**DIGITAL NOTICE BOARD USING RASPBERRY PI3**

*Submitted By*

**TAKKELLAPATI JAYAPRAKASH - 15BQ1A1250**

**DOPPALAPUDI ANUDEEP - 15BQ1A1211**

**GUDIMETLA VINEETH - 15BQ1A1214**

**NELAKUDITI AKHIL SAI – 15BQ1A1239**

Under the Supervision of

***Mr. B.RAJESH, M. Tech***

***Asst. Prof. of IT***

******

**Department Of Information Technology**

**Vasireddy Venkatadri Institute Of Technology**

**NAMBUR**

**ABSTRACT**

The Internet of Things (IoT) is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators and network conductivity which enable these objects to connect and exchange data. Notice Board is the primary thing in any institution or public utility places like bus stations, railway stations, colleges, malls, etc. But sticking various notices day to day is a difficult process. The main objective of this IoT system is to develop a wireless notice board that displays message sent from user. It is simple to design, easy to install, user friendly system, which can receive and display notice in a particular manner with respect to date and time, which help the user to easily keep the track of the notice board every day.

The project is built around ARM controller raspberry-pi which is the heart of the system. At any time we can add or remove or alter the text or notices according to our requirement using smart phones, tablets or computers. The raspberry-pi takes the data from the user through internet and displays the same on the notice board.

Keywords: Internet of things; Raspberry Pi; Notice Board; Wireless system.

**SOFTWARE REQUIREMENT SPECIFICATION**

**INTRODUCTION**

The Software Requirements Specification (SRS) is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are redefined by establishing a complete information description, a detailed functional and behavioural description, an indication of performance requirements and design constraints, appropriate validation criteria and other data pertinent to requirements.

**PURPOSE**

The purpose of the document is to describe the functionality and requirements. Its purpose is to automate and centralize the electrical connected devices of a home. The current system in the home automation is based on the webpage by using php. In such a system it is very difficult to access the webpage. The document also describes the functional and non-functional requirements. It also describes the design constraints that are to be considered when the system is to be designed, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

**SCOPE**

Our project “DIGITAL NOTICE BOARD USING RASPBERRY PI3” aims at creating an web application which will ease the process of connecting to raspberry pi. Using this it is possible to control electrical devices such as Notice Board. It has good user interface.

**DEFINTIONS, ACRONYMS AND ABBREVIATIONS**

**SRS -** Software Requirement Specification document.

**OS -** Operating System.

**GPIO -** General-purpose input/output

**REFERENCES**

The documents and websites referred to the project are as follows:

1. [https://fallthrough.io/2017/02/external -access -pi - ngrok/](https://fallthrough.io/2017/02/external%20-access%20-pi%20-%20ngrok/)

2.<https://www.raspberrypi.org/help/>

**OVERVIEW**

In the modern world of technology, computers are affecting our lives in more ways. Our project “DIGITAL NOTICE BOARD USING RASPBERRY PI3” aims at creating an web application which will ease the process of connecting to raspberry pi. Using this it is possible to control electrical devices such as Notice Board. It has good user interface.

The entire SRS is documented in view of USER and the following subsection are arranged to give a complete outlook of the software, its perspective, features, and system requirements.

After giving a brief introduction about the project the body of the report is divided into two parts. They are

**Overall description or General description**

This section will describe the major components of the system like the properties, functions and constraints of the project.

**Specific requirements**

This section will describe the details of functions and function constraints.

**OVERALL DESCRIPTION**

**PRODUCT PERSPECTIVE**

Notice Board is the primary thing in any institution or public utility places like bus stations, railway stations, colleges, malls, etc. The main objective of this IoT system is to develop a wireless notice board that displays message sent from user. It is simple to design, easy to install, user friendly system, which can receive and display notice in a particular manner with respect to date and time, which help the user to easily keep the track of the notice board every day.

**PRODUCT FUNCTION**

Digital Notice Board is developed to access the Notice Board connected the Web Browser. This helps to manage Updates Daily. It is also useful in a situation when daily sticking notices on notice board .

A summary of major functions that Digital Notice Board will perform:

* User must be identified.
* In a board it should be flexible to handle messages.
* Application also gives some information regarding the daily Notices.

