UNIVERSITY MANAGEMENT SYSTEM

PROJECT REVIEW – II

DATE:

Submitted by :

NAME	**:**	**S.SUDHARSAN**
REG. NO.	**:**	**23PCA528**
CLASS	**:**	**MCA II-YEAR**

Internal Project Guide:

Dr.S.CHIDAMBARANATHAN, M.Sc., M.C.A., M.Phil., Ph.D.,

UNIVERSITY MANAGEMENT SYSYTEM

A PROJECT REPORT

Submitted by

S. SUDHARSAN

(Register No: 23PCA528)

in partial fulfillment of the requirements
for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Under the Guidance of

Dr.S.CHIDAMBARANATHAN, M.Sc., M.C.A., M.Phil., Ph.D.,

DEPARTMENT OF COMPUTER APPLICATIONS

ST. XAVIER‘S COLLEGE (AUTONOMOUS)

(Recognized as _College with Potential for Excellence‘ by UGC)

(Accredited by NAAC with —A++‖ Grade with a CGPA of 3.66 out of 4 in IV Cycle)

PALAYAMKOTTAI – 627002

2024-2025

DEPARTMENT OF COMPUTER APPLICATIONS

ST. XAVIER’S COLLEGE (AUTONOMOUS)

(Recognized as ‗College with Potential for Excellence‘ by UGC)

(Accredited by NAAC with ―A++‖ Grade with a CGPA of 3.66 out of 4 in IV Cycle)

PALAYAMKOTTAI - 627002

BONAFIDE CERTIFICATE

This is to certify that the project work entitled **“UNIVERSITY MANAGEMENT SYSTEM”** is the bona fide work of **S. SUDHARSAN (23PCA528)** who carried out the project work under my supervision and submitted during the academic year 2024-25.

The Viva-voce held on ………………...

INTERNAL EXAMINER **EXTERNAL EXAMINER**

HEAD OF THE DEPARTMENT

ACKNOWLEDGEMENT

At first I extend my deepest gratitude to the **ALMIGHTY** for providing me all the requirements for completing this project.

I extend my deepest sense of gratitude to **Rev.Fr.Dr.Godwin RufusS.J.,** Principal, St. Xavier‘s College (Autonomous), Palayamkottai, **Dr. A. Lordusamy**, Deputy Principal (Shift-II) and **Dr. S. Chidambaranathan** and **Mrs. A. DhanaPraveena**, Vice Principals (Shift-II) for having permitted me to carry out this project work.

No one grows tired to thank, **Dr. S. Saraswathi**, M.C.A., M.Phil., Ph.D., Head of the Department, Department of Computer Applications, who has been a source of inspiration, all time motivator and played a major role in the conduct of my project work.

I wish to express my deep sense of gratitude to my Internal Guide and the Project Co-ordinator**, Dr. S. Chidambaranathan, M.Sc., M.C.A., M.Phil., Ph.D.,** for the guidance and useful suggestions, which helped me in completing the project work on time.

Words are inadequate in offering my thanks to all the **faculty members** of the Department of Computer Applications, for their encouragement and cooperation in carrying out the project work.

Finally ,yet importantly, I would like to express my heart felt thanks to my beloved **Parents** for their blessings and wishes for the successful completion of this project.

23PCA528

S SUDHARSAN

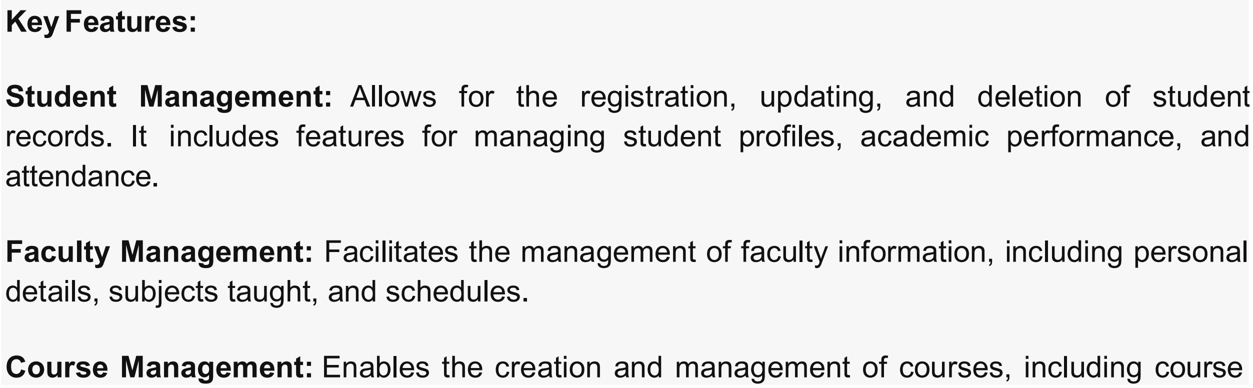
ABSTRACT

The University Management System (UMS) is a comprehensive software solution designed to streamline and automate the administrative processes of a University. Developed in Java, UMS provides a robust and scalable platform for managing various aspects of University operations, including faculty management, course management, department management, finance management, and HR management.

The University Management System in Java aims to enhance the efficiency and productivity of university operations, improve decision-making, and provide a better experience for admins.

Technical Specifications:

Programming Language: Java
Framework: Swing
Database: MySQL
User Interface: Web-based



Key Features:

Student Management: Allows for the registration, updating, and deletion of student records. It includes features for managing student profiles, academic performance, and attendance.

Faculty Management: Facilitates the management of faculty information, including personal details, subjects taught, and schedules.

Course Management: Enables the creation and management of courses, including course

TABLE OF CONTENTS

INDEX

INTRODUCTION

1. INTRODUCTION

Creating a University Management System using Java AWT (Abstract Window Toolkit),Swing, and MySQL involves several steps. Below is a high-level overview of how to structure and implement this system. The objective of the University Management System (UMS) project is to create a desktop application that facilitates the efficient management of University-related information. This system will allow users to manage student records, faculty details, and course information. By leveraging Java's AWT (Abstract Window Toolkit) and Swing for the graphical user interface (GUI) and MySQL for the database management, this project aims to provide a comprehensive solution for academic institutions.

System Requirements:

- **Java Development Kit (JDK):** Required for compiling and running Java applications.
- **MySQL Database Server:** For storing and managing data.
- **JDBC (Java Database Connectivity):** To connect Java applications with MySQL.

The University Management System project serves as an excellent exercise in applying Java, Swing, and MySQL to build a practical application. It demonstrates how to integrate various technologies to create a functional and user-friendly desktop application for managing University data. By following the outlined steps and continuously refining the application, you will gain valuable experience in software development and database management.

