.2. Testing

After the program's codes were written, it was gone through two tests which are alpha test and beta test. In alpha test, the program was tested by its developer. In beta test, the program was tested by external parties other than its developer such as its users.

3.4.3. Report

At the end of the project, a complete program was produced. A report consisting the development phases of the project and a complete report about it was handed over to the project supervisor, Mr. Khairul Shafee Bin Kalid.

3.4.4. Proposal, and Installation

Based on user acceptance level, if required, a proposal will be made to UTP management if the program is accepted by its user which is the student admin unit. There is a possibility where users may want the program to be installed in the system. For this purpose, UTP IT and Multimedia Services (ITMS) unit will be contacted to hand over the installation work to ITMS unit.

3.5. Resources and Other Supports

Basically, most of the resources needed to initiate the project were available to the student such as PC, connection to ABAP compiler, and reference materials especially from Internet and UTP library.

The platform to conduct the project was UTP SAP Campus Management system. The permission to access the system had been permitted by UTP Registrar. However, student was restricted to view the confidential data such as other students' results and other UTP staffs' salary.

Other than above, student had the opportunity to get help by one of SAP employees, Pn. Suhana Omar and by other experience students in SAP and ABAP especially Ahmad Fikri Bin Amer Hamzah and Nazrul Aminur Hafiz Bin Abdul Rahman.

4. RESULTS AND DISCUSSION

4.1 Results

4.1.1 Graduation process

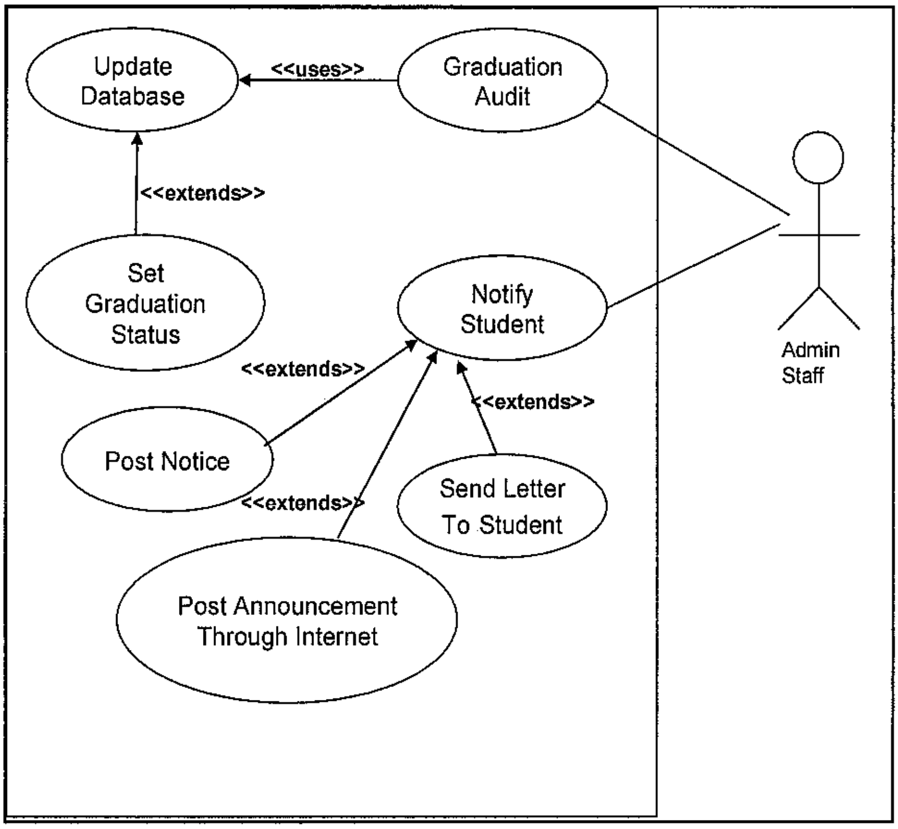


Figure 4.1: Use case diagram for graduation process

Above was the use case diagram for graduation process in UTP. At that time, students were not able to interact with the system. Students may only submit graduation audit form to Examination Unit by the end of their semester or one year before that and the admin staffs will once again do the graduation audit on the students. After the audit, admin staffs will update the database and change their graduation status to "Complete" if students have completed all the required courses along their study in UTP. Then, students will be notified whether through notices, online announcement, or through letters.

Admin staffs usually will do the graduation audit one year before students are estimated to graduate in order to help students to graduate in time if they accidentally unaware that they have some other courses that have not been completed yet. For example, if a student has been noticed one year or two semesters before he graduates telling him that he has not completed one course that he should have taken during his first year, he can enroll the course during his final semester. Else, if the notice was submitted to him during his final semester, he has to extend his study just for only one course.

Below was the data flow diagram for graduation process (see Figure 4.3). Admin staff gets student lists based on certain batches that have been scheduled to graduate in one year time. Based on that, student lists will be generated from the system and the graduation audit begins. Based on student status, student master database will be updated and notices will be sent to students and will be posted at notice boards.

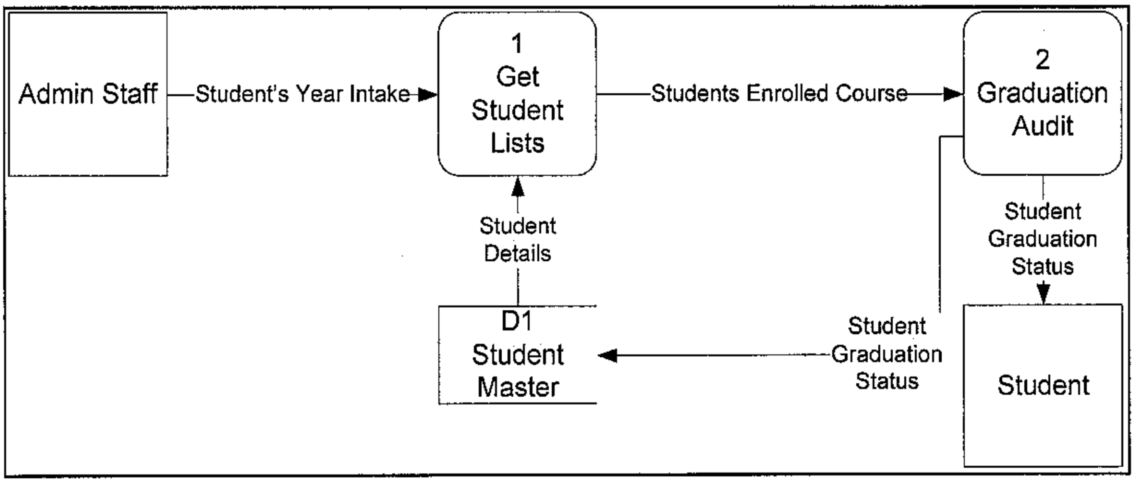


Figure 4.2: Data flow diagram of graduation process

4.1.2 User Requirements

Student had conducted several interview and discussion sessions with one of the system users from UTP Examination and Records unit, Mr. Azlan Zahuri. In the session, he had given an overview about the system and the business processes that involve in the system. Through time, he explained how the graduation process is being done in UTP and laid out the requirements that needed to be achieved in order to ensure the project can meet the organization's needs. Below are the current requirements that have been extracted from Mr. Azlan:

1. The program will help reduce the man power needed for working on graduation audit.
2. Admin staffs do the audit process based on single student intake or batch by batch for each semester.
3. The program can automatically compare students' enrolled courses with their own program academic structures.
4. The program can generate reports listing down students who are in track for their graduation and can generate report listing down students who need to extend to complete their study before graduating.
5. The program can store the academic structures for all programs.

4.1.3 System Architecture

A stand alone database was used by the program without any connection or link with SAP Campus Management database. Since it was hard to get any source code on existing SAP CM system and the data there were very confidential to students, the program was been developed as a stand alone program and as a prove of concept subject for the project. Below was the system architecture for the program that was been developed.

