KIET ALUMNI ASSOCIATION (KAA)

A PROJECT REPORT
for
Mini Project (KCA353)
Session (2023-24)

Submitted By
Sharad Nirankari
(2200290140140)

Submitted in partial fulfillment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
Dr. Shashank Bhardwaj
(ASSOCIATE PROFESSOR)

Submitted to
DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad Uttar Pradesh-201206
(MAR 2024)

CERTIFICATE

Certified that **Sharad Nirankari (University Roll No.- 2200290140140)** have carried out the project work having "**KIET Alumni Association (KAA)**" for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)**, Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the studenthimself/herself and the contents of the project report do not form the basis for the award of any otherdegree to the candidate or to anybody else from this or any other University/Institution.

Date:

Sharad Nirankari
(2200290140140)

This is to certify that the above statement made by the candidate is correct to the best of myknowledge.

Dr. Shashank Bhardwaj
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Tripathi
Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

KIET ALUMNI ASSOCIATION (KAA)

SHARAD NIRANKARI

ABSTRACT

The purpose of KIET Alumni Association is to automate the existing manual system by the help of computerized equipment's and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work.

Thus, it will help the organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

ACKNOWLEDGEMENT

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

The KIET Alumni Association (KAA) stems from the recognition of the growing need for an efficient and modern Alumni Management System. As the KIET alumni community continues to expand, there arises a requirement for a centralized platform to foster engagement, networking, and collaboration among former students.

1.2 PROJECT OVERVIEW

The KIET Alumni Association (KAA) project aims to establish a robust Alumni Management System that serves as a dynamic bridge connecting graduates with their alma mater. This platform will facilitate seamless communication, provide career support, and celebrate the achievements of KIET alumni.

1.3 OBJECTIVE

The primary objective of the KAA project is to create a comprehensive Alumni Management System that fosters a sense of community among KIET graduates. The system will enable efficient communication, networking opportunities, and collaboration among alumni and the institution.

1.4 KEY FEATURES

Key features of the KIET Alumni Association (KAA) include:
- Alumni Profile Management
- Networking and Collaboration Platform
- Job and Internship Portal
- Event Management for Alumni Gatherings
- Alumni News and Achievements Section
- Integration with Social Media Platforms
- Donation and Fundraising Module

1.5 SCOPE OF THE PROJECT

The scope of the KAA project involves the development and implementation of the core features mentioned above. Initial integration with the existing university database for alumni information is included, with provisions for scalability. The project will also involve user training sessions and documentation. Customization options and additional features may be considered based on user feedback and evolving alumni needs.

CHAPTER 2

PROBLEM IDENTIFICATION & FEASIBILITY STUDY

2.1 PROBLEM IDENTIFICATION

The KIET Alumni Association (KAA) project aims to address several key challenges, including limited alumni engagement, inefficient communication channels, gaps in career support, disparate alumni information, coordination issues for alumni events, a lack of alumni recognition, underutilization of social media, insufficient data security measures, and a scarcity of financial support channels. By recognizing and systematically addressing these issues, the project seeks to create a comprehensive Alumni Management System that fosters a vibrant and connected community, facilitates effective communication, provides valuable career resources, and ensures the security and privacy of alumni data. This will ultimately contribute to building a strong, engaged, and supportive network of KIET alumni.

2.2 FEASIBILITY STUDY

After studying and analyzing all the existing and required functionalities of the system, the next task is to do the feasibility study for the project. The feasibility study includes consideration of all the possible ways to provide a solution to a given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily made based on the future upcoming requirements.

2.2.1 Technical Feasibility

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionalities to be provided in the system, as described in the System Requirement Specification (SRS) and checked if everything was possible using different types of front end and backend platform.

2.2.2 Operational Feasibility

Operational feasibility study is an assessment of whether a proposed project is feasible from an operational standpoint. It evaluates the practicality of implementing the project based on factors such as technology, personnel, infrastructure, and processes. This study aims to identify potential operational issues that may arise during the project's implementation and determine whether the project can be realistically executed within the available resources and constraints. The operational feasibility study is an essential step in

project planning, as it helps ensure that the project will be successful and sustainable over the long term.

2.2.3 Behavioral Feasibility

A behavioral feasibility study is an examination of the potential impacts that a project or initiative may have on human behavior. It assesses whether the project aligns with societal norms and values, and whether it is likely to be embraced by the community it is intended to serve. This type of study explores how the target audience may react to the project and what factors could influence their behavior. It also considers the feasibility of implementing the project, considering the attitudes, beliefs, and cultural values of those who will be affected. The goal of a behavioral feasibility study is to ensure that a project is not only financially feasible, but also socially acceptable and culturally appropriate.

2.2.4 Economic Feasibility

For the economic feasibility, Economic analysis or cost/benefits analysis is most frequently used technique the effectiveness of a proposed system. It is a procedure to determine the benefits and saving those that are expected from the proposes system and compare them with cost. If the benefits outweigh the costs, a decision is taken to design and implement the system. otherwise, further justification or alternative in proposed system will have to be made if it is to have a chance of being approved this is ongoing effort that improves in accuracy at each phase of a system life cycle.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 INTRODUCTION

3.1.1 Background

The Alumni Management System is designed to streamline and enhance the management of alumni data, events, and donations within the organization. It aims to provide a centralized platform for alumni engagement, improving communication, and facilitating efficient organization of events and fundraising activities.

