Mansoura University
Faculty of Computers and Information
Computer Science Department

B.SC. PROJECT (2016/2017)

Integrated Smart System for Home Automation and Smart Car

(IControl)

Supervised by

Prof. Mohamed Fathy Al-Rahmawy

Student Name

Mohamed Elsayed Mohamed Sharshoup

mohamed_elsayed.1995@yahoo.com
www.linkedin.com/in/mohamed-elsayed-mohamed
https://www.facebook.com/mohamed.elsayed.1995

Table of Contents

Introduction

Modern technologies have evolved in last years that enable smart and more comfortable lives for the human beings. Internet of Things (IoT) is one such technologies that connect appliances to the internet for easier and location independent control of such appliances through the mobile and internet networks. In this project, we aim to develop an IoT system that enable the normal users to have a smart life by having a control over their homes and he appliances in them in addition to an integrated safer car driving system that monitor them during while driving to ensure that they do not fall asleep while driving. The system works through socket communication and mobile phones to ensure the highest levels of flexibility for its users and it was developed using state of the art algorithms and technologies for efficient and reliable operation.

Purpose

The purpose of this software design document is to provide a clear description of our system like

1) What system can do?
2) How it works?
3) Algorithms and methodology.
4) User interface and interaction.
5) Test cases and expected results.

Goals and objectives

1) Design an Intelligent system for home and car.
2) Controlling home devices remotely and reducing car accidents when the car driver sleep/drowsy while driving.
3) This can be useful for handicapped people which can make their lives easier and secure.
4) The final product must be fast, efficiency, ease to use and secure.

Project overview

We can divide our system into four subsystems which integrate together: -

1) Face Recognition.
2) Eye Blink.
3) Home control and Information.
4) User interface (IControl Application).

Core features

In this section, I'll briefly describe all subsystems components which we call them core features.

1) Face Recognition
   - This feature for securing our home from unauthorized users to entering our home, we just set a camera instead of magic eye on the door and when anyone front of the door the camera will recognize him and open the door if he/she is authorized user or send his/her image to all owners and tell them that this person tried to access the home.
   - The idea behind this feature is to reduce efforts for handicapped or blind people from opening the door.
2) Eye Blink
   - This feature in owner's car which monitoring car owner and produce alarm when the driver sleeping or close his/her eyes while driving, also when the car owner not focuses on the road the system will alarm him/her to watching the road.
   - Another sub feature embedded with this feature is opening/ closing the garage door automatically when the car is near to or far from the home.
3) Home Control and Information
   - This feature allows interaction between the end user and the system by using our android application
     - **Home Control**
       Control all functions and actuators in our home like opening/closing windows or Light, setting alarm, control music player, etc.
     - **Information**
       Getting some information about our home and the states of all components and this information includes: -
       1. System state (Is android app connected with the system ?!).
       2. Getting temperature degree of kitchen or any room.
       3. States of some actuators (Is bed room window open or close?!)
       4. State of alarm (on/off).
       5. State of unauthorized user notifications.
4) User interface (IControl)

   The (UI) for the end user which developed to be an android application that can have: -

   1. Control
      This is the main activity of our application which allows user to control the home and getting some information about the system.
   2. Check access
      In this activity, User check who tried to access the home and failed by to recognize him/her and the owner decide if this person known for him or not.
   3. Setting Alarm
      In this activity, User can set his/her alarm to wake him/her up, starts café machine before alarms on and also can open bed room window after alarm.
   4. Outside mode
      This mode switching all lights and windows off only in one click.
   5. Refresh
      Reconnect the application with the main system ().
   6. Music player.

Functional and Non-functional requirements

Functional	Non-Functional
Face Detection	Reliable AMAP in face recognition and Eyeblink
Face Recognition	Fast and Real Time System
Control Home	Alarm Starts After 10 Iteration When Eye Closed
Eye Detection	Start Café Machine before Alarm Starts
Eye Blink	Open window after Alarm Starts
Setting Alarm	Update System Information every 1 Minute
Control Garage	Save Image for Every One Tried to Access the Home
Outside mode	Support Night Vision Detection

Tools, Software and Hardware resources

Software	
Java	I'm using java programming language in our android application.
Python	Used to build system's classes on raspberry pi.
MySQL	Relational database to store all authorized and unauthorized users' information.
Arduino C	Used to get all information from sensors and send this information to raspberry pi.

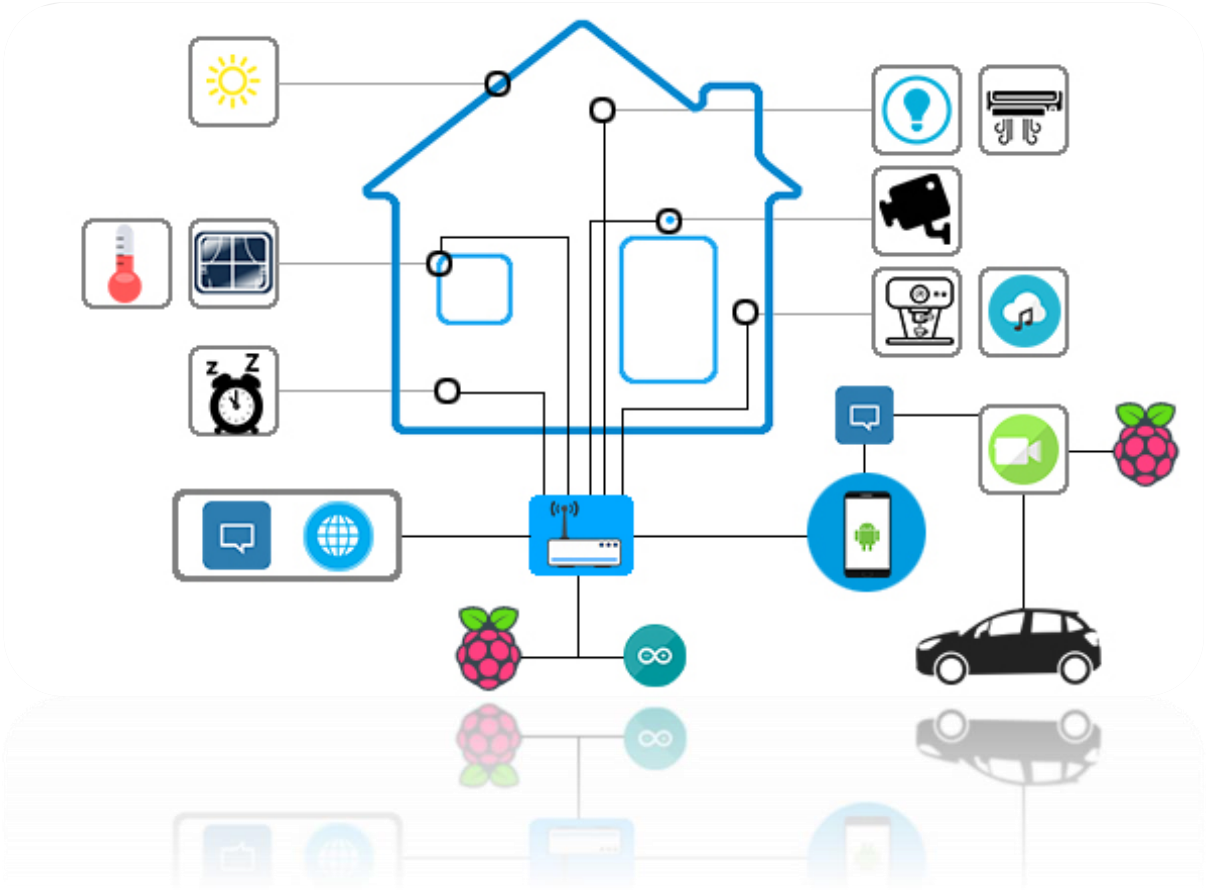
Hardware	
Raspberry pi 3	The system must contain two raspberry pi, one for the home which contains all classes and main functions of our system and use WI-FI for communication between it and the user, the second one contains all classes for eye blink detection on the car and it will use for more functions in the future plan.
Arduino Uno	Getting information from sensors, we can use raspberry pi instead it but raspberry pi doesn't support analogy input and it can't cover all functions of the system and Arduino will help in our future plan.
USB Camera	Also, we use two USB cameras, one for face recognition in home automation and the other one in the car, and we can replace this camera with raspberry pi camera which will be work better than USB cameras.
L 293D IC	Two ICs used to control DC Motors and turning them from left to right or reverse.

Sensors	
DHT22	DHT22 used to measure the temperature and humidity of any room and we use it in our bed room.
LM35	Used to measure the temperature of any room and we use it in our kitchen to measure its temperature and run the fan if it is high.
LDR	We use two LDRs which measures the light degree of current room or of nature light depend on usage.

Actuators	
DC Motors	We use basic DC-Motors, actually these motors I'm getting them from my joystick which acted as a vibration, you can use servo motors or steeper motors instead. These motors used for: - 1) Open/close bed room window. 2) Open/close garage door. 3) Running kitchen fan. 4) Running living room fan which simulates the air condition.
Buzzer	Two Buzzers one for alarm, and the other one is for alarming user when he/she sleep while driving.
Speaker	This speaker used for music player and connected with raspberry using audio jack port.
LEDs	Used for lighting and check if the user authorized to open the outside door (Green) or not (Red).

Other tools	
Thing Speak	ThingSpeak is an online platform for storing and visualizing your sensors data, and we use it to store the longitude and latitude of the car location.
Software ideas modeler	Used for UML design
Fritzing	Used for making electronic circuits schemas
Photoshop	Used for architecture design
Android studio	Making our android application
SQLite studio	Used for making our relational database
Python 2.7 IDE	Used for making our systems classes
Arduino IDE	Used for making our Arduino program
MS Project	Used for making our schedule plan
MS word	Don't forget this again

-



Icon	Usage
	Raspberry pi controller which contains our home features or our car eye blink.
	Arduino controller that get all sensors data.
	Specify current mode of day sunrise, sunset, sunshine or night.
	Specify the temperature of any room.
	The bed room window.
	Alarm.
	Thing Speak logo.
	Represent any light source.
	Represent our air conditional.
	Face recognition camera.
	Café machine.
	Music player.
	Eye Blink Camera.
	Our android application.

This diagram shows all system components that we can divide them into two main parts: -

1) Home Automation

   Contains raspberry pi and Arduino controllers which represent the kernel of the system and these controllers can: -

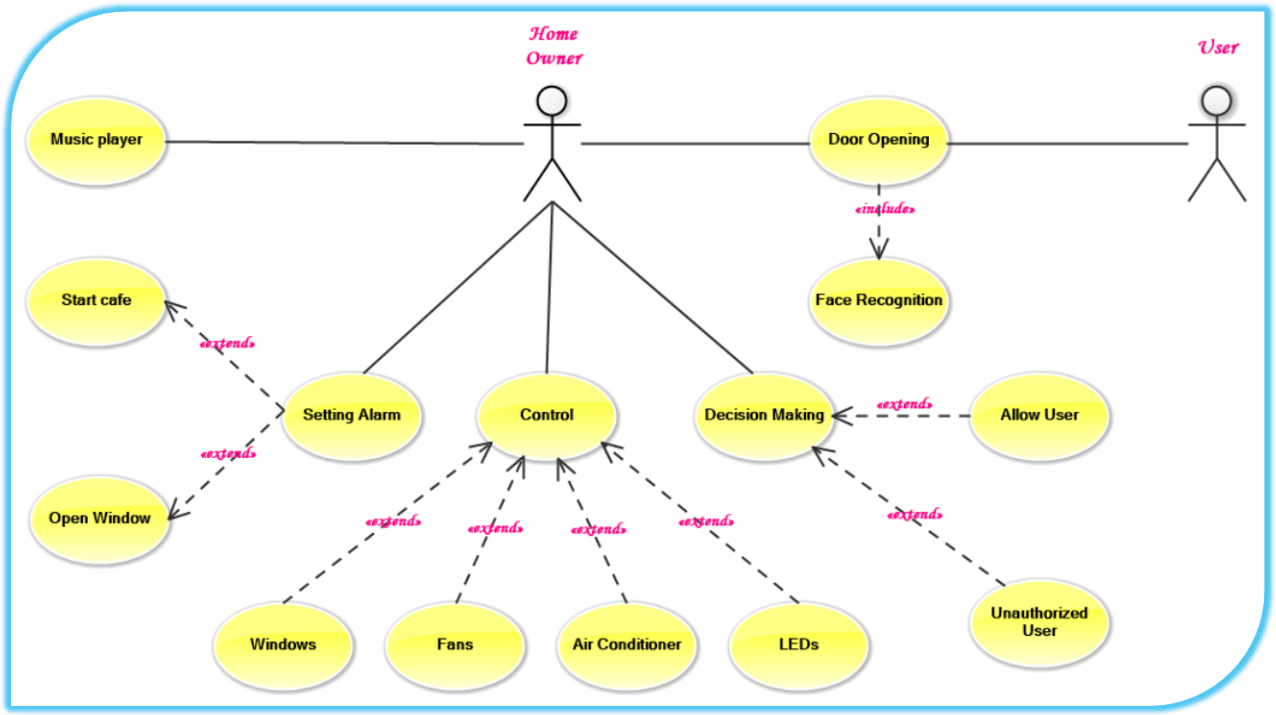
   - Get all sensors reading by using Arduino and transfer these readings to raspberry pi for processing by serial communication.
   - Control all actuators like LEDs and Windows by using raspberry pi and receive the commands transferred from user's Application using socket communication (LAN).
   - When there is any person near to the door the face recognition module will recognize that person, and send his/her image to all available owners if this person unauthorized.
   - Receiving all instructions that control actuators and sending all sensors data to all users using socket communication.
   - Raspberry pi also checks if the car is near to home or not and if it is, the garage door will open when it is nearby 100m or lower or close when the car about 100m or more from the home, the location which make the raspberry pi make this decision getting from ThinkSpeak platform which update the location of the car every 15 seconds.

2) Car Driver

   here we can make the system detect eye blinks for the car driver and alarm him, also we update the location by using android google maps or using independent GPS in the future and update Think Speak fields with current location.

Now we will discuss our system UML diagrams which will make the system easy to understand :)

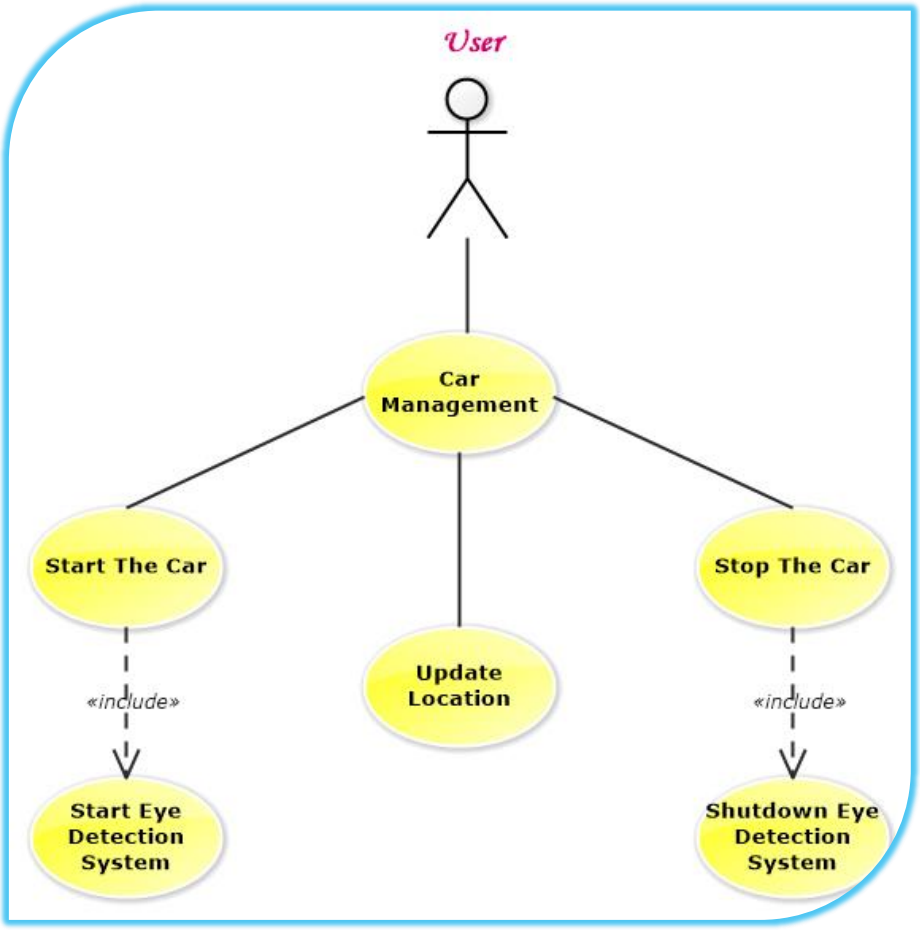
- Home automation



Case	Description
Door Opening	This function is active when we need to recognize a person in front of the door and it must run Face Recognition module for this operation.
Setting Alarm	In this case, the user already active the alarm and the system will start a new thread that make alarm deactivate until it is running, another two sub functions which they are [optional] for user are make the café machine running automatically before alarm starts to reduce wasted time when the user going to work, another function is to open window or curtains automatically.
Windows	In this case, the user can control windows states (Open/ Close) and these windows will open/close according to instruction from user.
Fans	We have only on fan in our system which is on the kitchen and this fan run automatically if the temperature of the kitchen is high and also stopping automatically if it is low, the user also can control the fan state (Open/close) and if the fan running based on high temperature on the kitchen the user can force stop it and it will ignore the high temperature for one minute then continue again in high temperature case.
Air condition	In this case, we simulate the air condition and control the speed of this air condition by the application, we make five speed degrees for this case.
LEDs	Also in this case, we simulate lamps in real homes, but generally the user can control this LEDs (ON/OFF) and control the degree of its brightness (25%, 50%, 75%, 100%).

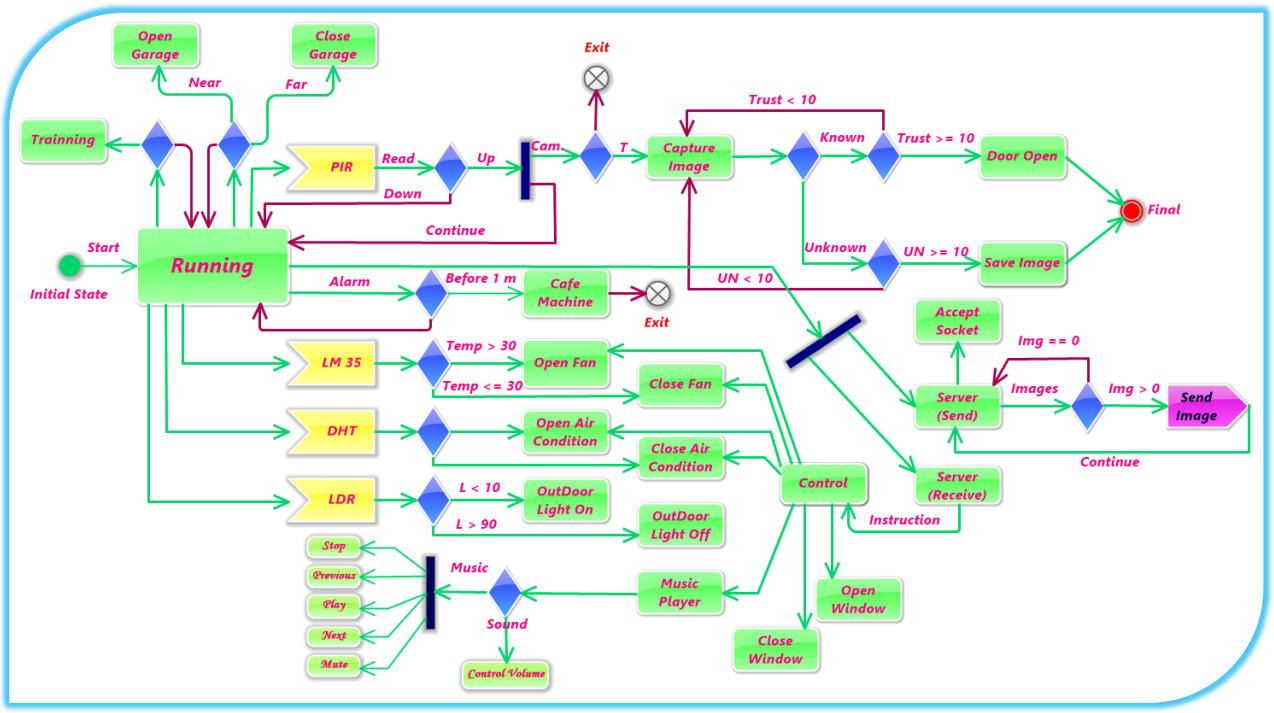
Case	Description
Decision Making	In this case, User making decision about unauthorized users when notification received, the user can recognize persons if he/she knows him/her and select which this image belongs to whom or he can make this person as unauthorized.
Music Player	Controlling the music player on the raspberry pi, first of all when the user opens the application the system will send all available music files from specific directory to the user then the user can choose any file to play, pause, resume or stop or control the volume of the music player.

- Car Driver



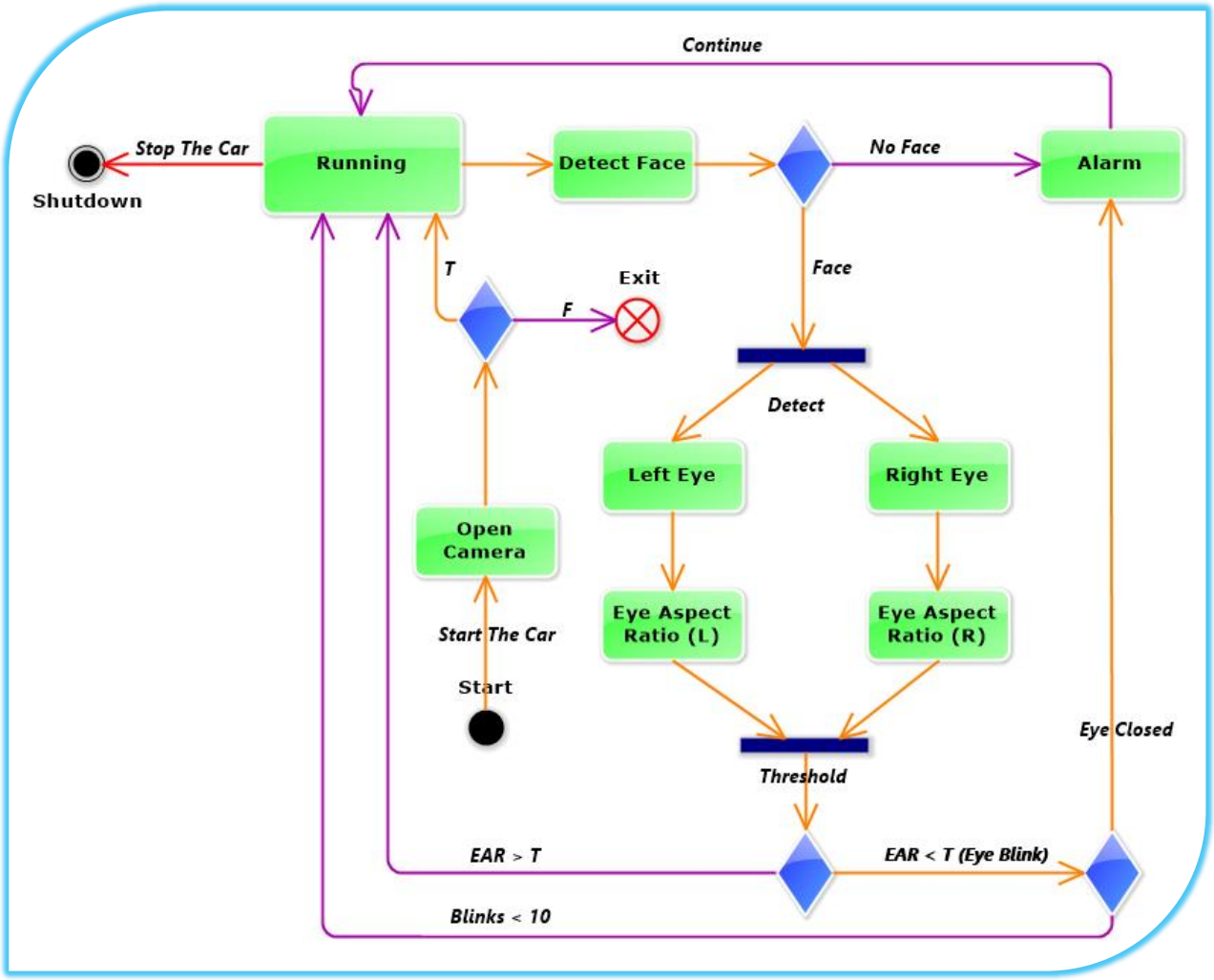
Case	Description
Car management	The user not interact with this case directly, but when he/she starting the car this function will start automatically to detect his/her eye if it is open or close, if the user stopping his car this function will stop automatically.
Update Location	Also, this feature starts automatically but after the user active location service (GPS) on his/her mobile, because we update the location of the car based on the mobile location which must be on his hand all time.

