Applications Of Mobile Visual Identification

Submitted By

Sunny Kachhia

13MCEC06

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

March 2015

Applications Of Mobile Visual Identification

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Sunny Kachhia

(13MCEC06)

Guided By

Prof. Vipul Chudasama

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

March 2015

Certificate

This is to certify that the major project entitled **"Applications Of Mobile Visual Identification"** submitted by **Sunny Kachhia (Roll No: 13MCEC06)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Institute of Technology, Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Vipul Chudasama
Guide & Associate Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. Vijay Ukani
Associate Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Sunny Kachhia**, Roll. No. **13MCEC06**, give undertaking that the Major Project entitled "**Applications Of Mobile Visual Identification**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Prof. Vipul Chudasama

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Vipul Chudasama**, Associate Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

See that you acknowledge each one who have helped you in the project directly or indirectly.

- Sunny Kachhia

13MCEC06

Abstract

The objective of the project is to develop an application that can be used in organizations, hospitals and educational institutions to keep record of daily attendances of people. Another advantage is that face recognition works well for both: images of single person or a group. There are three major components in this project: 1.Client Application (Desktop and Mobile to capture/upload images), 2.Server (Cloud where actual face detection/recognition algorithms reside) and 3.Web interface (to display result statuses). There is no need for any external hardware installation. Only a device with a camera (laptop with webcam, pc with webcam, smart phone) is required for capturing/uploading images to server.

Abbreviations

BYOD	Bring Your Own Device.
AVBPA	Audio and Video-Based Person Authentication.
AFGR	Automatic Face and Gesture Recognition.
FRT	Face Recognition Techniques.
FERET	Facial Recognition Technology.
FRVT	Face Recognition Vendor Test.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Knowledge Discovery Process

Face Detection and Face Recognition are utilized in extensive applications such as security in an organizaiton, marking attendance of the employees or students, verifying and authenticating faces. Illumiantion, Heavy Computation, Image Quality, Image Position are some important issue of such development.

1.2 Objective

The objective of the project is to develop an application that can be used in organizations, hospitals and educational institutions to keep record of daily attendances of people and is based on concepts of computer vision and machine learning. It is used to detect and recognize people and mark attendance in organizations or educational institutions.

1.3 Motivation

In organizations it would be useful for managers to supervise their employees attendance and which is important in generating their monthly salary. In educational institutions it can be used to mark attendance for various sessions in a day and which is crucial in determining the students eligibility to appear for exams. Another advantage is that face recognition works well for both: images of single person or a group.

1.4 Scope and Methodology

There are three major components in this project namely:

1. Client Application (Desktop and Mobile),
2. Server (Cloud) and
3. Web interface.

There is no need for any external hardware installation. Only a device with a camera (laptop with webcam, pc with webcam, smart phone) is required for capturing/uploading images to server.

The actual process of face recognition happens on server, where algorithms are written to do this. The client application invokes the server programs on the cloud.

Once the server programs finish execution, the results are displayed on the Web interface. The web interface is interactive and user friendly. The status of each person for that day will be shown on the web interface.

1.4.1 Limitations

The limitation of this project is that it cannot be applied for a real time scenario where the results are required immediately. This is because all the proessing has to be done on the server side and hence not on the client machine itself. High Computation power for image processing and generating result matrix reduces the speed of getting the results.

Chapter 2

Literature Survey

2.1 General

As one of the most successful applications of image analysis and understanding, face recognition has recently received significant attention, especially during the past few years. This is evidenced by the emergence of face recognition conferences such as the International Conference on Audio and Video-Based Person Authentication (AVBPA) since 1997 and the International Conference on Automatic Face and Gesture Recognition (AFGR) since 1995, systematic empirical evaluations of face recognition techniques (FRT), including the:

- FERET [Phillips et al. 1998b, 2000; Rizvi et al. 1998],
- FRVT 2000 [Blackburn et al. 2001],
- FRVT 2002 [Phillips et al. 2003], and
- XM2VTS [Messer et al. 1999]

protocols, and many commercially available systems. There are at least two reasons for this trend; the first is the wide range of commercial and law enforcement applications and the second is the availability of feasible technologies after 30 years of research.

In addition, the problem of machine recognition of human faces continues to attract researchers from disciplines such as image processing, pattern recognition, neural networks, computer vision, computer graphics, and psychology. The strong need for user-friendly systems that can secure our assets and protect our privacy without losing our identity in

a sea of numbers is obvious. At present, one needs a PIN to get cash from an ATM, a password for a computer, a dozen others to access the internet, and so on. Although very reliable methods of biometric personal identification exist, for example, fingerprint analysis and retinal or iris scans, these methods rely on the cooperation of the participants, whereas a personal identification system based on analysis of frontal or profile images of the face is often effective without the participants cooperation or knowledge.

Areas	Specific applications
Entertainment	Video game, virtual reality, training programs
	Human-robot-interaction, human-computer-interaction
Smart cards	Drivers' licenses, entitlement programs
	Immigration, national ID, passports, voter registration
	Welfare fraud
Information security	TV Parental control, personal device logon, desktop logon
	Application security, database security, file encryption
	Intranet security, internet access, medical records
	Secure trading terminals
Law enforcement and surveillance	Advanced video surveillance, CCTV control
	Portal control, postevent analysis
	Shoplifting, suspect tracking and investigation

Figure 2.1: Typical Applications of Face Recognition.[3]

2.2 Literature Review

2.2.1 Face Recognition From Still Images

As illustrated in Figure below, the problem of automatic face recognition involves three key steps/subtasks:

(1) Detection and rough normalization of faces,

(2) Feature extraction and accurate normalization of faces,

(3) Identification and/or verification.

Commercial and law enforcement applications of FRT range from static, controlled-format photographs to uncontrolled video images, posing a wide range of technical challenges and requiring an equally wide range of techniques from image processing, analysis,

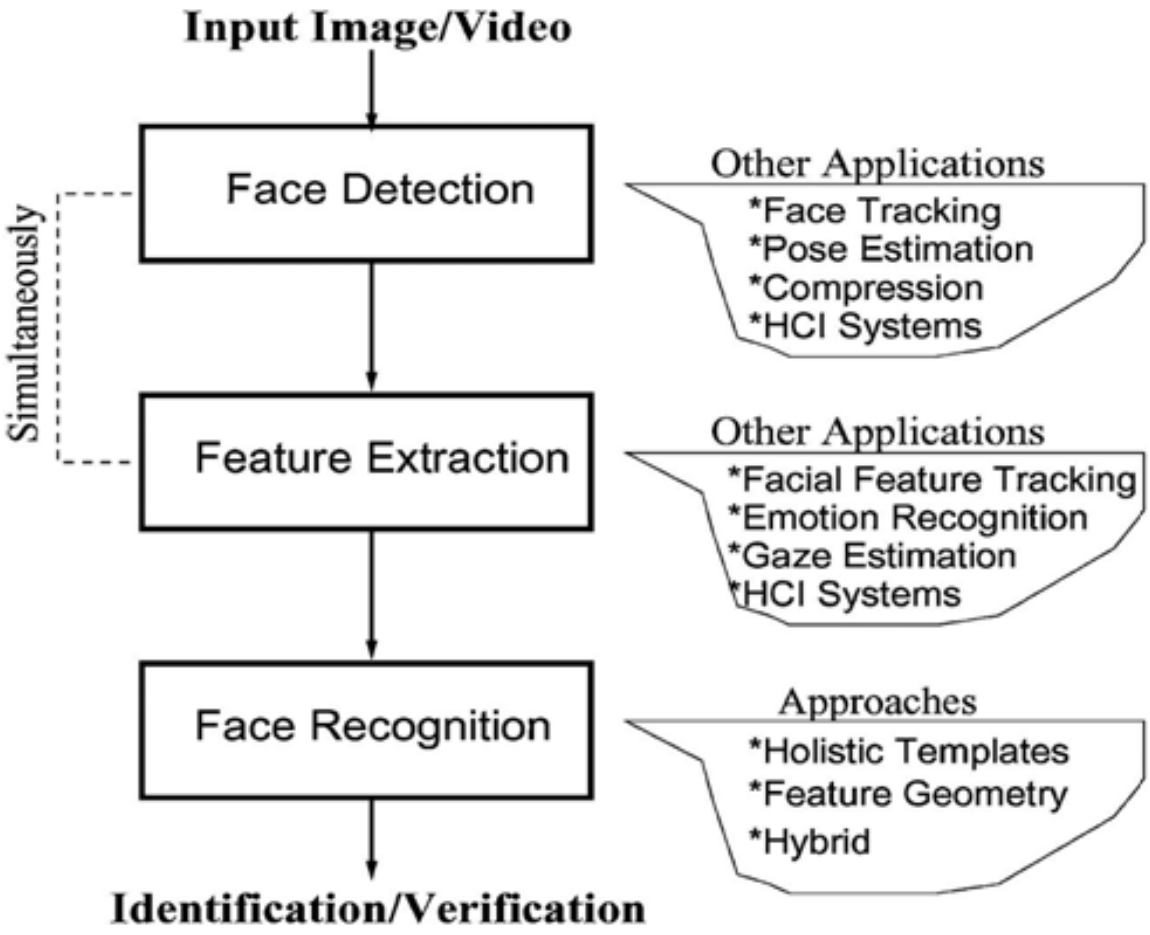

Figure 2.2: Configuration of a generic face recognition system.[3]

understanding, and pattern recognition. One can broadly classify FRT systems into two groups depending on whether they make use of static images or of video. Within these groups, significant differences exist, depending on the specific application. The differences are in terms of image quality, amount of background clutter (posing challenges to segmentation algorithms), variability of the images of a particular individual that must be recognized, availability of a well-defined recognition or matching criterion, and the nature, type, and amount of input from a user.