3.1.2 Objectives

The primary objectives of the Alumni Management System include:
- Efficiently manage and update alumni profiles.
- Facilitate seamless event organization and attendance tracking.
- Streamline the process of receiving and tracking alumni donations.
- Ensure secure user` authentication and access control.
- Meet performance, usability, security, and reliability standards.

3.2 FUNCTIONAL REQUIREMENTS

3.2.1 Alumni Profiles
- The system shall allow alumni to create and update their profiles with details such as name, contact information, graduation year, and current employment status.
- Administrators shall have the ability to manage and update alumni profiles.

3.2.2 Gallery
- Admin users shall have the capability to upload images to the gallery section, contributing to shared memories.
- Admin should be able to delete images from the gallery for content control.
- The system shall display upcoming gallery images, allowing alumni to preview and engage with upcoming events.

3.2.3 Donation System
- Alumni should be able to make donations with details such as donation amount,

purpose, and date.

- The system shall provide administrators with tools to track and manage donations, generate donation reports, and acknowledge donors.

3.2.4 User Authentication and Authorization

- The system shall implement secure user authentication mechanisms.
- Users shall have different roles (alumni, administrator) with role-based access control.
- Passwords shall be securely stored using encryption.

3.3 NON-FUNCTIONAL REQUIREMENTS

3.3.1 Performance

- The system should respond to user requests within a maximum of 2 seconds.
- It should be able to handle a concurrent user load of at least 500 users.

3.3.2 Usability

- The user interface should be intuitive and easy to navigate.
- The system shall comply with accessibility standards (WCAG) to ensure usability for individuals with disabilities.

3.3.3 Security

- Data transmission shall be encrypted using HTTPS.
- The system shall implement protection against common security threats, including SQL injection and cross-site scripting (XSS).
- User sessions shall have an idle timeout of 15 minutes.

3.3.4 Reliability

- The system shall have a minimum uptime of 99.9%.
- In case of system failures, there should be a recovery mechanism to restore the system to a consistent state.

CHAPTER 4

PROJECT PLANNING AND SCHEDULING

4.1 PERT CHART

A PERT chart is a project management tools used to schedule, organize, and coordinate tasks within a project. PERT stands for Program Evaluation Review Technique. A PERT chart presents a graphic illustration of a project as network diagram consisting of numbered nodes (either circles or rectangles) representing events, or milestones in the project linked by labelled vectors (directional lines) representing tasks in the project.

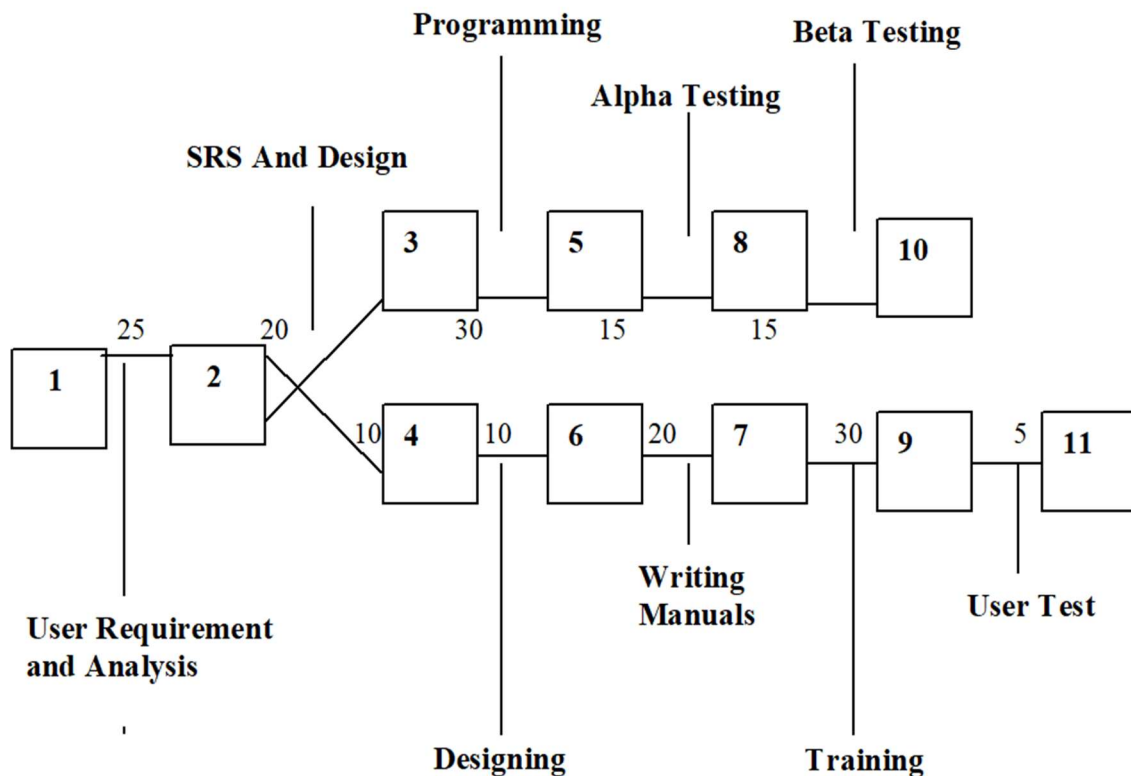The direction of the arrows on the lines indicates the sequence of tasks.

