

STUDENT ATTENDANCE MANAGEMENT SYSTEM



**2022-2023
A PROJECT REPORT**

Submitted by

KAVANA NAVADA	201231522269
KAVANA	201231522268
SANGEETHA	201231522222

In partial fulfillment for the award of the degree of
BACHELOR OF COMPUTER APPLICATIONS

Under the guidance of
Mrs. ASHWITHA
Lecturer
Department of Computer Science

**BHANDARKARS' ARTS AND SCIENCE COLLEGE
KUNDAPURA**

MANGALORE UNIVERSITY

BHANDARKARS' ARTS AND SCIENCE COLLEGE
KUNDAPURA -576201



Department of Computer Science

Certificate

Certified that the project work entitled

.....**STUDENT ATTENDANCE MANAGEMENT SYSTEM**.....
is a bonafied work carried out by

.....**KAVANA NAVADA, KAVANA, SANGEETHA**.....
in partial fulfilment for the award of degree of Bachelor of Computer Applications
of the Mangalore University during the year 2022-23. The project report has been
approved as it satisfies the academic requirements of project work prescribed for
the Bachelor of Computer Applications.

Project Guide

Head of the Department
Department of Computer Science

Name of the Student:

KAVANA NAVADA, KAVANA, SANGEETHA

Roll Number:

200653, 200681, 200666

University Register Number:

201231522269, 201231522268, 201231522222

Examiners: 1.

2.



BHANDARKARS' ARTS AND SCIENCE COLLEGE

KUNDAPURA-576201

ATTENDANCE CERTIFICATE

This is to certify that the following students of sixth semester BCA has got adequate attendance in the project work as stipulated by Mangalore University in BCA regulations.

KAVANA NAVADA	:201231522269
KAVANA	:201231522268
SANGEETHA	:201231522222

PRINCIPAL

DECLARATION

We hereby declare that this project work entitled “STUDENT ATTENDNACE MANAGEMENT SYSTEM” has been prepared by us during the year 2022–23 under the guidance of Mrs.ASHWITHA, Department of Computer Science, Bhandarkars’ Arts and Science College, Kundapura in the partial fulfillment of BCA degree prescribed by the Mangalore University.

We also declare that this project is the outcome of our own effort, that it has not been submitted to any other university for the award of any degree.

Date:

KAVANA NAVADA

Reg. Number: 201231522269

KAVANA

Reg. Number: 201231522268

SANGEETHA

Reg. Number: 201231522222

ACKNOWLEDGEMENT

It gives us an immense pleasure to present the project report on “**STUDENT ATTENDANCE MANAGEMENT SYSTEM**”

We consider it privilege to express our profound gratitude and respect to those who guided us in the successful completion of this project.

We would like to express our gratitude towards our project guide **Mrs. Ashwita**, lecturer in Department of Computer Science for his valuable guidance throughout the period of project.

We would like to express our gratitude to **Mrs. Vijayalakshmi N Shetty, Head of the Department, Computer Science** for her kind concern and encouragement during the completion of our project.

We are sincerely thankful to **Dr. N. P. Narayan Shetty, Principal of Bhandakars' Arts and science college Kundapura**, for granting an opportunity to work our project.

Our sincere thanks for all faculty members of Computer Science Department. We are thankful to our parents for their encouragement towards the project. Last but not the least, we whole heartly appreciates the co-operation of our friends.

Thank You,

Project Team,

Kavana Navada

Kavana

Sangeetha

TABLE OF CONTENTS

SNO	TITLE	PAGE NO
1	INTRODUCTION	1-7
	1.1. Introduction of the System	1
	1.1.1. Project Title	1
	1.1.2. Category	1
	1.1.3. Overview	1
	1.2. Background	1
	1.2.1. Introduction of the Company	1
	1.2.2. Brief note on Existing System	1
	1.3. Objective of the System	2
	1.4. Scope of the System	2
	1.5. Structure of the System	3
	1.5.1. Analysis	3
	1.5.2. Module Description	3
	1.5.2.1. Admin	3
	1.5.2.2.1. Login	3
	1.5.2.1.2. Dashboard	3
	1.5.2.1.2.1. Total No. of Students	3
	1.5.2.1.2.2. Total No. of Staff	3
	1.5.2.1.2.3 Total No. of Course	3
	1.5.2.1.3. Student Management	4
	1.5.2.1.4. Staff Management	4
	1.5.2.1.5. Subject Management	4
	1.5.2.1.6. Attendance Report	4
	1.5.2.1.7. My Profile	4
	1.5.2.1.8. Logout	4
	1.5.2.2. Staff	4

	1.5.2.2.1. Login	5
	1.5.2.2.2. Today's Attendance	5
	1.5.2.2.3. Update Attendance	5
	1.5.2.2.4. Attendance Report	5
	1.5.2.2.5. My Profile	5
	1.5.2.2.6. Logout	
	1.5.2.4. Student	5
	1.5.2.4.1. Login	5
	1.5.2.4.2. View Attendance	5
	1.5.2.4.3. My Profile	6
	1.5.4.2.4.3. Logout	6
	1.6. System Design Architecture	6
	1.7. End User	7
	1.8 Software/Hardware need for the development	7
	1.8.1 The following hardware specification is used to develop this project	7
	1.8.2 The following software specification is used to develop this project	7
	1.9 Software/Hardware need for the implementation	7
	1.9.1 Hardware needed for the implementation	7
	1.9.2 Software needed for the implementation	7
2	SOFTWARE REQUIREMENT SPECIFICATION	8-17
	2.1. Introduction	8
	2.2. Overall Description	8
	2.2.1. Product Perspective	8
	2.2.1.1. System Interfaces	8
	2.2.1.2. User Interfaces	8
	2.2.1.3. Hardware Interfaces	9
	2.2.1.5. Communication Interfaces	9
	2.2.1.6. Interface with Server	9
	2.2.2. Product Function	9
	2.2.3. User Characteristics	9
	2.2.4. General Constraints	9
	2.2.5. Assumptions and Dependencies	10

	2.3. Special Requirements	10
	2.4. Functional Requirements	10
	2.4.1. Admin	10
	2.4.1.1. Login	10
	2.4.1.2. Dashboard	10
	2.4.1.2.1. No. of Students	10
	2.4.1.2.2. No. of Staffs	11
	2.4.1.2.3. No. of Courses	11
	2.4.1.3. Student Management	11
	2.4.1.3.1. Add	11
	2.4.1.3.2. Update	11
	2.4.1.3.3. Delete	11
	2.4.1.3.4. View	12
	2.4.1.3.4.1. Search	12
	2.4.1.4 Staff Management	12
	2.4.1.4.1. Add	12
	2.4.1.4.2. Update	12
	2.4.1.4.3. Delete	12
	2.4.1.4.4. View	13
	2.4.1.6. Attendance Report	13
	2.4.1.7. My Profile	13
	2.4.1.8. Logout	13
	2.4.2. Staff	13
	2.4.2.1. Login	13
	2.4.2.2. Today's Attendance Entry	13
	2.4.2.3. Update Attendance	14
	2.4.2.4. Attendance Report	14
	2.4.2.5. My Profile	14
	2.4.2.6. Logout	14
	2.4.3. Student	14
	2.4.3.1. Login	14
	2.4.3.2. View Attendance	15
	2.4.3.3. My Profile	15

	2.4.3.4. Logout	15
	2.5. Design Constraints	15
	2.5.1. Hardware Constraints	15
	2.5.2. Software Constraints	15
	2.5.3. Fault Tolerance	16
	2.5.4. Security	16
	2.5.5. Standard Compliance	16
	2.6. System Attributes	17
	2.7. Other Requirements	17
3	SYSTEM DESIGN	18-55
	3.1. Introduction	18
	3.2. Assumption and Constraints	18
	3.3. Functional Decomposition	18
	3.3.1. System Software Architecture	19
	3.3.2. System Technical Architecture	20
	3.3.3. System Hardware Architecture	20
	3.3.4. External Interfaces	20
	3.4. Description of Programs	20
	3.4.1. Context Flow Diagram	20
	3.4.2. Data Flow Diagram	21
	3.4.2.1. DFD FOR LEVEL-0	22
	3.5. Description of components	23
	3.5.1. Admin	23
	3.5.1.1. Login	23
	3.5.1.1.1. Input	23
	3.5.1.1.2. Process	23
	3.5.1.1.3. Output	24
	3.5.1.1.4. Interface with other functional components	24
	3.5.1.1.5. Resource Allocation	24
	3.5.1.1.6. User Interface	24
	3.5.1.2. Dashboard	24
	3.5.1.2.1. Total No. of Students	25

	3.5.1.2.1.1. Input	25
	3.5.1.2.1.2. Process	25
	3.5.1.2.1.3. Output	25
	3.5.1.2.1.4. Interface with other functional components	25
	3.5.1.2.1.5. Resource Allocation	25
	3.5.1.2.1.6. User Interface	26
	3.5.1.2.2. Total No. of Staffs	26
	3.5.1.2.2.1. Input	26
	3.5.1.2.2.2. Process	26
	3.5.1.2.2.3. Output	26
	3.5.1.2.2.4. Interface with other functional components	26
	3.5.1.2.2.5. Resource Allocation	26
	3.5.1.2.1.6. User Interface	26
	3.5.1.2.3. Total No. of Courses	27
	3.5.1.2.3.1. Input	27
	3.5.1.2.3.2. Process	27
	3.5.1.2.3.3. Output	27
	3.5.1.2.3.4. Interface with other functional components	27
	3.5.1.2.3.5. Resource Allocation	27
	3.5.1.2.3.6. User Interface	27
	3.5.1.3 Student Management	28
	3.5.1.3.1. Add	28
	3.5.1.3.1.1. Input	28
	3.5.1.3.1.2. Process	28
	3.5.1.3.1.3. Output	28
	3.5.1.3.1.4. Interface with other functional components	28
	3.5.1.3.1.5. Resource Allocation	28
	3.5.1.3.1.6. User Interface	28
	3.5.1.3.2. Update	29
	3.5.1.3.2.1. Input	29
	3.5.1.3.2.2. Process	29
	3.5.1.3.2.3. Output	29
	3.5.1.3.2.4. Interface with other functional components	29

	3.5.1.3.2.5. Resource Allocation	30
	3.5.1.3.2.6. User Interface	30
	3.5.1.3.3. Delete	30
	3.5.1.3.3.1. Input	30
	3.5.1.3.3.2. Process	30
	3.5.1.3.3.3. Output	30
	3.5.1.3.3.4. Interface with other functional components	31
	3.5.1.3.3.5. Resource Allocation	31
	3.5.1.3.3.6. User Interface	31
	3.5.1.3.4. View	31
	3.5.1.3.4.1. Input	31
	3.5.1.3.4.2. Process	32
	3.5.1.3.4.3. Output	32
	3.5.1.3.4.4. Interface with other functional components	32
	3.5.1.3.4.5. Resource Allocation	32
	3.5.1.3.4.6. User Interface	32
	3.5.1.4. Staff Management	33
	3.5.1.4.1. Add	33
	3.5.1.4.1.1. Input	33
	3.5.1.4.1.2. Process	33
	3.5.1.4.1.3. Output	33
	3.5.1.4.1.4. Interface with other functional components	33
	3.5.1.4.1.5. Resource Allocation	33
	3.5.1.4.1.6. User Interface	33
	3.5.1.4.2. Update	34
	3.5.1.4.2.1. Input	34
	3.5.1.4.2.2. Process	34
	3.5.1.4.2.3. Output	34
	3.5.1.4.2.4. Interface with other functional components	35
	3.5.1.4.2.5. Resource Allocation	35
	3.5.1.4.2.6. User Interface	35
	3.5.1.4.3. Delete	35

	3.5.1.4.3.1. Input	35
	3.5.1.4.3.2. Process	35
	3.5.1.4.3.3. Output	35
	3.5.1.4.3.4. Interface with other functional components	36
	3.5.1.4.3.5. Resource Allocation	36
	3.5.1.4.3.6. User Interface	36
	3.5.1.4.4. View	36
	3.5.1.4.4.1. Input	36
	3.5.1.4.4.2. Process	36
	3.5.1.4.4.3. Output	37
	3.5.1.4.4.4. Interface with other functional components	37
	3.5.1.4.4.5. Resource Allocation	37
	3.5.1.4.4.6. User Interface	37
	3.5.1.5. Subject Management	38
	3.5.1.5.1. Add	38
	3.5.1.5.1.1. Input	38
	3.5.1.5.1.2. Process	38
	3.5.1.5.1.3. Output	38
	3.5.1.5.1.4. Interface with other functional components	38
	3.5.1.5.1.5. Resource Allocation	38
	3.5.1.5.1.6. User Interface	38
	3.5.1.5.2. Update	39
	3.5.1.5.2.1. Input	39
	3.5.1.5.2.2. Process	39
	3.5.1.5.2.3. Output	39
	3.5.1.5.2.4. Interface with other functional components	39
	3.5.1.5.2.5. Resource Allocation	40
	3.5.1.5.2.6. User Interface	40
	3.5.1.5.3. Delete	40
	3.5.1.5.3.1. Input	40
	3.5.1.5.3.2. Process	40
	3.5.1.5.3.3. Output	40
	3.5.1.5.3.4. Interface with other functional components	41

	3.5.1.5.3.5. Resource Allocation	41
	3.5.1.5.3.6. User Interface	41
	3.5.1.5.4. View	41
	3.5.1.5.4.1. Input	41
	3.5.1.5.4.2. Process	41
	3.5.1.5.4.3. Output	42
	3.5.1.5.4.4. Interface with other functional components	42
	3.5.1.5.4.5. Resource Allocation	42
	3.5.1.5.4.6. User Interface	42
	3.5.1.6. Attendance Report	42
	3.5.1.6.1. Input	42
	3.5.1.6.2. Process	43
	3.5.1.6.3. Output	43
	3.5.1.6.4. Interface with other functional components	43
	3.5.1.6.5. Resource Allocation	43
	3.5.1.6.6. User Interface	43
	3.5.1.7. My Profile	43
	3.5.1.7.1. Input	43
	3.5.1.7.2. Process	44
	3.5.1.7.3. Output	44
	3.5.1.7.4. Interface with other functional components	44
	3.5.1.7.5. Resource Allocation	44
	3.5.1.7.6. User Interface	44
	3.5.1.8. Logout	45
	3.5.1.8.1. Input	45
	3.5.1.8.2. Process	45
	3.5.1.8.3. Output	45
	3.5.2. Staff	45
	3.5.2.1. Login	46
	3.5.2.1.1. Input	46
	3.5.2.1.2. Process	46
	3.5.2.1.3. Output	46
	3.5.2.1.4. Interface with other functional components	46

	3.5.2.1.5. Resource Allocation	46
	3.5.2.1.6. User Interface	46
	3.5.2.2. Today's Attendance	47
	3.5.2.2.1. Input	47
	3.5.2.2.2. Process	47
	3.5.2.2.3. Output	47
	3.5.2.2.4. Interface with other functional components	47
	3.5.2.2.5. Resource Allocation	47
	3.5.2.2.6. User Interface	47
	3.5.2.3. Update Attendance	48
	3.5.2.3.1. Input	48
	3.5.2.3.2. Process	48
	3.5.2.3.3. Output	48
	3.5.2.3.4. Interface with other functional components	48
	3.5.2.3.5. Resource Allocation	48
	3.5.2.3.6. User Interface	48
	3.5.2.4. Attendance Report	49
	3.5.2.4.1. Input	49
	3.5.2.4.2. Process	49
	3.5.2.4.3. Output	49
	3.5.2.4.4. Interface with other functional components	49
	3.5.2.4.5. Resource Allocation	49
	3.5.2.4.6. User Interface	49
	3.5.2.5. My Profile	50
	3.5.2.5.1. Input	50
	3.5.2.5.2. Process	50
	3.5.2.5.4. Interface with other functional components	50
	3.5.2.5.5. Resource Allocation	50
	3.5.2.5.6. User Interface	50
	3.5.2.6. Logout	51
	3.5.2.6.1. Input	51
	3.5.2.6.2. Process	51
	3.5.2.6.3. Output	51

	3.5.3. Student	51
	3.5.3.1. Login	51
	3.5.3.1.1. Input	52
	3.5.3.1.2. Process	52
	3.5.3.1.3. Output	52
	3.5.3.1.4. Interface with other functional components	52
	3.5.3.1.5. Resource Allocation	52
	3.5.3.1.6. User Interface	52
	3.5.3.2. View Attendance	53
	3.5.3.2.1. Input	53
	3.5.3.2.2. Process	53
	3.5.3.2.3. Output	53
	3.5.3.2.4. Interface with other functional components	53
	3.5.3.2.5. Resource Allocation	53
	3.5.3.2.6. User Interface	53
	3.5.3.3. My Profile	54
	3.5.3.3.1. Input	54
	3.5.3.3.2. Process	54
	3.5.3.3.3. Output	54
	3.5.3.3.4. Interface with other functional components	54
	3.5.3.3.5. Resource Allocation	54
	3.5.3.3.6. User Interface	54
	3.5.3.4. Logout	55
	3.5.3.4.1. Input	55
	3.5.3.4.2. Process	55
	3.5.3.4.3. Output	55
4	DATABASE DESIGN	56-75
	4.1. Introduction	56
	4.2. Purpose and Scope	56
	4.3. Database Identification	57
	4.4. Schema Information	57-58
	4.5. Table Definition	58
	4.5.1. admin	58
	4.5.2. staffinfo	59

	4.5.3. studinfo	60
	4.5.4. subinfo	61
	4.5.5. attendance	61
	4.6. Physical Design	62
	4.7. Data Dictionary	62
	4.8. ER Diagram	62-71
	4.9. Database Administration	72
	4.9.1. DBMS System Information	72
	4.9.2. DBMS configuration	72
	4.9.3. Support software required	72
	4.9.4. Hardware (Storage) requirements	73
	4.9.5. Backup and recover	73-75
5	DEETAILED DESIGN	76-107
	5.1. Introduction	76
	5.2. Structure of software package	76
	5.3. Module decomposition of software	77-78
	5.3.1. Admin	79
	5.3.1.1. Login	79
	5.3.1.1.1. Input	79
	5.3.1.1.2. Procedural Details	79
	5.3.1.1.3. File I/O Interface	79
	5.3.1.1.4. Output	79
	5.3.1.1.5. Implementation aspects	79
	5.3.1.2. Dashboard	80
	5.3.1.2.1. Total No. of Students	80
	5.3.1.2.1.1. Input	80
	5.3.1.2.1.2. Procedural Details	80
	5.3.1.2.1.3. File I/O Interface	80
	5.3.1.2.1.4. Output	80
	5.3.1.2.1.5. Implementation aspects	80
	5.3.1.2.2. Total No. of Staffs	81
	5.3.1.2.2.1. Input	81
	5.3.1.2.2.2. Procedural Details	81
	5.3.1.2.2.3. File I/O Interface	81

	5.3.1.2.2.4. Output	81
	5.3.1.2.2.5. Implementation aspects	81
	5.3.1.2.3. Total No. of Courses	81
	5.3.1.2.3.1. Input	81
	5.3.1.2.3.2. Procedural Details	82
	5.3.1.2.3.3. File I/O Interface	82
	5.3.1.2.3.4. Output	82
	5.3.1.2.3.5. Implementation aspects	82
	5.3.1.3. Student Management	82
	5.3.1.3.1. Add	82
	5.3.1.3.1.1. Input	82
	5.3.1.3.1.2. Procedural Details	83
	5.3.1.3.1.3. File I/O Interface	83
	5.3.1.3.1.4. Output	83
	5.3.1.3.1.5. Implementation aspects	83
	5.3.1.3.2. Update	83
	5.3.1.3.2.1. Input	83
	5.3.1.3.2.2. Procedural Details	83
	5.3.1.3.2.3. File I/O Interface	84
	5.3.1.3.2.4. Output	84
	5.3.1.3.2.5. Implementation aspects	84
	5.3.1.3.3. Delete	84
	5.3.1.3.3.1. Input	84
	5.3.1.3.3.2. Procedural Details	85
	5.3.1.3.3.3. File I/O Interface	85
	5.3.1.3.3.4. Output	85
	5.3.1.3.3.5. Implementation aspects	85
	5.3.1.3.4. View	85
	5.3.1.3.4.1. Input	85
	5.3.1.3.4.2. Procedural Details	86
	5.3.1.3.4.3. File I/O Interface	86
	5.3.1.3.4.4. Output	86
	5.3.1.3.4.5. Implementation aspects	86
	5.3.1.4. Staff Management	86

	5.3.1.4.1. Add	86
	5.3.1.4.1.1. Input	86
	5.3.1.4.1.2. Procedural Details	87
	5.3.1.4.1.3. File I/O Interface	87
	5.3.1.4.1.4. Output	87
	5.3.1.4.1.5. Implementation aspects	87
	5.3.1.4.2. Update	87
	5.3.1.4.2.1. Input	87
	5.3.1.4.2.2. Procedural Details	87
	5.3.1.4.2.3. File I/O Interface	88
	5.3.1.4.2.4. Output	88
	5.3.1.4.2.5. Implementation aspects	88
	5.3.1.4.3. Delete	89
	5.3.1.4.3.1. Input	89
	5.3.1.4.3.2. Procedural Details	89
	5.3.1.4.3.3. File I/O Interface	89
	5.3.1.4.3.4. Output	89
	5.3.1.4.3.5. Implementation aspects	89
	5.3.1.4.4. View	90
	5.3.1.4.4.1. Input	90
	5.3.1.4.4.2. Procedural Details	90
	5.3.1.4.4.3. File I/O Interface	90
	5.3.1.4.4.4. Output	90
	5.3.1.4.4.5. Implementation aspects	90
	5.3.1.5. Subject Management	90
	5.3.1.5.1. Add	91
	5.3.1.5.1.1. Input	91
	5.3.1.5.1.2. Procedural Details	91
	5.3.1.5.1.3. File I/O Interface	91
	5.3.1.5.1.4. Output	91
	5.3.1.5.1.5. Implementation aspects	91
	5.3.1.5.2. Update	91
	5.3.1.5.2.1. Input	92

	5.3.1.5.2.2. Procedural Details	92
	5.3.1.5.2.3. File I/O Interface	92
	5.3.1.5.2.4. Output	92
	5.3.1.5.2.5. Implementation aspects	92
	5.3.1.5.3. Delete	93
	5.3.1.5.3.1. Input	93
	5.3.1.5.3.2. Procedural Details	93
	5.3.1.5.3.3. File I/O Interface	93
	5.3.1.5.3.4. Output	93
	5.3.1.5.3.5. Implementation aspects	93
	5.3.1.5.4. View	94
	5.3.1.5.4.1. Input	94
	5.3.1.5.4.2. Procedural Details	94
	5.3.1.5.4.3. File I/O Interface	94
	5.3.1.5.4.4. Output	94
	5.3.1.5.4.5. Implementation aspects	94
	5.3.1.6. Attendance Report	94
	5.3.1.6.1. Input	94
	5.3.1.6.2. Procedural Details	95
	5.3.1.6.3. File I/O Interface	95
	5.3.1.6.4. Output	95
	5.3.1.6.5. Implementation aspects	95
	5.3.1.7. My Profile	95
	5.3.1.7.1. Input	95
	5.3.1.7.2. Procedural Details	95
	5.3.1.7.3. File I/O Interface	96
	5.3.1.7.4. Output	96
	5.3.1.7.5. Implementation aspects	96
	5.3.1.8. Logout	96
	5.3.1.8.1. Input	96
	5.3.1.8.2. Procedural Details	96
	5.3.1.8.3. File I/O Interface	97
	5.3.1.8.4. Output	97

	5.3.1.8.5. Implementation aspects	97
	5.3.2. Staff	97
	5.3.2.1. Login	97
	5.3.2.1.1. Input	97
	5.3.2.1.2. Procedural Details	97
	5.3.2.1.3. File I/O Interface	98
	5.3.2.1.4. Output	98
	5.3.2.1.5. Implementation aspects	98
	5.3.2.2. Today's Attendance	98
	5.3.2.2.1. Input	98
	5.3.2.2.2. Procedural Details	98
	5.3.2.2.3. File I/O Interface	99
	5.3.2.2.4. Output	99
	5.3.2.2.5. Implementation aspects	99
	5.3.2.3. Update Attendance	99
	5.3.2.3.1. Input	99
	5.3.2.3.2. Procedural Details	99
	5.3.2.3.3. File I/O Interface	100
	5.3.2.3.4. Output	100
	5.3.2.3.5. Implementation aspects	101
	5.3.2.4. Attendance Report	101
	5.3.2.4.1. Input	101
	5.3.2.4.2. Procedural Details	101
	5.3.2.4.3. File I/O Interface	101
	5.3.2.4.4. Output	101
	5.3.2.3.5. Implementation aspects	101
	5.3.2.5. My Profile	101
	5.3.2.5.1. Input	101
	5.3.2.5.2. Procedural Details	101
	5.3.2.5.3. File I/O Interface	101
	5.3.2.5.4. Output	101
	5.3.2.5.5. Implementation aspects	101
	5.3.2.6. Logout	101
	5.3.2.6.1. Input	101

	5.3.2.6.2. Procedural Details	102
	5.3.2.6.3. File I/O Interface	102
	5.3.2.6.4. Output	102
	5.3.2.6.5. Implementation aspects	102
	5.3.3. Student	102
	5.3.3.1. Login	102
	5.3.3.1.1. Input	102
	5.3.3.1.2. Procedural Details	103
	5.3.3.1.3. File I/O Interface	103
	5.3.3.1.4. Output	103
	5.3.3.1.5. Implementation aspects	103
	5.3.3.2. View Attendance	103
	5.3.3.2.1. Input	103
	5.3.3.2.2. Procedural Details	104
	5.3.3.2.3. File I/O Interface	104
	5.3.3.2.4. Output	104
	5.3.3.2.5. Implementation aspects	104
	5.3.3.3. My Profile	104
	5.3.3.3.1. Input	104
	5.3.3.3.2. Procedural Details	105
	5.3.3.3.3. File I/O Interface	105
	5.3.3.3.4. Output	105
	5.3.3.3.5. Implementation aspects	105
	5.3.3.3. Logout	105
	5.3.3.4.1. Input	105
	5.3.3.4.2. Procedural Details	106
	5.3.3.4.3. File I/O Interface	106
	5.3.3.4.4. Output	106
	5.3.3.4.5. Implementation aspects	106
6	PROGRAM CODE LISTING	107-116
7	USER INTERFACE	117-126
8	TESTING	127-141
	8.1. Introduction	127
	8.2. Levels of Testing	127
	8.4. Test Case	127-141

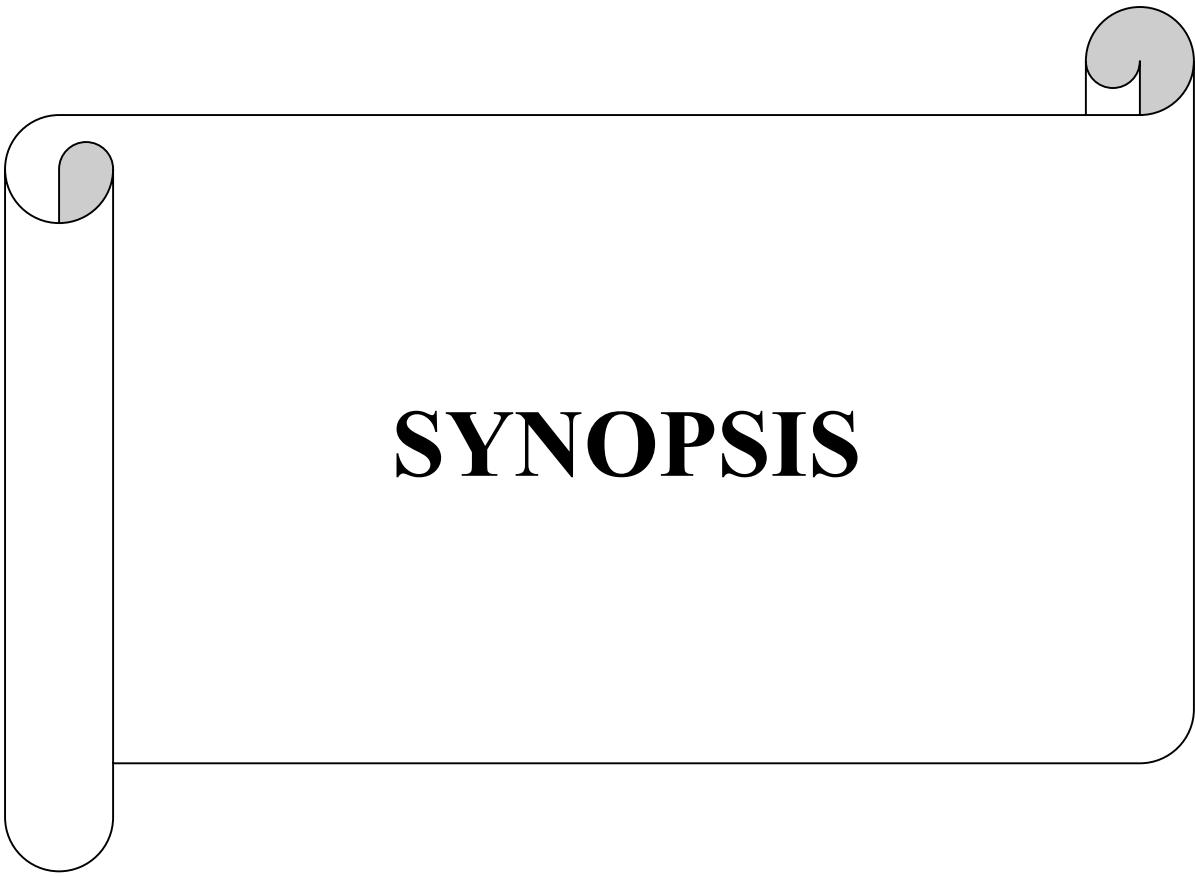
LIST OF FIGURES

Fig No.	TITLE	PAGE NO
1	1.1. System Architecture	6
2	3.1. System Software Architecture	19
3	3.2. System Technical Architecture	20
4	3.3. System Hardware Architecture	20
5	3.4. Context Flow Diagram	21
6	3.5. DFD for Level-0	22
7	3.6. Level 1 DFD for Admin Module	23
8	3.7 Level 1 DFD for Login Module	24
9	3.8. Level 2 DFD for Dashboard	24
10	3.9. Level 3 DFD for Total No. of Students	25
11	3.10. Level 3 DFD for Total No. of Staffs	26
12	3.11. Level 3 DFD for Total No. of Courses	27
13	3.12. Level 2 DFD for Student Management	28
14	3.13. Level 2 DFD for Add Module	29
15	3.14. Level 2 DFD for Update Module	30
16	3.15. Level 2 DFD for Delete Module	31
17	3.16. Level 2 DFD for View Module	32
18	3.17. Level 2 DFD for Staff Management	33
19	3.18. Level 2 DFD for Add Module	34
20	3.19. Level 2 DFD for Update Module	35
21	3.20. Level 2 DFD for Delete Module	36
22	3.21. Level 2 DFD for View Module	37
23	3.22. Level 2 DFD for Subject Management	38
24	3.23. Level 2 DFD for Add Module	39
25	3.24. Level 2 DFD for Update Module	40
26	3.25. Level 2 DFD for Delete Module	41
27	3.26. Level 2 DFD for View Module	42
28	3.27. DFD for Attendance Report Module	43

29	3.28. DFD for My Profile Module	44
30	3.29. Level 1 DFD for Logout Module	45
31	3.30. Level 1 DFD for Staff Module	45
32	3.31. Level 1 DFD for Login Module	46
33	3.32. DFD for Today's Attendance Module	47
34	3.33. DFD for Update Attendance Module	48
35	3.34. DFD for Attendance Report Module	49
36	3.35. DFD for My Profile Module	50
37	3.36. DFD for Logout Module	51
38	3.37. Level 1 DFD for Student Module	51
39	3.38. Level 1 DFD for Login Module	52
40	3.39. DFD for View Attendance Module	53
41	3.40. DFD for My Profile Module	54
42	3.41. DFD for Logout Module	55
43	4.1. Schema Diagram without relation	58
44	4.2. Schema Diagram with relation	58
45	4.3. ER Diagram	71
46	5.1. Structure Chart for Login	79
47	5.2. Flow Chart for Total No. of Students	80
48	5.3. Flow Chart for Total No. of Staffs	81
49	5.4. Flow Chart for Total No. of Courses	82
50	5.5. Structure Chart for Add	83
51	5.6. Flow Chart for Delete	85
52	5.7. Flow Chart for View	86
53	5.8. Structure Chart for Add	87
54	5.9. Flow Chart for Delete	89
55	5.10. Flow Chart for View	90
56	5.11. Structure Chart for Add	91
57	5.12. Flow Chart for Delete	93
58	5.13. Flow Chart for View	94
59	5.14. Flow Chart for logout	96
60	5.15. Structure Chart for Login	97
61	5.16. Flow Chart for Today's Attendance	98
62	5.17. Flow Chart for Logout	102
63	5.18. Structure Chart for Login	103
64	5.19. Flow Chart for View Attendance	104
65	5.20. Flow Chart for Logout	106

LIST OF TABLES

SNO	TITLE	PAGE NO
1	3.1. Data Flow Diagram	22
2	4.1. admin	58
3	4.2. staffinfo	59
4	4.3. studinfo	60
5	4.4. subinfo	61
6	4.5. attendance	61
7	4.6. ER Diagram	63
8	5.1. Structure Chart	77
9	5.2. Flow Chart	78
10	8.1. Testing for login module	142-143
11	8.2. Testing for add student details	144-145
12	8.3. Testing for update staff details	146-147
13	8.4. Testing for delete subject details	147-148
14	8.5. Testing for attendance report	148-149
15	8.6. Testing for today's attendance	150
16	8.7. Testing for update attendance	150-151
17	8.8. Testing for view attendance report by staff	152-153
18	8.9. Testing for view attendance shortage	153-154
19	8.10. Testing for view attendance	154-155



SYNOPSIS

1. INTRODUCTION

1.1 Introduction of the System

1.1.1 Project Title

Student Attendance Management System.

1.1.2 Category

Web application.

1.1.3 Overview

Attendance Management System is software developed for daily student attendance in schools, colleges and institutes. Here the staffs, who are handling the subjects, will be responsible to mark the attendance of the students and student can access their attendance information. Each staff will be given with a separate Email Id and Password. This system will also help in evaluating attendance eligibility criteria of student. Report of the student's attendance on weekly and monthly basis is generated.

1.2 Background

1.2.1 Introduction of the Company

- Not applicable.

1.2.2 Brief note on existing system

In Existing system all work is done on paper. Here the attendance will be carried out in the handwritten registers and only at the end of the session the reports are generated. It will be a tedious job to maintain the record for the user. The human effort is more here. The retrieval of the information is not as easy as the records are maintained in the hand written registers. So, the desire for the development of the proposed system had become essential.

Limitations of existing system:

- It is not user friendly because data retrieval is very slow.
- More calculations are required to generate report manually, it requires lots of paperwork.
- All calculations to generate report is done manually so there is a greater chance of errors.
- Report can be generated in the middle of the session or as per the requirement because it is time consuming.

1.3 Objective of the system

Objectives are the pre-determined goals or outcome of the system process. Object determine the final output of the process. Objectives are to be planned in the initial phase. The purpose of the project is to computerize the tradition way of taking attendance. This software generates the report at the end of the session or in the between of the session. The proposed system will reduce cumbersome paperwork.

1.4 Scope of the System

Scope is the limitation that process faces from the beginning to the end. It may help collecting perfect management in details. Scope can be altered in any phase including the initial one. The purpose is to maintain a centralized database of all event related information. It satisfies the user requirement.

The scope of the project is the system on which the software is installed, i.e., the project is developed as a desktop application, and it will work for a particular institute.

- As our project is a web application it provides easy-to-use functionality to the user, that means user need not to have high technical skill but we should have some basic knowledge to interact with user interface of our application.
- This application fit to desktop of the system rather than the mobile interface and requires active internet connection.

1.5 Structure of the system

1.5.1 Analysis

Usually, attendance is maintained and calculated manually. This process is cumbersome and time consuming and only at the end of session the reports are generated. This application will be useful to overcome this problem. It is used for storing attendance of each student in computerized system.

1.5.2 Module description

The system should be designed in such a way that only authorized people should be allowed to access some modules. The record should be modified only by the administrators and no one else. The user interface should be consistent so that the user can handle the application with ease and speed.

1.5.2.1 Admin

The admin can login to the system using Email Id and password. He is responsible for maintaining the staff, student and subject details like adding, deleting and updating the records. He can view student attendance also.

1.5.2.1.1 Login

In this admin can login to the system using Email Id and Password.

1.5.2.1.2 Dashboard

1.5.2.1.2.1 Total No. of students

It displays total number of students present in database.

1.5.2.1.2.2 Total No. of staffs

It displays total number of staffs present in database.

1.5.2.1.2.3 Total No. of courses

It displays total number of course.

1.5.2.1.3 Student management

In this module it deals with allocation of Register Number and personal details of student. The student records can be added, deleted, updated, and viewed based on Register Number by the admin.

1.5.2.1.4 Staff management

It provides the facility to have Email Id, Password, and personal details of staff. The admin will manage the staff records by adding, deleting, updating, and viewing records.

1.5.2.1.5 Subject management

In this module it deals with allocation of subject code and details of subject, name of the staff who handle those subjects. The subject details can be added, deleted, updated, and viewed based on subject code by the admin.

1.5.2.1.6 Attendance Report

Report can be taken by daily, weekly and consolidate:

Weekly reports get all hour details of attendance starting date to ending date and display the attendance percentage.

1.5.2.1.8 My Profile

In this module admin can view and update his profile.

1.5.2.1.9 Logout

By this module admin can logout from the Application.

1.5.2.2 Staff

Staff must authorize user of the college to access this application. Staff can mark the attendance of the student, update attendance and view the attendance report of the student. In this staff can login to the system using Email Id and Password.

1.5.2.2.1 Login

By this staff can login to the system using Email Id and Password.

1.5.2.2.2 Today's Attendance

It assists the staff to mark attendance to the students for their subject.

1.5.2.2.3 Update Attendance

It allowed the staff to update the attendance of the student for their subject.

1.5.2.2.4 Attendance Report

Weekly reports get all hour details of attendance from starting date to ending date and display the status and consolidate report get all student attendance details from starting date to ending date status help for the eligibility criteria of the student to attend the examination.

1.5.2.2.5 My Profile

By this module staff can view his profile.

1.5.2.2.6 Logout

By this module staff can logout from the Application.

1.5.2.4 Student

Student must be authenticated user of the college in order to access this application. Student can view his/her attendance. In this student can login to the system Email Id and Password.

1.5.2.4.1 Login

In this student can login to the system using Email Id and Password.

1.5.2.4.2 View Attendance

In this module it helps the student to view his attendance details.

1.5.2.4.3 My Profile

By this module student can view his profile.

1.5.2.4.4 Logout

By this module student can logout from the Application.

1.6 System Architecture

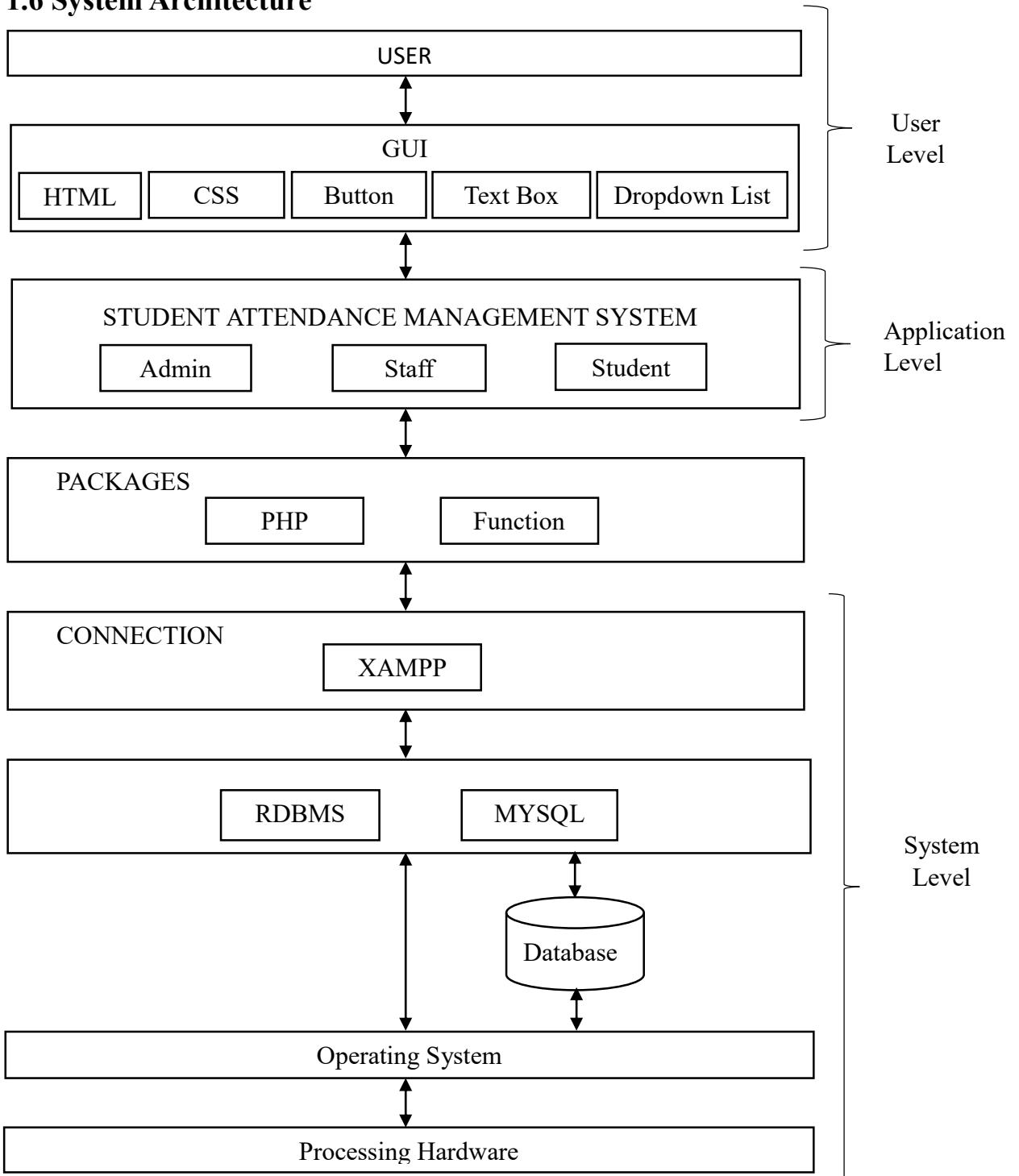


Figure 1.1 System Architecture

1.7 End user

- Admin
- Staff
- Student

1.8 Software/Hardware need for the development

1.8.1 The following hardware specification is used to develop this project

Processor : I3 5th Generation

RAM : 4 GB of free disk space or higher

Hard Disk : 500 GB

1.8.2 The following software specification is used to develop this project

Operating System : Windows 7 or higher

Front End : HTML/CSS/JAVASCRIPT

Back End : PHP 5.0

Web Server : Apache Tomcat

Other Tools : XAMPP

1.9 Software/Hardware need for the implementation

1.9.1 Hardware needed for the implementation

To run this app the basics requirements are as follows:

Hard Disk : 500 GB, 4GB RAM

Input Devices : Keyboard, Mouse etc.

