



Project report on

## **D-Notice Board**

submitted in partial fulfillment of one of the requirement for the award of the

Degree of

## **BACHELOR OF COMPUTER APPLICATION**

in

## **IDEAL COLLEGE FOR ADVANCED STUDIES**

Submitted by:

**HENNA FATHIMA**

**(Reg No:IKAVBCA013)**

Under the Guidance of

**Mrs.SUHAILA M P**

**DEPARTMENT OF COMPUTER SCIENCE**

**IDEAL COLLEGE FOR ADVANCED STUDIES, KADAKASSERY**

**MALAPPURAM - 679 573, KERALA.**

**2021-2024**

## DECLARATION

*“I hereby declare that the project entitled D-Notice Board submitted to the University of Calicut in partial fulfillment of one of the requirements for the award of the Bachelor of Computer Application is a record of original work done by me during November 2023 – February 2024 under the supervision and guidance of Mrs. SUHAILA MP, Assistant Professor, Department of Computer Science, Ideal College for Advanced Studies, Kadakassery and this is an original work which has not been submitted previously for the award of any degree or diploma to this University or any other University.”*

**KADAKASSERY**

**NAME & REG.NO**

**SIGNATURE**

**HENNA FATHIMA**

**(Reg No:IKAVBCA013)**

# CERTIFICATE



## IDEAL COLLEGE FOR ADVANCED STUDIES

Certified that the project entitled **D-Notice Board** is a record of the bonafide work done by **HENNA FATHIMA (Reg No:IKAVBCA013)**, under our guidance and supervision, in partial fulfillment of one of the requirements for the award of the Degree of Bachelor of Computer Application from the University of Calicut during the academic year 2021 - 2024.

**PROJECT GUIDE: Mrs. SUHAILA MP      HOD: Mr. SUDHEESH S**

Signature:

Signature:

Submitted for the Project -Work Examination held on .....

**INTERNAL EXAMINER:**

**EXTERNAL EXAMINER:**

Place: KADAKASSERY

Date:

## ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank our God almighty for his grace and blessing towards me throughout this project work.

With deep gratitude, I profoundly thank our Principal Prof. KOYAKUTTY.T.K, for acknowledging my request and permitting me to undertake my project.

I heartily thank our HOD Mr. SUDHEESH, department of computer science for his guidance and suggestions during this project work.

I owe my gratitude to my project guide Mrs. SUHAILA MP who took keen interest in my project work and guided us all along, till the completion of my project work by providing all the necessary information for developing a good system.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from my parents, friends and non-teaching staff of the computer science department which helped me in successfully completing my project.

**LIST OF TABLES**

S1.No.	Table Name	Page No.
1	login	39
2	Notice	40

**LIST OF FIGURES**

S1.No.	Table Name	Page No.
1	DataFlow Diagram	27-32
2	Entity Relationship Diagram	33-38

**CONTENTS**

Chapter	Index	Page No.
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
<b>2</b>	<b>SYSTEM ANALYSIS</b>	<b>2-12</b>
	Existing System	
	Proposed System	
	Feasibility Study	
<b>3</b>	<b>SYSTEM DESCRIPTION</b>	<b>13-20</b>
	Programming language	
	Python	
	Tools Used	
	PyCharm	
	Dreamweaver	

<b>4</b>	<b>SYSTEM REQUIREMENTS</b>	<b>21-22</b>
	Hardware Requirements	
	Software Requirements	
<b>5</b>	<b>SYSTEM DESIGN</b>	<b>23-40</b>
	Users of the System	
	Module Design	
	Architecture Design	
	DFD	
	ER Diagram	
	Database Design	
	Structure of Table	
<b>6</b>	<b>NORMALIZATION</b>	<b>41-42</b>
	First Normal Form	
	Second Normal Form	
	Third Normal Form	
<b>7</b>	<b>SYSTEM TESTING</b>	<b>43-47</b>
<b>8</b>	<b>SYSTEM IMPLEMENTATION</b>	<b>48-51</b>
<b>9</b>	<b>CONCLUSION</b>	<b>52-54</b>
	Future Enhancements	
<b>10</b>	<b>BIBLIOGRAPHY</b>	<b>55</b>
<b>11</b>	<b>APPENDIX</b>	<b>56-60</b>
	Screenshots	

## ABSTRACT

*This project explores the transformative role of digital notice boards in modern communication systems. Departing from traditional physical notice boards, digital counterparts leverage technology to deliver dynamic and engaging content, fostering enhanced communication within educational institutions, corporate offices, and public spaces. The versatility of digital notice boards extends beyond information dissemination, serving as platforms for showcasing achievements, recognizing individuals, and promoting a positive atmosphere. In particular, the project emphasizes the significant impact of these boards in educational settings, where they facilitate real-time updates, interactive learning experiences, and the display of student work. By examining the features, benefits, and applications of digital notice boards, this project underscores their growing importance in our digitally-driven world, where they act as catalysts for building connected, informed, and engaged communities.*

# **CHAPTER 1**

## **INTRODUCTION**

A digital notice board represents a modern and efficient way to disseminate information, announcements, and updates within various environments, ranging from educational institutions and corporate offices to public spaces. Unlike traditional notice boards, which rely on physical postings and manual updates, digital notice boards utilize technology to display dynamic content, making communication more engaging, timely, and versatile. In this digital age, these boards have emerged as a valuable tool for enhancing communication and information-sharing, fostering a more connected and informed community or organization. In this discussion, we will explore the features, benefits, and applications of digital notice boards, shedding light on their growing significance in our digitally-driven world.

The versatility of digital notice boards extends beyond their primary function of information dissemination. They serve as a powerful platform for showcasing achievements, highlighting employee or student recognition, and celebrating milestones. This not only fosters a sense of pride within the community but also contributes to a positive and motivating atmosphere. Furthermore, the interactive capabilities of some digital notice boards facilitate two-way communication.

The benefits of digital notice boards become especially pronounced in educational institutions. Teachers can utilize these boards to showcase students' work, display educational content, and provide real-time updates on class activities.



## **CHAPTER 2**

# **SYSTEM ANALYSIS**

System analysis is a general term that refers to an orderly, structured process for identifying and solving a problem. The system analysis process is called the life cycle methodology, since it relates to four significant phases in the life cycle of all business information systems: study, design, development and operation. The definition of system analysis includes not only the process but also the process of putting it together to form a new system. A system analyst is an individual who performs system analysis during any or all of the life cycle phases of a business information system. The system analyst not only analyzes business information system problems, but also synthesizes new systems to solve those problems or meet other information needs. The various techniques used in the study of the present system are:

- Observation
- Interview
- Site visits
- Discussions

### **Preliminary Investigation**

Preliminary investigation checks whether a system is developed by means of SDLC, a prototyping strategy or structured analysis method or combination of these methods. A project request should first be reviewed. The choice of the development strategy of the project is secondary to investment of an organization resource in an information system project. The entire proposal

for the required project is submitted to the selection committee for evaluation to identify that these projects are most beneficial to the organization. The preliminary investigation is thus carried out by the system analyst under the direction of the selection committee. In this stage at first visited an office to know how the working squad is going on. What all are the daily work done by officials and what the present system is.

### **Identification of needs**

The first step in the SDLC is the identification of needs. Since there is likely to be a stream of user's requests, standard procedures must be established to deal with them, the initial investigation is one way of handling this. The objective is to determine whether the request is valid and feasible before an improvement or modification of an existing or building a new system

### **Fact Finding Techniques**

There are several methods for gathering the sort of information. We can use all of these methods for gathering information from the user of the existing system. We can introduce seven fact-finding techniques.

### **Sampling**

Sampling is the process of collecting a representative sample of documents, forms and records. Because it would be impractical to study every occurrence of every form or record in a file or database, system analysts normally use sampling techniques to get a large enough cross section to determine what can happen in the system. The system analyst seeks to sample enough forms to represent the full nature and complexity of the data. First collected sample is a Sample receipt and conducted a study to know how these data can be converted to a digital method

## **Research and Visit Sites**

Another fact- finding technique is to thoroughly research the problem domains. Most problems are not unique. Others have solved them before us. We can contact or perform site visits at companies that have experienced similar problems. If these companies are willing to share, valuable information can be obtained, which may be tremendous time and cost in the development process. Computer trade journals and reference books are also a good source of information from the collected receipt that we found how collection details are recorded in a receipt. The next point is how it is recorded in an office register. We must design a database such that all that data must be maintained in that database.

## **Observation**

Observation is a fact finding technique where the system analyst either participates in or watches a person perform activities to learn about the system. This technique is often used when the validity of data collected through other methods is in question or when the complexity of certain aspects of the system prevents a clear explanation by the end users. This is an effective data- collection technique for obtaining an understanding of a system.