Fig 4.1 PERT Chart

CHAPTER 5

HARDWARE & SOFTWARE SPECIFICATION

5.1 HARDWARE SPECIFICATION

The Alumni Management System (KAA) will be developed and deployed on a hardware infrastructure that ensures optimal performance and reliability. The recommended hardware specifications are as follows:

5.1.1 Server
- Processor: Intel Core i5 or equivalent
- RAM: 8 GB or higher
- Storage: 256 GB SSD or higher

5.1.2 Database Server
- Processor: Intel Core i5 or equivalent
- RAM: 8 GB or higher
- Storage: 256 GB SSD or higher
- Network Interface: Gigabit Ethernet

5.1.3 Client Machines
- Processor: Intel Core i3 or equivalent
- RAM: 4 GB or higher
- Storage: 128 GB SSD or higher
- Network Interface: 100 Mbps Ethernet or Wi-Fi

5.2 SOFTWARE SPECIFICATION

The KAA will be developed using a combination of server-side and client-side technologies. The development and deployment environment will be facilitated by XAMPP, which provides a comprehensive stack for web application development. The software specifications include:

5.2.1 Server-Side Technologies
- Operating System: Windows Server 2016 or later
- Web Server: Apache 2.4
- Database Management System: MySQL 5.7 or later
- Server-Side Scripting Language: PHP 7.4 or later

5.2.2 Client-Side Technologies

- Web Browser: Latest versions of Chrome, Firefox, Safari, or Edge
- Client-Side Scripting: JavaScript, jQuery

5.2.3 Development Tools

- XAMPP: Version 8.0.9 or later for local development and testing
- Integrated Development Environment (IDE): Visual Studio Code or any preferred PHP IDE

5.2.4 Version Control

- Git: Version control for collaborative development

5.2.5 Security

- SSL/TLS: Ensure secure data transmission over the network.
- Firewall: Implement firewall rules to restrict unauthorized access
- Anti-malware Software: Regularly updated anti-malware software on server and client machines.

CHAPTER 6

CHOICE OF TOOLS & TECHNOLOGY

6.1 PHP

PHP (Hypertext Preprocessor) is a widely used server-side scripting language that is designed for web development. It is embedded within HTML code and executed on the server, producing dynamic content that is then sent to the client's web browser. PHP is renowned for its simplicity, versatility, and ease of integration with various databases, making it a cornerstone technology in modern web development.

Modern PHP embraces Object-Oriented Programming principles, allowing developers to organize code in a more modular and reusable manner. OOP in PHP involves the use of classes and objects, promoting better code organization, encapsulation, and the creation of scalable applications.

PHP has a vibrant ecosystem of frameworks that facilitate rapid and efficient web development. Frameworks like Laravel, Symfony, and CodeIgniter provide standardized architectures, pre-built modules, and tools that enhance developer productivity. These frameworks adhere to best practices, such as MVC (Model-View-Controller), promoting clean and maintainable code.

6.2 MYSQL

MySQL is a popular choice of database for use in web applications and is a central component of the widely used LAMP open-source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, and Perl/PHP/Python." Free-software-open-source projects that require a full-featured database management system often use MySQL.

6.3 DATA FLOW DIAGRAM

The data flow diagram shows the flow of data within any system. It is an important tool for the design phase of software engineering. Larry Constantine first developed it. It represents a graphical view of the flow of data. It's also known as BUBBLE CHART. The purpose of DFD is major transformation that will become in system design symbols used in DFD: -

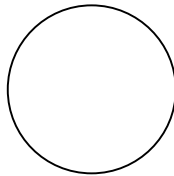In the DFD, four symbols are used, and they are as follows.
1. A square defines a source (originator) or destination of system data.

2. An arrow identifies data flow-data in motion. It is 2a pipeline through which information flows.

3. A circle or a "bubble "(Some people use an oval bubble) represents a process that transfers informing data flows into outgoing data flows.

4. An open rectangle is a data store-data at rest, or a temporary repository of data.
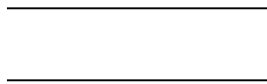
6.4 CONTEXT LEVEL DIAGRAM

