

# IoT Enabled Notice Board

Dr. Asawari Dudwadkar<sup>1</sup>, Omkar N. Tulaskar<sup>2</sup>, Mitesh R. Khedekar<sup>3</sup>, Anuja K. Merwade<sup>4</sup> and Shubham P. Sutrakar<sup>5</sup>

<sup>1</sup>Assistant Professor, Dept. of Electronics, V.E.S Institute of Technology, Mumbai, India

<sup>2,3,4,5</sup>UG Student, Dept. of Electronics, V.E.S Institute of Technology, Mumbai, India

**Abstract**—This paper deals with the implementation of software tools and the whole framework for IoT Enabled Notice Board. Notice boards can change the way people communicate with each other, providing important information to large people at the right time. Notice boards are used extensively in schools, colleges, hospitals, railway stations, hotels, malls, etc. The developed system includes the notice board being connected to a local server running on Raspberry Pi 24\*7. The notice to be displayed can be sent via the developed Android application or from the webpage. The data is then sent onto the server where it is pushed in a backend MySQL database. From the database, the contents of the notice are then displayed on the Monitor. The system provides an authentication layer to avoid any unauthorized activities since the target audience for the prototype developed is mainly schools and colleges.

**Keywords**—Raspberry Pi, Internet of Things, Database, E-notice

## I. INTRODUCTION

The main aim of the notice board is not only providing the information in a precise format to the general public but also providing the necessary information on time with negligible delay as many notices may be time-based. The currently employed system makes use of traditional methods which require human intervention right from deciding the information to delivering it to the audience. Also, an excess of manpower is needed for simple work[3]. The developed prototype involves the elimination of all this tedious procedure and makes use of the Internet of Things to ease the job[1]. The data to be displayed is sent from the Android application which connects to the local server running on the notice board[2]. The PHP script running on the local server extracts the payload and updates the database accordingly. Another PHP script keeps a check on the status of all the notices that are present in the database and displays the next set of notices in chronological order. For this, all the notices are flagged in the database for differentiation. This differentiation is based on the status of a notice being displayed on the monitor or not. As only one authorized person is involved from the generation of payload to displaying the notice, the developed prototype provides flexibility, ease of usage and zero paperwork as opposed to the traditional process.

## II. PROPOSED METHODOLOGY

The main objective of this project is to develop a wireless system that will enable the faculty/authorized user to display the latest information. At the implementation level, the whole prototype is developed as two different packages,

namely Web-based Package and Software-based package, for the ease of use as well as for providing flexibility. However, at maintenance levels, the efficiency of the web-based package is significantly high while in terms of development flexibility and user-level interaction the software-based package is considered to be more efficient and effective.

### A. Web-based package

The main working of the notice-board includes displaying the data that is being sent from the user-side. The android application makes use of HTTP GET request for delivering the payload from the user to the server. The data is then pushed onto the database. Webpage using HTML and CSS is developed to read the notices and display them accordingly. Here, two screens are used. One for idle mode and the other for notice mode. Idle mode is active when there is no notice to be displayed. Whereas the notice mode is active in the complementary situation.

### B. Software-based package

This package provides standard software developed using python to run on the Raspberry Pi. The python software is developed using python GUI library. The payload received from the user is first stored in the database. The python script makes a connection with the MySQL database using proper credentials. It then picks up the notices which are to be displayed and accordingly it updates its GUI on the screen.

## III. SOFTWARE MODELLING

### A. SIPOC MODEL

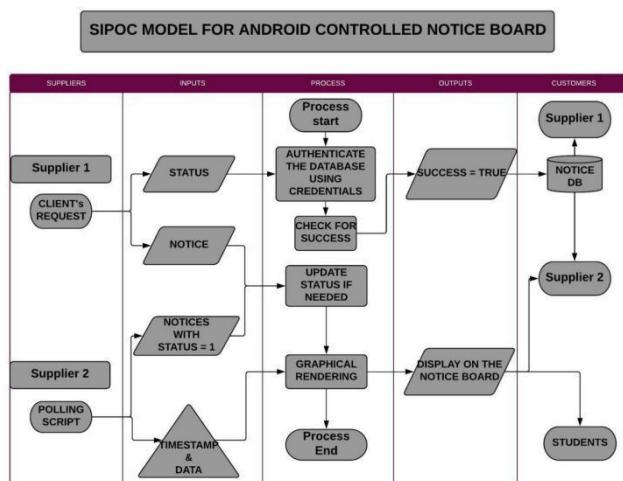


Fig. 1. SIPOC MODEL

The SIPOC model is one of the essential parts of the Six Sigma methodology. It is one of the most trusted and



primitive techniques for mapping critical business processes. This model is used predominantly by restaurants and manufacturing industries. SIPOC model is a tool to visualize the input and outputs of more than one process before implementation. It is also referred to as a high-level process map. It does not contain the in-depth information of the processes but rather provides an overall understanding of the interactions happening between consecutive processes. SIPOC stands for Suppliers, Inputs, Process, Output and Customers. The given SIPOC model consists of names of the columns. We have exploited the simplicity of this tool to enable the modelling of the entire project and provide a better understanding of the system. This further enables us to find any loopholes or improvements that could be taken up in the further versions of the system.