SYSTEM STUDY

2. SYSTEMSTUDY

2.1 EXISTING SYSTEM

The University currently relies on a manual system for managing student and faculty records. This traditional approach involves using paper forms and physical registers, which has several drawbacks:

Tedious Form Filling: Students are required to manually fill out forms for various processes, which is cumbersome and prone to errors. This manual entry not only increases the workload but also makes it difficult to track and manage records efficiently.

Inefficiency in Data Management: Maintaining records manually is highly time-consuming. The sheer volume of data—encompassing students, faculty, courses, and more—requires extensive effort to keep updated and organized.

Difficulty in Handling Frequent Updates: With frequent updates to records, maintaining them in physical registers becomes challenging. This manual updating process is prone to errors and delays, leading to potential discrepancies in the data.

Complicated Data Retrieval: The process of retrieving specific information from physical records is slow and inefficient. Searching through numerous registers to find relevant data consumes valuable time and resources.

Data Redundancy: Manual records often lead to data redundancy, where the same information may be entered multiple times across different registers. This redundancy not only wastes space but also creates inconsistencies in data.

2.2 PROPOSED SYSTEM

To address these issues, we propose developing a University Management System (UMS) using Java and MySQL. This computerized solution will streamline the management of student and faculty information, significantly improving efficiency and accuracy.

Advantages of the Proposed System:

Automated Data Entry and Management: The UMS will use Java Swing for creating an intuitive graphical user interface, allowing for easy data entry and management. This automation will eliminate the need for manual form filling and reduce the likelihood of errors.

Centralized Data Storage: MySQL will serve as the backend database, providing a centralized and organized repository for all records. This will simplify data management and ensure consistency across the system.

Efficient Data Handling: The system will handle frequent updates seamlessly, ensuring that records are always up-to-date. This will minimize the time and effort required for data maintenance.

Quick Data Retrieval: With a well-designed database structure, retrieving information will be fast and efficient. Users can quickly search for and access the data they need without sifting through physical records.

Reduction of Redundancy: By integrating data management into a single digital platform, the UMS will eliminate redundancy and ensure that each piece of information is stored only once. This will lead to more accurate and reliable data.

2.3 PROBLEMDEFINITIONANDPROJECTDESCRIPTION

The objective of the University Management System project is to develop a comprehensive desktop application that automates the management of University-related information. By leveraging Java for the application development and MySQL for database management, the UMS aims to enhance the efficiency and accuracy of handling student, faculty, and course records.

- Admin
- Faculty
- User(Students)

ADMIN MODULE

- **Total Students Registered:** View the total number of students currently registered in the system.
- **Total Faculty Members Registered:** See the total number of faculty members listed in the system.
- **Total Courses Offered:** View the total number of courses available in the University.

FACULTY MODULE

Total Number of Faculty Members: View the total number of faculty members currently registered in the system.

Total New Faculty Members: Display the number of new faculty members added recently.

Total Active Faculty Members: Show the number of faculty members currently active and involved in teaching or administrative roles.

Total Inactive Faculty Members: Show the number of faculty members who are not currently active (e.g., on leave, retired).

Assign Courses:

Functionality: Allocate specific courses to faculty members based on their expertise and availability.

Fields:

- **Course:** Course to be assigned.
- **Faculty Member:** Faculty member who will teach the course.
- **Semester:** Semester during which the course will be taught.

USER MODULE

Functionality: Allows users to view detailed information about individual students.

Features:

Search and Filter: Search for specific students by name, ID, department, or year of study. Filter the list based on criteria such as department or year.

Detailed View: View complete details including academic records, enrolled courses, and contact information.

E**nroll in Courses:**

Functionality: Allows students to enroll in courses for the current or upcoming semester.

Fields:

Course: Course to be enrolled in.

Semester: Semester during which the course will be taken.

View Enrolled Courses:

Functionality: View a list of courses that the student is currently enrolled in.

Functionality: Allows students and administrators to view academic progress, including grades and performance in courses.

Features:

Grades: View grades received in individual courses.

GPA: Calculate and display the Grade Point Average (GPA).

SYSTEM ANALYSIS

3. SYSTEMANALYSIS

3.1 REQUIREMENTS

SPECIFICATIONHARDWAREREQUI

REMENTS

Processor	:	AMD
Ram	:	4GB (3.89GB usable)
Hard disk	:	40 GB
Monitor	:	15‘‘mVGAmonitor
Keyboard	:	104keysKeyboard
Mouse	:	Optical mouse

SOFTWAREREQUIREMENTS

Operating System	:	64-bit OS
Local Host Server	:	Net Beans
Database Connectivity	:	Work Bench
Front End	:	JAVA
Back End	:	MYSQL

3.2 FEASIBILITYSTUDY

A feasibility analysis is conducted to decide if the solution considered to meet the criteria is feasible and workable in the software. During the feasibility study, information such as resource availability, cost estimates for software production, advantages of the software to the enterprise after its development, and cost to be expended on its maintenance is determined. The feasibility study aims to ascertain why developing software is appealing to users, adaptable to change, and compliant with applicable requirements. The system has been tested for feasibility in the following points:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

TECHNICAL FEASIBILITY

Technical feasibility evaluates the available infrastructure (such as hardware and software) andtechnologiesneededtomeettheuserneedsofsoftwareundertimeandbudgetconstraints.The following are the activities often performed by technical feasibility.

- ✓ Examines whether there are technical guarantees of accuracy, reliability, ease of access and data security.
- ✓ Determineswhethertheapplicationinfrastructureiswell-established.
- ✓ Ensureswhethertheproposedsystemprovidesadequateresponsetoinquiries,regardlessof the number or location of users

The current system developed is technically feasible as it provides the technical guarantee of accuracy, reliability, security and easy access to the users.

OPERATIONAL FEASIBILITY

The proposed system is beneficial only if it can be turned out into information system which will meet the operating requirements of the organization. The extent to which the required software completes a sequence of steps to address challenges and requirements of the developer and users respectively is measured by operational viability. The following are the operations carried out by operational feasibility:

- ✓ Determineswhethersufficientsupportfortheorganizationisprovidedfromtheusers.
- ✓ Ensures proper working of the system if it is being developed and implemented.
- ✓ Checkswhethertherewillbeanyresistancefromtheusersthatwillruinthepossiblebenefi -ts of the application

This University Management System would en sure the optimal utilization of computer resources and would help in the improvement of performance status.

ECONOMICFEASIBILITY

A system can be developed technically and that will be used if installed must still be a goodinvestmentfortheorganization.Economicfeasibilityneedstoconsidertheexpensesmadeonpurchasing, suchashardwarepurchasingandrequiredactivitiestocarryoutsoftwaredevelopment.Itis also necessary to consider the benefits that can be achieved by developing the software. Software is economically feasible when it focuses on the issues listed below.