**PRODUCT FEATURES**

Some of the features of the project Digital Notice Board System are

1.Interpolation  
2. Remote Access

3. Expandability

4. Upgradeability

5. Variety of User Interfaces

  6. Energy Management

7. Layers of Protection

**USER CHARACTERISTICS**

In this user can control the device with web browser Then the Notice will display on Notice Board They can Change the notice at any Time. To identify the user as genuine user can use default security application in their mobile otherwise they can set the some secret question kind of thing.

**GENERAL CONSTRAINTS**

* All the devices must be connected to the web browser so that they are under a network to access.
* User can set security kind of things by that they will not allow others to access from their network.
* Genuine user must be identify with some kind of security..

**ASSUMPTIONS AND DEPENDENCIES**

* The users must have how to control the system/ remote accessing.
* The application must need internet to control the things.
* System must respond as per the mobile application request with less responsive time.

**SPECIFIC REQUIREMENTS**

**FUNCTIONAL REQUIREMENTS**

***FUNCTIONAL REQUIREMENT 1***

**INTRODUCTION**

When we click on submit button on in the browser the signal is transmitted through WIFI to raspberry pi which is Displayed on The Digital Notice Board.

**INPUT**

Pressing the Submit button in Web Page.

**PROCESSING**

In the processing the signal is transmitted over WIFI to raspberry pi and raspberry pi sends the signals to the inbuilt programmable GPIO pins from where the signal is sensed by the LED.

**OUTPUT**

The Text Message is Displayed on The Digital Notice Board.

**EXTERNAL INTERFACE REQUIREMENTS**

**SOFTWARE INTERFACES**

Operating System - Raspbian

Mobile Operating System- Android(V4.0-V8.1)/Windows

Programming tool - Node-Red and Android Studio

Web Browser - Google Chrome, Mozilla Firefox

The product will host an android device where the user interface will be displayed via the android mobile application. The programming tool Node-Red is used for wiring hardware between Raspberry Pi and Relay. Android Studio is used to create android mobile application for the user to interact with the devices through android mobile.

**HARDWARE INTERFACES**

**Raspberry Pi 3**

SoC: Broadcom BCM2837

CPU: 1.2 GHZ quad-core ARM Cortex A53

GPU: Broadcom VideoCore IV

Memory: 1 GB LPDDR2-900 SDRAM

Storage: Samsung 32gb class 10 microsd card

The Application will communicate with Raspberry Pi 3. The user can communicate through android application installed in android mobile device

**LED**

16x4 LCD character

**Router**

WIFI standards: 802.11a/b/g

Wireless Frequency: 2.4GHz or 5GHz

**COMMUNICATION INTERFACE**

The devices are communicated through WIFI(802.11 standards)

**MEMORY CONSTRAINTS**

At least 1GB RAM and 32GB microSD card will be required for running the application.

**PERFORMANCE REQUIREMENTS**

* This development of the software will be based on the Prototype model.
* The performance of every module must be well.
* The system would exhibit high performance because it would be well optimized.
* The users must get the response within seconds i.e. the response time of a particular function should be minimum.
* The risk factors must be taken at initial step for better performance of the software.
* The overall performance of the software will be reliable and enable the user to work efficiently.

**DESIGN CONSTRAINTS**

The system must be designed to allow web usability. That is, the system must be designed in such a way that will be easy to use.

**SECURITY REQUIREMENTS**

Since the devices are connected within home network a basic firewall provided by the Raspbian OS is enough

**MAINTAINABILITY REQUIREMENTS**

After the deployment of the project if any error occurs then it can be easily maintained by the software developer. The requirements modules that are explained in this document are enough to satisfy the customer’s needs and wants in case of change or addition demand after completing the system or in development process of the system.

**AVAILABILITY REQUIREMENT**

Since the devices are connected with in home network through Router.they are available 24x7.

**SAFETY REQUIREMENTS**

* Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product.
* Defining any safeguards or actions that must be taken, as well as actions that must be prevented.

**OPERATIONAL REQUIREMENTS**

* The system will have user-friendly interfaces.

Only the user connected to the home network can control the electrical device

**INTRODUCTION TO UML**

**DEFINTION OF UML**

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and documents the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system’s architectural blueprints inlcuding elements such as

* Actors
* Business processes
* Components
* Activities
* Programming language statements
* Database schemes
* Reusable software components

The UML represents the culmination of best pratical object oriented modeling. The UML is the product of several years of hardworking which we focused on bringing about a unification of the methods used around the world.