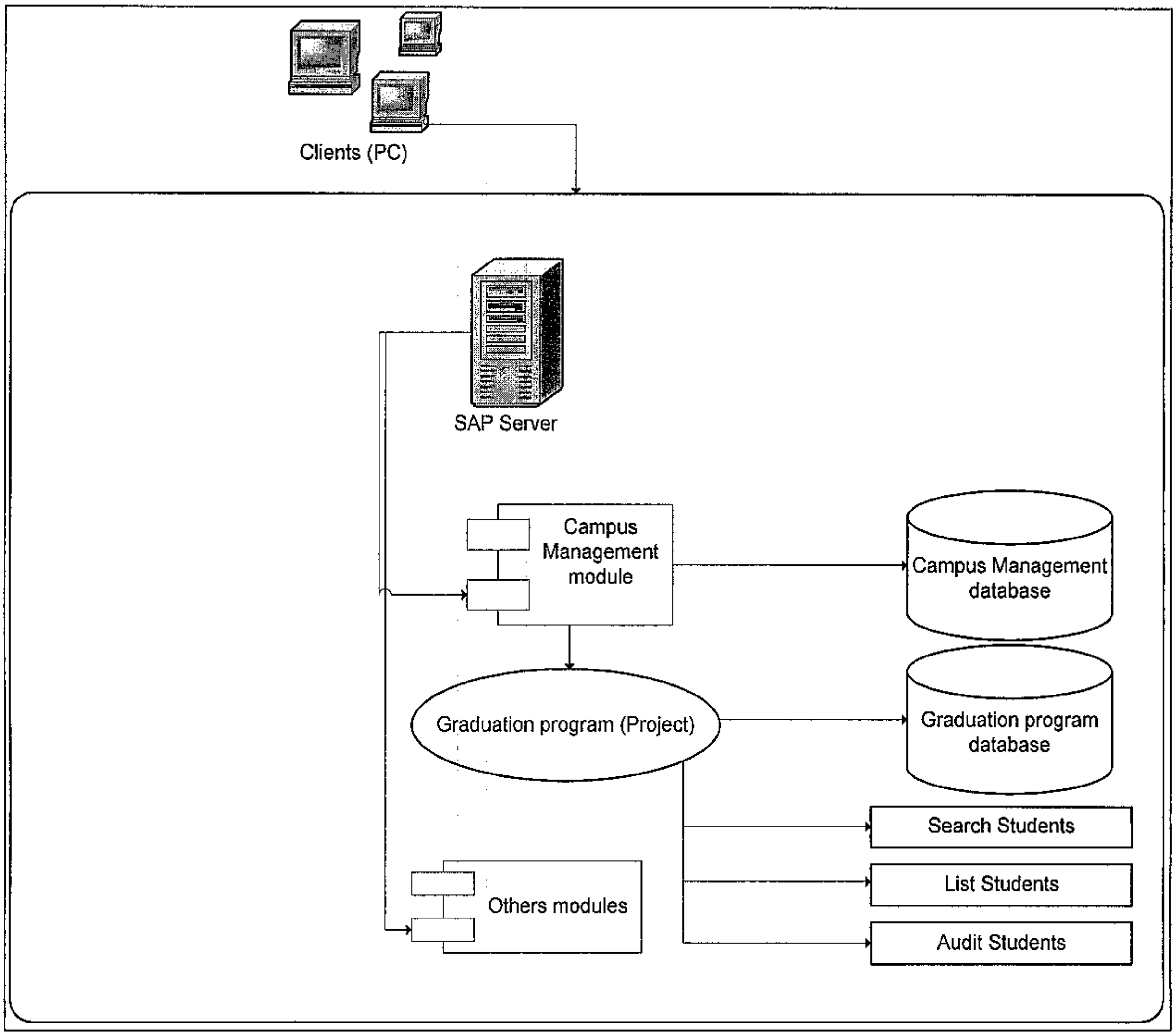


Figure 4.3: System architecture

4.1.4 Database Design

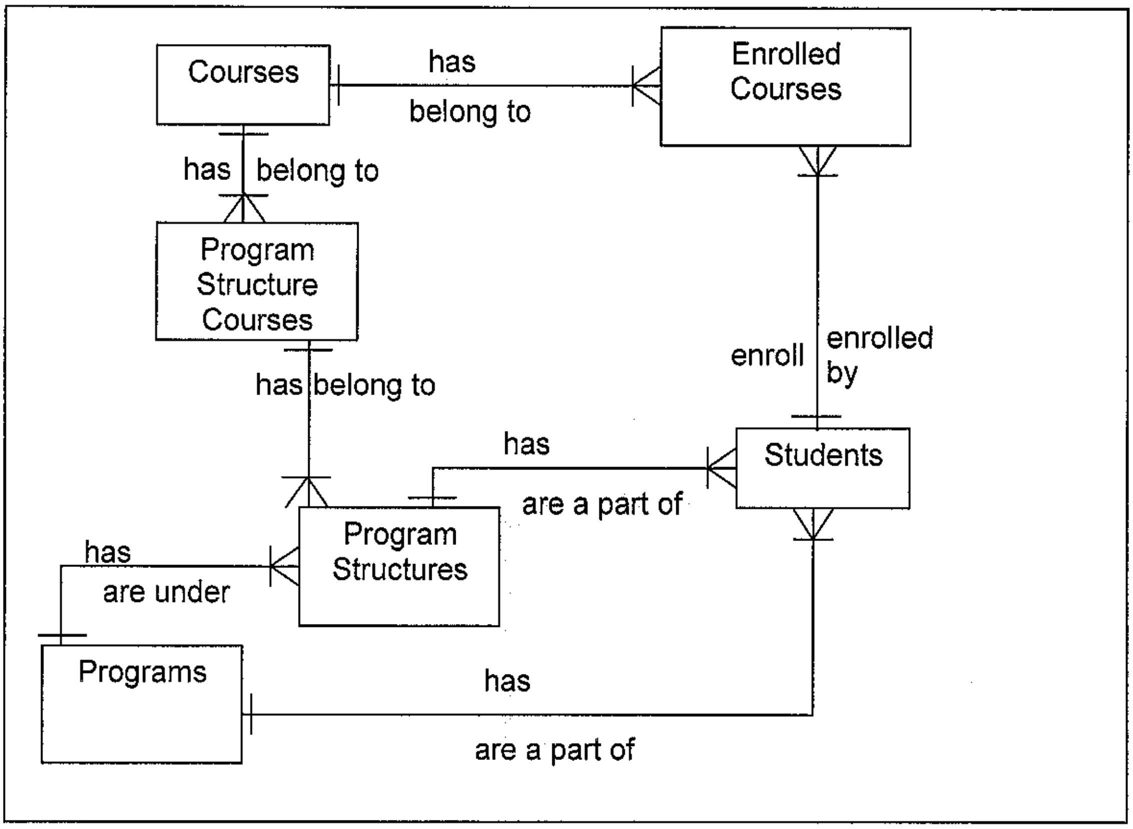


Figure 4.4: Entity-Relationship Diagram

The ERD above (see Figure 4.4) shows the relationship between six entities within the system. *Students* entity represents the students within the university. *Enrolled Courses* entity represents all the courses that have been enrolled by *Students* entity. *Courses* entity is all the courses that are offered by the university. *Programs* entity is the program of study the university has to offer to students. *Program Structures* entity is the type of structure of a program of study since university may have multiple program structures for a certain program. *Program Structure Courses* entity represents the list of courses that a student must complete in order to be eligible to graduate.

Next page is the database design that has been developed in the system.

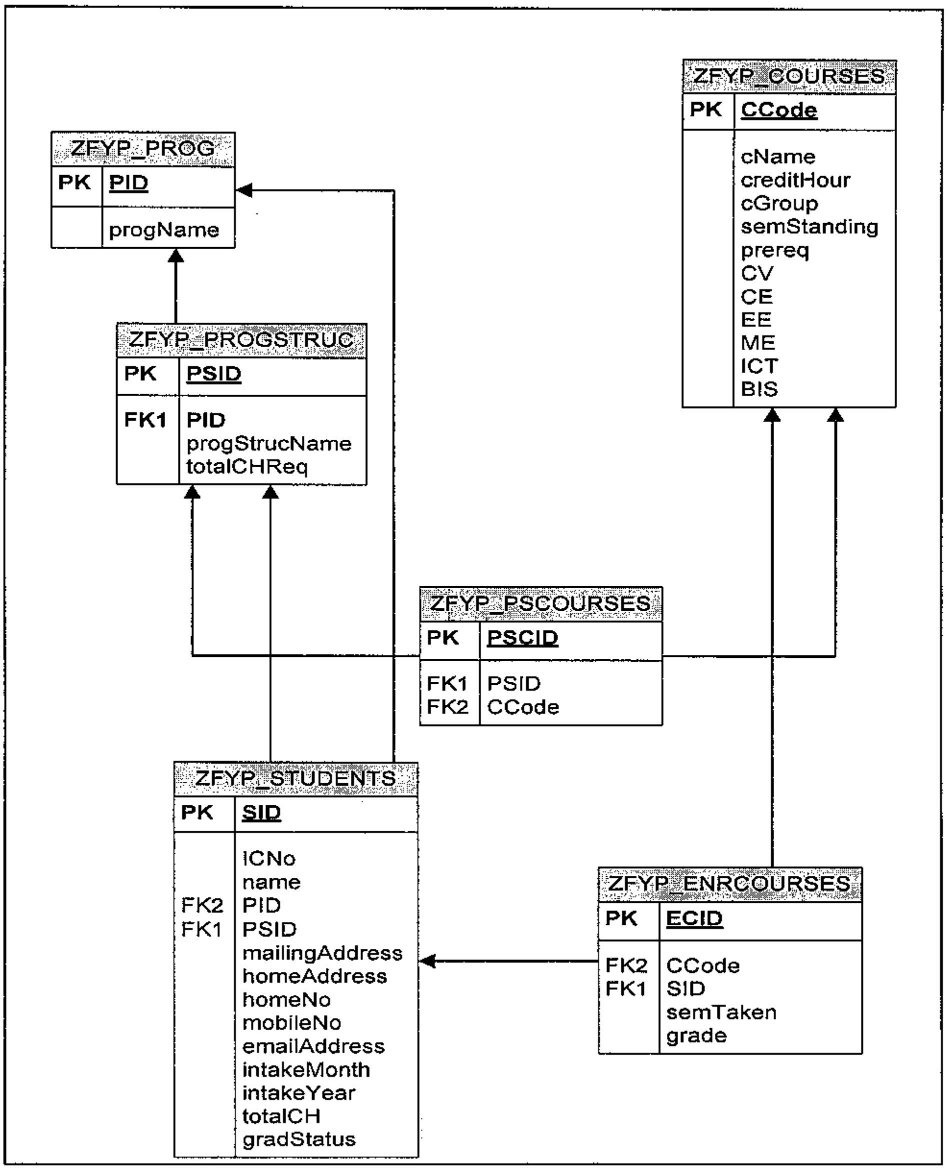
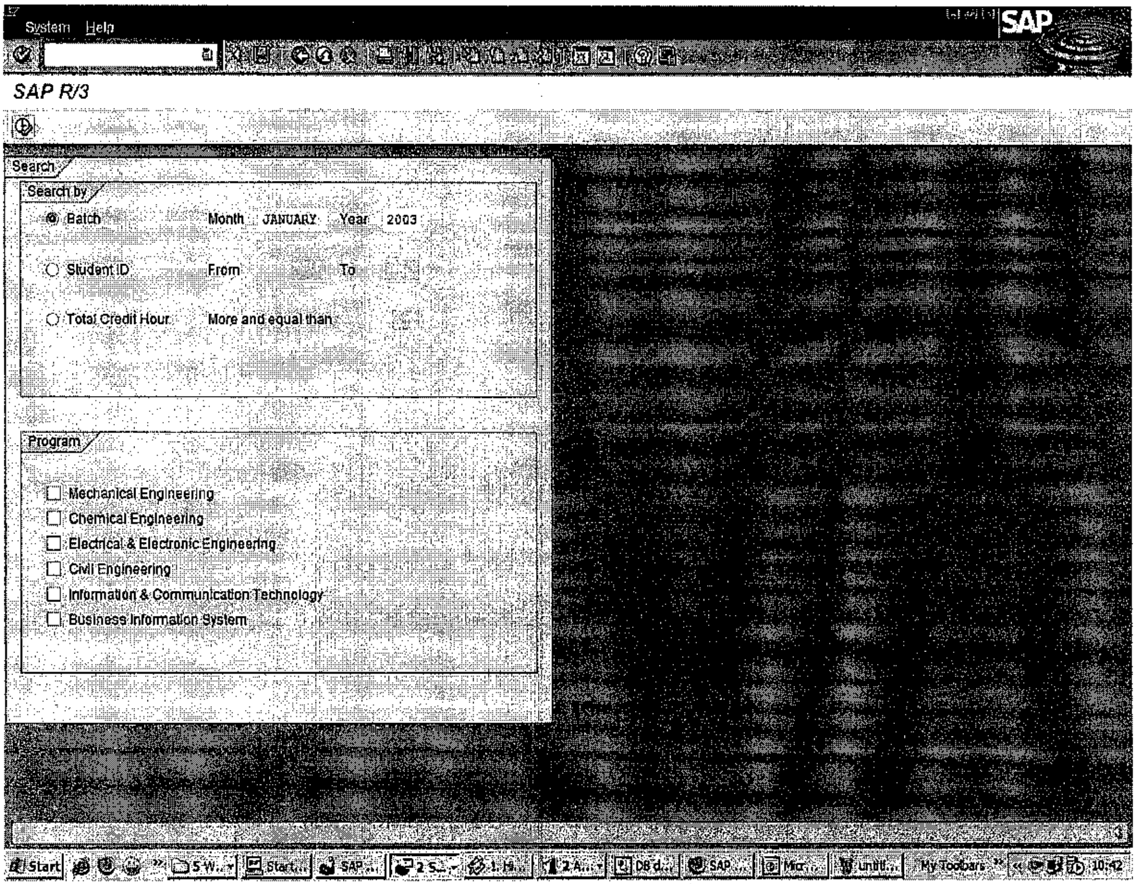


