# 3.0 SYSTEM ANALYSIS

This chapter intents to ascertain the functional and non-functional requirements of this project. System development methodology and project planning will be discussed here. Besides that, the determination of the programming language, database and hardware is needed. A mixture of various kinds of tools will be determined and used to build this project.

## 3.1 Methodology

There are many types of software process models in software engineering. The software model process selected for this project is software prototyping. This model wills clearly shows what goes on during the development process, and it suggests developing the sequence of events the developers should encounter. (Sommerville, 2002)

Prototyping is a worthwhile technique for gathering specific information before a " real " system is being developed. It enables the developer to create a model of the system and refine it into a newer version, or discard and start creating another model that will fulfill the developer's and user's needs.

Prototyping allows more flexibility to meet with new requirements and the addition of more innovative features. The prototyping model is shown in Figure 2.

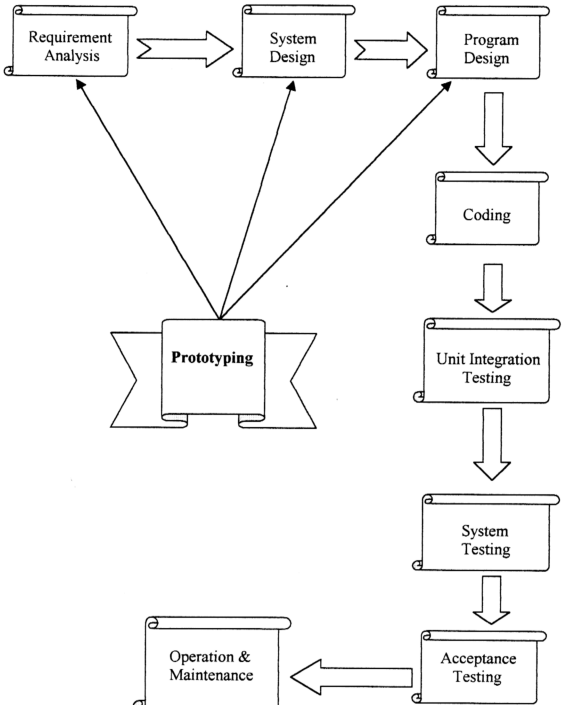


**Figure 2: Prototyping Method**

Prototyping starts with getting the requirements of analysis, where the developer has to come up with a broad conceptual solution. A detailed analysis of user's requirements has to be produced. The system design as well as the program design will then be constructed.

Based on the design, the prototype will be developed, then tested and refined. A process of iteration is needed to refine the prototype until all requirements are achieved or until the prototype has evolved into a system suitable to be implemented. During this phase, refining will be done with reference to the requirements of analysis, system and program design.

After confirming the program design, coding for database, user interface, and others will be carried out. Upon completion of this phase, all the units will be integrated into a complete system and tested to ensure that no flaws or bugs exist. Troubleshooting will also be done in this phase. The last phase is the last acceptance test and to maintain the system from time to time.

## 3.2 Project Schedule

To achieve the objectives of the project in time, a timeline is planned to manage tasks. The schedule will ensure the project is accomplished before the submission date. The project schedule is shown in Table 1.