2.2.2 Issues in Face Recognition

Illumination and pose variation[4]

Face recognition in an uncontrolled environment is still very challenging. For example, the FERET evaluations and FRVTs revealed that there are at least two major challenges: the illumination variation problem and the pose variation problem. Though many existing systems build in some sort of performance invariance by applying pre-processing methods such as histogram equalization or pose learning, significant illumination or pose change can cause serious performance degradation. In addition, face images can be partially occluded, or the system may need to recognize a person from an image in the database that was acquired some time ago (referred to as the duplicate problem in the FERET tests). These problems are unavoidable when face images are acquired in an uncontrolled, uncooperative environment, as in surveillance video clips.

Pros and cons of these approaches are pointed out so an appropriate approach can be applied to a specific task. The majority of the methods reviewed here are generative approaches that can synthesize virtual views under desired illumination and viewing conditions. Many of the reviewed methods have not yet been applied to the task of face recognition, at least not on large databases. This may be for several reasons; some methods may need many sample images per person, pixel-wise accurate alignment of images, or high-quality images for reconstruction; or they may be computationally too expensive to apply to recognition tasks that process thousands of images in near-real- time.

Machine recognition of faces has emerged as an active research area spanning disci-

plines such as image processing, pattern recognition, computer vision, and neural networks. There are numerous applications of FRT to commercial systems such as face verification-based ATM and access control, as well as law enforcement applications to video surveillance, etc. Due to its user-friendly nature, face recognition will remain a powerful tool in spite of the existence of very reliable methods of biometric personal identification such as fingerprint analysis and iris scans.

The accuracy and efficiency of the project can be improved by providing large data sets and re-enforced learning by machine. The advantage of this project is that it makes use of the concept of BYOD (Bring Your Own Device). External hardware installations are not required. Instead of maintaining a register for attendance and manually marking attendance, this application automates the attendance process which saves a lot of time and effort.

2.2.3 OpenCV for Face Detection [1]

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision, developed by Intel Russia research center in Nizhny Novgorod, and now supported by Willow Garage and Itseez. It is free for use under the open source BSD license. The library is cross-platform. It focuses mainly on real-time image processing. If the library finds Intel's Integrated Performance Primitives on the system, it will use these proprietary optimized routines to accelerate itself.

OpenCV is released under a BSD license and hence its free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 7 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

Chapter 3

System Architecture

3.1 Analysis of Engineering Problem

By applying the Face Recognition Techniques (FRT), this cloud based people recognition system will be developed which reduces the load on the client side while running the algorithms on the server. The first step is to capture and store images locally using a camera. Then upload these images to cloud using internet where image processing and face recognition happens on servers.

A general statement of the problem of machine recognition of faces can be formulated as follows: given still images of a scene, identify or verify one or more persons in the scene using a stored database of faces.

The solution to the problem involves segmentation of faces (face detection) from cluttered scenes, feature extraction from the face regions, recognition, or verification.

In identification problems, the input to the system is an unknown face, and the system reports back the determined identity from a database of known individuals, whereas in verification problems, the system needs to confirm or reject the claimed identity of the input face.

The final step is to display the results of recognition. This is our ultimate goal of face recognition. The accuracy of results is determined by light illumination, data sets and poses variation.

3.2 Details of Each Client Application

There are three main components in this project

3.2.1 Client Application

Client machine can be a computer system or a smart phone. The desktop application is developed using java swing and the mobile application using android. The two main functions of client application are to capture images of users and upload them to the server for new enrolment or recognition respectively.

3.2.2 Server (Cloud)

On the server side there are programs having algorithms for facial detection/ recognition, to train the images sent during enrolment for the first time and recognize the images sent during recognition against dictionary of images for a particular user for marking attendance later. All of these vital data is stored in appropriate tables in the database as objects. The OpenCV library is used in server programs for image processing. The training or enrolment is a one-time process done initially to create a database with known faces. These images will be matched during recognition to verify if they match, to mark their attendance. The php programs on server are the ones that populate and query the database.

3.2.3 Web interface

Thirdly, the web interface is developed using django which is a python framework. It has the feature of generating attendance, reports, manual tagging and manual un-tagging, displaying map showing location of users (if it is uploaded using mobile device with GPS on). The web is populated by querying the database tables using python. The web interface is where we will be able to see the results of recognition and attendance for different sessions, people, days, enrolled images and recognition images.

3.3 Processes

There are two main processes in this project namely

3.3.1 Training

Training, as the name insists, is the learning phase of our Face Recognition system. It is a one-time process for one user.

3.3.2 Recognition

Recognition is a daily process. Like Training, this also has to be initiated from the Client.

3.4 Tools

3.4.1 Programming languages used

- C++
- PHP
- Java
- Python
- HTML

3.4.2 Frameworks used

- Eclipse IDE
- Django
- Hibernate
- RESTEasy
- Phonegap
- Cordova

Chapter 4

Explanation of Engineering Problem

4.1 Design

Design is much more creative process than analysis. Design is the first step in the development of any system or product. Design can be defined as the process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization.

It involves four major steps. They are:

(1) Understanding how the system is working now.

(2) Finding out what the system does now.

(3) Understanding what the new system will do.

(4) Understanding how the new system will work.

System Design also called top-level design aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided. In system design the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for each of the modules.

Once the system has been designed, the next step is to covert the designed one in to actual code, so as to satisfy the user requirements as expected. If the system is approved

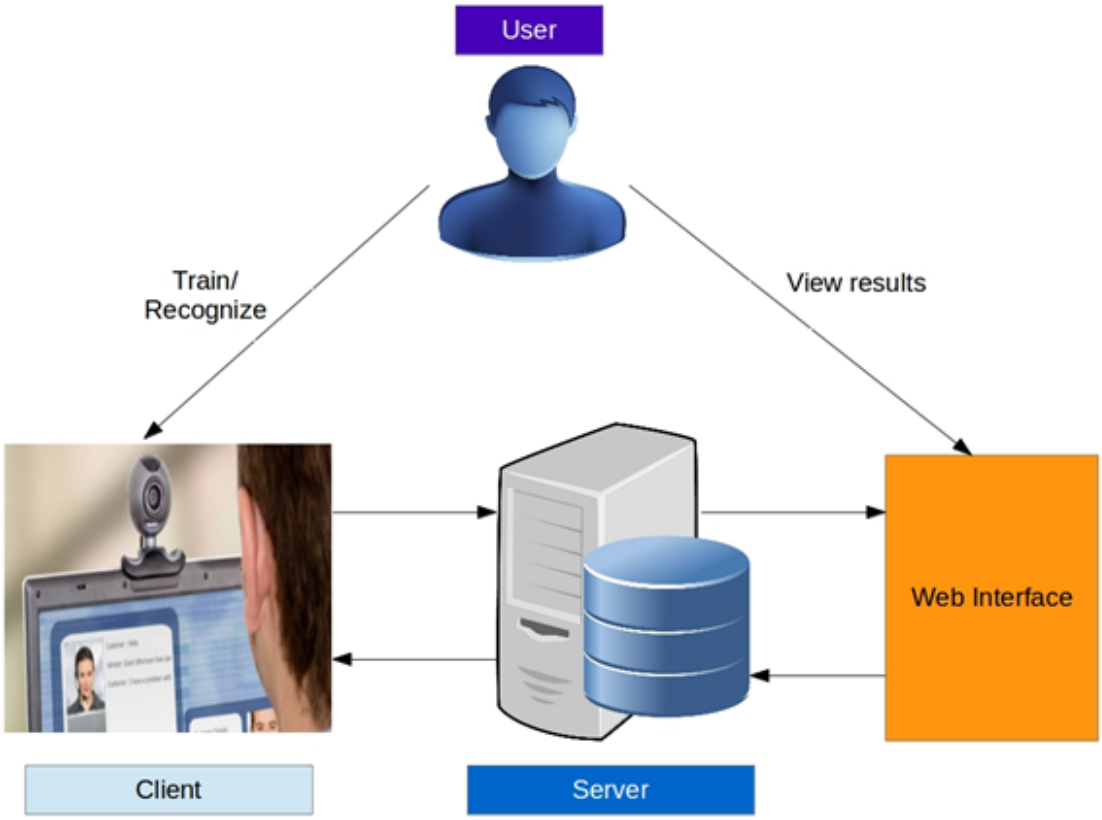

Figure 4.1: System Design.

to be error free it can be implemented.