## **Interviews**

These are fact finding techniques whereby the system analyst Collects information from individuals through face-to-face interaction. The personal interview is generally recognized as the most important and most often used fact finding techniques. Interviewing can be used to achieve any or all of the following goals: find facts, verify facts, clarify facts, generate enthusiasm, get the end-user involved, identify requirements and solicit ideas and opinions.

## **Cost Benefit Analysis**

“Cost Benefit Analysis (CBA) estimates and totals up the equivalent money value of the benefits and cost to the community of projects to establish they are worthwhile. “In order to reach a conclusion as to the desirability of a project, all aspects of the project, positive and negative, must be expressed in terms of a common unit; i.e. There must be a “bottom line”. The most convenient common unit is money. A program may provide benefits which are not directly expressed in terms of dollars but there is some amount of money the recipients of the benefits would consider just as the project’s benefits. When all data have been identified and broken down into cost categories, the analyst must select a method for evaluation. Several evaluation methods are available, each with pros and cons.

The common methods are:

- i. Net benefit analysis
- ii. Present value analysis
- iii. Net present value
- iv. Payback analysis
- v. Break even analysis
- vi. Cash flow analysis

After completing all these phases, a simple idea has been generated such that we can design a system that can limit the drawbacks of existing systems. The next point to be focused is its merit and demerit. One of the important facts that the limit the performance is that a position that can cover the main route of goods must be chosen. Integration of map application will arrange the felicity to do this.

## EXISTING SYSTEM

In the current landscape, Content Management Systems (CMS) play a pivotal role in digital notice board systems. These systems often incorporate web-based CMS platforms that empower users to effortlessly create, schedule, and manage content. Notable examples of digital signage software, such as "Xibo," "ScreenCloud," and "NoviSign," exemplify the capabilities of these CMS platforms. This existing system streamlines the process of content creation and management, offering user-friendly interfaces and scheduling functionalities.

1. **Privacy settings:** Ensure that your CMS provides robust privacy settings. Users should have the ability to control access to specific content, restricting it to authorized individuals or groups. This can include setting permissions for viewing, editing, or deleting content.
2. **Watermarking:** Incorporate watermarking features to protect against unauthorized use or distribution of content. Watermarks can serve as a visual deterrent and also help identify the source of leaked or misused content.
3. **Support For Large Networks:** Ability to manage a large number of display devices in a network.
4. **Multi display Support:** Ability to create and manage content across multiple displays.
5. **Offline Content Playback:** Ability to display cached content in case of internet outages.

## PROPOSED SYSTEM

In the implementation and technical features section of our digital notice board project report, we delve into the core aspects that make these boards efficient and user-friendly. Detailing the software components, and we explore how the digital notice boards seamlessly integrate with existing infrastructure. This section highlights the user interface, emphasizing its intuitive design for easy navigation and content management.

The incorporation of multimedia elements, such as videos and images, is explored to underscore the dynamic and engaging nature of the content displayed.

The scope of a proposed system could include:

- a. **User Friendly interface:** In the design and development of our digital notice board software, a paramount focus has been placed on ensuring a user-friendly interface. This means that the software has been crafted with an emphasis on simplicity and intuitiveness, allowing users, particularly administrators, to effortlessly navigate through various functionalities.
- b. **Security:** Our software prioritizes user privacy by ensuring that users only need to view notices and updates provided by the administrator, eliminating the need for them to input any personal information. This approach enhances the security of user data, as the system operates on a minimal information exchange basis, safeguarding user privacy and Maintaining a secure user experience.

- c. **Multiple Templates:** Administrators have the flexibility to enhance communication by selecting and applying notices across multiple templates, providing a versatile and efficient means of conveying information to users.
- d. **No Technical Knowledge Required:** digital notice board software is designed to be incredibly user-friendly, ensuring that individuals with varying levels of technical expertise or prior experience can navigate and utilize the platform effortlessly. With a clear and intuitive interface, the software eliminates the need for any specialized technical knowledge, making it accessible to a wide range of users.
- e. **Cross-Platform Accessibility:** One of the standout features of our project is its inherent ability to be accessed across various devices that support web browsing. Whether users are on a desktop computer, laptop, tablet, or smartphone, the website's responsive design ensures a seamless and consistent user experience.

Overall, You can use it on any device that connects to the internet, like your computer, tablet, or phone. Whether you want to create engaging digital notice boards or organize team tasks, our software is designed to be straightforward and flexible. It's like having a simple and powerful tool that anyone can use, no matter how tech-savvy they are:

## **FEASIBILITY STUDY**

A feasibility study is a test of a system proposal according to its work ability, impact on the organization, ability to meet user needs, and effective use of resources. The objective of feasibility study is acquiring a sense of the scope of the system. The feasibility of a project can be ascertained in terms of technical factors, economic factors, or both. A feasibility study is documented with a report showing all the ramifications of the project. It is very important to evaluate the feasibility study and risk analysis is related in many ways. If the project study is great, the feasibility of producing quality software is reduced. The key factors considered during the feasibility study are:

- Economic Feasibility
- Behavioral Feasibility
- Technical Feasibility
- Operational Feasibility
- Legal Feasibility
- Environment Feasibility



## ECONOMIC FEASIBILITY

Economic analysis is the most used method for evaluating the effectiveness of a system. Cost benefit analysis is the most important assessment of economic justification of the project. Cost-benefit analysis delineates the cost for project development and weighs them against tangible and intangible benefits of a system. This type of analysis varies with the characteristics of the system to be developed, the relative site of the project, and the expected return on investment. Benefits of a new system are always determined relative to the existing mode of operation. Economic feasibility deals with the economic impact faced by the organization to implement the new system. Not only the cost of hardware, software etc., is considered but also the form of reduced costs. The project, installed certainly will be beneficial since there will be a reduction in manual work and increase in speed of work.

The analysis raises financial and economic questions during the preliminary investigation to estimate the following:

- The cost to conduct a full systems investigation.
- The cost of hardware and software for the class of application of the project being considered.

To be judged feasible, a proposal for the specific project must pass all these tests, otherwise it is not considered as a feasible project. We gathered the details regarding the financial aspects incorporated in the system to make it cost efficient.

## BEHAVIORAL FEASIBILITY

This analysis involves how it will work when it is installed and the assessment of the political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and therefore it will accept a broad audience.

## OPERATIONAL FEASIBILITY

Operational feasibility assesses the extent to which a project can be successfully implemented within the existing operational environment of an organization. It involves evaluating whether the proposed software aligns with the organization's processes, resources, and strategic goals.

## LEGAL FEASIBILITY

Legal feasibility involves an assessment of the project's compliance with laws, regulations, and legal requirements governing the industry or jurisdiction in which it operates. In the context of our software project, legal feasibility examines whether the digital notice board and task management system adhere to relevant legal standards. This includes considerations such as data protection and privacy laws, intellectual property rights, accessibility standards, and any industry-specific regulations. Ensuring that the software aligns with legal frameworks is crucial for mitigating risks associated with legal challenges and safeguarding the interests of both the organization and its users.

## TECHNICAL FEASIBILITY

Technical feasibility assesses the practicality and viability of implementing a project from a technological standpoint. In the case of our software project, technical feasibility examines whether the proposed digital notice board and task management system can be effectively developed and integrated. This involves evaluating the availability of necessary technologies, assessing the scalability and adaptability of the system, and ensuring compatibility with existing infrastructure. A positive technical feasibility outcome indicates that the project can be successfully implemented, meeting performance standards and accommodating potential future enhancements. It assures stakeholders that the technology required for the software is both available and feasible, laying a solid foundation for the project's development and deployment.

## ENVIRONMENTAL FEASIBILITY

Environmental feasibility evaluates the compatibility of a project with its surroundings, considering the impact on the natural and social environment. For our software project, environmental feasibility involves assessing how the digital notice board and task management system align with sustainability goals and societal expectations. This includes considerations such as energy efficiency, carbon footprint, and social responsibility. A positive environmental feasibility outcome ensures that the software contributes positively to the broader environment, minimizing any potential adverse effects and promoting eco-friendly practices. This aligns with the growing emphasis on sustainable and socially responsible technology solutions, fostering a positive relationship between the software and its environmental context.

## **CHAPTER 3**

# **SYSTEM DESCRIPTION**

### **PROGRAMMING LANGUAGE**

#### **PYTHON:**

Originally named "ABC," Python, developed by Guido van Rossum in 1989, underwent a significant transformation and was officially released in 1991. Inspired by Monty Python's Flying Circus, the language prioritizes readability and simplicity. Python's clean design philosophy and versatility quickly made it popular for scripting, web development, data analysis, and AI. Its success lies in being a powerful yet accessible language, appealing to both beginners and seasoned programmers.

