**Chapter 1 OBJECTIVE AND SCOPE OF THE PROJECT**

The proposed system deals with all included activities handled by a computer center like enquiry, registration, faculty details, courses, fee information etc. This system is developed to eliminate all the complexities that can even occurred during the handling of a computer center. Now these days often seen that manual handling of a computer institute system has become more tedious task because a lot of many student comes for enquiring about different courses, few of them registered. A lot of courses are being conducted by the computer institute and to maintain faculty detail manually is also a tuff job. Since these works does manually thus it takes more time and there are chances of errors such as wrong fee details, wrong student registration etc.

**Chapter 2 THEORETICAL BACKGROUND AND PROBLEM DEFINATION**

The proposed system deals with all included activities handled by a computer center like enquiry, registration, faculty details, courses, fee information etc. This system is developed to eliminate all the complexities that can even occurred during the handling of a computer center. Now these days often seen that manual handling of a computer institute system has become more tedious task because a lot of many student comes for enquiring about different courses, few of them registered. A lot of courses are being conducted by the computer institute and to maintain faculty detail manually is also a tuff job. Since these works does manually thus it takes more time and there are chances of errors such as wrong fee details, wrong student registration etc. So to reduce these faults Computer Institute Management System has been developed. The proposed system. Computer Institute Management System deals with all the activities done by any computer institute. So this project is developed for all those institutes who are dealing with enquiry, registration, course details, and fee details processes. This system is more flexible and efficient than manual work and reduce the total time consume at error chances. As we know that if we use a new system on existing system then we must want to know about objective of proposed system. Following are the some key objective of proposed system that makes it better than existing system.

**Chapter 3 SYSTEM ANALYSIS**

**3.1 REQUIREMENT ANALYSIS**

**3.1.1 FUNCTIONAL REQUIREMENTS**

**a) Strong Data Validation:**

There is possibility that user might enter wrong data and wrong data may cause inconsistency to the database and hence to the system. To avoid this, data should be validated whenever entered.

**b) Automatic updating of the database:**

After any transaction is performed, it is necessary that the updating should be reflected in the database without any inconsistency.

**c) Provide efficiency querying based on user requests:**

The major purpose is to generate efficient reports on any user request. This will be done by our query processing system, which should be able to process any combination of queries will be done dynamically at run time depending on the user

**3.1.2 EXTERNAL INTERFACE REQUIREMENTS**

**a) User friendly interface:**

The interface should be developed in such a manner that it is very user friendly, this not only improve interaction but also saves data entry time.

**b) Making well designed forms for capturing data:**

The forms for capturing the data should be well-designed using pop-down menus and drag & drop facilities, which reduce the data entry effort on the part of the user.

**3.1.3 PERFORMANCE REQUIREMENTS**

**a) Security:**

All users are not allowed to access the database. Hence there is a need to check authority of every user. Username and Password validation helps to deny unauthorized access to the system.

There are 2 main types of users who will be using the software

They are:-

1) Admin

2) User

Each user is given the specific rights to access the data in Read only, Read Write, Delete.

**3.2 ERD**

Manages

Maintains

tbl\_Student

Register

Student

tbl\_Staff

Register

Computer Institute Management System

Maintains

tbl\_Course

tbl\_Login

tbl\_Fee

* 1. **DATA FLOW DIAGRAM**

.

Search and delete the Data

Admin

Check User Data

Details

Add Staff Details

Add Course Details

User

Add Student Details

Enter User Detail

User User Profile

Details

Student Student Record

Details

Course Course Details

Details

Staff Staff Record

Details

* 1. **CONTEXT LEVEL DAIGRAM**

Student

4. Course Details

Admin

8.Fees Details

7.Fees Details

5.Staff Details

3.Course Details

2. Student Details

6.Staff Details

1.Login

Staff

* 1. **Methodology adopted, system implementation and details of hardware and software used**