1.9.2 Software needed for the implementation

This product will use PHP and it operates on Windows 7 and other higher versions.



SOFTWARE REQUIREMENT SPECIFICATION

2. SOFTWARE REQUIREMENT SPECIFICATION

2.1 Introduction

A software requirement specification (SRS) is a document that describes what the software will do how it will be expected to perform. It also describes the functionality the product needs to be fulfil all user needs. A software requirement specification is a description of software system to be developed. It must bridge communication gap between developer and user. It describes functional and non-functional requirement and may include a set of cases that describes user interaction that the software must provide. SRS is an agreement between client and the developer.

The purpose of this document is to provide overall description of the event with functionalities and their dependencies with each other and their requirement. Using the SRS helps an enterprise confirm that the requirements are fulfilled and helps business lenders make decisions about the lifecycle of their product, such as when to retire a feature.

SRS document provides a reference for validation of the final product. A high-quality SRS is a prerequisite to high-quality software. In future one can refer the following SRS to extend the same software, by referring this SRS document other developers can develop and extend this software to network level. It specifies basic requirement constraints and interface for the proposed system.

2.2 Overall Description

This section describes the function of the project and their aim. It also includes the constraints and the requirements of the project.

2.2.1 Product Perspective

2.2.1.1 System Interfaces

This application best runs on web browser such as chrome and firefox on windows.

2.2.1.2 User Interfaces

The graphical user interface makes application more interactive which includes buttons and labels for selecting options and perform operations.

This application GUI provides menus, toolbars, buttons, panes, containers, grids allowing for easy control by a keyboard and a mouse.

2.2.1.3 Hardware Interfaces

Not applicable

2.2.1.4 Software Interfaces

- This application allows export data with MS Excel.
- This product uses PHP and HTML/CSS/JAVASCRIPT and it operates on Windows 7 or other higher versions.

2.2.1.5 Communication Interfaces

It requires stable internet connection.

2.2.1.6 Interfaces with Server

This application allows to interfaces with MYSQL Server, XAMPP, Apache Tomcat.

2.2.2 Product Functions

The main function of the project is to mark student attendance. Whenever attendance details are needed authenticated user can easily access it.

- Admin will add, update, delete, view the necessary details.
- Staff will mark the attendance and view student attendance.
- Student can view their attendance.

2.2.3 User characteristics

User (Admin, Student, Staff) should have basic computer knowledge and know about the functionalities of the application.

2.2.4 General constraints

This software operates on windows 7 or other higher versions.

2.2.5 Assumptions and Dependencies

These factors are not designing constraint on the software but any changes to these factors can affect the requirement in the SRS.

- This system requires 4 GB RAM and 500 GB Hard Disk.
- The system is dependent on the availability of an Apache Tomcat Server to run.

2.3 Special Requirements

Not applicable

2.4 Functional Requirements

Functional requirements in an SRS document indicate what a software system must do and how it must function; they are product features that focus on user needs.

2.4.1 Admin

2.4.1.1 Login

In this Admin can login to the system using Email Id and Password.

Input: Email Id, Password.

Process: Validates Email Id, Password.

Output: If Email Id, Password is valid, it will display respective page
else display error message.

2.4.1.2 Dashboard

2.4.1.2.1 Total No. of Students

Input: Button click.

Process: Retrieves total number of students from database.

Output: Displays total number of students.

2.4.1.2.2 Total No. of Staffs

Input: Button click.

Process: Retrieves total number of staffs from database.

Output: Displays total number of staffs.

2.4.1.2.3 Total No. of Courses

Input: Button click.

Process: Retrieves total number of courses from database.

Output: Displays total number of courses.

2.4.1.3 Student Management

2.4.1.3.1 Add

Input: First Name, Last Name, Roll Number, Register Number, Father Name, Mother Name, Contact Number, Parent Contact Number, Email Id, Password, Address, Gender, DOB, Year, Semester, Course.

Process: Stores the valid details to database.

Output: Displays inserted successfully message.

2.4.1.3.2 Update

Input: Register Number.

Process: Stores updated details into database.

Output: Displays updated successfully message.

2.4.1.3.3 Delete

Input: Register Number.

Process: Specified student details will be deleted from database.

Output: Displays deleted successfully message.

2.4.3.3.4 View

Input: Click the view student button.

Process: Retrieve the student information from database.

Output: Displays student information.

2.4.1.4 Staff Management

2.4.1.4.1 Add

Input: First Name, Last Name, Staff Id, Contact Number, Email Id, Password, Address, Gender, DOB, DOJ, Experience, Qualification, Department.

Process: Store valid details to database.

Output: Displays inserted successfully message.

2.4.1.4.2 Update

Input: Staff Id.

Process: Store updated details in database.

Output: Displays updated successfully message.

2.4.1.4.3 Delete

Input: Staff Id.

Process: Specified staff details will be deleted from database.

Output: Displays deleted successfully message.

2.4.1.4.4 View

Input: Click the view staff button.

Process: Retrieve the staff information from database.

Output: Displays staff information.

2.4.1.5 Subject Management

2.4.1.5.1 Add

Input: Subject Code, Subject Name, Year, Semester, Course, Department, Staff Name.

Process: Store valid details to database.

Output: Displays inserted successfully message.

2.4.1.5.2 Update

Input: Subject code.

Process: Store updated details in database.

Output: Displays updated successfully message.

2.4.1.5.3 Delete

Input: Subject code.

Process: Specified subject details will be deleted from database.

Output: Displays deleted successfully message.

2.4.1.5.4 View

Input: Click the view subject button.

Process: Retrieve the subject information from database.

Output: Displays subject information.

2.4.1.6 Attendance Report

Input: Year, Sem, Course, Subject Name.

Process: Retrieve attendance report from database.

Output: Displays attendance report.

2.4.1.7 My Profile

Input: Button click.

Process: Retrieve the admin profile from database.

Output: Displays profile information.

2.4.1.8 Logout

Input: Click the logout button.

Process: Logout process will be taken place.

Output: Redirect to the Home Page.

2.4.2 Staff

2.4.2.1 Login

In this Staff can login to the system using Email Id and Password.

Input: Email Id, Password.

Process: Validates Email Id, Password.

Output: If Email Id and Password is valid, it will display respective page else display error message.

2.4.2.2 Today's Attendance Entry

Input: Year, Sem, Course, Subject Name, Mark attendance.

Process: Store student attendance to database.

Output: Displays submitted successful message.

2.4.2.3 Update Attendance

Input: Year, Sem, Course, Subject Name, Date.

Process: Store updated attendance into database.

Output: Displays updated successful message.

2.4.2.4 Attendance Report

Input: Year, Sem, Course, Subject Name.

Process: Retrieves attendance report from database.

Output: Displays attendance report.

2.4.2.5 My Profile

Input: Button click.

Process: Retrieve the staff profile from database.

Output: Displays profile information.

2.4.2.6 Logout

Input: Click the Logout button.

Process: Logout process will be taken place.

Output: Redirect to the Home Page.

2.4.3 Student

2.4.3.1 Login

In this student can login to the system using Email Id and Password.

Input: Email Id, Password.

Process: Validates Email Id, Password.

Output: If Email Id, Password is valid, it will display respective page
else display error message.

2.4.3.2 View Attendance

Input: Input Subject Name.

Process: Retrieves attendance of specific student from database.

Output: Displays attendance of specific student.

2.4.3.3 My Profile

Input: Button click.

Process: Retrieve the student profile from database.

Output: Displays profile information.

2.4.3.4 Logout

Input: Click the Logout button.

Process: Logout process will be taken place.

Output: Redirect to the Home page.

2.5 Design Constraints

2.5.1 Hardware Constraints

The development of this application requires a computer system with 4 GB RAM and 500 GB Hard Disk.

2.5.2 Software Constraints

The system must include web browser such as Google Chrome or Firefox and must have stable net connection. It operates on Windows 7 or other higher versions.

2.5.3 Fault Tolerance

If the user enters any invalid input, then it will show the error message.

2.5.4 Security

There shall be strong mechanism should be placed in the server side of the system to keep unwanted users to damage the system. The software will authenticate each user who logs in using email id and password.

2.5.5 Standard Compliance

The software has Graphical user Interface (GUI) like button, textbox, dropdown list which is to ease of the user.

2.6 System Attributes

- **Availability**

Availability refers to the percentage of time that the infrastructure, system, or solution remains operational under normal circumstances in order to serve its intended purpose. This application is made available as Web Application.

- **Portability**

Portability, in relation to software, is a measure of how easily an application can be transferred from one computer environment to another. A computer software application is considered portable to a new environment if the effort required to adapt it to the new environment is within reasonable limits.

- **Reliability**

Reliability refers to the probability that the system will meet certain performance standards in yielding correct output for a desired time duration. This application system must be highly reliable and it should generate all the outputs in the correct order.

- **Maintainability**

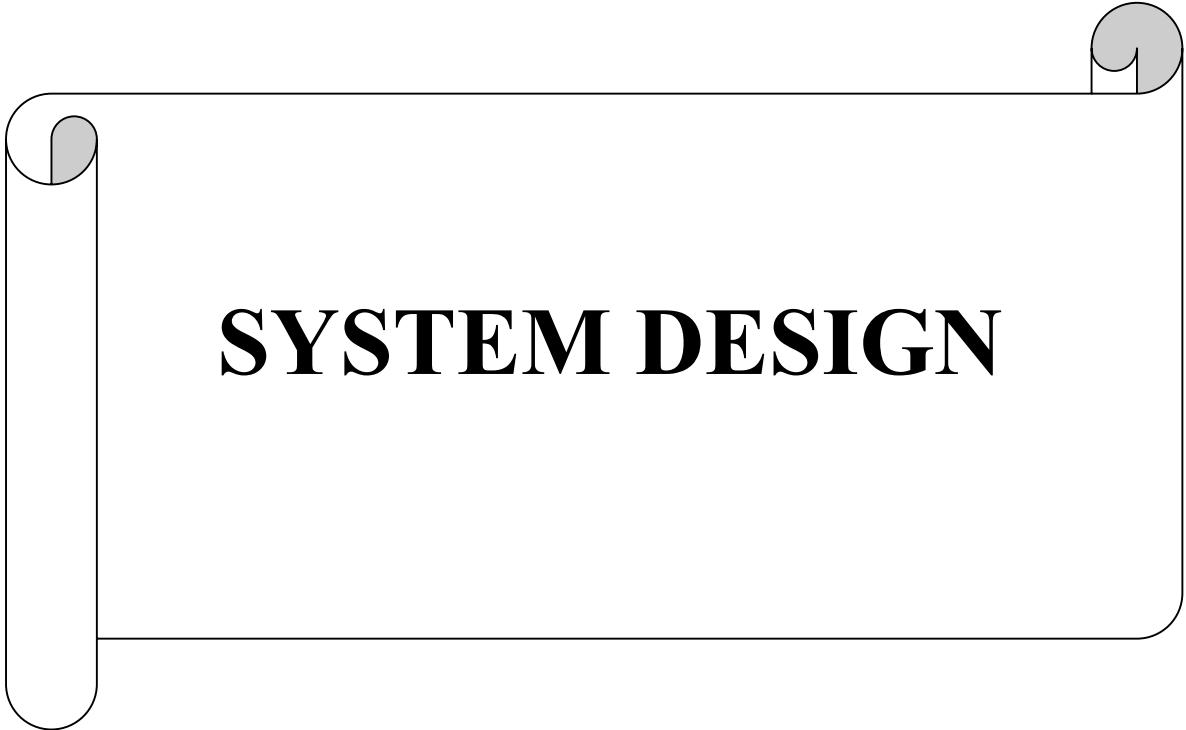
Maintainability refers to the ease with which you can repair, improve, and understand software code. Software maintenance is a phase in the software development cycle that starts after the customer has received the product. This application will be developed in such a way that it can be modified.

- **Scalability**

Software scalability is an attribute of a tool or a system to increase its capacity and functionalities based on its users' demand. Scalable software can remain stable while adapting to changes, upgrades, overhauls, and resource reduction. This application is scalable so that it can move from smaller to a larger operating system and take full advantage of the larger operating system in terms of performance.

2.7 Other requirements

Not applicable.



SYSTEM DESIGN

3. SYSTEM DESIGN

3.1 Introduction

- System design is the process of defining the architecture, module interfaces and data for a system to satisfy specified requirements.
- The purpose of the design phase is to plan the solution of the problem specified by the requirement documents.
- This is the first step that moving from problem domain to the solution domain.
- The design of the system is essentially a blueprint or a plan for a solution for the system.

3.2 Assumptions and Constraints

An assumption is a condition you think to be true and a constraint is fixed limitations of project development.

- All the functional requirements collected from client are sufficient for the project lifecycle.
- Time constraint-within 3 months.

3.3 Functional Decomposition

Functional decomposition is the process of taking a complex process and breaking it down into simpler parts. Using functional decomposition large or complex functionalities decomposed into smaller parts that are more easily understood. It is mainly used during project analysis phase, so each phase can be viewed as software. So, this has modular with some module.

3.3.1 System Software Architecture

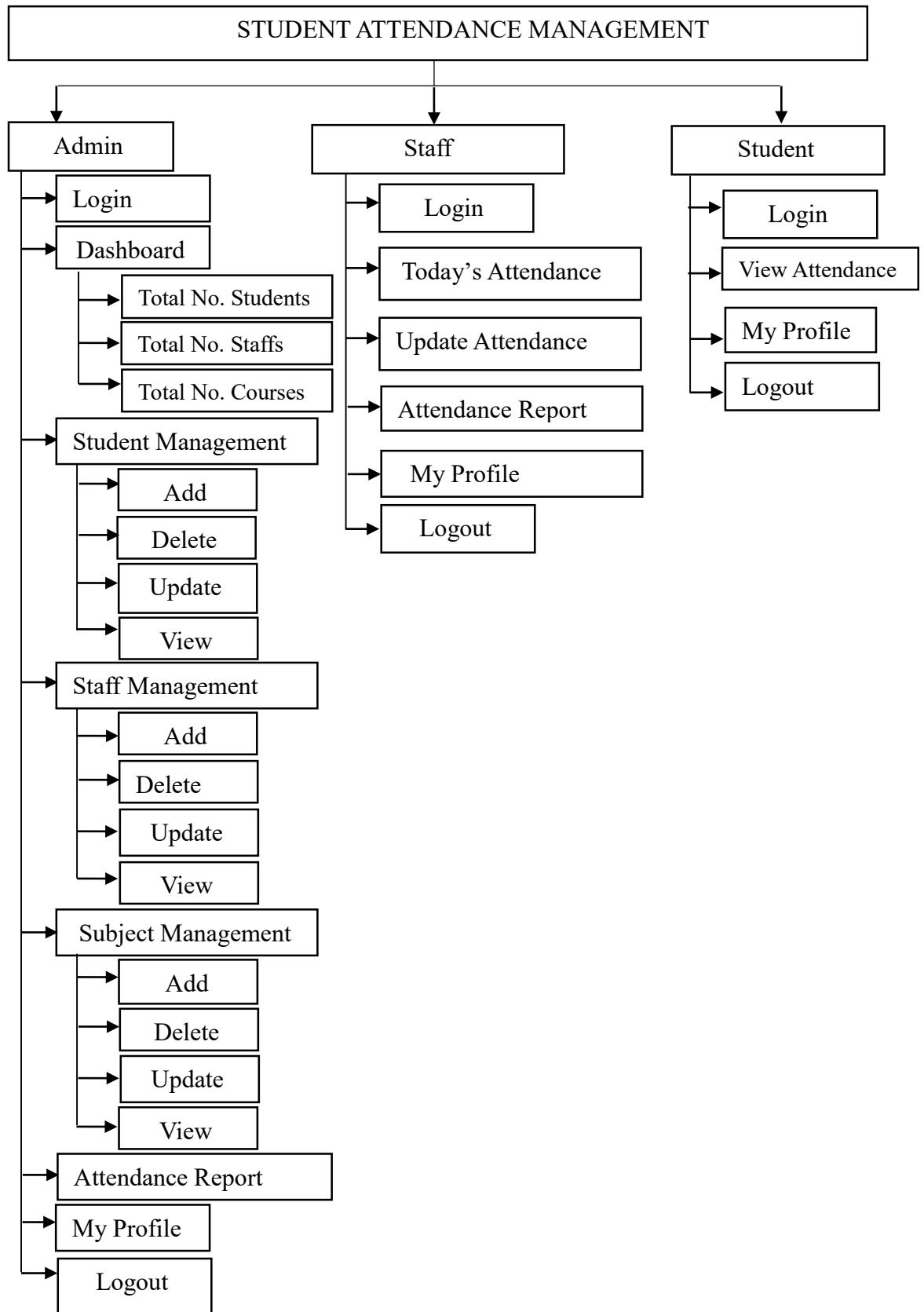


Figure 3.1 System Software Architecture

3.3.2 System Technical Architecture

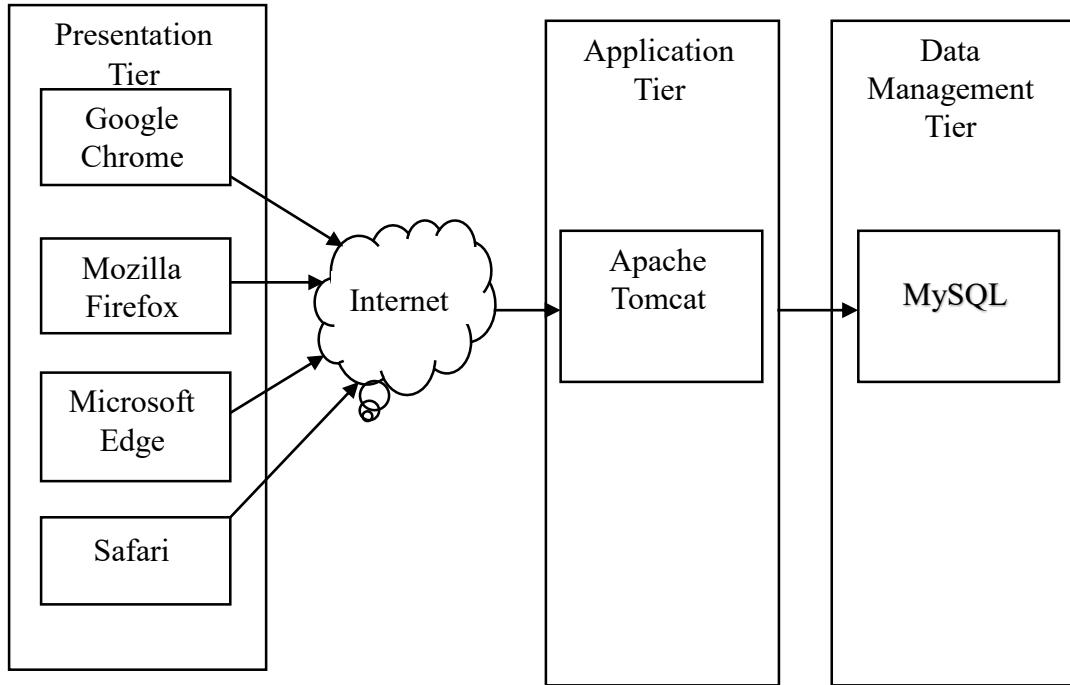


Figure 3.2 System Technical Architecture

3.3.3 System Hardware Architecture

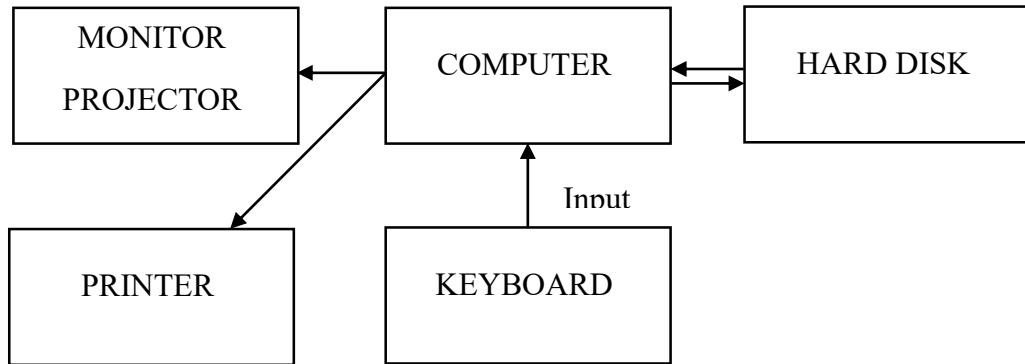


Figure 3.3 System Hardware Architecture

3.3.4 External Interfaces

Not applicable.

3.4 Description of Programs

3.4.1 Context Flow Diagram (CFD)

Context Flow Diagram shows the input and output of the system. It shows the external entities that interact with the system and how the data flow between these external entities and system.

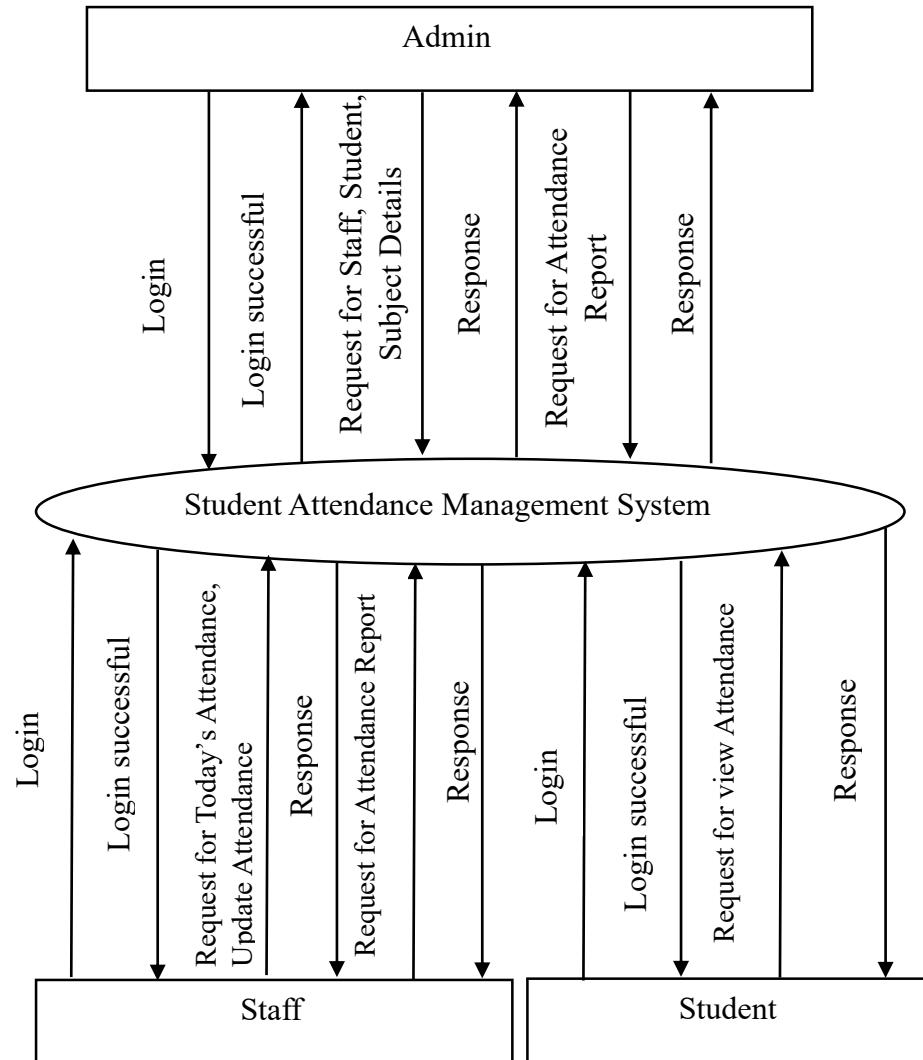


Figure 3.4 Context Flow Diagram

3.4.2 Data Flow Diagram

Data Flow shows the flow of data through system. This is also called the data flow graphs. It views system as a function that transforms input into desired output. Any complex system will not perform this transformation in a single step and a data will typically undergo a series of transformation before it becomes the output. It aims to capture the information that taken place within a system to the input data so that eventually the output data is produced.

Symbols	Name	Description
	Process	It performs transformation of data from one form to another.
	Source/Sink	It represents the external entity that may be either Source or Sink.
	Data Flow	It represents the flow of data from source to destination.
	Data Source/ Data Storage	It is the place where data is stored.

Table 3.1 Data Flow Diagram

3.4.2.1 DFD FOR LEVEL-0

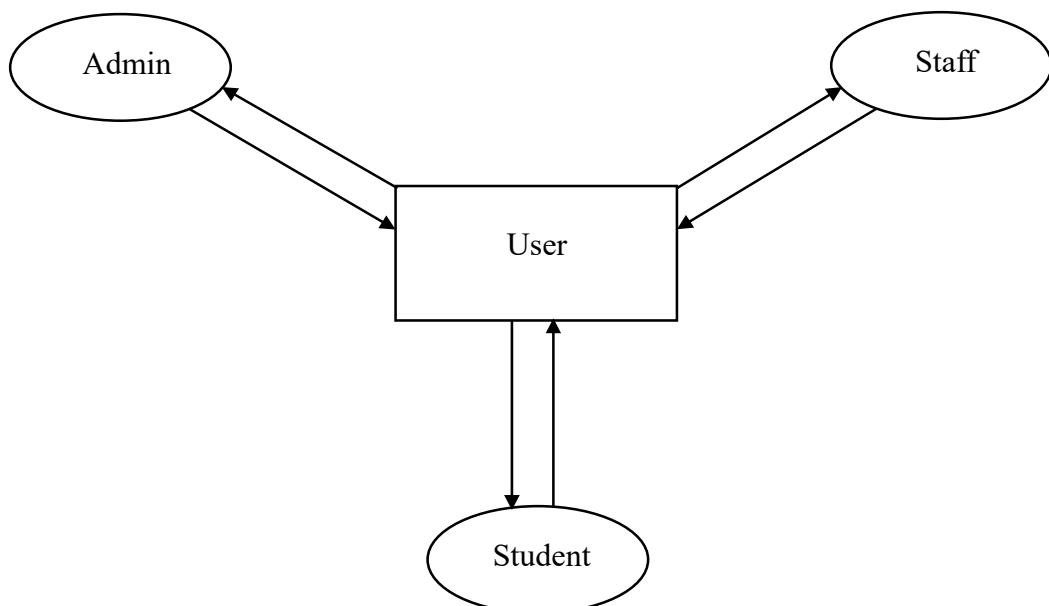


Figure 3.5 DFD for Level-0

3.5 Description of components

3.5.1 Admin

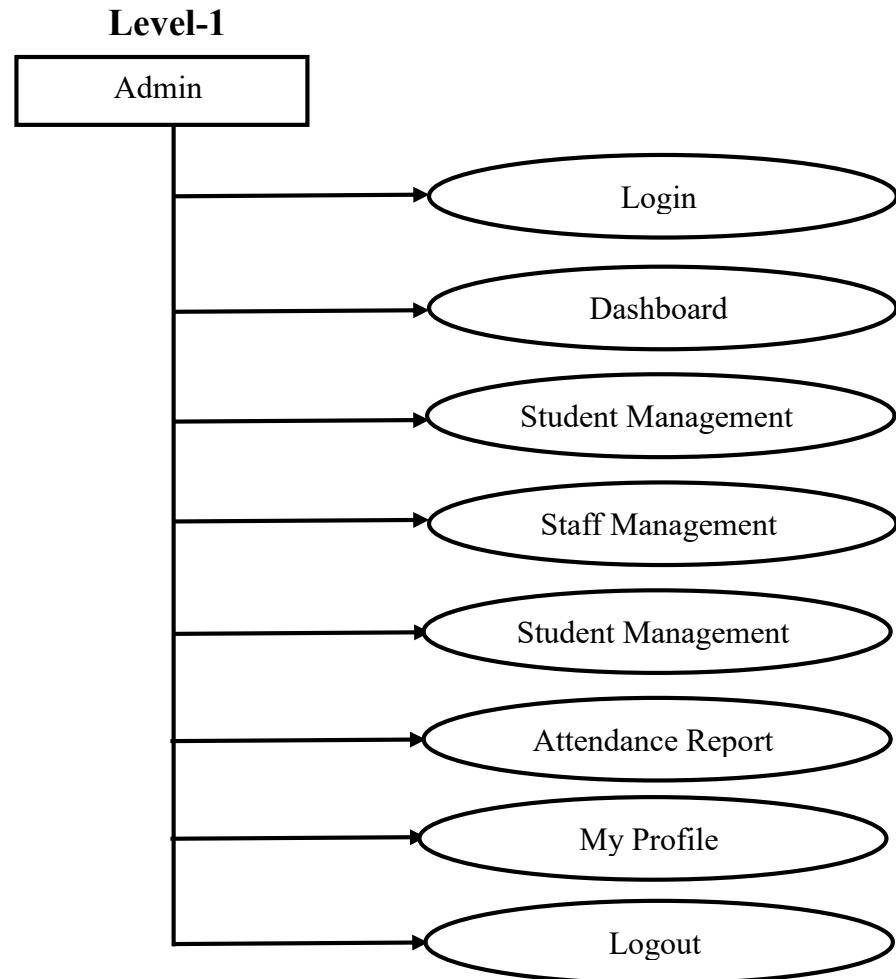


Figure 3.6 Level 1 DFD for Admin Module

3.5.1.1 Login

In this admin can login the system using Email-Id and Password.

3.5.1.1.1 Input

Email-Id, Password

3.5.1.1.2 Process

Validates for Email Id and Password.

3.5.1.1.3 Output

Respective page will be loaded.

3.5.1.1.4 Interface with other functional components

Independent module.

3.5.1.1.5 Resource Allocation

admin

3.5.1.1.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

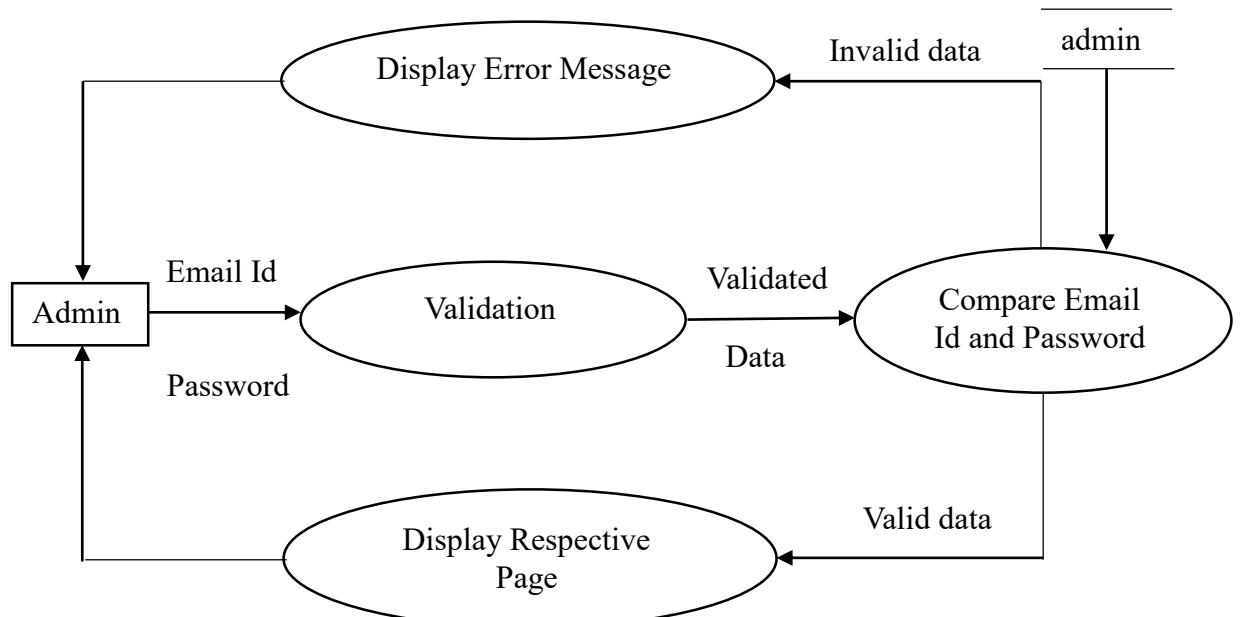


Figure 3.7 Level 1 DFD for Login Module

3.5.1.2 Dashboard

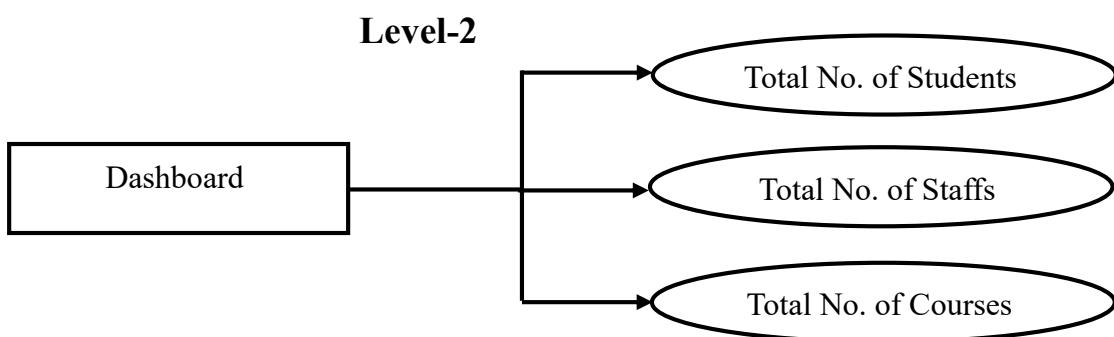


Figure 3.8 Level 2 DFD for Dashboard

3.5.1.2.1 Total No. of Students

3.5.1.2.1.1 Input

Button click.

3.5.1.2.1.2 Process

Retrieve total no. of students from database.

3.5.1.2.1.3 Output

Display total number of students.

3.5.1.2.1.4 Interface with others functional component

Add

3.5.1.2.1.5 Resource allocation

studinfo

3.5.1.2.1.6 User interfaces

Button, Edit Text, Text view are used to design the user interface.

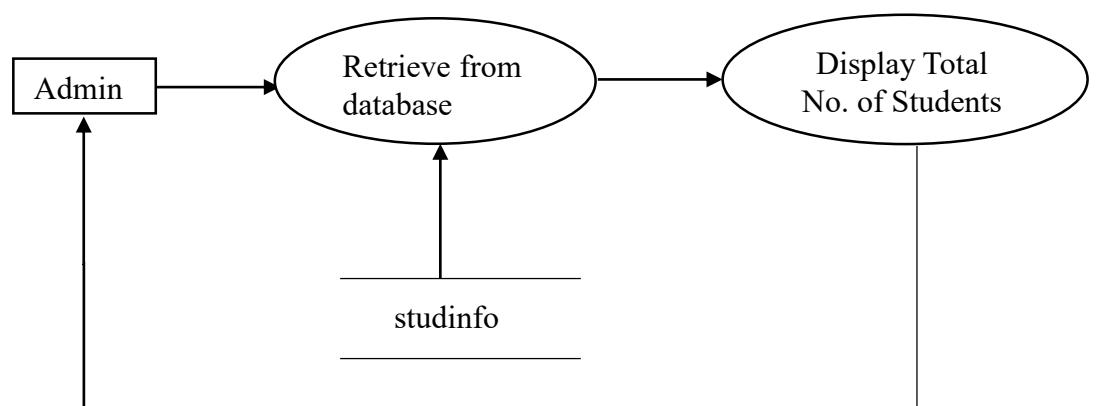


Figure 3.9 Level 3 DFD for Total No. of Students

3.5.1.2.2 Total No. of Staffs

3.5.1.2.2.1 Input

Button click.

3.5.1.2.2.2 Process

Retrieve total no. of staff from database.

3.5.1.2.2.3 Output

Display total number of staffs.

3.5.1.2.2.4 Interface with others functional component

Add

3.5.1.2.2.5 Resource allocation

staffinfo

3.5.1.2.1.6 User interfaces

Button, Edit Text, Text view are used to design the user interface.

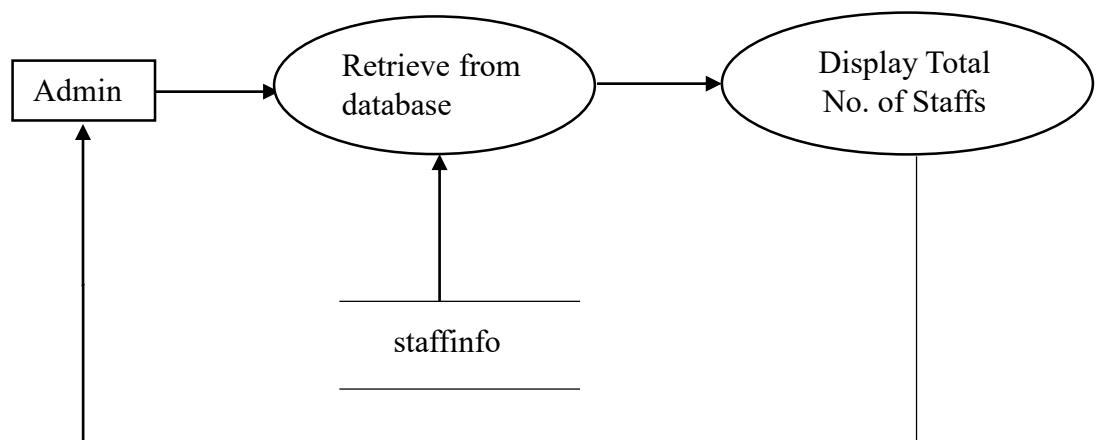


Figure 3.10 Level 3 DFD for Total No. of Staffs

3.5.1.2.3 Total No. of Courses

3.5.1.2.3.1 Input

Button click.

3.5.1.2.3.2 Process

Retrieve total no. of courses from database.

3.5.1.2.3.3 Output

Display total number of courses.

3.5.1.2.3.4 Interface with others functional component

Add

3.5.1.2.3.5 Resource allocation

studinfo

3.5.1.2.3.6 User interfaces

Button, Edit Text, Text view are used to design the user interface.

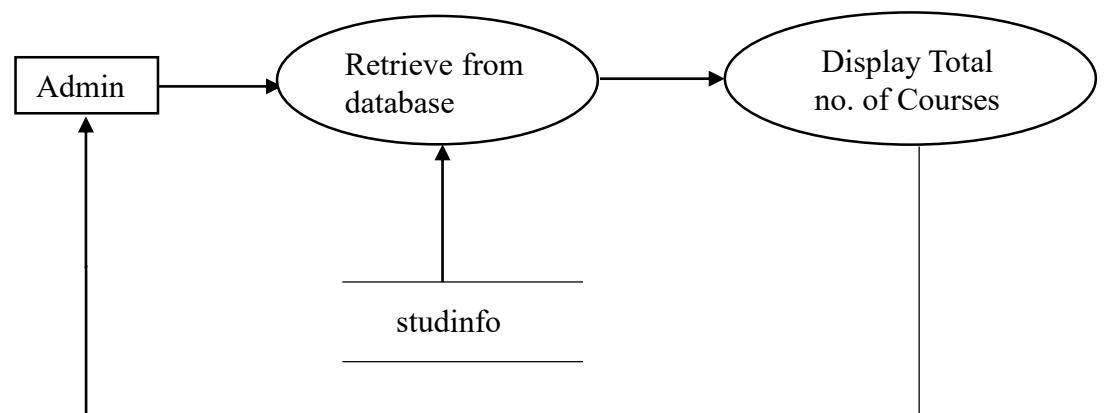


Figure 3.11 Level 3 DFD for Total No. of Courses

3.5.1.3 Student Management

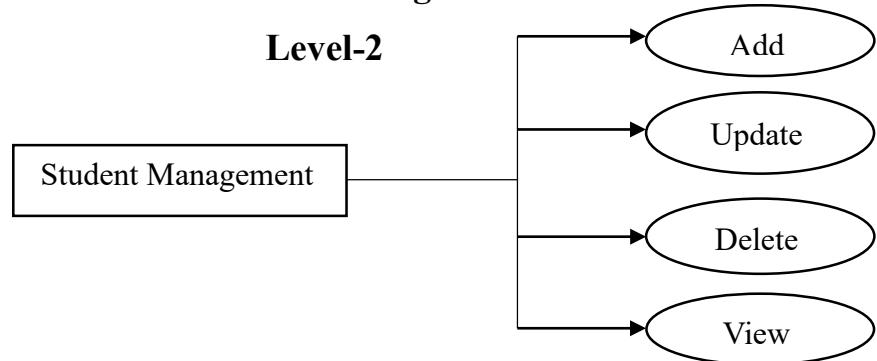


Figure 3.12 Level 2 DFD for Student Management

3.5.1.3.1 Add

3.5.1.3.1.1 Input

First name, Last name, Roll No, Register No, Contact number, Parent Contact Number, Father Name, Mother Name, Address, Gender, DOB, Email-ID, Password, Year, Semester, Course.