When the initial design was done for the system, the department was consulted for acceptance of the design so that the further proceedings of the system development can be carried on. After the development of the system a demonstration was given to them about working of the system. The aim of the system illustration was to identify any malfunctioning of the system.

The Web interface is developed using Django a python framework.

The templates which contain html pages are rendered by django.

We make use of bootstrap a front end framework, which helps build complex websites and web applications.

Bootstrap is a free collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

Bootstrap is compatible with the latest versions of all major browsers.

Bootstrap provides a set of style sheets that provide basic style definitions for all key HTML components. These provide a uniform, modern appearance for formatting text, tables and form elements.

In addition to the regular HTML elements, Bootstrap contains other commonly used interface elements. These include buttons with advanced features (e.g. grouping of buttons or buttons with drop-down option, make and navigation lists, horizontal and vertical tabs, navigation, breadcrumb navigation, pagination, etc.), labels, advanced typographic capabilities, thumbnails, warning messages and a progress bar. Bootstrap comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. They also extend the

functionality of some existing interface elements, including for example an auto-complete function for input fields.

To use Bootstrap in an HTML page, the developer downloads the Bootstrap CSS stylesheet and includes a link in the HTML file.

PhoneGap is the framework used along with ADT to build android (mobile) version of client application.

PhoneGap is a mobile development framework. It enables software programmers to build applications for mobile devices using JavaScript, HTML5, and CSS3, instead of device-specific languages such as Objective-C. The resulting applications are hybrid, meaning that they are neither truly native (because all layout rendering is done via web views instead of the platform's native UI framework) nor purely web-based (because they are not just web apps, but are packaged as apps for distribution and have access to native device APIs). The software underlying PhoneGap is Apache Cordova.

Hibernate ORM (Hibernate in short) is an object-relational mapping framework for the Java language, providing a framework for mapping an object-oriented domain model to a traditional relational database. Hibernate solves object-relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling functions.

Hibernate is free software that is distributed under the GNU Lesser General Public License.

Hibernate's primary feature is mapping from Java classes to database tables (and from Java data types to SQL data types). Hibernate also provides data query and retrieval facilities. It generates SQL calls and relieves the developer from manual result set handling and object conversion. Applications using Hibernate are portable to supported SQL databases with little performance overhead.[5]

RESTEasy is a JBoss project that provides various frameworks to help you build

RESTful Web Services and RESTful Java applications. It is a fully certified and portable implementation of the JAX-RS specification. JAX-RS is a new JCP specification that provides a Java API for RESTful Web Services over the HTTP protocol.[6]

4.2 Client

"Client" is the component of the system which is installed on the machines (can be computer systems or smart phones) of the customers who are also the users of our product. Desktop client app is developed in Java and Mobile client app is developed in android. As mentioned earlier, this is the part of our product which helps the users to enroll themselves and send recognition images for daily attendance. As soon as the app is started, the client has to enter License Key:

4.2.1 Train

This is a one-time process for every user. The end to end process of Enrolment has already been expounded in the Training process of the previous section. Ideally 50 images are required for a single user to train the system. When the images are uploaded from client application, they call a php url to fetch the image on server and store it at appropriate location. Once image is uploaded successfully then an ACK is sent to the client based on which the client triggers Enrollment Algorithm on the server.

4.2.2 Recognize

As mentioned earlier, it is a daily activity and the full flow is explained in the Recognition section. This is more like training process but with a few variations. In this case the number of images captured may vary from 1-5 only unlike Train.

4.2.3 Settings

Under settings multiple operations are possible:

- Reconfigure
- Location On/Off
- Auto Camera On/Off
- Auto Upload On/Off

- Change Action Type (i.e. Enroll/Attendance)
- Change Camera (i.e. Front/Rear)

4.3 Server

Server is the backbone for all the processes and activities that are done. In fact, it is the critical component of the system. It holds the incoming data from each and every client in the system. All the images reside on server. Database also sits on the server.

4.3.1 Dictionaries

Server maintains a dictionary of images that are used for verification.

4.3.2 Result File

Result files are the output of the whole process that stores the appropriate value after doing face detection/recognition.

4.4 Web Interface

Web interface is nothing but an UI which presents data to the users in varying granular levels and it has been built using Django. Indeed, Django is a free and open source web application framework, written in Python, which follows the model view controller architectural pattern.

4.4.1 Main Job

The main job of this web interface is storing the data in the database during training and recognition. Then, fetching them back from the appropriate tables in the database and presenting them to the user. This job is accomplished with the help of definitions and templates in Django.

4.4.2 Definitions

Since Django works on "Model View Controller Architectural Pattern", we have database tables associated with each model defined. The main advantage of using Django is, we do not directly query the database every time. Creating an object for the model, will automatically create respective entries in the database. Similarly while fetching the data, we do not use complex database querying. Instead, python statements will be

automatically translated as the respective query. This makes the job of a programmer much easier.

4.4.3 Templates

Templates are nothing but HTML pages presenting the data sent by the definitions to the users. Which HTML page has to be displayed is determined by the definitions only. In a nut shell, appearance, look and feel is role of these templates.

4.5 Database

Since Django works on "Model View Controller Architectural Pattern", we have database tables associated with each model defined. The main advantage of using Django is, we do not directly query the database every time. Creating an object for the model, will automatically create respective entries in the database. Similarly while fetching the data, we do not use complex database querying. Instead, python statements will be automatically translated as the respective query. This makes the job of a programmer much easier.

4.6 Activities

As mentioned earlier, we classify the activities into two, as the users do not have any direct interaction with the server. They can perform the following activities on the Client and Web Interface.

4.6.1 Activities on the client

- Enter License Key
- Enrolment
- Recognition
- Reconfigure

4.6.2 Activities on the Web Interface

- Login and Logout
- Select Organization

- Select View
- Date Navigation
- Generate Reports
- Self/Employee mode
- View Attendance
- Change Enrollment Image
- Manual Untagging
- Manual Tagging
- Review
- Accept Request
- Reject Request
- Assign Manager

Chapter 5

Proposed Architecture

5.1 Design Challenge

Since it is purely B2B solution, it becomes necessary to handle each and every scenario that user expects in the market. So we have come across three different design challenges that needs to be implemented in order to make the Smart Attendance solution robust and scalable.

Following are the three design challenges:

(1) Using our Client component as well as Web Interface.

(2) Using our Client component and Customer's Web Interface.

(3) Using Client's own component for applications and Web Interface.

5.1.1 Scenario 1

In this scenario customer will make use of the client component designed and developed by Aindra Systems i.e. Java based desktop application and android based mobile application. For accessing the attendance result also customer will use Web Interface developed by Aindra Systems.

5.1.2 Scenario 2

In this scenario customer will make use of the client component designed and developed by Aindra Systems i.e. Java based desktop application and android based mobile application. However for accessing the attendance result client will use their own management

system or website to display results to the users.

To handle this scenario Aindra Systems need to develop public APIs with use of which customer can make HTTP calls to the APIs and fetch the desired results.

5.1.3 Scenario 3

In this scenario customer would want to use the application developed by their own organization and would also want the end results to be displayed on their own Web portal. In this scenario customer will only buy the product as a service and not the whole system.

To handle this scenario Aindra Systems need to develop public APIs with use of which customer can do all the actions that are performed by Aindra Systems' own client application i.e. Enrollment and Recognition.