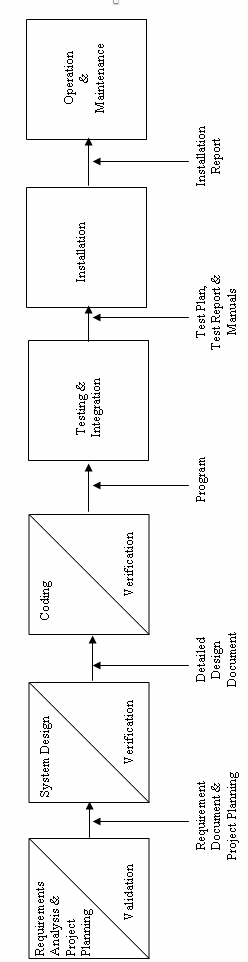
**3.5.1 PROCESS MODEL – SOFTWARE ENGINEERING**

In the development of software we have used **Waterfall Model**, the linear sequential mode. This model encompasses the following activities:

**a) Analysis Phase:**

System Analysis:-

This refers to the gathering of system requirements, with the goal of determining how these requirements will be accommodated in the system.



**b) System Design Phase:**

This is actually a multistep process. In this we tried to focus on some distinct attributes of a program like data structure, software architecture, interface representations and algorithmic detail. In this we tried to translate requirements into representation of the software which can be assessed for quality before coding begins. In the verifications, I have tried to ensure that the design is satisfying the requirements and is of good quality. I have tried to find out if there is any misinterpretation of specified any requirements.

**c) Code Generation Phase:**

In this phase, we translated design of a system into code which can be compiled and executed. In this phase we have done actual coding for all forms. In this we tried to produce simple program which are clear to understanding and modify.

We have used dynamic method to verify the code. We have executed program on some test data and output of the program examined to determine if there are any error present. I have read the code carefully to detect any discrepancies between the design specification and the actual implementation.

**d) Testing:**

Testing plays a critical role in quality assurance for software. Due to limitations of the verification methods for the previous phase, design and requirement faults also appear in the code. Testing is used to detect these errors, in addition to the errors introduced during the coding phase.

# 3.5.2 TOOLS/ENVIRONMENT USED

# SOFTWARE / HARDWARE REQUIREMENTS SPECIFICATION

**PLATFORM:**  Windows XP Professional

**FRONT END:** Visual Studio 2008.

**BACK END:** SQL Server 2005

**HARDWARE REQUIREMENTS:**

Intel Pentium III 733 MHz or Higher.

256 MB RAM or Higher.

**3.5.3 SYSTEM DESIGN**

# DATA STRUCTURE

**1) Table Name: tbl\_Course**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| COURSE\_No | 50 | varchar(50) | Course Number |
| COURSE\_COURSENAME | 50 | varchar(50) | Course Name |
| COURSE\_DURATION | 50 | varchar(50) | Duration |
| COURSE\_FEES | 50 | varchar(50) | Fess |
| COURSE\_BATCHTIMING | 50 | varchar(50) | Batch Timing |
| COURSE\_FACULTY | 50 | varchar(50) | Faculty |

**2) Table Name: tbl\_Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| username | 50 | varchar(50) | Username |
| password | 50 | varchar(50) | Password |

**3) Table Name: tbl\_Staff**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| Staff\_N0 | 50 | varchar(50) | ID |
| Staff\_NAME | 50 | varchar(50) | Name |
| Staff\_DOB |  | smalldatetime | Date of Birth |
| Staff\_QUALIFICATION | 50 | varchar(50) | Qualification |
| Staff\_EXPERIENCE | 50 | varchar(50) | Experience |
| Staff\_EMAILID | 50 | varchar(50) | Email ID |
| Staff\_PHONE | 50 | varchar(50) | Contact Number |
| Staff\_ADDRESS | 50 | varchar(50) | Address |