3.5.1.3.1.2 Process

It validate and stores the student information to the database.

3.5.1.3.1.3 Output

If valid it displays inserted successfully message or else displays error message.

3.5.1.3.1.4 Interface with other functional components

This is an independent module.

3.5.1.3.1.5 Resource allocation

studinfo

3.5.1.3.1.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

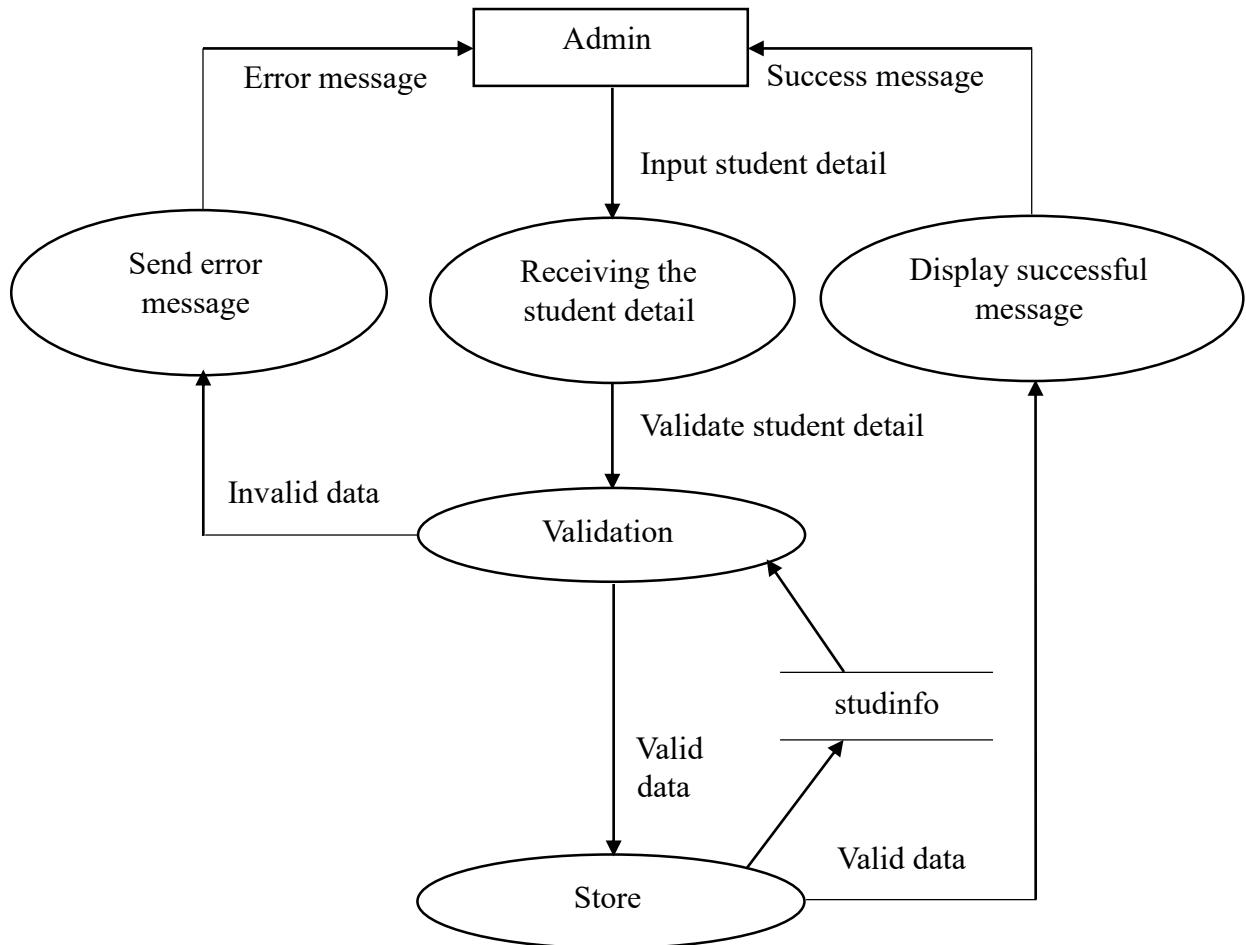


Figure 3.13 Level 2 DFD For Add Module

3.5.1.3.2 Update

3.5.1.3.2.1 Input

Register Number.

3.5.1.3.2.2 Process

It changes the student information and store it in database.

3.5.1.3.2.3 Output

If valid it displays updated successfully message or else displays error message.

3.5.1.3.2.4 Interface with other functional components

Add

3.5.1.3.2.5 Resource Allocation

studinfo

3.5.2.1.2.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

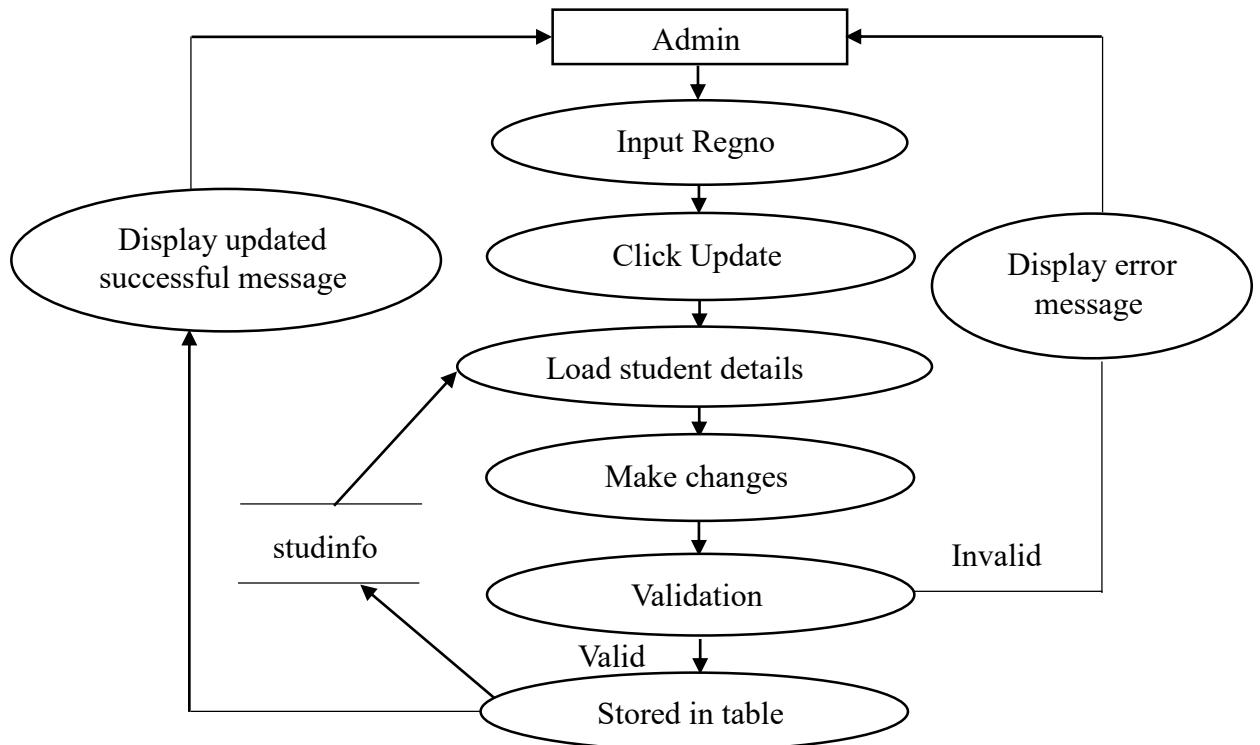


Figure 3.14 Level 2 DFD for Update Module

3.5.1.3.3 Delete

3.5.1.3.3.1 Input

Register Number.

3.5.1.3.3.2 Process

It removes the student information from database.

3.5.1.3.3.3 Output

Displays deleted successfully message.

3.5.1.3.3.4 Interface with other functional components

Add

3.5.1.3.3.5 Resource Allocation

studinfo

3.5.1.3.3.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

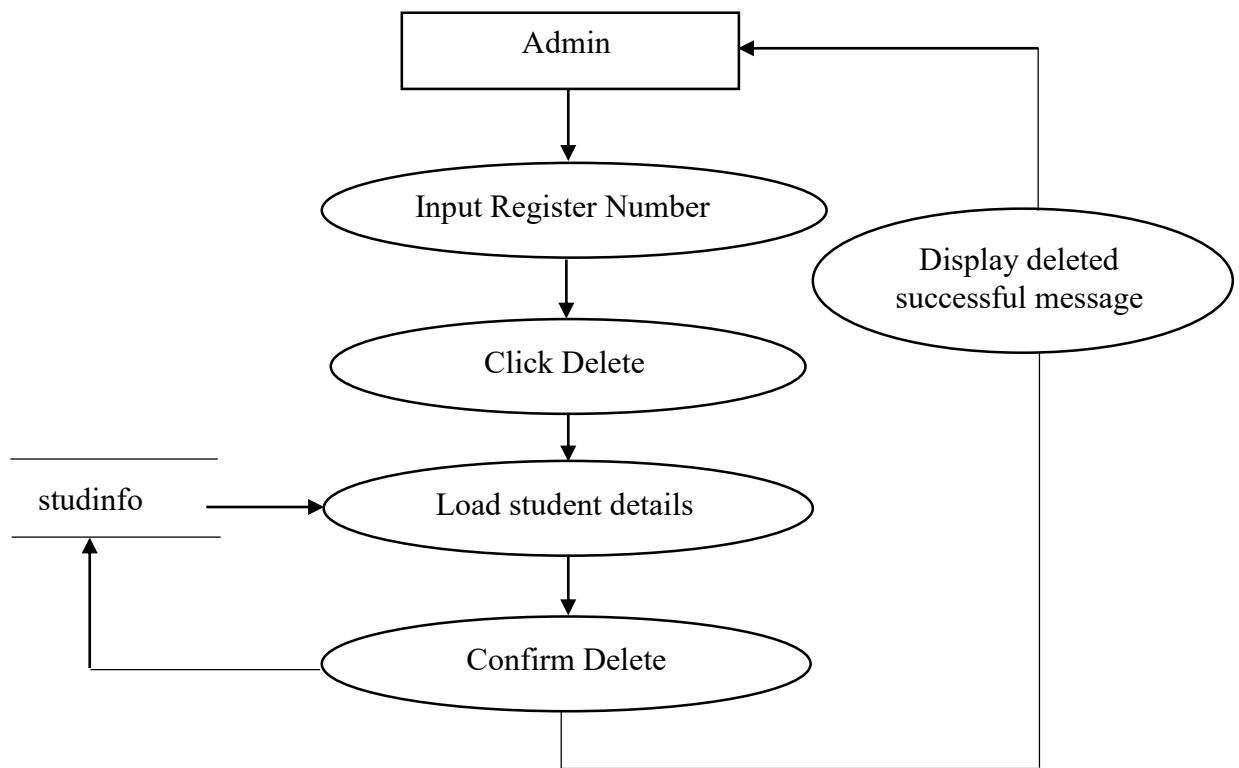


Figure 3.15 Level 2 DFD for Delete Module

3.5.1.3.4 View

3.5.1.3.4.1 Input

Button click.

3.5.1.3.4.2 Process

It retrieves the student information from database.

3.5.1.3.4.3 Output

Display the student detail from the database.

3.5.1.3.4.4 Interface with other functional components

Add

3.5.1.3.4.5 Resource Allocation

studinfo

3.5.1.3.4.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

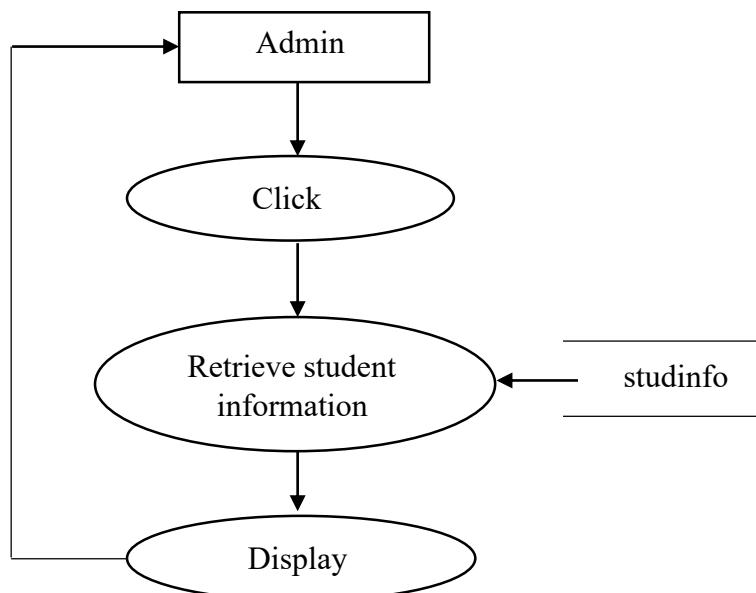


Figure 3.16 Level 2 DFD for View Module

3.5.1.4 Staff Management

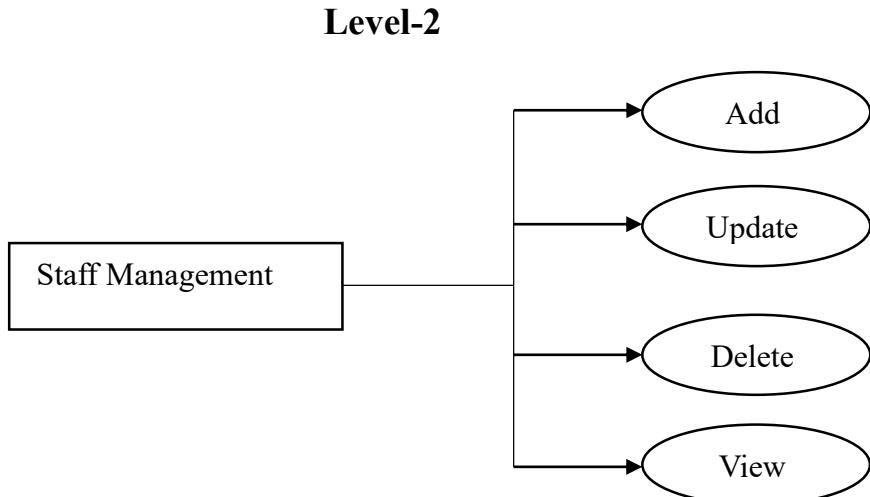


Figure 3.17 Level 2 DFD for Staff Management

3.5.1.4.1 Add

3.5.1.4.1.1 Input

First Name, Last Name, Staff Id, Contact Number, Address, Gender, DOB, DOJ, Experience, Qualification, Department.

3.5.1.4.1.2 Process

It validate and stores the staff information.

3.5.1.4.1.3 Output

If valid it displays inserted successfully message or else displays error message.

3.5.1.4.1.4 Interface with other functional components

This is an independent module.

3.5.1.4.1.5 Resource allocation

staffinfo

3.5.1.4.1.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

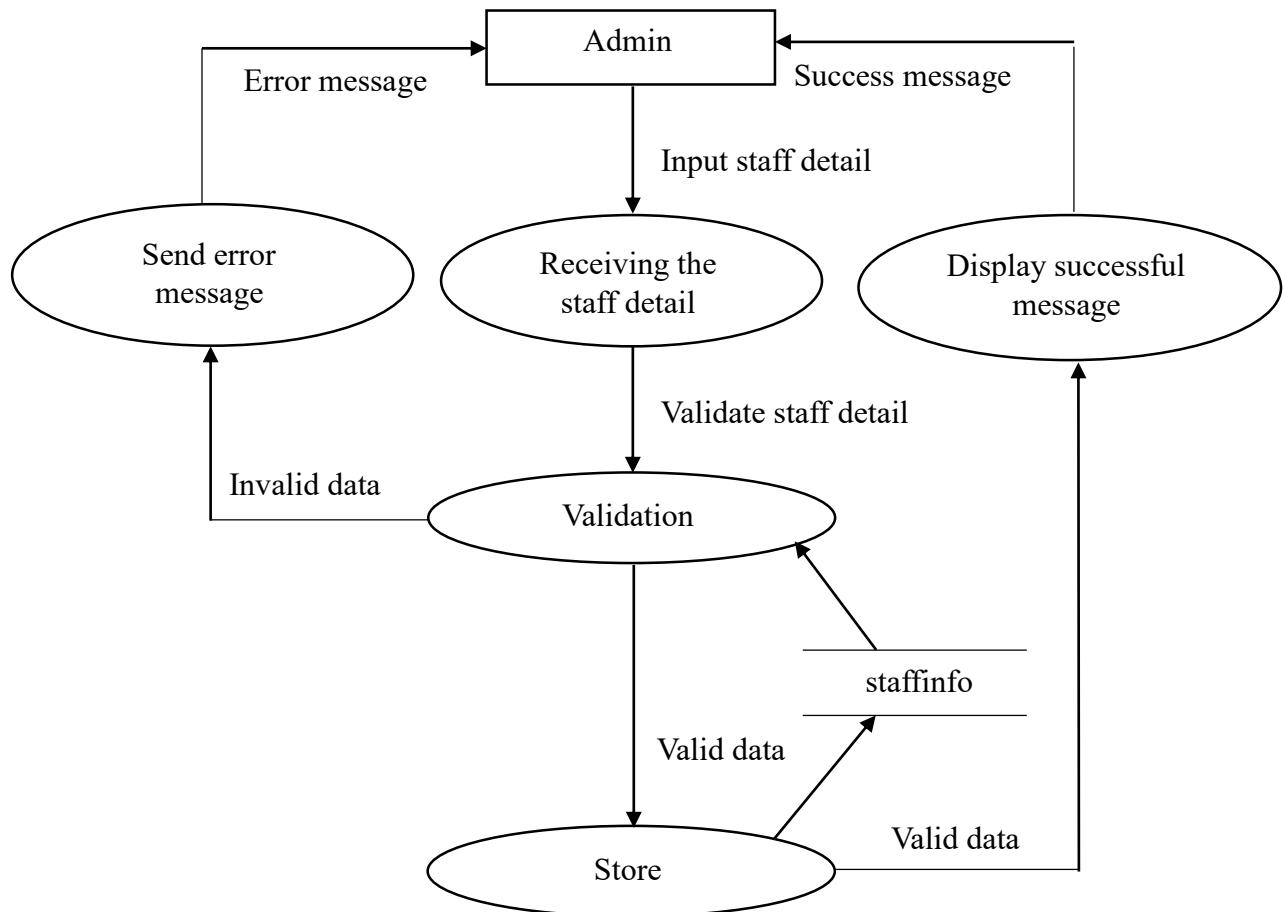


Figure 3.18 Level 2 DFD For Add Module

3.5.1.4.2 Update

3.5.1.4.2.1 Input

Staff Id.

3.5.1.4.2.2 Process

It changes the staff information and store it in database.

3.5.1.4.2.3 Output

If valid it displays updated successfully message or else displays error message.

3.5.1.4.2.4 Interface with other functional components

Add

3.5.1.4.2.5 Resource Allocation

staffinfo

3.5.1.4.2.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

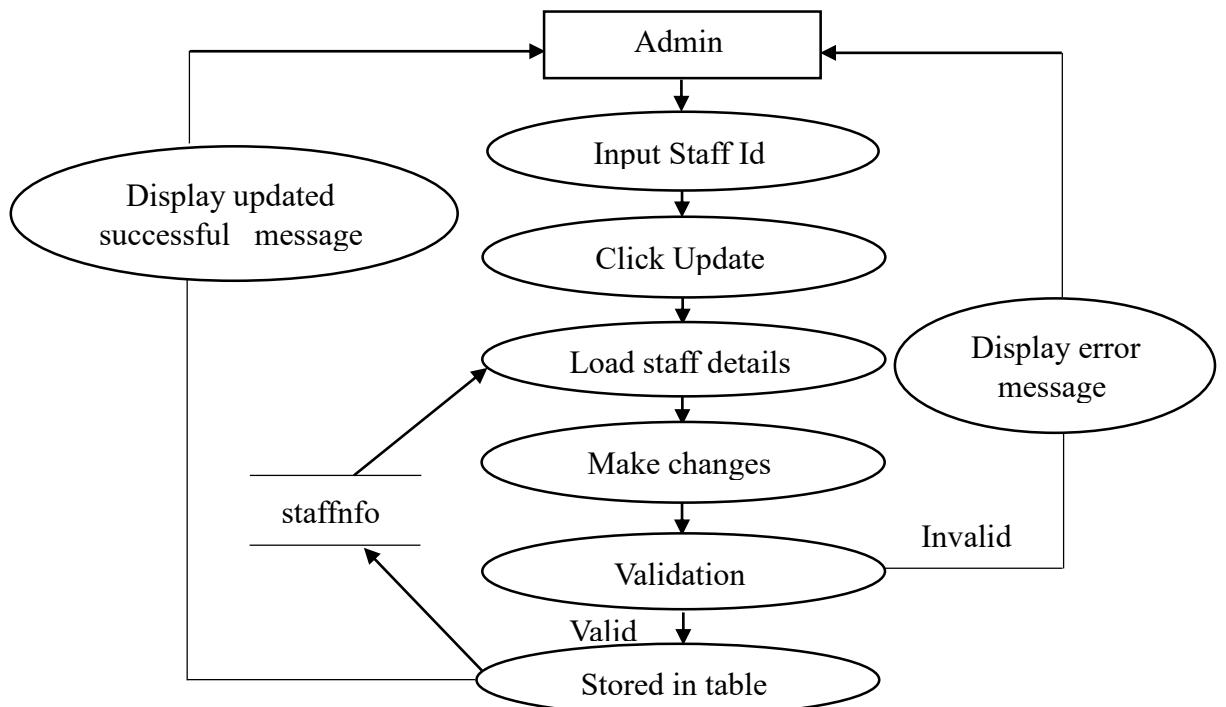


Figure 3.19 Level 2 DFD for Update Module

3.5.1.4.3 Delete

3.5.1.4.3.1 Input

Staff Id.

3.5.1.4.3.2 Process

It removes the staff information from database.

3.5.1.4.3.3 Output

Displays deleted successfully message.

3.5.1.4.3.4 Interface with other functional components

Add

3.5.1.4.3.5 Resource Allocation

staffinfo

3.5.1.4.3.6 User Interface

Button, Edit Text, Text view are used to design the user

interface.

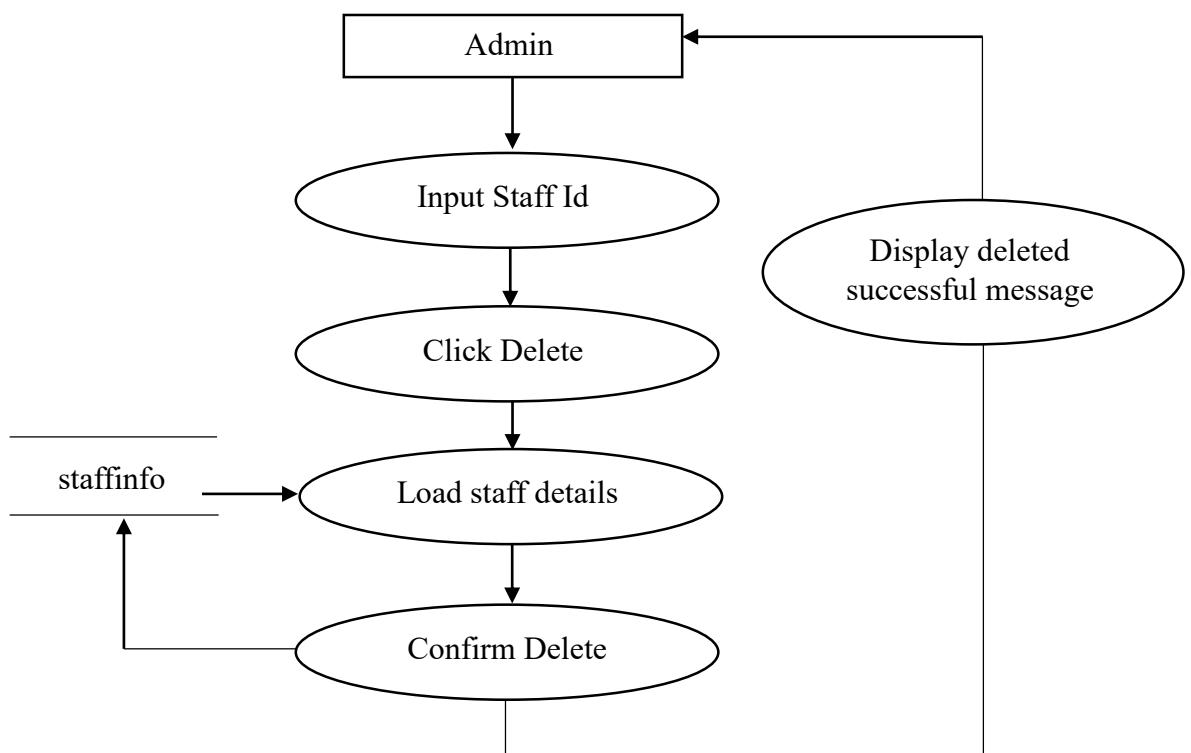


Figure 3.20 Level 2 DFD for Delete Module

3.5.1.4.4 View

3.5.1.4.4.1 Input

Button click.

3.5.1.4.4.2 Process

This functional component retrieves the staff information from database.

3.5.1.4.4.3 Output

Display the staff detail from the database.

3.5.1.4.4.4 Interface with other functional components

Add.

3.5.1.4.4.5 Resource Allocation

staffinfo

3.5.1.4.4.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

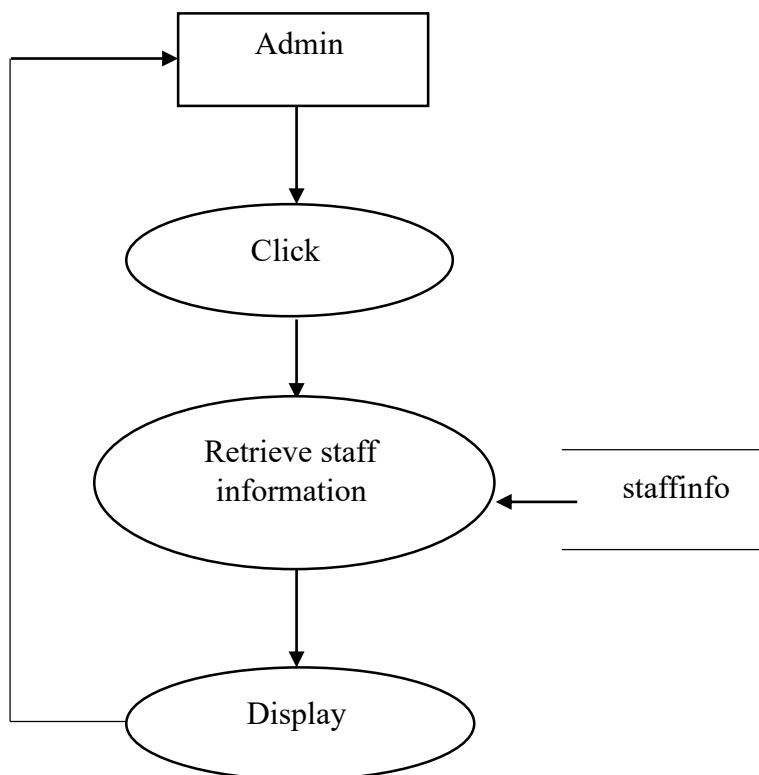


Figure 3.21 Level 2 DFD for View Module

3.5.1.5 Subject Management

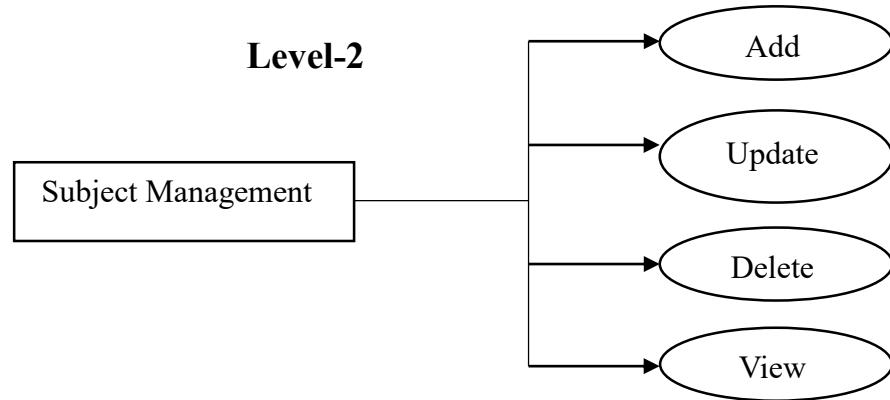


Figure 3.22 Level 2 DFD for Subject Management

3.5.1.5.1 Add

3.5.1.5.1.1 Input

Course, Year, Semester, Subject Name, Subject Code, Department, Staff Name.

3.5.2.3.1.2 Process

It allows us to input, validate and stores the subject information in database.

3.5.2.3.1.3 Output

If valid it displays inserted successfully message or else displays error message.

3.5.2.3.1.4 Interface with other functional components

This is an independent module.

3.5.2.3.1.5 Resource allocation

subinfo

3.5.2.3.1.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

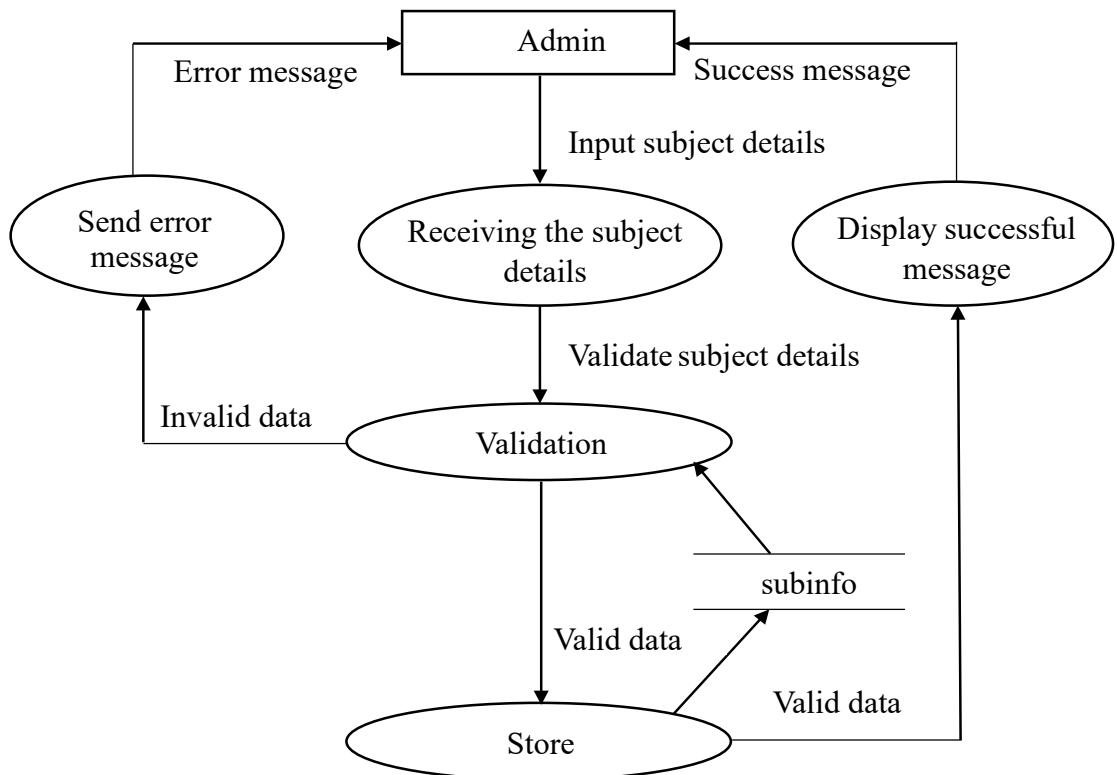


Figure 3.23 Level 2 DFD for Add Module

3.5.1.5.2 Update

3.5.1.5.2.1 Input

Subject Code

3.5.1.5.2.2 Process

It changes the subject information and store it in database.

3.5.1.5.2.3 Output

If valid it displays updated successfully message or else displays error message.

3.5.1.5.2.4 Interface with other functional components

Add

3.5.1.5.2.5 Resource Allocation

subinfo

3.5.1.5.2.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

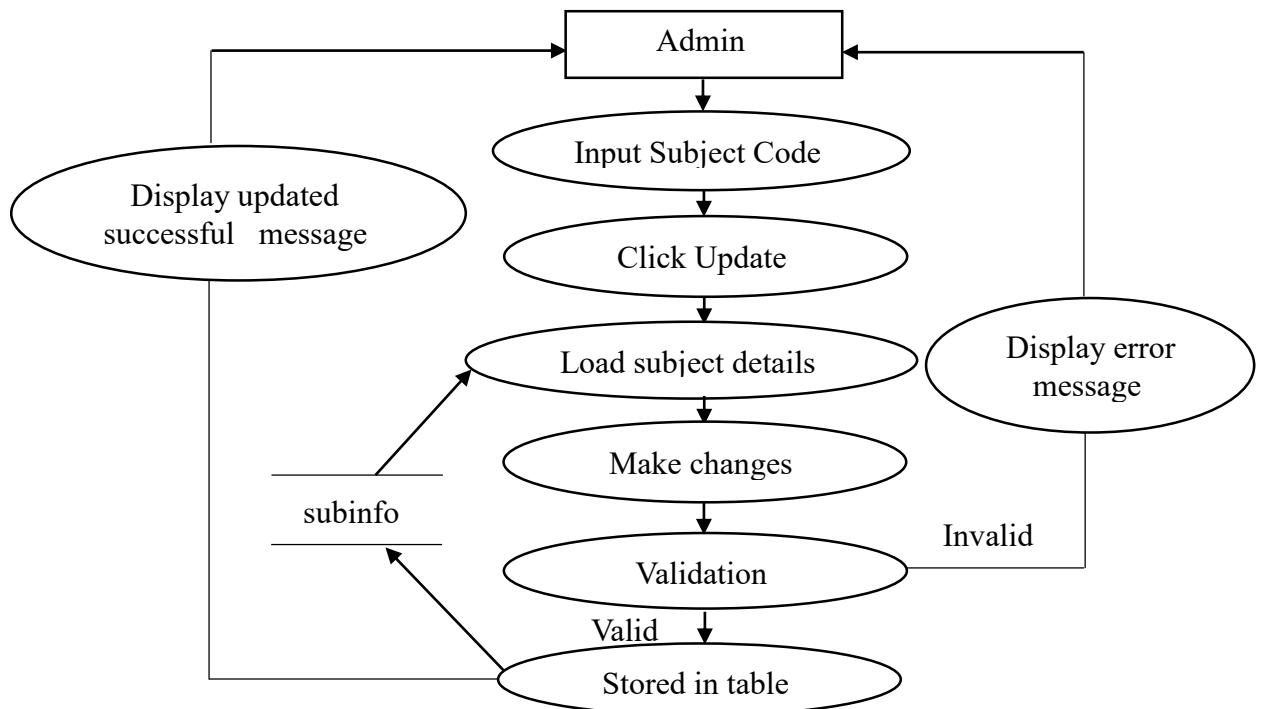


Figure 3.24 Level 2 DFD for Update Module

3.5.1.5.3 Delete

3.5.1.5.3.1 Input

Subject Code.

3.5.1.5.3.2 Process

It removes the subject information from data base.

3.5.1.5.3.3 Output

Displays deleted successfully message.

3.5.1.5.3.4 Interface with other functional components

Add

3.5.1.5.3.5 Resource Allocation

subinfo

3.5.1.5.3.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

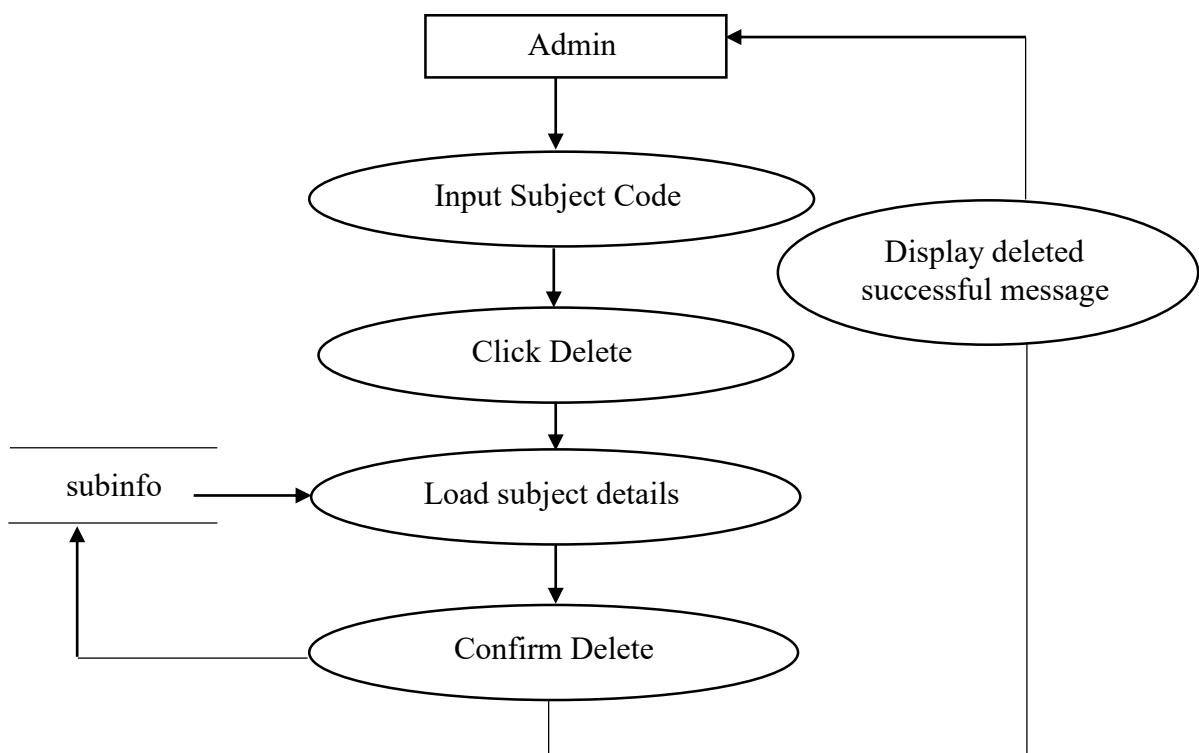


Figure 3.25 Level 2 DFD for Delete Module

3.5.1.5.4 View

3.5.1.5.4.1 Input

Button click.

3.5.1.5.4.2 Process

This functional component displays the subject information.

3.5.1.5.4.3 Output

Display the subject detail from the database.

3.5.1.5.4.4 Interface with other functional components

Add

3.5.1.5.4.5 Resource Allocation

subinfo

3.5.1.5.4.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

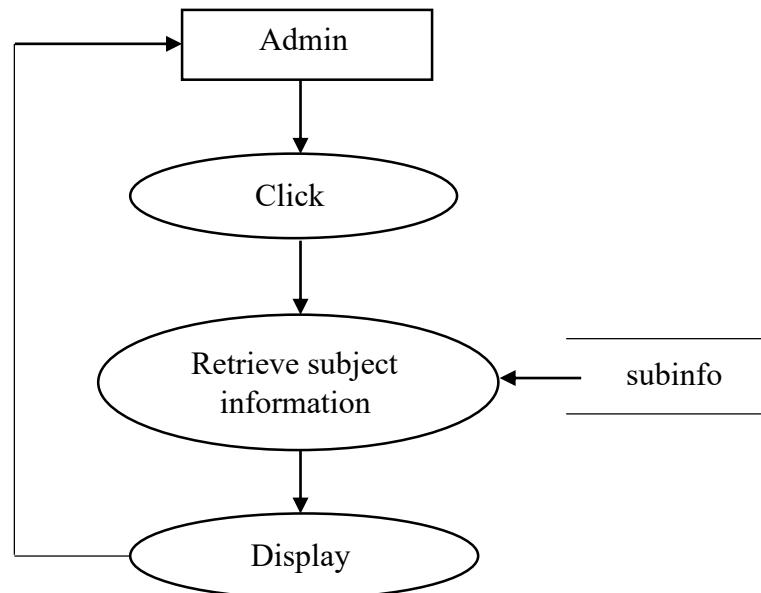


Figure 3.26 Level 2 DFD for View Module

3.5.1.6 Attendance Report

3.5.1.6.1 Input

Year, Semester, Class, Subject Name, Starting date,
Ending date.

3.5.1.6.2 Process

This functional component allows us to view the student attendance information.

3.5.1.6.3 Output

Retrieve the attendance details from the database.

