**CS341 - DESIGN PROJECT REPORT**

**ON**

**DIGITAL NOTICE BOARD AND NOTIFICATION APP**

***Submitted By***

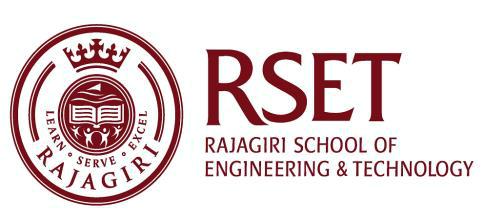
AKSHAY GEORGE (20),

AKSHAY KISHORE (21),

ASHEL AUGUSTINE (44),

ASWIN VIJAYAKUMAR (49)

**S5 CSα**



Department of Computer Science & Engineering

RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

Affiliated to APJ Abdul Kalam Technological University

November, 2017

**Rajagiri School of Engineering & Technology**

RAJAGIRI VALLEY, KAKANAD, COCHIN – 682 039



**Certificate**

*Certified that this is a Bonafide Record of the work done by……….…………..……in the* DESIGN PROJECT *during the FIFTH semester in year 2017 at Rajagiri School of Engineering & Technology, Kakkanad, Kochi.*

Ms. Shimmi Asokan

Project-In-Charge Head of the Department

Uni. Reg No. :

Branch : COMPUTER SCIENCE & ENGINEERING

Semester : S5

**ACKNOWLEDGEMENT**

It brings us great pleasure to be associated with this Design Project. The project enabled us to work together as a team and was indeed a wonderful learning experience. The project enabled us to learn more about the scope of wireless communication and how tedious tasks can be simplified with the help of technologies like a mobile application. The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of this project.

First of all, we respect and thank the Management of Rajagiri School of Engineering and Technology, Principal Dr. A Unnikrishnan, Head of Department CS RSET, Ms Shimmi Ashokan for providing us the opportunity to do the project.

We owe our deep gratitude to our project guides Prof.K S Mathew, Mr. Paul Augustine and Mr. Sivaram M who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good project.

We would also like to thank all the respondents whose responses and suggestions were of utmost importance for the successful completion of our project.

**ABSTRACT**

A notice board is a primary object in any institution or organisation or public utility may it be bus stations, railway stations, government institutions or parks. But sticking various notices day-to-day is a difficult process. A separate person is required to take care of how these notices go into the notice boards.

This project deals with an advanced hi-tech wireless notice board. The project is built around an LCD display linked to a server and a notification application that can be installed on mobile phones for students. The digital notice boards are placed in each and every classroom. The students can also easily access the notices from the app through their mobile phones even if they are not in the campus. The notices can only be altered by teachers or other authorised office staff. This is enables by the registration procedure on the related website. Notification application is linked to the display via a server. Any notice put in the board will also appear in the app. The students can login to the application and access details like exam seating arrangements, assignment details and other notifications with ease.

**Table of Contents**

1. Study Phase - Introduction 12
2. Design Considerations 13
3. External Search 15
4. How does it work? 18
5. Design Drawings/Models 19
6. Future Enhancements 21
7. Study Phase - Conclusion 22

Skin-status Evaluation: Evaluate skin health status of detected faces, and recognize common skin problems, such as acne, dark circles, and spots on face. Face++ provides high-precision confidence scores for you to measure and assess facial skin status. 22

1. Design Phase - Introduction 23

8.1 Problem Statement 23

8.2 Project Objectives 23

8.3 Design and Implementation Constraints 24

1. Literature Survey 24

10.System Overview and Requirement Specification 24

10.1 System overview 24

There were four main solutions under consideration when brainstorming was conducted to find a solution to the problem to find an alternative to the existing typical wooden notice board and for the issue of misplacing and the hectic work included in organising the notices and the wastage of paper. 24

The solutions were:: 24

* Email notification . 24
* Digital Displays on ID cards 24
* Digital Noticeboards. 24
* Notification app. 24

On carefully assessing the merits and demurest of each solution, we found that: 24

* Loads of emails regarding the notices create a tedious process to check the mail on a daily basis and, most important notices can go unnoticed. 24
* The implementation of ID cars with digital display feature was estimated to be too expensive and, once a student is graduated, the card becomes of no use. 24
* We concluded that both implementation of Digital noticeboards and the notifier app seemed to be fitting enough to solve the existing problem effectively and since both these solutions required a single database in common, we decided to club it into a single and more effective solution to the problem in hand. 24

10.2 Problem Domain Model 25

10.3 Functional and Non-Functional Requirements (Requirement Model) 25

10.4 List of Actors 27

10.5 List of Use Cases 27

10.6 Use Case Diagram 27

10.7 Use Case Specification 29

10.7.1 Login 29

10.7.2 Create Notice 29

10.7.3 Modify Notice 30

10.7.4 Delete Notice 31

10.7.5 View Notice 32

10.7.6 Send Feedback 33

10.7.7 Manage Database 34

11.High Level Design (Architecture Diagram) 38

12.Analysis and Design Models 40

12.1 Sequence Diagram 40

12.2 Class Diagram 43

12.2.1 List of Classes 44

12.2.2 List of Classes with Attributes and Operations 45

13. Data Model 47

14. Limitations 47

15.Conclusion 47

List of Figures

5.1. Comparison: Example………………………………………………………………………………………………………….……18

5.2. Searching: Example……………………………………………………………………………………………………………..……18

5.3. Comparison: Example…………………………………………………………………………………………………………….…19

10.2. UML Component diagram………………………………………………………………………………………………………24

10.6 Use Case Diagram………………………………………………………………………………………………………….………..27

10.8.1 Splash screen………………………………………………………………………………………………………………………..33

10.8.2 Login screen……………….………………………………………………………………………………………………………..33

10.8.3 Menu list screen…………………………………………………………………………………………………………………..33

10.8.4 Class notices screen……………………………………………………………………………………………………………..34

10.8.5 Exam schedules screen…………………………………………………………………………………….…………………..34

10.8.6 Feedback screen…………………………………………………………………….……………………………………………..35

10.8.7 Register screen……………………………………………………………………………………………………………………..35

10.8.8 Notice creation screen………………………………………………………………………………………………..………..36

10.8.9 Notice acceptance screen……………………………………………………………………………………………………..36

11.1 UML Component diagram………………………………………………………………………………………………………..37

11.2 UML Deployment diagram……………………………………………………………………………………………………….38

12.1.1 Login…………………………………………………………………………………………………………………………………….39

12.1.2 Create Notice……………………………………………………………………………………………………………………….40

12.1.3 Modify Notice………………………………………………………………………………………………………………………40

12.1.5 Delete Notice……………………………………………………………………………………………………………………….41

12.1.4 View Notice………………………………………………………………………………………………………………………….41

12.2 Class diagram…………………………………………………………………………………………………………………………..42

13 Data Model…………………………………………………………………………………………………………………………………45

List of Tables

2. Design Considerations…………………………………………………………………………………………………………………11

3.1. Benchmarking………………………………………………………………………………………………………………………….13

3.2. Patent information…………………………………………………………………………………………………………………..13

10.3 Functional and Non-Functional Requirements ………………………………………………………………………25

10.4 List of Actors…………………………………………………………………………………………………………………………..26

10.5 List of Use Cases……………………………………………………………………………………………………………………..26

10.7.1 Login…………………………………………………………………………………………………………………………………….28

10.7.2 Create Notice……………………………………………………………………………………………………………………….29

10.7.3 Modify Notice………………………………………………………………………………………………………………………29

10.7.4 Delete Notice……………………………………………………………………………………………………………………….30

10.7.5 View Notice………………………………………………………………………………………………………………………….30

10.7.6 Send Feedback……………………………………………………………………………………………………………………..31

10.7.7 Manage database…………………………………………………………………………………………………………………32

12.2.1 List of classes………………………………………………………………………………………………………………………..43

12.2.2 List of Classes with Attributes and Operations………………………………………………………………………44

**STUDY PHASE**

# Study Phase - Introduction

Face recognition is a process of identification of human face or faces similar to human face in a video or an image. Sometimes it is also referred as the process of identifying images which are similar to each other. The designed system should be able to recognize a particular person having all the different expressions and also should be proficient in differentiating other person’s face.The face recognition technology has improved over the years but still there are some drawbacks.

Face++ is a new vision platform built by Megvii Inc, providing top Face Detection, Face Recognition, and Face Analysis technology. Face++ was founded in October 2011 built by Megvii Inc which origins from Visionhacker Studio.

Face recognition is not a new technology, it has been created during late 1960’s but unlike any face recognition tech Face++ has its own unique algorithms.

For example, most rudimentary face recognition software analyses five points on the face – eyes, nose, and the corners of the mouth. Face++ analyses a whopping 83 points. The distance between them provides a means of identification .So it will be more accurate and error free.

