Chapter1
Introduction

1. INTRODUCTION

1.1 About the Organization:

1.1.1 Organization Name: IGKV (Indira Gandhi Krishi Vishwavidyalaya)

Indira Gandhi Krishi Vishwavidyalaya, situated in the capital of Chhattisgarh, Raipur, is only agricultural university in state with responsibility of agricultural Education, Research and Extension of technologies. It has a prestigious and long history of services to the tribal farming community of this region. The major objectives of the university are, making provision for the education in agriculture and other allied sciences, carrying out research in agriculture and other allied sciences, undertaking field extension programs and improving the level of living of rural people. The jurisdiction of IGKV is entire state which has three distinct agro-climatic zones and has diverse conditions. The university is recognized as one of the most important biodiversity repository in the world, particularly for rice and lathyrus. The teaching is imparted in agriculture and agricultural engineering faculties through its 33 constituent and 15 affiliated colleges. There are nine research stations carrying out research in different areas, and 27 Krishi Vigyan Kendras involved in dissemination of technologies to the farming community of state.

Indira Gandhi Krishi Vishwavidyalaya, is an autonomous, nonprofit, research and educational organization working for the up-liftment of farmer's livelihood of Chhattisgarh. Education, Research and Extension are the major activities of the Vishwavidyalaya.

Mission: Improving the competitiveness of agriculture sector, by building capacity, enhancing productivity and profitability, maintaining sustainability of production and consumption systems through efficient use of natural resources and energy, and catalyzing innovation across the agriculture in order to ensure food and nutritional security, and inclusive growth.

Vision: Sustainable integrated development of agriculture for rural livelihood enhancement, generation of economic growth and opportunities in state of Chhattisgarh.

1.1.2 Department: KTRC (Knowledge & Technology Resource Centre)

The Knowledge and Technology Resource Centre (KTRC) established at Indira Gandhi Krishi Vishwavidyalaya, Raipur dedicated to empowering students, farmers, and communities across Chhattisgarh with cutting-edge e-governance technology. Made possible through the generous funding support from RKVY, Government of India, this center is poised to revolutionize the dissemination of knowledge and technological advancements in the agricultural domain.

At KTRC, we have harnessed the power of digital imovation to create an all-encompassing platform for knowledge sharing and skill development. The centre boasts a state-of-the-art Digital KISAN Studio with an advanced control room, where a diverse array of videos are crafted and seamlessly broadcasted in real-time. These videos are tailor made to cater to the specific needs of both university students and farmers, enabling them to access vital information, best practices, and expert insights at their convenience.

One of our core strengths lies in the Complete Software Development Room, where a team of developers works tirelessly to create and refine various e-governance software solutions. These solutions are custom-designed to serve the unique requirements of the university and the farming community, streamlining processes, enhancing productivity, and ensuring seamless knowledge dissemination,

Ensuring a seamless digital ecosystem, our centre features a robust Network and Server Room, expertly managed to ensure smooth operations and uninterrupted access to online resources. Moreover, we have set up a dedicated Video Conferencing Room and Mini Theatre, facilitating seamless interaction and knowledge exchange among faculty, students, and farmers across districts of Chhattisgarh and national level. With our SMART Classrooms, faculty members can connect with students in remote colleges, transcending geographical barriers and fostering an inclusive learning environment.

The Knowledge and Technology Resource Centre at Indira Gandhi Agriculture University is not just a physical space; it symbolizes a vision of progress, innovation, and collaboration. We are committed to nurturing a culture of continuous learning, knowledge sharing, and digital transformation, ensuring that the agricultural community of Chhattisgarh can thrive in an increasingly digital world.

1.2About the Project:

Project Title: Uni-Connect WEB PORTAL

Uni-Connect is an advanced University Network Management System designed to streamline and secure the digital infrastructure of educational institutions. It serves as a centralized platform for managing all aspects of campus network operations, including user access control, bandwidth monitoring, device management, and network security. Tailored specifically for university environments, Uni-Connect supports real-time data traffic analysis, automated alerts, and detailed logging to ensure consistent network performance and quick issue resolution.

The system is built with scalability and ease of use in mind, allowing IT administrators to adapt to growing demands and evolving technologies. Whether managing hundreds or thousands of connected devices, Uni-Connect provides the tools needed to maintain a reliable, efficient, and secure digital ecosystem that supports both academic and administrative functions.

Chapter 2

System Analysis

SYSTEM ANALYSIS

System analysis is the first stage according to the System Development Life Cycle model. This system analysis is a process that starts with the analyst. Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. One aspect of analysis is defining the boundaries of the system and determining whether a candidate system should consider other related systems. During analysis, data are collected on the available files, decision points, and transaction handled by present system. Logical system models and tools that are used analysis. Training, experience and common sense are required for collection of the information needed to do the analysis.

Proposed System

The proposed system, **Uni-Connect** , is a centralized and intelligent network management platform specifically designed for university campuses. The aim is to replace fragmented, manual, and inefficient network monitoring methods with an integrated solution that enhances performance, control, and security across the institution's digital infrastructure.

- Store multiple IP addresses for a single device.
- Keep track of IP assignment history (optional)
- Allow one device to have both IPv4 and IPv6

Key Features of the Proposed Uni-Connect:

1. **Device Configuration Management**

- Store device configuration backups (for switches ,Routers)
- Track changes in configuration over time.
- Optionally auto-backup configuration via SNMP or SSH.

2 .IP Address Management (IPAM)

- IP pool/subnet management
- Track which IPs are free ,reserved or assigned
- Detect IP conflicts or duplicates

3. **Inventory Management**

- Add asset-level details:
    - Serial number
    - Purchase date
    - Warranty info
    - Vendor contact
- Download full inventory reports

4. Dashboard & Stats

- Summary of:
  - Total devices
  - Online/Offline count
  - Top device types
- Quick alerts for any changes

5 .Device Discovery & Monitoring

- Auto-scan devices using **Ping, SNMP, and** ARP across all IP ranges.
- Show **device online/offline status** in real-time.
- Schedule periodic network scans.

Advantages of the System

□ **Centralized Network Control**

Uni-Connect provides a unified dashboard that allows network administrators to manage all network activities from a single point, improving efficiency and visibility across the entire university infrastructure.

□ **Improved Security**

With features like user authentication, access control, and real-time alerts, Uni-Connect enhances network security by identifying unauthorized access attempts and potential threats instantly.

□ **Real-Time Monitoring**

Administrators can monitor network performance in real-time, including bandwidth usage, device connections, and traffic flow, enabling faster response to issues and better resource management.

□ **Efficient Bandwidth Management**

The system allows prioritized bandwidth allocation, ensuring essential academic services like online classes, research portals, and administrative systems function smoothly even during peak usage.

- **Scalability and Flexibility**

Uni-Connect is built to support future growth, allowing easy integration of new departments, campuses, or technologies without requiring major system overhauls.

- **Reduced Downtime and Faster Troubleshooting**

With automated issue detection and detailed system logs, administrators can quickly pinpoint and resolve network problems, minimizing service interruptions.

- **Enhanced User Experience**

Students, faculty, and staff benefit from a stable and responsive network, enabling smooth access to learning platforms, communication tools, and administrative services.

- **Audit and Compliance Support**

The system maintains logs and reports of user activity and network performance, supporting IT audits and helping meet compliance requirements for data and network usage.

- **Cost-Effective Management**

By automating tasks and optimizing network performance, Uni-Connect reduces the need for manual intervention and lowers the operational costs associated with network maintenance.

1.

Chapter 3

System Planning

SYSTEM PLANNING

System planning is a critical phase in the development of the Uni-Connect Web Portal. It involves identifying the project goals, determining system requirements, defining the scope, and preparing for implementation. Proper planning ensures that the system meets the needs of the university ,and manager while remaining scalable and secure.

3.1.1 Project Estimation: Accurate cost and effort estimation will be conducted for resource allocation and effective planning.

3.1.2 Work Planning and Scheduling: Meticulous work planning will ensure timely project completion with a well-defined timeline, milestones, and task dependencies.

3.1.3 User Involvement in Requirements Definition: Stakeholder collaboration will be prioritized for gathering comprehensive project requirements.

3.1.4 Development Stages: The project will be divided into distinct stages with specific objectives for better control and progress tracking.

3.1.5 Reviews for Control and Quality: Regular project reviews will assess progress and maintain high-quality standards.

3.1.6 Extensive Testing Planning: A comprehensive testing strategy will be implements to ensure system reliability and rectify issues

3.2 Pert Chart: A PERT (Program Evaluation and Review Technique) Chart is a visual tool used in project management to schedule, organize, and coordinate activities. It breaks down tasks into a sequence, estimating time and resources required.

Key characteristics are:

Key characteristics are:

1. **Decision Support:** PERT aids in strategic decision-making during project planning and execution.

2. **Comprehensive Planning:** It ensures thorough planning by identifying all tasks and defining the project scope.

3. **Optimized Resource Utilization:** PERT helps allocate resources efficiently, Page 11 minimizing wastage.

4. **Time Network Analysis:** It calculates the critical path and start/finish times for tasks to meet deadlines.

5. **Structured Reporting:** PERT provides a clear format for reporting project progress and milestones.

6. **Risk Management:** Incorporating risk management strategies allows proactive contingency planning.

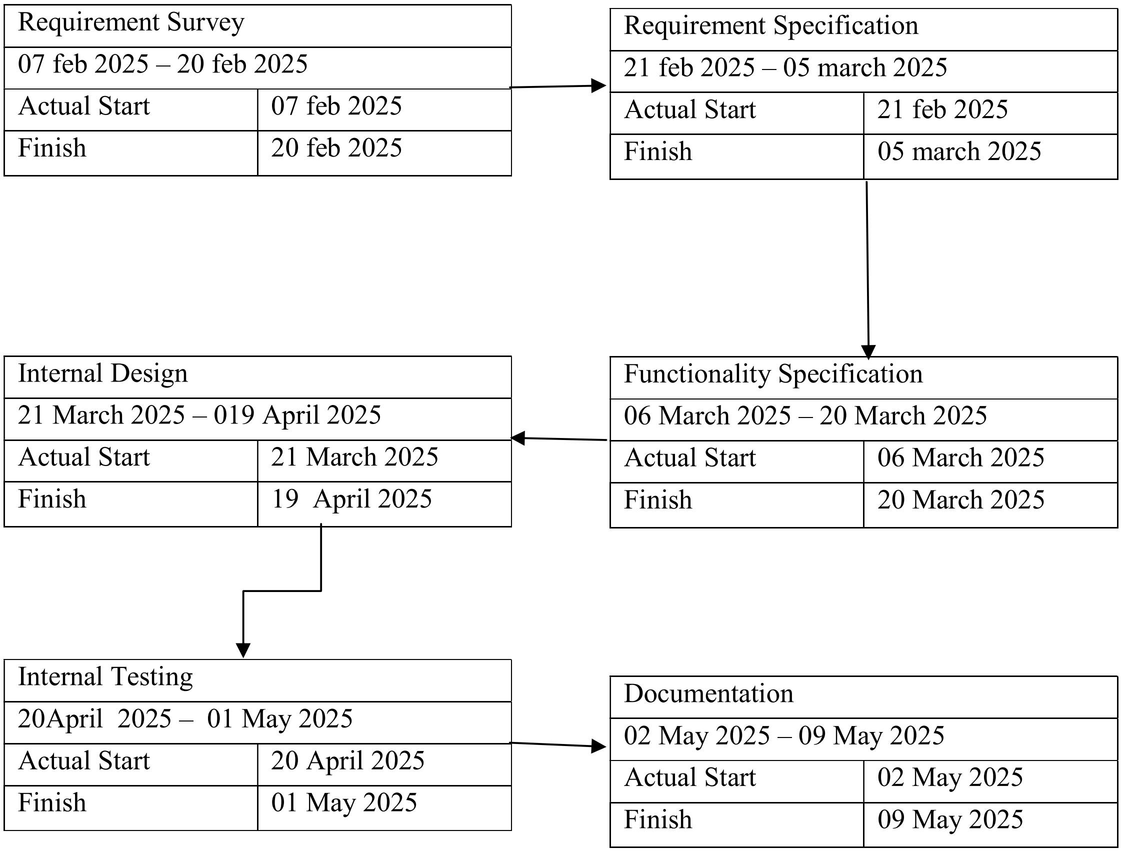
7. **Interdisciplinary Collaboration**: PERT fosters collaboration and understanding among team members.

8. **Real-Time Updates:** Digital PERT charts allow for dynamic real-time project tracking.

3.2.1 Objectives of the System

- Device Information Management
- Tagging and Grouping
- Geo-Mapping/Floor Plan View
- Location Mapping
- Device and IP Mapping (One-to-many)

PERT CHART :



Requirement Survey	
07 feb 2025 – 20 feb 2025	
Actual Start	07 feb 2025
Finish	20 feb 2025

Requirement Specification	
21 feb 2025 – 05 march 2025	
Actual Start	21 feb 2025
Finish	05 march 2025

Functionality Specification	
06 March 2025 – 20 March 2025	
Actual Start	06 March 2025
Finish	20 March 2025

Internal Design	
21 March 2025 – 019 April 2025	
Actual Start	21 March 2025
Finish	19 April 2025

Internal Testing	
20April 2025 – 01 May 2025	
Actual Start	20 April 2025
Finish	01 May 2025

Documentation	
02 May 2025 – 09 May 2025	
Actual Start	02 May 2025
Finish	09 May 2025

3.3 Cost and Benefit Analysis: For "**Uni-Connect**", Cost/benefit analysis gives a picture of the various costs, benefits, and rules associated with each alternative system. In developing cost estimates for a system, we need to consider several cost elements. Among them are hardware, personal, facilities, operating. Hardware cost relates to the actual purchase or lease of the computer and peripherals. Costs incurred during the development of a system are one-time costs and are labeled developmental costs. Once the system is installed the cost of operating and maintaining the system become a recurring cost.

Facility costs are expenses incurred in the preparation of the physical site where the application or the computer will be in operation.

