1. INTRODUCTION

1.1 BACKGROUND

Home automation is the control of any or all electrical devices in our home or office. In today's era, technology can enhance human life. Technology is evolving decade by decade. Automation was a science fiction earlier but not today. By combining latest technology with home, we can build an awesome home. With the Raspberry Pi and Windows 10, we can build a home automation system that is capable of operating home devices automatically.

Home Automation can be considered as an act of using electronic systems/devices and programming them to replace a number of human interactions for the control of basic home functions. This operates on the base of connecting sensors and devices to the IoT.

1.2 OBJECTIVE

The objective of our system is to take care of several domestic systems that may normally be difficult for those who are handicap or elderly to take care of. Home automation or Smart Homes can be described as introduction of technology within the home environment to provide convenience, comfort, security and energy efficiency to its occupants. Adding intelligence to home environment can provide increased quality of life. With the introduction of the Internet of Things (IoT), the research and implementation of home automation are getting more popular.

1.3 PROBLEM DEFINITION

The focus of my project is on helping users to operate home appliances with their own smartphones and to help elderly or handicapped people live a more independent life as long as possible. The objective of our system is to take care of several domestic systems that may normally be difficult for those who are handicap or elderly to take care of. The proposed idea will allow a user with any android enabled device to run a piece of downloadable software on any mobile device such as a smartphones. This application will allow the user to control a device that is connected to any home appliance that is Pi enabled. The focus of this application will be to direct a security system with webcam surveillance, door sensor notification and a light control system. Sensors will be connected to the home appliances with Pi so that they can be monitored and controlled.

1.4 PURPOSE

The proposed system does not require a dedicated server PC with respect to similar systems and offers a novel communication protocol to monitor and control the home environment with more than just the switching functionality. To demonstrate the feasibility and effectiveness of this system, devices such as light switches, power plug, temperature sensor and current sensor can be integrated with the home control system.

1.5 SCOPE

Can be expanded to include sensors that monitor temperature and other measurable quantities and make the data available to the user in real time. Gates and doors can be locked and unlocked using magnetic strips that can detect whether the door is open or closed.

A smart home concept can be integrated with the home automation system to develop an intelligent remotely controllable home system.

There are several ways this could be improved in future work. The use of surface mount components would dramatically decrease the overall size of the components. Surface mount components are also often less expensive as they require less material to produce. This would help reduce the overall cost of the devices as well as the size. Another area to help improve the size is the circuit board that is used. Currently for the prototype, a generic breadboard style board was used. If this device were to be commercially produced, a more compact circuit board could be designed.

1.6 APPLICABILITY

By the rapid developments of new technologies, monitoring, controlling services have been started to be served along with internet as an instrument providing interaction with machinery and devices. The system can be use in several places like banks, hospital, labs and other sophisticated automated system, which dramatically reduced the hazards of unauthorized entry.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

A commercially marketable home automation model based on IoT has not been developed in the state right now. Home Automation Systems are usually developed by hobbyists as pet projects. These projects only focus on personal requirements, as in what the developer needs, instead of tackling all the related aspects of Home Automation.

2.1.1 LIMITATIONS OF EXISTING SYSTEM

- Existing systems only focus on personal requirements
- They do not provide solutions for all aspects of Home Automation
- Once installed, major rework is necessary for modification to the system

2.2 PROPOSED SYSTEM

The Raspberry Pi based Home Automation System using IOT is a system that aims to make the distance between your home and office smaller. It ensures that all appliances you want to control are connected to the Internet and can be controlled (switched on or off) from anywhere provided the user is connected to the internet.

2.2.1 ADVANTAGES OF PROPOSED SYSTEM

- Security: Secure login that ensures only the home owner has access to the appliance status database.
- Energy Efficiency: Unwanted electricity consumption can be reduced.
- Savings: Less consumption, in turn, leads to savings.
- Elderly Assistance: Very helpful for Elderly citizens to lead an independent life not requiring extra assistance in this case

2.3 SOFTWARE AND HARDWARE REQUIREMENTS

Hardware Requirements: The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. The hardware requirements required for this project are:

1. Raspberry pi

The version of the board or the model (B) and connect it to your local network, so you will need a Wi-Fi dongle if you are using the A model which doesn't have an Ethernet port. In this paper, we used a Raspberry Pi model B with the Wi-Fi dongle.