Figure 4.5: Database Design

The ERD and database design (see Figure 4.4 and Figure 4.5 respectively) are structured based on UTP student handbook ("Undergraduate Handbook", 2002: 5). As above, Figure 4.5 shows the connections among the tables within the database. The names used within Figure 4.5 are the tables' physical names.

4.1.5 User Interface

User interfaces and screens for the program are still in drafts. The interfaces and screens will be drawn out during the implementation phase of the program. Below is the initial screen for the program (see Figure 4.6).



Search

Search by

● Batch Month JANUARY Year 2003

○ Student ID From To

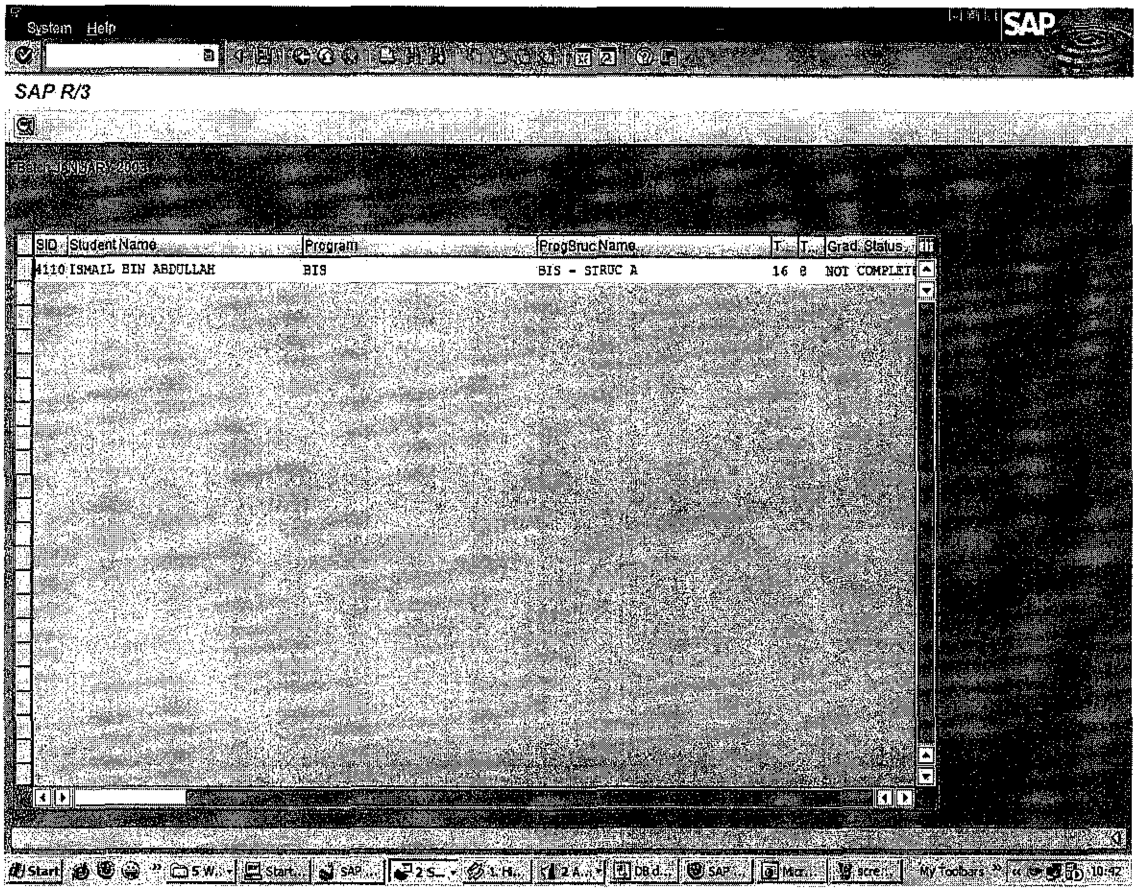
○ Total Credit Hour More and equal than

Program

□ Mechanical Engineering
□ Chemical Engineering
□ Electrical & Electronic Engineering
□ Civil Engineering
□ Information & Communication Technology
□ Business Information System

Figure 4.6: 1st screen (Screen 100)

The program is started by selecting the required batch of students that users want to check on. For example, an admin staff wants to get the student list for ICT and BIS student of Jan 2002 which is scheduled to be graduated next semester. Other than that, admin staffs may also search students by their ID and total credit hour. After that, the staff may check the checkboxes for to filter students' programs. Then, user need to click button "Search" or its shortcut, "F5" button; to go to the next screen.

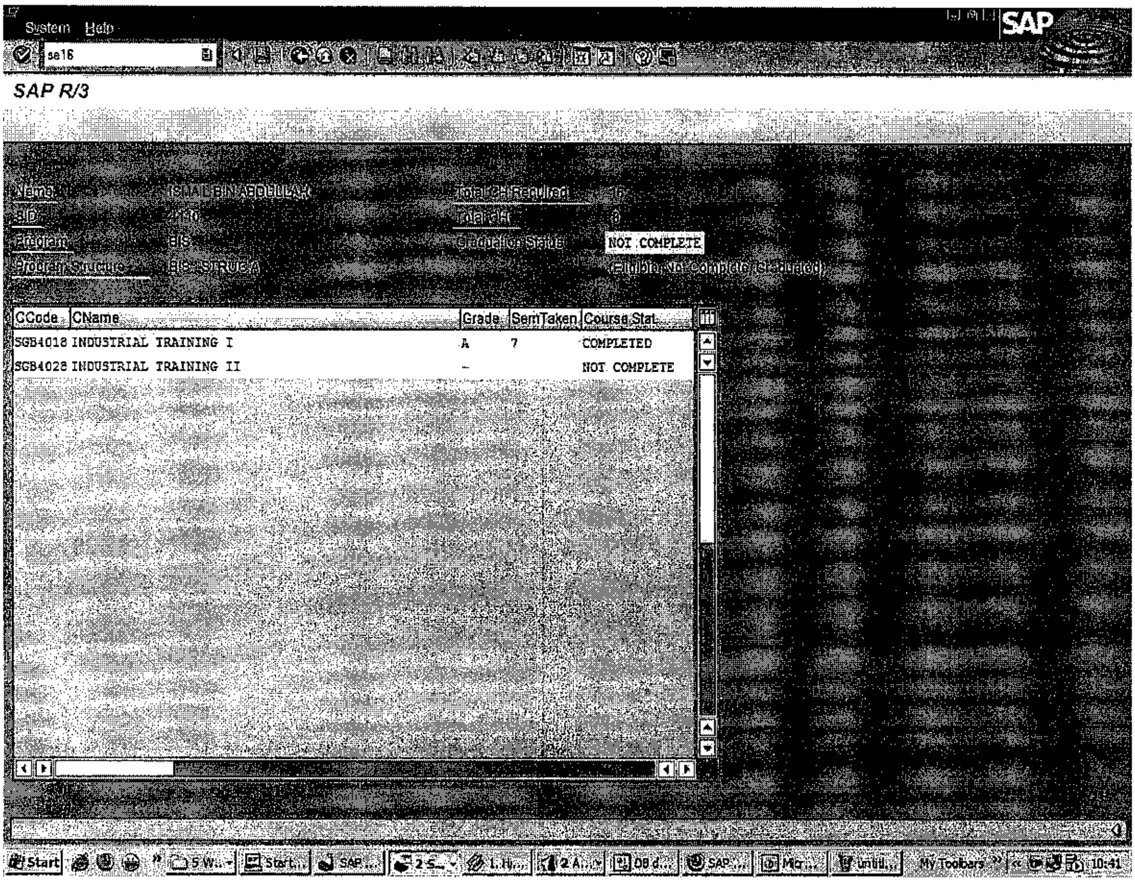


Batch JANUARY 2003

SID	Student Name	Program	ProgSrucName	T...	T...	Grad. Status
4110	ISMAIL BIN ABDULLAH	BIS	BIS - STRUC A	16	8	NOT COMPLETI

Figure 4.7: 2nd screen (Screen 200)

Next, list of students for the selected batch will appear and admin staff will do the audit for each of the student. As above (see Figure 4.7), in order to audit a student, admin staff has to select a student whether by clicking "Select" button or by double-click a student ID from the list of the students within the table control. Next page is showing the interface after clicking the "Select" button or after the double-clicking event.