5.2 Proposed Design

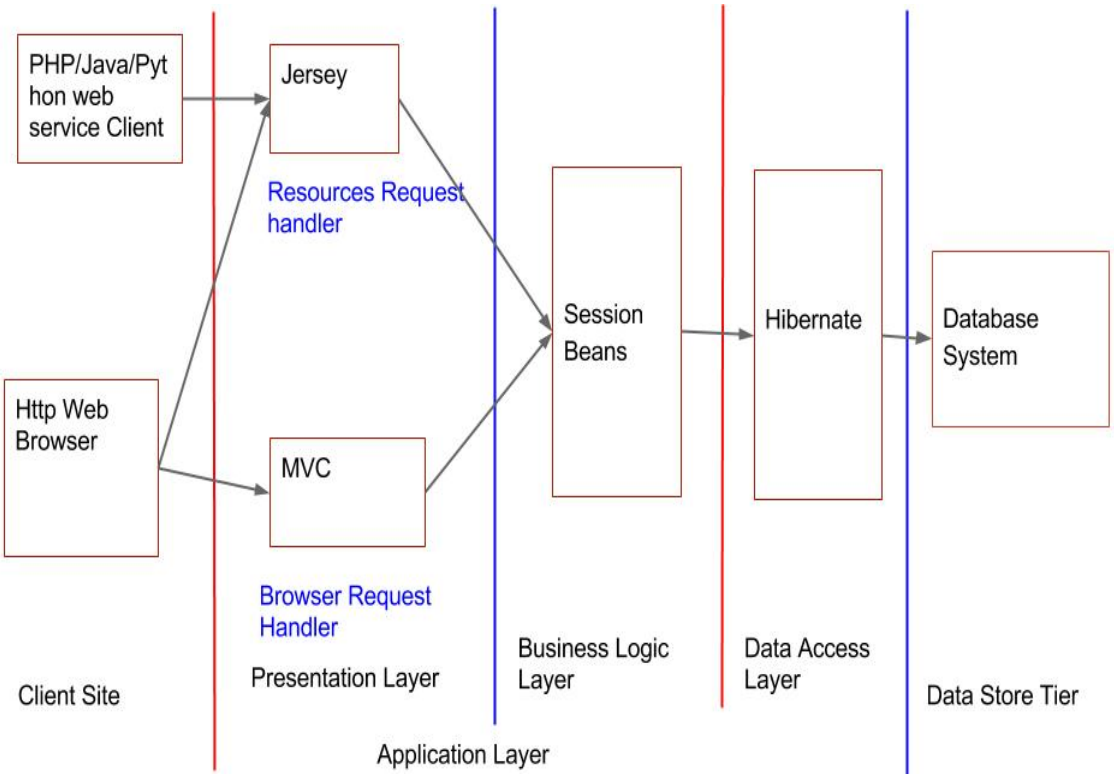

Figure 5.1: Hibernate Architecture.[5]

Even now the main system design remains the same. Only the implementation has

changed based on market requirement. Still the client component is thin and the main job of it is to capture images, store images locally and then push the images onto the server and deleting them locally. Server will process those images and generate appropriate results. But these actions can now be performed by making API calls.

Aindra Systems is developing Java based APIs using Hibernate Framework. These APIs are mapped to some URLs using RESTEasy Framework. The deployment of these APIs is handled by Jboss Enterprice Application Platform.

5.2.1 Programming Languages and Tools used

- J2EE
- XML I/O file format for APIs
- JSON I/O file format for Algorithms
- Jboss EAP 6.3
- RESTEasy Framework
- Hibernate Framework
- Eclipse(IDE)

5.3 Basic API call flow

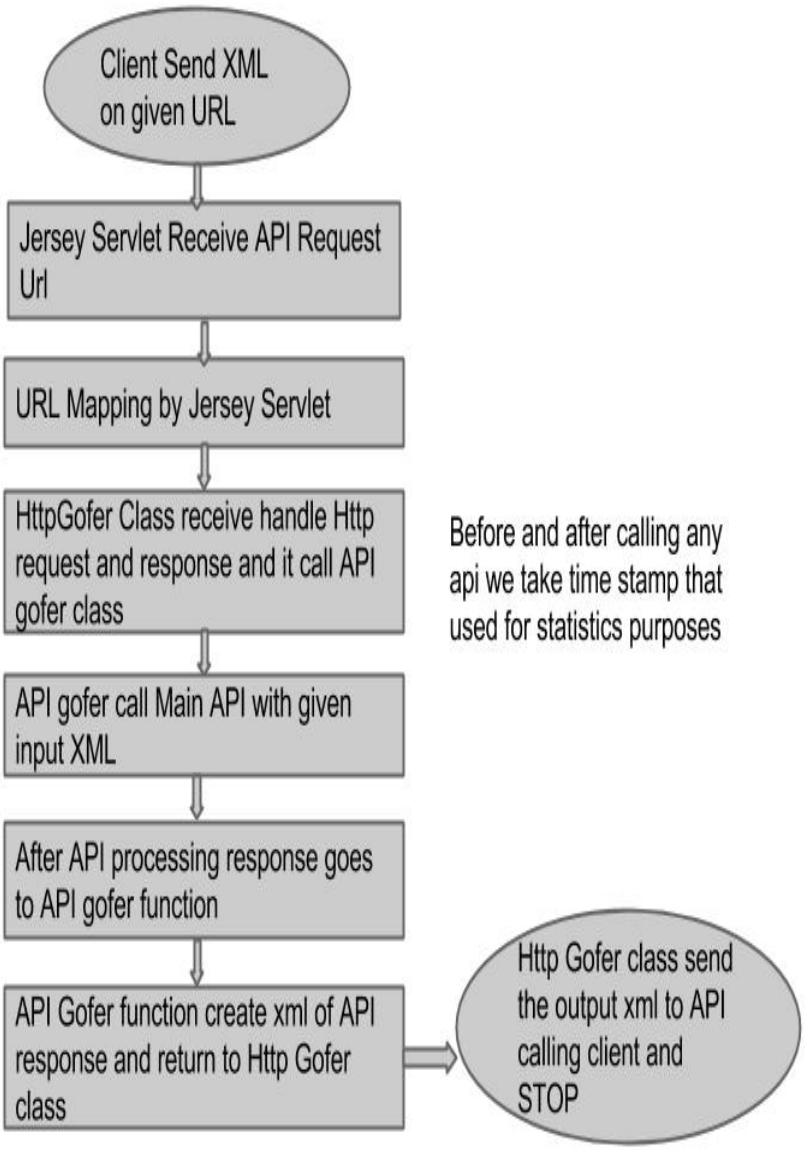

Figure 5.2: API Calling Flow.

5.4 Logging

Logging is an important module of any project. It helps to debug your source code and analyze the problem details. It can help you to keep record of your code usage. We have implemented log4j Framework in the java APIs.[2]

Apache log4j is a Java-based logging utility. The following table defines the log levels and messages in log4j, in decreasing order of severity. The left column lists the log level designation in log4j and the right column provides a brief description of each log level.[2]

Level	Description
OFF	The highest possible rank and is intended to turn off logging.
FATAL	Severe errors that cause premature termination.
ERROR	Other runtime errors or unexpected conditions.
WARN	Use of deprecated APIs, other runtime situations that are not necessarily "wrong".
INFO	Interesting runtime events (startup/shutdown).
DEBUG	Detailed information on the flow through the system.
TRACE	Most detailed information.

Table 5.1: Log4j Log Level.[2]

Chapter 6

Application Usage Flow

6.1 Overview

As mentioned earlier the client component is responsible for capturing images, storing images locally, pushing images on server and then deleting them form local client. This the functional implementation of client component. Now lets look at the application flow from user's perspective.

Smart Attendance by AIndra Systems is a B2B solution. Hence first step is to purchase license key from Aindra Systems. Organization officials can make this deal. Once licence is purchased, Aindra Systems would generate an organizational level License Key attached with the purchasing organization.

Following are the steps involved in using the Smart Attendance application:

6.2 Registering Organization

First step is to register your organization using Aindra Systems' Web Interface. Go to the Web Interface (www.aindra.in/attendance) and signup for your organization by giving the License Key that was generated while purchasing the system license.

