Software Design Document for Power Automation system (Classroom)

Group ID : 2

December 2, 2018

Contents

1 INTRODUCTION

1.1 Design overview

We propose a system in order to save electricity majorly in public places, our focus being on a classroom. Our project is simply an attempt to save electricity, and the system we're proposing would have an edge over the others as it is inexpensive and can be used in any area, and also doesn't require any special equipment or authorization. Few methods have been applied previously in order to automate the electricity supply using Image Processing. Methods previously implemented in order to save electricity have utilized motion detection algorithms in order to decide whether or not the electricity supply must be given but this is not efficient enough, as a slight change in the orientation of the benches or so will also mark a change and hence electricity supply will be provided. Hence, our approach would constitute not only the motion detection algorithm but also the face detection algorithm in order to know whether or not a human is present inside, hence only in the presence of a human will the supply be provided, assuring our main motive, which is saving electricity. We will also be providing a threshold in motion detection algorithms, so that minute changes which are minimal, shall be ignored. The usage of motion detection and face detection algorithms individually will not be efficient enough. Hence we'll be trying to use the combination of both and increase the efficiency of this system. The proposed system is inexpensive and can easily be used in public places.

1.2 Requirements Traceability Matrix

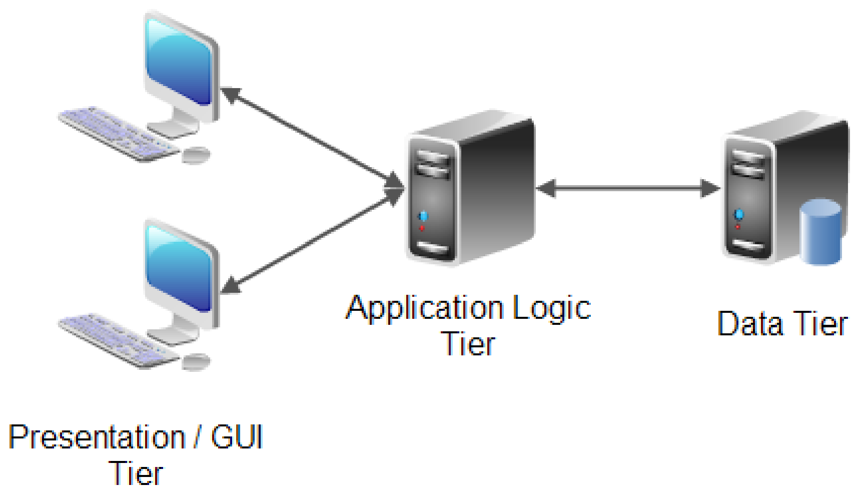
Requirements	**Camera**	**Raspberry-Pi**	**Relay Circuit**	**Led-lights**
System should allow camera to take live feed of the video	**X**			
System should able to detect faces efficiently	**X**	**X**		
System should able to detect motion efficietly	**X**	**X**		
All the modules should be properly interfaced	**X**	**X**	**X**	**X**
System should able to cut off and on the power supply efficiently		**X**	**X**	**X**
System should able to detect face and motion simultaneosly	**X**	**X**	**X**	**X**
Detect to able to switch off the power supply if there is no motion or face detection	**X**	**X**	**X**	**X**

2 SYSTEM ARCHITECTURAL DESIGN

2.1 3-Tier Architectural Style

N-tier and 3-tier are architectural deployment styles that describe the separation of functionality into segments in much the same way as the layered style, but with each segment being a tier that can be located on a physically separate computer. They evolved through the component-oriented approach, generally using platform specific methods for communication instead of a message-based approach.
N-tier application architecture is characterized by the functional decomposition of applications, service components, and their distributed deployment, providing improved scalability, availability, manageability, and resource utilization. Each tier is completely independent from all other tiers, except for those immediately above and below it. The nth tier only has to know how to handle a request from the n+1th tier, how to forward that request on to the n-1th tier (if there is one), and how to handle the results of the request. Communication between tiers is typically asynchronous in order to support better scalability.
There are three separate logical parts, each located on a separate physical server. Each part is responsible for specific functionality.
The business layer is deployed behind a firewall, which forces the deployment of the presentation layer on a separate tier in the perimeter network.The presentation layer is deployed on client machines and the business layer and data access layer are deployed on one or more server tiers.