This level shows the overall context of the system and its operating environment and shows the whole system as just one process. Alumni Management System is shown as one process in the context diagram, which is also known as zero level DFD, shown below. The context diagram plays an important role in understanding the system and determining the boundaries. The main process can be broken into sub-processes and the system can be studied with more details this is where 1st level DFD comes into play.
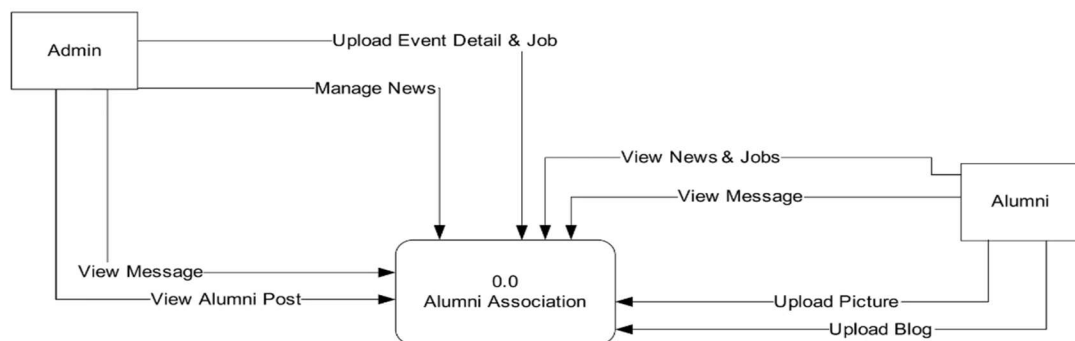
Fig 6.1 Zero Level DFD
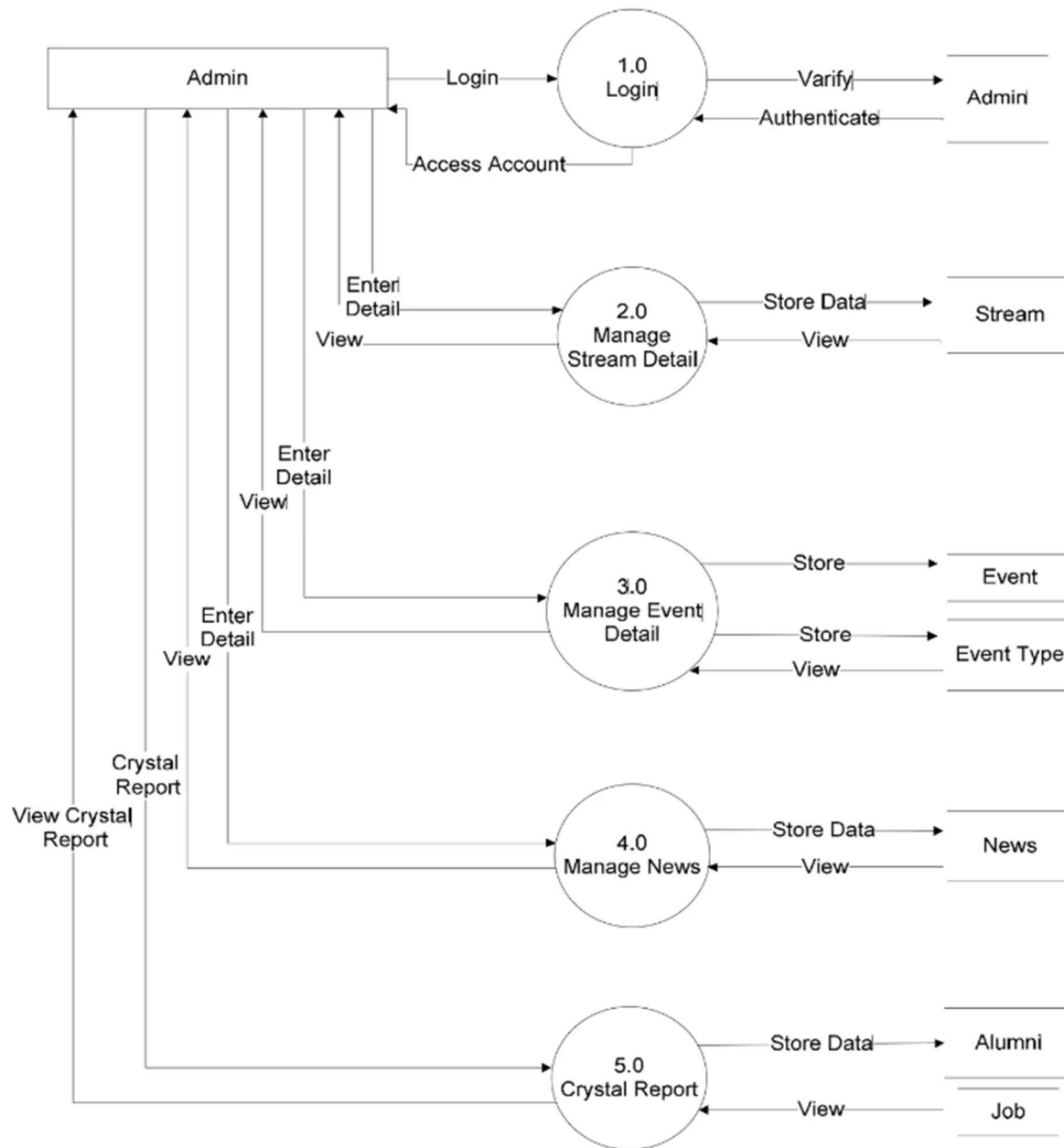
Fig 6.2 1ˢᵗ Level DFD

CHAPTER 7

ER DIAGRAM

7.1 ENTITY-RELATIONSHIP MODEL

Entities are represented by the rectangle shape. The entity will be our database table of Alumni Management System later.

Attributes are represented by the oval shape. This will be the columns or fields of each table in the Alumni Management System. Relationship is represented by diamond shape. This will determine the relationships among entities. This is usually in the form of primary key to foreignkey connection.

We will follow the 3 basic rules in creating the ER Diagram.
1. Identify all the entities.
2. Identify the relationship between entities.
3. Add meaningful attributes to our entities.

