**CHAPTER – 4**

**DETAILED DESIGN**

4.1 INTRODUCTION

Detailed design starts after the system design phase and system has been certified through the review. The goal of this phase is to develop the internal logic of each of the modules identified during system design. In the system design, the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for the modules. In other words, in system design attention is on what components are needed, while in the detailed design how the components can be implemented in the software is the issue, the design process for software system has two eves. At the first level focus is on deciding which modules are needed for the system, the specification of these modules and how the modules should be interconnected. This is called system design or top level design. In the specification of the module can be satisfied is decided. This design level is often called detailed design.

Because the detailed design is extension of system design, system design controls the major structural characteristics of the system. The system design has a major impact testability and modifiability of a system and impact its efficiency much of the design efforts for the designing software are spent creating the system design.

4.2 APPLICABLE DOCUMENTATION:

The detailed design refers the system document hence the first applicable document here is system design. Also we are referring the data structure. Hence second applicable document here is database design.

4.3 DATABASE DESIGN:

Data Design is used to manage large amount of information. The management of data involves both the definition of structure for storage and provision for the manipulation of information.

4.3.1 TABLE STRUCTURE:

Database is a collection of information and data systematically stored in tables the application “Automation of online inter collaged “consists of 13 tables. Table is a list of information organized into fields. Usually each field has a field name, data type with fixed length and description.

4.3.3.1 Field name tells the name of the field.

4.3.3.2 Data types property tells the type of data stored in that field and the length is

specified, so that the maximum length is set.

4.3.3.3 Attribute property is mentioned so that mandatory field are mentioned.

4.3.3.4 Description field tells the detail of field.

**4.3.1.1 Table Name: tblcollege**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | **Data Type** | **Constraints** | **Description** |
| College id | Int | Primary key | To store the ID of the college. |
| College name | Varchar(50) | Not Null | To store the name of the college. |
| Address | Varchar(MAX) | Not Null | To store the address of the college. |
| Contact no | Varchar(20) | Not Null | To store the contact of the college. |
| Contact no 2 | Varchar(20) | Not Null | To store the alternative no. |
| Email id | Varchar(20) | Not Null | To store the college email id. |
| photo path | Varchar(20) | Not Null | To store the photo path. |
| ID proof | Varchar(20) | Not Null | To store the certificates of the college. |
| Principle name | Varchar(20) | Not Null | To store the principle name of the college |
| Principle no | Varchar(20) | Not Null | To store the principle no. |
| Longitude | Varchar(20) | Not Null | To store the location of the college. |
| Colg\_password | Varchar(20) | Not Null | To store the college password. |
| Status | Varchar(20) | Not Null | To store the status of the college. |

**4.3.1.2 Table Name: tblevent**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | **Data Type** | **Constraints** | **Description** |
| Event id | Int | Primary key | To store the ID of the event. |
| Event name | Varchar(MAX) | Not Null | To store the name of the event. |
| Event description | Text | Not Null | To store the event description. |
| Event date | Date | Not Null | To store the event date. |
| Event photo | Varchar(20) | Not Null | To store the image of event. |
| Clog id | Int | Not Null | To store the college id. |
| Event type | Varchar(20) | Not Null | To store the event type. |
| Status | Varchar(20) | Not Null | To store the status of the event. |

**4.3.1.3 Table Name: tbldetail**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | **Data Type** | **Constraints** | **Description** |
| Event id | Int | Primary key | To store event id. |
| Event schedule | Text | Not Null | To store event schedule. |
| Event result | Text | Not Null | To store event result. |

**4.3.1.4 Table Name: tblelection**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | **Data Type** | **Constraints** | **Description** |
| Election id | Int | identity | To store the ID of the election. |
| Election post | Varchar(250) | Not Null | To store the election post. |
| Election description | Text | Not Null | To store the election description. |
| Election date | Date | Not Null | To store the election date. |
| Result date  2222222222222222222222222222222222 | Date | Not Null | To store the result date. |
| Election year | Varchar(20) | Not Null | To store the election year. |

**4.3.1.5 Table Name: tblelectiondetail**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | **Data Type** | **Constraints** | **Description** |
| Post id | Int | Primary key | To store the ID of the post. |
| Election id | Int | Not Null | To store election id. |
| Post name | Varchar(150) | Not Null | To store post name. |
| Post description | Text | Not Null | To store post description. |
| Status | Varchar(20) | Not Null | To store status of the election detail. |