Python: Where simplicity meets versatility, making coding a delight for developers of all levels.

- Python is renowned for its simplicity and consistency, offering a cohesive programming experience that prioritizes readability and ease of use.
  
- Python's power lies in its versatility, with a rich ecosystem of libraries for seamless development in web, data science, AI, and more.

## **Importance of Python**

Python's importance in the programming landscape is undeniable, owing to its versatility, readability, and a vast ecosystem of libraries and frameworks. As a general-purpose language, Python finds applications in various domains, from web development and data analysis to machine learning and artificial intelligence. Its clean and concise syntax makes it accessible for beginners, while its powerful capabilities attract seasoned developers. The open-source nature of Python encourages collaboration and continuous improvement within the community. The language's prominence in emerging technologies, coupled with its ease of learning and implementation, solidifies its position as a fundamental and influential tool for developers worldwide.

## **Python architectural**

Python's architecture, centered around an interpreter, libraries, and runtime environment, employs a bytecode compilation model. Source code is translated into bytecode, executed by the Python interpreter. The language boasts an extensive Standard Library and supports third-party libraries, enhancing its adaptability in areas like web development, scientific computing, and machine learning. Python's runtime environment manages memory, exceptions, and dynamic typing. Notably, the Global Interpreter Lock (GIL) impacts multi-threaded execution, simplifying memory management but potentially affecting performance in CPU-bound tasks. In essence, Python's design emphasizes simplicity, readability, and flexibility, enabling developers to create scalable solutions across diverse applications.

## Features of Python

- 1) **Readability:** Python's syntax is designed to be clear and readable, making it easy for developers to write and maintain code. The use of indentation for block delimiters enhances code readability.
- 2) **Versatility:** Python is a general-purpose language, suitable for a wide range of applications. It is used in web development, data science, machine learning, artificial intelligence, automation, scientific computing, and more.
- 3) **Extensive Standard Library:** Python comes with a rich and comprehensive Standard Library that includes modules and packages covering a broad range of functionalities. This eliminates the need for developers to write code from scratch for common tasks.
- 4) **Dynamically Typed:** Python is dynamically typed, meaning that the type of a variable is interpreted during runtime. This flexibility simplifies coding and allows for more natural expression of ideas.
- 5) **Interpreted and Interactive:** Python is an interpreted language, which means that code is executed line by line. It also supports an interactive mode, where developers can execute code interactively, making it great for learning and testing snippets of code.
- 6) **Object-Oriented:** Python supports object-oriented programming (OOP) principles, facilitating the creation and use of classes and objects. This promotes code organization and reusability.
- 7) **High-Level Language:** Python is a high-level language, abstracting many low-level details and providing a more straightforward and concise syntax. This allows developers to focus on solving problems rather than dealing with complex system intricacies.

- 8) **Community and Documentation:** Python has a large and active community of developers who contribute to its growth and development. The community support, along with extensive documentation, makes it easy for developers to find resources and solutions to problems.
- 9) **Cross-Platform Compatibility:** Python is platform-independent, meaning that Python code can run on various operating systems without modification. This cross-platform compatibility adds to its versatility.
- 10) **Libraries and Frameworks:** Python has a vast ecosystem of third-party libraries and frameworks that cater to specific needs. Popular libraries like NumPy, Pandas, and frameworks like Django and Flask contribute to the language's capabilities in data analysis, web development, and more.
- 11) **Garbage Collection:** Python features automatic memory management through garbage collection. This helps developers avoid manual memory allocation and deallocation, reducing the likelihood of memory leaks and making the language more user-friendly.
- 12) **Exception Handling:** Python has robust support for exception handling, allowing developers to manage errors gracefully. The try-except block structure makes it easy to identify and handle exceptions, contributing to the reliability of Python programs.
- 13) **Community-driven Development:** The open-source nature of Python encourages community-driven development. The language evolves through contributions from developers worldwide, resulting in regular updates, improvements, and the introduction of new features.
- 14) **Integration Capabilities:** Python seamlessly integrates with other languages like C and C++, facilitating the incorporation of existing codebases. This interoperability is crucial in scenarios where performance-critical tasks are implemented in languages with lower-level control.

- 15) Scalability:** Python's scalability is evident in its application across a broad spectrum of projects, from small scripts to large-scale enterprise applications. The language's flexibility and the availability of tools and frameworks contribute to its scalability in various contexts.

## **TOOLS USED:**

### **PyCharm**

PyCharm, developed by JetBrains, is a robust Python IDE known for its powerful tools and user-friendly interface. Widely used for efficient Python application development, it offers features like intelligent code completion, built-in testing, debugging tools, and seamless integration with version control. Supporting Python and web technologies like HTML, CSS, and JavaScript, PyCharm serves as a comprehensive solution for full-stack development. It emphasizes productivity and code quality, available in both a free community edition and a professional edition with advanced features for developers.

PyCharm offers a range of features to assist developers in their software development process. Some of the key features include:

1. **Code editor:** PyCharm's code editor is a powerful and user-friendly component that lies at the heart of the integrated development environment (IDE). Offering advanced features and tools, the code editor in PyCharm provides intelligent code completion, syntax highlighting, and error checking to enhance the developer's coding experience.



2. **Project management:** PyCharm's project management tools facilitate the creation and organization of projects, offering support for diverse project types such as Python, Django, Flask, JavaScript, and web frameworks, enhancing developers' flexibility and efficiency.
3. **Debugging:** PyCharm offers powerful debugging tools to help identify and fix issues in your code.
4. **Version control:** PyCharm seamlessly integrates with various version control systems, including Git, Subversion, and Mercurial, empowering developers to efficiently manage source code, collaborate seamlessly, and track changes with ease.
5. **Profiling:** PyCharm includes tools for profiling Python applications to identify performance issues.
6. **GUI builder:** The IDE provides a graphical user interface (GUI) builder to help developers create user interfaces quickly and easily.
7. **Plugins:** PyCharm supports a range of plugins that can be installed to extend its functionality.
8. **Supported platforms:** PyCharm can be installed on Windows, macOS, and Linux operating systems. It supports both 32-bit and 64-bit architectures.

9. Python interpreter: To use Python development environments like PyCharm or IDLE, you need to have the Python interpreter installed on your system. The environments typically support various Python versions, with compatibility for Python 3.x being the most common requirement.
10. Integrated Tools: PyCharm comes equipped with integrated tools for seamless application building and packaging. It supports popular build systems such as Ant, Maven, and Gradle, providing developers with flexibility in managing their projects efficiently.
11. HTML5 support: The IDE includes tools for developing HTML5 applications. It provides features like live preview, code completion, and debugging.
12. Database support: PyCharm includes tools for working with databases. It provides support for various databases like MySQL, Oracle, and PostgreSQL.
13. Community support: PyCharm has a large community of developers who contribute to its development. The community provides support through forums, mailing lists, and social media.
14. JetBrains PyCharm: PyCharm is a product of JetBrains, and its development is driven by a vibrant and collaborative community of developers. This open-source project consistently evolves and enhances the Python development experience within the PyCharm IDE.

In summary, PyCharm is a powerful Python IDE, renowned for its comprehensive tools, robust code editor, and seamless integration with version control. Widely favored by developers, its continuous improvement and strong community support make it a top choice for efficient and versatile Python development.

## **DREAMWEAVER**

Adobe Dreamweaver is a software program for designing web pages, essentially a more fully featured HTML web and programming editor. The program provides a what-you see-is-what-you- get (WYSIWYG) interface for users to create and edit web pages in a more user-friendly environment. Dreamweaver supports multiple mark-up languages including HTML and Extensible Mark-up Language (XML), style sheet languages like Cascading Style Sheets (CSS), and programming languages including JavaScript, C#, Visual Basic (VB), Active Server Pages (ASP), and others. The program is also available in a number of languages, including English, Spanish, French, German, Japanese, Chinese (both Simplified and Traditional), Italian, Russian, and more. Dreamweaver was originally developed and published by Macromedia in 1997. Adobe purchase Macromedia (which included the rights to Dreamweaver) in 2005 and continued the development of the program. The many features of Dreamweaver make it a versatile web editing tool, where it can be for creating complex or very simple sites.

## **CHAPTER 4**

# **SYSTEM REQUIREMENTS**

### **HARDWARE REQUIREMENTS**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application.

- Processor: i3 processor/Above
- Hard Disk: 40 GB or Above
- RAM: 8GB
- Display Type: Any Display.
- Input Device: Mouse, keyboard
- Output Device: Monitor

## SOFTWARE REQUIREMENTS

A software requirement specification (SRS), a requirements specification for a software system, is a complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition, it also contains non-functional requirements. Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

The software requirements specification document enlists all necessary requirements that are required for the project development. To derive the requirements, we need to have a clear and thorough understanding of the products to be developed. This is prepared after detailed communications with the project team and customer.