The Raspberry Pi is a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools. The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz, Video Core IV GPU, later upgraded to 512 MB. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and long-term storage

2. Relay circuit

A Relay is electrically operated switches, which allow low power circuits to switch a relatively high voltage or current on/off. For a relay to operate a suitable pull in and holding current should be passed through its coil. Relay coils are designed to operate from a particular voltage often its 5V or 12V. The function of relay driver circuit is to provide the necessary current energize the relay coil, when a LOGIC 1 is written on the PORT PIN thus turning on the relay. The relay is turn off by writing LOGIC 0 on the port pin. In our system four relays are used for device control

.

Software Requirements: Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed. The software requirements that are required for this project are:

1 .Python

It is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly braces or keywords), and a syntax which allows programmers to express concepts in fewer lines of code than possible. The language provides constructs intended to enable writing clear programs on both a small and large scale. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library.

Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

2. Raspbian OS

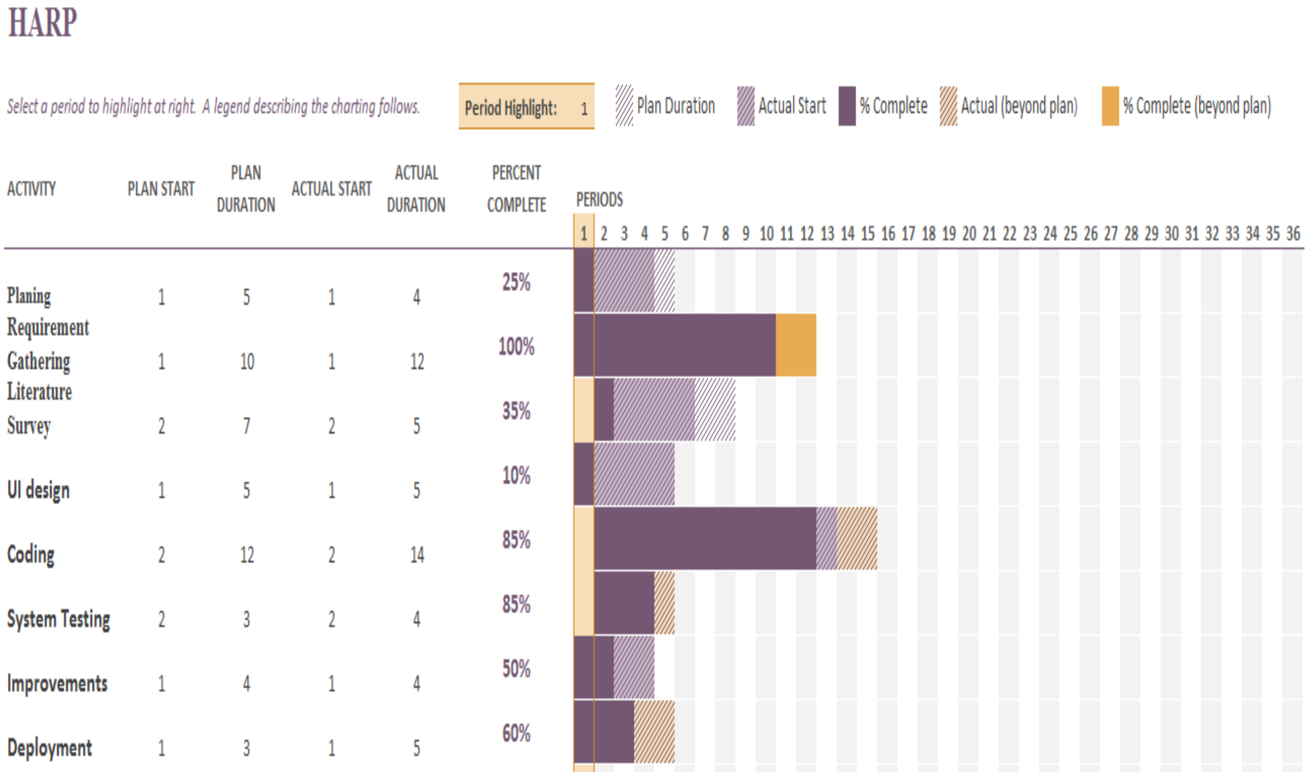
Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on the Raspberry Pi. The OS version used in this project is Raspbian Jessie with Pixel (Kernel Version 4.4)

3. REQUIREMENTS SPECIFICATION

3.1 PURPOSE AND OVERVIEW

The purpose of my project is on helping users to operate home appliances with their own smartphones and to help elderly or handicapped people live a more independent life as long as possible. The objective of our system is to take care of several domestic systems that may normally be difficult for those who are handicap or elderly to take care of. The proposed idea will allow a user with any android enabled device to run a piece of downloadable software on any mobile device such as a smartphones.

