

# Software Design Document for: Emotion Detection Project

Bhavya Bansal , Ritveeka Vashishta, Surabhi Gupta

April 2, 2017

## Contents

<b>1 INTRODUCTION</b>	<b>1</b>
1.1 PURPOSE . . . . .	1
1.2 PRODUCT SCOPE . . . . .	2
1.3 INTENDED AUDIENCE . . . . .	2
1.4 DEFINITIONS . . . . .	2
1.5 REFERENCES . . . . .	2
1.6 TOOLS USED . . . . .	3
<b>2 DESIGN OVERVIEW</b>	<b>3</b>
2.1 Description of problem . . . . .	3
2.2 Technologies used . . . . .	3
2.3 System Architecture . . . . .	3
<b>3 USER'S VIEW</b>	<b>5</b>
3.1 List of use cases : . . . . .	5
3.2 Use case diagram . . . . .	5
3.3 Use cases . . . . .	6
<b>4 STRUCTURAL VIEW</b>	<b>8</b>
4.1 Class Diagrams . . . . .	10
4.2 Objects . . . . .	11
<b>5 BEHAVIORAL VIEW</b>	<b>18</b>
5.1 Sequence Diagram . . . . .	18
<b>6 DATABASE DESIGN</b>	<b>21</b>
6.1 Entity Relation Diagrams . . . . .	21

## 1 INTRODUCTION

### 1.1 PURPOSE

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams.

## 1.2 PRODUCT SCOPE

This Software Design Document is for a base level system which will work as a proof of concept for the use of building a system that provides a base level of functionality to show feasibility for large scale production use. This Software Design is focused on the base level system and critical parts of the system. For this particular Software Design Document, the focus is placed on generation of the documents and modification of the documents. The system will be used in conjunction with other pre-existing systems and will consist largely of a document interaction facade that abstracts document interactions and handling of the document objects.

## 1.3 INTENDED AUDIENCE

primary audiences for this document are the software developers

## 1.4 DEFINITIONS

- 1) **face detection** : to detect the user's face in an image or a video
- 2) **emotion detection** : to detect the emotion of the person present in the image or as shown by front camera. it may be happy, sad, fear, disgust and focus.
- 3) **Float** : used to represent decimal numbers
- 4) **hashtable** : data structure to store data and supports linear search
- 5) **UML** : The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system. UML offers a way to visualize a system's architectural blueprints in a diagram
- 6) **object oriented** : using a methodology which enables a system to be modelled as a set of objects which can be controlled and manipulated in a modular manner.
- 7) **objects** : Objects represent an entity and the basic building block
- 8) **classes** : Class is the blue print of an object which represents both the main elements, interactions in the application, and the classes to be programmed.
- 9) **interface** : Interface defines a set of operations which specify the responsibility of a class.
- 10) **machine learning** : is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can change when exposed to new data.

## 1.5 REFERENCES

- 1) Software Engineering by K.K Agarwal and Yogesh Singh
- 2) Software engineering Pressman
- 3) [ieeexplore.ieee.org/document](http://ieeexplore.ieee.org/document)
- 4) [www.wikipedia.com](http://www.wikipedia.com)
- 5) [nptel.ac.in](http://nptel.ac.in)

## 1.6 TOOLS USED

- 1) lucidchart.com
- 2) Latex
- 3) Smart draw

## 2 DESIGN OVERVIEW

### 2.1 Description of problem

In this product we have to develop a system which can detect emotion in real time / image and share it with other people. For the first version our aim is to identify fear, happiness, disgust, focus and sadness.

### 2.2 Technologies used

Emotion detection is done via machine learning and deep learning algorithms. Technologies which help to share information will be used to share results. Several applications present in the mobile phone like gallery will also be used.

### 2.3 System Architecture

1) **Image selection interface :** The windowed interface for selecting image of your choice from the images stored in gallery of your mobile phone.

2 **Face detection interface :** The simple screen which shows your image or the video captured by front camera with the detected face outlined .

3 **Emotion detection interface :** The interface for showing detected emotion with the video which is seen by the front camera or the image selected from the gallery.