**4.3.1.6 Table Name: tblcandidate**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | **Data Type** | **Constraints** | **Description** |
| candidate id | Int | Primary key | To store candidate id. |
| Candidate name | Varchar(250) | Not Null | To store the candidate name. |
| Contact no | Varchar(20) | Not Null | To store candidate no. |
| Candidate photo | Varchar(20) | Not Null | To store candidate photo. |
| Candidate ID card | Varchar(20) | Not Null | To store candidate id card. |
| clog id | Int | Not Null | To store candidate clog id. |
| Post id | Int | Not Null | To store candidate post id. |

**4.3.1.7 Table Name: tblresult**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | **Data Type** | **Constraints** | **Description** |
| ID | Int | Primary key | To store id. |
| Post id | Int | Not Null | To store post id. |
| Student id | Int | Not null | To store student id. |

**4.3.1.8 Table Name: tblfund**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | **Data Type** | **Constraints** | **Description** |
| Fund id | Int | Identity, primary key | To store fund id. |
| Fund title | Varchar(20) | Not Null | To store fund detail. description. |
| Fund description | Text | Not Null | To store fund description. |
| Fund amount | Double(10,2) | Not Null | To store fund amount. |
| Fund date | Date | Not Null | To store fund date. |
| Student id | Int | Not Null | To store the status of the. election. |
| Fund proof | Varchar(max) | Not Null | To show the proof for fund required. |
| Status | Varchar(20) | Not Null | To store status of fund |

**4.3.1.9 Table Name: tblcourse**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | **Data Type** | **Constraints** | **Description** |
| Course id | Int | Identity, primary key | To store course ID. |
| Course name | Varchar(250) | Not Null | To store the course name. |

4.3.1.0.0 Table Name: security

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | **Data Type** | **Constraints** | **Description** |
| security id | Int | Primary key | To store security id. |
| clog id | Int | Not Null | To store college id. |
| Post name | Varchar(150) | Not Null | To store post name. |
| Candidate name | Text | Not Null | To store candidate name. |
| Year | Varchar(20) | Not Null | To store year. |

**4.4 ER DIAGRAM**

**4.4.1 ELEMENTS OF ER-MODEL**

**4.4.1.1Entities**

The first basic concept in E-R diagram is the entity. They are the items that have a definable existence in the World-Persons, Jobs, Salaries etc. They are logical categories of items rather than specific items.

Entities can be further characterized as strong or weak. A strong entity is a well-defined without reference to any other entity in the model, where as a weak one requires reference to something else in order for its individual instances to be meaningful or to be identifiable.

**4.4.1.2 Relationships**

Entities have relationships to one another. Relationships have a couple of important features. One of these is the degree of the relationship. This refers to the number of entities that participate in the relationship.

Another important characteristic of relationships is their connectivity, which refers to the mapping of instances of entity A to entity B.

There are 3 possibilities:

One -to –one (written 1:1). For each A there is no more than one B.

One –to-many (written 1: n). For each A, there are zero or more BS, but for each B, there is only one A.

Many –to-many (written m: n). For each Am there are zero or more BS, and for each B, there are zero or more As.

4.4.1.3 Attributes

These can apply to either entities or relationships. An attribute is a traceable characteristic of the entity or relationship. An attribute can be single- valued, meaning it has one value per instance of the entity or multi-valued, meaning it can have several such values.

**SYMBOLS USED IN ER-DIAGRAM**

|  |  |
| --- | --- |
| SYMOLS | **NAME** |
|  | ENTITY |
|  | RELATIONSHIP |
|  | ATTRIBUTE |
|  | KEY ATTRIBUTE |
|  | CARDINALITY  RATIO 1: N FOR E1:E2 IN R  IN R |
|  | CARDINALITY  RATIO M: N FOR E1:E2  IN R |