**Goals of UML**

The primary goals in the design of the UML were:

* Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
* Provide extensibility and specialization mechanisms to extend the core concepts.
* Be independent of particular programming languages and development processes.
* Provide a formal basis for understanding the modeling language.
* Encourage the growth of the OO tools market.
* Support higher-level development concepts such as collaborations, frameworks, patterns and components.
* Integrate best practices.

**SCOPE OF UML**

The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, Constructing and documenting the aircrafts of a software intensive system. The UML specifies a modeling language that incorporates the object oriented communities consensus on core modeling concepts. It allows deviations to be expressed in terms of extension mechanisms. The **UML provides the following**:

* Semantics and notation to address a wide variety of contemporary modeling issues in a direct and economical fashion.
* Semantics to address certain expected future modeling issues, specifically related to component technology, distributed computing, frame works and excitability.
* Extensibility mechanisms so individual projects can extend the Meta model for their application at low cost.
* Extensibility mechanisms so that future modeling approaches could be grown on the top of the UML.
* Semantics to facilitate the model interchange among variety of tools.
* Semantic to specify the interface to repositories for the sharing and storage of model aircrafts.

**There are several new components included in UML**

* Extensibility mechanisms
* Threads and process
* Distribution and concurrency
* Patterns / collaboration
* Activity diagrams
* Refinement
* Interface and components
* A constraint language

**USE CASE DIAGRAM**

The Use Case Diagram is a graphic depiction of the interactions among the elements of DigitalNoticeBoard System. It represents the methodology used in system analysis to identify, clarify and organize system requirements of the system.

**IMPORTANCE OF USE-CASE DIAGRAM**

Within the Object-Modeling approach, one of the earlier steps involves building an USE-CASE Model. The essence of this model is to capture user requirements of a new system, whether it is being developed from scratch or based on an existing system, by detailing all the scenarios that users will be performing.

**IDENTIFYING USE-CASES**

A use-case captures the interactions that occur between producers and consumers of information and the system itself. A use-case is a description of set of sequence of actions that a system performs that yields an observable result of value to a particular actor. A use-case is used to structure the behavioral things in a model. A use-case is realized by collaboration. Graphically, a use-case is rendered as an ellipse with solid lines, usually including only its name.

**DEVELOPING A SET OF USE-CASES**

The functions or activities performed by a specific actor are listed.These may be obtained from a list of required system functions, through conversations with customersor end-users.

Use-case diagram depict the functionality of the system. The use case diagram has the following types of elements.

1. Actor.

2. Use-case.

3. Association.

**Actor**

Shown as a stick figure icon. This represents users and external systems with

which the system we are discussing interacts.



**Use-case**

Shown as an ellipse. This represents a functional requirement that is described

from the perspective of the user’s of a system.

**Use case**

**Association**

Shown as a solid line path from an actor to use-case. This represents that the actor

uses the use-case.

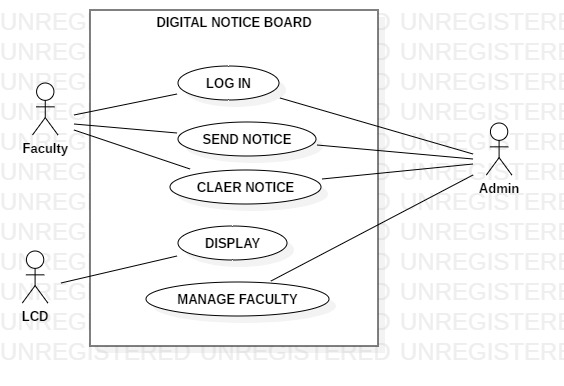
**Association**

**The main actors of DigitalNoticeBoard**

* Admin
* Faculty
* LCD

Relationships between and among the actors and the use cases of DigitalNoticeBoard

* **Admin Actor :** Use cases of Admin are manage Faculty, Send Notice, Clear Notice, login and logout of the system.
* **Faculty Actor :** Use cases of Faculty are send Notice, clear Notice, login .
* **LCD Actor :** Use cases of Display Notice.



**Use case diagram for DIGITAL NOTICE BOARD**

**Class Diagram**

Class diagram is used to display some of the classes and packages in your system.

* A class diagram includes a subset of the classes, attributes, operations, relationships, and packages of classes in the system.
* It gives us a static picture of the pieces in the system and of the relationships between them.
* Class diagrams help the developers see and plan the structure of the system before the code is written helping to ensure that the system is well designed from the beginning.