Name ISMAIL BIN ABDULLAH
SID 2110
Program BIS
Program Structure BIS_STRUC A

Total CH Required 16
Total CH 8
Graduation Status NOT COMPLETE
(Eligible, Not Complete, Graduated)

CCode	CName	Grade	SemTaken	Course Stat.
SGB4018	INDUSTRIAL TRAINING I	A	7	COMPLETED
SGB4028	INDUSTRIAL TRAINING II	–		NOT COMPLETE

Figure 4.8: 3rd screen (Screen 300)

When a student from the list is selected, a detail list will appear as Figure 4.8 and shows which courses that a student has or has not taken before the student graduates. In the table, it will list down all courses related to the program the student is in. For example, if the student is a BIS student, all courses the student should take will appear in the table. If a course has been completed by the student, than it will say "Completed" in column "Course Stat." and it will say "Not Completed" if the student has not completed the course yet.

4.2 Discussion

4.2.1 Problems

Below were the problems that existed during the project development:

- The SAP system is owned by PETRONAS thus making changes on it need to undergo several strict procedures before the changes or enhancement can be committed.
- The user has access to the screen of the program only but the ABAP codes behind those programs are out of user's reach thus making it is harder for student to take a look at the ABAP codes to enhance any existing program.
- The documentation on the system may not be with the one who currently maintain it which is i-Perintis since the real developer of the system is not exist anymore.

Although there is already permission letter from Registrar, there was still a problem when dealing on accessing the UTP Campus Management system at that time. The problem was:

- Cannot access and have a look at the UTP Campus Management system such as its interfaces and database structures due to confidentiality reasons. Although by using student own details, the permission to have a look at the system cannot be given.

4.2.2 Solutions

In the first three problems in section 4.2.1, a separate development from application or real system was being used. Thus, having a separate database but mimicking the real database was used. But, without having the access to see the system, Mr. Azlan Zahuri can only gave a sample SAP generated report of graduation check list in order to help with the project. The sample of the graduation check list is attached in appendix section by the end of this report. From the check list sample, the existing database design was adjusted accordingly.

5. CONCLUSION & RECOMMENDATION

5.1. Conclusion

SAP Campus Management is a robust and advanced solution and is primarily designed to support the management of teaching and studies in general and the progress of each individual student. It supports the process of student registration, examination and grading, student progression, and up to student graduation stage. Other than that, the solution is available to solve the complex requirements of each member of the campus community from executives and administrators to students and facility. Through this project, both student and UTP administration benefit from it. Student will gain his necessary experience as a consultant and to learn more about SAP and SAP Campus Management application and UTP administration may get the opportunity to improve its system and to improve the productivity of its users for free.

5.2. Recommendation

The project is a stand-alone program and is not attached and is not connected to UTP SAP Campus Management module. It is highly recommended to future FYP students who has interest to doing further work in this title to connect it to a SAP Campus Management module and if possible within UTP SAP Campus Management module. A function that can immediately notice students through e-mail regarding their graduation status is also another interesting part that can be looked for further improvement on this program.

6. REFERENCE

1. "SAP® Campus Management" (2005) [Online]: SAP AG, [Online], available: http://www.sap.com/industries/highered/pdf/BWP_Campus_Mgt.pdf [19 May 2006].
2. "SAP® Campus Management" (2005) [Online]: SAP AG, [Online], available: http://www.onepurdue.purdue.edu/contribute_pdf/SAPCampusManagementbrochure.pdf [19 May 2006].
3. "Student Master Data – SAP Screen Shot" (2004): University of Mississippi.
4. D. Alan, H.W. Barbara, and T. David (2002), "System Analysis & Design – An Object-Oriented Approach with UML": John Wiley & Sons Inc.
5. "Undergraduate Handbook" (2002): Universiti Teknologi PETRONAS.
6. E. K. Kenneth and E. K. Julie (2002), "Systems Analysis and Design": Prentice Hall.
7. L.S. Vicki (2000), "Entity-Relationship Diagram": University of Missouri, [Online], available: http://www.umsl.edu/~sauter/analysis/er/er_intro.html [19 May 2006].
8. Software Design Center (2006), "HOW TO DRAW ENTITY RELATIONSHIP DIAGRAMS": SmartDraw, [Online], available: http://www.smartdraw.com/tutorials/software-erd/erd.htm [19 May 2006].
9. Faculty of Informatics and Communication (2006), "Developing Entity Relationship Diagrams (ERDs)": Central Queensland University, [Online], available: http://infocom.cqu.edu.au/Courses/spr2000/95169/Extra_Examples/ERD.htm [19 May 2006].

7. APPENDIX

7.1. Developer Manual

7.1.1. Database Tables

<table>
<tr><th colspan="7">ZFYP_PROG</th></tr>
<tr><td colspan="2">Short description</td><td colspan="5">List of Program of Study</td></tr>
<tr><th colspan="7">Attributes</th></tr>
<tr><td colspan="2">Table category</td><td colspan="5">Transparent table</td></tr>
<tr><td colspan="2">Delivery class</td><td colspan="5">A</td></tr>
<tr><td colspan="2">Table maintenance</td><td colspan="5">Allowed</td></tr>
<tr><th colspan="7">Fields</th></tr>
<tr><th>Fields</th><th>Key</th><th>Init.</th><th>Frgn. Key</th><th>Data</th><th>Lgth.</th><th>Short text</th></tr>
<tr><td>PID</td><td>X</td><td>X</td><td></td><td>CHAR</td><td>4</td><td>Program ID</td></tr>
<tr><td>PROGNAME</td><td></td><td></td><td></td><td>CHAR</td><td>30</td><td>Program of Study Name</td></tr>
</table>

<table>
<tr><th colspan="7">ZFYP_PROGSTRUC</th></tr>
<tr><td colspan="2">Short description</td><td colspan="5">List of Program of Study Structure</td></tr>
<tr><th colspan="7">Attributes</th></tr>
<tr><td colspan="2">Table category</td><td colspan="5">Transparent table</td></tr>
<tr><td colspan="2">Delivery class</td><td colspan="5">A</td></tr>
<tr><td colspan="2">Table maintenance</td><td colspan="5">Allowed</td></tr>
<tr><th colspan="7">Fields</th></tr>
<tr><th>Fields</th><th>Key</th><th>Init.</th><th>Frgn. Key</th><th>Data</th><th>Lgth.</th><th>Short text</th></tr>
<tr><td>PSID</td><td>X</td><td>X</td><td></td><td>CHAR</td><td>5</td><td>Structure of Program of Study ID</td></tr>
<tr><td>PID</td><td></td><td></td><td>X</td><td>CHAR</td><td>4</td><td>Program ID</td></tr>
<tr><td>PROGSTRUCNAME</td><td></td><td></td><td></td><td>CHAR</td><td>30</td><td>Structure of Program of Study Name</td></tr>
<tr><td>TOTALCHREQ</td><td></td><td></td><td></td><td>INT1</td><td>3</td><td>Total Credit Hour Required for Graduating</td></tr>
</table>

<table>
<tr><td colspan="7"><b>ZFYP_COURSES</b></td></tr>
<tr><td colspan="2">Short description</td><td colspan="5">List of Courses</td></tr>
<tr><td colspan="7"><b>Attributes</b></td></tr>
<tr><td colspan="2">Table category</td><td colspan="5">Transparent table</td></tr>
<tr><td colspan="2">Delivery class</td><td colspan="5">A</td></tr>
<tr><td colspan="2">Table maintenance</td><td colspan="5">Allowed</td></tr>
<tr><td colspan="7"><b>Fields</b></td></tr>
<tr><td><b>Fields</b></td><td><b>Key</b></td><td><b>Init.</b></td><td><b>Frgn. Key</b></td><td><b>Data</b></td><td><b>Lgth.</b></td><td><b>Short text</b></td></tr>
<tr><td>CCODE</td><td>X</td><td>X</td><td></td><td>CHAR</td><td>7</td><td>Course Code</td></tr>
<tr><td>CNAME</td><td></td><td></td><td></td><td>CHAR</td><td>50</td><td>Course Name</td></tr>
<tr><td>CREDITHOUR</td><td></td><td></td><td></td><td>INT1</td><td>30</td><td>Course's Credit Hour</td></tr>
<tr><td>CGROUP</td><td></td><td></td><td></td><td>CHAR</td><td>10</td><td>Course Group</td></tr>
<tr><td>SEMSTANDING</td><td></td><td></td><td></td><td>INT1</td><td>3</td><td>Semester Standing Required to Enroll a Course</td></tr>
<tr><td>PREREQ</td><td></td><td></td><td></td><td>CHAR</td><td>7</td><td>Pre-Requisite Course Needed to Enroll for a Course</td></tr>
<tr><td>CV</td><td></td><td></td><td></td><td>CHAR</td><td>1</td><td>Is a course is for Civil Engineering student?</td></tr>
<tr><td>CE</td><td></td><td></td><td></td><td>CHAR</td><td>1</td><td>Is a course is for Chemical Engineering student?</td></tr>
<tr><td>EE</td><td></td><td></td><td></td><td>CHAR</td><td>1</td><td>Is a course is for EE Engineering student?</td></tr>
<tr><td>ME</td><td></td><td></td><td></td><td>CHAR</td><td>1</td><td>Is a course is for Mechanical Engineering student?</td></tr>
<tr><td>ICT</td><td></td><td></td><td></td><td>CHAR</td><td>1</td><td>Is a course is for ICT student?</td></tr>
<tr><td>BIS</td><td></td><td></td><td></td><td>CHAR</td><td>1</td><td>Is a course is for BIS student?</td></tr>
</table>

