Online Home Automation Control System
2019-2020 Graduation Project I

ريم علي الغامدي
437004875

ساره خالد آل حسين
436006939

ضحى نضال الزعبي
436200063

عبير أحمد عزت
436200058

منى سعود الخثلان
437004005

نوف عبد الله الدعجاني
437004100

Contents

List of Tables

List of Figures

Abstract

The aim for this project is to control lights, air conditioners, television or any other home appliance regardless of the person's location. The methodology is simple: an android app will send controlling requests to a web server. Raspberry Pi will be getting all the new requests from the server, processing it accordingly and controlling the hardware components connected to it. Such a system will allow someone in the United States to turn the lights in their house in Saudi Arabia on. However, an active connection to the internet must be present all the time.

1 Introduction

1.1 Problem statement & Significance

With the recent very rapid progress in technology and automation, there has become a need for remote control of almost all possible aspects of living, like using apps to control a cleaning robot or adjust the heating in the house or switch the house lights on or off. For the latter, there have been many applications that can do that, however they all work locally and there hasnt been one yet that uses the internet so it can be used remotely from outside the house to control the lights. A service like this is crucial for many people, like moms who are outside the house and want to switch the lights on at a certain time to wake their children up, or for a traveling family to turn on air conditioner before arrriving after summer vacation.

1.2 Proposed Solution

Because an important service like this is still not available, the proposed solution is as follows: we will create an app that should enable the user, by clicking on the appropriate buttons, to control a physical apparatus in the building where the lights are and have the lights turn on or off accordingly. We will do that by

1.3   Project Domain & Limitation

The domain in which our project operates in is home automation or domotics, and because our application will work through the Internet, it also operates in the domain of Internet of Things. The main limitation that we will face will be the time; given the period we have to work on the project, we may just be able to create an application that will control only lights, rather than other home appliances such as air conditioners, heaters and so on.

1.3.1   Domain

1.3.2   Limitation

1.4   Gantt Chart

2 Background Information & Related Work

2.1 Background Information

2.1.1 IoT

2.1.2 Hardware

- **Raspberry Pi:** a small general purpose computer. All hardware components will be connected to it. An active connection to the internet is needed for it to fetch data from the server.

- **Ubuntu Web Server:** hosts the web application. Digital Ocean severs were chosen for this project.

- **LED:** since the hardware components controlled depends heavily on the user needs, this project main aim will be controlling a small LED. LED stands for light-emitting diode. Basically a small light source.

- **Linear Solenoid:** once the LED works, linear solenoid will be installed for demonstrating the idea. It is a small component that generates a linear motion. It will be used to press in anything, such as lights, TV remote, and coffee machine.

2.1.3   Programming Languages & Frameworks

- **Python:** raspberry pi can be controlled by either c++ or python. Python was chosen because a REST API can be made using it fast.

  - **GPIO:** a library for controlling any hardware component connected to the GPIO pins.

  - **Flask:** a lightweight framework to build web applications.

- **Java:** mobile application are made in a native way with either swift or java.

  - **Android:** a framework for making android apps.

  - **Retrofit:** type-safe HTTP client for Android and Java. It will be used to send and receive commands and status from the web server.

- **PostgreSQL:** an open-source RDBMS. It will be installed on the server.

2.1.4   SDLC Model

2.2   Related Work

2.3   Proposed & Similar System Comparison

3 Requirement Determination & Analysis

3.1 System Analysis

3.2 Gathering Information

3.3 Requirement Specification

3.3.1 Functional Requirements

3.3.2 Non-Functional Requirements

3.4 Requirement Analysis

4 System Design

4.1 System Architecture

4.2 User Interface Design

5 Implementation

5.1 Implementation

5.2 Implementation Requirements

5.2.1 Software Requirements

5.2.2 Hardware Requirements

5.3 Implementation Detailed

5.4 I/O Screens

6 Testing

6.1 Testing

6.2 Test Plan

6.2.1 Unit Test

6.2.2 Functional Test

6.2.3 Acceptance Test

6.3 Test Items

6.3.1 Features to Be Tested

6.3.2 Schedule of Test Actions

6.3.3 Test Tasks

6.4 Test Case

6.5 Test Result

7 Conclusion

7.1 Conclusion

7.2 Evaluation

7.3 Future Work

Bibliography