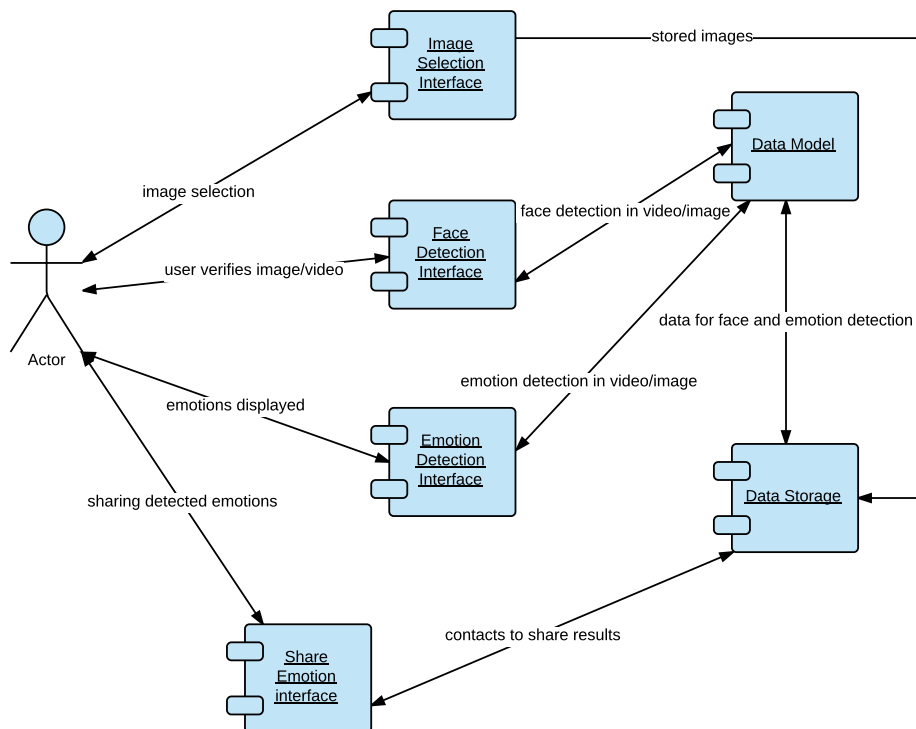
4 **Share result interface :** The interface for sharing the emotion detection results with friends and relatives.

5 **Data Model :** The classes needed to organise face detection, emotion detection, verification and video recording.

6 **Data storage :** The interface for storing, importing and exporting the data model and raw collected data.

## 2.4 SYSTEM ARCHITECTURE DIAGRAM

The fundamental organisation of a system, embodied in its components, their relationship with each other and to the environment, and the principles governing its design and evolution is illustrated by the following diagram. This diagram depicts the interaction between different interfaces, user, data models and data stores.