Face recognition has low accuracy compared to the finger print and iris recognition. Both these methods are way more precise than face recognition and are therefore given preference. Change in facial expressions, aging and other personal factors also add to the difficulty in recognizing faces. Different cameras carry different lenses which again has a great impact on the picture being captured as different lens have different power. So any change in the camera or the lenses adds to the difficulty. A little expression or accessory can change the whole look of a person. This leads to unnecessary dismissal of the same face. This is called face reject.

It is almost certain that the face recognition technologies like FACE++ have come up with algorithms that help make our lives easier as discussed previously. And at the same time, endangers our privacy.

When the flood of facial recognition data from Facebook and Microsoft combines with other biometric data including voice from Siri, and location from GPS etc, it clearly destroys our privacy and we can be tracked anywhere by anyone without our consent.

# Design Considerations

|  |  |  |  |
| --- | --- | --- | --- |
| **Design Considerations / Features** | **How is it addressed in the product/process /technique** | | |
| **Functional features** | | | |
| Face detection | | Face detection is the first step to analysing and processing faces, Face++ also allows you to store metadata of each detected face for future use. It uses Face++ Detect API to detect faces within images, and get back face bounding box and token for each detected face. | |
| Face searching | | Search API returns most similar-looking faces to a target face, from a given collection of faces, along with confidence scores and thresholds to evaluate the similarity. To set up face collection, you need to first detect and store face metadata in FaceSet. Search API is widely used in photo grouping and security monitoring. | |
| Face landmarks | | Use Face++ Detect API to detect faces within images, and get back face bounding box and token for each detected face. You can pass the face token to other APIs for further processing. Detect API also allows you to get back 83-point landmarks and attributes for the top 5 largest detected faces. | |
| Face Comparison | | When the software has arrived at a proper size and orientation for the face, it looks even more closely, seeking to create what is called a “[faceprint](https://www.springer.com/us/book/9780857299314).” Much like a fingerprint record, a faceprint is a set of characteristics that, taken together, uniquely identify one person’s particular face. | |
| **Non Functional features** | | | |
| Accuracy | | | Face++ maintains high accuracy in real world, and is robust against influences including face makeup, improper illumination or head pose.  Face++ searches faces at an extremely low error rate. We provide confidence scores with thresholds to help minimize risks. |
| Speed | | | Face++'s powerful search engine allows you to search in large-scale face database in seconds.  Generates landmarks extremely fast ,without any frame drop. |
| Unlimited Faces | | | No limits to the number of detectable faces in an image. You can easily process large group photos. |
| Safety | | | Face++ is very safe since it is highly accurate and is even used for identity verification. |
|  | | | |

Table.2.1

# External Search

**3.1 Benchmarking**

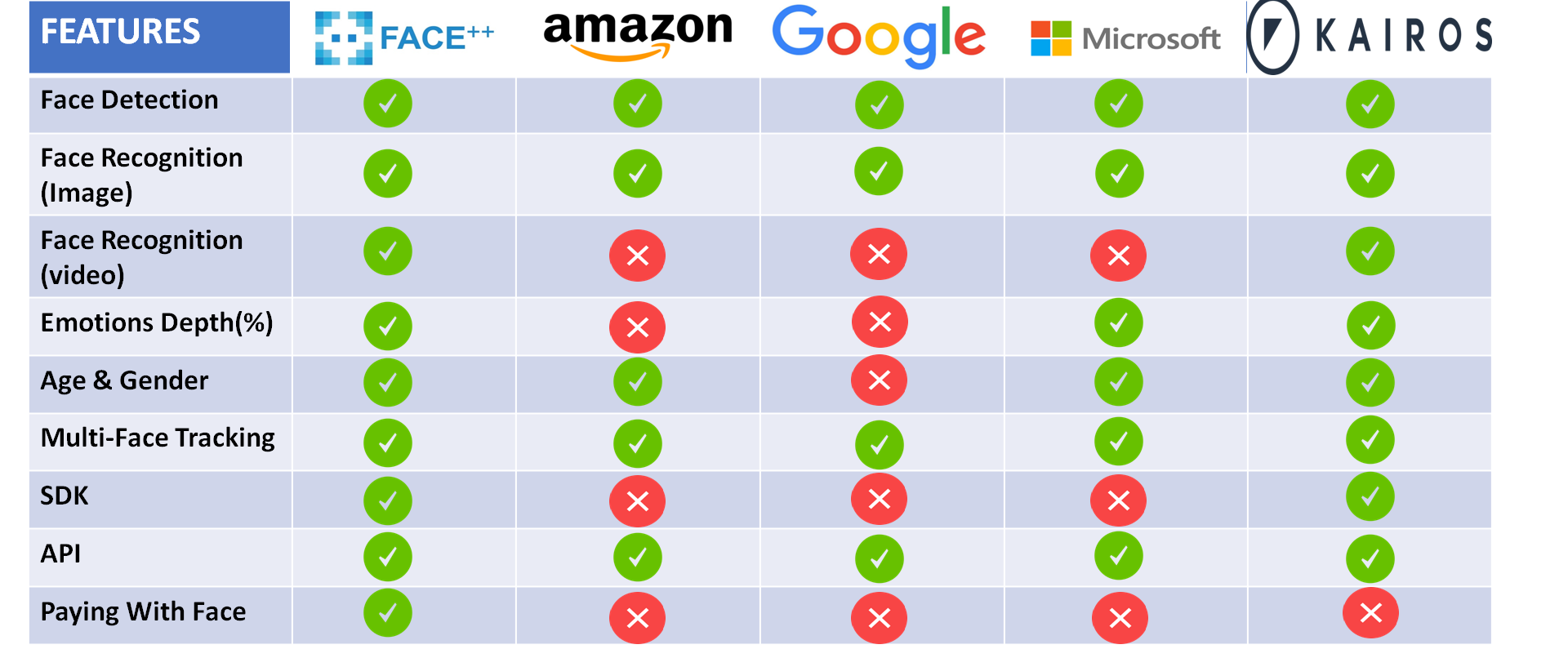
****

Table.3.11

**3.2 Applicable Patents**

**Display systems using facial recognition for viewership monitoring purposes   
(WO 2016187692 A1)**

|  |  |
| --- | --- |
| **Publication number** | WO2016187692 A1 |
| **Publication type** | Application |
| **Application number** | PCT/CA2015/050823 |
| **Publication date** | Dec 1, 2016 |
| **Filing date** | Aug 27, 2015 |
|  |  |
| **Priority date** | May 27, 2015 |
| **Inventors** | [Charlie TAGO](https://www.google.com/search?tbo=p&tbm=pts&hl=en&q=ininventor:%2522Charlie+TAGO%2522), [David Wang](https://www.google.com/search?tbo=p&tbm=pts&hl=en&q=ininventor:%2522David+Wang%2522), [Tago RANGINYA](https://www.google.com/search?tbo=p&tbm=pts&hl=en&q=ininventor:%2522Tago+RANGINYA%2522), [Jeffrey HIEBERT](https://www.google.com/search?tbo=p&tbm=pts&hl=en&q=ininventor:%2522Jeffrey+HIEBERT%2522) |
| **Applicant** | [Idk Interactive Inc.](https://www.google.com/search?tbo=p&tbm=pts&hl=en&q=inassignee:%2522Idk+Interactive+Inc.%2522) |

Table.3.2

**3.3 Applicable Standards**

**ANSI –ANSIINCITS 385-May 2004**, the US Standard for Digital Image Formats for use with the Facial Biometric - Referenced by DoC for US PIV ID Card, DHS for face capture.

**ISO –19794-5 FDIS–November 2004**, the Final Draft International Standard of Biometric Data Interchange Formats -Part 5 –Face Image Data.–Referenced by ICAO “Biometrics Deployment of Machine Readable Travel Documents”, TAG MRTD/NTWG May 2004.

**3.4 Present Market & Future Business Opportunity**

**Present Market**

**Ali Pay**

Ali pay is the leading online payment platform founded in China. Ali pay keeps improving user experience and transaction security, together with Face++ authentication technology. Users can reset their missing password using facial recognition.

**Camera 360**

Camera 360 is a popular photo editing and filter app, which achieved top rank in app stores in many countries. With Face++'s high-precision face detection and landmarks algorithms, Camera 360 allows users to edit their selfies with numerous effects.

**UCAR**

With facial recognition service by Face++, UCAR succeeded in building a safe and easy riding experience for both riders and drivers. Face Comparing offered great help in solving the problems of fraud and cheating.

**Lenovo**

Lenovo turned to Face++ for a leading and reliable biological identification solution. With Face++, Lenovo's video call application allows users to smile in front of their computer or mobile device, and directly log in their account.