7.1.1 Entities

- admin_profile
- admin_users
- ambassadors
- award
- banner
- categories
- committee
- contact_us
- gallery
- gallery_images
- massage
- notable
- sub_categories
- users

7.1.2 Identify the relationship between entities

After we have specified our entities, it is time now to connect or establish a relationship among the entities.

- **admin_profile - admin_users Relationship:**

  An admin_profile is managed by an admin_user, indicating a one-to-one relationship. Each admin_user can be associated with one admin_profile, and vice versa.

- **admin_users - committee Relationship:**

  An admin_user can be a part of the committee, suggesting a one-to-many relationship. One admin_user can be associated with multiple committees, but each committee is managed by only one admin_user.

- **users - massage Relationship:**

  Users can have massages (messages), indicating a one-to-many relationship. A user can send or receive multiple messages, but each message is associated with one user.

- **admin - gallery Relationship:**

  Users can contribute to the gallery, representing a one-to-many relationship. Each user can submit multiple images to the gallery, but each image is associated with one user.

- **users - contact_us Relationship:**

  Users may initiate contact_us requests, suggesting a one-to-many relationship. Each user can submit multiple contact_us requests, but each request is associated with one user.

- **gallery - gallery_images Relationship:**

  The gallery can have multiple images, indicating a one-to-many relationship. Each image is associated with one gallery, but one gallery may contain multiple images.

- **categories - sub_categories Relationship:**

  Categories can have sub-categories, implying a one-to-many relationship. Each category can contain multiple sub-categories, but each sub-category is associated with only one category.

7.1.3 Add Meaningful Attributes to Entities

admin_profile:
    Attributes: profile_id, user_id, full_name, email, phone, position.
admin_users:
    Attributes: user_id, username, password, role.
award:
    Attributes: award_id, award_name, description.
banner:
    Attributes: banner_id, image_url, ambassador_id.
categories:
    Attributes: category_id, category_name.
committee:
    Attributes: committee_id, committee_name.
contact_us:
    Attributes: request_id, user_id, subject, message, timestamp.
gallery:
    Attributes: gallery_id, user_id, category_id, timestamp.
gallery_images:
    Attributes: image_id, gallery_id, image_url, description.
notable:
    Attributes: notable_id, user_id, reason, timestamp.
sub_categories:
    Attributes: sub_category_id, category_id, sub_category_name.
users:
    Attributes: user_id, username, password, full_name, email, phone, registration_date

