

Diss  
COM  
1773

# DEPARTMENT INFORMATION SYSTEM



DEVELOPED  
BY

NOSHEEN FAROOQ  
MARIA MUNIR

DEPARTMENT OF COMPUTER CENTRE  
QUAID-I-AZAM UNIVERSITY  
ISLAMABAD, PAKISTAN  
YEAR 2007

**DEPARTMENT INFORMATION  
SYSTEM**

**DEVELOPED  
BY**

**NOSHEEN FAROOQ  
MARIA MUNIR**

**Supervised  
By  
MS SABA AKHTER**

**DEPARTMENT OF COMPUTER CENTRE  
QUAID-I-AZAM UNIVERSITY**

**ISLAMABAD, PAKISTAN**

**YEAR 2007**

## DECLARATION

I hereby declare that this web site, neither as a whole nor as a part there of has been Copied out from any source. It is further declared that we developed this web site and this report entirely on the basis of our personal efforts made the sincere guidance of our project supervisor. If any part of this web site to be copied of found to be a report of some other I shall standby the consequences.

No portion of the work presented in this report has been submitted in support of any application for any other degree of qualification of this or any other University or institute of learning. We further declare that this web site and all associated documents, reports, and records are submitted as partial requirement for the Diploma PGD-IT.

We understand and transfer copyrights for these materials to Quaid-I-Azam University, Islamabad.

We shall not sale this web site and documents and shall not get any financial gains from these.

**NOSHEEN FAROOQ**

  
SIGNATURE

**MARIA MUNIR**

  
SIGNATURE

**PROJECT SUPERVISOR**

  
SIGNATURE

## FINAL APPROVAL

This is to certify that we have read the project report submitted by the following students

**NOSHEEN FAROOQ**

**MARIA MUNIR**

It is our judgment that this report is of sufficient standard to warrant its expectance by Quaid-i-Azam University Islamabad, for the Diploma of PGD-IT.

**Examination Committee:**

**Internal Supervisor**

\_\_\_\_\_  
**SABA AKHTAR**

**Internal Examiner**

\_\_\_\_\_

**External Examiner**

\_\_\_\_\_

## ACKNOWLEDGMENT

All praises for **ALLAH**, The Creator of the Universe, who guided mankind to path of understanding, knowledge, reasoning, and research through his beloved **Holy Prophet (PBUH)**. First of all we would like to extend our sincere and humble gratitude to Almighty Allah who endowed us potential and ability to make solid contribution to the already existing oceans of knowledge.

We would like to express our profound regards to our project supervisor **Ms SABA AKHTAR** who provided us her cooperation, guidance and valuable support during the all phases of the project. The purpose of this acknowledgement will not to serve if we don't mention here the cooperation, assistance and guidance extended by our respectable teachers whose suggestions and criticism enabled us to complete this project.

We should love to acknowledge the sincere cooperation and moral support of all my friends in the university.

In the end we should like to acknowledge the support of our family members and it would not have been possible without the prayers and support of our parents.

**NOSHEEN FAROOQ**

**MARIA MUNIR**

## DEDICATION

We dedicate this project to our

Parents and our respectable Teachers

Who helped us in the preparation of this project

Because without their moral support and encouragement

We were unable to achieve our aims

## PROJECT IN BRIEF

<b>Project Title</b>	Department Information System
<b>Developed By:</b>	Nosheen Farooq Maria Munir
<b>Supervised By:</b>	Ms Saba Akhtar
<b>Tools Used:</b>	Dream weaver, HTML, PHP, MySQL, Flash MX, Adobe Photoshop, Microsoft Visio
<b>Operating System:</b>	Windows XP server pack 2.2
<b>System Used :</b>	Pentium III (PRO)

# CONTENTS

## 1. INTRODUCTION

1.1	PROJECT DEFINITION .....	02
1.2	SCOPE OF THSYSTEM.....	02
1.3	OBJECTIVES OF THE PROJECT.....	03

## 2. SYSTEM PLANING

1.4	INTRODUCTION .....	05
1.5	EXISTINSYSTEM.....	05
1.6	PROPOSEDSYSTEM.....	05

## 3. REQUIREMENT ANALYSIS

3.1	FUNCTIONAL REQUIREMENTS .....	07
3.2	NON FUNCTIONAL REQUIREMENTS .....	07
3.2.1	USABILITY .....	08
3.2.2	PERFORMANCE .....	08
3.2.3	RELIABILITY .....	09
3.2.4	SECURITY.....	09
3.2.5	HARDWARE.....	09
3.2.6	DEPLOYMENT.....	09
3.3	SYSTEM ATTRIBUTES .....	10
3.4	USE CASES .....	13

## 4. SYSTEM ANALYSIS

4.1	INTRODUCTION .....	19
4.2	ANALYSIS DIAGRAM .....	20
4.3	SEQUENCE DIAGRAM .....	24



## 5. SYSTEM DESIGN

5.1	INTRODUCTION .....	29
5.2	PRESENTATION LOGIC .....	29
5.2.1	INTERFACE DESIGN .....	29
1	THE ELEMENTS OF INTERFACE DESIGN .....	30.
5.3	INPUT DESIGN .....	32
5.4	OUTPUT DESIGN .....	32
5.5	DATABASE PLANNING .....	33
5.5.1	SYSTEM DEFINITION.....	33
2	MISSION STATEMENT .....	33
3	SYSTEM BOUNDARY .....	34
4	MAJOR USER VIEWS OF SYSTEM.....	34
5.6	REQUIREMENTS COLLECTION AND ANALYSIS .....	35
5.7	ENTITY RELATIONSHIP DIAGRAM .....	36
5.8	DATABASE DESIGN.....	37
5.8.1	ENTITIES .....	37
5.8.2	ENTITIES ALONG WITH ATTRIBUTES.....	38
5.8.3	REFRENTIAL INTEGRITY CONSTRAINTS .....	42

## 6. IMPLEMENTATION

6.1	INTRODUCTION .....	46
6.2	TECHNOLOGY SELECTION.....	46
6.2.1	PHP.....	46
6.2.2	MY SQL .....	47
6.2.3	JAVASCRIPT .....	47
6.3	DEVELOPMENT IDE TOOLS .....	47
6.3.1	MACROMEDIA DREAMWEAVE .....	48
6.3.2	MACROMEDIA FIREWORKS .....	48
6.4	HARDWARE REQUIREMENTS OF SYSTEM .....	49
6.5	SOFTWARE REQUIREMENTS OF SYSTEM .....	50

**7. SYSTEM TESTING**

5.3 INTRODUCTION .....52  
5.4 TESTING CHASSES .....53  
5.5 SYSTEM EVALUATION .....54  
5.6 FUTURE ENHANCEMENT .....54

**8. SYSTEM CODING**

5.7 INTRODUCTION .....56  
5.8 CODING .....56

**WEBLIOGRAPHY**

**BIBLIOGRAPHY**

**REFERENCES**

APENDIX A	WEB APPLICATION EXTENSION FOR UML
APENDIX B	FEASIBILITY STUDY
APENDIX C	RISK MANAGEMENT
APENDIX D	USER GUIDE
APENDIX E	PROJECT PLAN

# **CHAPTER 1**

## **INTRODUCTION**

- **PROJECT DEFINITION**
- **SCOPE OF THE SYSTEM**
- **OBJECTIVES OF THE PROJECT**

## **INTRODUCTION**

The internet has become a vast source of information for business people as well as students as they discover the advantages of getting online. It gives professional technologically advanced image. That's why we have chosen a project of developing a department information system. In order to cope with the international level we also need to automize our system to work more efficiently and timely.

### **1.1 PROJECT DEFINITION**

The purpose of the project is to maintain the examination data that a department gets and generate for the students and teacher and to facilitate the cooperation and sharing of information between staff, student and teacher.

### **1.2 SCOPE OF THE SYSTEM**

This proposed system will be a web application which will have different features according to the requirements of the university. The scope of the system includes

- 1 Student personal record information
- 2 Students marks detail record
- 3 Department detail
- 4 Course detail
- 5 Student course registration
- 6 Online result display
- 7 View student profile and result status semester wise
- 8 View result status of student.
- 9 Student result submission and approval.
- 10 Student profile for teachers to view

### **1.3 OBJECTIVES OF THE PROJECT**

The objectives of the project are to facilitate students, teachers and staff members of the department. The main objectives of this project are

1. To login an existing accept into the system.
2. To enable department staff to add, delete, modify, view and print student personal record.
3. To enable department staff to register students courses semester wise.
4. To enable department staff to add, delete, modify, view and print student results.
5. To enable teachers to add, view and print student results.
6. To enable students to view and print their results.
7. To display a list of currently available courses.
8. To provide all above mentioned facilities in a user friendly manner.

## **CHAPTER 2**

### **SYSTEM PLANING**

- **INTRODUCTION**
- **EXISTING SYSTEM**
- **PROPOSED SYSTEM**

## **2.1 INTRODUCTION**

The purpose of the project is to maintain the data that a department gets and generate for the students and teacher and to facilitate the cooperation and sharing of information between staff, student and teacher.

## **2.2 EXISTING SYSTEM**

The existing department system is out dated and old one. It is highly dependent on the examination department to get the students result information. There is no such system which provides them direct access to the student's results and date sheet. Students also suffer from this old system. They have to move to different department to get different information. It waste their time and money and create tension sometimes as well for them.

## **2.3 PROPOSED SYSTEM**

This project will cover all the shortcomings of the existing system which is totally out dated. This project will make the department information system more compatible and feasible for every one. This is more dynamic due to the availability of on line information facility. This project also has all the information related to the examination department. All the members of the department can get all required information any time they want. That information would be accurate and updated. All students can get their academic and official records and their on line results and date sheets. All information about available courses for the newcomers will also be available.

## **CHAPTER 3**

### **REQUIREMENT ANALYSIS**

- **FUNCTIONAL REQUIREMENTS**
- **NON FUNCTIONAL REQUIREMENTS**
- **SYSTEM ATTRIBUTES**
- **PROJECT BLOCK DIAGRAM**
- **USE CASES**



### 3. REQUIREMENT ANALYSIS

Requirement analysis normally partitioned into

- 1 Functional requirements
- 2 Non functional requirements

#### 3.1 FUNCTIONAL REQUIREMENTS

Functional requirements are associated specific functions, tasks or behaviors the system must support. A functional requirements statement is a useful skeleton upon which to construct a complete requirements statement [PRS97]. In this proposed project functional requirements are:

1. Student registration process that how students can register themselves.
2. Course registration system in which administration can register students in any selected course.
3. Students access to view courses, results and news.
4. Administration can add, delete, modify, view and print student personal records.
5. Teachers access to add, view and print student results.
6. Access to view currently available courses.

#### 3.2 NON FUNCTIONAL REQUIREMENTS

Non functional requirements are constraints on various attributes of these function or tasks. Non functional requirements are given in terms of constraints on the result of tasks which are given as functional requirements. It includes:

1. Usability
2. Performance
3. Reliability
4. Security
5. Hardware
6. deployment

### **3.2.1 USABILITY**

A usability requirement refers to those general aspects of the interface between the user & the system. It can be measured objectively via performance errors and productivity, and subjectively via users preferences and interface characteristics.

### **3.2.2 PERFORMANCE**

Performance requirements describe system execution performance & are usually related to time. A common requirement for web application is to specify a maximum load time for a page.

Performance standards are:

- 1 Based on the position, not the individual
- 2 Observable, specific indicators of success
- 3 Meaningful, reasonable and attainable
- 4 Describe "fully satisfactory" performance once trained
- 5 Expressed in terms of Quantity, Quality, Timeliness, Cost, Safety, or Outcomes

### **3.2.3 RELIABILITY**

The extent to which a measurement instrument yields consistent, stable, and uniform results over repeated observations or measurements under the same conditions each time. For example, a scale is unreliable if it weighs a child three times in three minutes and gets three different weights.

### **3.2.4 SECURITY**

Security requirements should also include access to the system by other “external” systems, if used. A typical security requirement might be stated, as “the system will be ensuring that all confidential information provided by customer via Internet shall be encrypted with at least SSL version 3 or later”.

### **3.2.5 Hardware:**

Hardware requirement often state the minimum hardware required to implement the system. We need a 40 GB hard disk, 1.8 GHZ processor and 128 MB Ram.

### **3.2.6 Deployment:**

Deployment requirements describe how the application is delivered to the end users. It provides constraints on how the system is to be installed, maintained, & accessed by maintenance staff. A deployment requirement for a web application may require all client software to be downloading & installed from the browser and not required the client to reboot or to perform a manual setup.

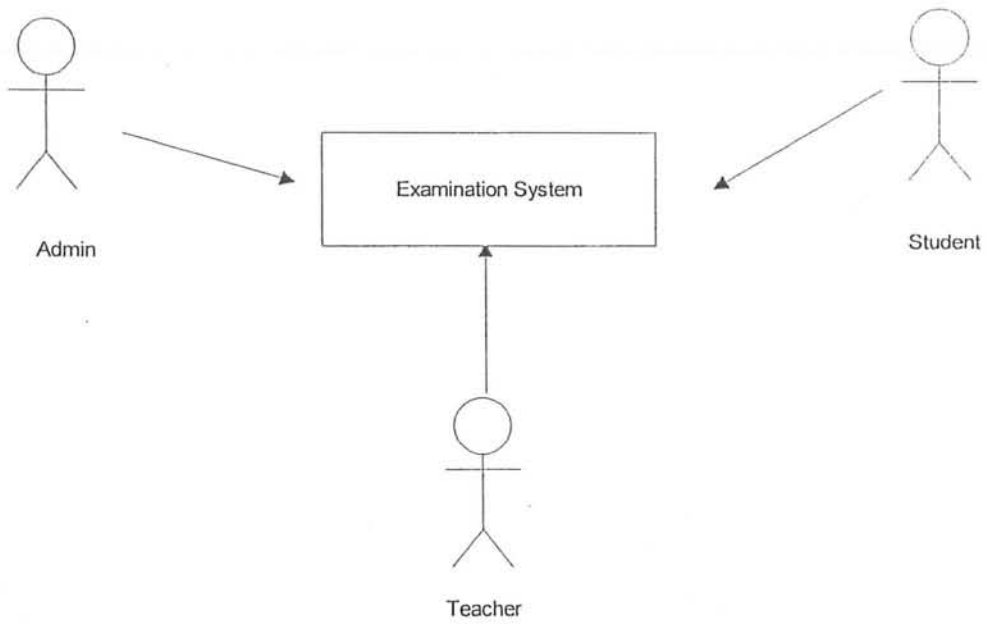
### 3.3 SYSTEM ATTRIBUTES:

System functions are nonfunctional requirements or that a product must fulfill

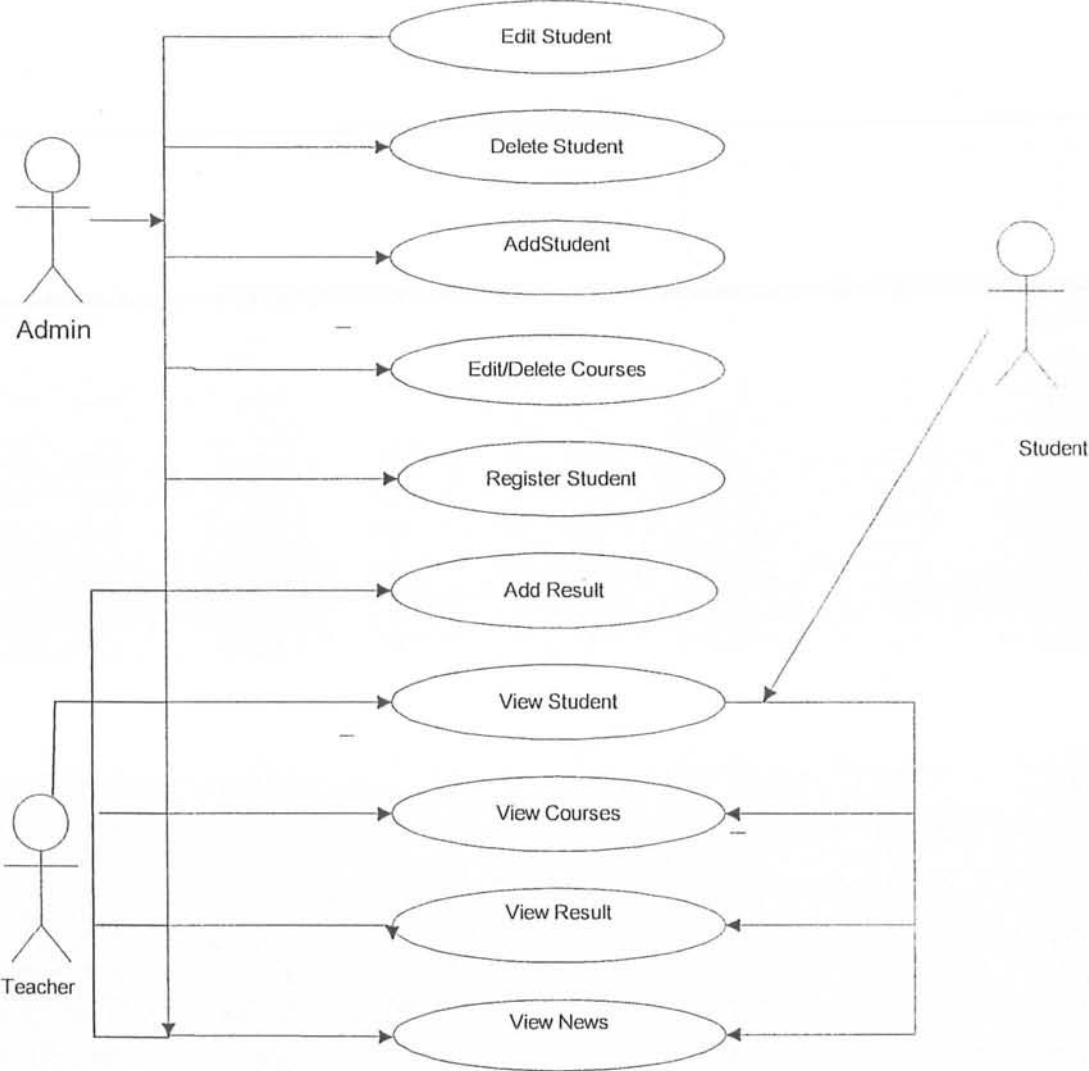
Following are non-function requirements for our system:

Ref#	Functions	Attributes	Details	Cat
2.4.1	User interface	Level of Expertise	Novice	Must
		Interface	From based, Colorful	Must Want
2.4.2	Documents	Deliverable	User manual Reports	Must
		Technical Documentation	Source code Configuration manual	Must Want
2.4.3	Hardware	Hard disk	40 GB	Want
		Processor	1.8 GHz	Want
		RAM	128 MB	Want
2.4.4	Performance	Response time	Web pages should not take Longer than 15 second to load	Must
2.4.5	Correctness	Error handing	Validity checks	Must
2.4.6	Robustness	Invalid input		
	Reliability	Availability		Want
2.4.7	Modification	Future changes	Availability to add future Changing facility	Want
2.4.8	Physical environment	Deployment	Anywhere	Must
2.4.9	Security	Confidentiality	The system will ensure all confidential information provided by users via the internet shall be encrypted.	Must
		Authentication	Authenticated access given to system shall be provided.	