**Future Business Opportunity**

**Better tools for law enforcement**

After the Boston Marathon bombing, the Boston police commissioner said that facial recognition software had not helped them identify Dzhokhar and Tamerlan Tsarnaev, despite the fact that the two were in public records databases—and photographed at the scene. Only, those images were taken from far away, the brothers were wearing sunglasses and caps, and many shots of them were in profile — all things that make facial recognition difficult.

**Facial recognition as advertising**

Could allow corporations and organizations to tap into our personal lives in unpredictable ways. Personalized ads as we walk down the street. There’s a potentially more subtle application of this technology too: ads that can identify us and our two favorite friends on Facebook. From there, it’s a snap to create a composite image of a person who’ll star in an ad targeted just to us.

**3.5 How is it developed or manufactured/assembled, what does it cost?**

It is developed in two types:

WEB API

Use Face++ capabilities online, with Face++ Web API service. Try out all Face++ features free. Offers flexible and cost-efficient pricing plans for businesses of all sizes.

1. Free $0

No credit card or upfront fees. Sign up and try out Face++ features

2. Pay As You Go

Starting from $0.0001 per request

Top up account and pay for what you consume. Use all features with guaranteed performance.

3. Hourly / Monthly Plan

Starting from $5 per hour, per QPS

Prepay for future usage by QPS capacity. Exploit APIs in the plan with lowest prices.

4. Hourly / Monthly Plan

Starting from $5 per hour, per QPS

Prepay for future usage by QPS capacity. Exploit APIs in the plan with lowest prices.

Mobile SDK

Use Face++ capabilities offline on devices, with Face++ Mobile SDKs. Test all Mobile SDKs free. We offer flexible and cost-efficient pricing plans for businesses of multiple sizes.

1. Online Licensing

Starting from $20,000

1000 licenses, per year, per platform

Prepay for license packs. Annual or monthly license available, with online activation.

2. Offline Licensing

Starting from $300,000 per year, per platform

Prepay for unlimited number of licenses. Annual license available. No need of online activation.

**3.6 Customer / User reviews of the product**

The face++ application is fast and accurate but differentiating twins is difficult.

The face++ technology is used by many companies and has a high level of acceptance in the face recognition market.

# How does it work?

Face++ detects and locates human faces within an image, and returns high-precision face bounding boxes.

Face detection is the first step to analysing and processing faces, Face++ also allows you to store metadata of each detected face for future use.

Use Face++ Detect API to detect faces within images, and get back face bounding box and token for each detected face. You can pass the face token to other APIs for further processing.

Detect API also allows you to get back face landmarks and attributes for the top 5 largest detected faces.

Face++ Compare API allows you to check the likelihood that two faces belong to the same person. You will get a confidence score and thresholds to evaluate the similarity. With Compare API, you can easily verify a user against a reference photo.

All of these systems take in data – often an image – from an unknown person, analyze the data in that input, and attempt to [match them to existing entries](https://doi.org/10.1109/TCSVT.2003.818349) in an database of known people’s faces or voices. Facial recognition does this in [three steps](https://www.gao.gov/products/GAO-15-621): detection, faceprint creation, and verification or identification.

Once the system has identified any potential faces in an image, it [looks more closely](https://cours.etsmtl.ca/sys828/REFS/Intro/Hanbook%2520of%2520Face%2520Recognition.pdf) at each one. Sometimes the image needs to be [reoriented or resized](https://cours.etsmtl.ca/sys828/REFS/Intro/Hanbook%2520of%2520Face%2520Recognition.pdf). A face very close to the camera may seem tilted or stretched slightly; someone farther back from the camera may appear smaller or even partially hidden from view.

When the software has arrived at a proper size and orientation for the face, it looks even more closely, seeking to create what is called a “[faceprint](https://www.springer.com/us/book/9780857299314).” Much like a fingerprint record, a faceprint is a set of characteristics that, taken together, uniquely identify one person’s particular face.

Find similar-looking faces to a new face, from a given collection of faces.

Face++'s fast and accurate search returns a collection of similar faces, along with confidence score and thresholds to evaluate the similarity.

With Face Searching, you can create an index of faces in your photo library. It is also applied to quickly determine the identities of people in large crowds , given a photo ID database.

Search API returns most similar-looking faces to a target face, from a given collection of faces, along with confidence scores and thresholds to evaluate the similarity.

To set up face collection, you need to first detect and store face metadata in FaceSet.

Search API is widely used in photo grouping and security monitoring.

Locate and return keypoints of face components, including face contour, eye, eyebrow, lip and nose contour. Face++ provides you with high-precision, most detailed landmarks with up to 106 points. Face++ landmarks fit the face seamlessly in images and videos.

High-precision landmarks enables you to apply sophisticated effects to face, such as face makeup, filters and avatar. Use Face++ Detect API to detect faces within images, and get back face bounding box and token for each detected face. You can pass the face token to other APIs for further processing. Detect API also allows you to get back 83-point landmarks and attributes for the top 5 largest detected faces. Get 83-point landmarks and attributes by passing its face token to Face Analyze API. You can get face token by using Detect API. Face Analyze API allows you to process 5 face token at a time. Face++ Face Landmark SDK enables your application to perform facial recognition on mobile devices locally. You can detect and track all the faces in videos streams in real time, and get back high-precision 106-point landmarks for each face.