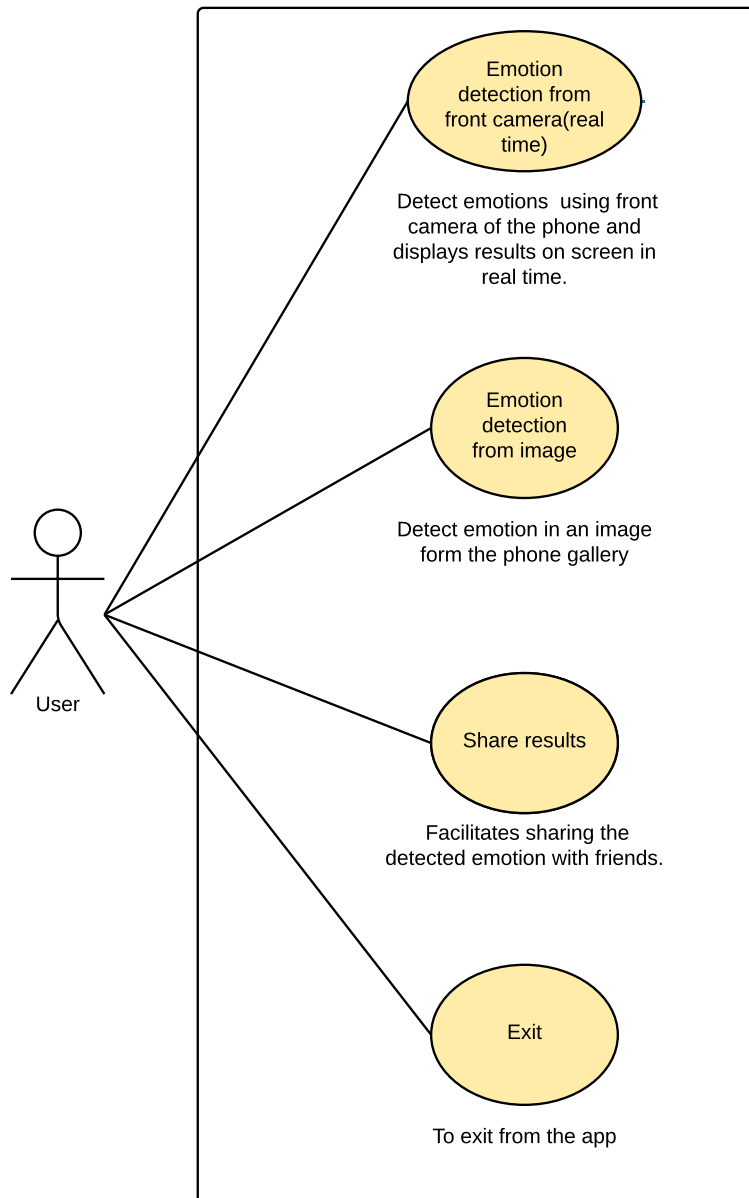


### 3. USER'S VIEW

**ACTOR :** Any person who has the app installed in her smartphone and wishes to get her emotion analysed

#### 3.1 List of use cases :

- 1) Real time emotion detection using front camera.
- 2) Emotion detection from image
- 3) Share results
- 4) Exit



#### 3.2 USE CASE DIAGRAM

### 3.3 Use cases

#### USE CASE-1 - EMOTION DETECTION IN REAL TIME

**Introduction :** This use case let's the user do emotion detection in real time using the front camera of her smart phone.

**Pre condition :** The user must have selected "emotion analysis in real time" option in the mode selection screen.

**Post condition :** If this use case is successful, the user can see her emotions in real time using the front camera of the smart phone.

**Main Line Sequence :**

**User :** selects the real time emotion detection mode

**System :** opens the front camera and shows the detected face by indicating a box around the most prominent face.

**user :** Gives confirmation that the detected face is the one for which she wants to see emotion for.

**System :** After getting the confirmation from the user, it does the emotion analysis and displays the result to the user.

**Alternate sequence :**

**User :** Does not confirm to the detected face by clicking the no button.

**System :** Goes back and displays the main screen which again prompts the user to choose between the mode.

#### USE CASE-2 - EMOTION DETECTION ON IMAGE

**Introduction :** This use case let's the user do emotion analysis of some image stored in her smart phone.

**Pre condition :** The user must have selected "emotion analysis from image" option in the mode selection screen.

**Post Condition :** - If this use case is successful, the user can see the emotions of people in images that are stored in her phone and subsequently share the result.

**Mainline Sequence :**

**User :** selects the 'emotion detection from image' option in the mode selection screen.

**System :** opens the phone gallery and displays all the images to the user.

**User :** selects an image from the list of displayed images.

**System :** performs face detection and displays to the user modified image with a box surrounding the more prominent face.

**user :** confirms if the detected face is the one for which she wants to perform emotion analysis.

**System :** performs emotion analysis and displays the result to the user.

**Alternate sequence :**

**User :** does not confirm to the detected face on the image.

**System :** Goes back and displays the main screen which again prompts the user to choose between the mode.

#### USE CASE-3 - SHARING

**Introduction :** This use case lets the user share her emotions detection results with friends and acquaintances.

**Pre condition :** The user have have performed emotion analysis using the "emotion detection from image mode".

**Post condition :** If this use case is successful, the emotion detection result will be shared with the contact chosen by the user

**Mainline Sequence :**

**User :** selects the share option after getting the emotion results

**System :** opens a list of contacts saved in the phone.

**User :** Select the contact with whom she wants to share the results.

**System :** Displays a list of applications available for sharing

**User :** selects an app from the displayed list

**System :** shares the results and displays a message indicating success (or failure)

#### **USE CASE 4 - EXIT**

**Introduction :** - This lets the user exit from the app.

**Pre condition :** The user must be present in the home page to be able to exit from the app.

**Post Condition :** The app gets closed

##### **Mainline Sequence :**

**User** - selects the exit button

**system** - seeks confirmation

**User** - confirms exit

**system** - closes the app

##### **Alternate Sequence :**

**User :** - does not confirm to exit

**system :** closes the confirmation dialogue box and remains in it's state

## 4. STRUCTURAL VIEW

### 4.1 Classes

Boundary class:

**<<boundary>>  
Home**

+ screenHeight() : Float  
+ screenWidth() : Float

+ displayEmotionResult( emotion data);  
+ displayImage (image data) ;  
+ displayVideo(video data) ;  
+ displayHomePage() ;  
+ displaySuccessMessage(String);  
+ confirmExitPrompt(String) : Boolean;

Entity Class :

**<<entity>>  
ImageRegister**

+ numberOfImages : Integer  
+ Imagelist : Hashtable

+ getAllImages() : all images;  
+ getImage(image id) : image;  
+ indexOutOfBounds(image id): Boolean;  
+ isEmpty() : Boolean

**<<entity>>  
Image**

+ id : String  
+ size : Float

+ setId(image id);  
+ getId() : image id;

**<<entity>>  
ContactRegister**

+ numberOfcontacts : Integer  
+ contactlist : Hashtable

+ getContacts(): all contacts ;  
+ getNumberOfContacts() : Integer;  
+ indexOutOfBounds(Integer): Boolean  
+ isEmpty(): Boolean

**<<entity>>  
Contact**

+ name: Integer  
+ contactId : Integer

+ setName(contact name);  
+ getName(contact id): contact name;



Controller Class :

<<controller>> FaceDetection
+ faceCoordinates : Float
+ detectFace(data) : face coordinates; + faceOutline(face coordinates) : video/image with outlined face

<<controller>> EmotionDetection
+ happiness : Float + focus : Float + stress : Float + fear : Float + disgust : Float
+ detectEmotion(face coordinates) : emotion result;

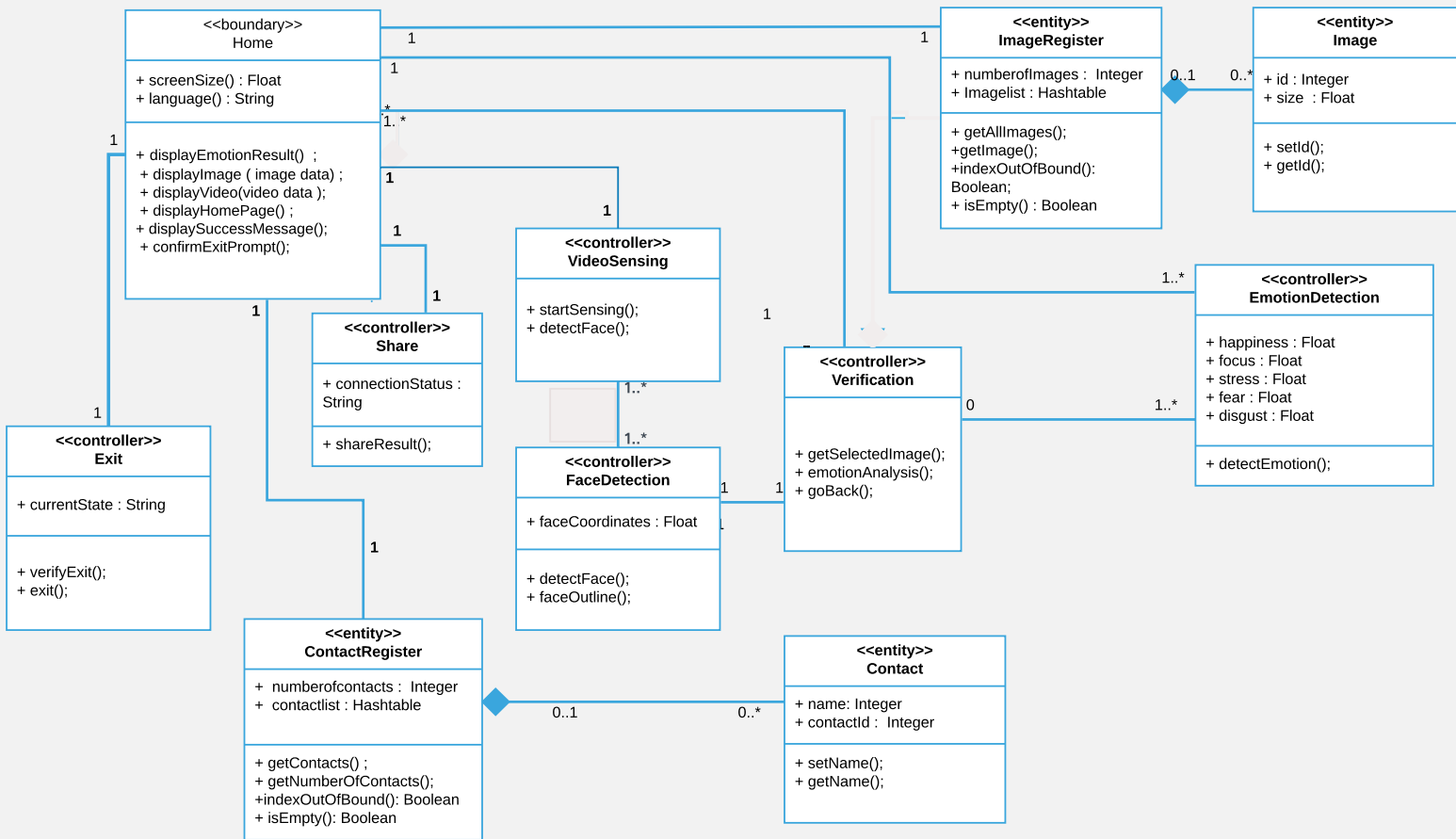
<<controller>> Verification
+getSelectedImage (image Id): image data; + emotionalAnalysis(); + goBack();