Home Automation



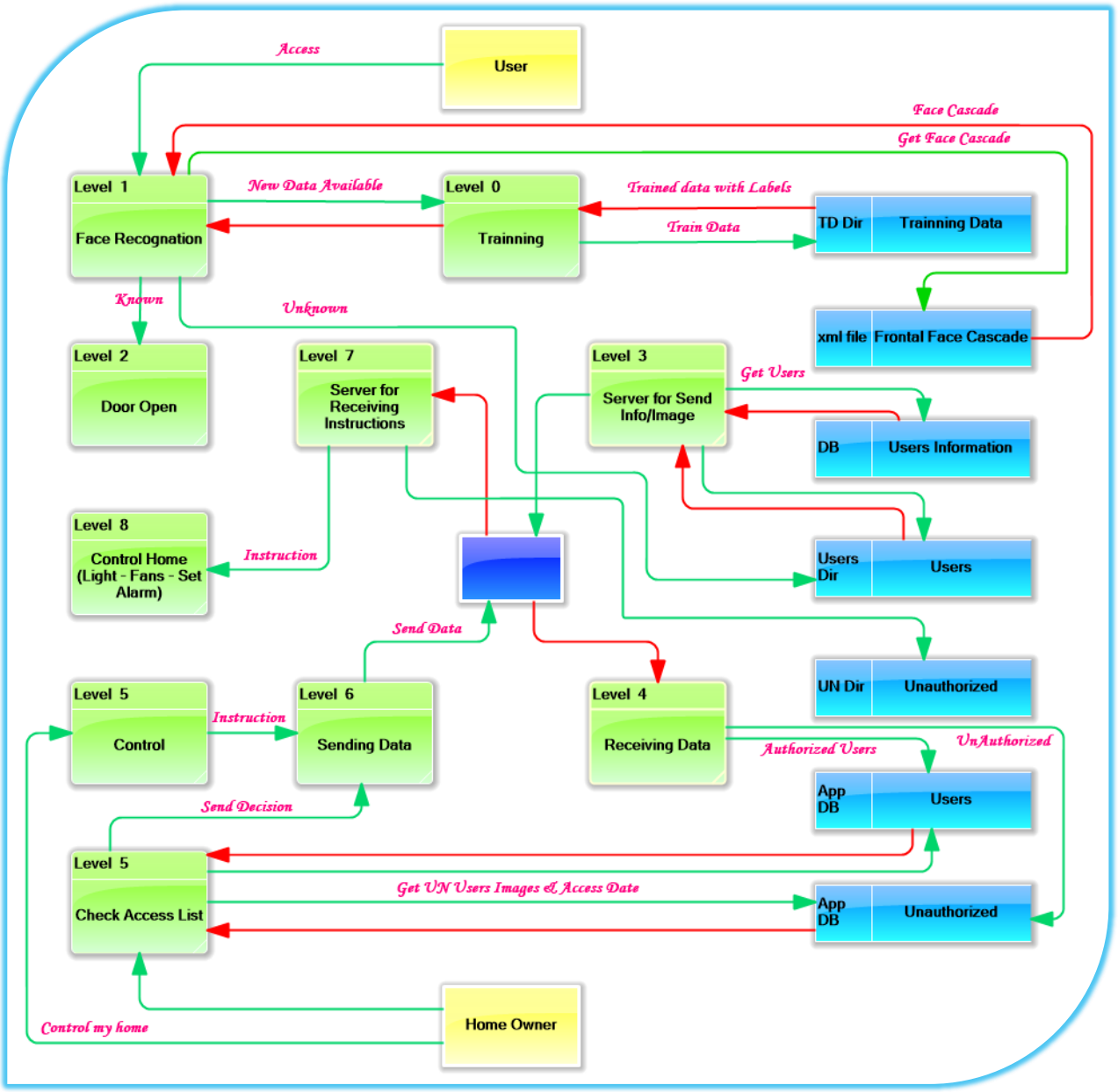
- This is our state machine diagram for the home automation subsystem.
- We starting from initial state to running state which: -
  1) Perceives instructions from users and convert it to commands to: -
     - Control Fan.
     - Control Air condition.
     - Playing music.
     - Opening/ closing window.
  2) Reading sensors data, as we can see from this figure there are four yellow reading sensors which we make decisions according to these readings.
  3) Open/close garage door when the car is near or far.
  4) Training the system when it starts for face recognition.
- Any instruction received by Server (Receive) and check what command can produce.
- Another state called Server (Send) which send all unauthorized users images to all available users.
- Control state convert any instruction message from user to a command and activate it.
- When PIR sensor detects any signal then it starts face recognition module to recognize user.

Car Driver



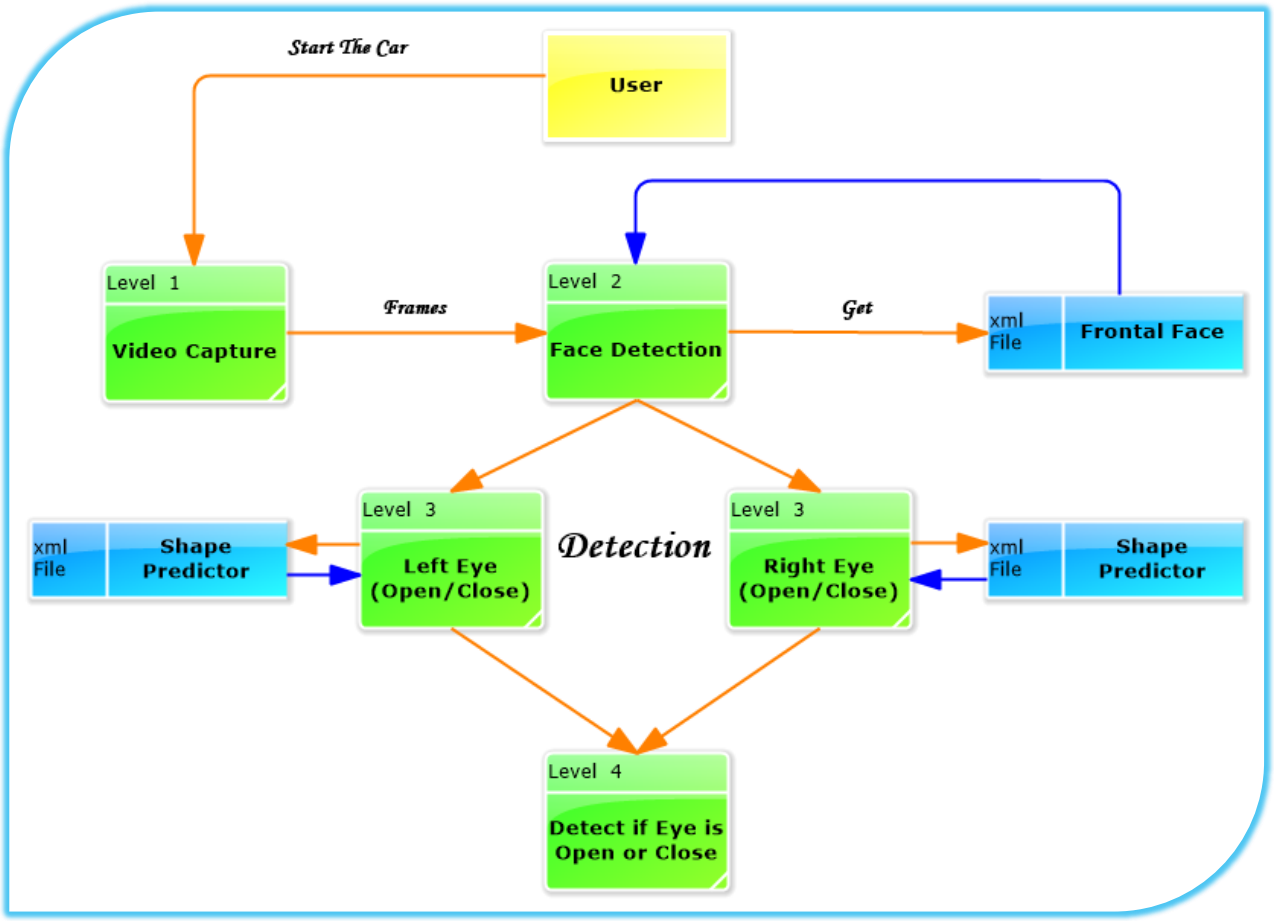
- This is the state machine which in our car.
- The system must run when the driver starts the car engine.
- The camera will start to detect car driver face in Detect Face state.
- If there is no face, alarm will start.
- After detecting face, it will detect left and right eyes from this face.
- After detecting eyes, the system will calculate the eye aspect ratio.
- According to the threshold the system will check if the eye was blinked or not.
- If eye blinked for 10 iterations continually then the eye was closed and the alarm will start.
- System will stop when the car stops.

Home Automation



Level	Description
Level 0	In this level, we get our users faces training images and train them using trainer class.
Level 1	In this level, the user stands front of the door and the camera recognize him/her after getting the training data from (level 0), then if the user known the door will open (Level 2), else the system will save this unauthorized user image in users' directory (Users Dir: Users) for transferring later to the home owners.
Level 2	An action for opening the door after recognition.
Level 3	In this level, we send unauthorized users images from (Users Dir: Users) to all owners using the WI-FI access in the middle with blue colour and if the image was sent successfully it will move from users' directory to unauthorized directory (UN Dir: Unauthorized) waiting for user's decision.
Level 4	This level in our android application and it receive images and some information from the system using the gateway then we save these images in unauthorized database (App DB: unauthorized), if the data contains users' information then it will store it into users' database table (App DB: Users) and use this information later in the GUI layer, if this information contains data from sensors we will use it for update the GUI information.
Level 5	We have two states in this level when the home owner accesses the GUI layer, first one when user checking access list which get all users information from database which we discuss it in (level 4), second state is control which allow user to control all systems features which discussed in use case diagram.
Level 6	This level also in our application and it used for sending the instructions and decisions about unauthorized users images.
Level 7	This level act as a server for receiving instructions and data, when the instruction related to unauthorized user image we check if the user authorized or not if it is we move the image from users' directory to the valid user, else we move the image to unauthorized directory.
Level 8	In this level, we control home fans, light, etc. based on user instruction.

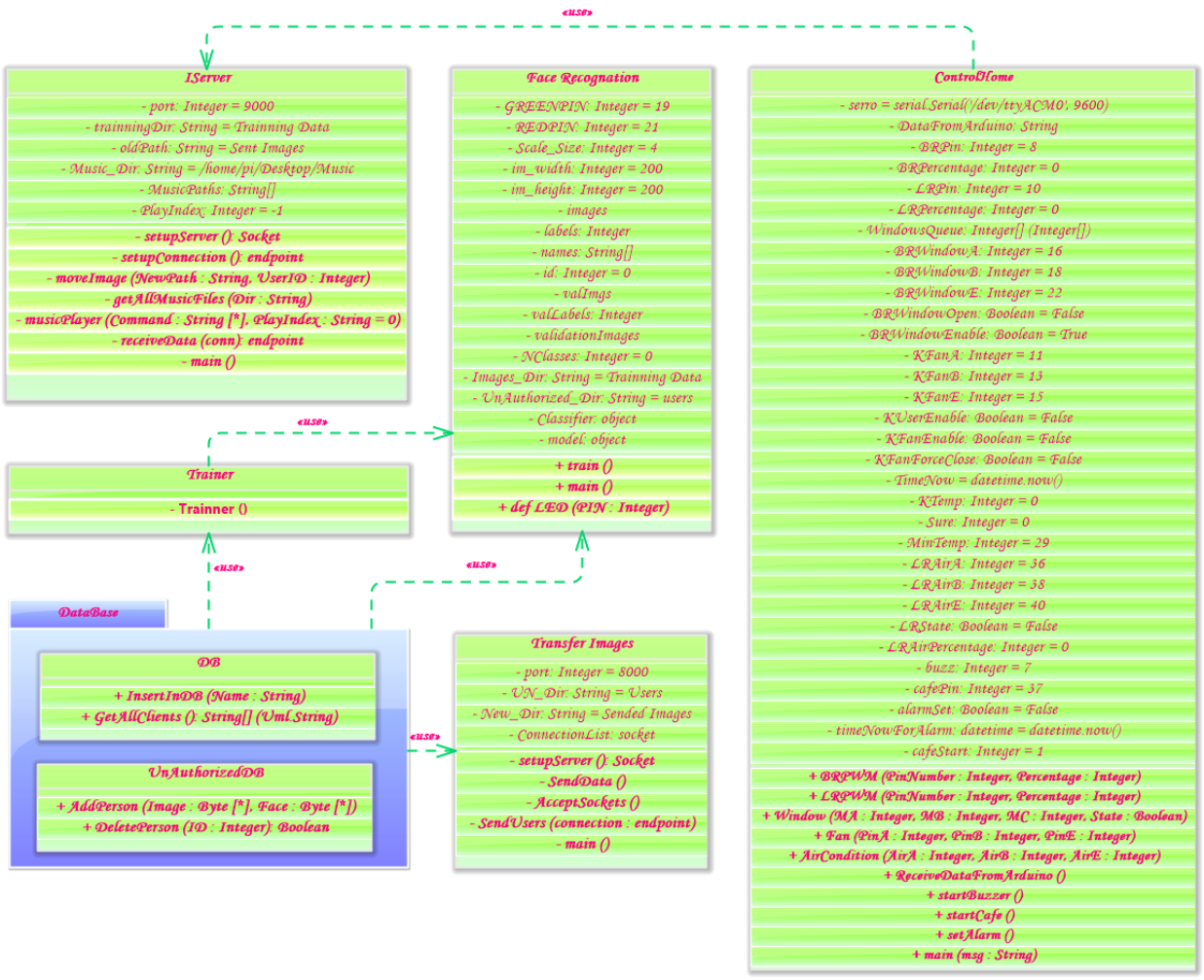
Car Driver

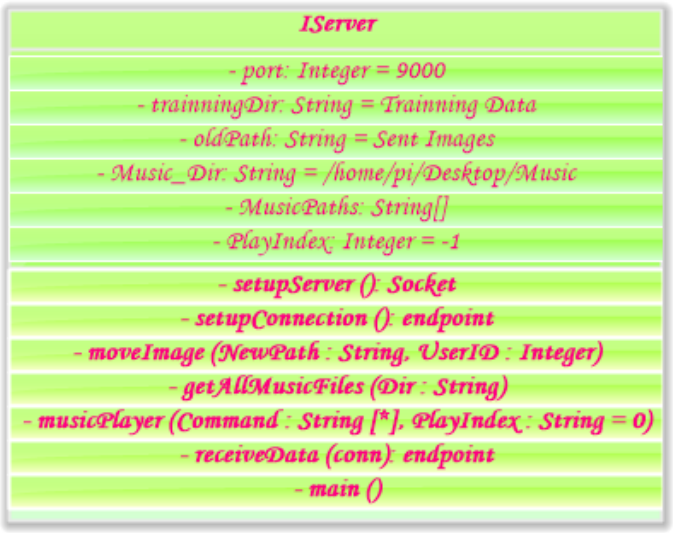


Level	Description
Level 1	When the user starts the car, this level will be activated to start video stream capture.
Level 2	This level starts to detect the driver's face using the frontal face cascade file (xml file: Frontal Face)
Level 3	Here we use the shape predictor (xml File: Shape Predictor) to detect the left or right eye and calculate eye aspect ratio for checking if eye was closed or still open.
Level 4	In this level, we checking if eye is open or closed and make a sound according to this checking.

Raspberry class diagram

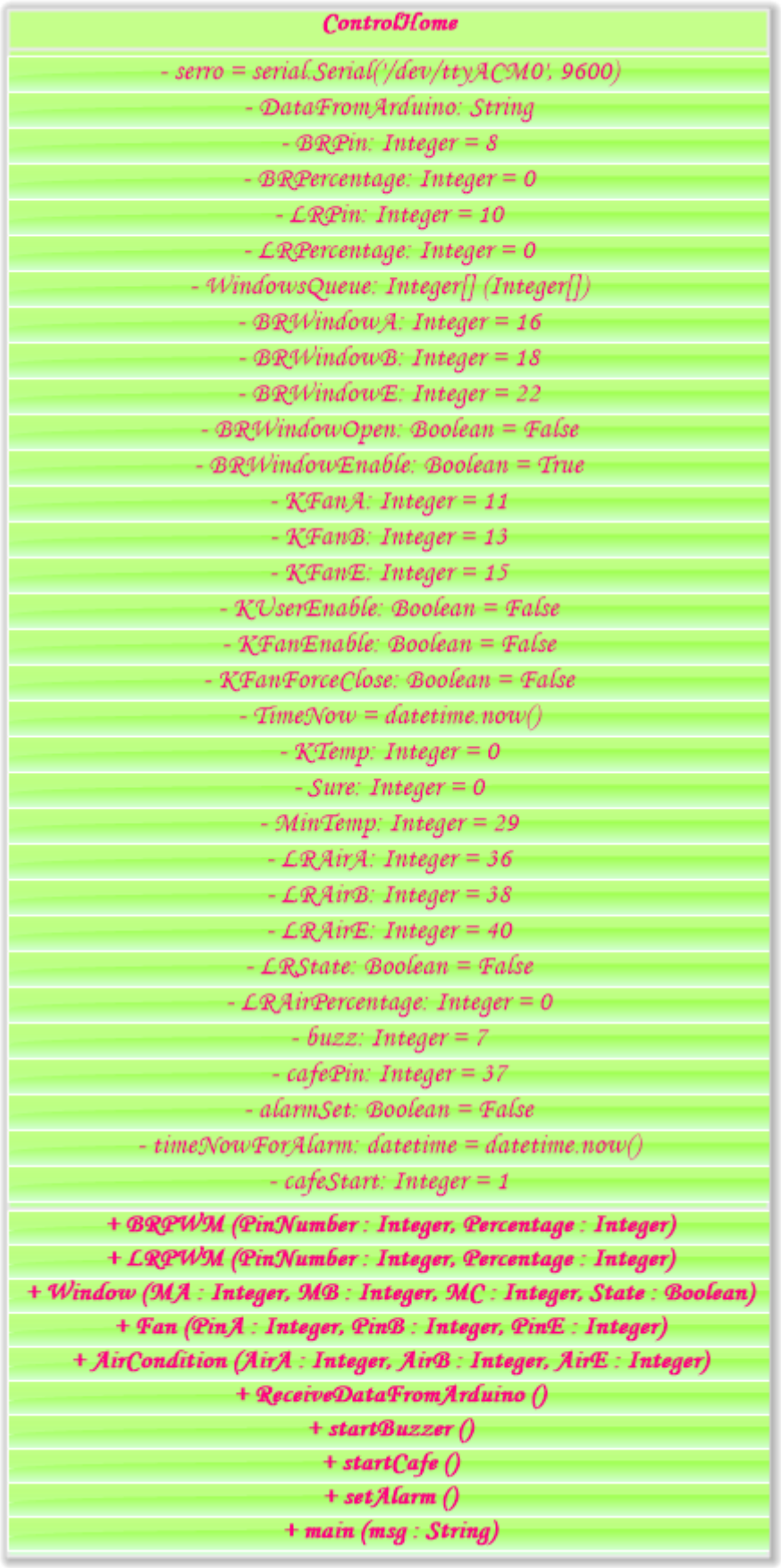
- We build these classes in our home subsystem which contains 7 different classes with its functionality and I'll describe each class variables, methods as much as possible.
- These classes are associated together and one or more class can be used into another as shown from this figure.





Global Variables	
port	This port is used for communication between this server and any device and we initialize it on port 9000.
trainingDir	String value that contains the training directory name to use it later when we need to train our dataset and this directory name is “Training Data”
oldPath	We talk about sending unauthorized users’ images to users before and every image sent successfully we move it to send images directory and after user’s decision we move this image to any authorized user training data or unauthorized directory so this oldPath directory represent the sent images directory “Sent images”.
Music_Dir	String value that represent the music directory “/home/pi/Desktop/Music”.
MusicPaths	List for all music files paths.
playIndex	The music file index which specified by the user to play music file.

Methods	
setupServer	This method initializes our server for accepting any IP address using port 9000.
setupConnection	This method waiting for any incoming connection, accept it and return the socket which communicate with it.
moveImage	This method used for moving images from directory to another and we call this method after any image sent to users.
getAllMusicFiles	Getting all music files and save its paths in MusicPaths list.
musicPlayer	This method is used for playing any music file, we sent the command “play”, “pause”, “resume” or “stop” and the index of the music file for this command.
receiveData	This method used for receiving data from user and split this data to extract the command to call the required method from ControlHome class which discussed below, also it used for sending information to users.
main	Main method for calling setupConnection and receiveData methods.


ControlHome
- serro = serial.Serial('/dev/ttyACM0', 9600)
- DataFromArduino: String
- BRPin: Integer = 8
- BRPercentage: Integer = 0
- LRPin: Integer = 10
- LRPercentage: Integer = 0
- WindowsQueue: Integer[] (Integer[])
- BRWindowA: Integer = 16
- BRWindowB: Integer = 18
- BRWindowE: Integer = 22
- BRWindowOpen: Boolean = False
- BRWindowEnable: Boolean = True
- KFanA: Integer = 11
- KFanB: Integer = 13
- KFanE: Integer = 15
- KUserEnable: Boolean = False
- KFanEnable: Boolean = False
- KFanForceClose: Boolean = False
- TimeNow = datetime.now()
- KTemp: Integer = 0
- Sure: Integer = 0
- MinTemp: Integer = 29
- LRAirA: Integer = 36
- LRAirB: Integer = 38
- LRAirE: Integer = 40
- LRState: Boolean = False
- LRAirPercentage: Integer = 0
- buzz: Integer = 7
- cafePin: Integer = 37
- alarmSet: Boolean = False
- timeNowForAlarm: datetime = datetime.now()
- cafeStart: Integer = 1
+ BRPWM (PinNumber : Integer, Percentage : Integer)
+ LRPWM (PinNumber : Integer, Percentage : Integer)
+ Window (WA : Integer, WB : Integer, WC : Integer, State : Boolean)
+ Fan (PinA : Integer, PinB : Integer, PinE : Integer)
+ AirCondition (AirA : Integer, AirB : Integer, AirE : Integer)
+ ReceiveDataFromArduino ()
+ startBuzzer ()
+ startCafe ()
+ setAlarm ()
+ main (msg : String)

Global Variables	
serro	This variable contains the communication port name that transfer data from Arduino to raspberry pi.
DataFromArduino	String variable that contains all data which sent from Arduino.
BRPercentage	Percentage of Bed Room light regularly can be (0 – 25 – 50 – 75 – 100).
LRPercentage	Same as BRPercentage but this for Living Room.
WindowQueue	Queue for saving all instructions about opening/closing any window.
BRWindowOpen	Boolean variable which indicate if the Window is open or close.
BRWindowEnable	Boolean variable which indicate if the window is opening or not if it is we set the instruction which want to open the window in WindowQueue.
KUserEnable	Indicate if the user Enable the kitchen Fan or not and this variable used for force close the kitchen Fan if it is running.
KFanEnable	Boolean variable that indicate if the kitchen fan enables or not!
KFanForceClose	Indicate if the fan was forced close or not.
TimeNow	Getting current time as we use it as starting time and after one minute if the kitchen fan was forced close we check the kitchen temperature again and running the fan if the degree is high.
KTemp	Get current kitchen temperature.
Sure	Make sure that the degree of temperature in the kitchen is high to run the fan.
MinTemp	Minimum temperature that make the fan running.
LRState	Check if living room air condition running or not!
LRAirPercentage	Living room air condition or fan speed percentage.
alarmSet	Boolean variable that indicate if the alarm is enabled or not.
timeNowForAlarm	Get current time which will be the initial time for running the alarm alter.
cafeStart	Check if user wants to make the café machine on before the alarm starts or not?!
...	Other variables are for GPIO Pins in the raspberry pi.

Methods	
BRPWM	This method acts as a thread and used to switch the Bed Room light on or off and control the brightness of this room which indicated by user.
LPPWM	This method like BRPWM method.
Window	This method used to open/close Bed Room window.
Fan	This method acts as a thread and it used to Open/ Close fan by the user instruction or if the temperature is high.
AirCondition	This method acts as a thread and it makes living room Air Condition on/off and control its speed.
ReceiveDataFromArduino	This method acts as a thread and it receive data from Arduino controller and set this reading to the DataFromArduino variable, also it checks if the temperature reading is high for three times the fan will start automatically.
startBuzzer	This method is used when alarm starts and it make sound.
startCafe	Start café machine one minute before alarm starts.
setAlarm	This method acts as a thread when the user setting alarm and it still running until the alarm starts, there are two optional features with it first is to start café machine before alarm, second is to open the bed room window.
main	Check incoming instruction from IServer module and run the specific method.