The main benefits of the N-tier/3-tier architectural style are:
1 .Maintainability. Because each tier is independent of the other tiers, updates or changes can be carried out without affecting the application as a whole.
2. Scalability. Because tiers are based on the deployment of layers, scaling out an application is reasonably straightforward.
3. Flexibility. Because each tier can be managed or scaled independently, flexibility is increased.
4. Availability. The modular architecture of enabling systems can be exploited using easily scalable components, which increases availability.

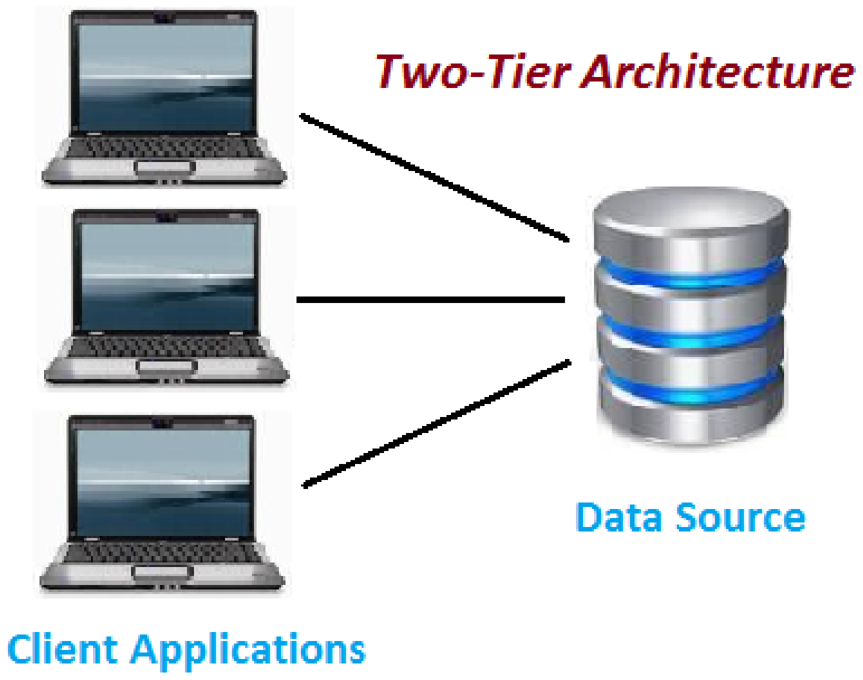


Problems:
Increased complexity/effort- In general is more difficult to build a 3-tier application compared to a 2-tier application because the points of communication are doubled.

2.2 Discussion of Alternative Designs- Client/server architectural style

The client/server architectural style describes distributed systems that involve a separate client and server system, and a connecting network. The simplest form of client/server system involves a server application that is accessed directly by multiple clients, referred to as a 2-Tier architectural style.
Historically, client/server architecture indicated a graphical desktop UI application that communicated with a database server containing much of the business logic in the form of stored procedures, or with a dedicated file server. More generally, however, the client/server architectural style describes the relationship between a client and one or more servers, where the client initiates one or more requests (perhaps using a graphical UI), waits for replies, and processes the

replies on receipt. The server typically authorizes the user and then carries out the processing required to generate the result. The server may send responses using a range of protocols and data formats to communicate information to the client.



Problems:
1. Because all business logic was implemented on the Client application, the code enforcing this business logic was spread all across the network and duplicated on each workstation. Changes to business logic or business rules usually implied redeploying new client software to all users. A big administrative effort at best, and downtime for large parts of the workforce at worst.
2. It makes the system vulnerable to attacks, as client systems can be compromised. With client software deployed on hundreds of computers throughout the company or even outside the controlled network, it cannot be guaranteed that malicious users do not try and succeed at "hacking" the client software or even write their own replacement software – directly accessing the database and bypassing all enforcement of business rules altogether.
How chosen system architecture avoids these problems: Multi-tier architecture solves this problem by simply partitioning data access into more tiers than the traditional Client/Server model (which is also sometimes referred to as two tier), with each tier performing the tasks for which it is best suited and

can be trusted.