Operating costs include costs associated with the day-to-day operation of the system. One approach is to treat operating costs like overhead. Another approach is to charge each authorized user for the amount of processing the request from the system.

A system is also expected to provide benefits. The first task is to identify each benefit and turns to assign a monetary value to it for cost/benefit analysis. Benefits may be tangible and intangible, direct or indirect.

Terms:

1. Tangible or Intangible Costs

Tangibility refers to the ease with which costs or benefits can be measured. An outlay of cash for a specific item or activity is referred to as a tangible cost. The purchase of hardware or software personnel training and employee salaries are examples of tangible costs. They are readily identified and measured. Costs that are known to exist but whose financial value cannot be accurately measured are referred to as intangible costs. For example, employee movable problems caused by a new system or lowered company image are an intangible cost.

Benefits

Tangible benefits such as completing jobs in fewer hours or producing reports with no errors are quantifiable. Intangible benefits such as more satisfied customers or an improved corporate image are not easily qualified. Both tangible and intangible costs and benefits, however, should be considered in the evaluation process.

2. Direct or indirect costs

Direct costs are those with which a rupee's figure can be directly associated in a project. They are applied directly to the operation. For example, the purchase of a box of diskettes for Rs.2800 is a direct cost because the diskettes can be associated with the rupees expanded. Indirect costs are the results of operations that are not directly associated with a given system or activity. They are often referred to as overhead. A system that reduces overhead realizes savings.

Benefits

Direct benefits also can be specifically attributable to a given project. For example, a new system than can handle 25 percent no transaction per day is a direct benefit. Indirect benefits are realized as a by-product of another activity or system. For example, a proposed safe deposit billing system that provides profits showing vacant boxes by sizes, location, and price, will help management decide on how much advertising to do for box rental.

3. Fixed or Variable Costs

Fixed costs are sunk costs. They are straight-line depreciation of hardware, exempt employee salaries and insurance. In contrast, variable costs are incurred on a regular basis. They are usually proportional to work volume and continue as long as the system is in operation. For example, the costs of computer forms vary in proportion to the amount of processing or the length of the reports required.

Benefits

Fixed benefits are also constant and do not change. For example, decrease in the number of personnel by 20 percent resulting from the use of a new computer. In this project, cost is incurred in terms of time consumed, electricity used etc.

Costs and benefits are as follows:

Cost

Hardware Cost: Single computer system is used in the development of an application. All hardware parts are working well and its quality is perfect**.**

Personnel Cost: Staff is not required. An administrator will be responsible for maintaining the system and its records. So, the developer is responsible for an extra cost.

Facility Cost: Electricity is being consumed in developing this application.

Benefits

➢ This application leads to less time-consuming.

➢ Our database can store a large amount of data can be stored serially and accessed frequently.

➢ Staff reduction-only single person is required which will act as an administrator to maintain the system

Chapter 4

Feasibility Study

Feasibility Study

Feasibility study is an important phase in the software development process. It enables the developer to have an assessment of the product being developed. It refers to the feasibility study of the product in terms of outcomes of the product, operational use and technical support required for implementing it.

One's the need of system requirement has been defined. The user may authorize the system analyst to create a more detailed for better understanding the opportunities and limitation associated with proposed project. Feasibility study guides the user to determine the process of the project. Feasibility study also guide user to determine the risk associated with project. The feasibility study includes Technical, Economic and Operational, the result of the study is combined known as Feasibility Study.

4.1 Technical Feasibility: We can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already.

The technical issues raised during the feasibility stage of the investigation are:

1. Does the necessary technology exist (can it be acquired) to do what is suggested?

2. Does the proposed equipment have the technical capacity to hold the data required to use the new system?

3. Will the proposed system and components provide adequate responses to queries, regardless of the number or location of users?

4. Can the system be expanded, if developed?

5. Are there technical guarantees of accuracy, reliability, ease of access and data security?

4.2 Economic Feasibility: Economic feasibility is the most frequently used methods for evaluating the effectiveness of a candidate system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with cost. The result of comparison is found and changed

if needed. If benefits outweigh costs, then the decision is made to design and implement the system. Otherwise, further justification or alternation in the proposed system will have to be made if it is to have a chance of being approved. As we are developing a completely new system the cost is on the higher side. The implementation costs involve the installation of a new hardware and software as well as the cost of hosting the website on the internet. Maintenance of the system is much costly. Training for the operating personnel is also expected to be by the people who have never been initialized to operating a computerized system.

In performing cost and benefit analysis it is important to identify cost and benefits factors. Cost and benefits can be categorized into the following categories:

1) Development cost: A Development cost is the costs that are incurred during the development of the system. It is one time investment.

2) Operating cost: Operating cost are the expenses required for the day to-day running of the system. As operating cost are wages, supplies and overheads.

3) Hardware/Software cost: It includes the cost of purchasing or leasing of computers and its peripherals. Software costs involve required software cost.

4) Supply cost: These are variable costs that are very proportionately with the amount of use of paper, ribbons, disks, and others.

Benefits:

1. Fast and easy access to all procedures and functions.
2. No need for large storage spaces sized of rooms for storing the cabinets because all the information about the members and other details is saved in the computer's hard disks.
3. High level of security and authentication of each and every user.
4. Reliability is increased, as backups of files, and records can be made and saved.
5. Different locations and information will be highly secure, unlike in file cabinets where entries can easily be ripped or tampered with by users.

4.3 Operational Feasibility: The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system. It is mainly related to human organizational and political aspects. The points to be considered are:

1. What changes will be brought with the system?

2. What new skills will be required?

3. Do the existing staff members have these skills?

People are inherently resistant to change and computers have been known to facilitate change. Now most people support computerized system. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a new system. Therefore, it is understandable that the introduction of the new system required special effort to educate and train the staff on way of operating system. Also required to give awareness to the customers. The staffs were not against the system; however the user would accept the concept.

Chapter 5
System Requirement Specification

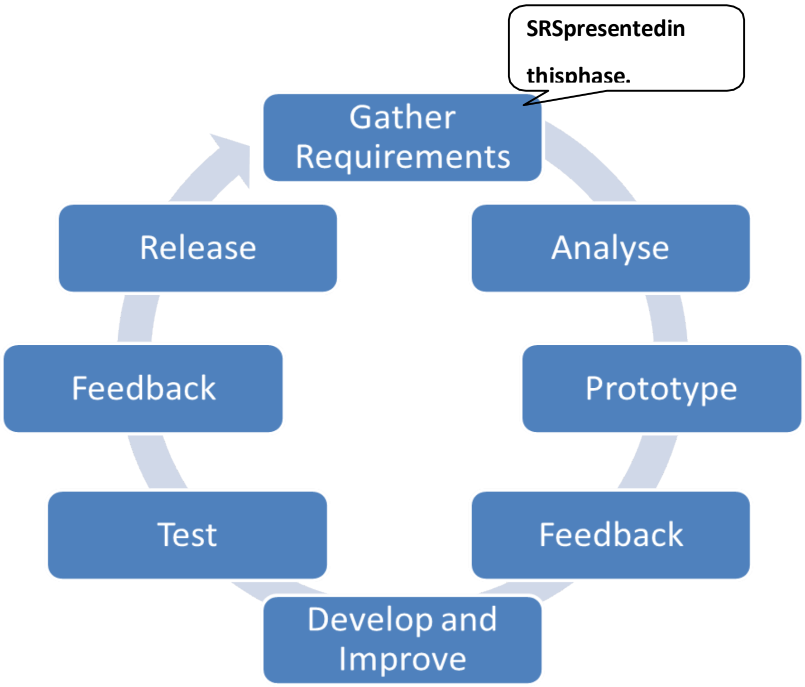
System Requirement Specifications

The requirement phase basically consists of three activities:

• Requirement Analysis

• Requirement Specification

• Requirement Validation



5.1.1 User Requirements:

5.1.2 Functional Requirements:

- Add or update device records manually.
- Group devices by:
  - Type
  - Location
  - Department
  - Custom tags
- Store and display physical location of each device:
  - Building name
  - Floor number
  - Room number

5.1.3 Non-Functional Requirements:

- System security and data privacy.
- High performance and fast load times.
- Scalability to accommodate growing users.
- Responsive and user-friendly interface.
- Backup and data recovery capabilities.

Informally these are sometimes called the "ileitis", from attributes like stability and portability. Qualities—that is non- functional requirements—can be divided into two main categories:

Execution Qualities: such as safety, security and usability, which are observable during operation (at run time).

Evolution Qualities: such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the system.

5.3 Minimum Hardware & Software Requirements:

5.3.1 Hardware Requirements: -

❖ **Processor:** Inteli3/i5 or higher (U series or higher).

❖ **RAM:** Minimum 4GB or above.

❖ **HDD:** 40 to 100 GB (Recommended) or above.

5.3.2 Software Requirements: -

❖ **Operating System:** Windows 7 or above

❖ **For front-end:** Languages: HTML, CSS, JavaScript, Type-Script.

Source-code editor: Visual Studio Code.

Framework: Angular Version 18

❖ **For Back-end: Languages:** JavaScript. Run time

Environment: Node.js.

Source-code editor: Visual Studio Code.

❖ **For Database:**

Database: Maria-DB.

Management Tool: Heidi SQL

- **Browser:** Google Chrome / Firefox / Microsoft Edge.

5.3.3 Runtime Requirement: -

- **Device:** Laptop/Desktop/Mobile phones.
- **Operating System:** Windows/Linux/Android
- **Processor:** 512 MB

5.4 Technology Description

5.4.1 Angular Framework: Angular is a popular open-source front-end web application framework developed and maintained by Google. It is written in Type-Script and allows developers to build dynamic, single-page web applications (SPAs) and progressive web apps (PWAs). Angular provides a structured and powerful framework for building client-side applications, and it follows the Model-View-Controller (MVC) architectural pattern**.**

Key features and characteristics of the Angular framework include:

1. Type-Script: Angular is built using Type-Script, which is a superset of JavaScript that adds static typing and other advanced features to JavaScript. Type-Script helps catch errors during development and provides better tooling support.

2. Component-Based Architecture: Angular applications are built using components. Components are self-contained and reusable building blocks that encapsulate the template, logic, and styles associated with a specific part of the user interface.

3. Data Binding: Angular provides powerful data binding capabilities that allow automatic synchronization of data between the component and the template, making it easier to manage the application state.

4. Directives: Angular uses directives to extend the functionality of HTML elements. Directives can be custom or built-in, and they allow developers to add behavior or manipulate the DOM elements.

5. Dependency Injection: Angular uses a robust dependency injection system, which promotes modularity and makes it easier to manage the application's dependencies and services.

6. Routing: Angular provides a powerful routing system that enables navigation within the application without requiring full page reloads.

7. Forms: Angular offers extensive support for creating forms, including template-driven forms and reactive forms, making it easier to handle form inputs and validation.

8. Testing: Angular has built-in support for unit testing and end-to-end testing, making it easier to ensure the quality and reliability of applications.

5.4.2 Node.js: Node.js is an open-source, cross-platform, server-side JavaScript runtime environment. It allows developers to execute JavaScript code on the server-side outside of a web browser. Node.js is built on the V8 JavaScript engine, which is developed by Google and used in their Chrome web browser. This engine compiles JavaScript code into machine code, providing high-performance execution.

Node.js was created by Ryan Dahl and was first released in 2009. Since then, it has gained significant popularity and has become a fundamental technology for building scalable and real-time applications

Key features and characteristics of Node.js include:

1. Asynchronous and Non-Blocking I/O: Node.js is designed with an event-driven architecture, enabling asynchronous processing and non-blocking I/O operations. This makes it well-suited for handling concurrent connections and I/O intensive tasks without getting blocked.

2. Single-Threaded Event Loop: Node.js uses a single-threaded event loop to handle multiple concurrent connections efficiently. It allows handling a large number of connections without the need for spawning new threads for each request, making it lightweight and memory-efficient.

3. NPM (Node Package Manager): Node.js comes with NPM, a powerful package manager that provides access to a vast ecosystem of open-source modules and libraries. NPM makes it easy for developers to install, manage, and share reusable code.

4. Cross-Platform: Node.js is compatible with various operating systems, including Windows, macOS, and Linux, making it highly portable and flexible.

5. Server-Side Development: Node.js is commonly used for server-side development to build web applications, APIs (Application Programming Interfaces), and backend services.

6. Full-Stack Development: With Node.js, developers can use the same language (JavaScript) for both client-side and server-side development, promoting code reuse and enhancing developer productivity

5.4.3 Maria-DB: Maria-DB is an open-source, relational database management system (RDBMS) and a popular alternative to My-SQL. It was developed by the original creators of My-SQL after My-SQL was acquired by Oracle Corporation. Maria-DB is designed to be a drop-in replacement for My-SQL, meaning it is mostly compatible with My-SQL and can be used as a seamless replacement without requiring significant changes to existing applications or databases.

Key features and characteristics of Maria-DB include:

1. Open-Source: Maria-DB is released under the GNU General Public License (GPL), making it freely available for use, distribution, and modification.

2. Performance: Maria-DB is known for its excellent performance, scalability, and optimization features. It has a robust query optimizer and supports various storage engines, including Inno-DB, My Rocks, and Aria, among others.

3. High Availability and Replication: Maria-DB supports features like Master-Slave replication and Master-Master replication, enabling data redundancy and high availability for critical applications.

4. Security: Maria-DB emphasizes security and provides features like data encryption, secure connections using SSL/TLS, and user authentication mechanisms.

5. Compatibility: As mentioned earlier, Maria-DB is designed to be compatible with . It supports the same SQL syntax, APIs, and connectors, making it easy to migrate to Maria-DB.

6. Active Community: Maria-DB has a large and active community of developers and contributors, ensuring continuous development and improvement of the database system.

7. Storage Engines: Maria-DB supports multiple storage engines, allowing users to choose the most suitable one for their specific use case. The default storage engine, Inno-DB, is a transactional storage engine widely used for its ACID-compliance (Atomicity, Consistency, Isolation, and Durability).

8. Tools and Ecosystem: Maria-DB comes with various tools and utilities for database management, administration, and performance tuning.