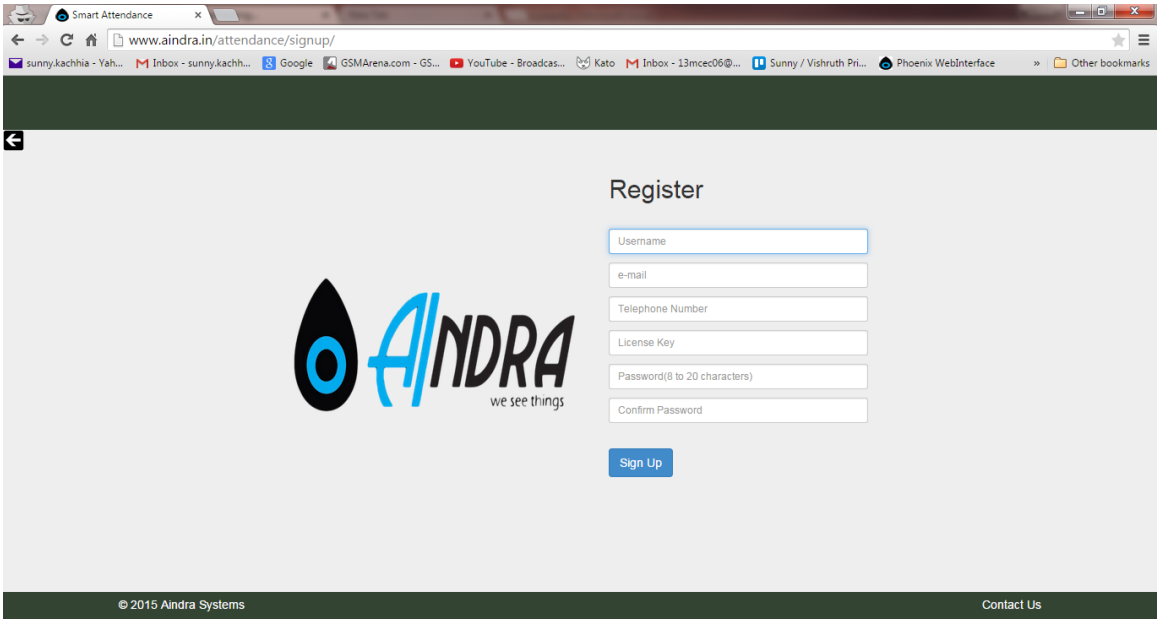

Figure 6.1: Sign Up page.

6.3 System Provisioning

Next step is to provision your system. Before you start using the client application it is necessary to provision your system. IT admin of the organization is likely to do the provisioning of the system. Admin has to login through the crdentials generated while signup process on Aindra Systems' Web Interface.

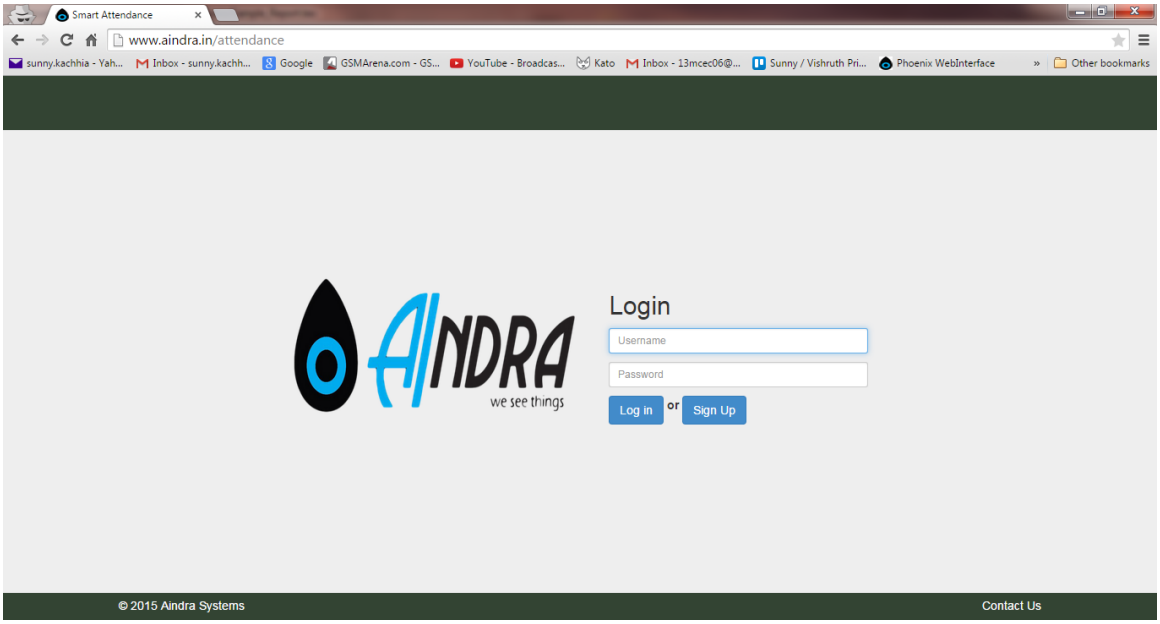

Figure 6.2: Login page.

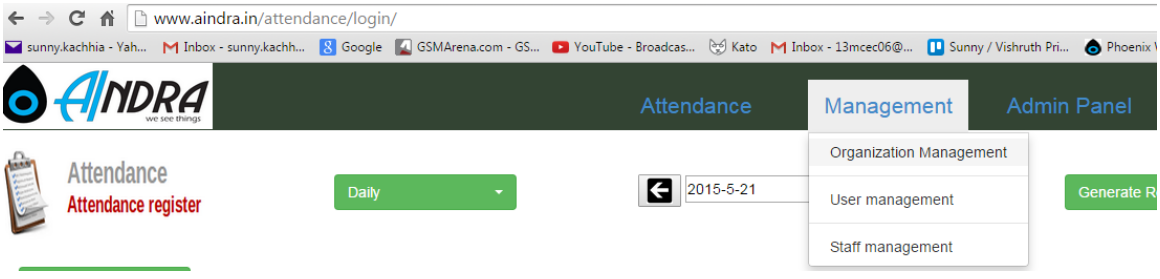

Figure 6.3: Menu bar options.

6.3.1 Organization Management

After login the Admin has to go to the Organization/Institute Management Tab on the top of the screen. This will lead to next page where actual provisioning has to be done. On this page Admin will add location/course and branch/batch that corresponds to their organization/institution.

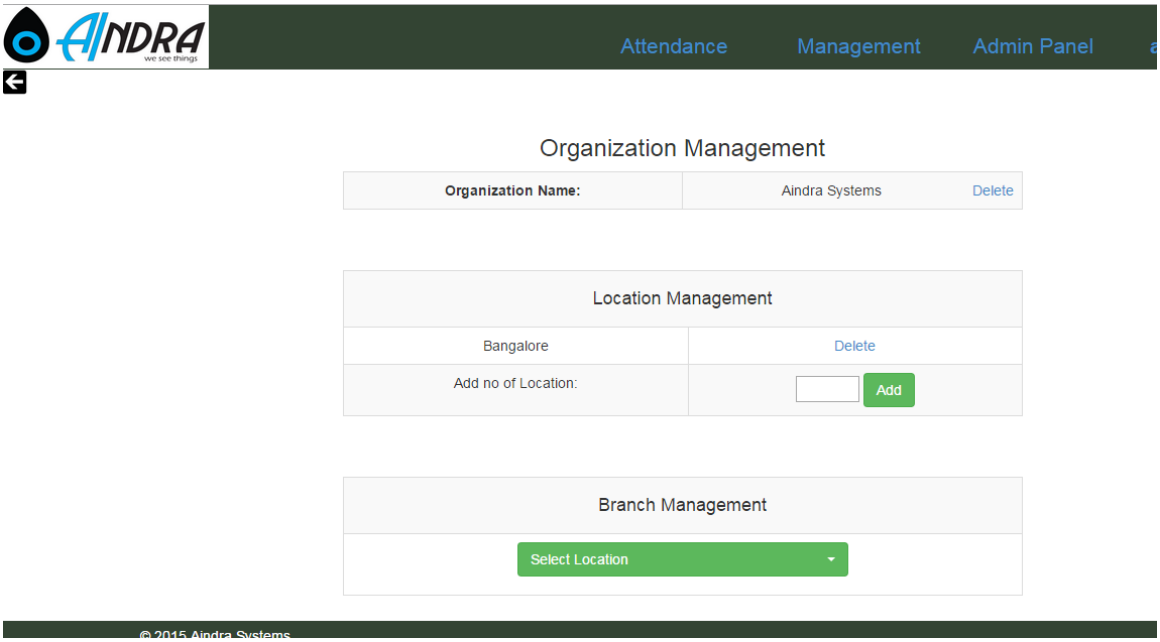

Figure 6.4: Organization Management Page.

