Comparative Study between Relational Database System (RDBMS) and Object Relational Database System (ORDBMS) in Data Modelling and Database Languages

By

Farhana binti Abdul Jalil

Dissertation submitted in partial fulfilment 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

Comparative Study between Relational Database Management System (RDBMS) and Object Relational Database Management System (ORDBMS) in Data Modelling and Database Languages

By

Farhana binti Abdul Jalil

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

Approved By,

Eliza

(Ms. Eliza Mazmee Mazlan)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
JANUARY 2006

CERTIFICATION OF ORIGINALITY

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

FARHANA BINTI ABDUL JALIL

ABSTRACT

This paper presents comparative study between Relational Database Management System (RDBMS) and Object Relational Database Management System (ORDBMS). The objective are to develop two systems with different models; Relational Database Management System (RDBMS) and Object-Relational Database Management System (ORDBMS) and also to choose which model is better from user and designer's point of view in terms of data modeling. Several problems are being identified in order to know which model is better; Relational Database Model or Object-Relational Database Model approach. Similarities and differences between the two models based on criteria such as data modeling are compared. This is to provide guidelines on which model users or designers to choose from based on different type of data that they wish to accommodate. The scope of research is limited to be on the development of RDBMS and ORDBMS. This project involves project planning, requirements gathering, requirements analysis, logical database design, physical database design and finally testing phase. Thus, for the data collection, a research and a survey has been conducted through readings and interviews. By developing this comparative study, this project is expected to be implemented in Admission and Registration Unit in which it can serve the better performance of database.

ACKNOWLEDGEMENT

In the name of ALLAH, The Most Gracious and Most Merciful.

Heartfelt thanks to ALLAH God Almighty, that finally I have fulfilled my Final Year Project. Hereby, I would like to take this opportunity to convey my greatest appreciation to all people who have been very cooperative and supportive to me during the accomplishment of this Final Year Project.

First of all, I would like to thank Universiti Teknologi PETRONAS (UTP) for arranging such beneficial course for the student, which will expose and enhance students' skill in the process of applying knowledge, expanding thoughts, solving problem independently and applying knowledge. Besides, I also to thank all of FYP Committee members who have approved my Final Year Project title and for managing this course proficiently.

I wish to warmly thank my Final Year Project supervisor, Ms. Eliza Mazmee Mazlan who has been supervised and guided me in the accomplishment of this Final Year Project. She is a very nice person and will not hesitate to give her cooperation to me throght the development of this project.

A token of appreciation to Puan Suhaidah bt Ismail from Admission and Registration Unit, UTP. for valuable information, guidance, knowledge and expertise; without her it will be hard to make the project successful.

Last but not least, I would like to give my greatest gratitude to my parents and friends who had given me their moral support and undivided encouragement to complete this Final Year Project successfully. Thank you very much.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

This chapter features the basic information of the project, which includes the background of the project, problem statement, the objectives and scope of the study. This project will be concentrating on two different types of database system that include Relational Database Management System (RDBMS) and Object-Relational Database Management System (ORDBMS).

1.1 Background of Study

1.1.1 Database System

Every company needs a database. Many different types of databases exist, some simple, others very complex. When we order a book from online bookstore in the Internet, we are accessing database. One of the simplest forms of database with which each one of us are familiar is a filing cabinet. Information is stored in cabinet drawers, in folders, and even subfolders. Many companies still shuffle paperwork on a day-to-day basis instead of storing their information in computer. Although the complete elimination of paperwork is virtually impossible for any company, the benefits of storing data in a database are obvious. [1]

The database is now such an integral part of our day-to-day life that often we are not aware we are using one. A database management system (DBMS) is a computer program designed to manage a database; a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMS use include accounting, human resources and customer support systems. Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office. [2]

There are four types of database models which are:

- Hierarchical database model
- Nctwork databasc modcl
- Relational database model
- Object Relational database model

Hierarchical database model

The architecture of hierarchical database model is based on the concept of parent/ child relationships. The structure of hierarchical database model appears as an inverted tree. A parent table can be associated with one or more child tables, but a single child table can be associated with only one parent table. [3] Figure 1.1 illustrates the hierarchical database model.

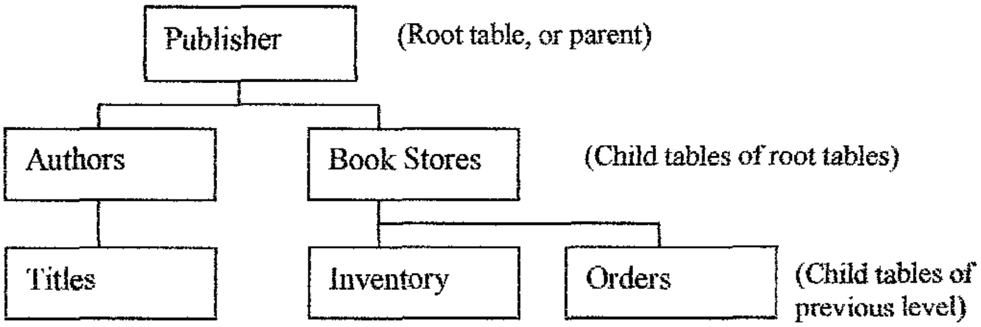


Figure 1.1: Hierarchical database model

Benefits of hierarchical model are:

- Data can be quickly retrieved
- Data integrity is easier to manage

Drawbacks of hierarchical model are:

- Users must be very familiar with the database structure
- Redundant data is stored

Network Database Model

Improvements were made to the hierarchical database model in order to derive the network model. Advantage of network model is the capability of parent tables to share relationship with child tables. This means that a child table can have multiple parent tables. [2] It is a transparent construction that relates a pair of nodes together by using one node as an owner and the other node as a member. [3] The relationship between tables in the network model is called a set structure. Set structures can represent a one-to-many relationship between tables. [2] Figure 1.2 illustrates set structures.

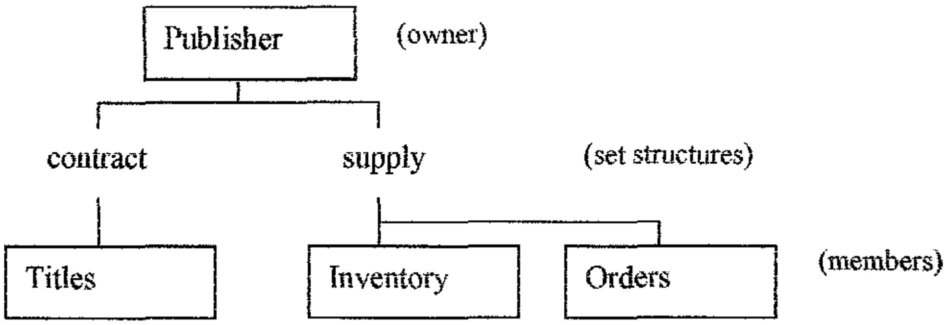


Figure 1.2: Network Database Model

The benefits of network database model are as follows:

- Data is accessed very quickly
- Users can access data starting with any table

The drawbacks of the network database model are as follows:

- The structure of the database is not easily modified
- Changes to the database structure definitely affect application programs that access the database
- The user has no understanding on the structure of the database

Relational Database Model

The relational database model is the most popular database model used today. Many improvements have been made to prior database models that simplify data management, data retrieval, and change propagation management. Three different types of table relationship are allowed: one-to-one, one-to-many, and many-to-many. Different relationships should be allowed to exist between tables in a database.[1]

In the relational model, there is no root table, although parent and chilled relationships of tables are allowed. A parent table can have multiple child tables, as a child table can have multiple child tables, as a child table can have multiple parent tables (bi-ditectional relationships). [1]

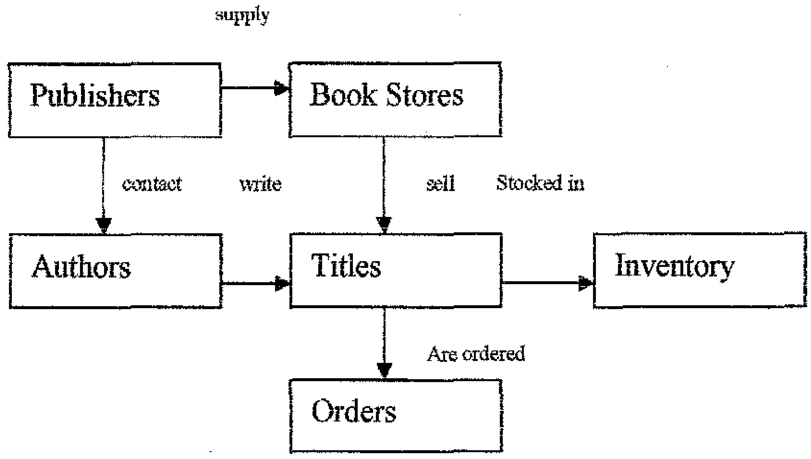


Figure 1.3: The Relational Model

Benefits of the relational model are as follows:

- Data is accessed very quickly
- The database structure is easy to change
- The data is represented logically, therefore users need not understand how the data is stored
- It is easy to develop complex queries to retrieve data
- It is easy to implement data integrity
- Data is generally more accurate
- It is easy to develop and modify application programs
- A standard language (SQL) has been developed

Drawbacks of the relational database model are as follows:

- Different groups of information, or tables, must be joined in many cases to retrieve data
- Users must be familiar with the relationship between tables
- Users must learn SQL

Object-Relational Database Model

Object-relational database management system (ORDBMS) is a relational database management system that allows developers to integrate the database with their own custom data types and methods. [4] This system simply puts an object oriented front end on a relational database management system (RDBMS). When applications interface to this type of database, it will normally interface as though the data is stored as objects. However the system will convert the object information into data tables with rows and columns and handle the data the same as a relational database. Likewise, when the data is retrieved, it must be reassembled from simple data into complex objects. [5]

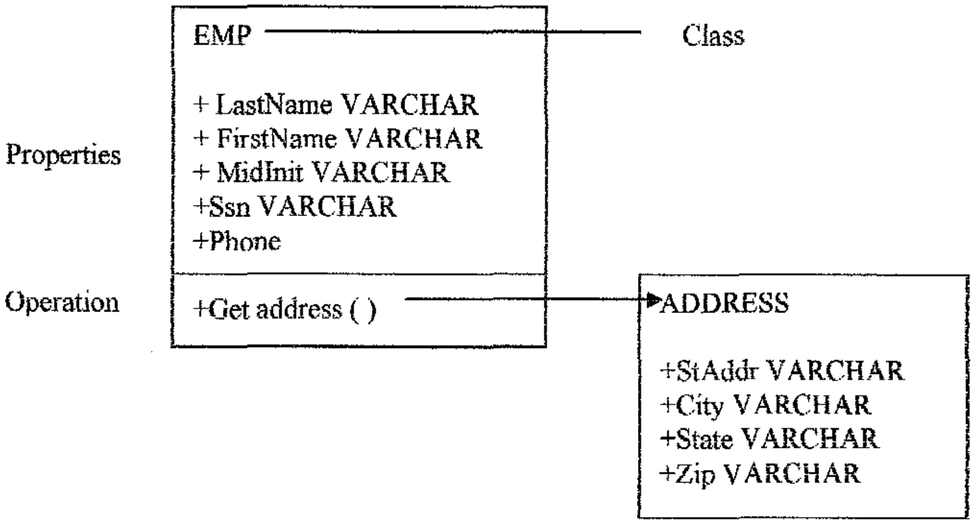


Figure 1.4: The Object-relational database

Benefits of the object-relational model are as follows:

- Objects can inherit property settings from other objects
- It is theoretically easier to manage objects

Drawbacks of the object-relational model are as follows:

- Users must learn Object Oriented (OO) concepts because the OO database does not work with traditional programming methods
- Stability is a concern since OO database have not been around for long.

1.1.2 Admission and Registration Unit, UTP

For this paper, Admission and Registration Unit of UTP is chosen as the case study. This unit is under Registrar Department. The scope of work for this unit are basically monitoring the admission of new students and handling registration courses of students. Recently, most of departments in UTP are using SAP R/3 System in implementing all activities. In this case the design of the database from this unit will be studied and that design will be used to come out with two systems with different type of model which are Relational Database Model and Object-Relational Database Model. From here, the models will be compared is in terms of data modeling and database language which are Data Definition Language (DDL) and Data Manipulation Language (DML)

1.2 Problem Statement

1.2.1 Problem Identification

Several problems had been identified for this paper which are:

- Which model is better; Relational Database Model or Object-Relational Database Model approach. Here relationship and differences will be discovered between these two models approaches based on data modeling and database languages.

- Since different model of databases is suitable with different type of data, people do not know which one is suitable to use in their situation. The approach used in practice does not mean that it produces the best results. It is often determined by the standards used in an organization. Modern approaches, leading to the better results, are therefore not recognized. [6]

In order to understand the advantages and disadvantages of Relational and OO technologies, features of each model is explored. Only then the decision can be made in which technology to apply in which problems.

Table 1.1: Uses of Relational and Object Technology

FUNCTIONS	RELATIONAL DATABASE	OBJECT RELATIONAL DATABASE	OBJECT DATABASE
Modeling complex entities for simulation, finance, scheduling, and configuration		/	/
Implementing document and object repositories		/	/+
Handling thousands of transaction per hour	/+	/	
Handling transactions of long duration			/
Maintaining data integrity and security, by separating data from applications	/		
Storing and retrieving multimedia and other user defined types		/	/
Easy integration with existing programming languages, and their types		/	/+
Handling data analysis and warehousing tasks	/-	/	

(Source: Robert Vermeulen, 1996, ***Upgrading Relational Databases with Objects,*** *New York,* page 105-110)

Indicator:

/ - – Acceptable

/ = Good

/+ = Excellent

1.2.2 Significant of the Project

There are several benefits Admission and Registration unit of UTP can gain in order to have a good database approach which are:

- Data is accurate and easy to manage

  The database will have the accurate data such as referential integrity is applied (primary key and foreign key constraints) and some other constraints also have been established to check the uniqueness or validity of data. [1]

- Redundant data is minimized

  One of the main goals when storing data in a database is to reduce or eliminate redundant information. Data should be stored one time in the database if possible. If an occurrence of data is stored in multiple times in the database, the data will be updated when changes are required. Redundant data is minimized though process called normalization. [1]

1.3 Objective, Scope of Study and Feasibility of the Project

1.3.1 Objective

The objectives of this project are:

- To develop two systems with different models; Relational Database Management System (RDBMS) and Object-Relational Database Management System (ORDBMS).
- To study differences between both models from user and designer view in terms of data modeling and database languages which are Data Definition Language (DDL) and Data Manipulation Language (DML).

1.3.2 Feasibility of the Project within Timeline and Scope

This project can be considered as technically feasible as the scope of the project is limited to the comparative study between RDBMS and ORDBMS. Here is no relative cost related to the project as this project can be developed using open sources application resources that is already available in the Internet. There are also adequate resources available to support the project such as Internet developers' forums, open sources, books, online resources as well as expertise from database field itself.

The time given to complete the project is also sufficient. The project timeline indicates the time allocated for each tasks and it serves as guidance for project execution. **(Refer Appendix 1)**

CHAPTER 2

LITERATURE REVIEW / THEORY

2. LITERATURE REVIEW / THEORY

In doing the comparative study between Relational Database Management System (RDBMS) and Object Relational Database Management System (ORDBMS), there are several things that have been taken in account. First, the author defines the definition of RDBMS and ORDBMS. Then, there are studies on features of these two models in order to get the clear comparison between them.

2.1 Definition of RDBMS and ORDBMS

According to Jagadish Chaterjee (2005) in his article " Introduction to RDBMS, ORBMS and ORDBMS":

> The relational model is based on the structure of a database. A database is simply a collection of one or more relations or tables with columns and rows. The use of set theory allows for data to be structured in a series of tables that has both columns and rows. Each column corresponds to an attribute of that relation, while each row corresponds to a record that contains data values for an entity.[5]

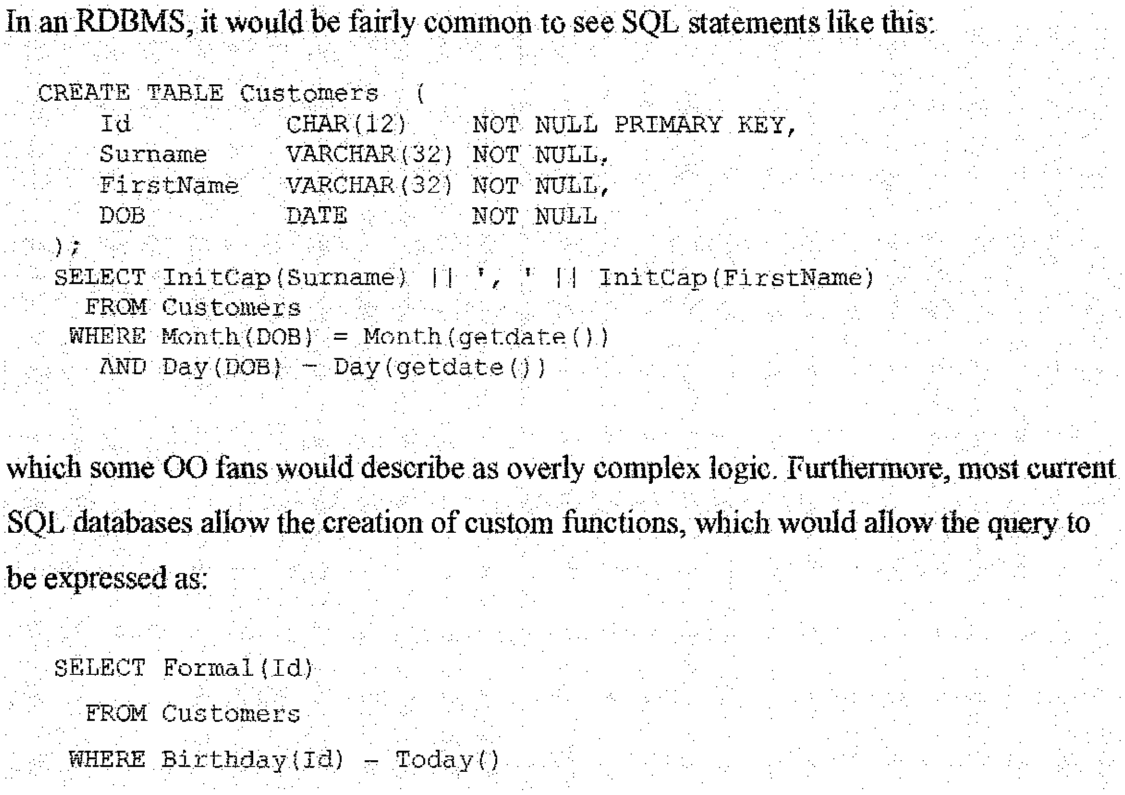
However, there is a little bit different with Object Relational Database Management System concepts.