# Design Drawings/Models

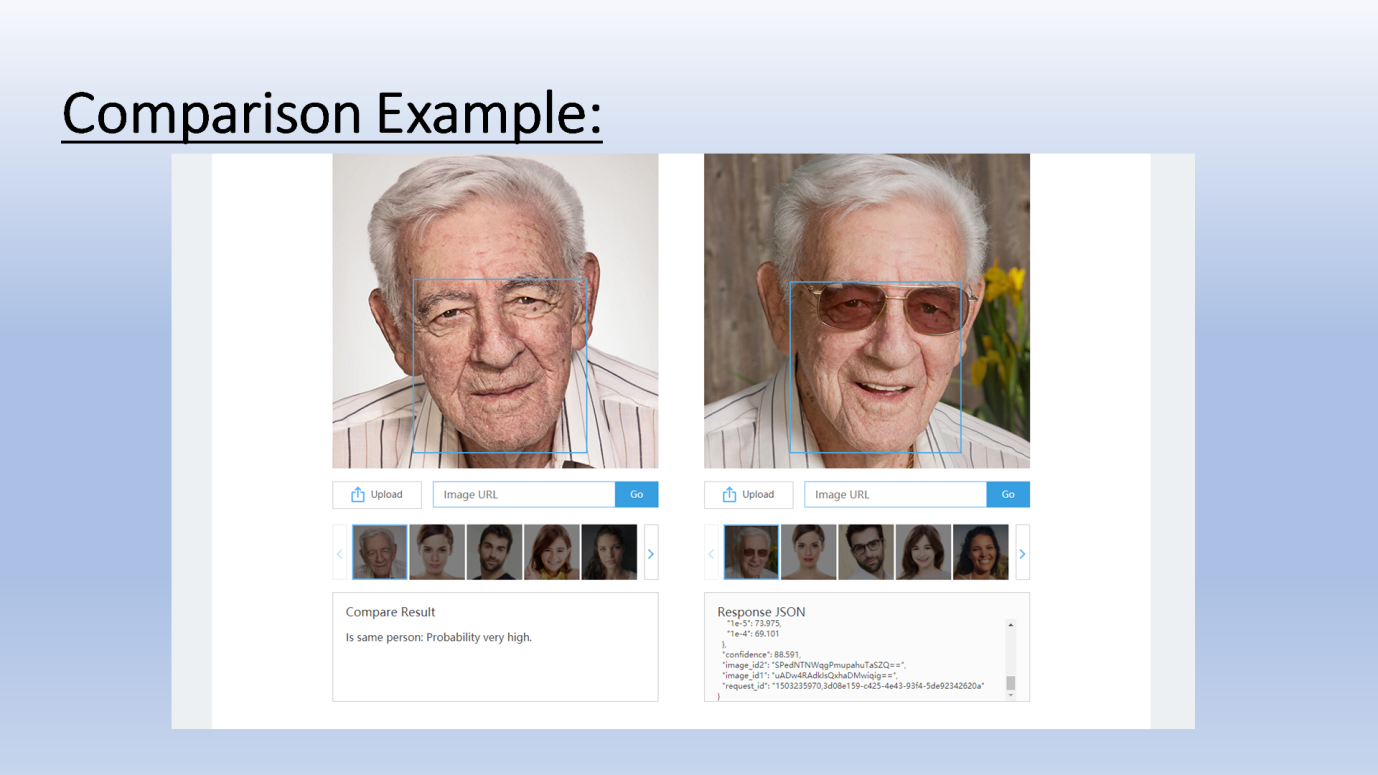
****

Fig.5.1

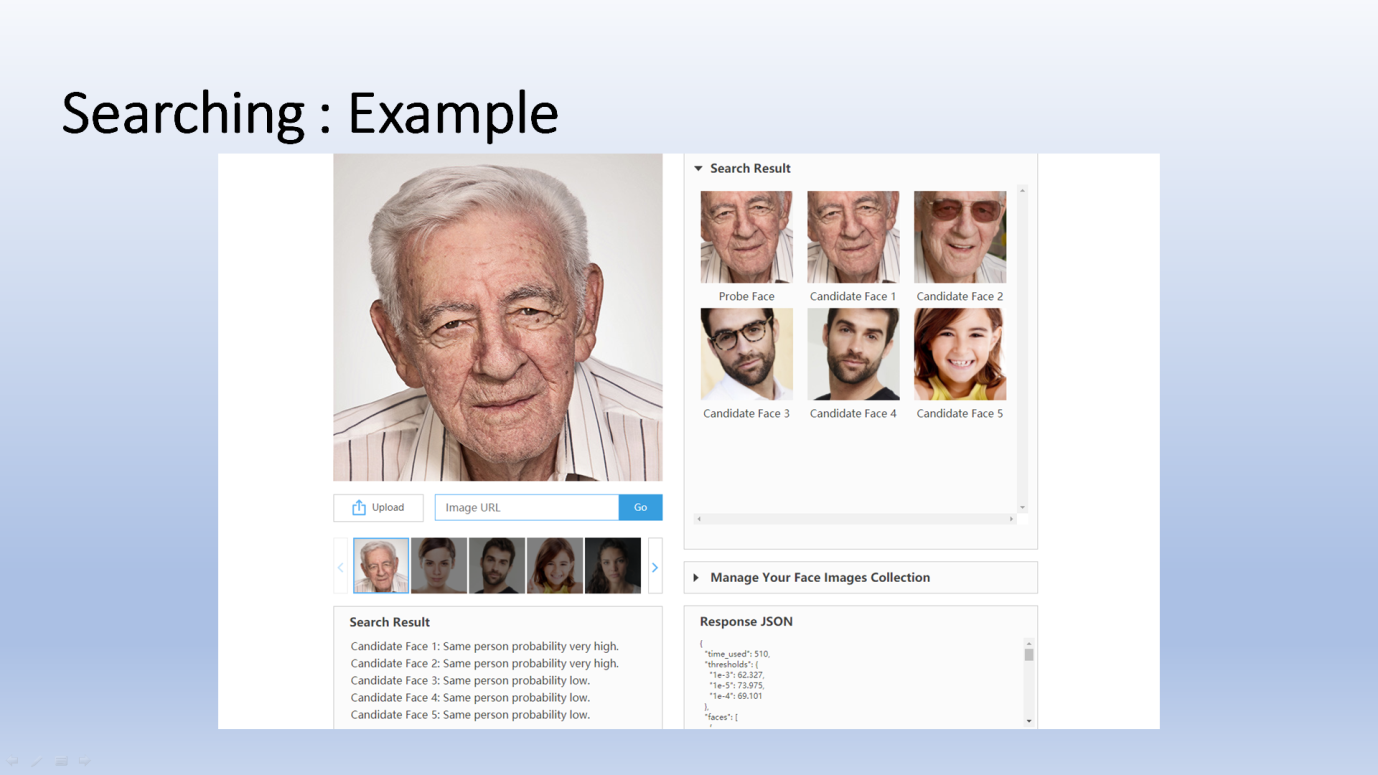
****

Fig.5.2

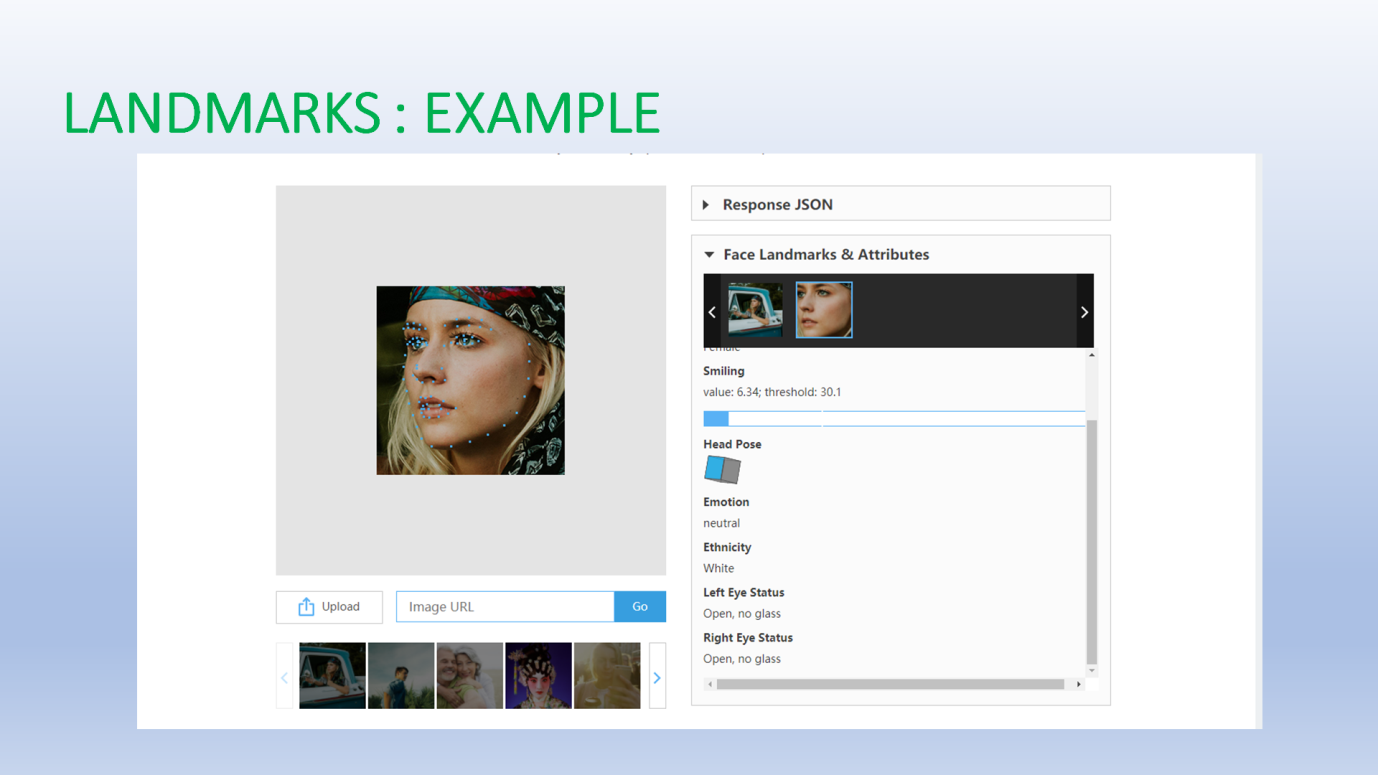
****

Fig.5.3

# Future Enhancements

**DIFFERENTIATING TWINS**

Face++ can differentiate between identical twins.

**BODY DETECTION**

Face++ detects and locates human bodies within an image, and returns high-precision body bounding boxes.

**BODY ATTRIBUTES**

Face++ analyzes a series of body related attributes based on machine learning technologies. You can get body attributes including gender, upper and lower body clothing colors.

**GESTURE RECOGNITION**

Detect and return locations of all the hands within images, and recognize hand gestures. For each hand gesture, Face++ provides high-precision confidence score. Face++ can recognize up to 19 kinds of gestures, such as "victory", "OK", "fist", "rock", etc.

# Study Phase - Conclusion

The product stands among the top few companies to have come up with an almost perfect face recognition algorithm.

A massive $47million investment as of 2017 clearly shows that the product has been delivering the best possible services to its clients.

It is competing with world class face recognition technologies like Amazon Rekognition, Google Vision and Microsoft Face. Even though these companies are light years ahead of FACE++, but as a startup tech, FACE++ is one of a kind.

Face++ comes under top 15 innovative technologies. It is ranked at 11th place.

Face Attributes: Get attributes including age, gender, head pose, emotion, beauty, eye gaze, and skin status.

Emotion Recognition: Analyze and identify emotion of detected faces. Analysis result of each detected face includes confidence scores for seven kinds of emotion: anger, disgust, fear, happiness, neutral, sadness, and surprise.

Beauty Score: Compute beauty scores for detected faces from both male's and female's perspective. Beauty score is not an index of the actual beauty, but indicates the average level of Asian men's and women's evaluation of the face.

Gaze Estimation: Face++ can estimate eye gaze direction in images, compute and return high-precision eye center positions and eye gaze direction vectors. You can perform real-time eye gaze tracking in videos.

## Skin-status Evaluation: Evaluate skin health status of detected faces, and recognize common skin problems, such as acne, dark circles, and spots on face. Face++ provides high-precision confidence scores for you to measure and assess facial skin status.