PROJECT BLOCK DIAGRAM



USE CASE DIAGRAM



## USE CASES

### USE CASE 1: LOGIN SYSTEM

Actors:	Teachers, Administrator
Purpose:	To login to the system.
Overview:	The user can login into the system.
Type:	Primary and essential.

#### Typical Course of events

Actor action	System Response
1. User enter the Login name and password to login to the system	2. System accepts the user request and authenticate if password and login name are correct.
	3. If user is a teacher then it will take the user to the main screen for teacher otherwise display message.
	4. If the user is an administrator then system will take the user to the main menu for admin otherwise display message

### USE CASE 2: STUDENT REGISTRATION

Actors:	Administrator
Purpose:	To register the student to the examination system.
Overview:	The user can do registration.
Type:	Primary and essential.

**Typical Course of events**

<b>Actor action</b>	<b>System response</b>
1. User will select the link for student registration from menu	2. System will display the student registration form on the screen
3. User will fill the required information on form and submit the form	4. System accepts that information and add the required information into the database
	5. Information is successfully added into the database and user is shown the entered information on the screen
	6. System will provide the option to register new student or go back to main menu

**USE CASE 3: EDIT STUDENT INFORMATION**

Actors:	Administrator
Purpose:	To edit student information.
Overview:	The user can edit student information from links.
Type:	Primary and essential.

**Typical Course of events**

<b>Actor Action</b>	<b>System Response</b>
1. User will select the edit/delete option from the menu.	2. System will display a form to enter the student registration id of student to be edited
3. User will enter the required information and submit the form to system.	4. System will accept the information and display the existing profile of that particular student.
5. User will do required changes into the profile and submit the information to the system.	6. System will accept and update the required changes into the database.

**USE CASE 4: VIEW STUDENT PROFILE**

Actors:	Students, Teachers, Administrator
Purpose:	To view the student profile.
Overview:	The user can select the student to view its profile from links.
Type:	Primary and essential.



**Typical Course of events**

<b>Actor Action</b>	<b>System Response</b>
1. User enters the registration number of the student to view student profile and submit it to the system.	2. System will display the profile of the particular student.

**USE CASE 5: REGISTER COURSES SEMESTER WISE**

Actors:	Administrator
Purpose:	To register courses semester wise.
Overview:	The user can register new courses from links.
Type:	Primary and essential.

**Typical Course of events**

<b>Actor Action</b>	<b>System Response</b>
1. User will select the link to register courses from main menu	2. System will prompt for the student registration Id, semester to be entered
3. User will enter the registration number and Semester of the student and submit that to the system.	4. System will display a list of courses along with credit hours to be selected by the user
5. User will select courses and credit hours for that particular semester and submit to the system.	6. System checks the validity criteria of that particular semester and adds the courses into the database.

**USE CASE 6: EDIT/DELETE COURSES SEMESTER WISE**

Actors:	Administrator
Purpose:	To edit/delete course semester wise.
Overview:	The user can edit/del course semester wise..
Type:	Primary and essential.

Typical Course of events

Actor Action	System Response
1. User will select the edit/delete option from the main menu	2. System will display a form to enter the student registration id and semester of student whose courses to be edited or deleted.
3. User will enter the required information and submit the form to system.	4. System will accept the information and display the existing courses of that particular student.
5. User will do required changes into the course and submit the information to the system	6. System will accept and update the required changes into the database.

**USE CASE 7: ADD STUDENT MARKS SEMESTER/SUBJECT WISE**

Actors:	Teachers, Administrator
Purpose:	To add students marks to the system.
Overview:	The user can add student marks from links.
Type:	Primary and essential.

Typical Course of events

Actor Action	System Response
1. User will select the Add Student Marks from the main menu	2. System will display a form with 2 options from user. First to enter student Reg_no and semester of a particular student for semester wise entry. Second to enter course id and semester for course wise entry.
3. User will enter the registration ID, semester and course id of that particular student.	4. System will display student mark sheet.
5. If the user is admin then it can enter student marks and submit that new information to the system.	6. System will update that required information.

**USE CASE 8: VIEW STUDENT RESULT SEMESTER/SUBJECT WISE**

Actors:	Students, Teachers, Administrator
Purpose:	To view the students results.
Overview:	The user can view results from links.
Type:	Primary and essential.

**Typical Course of events**

Actor Action	System response
1. User can enter the student registration ID number and semester to view the result.	2. System will check the registration ID number and will display the required information to the student.
3. User can enter the student registration ID number, semester and subject code to view the result.	4. System will check the registration ID number and will display the required information to the student

**USE CASE 9: VIEW SEMESTER DETAILS**

Actors:	Students, Teachers, Administrator
Purpose:	To view the semester details from system.
Overview:	The user can view semester details from links.
Type:	Primary and essential.

**Typical Course of events**

Actor Action	System response
1. After login to the system user can request system to view semester detail.	2. System login the user to the system.
3. User can see start and end date of semester.	4. System will display the date information.
5. User can request to view fee charges for each course.	6. System will display fee information for each course.
7. User can ask system to show offered courses.	8. Systems accept the request and display the required information to the user.

## **CHAPTER 4**

### **SYSTEM ANALYSIS**

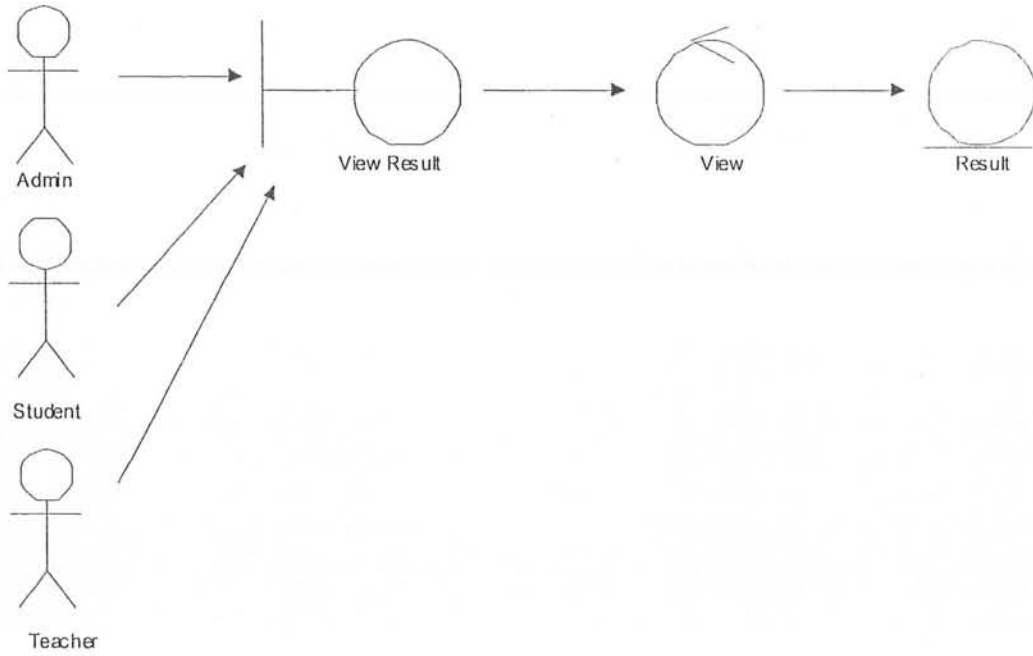
- **INTRODUCTION**
- **ANALYSIS DIAGRAM**
- **SEQUENCE DIAGRAM**

## **Introduction**

Assuming that a new system is to be developed, the next phase is system **analysis**. Analysis involved a detailed study of the current system, leading to specifications of a new system. Use case analysis diagram is another activity that is done when the use case near completion. It identifies the classes and objects that will perform a use case's flow of events. A sequence diagram, a specific type of interaction diagram, expresses the interaction between an actor and the system, with a special emphasis on the time line.