<table>
<tr><th colspan="7">ZFYP_PSCOURSES</th></tr>
<tr><td colspan="2">Short description</td><td colspan="5">List of Courses for a Program Structure</td></tr>
<tr><th colspan="7">Attributes</th></tr>
<tr><td colspan="2">Table category</td><td colspan="5">Transparent table</td></tr>
<tr><td colspan="2">Delivery class</td><td colspan="5">A</td></tr>
<tr><td colspan="2">Table maintenance</td><td colspan="5">Allowed</td></tr>
<tr><th colspan="7">Fields</th></tr>
<tr><th>Fields</th><th>Key</th><th>Init.</th><th>Frgn. Key</th><th>Data</th><th>Lgth.</th><th>Short text</th></tr>
<tr><td>PSCID</td><td>X</td><td>X</td><td></td><td>NUMC</td><td>9</td><td>ID for Courses of a Program Structure</td></tr>
<tr><td>PSID</td><td></td><td></td><td>X</td><td>CHAR</td><td>5</td><td>Structure of Program of Study ID</td></tr>
<tr><td>CCODE</td><td></td><td></td><td>X</td><td>CHAR</td><td>7</td><td>Course Code</td></tr>
</table>

ZFYP_STUDENTS	
Short description	List of Students and Their Details
Attributes	
Table category	Transparent table
Delivery class	A
Table maintenance	Allowed
Fields	

Fields	Key	Init.	Frgn. Key	Data	Lgth.	Short text
SID	X	X		NUMC	4	Student ID
ICNO				NUMC	12	Identification Card Number
NAME				CHAR	30	Student Name
PID			X	CHAR	4	Program ID
PSID			X	CHAR	5	Structure of Program of Study ID
MAILINGADDRESS				CHAR	50	Address for Mailing Purpose
HOMEADDRESS				CHAR	50	Home or Residential Address for a Student
HOMENO				CHAR	10	Home Phone Number
MOBILENO				CHAR	10	Mobile Phone Number
EMAILADDRESS				CHAR	40	Email Address
INTAKEMONTH				CHAR	10	Month of Intake of a Student
INTAKEYEAR				NUMC	4	Year of Intake
TOTALCH				INT1	3	Total Credit Hour a Student has Taken
GRADSTATUS				CHAR	12	Graduation Status of a Student

<table>
<tr><th colspan="7">ZFYP_ENRCOURSES</th></tr>
<tr><td colspan="2">Short description</td><td colspan="5">Enrolled Courses by Students</td></tr>
<tr><th colspan="7">Attributes</th></tr>
<tr><td colspan="2">Table category</td><td colspan="5">Transparent table</td></tr>
<tr><td colspan="2">Delivery class</td><td colspan="5">A</td></tr>
<tr><td colspan="2">Table maintenance</td><td colspan="5">Allowed</td></tr>
<tr><th colspan="7">Fields</th></tr>
<tr><th>Fields</th><th>Key</th><th>Init.</th><th>Frgn. Key</th><th>Data</th><th>Lgth.</th><th>Short text</th></tr>
<tr><td>ECID</td><td>X</td><td>X</td><td></td><td>NUMC</td><td>9</td><td>Enrolled Courses ID</td></tr>
<tr><td>CCODE</td><td></td><td></td><td>X</td><td>CHAR</td><td>7</td><td>Course Code</td></tr>
<tr><td>SID</td><td></td><td></td><td>X</td><td>NUMC</td><td>4</td><td>Student ID</td></tr>
<tr><td>GRADE</td><td></td><td></td><td></td><td>CHAR</td><td>2</td><td>Course Grade</td></tr>
<tr><td>SEMTAKEN</td><td></td><td></td><td></td><td>INT1</td><td>3</td><td>Semester which a Course was Taken</td></tr>
</table>

7.1.2. Structures

ZSSTUDENTS			
Short description	Structure Type for Students		
Components			
Components	**Data Type**	**Length**	**Short Text**
SID	NUMC	4	Student ID
NAME	CHAR	3	Student Name
PID	CHAR	4	Program ID
PROGNAME	CHAR	30	Program of Study Name
PSID	CHAR	5	Structure of Program of Study ID
PROGSTRUCNAME	CHAR	30	Structure of Program of Study Name
TOTALCHREQ	INT1	3	Total Credit Hour Required for Graduating
TOTALCH	INT1	3	Total Credit Hour a Student has Taken
GRADSTATUS	CHAR	12	Graduation Status of a Student
MARK	CHAR	1	Mark a Student

ZSPSCOURSESTAT			
Short description	Structure Type for Program Structure Course Status		
Components			
Components	**Data Type**	**Length**	**Short Text**
PSID	CHAR	5	Structure of Program of Study ID
CCODE	CHAR	7	Course Code
CNAME	CHAR	50	Course Name
GRADE	CHAR	2	Course Grade
SEMTAKEN	INT1	3	Semester which a Course was Taken
COURSESTAT	CHAR	12	Course Status for a Student

7.1.3. Screens

SCREEN 100					
Short description	Screen 100 - Search students				
Attributes					
Screen type	Normal				
Next screen	100				
Element List					
Name	**Type**	**Input**	**Output**	**Function Code**	**Text or I/O Field**
BOX_SEARCH	Box				Search_____________...
BOX_SEARCHBY	Box				Search_by__________...
RBTN_BATCH	Radio	X	X		
RBTN_BATCH	Radio				Batch
TXT_INTAKEMONTH	Text				Month
IN_INTAKEMONTH	I/O	X	X		__________
TXT_INTAKEYEAR	Text				Year
IN_INTAKEYEAR	I/O	X	X		_________
RBTN_SID	Radio	X	X		
RBTN_SID	Radio				Student_ID
TXT_SID_FROM	Text				From_
IN_SID_FROM	I/O		X		____
TXT_SID_TO	Text				To___
IN_SID_TO	I/O		X		____
RBTN_TCH	Radio	X	X		
RBTN_TCH	Radio				Total_Credit_Hour
TXT_TCH	Text				More_and_equal_than
IN_TCH	I/O		X		____
BOX_PROGRAM	Box				Program_____________...
CBOX_ME	Check	X	X		

CBOX_ME	Check				Mechanical_Engineering
CBOX_CE	Check	X	X		
CBOX_CE	Check				Chemical_Engineering
CBOX_EE	Check	X	X		
CBOX_EE	Check				Electrical_&_Electronic_Engineering
CBOX_CV	Check	X	X		
CBOX_CV	Check				Civil_Engineering
CBOX_ICT	Check	X	X		
CBOX_ICT	Check				Info._&_Comm_Technology
CBOX_BIS	Check	X	X		
CBOX_BIS	Check				Business_Info._System
Flow Logic					
PBO	MODULE STATUS_0100. MODULE RADIOBUTTON.				
PAI	MODULE SAVE_OK_CODE. MODULE USER_COMMAND_0100.				
GUI Status					
Status name	S_100				
Application toolbar	Item 1 – SEARCH				
Function keys	Standard Toolbar: • BACK • EXIT Freely assigned function keys: • F5 - SEARCH				

<table>
<tr><th colspan="6">SCREEN 200</th></tr>
<tr><td colspan="2">Short description</td><td colspan="4">Screen 200 - Listing Students</td></tr>
<tr><th colspan="6">Attributes</th></tr>
<tr><td colspan="2">Screen type</td><td colspan="4">Normal</td></tr>
<tr><td colspan="2">Next screen</td><td colspan="4">200</td></tr>
<tr><th colspan="6">Element List</th></tr>
<tr><th>Name</th><th>Type</th><th>Input</th><th>Output</th><th>Function Code</th><th>Text or I/O Field</th></tr>
<tr><td>OUT_SEARCHTITLE</td><td>I/O</td><td>X</td><td>X</td><td></td><td>____________________...</td></tr>
<tr><td>TCL_STUDENTS</td><td>Table</td><td></td><td></td><td></td><td></td></tr>
<tr><td>ZSSTUDENTS-SID</td><td>Text</td><td></td><td></td><td></td><td>SID_________</td></tr>
<tr><td>ZSSTUDENTS-NAME</td><td>Text</td><td></td><td></td><td></td><td>Student_Name___</td></tr>
<tr><td>ZSSTUDENTS-PROGNAME</td><td>Text</td><td></td><td></td><td></td><td>Program_Name_____</td></tr>
<tr><td>ZSSTUDENTS-PROGSTRUCNAME</td><td>Text</td><td></td><td></td><td></td><td>ProgStruc_Name</td></tr>
<tr><td>ZSSTUDENTS-TOTALCHREQ</td><td>Text</td><td></td><td></td><td></td><td>Total_CH_Required</td></tr>
<tr><td>ZSSTUDENTS-TOTALCH</td><td>Text</td><td></td><td></td><td></td><td>Total_CH________</td></tr>
<tr><td>ZSSTUDENTS-GRADSTATUS</td><td>Text</td><td></td><td></td><td></td><td>Grad._Status_____</td></tr>
<tr><td>ZSSTUDENTS-MARK</td><td>Check</td><td>X</td><td>X</td><td></td><td></td></tr>
<tr><td>ZSSTUDENTS-SID</td><td>I/O</td><td>X</td><td>X</td><td></td><td>_________...</td></tr>
<tr><td>ZSSTUDENTS-NAME</td><td>I/O</td><td>X</td><td>X</td><td></td><td>_________...</td></tr>
<tr><td>ZSSTUDENTS-PROGNAME</td><td>I/O</td><td>X</td><td>X</td><td></td><td>_________...</td></tr>
<tr><td>ZSSTUDENTS-PROGSTRUCNAME</td><td>I/O</td><td>X</td><td>X</td><td></td><td>_________...</td></tr>
</table>