3.2 PROJECT PLAN: GANTT CHART



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Planing	1	5	1	4	25%
Requirement Gathering	1	10	1	12	100%
Literature Survey	2	7	2	5	35%
UI design	1	5	1	5	10%
Coding	2	12	2	14	85%
System Testing	2	3	2	4	85%
Improvements	1	4	1	4	50%
Deployment	1	3	1	5	60%

3.3 GENERAL DESCRIPTION

3.3.1 PRODUCT PERSPECTIVE

Home automation or smart home is building automation for the home. It involves the control and automation of lighting, heating (such as smart thermostats), ventilation, air conditioning (HVAC), and security, as well as home appliances such as washer/dryers, ovens or refrigerators/freezers. Wi-Fi is often used for remote monitoring and control. Home devices, when remotely monitored and controlled via the Internet, are an important constituent of the Internet of Things. Modern systems generally consist of switches and sensors connected to a central hub sometimes called a "gateway" from which the system is controlled with a user interface that is interacted either with a wall-mounted terminal, mobile phone software, tablet computer or a web interface, often but not always via Internet cloud services.

3.3.2 PRODUCT FUNCTIONALITY

The following list offers a brief outline and description of the main features and functionalities of the Home Automation application:

- **Control appliances**

  It ensures that all appliances you want to control are connected to the Internet and can be controlled (switched on or off) from anywhere, provided the user is connected to the internet. We need not be at home to switch the appliances on/off.

- **Display status of appliances**

  It displays the status of each connected appliance on the web portal. If the home owner has any doubt relating to whether any appliance is on or off; it can be cleared by checking the status on the application or on the portal.

3.3.3 USER CHARACTERISTICS

End users are home owners or elderly citizens. They access the web portal through the android application or web browser; and can utilize the entire functionalities of the project. They access the database through the web portal, view each appliance status, and switch appliances on/off.

3.4 SPECIFIC REQUIREMENTS

3.4.1 FUNCTIONAL REQUIREMENTS

The Raspberry Pi used must be able to connect to the internet to read the current status of the various connected appliances. For this project, we have used Raspberry Pi 3 Model B which has in built Wi-Fi connectivity, so that it can connect to the available network. If previous models of Raspberry Pi are used, a Wi-Fi module or an Ethernet shield has to be interfaced to achieve the same result.

Android smartphones must have Internet access to run the application and view the web portal using it. The phone must be of adequate specifications that enable it to run the application without any issues.

3.4.2 NON-FUNCTIONAL REQUIREMENTS

HDMI Keyboard and Mouse are required to make modifications to the system in case of a new appliance installation or appliance port shifts. It is not needed for the normal functioning of the system.

3.4.3 USER INTERFACE REQUIREMENTS

The Web Portal and Android Application interact with the end user through GUI. The interface is simple, easy to handle, and self-explanatory. Once accessed, user will easily come into flow with the application and can easily use all interfaces properly.

3.5 SRS REVISOIN HISTORY

There have been no revisions to the SRS since the start of the project.

4. SYSTEM DESIGN

4.1 ARCHITECTURE (Design Diagrams- use case, activity, deployment)

4.1.1 ACTICITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- rounded rectangles represent actions;
- Diamonds represent decisions;
- Bars represent the start (split) or end (join) of concurrent activities;
- a black circle represents the start (initial state) of the workflow;
- an encircled black circle represents the end (final state).

Arrows run from the start towards the end and represent the order in which activities happen.

Activity diagrams may be regarded as a form of flowchart. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