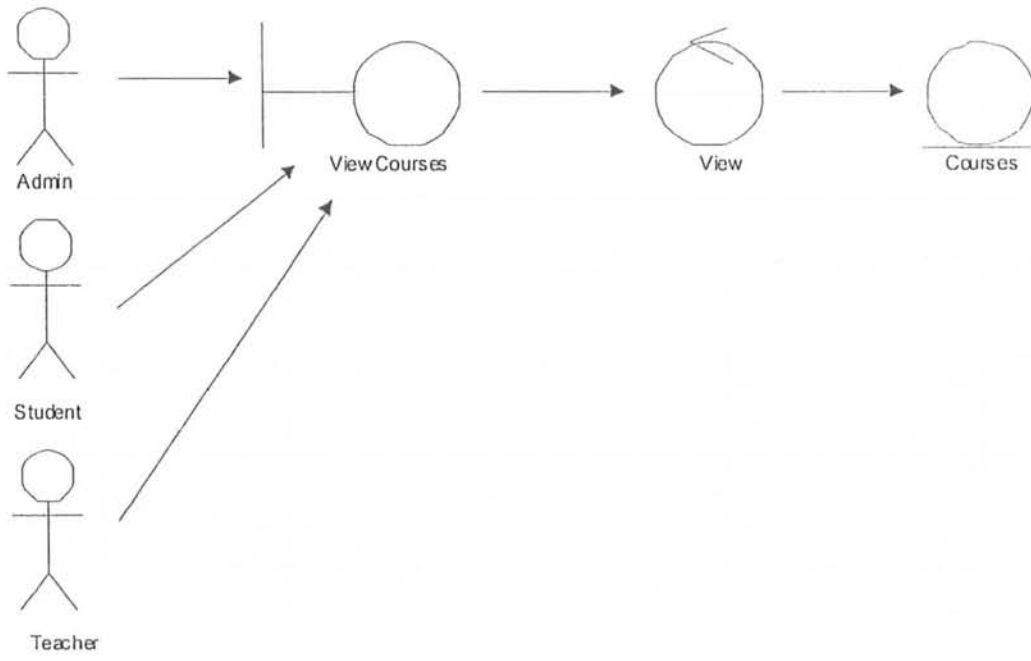
DEPARTMENT INFORMATION SYSTEM

ANALYSIS DIAGRAM

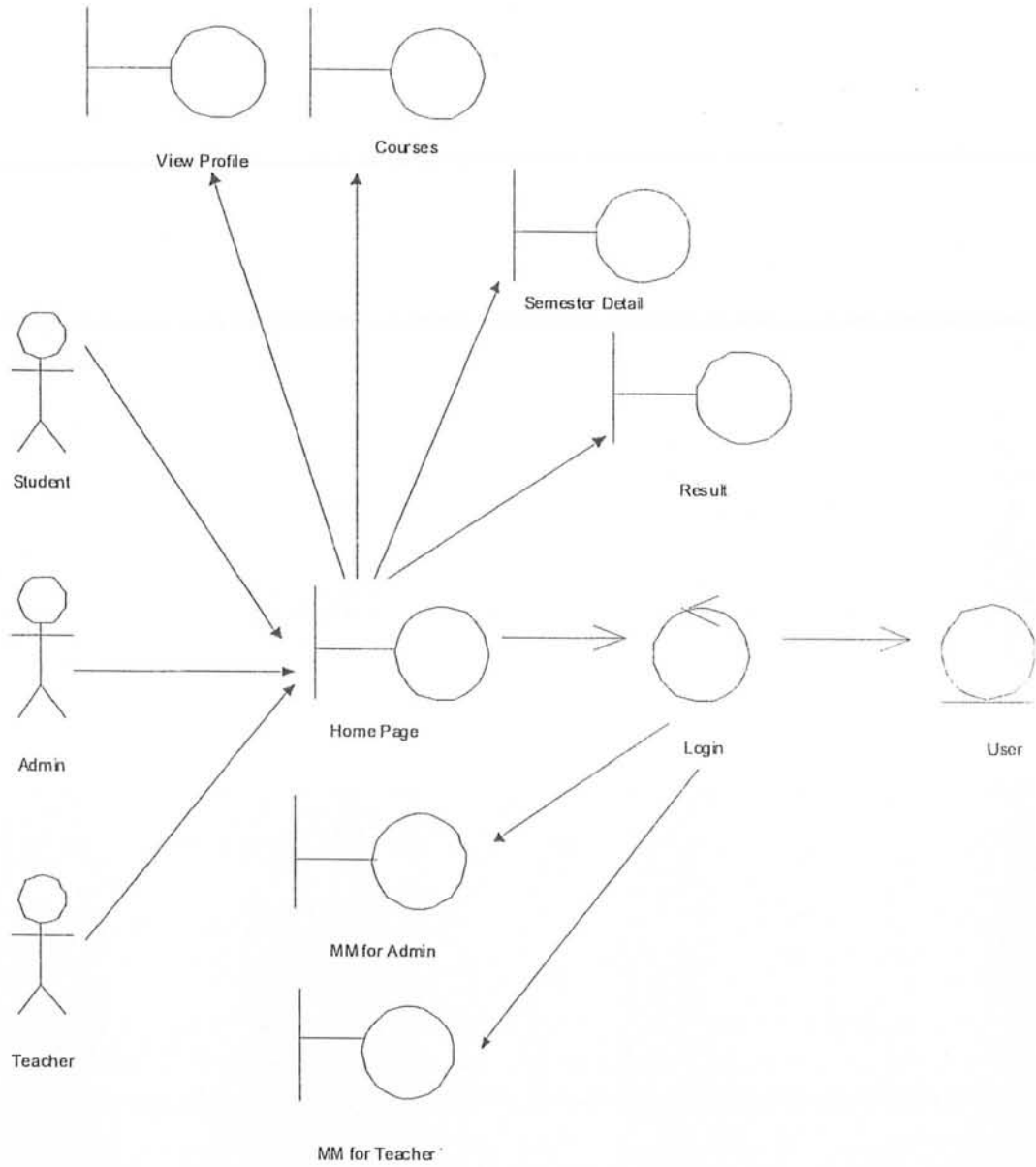


DEPARTMENT INFORMATION SYSTEM

ANALYSIS DIAGRAM



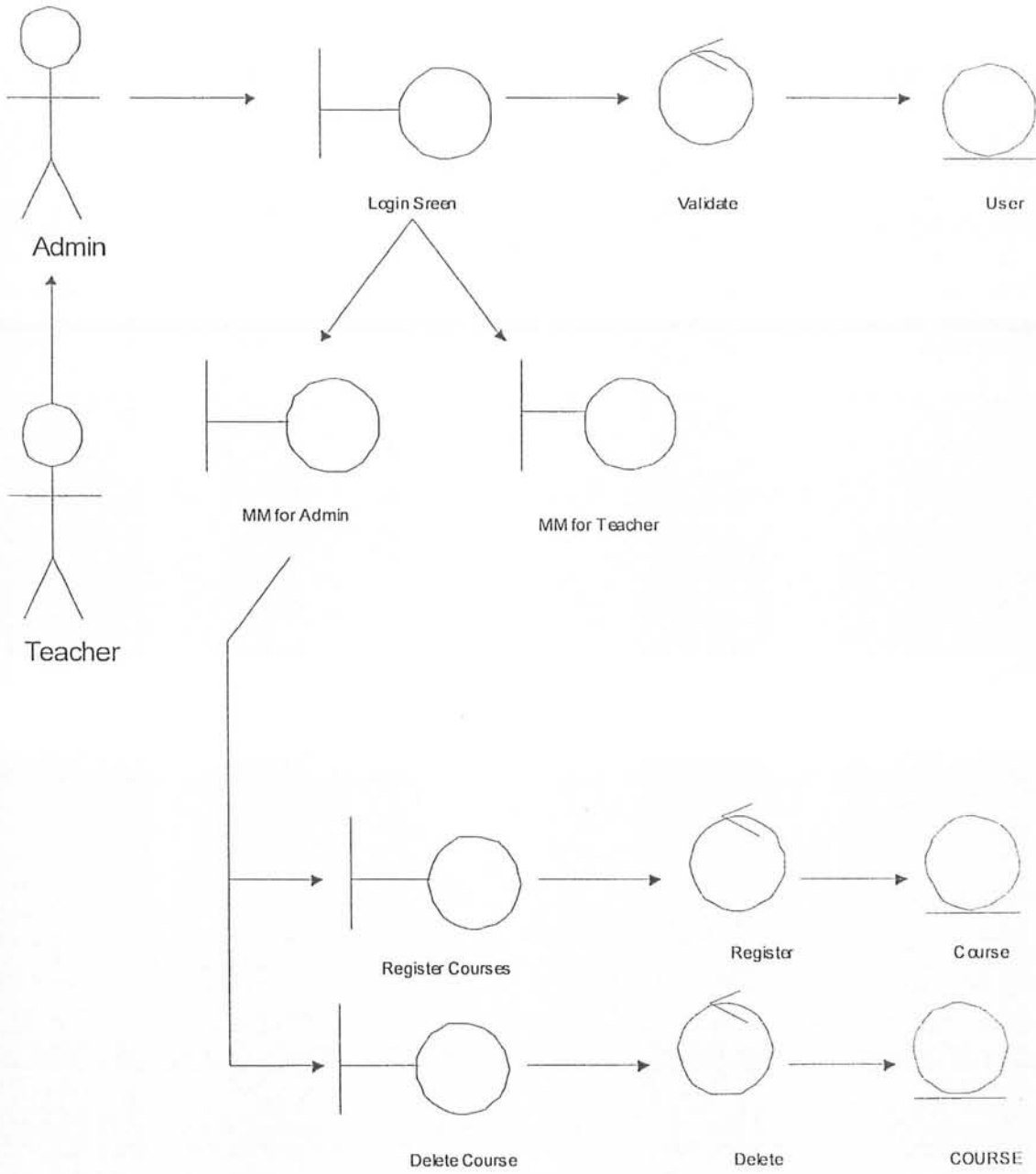
ANALYSIS DIAGRAM





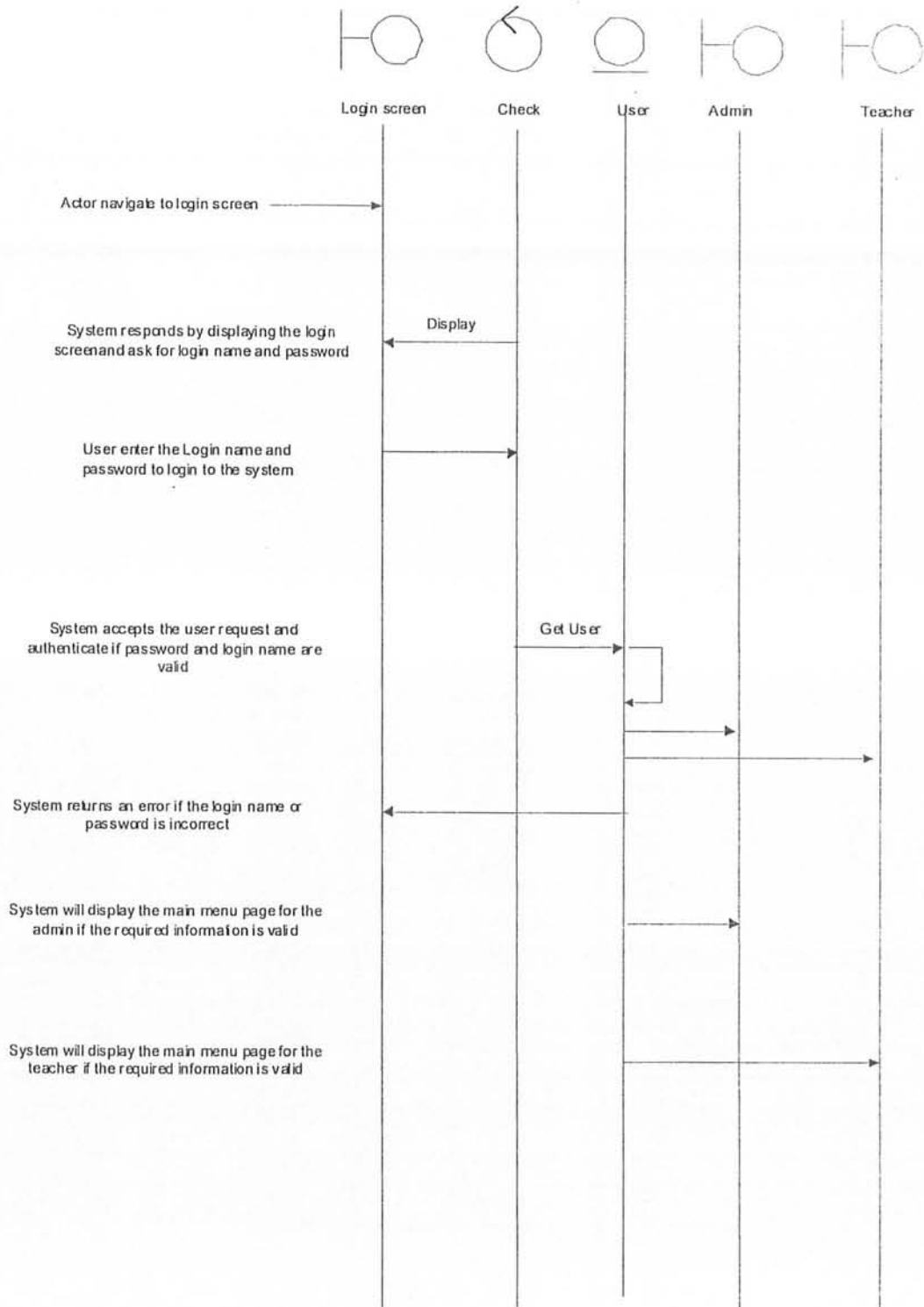
DEPARTMENT INFORMATION SYSTEM

ANALYSIS DIAGRAM



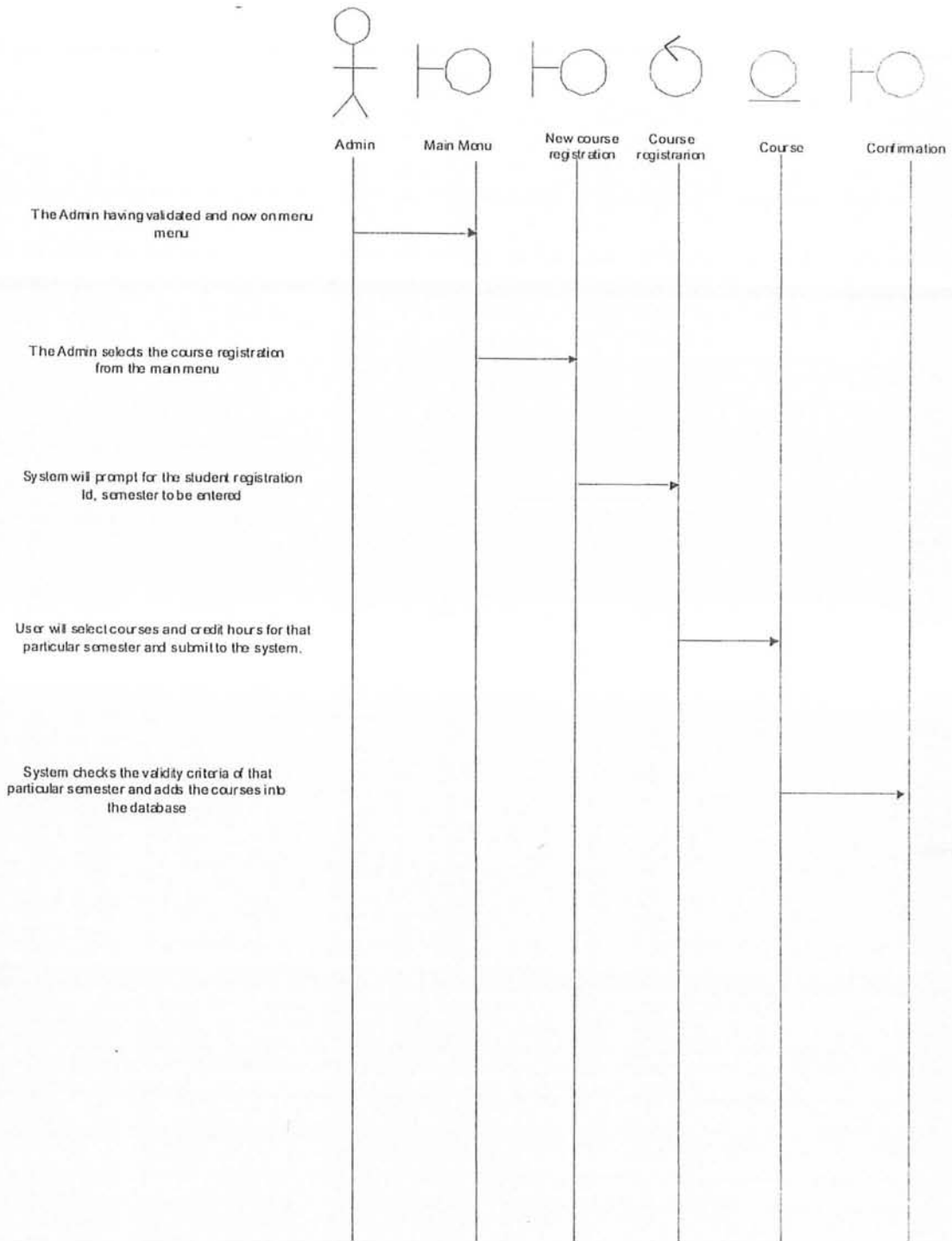
DEPARTMENT INFORMATION SYSTEM

SEQUENCE DIAGRAM



# DEPARTMENT INFORMATION SYSTEM

## SEQUENCE DIAGRAM



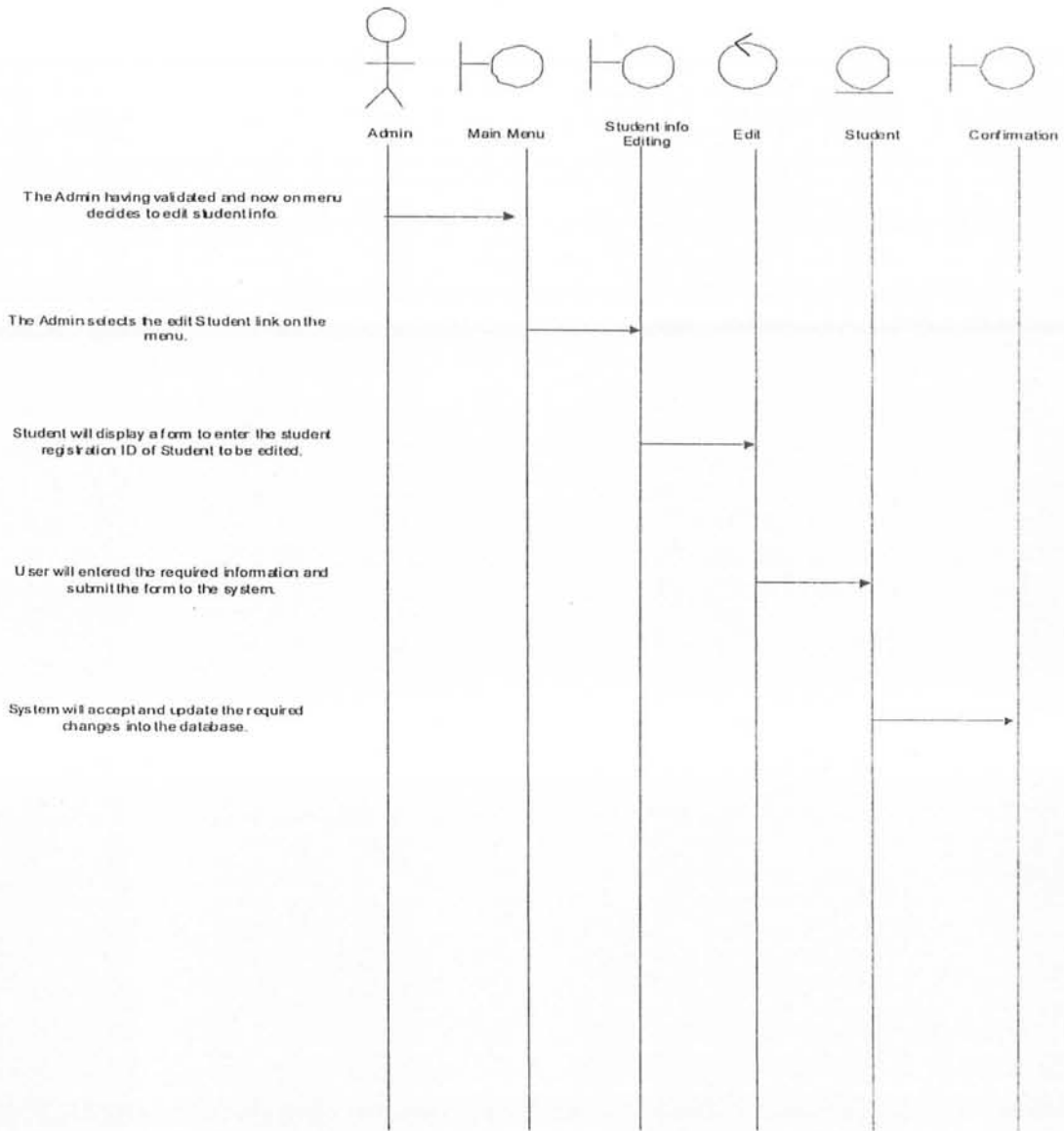
# DEPARTMENT INFORMATION SYSTEM

## SEQUENCE DIAGRAM



# DEPARTMENT INFORMATION SYSTEM

## SEQUENCE DIAGRAM



## **CHAPTER 5**

### **SYSTEM DESIGN**

- INTRODUCTION
- PRESENTATION LOGIC
- DATABASE PLANNING
- DATABASE DESIGN

## **5.1 INTRODUCTION**

Design is where the abstraction of the production takes its first step into the reality of software. Design starts with the analysis model as the major input. The principal activity of design is to refine the analysis model such that it can be implemented with the components of the architecture [PRS97]. Basic design activities include:

- 1 Presentation Logic
- 2 Business Logic
- 3 Data Service Logic

## **5.2 PRESENTATION LOGIC**

The presentation layer comprises the entire user interface. Not only does this layer allow the user to interact with the application, input data, and view the results of requests. It manages the browser application performs the tasks of the presentation layer. Presentation logic is interface design of Web application [PRS97]. It includes:

- 1 Interface Design
- 2 Site Design

### **5.2.1 INTERFACE DESIGN**

The user interface of the system is often the measure by which the user of the system interacts with it. If the system interface is very complex or presents information in a misleading way, the user may confuse the meaning of information presented. They may initiate the sequences of actions, which corrupt data or even cause catastrophic system failure. The design of web based application is not an easy task.

## 1 THE ELEMENTS OF INTERFACE DESIGN

Interface design encompasses three distinct, but related constructs--usability, visualization, and functionality [VAL1990]. Recently, a fourth component of interface design has emerged as a critical factor--accessibility. Interface design is most often associated with the development of Web pages, computer software, and multimedia, but is relevant to the creation of any instructional media or technical equipment. For an interface design, following are some of the principles to be used.[BRB00]:

### I. USABILITY

The key design issue is usability. It refers to how intuitively or easily your media item is navigated and processed (flow, sequence, instructions, and download time). My application supports the work of the user and easy navigation throughout the system. Keeping the user preferences in view the system has been maximized as user friendly. Numbers of hits required achieving user's objectives, have been maximized through the use of Java Script for front-end validation etc.

### II. USER FAMILIARITY.

The interface should use terms, which are drawn from the experience of the anticipated class of user. In my system, most of the things are memorable and recognized by user.

### III. CONSISTENCY

The interface should be consistent in that comparable operations should be activated in the same way. All the major functionality of my system is activated through links.



#### **IV. MINIMAL SURPRISE**

User should never be surprised by the behavior of the system. In my system less probability of error.

#### **V. USER GUIDANCE**

The interface should incorporate some form of context-sensitive user guidance and assistance. User guidance define the various components of the users desktop environment that a system administrator needs to manage e.g. the program that are available to users, the programs that appear on the users desktop and option for the start menu. Detail of subjects should be given.

#### **VI. FUNCTIONALITY**

It refers to the features of your media item and how useful they are for supporting a given task (e.g., interactive simulations, drill and practice quizzes, site maps, frequently asked questions, search engines).

#### **VII. ACCESSIBILITY**

It is an emerging Web interface design topic; if not addressed, it will negatively influence Web site usability for users with certain disabilities; tools that help users access your site in alternative formats (e.g., text, aural, visual) provide for increased functionality.

#### **VIII. SECURITY**

It is the ability to withstand malicious attacks. My application does not allow un authenticated users to access the system. We have applied the session management for proper authenticating of users accessing the system. We also have applied the password encryption so that no one can see the password.

### 5.3 INPUT DESIGN

It is important to design forms, which should reduce or minimize errors. As my application requires majority of the inputs from user, the field provided satisfy this criteria. For instance, in fields such as name etc only text can be added, numbers cannot be entered. Similarly for the numbers fields text is not allowed. To avoid invalid inputs, combo boxes with the relevant data provided. This will prevent user from entering invalid information. Apart from above, We have made the following considerations while designing input forms:

- 1 Uniform layout for all input screens.
- 2 Screens are user friendly. Since every thing on the screens (buttons, hyperlinks etc) are self-explanatory hence enhancing navigation speeds.
- 3 Screens the purpose for which they are designed.
- 4 Input fields have meaningful labels.
- 5 Avoid spelling mistakes.

### 5.4 OUTPUT DESIGN

The most important purpose of maintaining a database is to present the output in some meaningful way. For this reason, I have designed the forms keeping in mind the following considerations

- 1 Valid and to the point title is displayed on the top of each form.
- 2 Fields are labeled meaningfully.
- 3 Fonts and colors for the web pages have been selected after repeated testing with two most popular browsers i.e., Internet Explorer and Netscape Navigator.

- 4 Proper messages are displayed to the user if some invalid action is performed.
- 5 Consistent user interface through out the system.

## 5.5 DATABASE PLANNING

The purpose of Examination Database Application is to maintain the data that department get and generate for Students, Courses and Teachers and to facilitate the Cooperation and sharing of information between Students, Teachers and Administration.

### 5.5.1 SYSTEM DEFINITION:

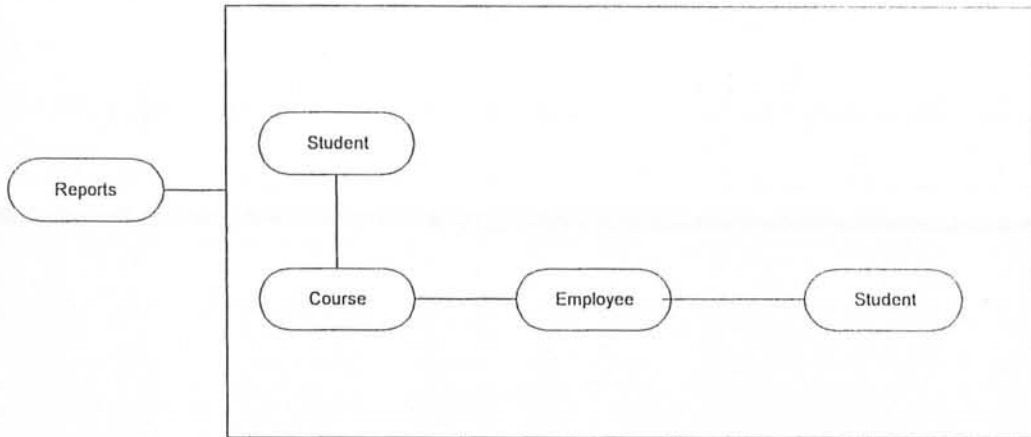
It includes the Scope and Objectives of the Database Application including its major User views.