5.4.4 JavaScript: JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

Key advantages of learning JavaScript:

1. JavaScript is the most popular programming language in the world and that makes it a programmer's great choice. Once you learnt JavaScript, it helps you developing great front-end as well as back-end software's using different JavaScript based frameworks like j-Query, Node.JS etc.

2. JavaScript is everywhere, it comes installed on every modern web browser and so to learn JavaScript you really do not need any special environment setup. For example, Chrome, Mozilla Firefox, Safari and every browser you know as of today, supports JavaScript.

3. JavaScript helps you create really beautiful and crazy fast websites. You can develop your website with a console like look and feel and give your users the best Graphical User Experience.

4. JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for you as JavaScript Programmer.

5. Due to high demand, there is tons of job growth and high pay for those who know JavaScript. You can navigate over to different job sites to see what having JavaScript skills looks like in the job market.

6. Great thing about JavaScript is that you will find tons of frameworks and Library.

5.4.5 J-Query: j-Query is a cross-platform JavaScript library designed to simplify the client side scripting of HTML. j-Query is the most popular JavaScript library in use today, with installation on 65% of the top 10 million highest-trafficked sites on the Web. J-Query is free, open-source software licensed under the MIT License.

5.4.6 SCSS

SCSS stands for "Sassy CSS," and it is a popular extension of CSS (Cascading Style Sheets). SCSS is a superset of CSS, which means that all valid CSS code is also valid SCSS code. It introduces additional features and syntax that make writing and organizing CSS styles more powerful and maintainable.

5.4.7 Bootstrap

Bootstrap is a free and open-source front end development framework for the creation of websites and web apps. The Bootstrap framework is built on HTML, CSS, and JavaScript (JS) to facilitate the development of responsive, mobile-first sites and apps. Responsive design makes it possible for a web page or app to detect the visitor’s screen size and orientation and automatically adapt the display accordingly; the mobile first approach assumes that smart phones, tablets and task-specific Mobile apps are employees' primary tools for getting work done and addresses the requirements of those technologies in design.

5.4.8 HTML

HTML is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current version of the HTML standard.HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a candidate for cross-platform mobile applications, because it includes features designed with low-powered devices in mind.

Chapter 6

System Design

System Design

System design is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. “how to implement?”
It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate.
In this phase, the complex activity of system development is divided into several smaller sub-activities, which coordinate with each other to achieve the main objective of system development.

6.1 PROTOTYPE MODEL: The concepts of software engineering have been implemented successfully and uniformly throughout the system. The performance of the integrated system will be uniform. For building this project, we followed Prototype Model as the requirements of this project are completed analyzed at the beginning of the project itself. The prototyping model is a systems development method (SDM) in which a prototype (an early approximation of a final system or product) is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved.

There are several steps in the Prototyping Model:
1. The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the departments or aspects of the existing system.

2. A preliminary design is created for the new system.

3. A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product, change approval process. For developing my project” Project Name”, firstly I gather some requirement of this project, then with this initial requirement I have start the project designing and a preliminary design of project is construct.

A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product. After making this prototype model and I present this model to the Project Name has.

Evaluate this first prototype, noting its strengths and weaknesses, what needs to be added, and what should to be removed. I collects and analyzes the remarks from the users. Then I modified the first prototype based on the comments supplied by the Project Name, and a second prototype of the new system is constructed. The second prototype is evaluated in the same manner as was the first prototype by Project Name.

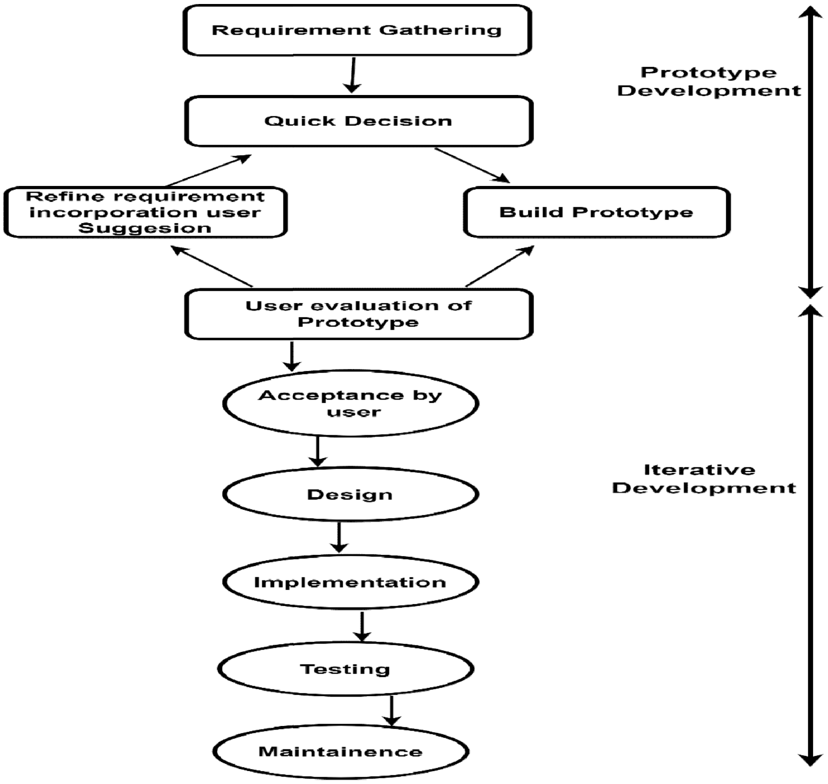


Fig : Prototype Model

Advantages:

1. Estimates (i.e. Budget, schedule, etc.) become more realistic work progresses, because important issues are discovered.

2. It is more able to cope with the software development generally entails.
Software engineers (who can get restless with protected design processes) can get their hands in and start working on a project earlier.

6.2 Database Design: A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected. The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS. In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- **Data Integrity**
- **Data independence**

NORMALIZATION

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updating, deletion anomalies. Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation. Insertion anomaly: Inability to add data to the database due to absence of other data. Deletion anomaly: Unintended loss of data due to deletion of other data. Update anomaly: Data inconsistency resulting from data redundancy and partial update Normal Forms: These are the rules for structuring relations that eliminate anomalies.

FIRST NORMAL FORM:

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

SECOND NORMAL FORM:

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules. • Primary key is a not a composite primary key. • No non key attributes are present. • Every non key attribute is fully functionally dependent on full set of primary key.

THIRD NORMAL FORM:

A relation is said to be in third normal form if their exits no transitive dependencies. Transitive Dependency: If two non-key attributes depend on each other as well as on the primary key then they are said to be transitively dependent. The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

DATA DICTIONARY: A data dictionary contains:

• The definitions of all schema objects in the database (tables, views, indexes, clusters, synonyms, sequences, procedures, functions, packages, triggers, and so on)

• How much space has been allocated for, and is currently used by, the schema objects

• Default values for columns.

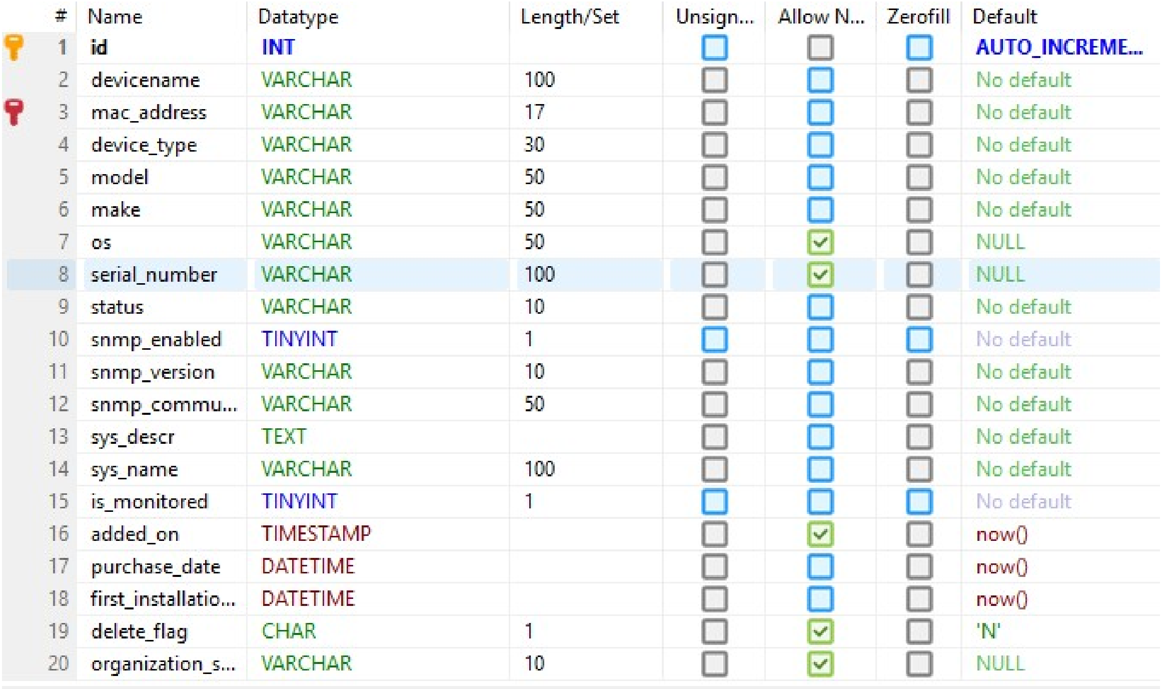
• Integrity constraint information.

• Privileges and roles each user has been granted.

• Auditing information, such as who has accessed or updated various schema objects.

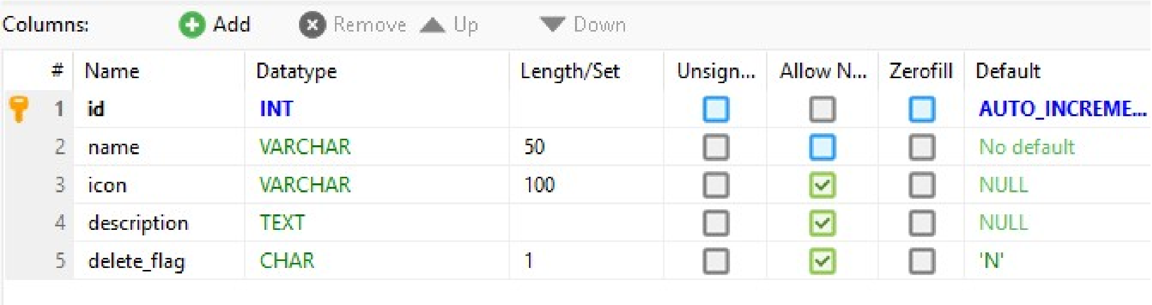
Tables Structure:

1.device_info Table



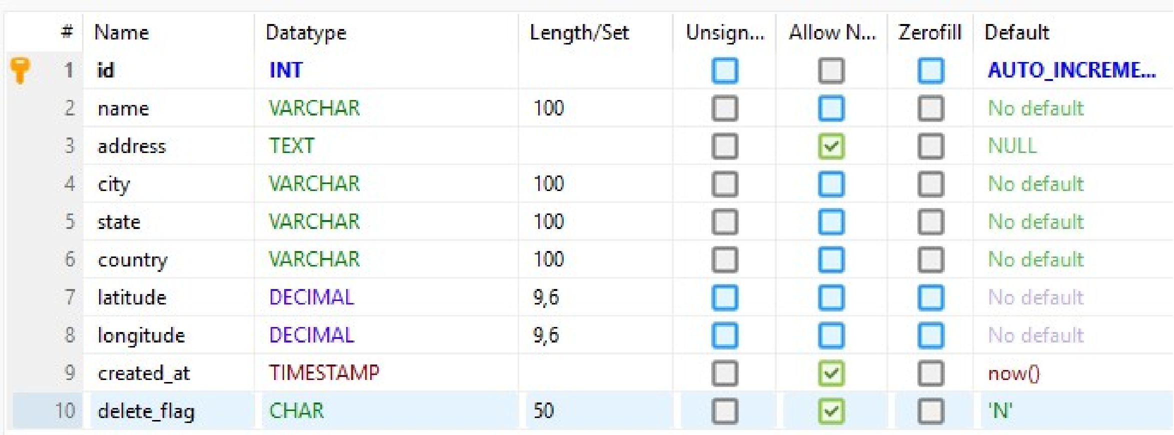
	#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
🔑	1	id	INT		□	□	□	AUTO_INCREME...
	2	devicename	VARCHAR	100	□	□	□	No default
🔑	3	mac_address	VARCHAR	17	□	□	□	No default
	4	device_type	VARCHAR	30	□	□	□	No default
	5	model	VARCHAR	50	□	□	□	No default
	6	make	VARCHAR	50	□	□	□	No default
	7	os	VARCHAR	50	□	☑	□	NULL
	8	serial_number	VARCHAR	100	□	☑	□	NULL
	9	status	VARCHAR	10	□	□	□	No default
	10	snmp_enabled	TINYINT	1	□	□	□	No default
	11	snmp_version	VARCHAR	10	□	□	□	No default
	12	snmp_commu...	VARCHAR	50	□	□	□	No default
	13	sys_descr	TEXT		□	□	□	No default
	14	sys_name	VARCHAR	100	□	□	□	No default
	15	is_monitored	TINYINT	1	□	□	□	No default
	16	added_on	TIMESTAMP		□	☑	□	now()
	17	purchase_date	DATETIME		□	□	□	now()
	18	first_installatio...	DATETIME		□	□	□	now()
	19	delete_flag	CHAR	1	□	☑	□	'N'
	20	organization_s...	VARCHAR	10	□	☑	□	NULL