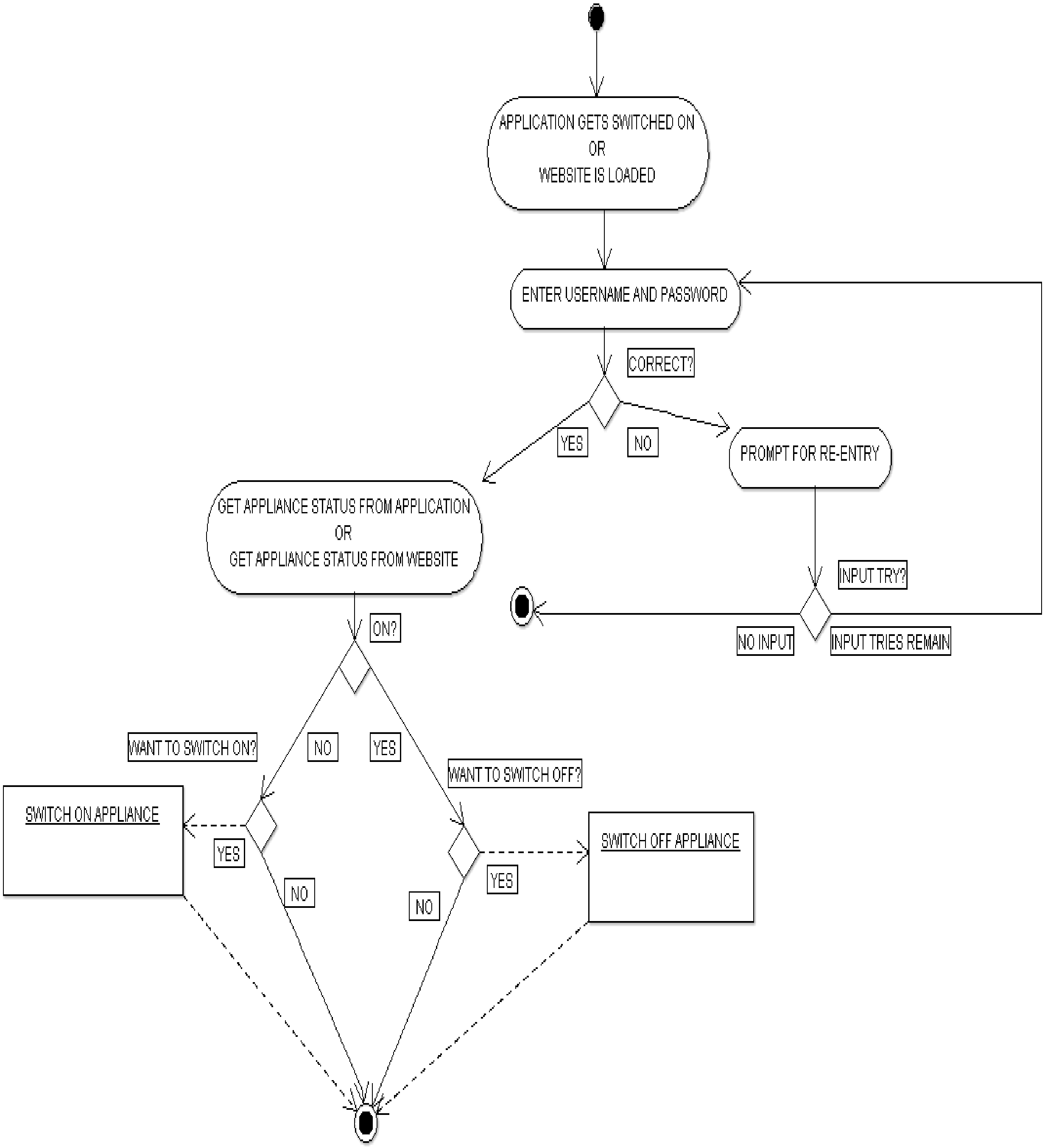


FIG: 4.1 ACTIVITY DIAGRAM

4.1.2 DEPLOYMENT DIAGRAM

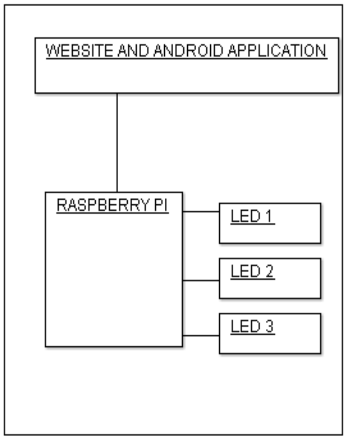


FIG: 4.2 DEPLOYMENT DIAGRAM

4.1.3 USE CASE DIAGRAM

Use case diagram identify the functionality provided by the system, the users who interact with the system (actors) and the association between the users and the functionality. Use cases are used in the analysis phase of software development to articulate the high level requirements of the system. The primary goal of use case diagram include providing a high- level view of what the system does, identifying the users(actor) of the system and determining areas needing human-computer interfaces. Use cases extend beyond pictorial diagrams. In fact, text-based use case descriptions are often used to supplement diagrams and explore use case functionality in more detail.