**4) Table Name: tbl\_Student**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| Student\_ROLLNO | 50 | varchar(50) | Roll Number |
| Student\_NAME | 50 | varchar(50) | Name |
| Student\_ADDRESS | 50 | varchar(50) | Address |
| Student\_PHONE | 50 | varchar(50) | Contact Number |
| Student\_QUALIFICATION | 50 | varchar(50) | Qualification |
| Student\_GENDER | 50 | varchar(50) | Gender |
| Student\_DOB |  | smalldatetime | Date of Birth |
| Student\_EMAIL | 50 | varchar(50) | Email ID |
| Student\_OCCUPATION | 50 | varchar(50) | Occupation |
| Student\_KNOWLEDGE | 50 | varchar(50) | Konowledge |
| Student\_TOTALFEES | 50 | varchar(50) | Total fees |

**5) Table Name: tbl\_Fee**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| Fee\_ROLLNO | 50 | varchar(50) | Roll Number |
| Fee\_RECEIPTNO | 50 | varchar(50) | Receipt Number |
| Fee\_Date |  | smalldatetime | Date |
| Fee\_AMOUNT | 50 | varchar(50) | Amount |
| Fee\_ModeOfPayment | 50 | varchar(50) | Mode of Payment |
| Fee\_BANKNAME | 50 | varchar(50) | Bank Name |
| Fee\_CHEQUENO | 50 | varchar(50) | Cheque Number |
| Fee\_BAMOUNT | 50 | varchar(50) | Balance Amount |
| Fee\_BasicFees | 50 | varchar(50) | Basic Fees |
| Fee\_DiscountPercentage | 50 | varchar(50) | Discount Percentage |
| Fee\_DiscountAmount | 50 | varchar(50) | Discount Amount |
| Fee\_TotalFees | 50 | varchar(50) | Total Fees |

**6) Table Name: Purchases**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| Serial\_No | 50 | varchar(50) | Serial No |
| Name | 50 | varchar(50) | Item name |
| Amount | 50 | varchar(50) | Amount |

**3.5.4 TESTING**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, coding.

**Testing objectives**

1. Testing is a process of executing a program with the intent of finding an error.

2. A good test case is the one that has high portability of finding an as-yet undiscovered error.

3. A successful test is one that uncovers an as-yet discovered error.

The main objective here will be to design test cases to uncover different classes of errors and to do so with minimum amount of time and efforts. If testing is conducted successfully it will uncover errors in the software. Another advantage is that it demonstrates that software functions appear to be working according to the specifications and performance requirement have been met.

But Testing cannot show the absence of defects it can show only that software errors are present.

**Strategies used for software testing**

The software engineering process is viewed as spiral as shown:

**Unit testing** begins at the vortex of the spiral and concentrates at the each unit of the software as implemented in the source code. Unit testing assures each module tested individually functions properly as a unit.

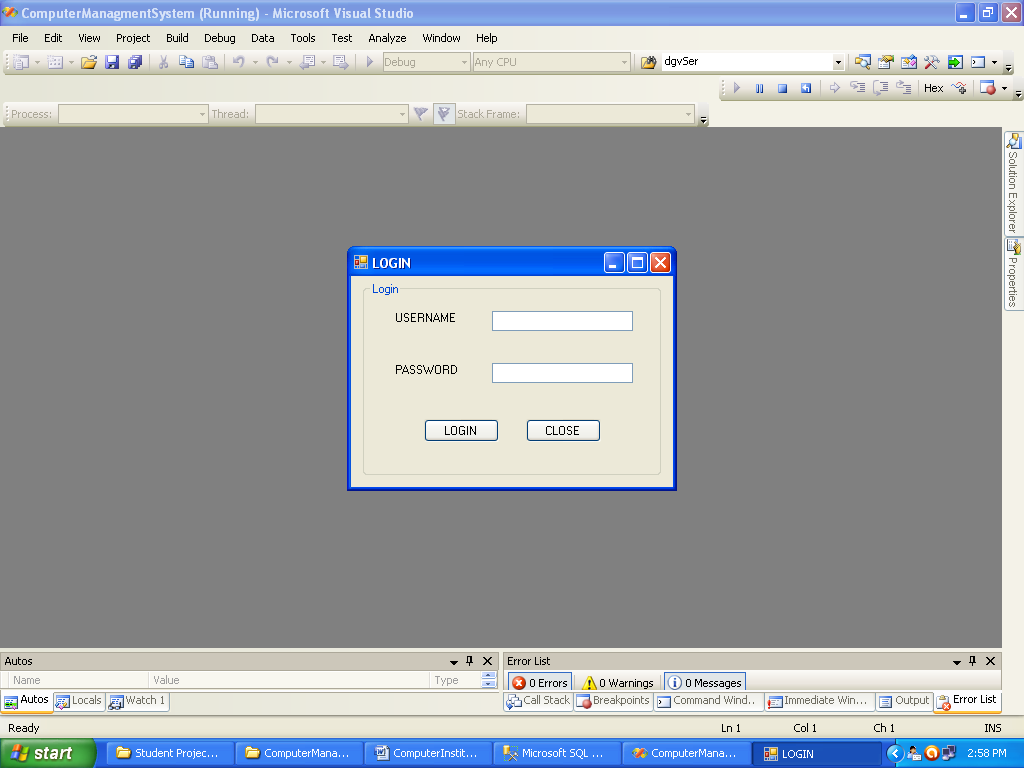
**Integration testing** focuses on the design and construction of the software architecture. It is a systematic technique for constructing a program structure while conducting tests to uncover errors associated with interfacing. The objective is to take unit-tested module and build a program structure that has been specified as design.