Generally, a class is rendered as rectangle, usually including its name, attributes and operations.

|  |
| --- |
| Name |
| Attributes |
| Operations |

* A class is depicted on the class diagram as a rectangle with three horizontal sections. The upper section shows the class's name; the middle section contains the class's attributes; and the lower section contains the class's operations
* We use classes to capture the vocabulary of the systemwe are developing. These classes may include abstractions that are part of the problem domain, as well as classes that make up an implementation.

The Class Diagram for DigitalNoticeBoard describes the structure of a Raspberrypi3 Database System classes, their attributes, operations and the relationships among the objects.

**Classes of DigitalNoticeBoard System**

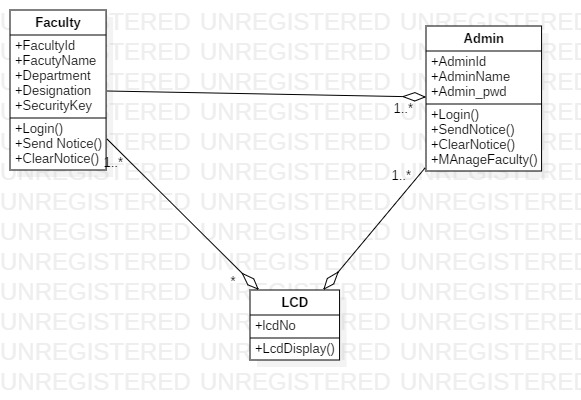
* Admin Class
* Faculty Class
* LCD Class

**Class and their Attributes of DigitalNoticeBoard System**

* **Admin Attributes :** admin\_id, admin\_name,Admin\_pwd.
* **Faculty Attributes :** faculty\_id, faculty\_name, securityKey
* **LCD Attributes:** LcdNo

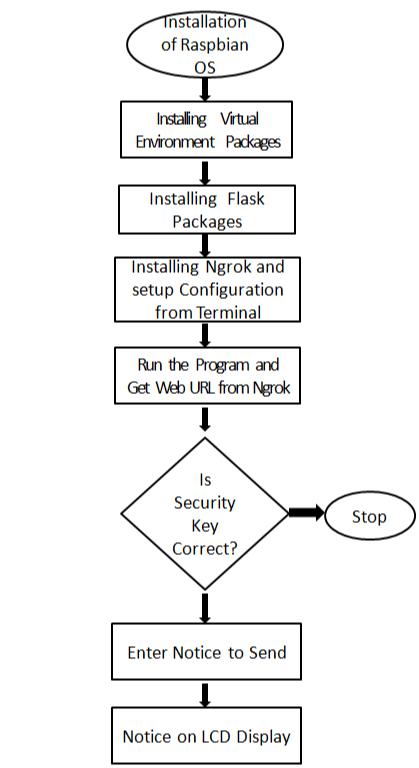
**Classes and their Methods of DigitalNoticeBoard System**

* **Admin Methods :** ManageFaculty, sendNotice,ClearNotice,Login.
* **Faculty Methods:** SendNotice,ClearNotice,Login
* **LCD Methods:** Display



**Class diagram for DIGITALNOTICEBOARD**

**FLOW DIAGRAM**



**SEQUENCE DIAGRAM**

Sequence diagram also known as interaction diagram depicts how elements interact over time.

A horizontal axis shows elements involved in the interaction and the vertical axis represents time proceeding down the page. The sequence diagram has following types of elements.

* Classes and objects
* A lifeline
* A communication between objects

**Classes and objects**

Classes are so much the same way as on class diagram. Objects may also be shown much the same way as an object diagram.