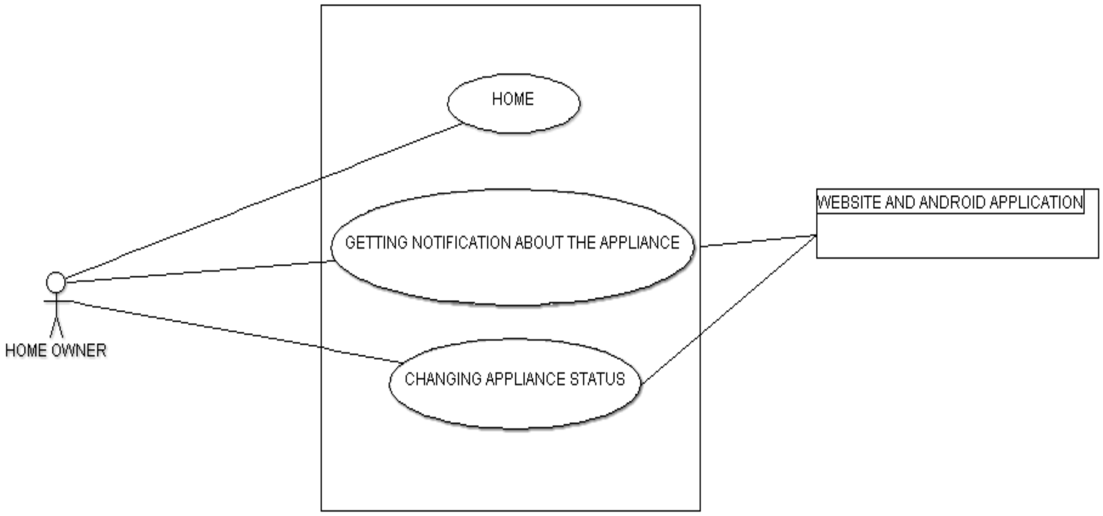

HOME
GETTING NOTIFICATION ABOUT THE APPLIANCE
CHANGING APPLIANCE STATUS
HOME OWNER
WEBSITE AND ANDROID APPLICATION

FIG: 4.3 USE CASE DIAGRAM

4.2 DATABASE DESIGN

There are two databases employed in this project. They are implemented using PHPMyAdmin. One database stores the username and password of each user and is used for validation of login details. The other database stores the status of each appliance connected to the system. This status database is accessed by the Raspberry Pi.

4.3 MODULE DESIGN

1. Raspberry Pi Microcontroller

The Raspberry Pi 3 is the third generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016. It has the following specifications:

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm Audio Jack and Composite Video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot (now push-pull rather than push-push)
- Video Core IV 3D graphics core

The Raspberry Pi 3 has an identical form factor to the previous Pi 2 (and Pi 1 Model B+) and has complete compatibility with Raspberry Pi 1 and 2

2. Web Interface

The server is the web interface consisting of buttons and UI (User Interface) that will allow you to turn ON/OFF a device. It consists of PHP files, Html files and a .txt file for each connected appliance (to store appliance status). The server usually stores information regarding the button press on the page (ON/OFF) on a device.txt file. This is the main website page called login.html, consisting of a login window. Once logged in, the site displays all the connected appliances and their status buttons. The clicking of the buttons will trigger the execution of a PHP file called button.php. This program serves as an API (Application Programming Interface) to store data on to a text file called device.txt. The data is a string: “ON”, if ON button is clicked and “OFF”, if OFF button is clicked. Thus the current button press state is recorded in the text file: device.txt.

3. Relay Circuit

A Relay is an electrically operated switch, which allows low power circuits to switch a relatively high voltage or current on/off. For a relay to operate, a suitable pull in and holding current should be passed through its coil. Relay coils are designed to operate from a particular voltage, often 5V or 12V. The function of relay driver circuit is to provide the necessary current to energize the relay coil, when a LOGIC 1 is written on the PORT PIN thus turning on the relay. The relay is turned off by writing LOGIC 0 on the port pin. In our system four relays are used for device control.

The relay inputs are received from the GPIO pins of the Raspberry Pi. The appliances are connected as load. The circuit checks for high voltage signal from the GPIO pins. The voltage triggers the relay switches and turns the appliance on. Four appliances can be connected to a single relay board. The number of appliances connected can be increased using Slave Arduinos for each room.

5. TESTING

5.1 TESTING STRATEGY

Software testing is the process used to access the quality of computer software. Software testing is an empirical technical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. This includes, but is not limited to the process of executing a program or application with the intent of finding software bugs. Quality is not an absolute; it is value to some person. With that in mind testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and behavior of the product against a specification. An important point is that software testing should be distinguished from the separate discipline of software quality assurance (or SQA) which encompasses all business process areas, not just testing.

Over its existence, computer software‘s has continued to grow in complexity and size. Every software product has a target audience. For e.g.: video game software has its audience completely different from backing software. Therefore, when an organization develops or otherwise in a software product, it presumably must assess whether a software product will be acceptable to its end users, its target audience, its purchasers and other stakeholders. Software testing is the process of attempting too makes this assessment.

In this phase the entire software system is tested. The reference document for this process is the requirements document, and the goal is to see whether the software meets the requirements. This is essentially a validation exercise, and in many situations it is only a validation activity. Testing focuses on the external behavior of the system.