6.3.2 User Management

Once this is done Admin has to add the members in the system. Members can be end users i.e. employee/student or managers/teachers who will track the attendance of end users. This can be done under User management section.

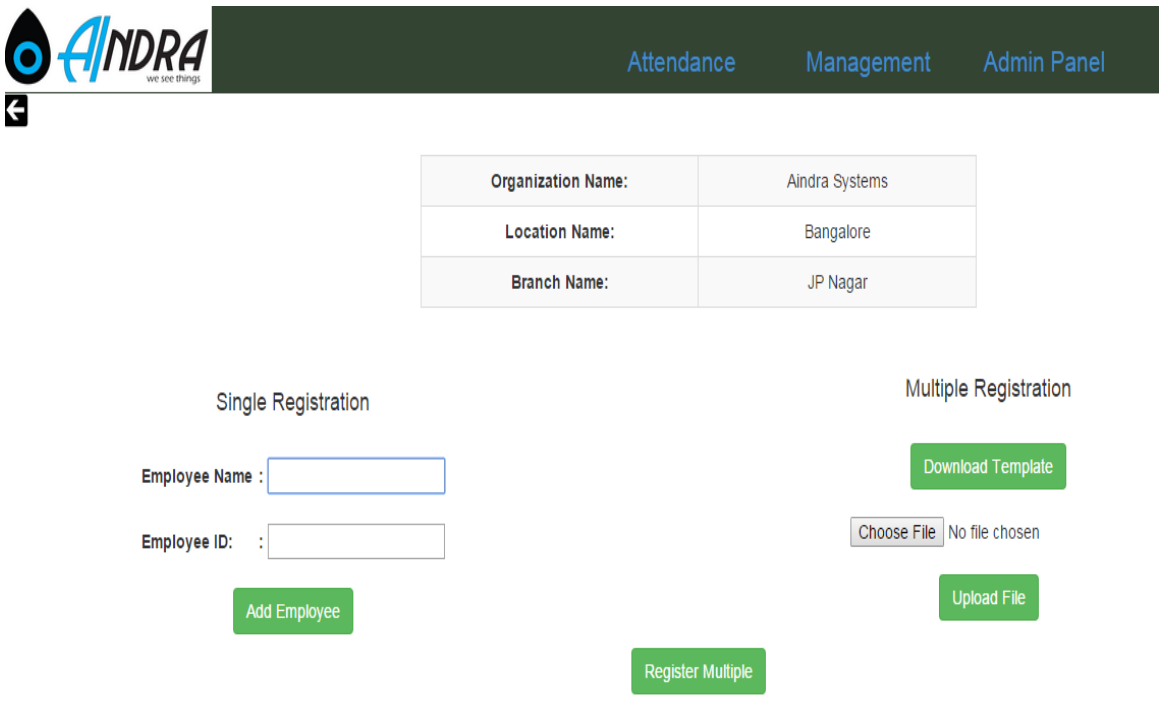

Figure 6.5: User Management Page.

While provisioning the system a JSON file is created for each organization. This file contains all the organization related details including number of locations/courses, number of branches/batched and number user in the organization. This JSON file is used on the client machine later.

This is the basic requirement to configure any system.

6.4 Client Application

Currently we two client components i.e. java based desktop application and android based mobile application.

Both of the applications are designed in such a way that all the requirements of Aindra Systems are met by them. Using the application is very simple and handy:

6.4.1 License Key Verification

Once the application is installed in your device (desktop/mobile), first step is to give your organization specific license key. Only if license key is valid then the application gets unlocked and then user can start using it.

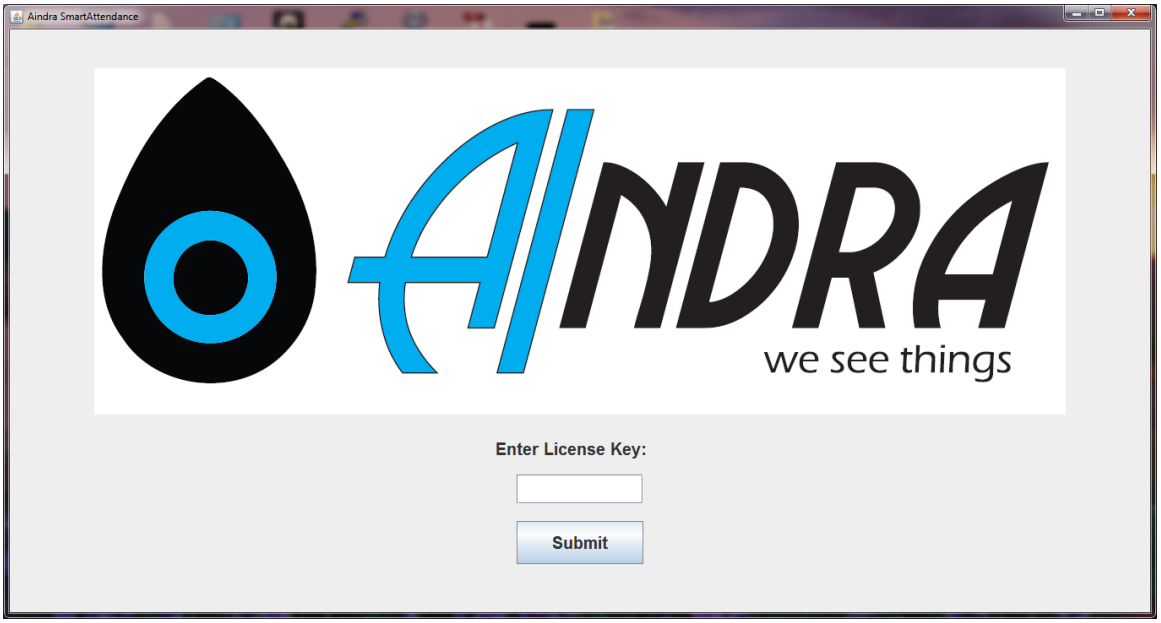

Figure 6.6: License Key Verification in Desktop Application.

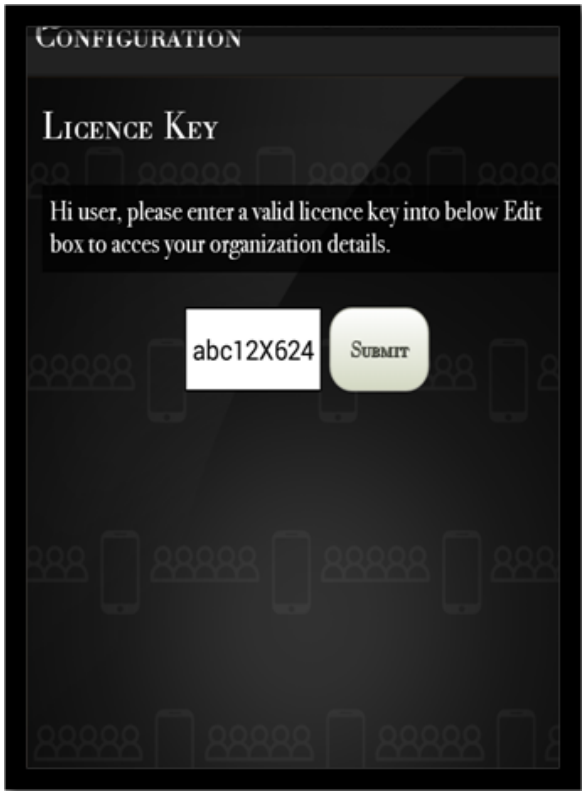

Figure 6.7: License Key Verification in Mobile Application.

6.4.2 Configuring application

After valid license key verification user has to configure the application. This process involves selection of location/course and branch/batch to which the user belongs who has installed the application. Remember this all was already provisioned by Admin using Aindra Systems' Web Interface.