2. device_group Table



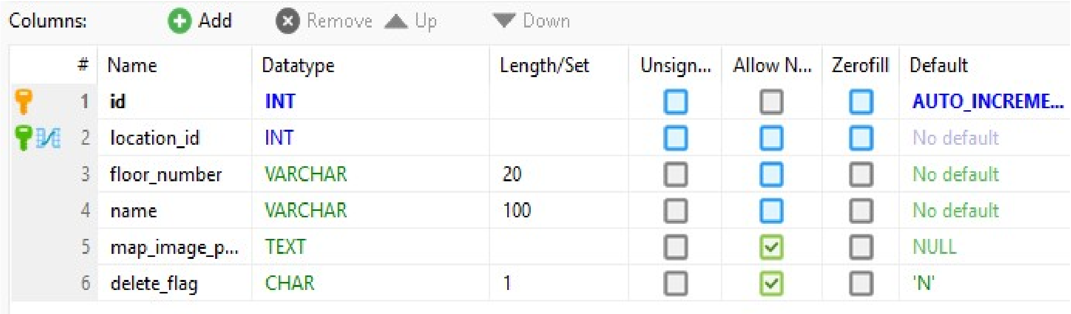
	#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
🔑	1	id	INT		□	□	□	AUTO_INCREME...
	2	name	VARCHAR	50	□	□	□	No default
	3	icon	VARCHAR	100	□	☑	□	NULL
	4	description	TEXT		□	☑	□	NULL
	5	delete_flag	CHAR	1	□	☑	□	'N'

3. locations Table



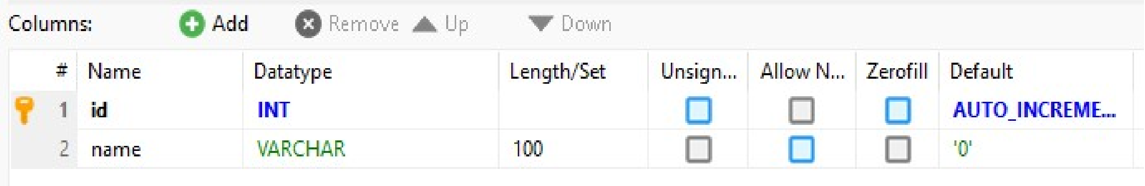
	#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
🔑	1	id	INT		□	□	□	AUTO_INCREME...
	2	name	VARCHAR	100	□	□	□	No default
	3	address	TEXT		□	☑	□	NULL
	4	city	VARCHAR	100	□	□	□	No default
	5	state	VARCHAR	100	□	□	□	No default
	6	country	VARCHAR	100	□	□	□	No default
	7	latitude	DECIMAL	9,6	□	□	□	No default
	8	longitude	DECIMAL	9,6	□	□	□	No default
	9	created_at	TIMESTAMP		□	☑	□	now()
	10	delete_flag	CHAR	50	□	☑	□	'N'

4. floors Table



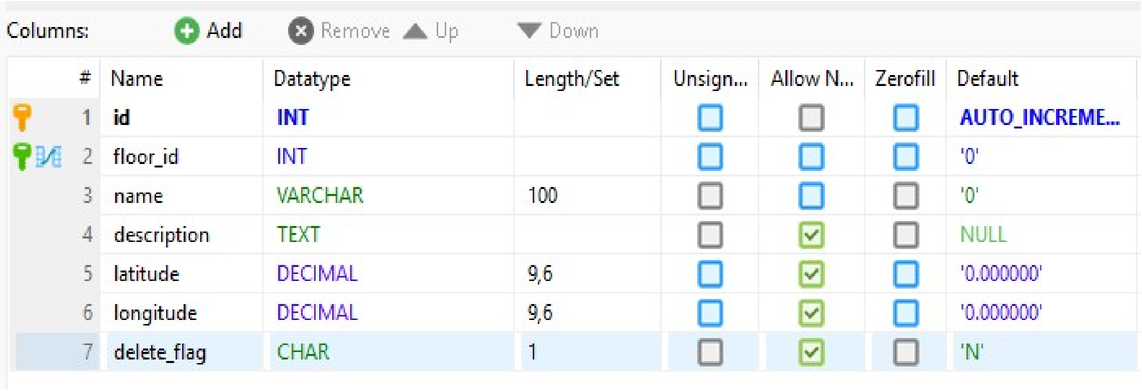
	#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
🔑	1	id	INT		□	□	□	AUTO_INCREME...
🔑	2	location_id	INT		□	□	□	No default
	3	floor_number	VARCHAR	20	□	□	□	No default
	4	name	VARCHAR	100	□	□	□	No default
	5	map_image_p...	TEXT		□	☑	□	NULL
	6	delete_flag	CHAR	1	□	☑	□	'N'

5. device_type Table



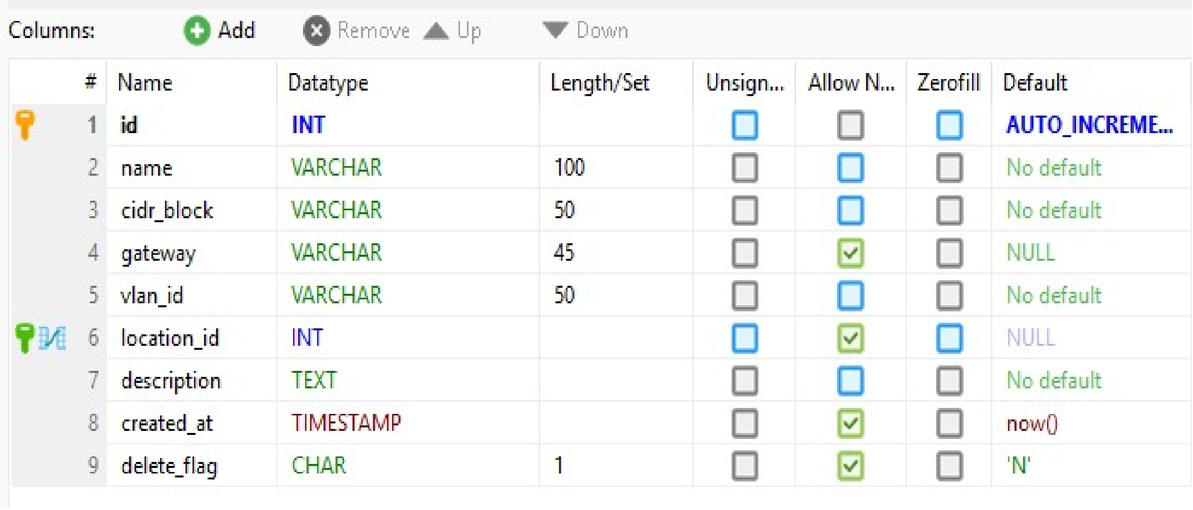
	#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
🔑	1	id	INT		□	□	□	AUTO_INCREME...
	2	name	VARCHAR	100	□	□	□	'0'

6.rooms Table



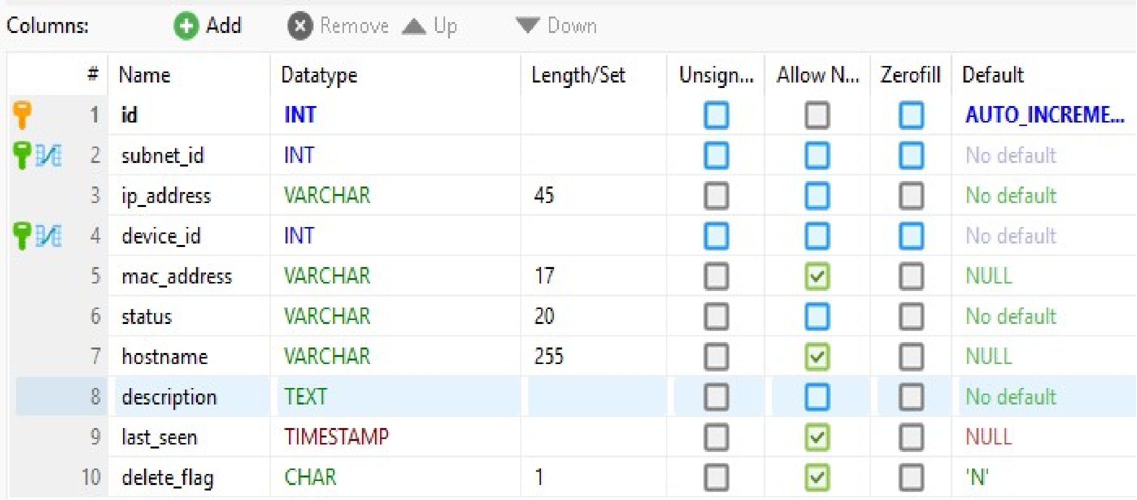
	#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
🔑	1	id	INT		□	□	□	AUTO_INCREME...
🔑	2	floor_id	INT		□	□	□	'0'
	3	name	VARCHAR	100	□	□	□	'0'
	4	description	TEXT		□	☑	□	NULL
	5	latitude	DECIMAL	9,6	□	☑	□	'0.000000'
	6	longitude	DECIMAL	9,6	□	☑	□	'0.000000'
	7	delete_flag	CHAR	1	□	☑	□	'N'

7. ip_subnets Table



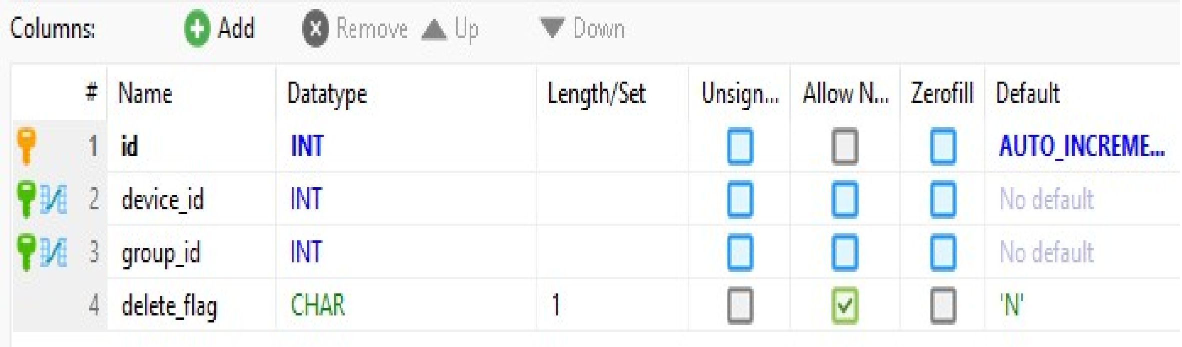
	#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
🔑	1	id	INT		□	□	□	AUTO_INCREME...
	2	name	VARCHAR	100	□	□	□	No default
	3	cidr_block	VARCHAR	50	□	□	□	No default
	4	gateway	VARCHAR	45	□	☑	□	NULL
	5	vlan_id	VARCHAR	50	□	□	□	No default
🔑	6	location_id	INT		□	☑	□	NULL
	7	description	TEXT		□	□	□	No default
	8	created_at	TIMESTAMP		□	☑	□	now()
	9	delete_flag	CHAR	1	□	☑	□	'N'

8. ip_addresses Table



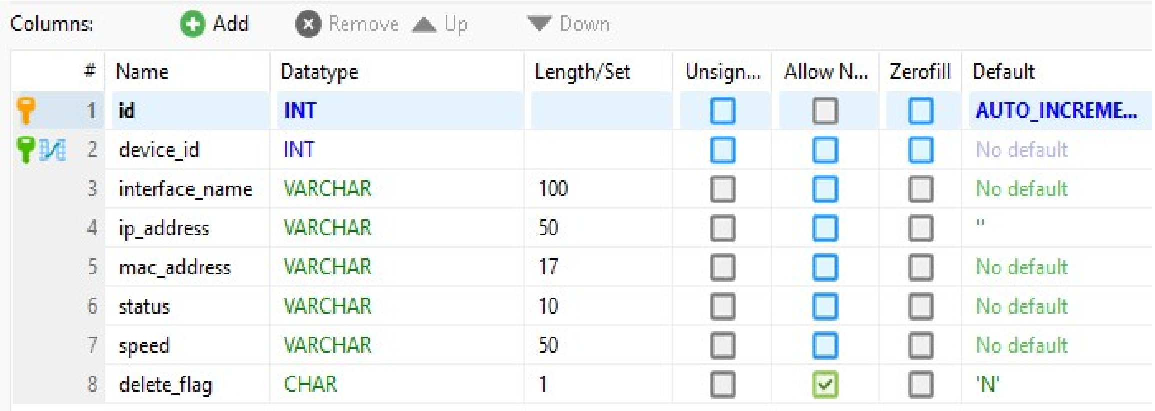
	#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
🔑	1	id	INT		□	□	□	AUTO_INCREME...
🔑	2	subnet_id	INT		□	□	□	No default
	3	ip_address	VARCHAR	45	□	□	□	No default
🔑	4	device_id	INT		□	□	□	No default
	5	mac_address	VARCHAR	17	□	☑	□	NULL
	6	status	VARCHAR	20	□	□	□	No default
	7	hostname	VARCHAR	255	□	☑	□	NULL
	8	description	TEXT		□	□	□	No default
	9	last_seen	TIMESTAMP		□	☑	□	NULL
	10	delete_flag	CHAR	1	□	☑	□	'N'

9.device_group_map Table



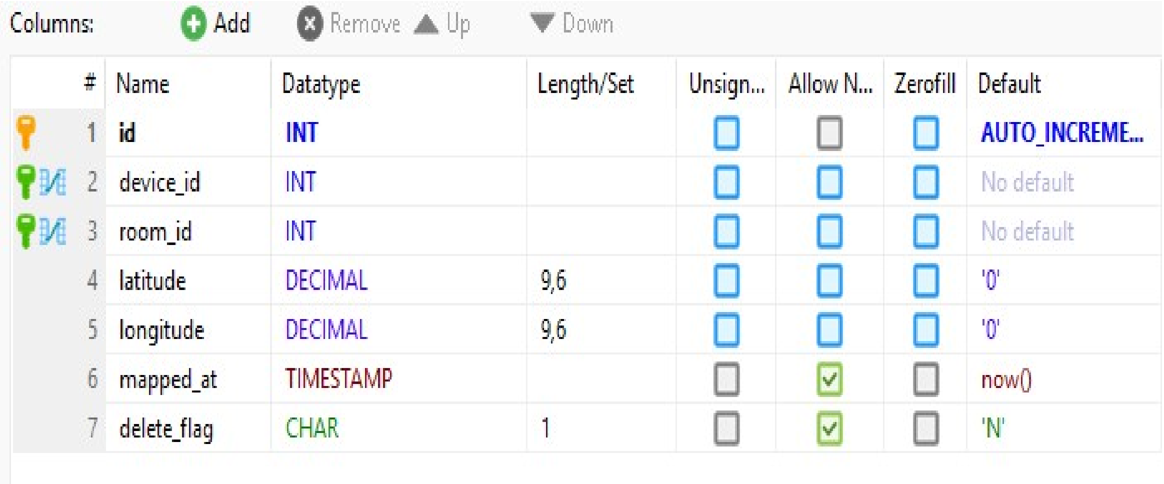
#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	id	INT		□	□	□	AUTO_INCREME...
2	device_id	INT		□	□	□	No default
3	group_id	INT		□	□	□	No default
4	delete_flag	CHAR	1	□	☑	□	'N'