3.5.1.6.4 Interface with other functional components

Today's attendance

3.5.1.6.5 Resource allocation

attendance

3.5.1.6.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

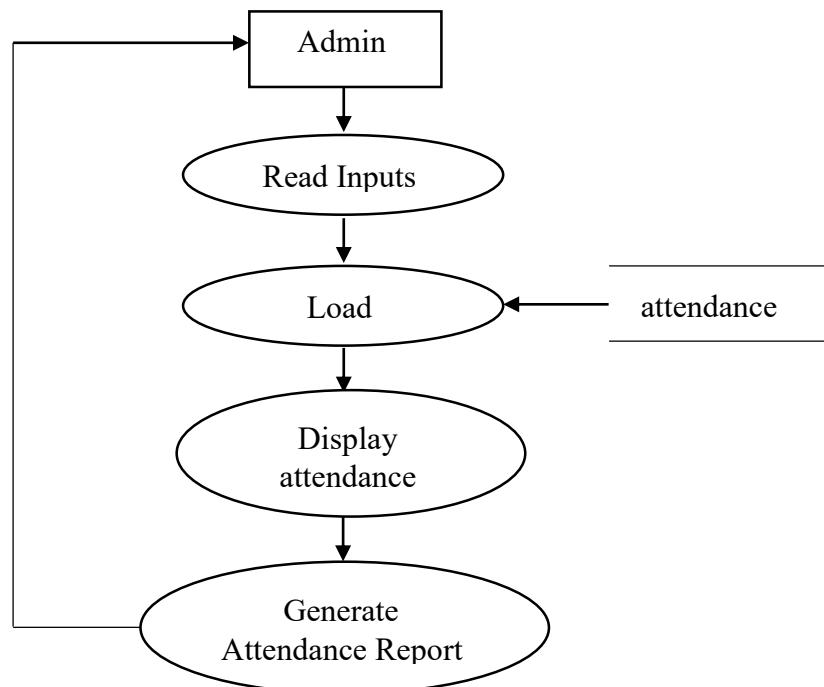


Figure 3.27 DFD for Attendance Report Module

3.5.1.7 My Profile

3.5.1.7.1 Input

Button click.

3.5.1.7.2 Process

Retrieve profile information from database.

3.5.1.7.3 Output

Display profile information.

3.5.1.7.4 Interface with others functional component

Not applicable

3.5.1.7.5 Resource allocation

admin

3.5.1.7.6. User interfaces

Button, Edit Text, Text view are used to design the user interface.

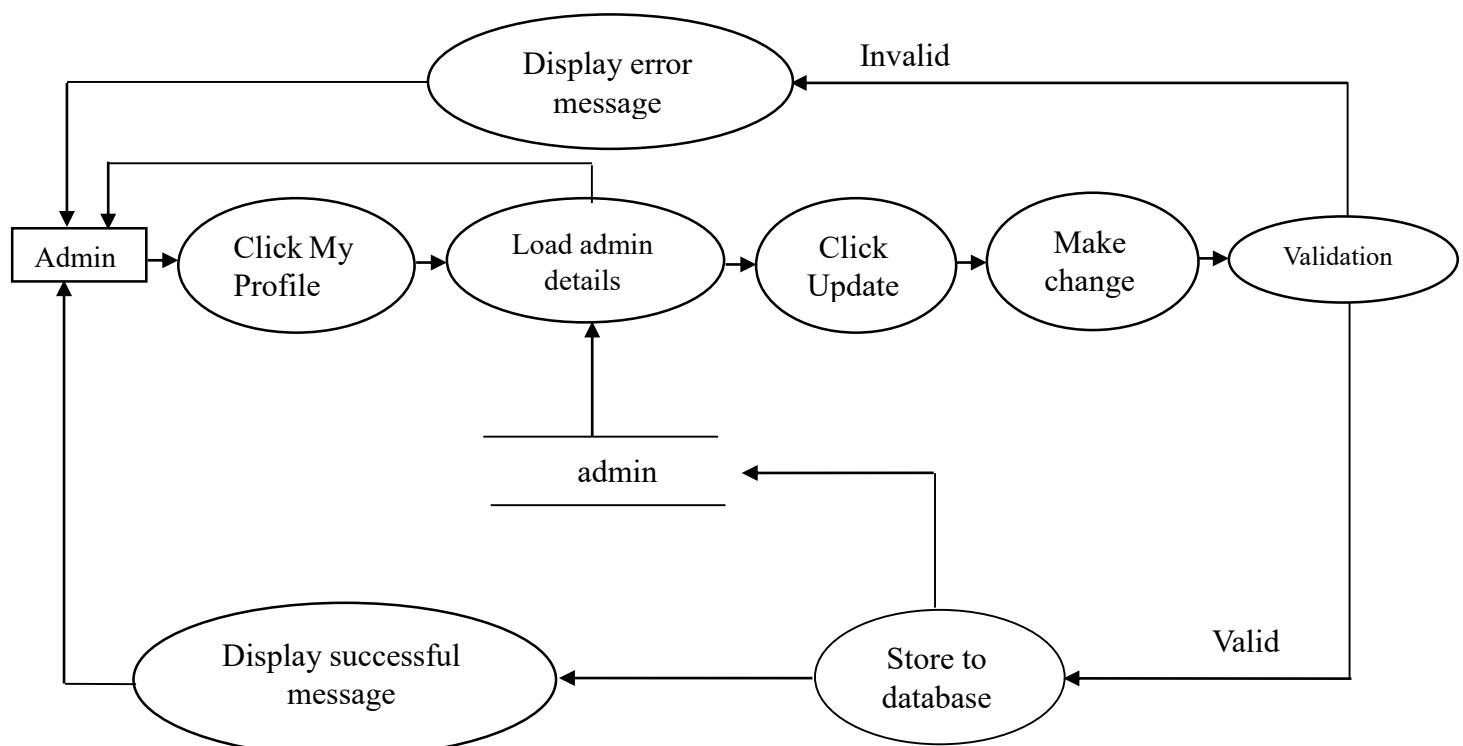


Figure 3.28 DFD for My Profile Module

3.5.1.8 Logout

This module informs that current user wishes to end the session.

3.5.1.8.1 Input

Button click.

3.5.1.8.2 Process

Ends the current session.

3.5.1.8.3 Output

After ending the session, it takes back to home page.

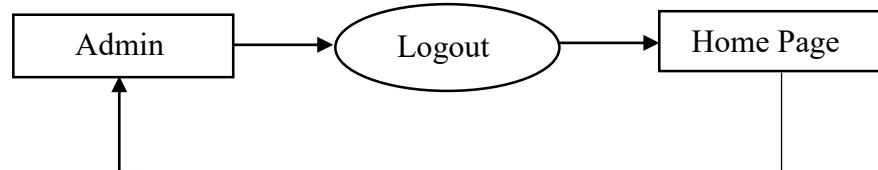


Figure 3.29 Level 1 DFD for Logout Module

3.5.2 Staff

Level-1

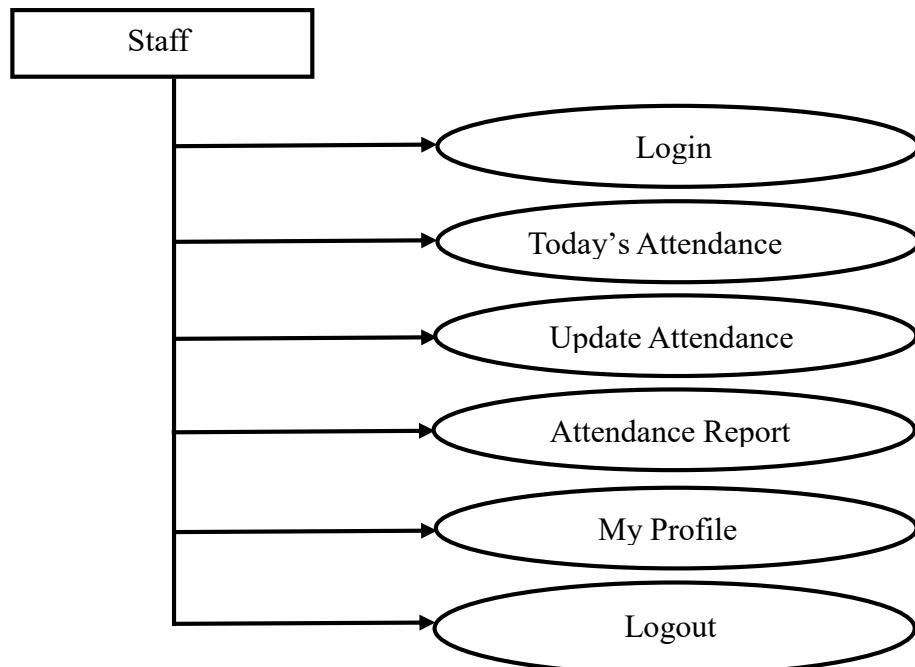


Figure 3.30 Level-1 DFD for Staff Module

3.5.2.1 Login

In this staff can login the system using Email Id and password.

3.5.2.1.1 Input

Email Id and Password

3.5.2.1.2 Process

Validates for Email Id and password.

3.5.2.1.3 Output

Respective page will be loaded.

3.5.2.1.4 Interface with other functional components

Add

3.5.2.1.5 Resource Allocation

staffinfo

3.5.2.1.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

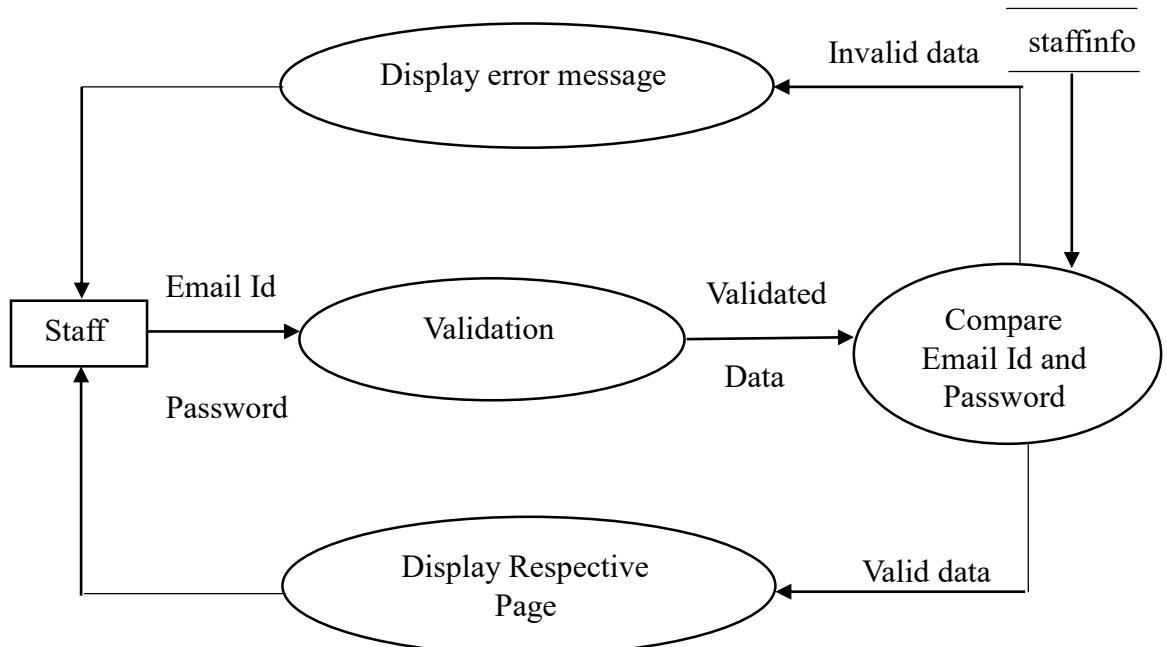


Figure 3.31 Level 1 DFD for Login Module

3.5.2.2 Today's Attendance

3.5.2.2.1 Input

Year, Course, Sem, Subject Name, tick checkbox.

3.5.2.2.2 Process

It allows us to input and store the student attendance information to database.

3.5.2.2.3 Output

Displays attendance submitted successfully.

3.5.2.2.4 Interfaces with the other functional component

Add

3.5.2.2.5 Resource allocation

attendance

3.5.2.2.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

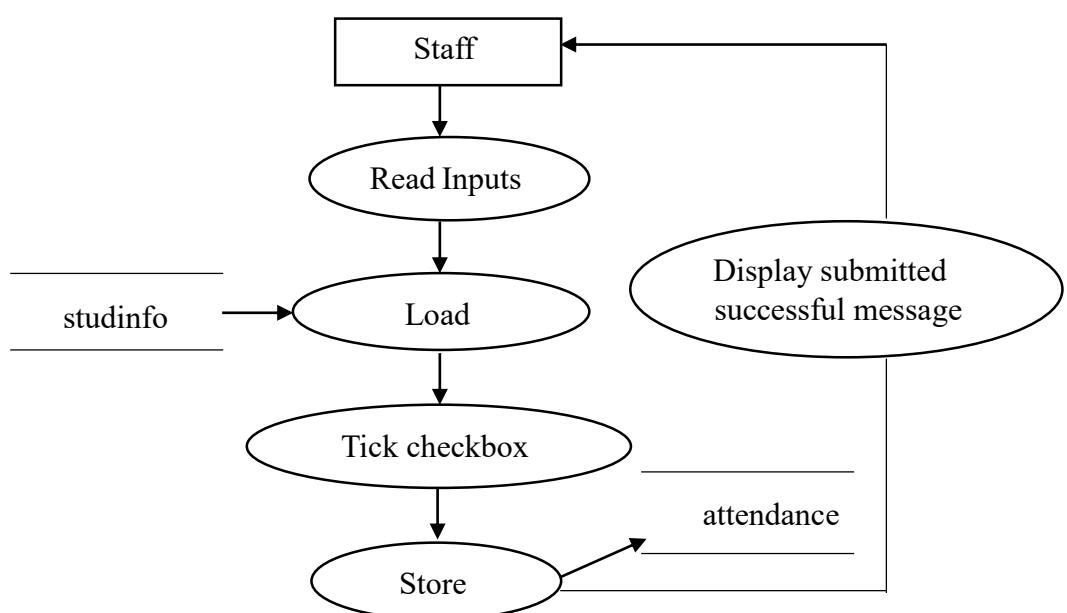


Figure 3.32 DFD for Today's Attendance Module

3.5.2.3 Update Attendance

3.5.2.3.1 Input

Year, Course, Subject Name, Sem, date.

3.5.2.3.2 Process

It allows staff to update and store the student attendance information to database.

3.5.2.3.3 Output

Displays attendance updated successfully message.

3.5.2.3.4 Interfaces with the other functional components

Today's attendance.

3.5.2.3.5 Resource allocation

attendance

3.5.2.3.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

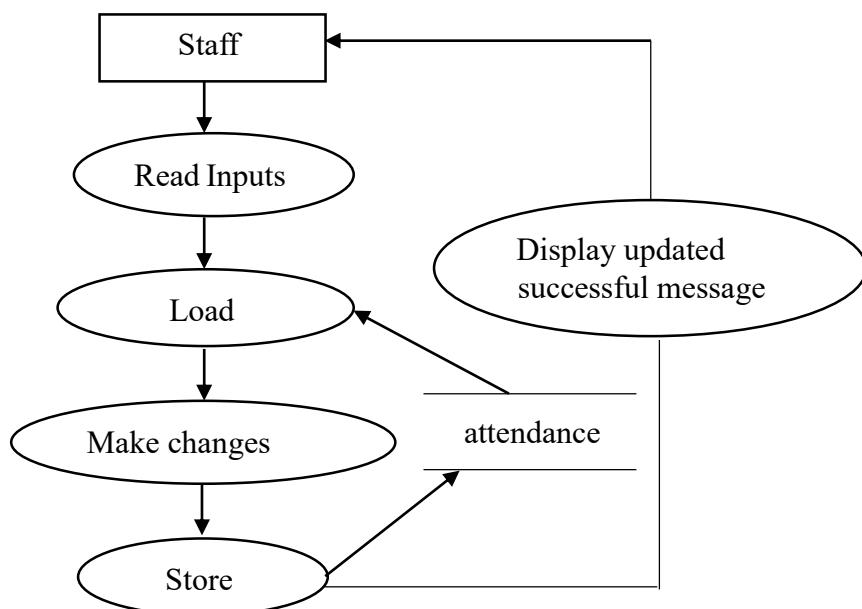


Figure 3.33 DFD for Update Attendance Module

3.5.2.4 Attendance Report

3.5.2.4.1 Input

Year, Course, Sem, Subject Name, Starting date, Ending date.

3.5.2.4.2 Process

It allows us to retrieve the student attendance information from database.

3.5.2.4.3 Output

Displays the attendance details from the database.

3.5.2.4.4 Interfaces with the other functional components

Today's attendance

3.5.2.4.5 Resource allocation

attendance

3.5.2.4.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

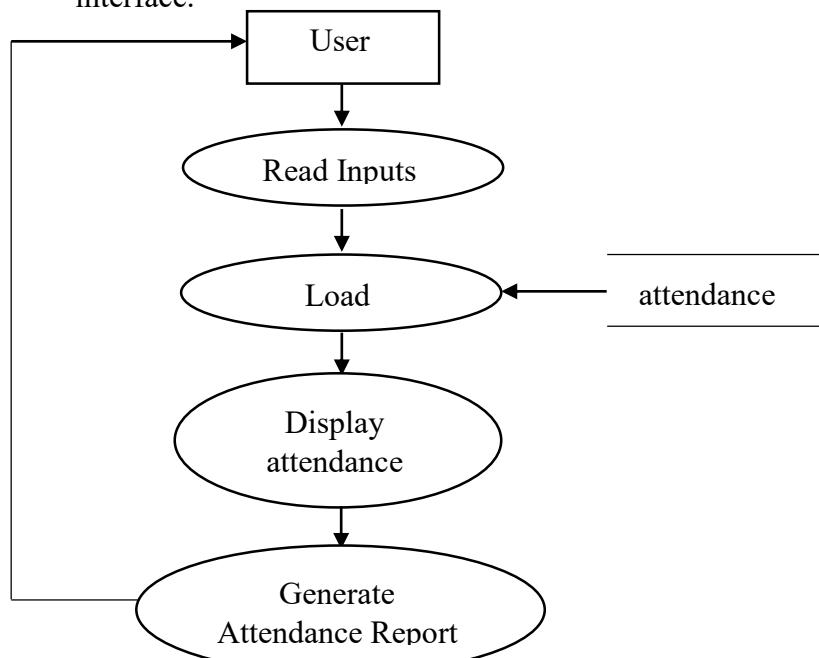


Figure 3.34 DFD for Attendance Report Module

3.5.2.5 My Profile

3.5.2.5.1 Input

Button click.

3.5.2.5.1 Process

Retrieve profile from database.

3.5.2.5.3 Output

Display profile information.

3.5.2.5.4 Interface with others functional component

Add

3.5.2.5.5 Resource allocation

staffinfo

3.5.2.5.6. User interfaces

Button, Edit Text, Text view are used to design the user interface.

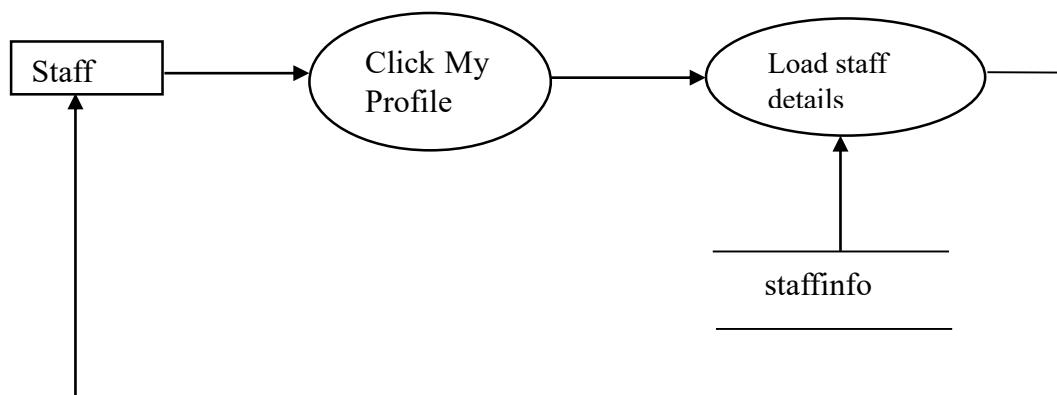


Figure 3.35 DFD for My Profile Module

3.5.2.6 Logout

This module informs that current staff wishes to end the session.

3.5.2.6.1 Input

Button click.

3.5.2.6.2 Process

Ends the current session.

3.5.2.6.3 Output

After ending the session, it takes back to home page.

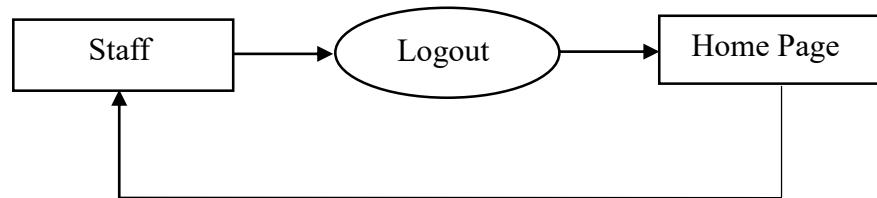


Figure 3.36 DFD for Logout Module

3.5.3 Student

Level-1

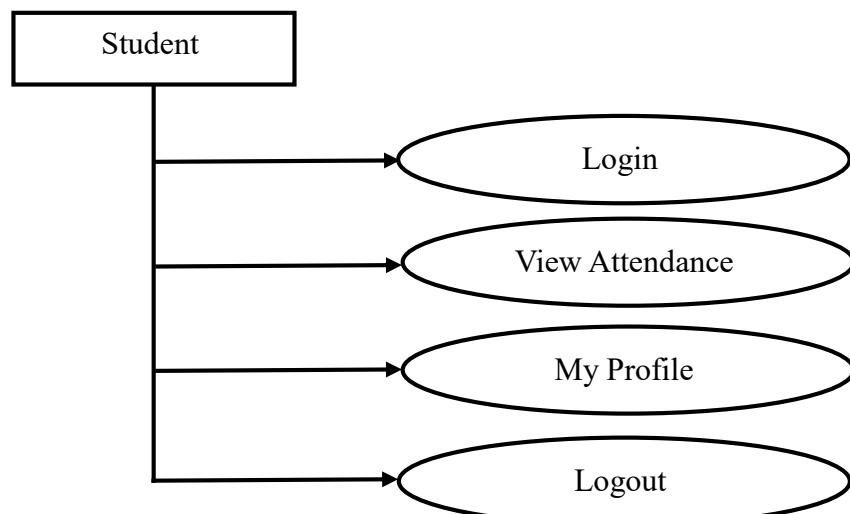


Figure 3.37 DFD for student Module

3.5.3.1 Login

In this student can login the system using Email Id and password.

3.5.3.1.1 Input

Email Id and Password

3.5.3.1.2 Process

Validates for Email Id and password.

3.5.3.1.3 Output

Respective page will be loaded.

3.5.3.1.4 Interface with other functional components

Add

3.5.3.1.5 Resource Allocation

studinfo

3.5.3.1.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

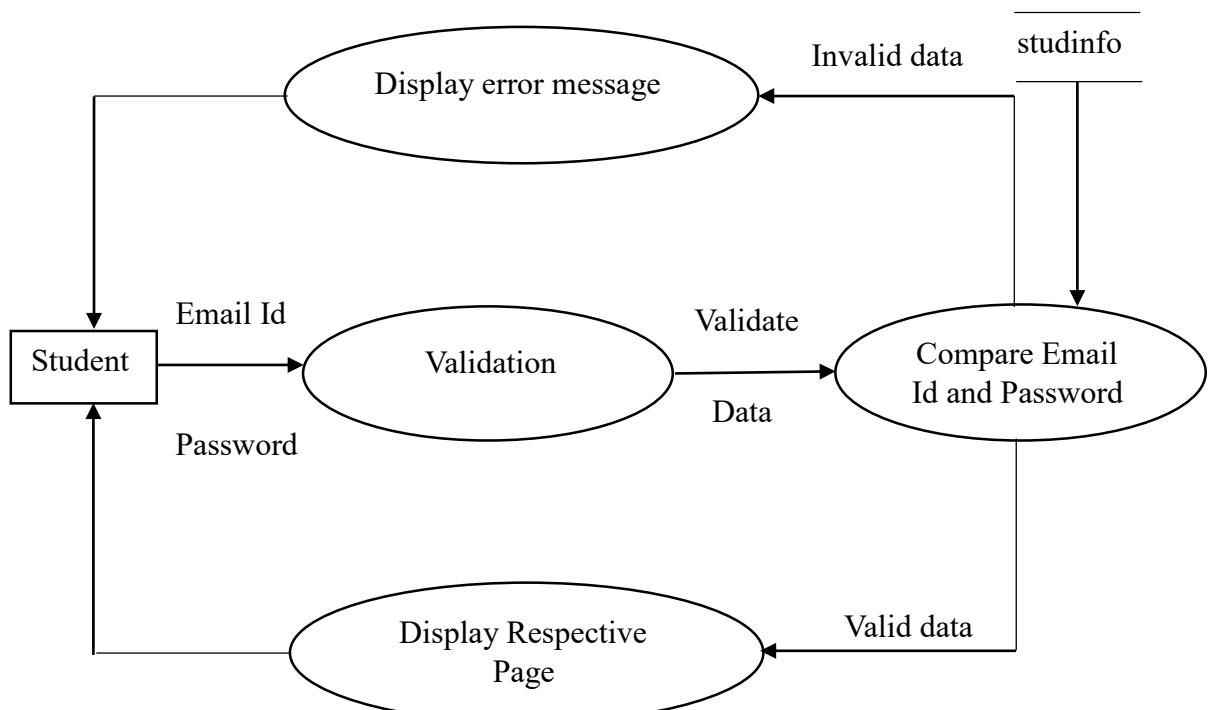


Figure 3.38 Level 1 DFD for Login Module

3.5.3.2 View Attendance

3.5.3.2.1 Input

Subject Name, Starting date, Ending date.

3.5.3.2.2 Process

It retrieves attendance information of the student from database.

3.5.3.2.3 Output

Display attendance of the student from database.

3.5.3.2.4 Interfaces with the other functional components

Today's attendance

3.5.3.2.5 Resource allocation

attendance.

3.5.3.2.6 User Interface

Button, Edit Text, Text view are used to design the user interface.

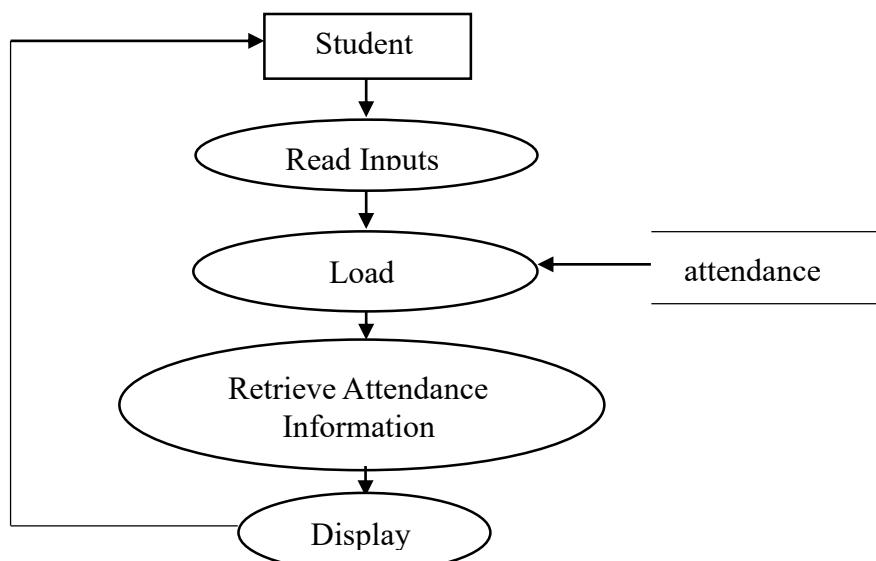


Figure 3.39 DFD for View Attendance Module

3.5.3.3 My Profile

3.5.3.3.1 Input

Button clicks

3.5.3.3.1 Process

Retrieve profile information from database.

3.5.3.3.3 Output

Display profile information.

3.5.3.3.4 Interface with others functional component

Add

3.5.3.3.5 Resource allocation

studinfo

3.5.3.3.6. User interfaces

Button, Edit Text, Text view are used to design the user interface.

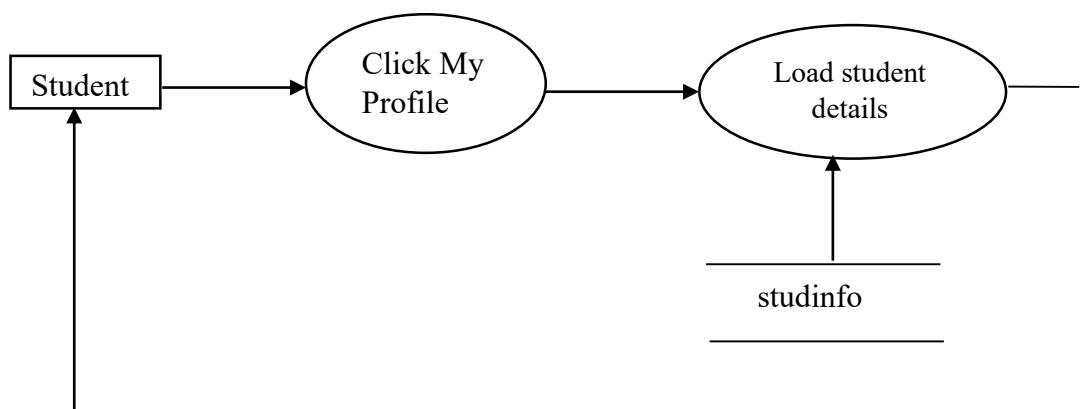


Figure 3.40 DFD for My Profile Module

3.5.3.4 Logout

This module informs that current staff wishes to end the session.

3.5.3.4.1 Input

Button click.

3.5.3.4.2 Process

Ends the current session.

3.5.2.4.3 Output

After ending the session, it takes back to home page.

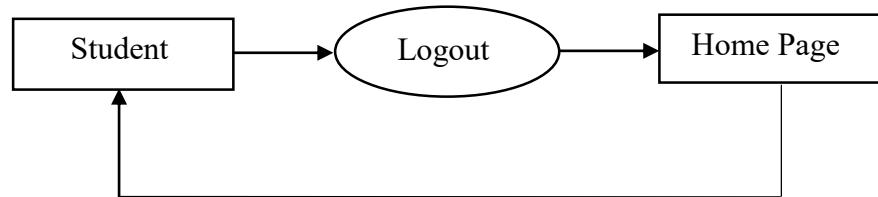


Figure 3.41 DFD for Logout Module



DATABASE DESIGN

4. DATABASE DESIGN

4.1 Introduction

Database is a collection of related data. Relational database stores data in a table or relations. The data stored in a relation are arranged in records. Each record consists of set of attributes the table that is used to design software, its attributes, data types, constraints and relationship among those tables.

The design process consists of the following steps:

- Determine the purpose of our database.
- Find and organize the information required.
- Divide the information into tables.
- Turn information items into columns.
- Specify primary keys.
- Set up the table relationships.
- Refine our design.
- Apply the normalization rules.

4.2 Purpose and Scope

Purpose

- **Avoid Redundant Data**

The table in the database should be constructed following standards and with utmost decision. It should have different fields and minimize redundant data. The table should always have a Primary Key that would be a unique id.

- **Faultless Information**

The database should follow the standard and conventions and provide meaningful information useful to the organization.

- **Data Integrity**

Integrity assists in guaranteeing that the values are valid and faultless. Data Integrity is set to tables, relationships, etc.

- **Modify**

The database developed should be worked upon with the conventions and standards, so that it can be easily modified whenever the need arises.

Scope

- Normalization of database.
- Imposing Integrity Constraint.
- Establishing the Relation between the tables.
- Accessing the data from multiple tables.

4.3 Database Identification

This identification of database by unique name given to the various database objects. The identifier is the name of database object. The following are the various database objects.

- Tables
- Columns
- Views
- Sequences
- Indexes
- Stored Procedures

4.4 Schema Information

The database schema is its structure described in a formal language supported by the database management system (DBMS). The term “schema” refers to the organization of database as a blueprint of how the database is constructed (divided into database in the case of relational database).

In a relational database, the schema defines the tables, fields, relationships, views, indexes, packages, procedures, functions, queues, triggers, types, sequences, materialized views.

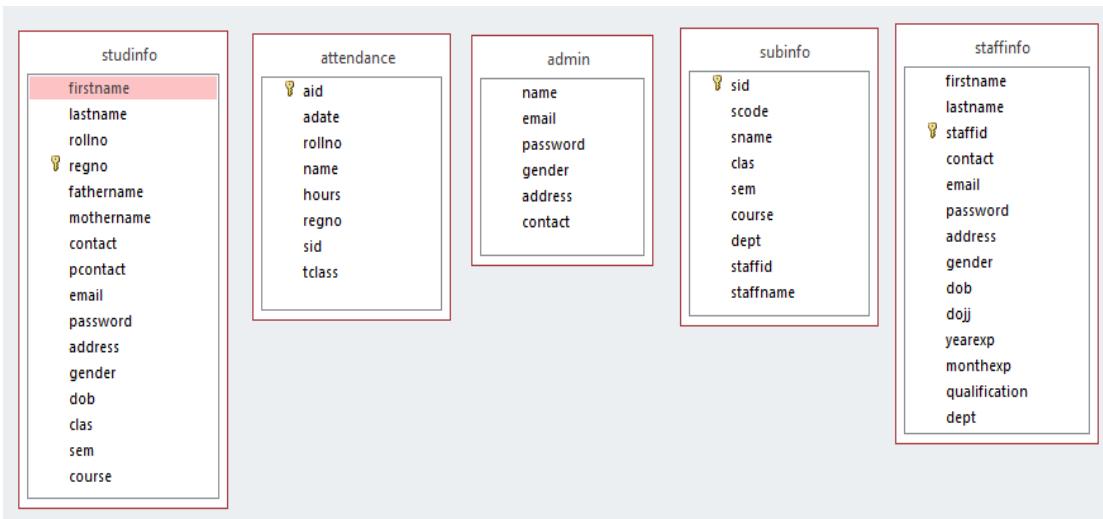


Figure 4.1 Schema Diagram without relation

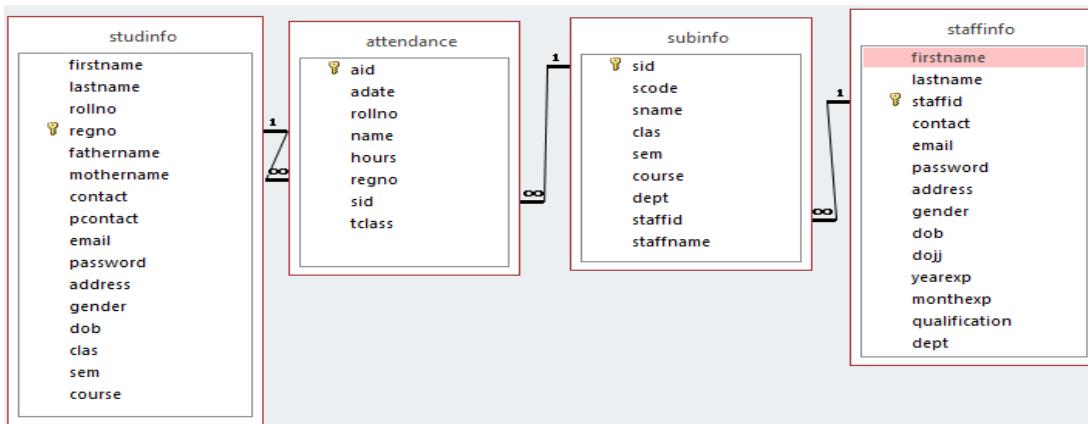


Figure 4.2 Schema Diagram with relation

4.5 Table Definition

4.5.1 admin

Field Name	Data Type	Size	Constraint	Description
name	varchar	20	NOT NULL	Name
email	varchar	50	NOT NULL	Email
password	varchar	30	NOT NULL	Password
gender	varchar	10	NOT NULL	Gender
address	varchar	20	NOT NULL	Address
contact	bigint	10	NOT NULL	Contact

Table 4.1 admin

4.5.2 staffinfo

Field Name	Data Type	Size	Constraint	Description
firstname	varchar	30	NOT NULL	First Name
lastname	varchar	30	DEFAULT NULL	Last Name
staffid	int	15	PRIMARY KEY	Staff Id
contact	bigint	10	NOT NULL	Contact
email	varchar	50	NOT NULL	Email
password	varchar	20	NOT NULL	Password
address	varchar	60	NOT NULL	Address
gender	varchar	10	NOT NULL	Gender
dob	date		NOT NULL	Date of Birth
dojj	date		NOT NULL	Date of Joining
yearexp	int	2	NOT NULL DEFAULT 0	Year of Experience
monthexp	int	2	NOT NULL DEFAULT 0	Month of Experience
qualification	varchar	20	NOT NULL	Qualification
dept	varchar	20	NOT NULL	Department

Table 4.2 staffinfo

4.5.3 studinfo

Field Name	Data Type	Size	Constraint	Description
firstname	varchar	15	NOT NULL	First Name
lastname	varchar	15	DEFAULT NULL	Last Name
rollno	bigint	15	NOT NULL	Roll Number
regno	bigint	15	PRIMARY KEY	Register Number
fathername	varchar	20	NOT NULL	Father Name
mothername	varchar	20	NOT NULL	Mother Name
contact	bigint	10	NOT NULL	Contact
pcontact	bigint	10	DEFAULT NULL	Parent Contact
email	varchar	40	NOT NULL	Email
password	varchar	15	NOT NULL	Password
address	varchar	50	NOT NULL	Address
gender	varchar	10	NOT NULL	Gender
dob	date		NOT NULL	Date of Birth
clas	varchar	10	NOT NULL	Department
sem	varchar	10	NOT NULL	Semester
course	varchar	15	NOT NULL	Course

Table 4.3 studinfo

4.5.4 subinfo

Field Name	Data Type	Size	Constraint	Description
scode	varchar	10	PRIMARY KEY	Subject Code
sname	varchar	20	NOT NULL	Subject Name
clas	varchar	15	NOT NULL	Class
Sem	varchar	15	NOT NULL	Semester
course	varchar	15	NOT NULL	Course
dept	varchar	20	NOT NULL	Department
staffname	varchar	30	NOT NULL	Staff Name
staffid	int	15	NOT NULL	Staff Id

Table 4.4 subinfo

4.5.5 attendnace

Field Name	Data Type	Size	Constraint	Description
aid	bigint	15	PRIMARY KEY	Attendance Id
adate	bigint	15	FOREIGN KEY	Attendance Date
rollno	bigint	15	DEFAULT NULL	Roll Number
name	varchar	30	NOT NULL	Name
hours	date		NOT NULL	Hours
regno	varchar	10	NOT NULL	Register Number
scode	varchar	10	NOT NULL	Subject code
staffid	bigint	10	NOT NULL	Staff Id

Table 4.5 attendance

4.6 Physical Design

The physical design is a where we translate schemas into actual database structure.

- Entity to Table
- Tuple to Rows
- Attribute to Column
- Primary Key and Alternate Key to Unique Index
- Domain into Constraints

4.7 Data Dictionary

A data dictionary is a file or a set of files that includes a database's metadata. The data dictionary holds records about other object in the database, such as data ownership, data relationships to other objects, and other data.

The data dictionary, in general, includes information about the following;

- Name of the data item
- Aliases
- Description/purpose
- Related data items
- Range of values
- Data structure definition

4.8 ER Diagram

ER-modelling is a data modelling method used in software engineering to produce conceptual data model of an information system. Diagram created using this ER-modelling method are called Entity-Relationship Diagrams or ER Diagram or ERDs.

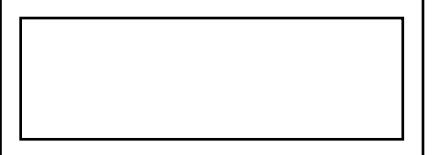
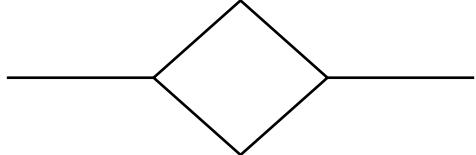
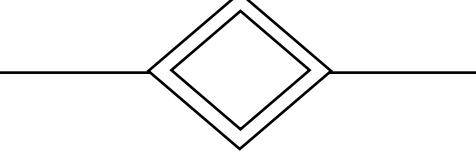
Symbols	Conversion
	Entity
	Weak Entity
	Relation
	Identity Relation
	Attribute
	Derived Attribute
	Cardinality ratio 1:N from E1 to E2

Table 4.6 ER Diagram

Components of an ER Diagrams

1. Entity

An entity can be a real –word object, either animate or inanimate, that can be merely identifiable.

An entity is denoted as a rectangle in an ER diagram. For example, in a school database, student, teachers, classes and courses offered can be treated as entities. All these entities have some attributes or properties that give them their identity.

Entity Set

An entity set is a collection of related types of entities.

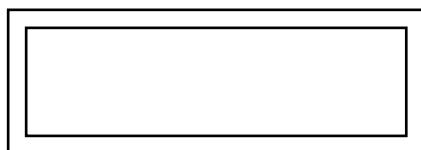
Strong Entity

An entity with uniquely identified by its attribute.



Weak Entity

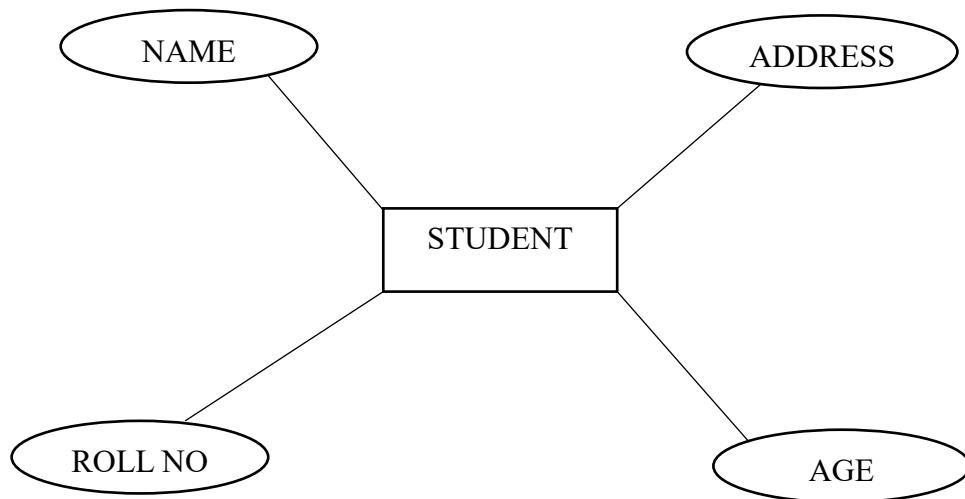
In a relational database a weak entity is an entity that cannot be uniquely identified by its attributes alone.