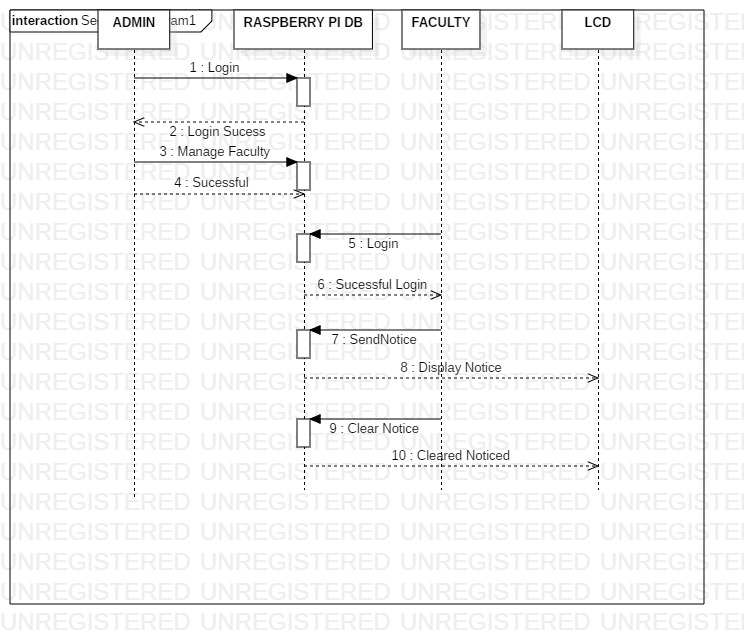
**Lifeline**

Shown as a vertically dashed life from an element. This represent the existence of the element over time.

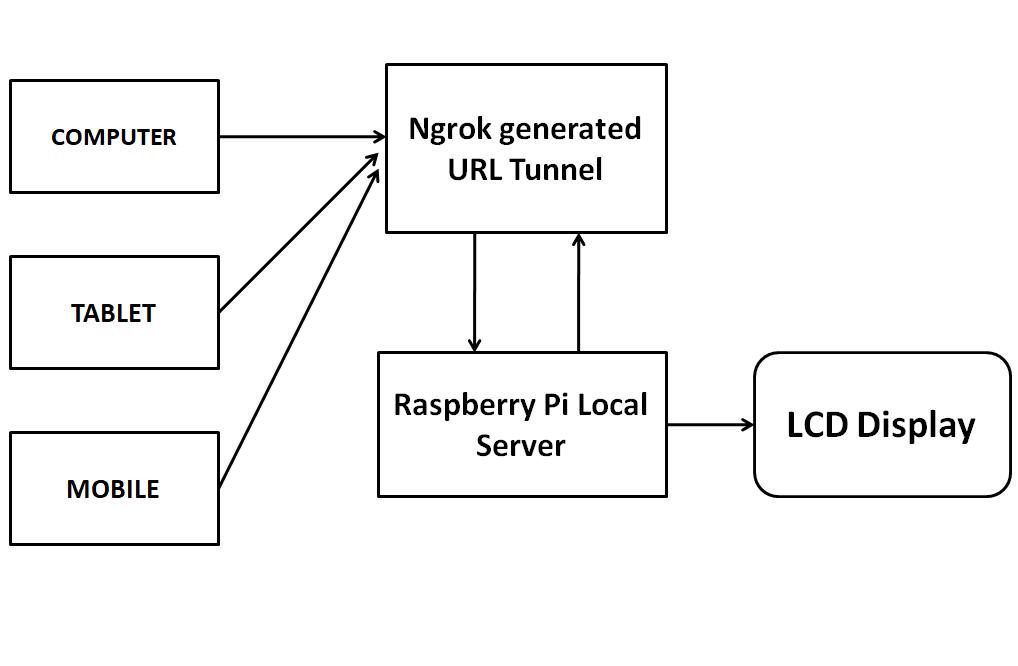
**Communication**

Shown as a horizontal solid arrow from the lifeline of the sender to the lifeline of the receiver and labeled with the name of the operation to be invoked. This represents that sender sends a message or stimulus to the receiver.

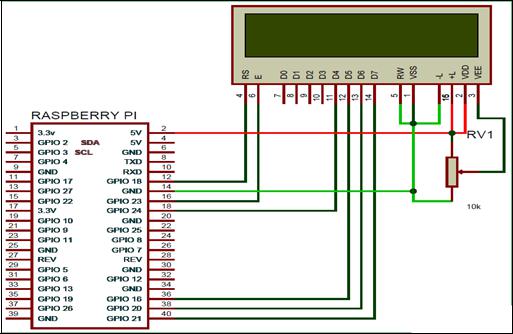
The UML Sequence Diagram of DigitalNoticeBoard System shows the interaction between the objects of Faculty, LCD. The diagram shows how the admin will be able to login to their accounts using their login credentials. After login user can manage all the operations . The diagram below demonstrates how the login page works.