**DESIGN PHASE**

# Design Phase - Introduction

## 8.1 Problem Statement

Delay in receiving notices and examination schedules from offices are a major concern in the modern world. Missing and misplacing of important notices and deadlines of assignments is still a persisting issue in most colleges. Wastage of paper for printing notices is yet another concern.

## 8.2 Project Objectives

* + - The project was proposed as a replacement to the typical wooden noticeboards where the students found it difficult to access the put up notices.
    - Planned on implementing a digital notice board linked to a notification app that allows a student to access notices without any issues.
    - The project was planned to be executed as a part of the Design Project Course.
    - The project will be effective if implemented at colleges or other institutions where checking notices is a tedious issue.
    - The beneficiaries to the project will be both the staff and students of a college, as, the project under consideration makes the putting up and accessing of notices much more easier.

## 8.3 Design and Implementation Constraints

It was required to establish a connection between the college servers and the notice board and app. The website if mishandled might cause a headache to the whole institution. The investment into the notice boards is also a constraint as a standard LCD display board costs around INR 12,000/-

# Literature Survey

The research was conducted on SedaoINFOrm, Sedao is a manufacturer of digital signage and communications systems. Based in the UK, this company provides digitalised notices boards for hospitals, offices and educational institutions. Their INFOrm makes use of a digital notice board and related software to display and update notices in schools.

# System Overview and Requirement Specification

## 10.1 System overview

## 

## There were four main solutions under consideration when brainstorming was conducted to find a solution to the problem to find an alternative to the existing typical wooden notice board and for the issue of misplacing and the hectic work included in organising the notices and the wastage of paper.

## The solutions were:

## Email notification

## Digital Displays on ID cards

## Digital Notice board and notification app

## On carefully assessing the merits and demurest of each solution, we found that:

## Loads of emails regarding the notices create a tedious process to check the mail on a daily basis and, most important notices can go unnoticed.

## 

## The implementation of ID cars with digital display feature was estimated to be too expensive and, once a student is graduated, the card becomes of no use.

## 

## We concluded that both implementation of Digital notice boards and the notification app seemed to be fitting enough to solve the existing problem effectively and since both these solutions required a single database in common, we decided to club it into a single and more effective solution to the problem in hand.

## 

## 10.2 Problem Domain Model

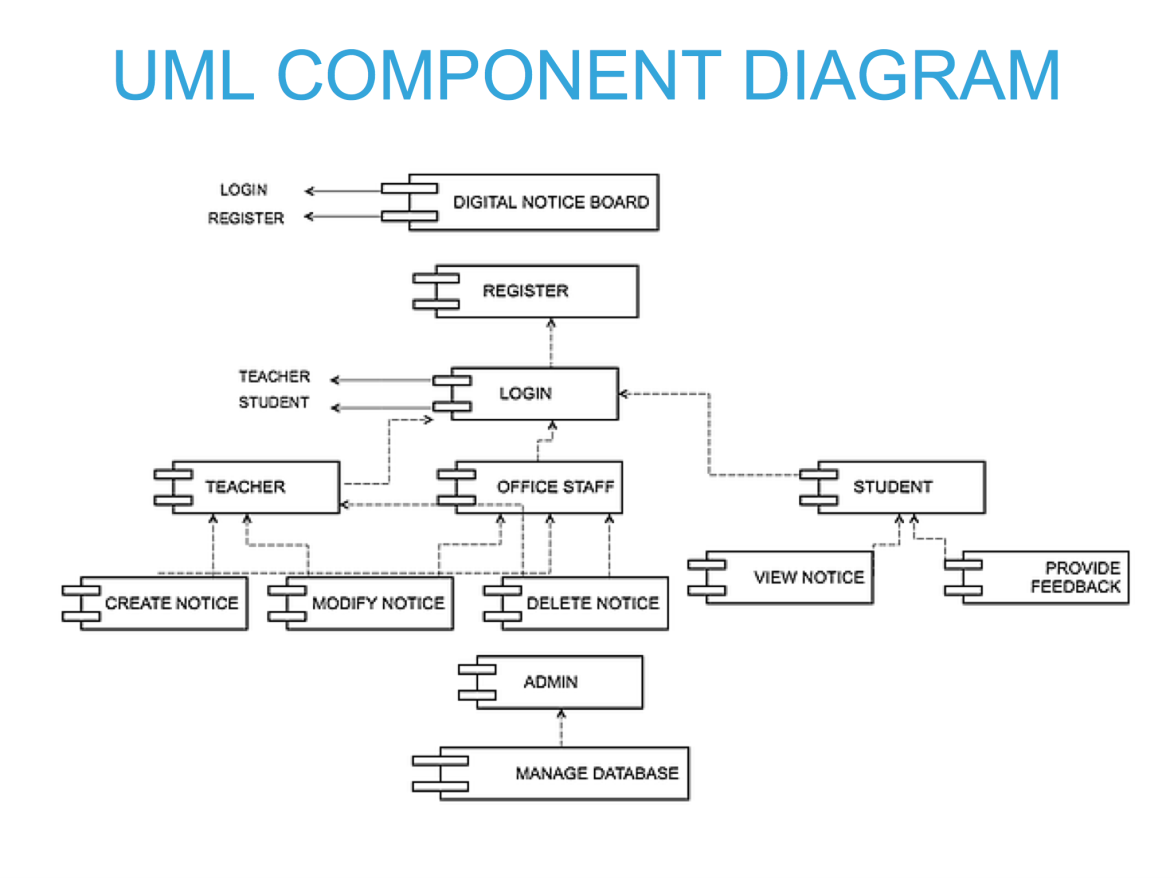


Fig.10.2

The above UML diagram clearly represents the relations between the various entities- Student, Teachers and Office staff, and how the entire problem revolves around them and how the solution is fit for the selected problem.

The usual tedious process of the authorities to set up a notice is simplified by coordinating the tasks of the authorised personnel, thereby making it easier for the students to access the notices in a hassle-free manner.

## 10.3 Functional and Non-Functional Requirements (Requirement Model)

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. No.** | **Requirements** | **Requirement Type (Functional or Non-Functional)** | **Priority or Importance (High, Medium, Low)** |
| 1 | Faster access to notices. | Non-functional | High |
| 2 | The authorities needed to easily upload notices without having to consult with other people. | Non-functional | High |
| 3 | The digital board need to be of a convenient size for better visibility. | Functional | Medium |
| 4 | The notification app needed to have a good user interface for hazzle-free experience . | Functional | Medium |
| 5 | A High-end server with an expert in managing server database should be assigned. | Functional | High |
| 6 | us  User interface for notice creation can be made much more | Functional | Medium |
| Table.10.3 | | | |

## 10.4 List of Actors

|  |  |
| --- | --- |
| **Actor** | **Description** |
| Teacher | Teachers can create, modify and delete notices whenever necessary. |
| Office staff | Office staff can create, modify and delete notices whenever needed. |
| Student | Students can login, view notice and send feedbacks whenever necessary. |
| Database Admin | The database admin monitors the servers for any errors if any and provides the required maintenance. |

Table.10.4

## 10.5 List of Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | **Use Case Name** | **Description** | **Mapping to Functional Requirement (Req. No.)** |
| 1 | Login | The teachers or office staff can log into the webpage using their respective usernames and passwords. | 2 |
| 2 | Create Notice | Notices can only be created by the authorised personnel. | 6 |
| 3 | Modify Notice | It enables the authorised people to modify the notices as per requirement. | 6 |
| 4 | Delete Notice | It enables the authorised people to delete the notices that are no longer needed. | 6 |
| 5 | View Notice | Students can view the notices created by the teachers and office staffs. | 3 |
| 6 | Send Feedback | The students can send their feedbacks or queries about a particular notice to the respective creators. | 4 |
| 7 | Manage Database | Database containing the notices stored in the servers need to be constantly monitored and maintained if necessary. This is done by the database admin. | 5 |

Table.10.5

## 10.6 Use Case Diagram

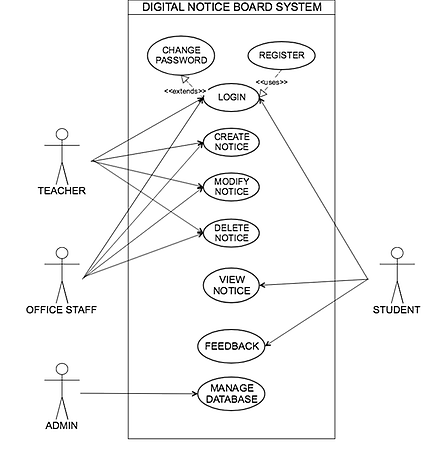


Fig 10.6

**PRIMARY ACTORS**

-**Teacher**. -**OfficeStaff**. -**Admin**. -**Student**.

**PRE-CONDITIONS**

-The server is up and running.

-There exists an active connection between the server and the digital notice boards and the notification app and website.