2. Attributes

Entities are denoted utilizing their properties, known as attributes. All attributes have values. For example, a student entity may have name, class and age as attributes.

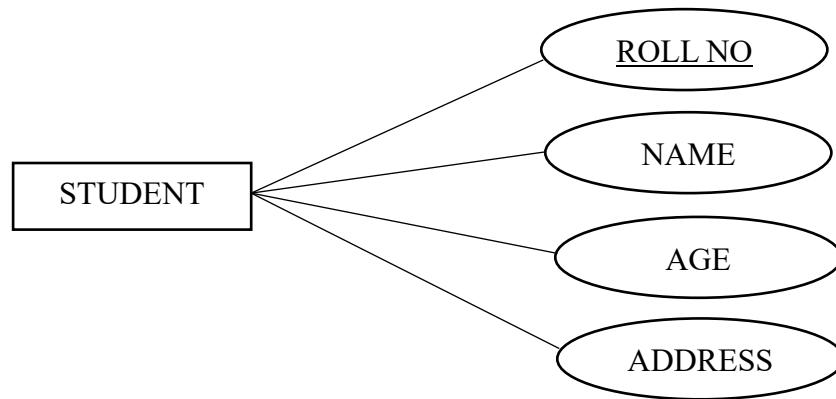
There exists a domain or range of values that can be assigned to attributes. For example, a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative.



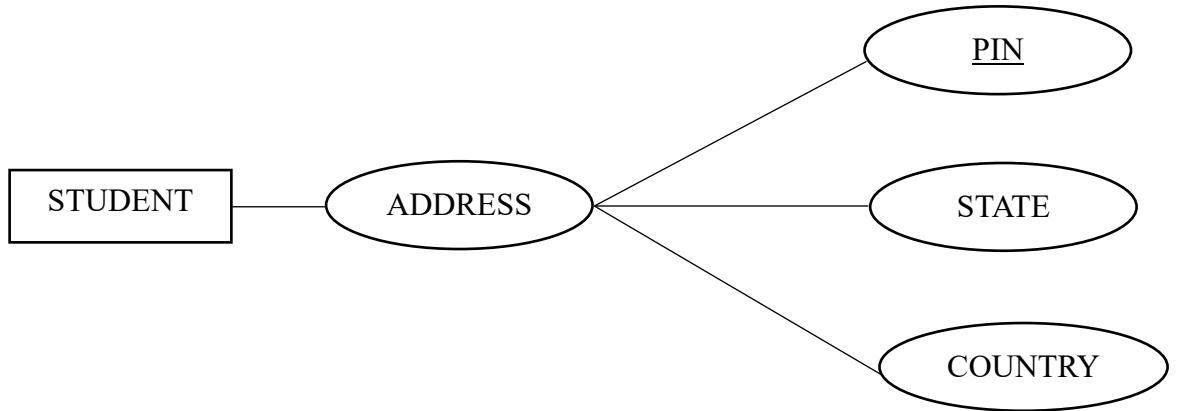
There are five types of Attributes:

1. Key Attribute
2. Composite Attribute
3. Single-valued Attribute
4. Multi-valued Attribute
5. Derived Attribute

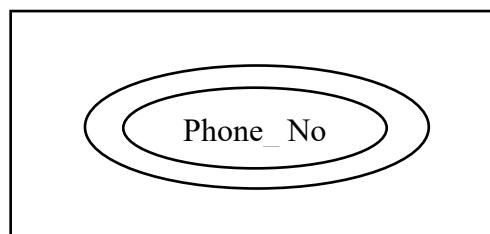
1. **Key attribute:** Key is an attribute or collection of attributes that are uniquely identifies an entity among the entity set. For example, the register number of a student makes him identifiable among students.



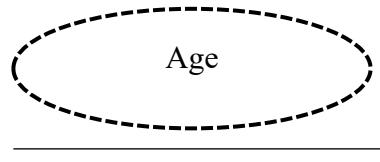
- 2. Composite attribute:** An attribute that is combination of other attributes is called composite attribute. For example, in student entity, the student address is a composite attribute as an address is composed of other characteristics such as pin code, state, country.



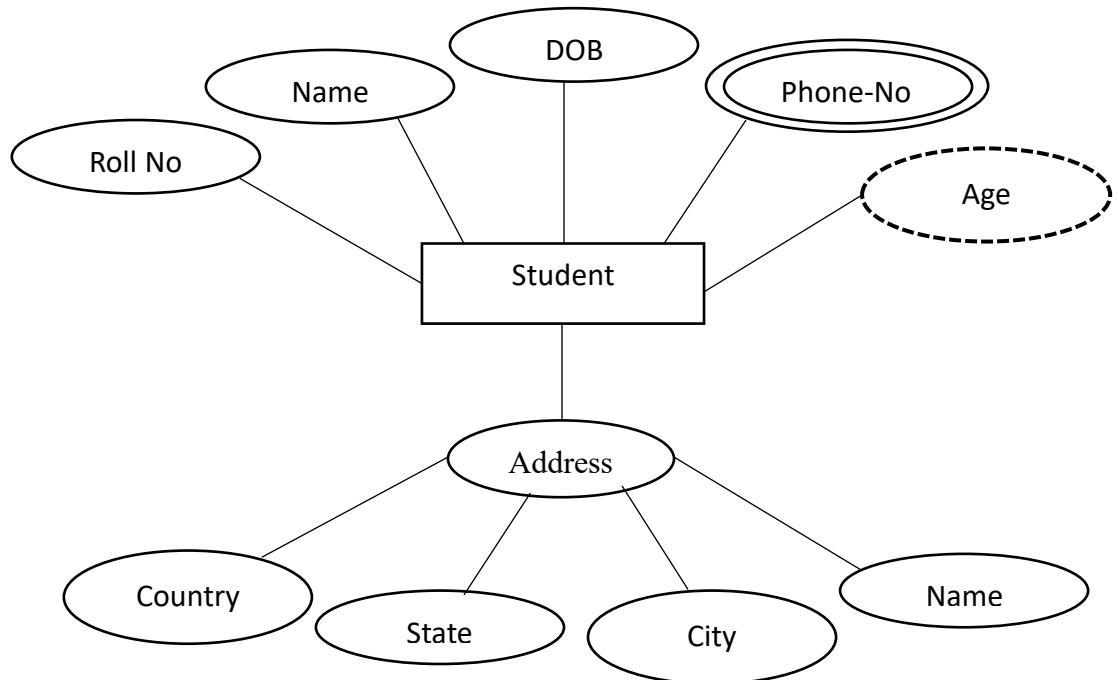
- 3. Single-valued attribute:** Single-valued attribute contain a single value. For example, Social_Security_Number.
- 4. Multi-valued attribute:** If an attribute can have more than one value, it is known as multivalued attribute. Multi-valued attributes are depicted by the double ellipse. For example, a person can have more than one phone number, email address, etc.



- 5. Derived attribute:** Derived attributes are the attribute that does not exist in the physical database, but their values are derived from other attributes present in the database. For example, age can be derived from date_of_birth. In the ER diagram, Derived attributes are depicted by the dashed ellipse.



The complete entity type student with its attributes can be represented as:



3. Relationships

The association among entities is known as relationship. Relationships are represented by the diamond-shaped box. For example, an employee works_at a department, a student enrolls in a course. Here, Works_at and enrolls are called relationships.

Degree of a relationship set

The number of participating entities in a relationship describes the degree of the relationship. The three most common relationships in E-R models are:

1. Unary (degree1)
2. Binary (degree2)
3. Ternary (degree3)

1. Unary relationship: This is also called recursive relationships. It is a relationship between the instances of one entity type. For example, one person is married to only one person.

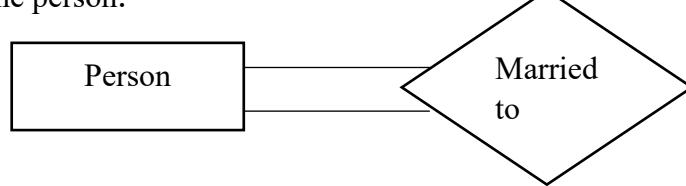


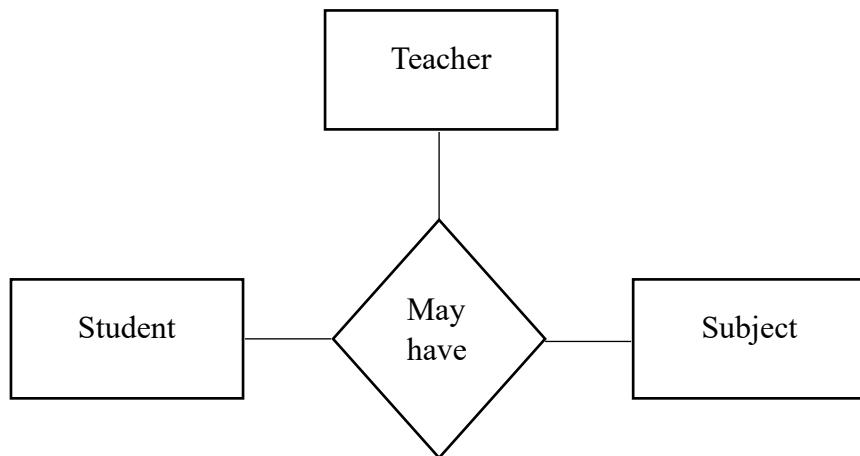
Fig: Unary Relation

2. Binary relationship: It is a relationship between the instances of two entity types. For example, the Teacher teaches the subject.



Fig: Binary Relation

3. Ternary relationship: It is a relationship amongst instances of three entity types. In fig, the relationships "may have" provide the association of three entities, i.e., TEACHER, STUDENT, and SUBJECT. All three entities are many-to-many participants. There may be one or many participants in a ternary relationship.



(Many to Many)

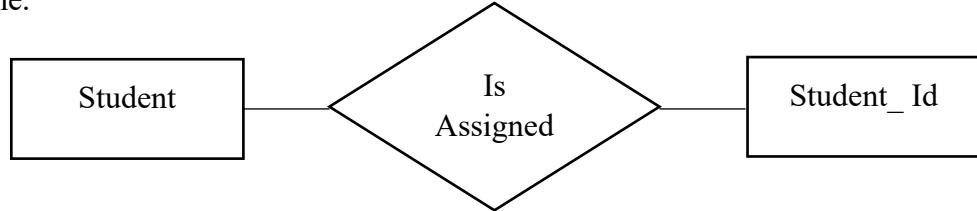
Fig: Ternary relationship

Cardinality Ratio

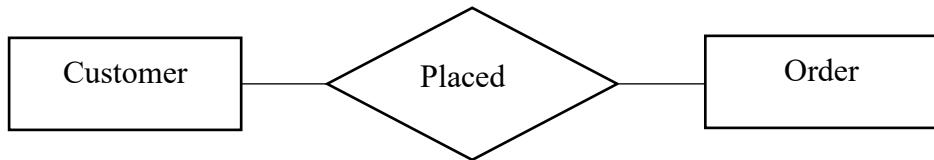
Cardinality describes the number of entities in one entity set, which can be associated with the number of entities of other sets via relationship set.

Types of Cardinalities

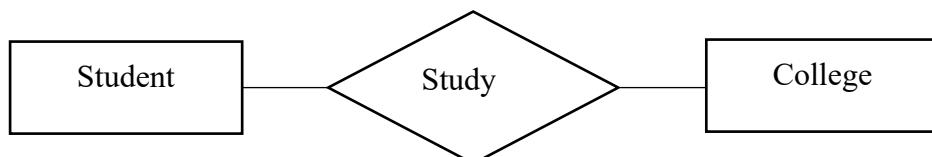
1. One to One: One entity from entity set A can be contained with at most one entity of entity set B and vice versa. Let us assume that each student has only one student ID, and each student ID is assigned to only one person. So, the relationship will be one to one.



2. One to many: When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationships. For example, a client can place many orders; a order cannot be placed by many customers.



3. Many to One: More than one entity from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A. For example - many students can study in a single college, but a student cannot study in many colleges at the same time.

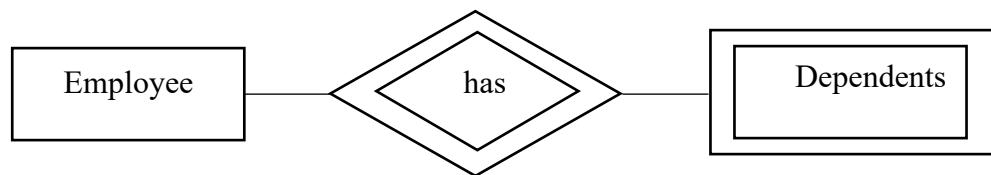


4. Many to Many: One entity from A can be associated with more than one entity from B and vice-versa. For example, the student can be assigned to many projects, and a project can be assigned to many students.



Identifying relationship

An identifying relationship is a relationship between two entities in which an instance of a child entity is identified through its association with a parent entity, which means the child entity is dependent on the parent entity for its identity and cannot exist without it.



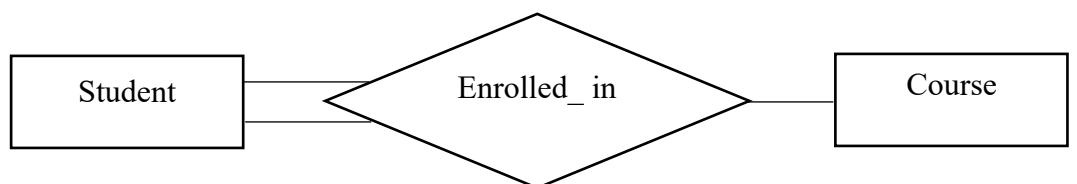
Participation Constraints

The participation constraint specifies the number of instances of an entity can participate in a relationship set.

Total participation – Each entity is involved in the relationship. Total participation is represented by double lines.

Partial participation – Not all entities are involved in the relationship. Partial participation is represented by single lines.

Example:



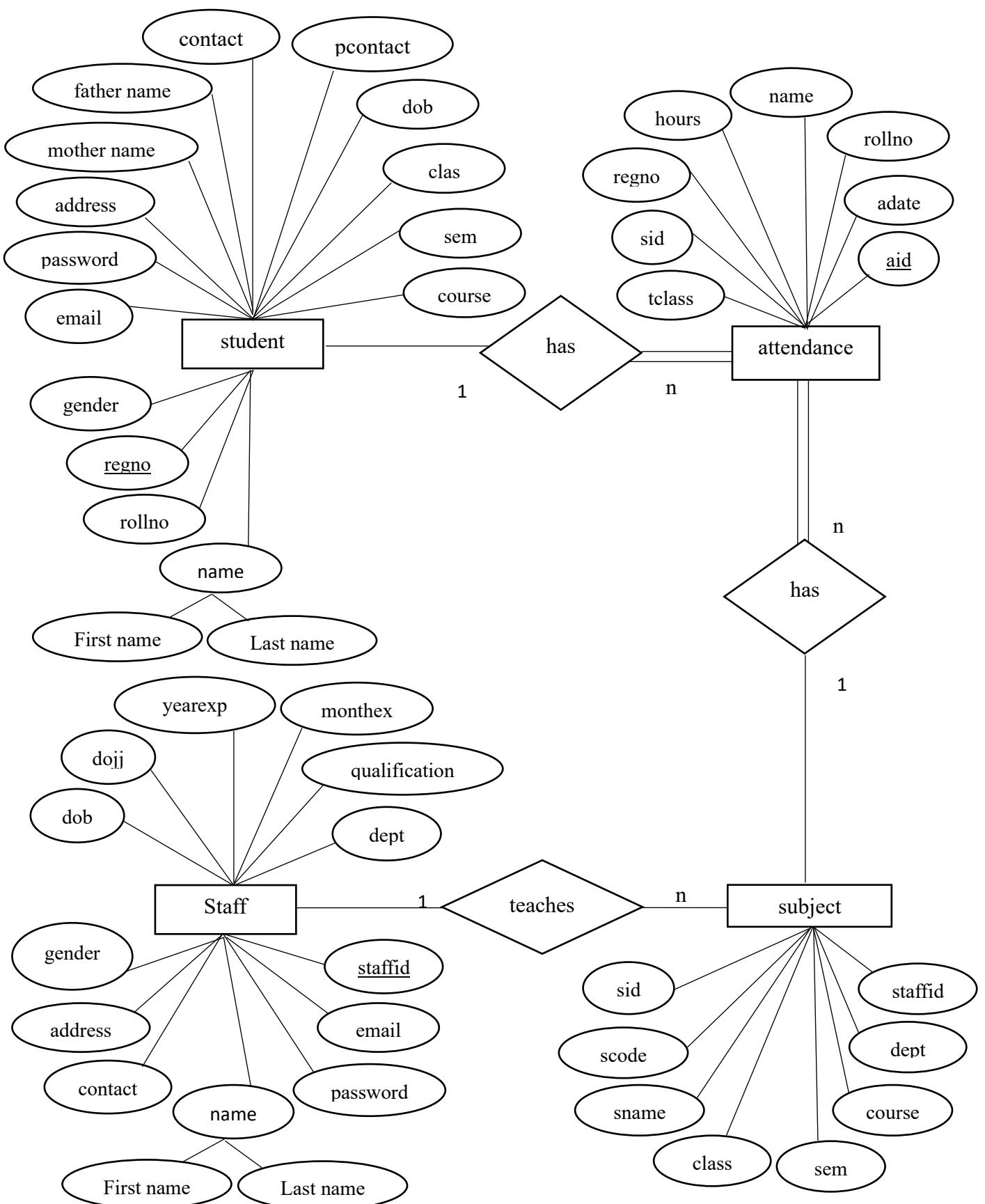


Figure 4.3 ER Diagram

4.9 Database Administration

4.9.1 DBMS System information

A database is an organized collection of structured information, or data typically stored electronically in a computer system. A data base is usually controlled by a database management system (DBMS).

(Specify the DBMS)

4.9.2. DBMS configuration

(Steps for configuration of DBMS)

Steps for configure Apache and MySQL in XAMPP

- 1.In phpMyAdmin, click the Users tab at the top.
- 2.Find the row that has User root and Host 127.0.0.1.
- 3.Click Edit Privileges.
- 4.Click Change password.
- 5.Enter the password twice (write it down somewhere if you're not sure you can remember it)
- 6.Click the Go button.

4.9.3. Support software required

MySQL Required XAMPP

(Specify any support Software Required)

Software Requirements

The following operating systems are officially supported:

- Windows 7 (64-bit, Professional level or higher)
- Mac OS X 10.6.1+
- Ubuntu 9.10 (64bit)
- Ubuntu 8.04 (32bit/64bit)

4.9.4. Hardware (Storage) requirements

(Specify the Disk place Require

The minimum hardware requirements are:

- Hard Disk: 1 GB Required 500GB(Recommended)
- CPU: Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMD CPU
- Cores: Single (Dual/Quad Core is recommended)
- RAM: 4 GB (6 GB recommended)
- Graphic Accelerators: nVidia or ATI with support of OpenGL 1.5 or higher
- Display Resolution: 1280×1024 is recommended, 1024×768 is minimum.

4.9.5. Backup and recover

Recovery is **the process of restoring a database to the correct state in the event of a failure**

Database backup is **a way to protect and restore a database**. It is performed through database replication and can be done for a database or a database server.

Using phpMyAdmin to Back Up or Restore MySQL

If you're running phpMyAdmin backing up and **restoring your MySQL database** is simple.

The **export** function is used as a backup, and the **import** function is used to restore.

Step 1: Create a MySQL Database Backup

1. Open phpMyAdmin. On the directory tree on the left, click the database you want to back up.

This should open the directory structure in the right-hand window. You'll also notice that, in the directory tree on the left, all the assets under the main database are highlighted.

2. Click **Export** on the menu across the top of the display.

You'll see a section called "Export Method." Use **Quick** to save a copy of the whole database. Choose **Custom** to select individual tables or other special options.

Leave the **Format** field set to **SQL**, unless you have a good reason to change it.

3. Click **Go**. If you select **Quick**, your web browser will download a copy of the database into your specified downloads folder. You can copy that to a safe location.

Step 2: Clear the Old Database Information

It's important to clear out old data before restoring a backup. If there's any old data, it isn't overwritten when you restore. This can create duplicate tables, causing errors and conflicts.

1. Open phpMyAdmin, on the navigation pane on the left, choose the database you want to restore.
2. Click the **check all** box near the bottom. Then, use the drop-down menu labelled **With selected** to select **Drop**.
3. The tool should prompt you to confirm that you want to go forward. Click **yes**.

This will get rid of all the existing data, clearing the way for your restoration.

Step 3: Restore Your Backed up MySQL Database

In phpMyAdmin, the **Import** tool is used to restore a database.

1. On the menu across the top, click **Import**.
2. The first section is labelled **File to import**. A couple of lines down, there's a line that starts with "Browse your computer," with a button labelled **Choose File**. Click that button.
3. Use the dialog box to navigate to the location where you've saved the export file that you want to restore. Leave all the options set to default. (If you created your backup with different options, you can select those here.)
4. Click **Go**.



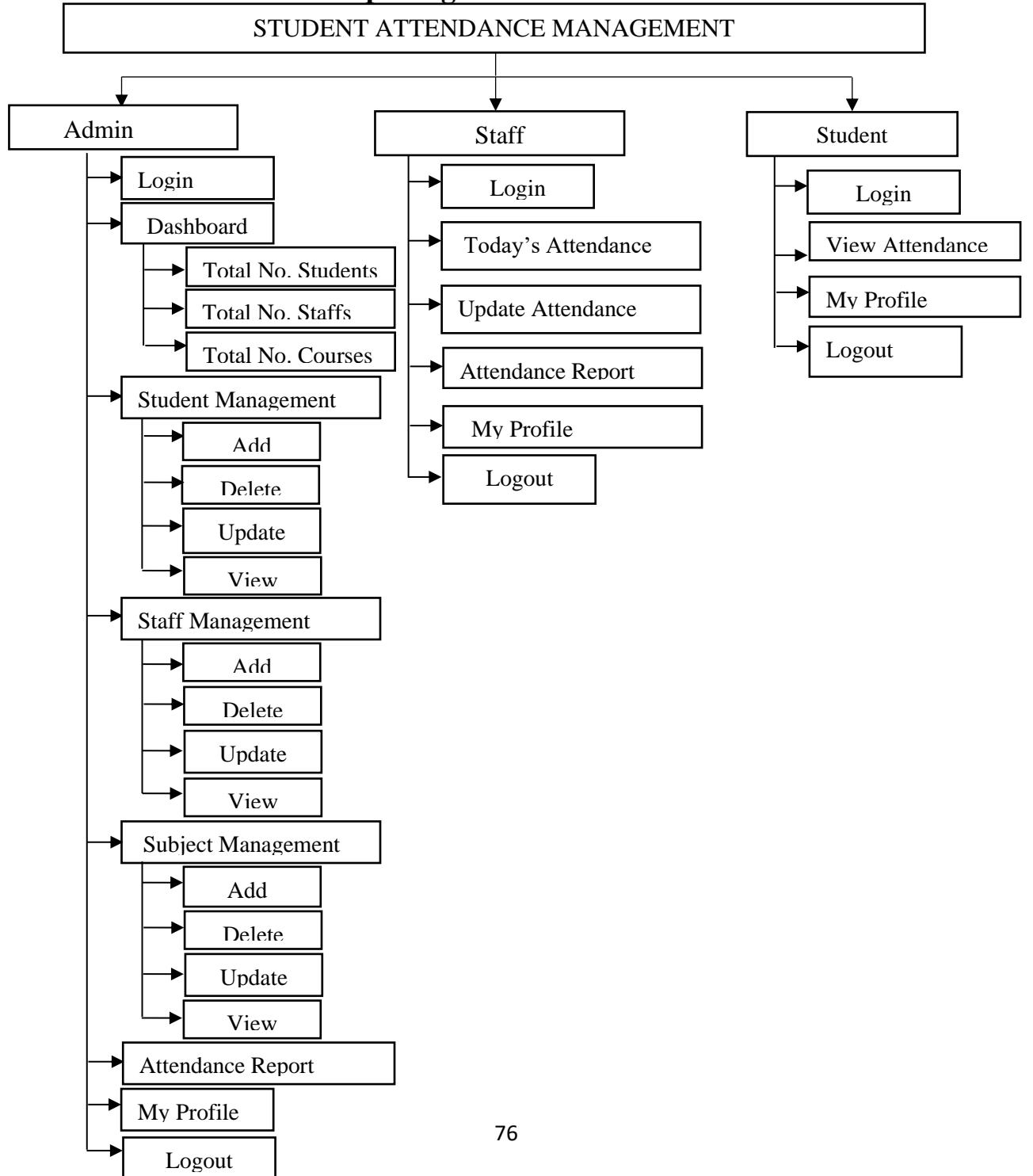
DETAILED DESIGN

5. DETAIL DESIGN

5.1 Introduction

During detailed design, the internal logic of each module specified in system design is decided. During this phase further details of the modules are decided. Design of each of the modules usually specified in a high-level description language which is independent of the language in which software eventually be implemented.

5.2 Structure of software package



5.3 Module decomposition of software

Structure chart:

Structure chart is a top-down modular design, consist of squares representing different models in a system and lines. Structure chart shows how program has been partitioned into manageable modules hierarchy and organization of those modules and communicational interface.

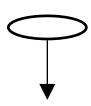
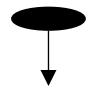
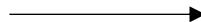
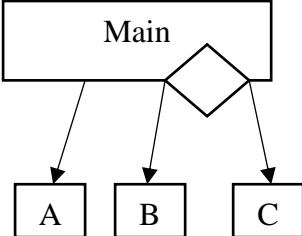
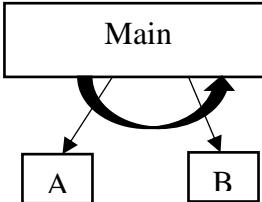
Symbol	Name	Process
	Data Flow	Show the direction flow of data.
	Control Flow	Show the direction flow of control.
	Processing	Show the manipulation, calculation and processing.
	Module Invocation	It represents subordinate module being invoked by superior ordinary module.
	Condition Invocation	It indicates that the invocation of subordinates. Module depends on the evaluation of condition.
	Iteration	It represents the iteration.

Table 5.1 Structure Chart

Flow chart:

Flow chart is a graphical representation of solution to the given problems. A Flowchart is pictorial representation of an algorithm, workflow or process. The diagrammatic representation illustrates a solution model to given problem. It uses the following symbol.

Symbol	Name	Purpose
	Terminator	It indicate the start and end process
	Input/output	Input/ Output Data
	Decision	It represents a comparison or question that determines an alternate path to be followed
	Flow Direction	Shows the direction of data flow
	Processing	It represents manipulation, calculation or information processing
	Direction access storage	File storage
	Preparation (Looping)	An instruction or Group of instruction
	In-page	
	Off-page	
	Delay	

Table 5.2 Flow Chart

5.3.1 Admin

5.3.1.1 Login

5.3.1.1.1 Input

Email Id and Password

5.3.1.1.2 Procedural Details

Structured Chart:

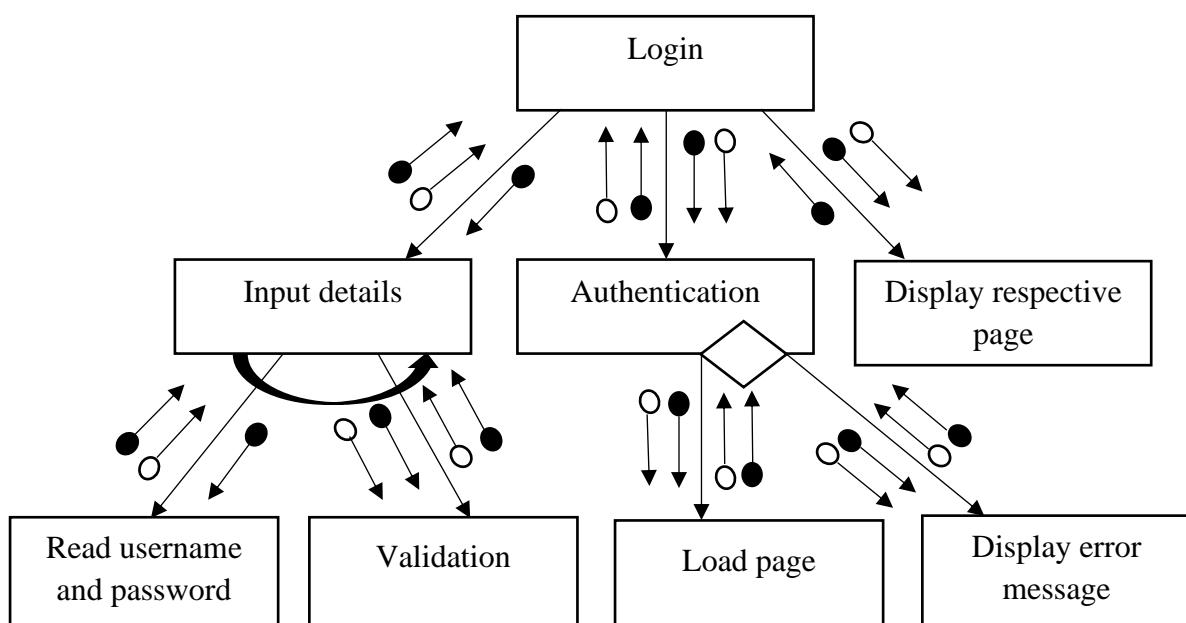


Figure 5.1 Structure Chart for Login

5.3.1.1.3 File I/O Interface

admin

5.3.1.1.4 Output

Respective Page will be loaded.

5.3.1.1.5 Implementation aspects

Edit text, button.

5.3.1.2 Dashboard

5.3.1.2.1 Total No. of Students

5.3.1.2.1.1 Input

Button click.

5.3.1.2.1.2 Procedural details

Flow Chart:

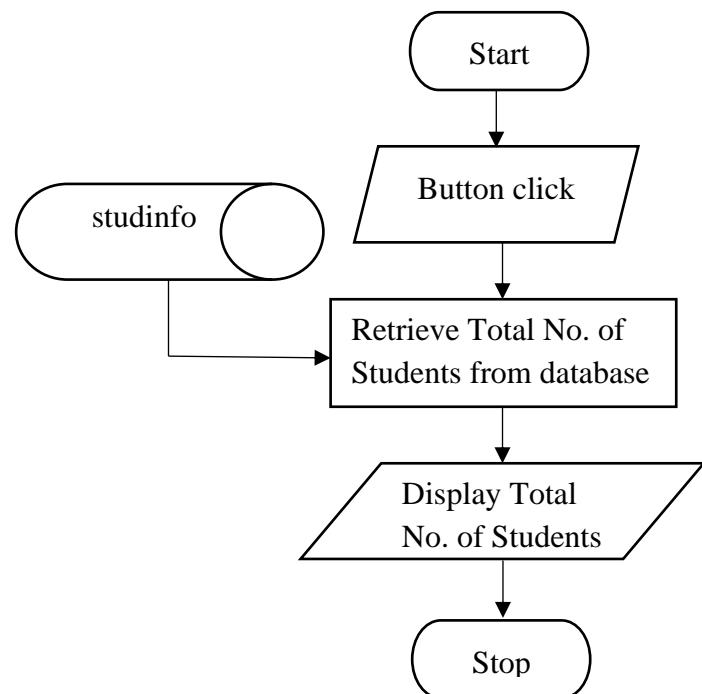


Figure 5.2 Flow Chart for Total No. of Students

5.3.1.2.1.3 File I/O Interface

studinfo

5.3.1.2.1.4 Output

Display total number of students.

5.3.1.2.1.5 Implementation aspects

Label

5.3.1.2.2 Total No. of Staffs

5.3.1.2.2.1 Input

Button click

5.3.1.2.2.2 Procedural details

Flow Chart:

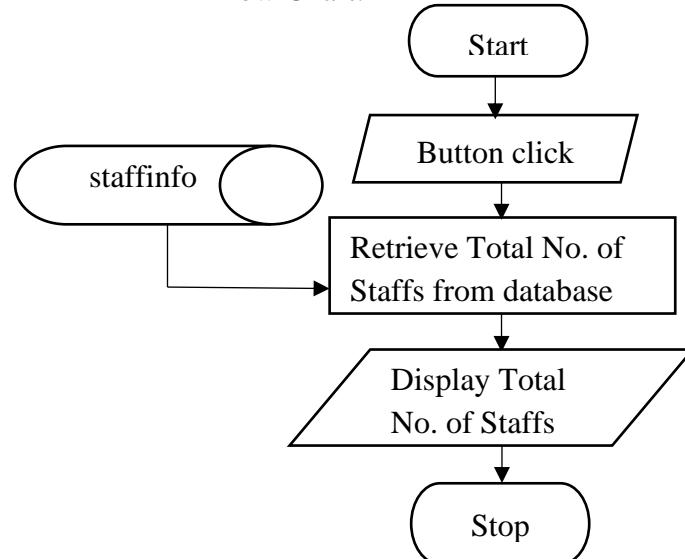


Figure 5.3 Flow Chart for Total No. of Staffs

5.3.1.2.2.3 File I/O Interface

staffinfo

5.3.1.2.2.4 Output

Display total number of staffs.

5.3.1.2.2.5 Implementation aspects

Label

5.3.1.2.3 Total No. of Courses

5.3.1.2.3.1 Input

Button click.

5.3.1.2.3.2 Procedural details

Flow Chart:

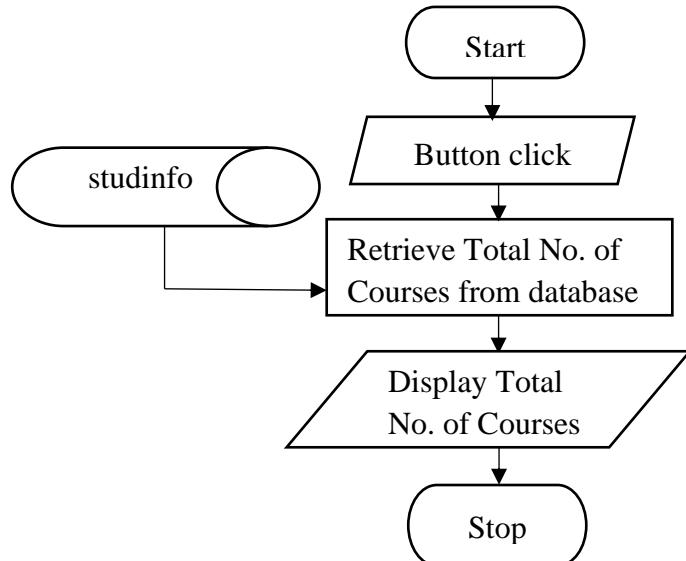


Figure 5.4 Flow Chart for Total No. of Courses

5.3.1.2.3.3 File I/O Interface

studinfo

5.3.1.2.3.4 Output

Display total number of courses.

5.3.1.2.3.5 Implementation aspects

Label

5.3.1.3 Student Management

5.3.1.3.1 Add

5.3.1.3.1.1 Input

First name, Last name, Roll No, Register No, Contact number, Parent Contact Number, Father Name, Mother Name, Address, Gender, DOB, Email-ID, Password, Year, Semester, Course.

5.3.1.3.1.2 Procedural details

Structured Chart:

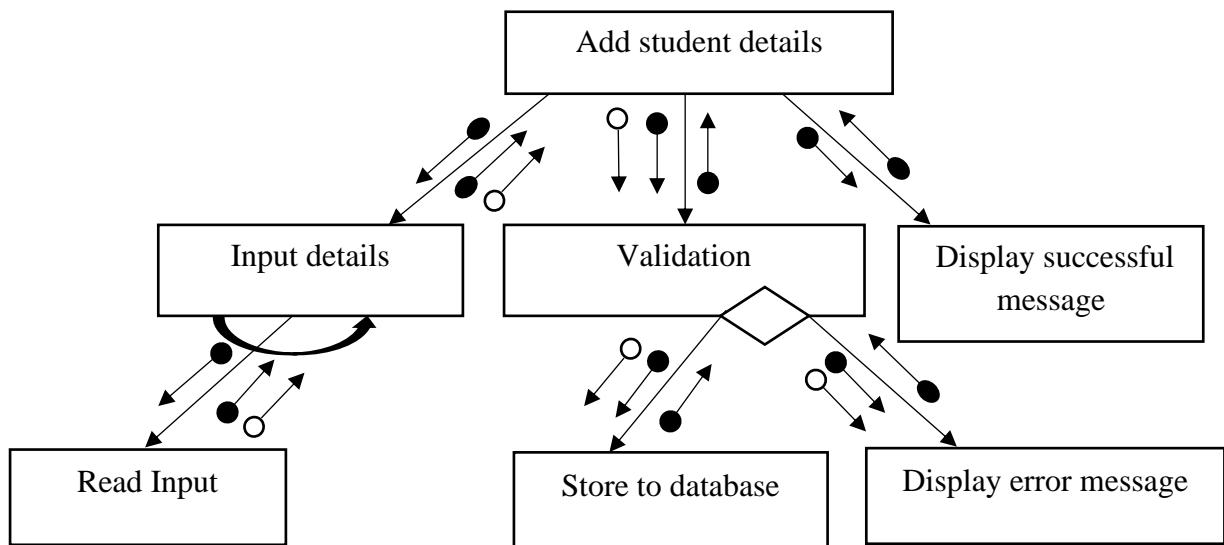


Figure 5.5 Structure Chart for Add

5.3.1.3.1.3 File I/O Interface

studinfo

5.3.1.3.1.4 Output

If added data is valid then store to database and displays successful message. Otherwise show error message.

5.3.1.3.1.5 Implementation aspects

Button, Textbox, Dropdown

5.3.1.3.2 Update

5.3.1.3.2.1 Input

Register Number

5.3.1.3.2.2 Procedural details

Algorithm:

Step 1: START

Step 2: INPUT Register Number.

Step 3: Retrieve student details from database.

Step 4: Perform update operation.

Step 5: IF updated detailed is valid then

 Store updated information to database.

 Display updated successful message.

ELSE

 Display Error message.

 GOTO Step2

END IF

Step 6: END

5.3.1.3.2.3 File I/O Interface

studinfo

5.3.1.3.2.4 Output

If updated data is valid then store to database and displays updated successful message. Otherwise show error message.

5.3.1.3.2.5 Implementation aspects

Button, Textbox, Dropdown.

5.3.1.3.3 Delete

5.3.1.3.3.1 Input

Register Number

5.3.1.3.3.2 Procedural details

Flow Chart:

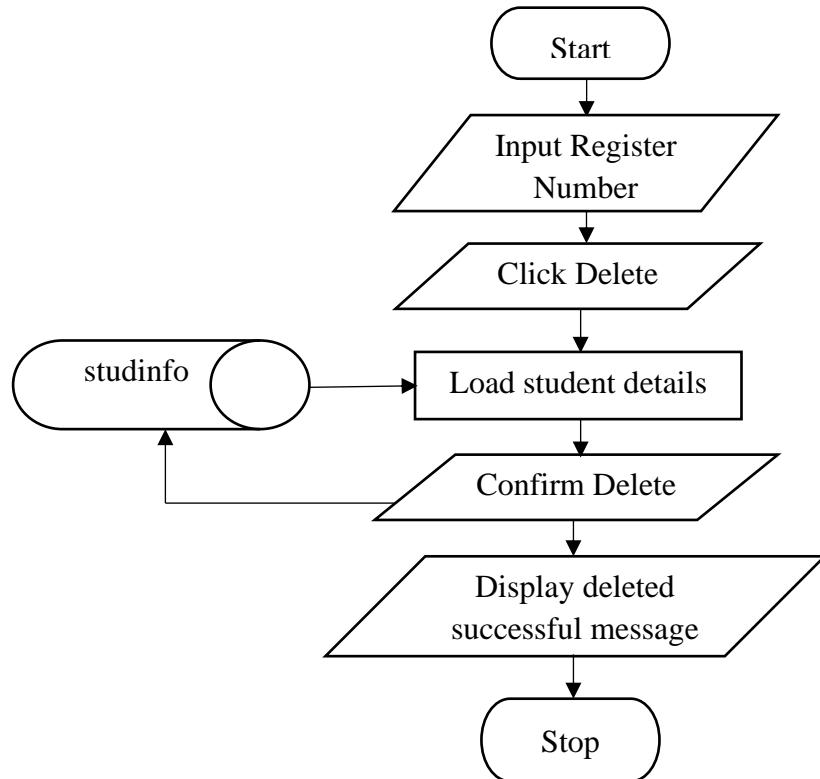


Figure 5.6 Flow Chart for Delete

5.3.1.3.3.3 File I/O Interface

studinfo

5.3.1.3.3.4 Output

Student details will be deleted from database and displays deleted successfully message.

5.3.1.2.3.5 Implementation aspects

Button, Textbox.