Global Variables	
images	List which contains all training images for training.
labels	List which contains all training labels or classes.
names	List which contains all names for each label or class.
id	Id increase automatically if there is new class.
valImgs	List which contains all validation images percentage that we split from all images.
valLabels	List which contains all validation labels.
validationImages	List which contains all validation images
NClasses	Integer variable which contains the number of classes we want to train.
Images_Dir	Directory name which include all training data directories (one for each user).
UnAuthorized_Dir	The directory name of unauthorized users' images.
Classifier	This is our harrcascade classifier object which we will use it to detect faces, this classifier will be discussed more in algorithms section.
model	This model object used for recognition, and we use FisherFaceRecognizer to recognize faces and we will talk about it in depth in algorithms section.

Methods	
train	We use this method for training our dataset when the system starts or when the training data updated and new images are added to it.
LED	Switch green LED on when the user was recognized by the system and this person known or switch red LED when the system can't recognize this person.
main	Start training and begin recognize any person near to the door.

Global Variables	
port	This our port which we can communicate with it and we set it to 8000.
UN_Dir	Unauthorized users' directory which we can get all images from it to transfer them to all users and we set it to "Users".
New_Dir	The images directory after we sent the image, we set it to "Sent Images"
ConnectionList	Save all connections into this list for using it later when we need to transfer image.

Methods	
setupServer	Initialize the server for accepting any IP address on port 8000.
sendData	This method acts as a thread and it used for getting all unauthorized users images and transfer it to the user, If the image was transferred successfully we will move it to the "Users" directory.
AcceptSockets	This method accepts new users who wants to join the server and save the connection into connectionList.
sendUsers	Sending all valid users information like (ID, Name) for the new socket.
main	Starting the server for accepting and sending data.

Methods	
Trainer	When we need to add new user in our system we need to capture some images for his/her face to recognize him/her face, so we use this class for training any user.

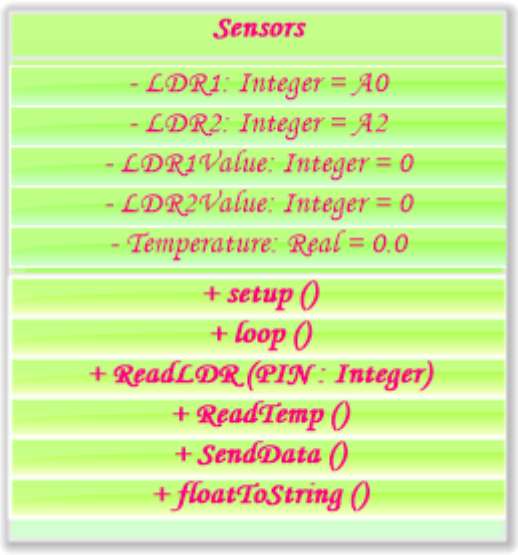
Methods	
Garage	This method used to open or close garage its input value.
main	Used for checking the location of the car and decide if it is near or far.

- In this package, we have two classes (DB and UnAuthorizedDB) and their methods are

Methods	
InsertInDB	Insert new user to our system, we just need his/her name and this method will add him/her to the database and getting the ID for this user and create directory for him/her in “Training” directory to train his/her face later.
GetAllClients	This method gets all users information from the database.

Methods	
AddPerson	Add unauthorized user information in the database like image and date of access.
DeletePerson	Delete unauthorized user information when the user recognizes him/her!

Arduino class diagram

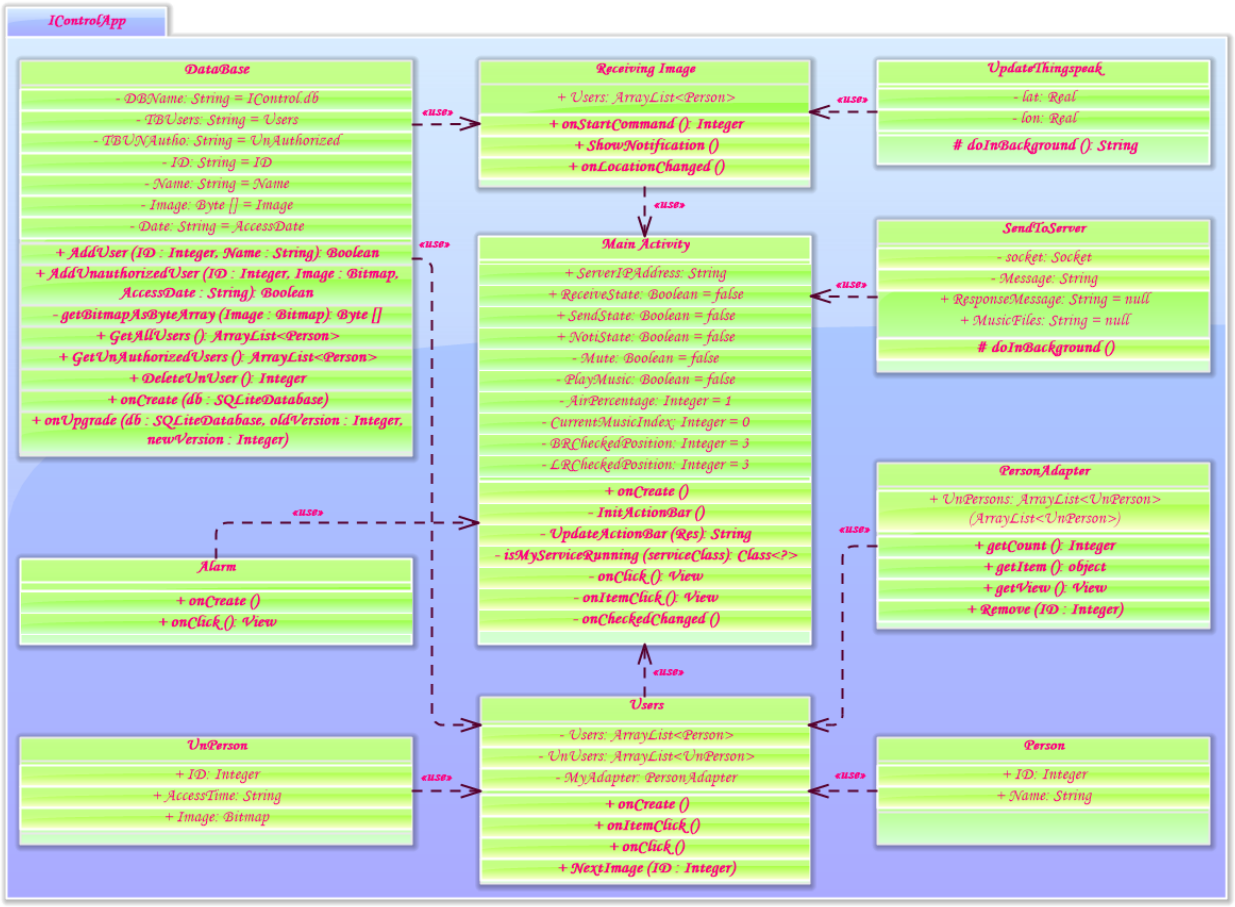


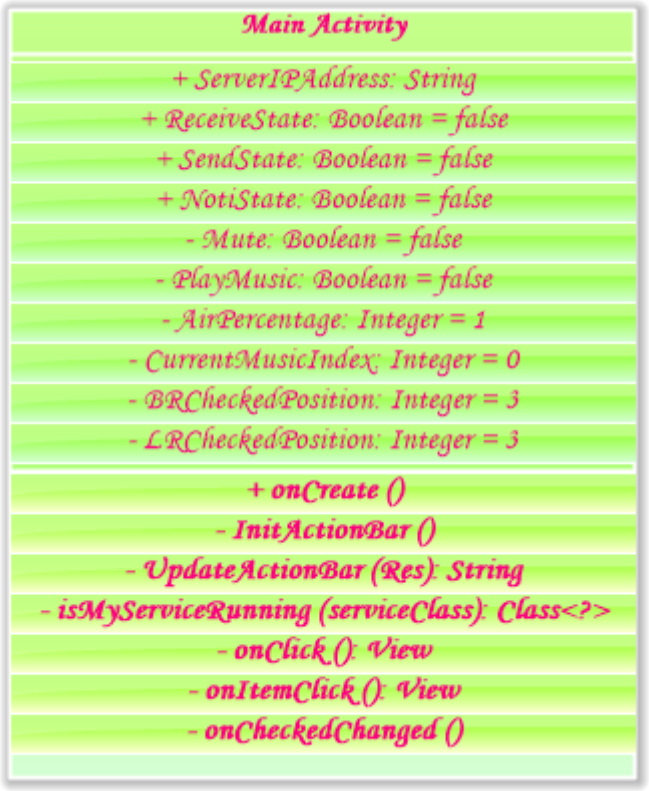
Global Variables	
LDR1Value, LDR2Value	These variables are assigned by the LDRs sensors values using Arduino.
Temperature	Value of kitchen Temperature, this temperature getting from LM35 sensor.

Methods	
setup	This method initializes all sensors for reading.
loop	This method used for reading all data from sensors.
ReadLDR	Reading light sensor value from specific Pin.
ReadTemp	Reading Temperature value from specific Pin.
SendData	Start sending data using serial port and 9600 bit-rate.
floatToString	This method converts float values to string.

Android class diagram

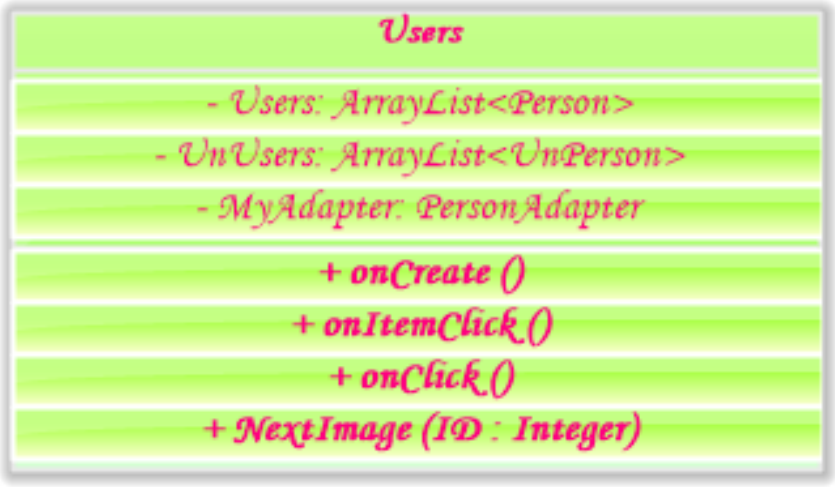
- Our main package (IControl) which contains all these classes.
- We build these classes in android application which contains 10 different classes with its functionality and I'll describe some of these classes.
- These classes are associated together and we one or more class into another as shown from this figure.





Global Variables	
ServerIPAddress	Static IP address for our system.
ReceiveState	Determine if the application connected with server for receiving images or not
SendState	Determine if the application connected with sever for sending instructions or not
NotiState	Determine if there is any notification for new unauthorized images or not!
Mute	Boolean variable to turn sound music on/off.
PlayMusic	Determine if the music player is playing a music now or it is stopped.
AirPercentage	Living room air condition or fan speed.
CurrentMusicIndex	Specify the current music file index, this index is sent to server for playing the specified music.
BRCheckPosition	Bed room light index which represent the percentage of light.
LRCheckPosition	Same as BRCheckPosition but for living room.

Methods	
onCreate	This method initializes the GUI for the user and manage some controls like starting connection with the system and getting all information.
InitActionBar	Use information from raspberry to update the GUI information for the user.
isMyServiceRunning	Check if the service which receive the unauthorized images running in background or not!
onClick	onClick method used when user click on some views like Buttons.
onItemClick	onItemClick used when the user checks any item from listView.
onCheckedChanged	Used when the user switch toggleButtons on/off.



Global Variables	
Users	List of all authorized users.
UnUsers	List of all unauthorized users.
MyAdapter	Custom adapter for showing any unauthorized user image and date of access.

Methods	
onCreate	Getting all users and unauthorized users from database and show them in GUI.
onItemClick	Used when user changes user name or unauthorized user image.
onClick	Confirm user and sending his/her information to the server for moving his/her image as authorized user or unauthorized user.
NextImage	Getting next image after button clicked.

Global Variables	
lat	Real variable that represent latitude of current location.
lon	Real variable that represent longitude of current location.

Methods	
doInBackground	Used to update the latitude and longitude of current location on thingspeak platform and use this location later on home for checking if the car is near or far!

Global Variables	
DBName	Static variable that represent our database name.
TBUsers	Users table that store all users' information like name and ID.
TBUNAutho	Unauthorized table that store all unauthorized images and date of access.
ID, Name, Image, Date	Columns name for all tables.

Methods	
AddUser	Add user information into the database.
AddUnauthorizedUser	Add unauthorized user ID, Image and AccessDate to the database.
getBitmapAsByteArray	Convert unauthorized user image to byte array for storing it in the database.
GetAllUsers	Getting all users information.
GetUnAuthorizedUsers	Getting all unauthorized users for the user to check them.
DeleteUnUser	Delete unauthorized user from the database after checking by the user.
onCreate	This method called when the DataBase class called and it used for creating the database if it is not existed.
onUpgrade	Upgrade the database.

Global Variables	
Socket	Specify the IP address and port number to connect the application with the server.
Message	The request message which contains the instruction for sending it to the sever.
ResponseMessage	The message from the sever which contains all information for updating the GUI.
MusicFiles	String variable that contains all music files names which will be split later to make them independent.

Methods	
doInBackground	This method is used for sending and receiving any message to/from the server using socket communication which specified before.

Global Variables	
Users	Static list which contains all authorized users in our system.

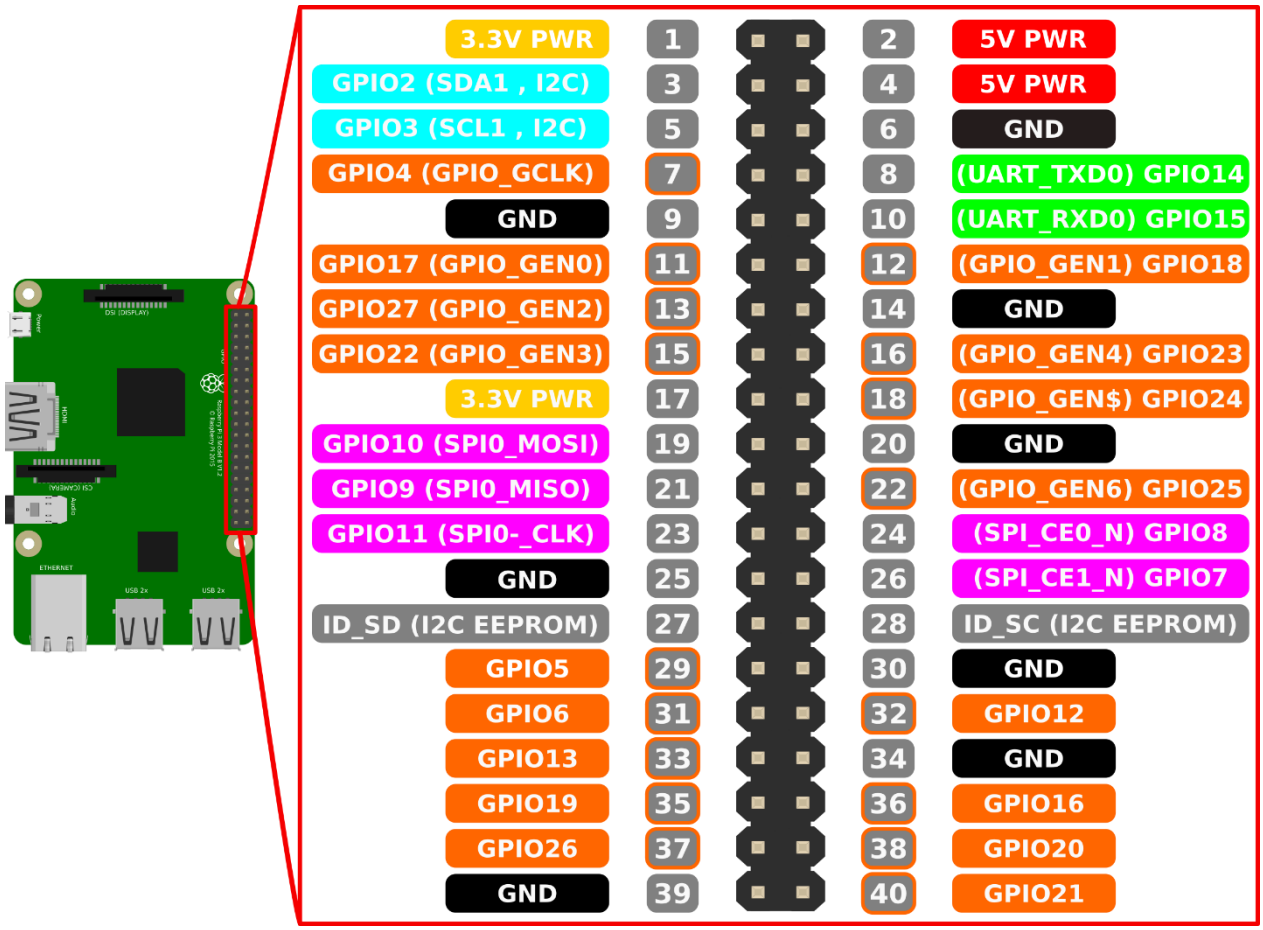
Methods	
onStartCommand	This method running in background as its class and it receives all authorized users' names, unauthorized user image and notify the user about any new image.
ShowNotification	This method is used for showing notifications.
onLocationChanged	This method called only when the user enable location service in his/her mobile and outside mode is active which is used for updating the location of the mobile phone when the user outside door and in his/her car, when the car is near to home, this feature may be integrate with GPS location on car in the future.

Car driver

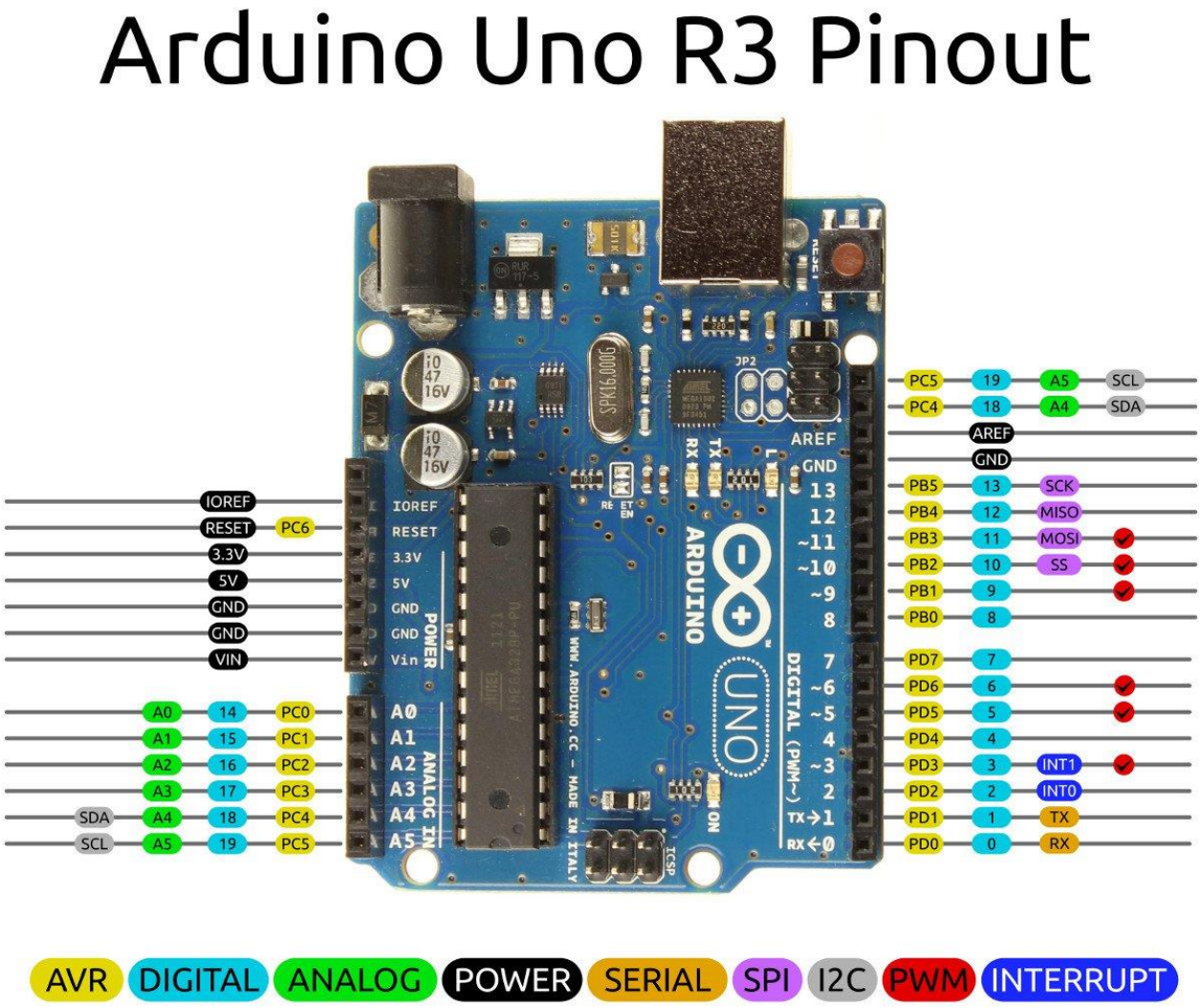
Global Variables	
eyeThreshold	The threshold which can be used to determine if eye is closed or not.
closedFrames	How many frames of images that make this eye is really closed not just blink for one time.
COUNTERl	Counter for left eye to count number of frames when eye blinked and make a decision later to determine if it is closed or not!
COUNTERr	Same as COUNTERl but here on right eye.
TOTAL	Total number of closed eyes.
scaleSize	This variable used for scaling the original image to small size which make its process faster.
detector	This is our detector for face and will be discussed in deep in Algorithms section.
predictor	Loading landmark predictor.
classifier	This classifier used for face detection.

Methods	
eyeAspectRatio	This method calculates the ratio between eyes landmarks to calculate the ratio between them.
alarm	This method used for alarm the user when his/her eye closed.
main	This method used for detecting the face, eye, check if eye closed or not and alarm the user when eye is closed.

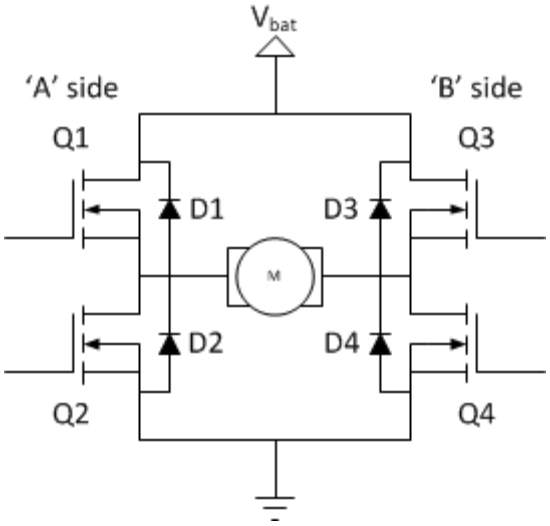
- In this section, I will describe the circuit design and what is input and what is output



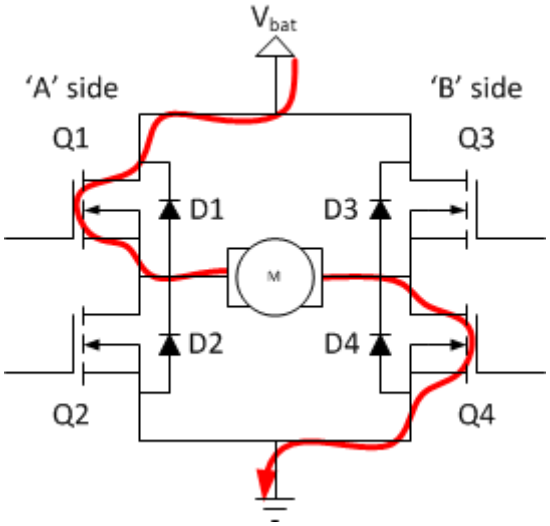
- This image shows all General-Purpose Input Output (GPIOs) for our raspberry pi circuit which we will use it for connecting our system components.
- There are two modes when you are using GPIOs (BOARD and BCM)
  - BOARD mode using normal numerical numbers from left to right (all left pins are odd and all right pins are even) and it is easy to use, when you want to add any pin to the raspberry just counting from left to your pin if you want to add jumper to pin 40 it will be right – bottom pin on the board and pin 25 in the left side and called GND (ground).
  - Second mode BCM which specified on any rectangle from this image for example pin 40 in the right-bottom it called GPIO 21 hence it is 21 in BCM mode and pin 3 in BOARD will be pin 2 in BCM and so on, and we will use BOARD mode in our system.
- All input/output pins use digital signal and there is no input/output for analog signal here.