10.device_interfaces Table



#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	id	INT		□	□	□	AUTO_INCREME...
2	device_id	INT		□	□	□	No default
3	interface_name	VARCHAR	100	□	□	□	No default
4	ip_address	VARCHAR	50	□	□	□	''
5	mac_address	VARCHAR	17	□	□	□	No default
6	status	VARCHAR	10	□	□	□	No default
7	speed	VARCHAR	50	□	□	□	No default
8	delete_flag	CHAR	1	□	☑	□	'N'

11.device_locations Table



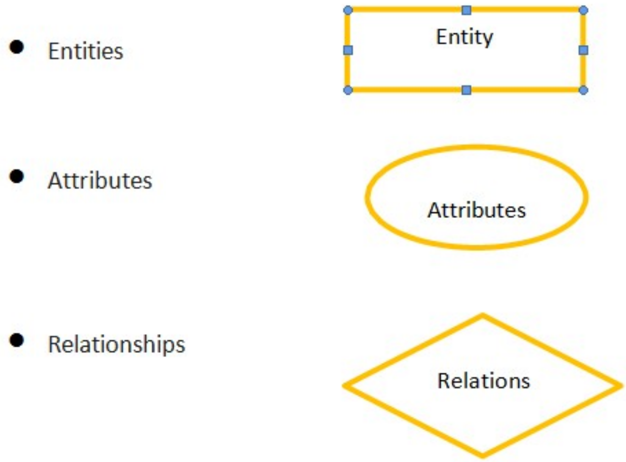
#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	id	INT		□	□	□	AUTO_INCREME...
2	device_id	INT		□	□	□	No default
3	room_id	INT		□	□	□	No default
4	latitude	DECIMAL	9,6	□	□	□	'0'
5	longitude	DECIMAL	9,6	□	□	□	'0'
6	mapped_at	TIMESTAMP		□	☑	□	now()
7	delete_flag	CHAR	1	□	☑	□	'N'

6.3 E-R Diagram: In software engineering, an entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs.

The first stage of information system design uses these models during the requirements analysis to describe information needs or the type of information that is to be stored in a database. The data modeling technique can be used to describe any ontology (i.e. an overview and classifications of used terms and their relationships) for a certain area of interest. In the case of the design of an information system that is based on a database, the conceptual data model is, at a later stage (usually called logical design), mapped to a logical data model, such as the relational model; this in turn is mapped to a physical model during physical design.

Components of the ER Diagram

This model is based on three basic concepts:



ER Diagram:-

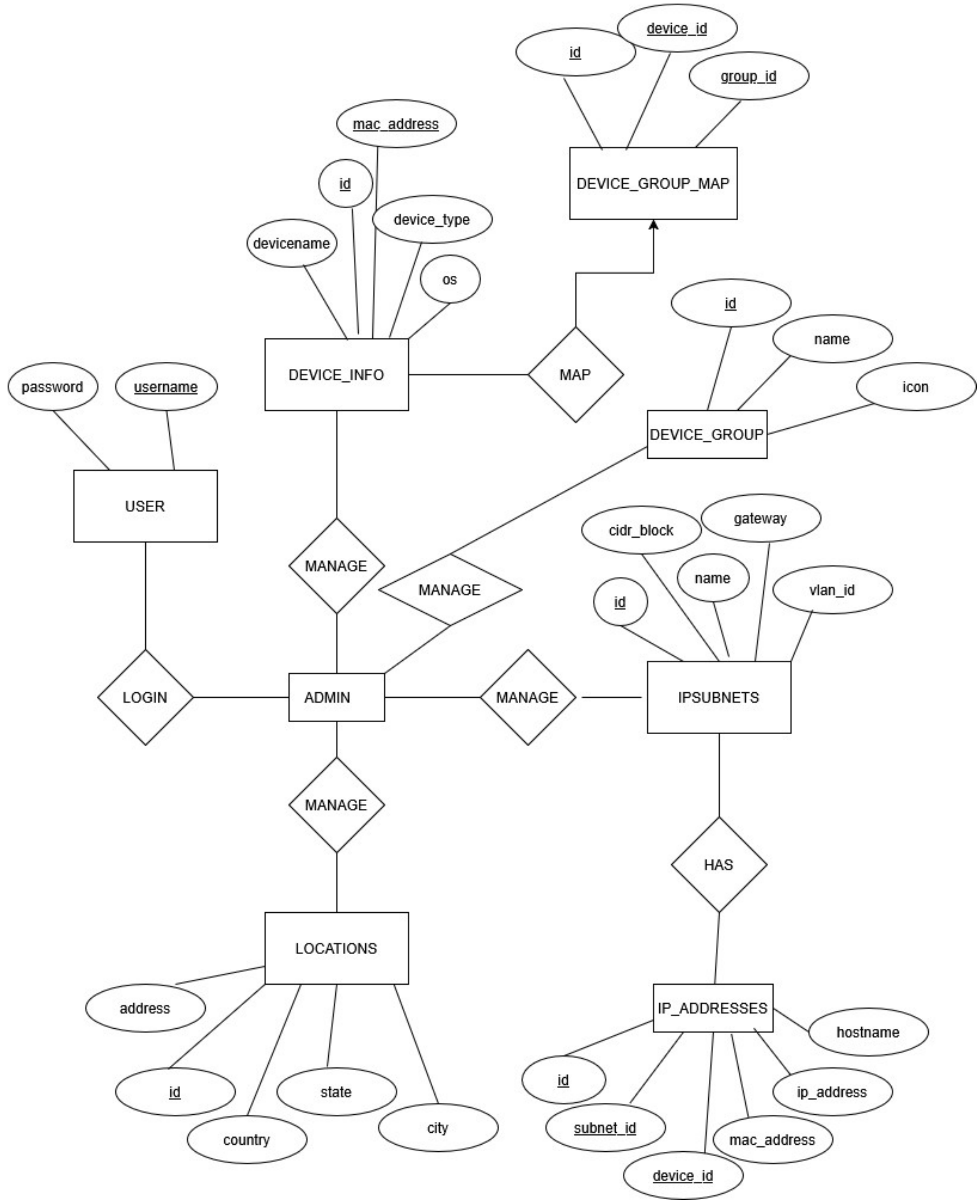


Diagram:- ER-Diagram of Uni-Connect

6.5 Data Flow Diagrams: A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored. The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.
Standard symbols for DFDs are derived from the electric circuit diagram analysis and are shown

Symbol	Description
→	Data Flow–Data flow are pipelines through the packets of information flow.
⬭	Process: A Processor task performed by the system.
▭	Entity: Entity is object of the system. A source or destination data of a system.
═	Data Store: A place where data to be stored.

Circle: A circle (bubble) shows a process that transforms data inputs into data outputs.

Data Flow: A curved line shows the flow of data into or out of a process or data store.

Data Store: A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements.

Source or Sink: Source or Sink is an external entity and acts as a source of system inputs or sink of system outputs.

6.5.1 Context Level DFD (Level 0): This diagram shows that UNI Connect WEB PORTAL consist actor ADMIN
Each actor requests login action to enter the system and does their tasks as shown

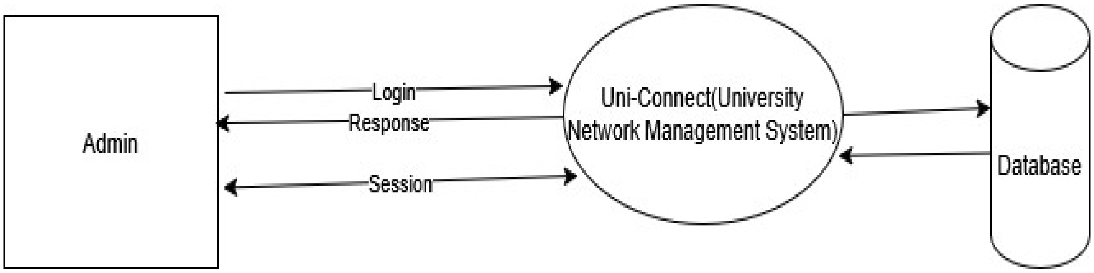


Figure (6.5.1) zero level DFD Uni-connect web Portal

6.5.2 Data Flow Diagram (Level 1):-

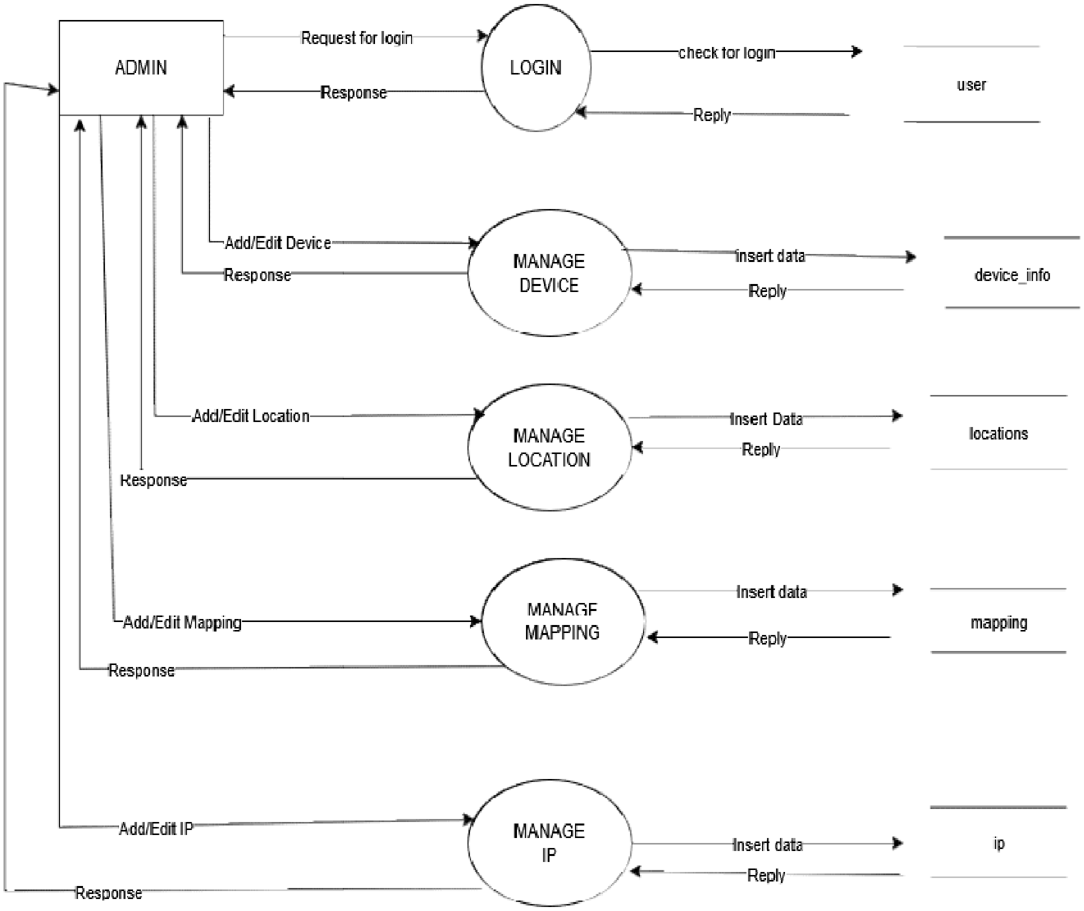


Figure (6.5.2) 1 level DFD Uni-connect web Portal

6.5.2 Data Flow Chart:-

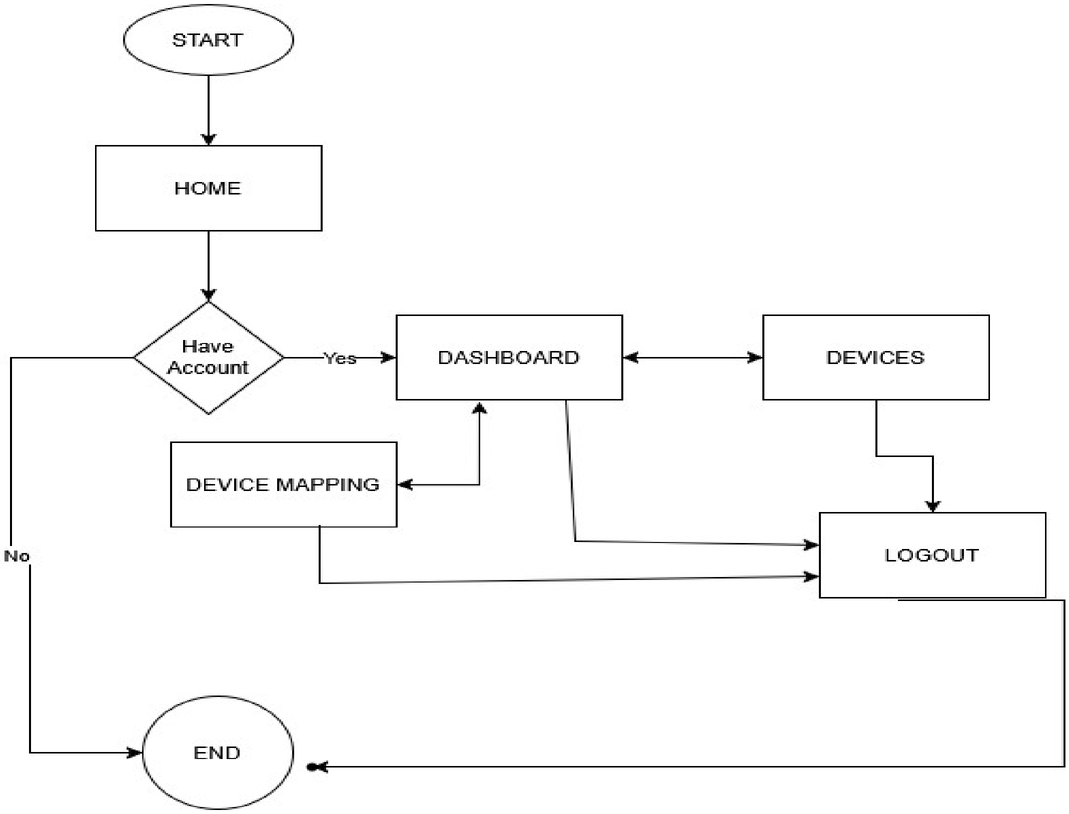


Figure (6.5.1) Flow Chart of Uni-connect web Portal

Chapter 7
System Testing

System Testing

System Testing is the testing of a complete and fully integrated software product. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system. System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.
Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software and application.