- Operating System: WINDOWS 10 or above for better performance
- Front end: HTML/CSS and JavaScript (For web application)
- Back end: MYSQL
- Software: PyCharm, Dreamweaver
- Web Browser: Internet Explorer/Google Chrome/Firefox
- Web Server: GlassFish Server

## **CHAPTER 5**

# **SYSTEM DESIGN**

### **INTRODUCTION**

The most creative and challenging phase of the system development is the system design. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through the logical and physical stages of development. In designing a new system, the analyst must have a clear understanding of the objectives, which the design is aiming to fulfill. The first step is to determine how the output is to be produced and in what format. Second input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing. Finally, details related to justification of the system and an estimate of the impact of the candidate system on the user and the organization are documented and evaluated by the management. Design of a system can be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Thus, system design is a solution, “how to” approach to the creation of a new system. The design step provides a data design, architectural design and a procedural design. The data design transforms the information domain created during analysis into the data structure that will be required to implement the software. The architectural design defines the relationship among major structural components and procedural description of the software. Source code is generated and testing conducted to integrate and validate the software.

System design goes through two phases of development:

- Logical design
- Physical design

## **1.Logical Design**

The part of the design process that is independent of any specific hardware or software platform is referred to as logical design. During logical design, all functional features of the system chosen for development in analysis phase are described independently of any computer platform. Logical design concentrates on the business aspects of the system and tends to be oriented to a high level of specificity.

## **2.Physical Design**

Physical design is the part of the design phase in which the logical specifications of the system from logical design are transferred into technology-specific details from which all programming and system construction can be accomplished. As a part of the physical design, analysts design the various parts of the system to perform the physical operation necessary to facilitate data capture, processing, and information output.

## **INPUT DESIGN**

The first step in system design is to design input and output within predefined guidelines. In input design, user originated inputs are converted into computer-based format. In output design, the emphasis is on producing the hard copy of the information requested or displaying the output on a CRT screen in a predefined format. The following features have been incorporated into the input design of the proposed system.

### **1.Easy Data Input**

Data entry has been designed in a manner much similar to the source documents.

Appropriate messages are provided in the message area, which prompts the user in entering the right data. Erroneous data inputs are checked at the end of each screen entry.

### **2.Data Validation**

The input data is validated to minimize errors in data entry. For certain data specific codes have been given and validation is done which enables the user to enter the required data or correct them if they entered wrong codes.

### **3.User Friendliness**

User is never left in a state of confusion as to what is happening, instead appropriate error and acknowledge messages are sent. Error maps are used to indicate the error codes and specific error messages.



## **4.Consistent Format**

A fixed format is adopted for displaying the title messages. Every screen has line, which displays the operation that can be performed after the data entry. They are normally done at the touch of a key.

## **5.Interactive Dialogue**

The system engages the user in an interactive dialogue. The system is able to extract missing or omitted information from the user by directing the user through appropriate messages, which are displayed.

## **OUTPUT DESIGN**

The output is the most important and direct source of information to the user. The output should be provided in a most efficient formatted way. Based on the options given by the users and the administrator various types of output screens have been generated. The computer output is the most important and direct source of information to the user. Efficient and intelligible output design improves the system's relationship with the user and helps in decision- making. Output design was studied going actively during the study phase. The objective of the output design is defined the contents and format of all documents and reports in an attractive and useful format.

# **ARCHITECTURE DESIGN**

## **DATA FLOW DIAGRAM (DFD)**

Data flow diagram (DFD) represents the flows of data between different processes in a business. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. It is a graphical technique that depicts information flow and the transforms that are applied as data move from input to output. It provides a simple, intuitive method for describing business processes without focusing on the details of computer systems. Data flow diagrams can be used in both Analysis and Design phases of the SDLC.

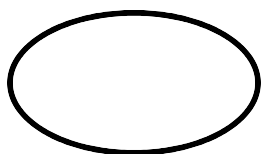
A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes or information about whether processes will operate in sequence or in parallel.

A DFD may look similar to a flow chart. However, there is a significant difference with the data flow diagram. The arrows in DFDs show that there is a flow of data between the two components and not that the component is sending the data that must be executed in the following component. A component in DFD may not continue execution when sending data and during execution of the component receiving the data. The component sending data can send multiple sets of data along several connections. In fact, a DFD node can be a component that never ends.

DFDs only involve four symbols. They are:

1. Process
2. Data Object
3. Data process
4. External entity

### 1. Process



Transform of incoming data flow(s) to outgoing flow(s).

### 2. Data Object



Movement of data in the system.

### 3. Data Store



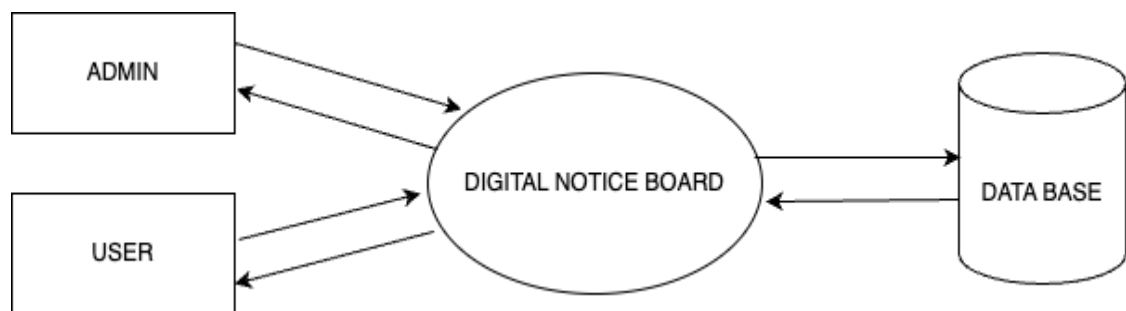
Data repositories for data that are not moving. It may be as simple as a buffer or a queue or as sophisticated as a relational database.

### 4. External entity

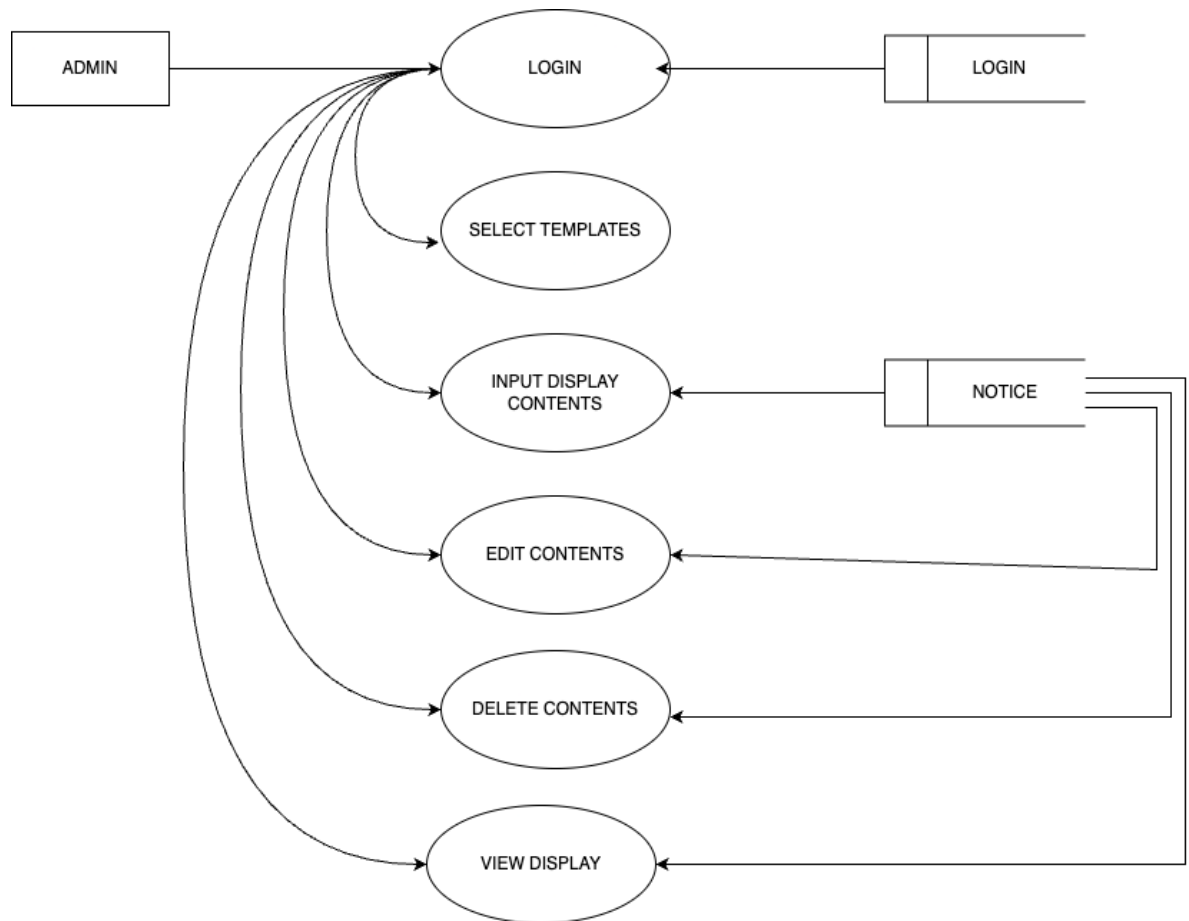


Sources or destinations outside the specified system boundary.