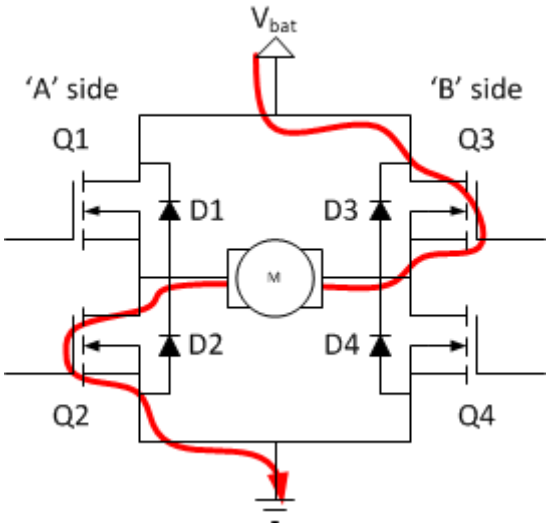
- Arduino GPIOs which shown from this picture divide into 3 main parts
  - From right, there are 14 digital pins before the GND which used for input/output digital signals.
  - On the bottom – left, there are 6 pins which used for input/output analog signals.
  - The remaining pins on the left used for power like Ground, 5V and 3.3V.
- First of all, I will describe how DC-Motors turning on from left to right or from right to left using H-Bridge and how we can control its speed using BWM.



- In general H-Bridge contains four switches (transistor) which make our circuit on/off from two sides with different ways of current.
- There are also two wires in the circuit one for (VCC) which connect to battery and the other one connected with the ground (GND).
- The DC-Motor connected with our switches in the middle making letter (H) like in the image above.
- If all switches turned off then there is no current can travel from VCC to the ground which make the motor turn.
- If we turn Q1 (switch 1), Q4 (switch 4) on and Q3, Q2 off then the current will pass from VCC to Q1, Q4 passing with the motor and make it turns let's say from left to right.



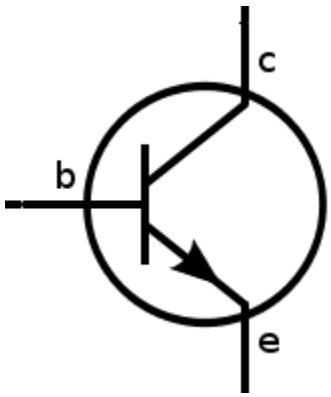
- If we turn Q3 and Q2 on and turning Q1 and Q4 off then the current will pass from the opposite way (from right to left).



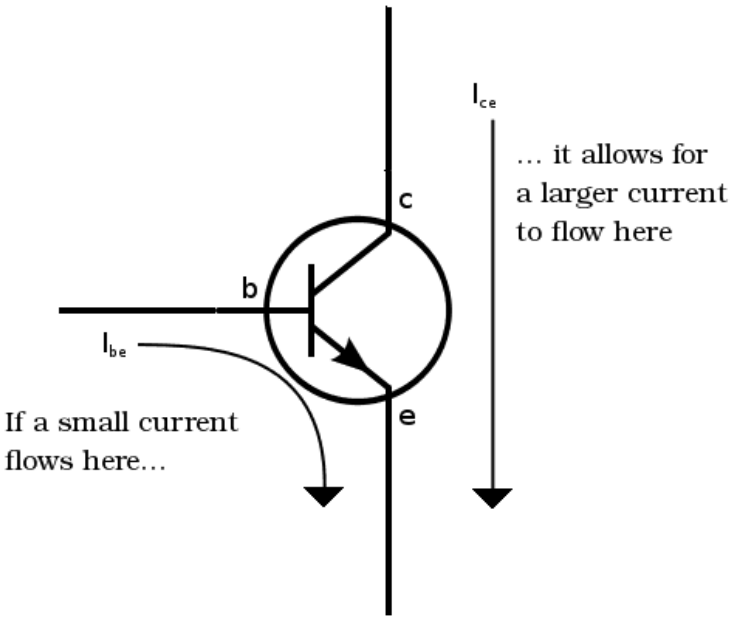
- These are the three ways which we can use in our circuit and DON'T try to make Q1 and Q2 on or Q3 and Q4 on or turning them all on because this may cause a short circuit which will damage your H-Bridge and may damage any connected device with it.
- So, these our states which we will use in our motors.

Q1	Q2	Q3	Q4	Turn
OFF	OFF	OFF	OFF	**Off**
ON	OFF	OFF	ON	**Left – Right**
OFF	ON	ON	OFF	**Right – Left**

- But how we can switch these switches on/ off automatically?
  - In this case we use the transistor instead of switch

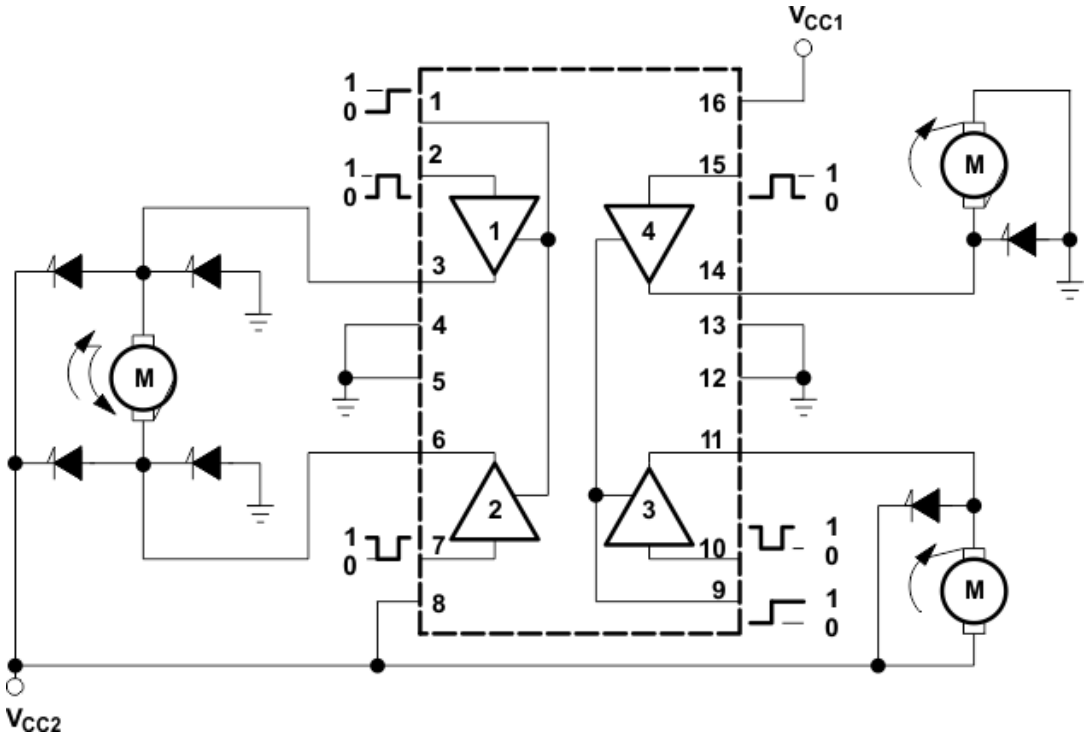


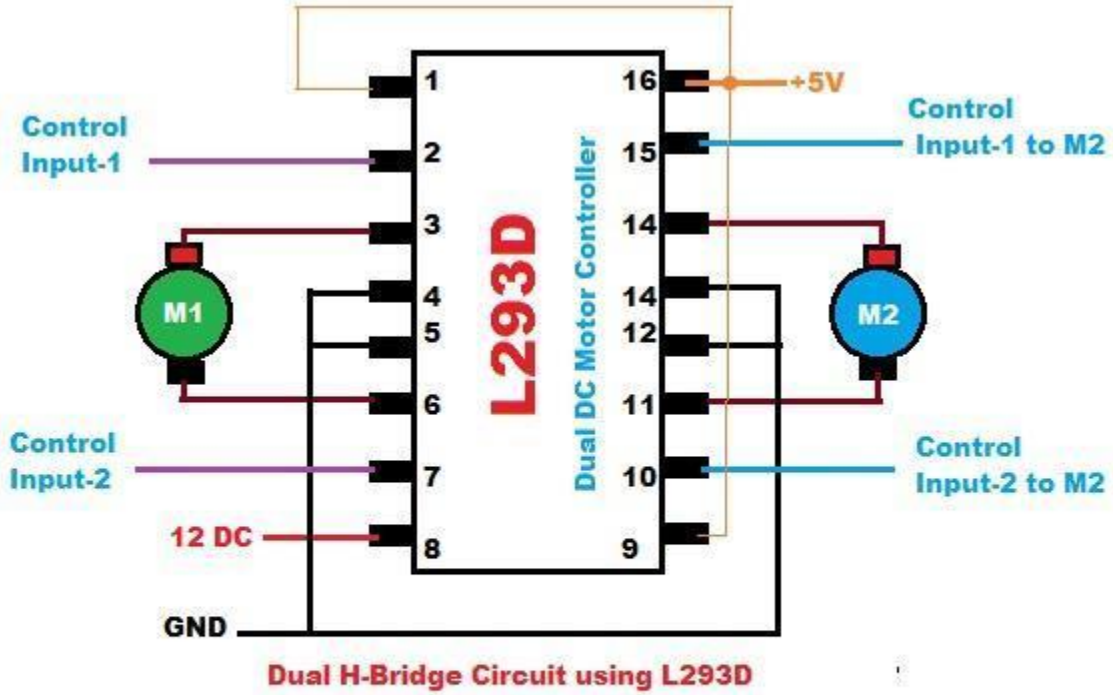
- There are three connectors here
  - Base (b)
  - Collector (c)
  - Emitter (e)
- The current must pass from c to e but this is a transistor which its mechanism says that there is no current pass from c to e if there is no current from b to e.
- So, if we need to make this switch on we must pass current from b to e make the current pass from c to e to.



- So, if we need to make this switch on at any time we must connect the base with high voltage to pass the current from base to emitter and this is all we need to know about H-Bridge and the transistor.

- In our project, we use L293D which works as dual H-Bridges (for two motors).



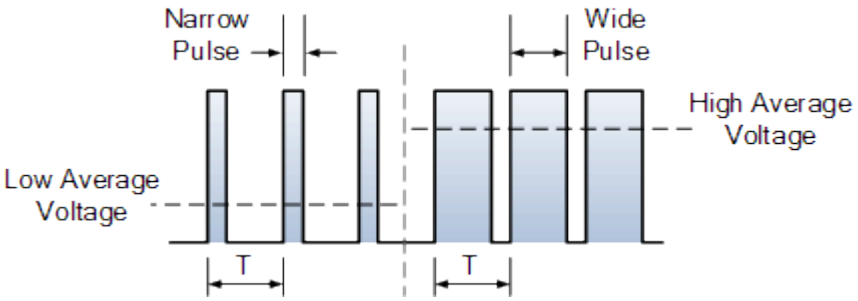


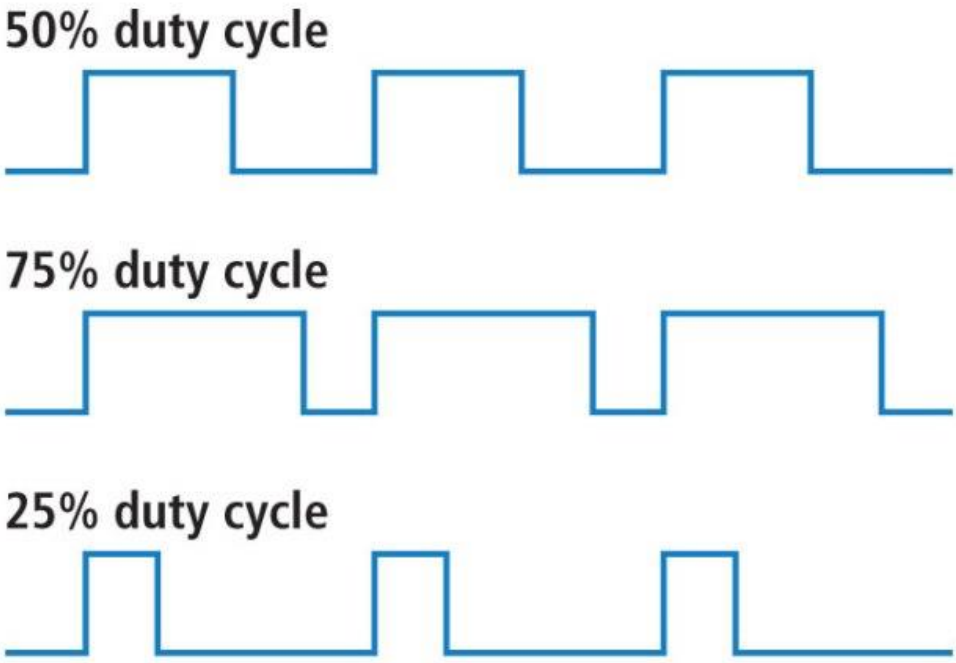
Dual H-Bridge Circuit using L293D

All inputs Low: Motor M1 & M2 = OFF.
Input-1 is High and Input-2 is Low: Motor M1 = Forward Direction.
Input-1 is Low and Input-2 is High: Motor M1 = Backward Direction.
Same Condition to M2.

Androiderode.com androiderode

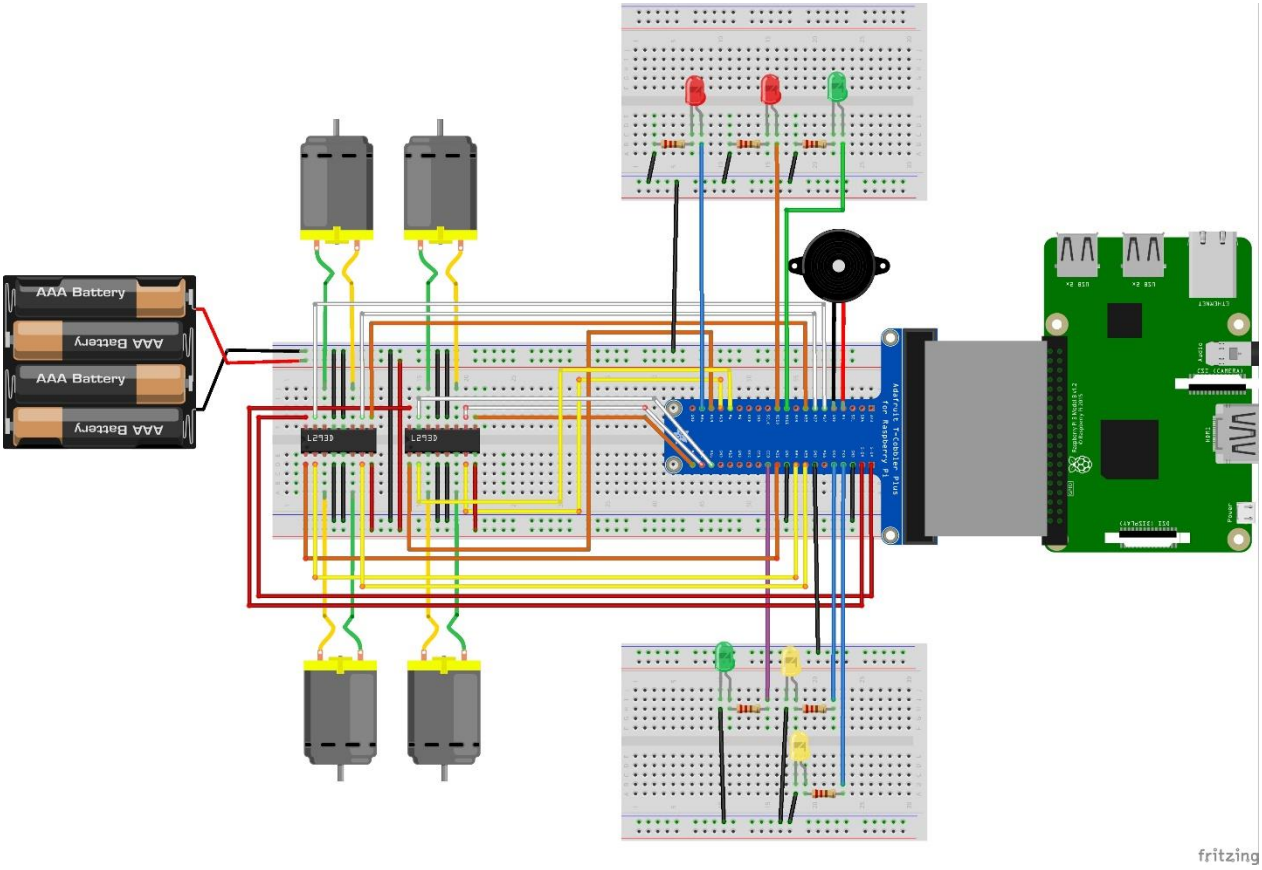
- Now Let's know how we can control the motor speed by using Pulse Width Modulation (PWM)?
- The main idea of PWM is to reduce the voltage with time (called the duty cycle) to make the motor run slowly.
- Some duty cycles with time.

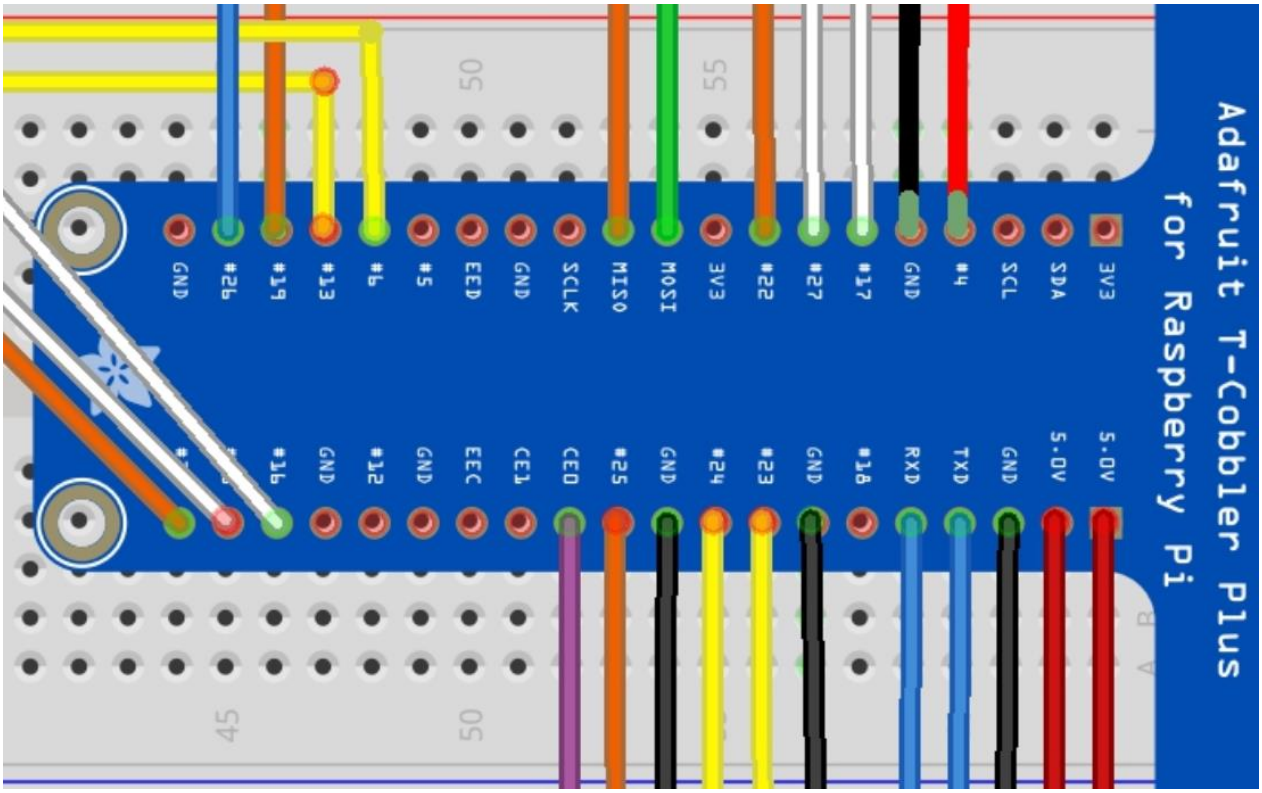




- Imagine that there are two axes, the X-axis represent the time and Y-axis represent the voltage and the spikes represent the voltage (High) with time and no spike represent (Low voltage), in case of 75% duty the voltage will pass more than in 25% with time and this is how we can do with the motors or LEDs to control the speed or the illumination.

- That's our circuit for home automation on raspberry pi which I will discuss its components now.





- Now we will discuss all these components briefly
  - Control A and Control B: control pins which make our motor run from left to right or reverse if the Enable pin on (remember transistor ?!).
  - one of these controls must be high and the other low to make the motor if the Enable on.

Motor	Usage	Control A	Control B	Enable
Top-left	Kitchen fan	11	13	15
Top-right	Living room air conditional	36	38	40
Bottom-left	Bed room window	16	18	22
Bottom-right	Garage door	31	33	35

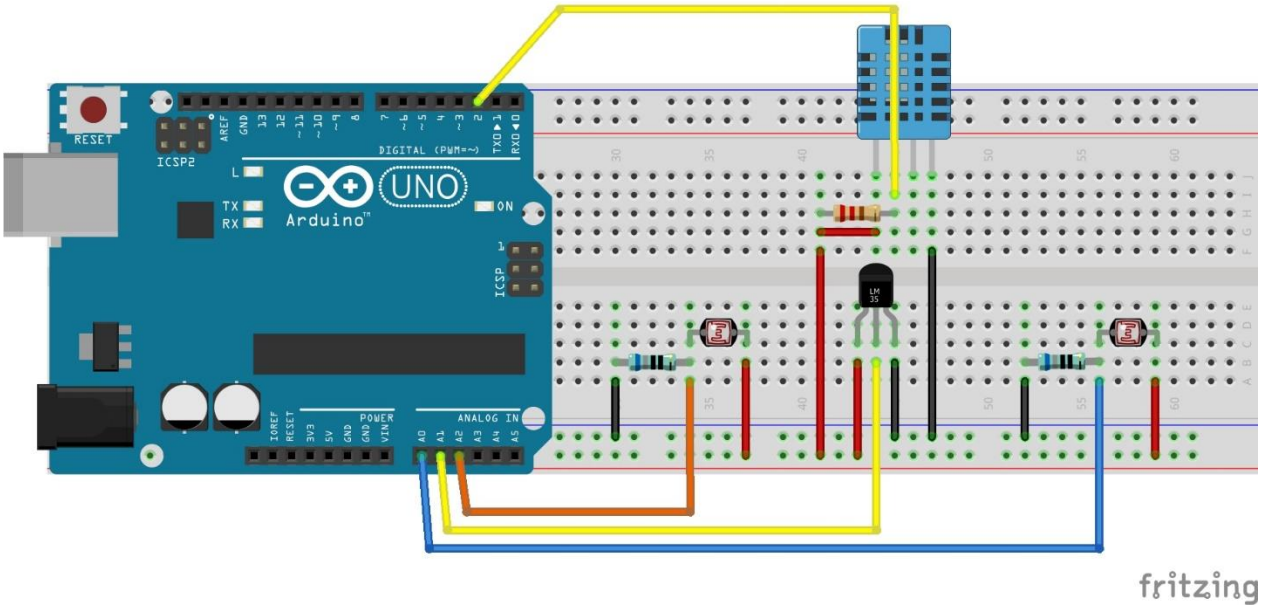
LED	Usage	Pin
Top-left (Red)	Simulate the café machine	37
Top-middle (Red)	Turning on when the system can't recognize the person.	19
Top-right (Green)	Turning on when the system recognizes the user well.	17
Bottom-left (Green)	Turning outside door on automatically.	24
Bottom-right (Yellow)	Living room LED	10
Bottom-Bottom-right (Yellow)	Bed room LED	8

Buzzer	Usage	Pin
Right-top	Alarm for user	7

- The other raspberry pi which connected to the car contains only one GPIO for a buzzer

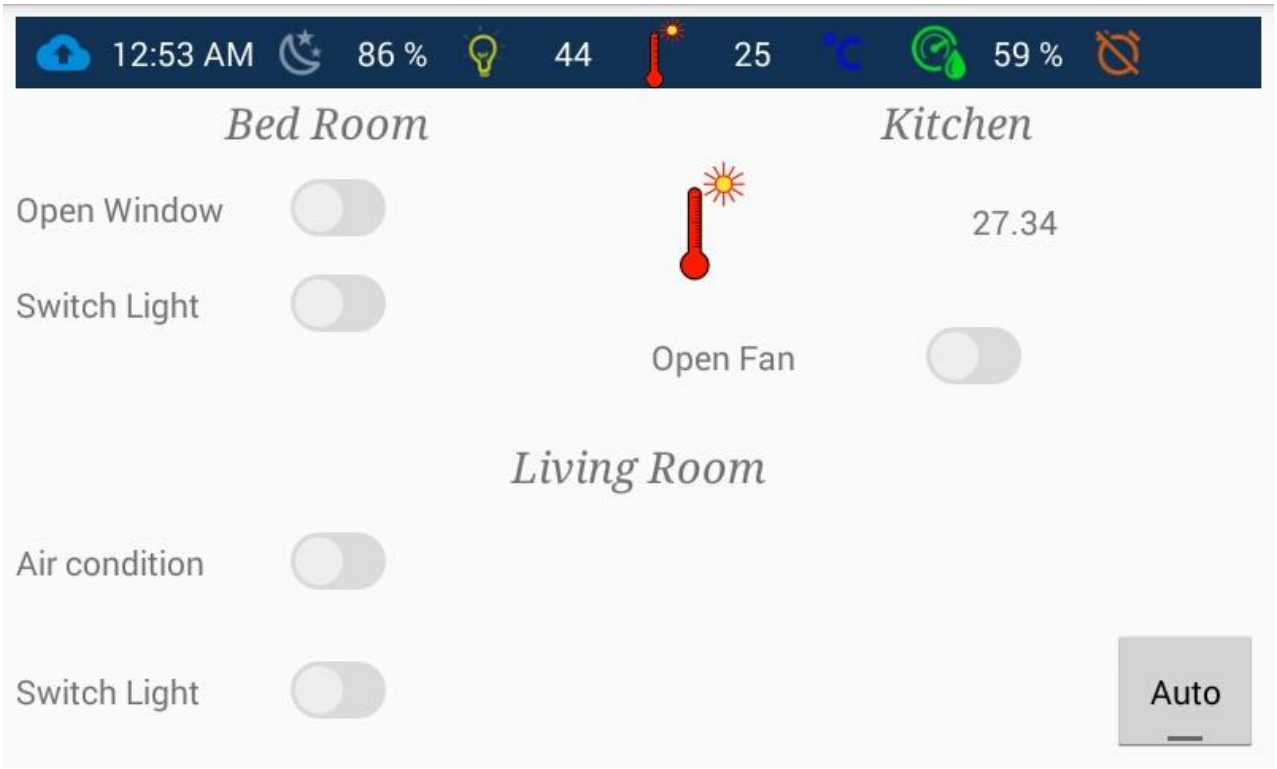
Buzzer	Usage	Pin
Alarm buzzer	Alarm when eye closed	30

- That's our Arduino circuit for the home automation which integrate with the raspberry pi for complete functions.