## 10.7 Use Case Specification

## 10.7.1 Login

|  |  |
| --- | --- |
| **Use Case ID** | 1 |
| **Use Case Name** | Login |
| **Description** | The teachers or office staff can log into the webpage using their respective usernames and passwords. |
| **Actor(s)** | Teacher, Office staff, Student |
| **Pre-Conditions** | User name & password |
| **Flow of Events (Main Flow & Alternate Flows – give as numbered bullet points)** | The authorised personnel, may it be a teacher or an office staff, registers on the website and logs in using the username and password |
| **Post-Conditions** | Logged in successfully. |
| Table.10.7.1 | |

## 10.7.2 Create Notice

|  |  |
| --- | --- |
| **Use Case ID** | 2 |
| **Use Case Name** | Create Notice |
| **Description** | Notices can only be created by the authorised personnel. |
| **Actor(s)** | Teacher, Office staff |
| **Pre-Conditions** | Logged in and create notice option selected |
| **Flow of Events (Main Flow & Alternate Flows – give as numbered bullet points)** | A notice is created in the webpage which is directly loaded into the server and then to the respective digital notice boards in the classrooms.  It is also sent to the notification app installed in the students’ mobile, which notifies the student about the new notice being created |
| **Post-Conditions** | Notice created and uploaded to server. |

## 

Table.10.7.2

## 10.7.3 Modify Notice

|  |  |
| --- | --- |
| **Use Case ID** | 3 |
| **Use Case Name** | Modify Notice |
| **Description** | It enables the authorised people to modify the notices as per requirement |
| **Actor(s)** | Teacher, Office staff |
| **Pre-Conditions** | Should be logged in and there should be notice present to be modified. |
| **Flow of Events (Main Flow & Alternate Flows – give as numbered bullet points)** | These notices can also further be modified accordingly or removed if necessary |
| **Post-Conditions** | Modifications applied in the notices stored in the server. |

Table.10.7.3

## 10.7.4 Delete Notice

|  |  |
| --- | --- |
| **Use Case ID** | 4 |
| **Use Case Name** | Delete Notice |
| **Description** | It enables the authorised people to delete the notices that are no longer needed. |
| **Actor(s)** | Teacher, Office staff |
| **Pre-Conditions** | User should be logged in and there should be notice’s to be deleted |
| **Flow of Events (Main Flow & Alternate Flows – give as numbered bullet points)** | A notice is automatically removed once the deadline is crossed |
| **Post-Conditions** | Notice removed from the server. |

Table.10.7.4

## 10.7.5 View Notice

|  |  |
| --- | --- |
| **Use Case ID** | 5 |
| **Use Case Name** | View Notice |
| **Description** | Students can view the notices created by the teachers and office staffs. |
| **Actor(s)** | Teacher, Office staff, Student |
| **Pre-Conditions** | There should be any notice to be viewed |
| **Flow of Events (Main Flow & Alternate Flows – give as numbered bullet points)** | The students can view the notice via the digital display board or the notification application |
| **Post-Conditions** | Created notice can be viewed from display board and app. |
| Table.10.7.5 | |

## 10.7.6 Send Feedback

|  |  |
| --- | --- |
| **Use Case ID** | 6 |
| **Use Case Name** | Send Feedback |
| **Description** | The students can send their feedbacks or queries about a particular notice to the respective creators. |
| **Assumptions, if any** |  |
| **Actor(s)** | Student |
| **Pre-Conditions** | Student should be logged in |
| **Flow of Events (Main Flow & Alternate Flows – give as numbered bullet points)** | The student can send a feedback if necessary which will be displayed in the website for the authorised people to see. |
| **Post-Conditions** | Feedback sent to corresponding staff. |

Table.10.7.6

## 10.7.7 Manage Database

|  |  |
| --- | --- |
| **Use Case ID** | 7 |
| **Use Case Name** | Manage Database |
| **Description** | Database containing the notices stored in the servers need to be constantly monitored and maintained if necessary. This is done by the database admin. |
| **Assumptions, if any** |  |
| **Actor(s)** | Database Admin |
| **Pre-Conditions** | Database admin should be logged in |
| **Flow of Events (Main Flow & Alternate Flows – give as numbered bullet points)** | The database admin monitors the servers for any errors if any and provides the required maintenance. |
| **Post-Conditions** | Errors resolved and database working set to normal.  Table.10.7.7 |