## LEVEL 0



## LEVEL 1.1



**LEVEL 1.2**

## **ENTITY RELATIONSHIP DIAGRAM**

An entity-relationship diagram (ER) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system. An ER is a data modeling technique that can help define business processes and can be used as the foundation for a relational database. This process has proved to enable us to produce a good database structure so that data can be stored and retrieved in a most efficient manner.

### **History of ER models**

ER diagrams are a visual tool which is helpful to represent the ER model. It was proposed by Peter Chen in 1971 to create a uniform convention which can be used for relational databases and networks. He aimed to use an ER model as a conceptual modeling approach.

### **USE OF ER DIAGRAM**

Here, are prime reasons for using the ER Diagram

- Helps you to define terms related to entity relationship modeling
- Provide a preview of how all your tables should connect, what fields are going to be on each table
- Helps to describe entities, attributes, relationships
- ER diagrams are translatable into relational tables which allows you to build databases quickly
- ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications



- The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram
- ERD is allowed you to communicate with the logical structure of the database to users

### Components of the ER Diagram

Three main components of an ER:

1. Entities
2. Relationship and
3. Cardinality

- 1. Entity:** A data entity is anything real or abstract about which we want to store data.

E.g.

Student

Teacher

Projects

- 2. Relationship:** A data relationship is a natural association that exists between one or more entities.

For example, an employee works at a department, a student enrolls in a course. Here, works at and enrolls are called relationships.

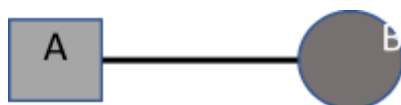
**3. Cardinality:** Cardinality notations define the attributes of the relationship between the entities in terms of numbers. Cardinalities can denote that an entity is optional.

The three main cardinal relationships are:

1. One-to-One
2. One-to-Many
3. Many-to-Many

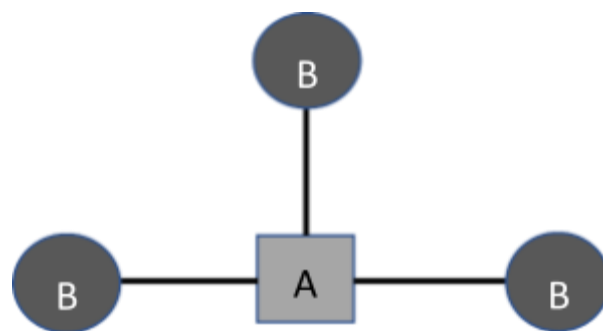
**1. One-to-one (1:1)**

For example, if each customer in a database is associated with one mailing address.



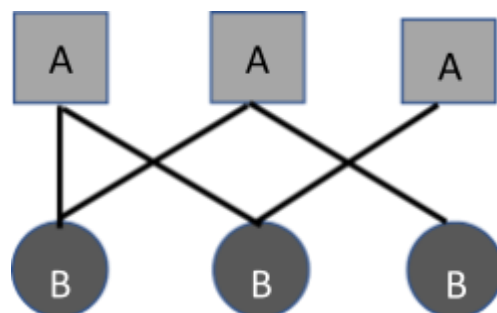
## 2. One-to-many (1:M)

For example, a single customer might place an order for multiple products. The customer is associated with multiple entities, but all those entities have a single connection back to the same customer.



## 3. Many-to-many (M:M)

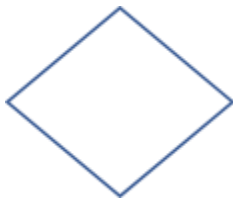
For example, at a company where all call-center agents work with multiple customers, each agent is associated with multiple customers, and multiple customers might also be associated with multiple agents.



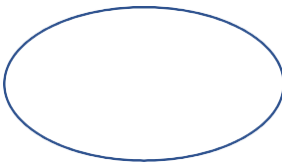
## BASIC SYMBOLS



Entity



Relationship



Attribute



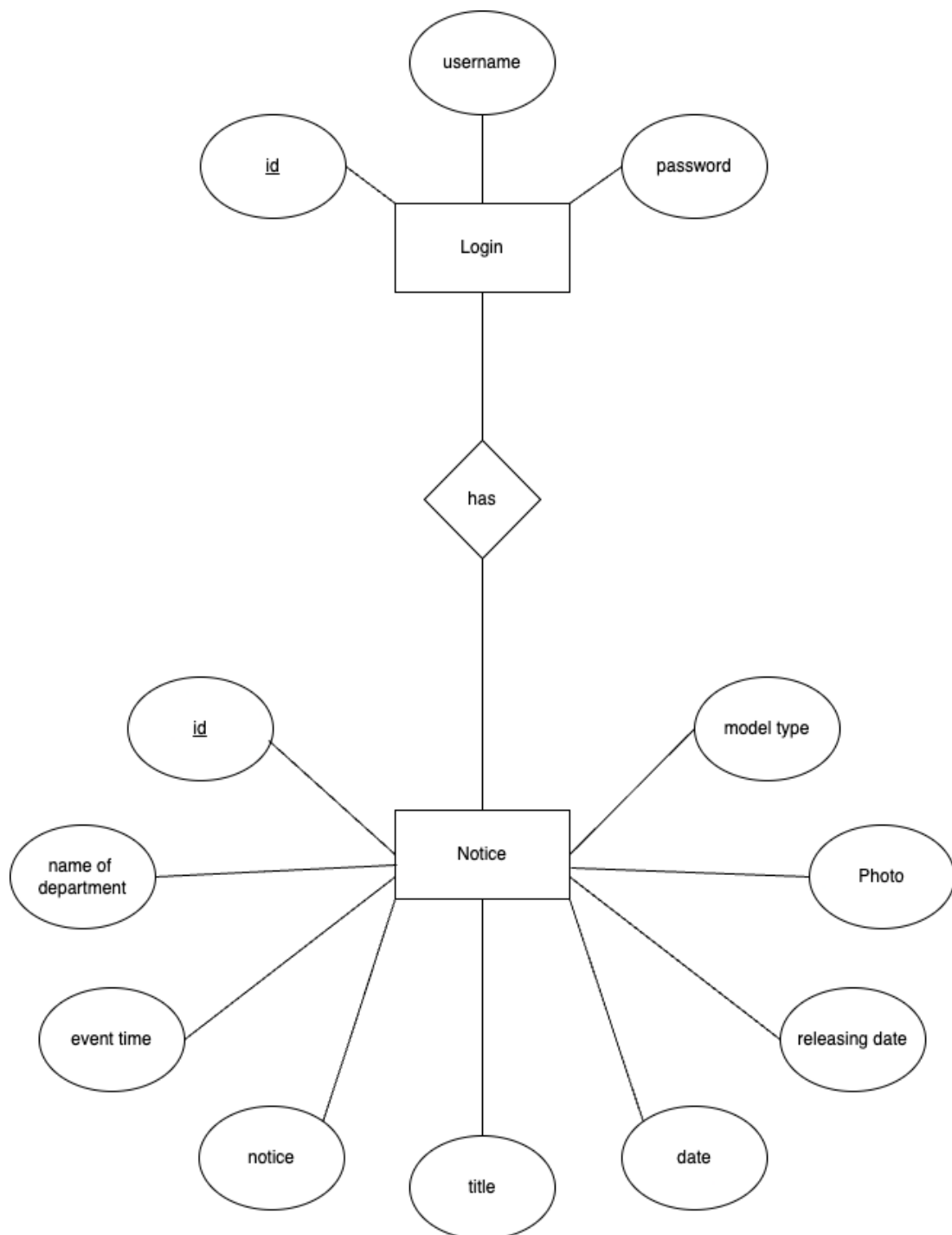
Weak Entity



Weak Entity Relationship



Multivalued Attribute



## DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general way is to make information accessing easy, quick, inexpensive and flexible for the user. In the database design several objectives are considered: controlling redundancy, ease of learning and use, data dependence, more information at low cost, accuracy and integrity are some of them.