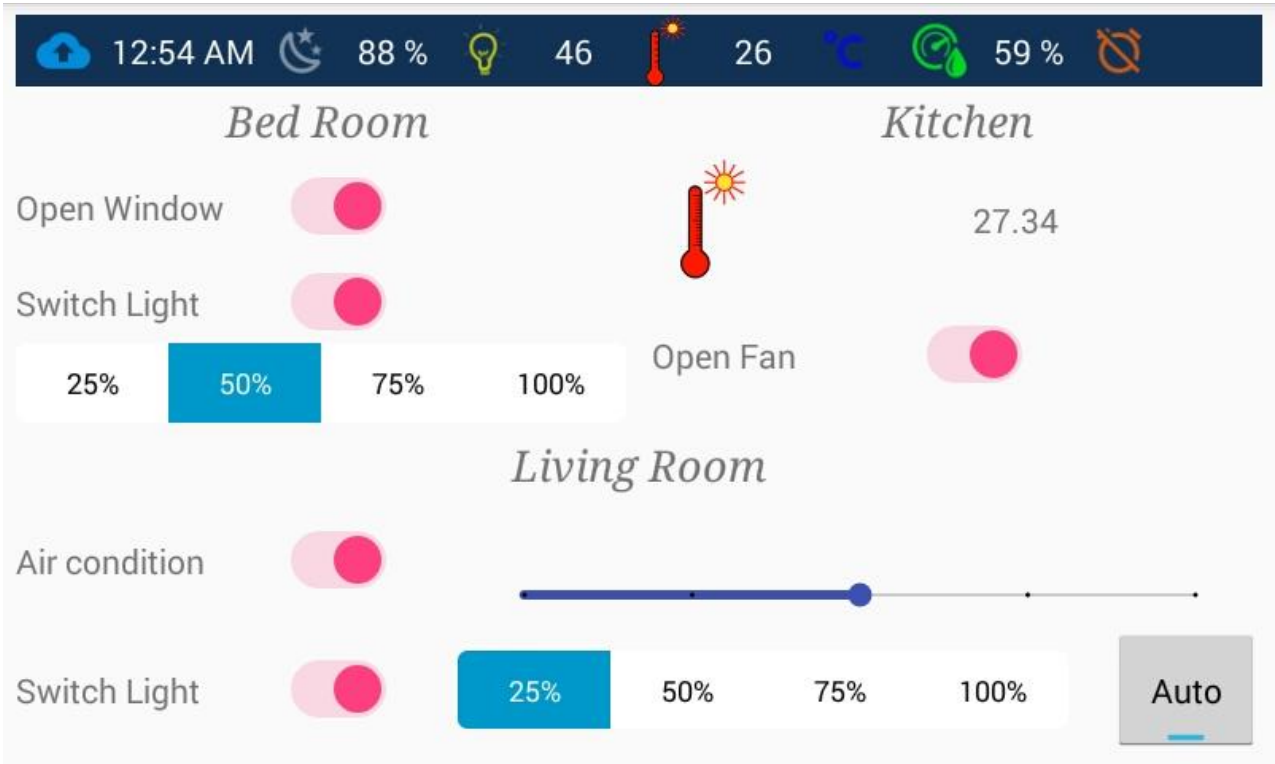


Sensor	Usage	Pin
DHT22	Measure bed room temperature and humidity	2
LDR (Left)	Measure nature light at day	A0
LM35	Measure temperature of kitchen	A1
LDR (Right)	Measure living room light or nature light.	A2

- In this chapter, we will see how the user interface activities looks like and its functions briefly.
  - This is the main activity which contains some information and controllers.



  - Let we control some lights, fans to show this clearly.



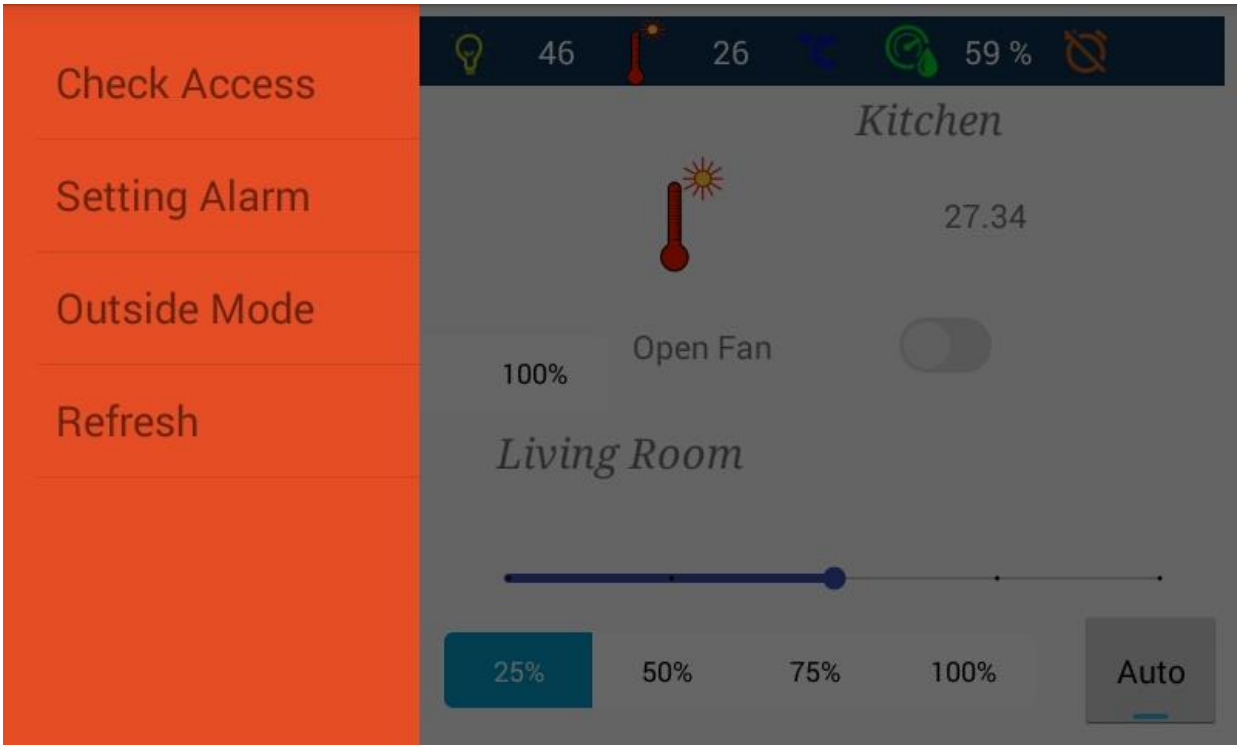
- The main activity contains: -
  - Action Bar
    Which contains all information about our system and these information is:

Image	Usage
	These images specify the system state: - 1) First one shows when the system can only accept your instructions but any unauthorized image can't be sent to user because it is not running in the main system. 2) Second one can send images to user but can't receive any instruction from him/her because it is not running. 3) Third one specifies that the system running perfectly and you can send and receive any data. 4) Fourth one specifies that the system not running or you have no access with the same Wi-Fi connection.
	These images specify the current state of day if it is sunrise, sunshine, sunset or night and the nature light percentage shows on the right of this image.
	This image shows the light degree of the Living room only and the user can use this information to make the living room light automatically adjusted by the system if the light degree is low in night or when it is cloudy day.
	These images specify the temperature statue of bed room or any room which contains DHT22 sensor and the degree beside this image with Celsius measurement.
	This image specifies the humidity percentage in the bed room and the value beside it.
	These images specify the state of the alarm (on/off).
	This image shows when unauthorized users images sent to authorized user which mean that there is a person or more unauthorized tried to access the system, when the user see and confirm all unauthorized images this image will disappear.
Second feature in action bar is a clock which specify current time from the android not from system.	

  - Bed Room section
    This section contains two functions in our bed room
    1) Open window: opening/ closing bed room window.
    2) Switch light: Turning light on/ off and control the degree of light when it is on.
  - Kitchen section
    Contains only information about current temperature of the kitchen and switch button for switching fan kitchen on/off.

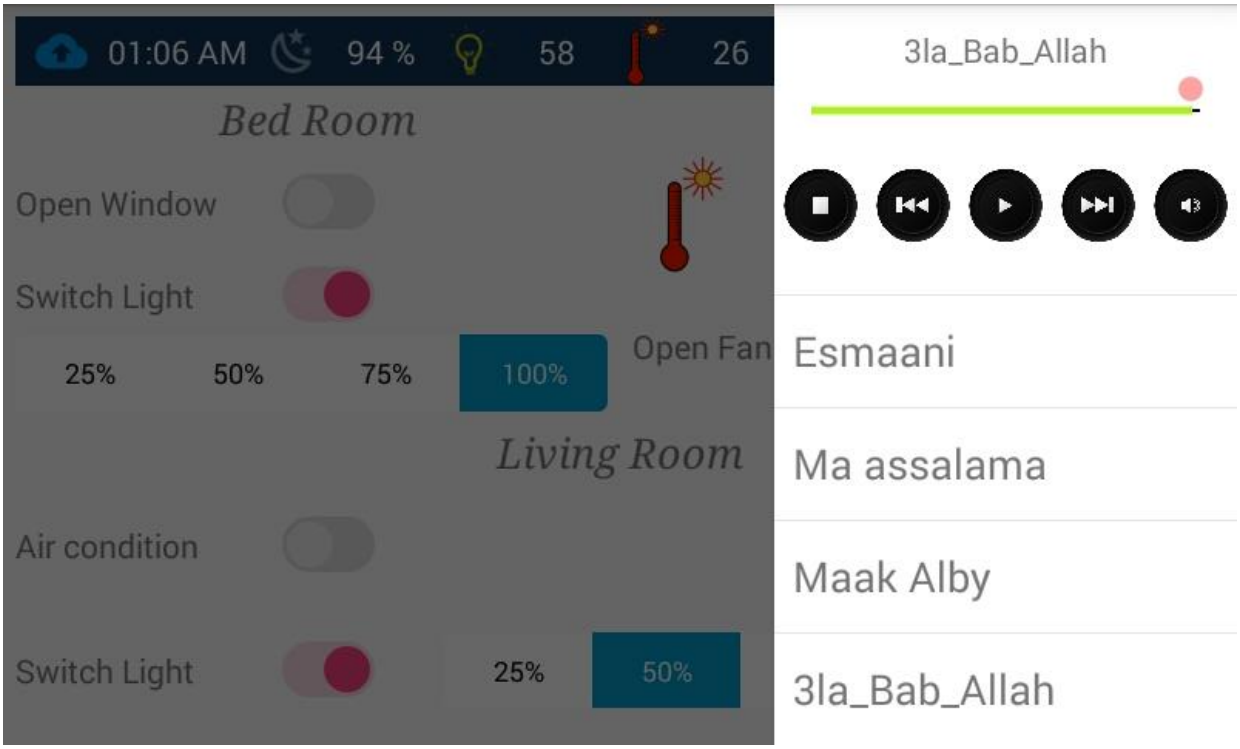
- Living room section
  Contains three functions:
  1) Air condition: we have no air conditional here but I simulate it and we use DC motor to do that function and we can assume it as a fan and this fan turning on/off by the switch button and control the speed (5 speeds).
  2) Switch light: same as bed room light.
  3) Auto: auto adjust the light in the living room according to outdoor light which described in action bar and when this button active it will ignore the degree of switch button and adjust the light according outdoor light, when it is off then it returns to the percentage degree which specified by the user but if the switch light button also off it turns the living room light off.
- Finally, this information updated every one minute by the system.

- This is main activity too, but with left hidden layout that contains: -



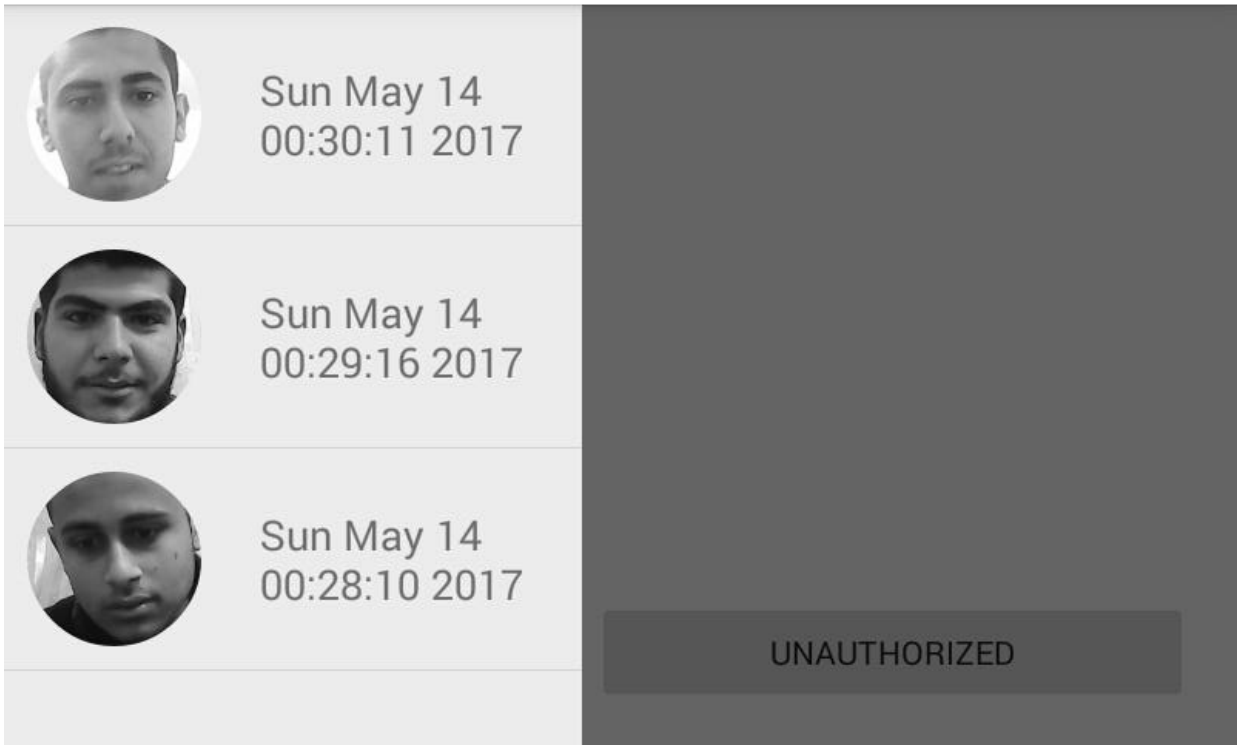
1) Check access: Move to another activity to check unauthorized users' images.
2) Setting Alarm: Move to another activity to setting the alarm.
3) Outside mode: This mode switching all lights, fans and music player off.
4) Refresh: Refresh the connection between the user and the main system when there is no connection between them after that it update all GUI information if connection succeed.

- Also, this is the main page with right hidden layout which contains: -

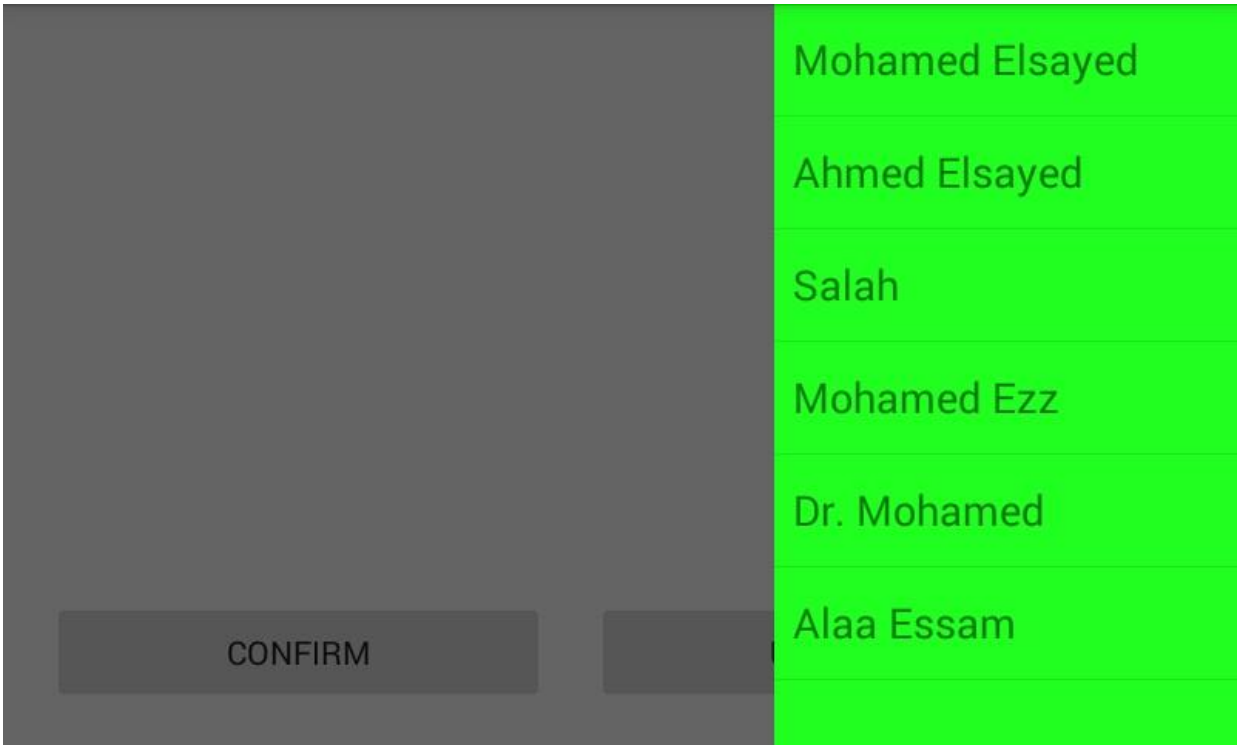


- Basic music player to play music on the system and control which music to play, stop, pause, etc. this music player contains: -
  - The current music name on the top “3la_Bab_Allah”.
  - Current volume of the music player.
  - The control panel of music player.
  - List of all available music files which in the music folder of the system

- Second activity that allows user to check all unauthorized users' images and make decisions.



- All unauthorized users' images with their access date as shown in left hidden layout.
- These images chosen by the user one by one and decide if this person authorized or not.
- In last image, we have person unknown person and last two persons are known.

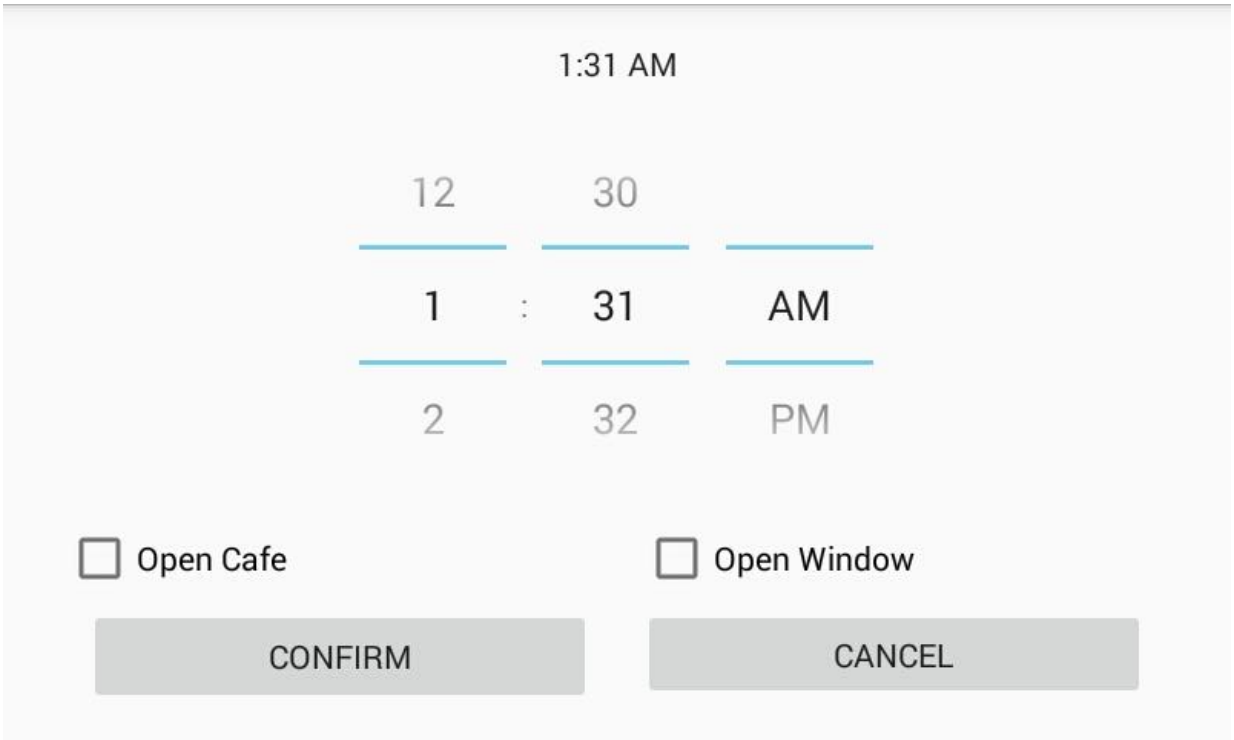


- Right layout contains all users' names which are known for us and also from the system.

- This person known for us and his name is “Salah” but not recognized well by the system so we need to add Salah’s image to its folder to make the system train this image later.
- Now we chose Salah’s image, its name and we press confirm button to confirm this operation.