<<controller>> Exit
+ currentState : String
+ verifyExit(); + exit();

<<controller>> Share
+ connectionStatus : String
+ shareResult(emotion result) : String;

<<controller>> VideoSensing
+ startSensing(): video data; + detectFace(video data);

## 4.2 CLASS DIAGRAM



## 4.3 Objects

Class Name : share	
<b>Brief Description</b> : The methods of this class facilitates the user to share the emotion result with friends and acquaintances.	
Attributes(fields)	Attribute Description
Connection Status	The connection status indicates the availability of network to facilitate sharing with a contact of choice.
Methods(operations)	Method Description
shareResult()	<i>Parameter</i> : int <i>return type</i> : void <i>Description</i> : This method is invoked by the home boundary class. It shares the emotion result with the selected contact.

Class Name : Video Sensing	
<b>Brief Description</b> : The methods of this class senses the real time image from the front camera and invokes methods from face detection class sends the modified image back to the user	
Methods(operations)	Method Description
startSensing()	<i>Parameters</i> : image data <i>Return Type</i> : Null <i>Description</i> : This method, invokled by the home boundary class object senses the image data from the front camera and and invokes face detection class.
detectFace()	<i>Parameters</i> : modified image data <i>Return type</i> : null <i>Description</i> : Invokes the method of face detection class to get the detected face

Class Name : Home	
<b>Brief Description :</b>	
Attributes(fields)	Attribute Description
Screen Size	This specifies the screen width and height
Language	This specifies the language that the boundary uses to interact with the user
Methods(operations)	Method Description
displayEmotionResult()	<i>Parameter :</i> Emotion data after being calculated from emotion detection class. <i>Return type :</i> Void <i>Description :</i> This method is invoked by the verification controller once it has recieved the calculated emotion data. It displays the final result to the user.
displayVideo()	<i>parameter :</i> video data <i>return type :</i> void <i>Description :</i> This method is invoked by the video sensing controller. It displays the result to the user when in real time mode
displayHomePage()	<i>Parameter :</i> void <i>Return Type :</i> void <i>Description :</i> If the user does not give confirmation to the detected face, this method takes the user back to the home page, the main activity page wherein the user can again select between the two modes and proceed accordingly.
displaySuccessMsg()	<i>Parameter :</i> string <i>Return Type :</i> void <i>Description :</i> This method is invoked by the share controller. It displays a message on the screen informing the user whether or not sharing was successful.
confirmExitPrompt()	<i>Parameter :</i> string <i>Return type :</i> void <i>Description :</i> This method is invoked by the exit controller. It prompts the user to confirm whether to exit or not
displayImage()	<i>Parameter :</i> Image data <i>Return type :</i> void <i>Description :</i> This method is invoked by the image register class to display all the images to the user at the boundary.