According to Wikipedia, the free encyclopedia :

> Object-relational database management system (ORDBMS) is a relational database management system that allows developers to integrate the database with their own custom data types and methods. The term *object-relational database* is sometimes used to describe external software products running over traditional DBMSs to provide similar features; these systems are more correctly referred to as object-relational mapping system.[4]

From these definitions, it shows that the RDBMS a simple collection of one or more relations or tables with columns and rows. Meanwhile, the ORDBMS is a relational database management system that allows developers to integrate the database with their own custom data types and methods which cannot be found in RDBMS.

To prove the theory of the articles saying that ORDBMS allows the developers to use their own data types and methods, the author had came out with the coding that can show the different compared with RDBMS.

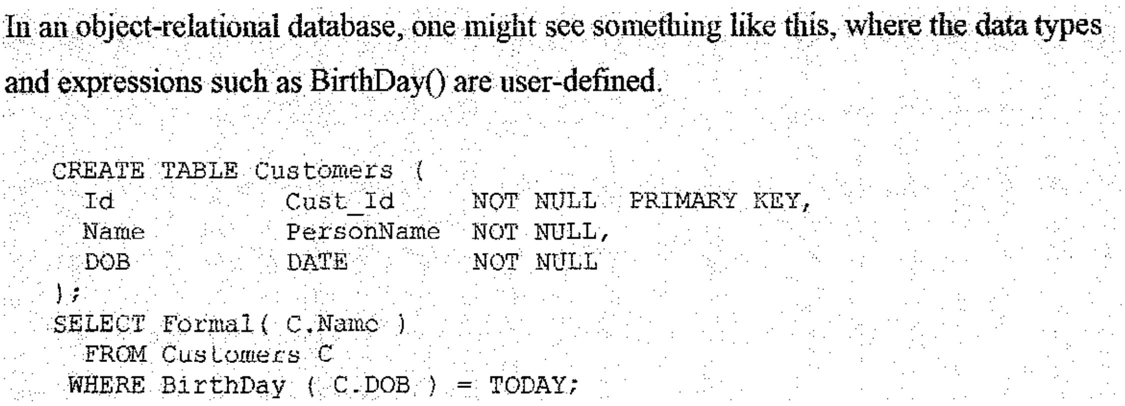


In an RDBMS, it would be fairly common to see SQL statements like this:

```
CREATE TABLE Customers  (
    Id          CHAR(12)    NOT NULL PRIMARY KEY,
    Surname     VARCHAR(32) NOT NULL,
    FirstName   VARCHAR(32) NOT NULL,
    DOB         DATE        NOT NULL
 );
 SELECT InitCap(Surname) || ', ' || InitCap(FirstName)
   FROM Customers
  WHERE Month(DOB) = Month(getdate())
    AND Day(DOB) = Day(getdate())
```

which some OO fans would describe as overly complex logic. Furthermore, most current SQL databases allow the creation of custom functions, which would allow the query to be expressed as:

```
 SELECT Formal(Id)
   FROM Customers
  WHERE Birthday(Id) = Today()
```



In an object-relational database, one might see something like this, where the data types and expressions such as BirthDay() are user-defined.

```
CREATE TABLE Customers (
  Id            Cust_Id      NOT NULL  PRIMARY KEY,
  Name          PersonName   NOT NULL,
  DOB           DATE         NOT NULL
);
SELECT Formal( C.Namc )
  FROM CusLomers C
 WHERE BirthDay ( C.DOB ) = TODAY;
```

2.2 Features of RDBMS and ORDBMS

Each model has their unique advantages and disadvantages in database development. The author has to find out several features of these two models as the author can figure out which is the best in which circumstances.

After some readings and study, the author came out with several benefit of using RDBMS which are:

- A simple data-storage concept, tables, and standard query language
- A logical separation of the database from application programs
- A powerful data consistency and security mechanism
- The ability to have multiple concurrent users, with transactions

This research had been proved by article below. According to Jagadish Chaterjee (2005) in his article " Introduction to RDBMS, ORBMS and ORDBMS":

> Benefits of RDBMS are that the system is simple, flexible, and productive. Because the tables are simple, data is easier to understand and communicate with others. RDBMS are flexible because users do not have to use predefined keys to input information. Also, RDBMS are more productive because SQL is easier to learn. This allows users to spend more time inputting instead of learning. More

> importantly, RDBMS's biggest advantage is the ease with which users can create and access data and extend it if needed. After the original database is created, new data categories can be added without the existing application being changed.[5]

As for the ORDBMS, according to Jamiee Soni, Barbara S, Sally T, (2000) in their articles:

> ORDBMS is an extension of the relational model which allows richer data types to be supported. These new data types include user-defined abstract data types (ADTs), including image, audio, and video. Constructed types such as sets, tuples, arrays, and sequences are another example. A key feature of ORDBMS is inheritance which takes advantage of the commonality between different data types. For example, voice data and music clips are different data types which have things in common. It is desirable to *inherit* properties of stereo music clips while defining voice data which may be recorded in mono.[7]

This proved the research of the author regarding the advantages of ORDBMS which it can make use of the relationships between data to easily collect related records. In an address book application, an additional table would be added to the ones above to hold zero or more addresses for each user. Using a traditional RDBMS, collecting information for both the user and their address requires a "join":

```
  SELECT InitCap(C.Surname) || ', ' || InitCap(C.FirstName), A.city
    FROM Customers C, Addresses A
   WHERE A.Cust_Id=C.Id -- the join
     AND A.city="New York"
```

The same query in an object-relational database is much simpler:

```
 SELECT Formal( C.Name )
   FROM Customers C
  WHERE C.address.city="New York"
```

2.3 Data Modeling of RDBMS and ORDBMS

In this project, two data models are developed; which are Entity Relationship Diagram (ERD) for RDBMS and Unified Manipulation Language (UML) for ORDBMS.ERD is best suited for relational database while UML is most suited for object relational database.

Data Modeling of RDBMS

According to Information Technology Services, University of Texas in its article "Introduction to Data Modeling":

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is [8]:

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.[8]
- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user. [8]
- In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software. [8]

Data Modeling of ORDBMS

From Wikipedia, the free encyclopedia;

> The Unified Modeling Language (UML) is a non-proprietary, object modeling and specification language used in software engineering. UML includes a standardized graphical notation that may be used to create an abstract model of a system: the *UML model*. UML is an extensible modeling language. If a concept you need is not present in the base language, you may introduce it by defining a stereotype.[9]
>
> UML is officially defined at the Object Management Group by the UML metamodel (a Meta-Object Facility metamodel serialized in XMI). UML is a General Purpose Modeling language. While UML was designed to specify, visualize, construct, and document software-intensive systems, UML is not restricted to modeling software. UML has its strengths at higher, more architectural levels and has been used for modeling hardware (engineering systems) and is commonly used for business process modeling, systems engineering modeling, and representing organizational structure among many other domains.[9]

Inheritance

A very important concept in object-oriented design, *inheritance*, refers to the ability of one class (child class) to *inherit* the identical functionality of another class (super class), and then add new functionality of its own. (In a very non-technical sense, imagine that I inherited my mother's general musical abilities, but in my family I'm the only one who plays electric guitar.) To model inheritance on a class diagram, a solid line is drawn from the child class (the class inheriting the behavior) with a closed, unfilled arrowhead (or triangle) pointing to the super class. Consider types of bank accounts: CheckingAccount and SavingsAccount classes inherit from the BankAccount class are shown below.[9]

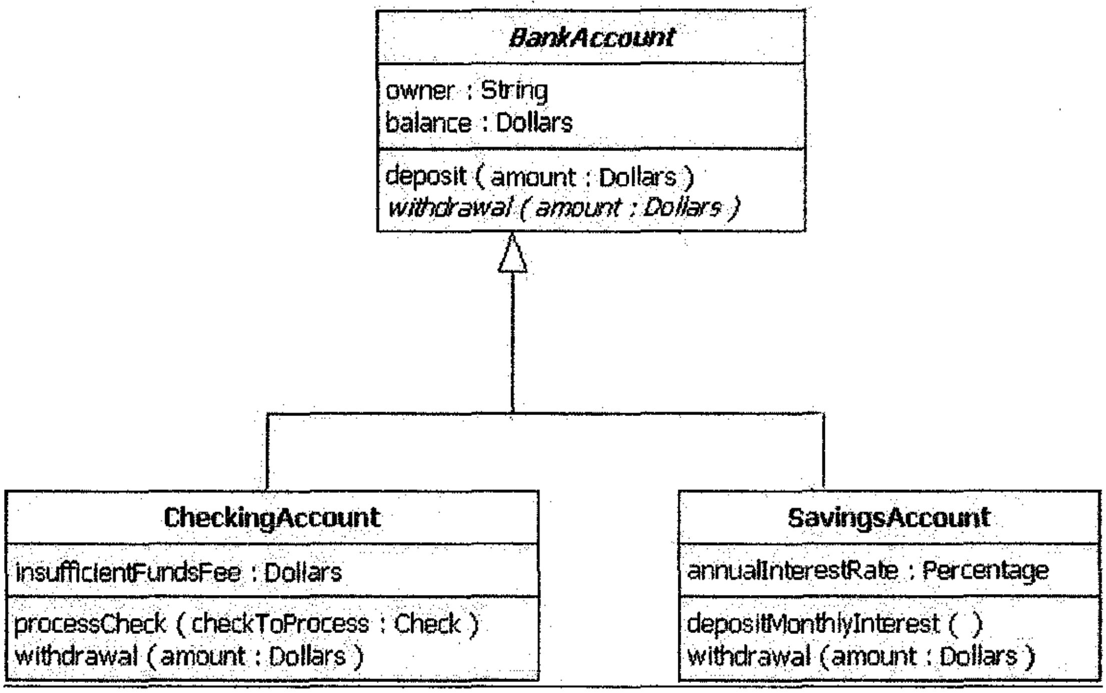

BankAccount
owner : String
balance : Dollars
deposit (amount : Dollars)
withdrawal (amount : Dollars)
CheckingAccount
insufficientFundsFee : Dollars
processCheck (checkToProcess : Check)
withdrawal (amount : Dollars)
SavingsAccount
annualInterestRate : Percentage
depositMonthlyInterest ( )
withdrawal (amount : Dollars)

Figure 1.5: The hierarchy of BankAccount class

Inheritance offers the following benefits:

- Subclasses provide specialized behaviors from the basis of common elements provided by the superclass. Through the use of inheritance, programmers can reuse the code in the superclass many times.
- Programmers can implement superclasses called *abstract classes* that define common behaviors. The abstract superclass defines and may partially implement the behavior, but much of the class is undefined and unimplemented. Other programmers fill in the details with specialized subclasses.

In summary, Relational and Object-relational database systems each have certain strengths as well as certain weaknesses. In general, the weakness of one type of system tends to be strength of the other.

CHAPTER 3
METHODOLOGY

3. METHODOLOGY AND PROJECT WORK

This chapter contains a detailed description of the methodologies and procedures used to complete and achieve the objectives of this project. This includes the development of the Relational Database and Object Relational Database for Registration Unit UTP.

3.1 Procedure Identification

Below is the methodologies used to complete this project;

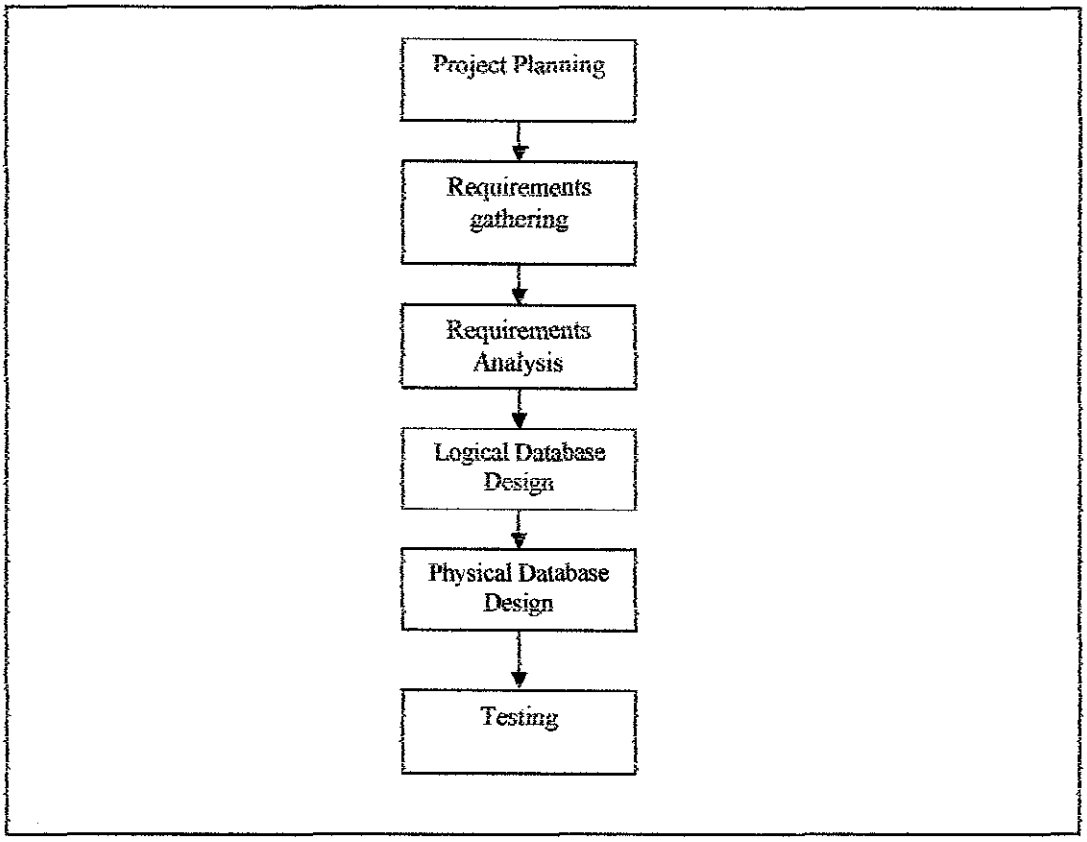


Figure 1.6 : Project Phases of the project

3.2 Project Phases Details

Table 1.2: Project Phases Details

Phase	Main Activities
Project Planning	Planning on how the phases can be realized most efficiently and effectively. ✤ Develop milestones of project A Gantt chart had been developed in order to provide a time line for the work and tasks that have been allocated during the development of the project. **(Refer to Appendix 1)**
Requirements Gathering and Analysis	Fact finding techniques ✤ Interview ➤ An interview is conducted with the Head of Admission and Registration Unit in order to know the business process and how this unit runs.**(Refer to Appendix 8)** ✤ Study on Literature Review ➤ Analysis had been done through reading the article on related fields and understanding the concept of Relational Database Management System) RDBMS and Object Oriented Database Management System (ORDBMS). ✤ Analyzing current database ✤ Determining System Requirements

	- Examining Documentation
Logical Database Design	Build and validate logical data model - Identify entity types - Identify relationship types - Determine attribute domains - Determine candidate, primary, and alternate key attributes. - Derive relations for logical data model (eg: one-to-one relationship) - Validate relations using normalization - Check integrity constraints
Physical Database Design	Translate logical data model for target DBMS - Design base relations - Design representation of derived data - Design the general constraints Construct/ Coding - Oracle9i is used as a tool in the development of the proposed system.
Testing	Testing for errors and validated against requirements specified by users

3.3 Tools

In order to ensure the development of the project is succeed, a specific hardware requirements and software used been specified to support the project development.

3.2.1 Hardware

A list of hardware requirements of the computer that required completing the project has been shows in Table 1.3 below.

Table 1.3: Minimum Hardware Requirement

Device	Requirement
Processor	Intel Pentium IV 2.66 Ghz
Memory	512MB of memory
Disk Space	20GB of free space

3.2.2 Software

Table 1.4 shows the software used throughout the development of monitoring service.

Table 1.4: Software Requirements

Software	Function
Oracle 9i	This development system will be developed using Oracle 9i.
Window XP	As a platform for this project.
Macromedia Dreamweaver MX	Software tool to build interface of the system.
PHP and Apache Server	Web server that can connect PHP and Oracle.
Microsoft Visio	Software tool to develop diagram; ERD and UML

CHAPTER 4.0

RESULTS AND DISCUSSION

4. RESULTS AND DISCUSSION

This chapter compiles the current findings or outcomes of the project work. There has been some information, coming from journal and online sources. According to findings from different sources, the differences in terms of data design and database languages have been discussed in this chapter.

4.1 Project Planning Implementation

The first step in developing database system is to clearly define the mission statement for the database project, which defines the major aims of the database system.

4.1.1 Creating the mission statement for the system developed.

The process of creating a mission statement for the system is began by conducting the interview with the person in-charged and other appropriate staff. Open-ended questions are normally the most useful at this stage of process. An interview had been conducted with one of the staff in Admission and Registration Unit, UTP. Her name is Puan Suhaidah Bt Ismail, the head of Registration. Below are several questions had being asked during the interview.

Farhana : "What is the purpose of your unit?"

Pn. Suhaidah : "The purpose of this unit is to do the process of students' course registration."

Farhana : "Why do you feel that you need a database?"

Pn. Suhaidah : "Yes. Since there is many data that have to be managed, it is sufficient for this department to have a well-organized database."

Farhana : "How do you know that a database will solve the problem?"

Pn. Suhaidah : "All I know is that we are drowning in paperwork. We need something that will speed up the way we work by automating a lot of the day-to-day tasks that seen to take for ever days."

Farhana : "What is your opinion with the system that you have now?"