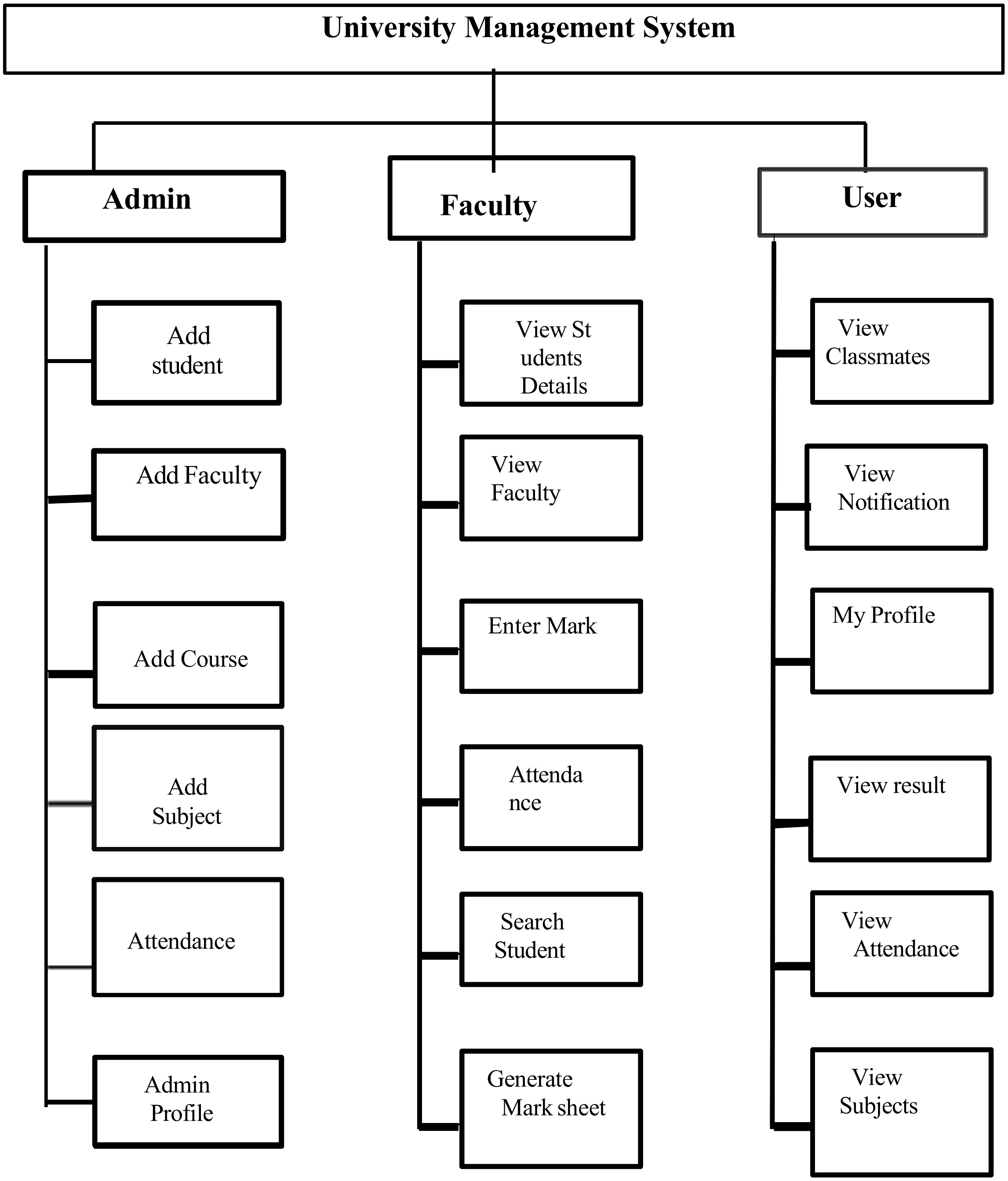
- ✓ Expense in curred on software development for achieving long- term gains for an organization.
- ✓ Expenses required to conduct elicitation and requirements analysis
- ✓ Hardware and software cost , development team, and training cost.

This system is economically feasible .Since this system is developed using the existing resources and technologies, there is nominal expenditure which ensures the economic feasibility of the system.