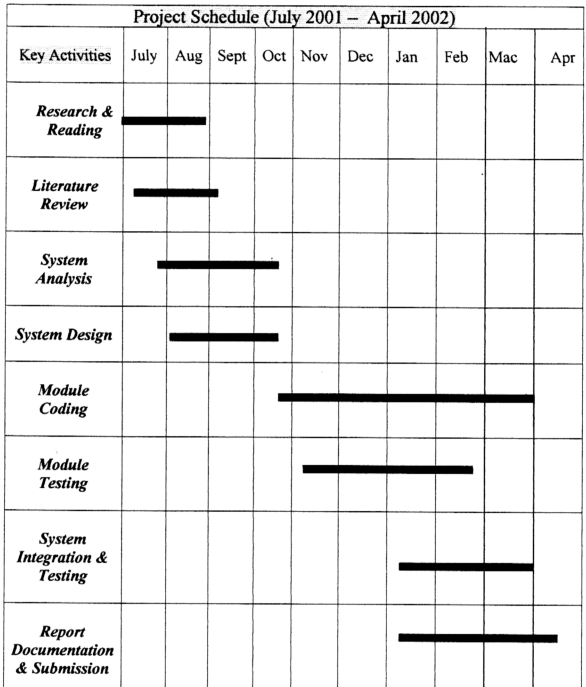


**Table 1: Project Schedule**

## 3.3 Requirements Analysis

Implementation of requirement analysis is to discuss more about abstract information and restrictions of the online course registration and payment system. Requirements analysis can be divided into two – functional requirements and non-functional requirements.

### 3.3.1 Functional Requirements

Functional requirements are the main features that will complete the operation and produce required results of the online course registration and payment system. The lists of functional requirements are as below.

#### 3.3.1.1 Students & Administrators Login Function

This function will allow or restrict students or administrators to access into the following page. Students and administrators need an ID and password to login.

#### 3.3.1.2 Change Password Function

This function allows students and administrators to change their password if they feel the old password is not safe.

### 3.3.1.3 Payment Calculating Function

This function will assist students to calculate the course registration payment based on the information from the database. The registration fees and examination fees will be automatically summed up. Therefore, the total amount that shown in the calculation would be the total fee that includes registration fee and examination fee.

### 3.3.1.4 Data Transaction

The modules involved here allow the end users *(UM administrators)* to create, retrieve, delete and update students' information, administrative records and course registration information. This component will ensure the database remains current with the 'real world' and support the information need of the UM administrators.

### 3.3.1.5 Data Queries

These modules allow UM administrators to retrieve the details of records from the database based on:

i. Keywords such as student's matrix number, administrator's ID, and course number.

ii. Non-keywords such as students and administrators login time.

### 3.3.2 Non-Functional Requirements

Non-functional modules are the modules that will not affect the system when they are operating and producing the required output. In this system, alteration in administrative records will not affect the required results. For example, a change in administrators' name, ID, and password will not affect the course registration and payment result. Besides that, changing information in the timetable will not affect the required result of the system as well.

#### 3.3.2.1 User Interface

The user interface in the system should present the required information in a user-friendly way. When designing the forms, some useful guidelines need to be followed, including meaningful title, logical grouping and sequencing of fields, visually appealing layout of the forms, familiar field labels, use of distinguished colors, visible space and boundaries for data-entry fields and convenient cursor movement.

#### 3.3.2.2 Storage Capacity

The storage capacity of the database must be enough for the postgraduate students in FSKTM. Besides that, the RAM (Random Access Memory) of the computer must be high enough to support the workload.

#### 3.3.2.3 Easy to Use and Understand

A standard navigation buttons and interface design of the system must be used. Furthermore, the information in the system has to be easy to understand.

#### 3.3.2.4 Reliable

The system has to be reliable with less error occurred. The security of the data must be control by User ID and password.

#### 3.3.2.5 Robust

The system has to be robust. Only a little of time need to spend to recover the system when system failure. Data error has to be minimized after system failure with the validation of data if the data are safe to transfer to database.

## 3.4 Run-Time Requirements

Online Postgraduate Course Registration and Payment System is developed to run on a standalone PC. This standalone PC can act as a server. The operating system for development work is Windows 95/98/2000/ME.

#### 3.4.1 Window 95/98/2000/ME

- At least a Pentium II 300 MHz machine

- 64 MB RAM (recommended for performance wise)
- 3.2 GB Hard Disk (recommended for performance wise)
- 52 X CD_ROM Drive
- Modem with at least 56 Kbps
- Other standard computer peripherals

## 3.5 Data Flow Diagram

Data Flow Diagram (DFD) is a graphical characterization of the data processes and flows in a system. DFD depicts the broadest possible overview of system inputs, processes, and output, which correspond to the data movement through the system. Data flow diagram in this system is shown as below (Figure 3).