Pn. Suhaidah : "Frankly speaking, the system that we use now can help us to do the job faster and more efficient. However, we do not know whether there will be any better system that can improve our performance."

A mission statement can be created after responses are analyzed. Here the mission statement will be

'The purpose of this project is to know which type of model of database is better to be used for Admission and Registration Unit in terms of the data modeling.'

4.2 Requirement Gathering Implementation

There are several data techniques that are used. Below are the techniques that had been done.

- Interview
    - An interview is conducted with the Head of Admission and Registration Unit in order to know the business process and how this unit runs.

- Research
  - Analysis had been done through reading the article on related fields and understanding the concept of Relational Database Management System (RDBMS) and Object Oriented Database Management System (ORDBMS).
- Analyzing current database
  - During the interview, some questions related to the current database are asked. Currently the system that Admission and Registration Unit is using is Campus Management (CM) System. This system can track the data for student registration.
- Examining Documentation
  - Documents related to the business process of the course registration are examined in getting the clear picture of the flow.

4.3 Requirement Analysis

4.3.1 Business Process

Academic and Central Services (ACS) is one of the department in UTP. It consists of several units such as Admission and Registration Unit, Exam and Record Unit, and Academic Administration Unit.

As for this project, Admission and Registration Unit is selected. In general, Admission specializes in process of registration of new students while Registration specializes in process of course registration. This project will focus on Registration part. **(Refer to Appendix 2 and 3)**

Basically, the process of the course registration is starting from creating and maintaining students master data. In other words, this unit have to come out with student master data which include student names, students ID no, year of enrollment, semester of enrollment, sponsors (if any), programme code, major / minor courses, credit transfer and waive course. After the student master data is created, the unit will proceed with creating the course plan. It comprises of courses taken and which category the courses are in. This activity will be done before course registration begins.

Before the students register their courses, they will get their advisor's approval first. The student will register the course during the course registration time. Usually it will be during the final week before semester ends. There are two types of registration; by batch or individual. Registration by batch will be done by Academic Administration clerk and individual registration will be done by students. Students will have access to the system through their ID no, IC no, or name.

After the student had registered the courses online, it will be checked by the Academic Administration in the system. This unit is using Campus Management (CM) System. **(Refer Appendix 4)** If there is any problem or invalid data, the student will be required to fill in again the form, usually done using manual form.

The examination slip will be issued to the students after their registration form is approved. Yet, the students have to check the slip whether it has correct information or vice versa.

Other department and unit such as Finance Department and Exam Unit can view the students' data through CM system. It is to know the courses taken by the students for payment and examination purposes.

4.4 Logical Database Design (RDBMS)

Logical Database design is basically a process of constructing a model of the data used in an enterprise based on a specific data model. The logical data model is based on the target data model for the database. For this project, relational database model and object relational database model are being used.

First of all an ER Diagram is developed. It is a top-down approach to database design that begins by identifying the important data called entities and relationships between the data that must be represented by the model.

4.4.1 Identify entity types

The basic concept of ER Model is the entity type. Entity type is a group of objects with the same properties. It has independent existence and can be objects with physical (or 'real') existence or objects with a conceptual (or 'abstract') existence. As for this project, several entities are identified which are:

Table 1.5: Entities with physical or conceptual existence

Physical existence
Student
Course
Clerk
Advisor
Course Enrolled
Registration Details
Finance Unit
Exam Unit
Sponsor
Slip Registration

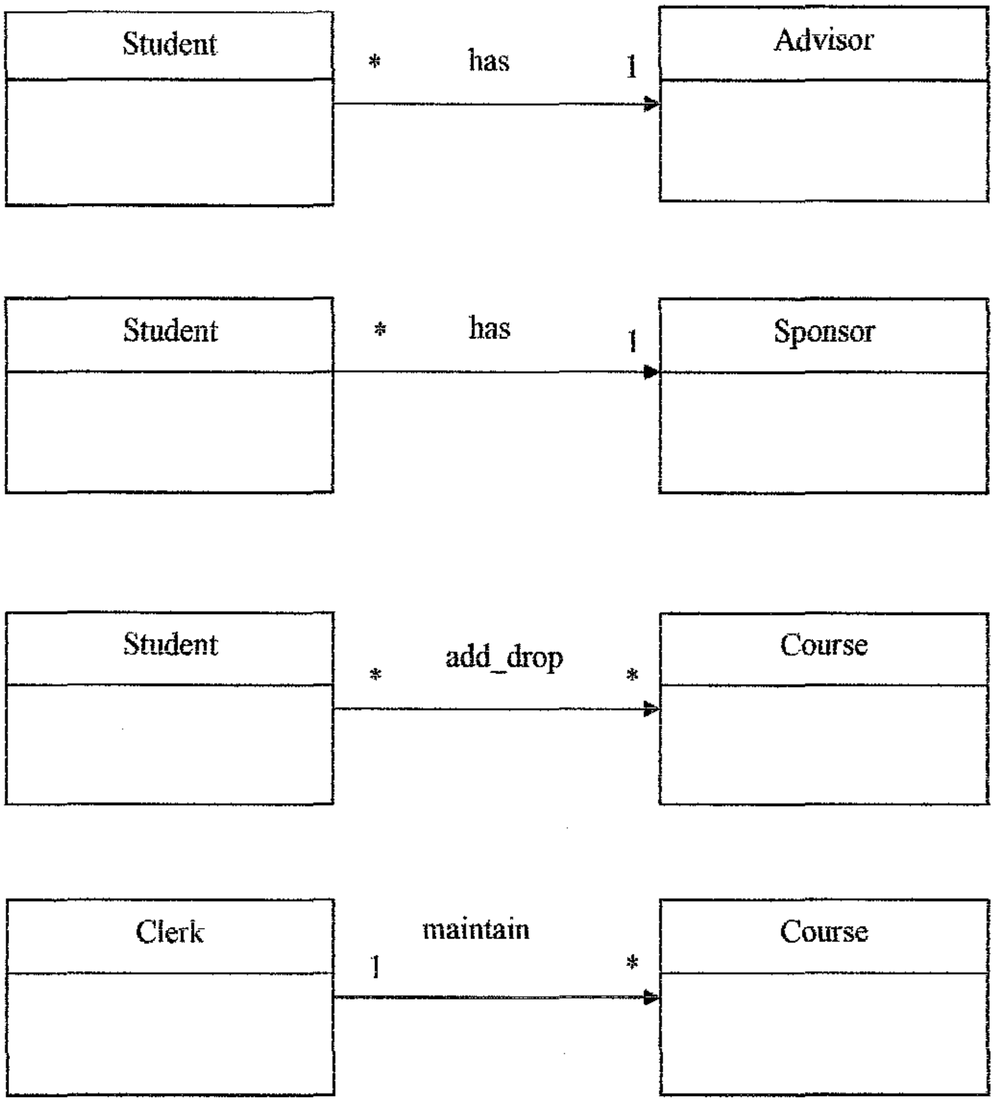
A relationship type is a set of associations between one or more participating entity types. Type used for this project is binary relationship

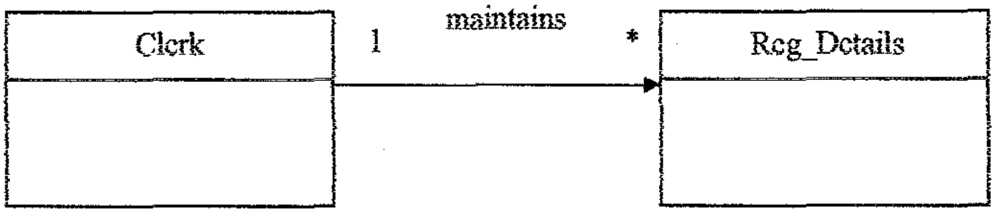
4.4.2 Derived relations for logical data model

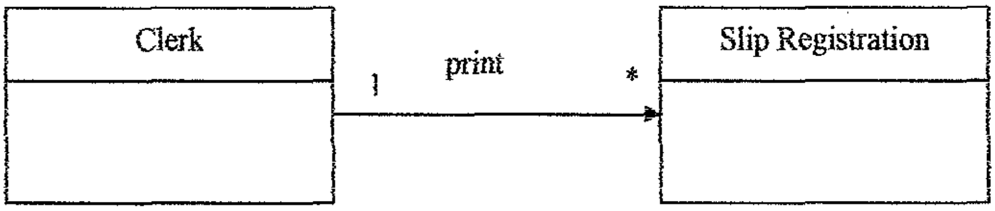
The main type of constraints on relationship is called multiplicity. There are three types of relationship which are:

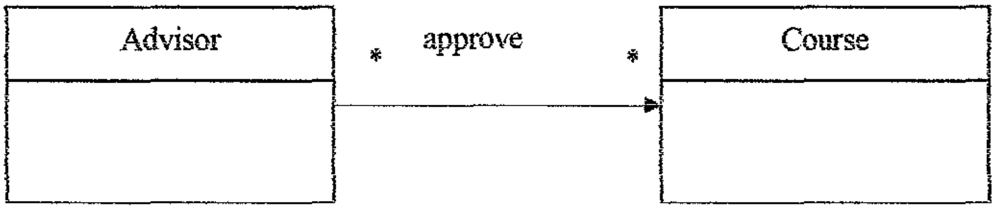
- One-to-one relationships
- One-to-many relationships
- Many-to-many relationships

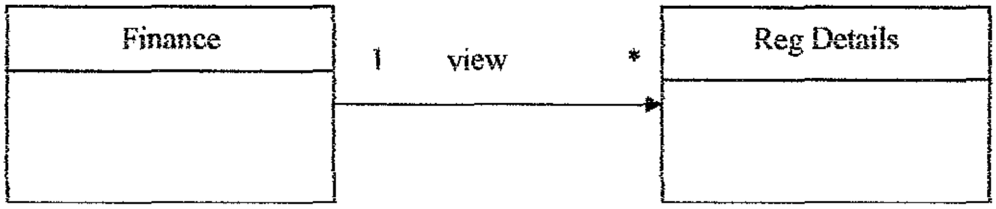
Below are the entity relationships that are identified.

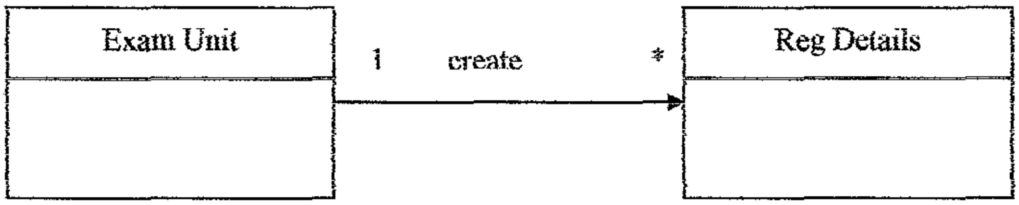



Clerk
1
maintains
*
Reg_Details


Clerk
1
print
*
Slip Registration


Advisor
*
approve
*
Course


Finance
1
view
*
Reg Details


Exam Unit
1
create
*
Reg Details

4.4.3 Identify and associate attributes with entity or relationship types

The objective is to associate attributes with entity or relationship types.
For this project, attributes are identified and associated with entities as follows:

Table 1.6: Entity and the attributes

No	Entity	Attribute
1	Student	ID{PK} Name Gender IC Status Address
2	Sponsor	S_Code{PK} S_Name S_IC S_ContactNo S_Address
3	Advisor	A_ID{PK} A_Name A_IC A_ContactNo A_Addreess
4	Course	CourseCode{PK} CourseName
5	SlipRegistration	Slip_No
6	Clerk	Clerk_ID{PK} Clerk_Name
8	Finance	Finance_Ref
9	Exam Unit	Exam_Ref
10	Course Enrolled	Student_ID Student_Name Student_Year Student_Course
11	Registration Details	Course_Name Student ID

4.4.4 Determine attribute domain

A domain is pool of values from which one or more attributes draw their values. A fully developed data model specifies the domains for each attribute and includes:

- Allowable set of values for the attribute
- Sizes and formats of the attribute

As attribute domain are identified, their names and characteristics are recorded in the data dictionary. **(Refer to Appendix 5)**

4.4.5 Determine candidate, primary, and alternate key attributes.

This step is concerned with identifying the candidate key. A candidate key is a minimal set of attributes of an entity that uniquely identifies each occurrence of that entity. More then one candidate key are identified, in which case one primary key will be choose, the remaining candidate keys are called alternate keys. **(Refer Appendix 5)**

4.5 User Interface for the Project

User Interface is necessary to be done since not all level of people are familiar with the SQL statement. Some people are expert on it but there are also certain people that naïve about SQL statement. Hence this is the alternative way for people to use the database so that they can compare between two types of models which are RDBMS and ORDBMS.

In fact, there are several software tools to create interface that may connect to Oracle. For instance, VB.Net, Macromedia Dreamweaver MX, C++ and some other languages can connect the interface to the database especially Oracle.

So, for this project, Macromedia Dreamweaver MX is used to developing the interfaces. There are some strong reasons why this tool is selected. Some of the reasons are it is easy to handle and there are lot of resources from the internet as well as references books that can show the connection between PHP and Oracle. To start the interface development, first the business process of the system is studied. It is to ensure how many interface will be developed. Also, the users of the system are being determined.

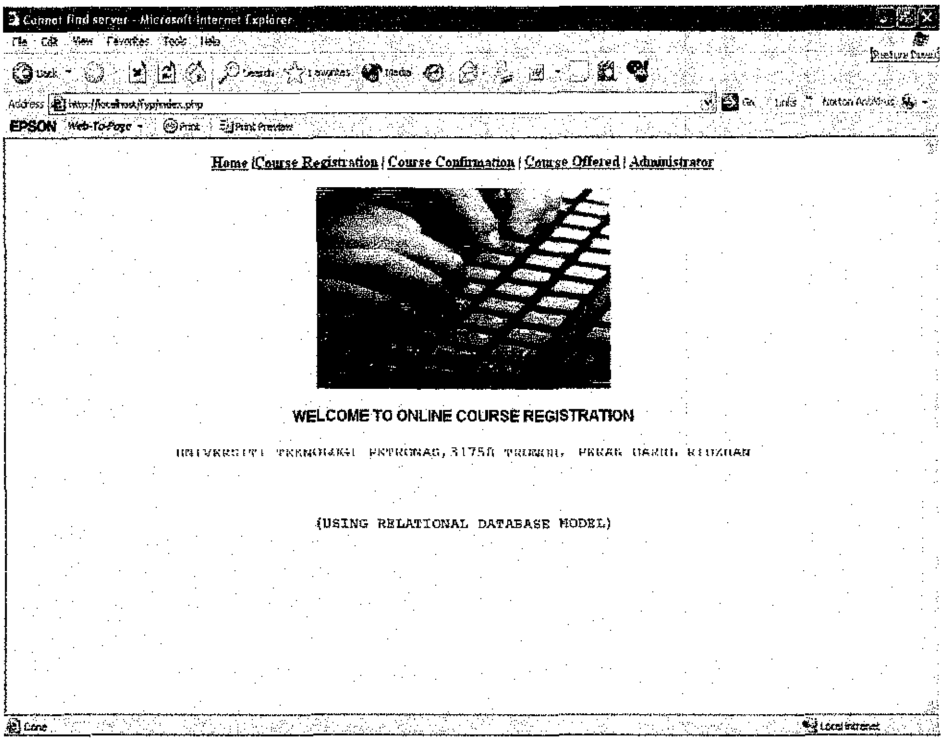
After some studies, below are the types of target user that will use the system followed by the Level of Authority (LOA) for each user.

Table 1.7: The user and Level of Authority for the system

User	Level of Authority (LOA)
Administrator	Add record, edit record, delete record, view record.
Student	Add record, drop record, view record.
Finance Unit	View record
Exam Unit	View record

The interface will be done based on the table above. For instance, there will be interfaces for add record, edit record, delete record, and query record. These is interfaces are distinguished from type of users. For example Finance Unit can view only selected record for student details compared to Administrator that able to view the master data of student details.

Below are the interfaces of the project.