#### 1 MISSION STATEMENT:

1. To maintain (Enter, Update and Delete) data on Students.
2. To maintain (Enter, Update and Delete) data on Courses.
3. To maintain (Enter, Update and Delete) data on Employees.
4. To maintain (Enter, Update and Delete) data on Registration.
5. To maintain (Enter, Update and Delete) data on Results.
6. To track status of Students Semester Wise.
7. To track status of Courses.
8. To track status of Registered Student Courses.
9. To track status of Employees of Department.
10. To Report on Students Mark Sheet.
11. To Report on Students Semester Result Card.
12. To Report on Students Progress Card.



## 2 SYSTEMS BOUNDARY



## 3 MAJOR USER VIEWS OF SYSTEM:

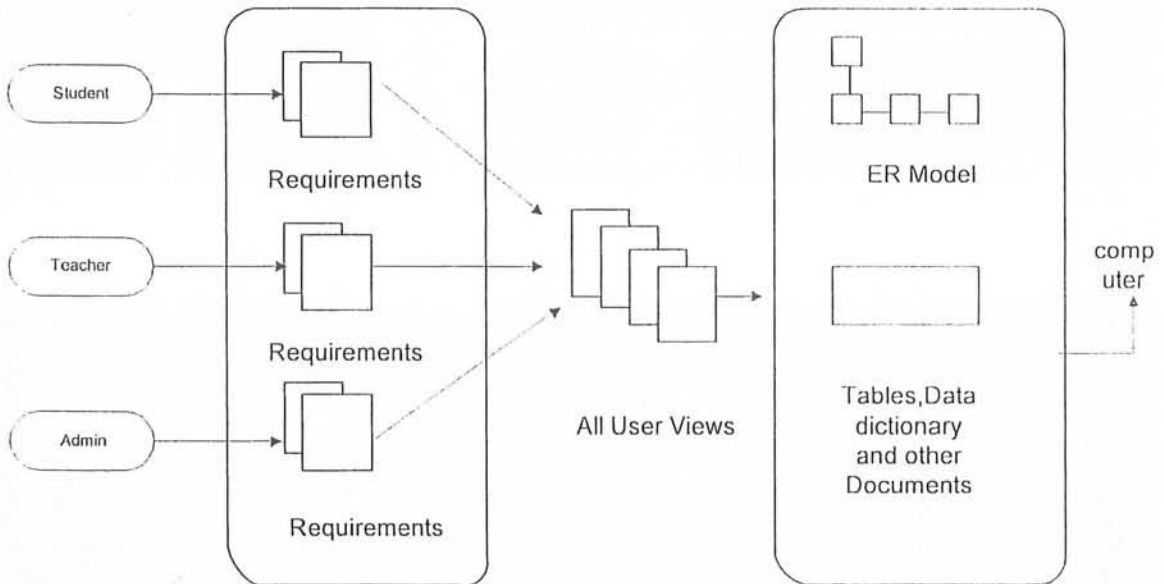
User View	Requirements
<b>Admin Section</b>	<ol style="list-style-type: none"> <li>1. To maintain (Enter, Update and Delete) data on Students.</li> <li>2. To maintain (Enter, Update and Delete) data on Courses.</li> <li>3. To maintain (Enter, Update and Delete) data on Employees.</li> <li>4. To maintain (Enter, Update and Delete) data on Registration.</li> <li>5. To maintain (Enter, Update and Delete) data on Results.</li> <li>6. To track status of Students profile.</li> <li>7. To track status of Students Semester Wise.</li> <li>8. To track status of Courses.</li> <li>9. To track status of Registered Student Courses.</li> <li>10. To track status of Employees of Department.</li> <li>11. To track status of Class Wise.</li> <li>12. To Report on Students Mark Sheet.</li> <li>13. To Report on Students Semester Result Card.</li> <li>14. To Report on Students Progress Card.</li> <li>15. To Report on Students Merit List.</li> </ol>
<b>Teacher</b>	<ol style="list-style-type: none"> <li>1. To maintain (Enter, update and delete) data on Sessional Marks.</li> </ol>

## 5.6 REQUIREMENTS COLLECTION AND ANALYSIS

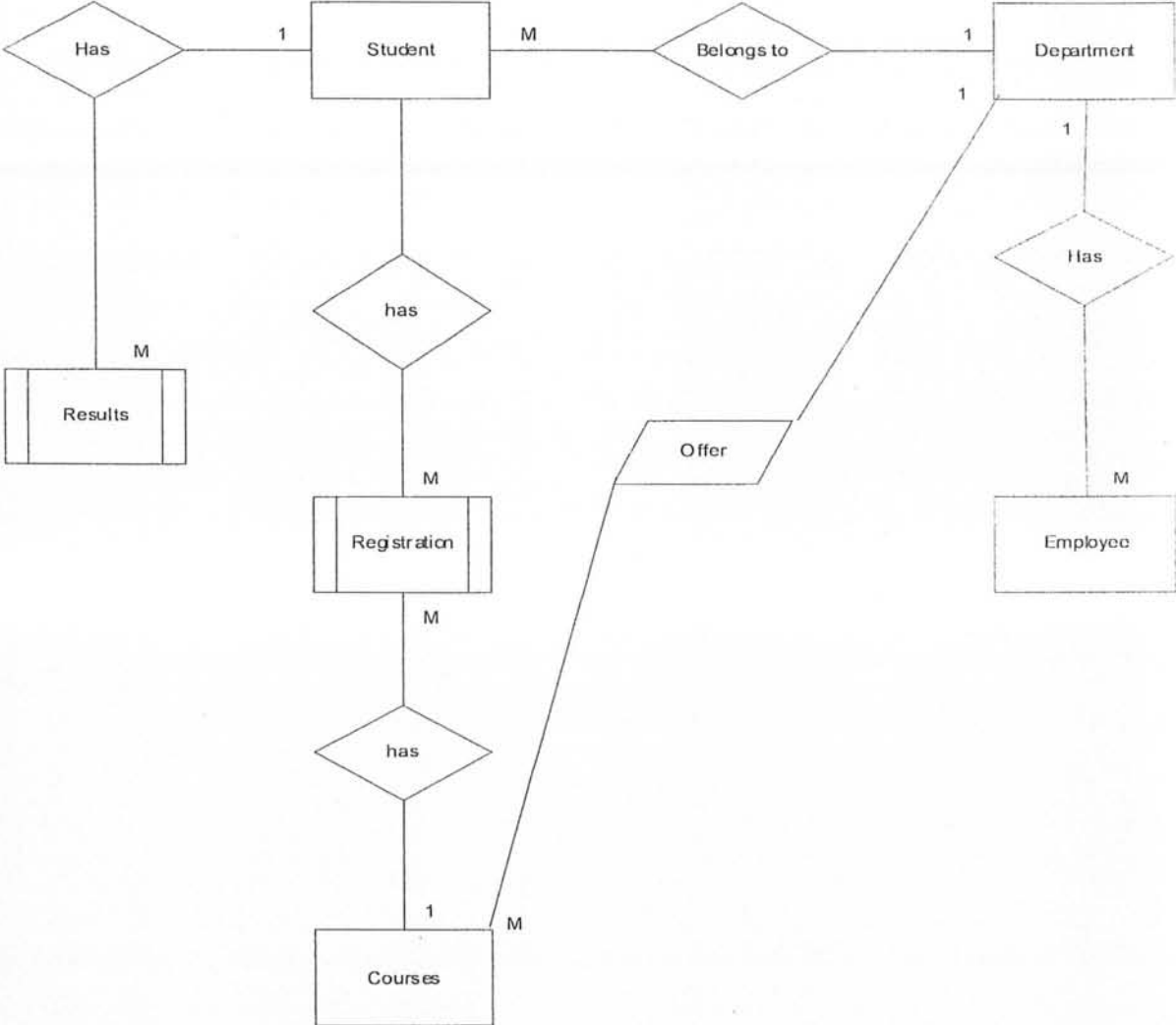
This includes

1. Description of data used or generated.
2. Details of how data is to be used or generated.
3. Additional requirements for new database Application.

Approach that I follow is centralized approach; in it requirements for each user view are merged into a single set of requirements for new database application.



Entity Relationship Diagram



## 5.7 DATABASE DESIGN

The process of creating a design of a database that will support system operations and objectives. Database design is the design of the database that supports the objectives and operations of the system in which

- 1 The data and relationships between the data, required by the users are represented.
- 2 The data models that supports any transactions required on the data is provided.
- 3 A design is specified that should achieve the requirement performance for the system.

The specification of different tables is as follows:

**Table Number:** 1

**Table Name:** Entities

**Purpose:** Stores all the entities

### 5.7.1 ENTITIES:

Entity Name	Relationship	Entity
Student	Belongs to	Department
	Has	Results
	Registers	Courses
Department	Offer	Courses
	Has	Employees
Courses	Taken by	Students
Employee	Has	Registration
Result	Belongs to	Students

**Table Number:** 2

**Table Name:** Students

**Purpose:** Stores all the attributes of students

**5.7.2 ENTITIES ALONG WITH ATTRIBUTES:**

**Entity: Student**

Attribute	Description	Data type & Length	Key	Nulls	Multi value	Composite	Derived	Default
Reg No	Registration Number, Primary Key	16 fixed char	P.K	No	No	No	No	00000-000/0000-00
Form No	Form Number	5 fixed char		No	No	No	No	
Roll No	Roll Number	5 fixed char		No	No	No	No	
Class	MSc/M-Phil/Phd	3 fixed char		No	No	No	No	
Sex	Sex of Student	1 fixed char		No	No	No	No	
NIC	National Identification Card	20 Variable Char		No	No	No	No	
Semester	Fall/Spring	1 fixed char		No	No	No	No	
Session	Year of enrollment e.g 1998-99	7 fixed char		No	No	No	No	
FName	First Name of Student	20 Variable Char		No	No	No	No	
LName	Last Name of Student	20 Variable Char		No	No	No	No	
Father name	Father's Name of Student	30 Variable Char		No	No	No	No	
Date Birth	Date of Birth of Student	Date		No	No	No	No	0000-00-00
Prov Code	Domicile Province of Student	1 fixed char		No	No	No	No	
Phone No	Phone No of Student	20 Variable Char		No	No	No	No	



*DEPARTMENT INFORMATION SYSTEM*

Address	Residential Address of Student	30 Variable Char		No	No	No	No	
Metric Marks	Metric marks of Student	4 fixed integers		No	No	No	No	
Metric Div	Division obtained in Metric	1 fixed integer		No	No	No	No	
Metric BU	Board/University of study	20 Variable Char		No	No	No	No	
F-sc Marks	F-sc Marks of Student	4 fixed integers		No	No	No	No	
F-sc Div	Division in F-sc	1 fixed integer		No	No	No	No	
F-sc Bu	Board/University of study	20 Variable Char		No	No	No	No	
B-sc Marks	B-sc Marks of Student	4 fixed integers		No	No	No	No	
B-sc Div	Division in B-sc	1 fixed integer		No	No	No	No	
B-sc BU	Board/University of Study	20 Variable Char		No	No	No	No	

*DEPARTMENT INFORMATION SYSTEM*

**Table Number:** 3

**Table Name:** Course

**Purpose:** Stores all the attributes of the courses

**Entity: Courses**

Attribute	Description	Data Type & Length	Key	Nulls	Multivalued	Composite	Derived	Default
Course Id	Course Code, Primary Key	7fixed char	PK	NO	NO	NO	NO	000-000
Course Name	Course Name	100variable char		NO	NO	NO	NO	
Credit	No of Credit Hours	1fixed char		NO	NO	NO	NO	1
Type	Type of course-Credit/NC	1fixed char		NO	NO	NO	NO	c
Sub Code	Foreign Key from Departments Table	3fixed integers						

*DEPARTMENT INFORMATION SYSTEM*

**Table Number:** 4

**Table Name:** Result

**Purpose:** Stores all the attributes of results

**Entity:** Results (Week Entity)

Attribute	Description	Data type & length	Key	Nulls	Multivalued	Composite	Derived
Reg_No	Registration Number, Primary key	16 fixed char	F.K	No	No	Yes	No
Semester	Semester, primary key	1 fixed integer	F.K	No	No	Yes	No
TMarks	Total marks	3 fixed integer	-	No	No	No	No
TCredits	Total Credits passed	3 fixed integer	-	No	No	No	No
Status	Status of student result, Passed, Failed, Chance, Promoted, Drop, Later On	1 fixed integer	-	No	No	No	No

5.8.3 REFERENTIAL INTEGRITY CONSTRAINTS (INSERTION DELETION UPDATION ANOMALIES)

Transaction	Tables Required
1. Insert new student	<b>Student</b> Table: Student Registration ID as P.K
2. Delete Student	<b>Registration</b> Table: F.K=Students_Id References Student <b>Results</b> Table: FK= Students_Id References Student ON DELETE CASCADE
3. Update Student	<b>Registration</b> Table: F.K=Students_Id References Student <b>Results</b> Table: FK= Students_Id References Student ON UPDATE NO ACTION: Not Allowed to update Registration id once entered.
4. List Students	<b>DEPARTMENT</b> Table: Student Registration ID as PK
5. insert New Department	<b>DEPARTMENT</b> Table: Subject_code as P.K
6. Delete Department	<b>DEPARTMENT</b> Table: Subject_code as P.K <b>Courses</b> Table: F.K=Students_code References Department <b>Student</b> Table: F.K=Students_code References Department <b>Employees</b> Table: F.K=Students_code References Department ON DELETE CASCADE
7. Update Department	<b>DEPARTMENT</b> Table: Subject_code as P.K <b>Courses</b> Table: F.K=Students_code References Department

*DEPARTMENT INFORMATION SYSTEM*

	<p><b>Student</b> Table: F.K=Students_code References Department</p> <p><b>Employees</b> Table: F.K=Students_code References Department ON UPDATE CASCADE</p>
8.List Department	<b>Department</b> Table: Subject_code as PK
9. Insert New Group	<b>Group</b> Table: Group_Code as P.K
10. Delete Group	<b>Group</b> Table: Group_Code as P.K <b>Employees</b> Table: FK= Group_Code References Group ON DELETE SET DEFAULT
11.Update Group	<b>Group</b> Table: Group_Code as P.K <b>Employees</b> Table: FK= Group_Code References Group ON UPDATE CASCADE
12.List Group	<b>Group</b> Table:Group_Code as P.k
13. Insert New Course	<b>Course</b> Table:Course_Code as P.k
14.Delete Course	<b>Course</b> Table:Course_Code as P.k <b>Registration</b> Table: F.K=Students_Id References course ON DELETE CASCADE
15.UPDATE Course	<b>Course</b> Table:Course_Code as P.k <b>Registration</b> Table: F.K=Students_Id References course ON UPDATE CASCADE
16.List Course	<b>Course</b> Table:Course_Code as P.k
17.List Final Result	<b>Results</b> Table: Student Registration Id,Semester as P.K
18.List Student Marks	<b>Registration</b> Table: Student Registration Id, Course_Code, Semester as P.K
19.List Student Registration	<b>Registration</b> Table: Student Registration Id, Course_Code, Semester as P.K
20.List DMC/Mark Sheet	<b>Student</b> Table:Student Registration Id as P.K <b>Course</b> Table:Course_Code as P.k <b>Department</b> Table: Subject_code as PK

*DEPARTMENT INFORMATION SYSTEM*

	<p><b>Registration</b> Table: Student Registration Id, Course_Code, Semester as P.K  <b>Results</b> Table: Student Registration Id, Semester as P.K</p>
21.List Student Grade Card	<p><b>Student</b> Table: Student Registration Id as P.K  <b>Department</b> Table: Subject_code as PK  <b>Student</b> Table: Student Registration Id as P.K</p>
22.List Gazette Notification	<p><b>Student</b> Table: Student Registration Id as P.K  <b>Registration</b> Table: Student Registration Id, Course_Code, Semester as P.K</p>
23.List Students Progress Card	<p><b>Student</b> Table: Student Registration Id as P.K  <b>Department</b> Table: Subject_code as PK  <b>Registration</b> Table: Student Registration Id, Course_Code, Semester as P.K</p>
24.List Students Semester Merit List	<p><b>Student</b> Table: Student Registration Id as P.K  <b>Department</b> Table: Subject_code as PK  <b>Results</b> Table: Student Registration Id, Semester as P.K</p>

## **CHAPTER 6**

### **IMPLEMENTATION**

- **INTRODUCTION**
- **WEB SCRIPTING TECHNOLOGY**
- **DEVELOPMENT IDE TOOLS**
- **HARDWARE AND SOFTWARE REQUIREMENTS**

## 6.1 INTRODUCTION

After the completion of the design phase, the development phase of the software starts. The purpose of the development phase is to transfer the design into executable computer software. In the development phase, the system is built to meet designed specification and is concerned with the tools used in the development work and the components used to implement the system[PRS97].

## 6.2 TECHNOLOGY SELECTION

The selection of tool to use in development nontrivial software is crucial. The choice of appropriate tools is essential for avoiding difficulties in the later stages of the system life cycle. Therefore, after considering the various alternatives in the development of Department Information System, the following technologies have been selected for my application.

- 1 PHP for putting the functionality in the web page.
- 2 My SQL
- 3 Java Script for the client side validation.

### 6.2.1 PHP (HYPERTEXT PROCESSOR)

PHP is a powerful server-side scripting language for creating dynamic and interactive websites. The PHP is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software application.

PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is perfectly suited for Web development and can be embedded directly into the HTML code and then saved with a .php file extension.

The PHP syntax is very similar to Perl and C. PHP is often used together with Apache



(web server) on various operating systems. It also supports ISAPI and can be used with Microsoft's IIS on Windows.