5.3.1.3.4 View

5.3.1.3.4.1 Input

Button click

5.3.1.3.4.2 Procedural details

Flow Chart:

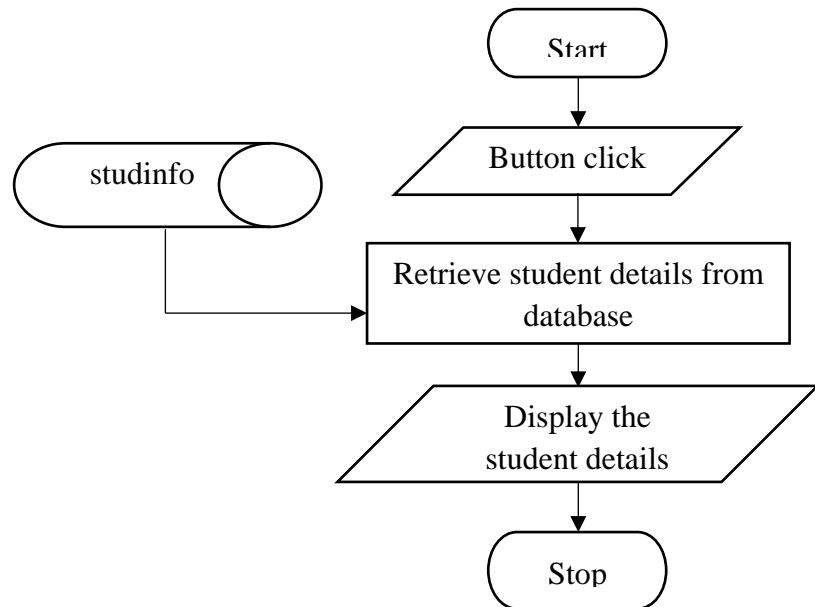


Figure 5.7 Flow Chart for View

5.3.1.3.4.3 File I/O Interface

studinfo

5.3.1.3.4.4 Output

Retrieve details from database.

5.3.1.3.4.5 Implementation aspects

Button

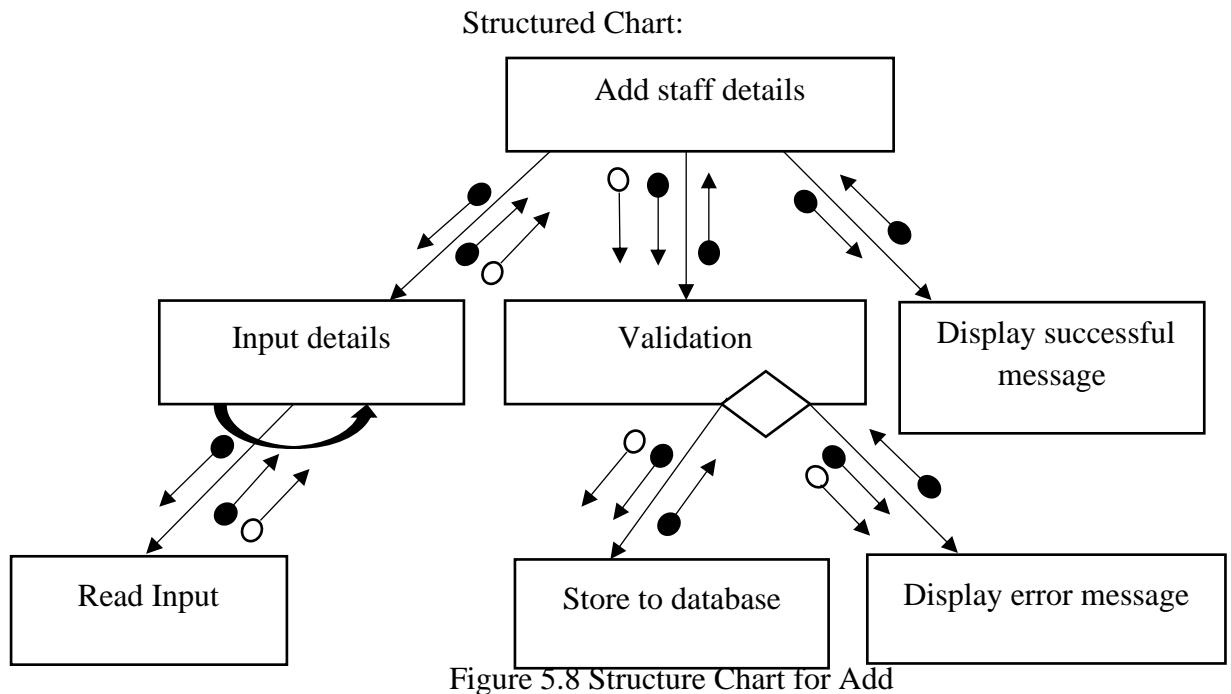
5.3.1.4 Staff Management

5.3.1.4.1 Add

5.3.1.4.1.1 Inputs

First name, Last name, Staff ID, Contact number, Address, Gender, DOB, DOJ, Experience, Qualification, Department.

5.3.1.4.1.2 Procedural details



5.3.1.4.1.3 File I/O Interface

staffinfo

5.3.1.4.1.4 Output

If added data is valid then store to database and displays successful message. Otherwise show error message.

5.3.1.4.1.5 Implementation aspects

Button, Textbox, Dropdown

5.3.1.4.2 Update

5.3.1.4.2.1 Input

Staff Id

5.3.1.4.2.2 Procedural details

Algorithm:

Step 1: START

Step 2: INPUT Staff Id

Step 3: IF present:

 Retrieve staff details from database.

ELSE

 Display Record not found

Step 4: Perform update operation.

Step 5: IF updated detailed is valid then

 Store updated information to database.

 Display updated successful message.

ELSE

 Display Error message.

 GOTO Step2

END IF

Step 6: END

5.3.1.4.2.3 File I/O Interface

staffinfo

5.3.1.4.2.4 Output

If updated data is valid then store to database and displays updated successful message. Otherwise show error message.

5.3.1.4.2.5 Implementation aspects

Button, Textbox, Dropdown.

5.3.1.4.3 Delete

5.3.1.4.3.1 Input

Staff Id

5.3.1.4.3.2 Procedural details

Flow Chart:

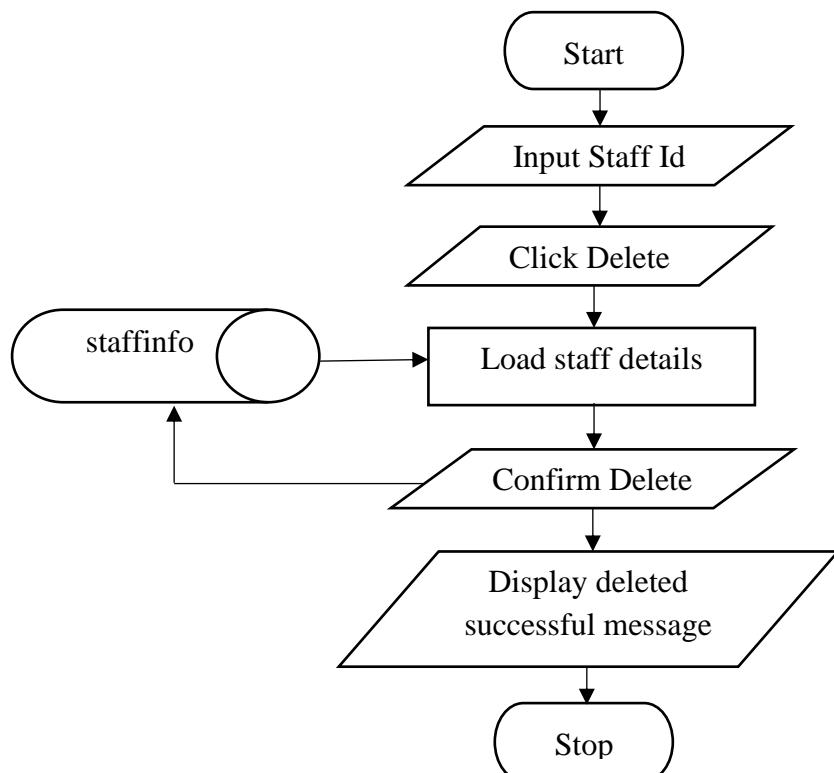


Figure 5.9 Flow Chart for Delete

5.3.1.4.3.3 File I/O Interface

staffinfo

5.3.1.4.3.4 Output

Staff details will be deleted from database and displays deleted successfully message.

5.3.1.4.3.5 Implementation aspects

Button, Editbox.

5.3.1.4.4 View

5.3.1.4.4.1 Input

Button click

5.3.1.4.4.2 Procedural details

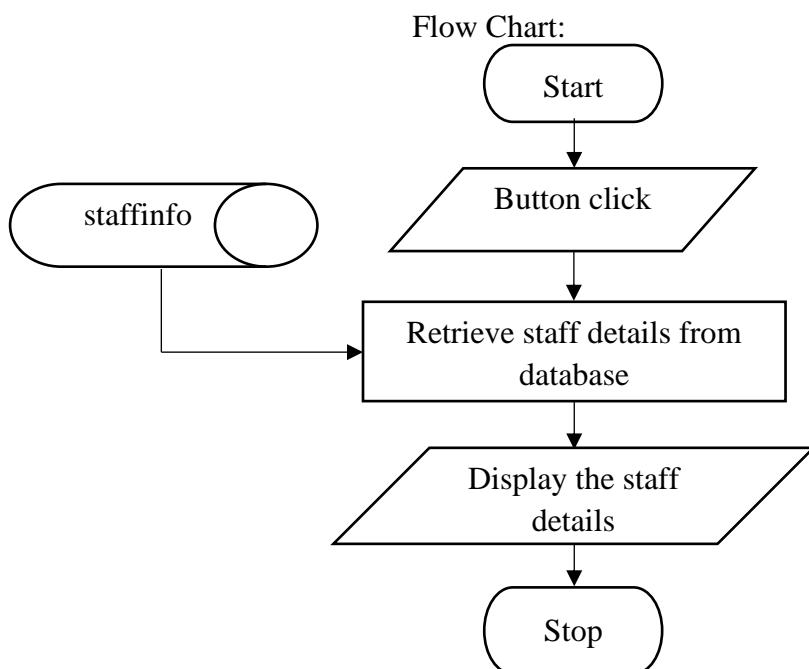


Figure 5.10 Flow Chart for View

5.3.1.4.4.3 File I/O Interface

staffinfo

5.3.1.4.4.4 Output

Retrieve details from database.

5.3.1.4.4.5 Implementation aspects

Button

5.3.1.5 Subject Management

5.3.1.5.1 Add

5.3.1.5.1.1 Input

Course, Year, Semester, Subject name, Subject Code,
Department, Staff Name, Staff-ID.

5.3.1.5.1.2 Procedural details

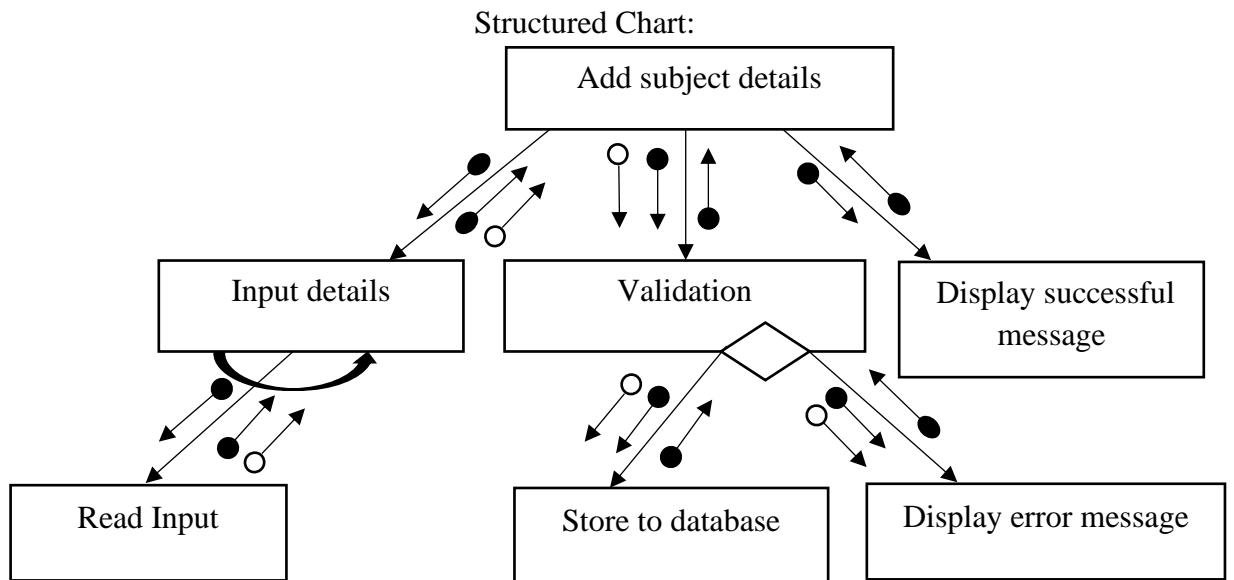


Figure 5.11 Structure Chart for Add

5.3.1.5.1.3 File I/O Interface

subinfo

5.3.1.5.1.4 Output

If added data is valid then store to database and displays successful message. Otherwise show error message.

5.3.1.5.1.5 Implementation aspects

Button, Editbox, Dropdown

5.3.1.5.2 Update

5.3.1.5.2.1 Input

Subject Code

5.3.1.5.2.2 Procedural details

Algorithm:

Step 1: START

Step 2: INPUT Subject Code.

Step 3: Retrieve subject details from database.

Step 4: Perform update operation.

Step 5: IF updated detailed is valid then

 Store updated information to database.

 Display updated successful message.

ELSE

 Display Error message.

 GOTO Step2

END IF

Step 6: END

5.3.1.5.2.3 File I/O Interface

subinfo

5.3.1.5.2.4 Output

If updated data is valid then store to database and displays successful message. Otherwise show error message.

5.3.1.4.2.5 Implementation aspects

Button, Textbox, Dropdown.

5.3.1.5.3 Delete

5.3.1.5.3.1 Input

Subject Code

5.3.1.5.3.2 Procedural details

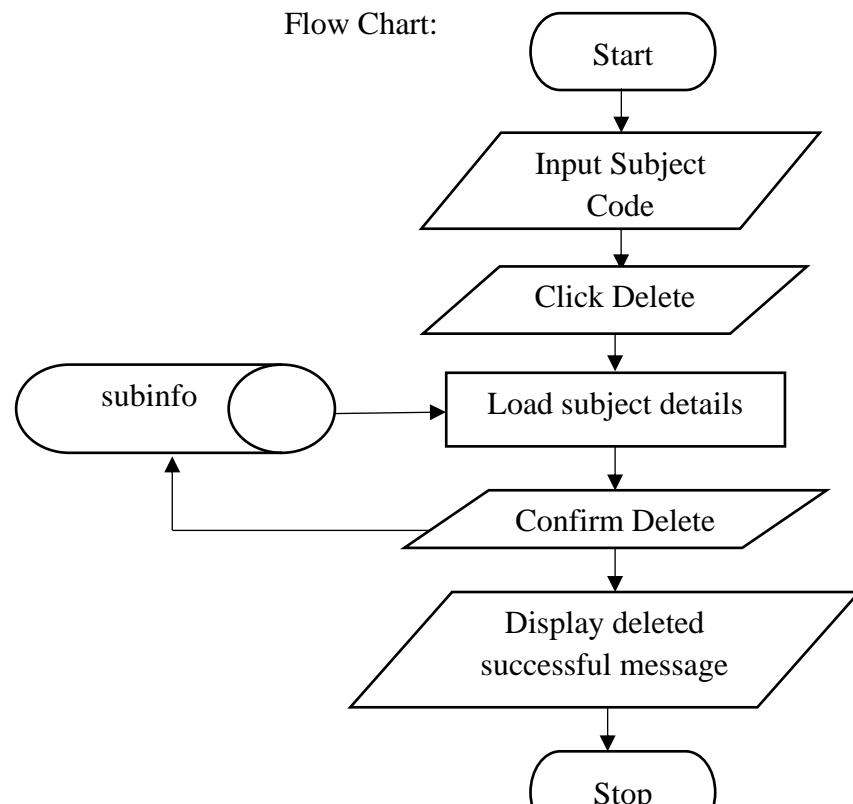


Figure 5.12 Flow Chart for Delete

5.3.1.5.3.3 File I/O Interface

subinfo

5.3.1.2.5.3.4 Output

Subject details will be deleted from database and displays deleted successfully message.

5.3.1.5.3.5 Implementation aspects

Button, Textbox.

5.3.1.5.4 View

5.3.1.5.4.1 Input

Button click

5.3.1.4.4.2 Procedural details

Flow Chart:

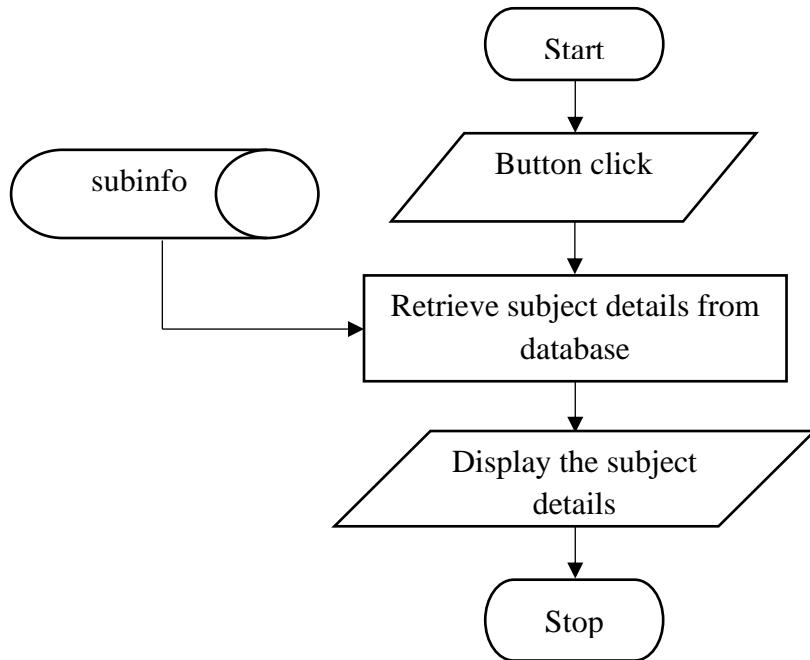


Figure 5.13 Flow Chart for View

5.3.1.4.4.3 File I/O Interface

subinfo

5.3.1.4.4.4 Output

Retrieve details from database

5.3.1.4.4.5 Implementation aspects

Button.

5.3.1.6 Attendance Report

5.3.1.6.1 Input

Year, Semester, Class, Subject Name.

5.3.1.6.2 Procedural Details

Algorithm:

Step1: START

Step2: INPUT Year, Semester, Course, Subject
Name, starting date, ending date.

Step3: Retrieve attendance details from the
database.

Step4: Display Attendance Report

Step5: END

5.3.1.6.3 File I/O Interface

attendance

5.3.1.6.4 Output

Respective Page will be loaded.

5.3.1.6.5 Implementation aspects

Button, Dropdown.

5.3.1.7 My Profile

5.3.1.7.1 Input

Button click

5.3.1.7.2 Procedural Details

Algorithm:

Step1: START

Step2: INPUT button click.

Step3: Retrieve information from database.

Step4: Display information.

Step5: END

5.3.1.7.3 File I/O Interface

admin

5.3.1.7.4 Output

Displays detail from database.

5.3.1.7.5 Implementation aspects

Edit text, button

5.3.1.8 Logout

5.3.1.8.1 Input

Button click

5.3.1.8.2 Procedural Details

Flow Chart:

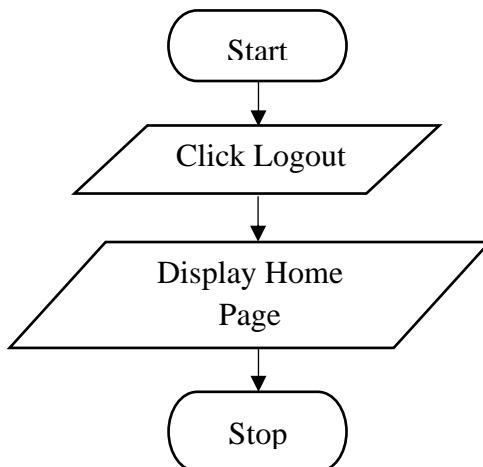


Figure 5.14 Flow Chart for Logout

5.3.1.8.3 File I/O Interface

None

5.3.1.8.4 Output

Display Home Page.

5.3.1.8.5 Implementation aspects

Button

5.3.2 Staff

5.3.2.1 Login

5.3.2.1.1 Input

Email Id and Password

5.3.2.1.2 Procedural Details

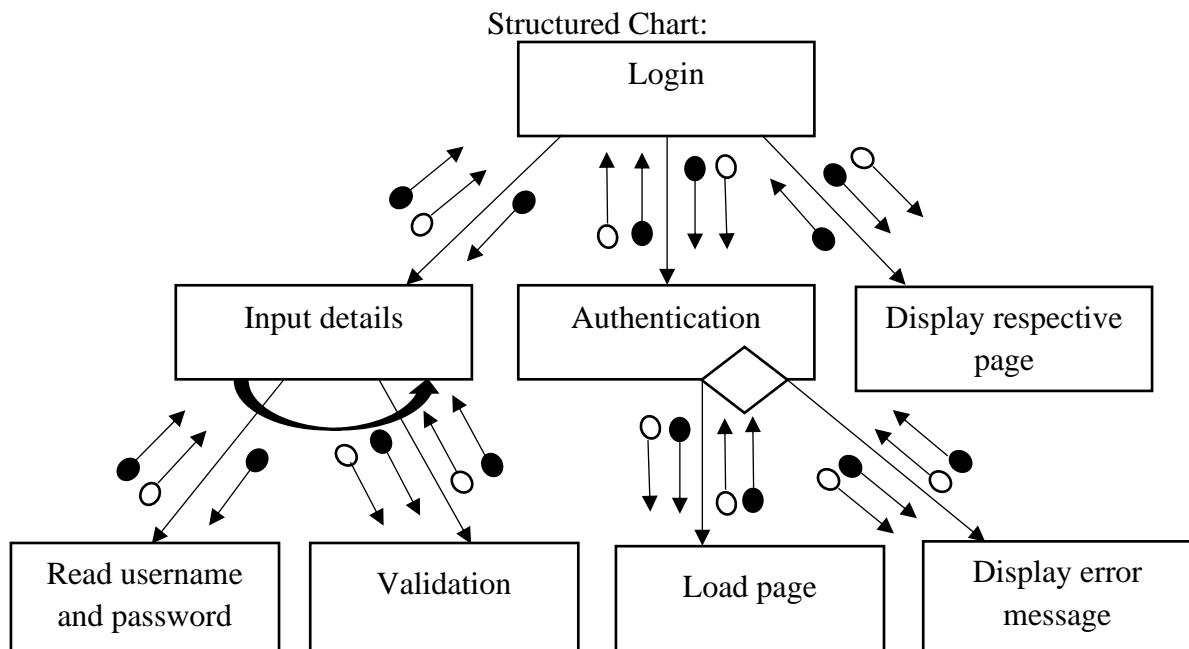


Figure 5.15 Structure Chart for Login

5.3.2.1.3 File I/O Interface

staffinfo

5.3.2.1.4 Output

Respective Page will be loaded.

5.3.2.1.5 Implementation aspects

Edit text, button.

5.3.2.2 Today's Attendance

5.3.2.2.1 Input

Year, Semester, Course, Hours, Subject Name, hour.

5.3.2.2.2 Procedural Details

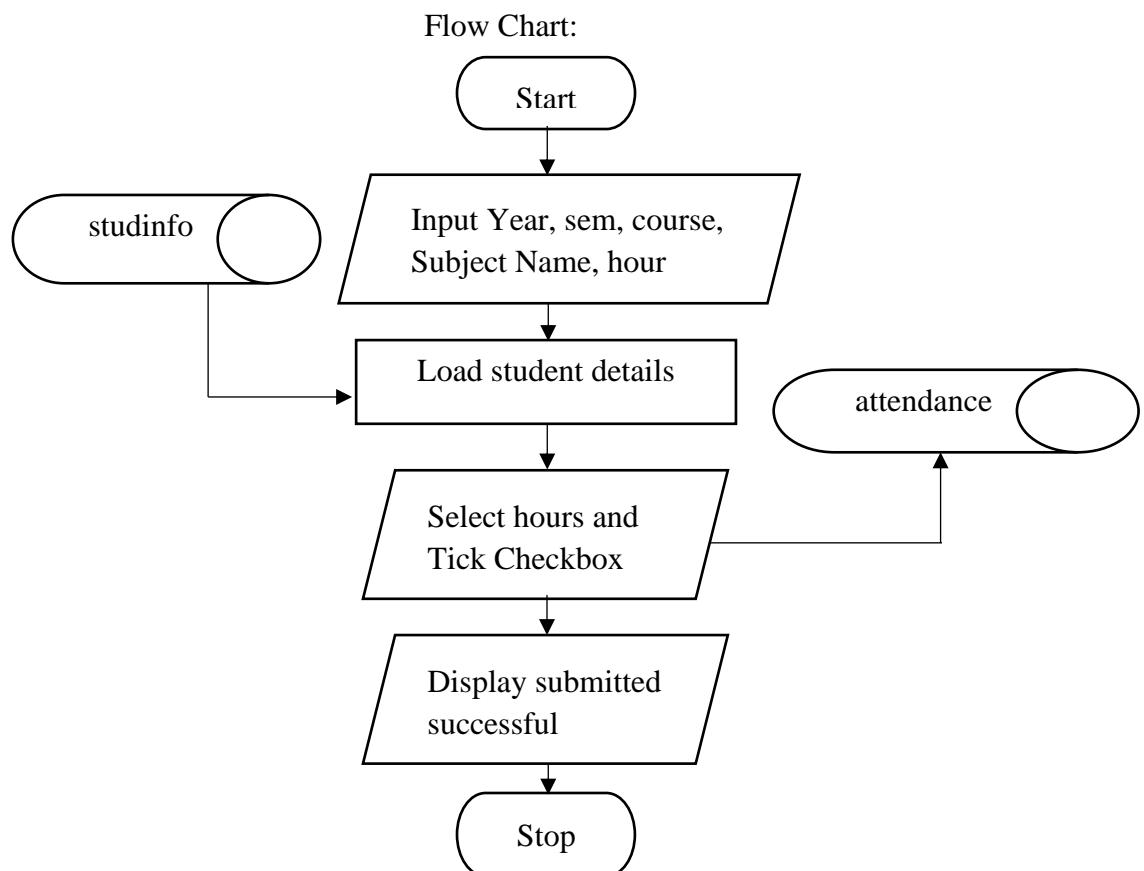


Figure 5.16 Flow Chart for Today's Attendance

5.3.2.2.3 File I/O Interface

attendance

5.3.2.2.4 Output

Marks the attendance to the database

5.3.2.2.5 Implementation aspects

Button, Dropdown, Checkbox.

5.3.2.3 Update Attendance

5.3.2.3.1 Input

Year, Semester, Course, Date, Hours

5.3.2.3.2 Procedural Details

Algorithm:

Step 1: START

Step 2: INPUT Year, Semester, Course, Subject Name, date,
hours.

Step 3: IF date is present then

 Retrieve student attendance from database and

 Perform update operation and display updated
 successful message.

ELSE

 Mark attendance and store it in the database
 and display submitted successful message

END IF

Step 4: END

5.3.2.3.3 File I/O Interface

attendance

5.3.2.3.4 Output

Marks the attendance to the database.

5.3.2.3.5 Implementation aspects

Button, Dropdown, Checkbox.

5.3.2.4 Attendance Report

5.3.2.4.1 Input

Year, Semester, Course, Subject Name, starting date, ending date.

5.3.2.4.2 Procedural Details

Algorithm:

Step1: START

Step2: INPUT Year, Semester, Course, Subject Name,
starting date, ending date

Step3: Retrieve attendance details from the database.

Step4: Display Attendance Report.

Step5: END

5.3.2.4.3 File I/O Interface

attendance

5.3.2.4.4 Output

Displays attendance report

5.3.2.4.5 Implementation aspects

Button, Dropdown.

5.3.2.5 My Profile

5.3.2.5.1 Input

Button click

5.3.2.5.2 Procedural Details

Algorithm:

Step1: START

Step2: INPUT button click.

Step3: Retrieve information from database.

Step4: Display information.

Step5: END

5.3.2.5.3 File I/O Interface

staffinfo

5.3.2.5.4 Output

Displays detail from database

5.3.2.5.5 Implementation aspects

Button

5.3.2.6 Logout

5.3.2.6.1 Input

Button click

5.3.2.6.2 Procedural Details

Flow Chart:

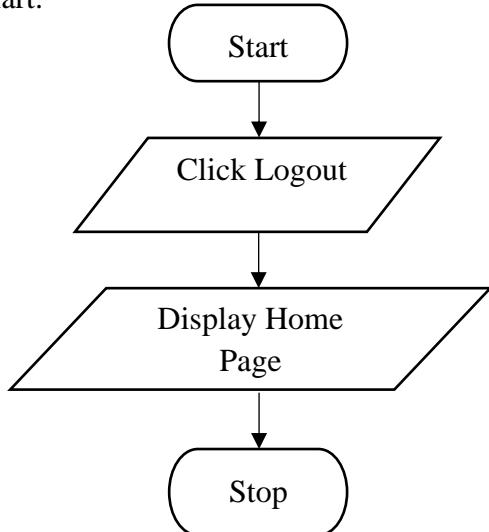


Figure 5.17 Flow Chart for Logout

5.3.2.6.3 File I/O Interface

None

5.3.2.6.4 Output

Display Home Page.

5.3.2.6.5 Implementation aspects

Button

5.3.3 Student

5.3.3.1 Login

5.3.3.1.1 Input

Email Id and Password

5.3.3.1.2 Procedural Details

Structured Chart:

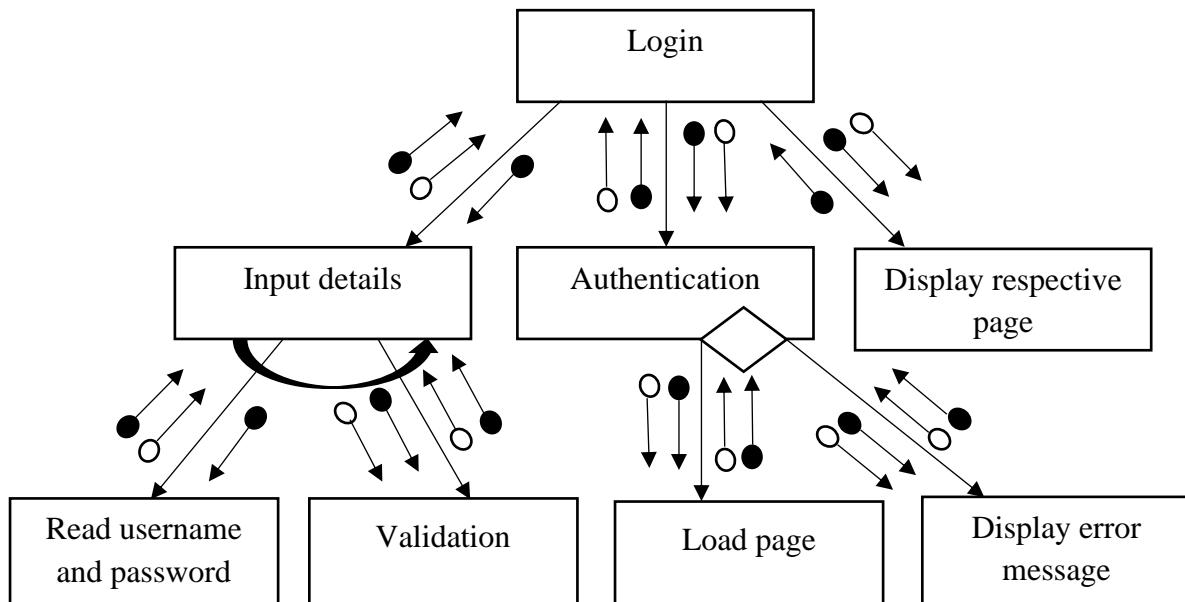


Figure 5.18 Structure Chart Login

5.3.3.1.3 File I/O Interface

studinfo

5.3.3.1.4 Output

Respective Page will be loaded.

5.3.3.1.5 Implementation aspects

Edit text, button.

5.3.3.2 View Attendance

5.3.3.2.1 Input

Subject Name, starting date, ending date

5.3.3.2.2 Procedural details

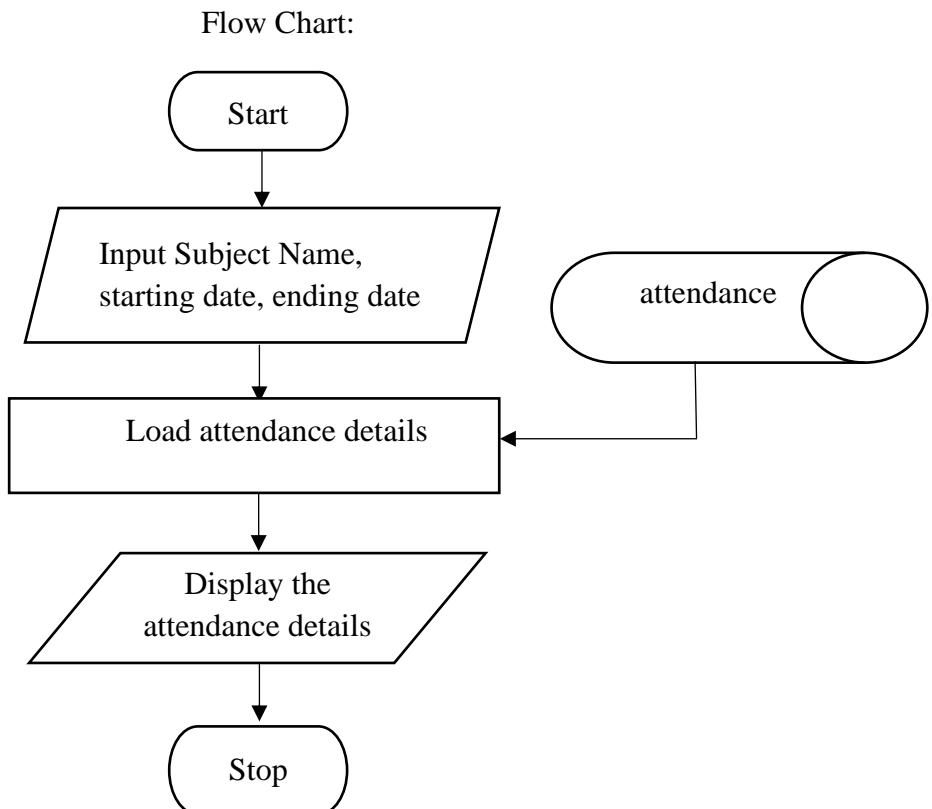


Figure 5.19 Flow Chart for View Attendance

5.3.3.2.3 File I/O Interface

attendance

5.3.3.2.4 Output

Display details from database

5.3.3.2.5 Implementation aspects

Button

5.3.3.3 My Profile

5.3.3.3.1 Inputs

Button click

5.3.3.3.2 Procedural Details

Algorithm:

Step1: START

Step2: INPUT Button click

Step3: Retrieve information from database.

Step4: Display information.

Step5: END

5.3.3.3.3 File I/O Interface

studinfo

5.3.3.3.4 Output

Display details from the database.

5.3.3.3.5 Implementation aspects

Button.

5.3.3.4 Logout

5.3.3.4.1 Input

Button click.

5.3.3.4.2 Procedural Details

Flow Chart:

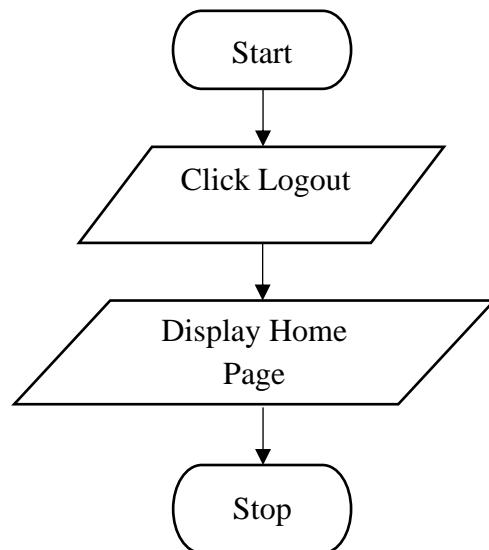


Figure 5.20 Flow Chart for Logout

5.3.3.4.3 File I/O Interface

None

5.3.3.4.4 Output

Display Home Page.