Cannot find server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/fyp/index.php
Go Links Norton AntiVirus
EPSON Web-To-Page Print Print Preview
Done
Local intranet

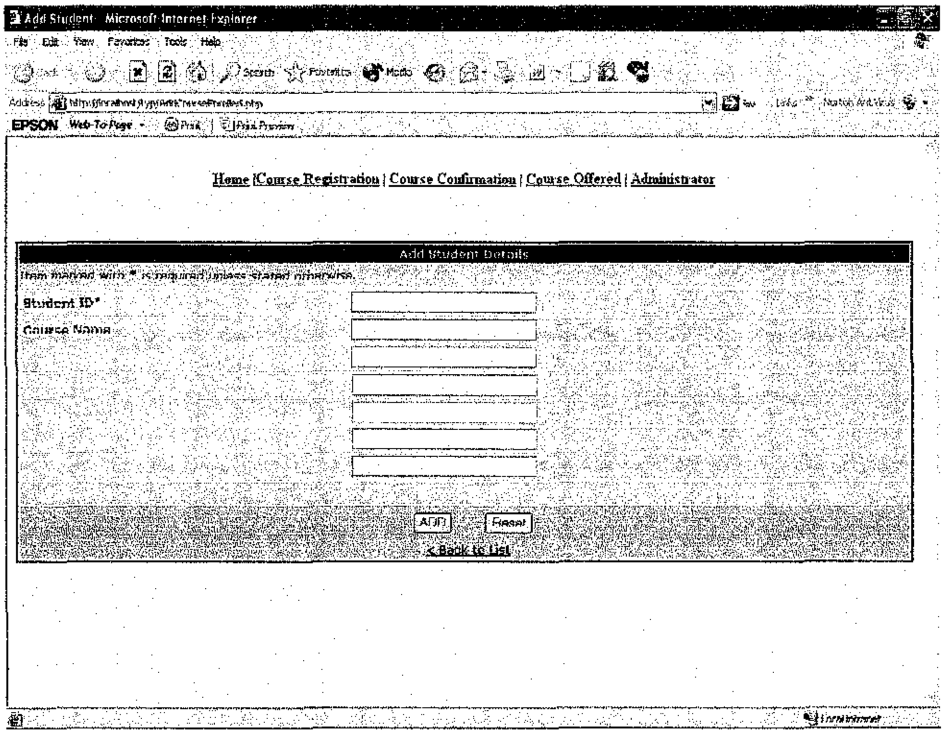
Home |Course Registration | Course Confirmation | Course Offered | Administrator

WELCOME TO ONLINE COURSE REGISTRATION

UNIVERSITI TEKNOLOGI PETRONAS, 31750 TRONOH, PERAK DARUL RIDZUAN

(USING RELATIONAL DATABASE MODEL)

Figure 1.7 : Interface of main page


Add Student - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go Links Norton AntiVirus
EPSON Web-To-Page Print Print Preview
ADD
Reset
Local intranet

Home |Course Registration | Course Confirmation | Course Offered | Administrator

Add Student Details

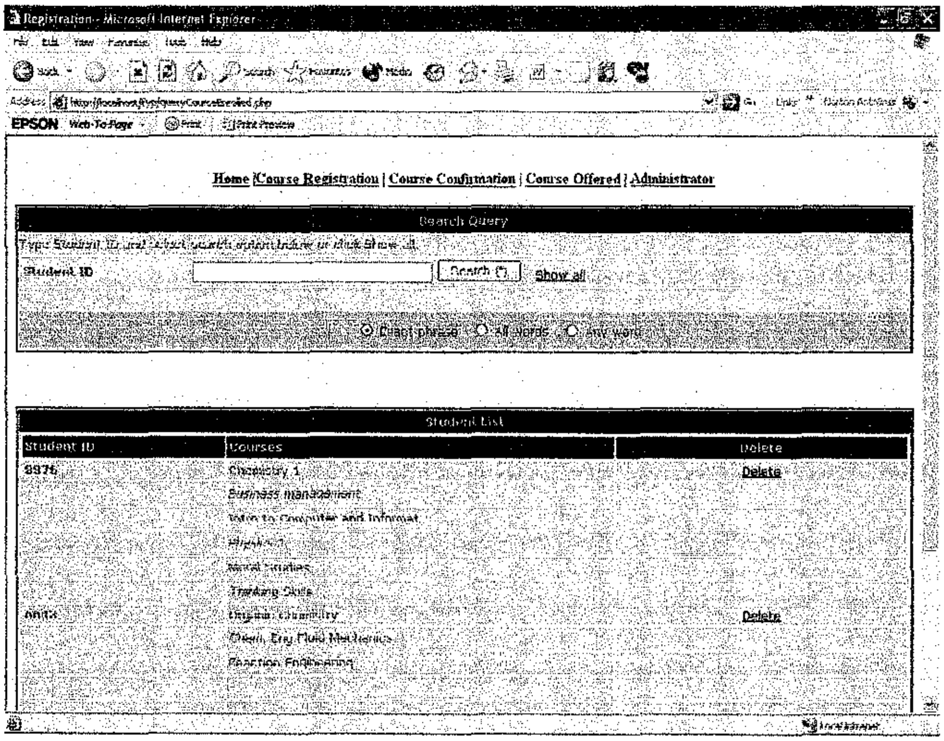
Item marked with * is required unless stated otherwise.

Student ID*

Course Name

< Back to List

Figure 1.8 : Interface of course registration (student's view)



Registration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/fyp/queryCourseEnrolled.php
EPSON Web-To-Page Print Print Preview
Local intranet

Home |Course Registration | Course Confirmation | Course Offered | Administrator

Search Query

Type Student ID and select search option below or click Show all

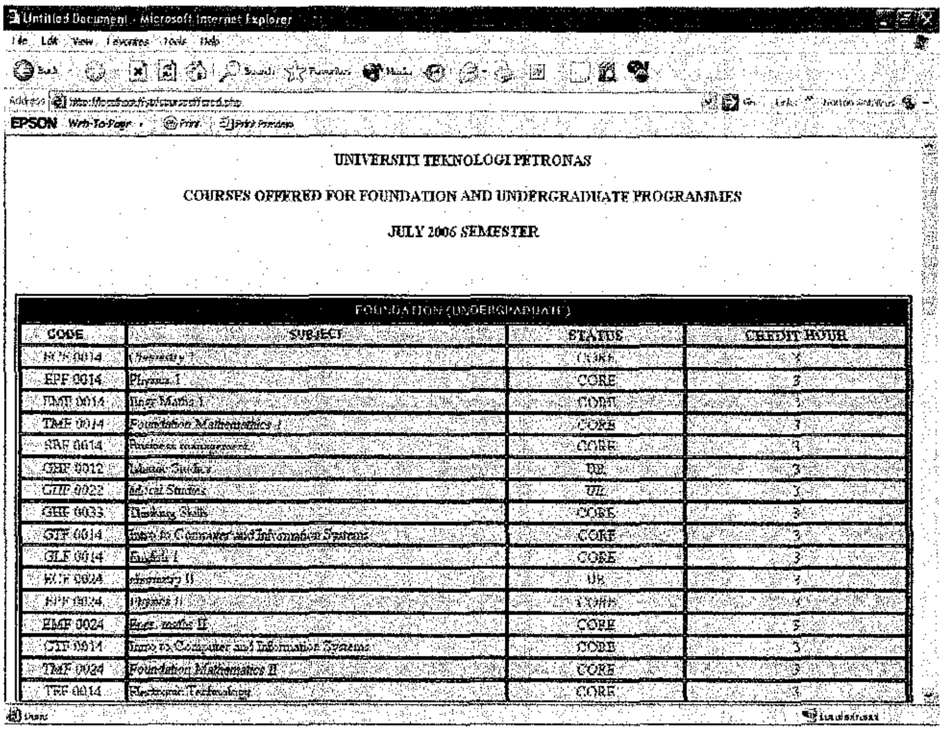
Student ID □ Search Show all

● Exact phrase ○ All words ○ Any word

Student List

Student ID	Courses	Delete
8976	Chemistry 1 Business management Intro to Computer and Informat Physics 1 Moral Studies Thinking Skills	Delete
[illegible]	Organic Chemistry Chem. Eng Fluid Mechanics Reaction Engineering	Delete

Figure 1.9 : Interface of view course confirmation (student's view)



Untitled Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/fyp/courseoffered.php
EPSON Web-To-Page Print Print Preview
Done
Local intranet

UNIVERSITI TEKNOLOGI PETRONAS

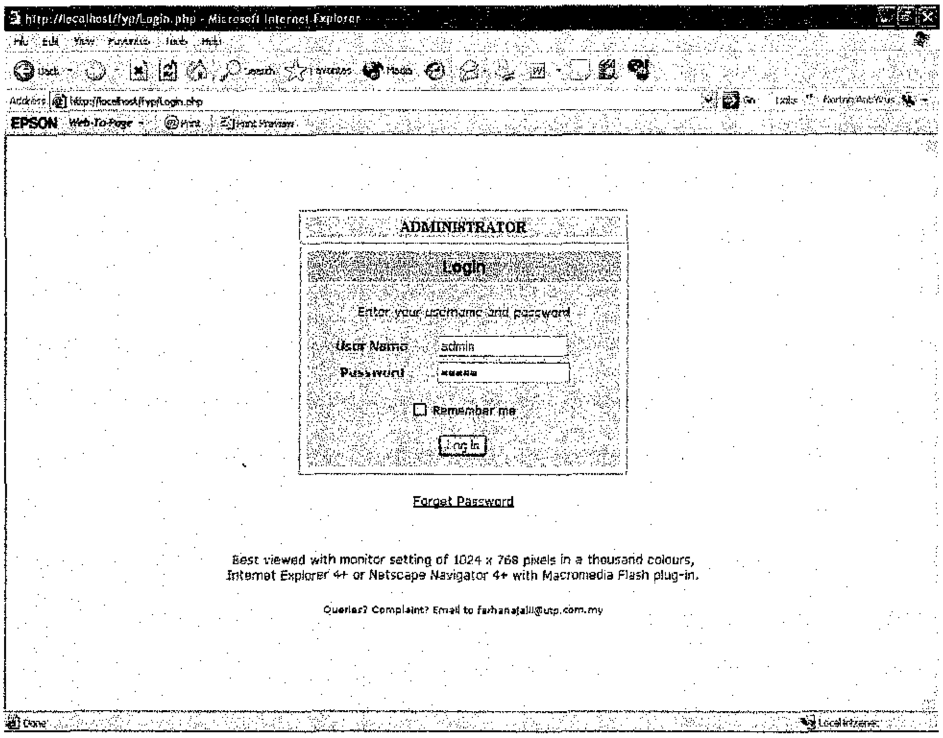
COURSES OFFERED FOR FOUNDATION AND UNDERGRADUATE PROGRAMMES

JULY 2006 SEMESTER

FOUNDATION (UNDERGRADUATE)

CODE	SUBJECT	STATUS	CREDIT HOUR
KCF 0014	Chemistry 1	CORE	3
EPF 0014	Physics 1	CORE	3
EMF 0014	Engr Maths 1	CORE	3
TMF 0014	Foundation Mathematics I	CORE	3
SBF 0014	Business management	CORE	3
GHF 0012	Islamic Studies	UR	3
GHF 0022	Moral Studies	UR	3
GHF 0033	Thinking Skills	CORE	3
GTF 0014	Intro to Computer and Information Systems	CORE	3
GLF 0014	English I	CORE	3
KCF 0024	Chemistry II	UR	3
EPF 0024	Physics II	CORE	3
EMF 0024	Engr. maths II	CORE	3
GTF 0014	Intro to Computer and Information Systems	CORE	3
TMF 0024	Foundation Mathematics II	CORE	3
TEF 0014	Electronic Technology	CORE	3

Figure 2.0 : Interface of course offered page



http://localhost/fyp/Login.php - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/fyp/Login.php
Go Links Norton AntiVirus
EPSON Web-To-Page Print Print Preview
Done
Local intranet

ADMINISTRATOR

Login

Enter your username and password

User Name: admin

Password: *****

□ Remember me

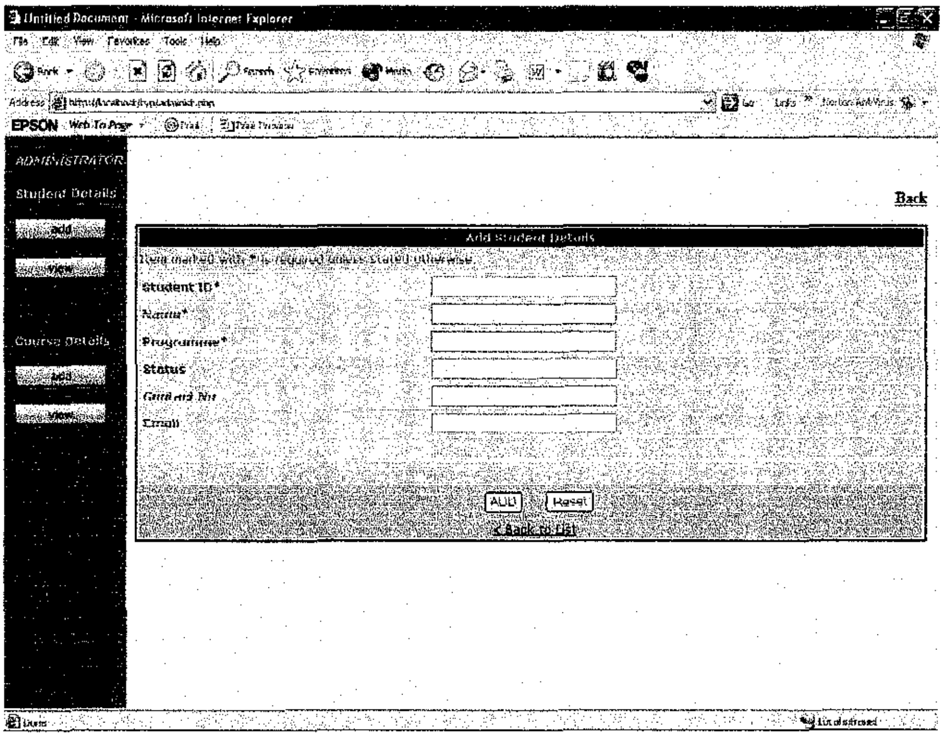
Log In

Forget Password

Best viewed with monitor setting of 1024 x 768 pixels in a thousand colours, Internet Explorer 4+ or Netscape Navigator 4+ with Macromedia Flash plug-in.

Queries? Complaint? Email to farhanajalil@utp.com.my

Figure 2.1: Interface of administrator's login



Untitled Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/fyp/admin.php
Go Links Norton AntiVirus
EPSON Web-To-Page Print Print Preview
ADMINISTRATOR
Student Details
add
view
Course Details
add
view
Back
Done
Local intranet

Add Student Details

Item marked with * is required unless stated otherwise.

Student ID*

Name*

Programme*

Status

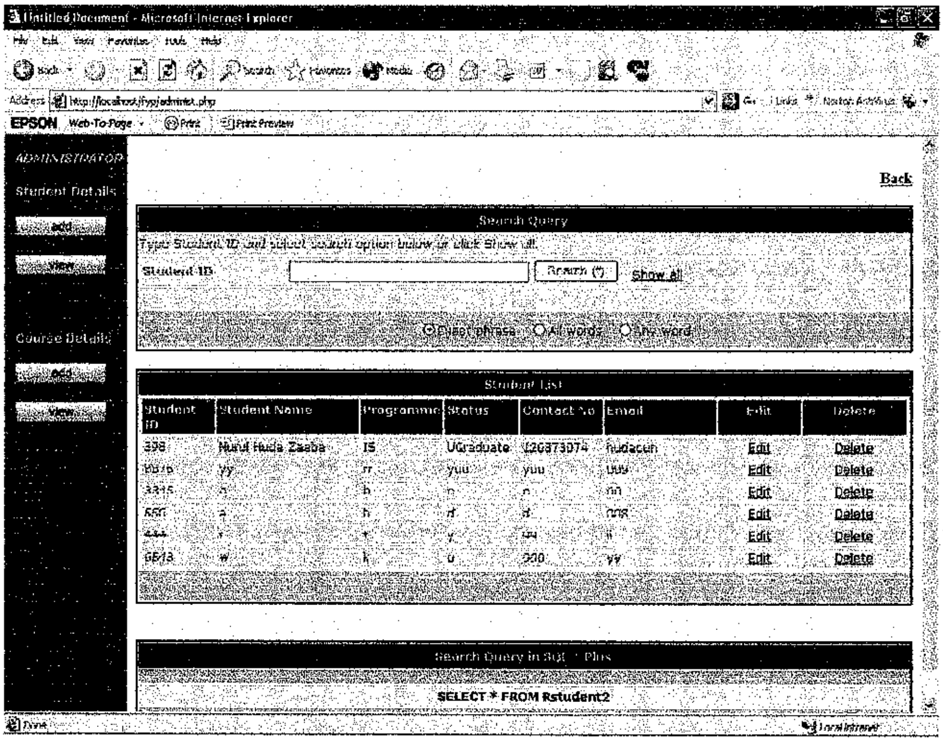
Contact No

Email

ADD Reset

< Back to List

Figure 2.2: Interface of add student record (administrator's view)


Untitled Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost/fyp/admintst.php
ADMINISTRATOR
Student Details
add
view
Course Details
add
view
Back
Search (*)
Show all
Done
Local intranet

Search Query

Type Student ID and select search option below or click Show all.

Student ID

○ Exact phrase ○ All words ○ Any word

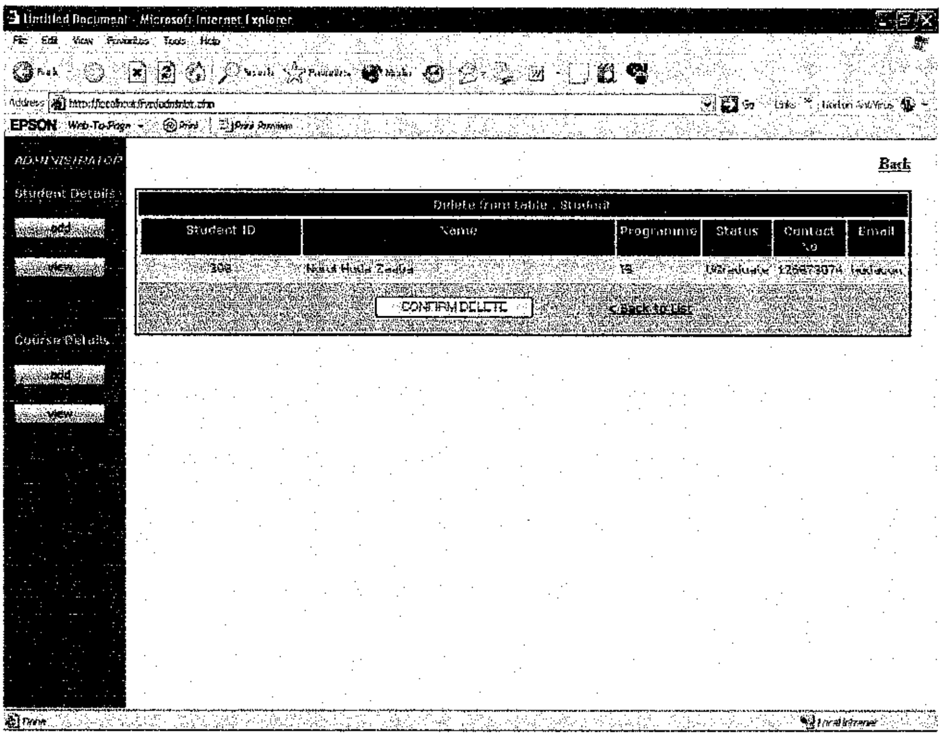
Student List

Student ID	Student Name	Programme	Status	Contact No	Email	Edit	Delete
398	Nurul Huda Zaaba	IS	UGraduate	120873074	hudacuh	Edit	Delete
8676	yy	rr	yuu	yuu	uuu	Edit	Delete
3315	n	h	n	n	nn	Edit	Delete
556	a	h	d	d	ddd	Edit	Delete
444	r	r	y	44	ii	Edit	Delete
6613	w	k	u	000	yy	Edit	Delete

Search Query in SQL * Plus

SELECT * FROM Rstudent2

Figure 2.3: Interface of query student record (administrator's view)


Untitled Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost/fyp/admintst.php
ADMINISTRATOR
Student Details
add
view
Course Details
add
view
Back
CONFIRM DELETE
< Back to List
Done
Local intranet

Delete from table : Student

Student ID	Name	Programme	Status	Contact No	Email
398	Nurul Huda Zaaba	IS	UGraduate	120873074	hudacuh

Figure 2.3: Interface of query student record (administrator's view)

The syntax of the interfaces is attached **(See appendix 6).**

4.5.1 Connection between Oracle and PHP

First of all, research is done on which is the suitable application to integrate with Oracle. I found out that PHP is a good application to integrate with Oracle.