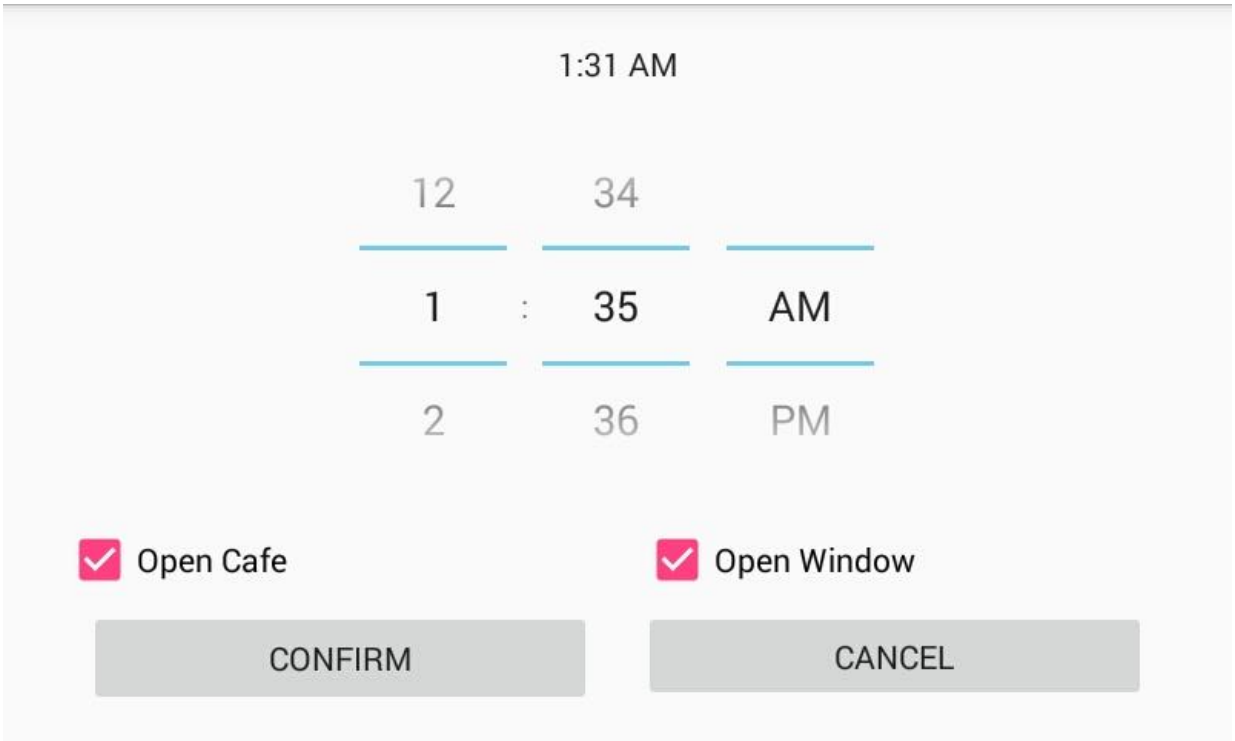
- Unknown person for us and we want to make him as unauthorized user and save his image in the system, so we chose his image (we don’t need to choose any name in this case) and press unauthorized button to move his image to unauthorized images directory in the system.

- Another activity that allows user to setting his/her alarm.

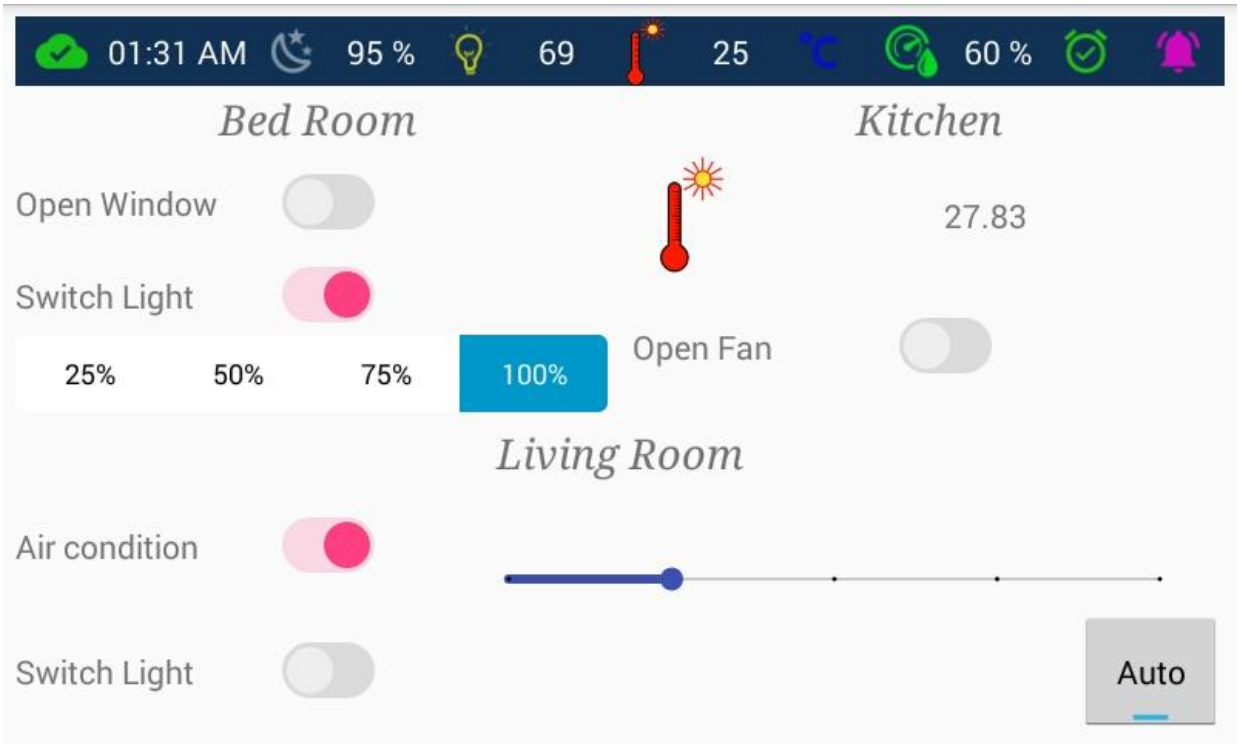


- This is the main activity of alarm which contains: -
  - Current time on the top.
  - Digital clock to choose the time of alarm, this alarm between 1 min – 24 hours, it set for only once and any alarm will override the last one.
  - Open café: Optional for user if he/she setting alarm and wants the café machine starts before running alarm then he/she can check this checkbox.
  - Open window: Also, it is optional and it opens the bed room window after alarm starts.
  - Confirm: This button confirms the alarm setting and send the required time for alarm to the system.
  - Cancel: cancel operation and get back to main activity.

- Here we are setting the alarm and confirm it.



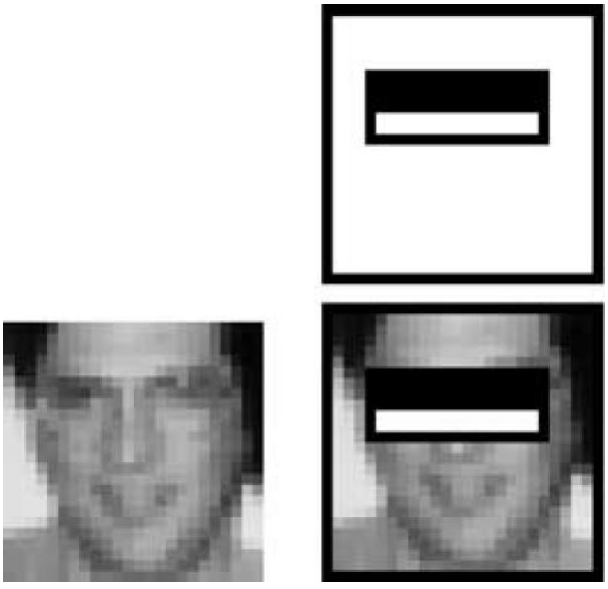
- Another picture for main activity which we can notes that: -
  - System state is fully connected and work perfectly.
  - The alarm state changed from off to on.
  - The notification state image was shown after we receive unauthorized images.



- In this chapter, I will discuss all algorithms and methodologies that were used in our system and I will focus on the algorithms of face detection, face recognition and eye blink.

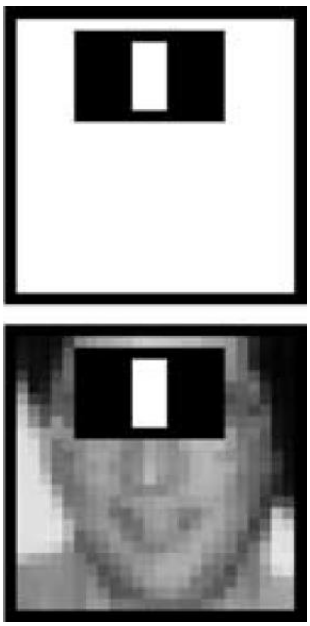
  - In this section, I will describe the Viola and Jones face detection algorithm which enables us to detect face from image in real time so we can use this face image to recognize it later in face recognition section.
    - Now we have three main steps to achieve this goal: -
      1- Feature computation (Integral image)
      2- Feature selection (AdaBoost)
      3- Realtime detection (Cascade classifier)

- Feature computation
  - All faces share some similar properties like eyes, nose and mouth so if we can detect like these properties we say this is a face.
    - The eyes region is darker than upper cheeks so we assume that this region as a feature and can be detected by some operations will be discussed now but we need to make this region special when we need to operate it so we use a rectangle which specify this feature like this.

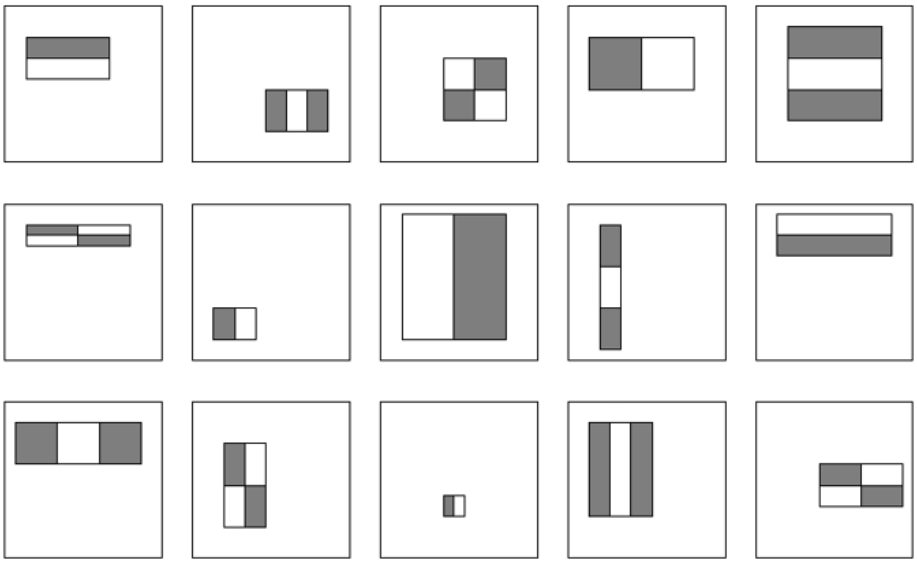


    - This rectangle is called two rectangle features as it contains two sub rectangles (white and black) which will help to compute our feature.

- Another example is the nose bridge region which is brighter than the eyes or the shadow around nose and this called a three rectangle features



- This image contains some features with all types in the Viola and Jones.



- Feature can be calculated by subtract the summation of pixels in white rectangle(s) from the summation of pixels in grey or black rectangle(s).

$$\text{Value} = \sum(pixels\ in\ black\ rectangle) - \sum(pixels\ in\ white\ rectangle)$$

- As we mentioned before that we share the same properties but we don't share the same eye size or position and rotation of the head, you can rotate your head a little which will differ your eye location or position from another one, your nose may be smaller than your friend's noise, so we have a problem here and we need to solve it.

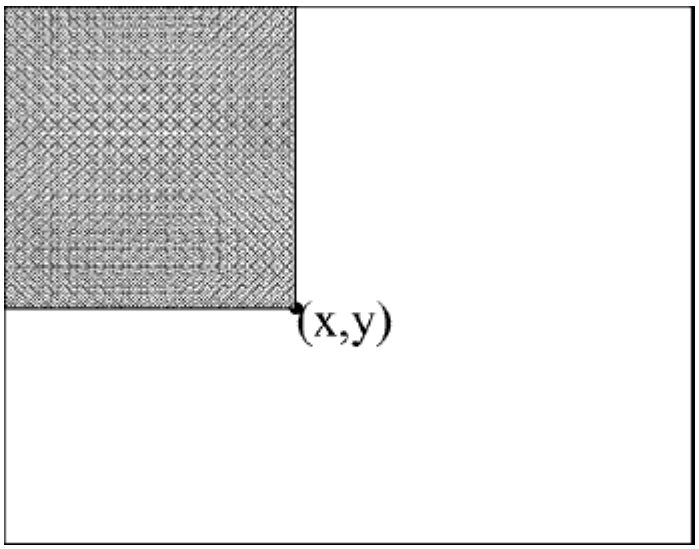
- Voila and Jones features can solve this problem by changing the size, location, orientation, etc. of these features to be suitable for all faces and this changes lead to make our features to be ~16,000 feature with different location and size.
  - In any feature rectangle, it starts from size 24 x 24 pixels and it scanned for 12 times with different scale, each scale is a factor of 1.25 larger than the last.
  - For more speed in this case we use the integral image representation for each sub window or scale.
  - The integral image at location (x, y) contains the sum of pixels above and to the left of x, y inclusive:

$$ii(x,y) = \sum_{x' \le x,\ y' \le y} i(x',y')$$

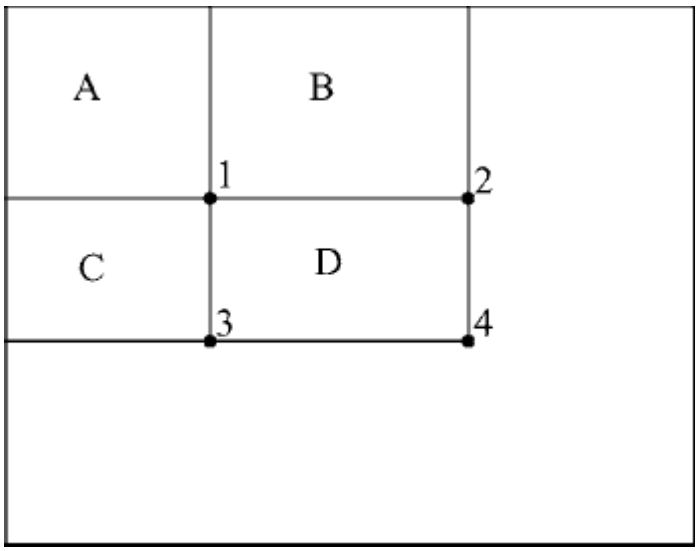
Where $ii(x,y)$ is the integral image and $i(x,y)$ is the original image using the following pair of recurrences?

$$s(x,y) = s(x,y-1) + i(x,y) \quad (1)$$
$$ii(x,y) = ii(x-1,y) + s(x,y) \quad (2)$$

Where $s(x,y)$ is the cumulative row sum, $s(x,-1) = 0$, $ii(-1,y) = 0$



- The integral image can be computed in one pass over the original image.

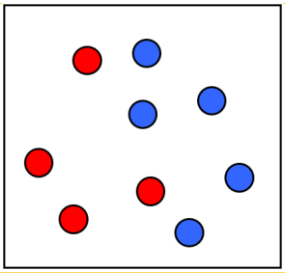


- The sum of pixels within rectangle D can be computed with four array references.
    1) The value of the integral image at location 1 is the sum of pixels of the sum of the pixels in rectangle A.
    2) The value at location 2 is A + B.
    3) Location 3 is A+C.
    4) Location 4 is A+B+C+D.
- The sum within D can be computed as $4 + 1 - (2 + 3)$.
- Now we have our features after computing integral image for each sub window, but we don't need all 16,000 features it still a problem as it may take many time in processing especially when we need this detection in our system which is real time system.

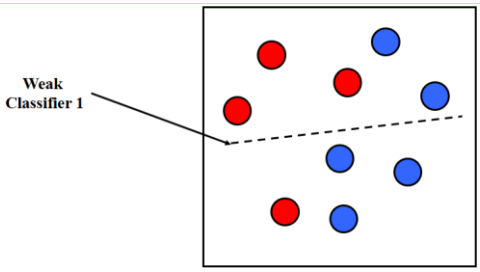
Feature selection

- As I mentioned before that we don't need all 16,000 features, because some features not necessary in our detection.
  - For example, this feature represents the noise bridge as we can see that this feature (left figure) is useful as it detects the bridge of noise well but the same feature with small size on the right not useful for us as it can't detect noise bride in this region so we need to remove this size of feature and keep the larger one, this processing done by Adaptive Boosting algorithm which detects the most important features.

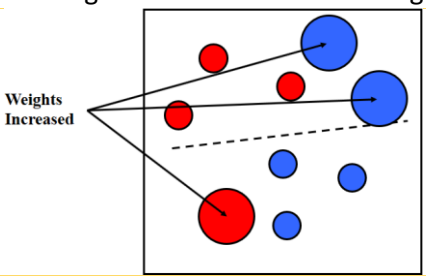
- AdaBoost constructs a strong classifier as a linear combination of weighted simple weak classifiers.
- Training consists of multiple boosting rounds
  1) During each boosting round, we select a weak learner that does well on examples that were hard for the previous weak learners.
  2) "Hardness is captured by weights attached to training examples"
- Training Algorithm
  3) Given an example of images $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
  4) Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
  5) For $t = 1, \dots, T$:
     1. Normalize the weights, $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
     2. Select the best weak classifier with respect to the weighted error
        $$\epsilon_t = min_{f,p,\theta} \sum_i w_i \left| h(x_i, f, p, \theta) - y_i \right|$$
     3. Define $h_t(x) = h(x,\ f_t,\ p_t,\ \theta_t)\ where\ f_t,\ p_t\ and\ \theta_t\ are\ the\ minimizers\ of\ \epsilon_t$
     4. Update the weights:
        $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
        Where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$
  6) The final strong classifier is:
     $$C(x) = \begin{cases} 1, & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0, & otherwise \end{cases}$$
     Where $\alpha_t = \log \frac{1}{\beta_t}$



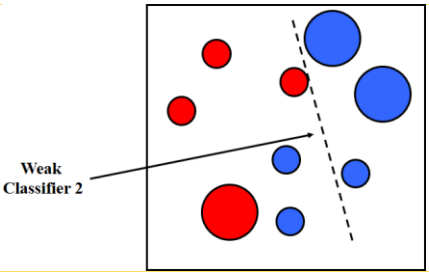
Step 1: Training examples (x, y) where x is red balls which represent negative (0) and blue represent positive (1).
Step 2: All weights are initialized according.



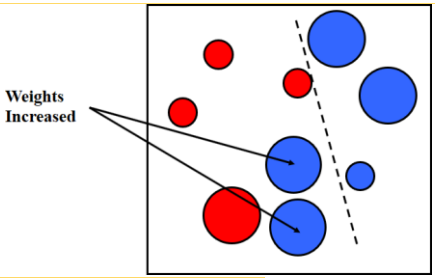
Step 3.1: Normalize weights.
Step 3.2: Select the weak learner that achieves the lowest weighted training error.

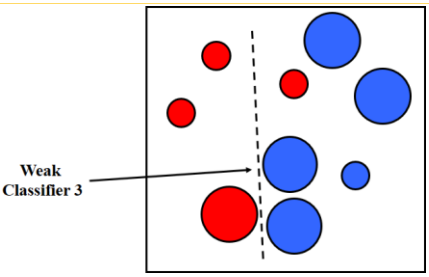


Step 3.3: Define $h_t(x)$.
Step 3.4: Update weights.

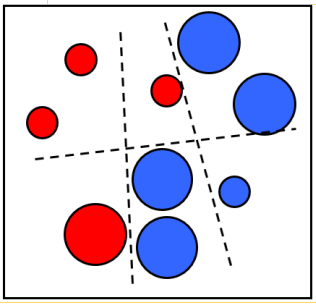


Repeat step 3 for each feature.





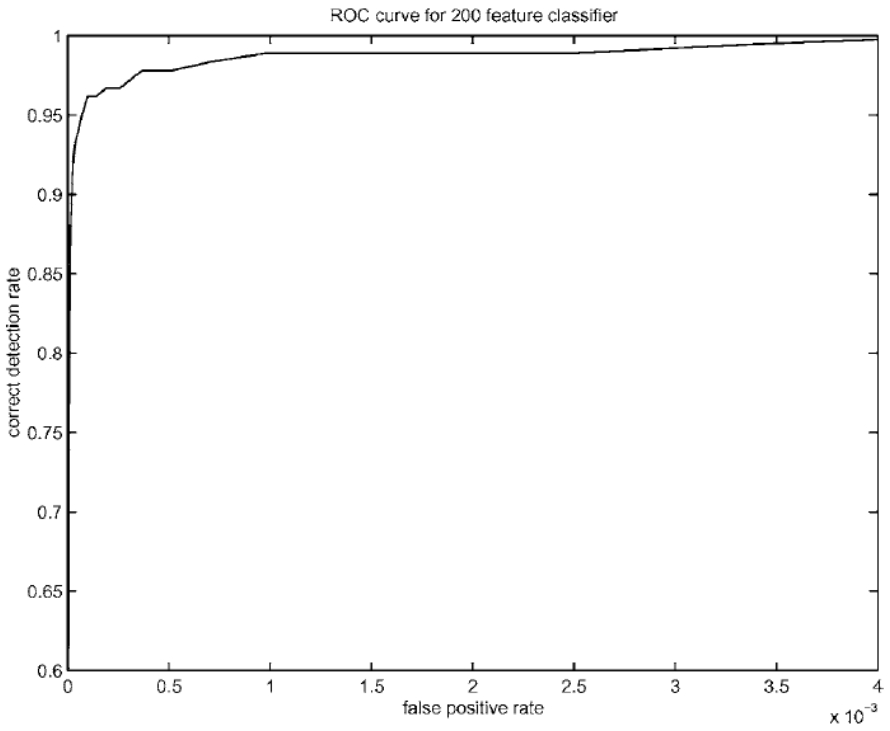
Final classifier is a combination of weak classifiers



Step 4: Final stronger classifier as lineear combination of all weak learners.

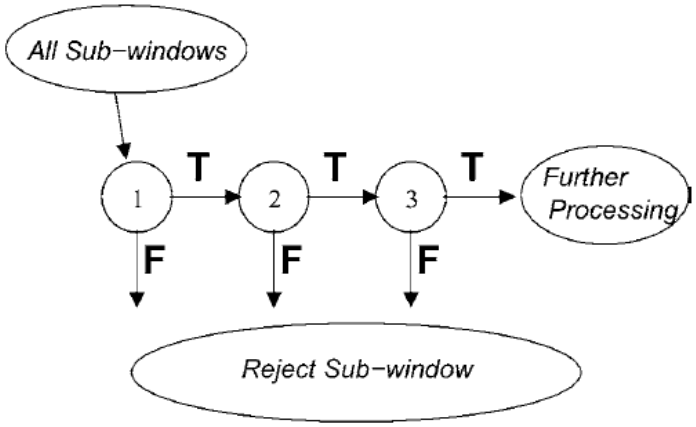
- Computational complexity of learning O(MNK)
  Where M rounds, N examples, K features
- Now we build a 200-feature classifier from 16,000

- Experiments showed that a 200-feature classifier achieves 95 % detection rate.



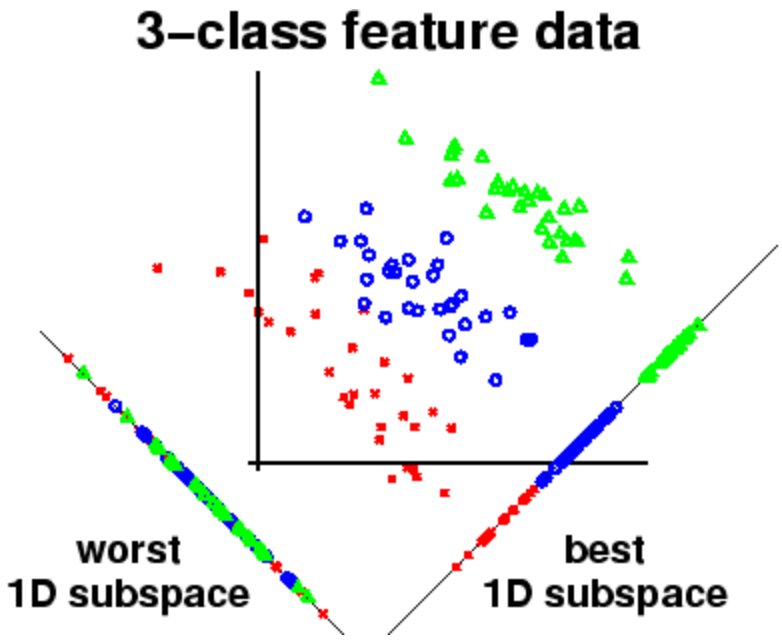
Realtime detection

- After choosing best features, we need to apply them on real time system but if sub window A not contains a face the system will apply all features on it to check if it contains a face or not and repeat this step for each sub window in the image which will increase the processing for each sub window, but this problem can solve using the decision tree which called cascade.
- The idea is to filter each window using the most common features for example (eyes and noise bridge) features can yield 100 % detection rate (faces windows) and 50% false positive rate (not face but system tell us it is a face), so the remaining 50% of windows are not useful for us and we can ignore them only in one step instead applying all features on the window.
- So, this is the main idea we divide features to classifiers each classifier contains the features which reject many negative windows and if it passes from first classifier we move the image to the second classifier which contains more complex features and check if it can pass or not and so on to pass all classifiers to get our face!