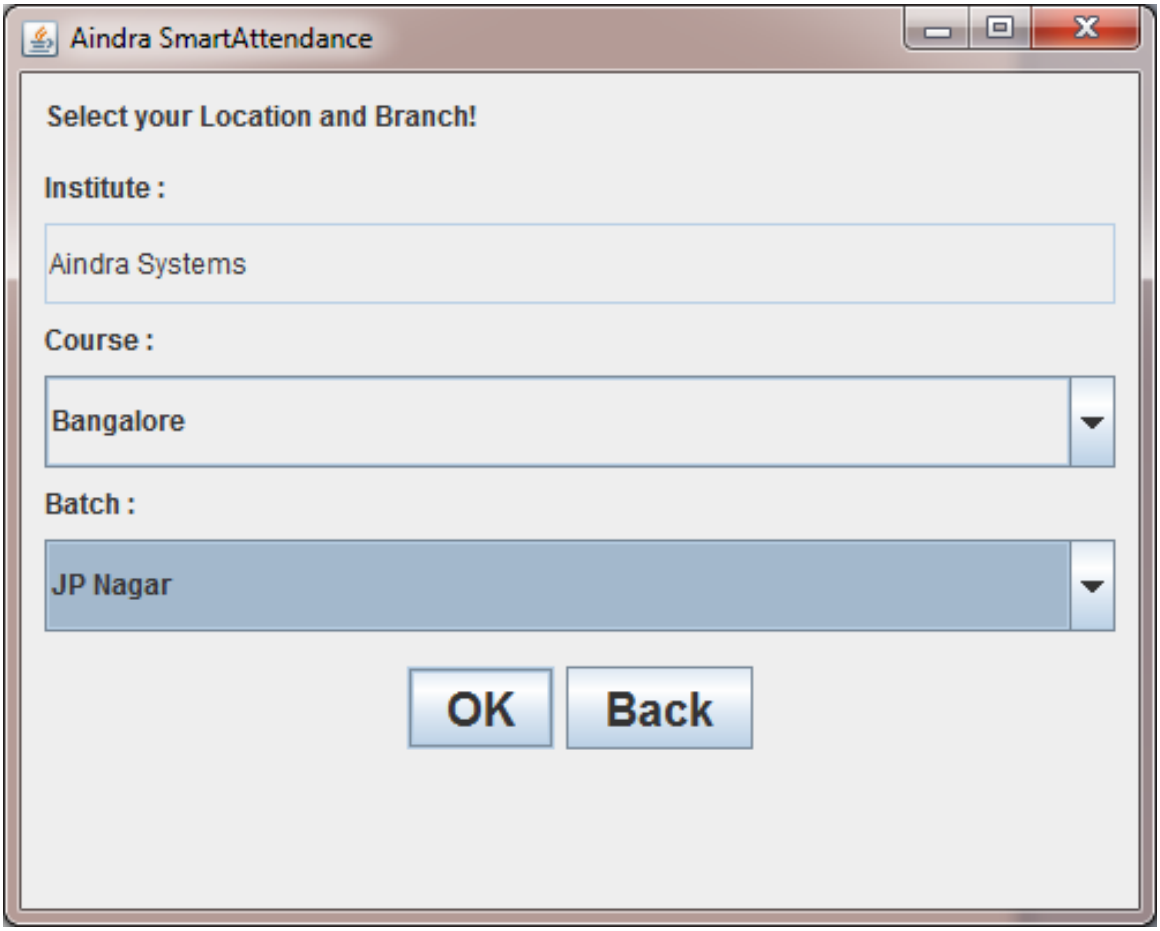

Figure 6.8: Configuring Desktop Application.

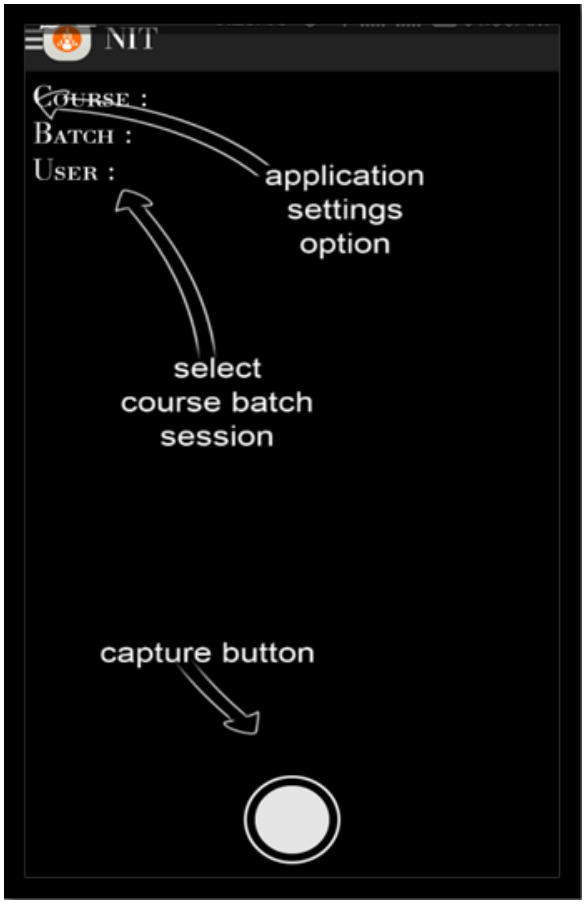

Figure 6.9: Configuring Mobile Application.

6.4.3 Training

Now the user can select Training type and capture his/her images. Once images are captured then user can upload those images according to their convenience whenever internet service is available. We do capture time of the image captured hence no need to worry about internet availability for uploading images as an when they are captured.

This is a one-time process for each user.

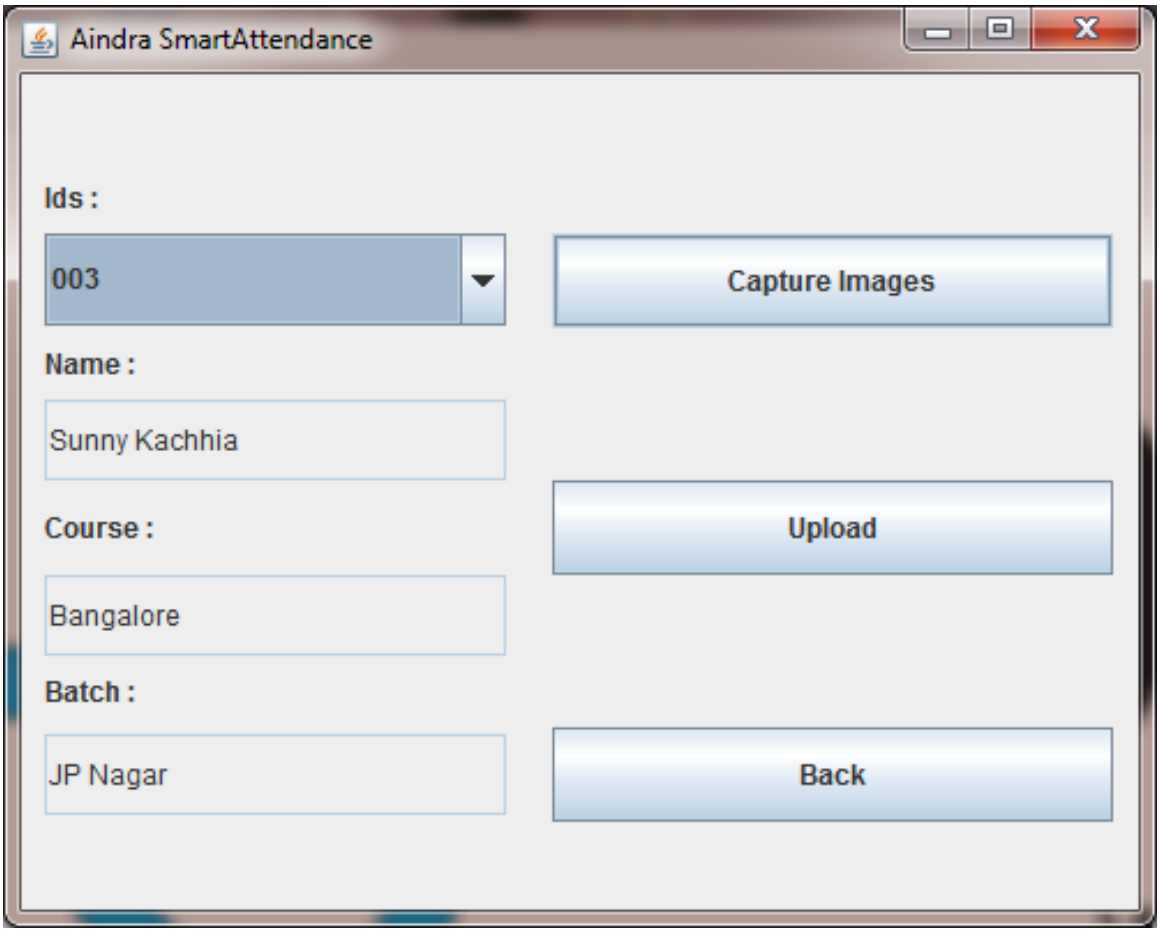

Figure 6.10: Train page in desktop application.