Class Name : <u>FaceDetection</u>	
<b>Brief Description :</b> It is a controller class which implements the logic of Face Detection in image or video from front camera . It is important to detect face before starting face detection because if correct face is not detected in the data then there is no point in doing emotion detection.	
Attributes (fields)	Attribute Description
faceCoordinates	It stores the coordinates of face in the data for which face detection has to be done
Methods(operations)	Method Description
detectFace()	<i>parameters :</i> bit data <i>return type :</i> none <i>description :</i> It runs a machine learning algorithm on the image and detect the face in the image and sets the faceCoordinates
faceOutline()	<i>parameters :</i> bit data <i>return type :</i> bit data <i>description :</i> it draws a rectangle around the face according to the faceCoordinates and returns the modified bit data

Class Name : <u>Verification</u>	
<b>Brief Description :</b> It is a controller class , it's logic is to verify if the user wants to proceed with emotion detection after seeing the detected faces in the image and the video from front camera	
Methods(operations)	Method Description
getSelectedImage()	<i>parameters :</i> int - image id <i>return type :</i> image data <i>description :</i> It returns the image which has the same id as the parameter . It uses the getImage method of imageRegister to implement it.
emotionAnalysis()	<i>parameters :</i> none <i>return type :</i> none <i>description :</i> This method gets the data from videoSensing class and uses emotion detection controller class to do the emotion analysis of the data and displays the processed data on the boundary using displayVideo method.

Class Name : <u>ImageRegister</u>	
<b>Brief Description</b> : Object of this class stores all images	
Attributes (fields)	Attribute Description
numberOfImages	It is the total number of images stored in image register
imageList	This data structure store all images
Methods(operations)	Method Description
getAllImages()	<i>parameters</i> : none <i>return type</i> : list of images <i>description</i> : this method returns all the images stored in the attribute and will send error if it is empty.
getImage()	<i>parameters</i> : int - image id <i>return type</i> : image <i>description</i> : this method returns image of the specified id stored in imageList
indexOutOfBounds()	<i>parameters</i> : long - image id <i>return type</i> : boolean <i>description</i> : this method checks if the index is more than the index of the last image stored in the imageList
isEmpty()	<i>parameters</i> : none <i>return type</i> : boolean <i>description</i> : this method checks if the imageList is empty or not

Class Name : Home	
<b>Brief Description :</b>	
Attributes(fields)	Attribute Description
Screen Size	This specifies the screen width and height
Language	This specifies the language that the boundary uses to interact with the user
Methods(operations)	Method Description
displayEmotionResult()	<i>Parameter :</i> Emotion data after being calculated from emotion detection class. <i>Return type :</i> Void <i>Description :</i> This method is invoked by the verification controller once it has recieved the calculated emotion data. It displays the final result to the user.
displayVideo()	<i>parameter :</i> video data <i>return type :</i> void <i>Description :</i> This method is invoked by the video sensing controller. It displays the result to the user when in real time mode
displayHomePage()	<i>Parameter :</i> void <i>Return Type :</i> void <i>Description :</i> If the user does not give confirmation to the detected face, this method takes the user back to the home page, the main activity page wherein the user can again select between the two modes and proceed accordingly.
displaySuccessMsg()	<i>Parameter :</i> string <i>Return Type :</i> void <i>Description :</i> This method is invoked by the share controller. It displays a message on the screen informing the user whether or not sharing was successful.
confirmExitPrompt()	<i>Parameter :</i> string <i>Return type :</i> void <i>Description :</i> This method is invoked by the exit controller. It prompts the user to confirm whether to exit or not
displayImage()	<i>Parameter :</i> Image data <i>Return type :</i> void <i>Description :</i> This method is invoked by the image register class to display all the images to the user at the boundary.

Class Name : <u>Image</u>	
<b>Brief Description :</b> Object of this class is a image which is stored in gallery of user's mobile	
Attributes (fields)	Attribute Description
id	each image has a id , this id is helpful in finding the image in imageList
size	It stores the size of image as width and height (width,height)
Methods(operations)	Method Description
setId()	<i>parameters :</i> int <i>return type :</i> none <i>description :</i> It sets the id of the image object to the parameter
getId()	<i>parameters :</i> none <i>return type :</i> int <i>description :</i> this method returns the id of the image object