admin_users

user_id (PK)
username
password
role

admin_profile

user_id (FK)
full_name
email
phone

committee

committee_id (PK)
committee_name

users

user_id (PK)
username
password
full_name
email
phone
reg_date

message

message_id (PK)
user_id (FK)
message
timestamp

gallery

gallery_id (PK)
user_id (FK)
category_id (FK)
timestamp

gallery_images

image_id (PK)
gallery_id (FK)
image_url
description

categories

category_id (PK)
category_name

sub_categories

sub_category_id
category_id (FK)
sub_category_name

Fig 7.1 Database Design

7.2 CLASS DIAGRAM
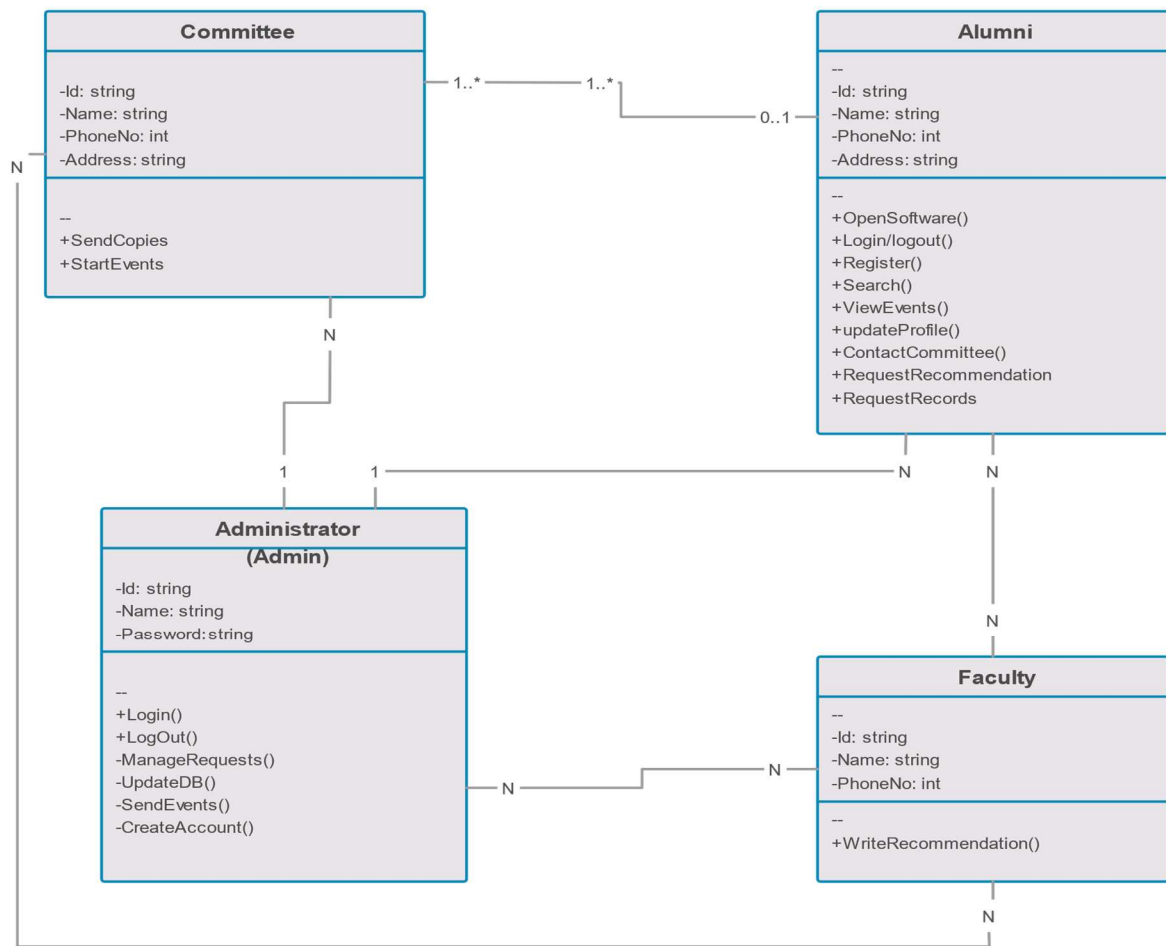
Fig 7.2 Class Diagram

CHAPTER 8

DATABASE

8.1 DATABASE: KAA

The database named "KAA" encompasses tables that store information related to administrators, users, awards, banners, categories, committees, messages, gallery, notable events, sub-categories, and user interactions.

8.1.1 Table: admin_users

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	admin_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	admin_name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	admin_email	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	admin_pass	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
5	register_time	timestamp(6)			No	current_timestamp(6)		ON UPDATE CURRENT_TIMESTAMP(6)	Change Drop More

Table 8.1 admin_users table

8.1.2 Table: admin_profile

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
3	password	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
4	role	int(11)			No	None			Change Drop More
5	email	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
6	mobile	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
7	status	int(11)			No	None			Change Drop More

Table 8.2 admin_profile table

8.1.3 Table: committee

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	committee_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	committee_name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	committee_post	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	committee_desc	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
5	committee_image	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
6	member	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
7	status	int(11)			No	None			Change Drop More
8	order_no	int(255)			No	None			Change Drop More

Table 8.3 committee table

8.1.4 Table: massage

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	msg_id 🔑	int(11)			No	None		AUTO_INCREMENT	Change	Drop	More
2	msg_title	varchar(1000)	utf8mb4_general_ci		No	None			Change	Drop	More
3	msg_description	varchar(2000)	utf8mb4_general_ci		No	None			Change	Drop	More
4	msg_dp	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
5	msg_status	int(1)			No	None			Change	Drop	More

Table 8.4 massage table

8.1.5 Table: gallery

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	id 🔑	int(255)			No	None		AUTO_INCREMENT	Change	Drop	More
2	categories_id	int(11)			No	None			Change	Drop	More
3	sub_categories_id	int(11)			No	None			Change	Drop	More
4	name	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
5	image	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
6	status	int(255)			No	None			Change	Drop	More

Table 8.5 gallery table

8.1.6 Table: gallery_images

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	id 🔑	int(11)			No	None		AUTO_INCREMENT	Change	Drop	More
2	gallery_id	int(11)			No	None			Change	Drop	More
3	gallery_images	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More

Table 8.6 gallery_image table

8.1.7 Table: categories

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	id 🔑	int(11)			No	None		AUTO_INCREMENT	Change	Drop	More
2	categories	varchar(255)	latin1_swedish_ci		No	None			Change	Drop	More
3	status	tinyint(4)			No	None			Change	Drop	More

Table 8.7 categories table

8.1.8 Table: sub_categories

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	id 🔑	int(11)			No	None		AUTO_INCREMENT	Change	Drop	More
2	categories_id	int(11)			No	None			Change	Drop	More
3	sub_categories	varchar(100)	utf8mb4_general_ci		No	None			Change	Drop	More
4	status	int(11)			No	None			Change	Drop	More

Table 8.8 sub_categories table

8.1.9 Table: contact_us

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	id 🔑	int(11)			No	None		AUTO_INCREMENT	Change	Drop	More
2	name	varchar(255)	latin1_swedish_ci		No	None			Change	Drop	More
3	email	varchar(75)	latin1_swedish_ci		No	None			Change	Drop	More
4	mobile	varchar(15)	latin1_swedish_ci		No	None			Change	Drop	More
5	comment	text	latin1_swedish_ci		No	None			Change	Drop	More
6	added_on	datetime			No	None			Change	Drop	More

Table 8.9 contact_us table

8.1.10 Table: notable

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	id 🔑	int(11)			No	None		AUTO_INCREMENT	Change	Drop	More
2	alumni_name	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
3	alumni_branch	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
4	alumni_desc	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
5	alumni_image	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
6	alumni_category	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
7	status	int(11)			No	None			Change	Drop	More
8	order_no	int(11)			No	None			Change	Drop	More

Table 8.10 notable table

8.1.11 Table: award

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	id 🔑	int(11)			No	None		AUTO_INCREMENT	Change	Drop	More
2	name	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
3	achievement	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
4	course	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
5	branch	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
6	passout	int(11)			No	None			Change	Drop	More
7	affiliation	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
8	awardyear	int(11)			No	None			Change	Drop	More
9	orderno	int(11)			No	None			Change	Drop	More