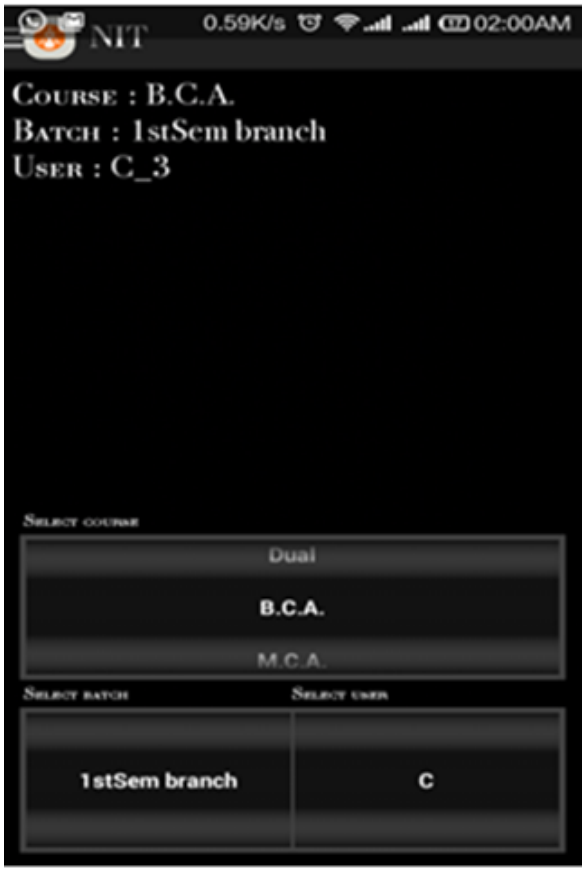

Figure 6.11: Train page in desktop application.

6.4.4 Recognition

After Training is done for the user, he/she can start sending their attendance. User has to select Attendance type in the application and then again capture images and upload it to server based on internet availability.

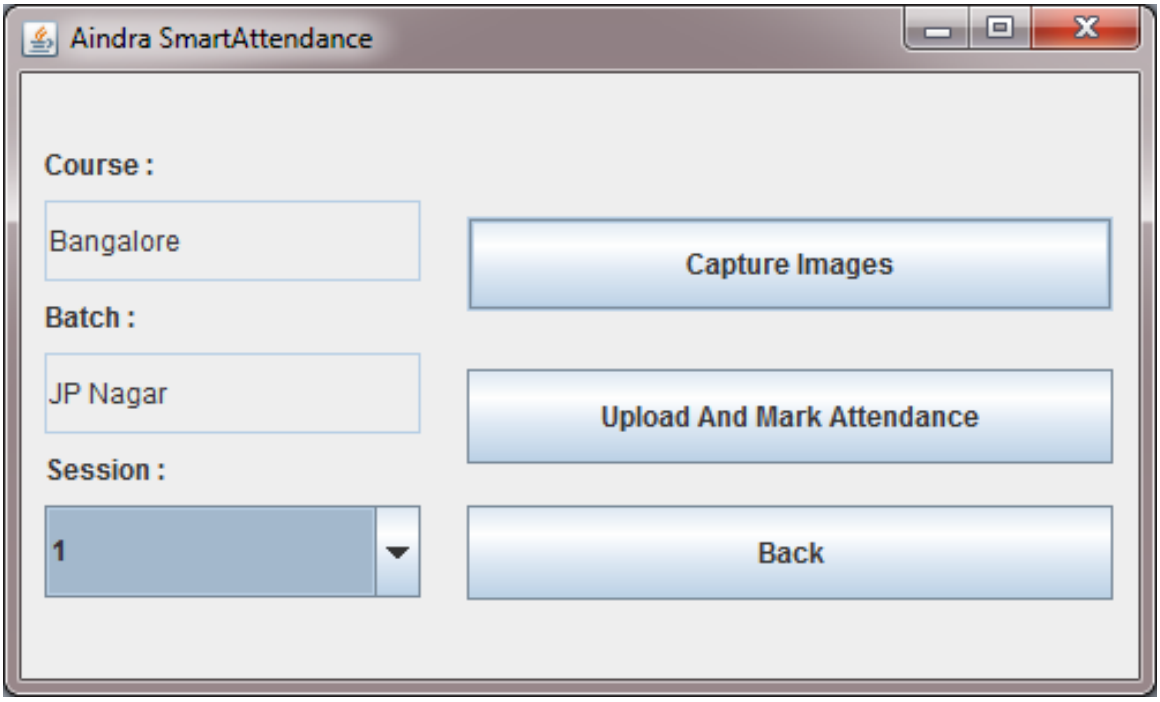

Figure 6.12: Attendance page in desktop application.

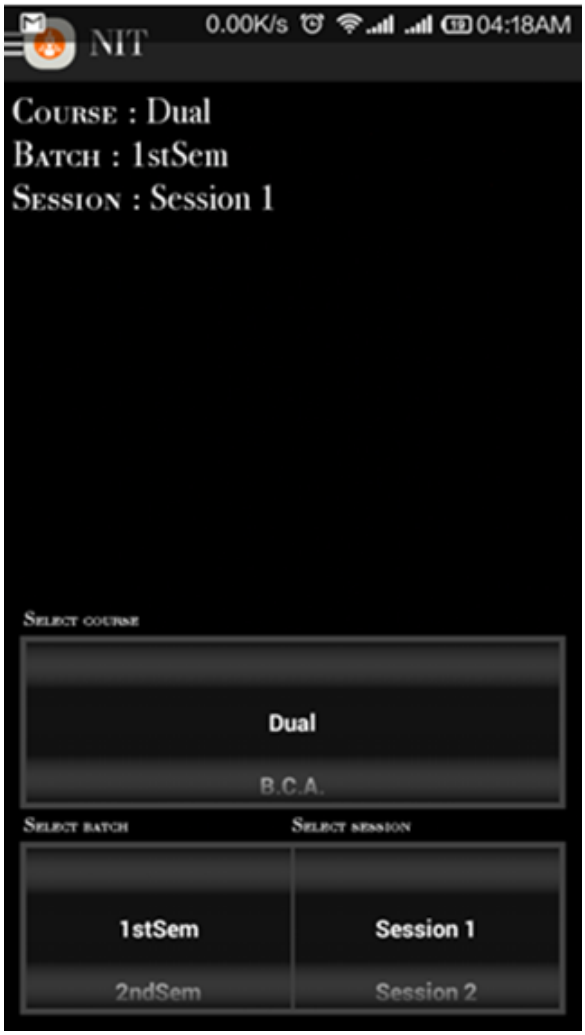

Figure 6.13: Attendance page in mobile application.

6.5 Results on WebInterface

User can now anytime login to the Web Interface and verify his/her attendance record. For teachers/managers it is the same but for them they can see the attendance of all the students/employees under them. User has to login to the Web Interface (www.aindra.in/attendance) using their login credentials. User can also generate reports of their attendance through Web Interface.

Location	Branch	Name	ID	Session 1
Bangalore	JP Nagar	Sunny Kachhia	003	

Figure 6.14: Attendance Result.

This covers an entire end to end flow of using Smart Attendance application. On top of this user has access to misc features in their client application as well as Web Interface.

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

Thus an intuitive Face Recognition System has been developed which makes use of the Face Recognition Techniques (FRT) to train (learn) itself and uses it to recognize people from the dictionary it created during learning process. Once Recognition is done, the result is forwarded to the web interface where managers and employees can view the consolidated attendance in various levels of granularity. When a person is wrongly marked absent, with the help of manual tagging, reinforced learning is also made possible so that the machine can recognize that person properly in the future.

7.2 Future Scope

Though marking attendance for a group of people in a single image is working, its scalability is still a challenge. The illumination conditions and face postures are important factors impacting the results. The efficiency can be increased by improving the core algorithm which takes care of the recognition process for a group of people.

References

[1] "OpenCV description." http://en.wikipedia.org/wiki/Hibernate_(Java).

[2] C. Glc, "Log4j description." http://en.wikipedia.org/wiki/Log4j.

[3] J. Beveridge, K. She, B. Draper, and G. Givens, *A nonparametric statistical comparison of principal component and linear discriminant subspaces for face recognition*, vol. 1. 2001. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

[4] V. Blanz, S. Romdhami, and T. Vetter, *Face identification across different poses and illuminations with a 3D morphable model*. 2002. Proceedings of International Conference on Automatic Face and Gesture Recognition.

[5] "Hibernate orm." http://en.wikipedia.org/wiki/Hibernate_(Java).

[6] "RESTEasy description." http://resteasy.jboss.org/.