There are several advantages using PHP to integrate with Oracle. Oracle is a powerful database for building web-based applications. PHP is famous for being quick and efficient; it can help us build fast applications that are not going to weigh down our database. Pair Oracle up with PHP, and we get a powerhouse combination.

Information about PHP

PHP (an acronym for PHP: Hypertext Preprocessor) has grown by leaps and bounds into one of the most popular web programming languages around. (According to netcraft.com, Apache commands 55% of the entire web server market, and PHP claims 38% of all those Apache servers.) It has done so by allowing us to quickly and efficiently get the job done while providing sophisticated features for more complex applications. With efficiency and low overhead serving as its prime directives, it helps produce some of the fasted web-based applications around. PHP is also open source, we do not have to wait for the vendor to fix bugs, and plenty of peer review uncovers and irons out the ones there.

PHP is basically a set of scripts much like those we might write in Perl or Python that we can directly embed in your HTML pages. In this approach, HTML serves as the basic framework for a page, while dynamic PHP code draws content and information from our Oracle database.

Oracle and PHP

PHP and Oracle integration is nothing new; in fact, Oracle was one of the first databases other than MySQL to which PHP could connect. Programmers have been building PHP applications for Oracle for years, usually by building Apache with PHP + Oracle support. What is new, however, is Oracle support for this combination and for users building PHP-based applications including documentation on OTN, as well as Metalink support for installing mod_php with Oracle Application Server.

Many books and materials are being studied in order to find the easiest way in connecting PHP and Oracle. Not forget, the internet is also surfed to find the best methods. All the information from the books and internet are gathered for the future use. There are several matters to be considered before the coding activities begin.

Below are several matters that have to be considered:

- What is PHP and what's it got to do with Oracle.
- What is the difference between the OCI and ORA extension modules.
- How does one configure PHP to use Oracle.
- How does one connect to Oracle.
- Why do we get error "Call to undefined function: ora_logon()/ ocilogon()".
- How does one SELECT, INSERT, UPDATE and DELETE data from PHP.
- How are database transactions handled in PHP.
- How are database errors handled in PHP.
- How does one call stored procedures from PHP.
- Does PHP offer Oracle connection pooling.

Before anything can be done with PHP, of course, we have to ensure that it is installed and working on our system. In this case, the application named easyPHP version 1.8 that consists of PHP application and Apache is used for this project.

The next step is how we tie PHP to Oracle. In keeping with the PHP tradition, the process will be like below:

Table 1.7 : The code and the explanation in connecting Oracle with PHP

Code	Explaination
OCILogon()	Opens a connection to Oracle. Requires that the environment variable ORACLE_SID has been set and that we have a valid username and password.
OCIParse()	Parses an SQL statement.
OCIExecute()	Executes the SQL statement.
OCI NumCols()	Gets the number of columns used in the SQL statement.
OCIFetch()	Gets the next row in the result of a SQL statement and places it in a result buffer.
OCIResult()	Gets the value of the named column in the current result row.
OCIFreeStatement()	Frees the resources in used by the current statement.
OCILogoff()	Closes the connection to Oracle.

(Source: PHP Manual,<http://www2.stack.ru/~julia/PHP4/function.ocilogon.html >

4.6 Comparison between RDBMS and ORDBMS

For this project the comparative study between RDBMS and ORDBMS will be divided by two categories; first is Logical Database Design and second one is Physical Database Design. For Logical Database Design, the study will focus on data design which is the diagram representational. In other word, there will be two diagrams that represent RDBMS and ORDBMS each and further comparison will be run.

Second category is Physical Database Design. SQL commands can be divided into two main sublanguages. In brief, The Data Definition Language (DDL) contains the commands used to create and destroy databases and database objects. After the database structure is defined with DDL, database administrators and users can utilize the Data Manipulation Language (DML) to insert, retrieve and modify the data contained within it.

4.6.1 Logical Database Design Comparison

The comparison between RDBMS and ORDBMS is done by comparing both data modeling. In this project system architecture is developed for each model. For RDBMS, diagram called Entity Relational Database (ERD) is developed that represents relational data model and Unified Manipulation Language (UML) is developed to represent object relational data model. Appendix 7 shows the ERD and UML for RDBMS and RDBMS.

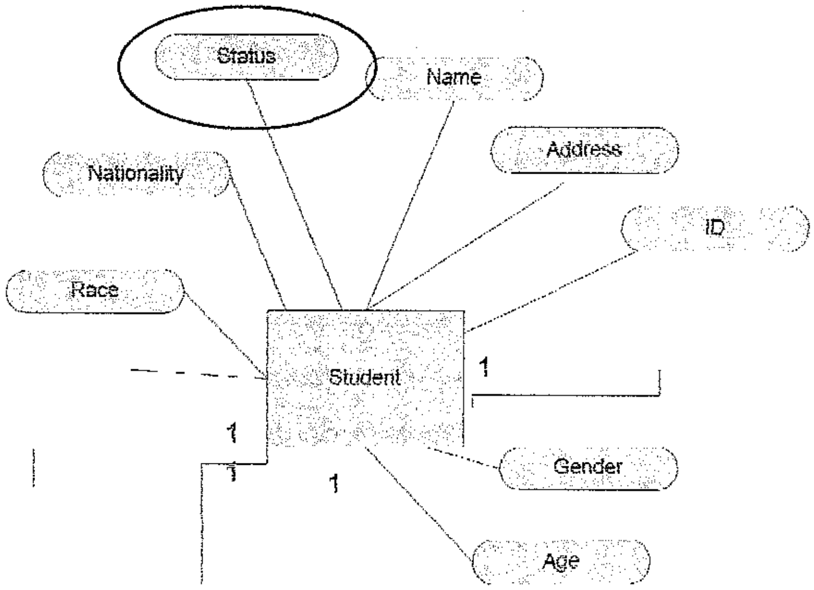


Figure 2.1: Entity of Student uses Status as one of attributes to represent status of student using ERD. For example: Undergraduate or Foundation

In this case, RDBMS approach use attribute named Status to represents the status of student by using ERD. On the other hand, ORDBMS approach creates subclasses that for status which are Foundation and Undergraduate. This is called inheritance which the subclasses will share the same attributes and/or the same methods with the superclass.

The advantage of this is that we do not have to write the same code repeatedly, we want a mechanism that takes advantage of these similarities. Inheritance is that mechanism. Inheritance models "is a" and "is like" relationships, enabling us to reuse existing data and code easily. When *A* inherits from *B*, we say *A* is the subclass of *B* and *B* is the superclass of *A*. Furthermore, we say we have "pure inheritance" when *A* inherits all the attributes and methods of *B*. The UML modeling notation for inheritance is a line with a closed arrowhead pointing from the subclass to the superclass.

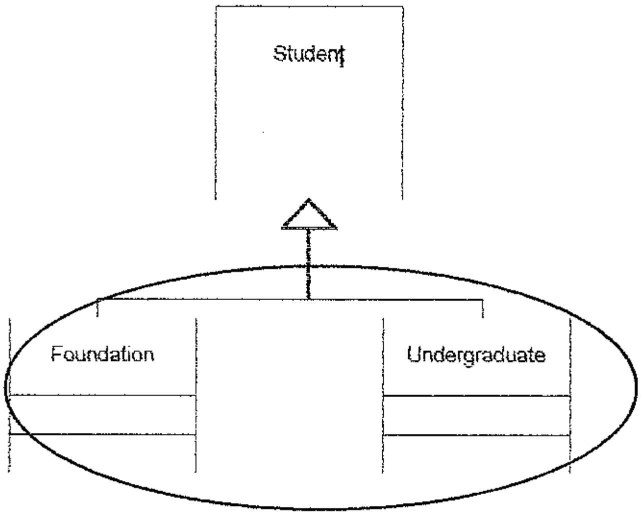


Figure 2.2: UML Diagram that shows the inheritance between Student as superclass, and two subclasses; which are Foundation and Undergraduate

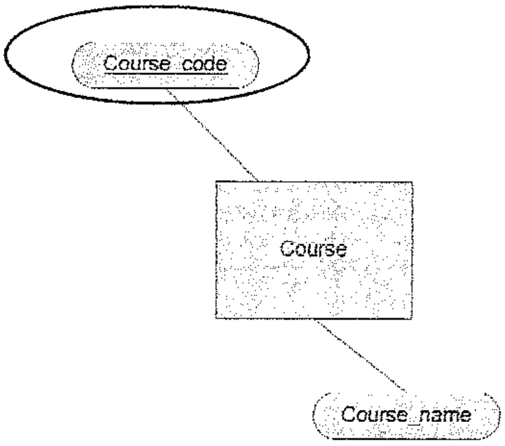


Figure 2.3: Entity of Course uses course name as one of attributes to represent status of student using ERD. For example: Year 1.

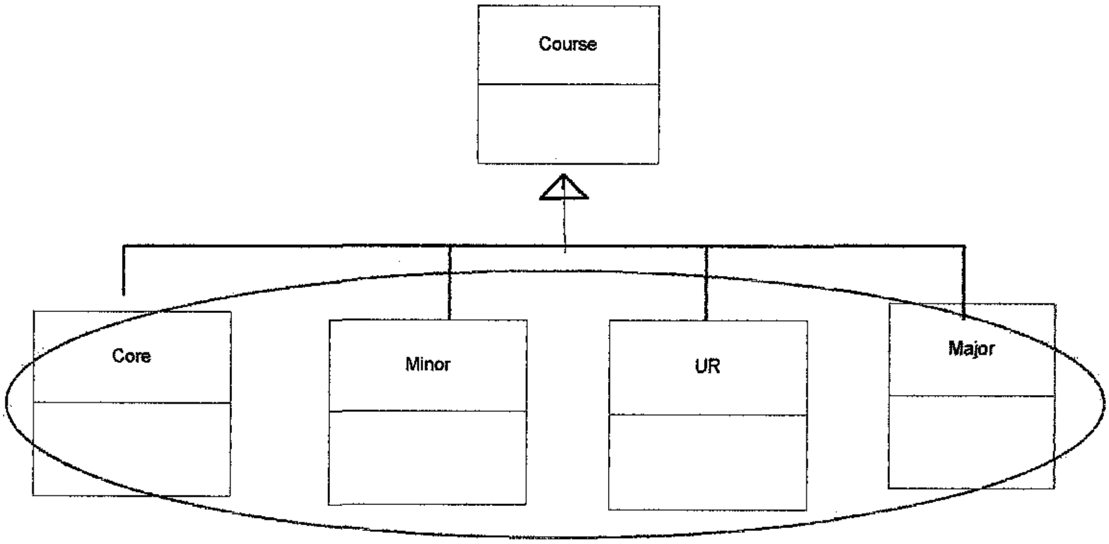


Figure 2.4: UML Diagram that shows the inheritance between Course entity as superclass, and four subclasses; which are Core, Minor,UR, Major

4.6.2 Physical Database Design Comparison

Data Definition Language

A Data Definition Language is a computer language for defining data. For example in Oracle the DDL statements refer to CREATE, DROP, ALTER, etc.

These SQL statements define the structure of a database, including rows, columns, tables, indexes, and database specifics such as file locations. DDL SQL statements are more part of the DBMS and have large differences between the SQL variations. DDL SQL commands include the following:

- Create - To make a new database, table, index, or stored query.
- Drop - To destroy an existing database, table, index, or view.
- Alter - To modify an existing database object.

In fact, there are obvious contrast between RDBMS and ORDBMS in terms of DDL.

To create a table, the command language for RDBMS is:

```
CREATE TABLE StudentR
  ( RID              INTEGER NOT NULL,
   RName             VARCHAR2(20) NOT NULL,
   RGender           VARCHAR2(6),
   RIC               VARCHAR2(14),
   RDOB              DATE,
   RProgramme        VARCHAR2(20),
   RNationality      VARCHAR2(20),
   RRace             VARCHAR2(10),
   RStatus           VARCHAR2(10),
   RContactNo        VARCHAR2(10),
   RAddress          VARCHAR2(30),
CONSTRAINT StudentR_PK PRIMARY KEY(RID));

  /
```

Use code create table

While for ORDBMS, the command language will be like this:

```
CREATE TYPE Student_Type AS OBJECT
  ( ID               Number(6),
   Name              Varchar2(20),
   Gender            Varchar2(6),
   IC                Number(14),
   DOB               Varchar2(10),
   Age               Number(6),
   Programme         Varchar2(20),
   Nationality       Varchar2(20),
   Race              Varchar2(10),
   ContactNo         Number(15),
   Address           Varchar2(30),
   Sponsor           Sponsor_Type,
   Advisor           Advisor_Type,
   Clerk             Clerk_Type,
   Course            CourseTab,
   SlipRegistration  SlipRegistration_Type,
  MEMBER FUNCTION StudentInfo RETURN VARCHAR2) NOT FINAL;
 /
```

Create type first in table creation

```
*subclass for student

CREATE or replace TYPE FoundationStudent_Type UNDER Student_Type
  ( Status           Varchar2(10),
   OVERRIDING MEMBER FUNCTION StudentInfo RETURN VARCHAR2);
  /
```

Create type for subclasses (inheritance)

```
*subclass for student
CREATE or replace TYPE UndergraduateStudent_Type UNDER Student_Type
```

```
( Status                Varchar2(10),
   OVERRIDING MEMBER FUNCTION StudentInfo RETURN VARCHAR2);
/
```

There are additional commands for ORDBMS which are creating type and the subclasses also being created under a specific class.

Type in ORDBMS

A type, in an object oriented system, summarizes the common features of a set of objects with the same characteristics. An object type has attributes, which reflect the entity's structure, and methods, which implement the operations on the entity. In programming languages, types are tools to increase programmer productivity, by insuring program correctness. If the type system is designed carefully, the system can do the type checking at compile-time, otherwise some of it might have to deferred at compile time, thus, types are mainly used at compile time to check the correctness of the programs.

So when creating type, the developer do no has to create the same set of objects with the same characteristics later on; but the developer can use existing type.

Inheritance in ORDBMS

Data inheritance is another extension to ORDBMS's type system. In this case, we recognize that Foundation student and Undergraduate student are Student; thus they have something in common (the fact of being Student), and they also have something specific. So type Student is introduced, then Foundation_student is declared as special types of Student, who inherits attributes from Student, yet has own attributes, Similarly Undegraduate_student is declared as special kind of Student, with its specific attributes.

The advantage of having this kind of formula is it leads to a better structured and more concise description of the schema. Inheritance also helps code reusability, because every program is at level at which the largest numbers of objects can share it.

Data Manipulation Language

Data Manipulation Language (DML): is a family of computer languages used by computer programs or database users to retrieve, insert, delete and update data in a database.

Currently, the most popular data manipulation language is that of SQL, which is used to retrieve and manipulate data in a Relational database.

Data manipulation languages have their functional capability organized by the initial word in a statement, which is almost always a verb. In the case of SQL, these verbs are "select", "insert", "update", and "delete". This makes the nature of the language into a set of imperative statements (commands) to the database.

After some research and study about the comparison between RDBMS and ORDBMS in terms of DML, the result is there are not major differences between them. The example below proves that the command for RDBMS and ORDBMS are the slightly the same.

DML command for RDBMS:

```
INSERT INTO Student_Table VALUES
(000398, ‘Nurul Huda Zaaba’,‘Female’,841102064432,‘02-11-1984’, 21, ‘Information
System’,‘Malaysia’,‘Malay’, 0126873074, ‘88,Jln Raya,Tmn Permata’,
Sponsor_Type(2222,‘Zaaba Ahmad’, 590812062234, ‘88,Jln Raya,Tmn Permata,Kuala
Selangor’,89424598,‘Veterinar’, ‘Father’),
Advisor_Type(1111,‘Hamidah Wahab’, 590723143390, ‘88,Jln Raya,Tmn Permata’,
89424598,‘Lecturer’),
Clerk_Type( ‘3456’, ‘Nur Safiyah Aleya Baharudin’),
CourseTab (Course_Type(‘STB 4153’, ‘ABAP II’, ‘Mr Khairul Shafee’)),
SlipRegistration_Type(567843,
(Clerk_Type( ‘3456’, ‘Nur Safiyah Aleya Baharudin’))))
;
```

DML command for ORBMS:

```
INSERT INTO StudentR VALUES
(000398, ‘Nurul Huda Zaaba’,‘Female’,841102064432, ‘IS’,‘Malaysia’,‘Malay’,
‘UGraduate’,0126873074, ‘88,Jln Raya,Tmn Permata’)
;
```

The only difference between both command languages is that the ORDBMS has the additional command which is INSERT values for other tables and not only student table itself. The reason why there is additional command is because ORBMS do not have the primary key. So in this case, values for tables that have relationship with student table such as clerk, advisor, and sponsor table is also being inserted in student table. Unlike RDBMS, the student table gets the data from other table by using the foreign key. So the values form other tables will be called using the foreign key in student table.

CHAPTER 5.0
CONCLUSION AND RECOMMENDATIONS

The project has presented an approach for using database design which are ERD and UML as the basis of comparative analysis between Relational Database and Object Relational Database designs. The result shows that Object Relational Database implement inheritance concept unlike Object Relational Database.

Besides, a comparative study also covered the differences in terms of database languages which are Database Definition Language (DDL) and Database Manipulation Language (DML). Several results had been found which were ORDBMS applied complex type and inheritance. In contrast, RDBMS applied entity integrity and referential integrity which are primary key and foreign key.

By having the comparative study between RDBMS and ORBMS, it can help the Admission and Registration unit in choosing the right model of their database. It is to ensure the smoothness and effectiveness of usage. Furthermore, it may reduce the cost of maintaining the system since the organization manages to use the right type of model for their databases.