### i. Context Diagram

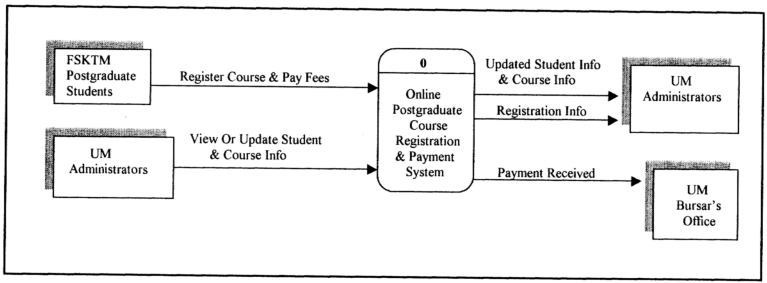


**Figure 3: Context Level Diagram For An Online Postgraduate Course Registration and Payment System**

There are three external entities for this system, namely FSKTM Postgraduate Students UM Bursar's Office, and UM Administrators. These entities are shown in the Context Diagram. The Online Postgraduate Course Registration and Payment System process symbol is a parent process and the details of this process are defined in its child diagram (Diagram 0).

ii. Diagram 0 (Level 0)

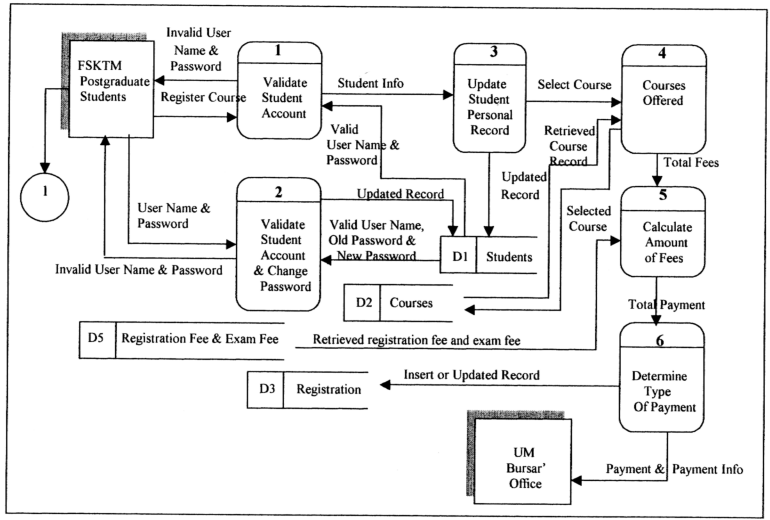


**Figure 4: Diagram 0 For An Online Postgraduate Course Registration and Payment System**

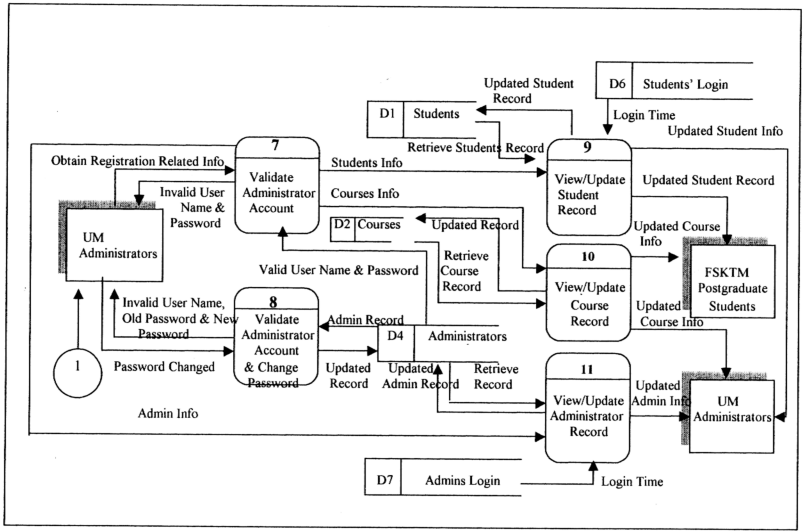


**Figure 5: Diagram 0 (Continue)**

These diagrams (Figure 4 & Figure 5) are the more detailed level for Online Postgraduate Course Registration and Payment System.

Process and data stores are numbered. There are eleven processes and seven data stores.