**ER-DIAGRAM**

1

ADMIN

VERFIES

**1 1**

PUBLISHED

ADMIN NAME

**1**

COLLEGE

EVENT NOTICE

**1 1**

HAS

ELECTION

REGISTER

EVENT DATE

**1 1**

CONTAINS

**N**

**N**

VIEW

STUDENT

ELECTION DETAIL

**1**

HAsS

REGISTER

voter

**N**

VIEW

CANDIDATE

RESULT

FUNDPAY

HAS

FUND

**4.6.1 System Software Architecture**

**Web server (IIS)**

**SQL server**

**ASP.NET**

**Local host**

**SQL query**

**Database**

Send

Receive

Retrieve

Add

4.6.1.1 The software has following software, database and other requirements.

4.6.1.2 Operating system: Windows XP or higher.

4.6.1.3 Frontend: ASP.Net.

4.6.1.4 Backend: SQL Server 2008.

**4.6.2 SYSTEM TECHNICAL ARCHITECTURE.**

SQL SERVER

IIS SERVER

Sql Server

Inter College Event

DATABASE COMPILER

DATABASE SERVER

S

DATABASE ADMINISTRTOR

**4.6.3 SYSTEM HARDWARE ARCHITECTURE:**

**LOCAL AREA NETWORK**

USER

ADMIN SERVER

DATABASE SERVER

4.6.3.1 The system should have these hardware requirements.

4.6.3.2 Processor: Intel dual core or above

4.6.3.3 Processor Speed: 1.0GHZ or above

4.6.3.4 RAM: 1 GB RAM or above

4.6.3.5 Hard disk: 20 GB hard disk or above

4.6.3.6 Operating system: Windows XP or above

4.6.3.7 Monitor: VGA or higher resolution.

**4.6 DETAILED DESGIN**

**4.6.1 FLOW CHART**

Flow chart is a graphic picture of the logical steps and sequence involved in a procedure or **a** program.

**Some of the symbol used in drawing a flowchart-**

|  |  |
| --- | --- |
| **Terminator** | **Terminator or oval symbol is used to define the beginning of a flow chart and to show the termination point of flow chart.** |
| **Flow Lines** | **Flow lines connect the other symbol in flow chart. The arrows with the sequence of steps to follow.** |
| **Input/output Box** | **A parallelogram is used represent the Input and Output operation.** |
| **Decisions Box** | **The diamond symbol is used when want the program to decide between alternate courses of action.** |
| **Process Box** | **A rectangle or process symbol is generally used to represent any processing operation.** |

4.6.1.1 Login

Input: User name and Password

Procedural details: If the user name and password is correct then it goes to theHomepage otherwise display error message.

File Input/output Interfaces:

****

4.6.1.2 COLLEGE ADMIN

Input: User and Password

Procedural details: If the user name and password is correct then it goes to the Homepageotherwise display error message.

File Input/output Interfaces:

****

4.6.1.3 STUDENT

Input: User and Password

Procedural details: If the user name and password is correct then it goes to the Homepageotherwise display error message.

File Input/output Interfaces:

****

4.6.2 PESUDO CODES(ALGORTHIM)

4.6.1 LOGIN

Input username, password,

If user name exits, then

Check for whether password matches

If password matches, then

Insert it to data base then login

Else

display error message

End if

End if

4.6.2 COLLEGE MODULE

Input college name, address contact no, email id

If its exits, then

Insert into database

Else display error message

End if

4.6.3 STUDENT MODULE

Input student name, address contact no, email id

If its exits, then

Insert into database

Else display error message

End if

4.6.4 EVENT MODULE

Input Event name, event date event detail, email id

If its exits, then

Insert into database

Else display error message

End if

4.6.2 EVENT SCHEDULE MODULE

Input Event location, event description, event location

If its exits, then

Insert into database

Else display error message

End if

**4.7 User Interface Design:**

**Label:**

**Description**: The label provides a mnemonic name that indicates what type of information

Is stored in a media. The name of label starts with lab.

Example: Username, Password etc.

**Textbox:**

**Description:** Textbox is a rectangular area on the screen where you can enter text.

There are two types of textboxes, text field and text area. the name of the textbox starts with txt.

**Text Field:** It is a small box that allows you to enter a single line of text.

Example –username, email etc.

**Text Area:** It is a larger box that allows you to enter multiple lines of text.

Example address, description etc.

**Dropdown list:**

**Description** dropdown list is a user input device in which the user can select an option

From a dropdown list or type in a value into a textbox so that the appropriate choice can be selected.

Example: select subject etc.

**Button:**

**Description**: button is a small outline area in a dialog box that you can click to select an

option or command.

Example: Add update etc.

**Radio button:**

**Description:** It is a group of buttons of which only one can be on at a time. When you select one button all the others are automatically deselected.

Example: Gender.