In this phase, information from the ER-diagram is used to design the database. The entities in the ER-diagram represent the table that have to be created and the attributes represent the fields that are in each table.

## STRUCTURE OF TABLE

**Database Name:** dnoticeboard

**Table Name:** login

**Table Description:** login

**Table No:** 1

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS	DESCRIPTION
id	int		PrimaryKey	id
username	varchar	50	Not Null	Username
password	varchar	20	Not Null	Password

**Table Name:** Notice

**Table Description:** Notice

**Table No:** 2

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS	DESCRIPTION
id	int		Primary Key	id
name_of_Department	varchar	50	Not Null	Department
event_Time	varchar	50	Not Null	Time
notice	varchar	50	Not Null	Description
title	varchar	50	Not Null	Title
date	varchar	50	Not Null	Date
releasing_date	varchar	50	Not Null	Release Date
Photo	varchar	500	Not Null	Image
model_type	varchar	50	Not Null	Model

## CHAPTER 6

# NORMALIZATION

Normalization is the process of decomposing a set of relations with anomalies to produce smaller and well structured relations that contain minimum redundancy. It is a formal process of deciding which attributes should be grouped together in a relation.

### **FIRST NORMAL FORM:**

First Normal form (1NF) is now considered to be part of the formal definition of relational model. 1NF is designed to disallow multi valued attributes, composite attributes, and their combinations. It states that the domain of an attribute must include only atomic values. A domain is atomic, if elements of the domain are considered to be indivisible units. We say that a relational schema R is in 1NF if the domain of all attributes of 'R' is atomic.

### **SECOND NORMAL FORM:**

The relation is said to be in 2NF if each attribute A in R meets one of the following criteria:

- a) It appears in the primary key.
- b) It is fully functionally dependent on the primary key.

The tables designed in the proposed system contain a primary key for uniquely identifying each user.



**THIRD NORMAL FORM:**

Third Normal form (3NF) is based on the concept of transitive dependency. A relation is said to be in 3NF if it is in 2NF and has no transitive dependencies.

That is all the non key attributes should be functionally determined by the primary key. In the proposed system all attributes of tables fully depends on the primary key only that is all non key attributes are mutually independent.

## **CHAPTER 7**

# **SYSTEM TESTING**

### **SYSTEM IMPLEMENTATION AND TESTING**

System testing is actually a series of different testings whose primary purpose is to fully exercise the computer based system. Software testing is a critical element of software quality assurance and represents the ultimate review of the specification, design and coding. System testing makes a logical assumption that all the parts of the system are correct; the goal will be successfully achieved. Testing is the final verification and validation activity within the organization itself. During testing, the major activities are concerned with the examinations and modifications of the source code.

Testing is a process of executing a program with the intent of finding an error. However, not restricted to being performed after the development phase is complete, but this is to be carried out in parallel with all stages of system development, starting with requirements specification. Testing results, once gathered and evaluated, provide a qualitative indication of software quality and reliability and serve as a basis for design modification if required. A good test is one that uncovers an as yet undiscovered error.

Testing Objectives are:

- Unit Testing
- Integration Testing
- Validation Testing
- Output Testing

**UNIT TESTING:**

Unit testing enables a programmer to detect errors in coding. A unit test focuses on verification of the smallest unit of software design. This testing was carried out during the coding itself. In this testing step, each module is going to be work satisfactory as the expected output from the module.

**INTEGRATION TESTING:**

System testing / integration testing is a systematic technique for constructing tests to uncover errors associated within the interface. It comprises two or more developers or development teams combining the programs to determine how they work together like login, registration, etc. All the errors uncovered in the system testing were corrected for the next testing steps.

**VALIDATION TESTING:**

The next level of testing is validation testing. Here the entire software is tested. The reference document for this process is the requirement and the goal is to see if the software meets its requirements. The requirement document reflects and determines whether the software functions as the user expected. At the culmination of integration testing, software is completely assembled as a package and corrected and a final series of software test validation tests begins. The proposed system under construction has been tested by using validation testing and found to be working satisfactory. Data validation checking is done to see whether the corresponding entries made in different tables are done correctly. Proper validation checks are done in case of insertion and updating of tables, in order to see that no duplication of data has occurred. If any such case arises proper warning messages will be displayed. Double configuration is done before the administrator deletes the data in order to get positive results and to see that no data has been deleted by accident.

**OUTPUT TESTING:**

After performing the validation testing, the next step is the output testing of the enhanced system. No system could be useful if it does not produce the required output in the required format. The outputs generated or displayed by D-NoticeBoard are defined during the system analysis. During output testing the developer required some new report layout changes. These changes were done and tested before the final delivery of the system.

**TEST CASE DESIGN METHODS**

The primary objectives of test case design methods are to derive a set of tests that has the highest likelihood of uncovering the defects. To accomplish this objective, two categories of test case design techniques are used:

- Black box testing
- White box testing

**WHITE BOX TESTING:**

White box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using White box testing methods, we can derive test cases, that

1. Guarantee that all independent paths within a module have been exercised at least once.
2. Exercise all logical decisions on their true and false sides.
3. Execute all loops at their boundaries and within their operational bounds
4. Exercise internal data structures to ensure their validity.

**BLACK BOX TESTING:**

Black box testing methods focus on the functional requirements of the software. That is, black box testing enables us to derive sets of input conditions that will fully exercise all functional requirements of the program.

Black box testing attempts to find errors in the categories:

1. Incorrect or missing functions.
2. Interface errors.
3. Errors in data structure or external database access.
4. Performance errors.

**TESTING STRATEGIES**

Testing strategy is used to perform testing in a planned and systematic manner. It identifies what levels of testing are to be applied, methods and techniques, and tools to be used. This strategy also decides test cases, test specifications, test case designs, and puts them together for execution.

The choice of software testing strategy depends on the nature of the software. There are different types of software testing strategies, selected by the testers depending upon the nature & size of the software:

- Model-based testing strategy
- Methodical testing strategy
- Process-oriented testing strategy
- Dynamic testing strategy
- Philosophical testing strategy
- Analytic testing strategy

The document that describes an input, action, or event and its expected result, in order to determine whether the software or a part of software is working correctly or not is a test case. Incomplete and incorrect test cases lead to incorrect and erroneous test outputs. A test case should be developed in such a manner that it checks software with all possible inputs. This is exhaustive testing and the test case, which is able to perform exhaustive testing, is an ideal test case. The test case comprised of testing the modules for:

1. Common errors
2. Simple boundaries- minimum, maximum
3. Wrong kind of data
4. Database specific errors

## **CHAPTER 8**

### **SYSTEM IMPLEMENTATION**

The implementation of our digital notice board project encompasses a comprehensive approach to seamlessly integrate hardware and software components. The user-friendly interface design prioritizes accessibility, allowing content creators to effortlessly manage real-time updates and announcements. Dynamic content management, including scheduling features, ensures the timely dissemination of information. Interactive elements. The system features a dedicated section for showcasing achievements, employee or student recognition, fostering a positive and motivating atmosphere. Security measures, including encryption and secure login processes, safeguard against unauthorized access. In educational settings, seamless integration with existing systems allows teachers to effortlessly showcase students' work and provide real-time updates on class activities. Regular system updates, user training, and compliance with data protection laws are integral components, ensuring a robust and efficient digital notice board system. Performance monitoring tools guarantee optimal functionality, contributing to effective communication and a connected community or organization.

In conclusion, the system implementation for our digital notice board project embodies a comprehensive strategy designed to revolutionize communication and information-sharing. By departing from traditional methods and leveraging technology, the platform successfully delivers

dynamic content, fostering engagement within diverse environments. The integration of interactive features and real-time updates ensures a versatile and impactful user experience, particularly beneficial in educational settings. The implementation prioritizes user satisfaction by providing a user-friendly interface, facilitating seamless navigation and content management. The choice of Java in NetBeans for the back-end and HTML/CSS for the front-end underscores the commitment to scalability and maintainability. With a focus on detailed documentation, the project not only ensures current users' comprehension but also paves the way for future developers to build upon and enhance the system. The success of the system implementation lies in its ability to not just disseminate information but to catalyze a positive and connected community through innovative and transformative digital notice board functionalities.

## **IMPLEMENTATION OF THE PROJECT**

**Problem Identification:** Recognizing the limitations of traditional notice boards, including static content, manual updates, and restricted interactivity.

**Goal Definition:** Clearly defining project goals, such as transforming communication through dynamic content, real-time updates, and fostering engagement within educational institutions, corporate offices, and public spaces.

**Conduct research:** Undertaking comprehensive research, including studying existing literature on digital communication systems, conducting user surveys, and analyzing the communication dynamics in diverse environments.