**Validation testing** takes care of the requirements established as part of the software requirements analysis are validated against the software that has been constructed. It is said to be successful when the software functions in a manner that can be reasonably expected by the customer.

**System testing** tests software and other system elements as a whole. These tests fall outside the scope of software engineering process and are not conducted solely by the software developer.

**3.5.5 DATA MODULES AND THEIR DESCRIPTION**

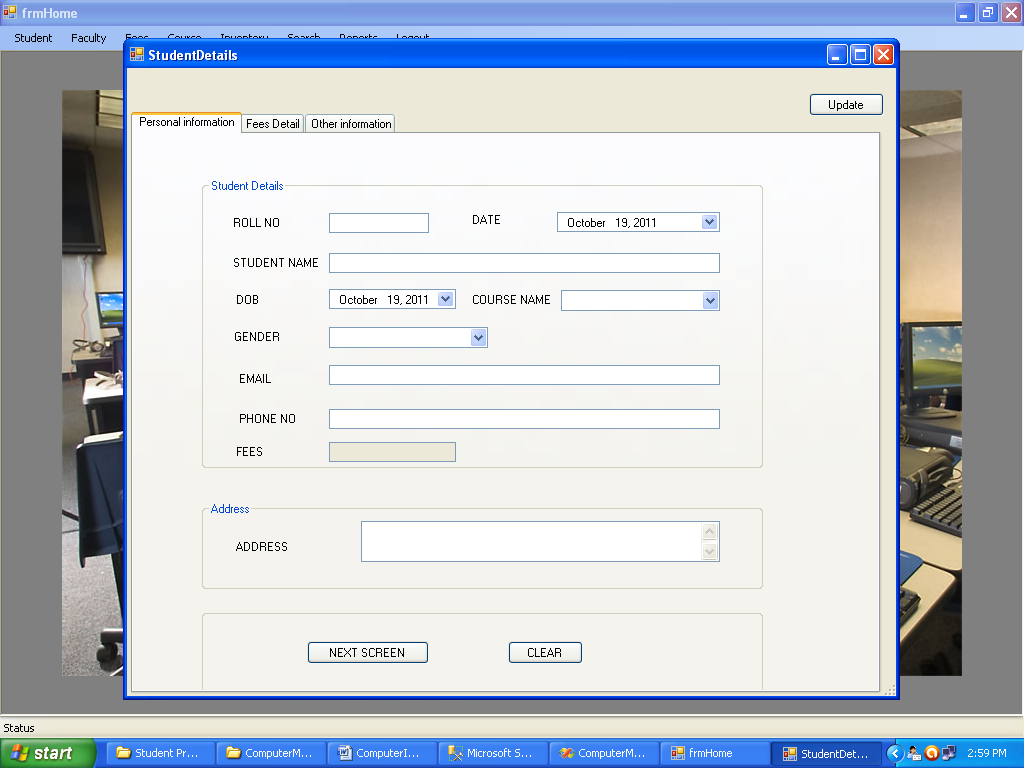
**1) Login**



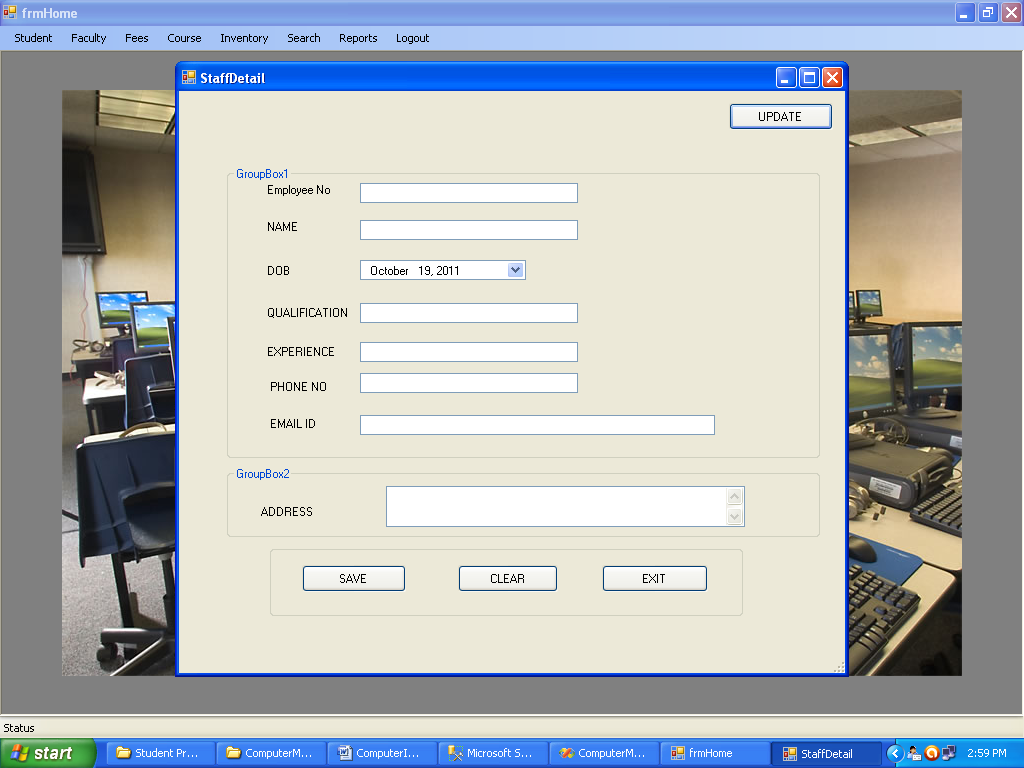