### **6.2.2 My SQL SERVER**

MySQL is a database. A database defines a structure for storing information. In a database, there are tables. Just like HTML tables, database tables contain rows, columns, and cells. Following are the some important features in My SQL :

- 1 It is ideal for small and medium applications.
- 2 It supports standard SQL.
- 3 It compiles on number of platform.
- 4 It has built- in support for SQL statements to check, optimize, and repair table .
- 5 A privilege and password system that is very flexible and secure and allows host based verification.

### **6.2.3 JAVASCRIPT**

The most common scripting technology available in browsers today is JavaScript. JavaScript, one part of the whole JavaScript revolution, is an implementation of a scripting language that has its roots in Java language. JavaScript is intended to be easier than Java.

## **6.3 DEVELOPMENTS IDE TOOLS**

The development tools which are used in this project are

1. Macromedia Dream weaver MX 2004
2. Macromedia Fireworks MX 2004

### **6.3.1 MACROMEDIA DREAM WEAVER MX 2004**

Dream weaver is Haverford's supported web editor. It has features for both the beginning and advanced web page creator. Dream weaver integrates many aspects of Web development, including page creation, site management, and web server tools, giving the user a good perspective of an entire web site. Dream weaver also supports JavaScript and Cascading Style Sheets, along with other advanced functions. We have used macromedia Dream weaver MX 2004 as a text editor for php files.

### **6.3.2 MACROMEDIA FIREWORKS MX 2004**

Macromedia Fireworks MX 2004 is the complete solution for creating, optimizing, and integrating great-looking Web graphics. Fireworks MX 2004 gives Web developers professional control to quickly create original Web graphics and interactivity, from simple graphical buttons to sophisticated rollover effects and pop-up menus. Fireworks MX 2004 increases flexibility, allowing easy import, integration, and editing of vector and bitmap images in all the major graphics formats, including exporting to other Macromedia products, such as Flash and Dream weaver, and third-party Web development tools. Streamline any design or development work flow with the program's significant performance and usability enhancements. We have used macromedia fireworks MX 2004 for editing our images.

## 6.4 HARDWARE REQUIREMENT OF SYSTEM

### 6.4.1 SERVER SIDE REQUIREMENTS

### 6.4.2 CLIENT SIDE REQUIREMENTS

SERVER SIDE REQUIREMENTS		
DEVICE	MINIMUM	RECOMMENDED
PROCESSOR	2 GHZ	3GHZ
HARD DISK	40GB	80GB
RAM	256MB	1GB
LAN CARD	100MBPS	1000MBPS

CLIENT SIDE REQUIREMENTS		
DEVICE	MINIMUM	RECOMMENDED
PROCESSOR	256MHZ	500MHZ
HARDDISK	10GB	20GB
RAM	128MB	500MB
LAN CARD	100MBPS	500MBPS

## **6.5 SOFTWARE REQUIREMENT OF SYSTEM**

- 1 For server side
- 2 Operating system: Microsoft Window XP-2003
- 3 APACHE Server
- 4 Office XP-2002
- 5 Macromedia FLASH
- 6 Macromedia Dreamweaver MX
- 7 Microsoft Visio

### **1.5 FOR CLIENT SIDE**

- 1 Web Browser
- 2 Flash plugging

## **CHAPTER 7**

### **SYSTEM TESTING**

- **Introduction**
- **Testing Chasses**
- **System Evaluation**
- **Future Enhancement**

## 7.1 INTRODUCTION

Before actually implementing the new system into operations, a test run of the system is done removing all the bugs, if any. It is an important phase of a successful system. After codifying the whole programs of the system, a test plan should be developed and run on a given set of test data. The output of the test run should match the expected results.

Using the test data following test run are carried out:

- 1 Unit test
- 2 System test

Unit test: When the programs have been coded and compiled and brought to working conditions, they must be individually tested with the prepared test data. Any undesirable happening must be noted and debugged (error corrections).

System Test: After carrying out the unit test for each of the programs of the system and when errors are removed, then system test is done. At this stage the test is done on actual data. The complete system is executed on the actual data. At each stage of the execution, the results or output of the system is analyzed. During the result analysis, it may be found that the outputs are not matching the expected out of the system. In such case, the errors in the particular programs are identified and are fixed and further tested for the expected output.

When it is ensured that the system is running error-free, the users are called with their own actual data so that the system could be shown running as per their requirements.

## 7.2 TESTING CHASSES

We performed various tests to analyze our web contents.

### 1 HTML Pages static Information verification

We submitted all final web pages to the Administration department to review the published contents on the HTML pages. We had spent 1 week in this process and rectified our composing and drafting errors.

### 2 php Pages results

This was the check of correctness, completeness and accuracy of the MYSQL query results displayed on PHP pages. We had passed every possible query to recoil MYSQL query results from our MYSQL Database. This Test was executed for:

### 3 QUERY

We passed different values for

- Reg\_id
- CId
- Employee\_id
- Semester

And get results on PSP pages. For all invalid values we put appropriate error messages like:

- 1 Data not found
- 2 Invalid entry
- 3 will be available soon ...

### **7.3 SYSTEM EVALUATION**

We have achieved almost 90% of our general as well as specific objectives out of our project.

- 1 More closer and updated information at user's door
- 2 Student result system will become more effective
- 3 Online News & updates

### **7.4 FUTURE ENHANCEMENT**

#### **1 PROPOSED EXTENSION**

Not every system is absolute and 100% perfect. There exists always a room to enhance and extend the systems. We assess has a plenty of room for future extensions both due to enhancements as well as growth. That is the reason; we have hopeful that after the successful implementation that we can further extend the site after 1-2 years with following features:

- 2 Currently, site is only providing limited information about Examination; we can extend it to provide all information of each Department of QAU.
- 3 We can provide search facility in our site so user can easy search information that he or she needed in our site..
- 4 All information provided in our site is downloadable in format of word documents (.doc), Adobe acrobat Reader (.PDF) and PowerPoint (ppt).



## **CHAPTER 8**

### **SYSTEM CODING**

- **INTRODUCTION**
- **CODE FOR ADDING**
- **CODE FOR EDITING**
- **CODE FOR DELETION**

## INTRODUCTION

After designing the new system, the whole system is required to be converted into computer understanding language. **Coding** the new system into computer programming language does this. It is an important stage where the defined procedures are transformed into control specifications by the help of a computer language. This is also called the programming phase in which the programmer converts the program specifications into computer instructions, which we refer as **programs**. The programs coordinate the data movements and control the entire process in a system.

### 8.2 CODING

#### 1 CODING FOR ADDING STUDENT

```
<?php
include "./mysql_connect.inc";

$count=0;
//Get values from Form

$Class   = $HTTP_POST_VARS["Class"];
$Session = $HTTP_POST_VARS["Session"];
$Semester = $HTTP_POST_VARS["Semester"];
$Province = $HTTP_POST_VARS["Province"];
$FName   = $HTTP_POST_VARS["FName"];
$LName   = $HTTP_POST_VARS["LName"];
$FatherName = $HTTP_POST_VARS["FatherName"];
$Month   = $HTTP_POST_VARS["Month"];
$Day     = $HTTP_POST_VARS["Day"];
$Year    = $HTTP_POST_VARS["Year"];
$Date    = $Year . '-' . $Month . '-' . $Day ;
$Sex     = $HTTP_POST_VARS["Sex"];
$IdCard  = $HTTP_POST_VARS["IdCard"];
$Email   = $HTTP_POST_VARS["Email"];
$Phone   = $HTTP_POST_VARS["Phone"];
$Address = $HTTP_POST_VARS["Address"];
$Reg_no  = $HTTP_POST_VARS["Reg_no"];
```

**QUERY TO ADD DATA IN DATABASE**

```
mysql_query ("INSERT INTO student(Reg_No, Class, Session, FName, LName, Sex,
Date_of_Birth, NIC, Phone No, Address, Province, Email, Semester)
VALUES
('$Reg_no','$Class', '$Session', UPPER('$FName'),UPPER('$LName'), '$Sex','$Date',
'$IdCard', '$Phone', UPPER('$Address'), '$Province',LOWER('$Email'), '$Semester')");
print "You inserted the following
data Successfully";

?>
```

**2 CODE FOR ADDING COURSE**

```
<?php
include "./mysql_connect.inc";
//Get values from Form
$CId      = $HTTP_POST_VARS["Cid"];
$CName    = $HTTP_POST_VARS["CName"];
$Class    = $HTTP_POST_VARS["Class"];
$CType    = $HTTP_POST_VARS["Type"];
$CHours   = $HTTP_POST_VARS["CHours"];
```

**QUERY TO ADD DATA IN DATABASE**

```
mysql_query ("INSERT INTO courses(CId, CName, CHours,CType,Class)
VALUES('$CId','$CName','$CHours','$CType','$Class')");
print "You inserted the following data Successfully";

?>
```

**3 CODE FOR ADDING EMPLOYEE**

```

<?php
include "./mysql_connect.inc";
$Sid          = $HTTP_POST_VARS['employee_id'];
$Semp_name    = $HTTP_POST_VARS['emp_name'];
$Semp_login   = $HTTP_POST_VARS['emp_login'];
$Semp_pwd     = $HTTP_POST_VARS['emp_pwd'];

$Semp_group   = $HTTP_POST_VARS['group'];
$Semp_department = $HTTP_POST_VARS['department'];
$Semp_email   = $HTTP_POST_VARS['emp_email'];
$Semp_home_no = $HTTP_POST_VARS['emp_home_no'];
$Semp_office_no = $HTTP_POST_VARS['emp_office_no'];
$Semp_cell_no = $HTTP_POST_VARS['emp_cell_no'];
$Semp_address = $HTTP_POST_VARS['emp_address'];

```

**QUERY TO ADD DATA INTO THE DATABASE**

```

// Start of Query
$query = "select employee_id from employees";
$result = mysql_query($query);
$count=0;
while ($rows = mysql_fetch_array($result))
    {
        $sid[$count] = $rows['employee_id'];
        $count++;
    }
$j = $count;
$j = $j - 1;
$id = $sid[$j] ;
$id = $id + 1;

mysql_query ("insert into employees(employee_id ,group_id, sub_code, employee_login,
employee_password, employee_name, email_address, phone_home, phone_office,
phone_cell, address)
VALUES('$id'          , '$Semp_group',          '$Semp_department',          '$Semp_login'
, '$Semp_pwd', '$Semp_name', '$Semp_email', '$Semp_home_no', '$Semp_office_no', '$Semp_cell_n
o', '$Semp_address')");

?>

```

#### 4 CODE FOR EDITING

```
<?php
include "./mysql_connect.inc";

$CIId      = $HTTP_GET_VARS['id'];
$CName     = $HTTP_POST_VARS['CName'];
$CHours    = $HTTP_POST_VARS['CHours'];
$CType     = $HTTP_POST_VARS['CType'];
$Class     = $HTTP_POST_VARS['Class'];
print $CIId;
print $CName;
print $CHours;
print $CType;
print $Class;
```

#### QUERY TO EDIT DATA

```
mysql_query("Update      courses set      CName='$CName',  CHours='$CHours',
Class='$Class', CType='$CType' WHERE CIId = '$CIId'");
?>
```

## 5 CODE FOR LOGIN

```

<?php
// start the session
//session_start();
//header("Cache-control: private"); //IE 6 Fix

include "./mysql_connect.inc";

// Get the user's input from the form
$name = $_HTTP_POST_VARS['logid'];
$password = $_HTTP_POST_VARS['userpass'];
$type = $_HTTP_POST_VARS['type'];
$counter = 0;

switch($type)
{
    case '1':
        {
            $query = "select * from employees where employee_login
            =\"$name\" and employee_password=\"$password\" and group_id = '1'";
            $result = mysql_query($query);
            while($rows = mysql_fetch_array($result))
            {
                $login = $rows['employee_login'];
                $password = $rows['employee_password'];

                $counter++;
            }
            if($counter > 0)
            {
                // Register session key with the value
                $_SESSION['name'] = $login;
                $_SESSION['id'] = $type;
                $redirect = $redirect
                dirname($_HTTP_SERVER_VARS['PHP_SELF']). "/";
                // $redirect == $redirect

                $redirect = "Admin.php";
                header("Location: $redirect");
            }
        }
    else
        {

```

DEPARTMENT INFORMATION SYSTEM

```
print "Your requested record does
not exist";
}
break;
}
case '2':
{
$query = "select * from employees where employee_login
= \"\$lname\" and employee_password='\$pwd' and group_id ='2' ";
$result = mysql_query($query);
while($rows = mysql_fetch_array($result))
{
$login = $rows['employee_login'];
$password = $rows['employee_password'];
$counter++;
}
if($counter > 0)
{
// Register session key with the value
$_SESSION['name'] = $login;
$_SESSION['id'] = $type;
$redirect = $redirect .
dirname($_SERVER_VARS['PHP_SELF']) . "/";
//$redirect = $redirect .
"Admin.php";
$redirect = "Admin.php";
header("Location: $redirect");
}
else
{
print "Your requested record does
not exist";
}
break;
}
default:
{
print "Requested User does not exist";
}
}
?>
```

## CODE FOR DELETION

```
<?php
include "./mysql_connect.inc";
        $id = $_HTTP_GET_VARS['id'];
print $id;
?>
```

## QUERY TO DELETE DATA FROM DATABASE

- 1 <?php  
mysql\_query("delete from courses where CId = '\$id'");  
print "You're requested record has been successfully deleted";  
?>
  
- 2 <?php  
mysql\_query("delete from student where reg\_no = '\$id'");  
print "You're requested record has been successfully deleted";  
?>



## WEBLIOGRAPHY

### SITES

- 1 [www.vbip.com](http://www.vbip.com)
- 2 [www.vbsquare.com](http://www.vbsquare.com)
- 3 [www.slipstick.com](http://www.slipstick.com)
- 4 [www.asptoday.com](http://www.asptoday.com)
- 5 [www.4guysfromrolla.com](http://www.4guysfromrolla.com)
- 6 [www.aspsimply.com](http://www.aspsimply.com)
- 7 [www.internet.com](http://www.internet.com)
- 8 [www.hotscripts.com](http://www.hotscripts.com)
- 9 [www.planetsourcecode.com](http://www.planetsourcecode.com)

## BIBLIOGRAPHY

### BOOKS

- 1 Conallen, Jim. (2000). Building Web Application with UML
- 2 Buser, David. (2000). Beginning Active Server Pages 3.0, Wrox
- 3 Sussman, Dave. (1999). Professional Active Server Pages 3.0, Wrox
- 4 Petroustos, Evangelos. (1998). Mastering Visual Basic
- 5 Bruegge, Bernd. (2000). Object-Oriented Software Engineering
- 6 Jaworski, James. (1999). Mastering JavaScript and JScript

## REFERENCES

### BOOKS

- 1 [JIM00] Conallen, Jim. (2000). Building Web Application with UML
- 2 [AND99] Anderson, Richard, Professional Active Server Pages 3.0, Wrox, 1999
- 3 [MSDN]
- 4 [PRS97] Pressman, Roger. (1997). Software Engineering A Practitioner's Approach, 4<sup>th</sup> Edition
- 5 [SQL90] Gruber, Martin. (1990). Understanding SQL
- 6 [BRB00] Bruegge, Bernd. (2000). Object-Oriented Software Engineering

**APPENDIX A**

## WEB APPLICATION EXTENSION FOR UML

### DESCRIPTION

The extension UML defines a set of stereotypes, tagged values, and constraint that enable us to model Web applications. The stereotypes and constraints are applied to certain components that are particular to Web applications, allowing us to represent them in the same model and on the same diagrams that describe the rest of the system.

### STEREOTYPES

### USE CASE

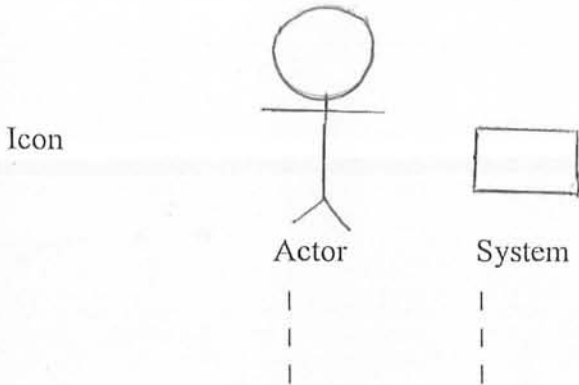
Description            A Use Case, a textual description of the interaction between an actor and the system, contains a narrative description of a specific usage scenario, one in which an actor supplies input and the system exhibits an observable output.


Icon

Constraints            None

**SEQUENCE DIAGRAM**

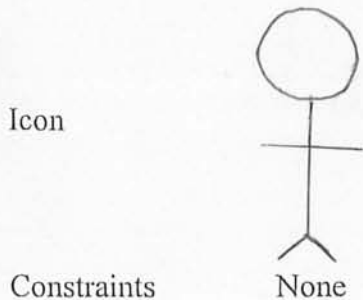
Description            A sequence diagram, a specific type of interaction diagram, expresses the interaction between an actor and the system, with a special emphasis on the time line.



Constraints            None

**ACTOR**

Description            An actor represents a role that a user can play with the system. The actor is associated with the Log-on use case. This means that the actor can “invoke” this use case and participate in the scenario described in it.



ANALYSIS DIAGRAM

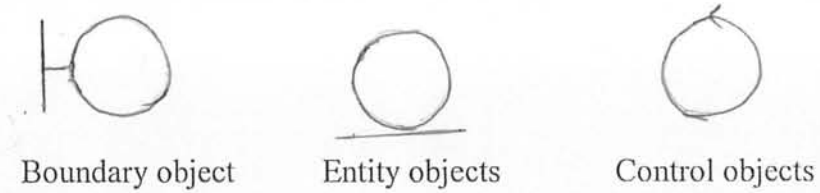
Description Use case analysis is another activity that is done when the use case near completion. It identifies the classes and objects that will perform a use case's flow of events. Analysis classes can be stereotyped into three types: boundary, entity, and control.

**Boundary objects** represent the interface between the actor and the system. In Web applications, these may represent the whole page.