ZSSTUDENTS-TOTALCHREQ	I/O	X	X		________...
ZSSTUDENTS-TOTALCH	I/O	X	X		________...
ZSSTUDENTS-GRADSTATUS	I/O	X	X		________...
OK_CODE	OK				
Flow Logic					
PBO	MODULE STATUS_0200. MODULE FILL_ITAB_STUDENTS. MODULE SET_SEARCHTITLE. LOOP AT itab_students INTO wa_students WITH CONTROL tcl_students. MODULE MOVE_TO_ZSSTUDENTS. ENDLOOP.				
PAI	LOOP AT itab_students. ENDLOOP. MODULE SAVE_OK_CODE. MODULE USER_COMMAND_0200.				
GUI Status					
Status name	S_200				
Application toolbar	Item 1 – SELECT				
Function keys	Standard Toolbar: • BACK • EXIT Freely assigned function keys: • F5 - SELECT				

SCREEN 300					
Short description	Screen 300 - Detail List of a Student				
Attributes					
Screen type	Normal				
Next screen	300				
Element List					
Name	**Type**	**Input**	**Output**	**Function Code**	**Text or I/O Field**
TXT_NAME	Text				Name
OUT_NAME	I/O	X	X		_______________...
TXT_TOTALCHREQ	Text				Total_CH_Required
OUT_TOTALCHREQ	I/O	X	X		___
TXT_SID	Text				SID
OUT_SID	I/O	X	X		____
TXT_TOTALCH	Text				Total_CH
OUT_TOTALCH	I/O	X	X		___
TXT_PROG	Text				Program
OUT_PROG	I/O	X	X		________________...
TXT_GRADSTAT	Text				Graduation_Status
IN_GRADSTAT	I/O	X	X		__________...
TXT_PROGSTRUC	Text				Program_Structure
OUT_PROGSTRUC	I/O	X	X		_________________...
TXT_GRADSTATHELP	Text				(Eligible,_Not_Complete,_Graduated
TCL_PSCOURSESTAT	Table				
ZSPSCOURSESTAT-CCODE	Text				CCode_________
ZSPSCOURSESTAT-NAME	Text				CName____
ZSPSCOURSESTAT-GRADE	Text				Grade

ZSPSCOURSESTAT-SEMTAKEN	Text				SemTaken___
ZSPSCOURSESTAT-COURSESTAT	Text				Course_Stat___
ZSPSCOURSESTAT-CCODE	I/O				_______________...
ZSPSCOURSESTAT-NAME	I/O				_______________...
ZSPSCOURSESTAT-GRADE	I/O				_______________...
ZSPSCOURSESTAT-SEMTAKEN	I/O				_______________...
ZSPSCOURSESTAT-COURSESTAT	I/O				_______________...
OK_CODE	OK				

Flow Logic	
PBO	MODULE STATUS_0300. MODULE FILL_STUDENT_DETAILS. MODULE FILL_ITAB_PSCOURSESTAT. LOOP AT itab_pscoursestat INTO wa_pscoursestat WITH CONTROL tcl_pscoursestat. MODULE MOVE_TO_ZSPSCOURSESTAT. ENDLOOP.
PAI	LOOP AT itab_pscoursestat. ENDLOOP. MODULE SAVE_OK_CODE. MODULE USER_COMMAND_0300.

GUI Status	
Status name	S_300
Function keys	Standard Toolbar: • BACK • EXIT • CANCEL

7.1.4. Include Files

ZFYP_GRADPROG_F01

```
*----------------------------------------------------------------------*
*   INCLUDE ZFYP_GRADPROG_F01                                          *
*----------------------------------------------------------------------*
*&---------------------------------------------------------------------*
*&      Form  radiobutton
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form radiobutton.
  IF rbtn_batch = 'X'.
      CLEAR: in_sid_from, in_sid_to, in_tch.
  ENDIF.

  IF rbtn_sid = 'X'.
    LOOP AT SCREEN.
      IF screen-name = 'IN_SID_FROM'
       OR screen-name = 'IN_SID_TO'.
        screen-input = 1.
      ENDIF.

      IF screen-name = 'IN_INTAKEMONTH'
       OR screen-name = 'IN_INTAKEYEAR'.
        screen-input = 0.
      ENDIF.

      MODIFY SCREEN.
    ENDLOOP.

    CLEAR: in_intakemonth, in_intakeyear, in_tch.
  ENDIF.

  IF rbtn_tch = 'X'.
    LOOP AT SCREEN.
      IF screen-name = 'IN_TCH'.
        screen-input = 1.
      ENDIF.
      IF screen-name = 'IN_INTAKEMONTH'
       OR screen-name = 'IN_INTAKEYEAR'.
        screen-input = 0.
      ENDIF.
      MODIFY SCREEN.
    ENDLOOP.

    CLEAR: in_intakemonth, in_intakeyear, in_sid_from, in_sid_to.
  ENDIF.

endform.                           " radiobutton
```

```
*&---------------------------------------------------------------------*
*&      Form  set_searchtitle_screen_200
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form set_searchtitle_screen_200.
  IF rbtn_batch = 'X'.
    s1 = 'Batch'.
    s2 = in_intakemonth.
    s3 = in_intakeyear.

    CONCATENATE s1 s2 s3 INTO s_combine SEPARATED BY SPACE.

    out_searchtitle = s_combine.
  ENDIF.

  IF rbtn_sid = 'X'.
    s1 = 'SID from'.
    s2 = in_sid_from.
    s3 = 'to'.
    s4 = in_sid_to.

    CONCATENATE s1 s2 s3 s4 INTO s_combine SEPARATED BY SPACE.

    out_searchtitle = s_combine.
  ENDIF.

  IF rbtn_tch = 'X'.
    s1 = 'Total credit'.
    s2 = 'hour more and'.
    s3 = 'equal than'.
    s4 = in_tch.

    CONCATENATE s1 s2 s3 s4 INTO s_combine SEPARATED BY SPACE.

    out_searchtitle = s_combine.
  ENDIF.

endform.                    " set_searchtitle_screen_200
```

```
*&---------------------------------------------------------------------*
*&      Form  select_student_by_batch
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form select_student_by_batch.
  SELECT zfyp_students~sid zfyp_students~name zfyp_students~pid
         zfyp_students~psid zfyp_students~totalch
         zfyp_students~gradstatus
         zfyp_prog~progname
         zfyp_progstruc~progstrucname zfyp_progstruc~totalchreq
   INTO CORRESPONDING FIELDS OF TABLE itab_students
   FROM zfyp_students INNER JOIN zfyp_prog
   ON zfyp_students~pid = zfyp_prog~pid
   INNER JOIN zfyp_progstruc
   ON zfyp_prog~pid = zfyp_progstruc~pid
   WHERE zfyp_students~intakemonth = in_intakemonth
   AND zfyp_students~intakeyear = in_intakeyear.

endform.                    " select_student_by_batch
*&---------------------------------------------------------------------*
*&      Form  select_student_between_ids
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form select_student_between_ids.
    SELECT zfyp_students~sid zfyp_students~name zfyp_students~pid
           zfyp_students~psid zfyp_students~totalch
           zfyp_students~gradstatus
           zfyp_prog~progname
           zfyp_progstruc~progstrucname zfyp_progstruc~totalchreq
    INTO CORRESPONDING FIELDS OF TABLE itab_students
    FROM zfyp_students INNER JOIN zfyp_prog
    ON zfyp_students~pid = zfyp_prog~pid
    INNER JOIN zfyp_progstruc
    ON zfyp_prog~pid = zfyp_progstruc~pid
    WHERE zfyp_students~sid BETWEEN in_sid_from
    AND in_sid_to.

endform.                    " select_student_between_ids
```

```
*&---------------------------------------------------------------------*
*&      Form  select_student_by_id
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form select_student_by_id.
    SELECT zfyp_students~sid zfyp_students~name zfyp_students~pid
           zfyp_students~psid zfyp_students~totalch
           zfyp_students~gradstatus
           zfyp_prog~progname
           zfyp_progstruc~progstrucname zfyp_progstruc~totalchreq
    INTO CORRESPONDING FIELDS OF TABLE itab_students
    FROM zfyp_students INNER JOIN zfyp_prog
    ON zfyp_students~pid = zfyp_prog~pid
    INNER JOIN zfyp_progstruc
    ON zfyp_prog~pid = zfyp_progstruc~pid
    WHERE zfyp_students~sid = in_sid_from.

endform.                    " select_student_by_id
*&---------------------------------------------------------------------*
*&      Form  select_student_by_totalch
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form select_student_by_totalch.
  SELECT zfyp_students~sid zfyp_students~name zfyp_students~pid
         zfyp_students~psid zfyp_students~totalch
         zfyp_students~gradstatus
         zfyp_prog~progname
         zfyp_progstruc~progstrucname zfyp_progstruc~totalchreq
  INTO CORRESPONDING FIELDS OF TABLE itab_students
  FROM zfyp_students INNER JOIN zfyp_prog
  ON zfyp_students~pid = zfyp_prog~pid
  INNER JOIN zfyp_progstruc
  ON zfyp_prog~pid = zfyp_progstruc~pid
  WHERE zfyp_students~totalch GE in_tch.

endform.                    " select_student_by_totalch
```

```
*&---------------------------------------------------------------------*
*&      Form  fill_student_details
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form fill_student_details.
  out_sid =  wa_students-sid.
  out_name = wa_students-name.
  out_prog = wa_students-progname.
  out_progstruc = wa_students-progstrucname.
  out_totalch = wa_students-totalch.
  out_totalchreq = wa_students-totalchreq.

  in_gradstat = wa_students-gradstatus.

  get_psid = wa_students-psid.
endform.                    " fill_student_details
*&---------------------------------------------------------------------*
*&      Form  select_pscoursestat
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form select_pscoursestat.

  SELECT zfyp_pscourses~psid zfyp_pscourses~ccode
         zfyp_courses~cname
  INTO CORRESPONDING FIELDS OF TABLE itab_pscoursestat_temp1
  FROM zfyp_pscourses LEFT OUTER JOIN zfyp_courses
  ON zfyp_pscourses~ccode = zfyp_courses~ccode
  WHERE zfyp_pscourses~psid = get_psid.

  SELECT zfyp_enrcourses~ccode
         zfyp_enrcourses~semtaken zfyp_enrcourses~grade
  INTO CORRESPONDING FIELDS OF TABLE itab_pscoursestat_temp2
  FROM zfyp_enrcourses
  WHERE zfyp_enrcourses~sid = out_sid.
  LOOP AT itab_pscoursestat_temp1 INTO wa_pscoursestat_temp1.
    READ TABLE itab_pscoursestat_temp2 INTO wa_pscoursestat_temp2
     WITH KEY ccode = wa_pscoursestat_temp1-ccode.

    IF sy-subrc = 0.
      IF wa_pscoursestat_temp2-grade EQ 'F'.
        wa_pscoursestat-ccode      = wa_pscoursestat_temp1-ccode.
        wa_pscoursestat-cname      = wa_pscoursestat_temp1-cname.
        wa_pscoursestat-semtaken   = wa_pscoursestat_temp2-semtaken.
        wa_pscoursestat-grade      = wa_pscoursestat_temp2-grade.
        wa_pscoursestat-coursestat = 'FAILED'.
      ELSE.
        wa_pscoursestat-ccode      = wa_pscoursestat_temp1-ccode.
        wa_pscoursestat-cname      = wa_pscoursestat_temp1-cname.
        wa_pscoursestat-semtaken   = wa_pscoursestat_temp2-semtaken.
        wa_pscoursestat-grade      = wa_pscoursestat_temp2-grade.
        wa_pscoursestat-coursestat = 'COMPLETED'.
      ENDIF.
    ELSE.
      wa_pscoursestat-ccode      = wa_pscoursestat_temp1-ccode.
      wa_pscoursestat-cname      = wa_pscoursestat_temp1-cname.
      wa_pscoursestat-semtaken   = SPACE.
      wa_pscoursestat-grade      = '-'.
      wa_pscoursestat-coursestat = 'NOT COMPLETE'.
    ENDIF.

    APPEND wa_pscoursestat TO itab_pscoursestat.
    CLEAR: wa_pscoursestat_temp1, wa_pscoursestat_temp2.
  ENDLOOP.

endform.                     " select_pscoursestat
```