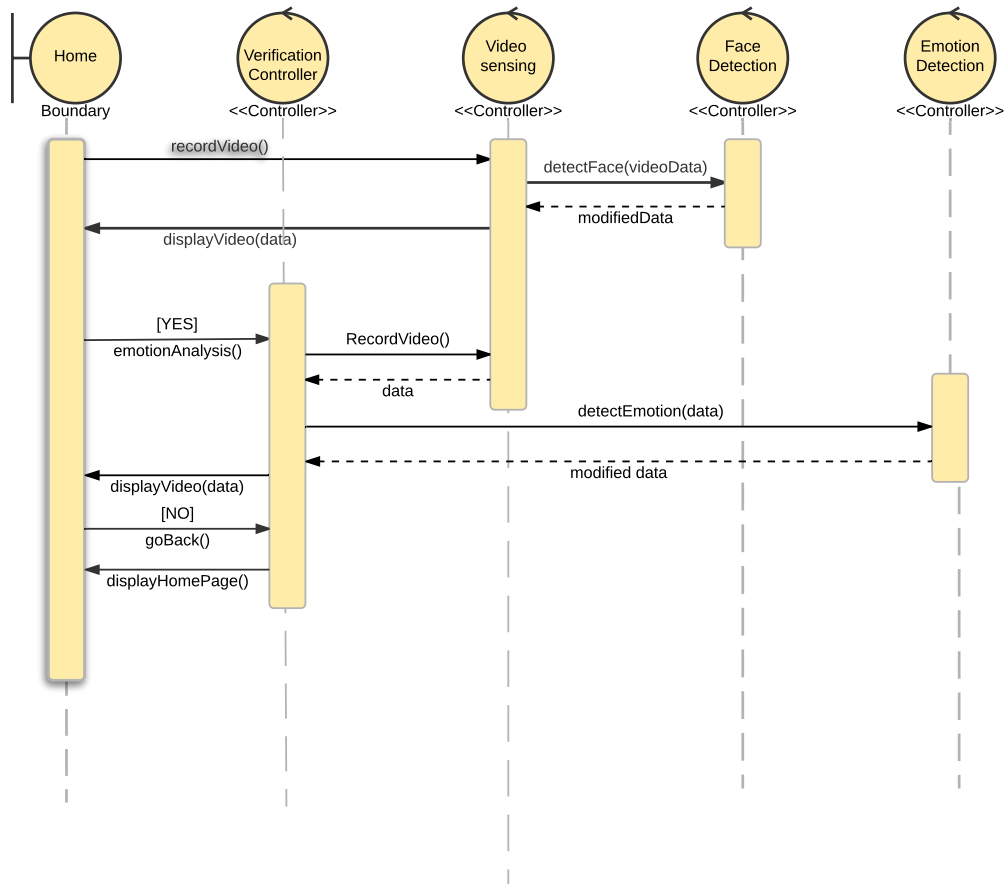
Class Name : <u>Contact</u>	
<b>Brief Description :</b> Object of this class is a contact to which user can contact and share things with	
Attributes (fields)	Attribute Description
contactId	each image has a contactId , this id is helpful in finding the conatct in conatctList
name	each contact has a name so that we can address our messages by this name
Methods(operations)	Method Description
setName()	<i>parameters :</i> string <i>return type :</i> none <i>description :</i> It sets the name of the contact object to the parameter
getId()	<i>parameters :</i> none <i>return type :</i> string