SYSTEM DESIGN

4. SYSTEMDESIGN

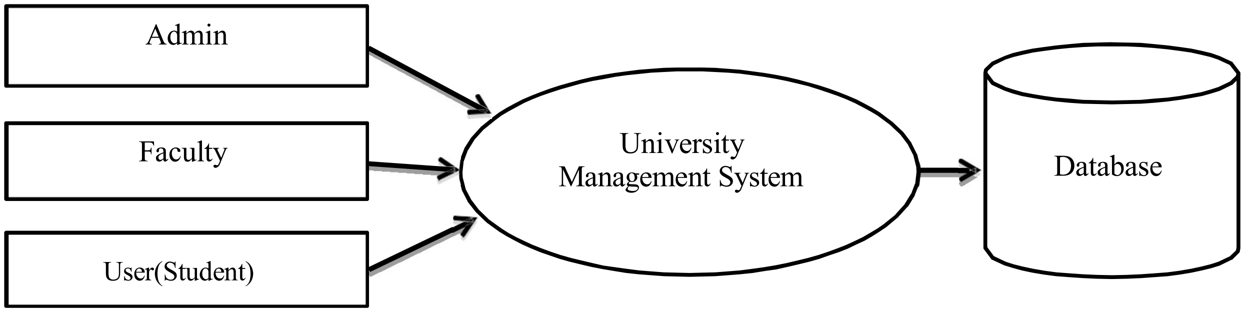
4.1 ARCHITECTURALDESIGN



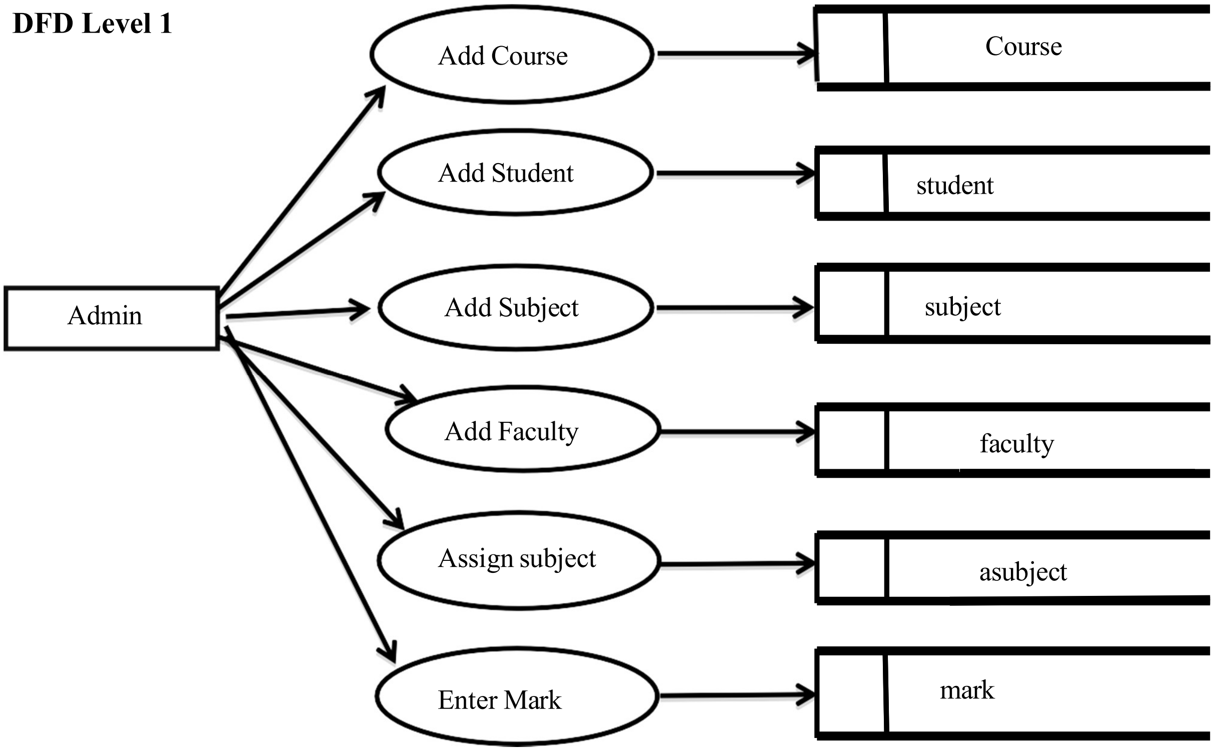
4.2 DATA FLOW

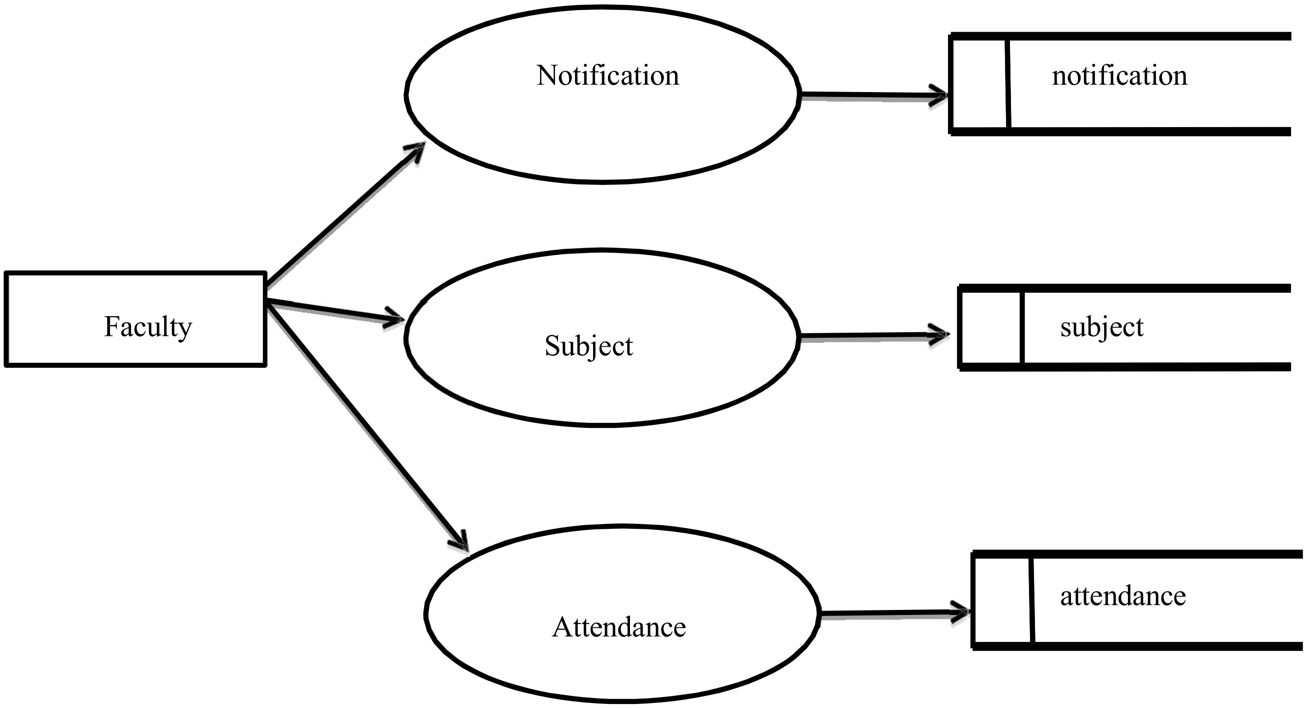
DIAGRAM

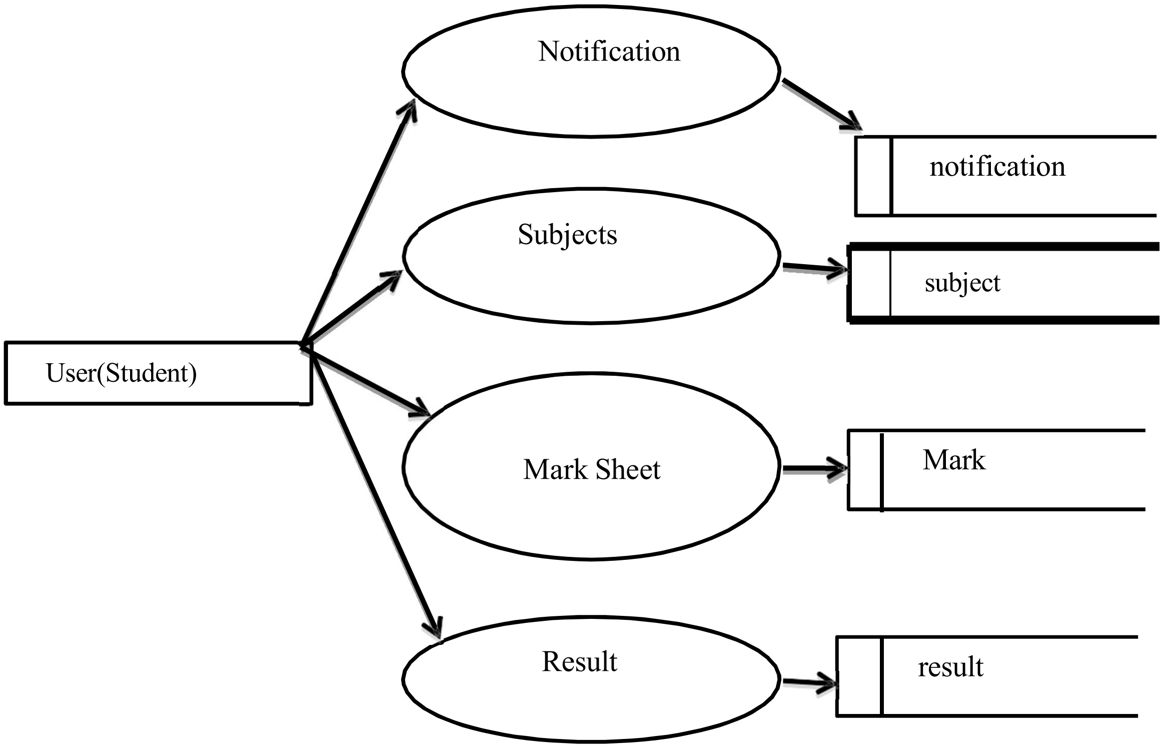
DFD Level 0



DFD Level 1




Notification
notification
Faculty
Subject
subject
Attendance
attendance


Notification
notification
Subjects
subject
User(Student)
Mark Sheet
Mark
Result
result

4.3 DATADICTIONARY

Table Name: admin

Purpose: To Store the Admin Details

S. No.	Field Name	Data type	Size	Constraint	Description
1	Universityname	varchar	11	Not Null	Universityname
2	address	Varchar	25	NotNull	Universityaddress
3	emailid	Varchar	25	NotNull	Universityemail
4	contactnumber	Varchar	25	NotNull	Universitycontactnumber
5	website	Varchar	8	NotNull	Universitywebsite
6	Lastlogin	Varchar	30	NotNull	lastlogin
7	password	Varchar	25	NotNull	password
8	facebook	Varchar	15	NotNull	Universityfbid
9	instagram	int	10	NotNull	Universityinstagramid
10	twitter	Varchar	10	NotNull	Universitytwitterid
11	linkedin	Varchar	10	NotNull	Universitylinkedin id
12	logo	Varchar	15	NotNull	Universitylogo
13	activestatus	tinyint	4	Default 0	Active status

Table Name: attendance

Purpose : To store the student attendance

S. No.	Field Name	Data type	Size	Constraint	Description
1	subjectcode	varchar	10	Default Null	scode
2	date	Varchar	30	NotNull	Date
3	rollnumber	Bigint	4	NotNull	Rollnumber
4	present	Tinyint	4	Default 0	Present/absent
5	coursecode	varchar	20	Default	Ccode
6	semoryear	Int	11	Not Null	Sem/year

Table Name: course

Purpose: To store the course details

S. No.	Field Name	Data type	Size	Constraint	Description
1	Sr_no	Int	10	PrimaryKey	Serial no
2	Coursecode	Varchar	50	NotNull	Course Code
3	Coursename	Varchar	30	NotNull	Name of the Course
4	Semoryear	Varchar	10	NotNull	Semoryear
5	Totalsemoryear	Int	20	NotNull	Totalsemoryear

Table Name: faculties

Purpose: To store the Faculties Detail

S. No.	Field Name	Data type	Size	Constraint	Description