**10.8. Wireframe or Screen Format:**

Notification App

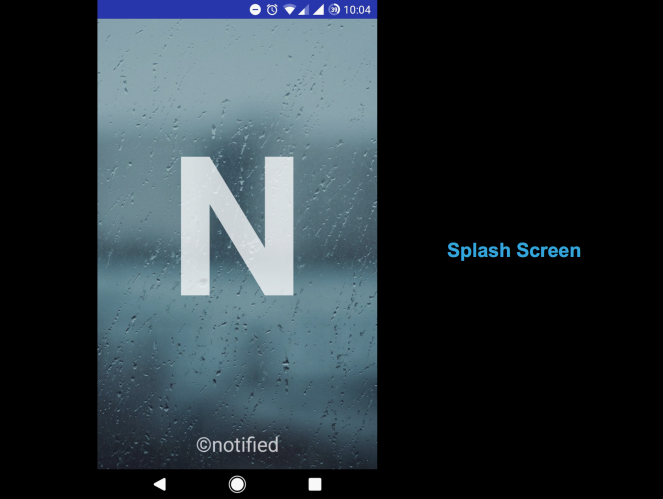


Fig.10.8.1

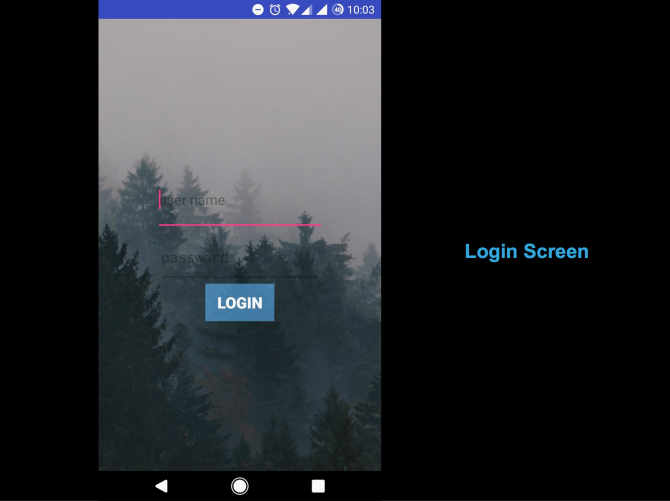


Fig.10.8.2

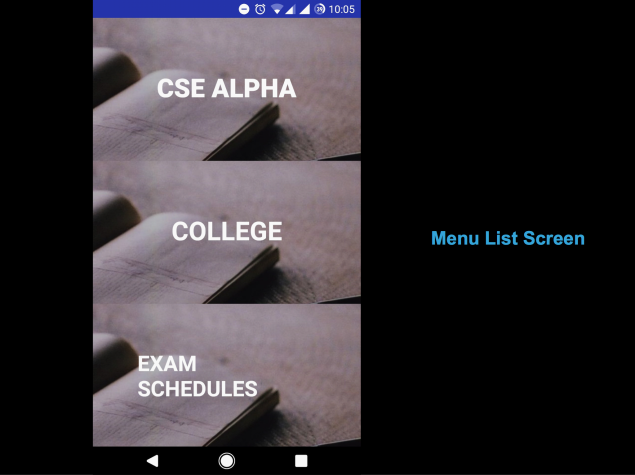


Fig.10.8.3

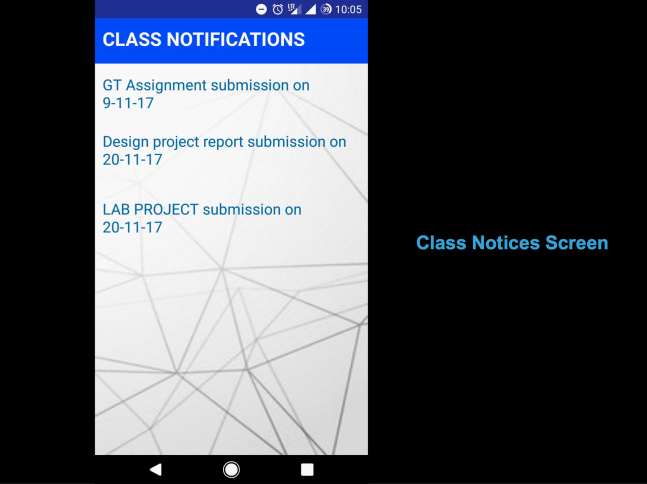


Fig.10.8.4

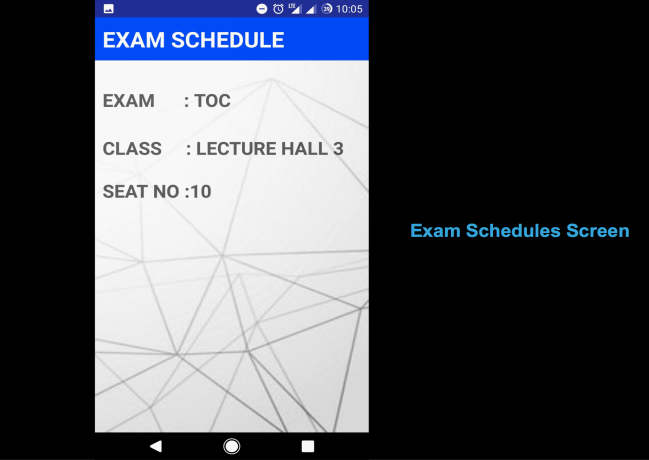


Fig.10.8.5

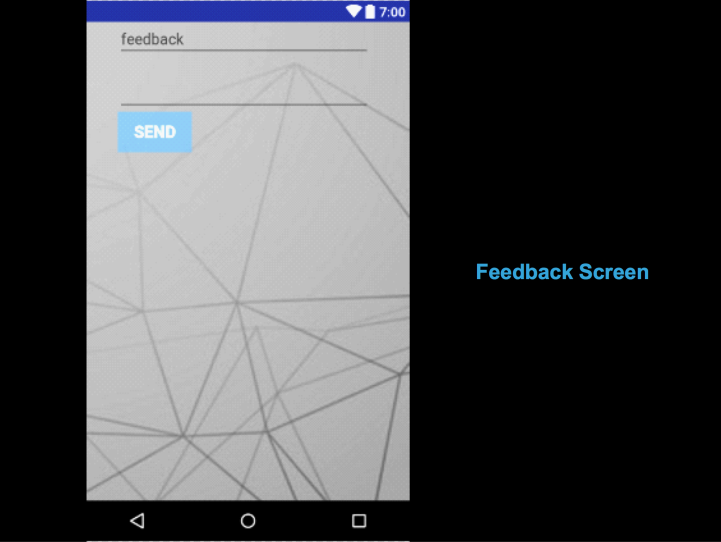


Fig.10.8.6

**Website for notice creation**

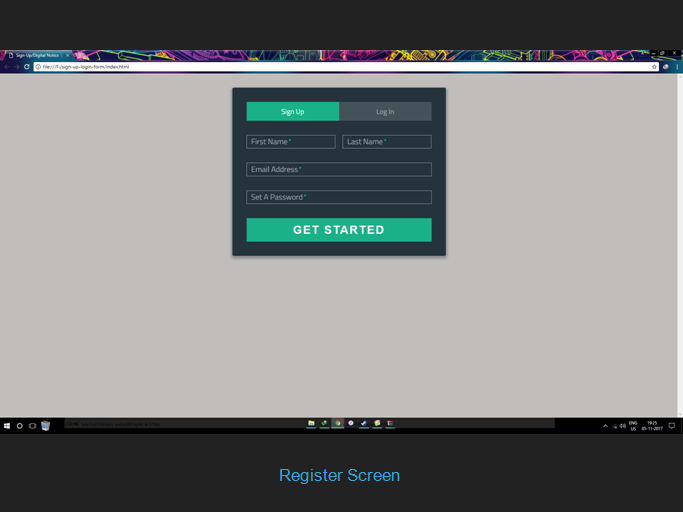
****

Fig.10.8.7

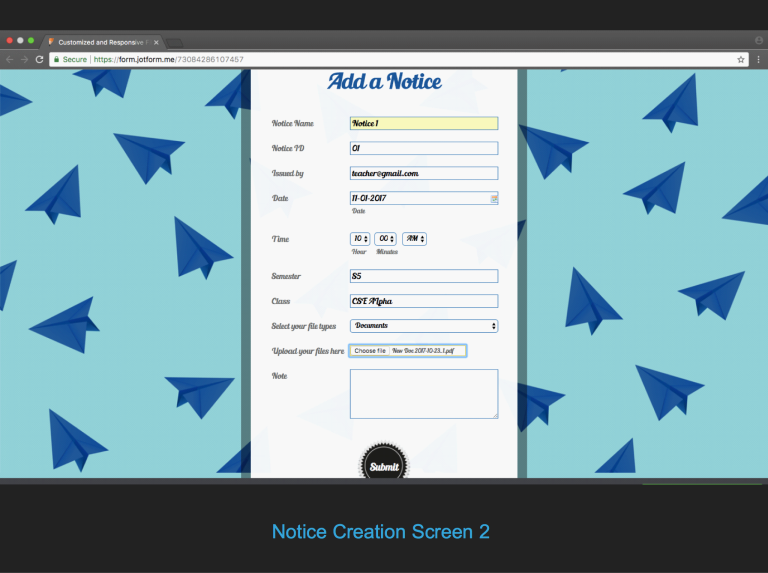


Fig.10.8.8

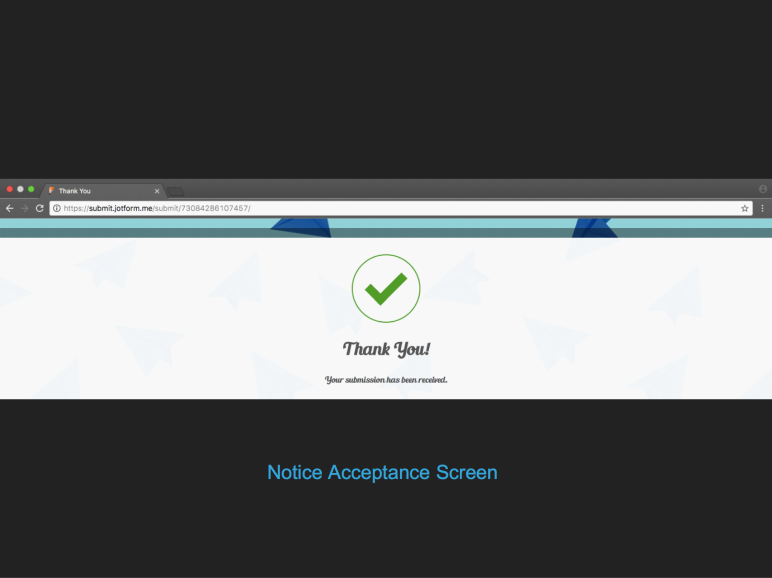
****

Fig.10.8.9

# High Level Design (Architecture Diagram)

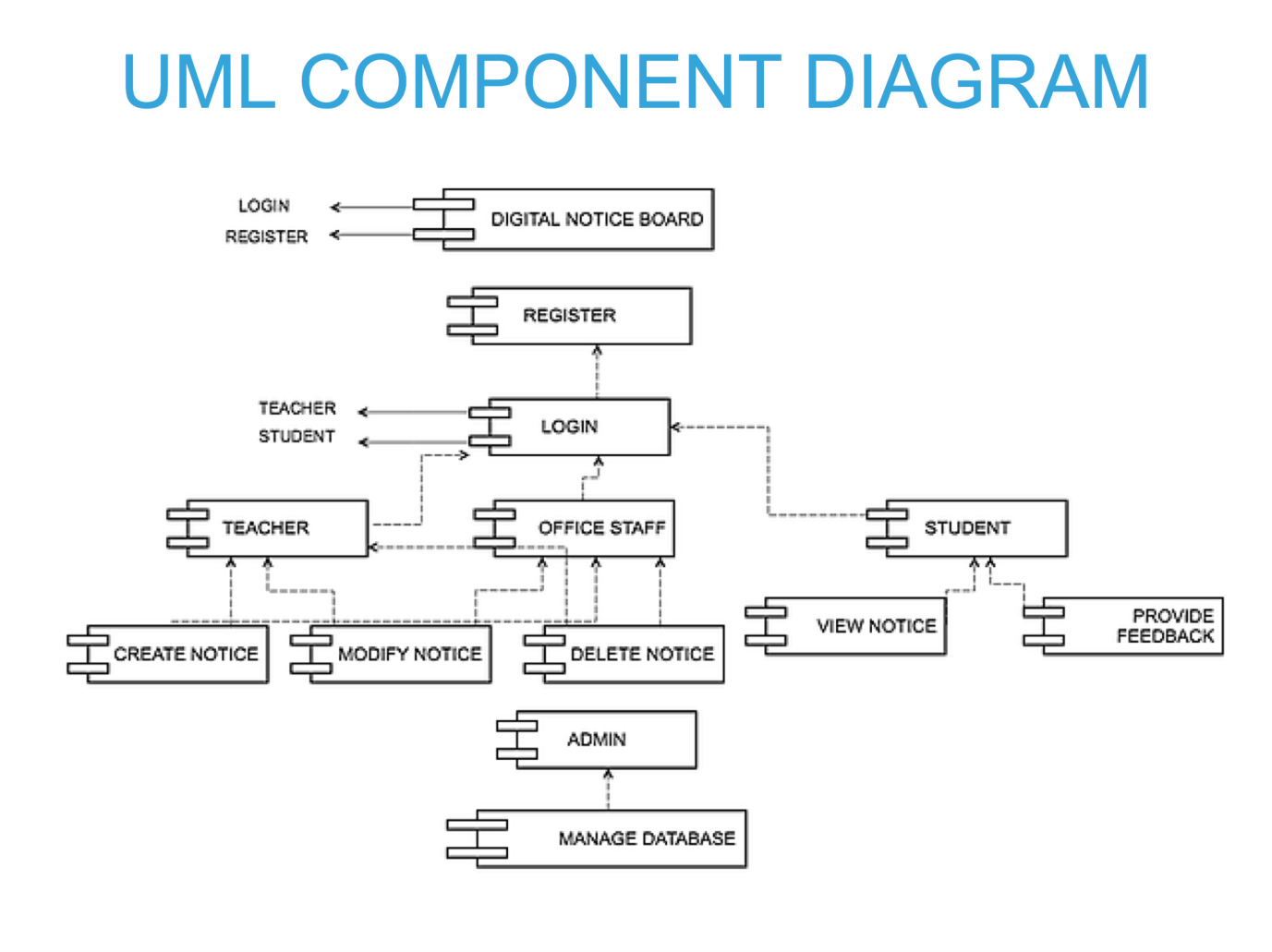


Fig.11.1

It is mainly used to represent the relationship between the required interfaces and ports.

Here the non-related interfaces are separately listed.

The various interfaces are connected by lines to represent the relation between them.