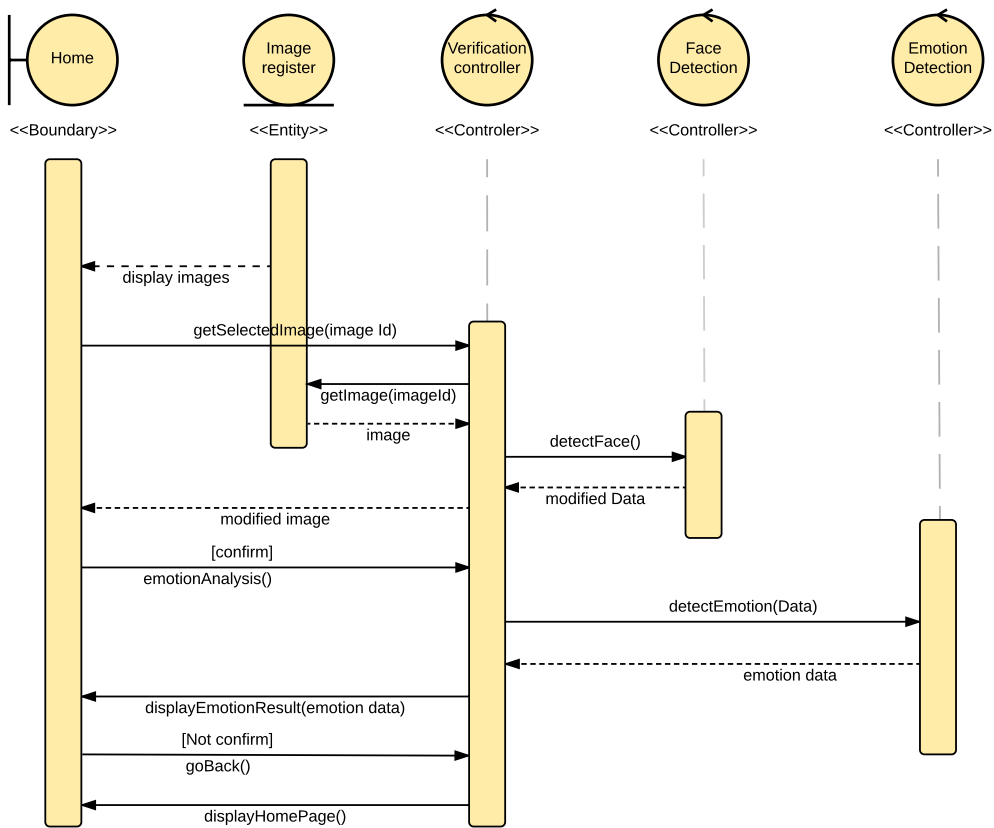
<b>Class Name</b> : Contact register	
<b>Brief Description</b> : The object of this class stores all contacts	
<b>Attributes(fields)</b>	<b>Attribute Description</b>
number of contacts	The total number of contacts stored in the database
contact list	This hash table stores all contacts
<b>Methods(operations)</b>	<b>Method Description</b>
getContacts()	<i>Parameter</i> : int <i>Return type</i> : list of contats <i>Description</i> : This method, when invoked by an instance of the boundary class sends a list of all the contacts stored in the phone
getNumberOfContacts()	<i>Parameter</i> : Null <i>Return Type</i> : Int <i>Description</i> : Returns the total number contacts
indexOutOfBounds()	<i>Parameter</i> : Int <i>Return Type</i> : Bool <i>Description</i> : checks if the index of the contact required is within bounds
isEmpty()	<i>Parameter</i> : void <i>Return Type</i> : Bool <i>Description</i> : checks if the contact list is empty

## 5. BEHAVIOURAL VIEW

### 5.1 Sequence Diagrams

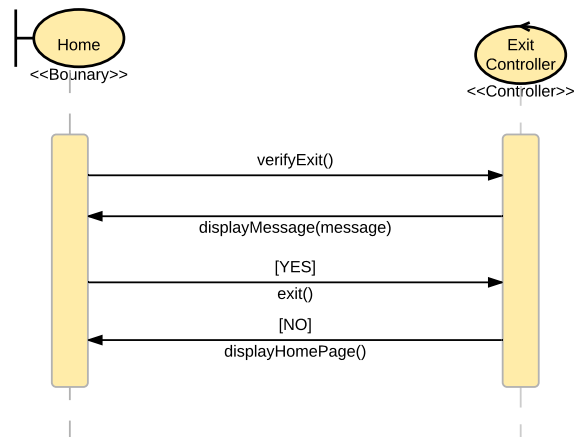
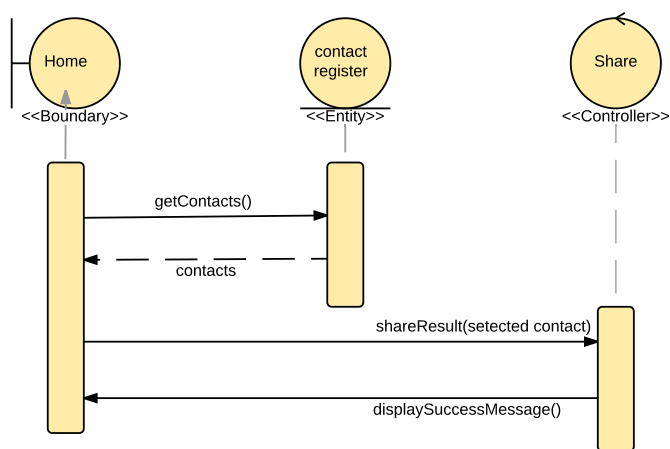


#### 5.1.1 Sequence Diagram : Real time emotion detection



5.1.2 Sequence Diagram : Emotion detection in Image

### 5.1.3 Sequence Diagram : Share



### 5.1.4 Sequence Diagram : Exit

## 6. DATABASE DESIGN

### 6.1 Entity Relation Diagram