So, for this project paper, I can say both type of models has their own strengthens and weaknesses. As for Admission and Registration Unit, both models can be used. However, relational database model is most suited for this set of data. The reason is it provides a simple data storage concept, tables and standard query language compared to ORDBMS. ORDBMS is best suited to work with complex data like images, audio, and video. GIS application is one example that suits this type of model.

Recommendations for enhancing and continuing this project are to compare and determine the differences between RDBMS and ORDBMS in terms of Data Control Language. The idea is to identify whether a user of one subclass can has the privilege of the superclass. It is because object relational model has the inheritance concept which consists of superclass and subclasses. Hence, the research will determine whether the user of subclass has the grant to view the superclass or not. The result will show which model is good in security mechanism.

Besides, this comparison also can be done in term of performance comparison. For example, we can know which model has the ability to have multiple concurrent users or the fastest transaction time which it can shows the performance of data retrieval.

In addition, we can compare in terms of data storage. Here, we can see which type of model able to store a large volume of data in a single of database.

REFERENCES

1. Greg Riccardi, 2000, ***Principles of Database Systems with Internet and Java Applications,*** Florida, Addison Wesley, page 2-16.

2. Ryan K. Stephens, Ronalds R. P., 2000, ***Database Design***, Indiana, Sams, page 38-55.

3. Micheal J. Hernandez, 2003, ***Database Design for Mere Mortals.***, Boston, page 3-24.

4. Wikipedia, the free encyclopedia,< http://en.wikipedia.org/wiki/Object-relational_database>, *retrieved on* 18th October 2005, 4.44pm.

5. Jagadish Chaterjee,ESRI, New Delhi, ***Introduction to RDBMS, OODBMS, ORDBMS***,<http://www.comptechdoc.org/independent/database/basicdb/dataordbms.html > *retrieved on* 19th October 2005, 4.40 p.m.

6. Robert Vermeulen, 1996, ***Upgrading Relational Databases with Objects,*** *New York*, page 105-110.

7. Jaymee Soni, Barbara S, Sally T, (2000), ***Object Relational and Object Oriented Database Management Systems*** <http://www.sims.berkeley.edu/academics/courses/is206/f97/GroupE/dbs.html# ORDBMS__Extending_the_Relational_Model> *retrieved on* 1st October 2005, 7.53 p.m.

8. Information Technology Services, University of Texas, ***Introduction to Data Modeling,***,<http://www.utexas.edu/its/windows/database/datamodeling/rm/overview.html > *retrieved on* 16th May 2006, 7.24 a.m.

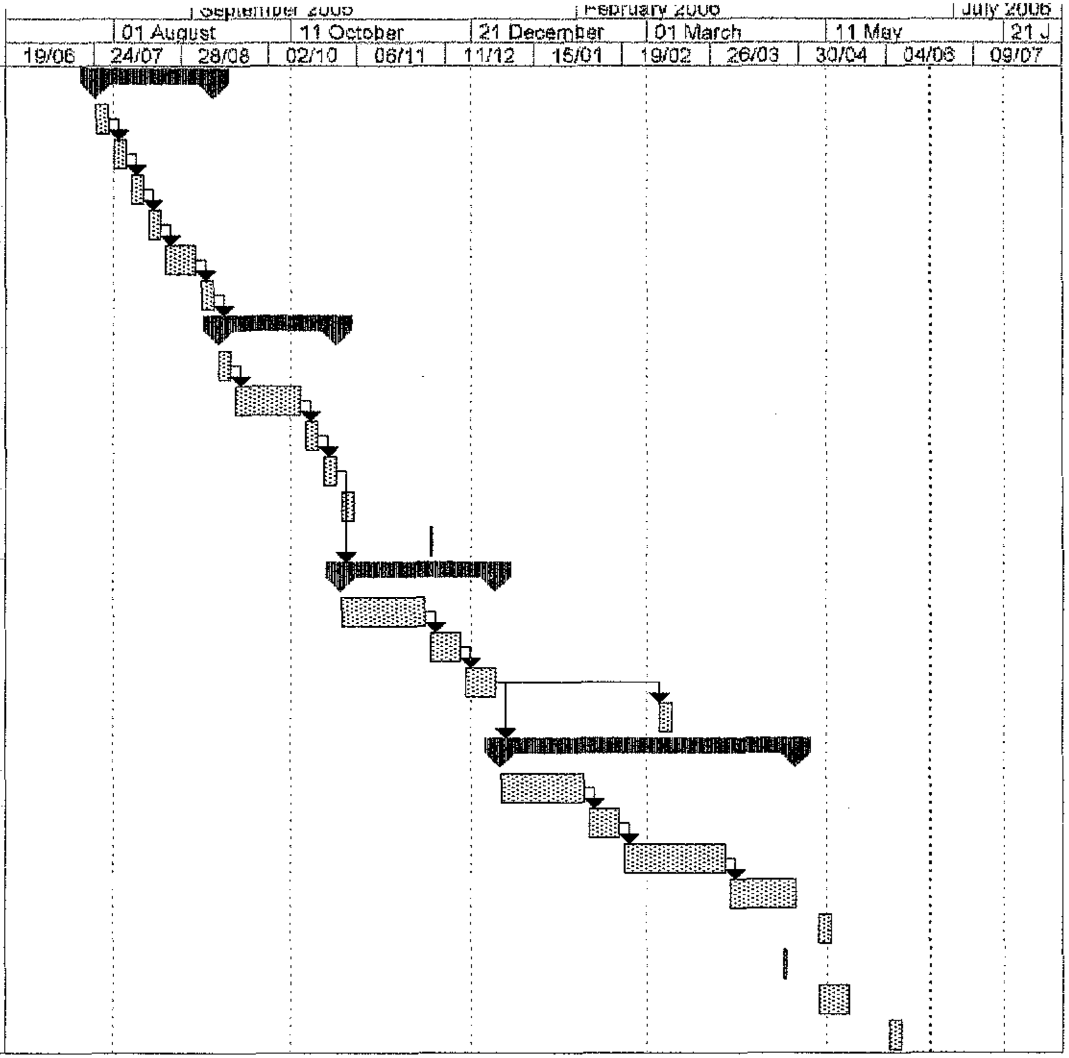
9. Sun Microsystems, ***Object-Oriented Programming Concepts***, <http://java.sun.com/docs/books/tutorial/java/concepts/inheritance.htm> *retrieve on* 16th May 2006, 11.35pm.

APPENDIX 1

MILESTONE FOR FINAL YEAR PROJECT

PART A AND PART B

1	**35 days**	**Mon 25/07/05**	**Fri 09/09/05**		
2	5 days	Mon 25/07/05	Fri 29/07/05		
3	5 days	Mon 01/08/05	Fri 05/08/05	2	
4	5 days	Mon 08/08/05	Fri 12/08/05	3	
5	5 days	Mon 15/08/05	Fri 19/08/05	4	
6	10 days	Mon 22/08/05	Fri 02/09/05	5	
7	5 days	Mon 05/09/05	Fri 09/09/05	6	
8	**35 days**	**Mon 12/09/05**	**Fri 28/10/05**	**7**	
9	5 days	Mon 12/09/05	Fri 16/09/05		
10	20 days	Mon 19/09/05	Fri 14/10/05	9	
11	5 days	Mon 17/10/05	Fri 21/10/05	10	
12	5 days	Mon 24/10/05	Fri 28/10/05	11	
13	5 days	Mon 31/10/05	Fri 04/11/05		
14	1 day	Mon 05/12/05	Mon 05/12/05		
15	**45 days**	**Mon 31/10/05**	**Fri 30/12/05**	**12**	
16	25 days	Mon 31/10/05	Fri 02/12/05		
17	10 days	Mon 05/12/05	Fri 16/12/05	16	
18	10 days	Mon 19/12/05	Fri 30/12/05	17	
19	5 days	Mon 06/03/06	Fri 10/03/06	18	
20	**85 days**	**Mon 02/01/06**	**Fri 28/04/06**	**18**	
21	25 days	Mon 02/01/06	Fri 03/02/06		
22	10 days	Mon 06/02/06	Fri 17/02/06	21	
23	30 days	Mon 20/02/06	Fri 31/03/06	22	
24	20 days	Mon 03/04/06	Fri 28/04/06	23	
25	5 days	Mon 08/05/06	Fri 12/05/06		
26	1 day	Mon 24/04/06	Mon 24/04/06		
27	10 days	Mon 08/05/06	Fri 19/05/06		
28	5 days	Mon 05/06/06	Fri 09/06/06		



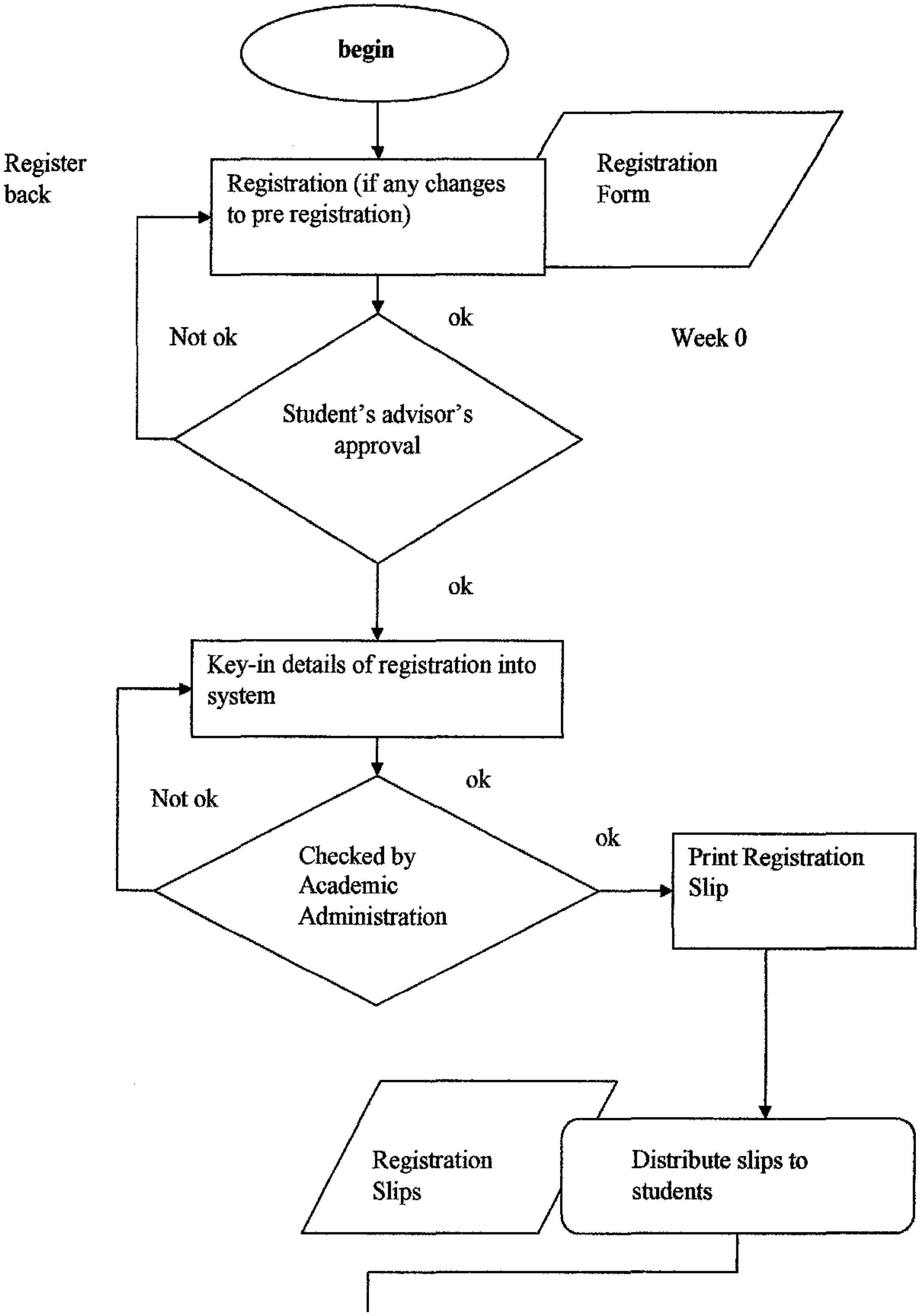
Project: duration fyp
Date: Wed 21/06/06

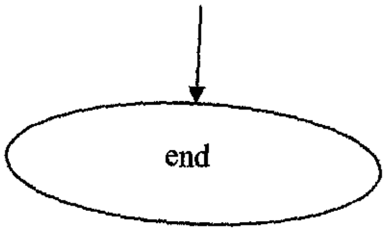
Task	▨	Milestone	◆	External Tasks	▨
Split	┆┆┆┆┆┆┆┆┆┆┆┆┆┆┆┆┆┆┆┆	Summary	▬	External Milestone	◆
Progress	▬	Project Summary	▬	Deadline	⇩

APPENDIX 2

FLOWCHART OF STUDENT MANAGEMENT SYSTEM AT ADMMISION AND REGISTRATION UNIT, UTP

Flow Chart of Admission

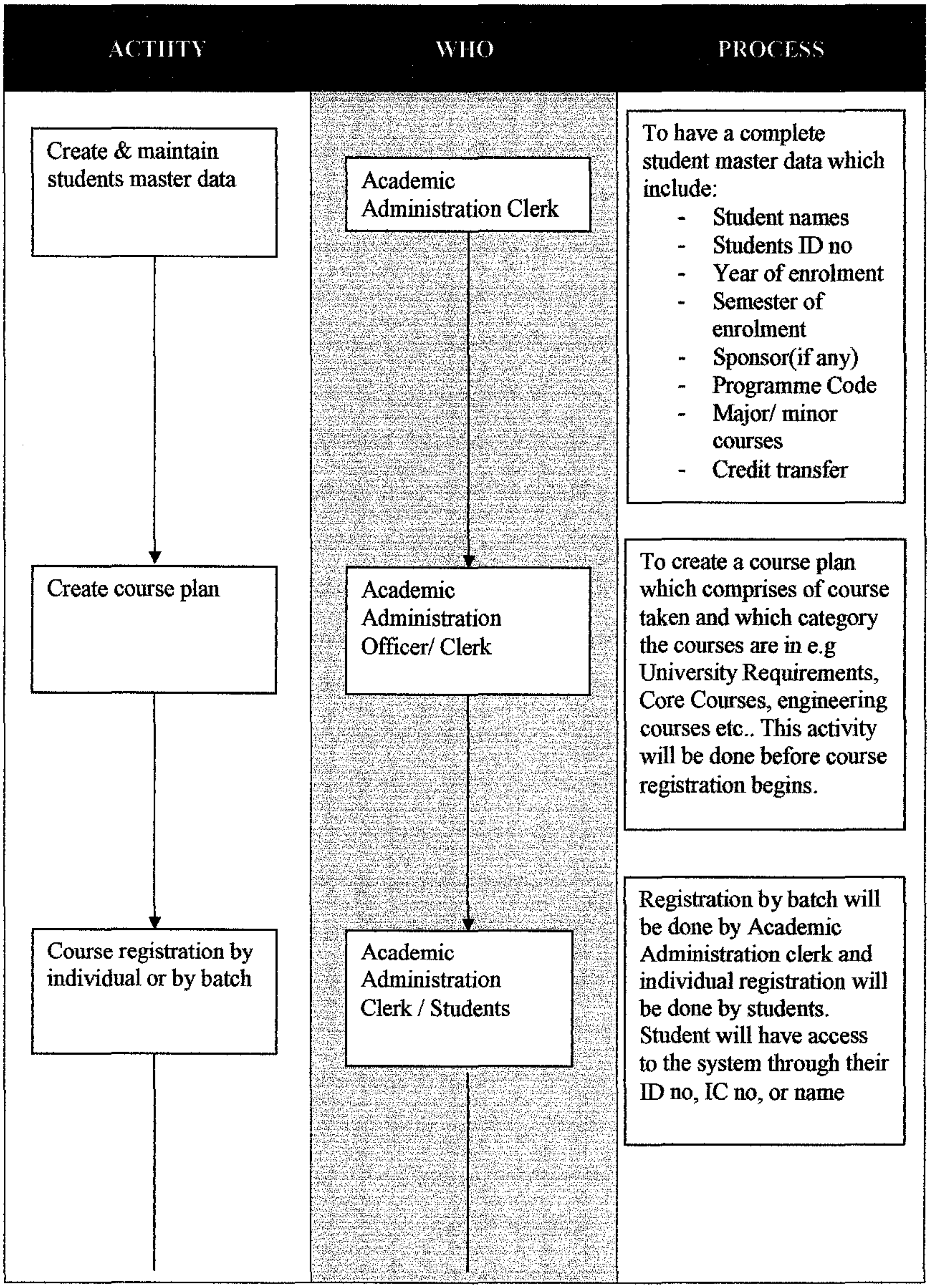

begin
Register back
Registration (if any changes to pre registration)
Registration Form
Not ok
ok
Week 0
Student's advisor's approval
ok
Key-in details of registration into system
Not ok
ok
ok
Checked by Academic Administration
Print Registration Slip
Registration Slips
Distribute slips to students