Here, the SIPOC model involves the interaction of the front-end user and the backend server. Supplier 1 is the 'CLIENTS REQUEST' for the payload that is needed to be sent. The user is first authenticated at the server side which is a normal login page using a MySQL database for storing the user credentials. Accordingly, the server accepts the data from the user using HTTP GET Request. A PHP script running on the server side extracts the payload and updates the notice database with status as 1 for the received notice. The script also stores the timestamp for the notice.

The POLLING SCRIPT is responsible for polling the status of all the notices present in the database. A PHP script continuously checks for the status of notices. If it finds any status as '1', it will take up that notice along with the timestamp and display it on the screen.

#### B. Python Based Software

Another module for the system is the implementation of python-based software. The PYTHON script provides a complete package from software implementation for polling algorithm to handling all the HTTP requests. The script will handle the incoming HTTP requests and extract the payload. Once requested the script will query the database to check if the credentials are authentic. Only after this verification, the

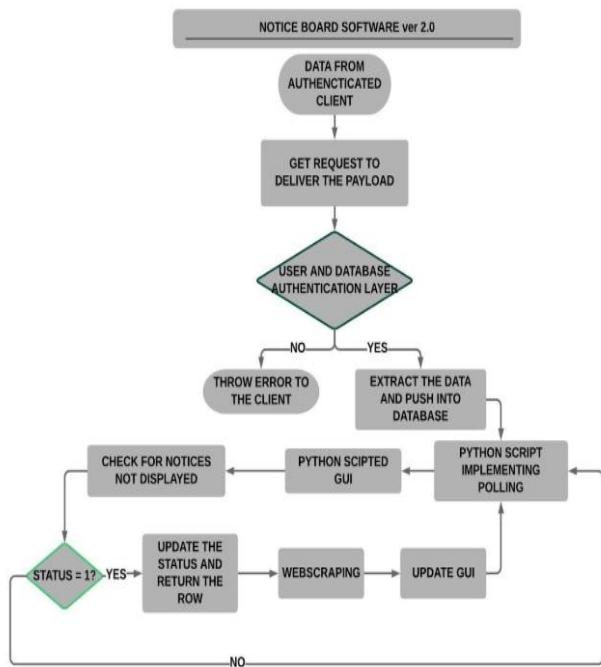


Fig. 2. Python software flowchart for version 2.0

script allows the user to update the notices. A dirty bit implementation is used to manage requests for the new notices. Every time a new notice is updated in the database, a dirty bit gets assigned to it. This enables the script to know if the notice has been displayed on the screen or not. This particular implementation will load the image only once thus saving the need for processing time.

It then also updates the database. The same script will then check for notices with status as 1 and display them along with their timestamp on the screen. Since Raspberry Pi supports mainly Python, the script development will be much flexible and smoothen the software development process.

#### IV. TOOLS DEPLOYED

The following HARDWARE and SOFTWARE tools were required for the development of the entire system along with the description.

##### A. Hardware tool



Fig. 3. Raspberry Pi 3

The Raspberry Pi model-3 is a development board that can host an entire light-weighted OS such as the Linux Distros. This powerful credit-card sized single-board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings along a more powerful processor. This processor is 10 times faster than the first-generation Raspberry Pi module. Additionally, it contains wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs. The Raspbian OS was loaded as the OS for this project along with all the necessary softwares such as Python, php and MySQL server.

##### Features:

- 1) Broad-com BCM2837 ARM7 Quad-Core Processor powered Single Board Computer running at 1.2 GHz.
- 2) 1 Gb of RAM
- 3) 40 pin extended GPIO
- 4) Full-size HDMI port
- 5) Memory: Micro SD Card

##### B. Software Tools

###### HTML

HTML stands for HyperText Markup Language. It is a basic markup language used to design documents to be displayed on webpages. It describes the structure of a webpage semantically and originally included cues for the appearance of the document. Elements of an HTML code are

delineated using tags and are written using angle brackets. Some of the other significant elements of an HTML code are headings, body, paragraph, list, etc.

#### PHP

It is an open source, general purpose, server-side scripting language. It is specifically used for the development of web applications that are dynamic and interactive in nature. Its speed, agility and flexibility enables everything from a blog to be written to some of the most renowned websites.

#### MySQL DB

MySQL database is a fully managed relational database service. It provides an implementation of SQL database and is very well suited for small to medium web pages. It is a Database Management System that manages databases, connects them to software and also provides cloud-native deployment solutions. The application is used for a variety of purposes like data warehousing, maintaining e-commerce databases as well as logging applications.