2.3 System Interface Description

Description of user interface representations provided in section 4 of this document
As our project is completely hardware based, we won't be having any user interface in particular, but we have a number of components which have been explained in the same document further.

3 DETAILED DESCRIPTION OF COMPONENTS

3.1 Component-1- Camera Component

Camera component will basically consist of a CCTV Cameras in order to record the video of whatever is going on in the classroom and accordingly taking the action. Basically this component will be important to get live video and find out whether students are present in the class or they're not.

Component-Validate.jsp

Responsibilities- Recording of whatever is going on the classroom at a given time and capturing frames in given intervals.

Constraints-This component will be active when required and not 24 hours and the lighting shall be taken into consideration too.

Composition-Since this is already a sub-component, all functional responsibilities are covered. In other words, there are no sub-components.

Interactions-This component interacts with the other components in order to automate the power supply.

3.2 Component-2- Raspberry Pi component

There is a need for us to Connect the computer with the hardware of relay circuit, this calls for an appropriate microcontroller, it is better to use Arduino or Raspberry Pi microcontroller as a utile platform.
Component-Associate.java

Responsibilities: The responsibility of this component is to provide a proper interface between the required components.

Constraints:Some libraries that are working fine on your laptop may give you some surprises when you try to use them on your RPI .

Composition-Since this is already a sub-component, all functional responsibilities are covered. In other words, there are no sub-components.

Interactions-This component is basically used in order to provide an interface between the computer and relay circuit.

3.3 Component-3- Relay circuit component

It is used to control the power i.e. to on and off the lights(Here Led light in our project). i.e the decision made will be sent to the relay circuit in order to take the appropriate action.