5.2 TEST CASES

Test case is an object for execution for other modules in the architecture doesn’t represent any interaction with itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases-manual and automated. A manual test is executed manually while automated test is executed automatically.

Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if functional test cases are not written yet, it is ok for testers to check error conditions when performing normal functional test cases. It should be clear which test data, if any is expected to trigger error.

Step no	Test Steps	Expected Result	Actual	Status
1	Web Browser launched (Desktop)	Site open	Site open	Success
2	Web Browser launched (Smartphone)	Site open	Site open	Success
3	Correct Login Details Provided	Log In Success	Log In Success	Success
4	Incorrect Login Details Provided	Log In Failure	Log In Failure	Success
5	Devices Switched On	Devices Turned On	Devices Turned On	Success
6	Devices Switched Off	Devices Turned Off	Devices Turned Off	Success
7	Logged Out	Log Out Success	Log out Success	Success
8	Internet Connection Switched Off	Error message displayed	Error message displayed	Success

TABLE 5.1 TEST CASES

6. CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

Any system which has been in use for a number of years gradually decays and becomes less effective because of the change in environment to which has to adapt for a time it is possible to overcome problems by amendments and minor modification to acknowledge the need of fundamental changes. Computerization was proposed as a solution to the problem of being out dated with the fast present technologies. In this project, our aim is to minimize human effort and whole system has been done with sample data and output obtained is according to the requirements. In this project we have taken more care to bring this system well above these aims of the manual system. The project has been successfully implemented and is found to replace the manual system efficiently and effectively. We hoped to make this system as user friendly and to use as possible. We tried to achieve this and other objectives to our best. The benefits expected from the developed system outweigh the implementation costs. Data security is provided. The system developed here can be upgraded or expanded, if necessary.

6.2 FUTURE SCOPE

This project can be expanded to include sensors that monitor temperature and other measurable quantities and make the data available to the user in real time. Gates and doors can be locked and unlocked using magnetic strips that can detect whether the door is open or closed. A smart home concept can be integrated with the home automation system to develop an intelligent remotely controllable home system.

6.3 APPLICATIONS

Provides safety from electrical power short circuits while using conventional wall switches to operate loads. Home automation system with automated door locking and security cameras facilitates more security. By using a home automation system, we can save a lot of time to operate home appliances from anywhere (without wasting time to move from office to home for just unlocking door for family members to enter the home).

7. REFERENCE

- Home Automation : A Study – Satish Palaniappan, Naveen Hariharan[International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 11, April 2015]

- Home Automation Using Internet of Things - Vinay Sagar K N, Kusuma S M [International Research Journal of Engineering and Technology (IRJET) Volume: 02 Issue: 03 | June-2015]

- Web Interfaces:

- https://diyhacking.com/raspberry-pi-home-automation
- https://www.hackster.io/AnuragVasanwala/home-automation-0dcefc
- https://www.hackster.io/Salman_faris_vp/raspberry-pi-azure-based-home-automation-44e847?ref=search&ref_id=home%20automation%20raspberry%20pi&offset=2

APPENDIX

A.1 ABBREVIATIONS

- IP – Internet Protocol
- PC – Personal Computer
- WI-FI – Wireless Fidelity
- GPU – Graphical User Interface
- SD – Secure Digital
- SRS – Software Requirement Specification
- CPU – Control Processing Unit
- RAM – Random Access Memory
- OS – Operating System
- HTML – Hypertext Markup Language
- UI – User Interface
- API – Application Program Interface
- GPIO – General Purpose Input/output

A.2 USERMANUAL

HARP

GROUP MEMBERS

 Athira P Haridas

 Jilu Thomas

 John Mathew

HARDWARE REQUIREMENT

- Raspberry Pi 3 Model B
- Relay Board
- HDMI Keyboard and Mouse
- Breadboard
- Connection Elements
- Android Smartphone
- Quad-Core 1.2GHz Processor
- 256 MB RAM

SOFTWARE REQUIREMENT

- Smartphone: Android OS (KitKat or higher)
- Raspberry Pi: Raspbian OS (Jesse or higher)

HOW TO USE?