**Sequence diagram for Admin Login of DIGITALNOTICEBOARD**

**BLOCK DIAGRAM**

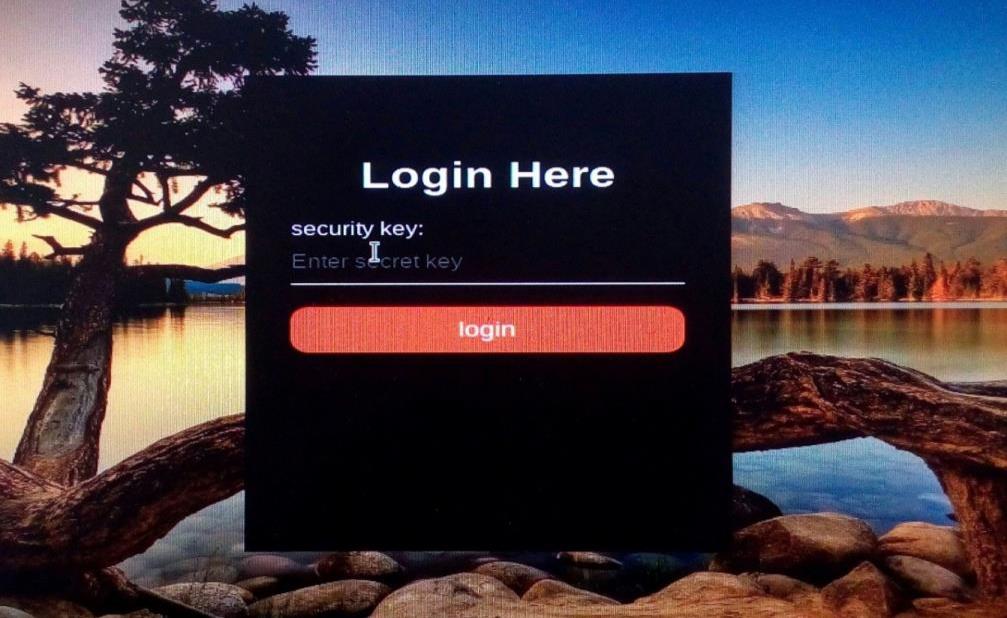
****

**CIRCUIT DIAGRAM**

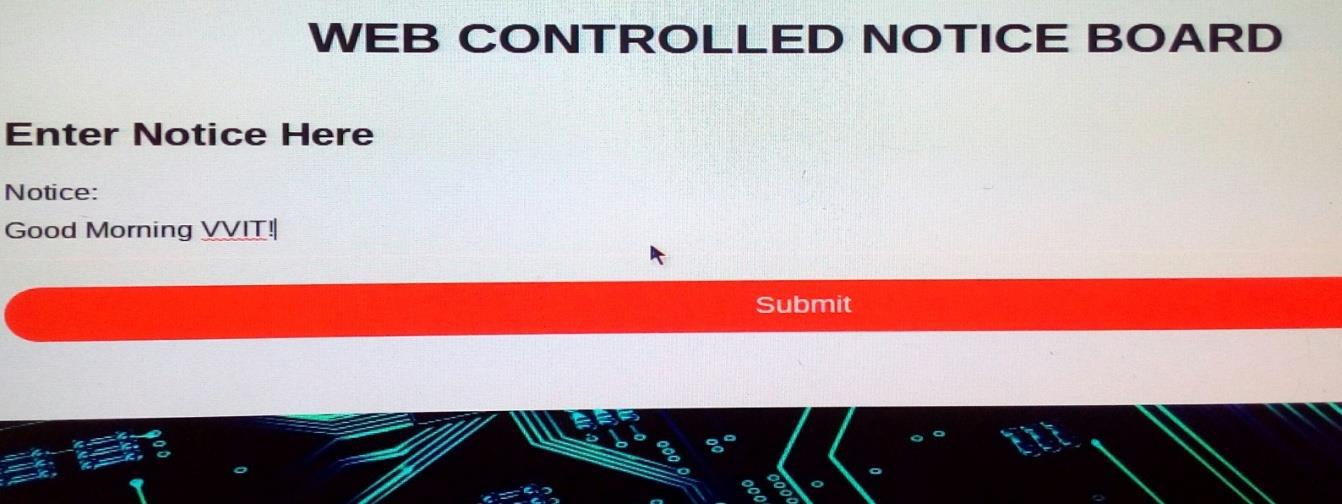


.

**IMAGES OF PROPOSED SITE**



**LOGIN PAGE**



**SEND NOTIFICATION**



**DISPLAY NOTICE USING RASPBERRYPI 3**