4 System Architecture

5 Design Document

5.1 Level 0 DFD with description

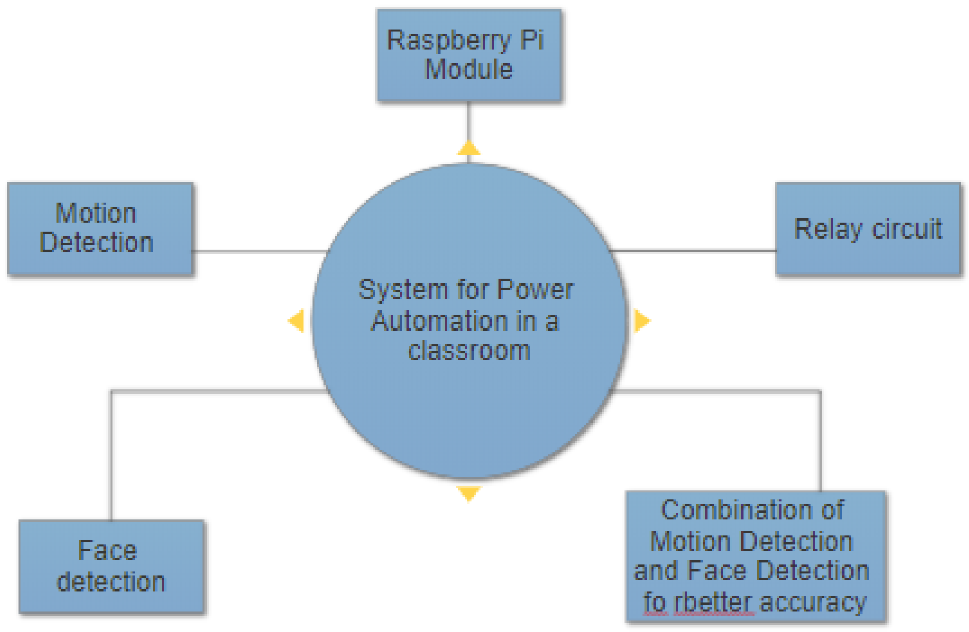
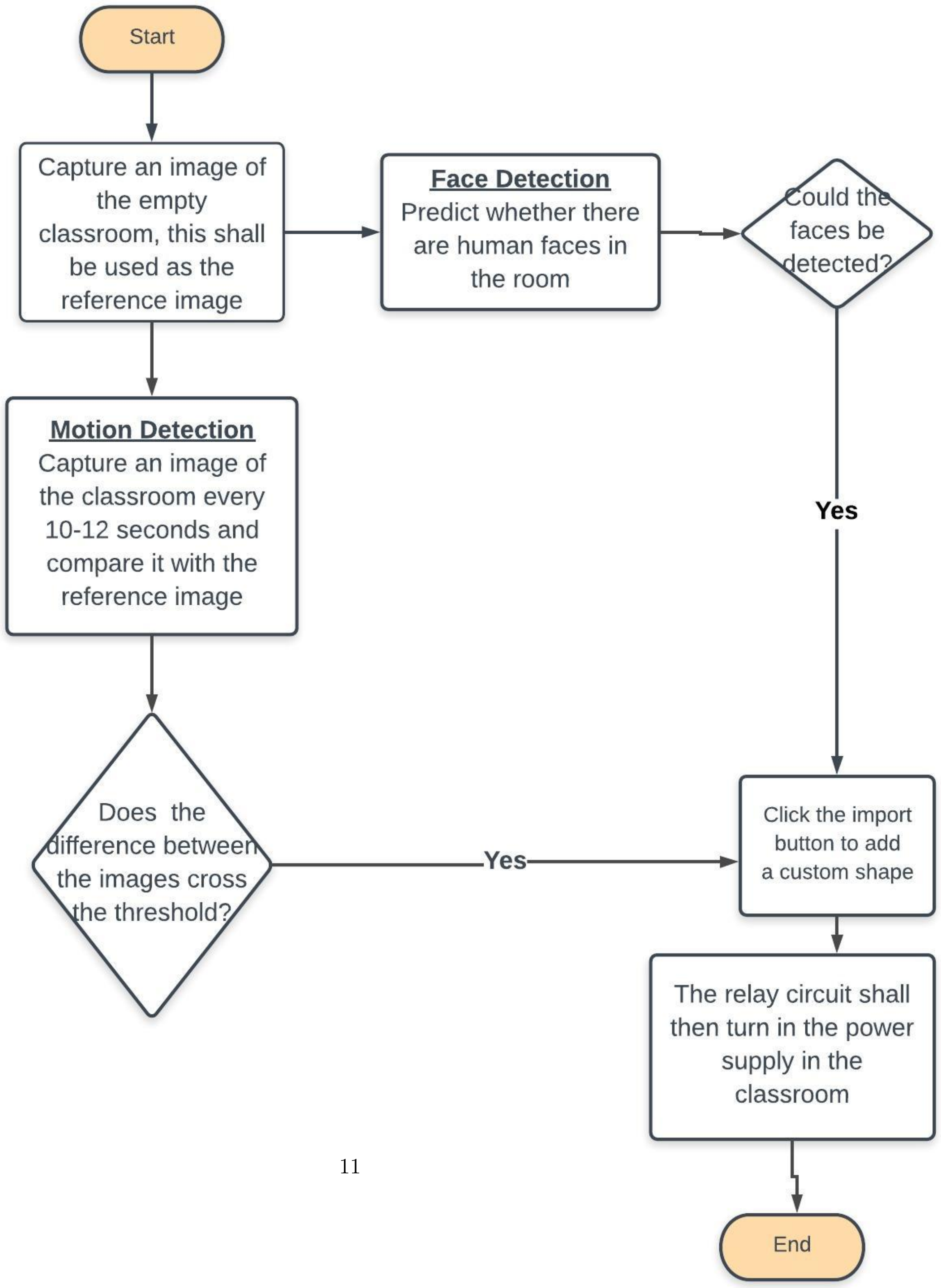

Raspberry Pi Module
Motion Detection
Relay circuit
System for Power Automation in a classroom
Face detection
Combination of Motion Detection and Face Detection fo rbetter accuracy

Figure 1: Level 0 DFD

5.2 DFD with description


Start
Capture an image of the empty classroom, this shall be used as the reference image
Face Detection
Predict whether there are human faces in the room
Could the faces be detected?
Yes
Motion Detection
Capture an image of the classroom every 10-12 seconds and compare it with the reference image
Does the difference between the images cross the threshold?
Yes
Click the import button to add a custom shape
The relay circuit shall then turn in the power supply in the classroom
End