**2) Home**

****

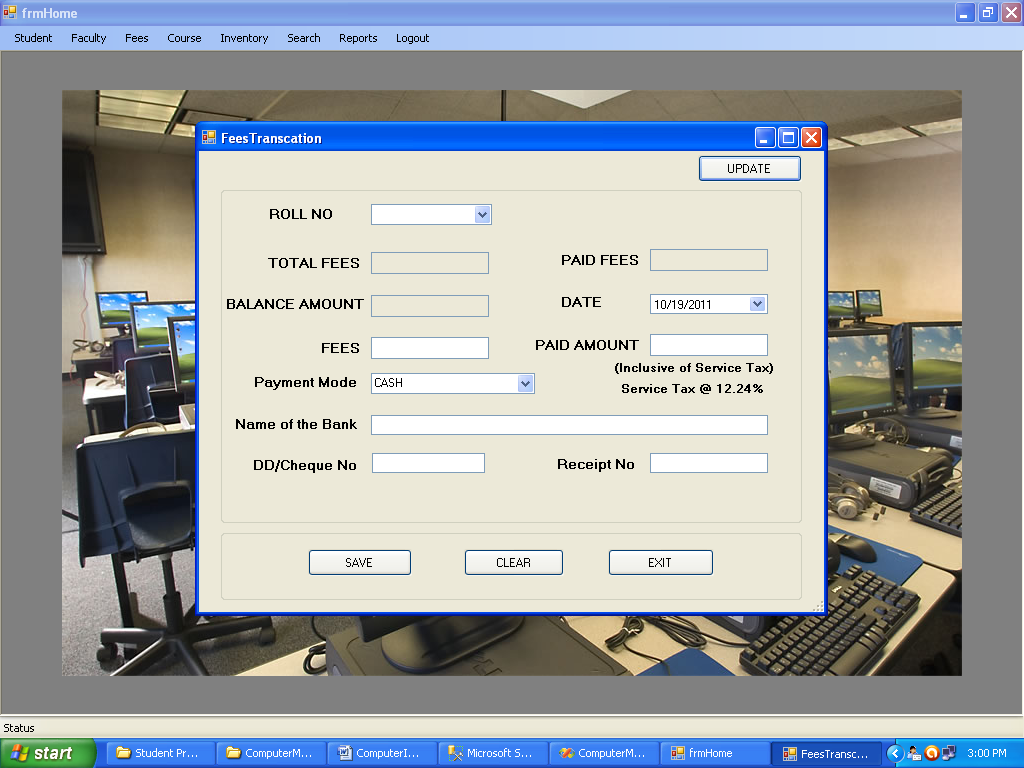
**3) Student**

****

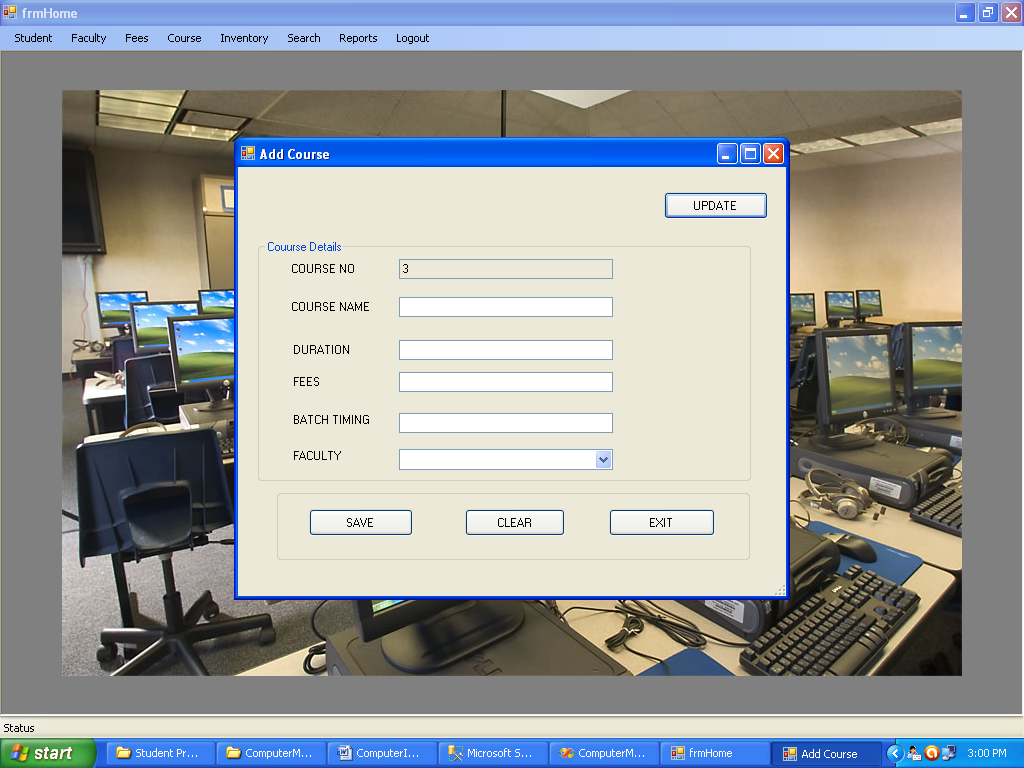
**4) Faculty**

****

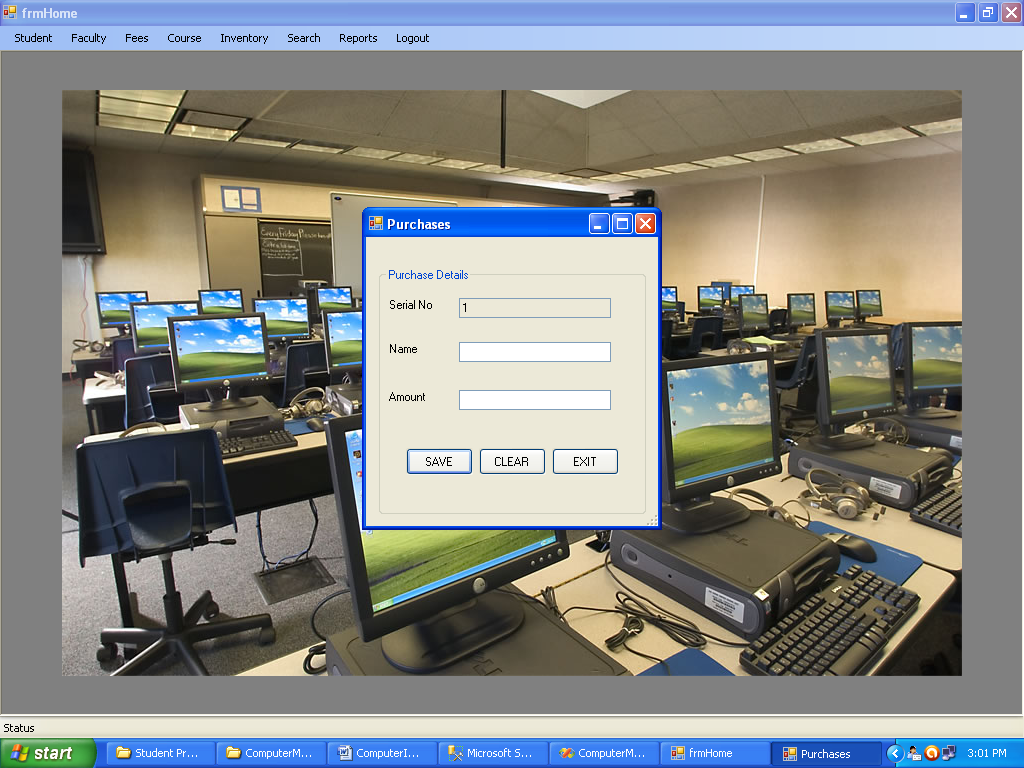