1	facultyid	int	11	PrimaryKey	Faculty Id
2	facultyname	Varchar	20	NotNull	Name
3	state	Varchar	11	NotNull	State
4	city	Varchar	11	NotNull	City
5	emailid	Varchar	19	NotNull	Email Id
6	contactnumber	Varchar	20	NotNull	Contact
7	qualification	Varchar	50	NotNull	Qualification
8	experience	Varchar	10	NotNull	Experience
9	birthdate	Varchar	50	NotNull	Date of birth
10	gender	Varchar	25	NotNull	Gender
11	profileepic	Longblob	34	NotNull	Picture
12	courcecode	Varchar	44	NotNull	Cource code
13	semoryear	Int	11	NotNull	Year
14	Subject	Varchar	22	NotNull	Subject
15	Position	Varchar	11	NotNull	Position
16	Sr_no	Int	33	NotNull	Serial no
17	Lastlogin	Varchar	22	NotNull	Lastlogin
18	Password	Varchar	30	NotNull	Password

19	Activestatus	Tinyint	5	NotNull	Activestatus
20	joineddate	Varchar	50	NotNull	Joining date

Table Name: marks
Purpose: To store the student Marks detail

S. No.	Field Name	Data type	Size	Constraint	Description
1	Coursecode	Varchar	10	Not Null	Courcecode
2	Semoryear	Int	50	NotNull	Semoryear
3	Subjectcode	Varchar	30	NotNull	Subjectcode
4	Subjectname	Varchar	10	NotNull	Subjectname
5	Rollnumber	Bigint	20	Primary key	Student rollnumber
6	Theory_mark	Int	11	Not Null	Mark
7	Practical_mark	int	11	Default 0	Mark

Table Name: notification
Purpose: To display the Notification for students

S. No.	Field Name	Data type	Size	Constraint	Description
1	Sr_no	Int	10	Not Null	Serial no
2	userprofile	Varchar	50	Default Null	Picture
3	Coursecode	Varchar	30	NotNull	coursecode
4	Semoryear	Int	10	NotNull	Semor year
5	Userid	Varchar	20	Primary key	User Id
6	Title	Varchar	20	Not Null	Title
7	Message	Varchar	20	Default 0	Message
8	Time	Varchar	22	Default 0	Time
9	Readby	Longtext	33	Default Null	Readby

Table Name: result

Purpose: To store the student result

S. No.	Field Name	Data type	Size	Constraint	Description
1	Coursecode	Varchar	50	PrimaryKey	Course code
2	Semoryear	Int	11	NotNull	Sem or year
3	Isdeclared	Tinyint	5	NotNull	Is declared