Table 8.11 award table

8.1.12 Table: banner

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	id 🔑	int(11)			No	None		AUTO_INCREMENT	Change	Drop	More
2	heading1	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
3	heading2	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
4	btn_txt	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
5	btn_link	varchar(55)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
6	image	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
7	order_no	int(11)			No	None			Change	Drop	More
8	status	int(11)			No	None			Change	Drop	More
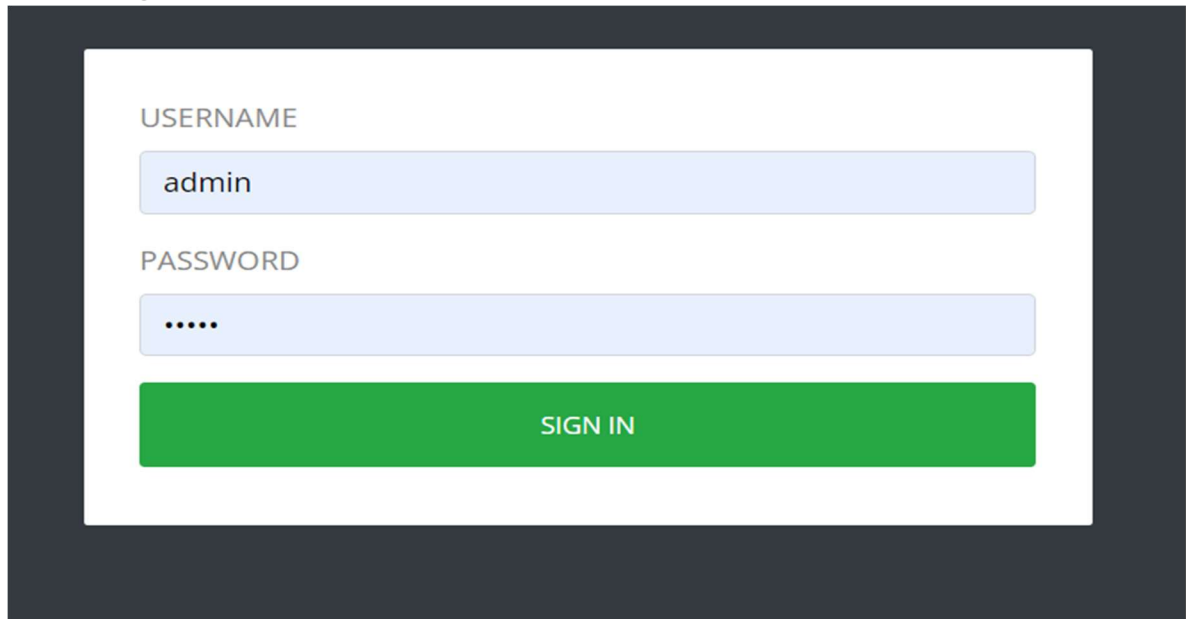
Table 8.12 banner table