**Entity objects** are those things that are described in the use case but That will outlast it. Orders, customers, products, and payroll are entity, objects whose instances may appear in many invocations of use cases.

**Control objects** represent processes. These objects represent system activities that can often b named. Control objects direct the activities of the entity and the interface objects.

Icon



Constraints None

**APPENDIX B**



## **FEASIBILITY REPORT**

It is defined as an evaluation or analysis of the potential impact of a project or program. This report is conducted to assist decision makers in determining whether or not to implement a particular project.

A critical part of this feasibility report is the discussion of the requirements use to reach the final discussion or recommendation. This report shows economic, operational as well as technical feasibility. It also includes the brief description about this project and the tools used to develop it.

## **CONTENTS OF REPORT**

- 1 TECHNICAL FEASIBILITY
  1. DATA BASE SELCTION
  2. SCRIPTING LANGUAGE
  3. OPERATING SYSTEM
  4. BROWSER
  5. SOFTWARE DEVELOPMENT TOOLS
  6. DOCUMENTATION TOOLS
  7. WEB SERVER
  8. HARDWARE REQUIREMENTS
- 2 ECONOMIC FEASIBILITY
- 3 OPERATIONAL FEASIBILITY
- 4 PROCESS

## TECHNICAL FEASIBILITY

The process of assessing the development organization's ability to construct a propose system. It includes:

1. **DATA BASE SELECTION** : There are different database available which could be used in the development of this particular project as SQL and MYSQL. But the most suitable database is MYSQL for this project.

### 1 REASONS FOR USING MYSQL

There are many reasons for using MYSQL in this project.

1. It is ideal for small and medium applications.
2. It supports standard SQL.
3. It compiles on number of platform.
4. It has built- in support for SQL statements to check , optimize, and repair table .
5. A previlege and password system that is very flexible and secure and allows host based verification.

2. **SCRIPTING:** Two scripting languages are being used in this project.using JAVA SCRIPT for client side and PHP for server side .

PHP stands for HYPERTEXT PROCESSOR. It supports many database MYSQL.

### **WHY PHP?**

1. It runs on different platforms(windows ,linux ,unix)
2. Compatible with almost all services available today (Apache,IIS , etc)
3. Easy to learn and run efficiently on tne server side.

**3. OPERATING SYSTEM:**

1. Using WINDOWS XP for the development of the project.
2. Using linux and windows both for deployment.

**4. BROWSER:** Using Internet Explorer

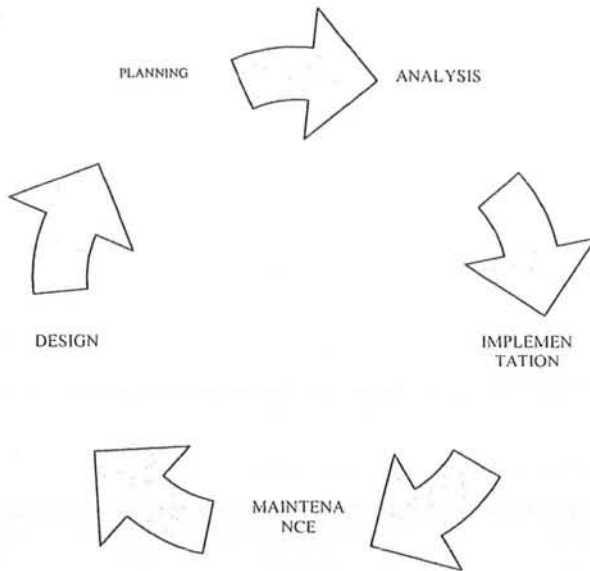
**5. DOCUMENTATION TOOL:** Ms Word 2000

**6. SOFTWARE DEVELOPMENT TOOL:** Dream weaver ,  
Fire works.

**7. WEB SERVER:** Apache.

**8.PROCESS:**

Iterative process creates a framework for the participation of users through out the design and development of the project. Each successive iteration adds detail to the design such that the process result in more precise requirements that better meet users need.



### ITERATION PROCESS

#### ECONOMIC FEASIBILITY

1. Latest Pentium computer are the available in department.
2. Net services available via Fiber optic cable.
3. It is secure and not expensive.
4. University has its own web hosting facility.
5. It is well protected from Viruses.

#### OPERATIONAL FEASIBILITY

Operational feasibility depends on:

1. Management and User support
2. User involvement in planning
3. Impact on performance, customer and company image
4. Reasonable schedules

**APPENDIX C**

## **RISK MANAGEMENT**

### **INTRODUCTION**

Despite much research and progress in the area of Software Project Management, software development projects still are not achieving the target of delivering desired systems on time, within the available financial resources and desired quality. Much of the failure in achieving those targets could be avoided by managers proactive planning for dealing with risk factors rather than waiting for problems to occur and then trying to react on the time of occurrence. Usually this reaction is too little and too late, because by the time the problem is fully recognized, the schedule has already been disturbed, a considerable amount of resources has been utilized, and the product quality has suffered due to introduction of errors. Risk management has been proposed as a solution to for overcoming errors appeared insight into potential problem areas and to identify these problems, address and eliminate them before they can create any problems in the project.

The system should support Risk Assessment during the initial

### **PURPOSE**

The purpose of this document is to describe how we can perform the job of managing risks for online testing. It identifies risks which may occur in the project, defines roles and responsibilities for participants in the risk management process, the risk management activities that will be carried out, the schedule and budget for risk management activities and tools and techniques that will be used during this process.

### **RISK LIST**

The risk factors identified and managed for this project will be accumulated in a risk list. The Risk list contains the following items:

## TIGHTENED SCHEDULE

- 1 Lack of skills.
- 2 Unavailability of the expert
- 3 System Crashing.
- 4 Load shedding
- 5 In availability of the developer

## RISK IDENTIFICATION

Risk #	Risk description	Category
1	Tough schedule	Schedule Risk
2	Lack of skills	Technical Risk
3	Unavailability of expert	Project Risk
4	System Crashing	Project Risk
5	Loadshedding	System Risk
6	Unavailability of developer	Technical Risk

**Risk#1:** we have only two and half month to complete this project and that is not a huge amount of time to complete a large project. Time and schedule is may be a risk for our project.

**Risk #2:** Due to summer vocations, university is off and there is no one who guides us so it is another risk for our project.

**Risk#3:** Viruses is a big challenge for windows; because we also use Windows XP; it becomes a permanent risk that our system may stop working at any time.

**Risk#4:** In our area of living; loadshedding is another risk that may occur at any day, any time and anywhere.

**Risk#5:** Although we spent some time here in PGD but we may not able to get expertise in developing large software projects so it is another risk for our project.

**Risk#6:** One of our developers is ill during the start of the project; it affects the project badly.

## **HOW TO AVOID FROM RISK**

- 1 We will try to manage the time before the summer vocations.
- 2 Although one of our developer is absent; one developer is trying to manage the things individually
- 3 For managing the avoiding from the system crashing risk; we put one of our project's backup on another system.[P S]



**APPENDIX D**

## USER GUIDE

### **Install & Configure Apache with PHP and MySQL on Windows XP Pro**

#### **Versions used:**

Apache 2.0.39

PHP 4.2.1

MySQL 3.23.51

MyODBC 2.50.39

PHPMyAdmin 2.2.6

Modified php.ini. Changed the following:

include\_path = ".;c:\php\includes"

session.save\_path = C:\Temp

session.cookie\_path = \

2002/07/31 Modified workers.properties file – some paths were for older versions of Tomcat

2002/07/26 Modified step 5 of JSP install - hardcoded the Tomcat configuration into the Apache httpd.conf file.

Also added MySQL installation instructions.

#### **DOWNLOAD THE COMPONENTS**

Before you can begin, you must first download all the required components. The components

used in this tutorial are listed below. You may want to download the latest versions. The versions used in the tutorial are listed.

#### **APACHE WEB SERVER**

- <http://httpd.apache.org/>
- <http://httpd.apache.org/dist/httpd/binaries/win32/>
- [apache\\_2.0.39-win32-x86-no\\_ssl.msi](#)

#### **PHP**

- PHP Win32 Binaries
- <http://www.php.net/>
- <http://www.php.net/downloads.php>
- [php-4.2.1-Win32.zip](#)
- PHP > Apache connector
- <http://FReemIRC.com/miki/php4apache2.dll>
- or <http://Rafigh.com/miki/php4apache2.dll>
- or <http://IRan4You.com/miki/php4apache2.dll>
- [php4apache2.dll](#)

#### **MySQL**

- MySQL Database
- <http://www.mysql.com/downloads/mysql.html>
- Download the latest stable release for Windows
- [mysql-3.23.51-win.zip](#)
- MyODBC
- <http://www.mysql.com/downloads/>
- Download the latest stable release of MyODBC for WinXP
- [myodbc-2.50.39-nt.zip](#)

## INSTALL COMPONENTS

Now that you have the necessary files, you can begin the installation process. We will install

Apache, PHP, then the Java SDK and Tomcat.

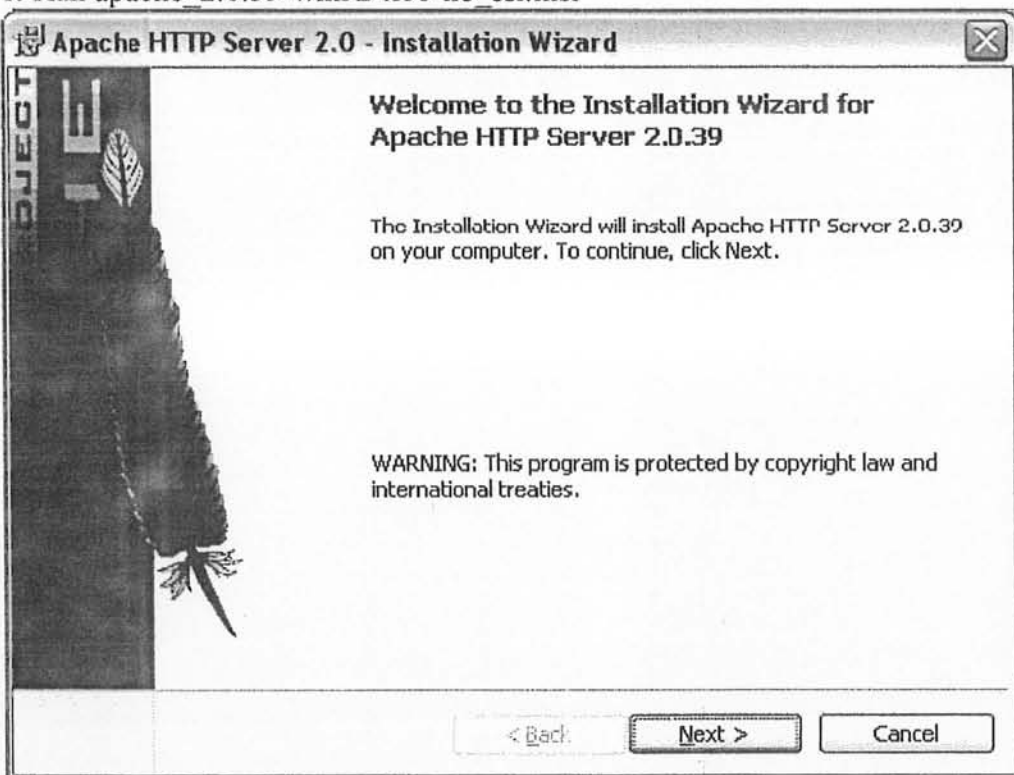
### **Note to Windows2000 users:**

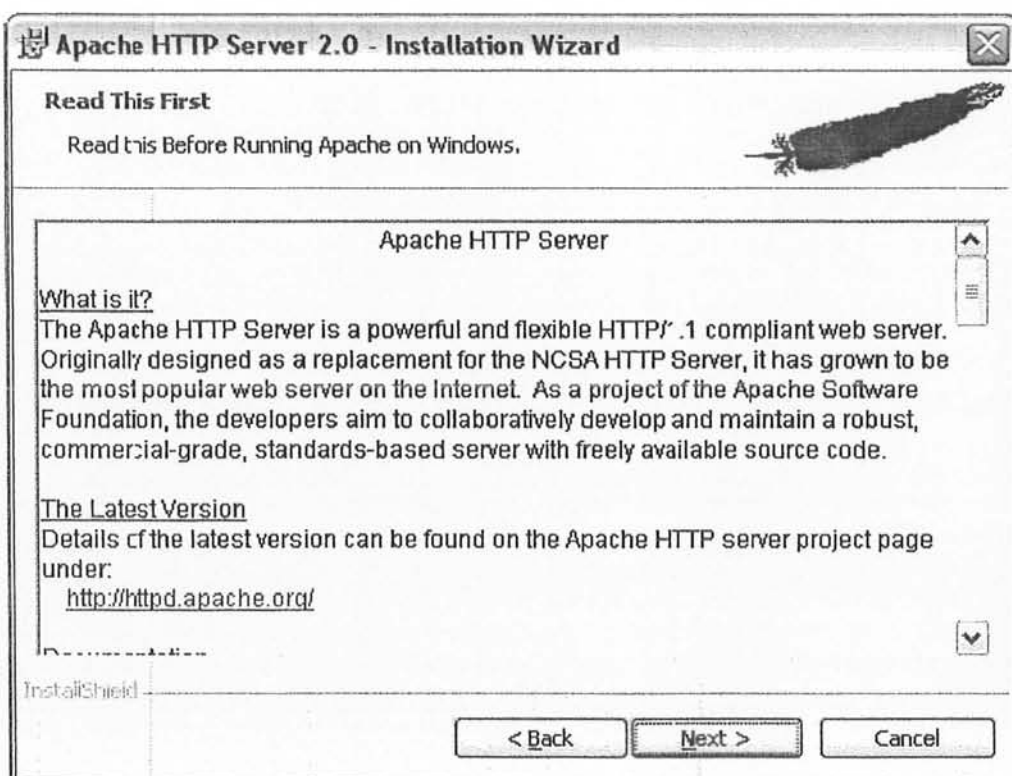
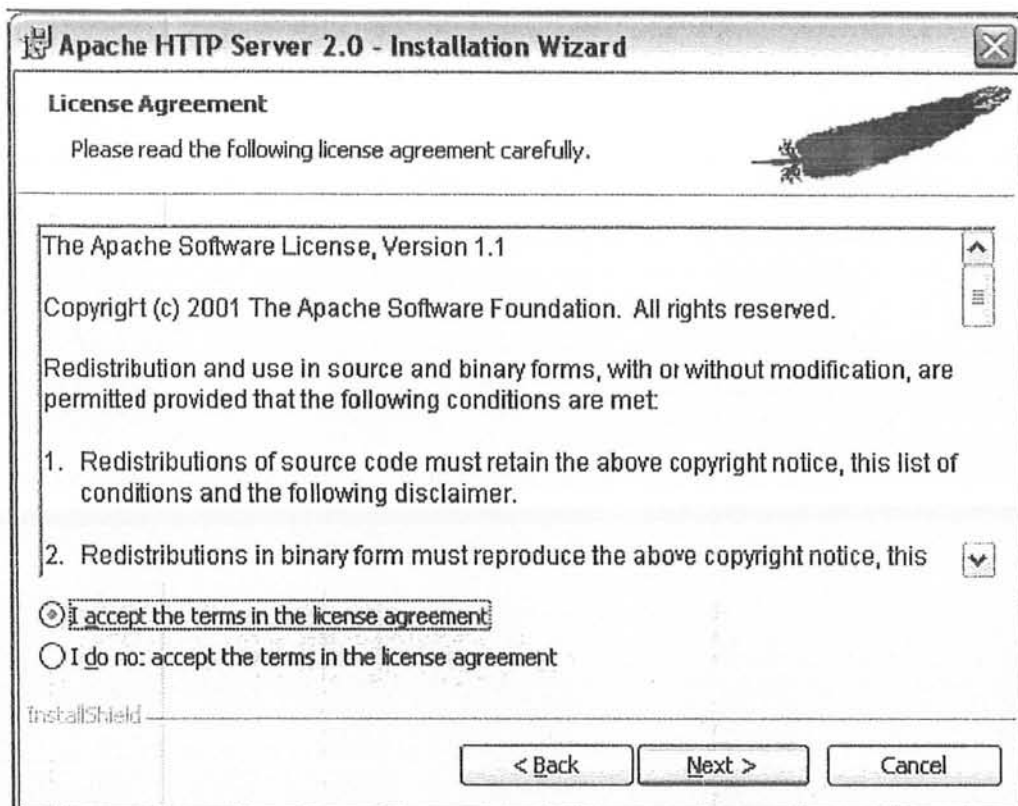
This guide should work with Windows2000 as well. Remember to replace C:\WINDOWS with

C:\WINNT anywhere in this document.

## Apache

1. Run apache\_2.0.39-win32-x86-no\_ssl.msi





2. Choose “Run as a service for All Users – Recommended” when prompted.

**Apache HTTP Server 2.0 - Installation Wizard**

**Server Information**

Please enter your server's information.

Network Domain (e.g. somenet.com)

Server Name (e.g. www.somenet.com):

Administrator's Email Address (e.g. webmaster@somenet.com):

Install Apache HTTP Server 2.0 programs and shortcuts for:

For All Users, on Port 80, as a Service – Recommended.

Only for the Current User, on Port 8080, when started Manually.

InstallShield

< Back    Next >    Cancel


**Apache HTTP Server 2.0 - Installation Wizard**

**Setup Type**


Choose the setup type that best suits your needs.

Please select a setup type.