White Box Testing: To follow the concept of white box testing we have tested each form we have created independently to verify that Data flow is correct. All conditions are exercised to check their validity. All loops are executed on their boundaries**.**

Black Box Testing: The Testing Method focuses on the functional Requirements of the software. Here each module will be treated as a black box that will take some input and generate output. Output for a given set of Input combinations is forwarded to other modules**.**

Unit Testing: Each module has been tested by giving different sets of inputs. The inputs are validated when accepting from the user.

Validation Testing: In the present system, validations are been written for Reference, Pay Type etc., entries and proper error messages are displayed when any validation error occurs. Validations such as a Text field should accept only Character data but no any other Characters and only Number data but no any other Characters.

Testing Strategies:

Testing is a set of activities that can be planned in advanced and conducted systematically. A strategy for software testing must accommodation low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against customer requirements. Software testing is one element of verification and validation. Verification refers to the set of activities that ensure that software correctly implements as specific function. Validation refers to a different set of activities that ensure that the software that has been built is traceable to customer requirements.
The main objective of software testing is to uncover errors. To fulfill this objective, a series of test steps unit, integration, and system tests are planned and executed. Each test step is accomplished through a series of systematic test technique that assist in the design of test cases. With each testing step, the level of abstraction with which software is considered is broadened.

UNIT TESTING:

Unit testing is usually performed by the developer who writes different code units that could be related or unrelated to achieve a particular functionality. Here individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

The unit-testing we have is white box oriented and some modules steps are conducted in parallel.

Test No.	Test Cases	Input Value	Expected Output	Pass/Fail
1	Login with wrong User name and Password	**Username:**User1 **Password:**User1@123	Incorrect User name or Password	Pass
2	Login with Right User name and Password	User name: Right User name Password: Right Password	Login success and User redirected to the User Dashboard.	Pass
3	Select files	Select files and submit	Files uploaded	Pass

Above test cases proved that all the functions, loops, conditions are working fine in this project

System Testing: System testing can be considered as a black-box test technique. Black box Testing technique does not require internal knowledge of the code.

System testing is actually a series of different tests (i.e., performance, security, recovery) whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions.

Test No.	Test Case Objective	Actual Output	Expected Output
1	Correct Workflow?	Each module is correctly connected with each other.	Yes
2	Interface	Interface of the system is user friendly.	User Friendly
3	Bug Free?	Changes can be made easily.	Yes

Performance Testing:

➢ Better response time because website take less time to load because using of lazy loading.

➢ No refreshing the website while doing any action.

➢ Process activities like redirecting other pages and file uploading etc. on the website are tested.

➢ Interoperability verified i.e.; an application should be able to inter-operate with the other computer and mobile devices.

Security testing:

➢ Authentication: Only the authenticated user should be able to Login.

➢ Authorized: User should be able to log into those modules only for which he is authorized or for which the user has been provided access to.

➢ Password: Password requirements are verified i.e., password should be as per how the requirement defines i.e., length, special characters, numbers etc.

➢ Internal links to the web application are not accessible if placed directly in the browser.

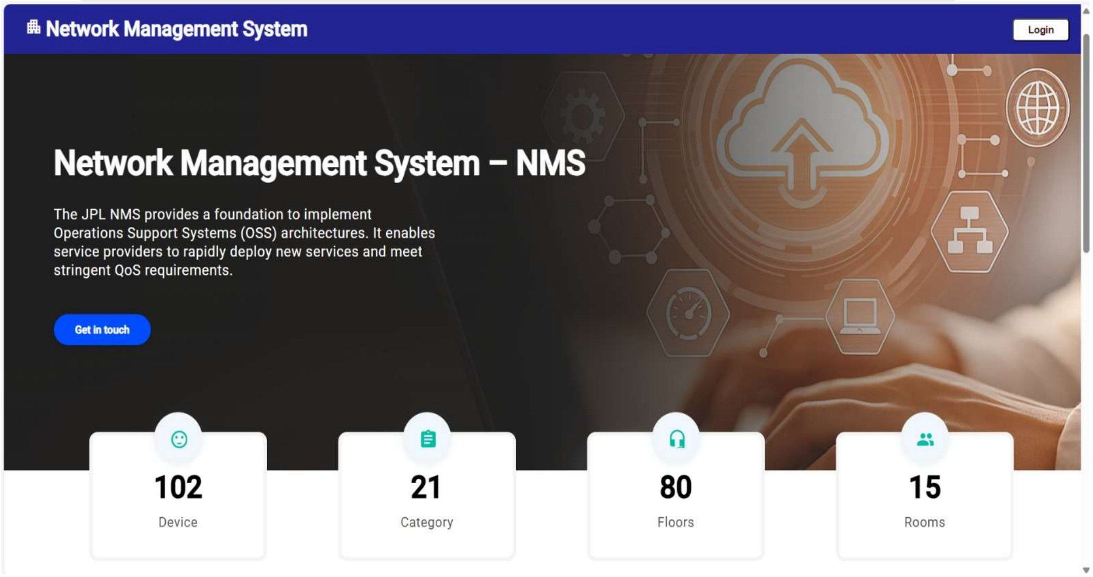
➢ All the communication are encrypted.

Chapter 8

INPUT – OUTPUT FORMS

Input and Output Screens:

1. Home Page:

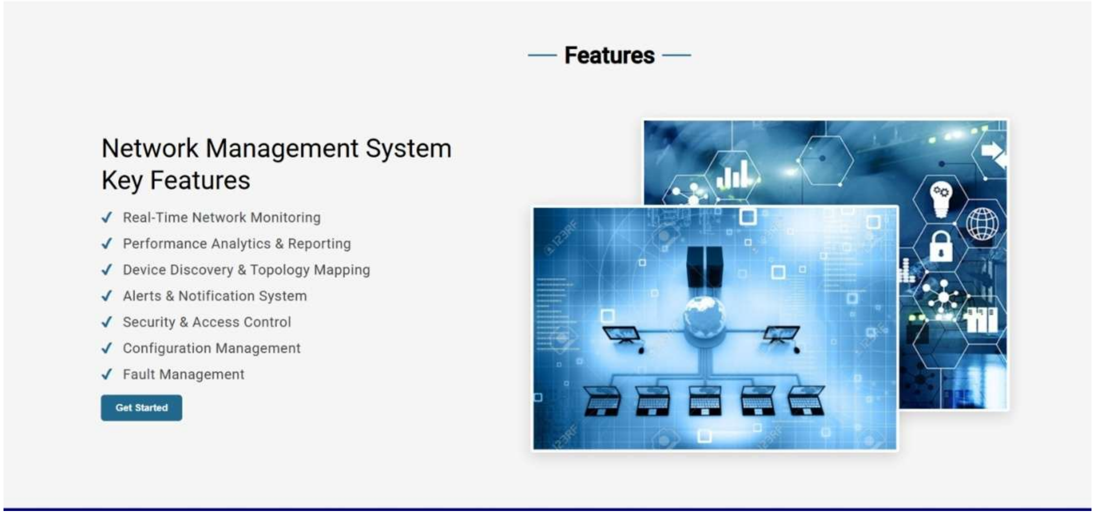


Network Management System

Network Management System – NMS

The JPL NMS provides a foundation to implement Operations Support Systems (OSS) architectures. It enables service providers to rapidly deploy new services and meet stringent QoS requirements.

102	21	80	15
Device	Category	Floors	Rooms

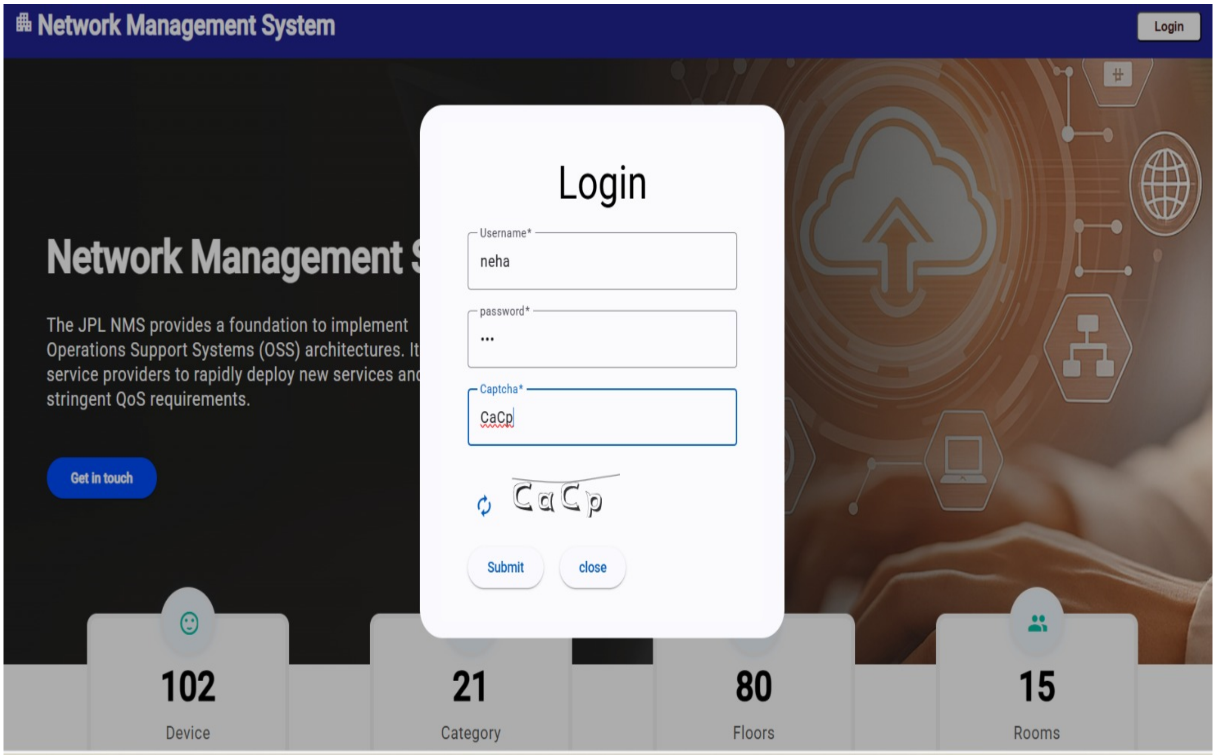


— Features —

Network Management System Key Features

- ✓ Real-Time Network Monitoring
- ✓ Performance Analytics & Reporting
- ✓ Device Discovery & Topology Mapping
- ✓ Alerts & Notification System
- ✓ Security & Access Control
- ✓ Configuration Management
- ✓ Fault Management

2. Login Page:



Network Management System

Login

Network Management S

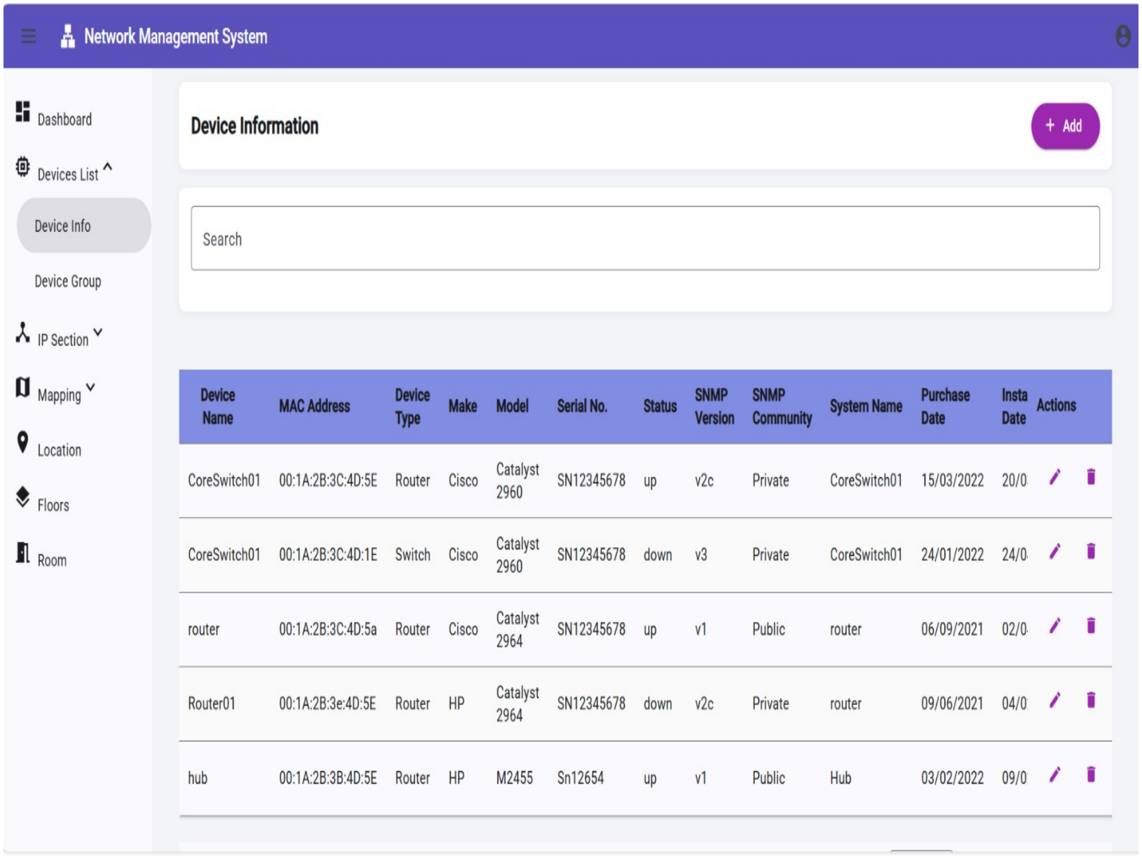
The JPL NMS provides a foundation to implement Operations Support Systems (OSS) architectures. It service providers to rapidly deploy new services and stringent QoS requirements.