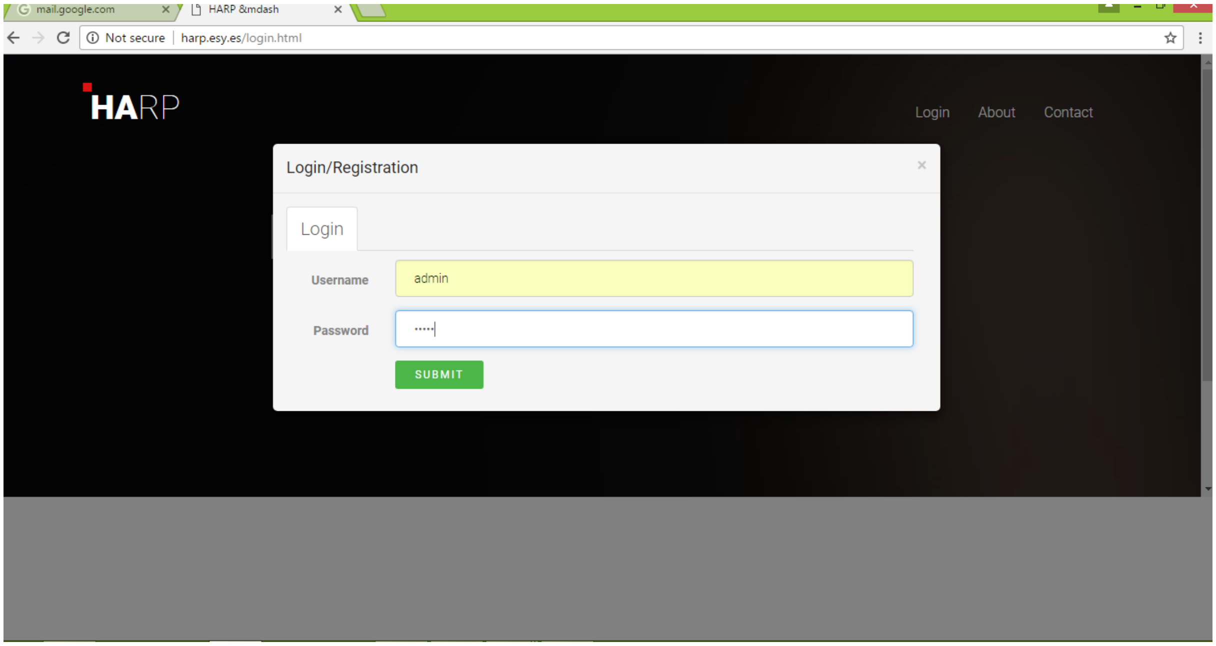
- The web portal is accessed via Desktop Browser or the Android Application
- The user logs in using his/her login ID
- The appliance status is displayed on the screen
- The user clicks the button corresponding to the appliance to be controlled
- The user is notified that the appliance has been switched on/off