**Typical**

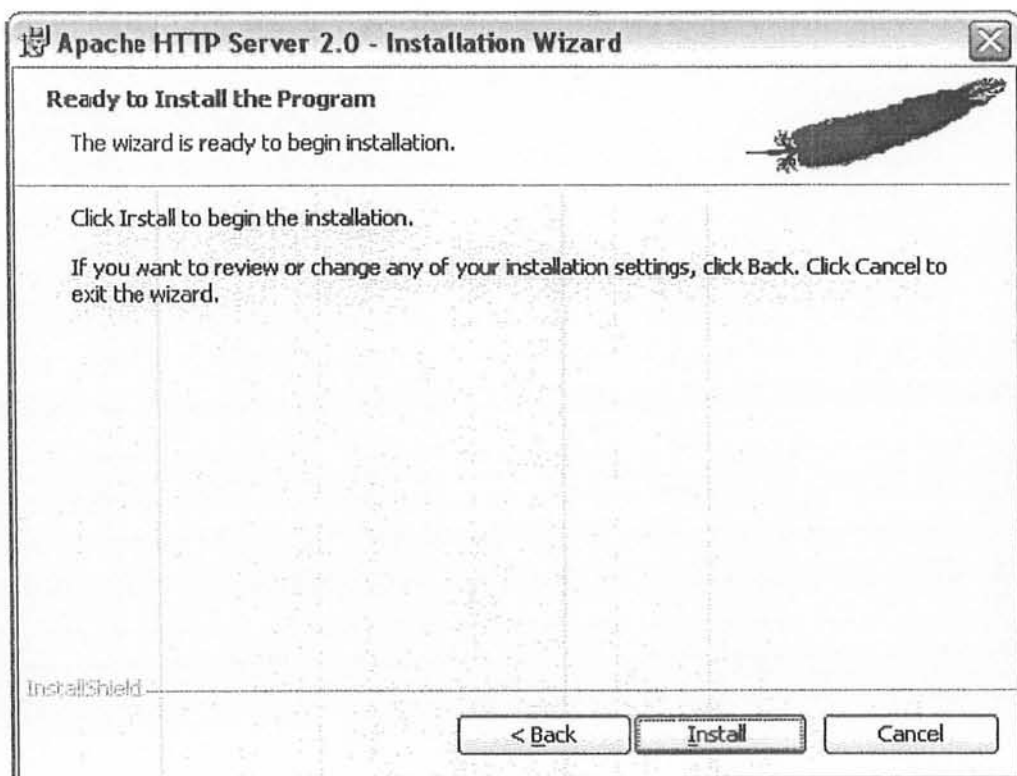
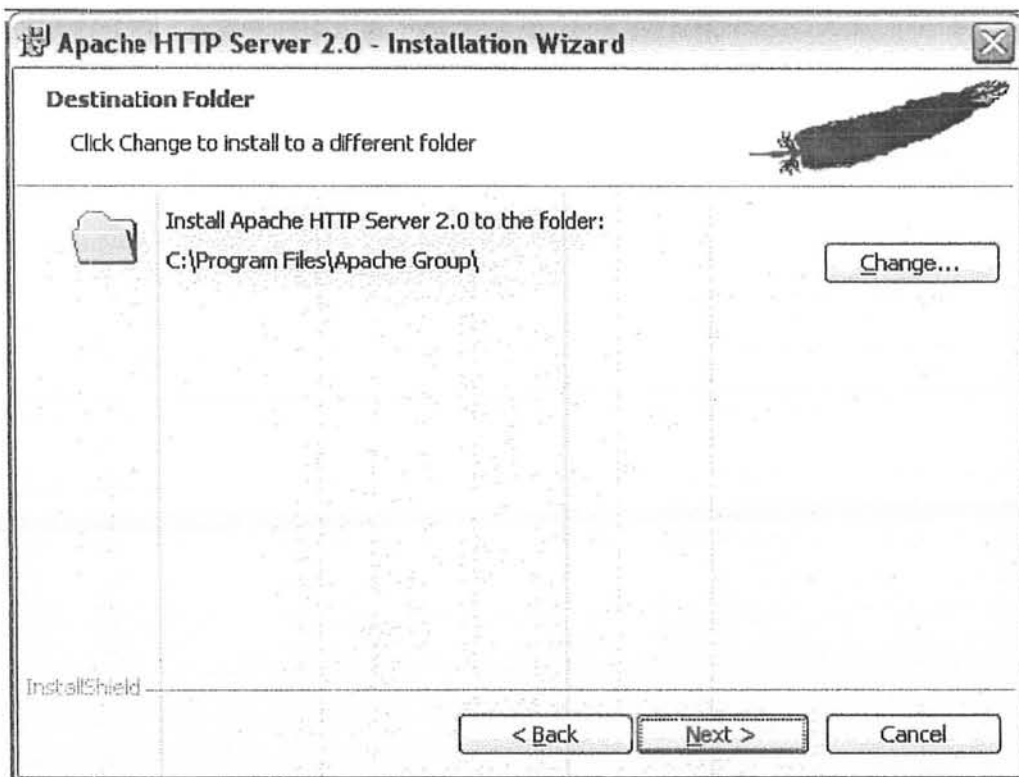
 Typical program features will be installed. (Headers and Libraries for compiling modules will not be installed.)

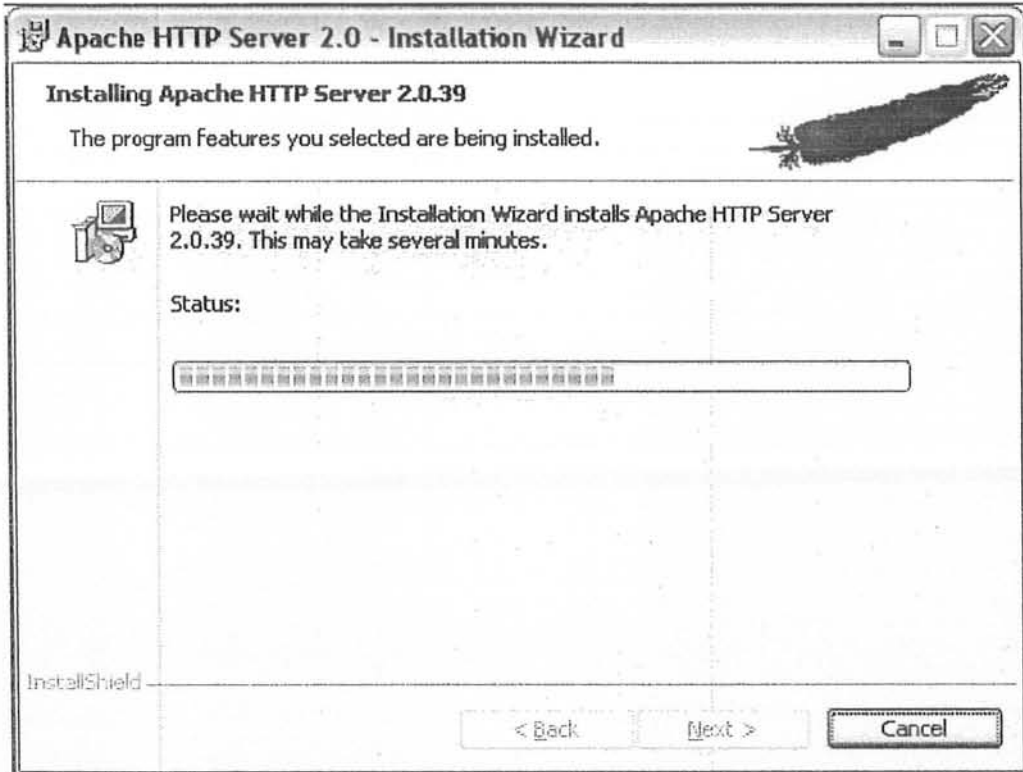
**Custom**

 Choose which program features you want installed and where they will be installed. Recommended for advanced users.

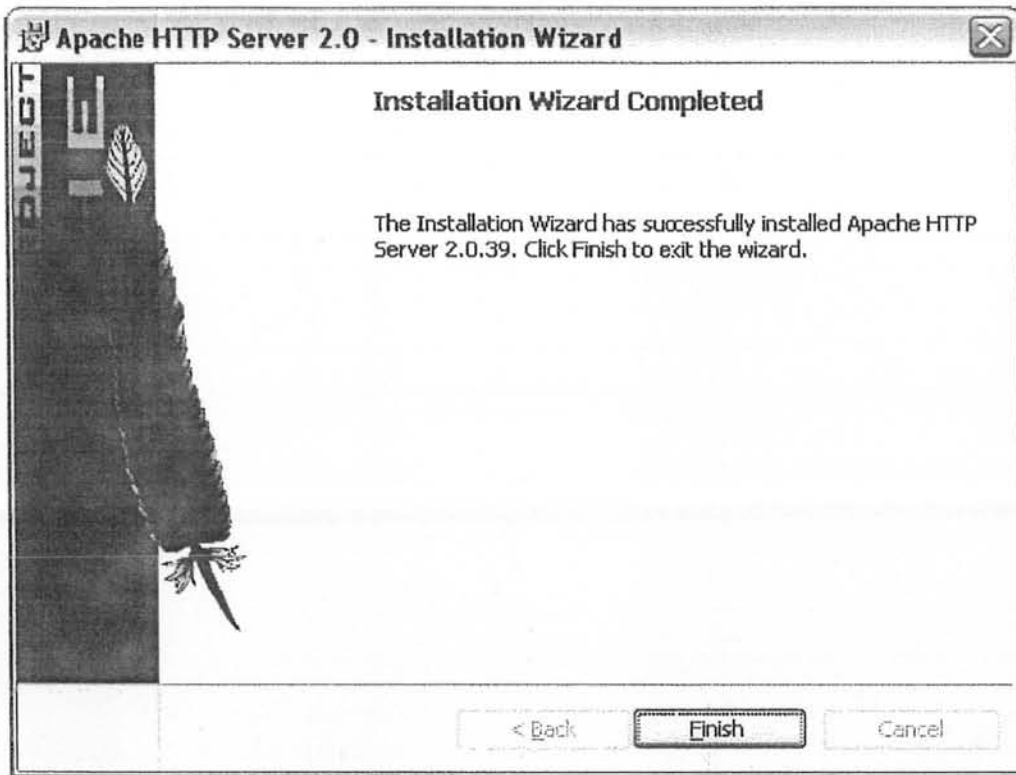
InstallShield

< Back    Next >    Cancel



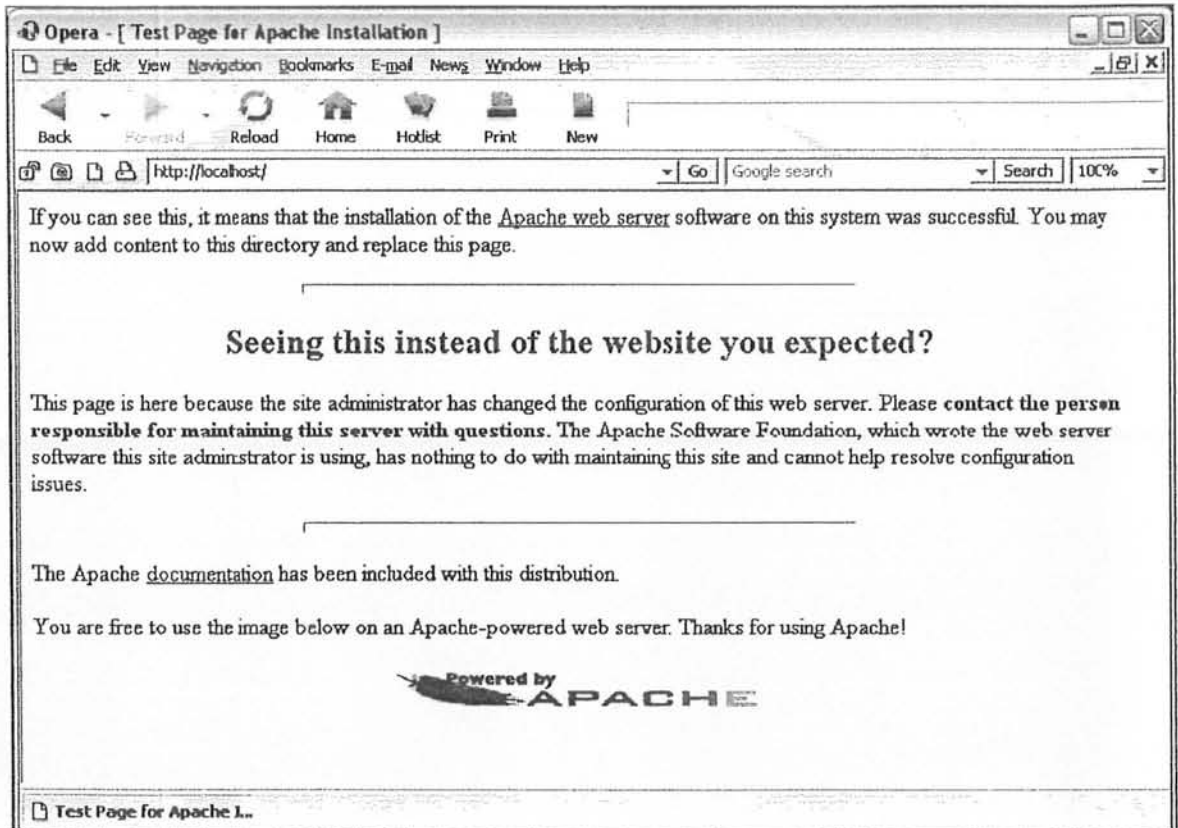


3. Finish the installation program.



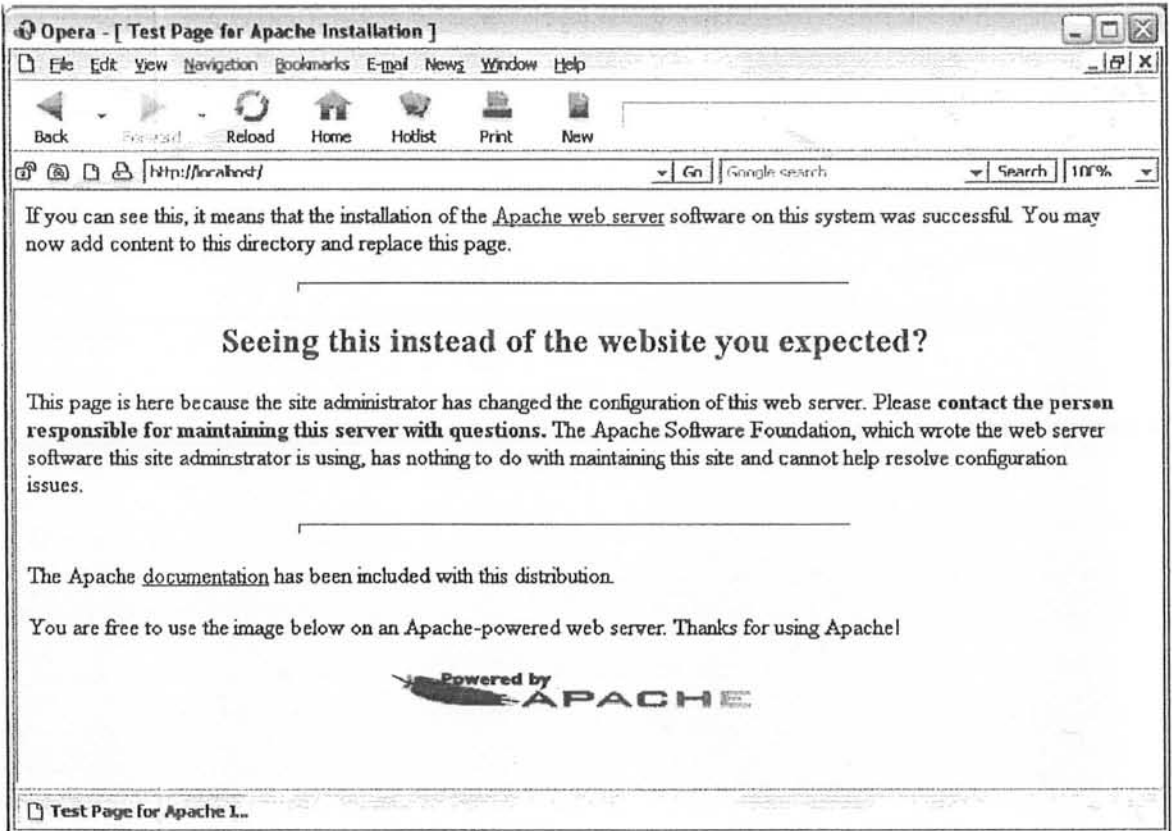


5. Test: Open `http://localhost` in a web browser.



## PHP

1. Extract `php-4.2.1-Win32.zip` into `C:\php`
2. Replace `c:\php\sapi\php4apache2.dll` with the one you downloaded.
3. Copy and rename `c:\php\php.ini-dist` to `C:\WINDOWS\php.ini`
4. Open `php.ini` and change the following:  
`include_path = ".;c:\php\includes"`  
`session.save_path = C:\Temp`  
`session.cookie_path = \`
5. Add `C:\php` to your PATH System Environment Variable.
6. Add the following lines to `httpd.conf`:  
`LoadModule php4_module c:\php\sapi\php4apache2.dll`  
`AddType application/x-httpd-php .php .php3 .phtml`
7. Test: Restart Apache & open `http://localhost` in a web browser. Hopefully you see the same thing as before:



8. **Test:** Create a new document called *phpinfo.php* and paste the following into it:

```
phpinfo.php
<?php
phpinfo();
?>
```

Save the document under C:\Program Files\Apache Group\Apache2\htdocs. Now open <http://localhost/phpinfo.php> in a web browser. You should see something like the following:


Opera - [ phpinfo() ]

File Edit View Navigation Bookmarks E-mail News Window Help

Back Forward Reload Home Hotlist Print New


http://localhost/phpinfo.php Go Google search Search 100%

# PHP Version 4.2.1



System	Windows NT 5.1 build 2600
Build Date	May 12 2002 23:51:56
Server API	Apache 2.0 Filter
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS\php.ini
Debug Build	no
Thread Safety	enabled

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v1.2.0, Copyright (c) 1998-2002 Zend Technologies