5.3.3.4.5 Implementation aspects

Button



PROGRAM CODE LISTING

6. CODING

home.html:

```
<!DOCTYPE html>
<html>
<head><title>Home</title></head>
<body>
<div class="container">
<nav id="navigationbar">

<span>Bhandakars' Arts and Science College</span>
<ul><li><a href="login.php">Admin</a></li>
<li><a href="login.php">Staff</a></li>
<li><a href="login.php">Student</a></li></ul>
</nav>
<section>
<h1>WELCOME TO</h1>
<h2>Student Attendance Management System</h2>
</section>
</div>
</body>
</html>
```

login.php:

```
<?php session_start();?>
<!DOCTYPE html>
<html>
<head><title>Login</title></head>
<body>
<div class="container">
<nav id="navigationbar">

<ul><li><span>Bhandakars' Arts and Science College</span></li>
```

```

<li><a href="home.html">Home</a></li> </ul>
</nav>

<div class="leaf">
    <div class="middle"> <!--login-box-->
        <h2><b>LOGIN</b></h2>
        <form action="login.php" method="POST">
            <div class="input-box">
                <input type="text" id="email" name="email" class="form__input"
maxlength="30" required>
                <label for="username" class="form__label">Email</label>
            </div>
            <div class="input-box">
                <input type="password" id="password" name="password"
maxlength="15" required>
                <label for="password">Password</label>
            </div>
            <div class="forgotpass">
                <a class="link1" id="link1" href="changepassphp.php">Forgot
password?</a>
            </div>
            <button type="submit" class="btn" value="login"
name="login">Login</button><br>
        </form>
    </div>
    </div>
</body>
</html>
<?php
$_SESSION['user'] = "";
if (isset($_POST['login'])) {
    $em = $_POST['email'];
    $pass = $_POST['password'];

```

```

@$con = new mysqli('localhost', 'root', '', 'attendance');

$qry = "SELECT * FROM admin WHERE email='$em'";
$result = mysqli_query($con, $qry);
if (mysqli_num_rows($result)) {
    $row = mysqli_fetch_assoc($result);
    if ($pass == $row['password']) {
        $_SESSION['user'] = $em;
        echo "<script>window.location.href='admin.php';</script>";
        exit;
    }
    else {
        echo "<script>alert('Wrong Password');</script>";
        echo "<script>window.location.href='login.php';</script>";
    }
}

$qry = "SELECT * FROM studinfo WHERE email='$em'";
$result = mysqli_query($con, $qry);
if (mysqli_num_rows($result)) {
    $row = mysqli_fetch_assoc($result);
    if ($pass == $row['password']) {
        $_SESSION['user'] = $em;
        echo "<script>window.location.href='student/viewattendance.php';</script>";
        exit;
    }
    else {
        echo "<script>alert('Wrong Password');</script>";
        echo "<script>window.location.href='login.php';</script>";
    }
}

$qry = "SELECT * FROM staffinfo WHERE email='$em'";
$result = mysqli_query($con, $qry);
if (mysqli_num_rows($result)) {
    $row = mysqli_fetch_assoc($result);
}

```

```

if ($pass == $row['password']) {
    $_SESSION['user'] = $em;
    echo "<script>window.location.href='staff/todaysattendance.php';</script>";
    exit;
} else {
    echo "<script>alert('Wrong Password');</script>";
    echo "<script>window.location.href='login.php';</script>";
}
$qry = "SELECT * FROM admin WHERE password='$pass'";
$result = mysqli_query($con, $qry);
if (mysqli_num_rows($result)) {
    $row = mysqli_fetch_assoc($result);
    if (!$em == $row['email']) {
        echo "<script>alert('Wrong Email-ID');</script>";
        echo "<script>window.location.href='login.php';</script>";
    }
}
$qry = "SELECT * FROM staffinfo WHERE password='$pass'";
$result = mysqli_query($con, $qry);
if (mysqli_num_rows($result)) {
    $row = mysqli_fetch_assoc($result);
    if (!$em == $row['email']) {
        echo "<script>alert('Wrong Email-ID');</script>";
        echo "<script>window.location.href='login.php';</script>";
    }
}
$qry = "SELECT * FROM studinfo WHERE password='$pass'";
$result = mysqli_query($con, $qry);
if (mysqli_num_rows($result)) {
    $row = mysqli_fetch_assoc($result);
    if (!$em == $row['email']) {

```

```

echo "<script>alert('Wrong Email-ID');</script>";
echo "<script>window.location.href='login.php';</script>";
}
}

echo "<script>alert('You are not an authenticated user');</script>";
echo "<script>window.location.href='login.php';</script>";
}

?>

```

studregisterphp.php:

```

<?php
if (isset($_POST['register'])) {
    $fn=$_POST['firstname'];
    $ln=$_POST['lastname'];
    $rn=$_POST['rollno'];
    $reg=$_POST['regno'];
    $fname=$_POST['fathername'];
    $mname=$_POST['mothername'];
    $cont=$_POST['contact'];
    $pcont=$_POST['pcontact'];
    $em=$_POST['email'];
    $pass=$_POST['password'];
    $add=$_POST['address'];
    $gen=$_POST['gender'];
    $dob=$_POST['dob'];
    $cls=$_POST['clas'];
    $sem=$_POST['sem'];
    $cours=$_POST['course'];
    if($ln=="") {$ln="N/A";}
    if($pcont=="") { $pcont="0";}

    @$con=new mysqli('localhost','root','','attendance');

    if(mysqli_connect_errno()){
        echo"could not connect";
    }
}
```

```

}

else{
    $existqry="SELECT * FROM studinfo where regno='$reg'";
    $rslt=mysqli_query($con,$existqry);
    if($rslt){
        $num=mysqli_num_rows($rslt);
        if($num>0){
            echo"<script>alert('This registration number already existed');</script>";
            echo"<script>window.location.href='studmanagementphp.php';</script>";
        }
    }

$existqry="select * from studinfo where email='$em'";
$rslt=mysqli_query($con,$existqry);
if($rslt){
    $num=mysqli_num_rows($rslt);
    if($num>0) {
        echo"<script>alert('This Email-Id already existed ');</script>";
        echo"<script>window.location.href='studmanagementphp.php';</script>";
    }else{
        $qry="INSERT INTO studinfo
VALUES('".$fn."','".$ln."','".$rn."','".$reg."','".$fname."','".$mname."','".$cont."','".$_pcont."','".$em."','".$pass."','".$add."','".$gen."','".$dob."','".$cls."','".$sem."','".$cours."');

$result=mysqli_query($con,$qry);
if($result){
            echo"<script>alert('Registerd successfully');</script>";
            echo"<script>window.location.href='studmanagementphp.php';
            </script>";
        } else{
            echo"<script>alert('Registration failed!');</script>";
            echo"<script>window.location.href='studmanagementphp.php';
            </script>";
        }
    }
}

```

```

        }
    }
}
}

?>

```

tattendance.php:

```

<!DOCTYPE html>
<head><title>Today attendance table</title></head>
<script>

function toggleCheckboxes() {

    var checkboxes = document.querySelectorAll('input[type="checkbox"]');

    var selectAllCheckbox = document.getElementById('select-all');

    for (var i = 0; i < checkboxes.length; i++) {

        checkboxes[i].checked = selectAllCheckbox.checked;
    }
}

</script><body></body>
</html>
<?php

$course = $_GET['course'];
$class = $_GET['class'];
$sub = $_GET['sub'];
$sem = $_GET['sem'];
$sid = $_GET['sid'];
$conn = mysqli_connect('localhost', 'root', "", 'attendance');session_start();
$email = $_SESSION['user'];

$query = "SELECT * FROM studinfo WHERE course = '$course' and clas = '$class' and sem = '$sem'";
$result = mysqli_query($conn, $query);
$currentTime = date('d-m-Y');
$cDate = date('Y-m-d');
$query1 = "SELECT * FROM attendance WHERE adate = '$cDate' and sid = '$sid'";


```

```

$result1 = mysqli_query($conn, $query1);
if ($result1->num_rows > 0) {
    $issued = 'yes';
    echo "<p>Attendance has already been issued on this date, submitting will add additional attendance</p>";
} else {
    $issued = 'no';
}

echo '<div class="tabletop"><label>Date: ' . $currentDate . '</label>';
echo '<form action="submit.php?course=' . $course . '&class=' . $class . '&sem=' . $sem .
'&sub=' . $sub . '&sid=' . $sid . '&issued=' . $issued . '" method="POST">
<label>Select Hours:</label>
<select name="hours">
    <option value="1">1 hour</option>
    <option value="2">2 hours</option>
    <option value="3">3 hours</option>
    <option value="4">4 hours</option>
</select></div>
<div class="table_selection">
<table> <thead> <tr><td>Roll number</td>
    <td>Name</td>
    <td><input type="checkbox" id="select-all" onchange="toggleCheckboxes()"> All
        Present</td></tr> </thead>';
while ($row = mysqli_fetch_assoc($result)) {
    echo ' <tr id="tablerow">
        <td>' . $row["rollno"] . '</td>
        <td>' . $row["firstname"] . ' ' . $row["lastname"] . '</td>
        <td><input type="checkbox" name="' . $row['rollno'] . '" value="present"></td>
    </tr>';
}
echo '</table>
<div class="btns">
    <button type="submit" id="submitAttendance" name="submitAttendance">Submit
        Attendance</button>
    <button type="submit" id="backbtn" name="backbtn" value="Back">Back</button>

```

```
</div></div></form>';
```

```
?>
```

submit.php:

```
<!DOCTYPE html>

<head><title>Today attendance table</title></head>

<script>

    function toggleCheckboxes() {

        var checkboxes = document.querySelectorAll('input[type="checkbox"]');

        var selectAllCheckbox = document.getElementById('select-all');

        for (var i = 0; i < checkboxes.length; i++) {

            checkboxes[i].checked = selectAllCheckbox.checked;
        }
    }

</script>

<body></body></html>

<?php

$course = $_GET['course'];

$class = $_GET['class'];

$sub = $_GET['sub'];

$sem = $_GET['sem'];

$sid = $_GET['sid'];

$conn = mysqli_connect('localhost', 'root', '', 'attendance');

session_start();

$email = $_SESSION['user'];

$query = "SELECT * FROM studinfo WHERE course = '$course' and clas = '$class' and sem = '$sem'";

$result = mysqli_query($conn, $query);

$currentTime = date('d-m-Y');

$cDate = date('Y-m-d');

$query1 = "SELECT * FROM attendance WHERE adate = '$cDate' and sid = '$sid'";

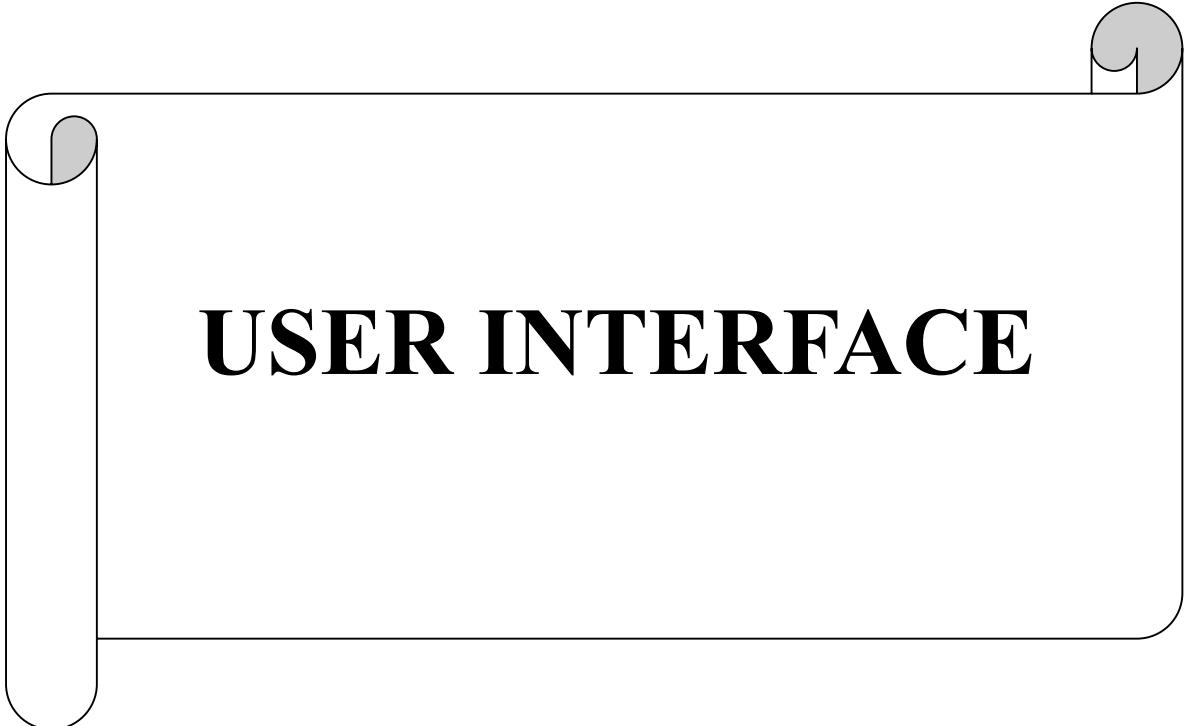
$result1 = mysqli_query($conn, $query1);

if ($result1->num_rows > 0) {

    $issued = 'yes';
}
```

```

echo "<p>Attendance has already been issued on this date, submitting will add additional
attendance</p>";
} else {
    $issued = 'no';
}echo '<div class="tabletop"><label>Date: ' . $currentDate . '</label>';
echo '<form action="submit.php?course=' . $course . '&class=' . $class . '&sem=' . $sem .
'&sub=' . $sub . '&sid=' . $sid . '&issued=' . $issued . '" method="POST">
    <label>Select Hours:</label>
    <select name="hours">
        <option value="1">1 hour</option>
        <option value="2">2 hours</option>
        <option value="3">3 hours</option>
        <option value="4">4 hours</option>
    </select></div>
<div class="table_selection">
    <table> <thead>
        <tr><td>Roll number</td>
        <td>Name</td>
        <td><input type="checkbox" id="select-all" onchange="toggleCheckboxes()"> All
Present</td></tr>
    </thead>';
while ($row = mysqli_fetch_assoc($result)) {
    echo '<tr id="tablerow">
        <td>' . $row["rollno"] . '</td>
        <td>' . $row["firstname"] . ' ' . $row["lastname"] . '</td>
        <td><input type="checkbox" name="' . $row['rollno'] . '" value="present"></td>
    </tr>';
}
echo '</table> <div class="bt�s">
    <button type="submit" id="submitAttendance" name="submitAttendance">Submit
Attendance</button>
    <button type="submit" id="backbtn" name="backbtn" value="Back">Back</button>
</div></div></form>';
?>
```



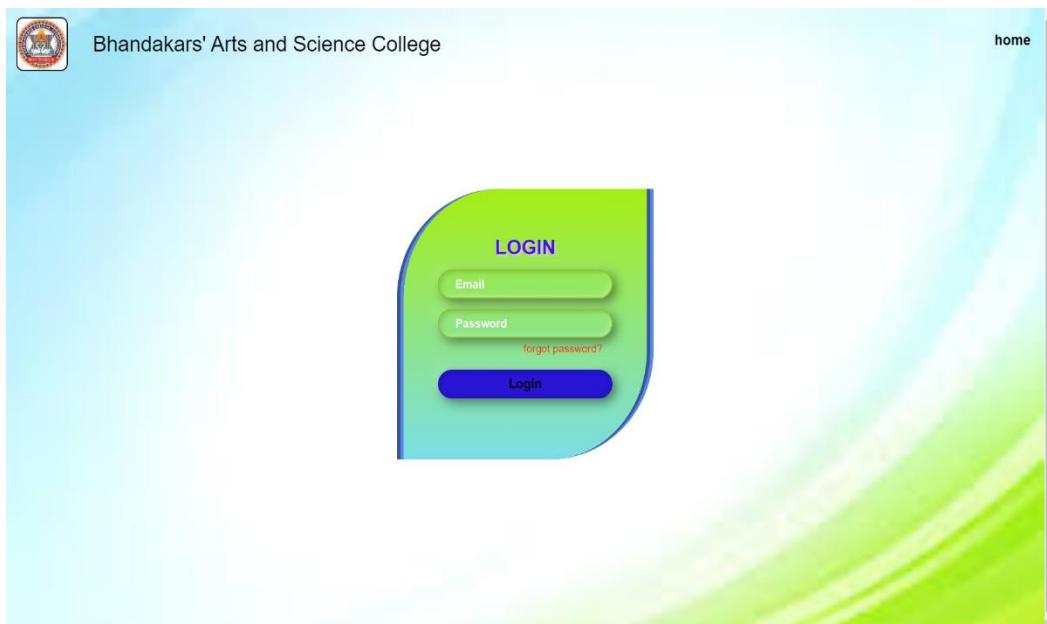
USER INTERFACE

7.USER INTERFACE

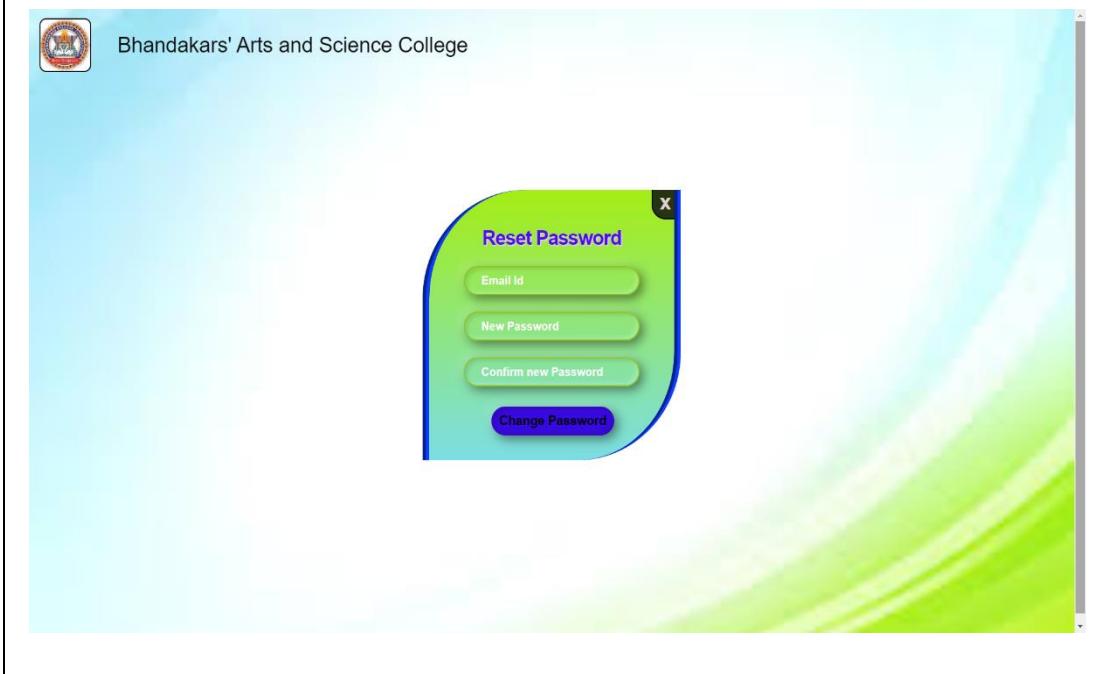
Home page



Login



Change password



Admin Dashboard

A screenshot of the Admin Dashboard. The dashboard has a light blue header with the word "DASHBOARD" and a user profile icon. On the left, there is a vertical sidebar with icons for users, courses, staff, and calendar. The main area displays three summary cards: "Total Number of Students" (100), "Total Number of Staff" (41), and "Total Number of Courses" (5). To the right, there is a profile section for "Narayana Shetty" with details: Email: admin@gmail.com, Address: Kundapura, Gender: Male, Contact: 9988998897. A "Update My Profile" button is also present. The overall layout is clean and modern.

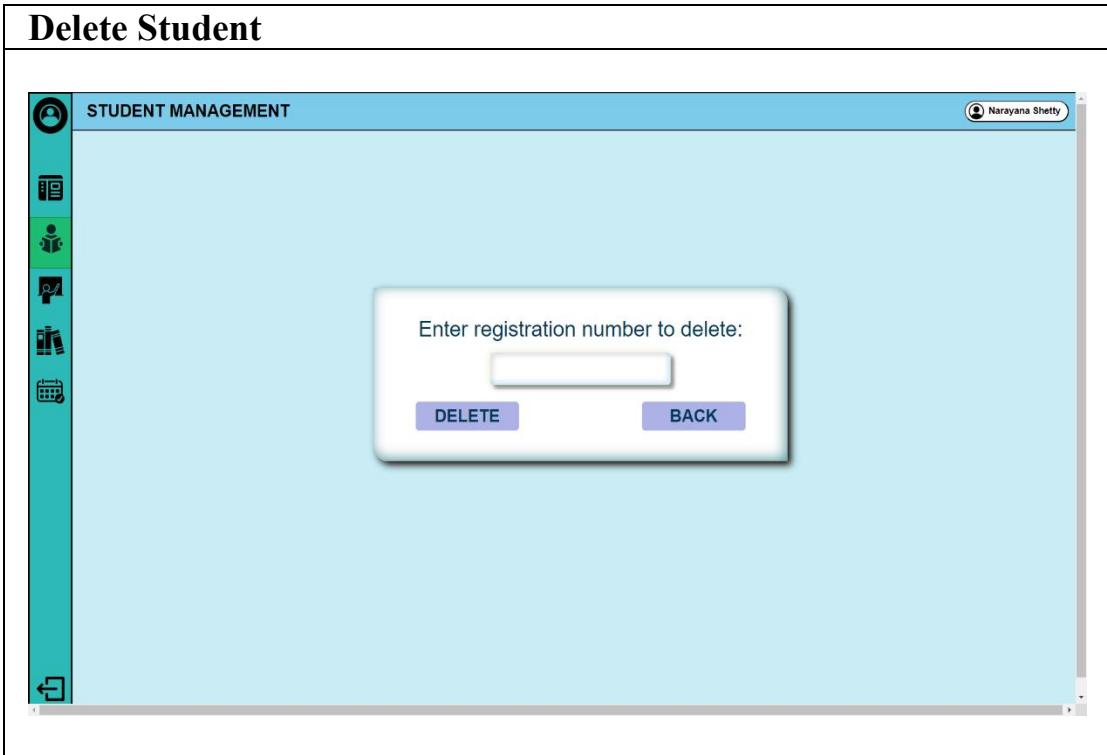
Student Management

The screenshot shows the 'STUDENT MANAGEMENT' application interface. On the left, there is a vertical sidebar titled 'ADMIN' containing icons and links for 'Dashboard', 'Student Management' (which is highlighted in green), 'Staff Management', 'Subject Management', and 'Attendance Report'. At the bottom of the sidebar is a 'Logout' button. The main content area is titled 'STUDENT MANAGEMENT' and contains four buttons: 'Add New Student', 'Delete Student', 'View Student', and 'Update Student'. In the top right corner, there is a user profile icon labeled 'Narayana Shetty'.

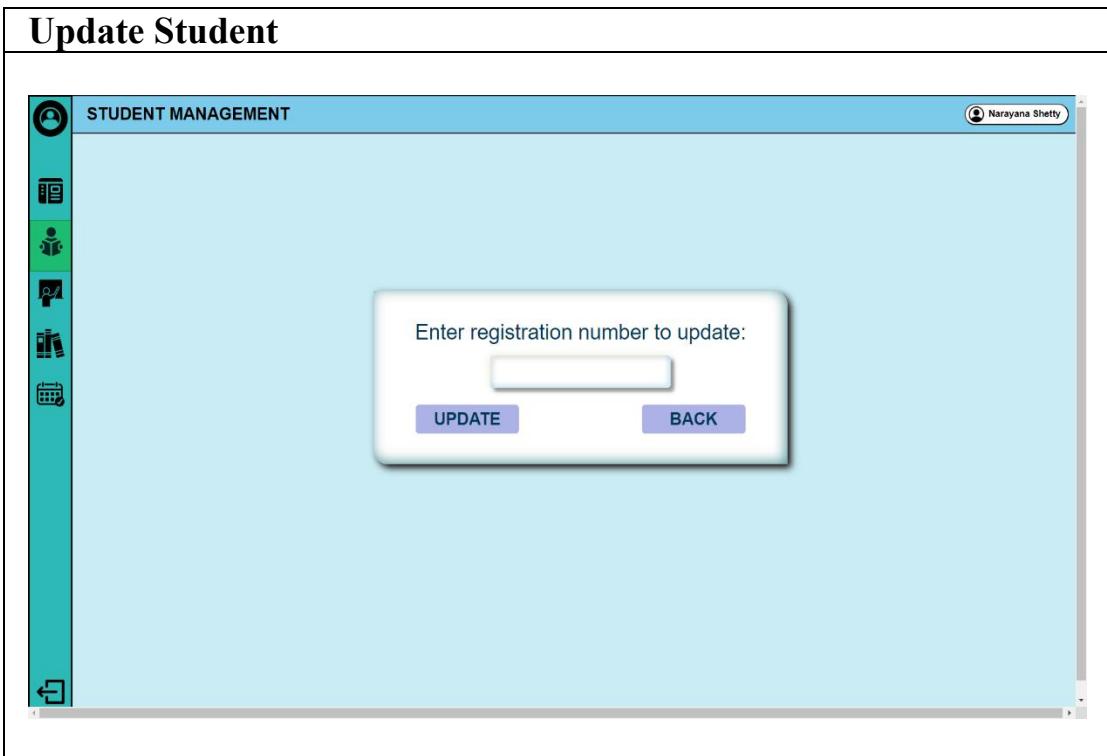
Add Student

The screenshot shows the 'STUDENT MANAGEMENT' application interface with the 'Add Student' form. The sidebar on the left has the 'Student Management' link selected. The main form is titled 'Student Registration' and contains the following fields: First Name, Last Name, Roll Number, Register Number, Father Name, Mother Name, Contact Number, Parent Contact Number, Email Id, Password, Address, Gender (Male, Female, Other), DOB (dd-mm-yyyy), Select Year, Select Semester, and Select Course. At the bottom of the form are three buttons: 'RESET', 'REGISTER' (highlighted in blue), and 'CANCEL'.

Delete Student



Update Student



View student

STUDENT MANAGEMENT									Narayana Shetty
Student details									
First Name	Last Name	Roll Number	Register Number	Father Name	Mother Name	Contact Number	Parent Contact	Email	
Anusha	Rai	200701	2012315701	ganira	Parvathi	8213212345	8887766555	anush@	
Monica	Shetty	200702	2012315702	Nagendra	Manasa	9845623415	7546732456	monica@	
Namratha	Poojari	200703	2012315703	Praveen	Nisha	7657456323	9856745783	namratha@	
Niranjan	Shetty	200704	2012315704	Rajesh	Nimitha	8792345678	9878943523	niru@g	
Nishanth	Shetty	200705	2012315705	Krishna	Nirmala	7845793245	8967584532	nishanth@	
Pratheeksha	Shetty	200706	2012315706	Pavan	Parvathi	6578456324	7894567356	pratheeksha@	
Udith	Devaiga	200201	2012315501	Uday	Usha	7687967845	7945678934	Udith@g	
Unnathi	Rav	200202	2012315502	Umesh	Urmila	6787954678	7896754698	unnathi@	
Diya	Hegde	200203	2012315503	Dananjay	Deeksha	6578987890	6570982345	diya@g	
Gandana	Kharvi	200204	2012315504	Ganesh	Gowri	6789890878	6898789453	gandana@	
Anil	Kumar	200205	2012315505	Avinash	Amrutha	7678986789	7896785678	anil@g	
Sharath	Poojari	200301	2012315401	Shekar	Saroja	8794567291	7896789355	sharath@g	
Ishan	Devadiga	200302	2012315402	Suresh	Sneha	6786789546	7896789567	ishan@e	

Add Staff

STAFF MANAGEMENT

Staff Registration

First Name	Last Name
Staff Id	Contact Number
Email Id	Password
Address	Gender: <input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other
Experience: <input type="text"/> years <input type="text"/> months	DOB: dd-mm-yyyy <input type="button" value=""/>
Select Qualification	DOJ: dd-mm-yyyy <input type="button" value=""/>
Select Department	
RESET REGISTER CANCEL	

Add Subject

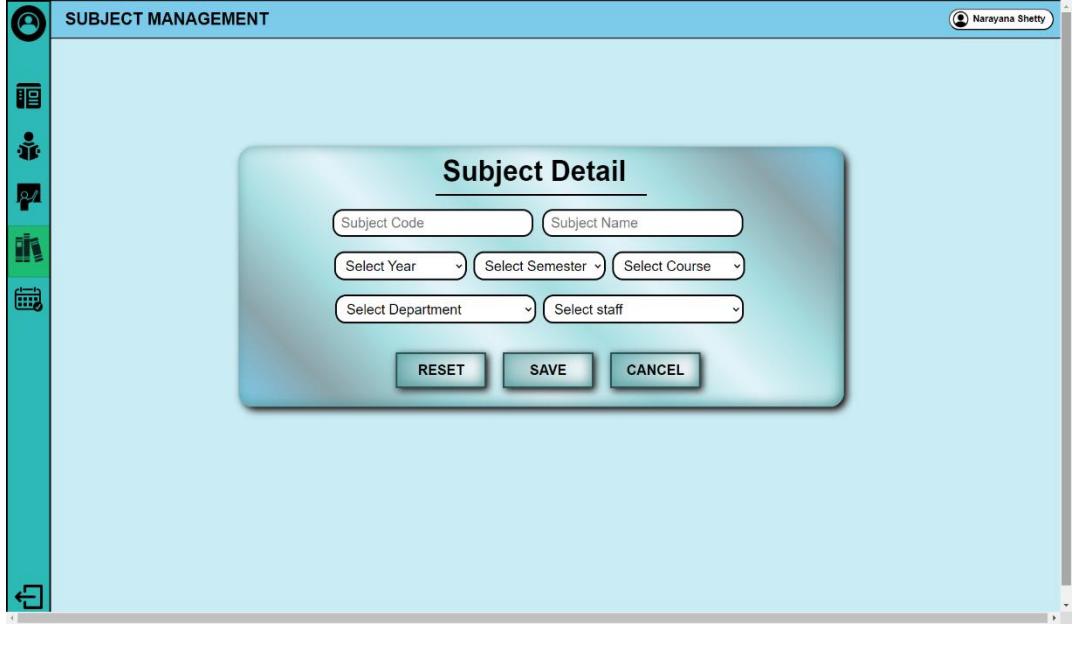
SUBJECT MANAGEMENT

Narayana Shetty

Subject Detail

Subject Code	Subject Name	
Select Year	Select Semester	Select Course
Select Department	Select staff	

RESET SAVE CANCEL



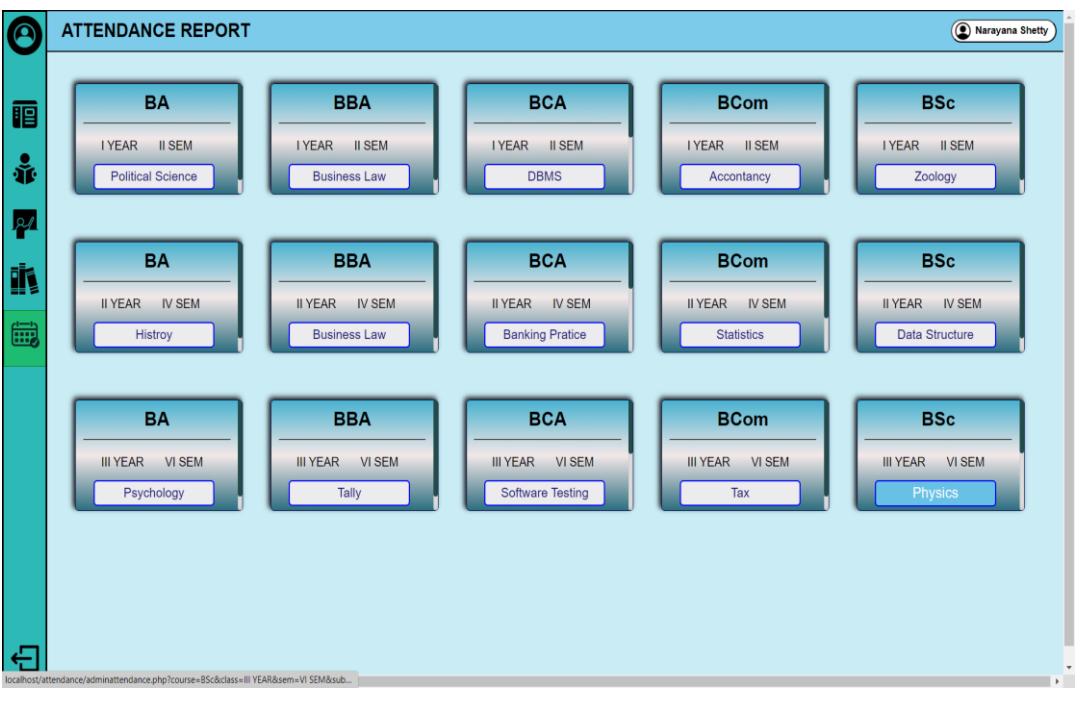
Attendance Report

ATTENDANCE REPORT

Narayana Shetty

BA I YEAR II SEM Political Science	BBA I YEAR II SEM Business Law	BCA I YEAR II SEM DBMS	BCom I YEAR II SEM Accountancy	BSc I YEAR II SEM Zoology
BA II YEAR IV SEM Histroy	BBA II YEAR IV SEM Business Law	BCA II YEAR IV SEM Banking Pratice	BCom II YEAR IV SEM Statistics	BSc II YEAR IV SEM Data Structure
BA III YEAR VI SEM Psychology	BBA III YEAR VI SEM Tally	BCA III YEAR VI SEM Software Testing	BCom III YEAR VI SEM Tax	BSc III YEAR VI SEM Physics

localhost/attendance/adminattendance.php?course=BSc&class=III YEAR&sem=VI SEM&sub...



View Report

ATTENDANCE REPORT

Starting Date: 19-06-2023

Ending Date: 21-06-2023

Filter Student

Back

Name	Roll No	19-06-2023 Total class : 1	20-06-2023 Total class : 1	21-06-2023 Total class : 2	Present/Total Class	Percentage
Samanth Shetty	200501	0	1	0	1/4	25.00 %
Rachana Devadiga	200502	1	1	1	3/4	75.00 %
Rohan Devadiga	200503	1	1	2	4/4	100.00 %
Nikhil Shetty	200504	1	1	2	4/4	100.00 %
Nayana Kharvi	200505	0	1	2	3/4	75.00 %
Poorvi Devadiga	200506	1	1	2	4/4	100.00 %
Manu Ganiga	200507	1	1	2	4/4	100.00 %
Ruthuja Shetty	200508	0	1	0	1/4	25.00 %
Mithun Bhat	200509	1	1	2	4/4	100.00 %
Roja Kharvi	200510	1	1	2	4/4	100.00 %

Update Admin Profile

My Profile

Name : Narayana Shetty
Email : admin@gmail.com
Password :
Contact : 9988998897
Address : Kundapura
Gender : Male

Update

Back

Mark Today's Attendance

Date: 23-06-2023

Select Hours: 1 hour

Roll number	Name	<input checked="" type="checkbox"/> All Present
200501	Samanth Shetty	<input checked="" type="checkbox"/>
200502	Rachana Devadiga	<input checked="" type="checkbox"/>
200503	Rohan Devadiga	<input checked="" type="checkbox"/>
200504	Nikhil Shetty	<input checked="" type="checkbox"/>
200505	Nayana Kharvi	<input checked="" type="checkbox"/>
200506	Poorvi Devadiga	<input checked="" type="checkbox"/>
200507	Manu Ganiga	<input checked="" type="checkbox"/>
200508	Ruthuja Shetty	<input checked="" type="checkbox"/>
200509	Mithun Bhat	<input checked="" type="checkbox"/>
200510	Roja Kharvi	<input checked="" type="checkbox"/>

SUBMIT ATTENDANCE

BACK

Update Attendance

Select Date: 22 - 06 - 2023

Roll number	Name	Present
200501	Samanth Shetty	<input type="checkbox"/>
200502	Rachana Devadiga	<input checked="" type="checkbox"/>
200503	Rohan Devadiga	<input type="checkbox"/>
200504	Nikhil Shetty	<input type="checkbox"/>
200505	Nayana Kharvi	<input type="checkbox"/>
200506	Poorvi Devadiga	<input type="checkbox"/>
200507	Manu Ganiga	<input type="checkbox"/>
200508	Ruthuja Shetty	<input type="checkbox"/>
200509	Mithun Bhat	<input type="checkbox"/>
200510	Roja Kharvi	<input type="checkbox"/>

UPDATE ATTENDANCE

BACK

DELETE ATTENDANCE

View Report

Attendance Report

Starting Date: 19-06-2023

Ending Date: 21-06-2023

[Filter Student](#)

[Back](#)

Name	Roll No	19-06-2023 Total class : 1	20-06-2023 Total class : 1	21-06-2023 Total class : 2	Present/Total Class	Percentage
Samanth Shetty	200501	0	1	0	1/4	25.00 %
Rachana Devadiga	200502	1	1	1	3/4	75.00 %
Rohan Devadiga	200503	1	1	2	4/4	100.00 %
Nikhil Shetty	200504	1	1	2	4/4	100.00 %
Nayana Kharvi	200505	0	1	2	3/4	75.00 %
Poorvi Devadiga	200506	1	1	2	4/4	100.00 %
Manu Ganiga	200507	1	1	2	4/4	100.00 %
Ruthuja Shetty	200508	0	1	0	1/4	25.00 %
Mithun Bhat	200509	1	1	2	4/4	100.00 %
Roja Kharvi	200510	1	1	2	4/4	100.00 %

View Shortage

Attendance Report

Starting Date: 20-06-2023

Ending Date: 22-06-2023

[Filter Student](#)

[Back](#)

Name	Roll No	20-06-2023 Total class : 1	21-06-2023 Total class : 2	22-06-2023 Total class : 1	Present/Total Class	Percentage
Samanth Shetty	200501	1	0	0	1/4	25.00 %
Rachana Devadiga	200502	1	1	0	2/4	50.00 %
Ruthuja Shetty	200508	1	0	1	2/4	50.00 %

Student View attendance

VIEW ATTENDANCE

Rachana Devadiga

Rachana Devadiga	
Reg No	: 2012315202
Roll No	: 200502
Course	: BSc
Year	: III YEAR
Sem	: VI SEM
Email	: rachana@gmail.com
Contact	: 8792456824
Address	: Mogeri
Gender	: female

Java
BSCCAC200
[View Attendance](#)

Physics
BSCCAC203
[View Attendance](#)

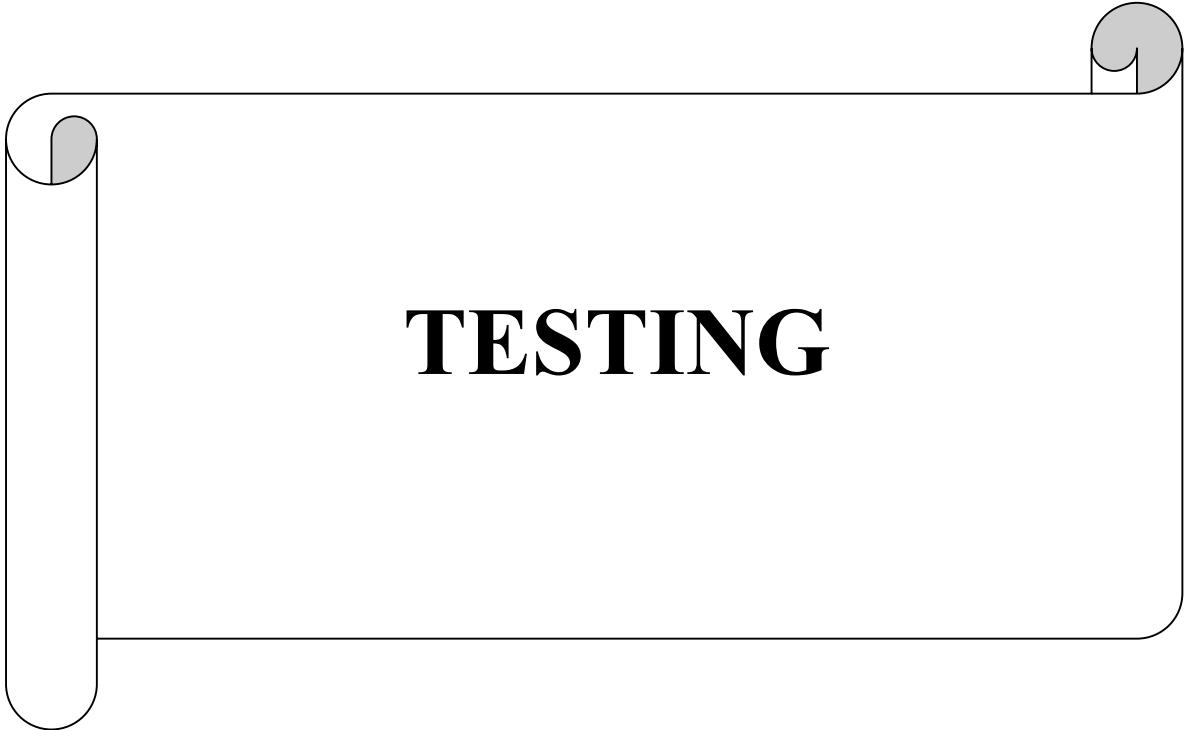
VB.NET
BSCCAC201
[View Attendance](#)

Maths-1
BSCCAC204
[View Attendance](#)

Student Attendance Report

ATTENDANCE REPORT

Name	Roll No	19-06-2023 Total class : 1	20-06-2023 Total class : 1	21-06-2023 Total class : 2	Present/Total Class	Percentage
Rachana Devadiga	200502	1	1	1	3/4	75.00 %



TESTING

8. TESTING

8.1 Introduction

Testing is the major quality control measures and during the software development it is used to detect errors that could have occurred during any of the phase like requirement analysis, design, coding. The goal of the testing is to uncover errors in the program.

8.2 Levels of Testing

Testing is done in different levels which includes the following.

- Unit Testing
- Integration Testing
- System testing
- Acceptance testing

- **Unit Testing**

In Unit testing each module gets tested during the coding phase itself.

The purpose is to exercise the different parts of the module code to detect the coding errors.

- **Integration Testing**

After new testing the modules are gradually integrated into sub systems. It is performed to detect design errors by focusing on testing the interconnection between modules.

- **System Testing**

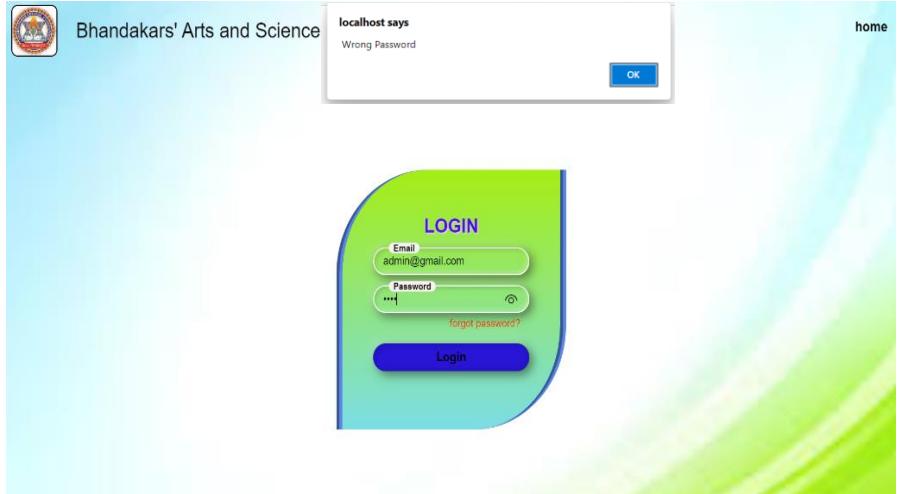
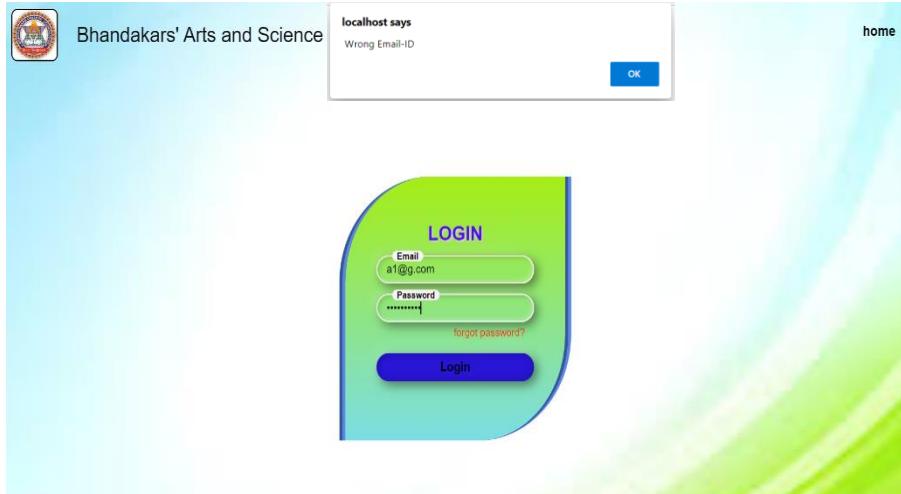
System is tested against the system requirement if all the requirements are met and if the system performs as specified by the requirement.

- **Acceptance Testing**

It is performed to demonstrate to the client on real life data of the client, the operation of the system.

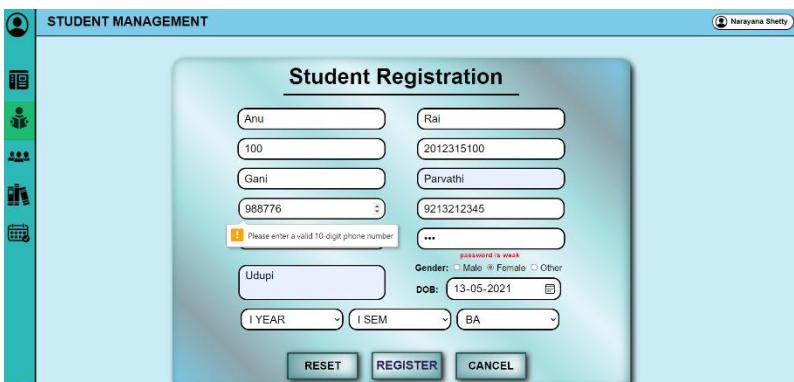
8.3 Test Case

It is the input that tests the genuineness of the program and successful execution of the test case reveals. That there are no errors in the program that are under testing. It is a set of conditions or variables under which tester will determine whether an application or software is working currently.