Table Name: student

Purpose : To store the student details

S .No.	Field Name	Data type	Size	Constraint	Description
1	Coursecode	Varchar	50	Not Null	Course code
2	Semoryear	Int	11	NotNull	Sem or year
3	Rollnumber	Bigint	5	NotNull	Rollno
4	Optionalsubject	Varchar	20	Default Null	Optional
5	Firstname	Varchar	30	Not Null	Name of student
6	Lastname	Varchar	22	Not Null	Lastname
7	Emailid	Varchar	33	Not Null	Mail id
8	Contactnumber	Varchar	44	Not Null	Contact
9	Dateofbirth	Varchar	20	Not Null	Date of birth
10	Gender	Varchar	20	Not Null	Gender
11	State	Varchar	30	Default Null	State
12	City	Varchar	20	Default Null	City
13	Fathername	Varchar	22	Default Null	Fathername
14	Father_Occupation	Varchar	25	Default Null	Occupation
15	Mothername	Varchar	27	Default Null	Mothername
16	Mother_Occupati On	Varchar	30	Default Null	Occupation
17	Profilepic	Longblob	30	Default Null	Picture
18	Sr_no	Int	11	Default Null	Serial no

19	Lastlogin	Varchar	40	Default Null	Lastlogin
20	Userid	Varchar	30	Not Null	User id
21	Password	Varchar	25	Not Null	Password
22	Activestatus	Tinyint	10	Default Null	Active status
23	Admissiondate	Varchar	20	Default Null	Admission date

Table Name: users

Purpose: To store the user details

S. No.	Field Name	Data type	Size	Constraint	Description
1	Sr_no	Int	11	Not Null	Serial no
2	Coursecode	Int	11	NotNull	Course code
3	Semoryear	Tinyint	5	NotNull	Sem or year
4	User id	Varchar	20	Not Null	User id
5	Login time	Tinyint	6	Default Null	Login time
6	User profile	longblob	20	Default Null	Picture

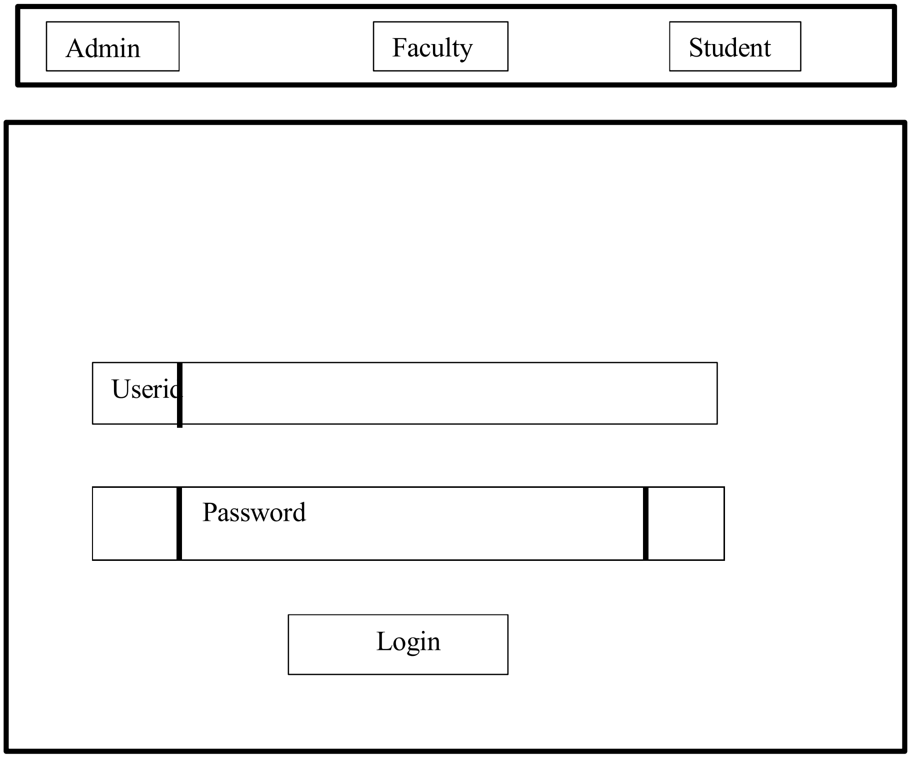
Table Name: Subjects

Purpose: To store the Subject details

S. No.	Field Name	Data type	Size	Constraint	Description
1	Subjectcode	Varchar	20	Not Null	Subject code
2	Subjectname	Varchar	25	NotNull	Subject name
3	Coursecode	Varchar	20	NotNull	Course code
4	Semoryear	Int	11	Not Null	Sem or year
5	Subjecttype	Varchar	30	Not Null	Subject type
6	Theorymark	Int	11	Default Null	Mark
7	Practicalmark	Int	11	Default Null	Mark