Username* neha

password* •••

Captcha* CaCp

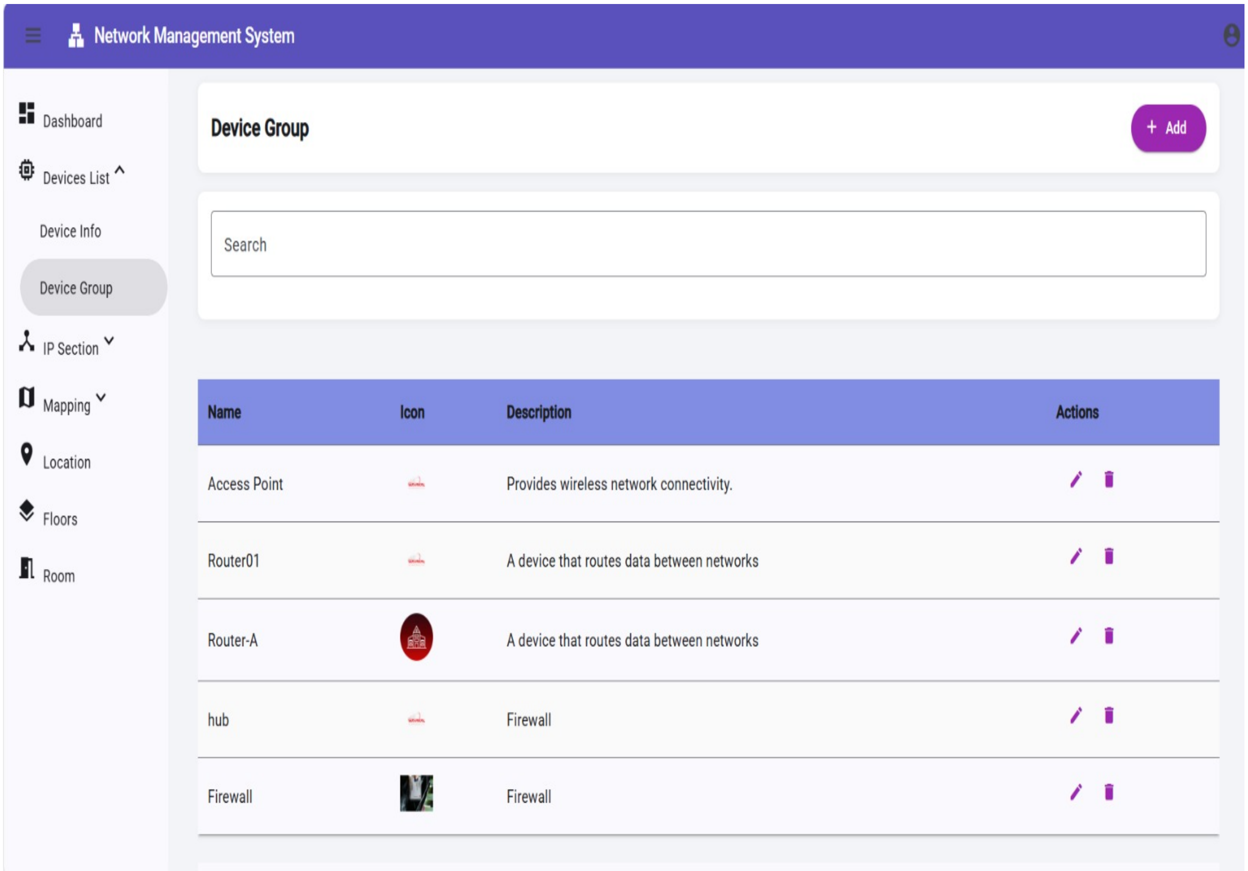
3. Device Information Page:



Device Information

Device Name	MAC Address	Device Type	Make	Model	Serial No.	Status	SNMP Version	SNMP Community	System Name	Purchase Date	Insta Date	Actions
CoreSwitch01	00:1A:2B:3C:4D:5E	Router	Cisco	Catalyst 2960	SN12345678	up	v2c	Private	CoreSwitch01	15/03/2022	20/0	✎ 🗑
CoreSwitch01	00:1A:2B:3C:4D:1E	Switch	Cisco	Catalyst 2960	SN12345678	down	v3	Private	CoreSwitch01	24/01/2022	24/0	✎ 🗑
router	00:1A:2B:3C:4D:5a	Router	Cisco	Catalyst 2964	SN12345678	up	v1	Public	router	06/09/2021	02/0	✎ 🗑
Router01	00:1A:2B:3e:4D:5E	Router	HP	Catalyst 2964	SN12345678	down	v2c	Private	router	09/06/2021	04/0	✎ 🗑
hub	00:1A:2B:3B:4D:5E	Router	HP	M2455	Sn12654	up	v1	Public	Hub	03/02/2022	09/0	✎ 🗑

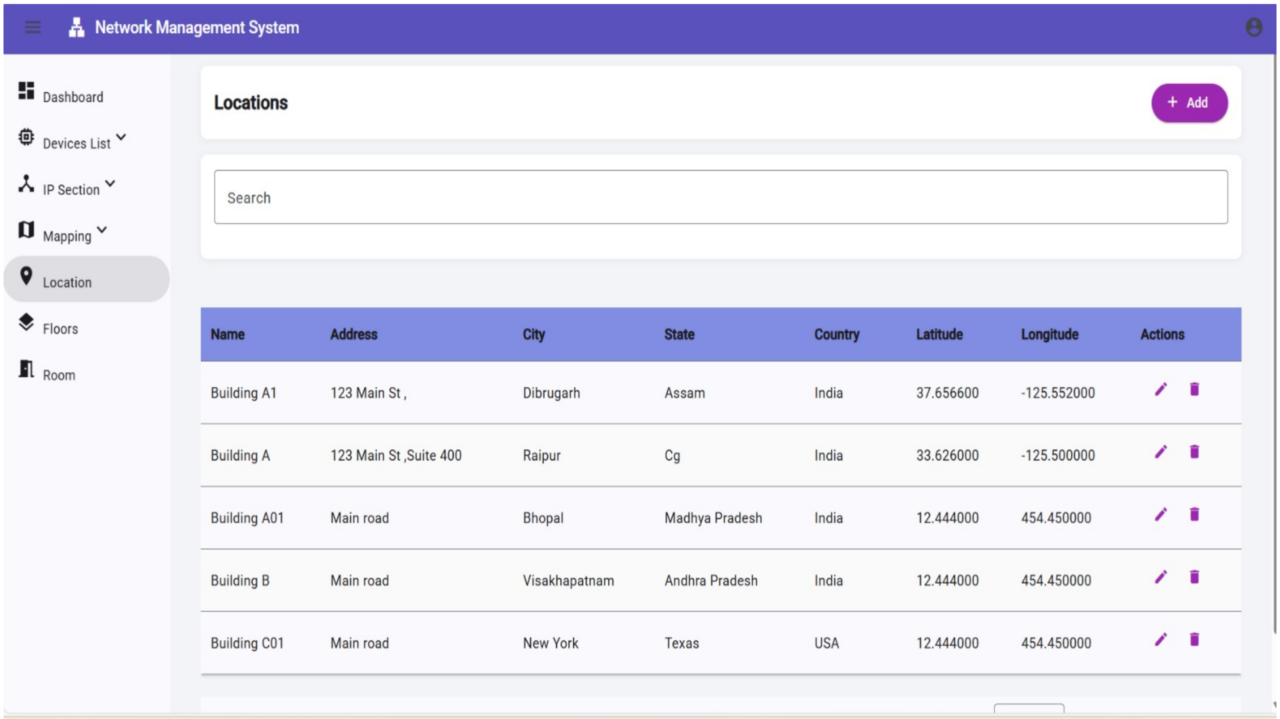
4. Device Group Page:



Device Group

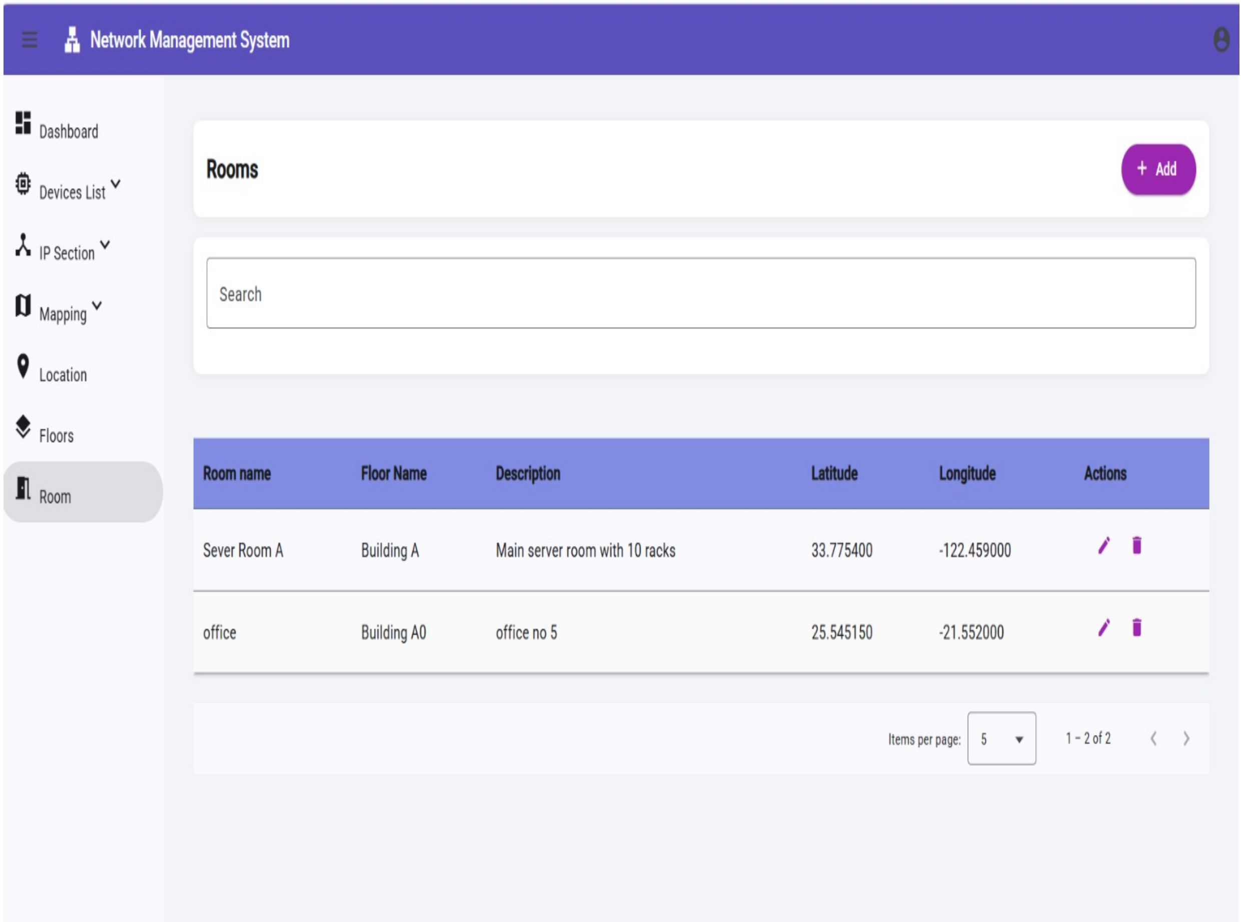
Name	Icon	Description	Actions
Access Point		Provides wireless network connectivity.	✎ 🗑
Router01		A device that routes data between networks	✎ 🗑
Router-A		A device that routes data between networks	✎ 🗑
hub		Firewall	✎ 🗑
Firewall		Firewall	✎ 🗑

5. Location Page:



Name	Address	City	State	Country	Latitude	Longitude	Actions
Building A1	123 Main St ,	Dibrugarh	Assam	India	37.656600	-125.552000	
Building A	123 Main St ,Suite 400	Raipur	Cg	India	33.626000	-125.500000	
Building A01	Main road	Bhopal	Madhya Pradesh	India	12.444000	454.450000	
Building B	Main road	Visakhapatnam	Andhra Pradesh	India	12.444000	454.450000	
Building C01	Main road	New York	Texas	USA	12.444000	454.450000	