Test case-ID	01
Test case title	Login
Purpose of testing	Testing the login
Test data	Email Id and Password
Steps	<ol style="list-style-type: none"> 1. If Email Id and Password is valid then 2. Redirects to certain page. 3. Else display warning message with description.
Expected output	<p>Invalid output:</p>  

<p>Valid output:</p>	

Table 8.1 Testing for login module

Test case-ID	02
Test case title	Add student details
Purpose of testing	Creating new student profile
Test data	First name, Last name, Roll No, Register No, Contact number, Parent Contact Number, Father Name, Mother Name, Address, Gender, DOB, Email-ID, Password, Year, Semester, Course.
Steps	<ol style="list-style-type: none"> 1. Click on add new student 2. Enter First name, Last name, Roll No, Register No, Contact number, Parent Contact Number, Father Name, Mother Name, Address, Gender, DOB, Email-ID, Password, Year, Semester, Course. 3. Click on Register Button. 4. If record is valid display register successful. 5. Else display warning message with description.
Expected output	<p>Invalid output:</p>  <p>The screenshot shows the 'Student Registration' form with two validation errors highlighted in yellow:</p> <ul style="list-style-type: none"> The 'Roll Number' field has an error message: "Field is required. Please fill in this field." The 'Contact Number' field has an error message: "Please enter a valid 10-digit phone number." <p>Other fields like First Name, Last Name, Register Number, Father Name, Mother Name, Email Id, Password, Address, DOB, Gender, Year, Semester, and Course are filled correctly.</p>  <p>The screenshot shows the 'Student Registration' form with validation errors for the 'Contact Number' and 'DOB' fields:</p> <ul style="list-style-type: none"> The 'Contact Number' field has an error message: "Please enter a valid 10-digit phone number." The 'DOB' field has an error message: "Please enter a valid date." <p>Other fields like First Name, Last Name, Register Number, Father Name, Mother Name, Email Id, Password, Address, DOB, Gender, Year, Semester, and Course are filled correctly.</p>

STUDENT MANAGEMENT

Student Registration

Anu12	Rai
20	1 Don't enter the numbers
ganira	Parvathi
8213212345	8887766555
anush@gmail.com	*****
Udupi	password is strong
Gender: <input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other	
DOB: 04-06-2003	
I YEAR	I SEM
BBA	

RESET REGISTER CANCEL

STUDENT MANAGEMENT

localhost says
This registration number already existed

OK

Anush	Rai
200	2012315100
ganira	Parvathi
8213212345	8887766555
anush@gmail.com	*****
Udupi	password is strong
Gender: <input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other	
DOB: 04-06-2003	
I YEAR	I SEM
BBA	

RESET REGISTER CANCEL

Valid output:

STUDENT MANAGEMENT

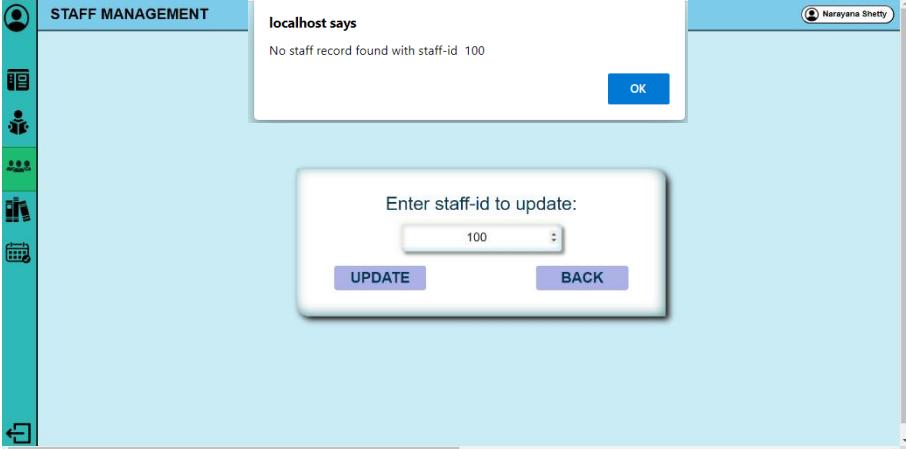
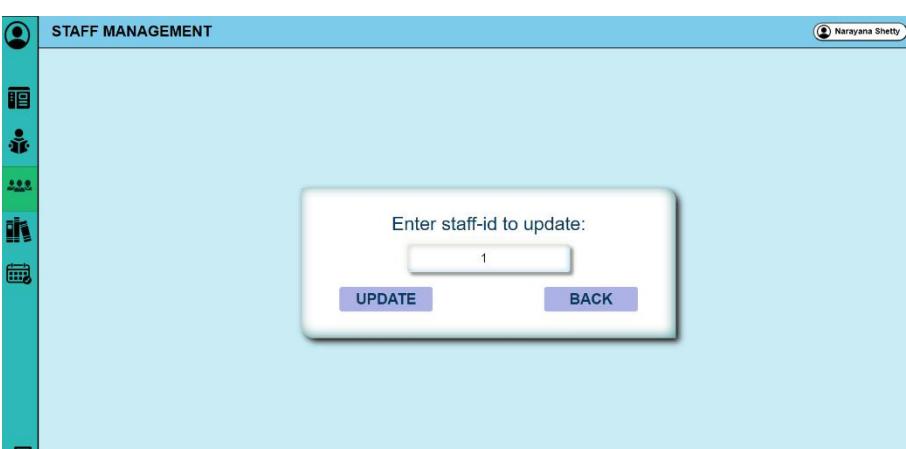
localhost says
Registered successfully

OK

Anusha	Rai
200	2012315500
ganira	Parvathi
8213212345	8887766555
anush@gmail.com	*****
Udupi	password is strong
Gender: <input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other	
DOB: 04-06-2003	
I YEAR	I SEM
BBA	

RESET REGISTER CANCEL

Table 8.2 Testing for add student details

Test case-ID	03
Test case title	Update staff details
Purpose of testing	Testing for update
Test data	Staff Id
Steps	<ol style="list-style-type: none"> 1. Click on update staff. 2. Enter the Staff Id 3. Click on Update Button. 4. If Staff Id is existed then allows to update record. 5. If updated record is valid display updated successful. 6. Else display warning message with description. 7. If Staff Id does not exist then display warning message with description.
Expected output	<p>Invalid output:</p>  <p>Valid output:</p> 

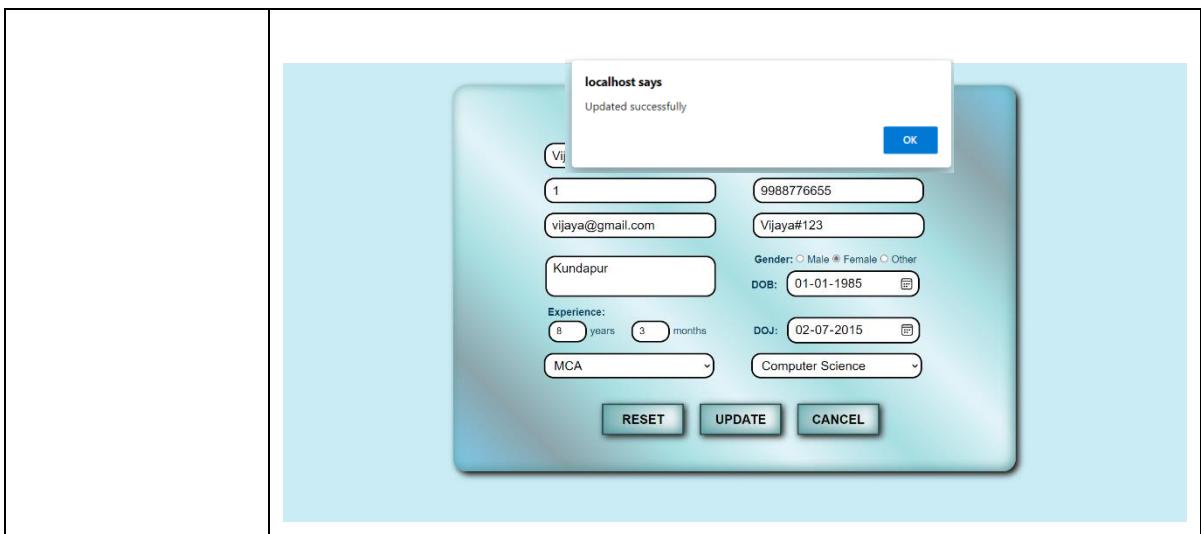


Table 8.3 Testing for update staff details

Test case-ID	04
Test Case Title	Delete subject details
Purpose of testing	Testing for delete
Test data	Subject Code
Steps	<ol style="list-style-type: none"> 1. Click on delete subject 2. Enter the Subject Code 3. Click on Delete Button 4. If Subject Code is existed then display subject record 5. If confirm delete button is clicked then displays deleted successful message. 6. If Subject code does not exist then display warning message with description.
Expected output:	<p>Invalid output:</p> <p>The screenshot shows a sidebar menu with icons for user management, subjects, courses, and reports. The main area is titled "SUBJECT MANAGEMENT". A modal dialog box displays the message "localhost says No subject record found with scode zzz" with an "OK" button. Below this, another modal dialog box asks "Enter subject code to delete:" with a text input field containing "zzz" and buttons for "DELETE" and "BACK".</p>

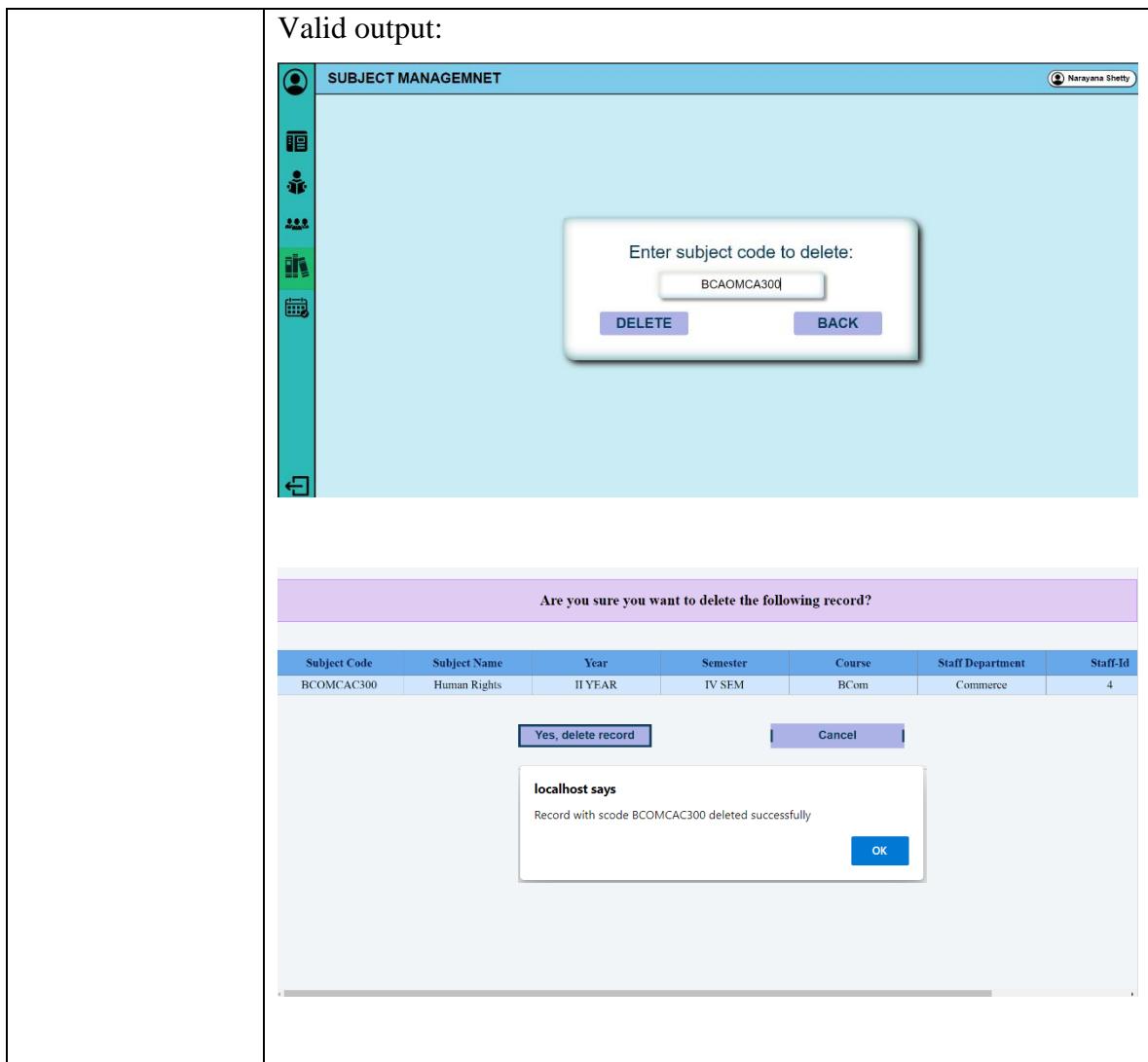


Table 8.4 Testing for delete subject details

Test case-ID	05
Test Case Title	Attendance report by admin
Purpose of testing	Testing for attendance report
Test data	Year, Course, Sem, Subject Name, Starting date, Ending date
Steps	<ol style="list-style-type: none"> 1. Select Year, Course, Sem. 2. Click on view report. 3. Input starting date and ending date. 4. Click on filter student button. 5. If selected date is valid then display attendance report. 6. Else display warning message with description.

Expected Output

Invalid output:

The screenshot shows a web-based attendance reporting system. At the top, there's a header bar with the title "ATTENDANCE REPORT". Below it, there are two input fields: "Starting Date" set to "19-06-2023" and "Ending Date" set to "15-06-2023". To the right of these are two buttons: "Filter Student" (green) and "Back" (orange). A modal dialog box is centered on the page, titled "localhost says". It contains the message "Invalid date range. First date should be less than or equal to the second date." at the top, followed by an "OK" button at the bottom.

This screenshot is similar to the one above, showing the same "ATTENDANCE REPORT" interface. The "Starting Date" field is now empty ("dd-mm-yyyy") and the "Ending Date" field is set to "01-06-2023". The "Filter Student" and "Back" buttons remain the same. The modal dialog box displays the message "Please select two dates." and an "OK" button.

Valid output:

The screenshot shows the same "ATTENDANCE REPORT" interface with valid date inputs: "Starting Date" is "20-06-2023" and "Ending Date" is "22-06-2023". The "Filter Student" and "Back" buttons are present. The main content area displays a table of student attendance data:

Name	Roll No	20-06-2023 Total class : 1	21-06-2023 Total class : 2	22-06-2023 Total class : 1	Present/Total Class	Percentage
Samanth Shetty	200501	1	0	0	1/4	25.00 %
Rachana Devadiga	200502	1	1	0	2/4	50.00 %
Rohan Devadiga	200503	1	2	1	4/4	100.00 %
Nikhil Shetty	200504	1	2	1	4/4	100.00 %
Nayana Kharvi	200505	1	2	1	4/4	100.00 %
Poorvi Devadiga	200506	1	2	1	4/4	100.00 %
Manu Ganiga	200507	1	2	1	4/4	100.00 %
Ruthuja Shetty	200508	1	0	1	2/4	50.00 %
Mithun Bhat	200509	1	2	1	4/4	100.00 %
Roja Kharvi	200510	1	2	1	4/4	100.00 %

Table 8.5 Testing for attendance report

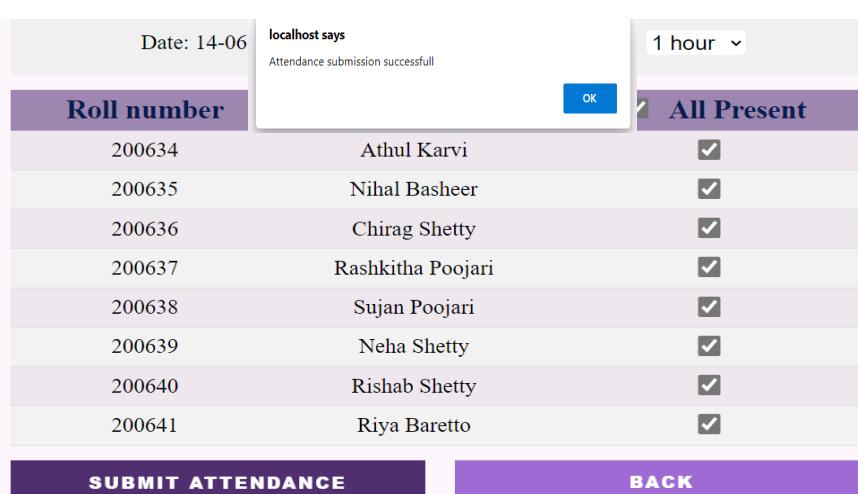
Test case-ID	06																		
Test Case Title	Today's attendance																		
Purpose of testing	Testing for today's attendance																		
Test data	Year, Course, Sem, Subject Name, Marks attendance.																		
Steps	<ol style="list-style-type: none"> 1. Click on mark attendance for particular year, sem, course. 2. Select number of hours and tick the checkbox. 3. Click on Submit Attendance Button. 4. It display attendance submission Successful. 																		
Expected output	<p>Valid output:</p>  <table border="1" style="margin-top: 10px; width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #546A9B; color: white;">Roll number</th> <th style="background-color: #546A9B; color: white;"></th> </tr> </thead> <tbody> <tr><td>200634</td><td>Athul Karvi</td></tr> <tr><td>200635</td><td>Nihal Basheer</td></tr> <tr><td>200636</td><td>Chirag Shetty</td></tr> <tr><td>200637</td><td>Rashkitha Poojari</td></tr> <tr><td>200638</td><td>Sujan Poojari</td></tr> <tr><td>200639</td><td>Neha Shetty</td></tr> <tr><td>200640</td><td>Rishab Shetty</td></tr> <tr><td>200641</td><td>Riya Baretto</td></tr> </tbody> </table>	Roll number		200634	Athul Karvi	200635	Nihal Basheer	200636	Chirag Shetty	200637	Rashkitha Poojari	200638	Sujan Poojari	200639	Neha Shetty	200640	Rishab Shetty	200641	Riya Baretto
Roll number																			
200634	Athul Karvi																		
200635	Nihal Basheer																		
200636	Chirag Shetty																		
200637	Rashkitha Poojari																		
200638	Sujan Poojari																		
200639	Neha Shetty																		
200640	Rishab Shetty																		
200641	Riya Baretto																		

Table 8.6 Testing for today's attendance

Test case-ID	07
Test Case Title	Update Attendance
Purpose of testing	Testing for update attendance
Test data	Year, Course, Sem, Subject Name, Starting date, Ending date.
Steps	<ol style="list-style-type: none"> 1. Click on update attendance for particular year, sem, course. 2. Select date. 3. If attendance marked on selected date then 4. Make changes and click on update attendance button. 5. If valid display attendance updated Successful. 6. Else select number of hours and tick checkbox. 7. Click on submit attendance button. 8. It display attendance submission Successful.

Expected Output

Invalid output:

Roll number	Name	Present
200501	Samantha Shetty	4+
200502	Rachana Devadiga	1
200503	Rohan Devadiga	1
200504	Nikhil Shetty	1
200505	Nayana Kharvi	0
200506	Poorvi Devadiga	1
200507	Manu Ganiga	1
200508	Ruthuja Shetty	0
200509	Mithun Bhat	1
200510	Roja Kharvi	1

UPDATE ATTENDANCE BACK DELETE ATTENDANCE

Valid output:

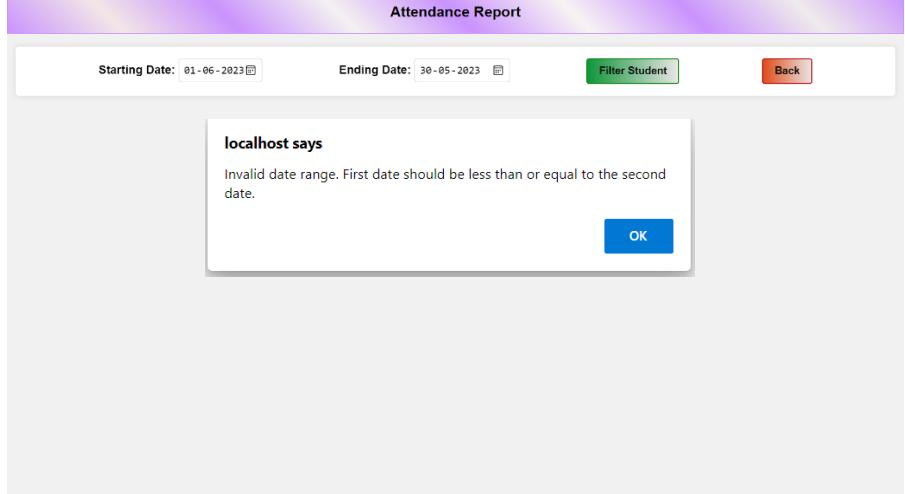
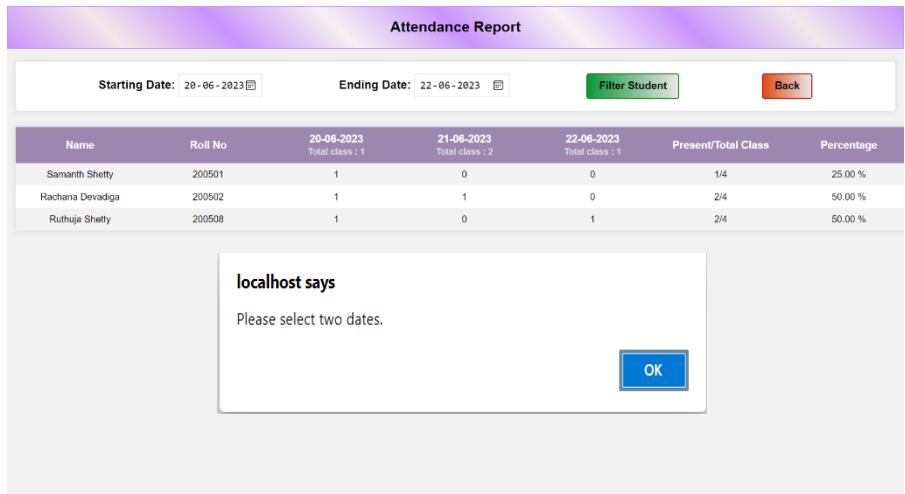
Roll number	Name	Present
200501	Samantha Shetty	1
200502	Rachana Devadiga	1
200503	Rohan Devadiga	1
200504	Nikhil Shetty	1
200505	Nayana Kharvi	1
200506	Poorvi Devadiga	1
200507	Manu Ganiga	1
200508	Ruthuja Shetty	1
200509	Mithun Bhat	1
200510	Roja Kharvi	1

UPDATE ATTENDANCE BACK DELETE ATTENDANCE

Roll number	Name	Present
200501	Samantha Shetty	✓
200502	Rachana Devadiga	□
200503	Rohan Devadiga	□
200504	Nikhil Shetty	✓
200505	Nayana Kharvi	✓
200506	Poorvi Devadiga	✓
200507	Manu Ganiga	✓
200508	Ruthuja Shetty	✓
200509	Mithun Bhat	✓
200510	Roja Kharvi	✓

SUBMIT ATTENDANCE BACK

Table 8.7 Testing for update attendance

Test case-ID	08
Test Case Title	View attendance report by staff
Purpose of testing	Testing for attendance report
Test data	Year, Course, Sem, Subject Name, Starting date, Ending date
Steps	<ol style="list-style-type: none"> 1. Select Year, Course, Sem. 2. Click on view report. 3. Input starting date and ending date. 4. Click on filter student button. 5. If selected date is valid then display attendance report. 6. Else display warning message with description.
Expected Output	<p>Invalid output:</p>  

	<p>Valid output:</p> <p>The screenshot shows an 'Attendance Report' interface. At the top, there are input fields for 'Starting Date' (19-06-2023) and 'Ending Date' (21-06-2023), a 'Filter Student' button, and a 'Back' button. Below this is a table with columns: Name, Roll No, 19-06-2023 (Total class : 1), 20-08-2023 (Total class : 1), 21-06-2023 (Total class : 2), Present/Total Class, and Percentage. The data rows show student names like Samanth Shetty, Rachana Devadiga, Rohan Devadiga, etc., along with their roll numbers and attendance counts.</p>
--	---

Table 8.8 Testing for view attendance report by staff

Test case-ID	09
Test Case Title	View Shortage Report
Purpose of testing	Testing for shortage report
Test data	Year, Course, Sem, Subject Name, Starting date, Ending date
Steps	<ol style="list-style-type: none"> 1. Select Year, Course, Sem. 2. Click on view shortage. 3. Input starting date and ending date. 4. Click on filter student button. 5. If selected date is valid then display attendance shortage. 6. Else display warning message with description.
Expected Output	<p>Invalid output:</p> <p>The screenshot shows a 'localhost says' dialog box with the message: 'Invalid date range. First date should be less than or equal to the second date.' There is an 'OK' button at the bottom right of the dialog.</p>

	<p>The screenshot shows a web-based application titled "ATTENDANCE REPORT". At the top, there are input fields for "Starting Date" (08-06-2023) and "Ending Date" (dd-mm-yyyy), a "Filter Student" button, and a "Back" button. A modal window is displayed with the title "localhost says" and the message "Please select two dates." A blue "OK" button is at the bottom right of the modal.</p>																												
	<p>Valid output:</p> <p>The screenshot shows the same "Attendance Report" interface. The "Starting Date" is now 20-06-2023 and the "Ending Date" is 22-06-2023. The "Filter Student" and "Back" buttons are visible. Below these, a table displays student attendance data:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Roll No</th> <th>20-06-2023 Total class : 1</th> <th>21-06-2023 Total class : 2</th> <th>22-06-2023 Total class : 1</th> <th>Present/Total Class</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Samanth Shetty</td> <td>200501</td> <td>1</td> <td>0</td> <td>0</td> <td>1/4</td> <td>25.00 %</td> </tr> <tr> <td>Rachana Devadiga</td> <td>200502</td> <td>1</td> <td>1</td> <td>0</td> <td>2/4</td> <td>50.00 %</td> </tr> <tr> <td>Ruthuja Shetty</td> <td>200508</td> <td>1</td> <td>0</td> <td>1</td> <td>2/4</td> <td>50.00 %</td> </tr> </tbody> </table>	Name	Roll No	20-06-2023 Total class : 1	21-06-2023 Total class : 2	22-06-2023 Total class : 1	Present/Total Class	Percentage	Samanth Shetty	200501	1	0	0	1/4	25.00 %	Rachana Devadiga	200502	1	1	0	2/4	50.00 %	Ruthuja Shetty	200508	1	0	1	2/4	50.00 %
Name	Roll No	20-06-2023 Total class : 1	21-06-2023 Total class : 2	22-06-2023 Total class : 1	Present/Total Class	Percentage																							
Samanth Shetty	200501	1	0	0	1/4	25.00 %																							
Rachana Devadiga	200502	1	1	0	2/4	50.00 %																							
Ruthuja Shetty	200508	1	0	1	2/4	50.00 %																							

Table 8.9 Testing for view attendance shortage

Test case-ID	10
Test Case Title	View attendance by student
Purpose of testing	Testing for view attendance
Test data	Subject Name, Starting date, Ending date.
Steps	<ol style="list-style-type: none"> 1. Click on view report. 2. Input starting date and ending date. 3. Click on Filter Button. 4. If selected date is valid then display Attendance Report. 5. Else display warning message with description.

Expected
Output

Invalid output:

The screenshot shows a web-based application titled "ATTENDANCE REPORT". At the top, there are two input fields: "Starting Date" with the value "02-06-2023" and "Ending Date" with the value "29-05-2023". Below these are two buttons: a green "Filter" button and a red "Back" button. A modal dialog box is displayed in the center, titled "localhost says". It contains the message "Invalid date range. First date should be less than or equal to the second date." with an "OK" button at the bottom.

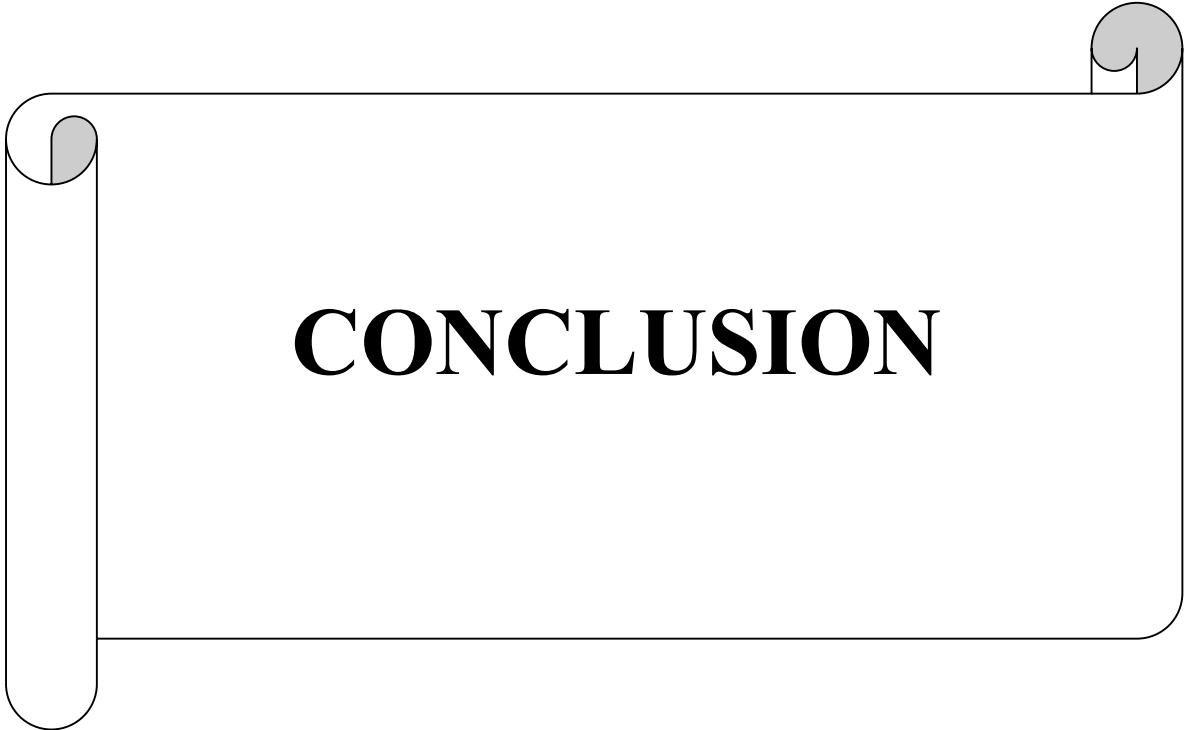
The screenshot shows the same "ATTENDANCE REPORT" interface. The "Starting Date" field is empty ("dd-mm-yyyy") and the "Ending Date" field has the value "01-06-2023". The "Filter" and "Back" buttons are present. A modal dialog box titled "localhost says" displays the message "Please select two dates." with an "OK" button.

Valid output:

The screenshot shows the "ATTENDANCE REPORT" interface with valid date inputs: "Starting Date" is "19-06-2023" and "Ending Date" is "21-06-2023". The "Filter" and "Back" buttons are visible. Below the date inputs is a table with the following data:

Name	Roll No	19-06-2023 Total class : 1	20-06-2023 Total class : 1	21-06-2023 Total class : 2	Present/Total Class	Percentage
Rachana Devadiga	200502	1	1	1	3/4	75.00 %

Table 8.10 Testing for view attendance



CONCLUSION

CONCLUSION

In conclusion, this project was successfully implemented using PHP, CSS, HTML. In this project the staff will not enter the student attendance manually rather he will do it in a computerized system. Admin will add all the academic details of each and every staff and the student and the attendance report generated by the staff can be viewed by the admin.

During the implantation we have faced many challenges in creating the modules like attendance report, update attendance and some coding errors but we handled all the challenges successfully during the development of the project work.

We have learned about the design, webpage, user interfaces and also logical working of the PHP backend.

Moreover, this project helped for us to understand Software Development life Cycle (SDLC), time bound work, team spirit and preparing project document, project testing, GUI designing and presentation.

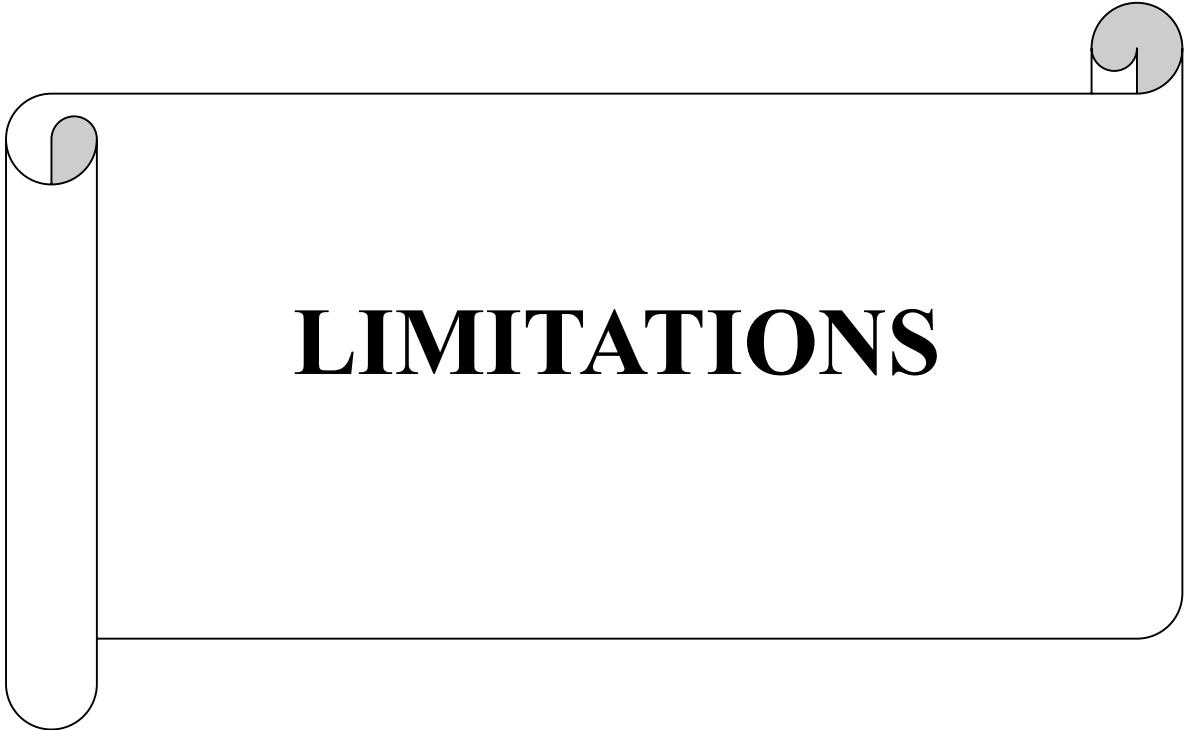
In addition to that, we learned PHP coding and HTML front end design, coding and working of this web application.

Finally concluded that we tried to fulfill the objectives of project work and goal of our project STUDENT ATTENDANCE MANAGEMENT SYSTEM.

Kavana Navada : 201231522269

Kavana : 201231522268

Sangeetha : 201231522222



LIMITATIONS

LIMITATIONS

- Not mobile optimized
- Network related issue is also one of the limitations of the student attendance management system
- Although attendance systems aim to minimize manual errors, there is still a possibility of human error during data entry or attendance marking.

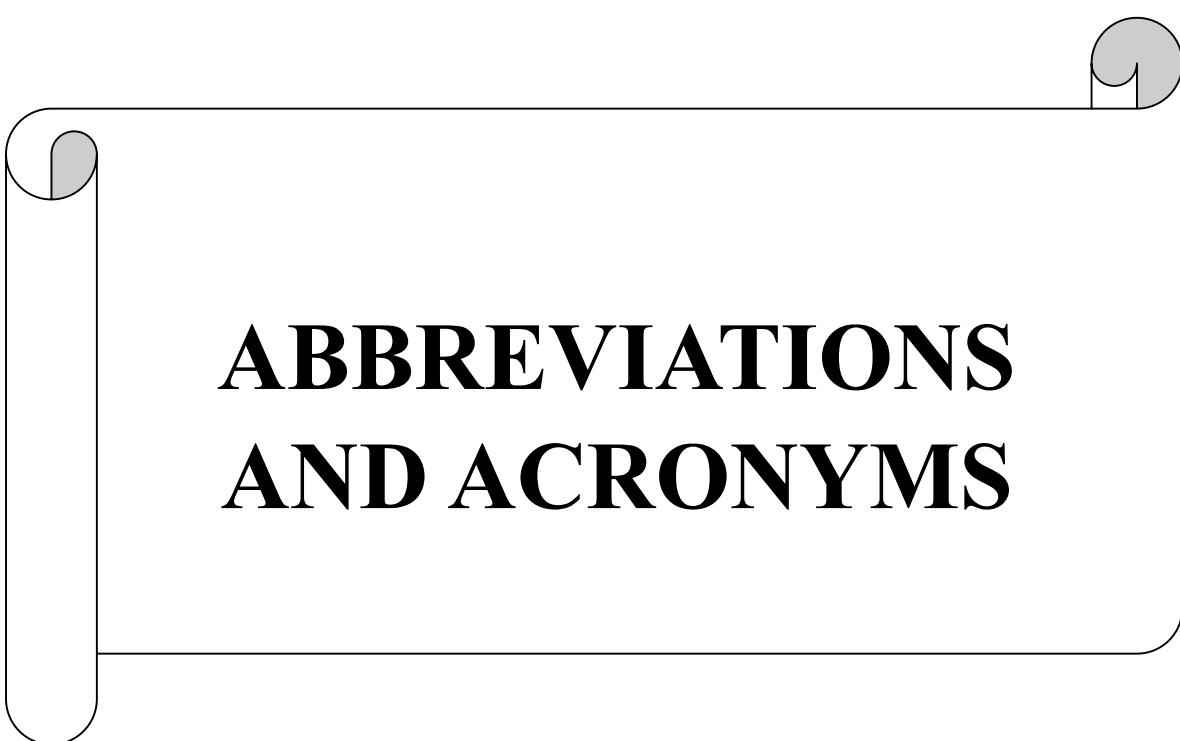


FUTURE SCOPE

FUTURE SCOPE

The project has very large scope in future. The project can be implemented in other colleges with extra technologies. It can also be implemented in some other institutes also. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion.

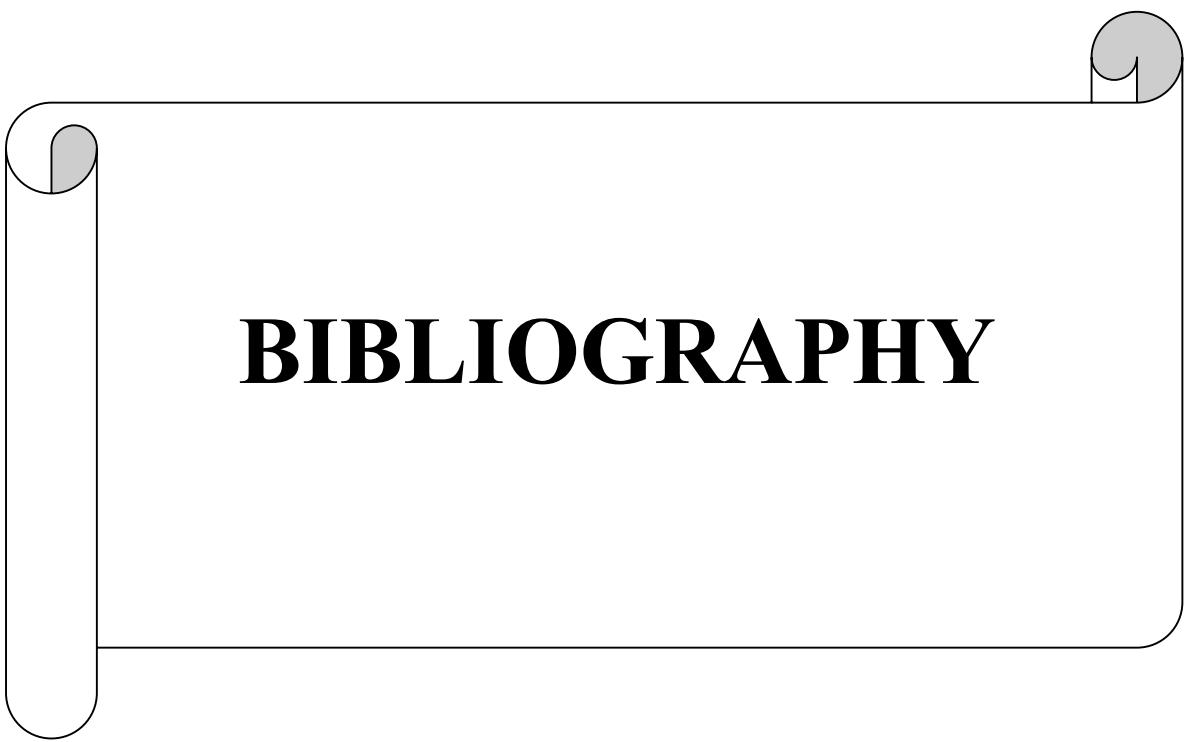
- With the help of QR Code scanner every staff and student can scan their own ID Card and can get their information and attendance also.
- To secure the data we can use figure print machine as an identification method of staff, student biometrics
- Integrate the fine payment module with the institution's student information system to ensure accurate tracking of the student details, fine history



ABBREVIATIONS AND ACRONYMS

ABBREVIATIONS AND ACRONYMS

CFD	: Control Flow Diagram
DFD	: Data Flow Diagram
ER	: Entity Relationship
SQL	: Structured Query Language
HTML	: Hypertext Markup Language
CSS	: Cascading Style Sheet
PHP	: Hypertext Pre-processor
AJAX	: Asynchronous JavaScript And XML
XAMPP	: X-operating system, Apache, Mysql, Php, Perl
RDBMS	: Relational Database Management System
MYSQL	: My Structured Query Language
DML	: Data Manipulation Language
DDL	: Data Definition Language
RAM	: Random Access Memory
DBMS	: Data Base Management System
OS	: Operating System
CPU	: Central Processing Unit
I/O	: Input Output



BIBLIOGRAPHY

BIBLIOGRAPHY

- **Text Book referred**

- An Integrated approach to Software Engineering-Pankaj Jalote.
- HTML5 and CSS3 made simple – Ivan Bayross, BPB Publication
- PHP and MYSQL Web Development – Luke Welling, Laura Thomson.

- **Website referred**

- <https://www.w3schools.com/php/>
- <https://stackoverflow.com>
- <https://www.google.com/>
- <https://chat.openai.com/>