#### CSS

CSS stands for Cascading Style Sheets. It is used to design and format a webpage built using HTML. It enables the user to place elements anywhere on the webpage and provides strong design rules and flexibility. It describes the way a webpage written in a markup language like HTML must be displayed.

#### Android Studio

It is a software used for designing Android applications for Android phones, TVs, tablets, etc. It provides an efficient user interface for connection with the database and also has individual units that can be used from building to deploying the application.

### V. SYSTEM SETUP



Fig. 4. Experimental Setup

Here, the monitor is interfaced with Raspberry Pi and the corresponding webpage is displayed. This webpage displays the notice sent from the app or login webpage by the authorized user.

### VI. IMPLEMENTATION AND RESULTS

The login webpage is developed using HTML and CSS. This webpage generates the payload which includes the credentials of the user. The login credentials are already

stored in the database which is used to check against this payload. As per the departmental process, these credentials are made available only to the authorized users at the beginning of the academic year. Due to the sensitivity of the project, under no circumstances, this login credential can be changed by the user. Only the owner of the database has the authority with a master credential to access the database and update user credentials.



Fig. 5. Login webpage

Fig. 5: A login webpage is designed for authorized users to log in to the portal and send notices as text or images to be displayed on the E-notice board.

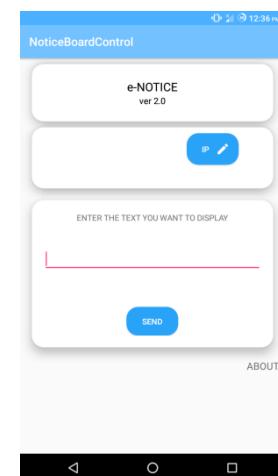


Fig. 6. Android application

Fig. 6: An Android application is made to provide easy access to authorized users to log in to the system and send appropriate notices as and when required.

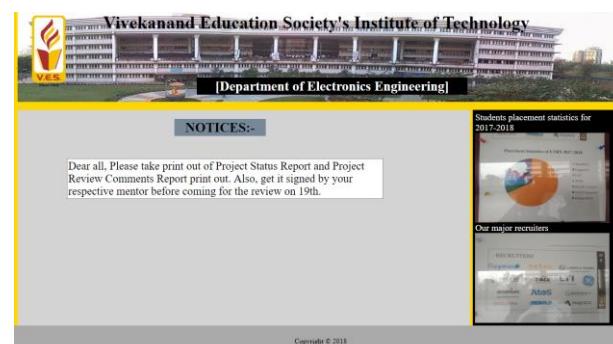


Fig. 7. Displayed notice

Fig. 7: The notice sent by an authorized user is displayed directly on the screen. It may be a text notice or a file (image, document, etc.)

The other way to update the notices is through the Android application which also supports the same amount of implementation as it is for the webpage-based service. The user needs to enter the secure IP address which can be used to connect only on the WiFi of the department. Once connected the user can then upload the required notice either in the text format or in the form of images. No pdf support has been provided due to the constraints of the size of the database being deployed for the pilot project and hence only valid image formats are allowed for the users to be uploaded

TABLE I. DATABASE SCHEMATIC

<b>ID</b>	<b>TIMESTAMP</b>	<b>DATA</b>	<b>STATUS</b>
1	Unix Timestamp	NOTICE-1	1
2	Unix Timestamp	NOTICE-2	0
3	Unix Timestamp	NOTICE-3	0
4	Unix Timestamp	NOTICE-4	1

The above table shows the general structure of the database where the notices are stored. Depending on the status of a particular notice, further action is taken whether to display the notice or not.

## VII. CONCLUSION

By introducing the concept of wireless technology in the field of communication, we can make our communication faster and much more efficient. The information can be displayed with fewer errors and less maintenance. This system can be employed in schools, colleges, offices, and for commercial as well as personal use. Latency involved in using papers for displaying notices is avoided and the information can be updated by the authorized person(s).

## ACKNOWLEDGMENT

This project is supported by Dept. of Electronics, V.E.S Institute of Technology, Mumbai, India.

## REFERENCES

- [1] Vinod B. Jadhav, Tejas S. Nagwanshi, Yogesh P. Patil, Deepak R. Patil "Digital Notice Board using Raspberry Pi" IJRET, Volume: 03, Issue: 05 | May-2016, published.
- [2] Dr. P. Gnana Sundari, P. Sangeetha, M. Sowmiya, N. Soundarya. "Wireless E-notice board using Raspberry Pi 3" IJRTER, Special issue: March 2018, published.
- [3] Er. G. Jalalu, Er. Polepogu Rajesh "Wireless Electronic Notice Board Using Raspberry Pi 3", IJRET, Volume: 05, Issue: 06 | June-2017, published.