6. Room Page:



Rooms

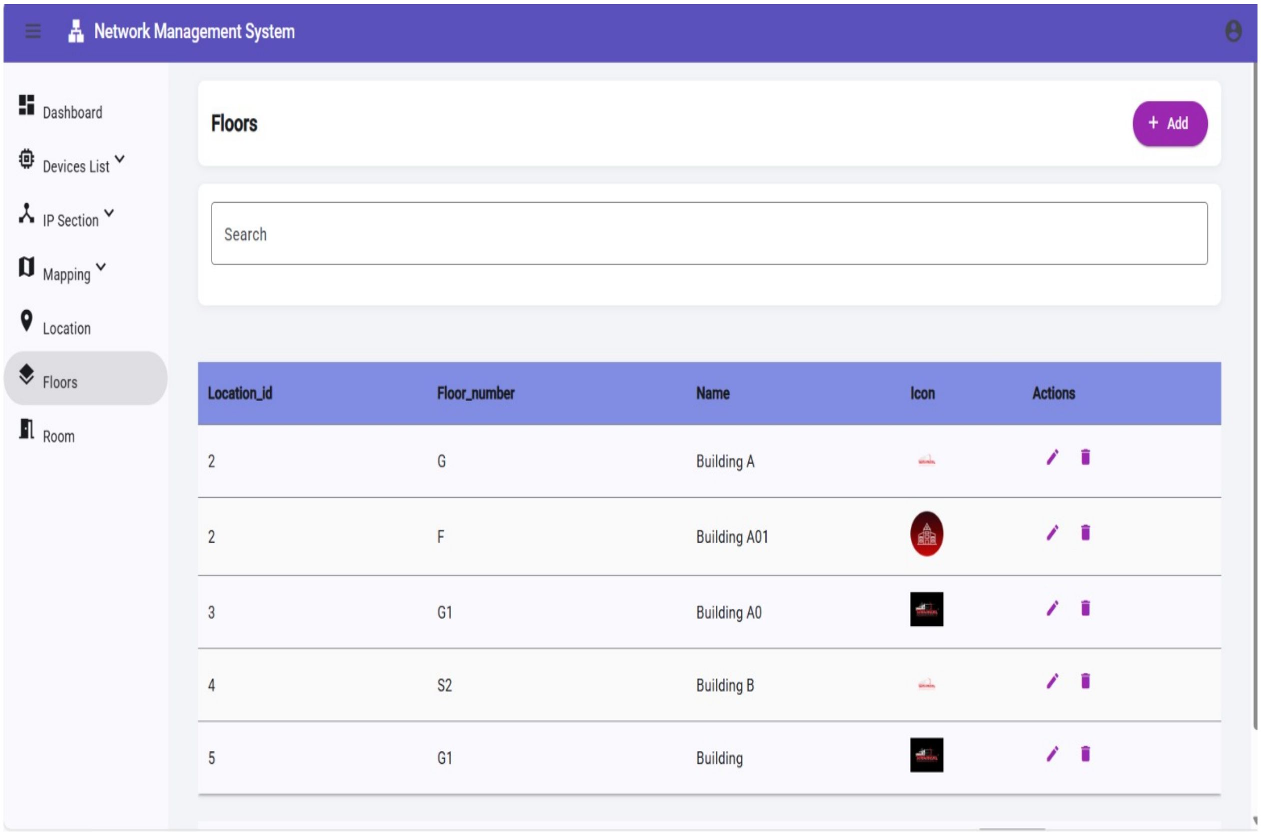
+ Add

Search

Room name	Floor Name	Description	Latitude	Longitude	Actions
Sever Room A	Building A	Main server room with 10 racks	33.775400	-122.459000	✎ 🗑
office	Building A0	office no 5	25.545150	-21.552000	✎ 🗑

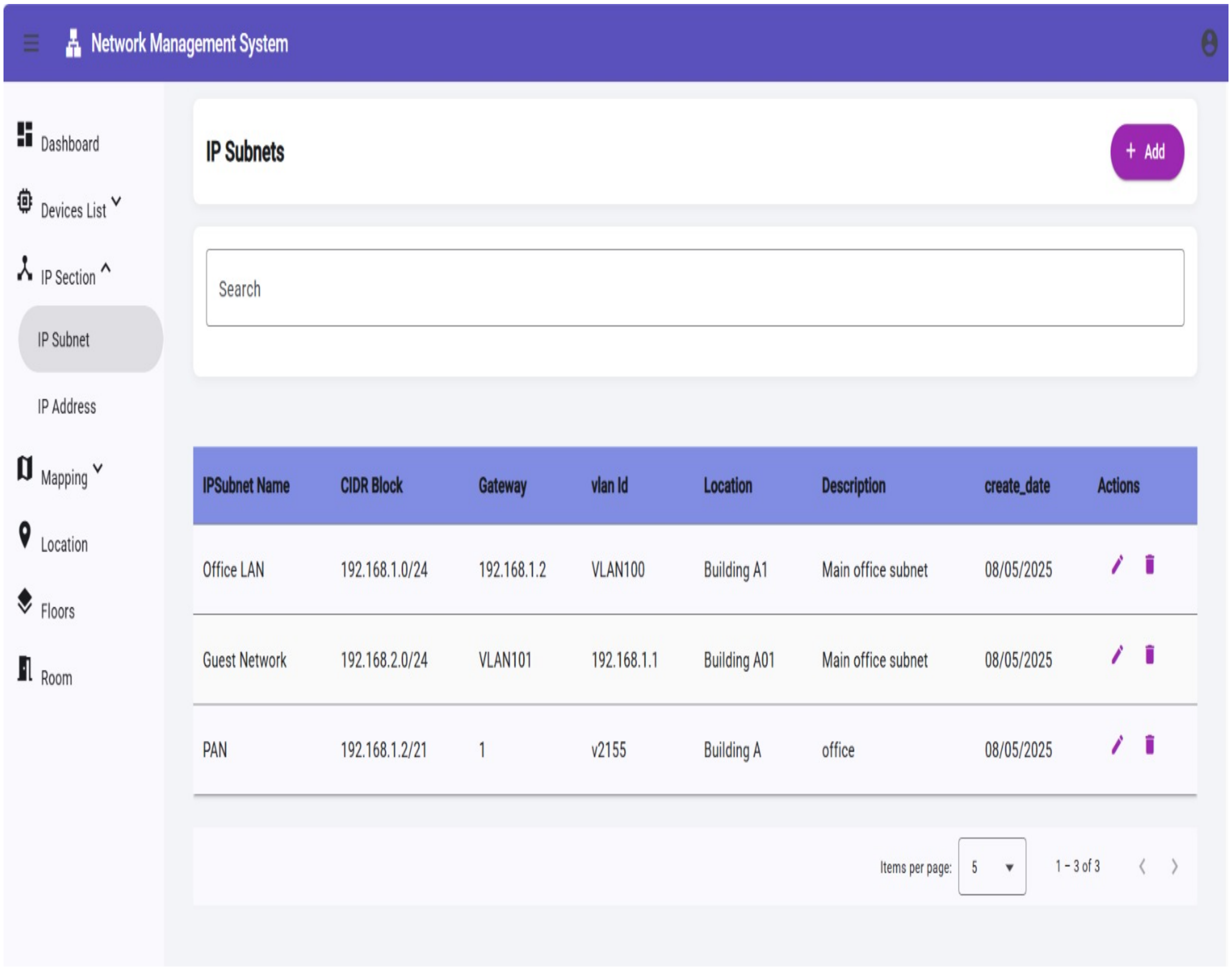
Items per page: 5 ▾ 1 – 2 of 2 ‹ ›

7. Floor Page



Location_id	Floor_number	Name	Icon	Actions
2	G	Building A		
2	F	Building A01		
3	G1	Building A0		
4	S2	Building B		
5	G1	Building		

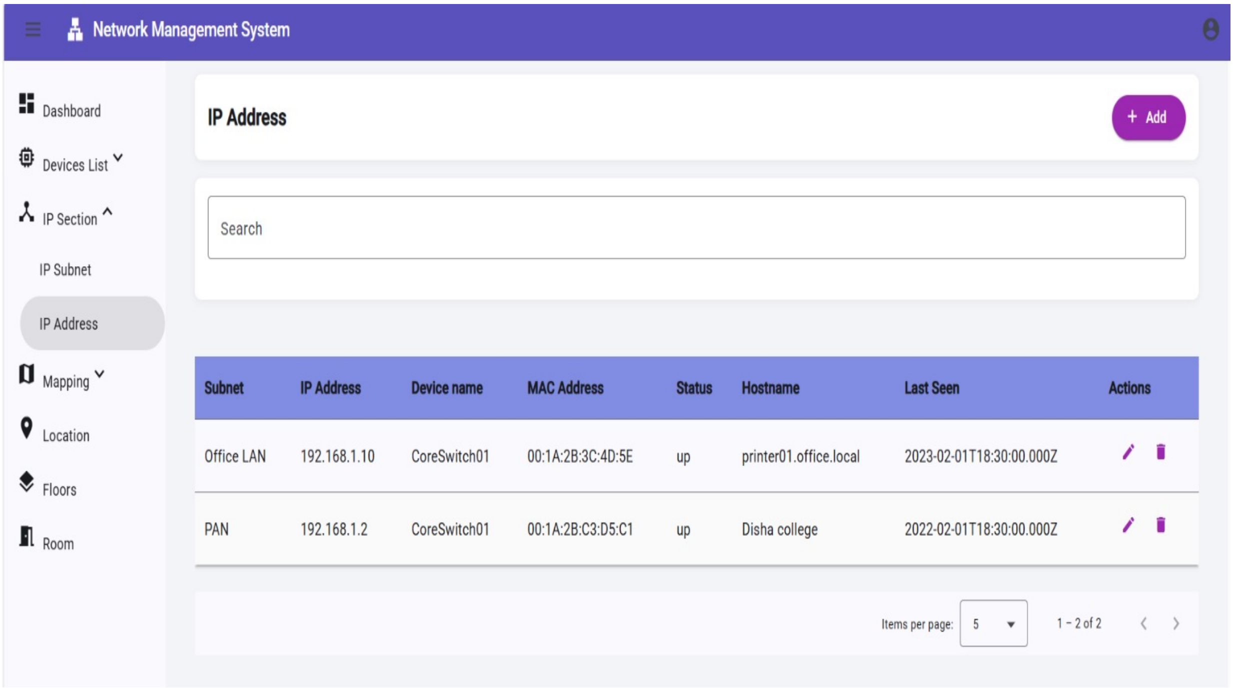
8. IP Subnet Page



IP Subnets

IPSubnet Name	CIDR Block	Gateway	vlan Id	Location	Description	create_date	Actions
Office LAN	192.168.1.0/24	192.168.1.2	VLAN100	Building A1	Main office subnet	08/05/2025	✎ 🗑
Guest Network	192.168.2.0/24	VLAN101	192.168.1.1	Building A01	Main office subnet	08/05/2025	✎ 🗑
PAN	192.168.1.2/21	1	v2155	Building A	office	08/05/2025	✎ 🗑

9. IP Address Page



Subnet	IP Address	Device name	MAC Address	Status	Hostname	Last Seen	Actions
Office LAN	192.168.1.10	CoreSwitch01	00:1A:2B:3C:4D:5E	up	printer01.office.local	2023-02-01T18:30:00.000Z	
PAN	192.168.1.2	CoreSwitch01	00:1A:2B:C3:D5:C1	up	Disha college	2022-02-01T18:30:00.000Z	

Chapter 9

SYSTEM SECURITY

Introduction: The protection of applications that include hardware, software, data, procedures and people against unauthorized use or natural disaster is known as Application Security.

Application security refers to various validations on data in the form of checks and controls to avoid the application from failing. It is always important.

9.1 Security:

➢ A secure login and logout facility is provided. Only Registered User can login into the application.

➢ Parameterized queries are used to prevent SQL injection.

➢ Internal pages are only accessible when the user is authenticated.

➢ Here I am using two types of validation i.e., client-side validation and server-side validation to ensure security and to ensure that only valid data is entered and only valid operations are performed on the system.

9.1.1 Client-Side Validation: Client-side validation saves server time and loads to handle invalid data. Some checks imposed are:

➢ Type-Script is used to ensure that all required fields are filled with valid data only. Maximum lengths of the fields of the forms are appropriately defined.

➢ Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client-side to save the server time and load. 9.1.2 Server-Side Validation: Server-side validation is also used to protect against malicious users, who can easily bypass our Client-Side scripting language and submit dangerous input to the server. Some of the server-side checks imposed is:

➢ Server-side constraint has been imposed to check for the validity of primary key and foreign key. If any attempt does not satisfy this key, then a proper error message are showing to the user.

➢ Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the server-side.

➢ User is intimating through appropriate messages about the successful operations or exceptions occurring at server side.

Chapter 10

Limitation/Future Enhancement

Limitation / Future enhancement

10.1 Limitation of system

• This is a web-based application, so this need internet facility.

• This application needs high performance processor and RAM.

• Data processing can be slow if internet will be slow.

10.2 Future enhancement:

- Mobile app for real-time alerts
- Add more features
- Make better responsive for all screen sizes
- Cross platform software app
- Better user-interface
- Update with new technology on demand

Chapter 11

Conclusion

Conclusion: In conclusion, an effective **University Network Management System (NMS)** is vital for maintaining the reliability, security, and performance of the institution's IT infrastructure. As universities increasingly rely on digital learning platforms, research data transmission, and hybrid learning environments, the demand for robust network management grows. A well-designed NMS enables real-time monitoring, fault detection, configuration management, and performance optimization across all campus network devices and services.

Looking ahead, integrating advanced features such as AI-driven automation, predictive analytics, and cloud-native support will significantly enhance the system's ability to meet evolving academic and administrative needs. These enhancements will ensure that the university network remains scalable, secure, and resilient—supporting both current operations and future digital transformation initiatives.

The project is still in the developing stage, at the time of submitting the dissertation only the development of the UNI-Connect portal has been completed, the following work is to be done in the future.

Future enhancement:

- Mobile app for real-time alerts
- Add more features
- Make better responsive for all screen sizes
- Cross platform software app
- Better user-interface

References and Bibliography

BIBLIOGRAPHY

While developing this project, I have used some websites which helped me in our development process.

WEB REFERENCES:

1. **https://angular.io/docs**
2. **https://stackoverflow.com**
3. **https://nodejs.org/en/docs**
4. **https://www.youtube.com/@TechnicalBabaji1**
5. **https://github.com/topics/**
6. **https://mariadb.org/documentation/**