**Develop a plan:** Crafting a strategic plan based on research findings. This may involve hardware and software integration, user-friendly interface design, dynamic content management, and interactive features to enhance the overall user experience.

**Implement the plan:** Executing the plan by integrating hardware components, developing user-friendly interfaces, and implementing features like real-time updates and interactive capabilities. Collaboration with developers ensures a scalable and maintainable solution.

**Monitor and evaluate:** Continuously monitoring the functionality and responsiveness of the digital notice board system. Evaluation includes assessing user satisfaction, engagement, and the impact of dynamic content on communication within the targeted environments.

**Iterate and improve:** Making iterative adjustments based on evaluation outcomes. This could involve refining the user interface, adding new interactive features, and exploring additional functionalities to enhance the digital notice board system.

The overall approach centers around addressing the limitations of traditional notice boards through a strategic blend of hardware and software integration, user-friendly design, and dynamic content management. By prioritizing user engagement and communication enhancement, the digital notice board project aims to transform communication dynamics in various settings, fostering a connected and informed community. The iterative process ensures continuous improvement, aligning the project with evolving user expectations and communication needs.

## **Post Implementation and Software Maintenance**

Maintenance involves the software industry captive, typing up system resources. It means restoring something to its original condition. Maintenance involves a wide range of activities including correcting, coding, and design errors, updating documentation and test data, upgrading user support. Maintenance was done after the successful implementation. Maintenance is continued till the product is re- engineered or deployed to another platform. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

## **CHAPTER 9**

# **CONCLUSION**

In conclusion our Digital Notice Board project aims to improvise the Digital Notice Board experience by providing a simple and seamless user interface. The software ensures that anyone can use the technology without having in-depth technical knowledge or experiences. Our system consists of two modules, which are Admin and User.

The project is developed using HTML, CSS and Javascript as front-end. This language has been selected for future improvements based on user specification and current system analytics flexibility. Since the system is developed in modules, future enhancement is very easy. This project has been developed, tested, documented and implemented successfully. The main objective of the system was brought into effect. This has been developed as versatile and user friendly as possible keeping in mind the advanced features in this technology. We use technologies like HTML/CSS, Python, Ajax, SQL, Javascript to develop the system along with JetBrains PyCharm and Adobe Dreamweaver.

## FUTURE ENHANCEMENTS

**Integration with other social media platforms:** We can consider integrating your social media platform with other popular social media platforms like Facebook, Twitter, and Instagram. This will allow users to share their posts on multiple platforms and increase your user base.

**Advanced facial recognition algorithms:** We can explore more advanced facial recognition algorithms to improve the accuracy of face detection and recognition. This will enhance the security and privacy of your platform and reduce the chances of false positives or negatives.

**Machine learning-based content moderation:** We can consider using machine learning algorithms to automatically moderate content on your platform. This will help you to detect and remove inappropriate or offensive content more efficiently.

**Cloud-based storage:** We can consider moving your database to a cloud-based storage platform like Amazon Web Services or Microsoft Azure. This will improve the scalability, availability, and performance of your platform.

**Mobile application:** We can develop a mobile application for our social media platform. This will allow users to access your platform from anywhere and at any time.

**Two-factor authentication:** Implementing two-factor authentication for user login can significantly increase the security of our platform.

**Advanced search and filtering options:** We can add advanced search and filtering options to allow users to find posts and other users based on their preferences.

## CHAPTER 10

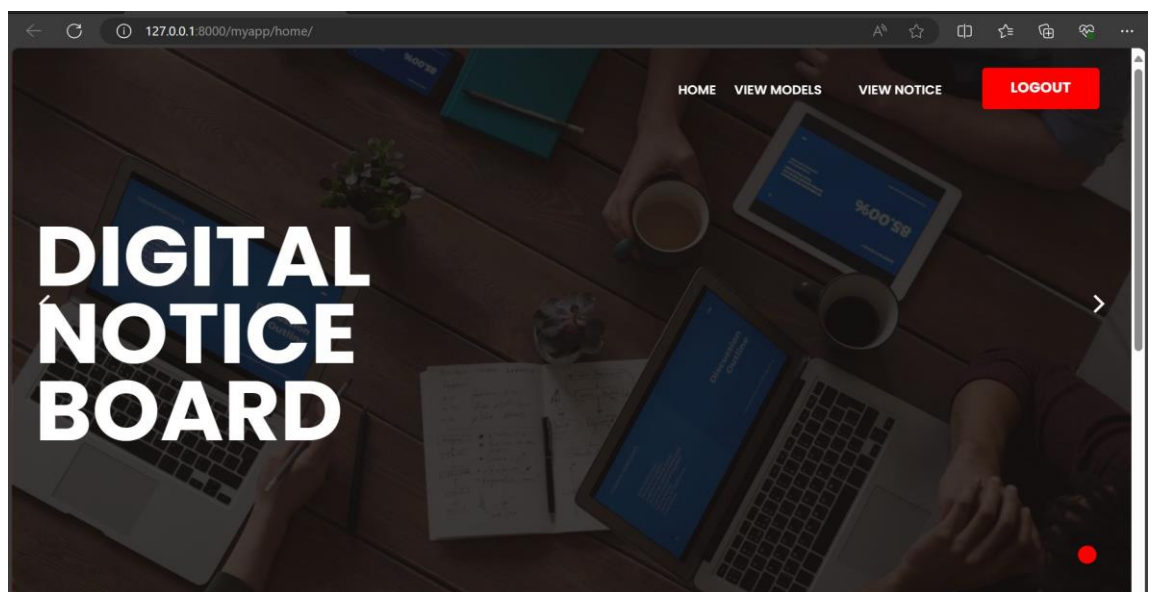
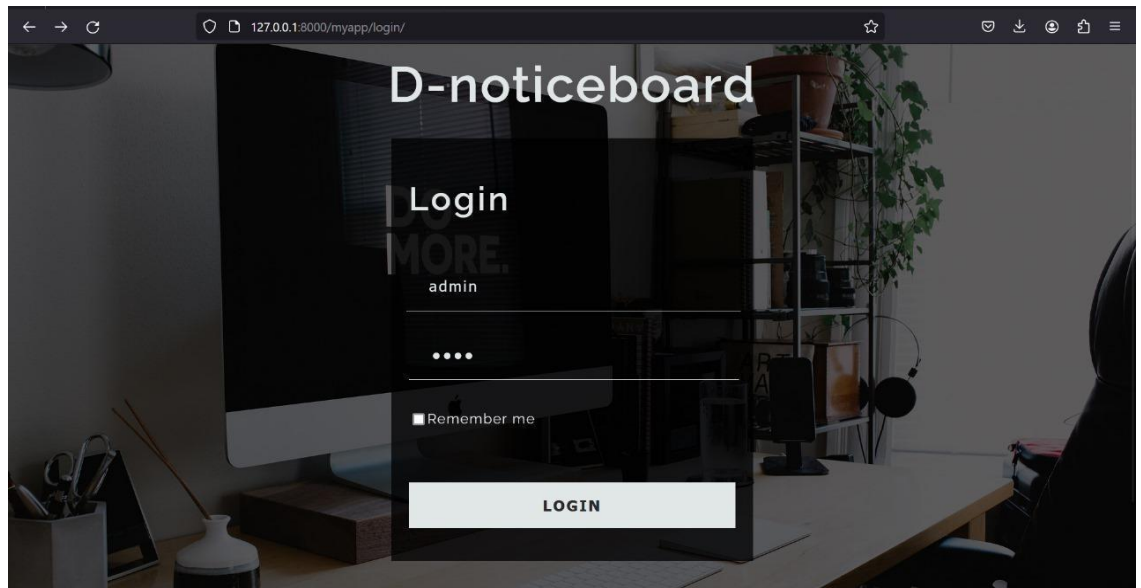
# BIBLIOGRAPHY