- Now we have our face image from face detection and we need to recognize this face.
- Let we have three persons (Tom, Jerry and Adam) Tom and Jerry home owners but Adam not!
- If you saw Tom and Jerry before you can recognize them immediately but Adam you didn't seen him before so you can't recognize him, this process is called face recognition and this is our problem for now.
- We need to make our system recognize persons based on previous images by using the camera and tell us this person known for him or not.
- The algorithm we are going to use is called Fisher algorithm which is based on Linear Discriminant Analysis (LDA) for face recognition.

    - LDA seeks directions that are efficient for discrimination between the data.
    - LDA maximizes the between class scatter and minimizes the within class scatter.



- In this figure, we represent the data on subspace (dimensionality reduction) which contains all classes perfectly classified in the right subspace but the worst case was represented on the left, that what LDA do on that data is representing them on lower dimensional with less loss in data form each class.

- LDA Algorithm
  - Let X be a random vector with samples drawn from c classes:

$$X = \{X_1, X_2, \dots, X_c\}$$
$$X_i = \{x_1, x_2, \dots, x_n\}$$

  - The scatter matrices S_B and S_w are calculated as:

$$S_B = \sum_{i=1}^{c} N_i (\mu_i - \mu)(\mu_i - \mu)^T$$
$$S_W = \sum_{i=1}^{c} \sum_{x_j \epsilon X_i} (x_j - \mu_i)(x_j - \mu_j)^T$$

    , Where μ is the total mean:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

    And μ_i is the mean of class $i \in \{1, \dots, c\}$

$$\mu_i = \frac{1}{|X_i|} \sum_{x_j \in X_i} x_j$$

  - Fisher's class algorithm now looks for a projection W, that maximizes the class reparability criterion:

$$W_{opt} = argmax_W \frac{|W^T S_B W|}{|W^T S_W W|} = [W_1\ W_2 \dots W_m]$$

    , Where $\{W_j | i = 1, 2, \dots, m\}$ is the set of generalized eigen vectors of S_B and S_w corresponding to the m largest generalized eigenvalues $\{\lambda_j | i = 1, 2, \dots, m\}$

$$S_B W_j = \lambda_j S_W W_j, \qquad i = 1, 2, \dots, m$$

  - There's one problem left to solve: The rank S_W is at most (N – c), with N samples and c classes.
    - In pattern recognition problems, the number of samples N is almost always smaller than the dimension of the input data (the number of pixels), so the scatter matrix S_W becomes singular.
    - So, we solve this problem by performing a Principal Component Analysis (PCA) on the data and projecting the samples into the (N-c) dimensional space.
    - LDA was performed on the reduced data, because S_W isn't singular anymore.
    - The optimization problem can be rewritten as:

$$W_{pca} = argmax_W |W^T S_T W|$$
$$W_{fld} = argmax_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|}$$

    - The transformation matrix W, that projects a sample into the (c – 1) dimensional space is then given by:

$$W = W_{fld}^T W_{pca}^T$$

- LDA assumes that all training and testing images with the same size.
- If we have 4 training data images 2 for each person so the image matrixes for each image will be:

Tom	
$x_1 = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{N^2} \end{pmatrix}$	$x_2 = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{N^2} \end{pmatrix}$

Jerry	
$x_3 = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N^2} \end{pmatrix}$	$x_4 = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N^2} \end{pmatrix}$

- So, we can compute the mean of all faces by: -

$$\mu = \frac{1}{M}\begin{pmatrix} a_1 + b_1 + c_1 + d_1 \\ a_2 + b_2 + c_2 + d_2 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ a_{N^2} + b_{N^2} + c_{N^2} + d_{N^2} \end{pmatrix}, \qquad where\ M = 4$$

- And the mean of each face can be calculated by: -

$$\mu_1 = \frac{1}{2}\begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \quad \vdots \\ a_{N^2} + b_{N^2} \end{pmatrix} \qquad \mu_2 = \frac{1}{2}\begin{pmatrix} c_1 + d_1 \\ c_2 + d_2 \\ \vdots \quad \vdots \\ c_{N^2} + d_{N^2} \end{pmatrix}$$

- Now we can compute S_B and S_W

$$S_B = 2\big((\mu_1 - \mu)(\mu_1 - \mu)^T\big) + 2\big((\mu_2 - \mu)(\mu_2 - \mu)^T\big)$$

$$S_W = \big((x_1 - \mu_1)(x_1 - \mu_1)^T + (x_2 - \mu_1)(x_2 - \mu_1)^T\big) + \big((x_3 - \mu_2)(x_3 - \mu_2)^T + (x_4 - \mu_2)(x_4 - \mu_2)^T\big)$$

- And now we need to optimize the between class variance

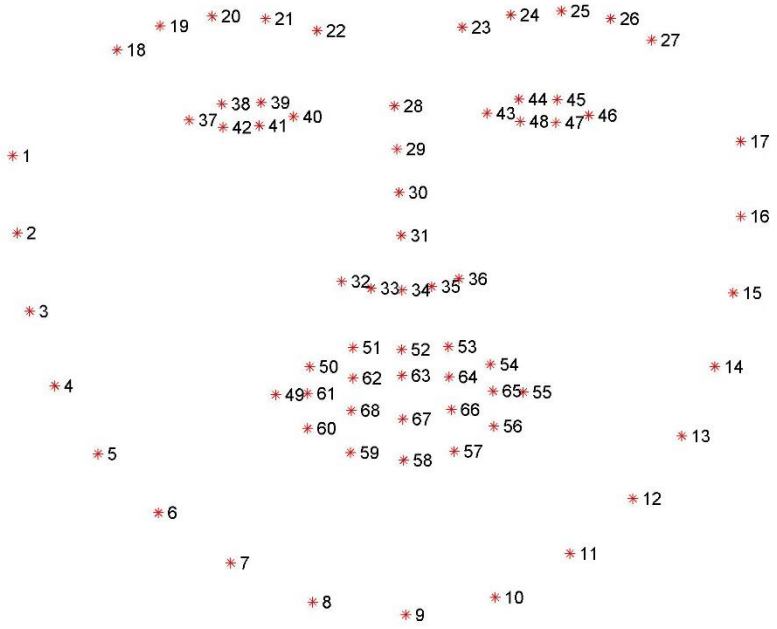
$$W_{opt} = argmax_W \frac{|W^T S_B W|}{|W^T S_W W|}$$

- Finally, we have one value of S_W
  - if S_W is non-singular $(M \geq N^2)$
    Then columns of W are eigenvectors of ${S_W}^{-1} S_B$ from this equation: -

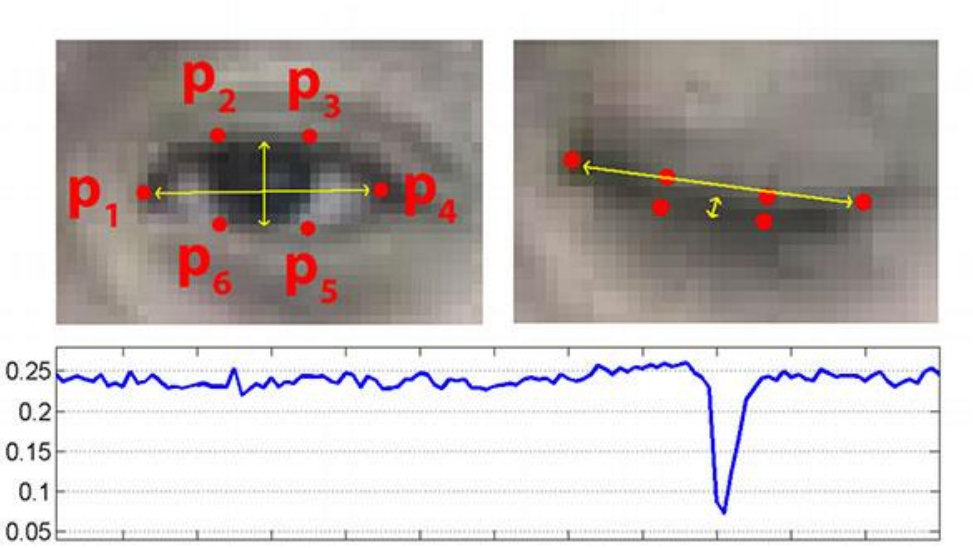
$$S_B W_j = \lambda_j S_W W_j$$

  - if S_W is singular $(M < N^2)$
    Then we apply PCA first which will reduce the dimension of faces from N^2 to M

- In this section, I'll describe briefly how we can detect if eye is open or close using facial landmarks.
  - Facial landmarks are set of salient points, usually located on the corners, tips or mid points of the facial components.
  - The facial landmark detector implemented 68 *(x, y)*-coordinates that map to *specific facial structures*.
  - These 68 point mappings were obtained by training a shape predictor.



  - As we can see from this figure the left and right eyes have 6 points for each other and these points locations can vary little according the eye of that person or if his/her eye is open or closed.



- To calculate the distance between these points we use the eye aspect ratio (EAR) using Euclidian distance between two points.

$$EAR_x = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

, Where $x = \{left,\ right\}$

- The EAR is mostly constant when an eye is open and is getting close to zero while closing the eye.
- After we calculate the EAR we can decide if eye E_x is Blinking for one time (1) or not (0) by using annotated threshold θ.

$$E_x = \begin{cases} +1, & EAR_x \geq \theta \\ 0, & EAR_x < \theta \end{cases}$$

- Then we can say that EYE_x is closed if E_x greater than α threshold which represent the number of closed frames.

$$EYE_x = \begin{cases} Close, & E_x \geq \alpha \\ Open, & E_x < \alpha \end{cases}$$

- Let we have three persons
  1) Ahmed: the owner of our home.
  2) Alaa: Ahmed's brother and he is owner too.
  3) Salah: Neighbor's son.

- At 11:00 pm Ahmed wants to sleep to wake up early at 07:00 AM and he need to adjust his alarm to wake him up so he picks up his phone and opens IControl application, opens "Setting alarm" activity and setting its alarm at 07:00 AM and check open café feature to starts his café machine then he confirms this and he get back to the main activity to close his room window and light off and he is going to sleep now!

- At 06:57 AM the system automatically starts the café machine to make coffee and at 07:00 AM system alarm starts to wake Ahmed up.

- Ahmed wakes up at 07:01 AM and he wants to go for his job after 15 minutes but while he is drinking his coffee he saw that the living room humidity is high therefor he turns the air conditional on with middle speed and it is 07:30 AM and he will late for his job, so he is going now but before going he opens the application and check outside mode which turns the air conditional and any lights off.

- Before Ahmed getting on his car he active location service on his device on which identify his location now and when he starts driving his car to his job and after 100 m from his home the garage door turns off automatically.

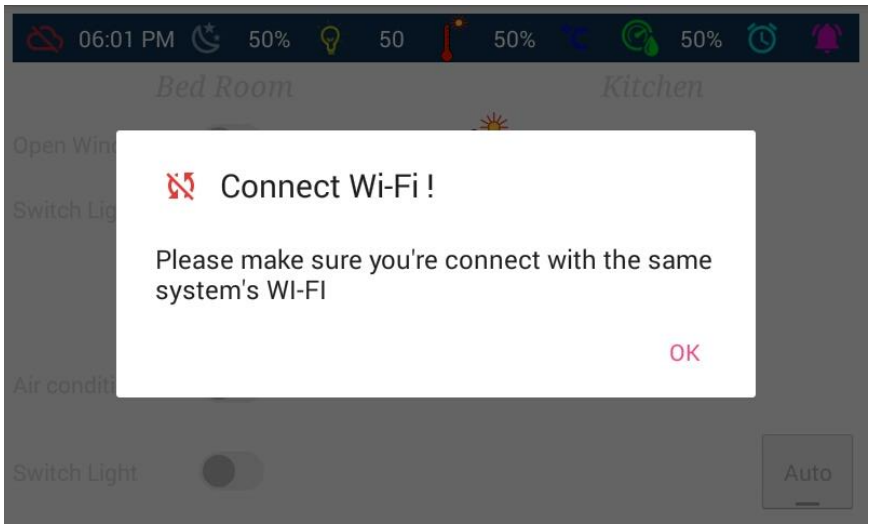
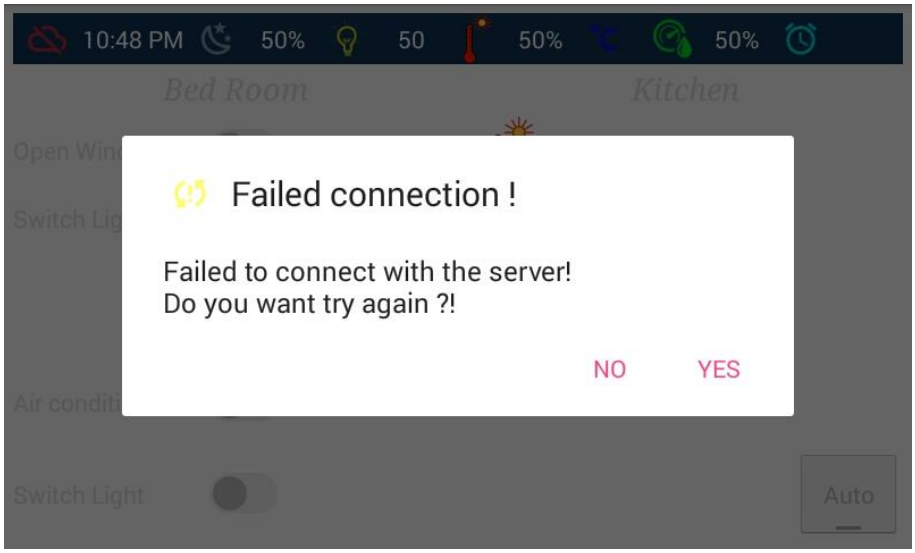
- At 03:00 PM it is hot day and Ahmed going back to his home and he is near about 100 m from his home then the garage door opens, after that Ahmed set down on living room and starts the air conditional on its high speed, after 5 minutes he is in the kitchen making his favorite food and the kitchen temperature increased to be $29\ ℃$ then kitchen fan running to reduce this temperature according its degree, but Ahmed force stop this fan using his application and after one minute the temperature degree becomes $33\ ℃$ and the fan running again with high speed to reduce this high temperature.

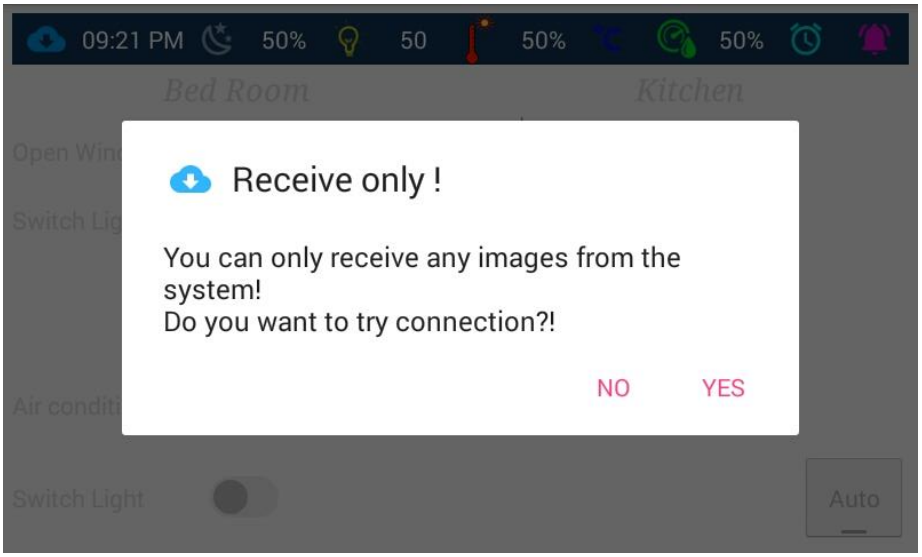
- At 05:00 PM: while Ahmed watching the TV, he get the notification telling him that there is unauthorized person tried to access his home and when check the image he notice that this person known for him (his brother Alaa) and he can access the home too but the system not recognize him well at first and Alaa in the home now after the system tried to recognize him in the second one, now Ahmed checking the unauthorized users activity and he get 21 images for unauthorized users one for his brother Alaa and 20 for Salah (Neighbor's son) who plays every day on Ahmed's outdoor. Ahmed confirm his brother's image and the system add this image into his dataset and add Salah's images to unauthorized users.

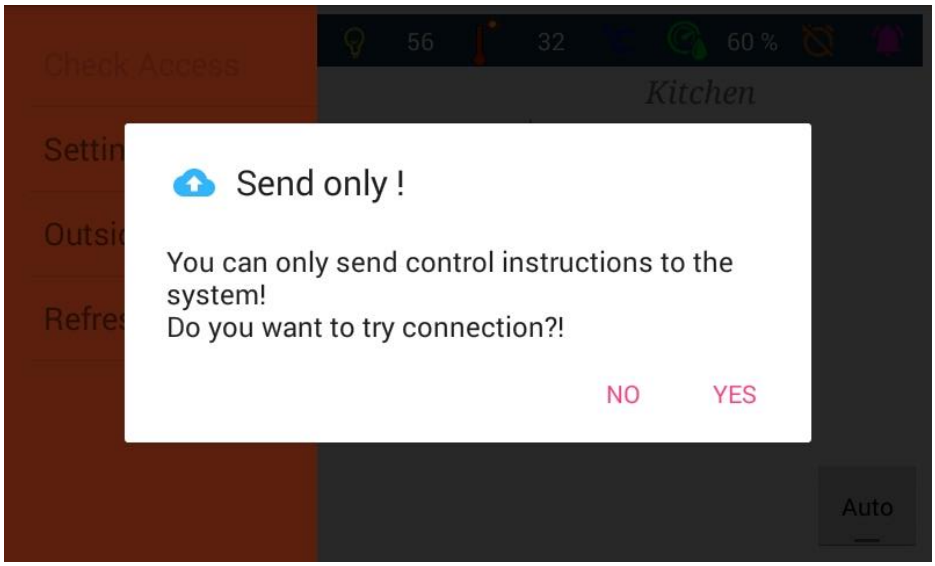
- At 12:00 AM after a good party Ahmed going back to his home but he is tired and wants to sleep, while he is driving his eyes blinks and will sleep now but the system alarms him to wake up, and he repeat this again but the system wakes him up.

- Next morning it is a holiday and Ahmed wants to listing to some music so he opens the application again and select his favorite music "Maak Alby" and plays it and enjoy for listening

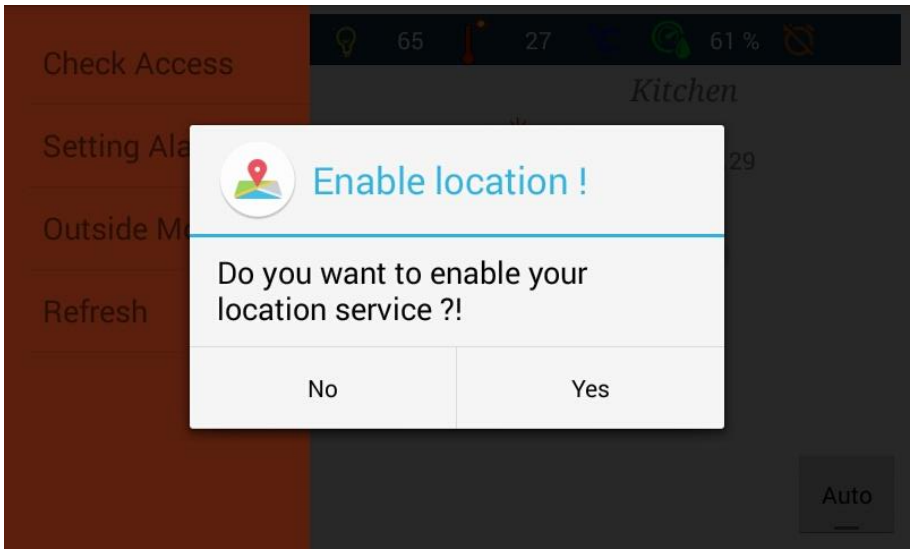
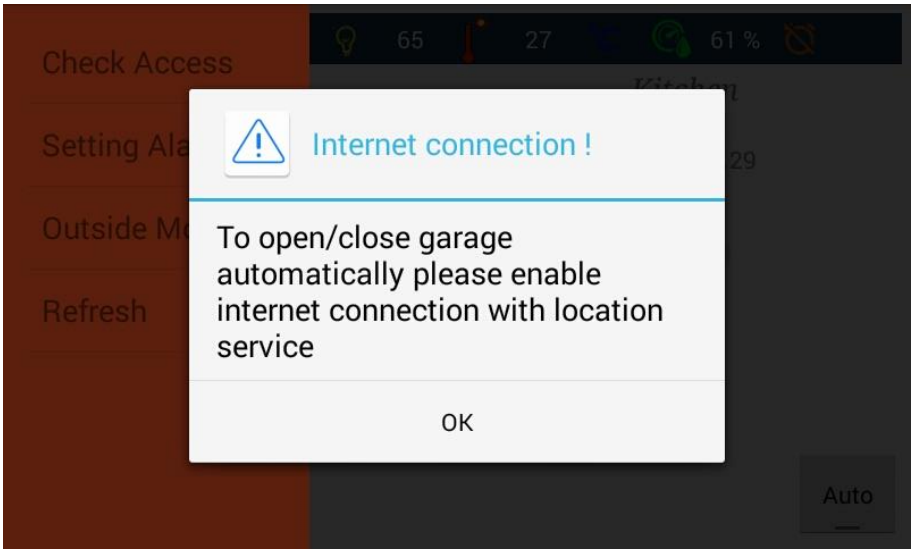
In this chapter, we will see how the system can handle errors or any invalid operations from user let's begin with the main window.

Case	No WI-FI connection
Causes	This occurs in these situations: - 1) User's device doesn't connect with any WI-FI which make the connection failed between the application and server. 2) System down or the required classes not running.
Solutions	1) We handle this case by disable all controls expect refresh and outside mode controls and showing alert dialog every one minute asking the user to reconnect with the server.  2) In this case the user must ensure that the application connects with the same WI-FI as the system otherwise user must restart the system or call system support. 

Case	Can't send instructions
Cause	This may occur when the user can't send any instructions to the system like opening the window, and this occurs when the class which creates the connection and receive instructions doesn't work well.
Solution	We handle this case by asking the user to reconnect the server, and if this failed the user must restart the system. 

Case	Can't receive any images or notifications
Cause	This occurs when the application not connected with the server class that send all unauthorized users to him/her.
Solution	We handle this case by asking the user to reconnect the server, and if this failed the user must restart the system. 

Case	User open and close bed room window every second.
Cause	This may cause the class which receive instructions down or it may make the DC-Motor off.
Solution	We handle this case by using queue data structure which store all instructions by the user and execute them one by one.

Case	No location provider
Cause	This occurs when the user outside door and getting on his car without opening the location service while driving or he didn't connect the mobile phone with the internet, and this will make the system never close or open the garage door automatically.
Solution	The solution for this problem can handle by the user if he/she enable the location service and internet connection manually. We also handle these cases but when the user press on the outside mode, then alert dialog will show asking user to enable the location service after that another alert dialog will remember him to enable the internet connection.  

Case	Exception while receiving unauthorized user image
Cause	This occurs rarely when the server sent unauthorized user to the application which make the application crash because the size of the image is big to store it in the database in the main thread or the image wasn't sent well.
Solution	We handle this case by using Handler which run a new thread for each image sent by the server to avoid this crash at runtime.

Case	Exception while delete unauthorized user image
Cause	This exception also like above, but occurs when the user wants to check unauthorized users and when he confirms or make any user unauthorized this exception occurs when application delete the image from database.
Solution	We handle this case by using Handler which run a new thread in background to avoid this crash at runtime.

Case	Face not recognized well
Cause	Occurs when the user outside door and the system recognize him, but how the user knows if the system recognizes him well or not?!
Solution	I'm using three LEDs for this situation: - 1) Yellow: Which tell tells the be ready! 2) Green: When the system recognizes the user well. 3) Red: When the system can't recognize the user well.

Case	Can't detect face while driving
Cause	This causes when the user doesn't look for his road.
Solution	The system alarms him to look the road.

Case	Eye closed while driving
Cause	This causes when the user sleeping while driving.
Solution	The system alarms him to wake up.

- I'll be fresh graduate student in a few weeks and I know that this system will be just "a graduation project" and it will be forgotten with time, so I built this system to be simple and open source in multiple domains as much as possible for anyone wants to work in embedded systems, development, designing, networks, algorithms, databases or just want to make any touch on it and this is the main idea from this project not just as a graduation project.
- Now I'll describe what you can do on this system in the future!