19

CHAPTER 9

DOCUMENTATION

9.1 ADMIN MODULE
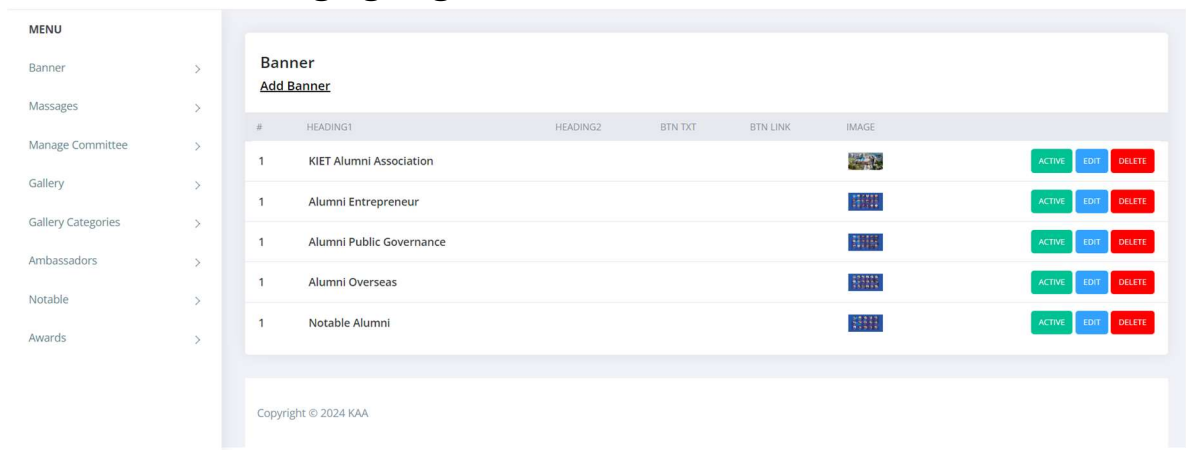
9.1.1 Login Form

Fig 9.1 Login Page

9.1.2 Banner Managing Page

Fig 9.2 Banner Managing Page

9.1.3 Committee Managing Page

Fig 9.3 Committee Managing Page

9.1.4 Gallery Managing Page

Fig 9.4 Gallery Managing Page

9.1.5 Notable Managing Page

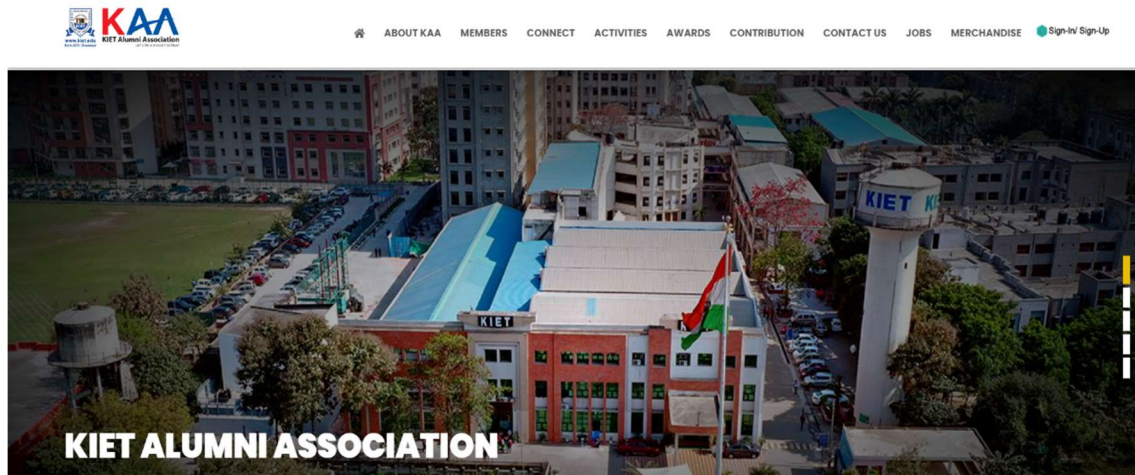Fig 9.5 Notable Managing Page

9.2 USER MODULE
9.2.1 Home Page

Fig 9.6 Home Page

9.2.2 Member Page

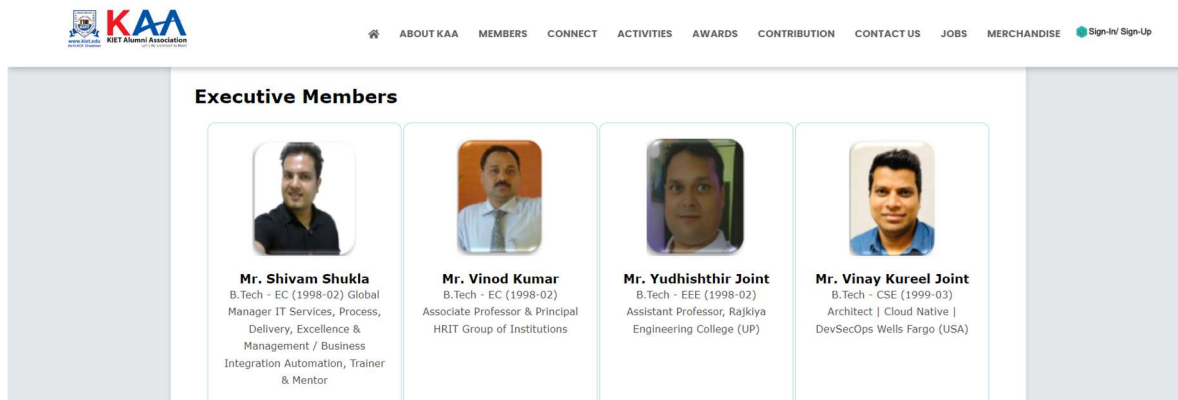Fig 9.7 Members Page

9.2.3 Gallery Page

Fig 9.8 Gallery Page

CHAPTER 10

CONCLUSION

Our project is only a humble venture to satisfy the needs to manage their project work. Several user-friendly coding has also been adopted. Theis's package shall prove to be a powerfulpackage in satisfying all the requirements of the school. The objective of software planning is to provide a frame word that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points:

- A description of the background and context of the project and its relation to workalready done in the area.
- Made a statement of the aims and obj dives of the project.
- Description of Purpose, Scope, and applicability.
- We define the problem on which we are working in the project.
- We describe the requirement specifications of the system and the actions that can bedone on these things.
- We understand the problem domain and produce a model of the system, whichdescribes operations that can be performed on the system.
- We included features and operations in detail, including screen layouts.
- We designed user interface and security issues related to the system.
- Finally, the system is implemented and tested according to test cases.

CHAPTER 11

TESTING

11.1 INTRODUCTION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionalities of components, sub-assemblies, and/or a finished product it is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

11.2 TYPES OF TESTING

11.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

11.2.2 Integration Testing

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. Thus, the system testing is confirmation that all is correct and an opportunity to show the user that the system works.

The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface.

11.2.3 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

CHAPTER 12

BIBLIOGRAPHY

We have successfully designed a project on Web Based Alumni Management. This project is designed in PHP (Hypertext Preprocessor) which runs on XAMP Web Server.

Keeping in view these facts we will develop successfully. Developing the project will help us gain some experience in website development.

12.1 REFERENCES

12.1.1 Coding phase

- PHP (Complete Reference)

12.1.2 Referenced Sites

- www.w3school. com
- www.php.net