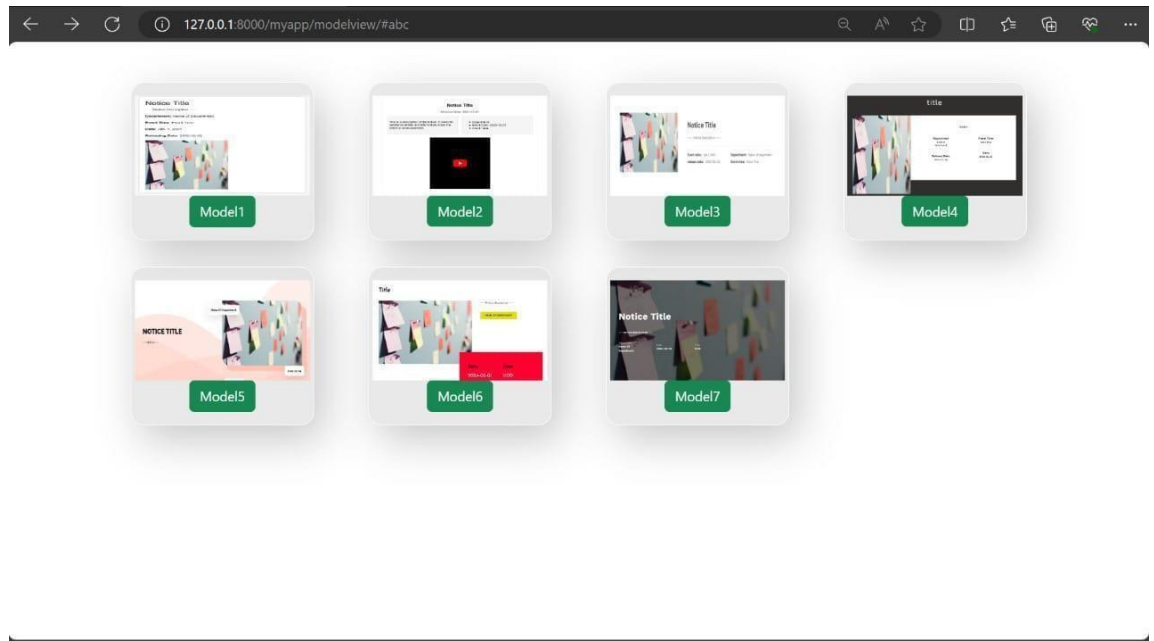
- Willian Jordan, "Python Django Web Development: The Ultimate Django Web framework guide for Beginners". Kindle Edition.
- Walter Shields, "SQL QuickStart Guide: The Simplified Beginner's Guide To Managing Analyzing, and Manipulating Data With SQL"
- "Python Django Web Development: The Ultimate Django web framework guide for Beginners". Kindle Edition.
- Steven Sanderson, "Ajax in Action".
- W3Layouts (2020). Model templates, Home templates.Retrieved from <https://w3layouts.com/>
- W3Schools. (2021). HTML CSS JavaScript. Retrieved from <https://www.w3schools.com/html/default.asp>
- Adobe Dreamweaver. (2022). Retrieved from <https://www.adobe.com/in/products/dreamweaver.html>
- SQL Database Reference. (2021). Retrieved from <https://dev.mysql.com/doc/refman/8.0/en>

## CHAPTER 11

## APPENDIX

### SCREENSHOTS:



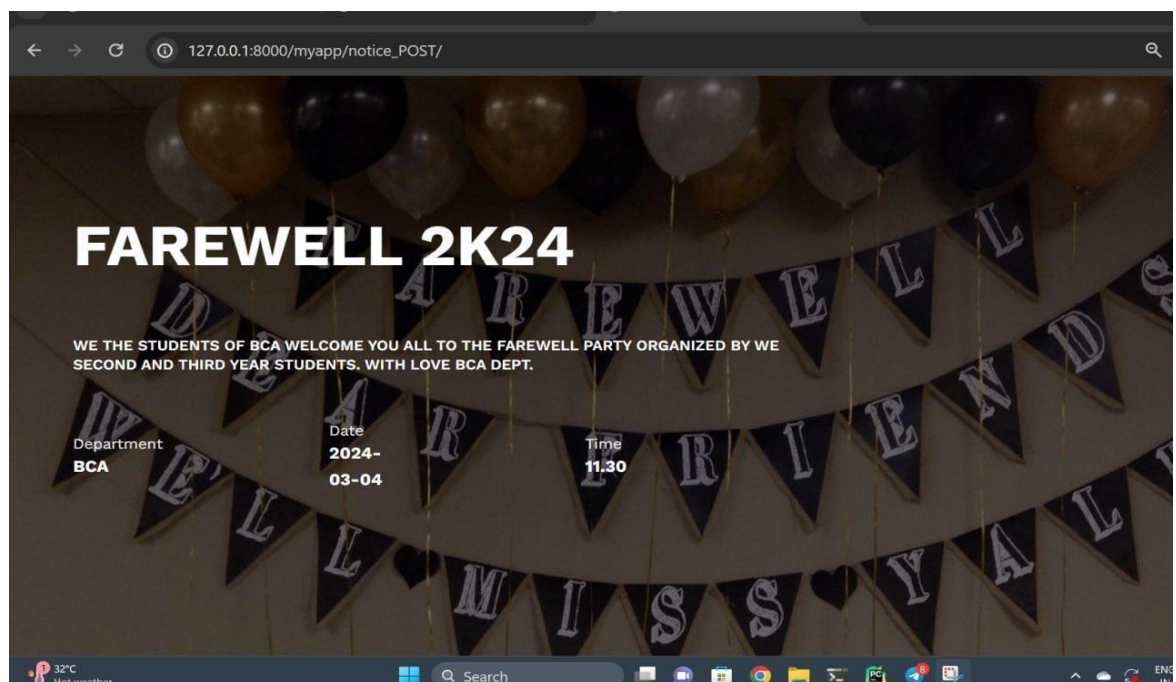
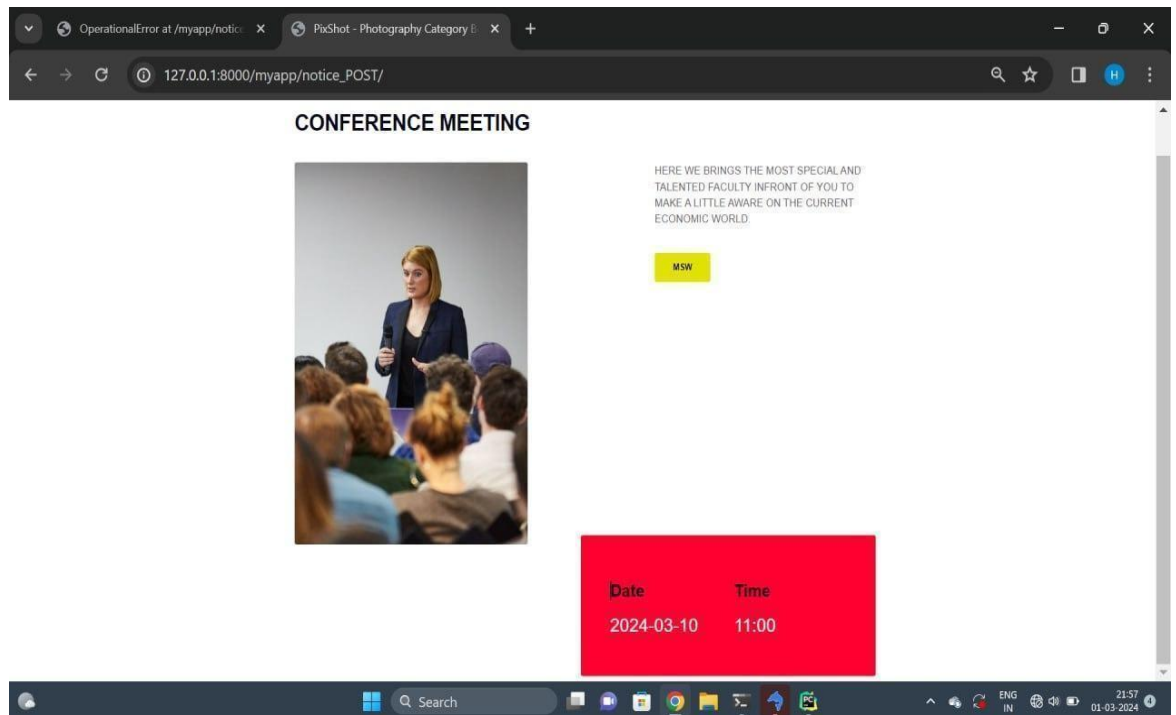



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/myapp/notice/1'. The page has a navigation bar with links for HOME, VIEW MODELS, VIEW NOTICE, and a LOGOUT button. The main form contains the following fields:

Name of Department	<input type="text"/>
Event Time	<input type="text"/>
Notice	<div></div>
Upload Image	<input type="button" value="Choose File"/> No file chosen
Title	<input type="text"/>
Date	<input type="text" value="dd-mm-yyyy"/> <input type="button" value="Calendar"/>
Releasing Date	<input type="text" value="dd-mm-yyyy"/> <input type="button" value="Calendar"/>









## FIESTA 2K24

HERE WE IDEAL CAMPUS HOST THE ANNUAL EVENT OF 2K24 – FIESTA WITH LOVE AND GRACIOUS WE ALL INVITE EVERYONE TO SUPPORT OUR STUDENTS AND BE A PART IN THEIR EVENT

---

**Event date:** 2024-03-25    **Department:** KINDER GARDEN  
**Release date:** 2024-03-27    **Event time:** 3:00