- Level 1 (Rise): In this level, we don't add any new functions but we will upgrade current functions to be more powerful and effective.
  - Update all GUI design to be more comfortable and accessible for users, you can remove some activities like setting alarm activity and add it with unauthorized users under control Activity or something like that. (Design)
  - I'm using socket communication with LAN for sending and receiving data between the system components, but what about if the user outside door and wants to make some changes?! So, you can modify it to make communication on the internet if LAN not available using web service like Microsoft azure or google firebase for publishing your messages or notifications. (Web Service, Network)
  - Update face recognition algorithm to be more reliable and get more accuracy by using deep neural network approach and you can use TensorFlow framework or any deep neural network frame work for that reason but make sure that your raspberry pi will work with it perfectly, I think it will be work fine with pi 3. (Development, Algorithm)
  - If there is any algorithm to make driver drowsiness more reliable and can't affected by illumination or motion while driving update it now. (Development, Algorithm)
  - Update music player to make it more powerful as you can add it in separate activity and add seekbar to it, make playlists, create directories and move/remove music files using your mobile application, transfer music files between your mobile and the system and show music file description like author, album, etc. (Development)
  - I'm using DC-Motor for opening window and garage and I control them using BWM for a few seconds you can use servo or stepper motors for different functions to make it simple. (Embedded)
  - Update your application database to save all unauthorized users images after applicable by the user. (Database)
  - Send notification to owners when any one entered the home. (Development)
  - Send notification to all home owners when car driver sleeps while driving. (Development)

- Level 2 (Professional): In this level, we will add some functions to our system.
  - Make registration and login activities for the new users. (Design, Database)
  - In our system, if we need to add new user we must capture some images for his/her face using trainer class from the system, you can add this function on your application that enable authorized users to add new users by opening mobile camera and detect new user face and capture his image with the same size of current dataset images size or the user can check his/her gallery to select new user image and the application detects his/her face after that user must add the new user name and then transfer these training images to the system. (Design, Development, Network)
  - Add GPS feature in car raspberry pi instead of mobile location service. (Embedded)
  - Add all lights and windows functions according to your view. (Electronics, Development)
  - Add setting activity which enable user to adjust his home features like min temperature that run the kitchen fan, what time required to start the café machine before the alarm, control which devices can turned on or off when outside mode active, etc. (Development)
  - Using your application to control your Curtains, TV, etc. (Embedded, Development)
  - Turning your windows and curtains off automatically when it is rainy according your sensors and switch lighting on if there is any person in the room using motion sensor. (Embedded, Development)
  - Turning some functions like TV, lights and music off when there is any gust outdoor and open the door automatically when he is coming (not for all guests see level 4). (Development)
  - Add small LEDs (color LEDs will make it awesome) to your stairs and turn them on with slow motion at night when the user's feet on the first or last stair. (Embedded, Development)
  - Add some location sensors to identify your objects location which usually loss. (Embedded, Development)
  - Add some sensors and actuators to your car like temperature sensor to measure your car temperature and switch the air conditional on if it is hot. (Embedded, Development)
  - Add screen to your car to show google map and your location (like GPS) but we will show some information and instructions from our home while driving (see level 4). (Embedded)
  - Use motion sensors to check if there is any person in the room or not and if there isn't switching lights and fans off after some time of checking. (Embedded, Development)
  - Open windows or air condition when the humidity is low. (Development)
  - Make our alarm powerful by adding new functions for user to add more than one alarm, use snooze and change alarm sound. (Development)

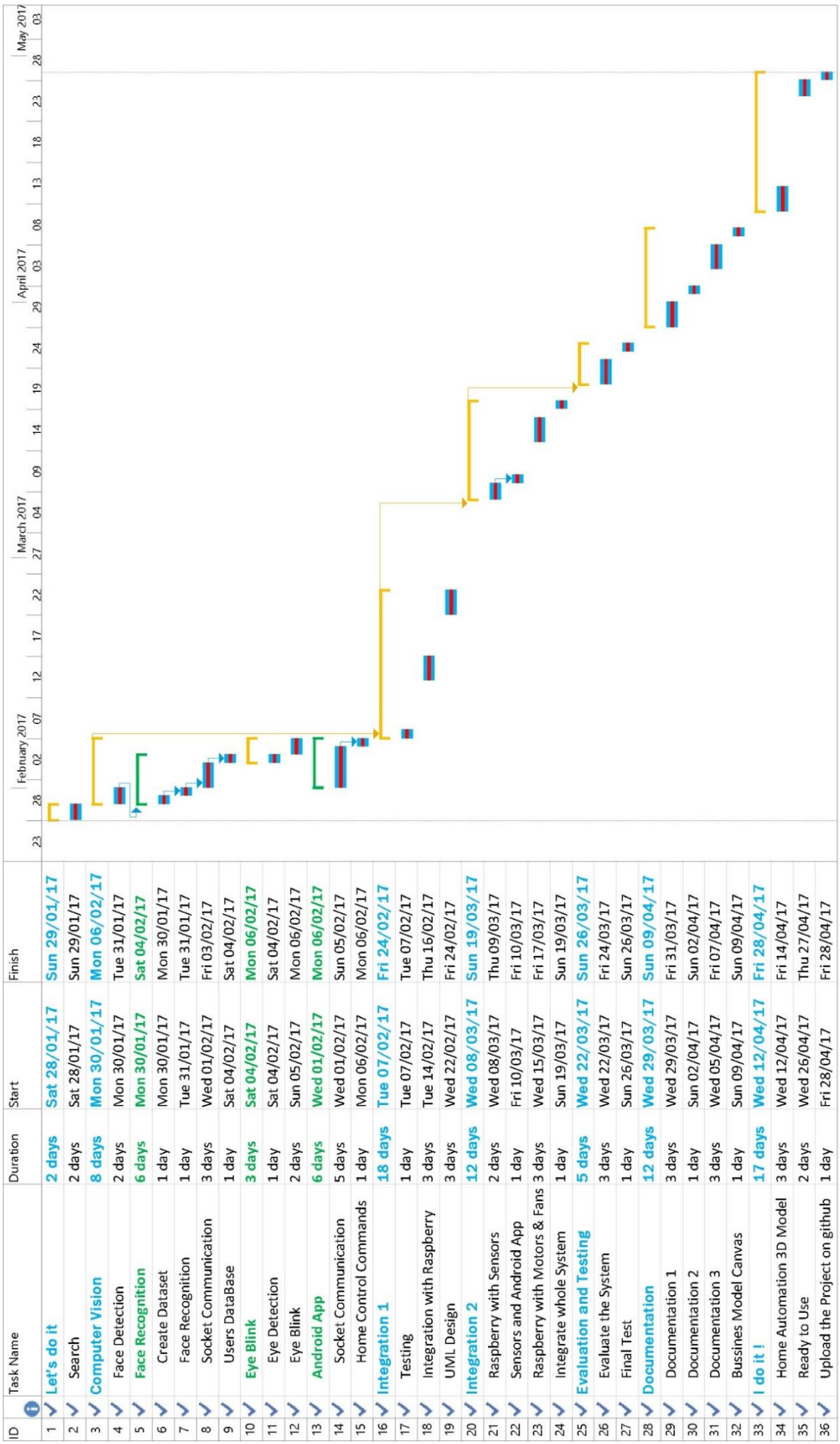
- Level 3 (Secure): In this level, we will add some security functions to our system
  - Use finger print recognition to your garage door to make it more secure. (Development)
  - Use encryption algorithm to secure your data communication. (Algorithm, Development)
  - Add face recognition in your car to identify who will drive the car and notify the user if the driver unauthorized and make some functions to force stop your car automatically for some time, if the user authorized then stop face recognition and continue with the eye blink detection in our system. (Development)
  - Add break glass sensors to your home and car windows and if any glass was broken notify the user and tell him/her which glass was broken. (Embedded, Development)
  - Use gas sensor in the kitchen for any gas leaking and notify user. (Embedded, Development)
  - Add Ultrasonic sensor to avid obstacles in while driving. (Embedded)
  - First user ever on the system we will called him/her Admin and can produce random code from the system to use it by any new user wants to register on the system and this function will be added only for Admin. (Development)

- Level 4 (Intelligent): In this level, we will make our system more intelligent and attractive
  - Use machine learning algorithm for voice recognition to make all users' instructions by his/her voice. (Algorithm, Development)
  - Use the same approach in your car to know what are best songs for the user which he/she listen to it regularly in home or car, at which time he/she listen to these songs and automatically starts it when the user drive or in his home. (Algorithm, Development)
  - Also make an algorithm to detect which places the user going on every day with his/her car and use this information to show the best road for him on car's screen. (Algorithm, Development)
  - Add an activity to the application which act as note for user and he can add some products he/she wants to buy it later from supermarket or any shop and notify him/her when he is near to the supermarket while driving or notify him/her about these products when he is going back to home and support this feature to add products using voice recognition. (Algorithm, Development)
  - Also in the same activity, user can add some places to visit it later and the system notify him/her about this when it is near. (Algorithm, Development)
  - Add function which allow any user not in your system to access your home or car (a guest) for a specific time and if the guest outdoor the door open automatically or if he/she is in the car face recognition ignores this user. (Development)
  - If the user wants to make any food like (Pizza) the kitchen notify user about all requirements and user can add any missing product to the activity which described before. (Algorithm, Development)
  - To make your kitchen smarter, in this case you must have a quadcopter which will buy all missing products which specified before and notify you to buy online if the supermarket support this feature or you can use it in shopping and all products shows in user's activity using video stream, in the future you can make a robot cooking any food independently using the expert of humans (Expert system). (Embedded, Development)
  - Use feature extraction by deep learning to extract some TV programs or matches from common websites and telling you what time of your favorite program or match and you can choose any program with different times then the system will notify you asking for watching the program and if you accept this, your TV turning on at the specific time of your program and open the channel which shows this program automatically. (Algorithm, Development, Embedded)
  - You can make your car smarter by turning it to self-driving car using camera vision which detects the road and avoid obstacles or other cars, use GPS with the camera vision, make your car control its speed to arrive your destination in time if you early or late but not decrease the speed to be like turtle or increase it much time and this problem also can be solved using machine learning the speed of the car in consideration the time, distance and road obstacles, when the car self-driving the eye detection must pause and alert user to drive again after some time and resume the eye detection in this case. (Algorithm, Development, Embedded)
  - Dream !!!

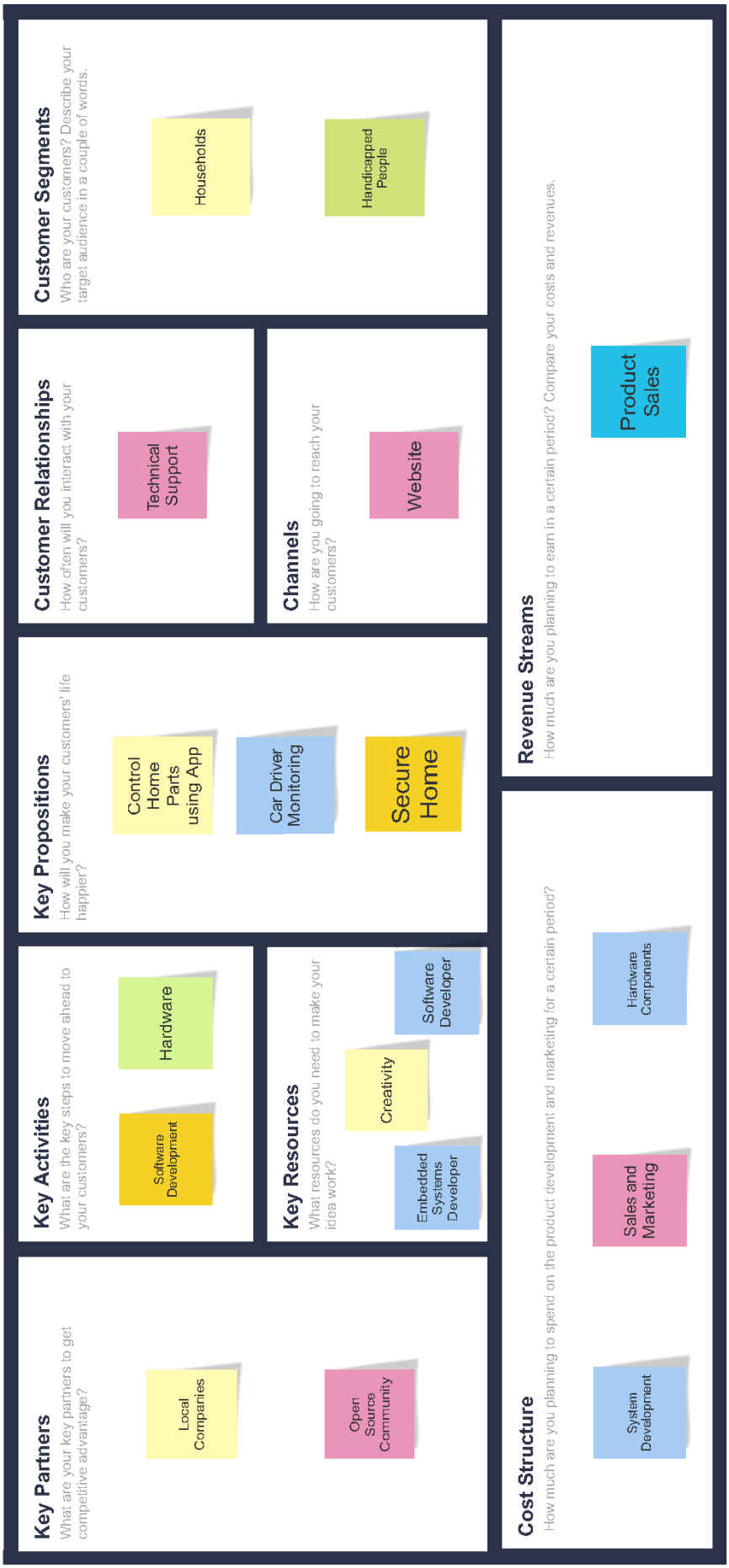
In this chapter, I will describe the budget of this project in dollar, Schedule planning and our Business model canvas.

- System cost

Tool	Quantity	Price	Average	Total price
Raspberry pi 3	2	40 – 60	50	100
Power adapter	1	5	5	5
8G micro SD card	1	6 – 8	7	7
Arduino Uno	1	15 – 20	17.5	17.5
Bread board	1	5 – 7	6	6
DC Motors	4	2 – 3	2.5	10
USB Camera	2	10	10	20
L293D IC	2	6 – 9	7.5	15
Buzzer	2	4 – 5	4.5	9
LM35	1	3	3	3
DHT22	1	10	10	10
USB Cable	1	5	5	5
4AA Battery holder	1	3	3	3
LDR	2	0.2 – 0.3	0.25	0.5
Jumpers and LEDs	...	5	5	5
Total				216



ID	Task Name	Duration	Start	Finish
1	**Let's do it**	**2 days**	**Sat 28/01/17**	**Sun 29/01/17**
2	Search	2 days	Sat 28/01/17	Sun 29/01/17
3	**Computer Vision**	**8 days**	**Mon 30/01/17**	**Mon 06/02/17**
4	Face Detection	2 days	Mon 30/01/17	Tue 31/01/17
5	**Face Recognition**	**6 days**	**Mon 30/01/17**	**Sat 04/02/17**
6	Create Dataset	1 day	Mon 30/01/17	Mon 30/01/17
7	Face Recognition	1 day	Tue 31/01/17	Tue 31/01/17
8	Socket Communication	3 days	Wed 01/02/17	Fri 03/02/17
9	Users DataBase	1 day	Sat 04/02/17	Sat 04/02/17
10	**Eye Blink**	**3 days**	**Sat 04/02/17**	**Mon 06/02/17**
11	Eye Detection	1 day	Sat 04/02/17	Sat 04/02/17
12	Eye Blink	2 days	Sun 05/02/17	Mon 06/02/17
13	**Android App**	**6 days**	**Wed 01/02/17**	**Mon 06/02/17**
14	Socket Communication	5 days	Wed 01/02/17	Sun 05/02/17
15	Home Control Commands	1 day	Mon 06/02/17	Mon 06/02/17
16	**Integration 1**	**18 days**	**Tue 07/02/17**	**Fri 24/02/17**
17	Testing	1 day	Tue 07/02/17	Tue 07/02/17
18	Integration with Raspberry	3 days	Tue 14/02/17	Thu 16/02/17
19	UML Design	3 days	Wed 22/02/17	Fri 24/02/17
20	**Integration 2**	**12 days**	**Wed 08/03/17**	**Sun 19/03/17**
21	Raspberry with Sensors	2 days	Wed 08/03/17	Thu 09/03/17
22	Sensors and Android App	1 day	Fri 10/03/17	Fri 10/03/17
23	Raspberry with Motors & Fans	3 days	Wed 15/03/17	Fri 17/03/17
24	Integrate whole System	1 day	Sun 19/03/17	Sun 19/03/17
25	**Evaluation and Testing**	**5 days**	**Wed 22/03/17**	**Sun 26/03/17**
26	Evaluate the System	3 days	Wed 22/03/17	Fri 24/03/17
27	Final Test	1 day	Sun 26/03/17	Sun 26/03/17
28	**Documentation**	**12 days**	**Wed 29/03/17**	**Sun 09/04/17**
29	Documentation 1	3 days	Wed 29/03/17	Fri 31/03/17
30	Documentation 2	1 day	Sun 02/04/17	Sun 02/04/17
31	Documentation 3	3 days	Wed 05/04/17	Fri 07/04/17
32	Bussines Model Canvas	1 day	Sun 09/04/17	Sun 09/04/17
33	**I do it !**	**17 days**	**Wed 12/04/17**	**Fri 28/04/17**
34	Home Automation 3D Model	3 days	Wed 12/04/17	Fri 14/04/17
35	Ready to Use	2 days	Wed 26/04/17	Thu 27/04/17
36	Upload the Project on github	1 day	Fri 28/04/17	Fri 28/04/17


Key Partners
What are your key partners to get competitive advantage?
Local Companies
Open Source Community
Key Activities
What are the key steps to move ahead to your customers?
Hardware
Software Development
Key Resources
What resources do you need to make your idea work?
Creativity
Software Developer
Embedded Systems Developer
Key Propositions
How will you make your customers' life happier?
Control Home Parts using App
Car Driver Monitoring
Secure Home
Customer Relationships
How often will you interact with your customers?
Technical Support
Channels
How are you going to reach your customers?
Website
Customer Segments
Who are your customers? Describe your target audience in a couple of words.
Households
Handicapped People
Revenue Streams
How much are you planning to earn in a certain period? Compare your costs and revenues.
Product Sales
Cost Structure
How much are you planning to spend on the product development and marketing for a certain period?
System Development
Sales and Marketing
Hardware Components

- Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.
  - http://ieeexplore.ieee.org/abstract/document/990517/
- Viola, Paul, and Michael J. Jones. "Robust real-time face detection." International journal of computer vision 57.2 (2004): 137-154.
  - https://link.springer.com/article/10.1023/B:VISI.0000013087.49260.fb
- Edward kim, regression boost
  - http://www.csc.villanova.edu/~ekim/classes/CSC9010F16/lect/lect8_regression_boost.pdf
- Adaboost for Face Detection
  - http://www.cse.msu.edu/~cse802/adaboostforfacedetection-1.ppt
- Robust Real-time Face Detection
  - http://users.utcluj.ro/~tmarita/HCI/C7-8-extra/Face-detect/violajones.ppt
- Belhumeur, Peter N., Joao P. Hespanha, and David J. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection." European Conference on Computer Vision. Springer Berlin Heidelberg, 1996.
  - https://link.springer.com/chapter/10.1007/BFb0015522
- Introduction to Pattern Analysis, Ricardo Gutierrez-Osuna, Texas A&M University
  - http://vision.eecs.ucf.edu/courses/cap5415/fall2011/Lecture-14.5-LDA.pdf
- Wagner, Philipp. "Face recognition with python." Tersedia dalam: www. bytefish. de [diakses pada 16 Februari 2015] (2012).
  - www.academia.edu/download/35754654/facerec_python.pdf
- Face Recognition using PCA and LDA
  - https://www.utc.edu/center-information-security-assurance/course-listing/b3.2.face.pca.lda.ppt
- Face Recognition and Feature Subspaces
  - http://dhoiem.cs.illinois.edu/courses/vision_spring10/lectures/Lecture15%20-%20Face%20Recognition.ppt
- Soukupová, Tereza. "Eye Blink Detection Using Facial Landmarks." (2016).
  - ftp://147.32.84.2/pub/cmp/articles/cech/Soukupova-TR-2016-05.pdf
- Kazemi, Vahid, and Josephine Sullivan. "One millisecond face alignment with an ensemble of regression trees." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014.
  - www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Kazemi_One_Millisecond_Face_2014_CVPR_paper.pdf
- Real-Time Eye Blink Detection using Facial Landmarks
  - https://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf
- Shen, Jie, et al. "The first facial landmark tracking in-the-wild challenge: Benchmark and results." Computer Vision Workshop (ICCVW), 2015 IEEE International Conference on. IEEE, 2015.
  - http://ieeexplore.ieee.org/abstract/document/7406481/
- Tie, Yun, and Ling Guan. "Automatic landmark point detection and tracking for human facial expressions." EURASIP Journal on Image and Video Processing 2013.1 (2013): 8.
  - https://link.springer.com/article/10.1186/1687-5281-2013-8
- Review Paper on Home Automation System using Raspberry PI
  - http://www.ijritcc.org/download/browse/Volume_4_Issues/October_16_Volume_4_Issue_10/1477755301_29-10-2016.pdf
- Rao, P. Bhaskar, and S. K. Uma. "Raspberry Pi home automation with wireless sensors using smart phone." International Journal of Computer Science and Mobile Computing 4.5 (2015): 797-803.
  - http://www.raspberrypi4you.com/wp-content/uploads/2016/11/raspberry-pi-and-domotica.pdf
- Android Based Home Automation Using Raspberry Pi
  - http://ijcat.org/IJCAT-2014/1-1/Android-Based-Home-Automation-Using-Raspberry-Pi.pdf
- Tiwari, Swati, and Rahul Gedam. "A Review Paper on Home Automation System Based on Internet of Things Technology." International Research Journal of Engineering and Technology (IRJET) 3.5 (2016).
  - https://www.irjet.net/archives/V3/i5/IRJET-V3I5240.pdf
- Home Automation through E-Mail using Raspberry Pi
  - http://ijarece.org/wp-content/uploads/2015/09/IJARECE-VOL-4-ISSUE-9-2475-2480.pdf

- Adam Harvey Explains Viola-Jones Face Detection
  - http://www.makematics.com/research/viola-jones/
- Viola-Jones Face Detection
  - https://sites.google.com/site/5kk73gpu2012/assignment/viola-jones-face-detection
- Face Detection and Recognition (Theory and Practice)
  - http://eyalarubas.com/face-detection-and-recognition.html
- OpenCV and Python Tutorial
  - https://www.youtube.com/playlist?list=PLnjEM1fs09cGGjdCLSue8Kw7GmWDhGIMh
- Face Detection and Recognition in Python with OpenCV
  - https://noahingham.com/blog/facerec-python.html
- Face Recognition in videos with OpenCV
  - http://docs.opencv.org/2.4/modules/contrib/doc/facerec/tutorial/facerec_video_recognition.html
- Face Recognition using Python and OpenCV
  - http://hanzratech.in/2015/02/03/face-recognition-using-opencv.html
- Linear Discriminant Analysis – Bit by Bit
  - http://sebastianraschka.com/Articles/2014_python_lda.html
- Facial landmarks with dlib, OpenCV, and Python
  - http://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/
- Real-time facial landmark detection with OpenCV, Python, and dlib
  - http://www.pyimagesearch.com/2017/04/17/real-time-facial-landmark-detection-opencv-python-dlib/
- Eye blink detection with OpenCV, Python, and dlib
  - http://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/
- Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning
  - https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78
- Client Server System with Sockets
  - https://pythonprogramming.net/client-server-python-sockets/
- Sending images over websockets in Python 2.7
  - http://laht.info/sending-images-over-websockets-in-python-2-7/
- Android Client-Server Using Sockets – Server Implementation
  - http://androidsrc.net/android-client-server-using-sockets-server-implementation/
- Android Client-Server Using Sockets – Client Implementation
  - http://androidsrc.net/android-client-server-using-sockets-client-implementation/
- DC Motor Dual H-Bridge IC-L293D
  - http://www.androiderode.com/dc-motor-dual-h-bridge-ic-l293d/
- Pulse-width Modulation
  - https://learn.sparkfun.com/tutorials/pulse-width-modulation
- 78 Home Automation Ideas
  - https://www.control4.com/blog/2015/10/78-home-automation-ideas
- OpenCV Library
  - http://opencv.org/
- dlib for facial landmarks
  - https://pypi.python.org/pypi/dlib
- iconfinder for some icons
  - https://www.iconfinder.com/