SUPPORT & MAINTENANCE

jilut95@gmail.com

A3 SAMPLE SCREEN SHOTS

A3.1 LOGIN PAGE



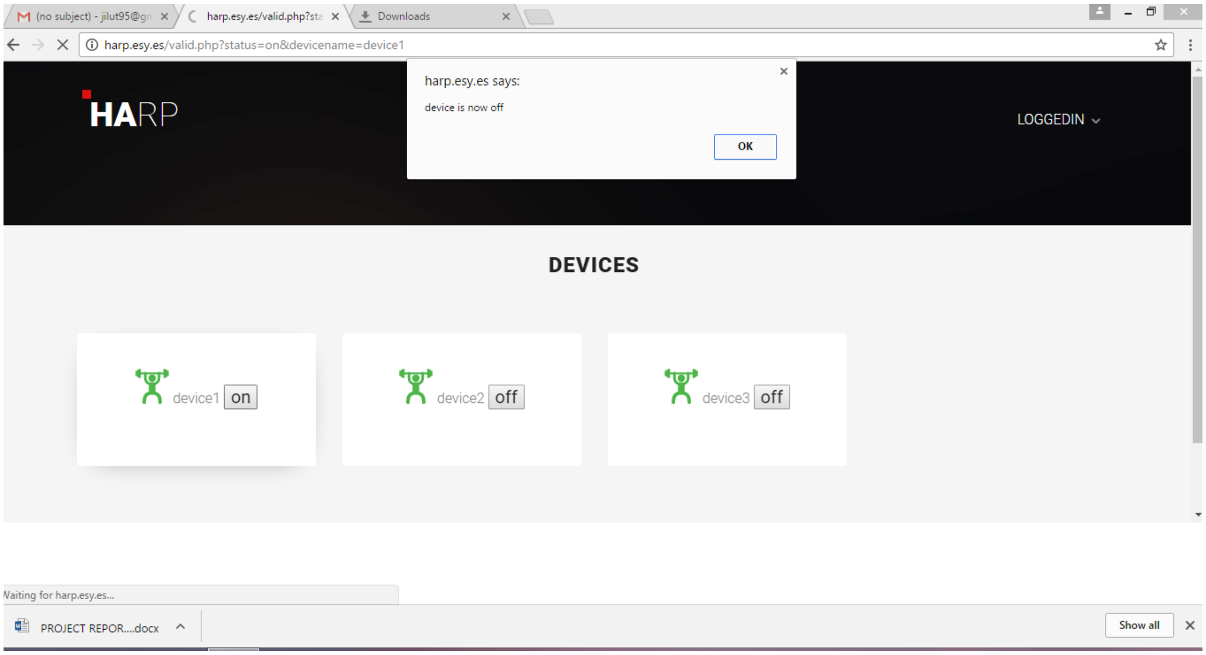
Not secure | harp.esy.es/login.html

Login/Registration

Username: admin

Password: •••••

A3.2 DEVICE STATUS: DEVICE1 IS PRESSSED



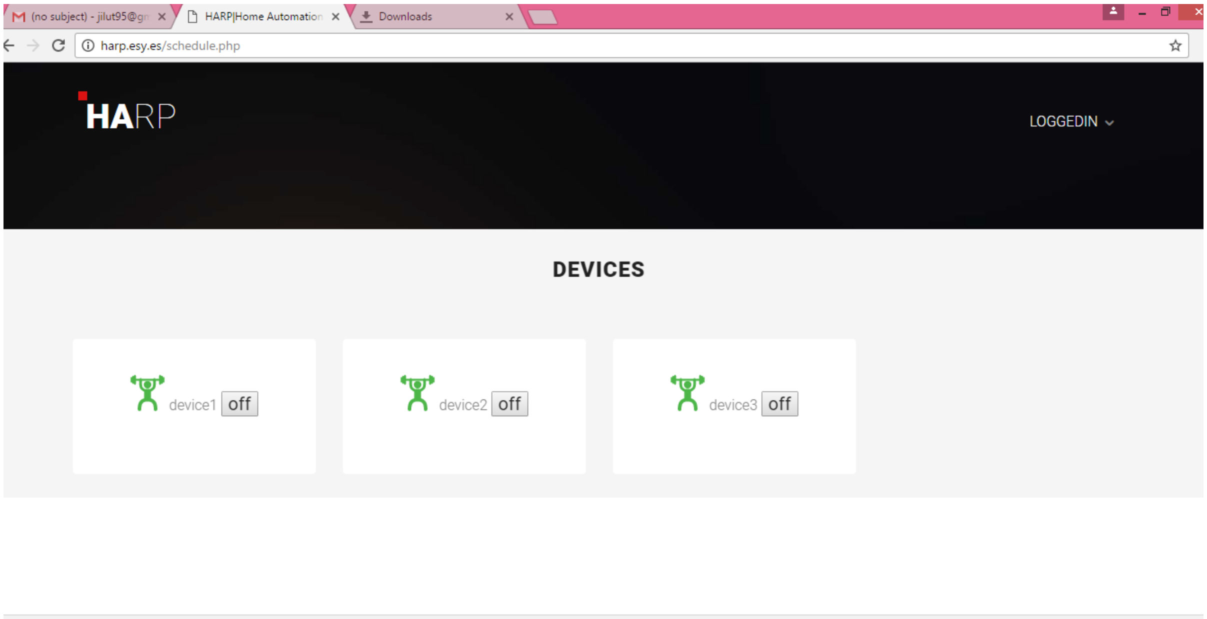
harp.esy.es/valid.php?status=on&devicename=device1

harp.esy.es says:

device is now off

Waiting for harp.esy.es...

A3.3 THE DEVICE1 IS NOW OFF



A4. SAMPLE SOURCE CODE

```
import RPi.GPIO as GPIO
import urllib2
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(5,GPIO.OUT)
GPIO.setup(6,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)
true = 1
while(true):
        try:
            for line1 in urllib2.urlopen('http://www.harp.esy.es/device1.txt'):
              status1 = line1
            for line2 in urllib2.urlopen('http://www.harp.esy.es/device2.txt'):
              status2 = line2
            for line3 in urllib2.urlopen('http://www.harp.esy.es/device3.txt'):
              status3 = line3
        except urllib2.HTTPError, e:
                    print e.code

        except urllib2.URLError, e:
                    print e.args

        print status1
print status2

print status3

if status1=='ON':

        GPIO.output(5,True)

elif status1=='OFF':

        GPIO.output(5,False)

                  if status2=='ON':

        GPIO.output(6,True)

elif status2=='OFF':

        GPIO.output(6,False)

                  if status3=='ON':

        GPIO.output(13,True)

elif status3=='OFF':

        GPIO.output(13,False)
```