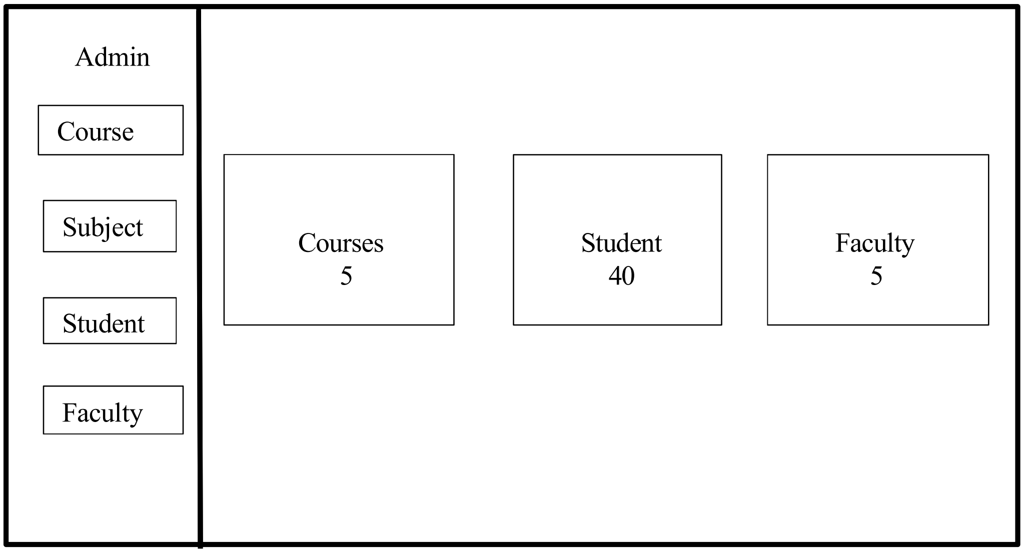
4.4 USERINTERFACEDESIGN

Login Page

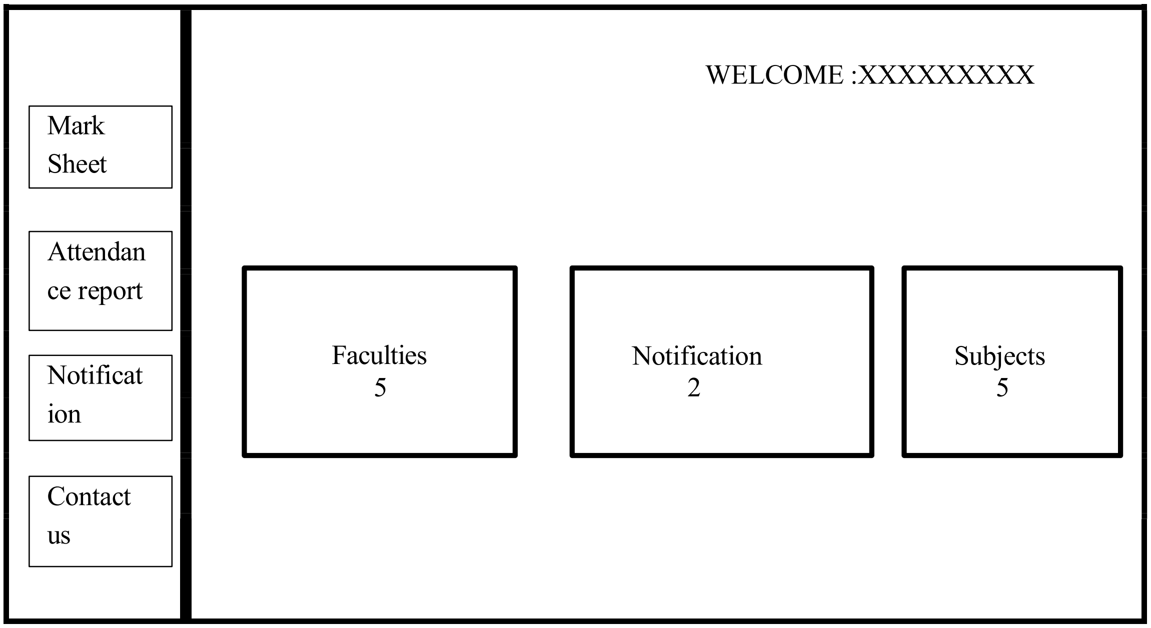


Dashboard

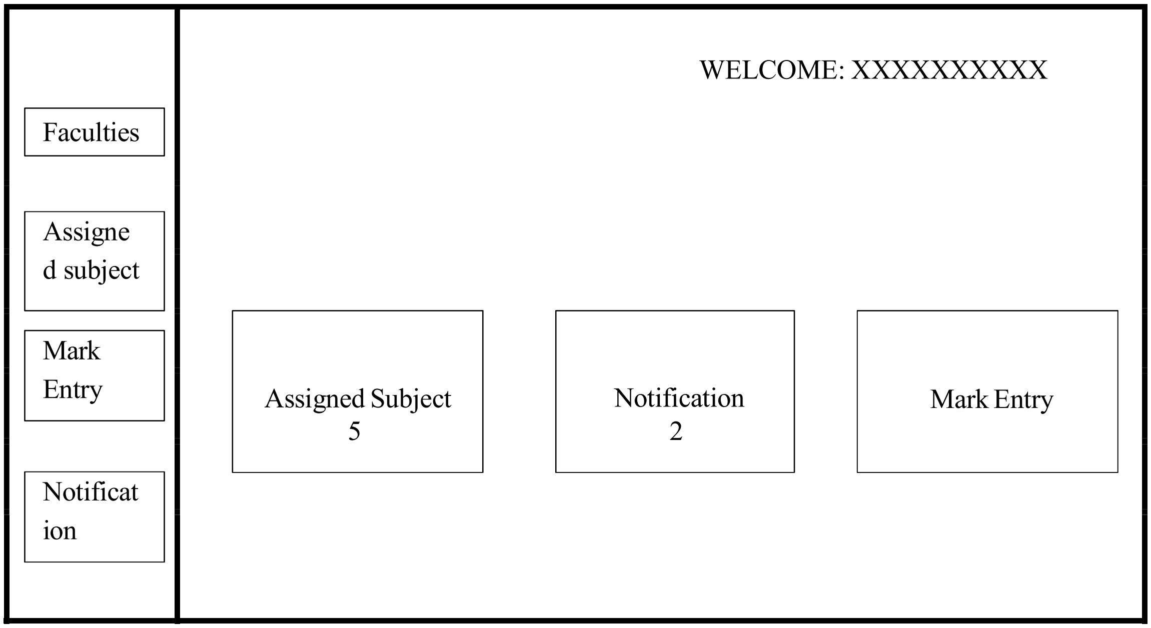
Admin DASHBOARD



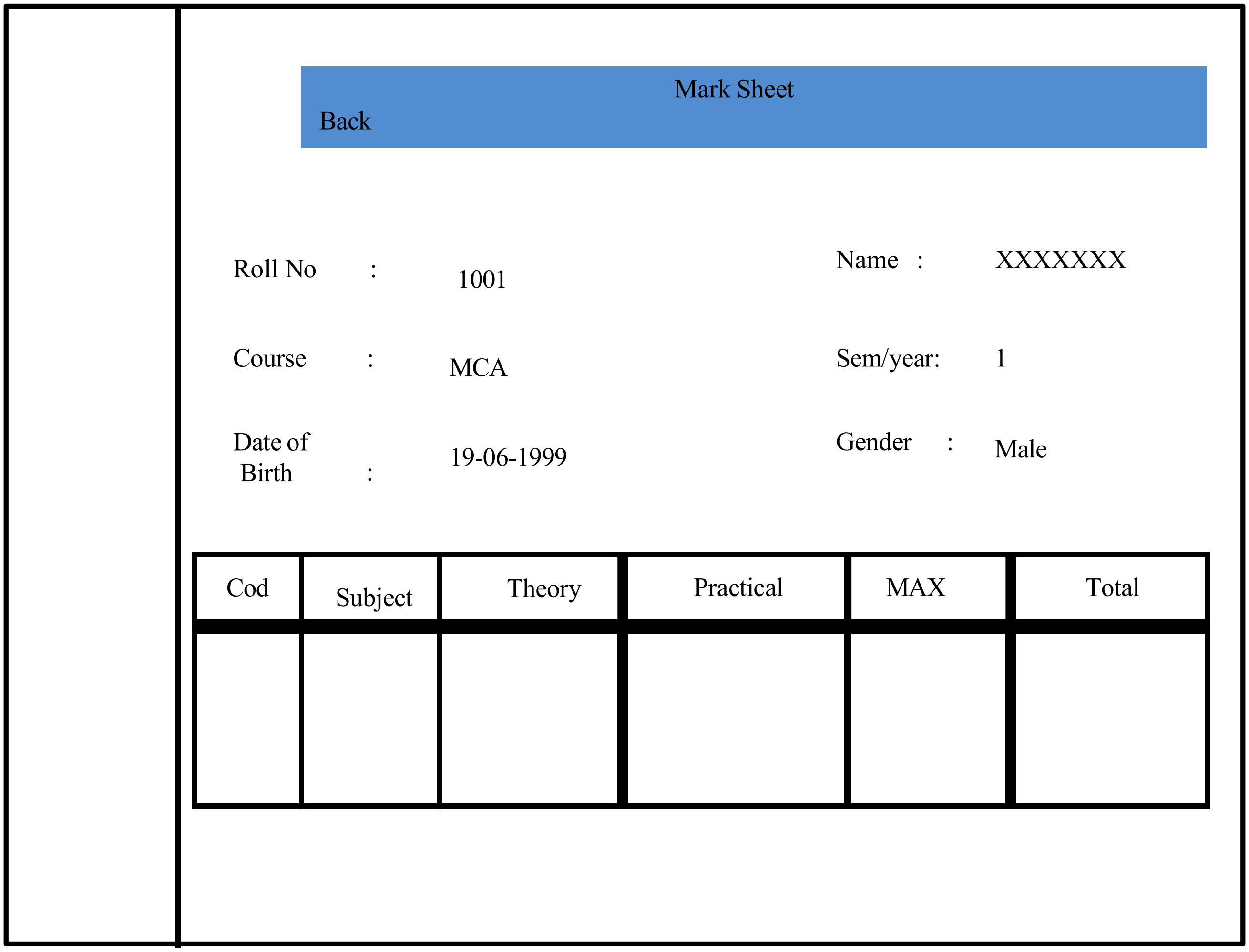
Student Dashboard



Faculty Page:



Mark Sheet Page:



Mark Sheet

Roll No : 1001

Name : XXXXXXX

Course : MCA

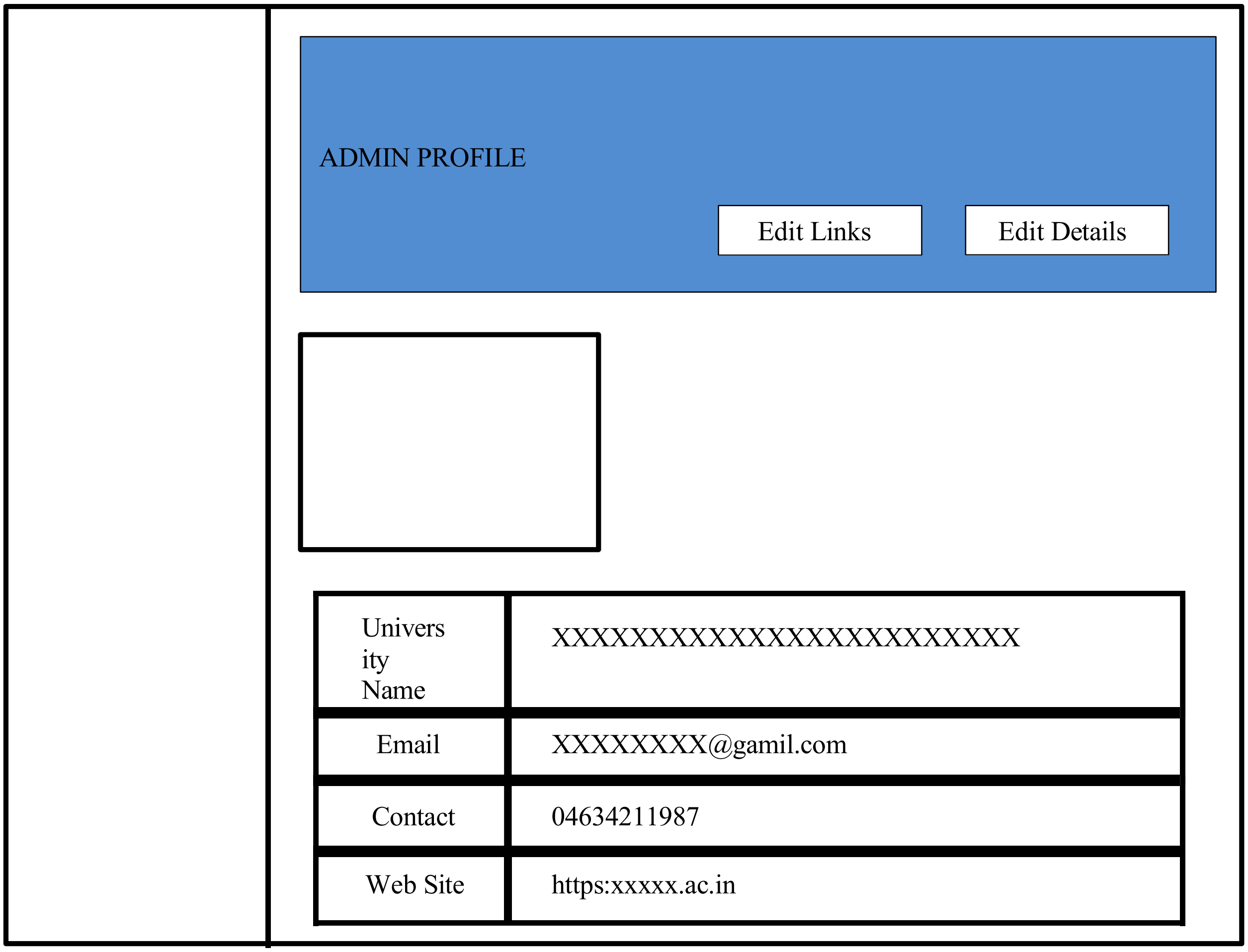
Sem/year: 1

Date of Birth : 19-06-1999

Gender : Male

Cod	Subject	Theory	Practical	MAX	Total

Admin Profile:



University Name	XXXXXXXXXXXXXXXXXXXXXXXXX
Email	XXXXXXXX@gamil.com
Contact	04634211987
Web Site	https:xxxxx.ac.in

4.5 NORMALIZATION

Normalization is the process of organizing the data in the database. Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization divides the larger table into the smaller table and links them using relationship. The normal form is used to reduce redundancy from the database table.

First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4rules:

1. It should only have single(atomic)valued attributes/columns
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names
4. And the order in which data is stored, does not matter

Second Normal Form (2NF)

For a table to be in the Second Normal Form,

1. It should be in the First Normal form
2. And, it should not have Partial Dependency. Partial Dependency occurs when anon- prime attribute is functionally dependent on part of a candidate key

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

Boyce and Codd Normal Form (BCNF)

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

Rmustbein3rdNormalForm
For each functional dependency (X→Y),X should be a super Key.

Fourth Normal Form (4NF)

A table is said to be in the Fourth Normal Form when,

1. It is in the Boyce-Codd Normal Form.
2. And , it doesn't have Multi-Valued Dependency.