end

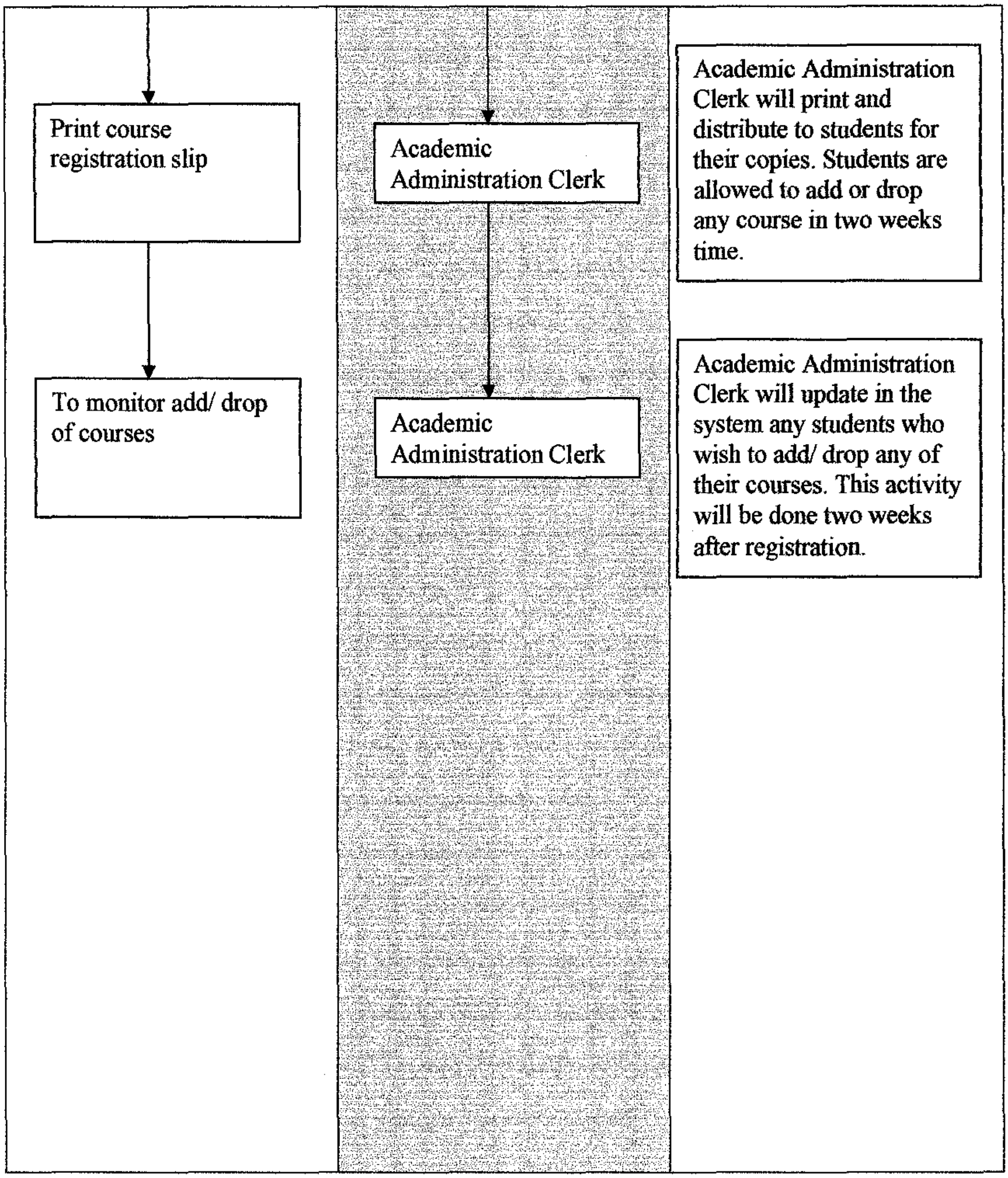
APPENDIX 3

BUSINESS FLOW OF ADMISSION

Business Flow of Admission



ACTIITY	WHO	PROCESS
Create & maintain students master data	Academic Administration Clerk	To have a complete student master data which include: - Student names - Students ID no - Year of enrolment - Semester of enrolment - Sponsor(if any) - Programme Code - Major/ minor courses - Credit transfer
Create course plan	Academic Administration Officer/ Clerk	To create a course plan which comprises of course taken and which category the courses are in e.g University Requirements, Core Courses, engineering courses etc.. This activity will be done before course registration begins.
Course registration by individual or by batch	Academic Administration Clerk / Students	Registration by batch will be done by Academic Administration clerk and individual registration will be done by students. Student will have access to the system through their ID no, IC no, or name


Print course registration slip
To monitor add/ drop of courses
Academic Administration Clerk
Academic Administration Clerk

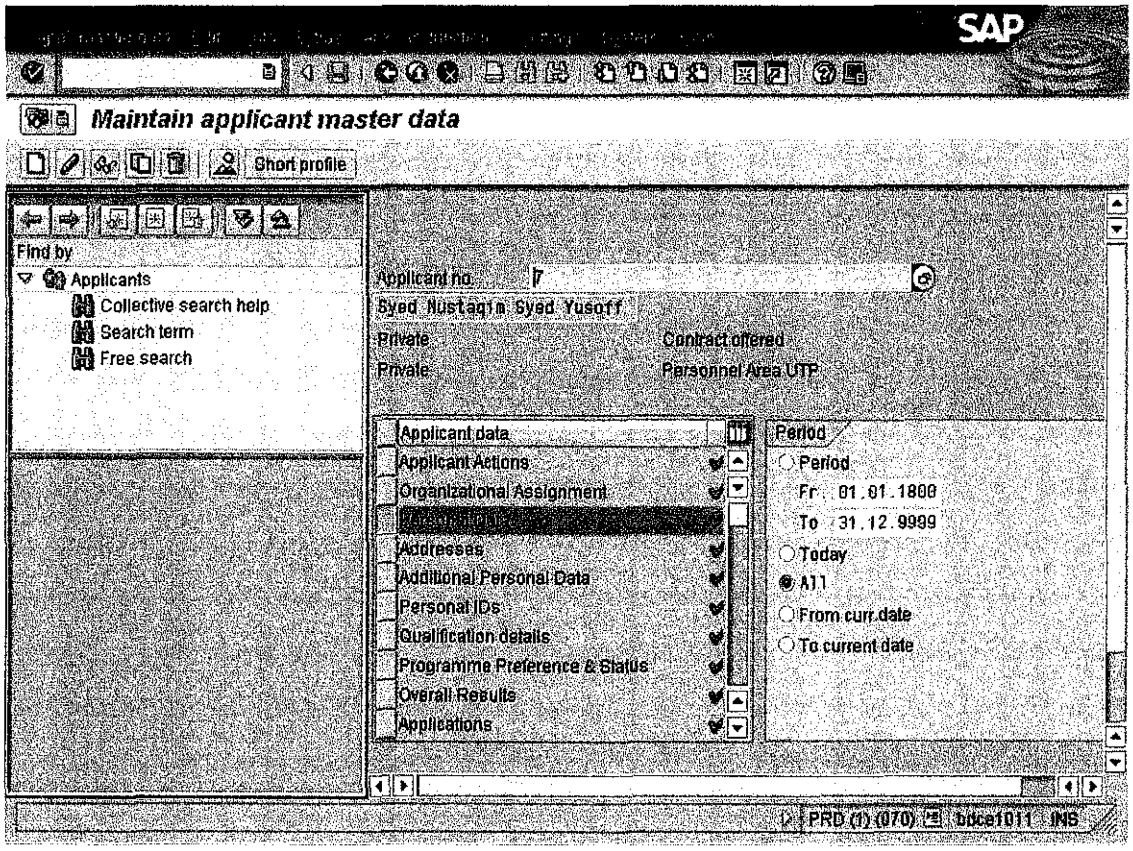
Academic Administration Clerk will print and distribute to students for their copies. Students are allowed to add or drop any course in two weeks time.

Academic Administration Clerk will update in the system any students who wish to add/ drop any of their courses. This activity will be done two weeks after registration.

APPENDIX 4

GRAPHICAL USER INTERFACE (GUI) OF CAMPUS MANAGEMENT (CM) SYSTEM

GRAPHICAL USER INTERFACE (GUI) OF CAMPUS MANAGEMENT (CM) SYSTEM



Maintain applicant master data

Applicant no. 7

Syed Mustaqim Syed Yusoff

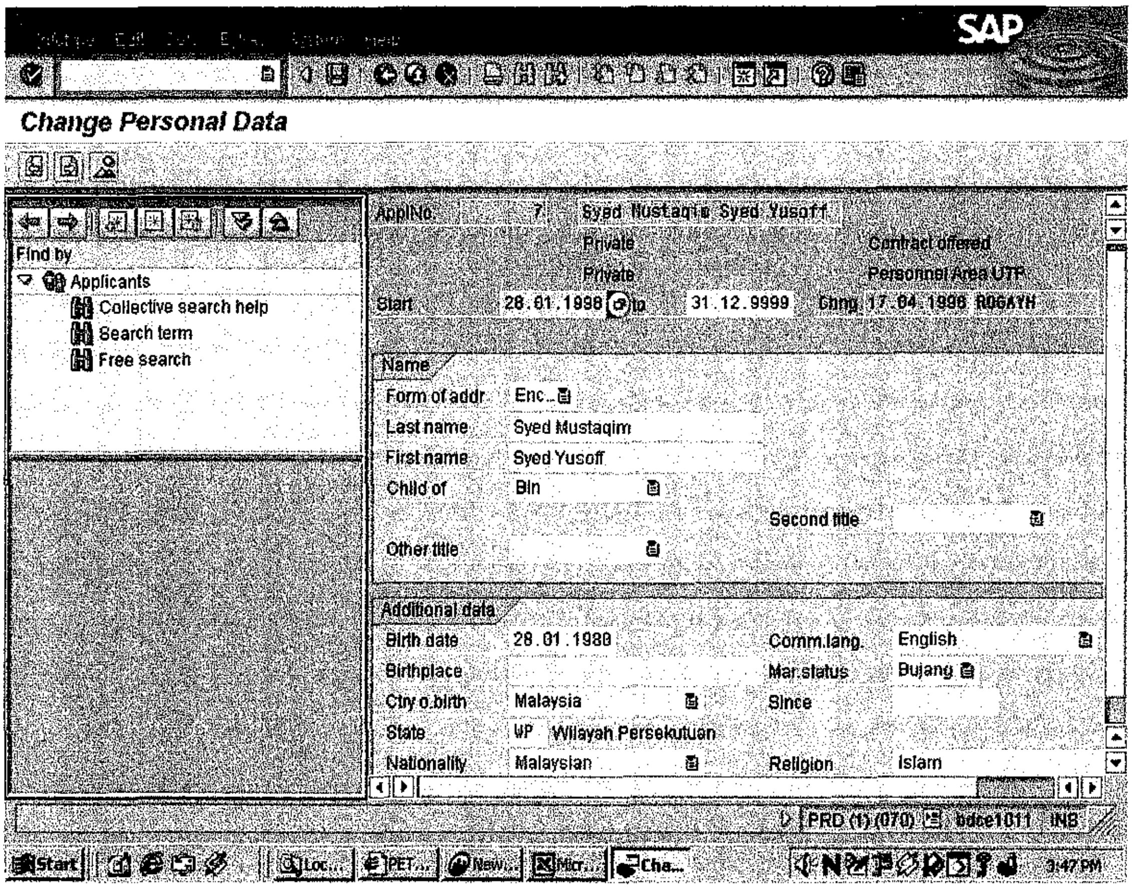
Private	Contract offered
Private	Personnel Area UTP

Applicant data
- Applicant Actions ✓
- Organizational Assignment ✓
- [illegible]
- Addresses ✓
- Additional Personal Data ✓
- Personal IDs ✓
- Qualification details ✓
- Programme Preference & Status ✓
- Overall Results ✓
- Applications ✓

Period
- ○ Period
  - Fr. 01.01.1800
  - To 31.12.9999
- ○ Today
- ● All
- ○ From curr.date
- ○ To current date

PRD (1) (070) bdce1011 INS

Figure 1: Student Master Data using SAP(Currently used in Admission and Registration Unit)

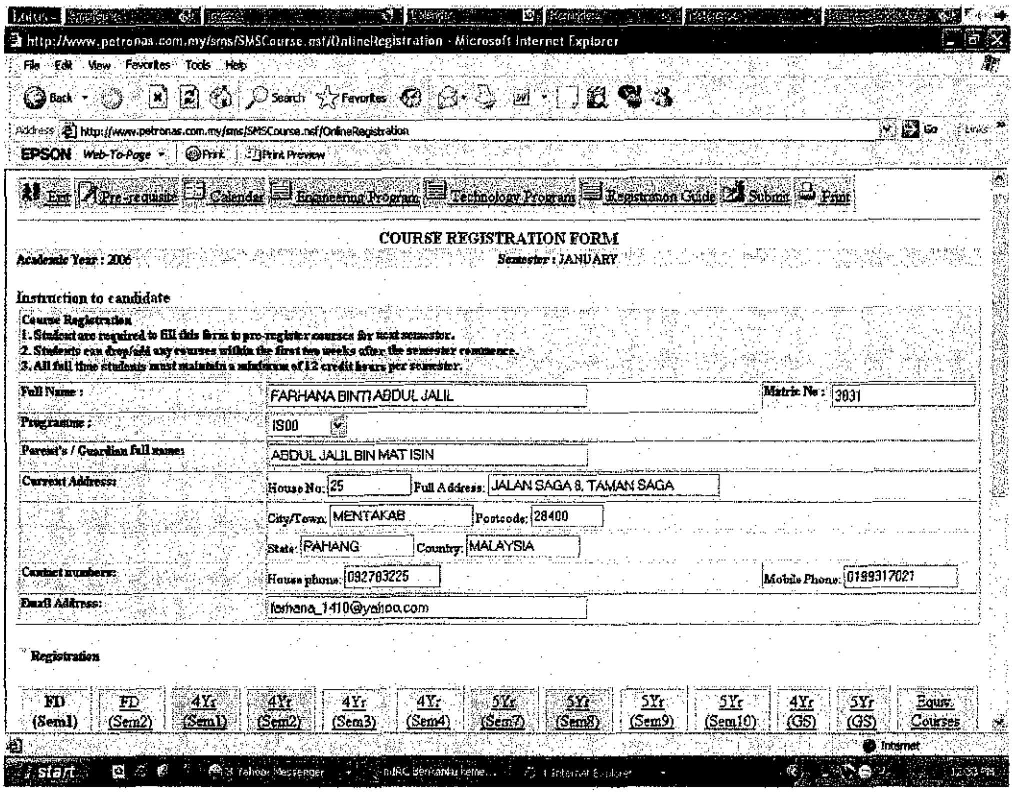


ApplNo. 7 Syed Mustaqim Syed Yusoff
Private
Contract offered
Private
Personnel Area UTP
Start 28.01.1998 to 31.12.9999 Chng 17.04.1998 ROGAYH

Form of addr Enc..
Last name Syed Mustaqim
First name Syed Yusoff
Child of Bin
Second title
Other title

Birth date 28.01.1980
Comm.lang. English
Birthplace
Mar.status Bujang
Ctry o.birth Malaysia
Since
State WP Wilayah Persekutuan
Nationality Malaysian
Religion Islam

Figure 2: Student Information Details



http://www.petronas.com.my/sms/SMSCourse.nsf/OnlineRegistration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.petronas.com.my/sms/SMSCourse.nsf/OnlineRegistration
Go Links
EPSON Web-To-Page Print Print Preview
Exit
Pre-requisite
Calendar
Engineering Program
Technology Program
Registration Guide
Submit
Print
Internet
start
Yahoo! Messenger
1 Internet Explorer

COURSE REGISTRATION FORM

Academic Year : 2006 Semester : JANUARY

Instruction to candidate

Course Registration
1. Student are required to fill this form to pre-register courses for next semester.
2. Students can drop/add any courses within the first two weeks after the semester commence.
3. All full time students must maintain a minimum of 12 credit hours per semester.

Full Name : FARHANA BINTI ABDUL JALIL
Matric No : 3031
Programme : IS00
Parent's / Guardian full name: ABDUL JALIL BIN MAT ISIN
Current Address: House No: 25 Full Address: JALAN SAGA 8, TAMAN SAGA
City/Town: MENTAKAB Postcode: 28400
State: PAHANG Country: MALAYSIA
Contact numbers: House phone: 092783225 Mobile Phone: 0199317021
Email Address: farhana_1410@yahoo.com

Registration

FD (Sem1)	FD (Sem2)	4Yr (Sem1)	4Yr (Sem2)	4Yr (Sem3)	4Yr (Sem4)	5Yr (Sem7)	5Yr (Sem8)	5Yr (Sem9)	5Yr (Sem10)	4Yr (GS)	5Yr (GS)	Equv. Courses

Figure 3: Course Registration Form at http://www.petronas.com.my/sms/SMSCourse.nsf/OnlineRegistration

APPENDIX 5

DATA DICTIONARY (RELATIONAL MODEL)

Dictionary

Entity	Attribute	Data Type	Length	Method	Remark
Student	ID{PK}	NUMBER	6		
	Name	VARCHAR2	6		
	Status	VARCHAR2	14		
Sponsor	S_Code{PK}	NUMBER	6		
	S_Name	VARCHAR2	40		
	S_ContactNo	NUMBER	14		
Advisor	A_ID{PK}	NUMBER	6		
	A_Name	VARCHAR2	40		
	A_IC	NUMBER	14		
	A_ContactNo	NUMBER	8		
Course	CourseCode{PK}	NUMBER	40		
	CourseName	VARCHAR2	6		
			40		
SlipRegistration	Slip_No	NUMBER	6		
Clerk	Clerk_ID{PK}	NUMBER	6		
	Clerk_Name	VARCHAR2	40		
Finance	Finance_Ref	VARCHAR2	40		
Exam Unit	Exam_Ref	VARCHAR2	40		
Reg_details	Student_ID	NUMBER	40		
	Student_Name	VARCHAR2	6		
	Student_Year	NUMBER	40		
	Student_Course	NUMBER	10		
Course_Plan	Course_Name	VARCHAR2	40		

APPENDIX 6

-SYNTAX OF INTERFACE OF THE PROJECT

-CODE OF TABLE CREATION (RDBMS AND ORBMS)