**5) Fees**

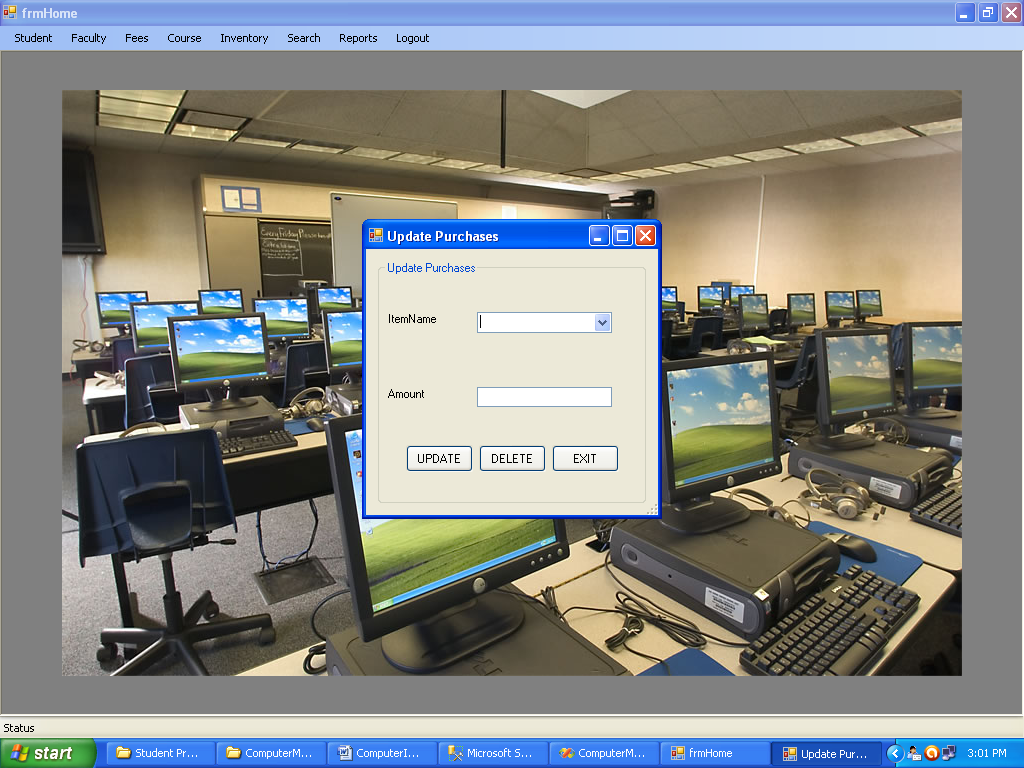
****

**6) Course**

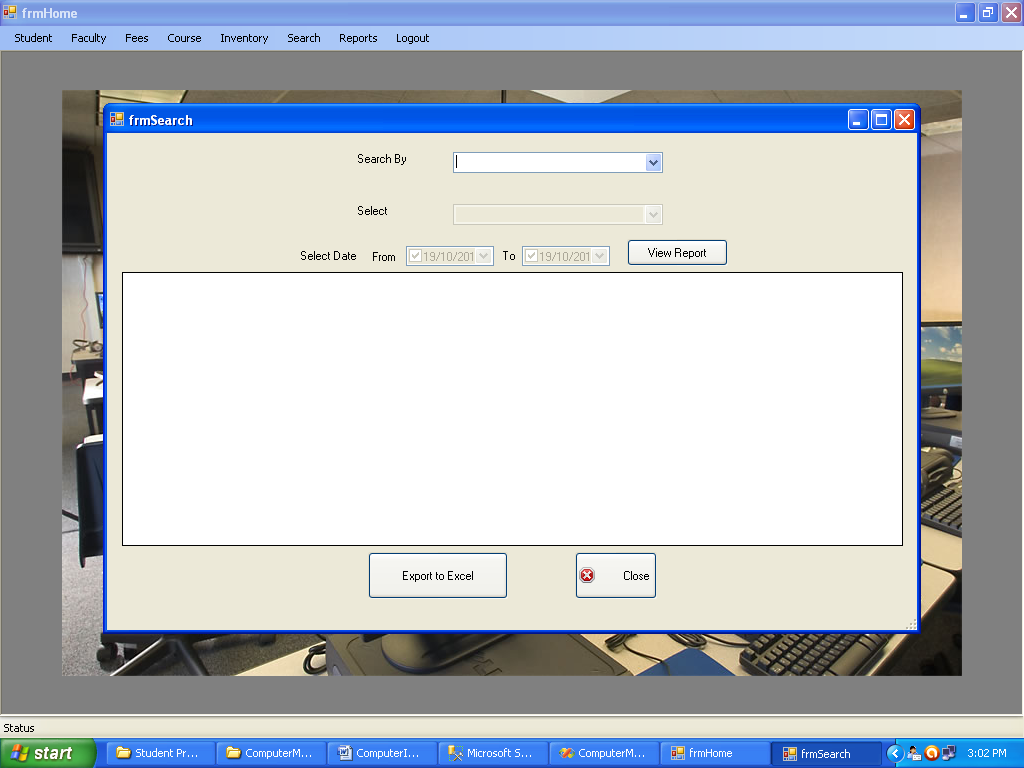
****

**7) Inventory**

****

****

**8) Search**

****

**9) Reports**

****