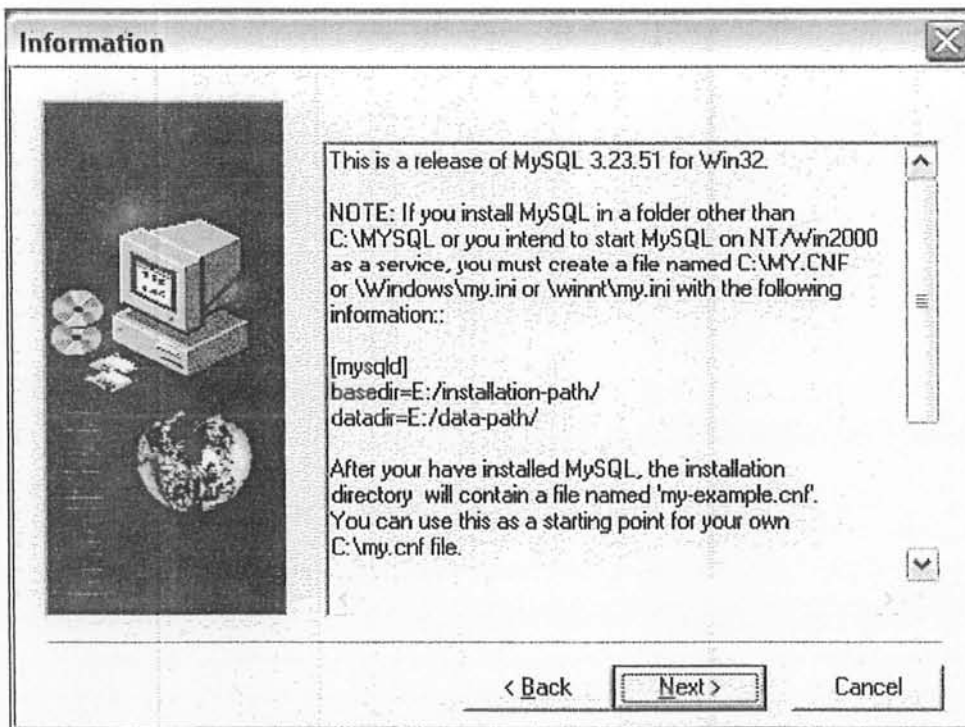
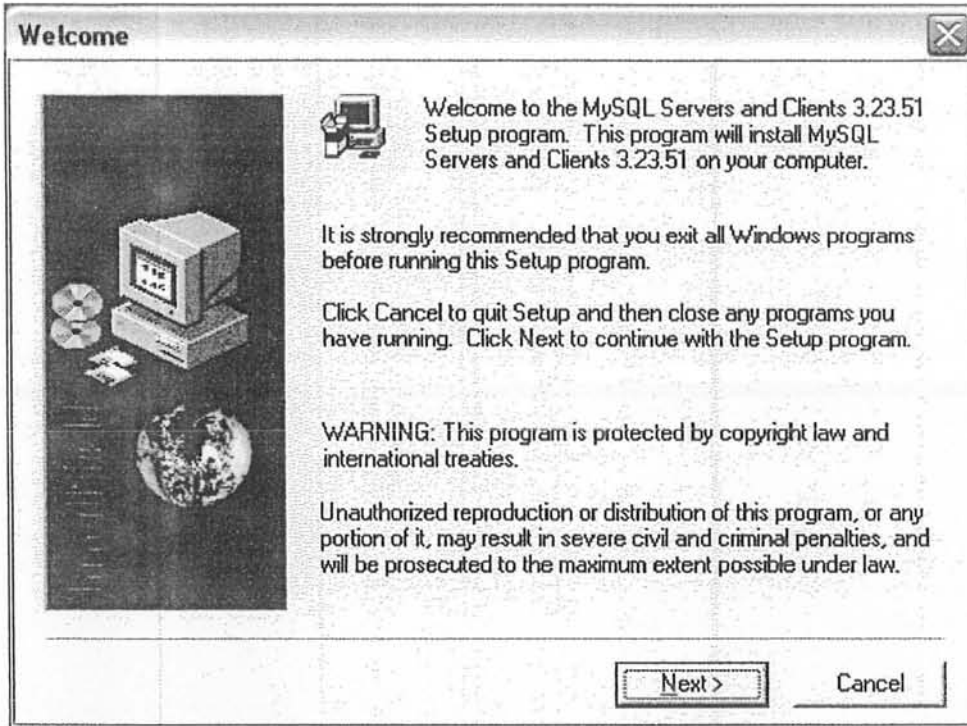


PHP 4 Credits

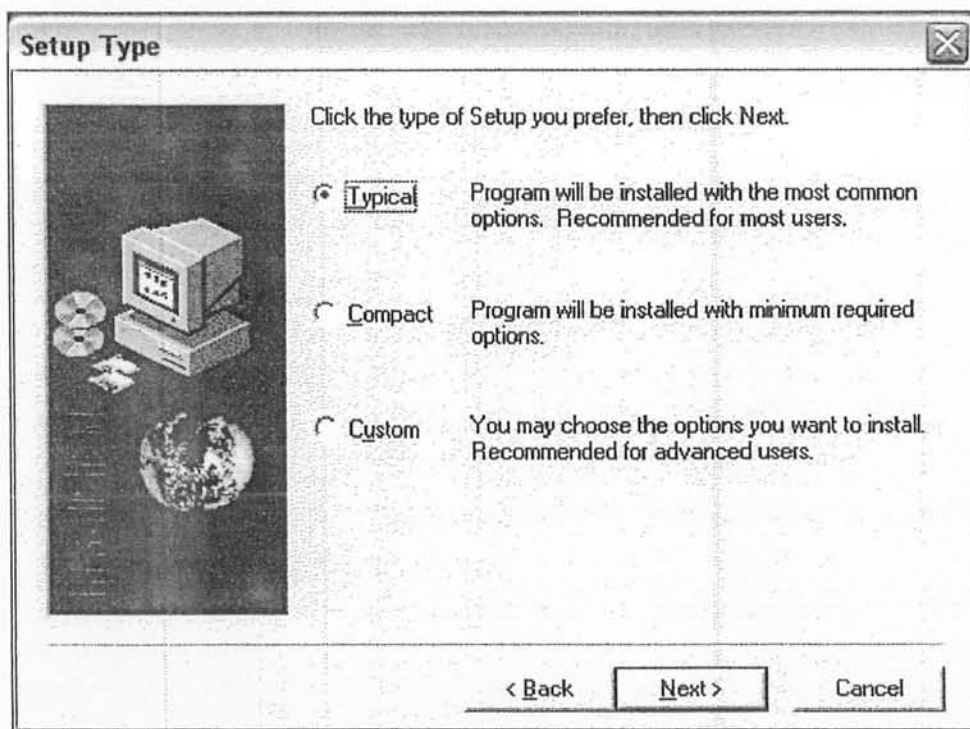
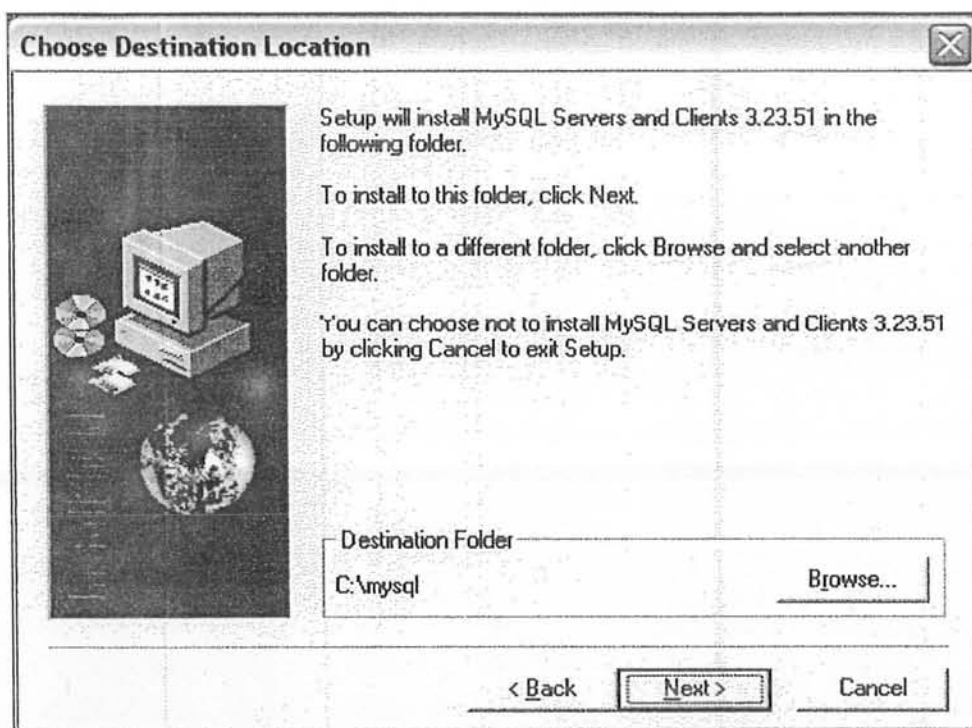
phpinfo()

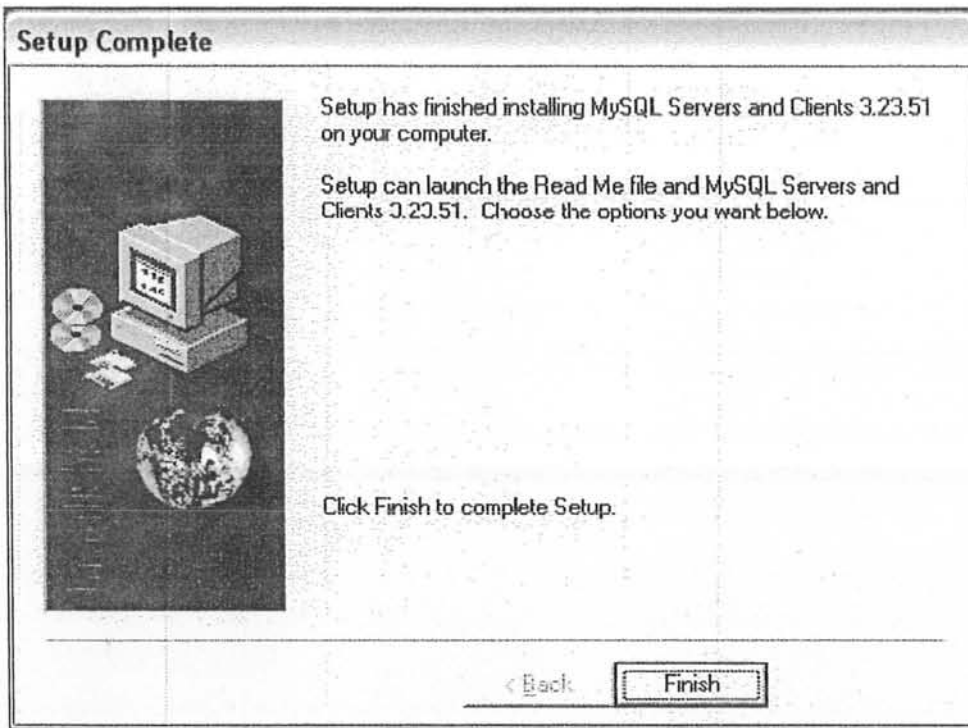
## MySQL

1. Extract mysql-3.23.51-win.zip to a temporary directory and run Setup.exe.



**Note:** Keep the default location. It will save you from frustration later on.





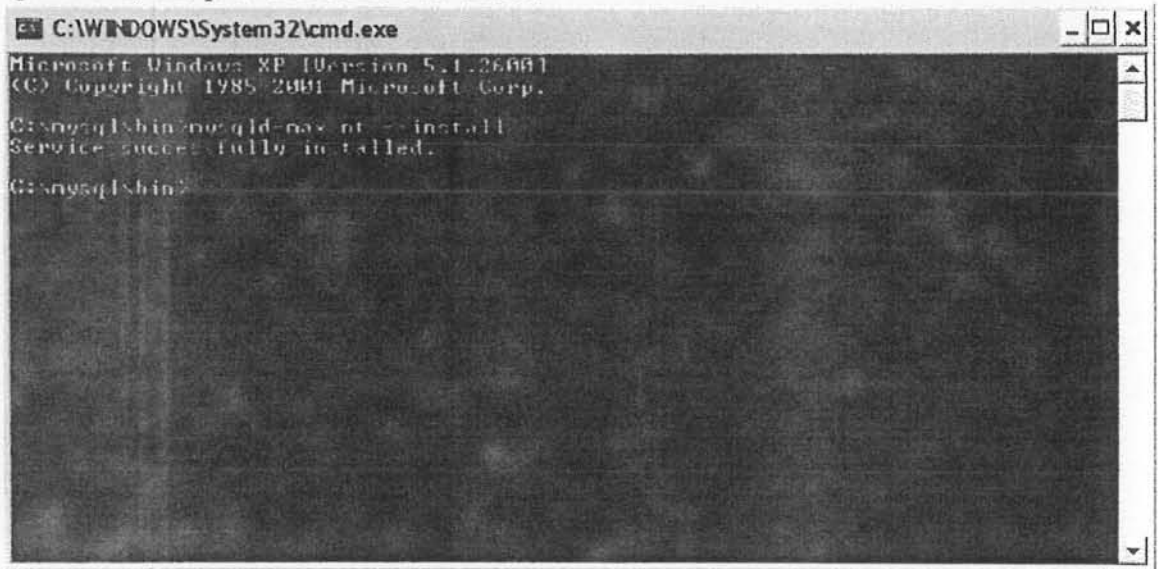
You may delete the temporary setup files at this time.

2. Open a command prompt and run the following command to install mysql as a Windows

Service:

```
C:\mysql\bin> mysql-d-max-nt --install
```

**Note:** It is advisable to close the Services utility while performing the --install or --remove operations, this prevents some odd errors.



3. **Test:** At the command prompt, type in NET START SQL. This should start up the service you just setup.

```

C:\WINDOWS\System32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\mysql\bin>mysql --install
Service successfully installed.

C:\mysql\bin>NET START MySQL
The MySQL service is starting.
The MySQL service was started successfully.

C:\mysql\bin>
  
```

**Note:** If you decided to install mysql somewhere besides the default c:\mysql, you may run into problems running it as a service. Check the *mysql.err* log file. It should tell you what's going on. It most likely has something to do with the my.cfg or my.ini files.

4. **Test:** Type in the command MYSQL SHOW . It should display the current databases.

```

C:\WINDOWS\System32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\mysql\bin>mysql --install
Service successfully installed.

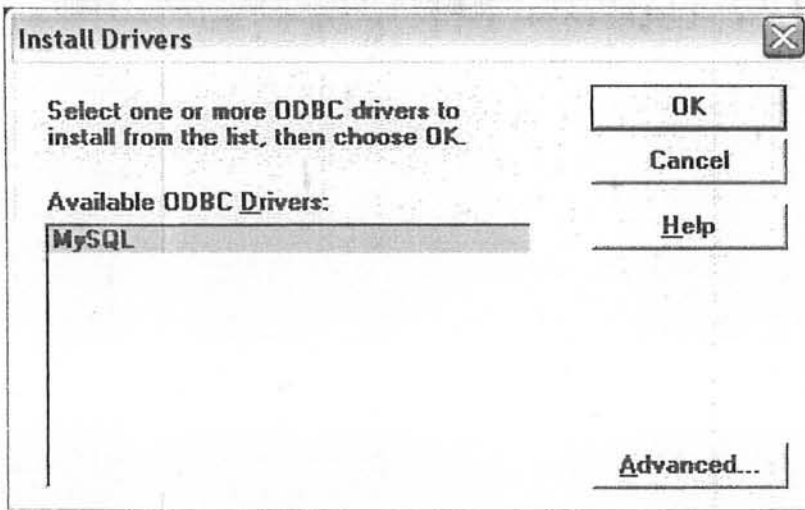
C:\mysql\bin>NET START MySQL
The MySQL service is starting.
The MySQL service was started successfully.

C:\mysql\bin>mysqlshow
+-----+
| Database |
+-----+
| mysql   |
| test    |
+-----+

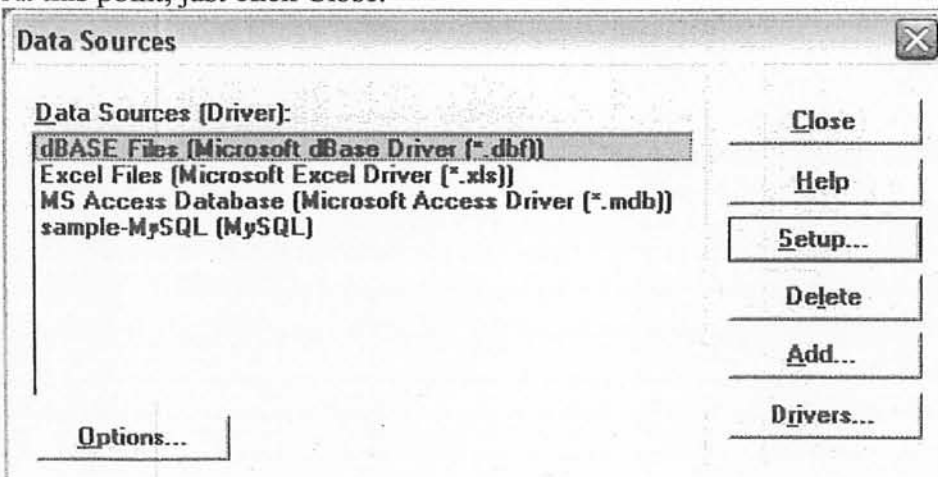
C:\mysql\bin>
C:\mysql\bin>
  
```

5. Extract myodbc-2.50.39-nt.zip to a temporary directory and run SETUP.EXE. (Don't

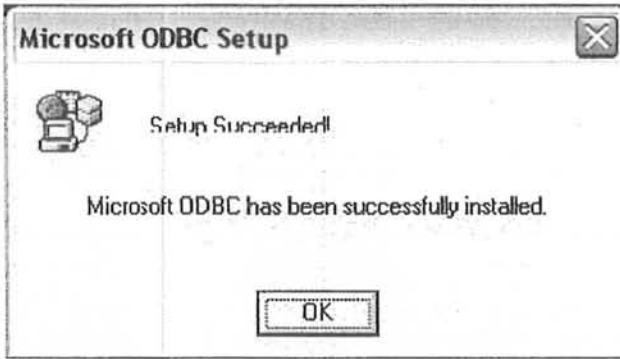
worry if the Installer looks like it's for Windows 3.1. It's normal.)



At this point, just click Close.







**Note:** You may delete the temporary installation directory at this time.

**APPENDIX E**

## PROJECT PLAN

Following is the project plan of the system