SQL STATEMENT FOR RDBMS TABLE CREATION

```
CREATE TABLE Sponsor2
  ( RS_ID                              INTEGER,
    RS_Name                   VARCHAR2(20) NOT NULL,
    RS_ContactNo                 INTEGER NOT NULL,
    SID                          INTEGER,
    CONSTRAINT Sponsor2_PK PRIMARY KEY (RS_ID),
   CONSTRAINT Sponsor2_FK1 FOREIGN KEY (SID) REFERENCES        Rstudent2(SID));


  /


CREATE TABLE advisor2
  ( AID                         INTEGER,
    Aname                 VARCHAR2(10),
    AOccupation                 VARCHAR2(10),
    RCourseCode          VARCHAR2(8) NOT NULL,
 SID                            INTEGER,
studentEnrolled     VARCHAR2(7),
CONSTRAINT advisor2_PK PRIMARY KEY (AID),
   CONSTRAINT advisor2_FK1 FOREIGN KEY (SID) REFERENCES Rstudent(SID),
CONSTRAINT advisor2_FK2 FOREIGN KEY (StudentEnrolled) REFERENCES CourseEnrolled2(StudentEnrolled));


CREATE TABLE clerk2
  ( CID                   INTEGER,
    cname                VARCHAR2(40),
SID      INTEGER,
CONSTRAINT clerk2_PK PRIMARY KEY (CID),
CONSTRAINT clerk2_FK1 FOREIGN KEY (SID) REFERENCES Rstudent2(SID));

  /



CREATE TABLE RegDetails2
  ( RRegDetails_ID           INTEGER NOT NULL,
    RCourseTaken             VARCHAR2(14),
    CID                     INTEGER,
   CONSTRAINT RegDetails2_FK1 FOREIGN KEY (CID) REFERENCES Clerk2(CID));
/



 CREATE TABLE ExamUnit2
  ( RExam_ref                    VARCHAR2(40));
     /



CREATE TABLE Finance2
  (RFinance_ref             VARCHAR2(40));

  /


CREATE TABLE RStudent2
  ( SID              INTEGER,
    SName            VARCHAR2(20) NOT NULL,
   SProgramme        VARCHAR2(20),
    SStatus          VARCHAR2(10),
    SContactNo        VARCHAR2(10),
    SEmail           VARCHAR2(30),
  CONSTRAINT RStudent2_PK PRIMARY KEY(SID));




  /
```

*link at many side between student and Course using complex datatype
*superclass

```
CREATE TABLE Rcourse2
   (Rcoursecode              VARCHAR2(7) NOT NULL,
    Rcoursename              VARCHAR2 (25) NOT NULL,
    Rcredithour               INTEGER NOT NULL,
    SID                     INTEGER,
CONSTRAINT RCourse2_PK PRIMARY KEY (Rcoursecode),
CONSTRAINT RCourse2_FK2 FOREIGN KEY (SID) REFERENCES RStudent2(SID));
```

```
CREATE TABLE  courseEnrolled2
( StudentEnrolled          VARCHAR2(7) NOT NULL,
CourseName1              VARCHAR2(30) NOT NULL,
CourseName2              VARCHAR2(30) NOT NULL,
CourseName3              VARCHAR2(30) NOT NULL,
CourseName4              VARCHAR2(30) NOT NULL,
CourseName5              VARCHAR2(30) NOT NULL,
CourseName6              VARCHAR2(30) NOT NULL,
   CONSTRAINT RCourseEnrollled2_PK PRIMARY KEY (StudentEnrolled));
```

*link on many side between SlipRegistration and Clerk

```
CREATE TABLE SlipRegistration2
   ( RSlipNo                INTEGER NOT NULL,
     CID                   INTEGER,
CONSTRAINT SlipRegistration2_PK PRIMARY KEY (RSlipNo),
 CONSTRAINT SlipRegistration2_FK FOREIGN KEY (CID) REFERENCES clerk2(CID));
```

```
  /
```

<u>DESCRIBE TABLE</u>

```
DESC  RStudent2

DESC Rcourse2

DESC Sponsor2
```

SQL STATEMENT FOR ORDBMS TYPE AND TABLE CREATION

```
CREATE TYPE ORSponsor1_Type AS OBJECT
  ( SponsorID                          Number(6),
    SponsorName                        Varchar2(40),
    SponsorContactNo          Number(10),
    ORStudent1                      ORStudent1Tab);

  /


CREATE TYPE ORAdvisor1_Type AS OBJECT
  ( A_Code              Number(6),
    A_Name                  Varchar2(20),
    A_ContactNo             Number(10),
    ORStudent               ORStudent1Tab,
CourseEnrolled            ORCourseEnrolled1_Type);
  /

CREATE TYPE ORClerk1_Type AS OBJECT
  ( C_ID                Number(6),
    C_Name              Varchar2(40),
   ORStudent1                        ORStudent1Tab,
 ORSlipReg1                          ORSlipReg1Tab,
  ORRegDetails1              ORRegDetails1Tab);

   /



CREATE TYPE ORRegDetails1_Type AS OBJECT
  ( S_ID                 Number(6),
     S_Name                 Varchar2(40),
     S_Programme         Varchar2(20),
    CourseTaken          Number(14));



CREATE TYPE ORCourseEnrolled1_Type AS OBJECT
 ( StudentEnrolled        VARCHAR2(7),
CourseName1             VARCHAR2(50),
CourseName2             VARCHAR2(50),
CourseName3             VARCHAR2(50),
CourseName4             VARCHAR2(50),
CourseName5             VARCHAR2(50),
CourseName6             VARCHAR2(50));


  /




 CREATE TYPE ORExamUnitE1_Type AS OBJECT
   ( ReferencePersonE                  Varchar2(40),
    ORRegDetails1                      ORRegDetails1Tab);

  /

CREATE TYPE ORFinanceF1_Type AS OBJECT
  ( ReferencePersonF          Varchar2(40),
   ORRegDetails1                       ORRegDetails1Tab);

  /


CREATE TYPE ORStudent1_Type AS OBJECT
  ( studentID                  Number(6),
    studentName             Varchar2(20),
    studentProgramme        Varchar2(20),
    studentContactNo         Number(15),
    studentEmail            Varchar2(30))NOT FINAL;
  /
*subclass for student
```

```
CREATE or replace TYPE FoundationStudent1_Type UNDER ORStudent1_Type
   ( Status                    Varchar2(10));
   /

*subclass for student
CREATE or replace TYPE UndergraduateStudent1_Type UNDER ORStudent1_Type
( Status                    Varchar2(10));
/


*link at many side between student and Course using complex datatype
*superclass
CREATE TYPE ORCourse1_Type AS OBJECT
   (C_CourseCode               Varchar2(14),
    C_CourseName          Varchar2 (40),
    C_credithour      Varchar2 (5)) NOT FINAL;
  /

*subclass
CREATE or replace TYPE ORMinor1_Type UNDER ORCourse1_Type
   ( C_Status1                  Varchar2(14));
   /

*subclass
CREATE or replace TYPE ORMajor1_Type UNDER ORCourse1_Type
   ( C_Status2                     Varchar2(14));
   /

 *subclass
CREATE  or replace TYPE ORUR1_Type UNDER ORCourse1_Type
   ( C_Status3                  Varchar2(14));
   /

*subclass
CREATE or replace TYPE ORCore1_Type UNDER ORCourse1_Type
   ( C_Status4                              Varchar2(14));
   /



*link on many side between SlipRegistration and Clerk
CREATE TYPE ORSlipReg1_Type AS OBJECT
   ( SlipNo               Number(6));

  /
*link on many side between CoursePlan and Clerk
CREATE TYPE ORCoursePlan1_Type AS OBJECT
   ( CourseNameP            Varchar2(40),
     CourseYearP            Number(6));
  /



CREATE TYPE ORCourse2Tab AS TABLE OF ORCourse1_Type;
   /

CREATE TYPE ORAdvisor1Tab AS TABLE OF ORAdvisor1_Type;
   /

CREATE TYPE ORStudent1Tab AS TABLE OF ORStudent1_Type;
   /

CREATE TYPE ORSlipReg1Tab AS TABLE OF ORSlipReg1_Type;
   /

CREATE TYPE ORRegDetails1Tab AS TABLE OF ORRegDetails1_Type;
/
CREATE TYPE ORCoursePlan AS TABLE OF ORCoursePlan1_Type;
/


*table for association class
CREATE TABLE ORCourseRegDetails1
```

```
   (C_coursecode      ORcourse2tab,
    StudentID         ORstudent1tab)
   NESTED TABLE C_coursecode    STORE AS ORcourse_List2,
   NESTED TABLE studentID STORE AS ORstudent_List2;
```

```
*table for association class
CREATE TABLE ORAdvisedBy
   ( GroupName          Varchar2(40),
    C_CourseCode          ORCourse2Tab,
    A_Name          ORAdvisor1Tab)
   NESTED TABLE C_CourseCode    STORE AS ORCourse2_List2,
   NESTED TABLE A_Name     STORE AS ORAdvisor_List;


CREATE TABLE ORSponsor1_Table OF ORSponsor1_Type
NESTED TABLE ORStudent1 STORE AS orStudentTable;


CREATE TABLE ORExamUnit1_Table OF ORExamUnitE1_Type
NESTED TABLE ORRegDetails1 STORE AS ORRegDetailsTable;

CREATE TABLE ORFinanceUnit1_Table OF ORFinanceF1_Type
NESTED TABLE ORRegDetails1 STORE AS ORRegDetailsTableF;

CREATE TABLE ORClerk1_Table OF ORClerk1_Type
NESTED TABLE ORSlipReg1  STORE AS ORSlipRegTableC
NESTED TABLE ORRegDetails1 STORE AS ORRegDetailsTableC
NESTED TABLE ORStudent1 STORE AS StudentTableC;

CREATE TABLE ORAdvisor_Table1 OF ORAdvisor_Type
NESTED TABLE ORStudent STORE AS ORStudentList;



CREATE TABLE ORStudent1_Table OF ORStudent1_Type;

CREATE TABLE ORCourse1_Table OF ORCourse1_Type;

CREATE TABLE  ORMinor1_Table OF ORMinor1_Type;

CREATE TABLE ORMajor1_Table OF ORMajor2_Type;

CREATE TABLE ORUR1_Table OF ORUR1_Type;

CREATE TABLE ORCore1_Table OF ORCore1_Type;

CREATE TABLE ORRegDetails1_Table OF ORRegDetails1_Type;

CREATE TABLE  ORFoundationStudent1_Table OF FoundationStudent1_Type;

CREATE TABLE ORUndergraduateStudent1_Table OF UndergraduateStudent1_Type;

CREATE TABLE ORCourseEnrolled1_Table OF ORCourseEnrolled1_Type;
```

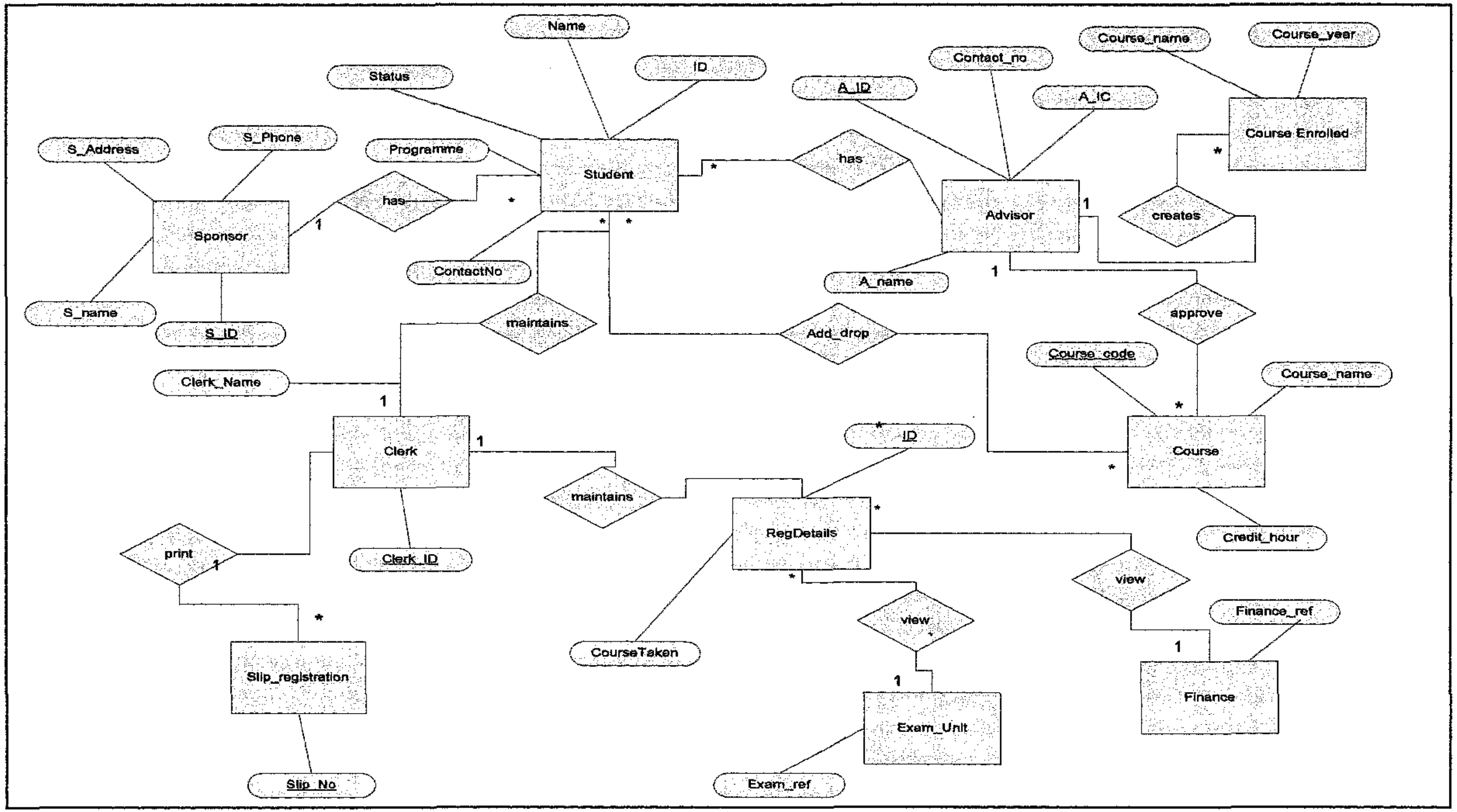
DESCRIBE TABLE

```
DESC ORCourse1_Table

DESC  ORFoundationStudent1_Table

DESC ORUndergraduateStudent1_Table
```

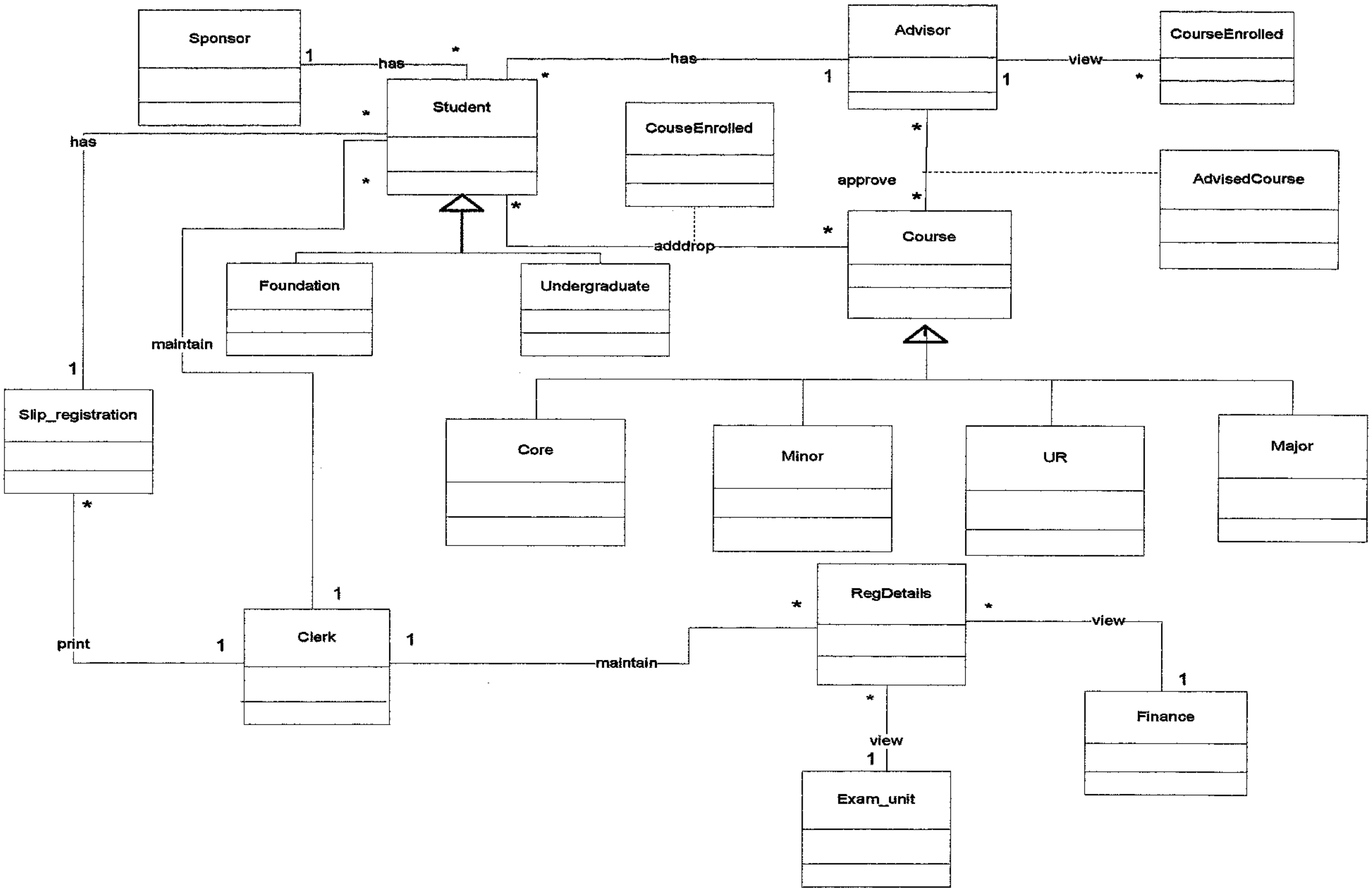
APPENDIX 7

ER DIAGRAM (RELATIONAL MODEL)

UML (OBJECT RELATIONAL MODEL)



Appendix 7.2: ERD for Object Relational Database Model



Appendix 7.2: ERD for Object Relational Database Model

APPENDIX 8

INTERVIEW QUESTIONS WITH ADMISSION AND REGISTRATION UNIT, UTP

INTERVIEW QUESTIONS
Final Year Project (Research)
Information Technology and Information System Dept

UNIVERSITI
TEKNOLOGI
PETRONAS

"What is the purpose of your unit?"

"Why do you feel that you need a database?"

Why is the database important from your standpoint?

"How do you know that a database will solve the problem?"

"What is your opinion with the system that you have now?"

Have upgrades been forecasted for the near future?