ZFYP_GRADPROG_I01

```
*----------------------------------------------------------------------*
*   INCLUDE ZFYP_GRADPROG_I01                                          *
*----------------------------------------------------------------------*
*&---------------------------------------------------------------------*
*&      Module  SAVE_OK_CODE  INPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
module SAVE_OK_CODE input.
  save_ok = ok_code.
  CLEAR ok_code.
endmodule.                 " SAVE_OK_CODE  INPUT

*&---------------------------------------------------------------------*
*&      Module  USER_COMMAND_0100  INPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
module USER_COMMAND_0100 input.
  CASE save_ok.
    WHEN 'BACK'.
      LEAVE TO SCREEN 0.
    WHEN 'EXIT'.
      LEAVE PROGRAM.
    WHEN 'SEARCH'.
      LEAVE TO SCREEN 200.
  ENDCASE.
endmodule.                 " USER_COMMAND_0100  INPUT

*&---------------------------------------------------------------------*
*&      Module  USER_COMMAND_0200  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
module USER_COMMAND_0200 input.
  CASE save_ok.
    WHEN 'BACK'.
      LEAVE TO SCREEN 100.
    WHEN 'EXIT'.
      LEAVE PROGRAM.
    WHEN 'SELECT'.
      LOOP AT tcl_students-cols INTO wa_students WHERE selected = 'X'.
      ENDLOOP.
      LEAVE TO SCREEN 300.

  ENDCASE.

endmodule.                 " USER_COMMAND_0200  input
```

```
*&---------------------------------------------------------------------*
*&      Module  USER_COMMAND_0300  INPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
module USER_COMMAND_0300 input.
  CASE save_ok.
    WHEN 'BACK'.
      LEAVE TO SCREEN 200.
    WHEN 'EXIT'.
      LEAVE PROGRAM.
    WHEN 'CANCEL'.
      LEAVE TO SCREEN 100.
  ENDCASE.

endmodule.                 " USER_COMMAND_0300  INPUT
```

ZFYP_GRADPROG_001

```
*----------------------------------------------------------------------*
*   INCLUDE ZFYP_GRADPROG_001                                          *
*----------------------------------------------------------------------*
*&---------------------------------------------------------------------*
*&      Module  STATUS_0100  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE status_0100 OUTPUT.
  SET PF-STATUS 'S_100'.
*  SET TITLEBAR 'xxx'.

ENDMODULE.                 " STATUS_0100  OUTPUT
*&---------------------------------------------------------------------*
*&      Module  STATUS_0200  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE status_0200 OUTPUT.
  SET PF-STATUS 'S_200'.
*  SET TITLEBAR 'xxx'.

ENDMODULE.                 " STATUS_0200  OUTPUT
*&---------------------------------------------------------------------*
*&      Module  STATUS_0300  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE status_0300 OUTPUT.
  SET PF-STATUS 'S_300'.
*  SET TITLEBAR 'xxx'.

ENDMODULE.                 " STATUS_0300  OUTPUT
*&---------------------------------------------------------------------*
*&      Module  FILL_ITAB_STUDENTS  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE fill_itab_students OUTPUT.
  IF in_intakemonth NE SPACE AND in_intakeyear NE SPACE.
    PERFORM select_student_by_batch.
  ENDIF.

  IF in_sid_from NE SPACE.
*   if both in_sid_from and in_sid_to have values
    IF in_sid_to NE SPACE.
      PERFORM select_student_between_ids.
    ELSE.
*     if in_sid_from has value only
      PERFORM select_student_by_id.
    ENDIF.
  ENDIF.

  IF in_tch NE SPACE.
    PERFORM select_student_by_totalch.
  ENDIF.

  DESCRIBE TABLE itab_students LINES tcl_students-lines.
ENDMODULE.                 " FILL_ITAB_STUDENTS  OUTPUT
*&---------------------------------------------------------------------*
*&      Module  RADIOBUTTON  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE radiobutton OUTPUT.
  PERFORM radiobutton.

ENDMODULE.                 " RADIOBUTTON  OUTPUT
```

```
*&---------------------------------------------------------------------*
*&      Module  MOVE_TO_ZSSTUDENTS  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE move_to_zsstudents OUTPUT.
  MOVE-CORRESPONDING wa_students TO zsstudents.
ENDMODULE.                 " MOVE_TO_ZSSTUDENTS  OUTPUT
*&---------------------------------------------------------------------*
*&      Module  SET_SEARCHTITLE  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE set_searchtitle OUTPUT.
  PERFORM set_searchtitle_screen_200.

ENDMODULE.                 " SET_SEARCHTITLE  OUTPUT
*&---------------------------------------------------------------------*
*&      Module  FILL_STUDENT_DETAILS  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
module FILL_STUDENT_DETAILS output.
  PERFORM fill_student_details.
endmodule.                 " FILL_STUDENT_DETAILS  OUTPUT

*&---------------------------------------------------------------------*
*&      Module  FILL_ITAB_PSCOURSESTAT  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
module FILL_ITAB_PSCOURSESTAT output.
  PERFORM select_pscoursestat.

  DESCRIBE TABLE itab_pscoursestat LINES tcl_pscoursestat-lines.
endmodule.                 " FILL_ITAB_PSCOURSESTAT  OUTPUT
*&---------------------------------------------------------------------*
*&      Module  MOVE_TO_ZSPSCOURSESTAT  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
module MOVE_TO_ZSPSCOURSESTAT output.
  MOVE-CORRESPONDING wa_pscoursestat TO zspscoursestat.
endmodule.                 " MOVE_TO_ZSPSCOURSESTAT  OUTPUT
```

ZFYP_GRADPROG_TOP

```
*&---------------------------------------------------------------------*
*& Include ZFYP_GRADPROG_TOP                                           *
*&                                                                     *
*&---------------------------------------------------------------------*

PROGRAM  ZFYP_GRADPROG                          .

TABLES: zsstudents,
        zspscoursestat.

*table controls
CONTROLS: tcl_students TYPE TABLEVIEW USING SCREEN '200',
          tcl_pscoursestat TYPE TABLEVIEW USING SCREEN '300'.

*internal tables
DATA: itab_students TYPE TABLE OF zsstudents,
      itab_pscoursestat TYPE TABLE OF zspscoursestat,
      itab_pscoursestat_temp1 TYPE TABLE OF zspscoursestat,
      itab_pscoursestat_temp2 TYPE TABLE OF zspscoursestat.

*work area
DATA: wa_students TYPE zsstudents,
      wa_pscoursestat TYPE zspscoursestat,
      wa_pscoursestat_temp1 TYPE zspscoursestat,
      wa_pscoursestat_temp2 TYPE zspscoursestat.

*ok code
DATA: ok_code LIKE sy-ucomm,
      save_ok LIKE ok_code.

*radio buttons at Screen 100
DATA: rbtn_batch(1) TYPE C,
      rbtn_sid(1) TYPE C,
      rbtn_tch(1) TYPE C.

*input fields at Screen 100
DATA: in_intakemonth(12) TYPE C,
      in_intakeyear(4) TYPE C,
      in_sid_from(4) TYPE C,
      in_sid_to(4) TYPE C,
      in_tch(3) TYPE C.

*output fields at Screen 200
DATA: out_searchtitle(50) TYPE C.

*strings for setting the search title at Screen 200
DATA: s_combine(50) TYPE C,
      s1(15) TYPE C,
      s2(15) TYPE C,
      s3(15) TYPE C,
      s4(15) TYPE C.

*input fields at Screen 300
DATA: in_gradstat(12) TYPE C.


*output fields at Screen 300
DATA: out_sid(4) TYPE C,
      out_name(30) TYPE C,
      out_prog(30) TYPE C,
      out_progstruc(30) TYPE C,
      out_totalchreq(3) TYPE C,
      out_totalch(3) TYPE C.

*temporary data for use in Screen 300
DATA: get_psid(5) TYPE C.
```

7.2. Graduation Checklist Samples

GRADUATION CHECKLIST (COURSE COMPLETION)
BACHELOR OF TECHNOLOGY (HONS) INFORMATION TECHNOLOGY

The Revised Programme Structures

Student's name : ______________________ Intake : ______________

Student's ID : ______________________ Exp. Graduate: ______________

Sex : ______________________

IC / Passport : ______________________

Date of birth ______________________

Citizenship : ______________________

Course Group	Code	Courses	Prev Course Description	Sem taken	Grade	Req Crh	Act crH	Remark
Foundation	STB1013	Computers & the Information Age				3		
	SMB1103	Foundation Mathematics I				3		
	SLB1000	English 1000				0		
	SLB1013	English I (Language Awareness & Study Skills)				3		
	STB1063	Programming I				3		
	SBB1013	Business Organisation				3		
	SEB1013	Economics I				3		
	SMB1123	Foundation Mathematics II				3		
	SLB1023	English II (Technical & Profesional Writing)				3		
	SAB1013	Business Accounting				3		
	SLB3013	Professional Communication Skill				3		
	SMB2033	Discrete Mathematics				3		
	SMB2133	Quantitative Methods				3		
		Total Credit hour				**36**		
Universiti Requirements	SHB2013	Malaysian Studies				3		
	SHB1013	Islamic Studies/Moral Studies				3		
Co-Curriculum						1		
Co-Curriculum						1		
Co-Curriculum						1		
	SLB1043	Bahasa Kebangsaan/M'ment Elective				3		
		Total Credit hour				**12**		
Core								
	STB1073	Programming II				3		
	SSB2063	System Analysis & Design				3		
	STB2023	Database Systems				3		
	STB2033	Computer Organisation				3		
	STB2053	Interactive Multimedia				3		
	STB2083	Visual Programming				3		
	STB2153	Data Structure				3		
	STB3023	Computer Graphics				3		
	STB3053	Object Oriented Programming				3		
	STB3063	Data Communications				3		
	STB3073	Software Engineering				3		
	STB3083	Internet Programming				3		
	STB3093	Computer Network Managemnt				3		
	STB3103	Distributed Computing				3		
	STB3113	Operating Systems				3		
	STB3133	Interactive Multimedia Instructions				3		
	STB3143	Human Computer Interaction				3		
	STB3153	Computer Project Management				3		
	STB4123	Networked Multimedia Systems				3		
	STB4132	Project - Part A				2		
	STB4142	Ethical and Professional Issues				2		
	STB4153	Software Testing & Reliability				3		
	STB5013	Computer Vision & Image Development				3		
	STB5023	Virtual Reality				3		
	STB5034	Project - Part B				4		
	STB5043	Computer Security				3		
	STB5053	Wireless Communication				3		
	SGB4018	Industrial Training I				8		
	SGB4028	Industrial Training II				8		
		Total Credit Hour				**96**		

GRADUATION CHECKLIST (COURSE COMPLETION)
BACHELOR OF TECHNOLOGY (HONS) INFORMATION TECHNOLOGY

The Revised Programme Structures

Student's name : ______________________

Student's ID : ______________________

Sex : ______________________

IC / Passport : ______________________

Date of birth ______________________

Citizenship : ______________________

Intake : ______________________

Exp. Graduate: ______________________

Course Group	Code	Courses	Prev Course Description	Sem taken	Grade	Req Crh	Act crH	Remarks
Minor								
Corporate	SNB2013	Organisational Behaviour				3		
Management	SWB1013	Business Cyber Law				3		
	SNB3033	Corporate Ethics				3		
	SNB4023	Strategics Management				3		
	SNB2023	Corporate Communication				3		
	SNB3023	Human Resource Management				3		
	SNB4013	Small Business & Entrepreneurship				3		
	SFB1013	Introduction to Finance				3		
Financial	SFB1013	Introduction to Finance				3		
Management	SFB3013	Corporate Finance				3		
	SFB3023	Multinational Corporate Finance & International Business				3		
	SNB2013	Organisational Behaviour				3		
	SNB2023	Corporate Communication				3		
	SNB3023	Human Resource Management				3		
	SNB4013	Small Business & Entrepreneurship				3		
	SNB4023	Strategic Management				3		
		Total Credit Hour				15		
Elective	EGB3013	Health, Safety & Environment				3		
		Total Credit Hour				3		
		Overall Credit Hour				162		

dent Name	: AMIR ISMAIL B ABDUL KADAR	Intake :	Jan 2002
dent ID	: 3846	Program :	C004
tus	: Not Completed		

rse Grp	Courses	Sem.	Taken	Grade	Credit Taken	Credit Passed		Course Status
1000	Foundation English	Jan	2002	P		0		Standard
1013	Language Awareness & Study Skills	Jul	2002	B	3	3		Standard
1023	English II (Tech & Professional Wri	Jan	2003	B	3	3		Standard
1103	Foundation Mathematics I	Jan	2002	B+	3	3		Standard
1123	Foundation Mathematics II	Jul	2002	B+	3	3		Standard
1013	Computer And The Information Age	Jan	2002	B	3	3		Standard
1063	Programming I	Jan	2002	B+	3	3		Standard
1073	Programming II	Jul	2002	B+	3	3		Standard
al Credit Hour Foundation :					033	033		

NICAL

4143	Database Administration	Jul	2005	B+	3	3		Standard
al Credit Hour Technical :					003	003		

VERSITY REQUIREMENTS

1011	Gamelan I	Jan	2004	P	1	0		Standard
1021	Modern Music I	Jul	2002	P	1	0	*	Credit hr excluded from GPA/CGPA calculation
2021	Modern Music II	Jan	2003	P	1	0		Standard
1013	Religous Studies(Islamic Studies)	Jan	2003	A	3	3		Standard
2013	Malaysian Studies	Jul	2002	C+	3	3		Standard
al Credit Hour University Requirements :					009	006		
rall Credit Hour					139	123		

t of Exempted Courses :

ourses	Semester	UTP Grade

o exempted course.

dent Name : AMIR ISMAIL B ABDUL KADAR　　　Intake : Jan 2002
dent ID : [illegible]　　　Program : C004
tus : Not Completed

rse Grp	Courses	Sem. Taken	Grade	Credit Taken	Credit Passed	Course Status
[illegible]	Multinational Corp Fin & Intern Bus	Jan 2006		3	0	Standard
[illegible]	Industrial Training I	Jan 2005	B+	8	8	Standard
[illegible]	Professional Communication Skills	Jul 2003	B	3	3	Standard
[illegible]	Object Oriented Programming	Jan 2004	B+	3	3	Standard
[illegible]	Human Computer Interaction	Jul 2004	B+	3	3	Standard
al Credit Hour						
E						
3013	Health, Safety & Environment	Jan 2004	A-	3	3	Standard
1013	Introduction To Finance	Jul 2005	A-	3	3	Standard
4028	Industrial Training II	Jan 2005	B+	8	8	Standard
033	Discrete Mathematics	Jan 2003	A-	3	3	Standard
133	Quantitative Method	Jan 2003	A-	3	3	Standard
063	System Analysis & Design	Jan 2003	C+	3	3	Standard
043	Management Information Systems	Jul 2003	B+	3	3	Standard
073	E Business	Jan 2004	B	3	3	Standard
133	Business Application Programming I	Jul 2004	B	3	3	Standard
153	Decision Support Systems	Jul 2004	B+	3	3	Standard
013	Database Management In GIS	Jul 2005	B	3	3	Standard
153	Business Application Programming II	Jul 2005	A-	3	3	Standard
073	Advanced Database Systems	Jul 2004	B	3	3	Standard
093	Knowledge Management	Jan 2006		3	0	Standard
023	Database Systems	Jul 2003	B	3	3	Standard
033	Computer Organisation	Jul 2003	B	3	3	Standard
123	Operating Systems	Jan 2004	B+	3	3	Standard
153	Data Structure	Jul 2003	B	3	3	Standard
043	Visual Programming	Jan 2003	A-	3	3	Standard
063	Data Communications	Jan 2004	B	3	3	Standard
073	Software Engineering	Jul 2003	C	3	3	Standard
083	Internet Programming	Jan 2004	B	3	3	Standard
033	Computer Project Management	Jul 2004	C+	3	3	Standard
132	Final Year Project (Part I)	Jul 2005	B+	2	2	Standard
142	Ethical and Professional Issues	Jul 2005	B+	2	2	Standard
034	Final Year Project (Part II)	Jan 2006		4	0	Standard
l Credit Hour Core :				082	075	
TIVE						
033	Internet Marketing	Jan 2006		3	0	Standard
013	Corporate Finance	Jan 2006		3	0	Standard
013	Organisational Behaviour	Jul 2004	B+	3	3	Standard
023	Strategics Management	Jul 2005	B	3	3	Standard
Credit Hour Elective :				012	006	
ATION						
013	Business Accounting	Jul 2002	C+	3	3	Standard
013	Business Organisation	Jan 2002	B+	3	3	Standard