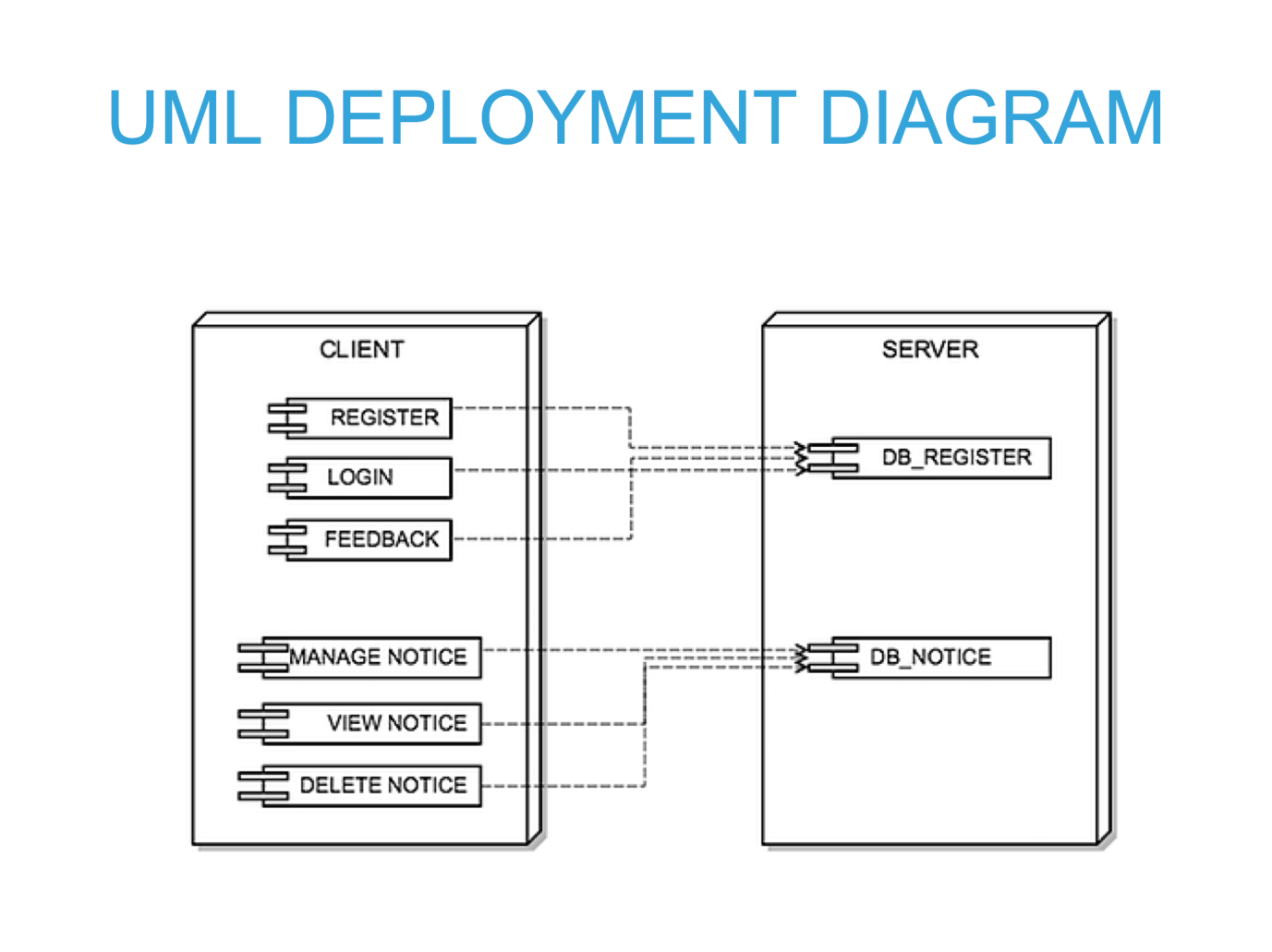


Fig 11.2

This diagram shows the configuration of the processing nodes at run-time and the components that live on them.

Here, the deployment target or node is the SERVER.

And the components relying on it are represented the CLIENT environment.

Here the registration details are stored in the DB\_REGISTER database.

The notice related details are stored in the DB\_NOTICE database.

The REGISTER, LOGIN, FEEDBACK use cases depend on DB\_REGISTER.

The MANAGE NOTICE, VIEW NOTICE, DELETE NOTICE use cases depend on DB\_REGISTER.

# Analysis and Design Models

## 12.1 Sequence Diagram

## 12.1.1 Login

## 



Fig.12.1

## 12.1.2 Create notice

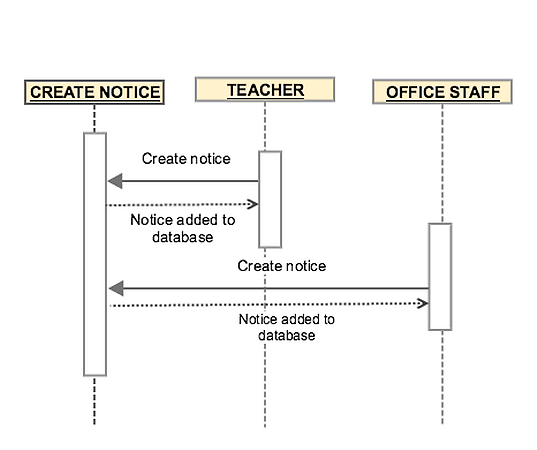


Fig.12.2

## 12.1.3 Modify notice

Fig.2.3

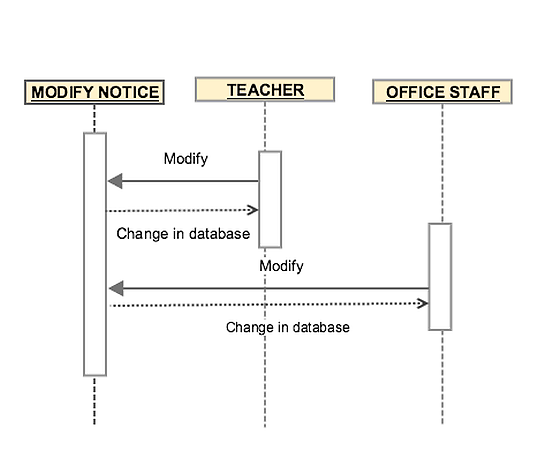


Fig.12.3

## 12.1.4 View notice

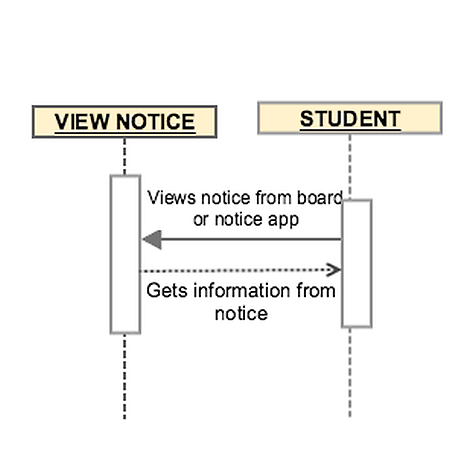


Fig.12.4

## 12.1.5 Delete notice

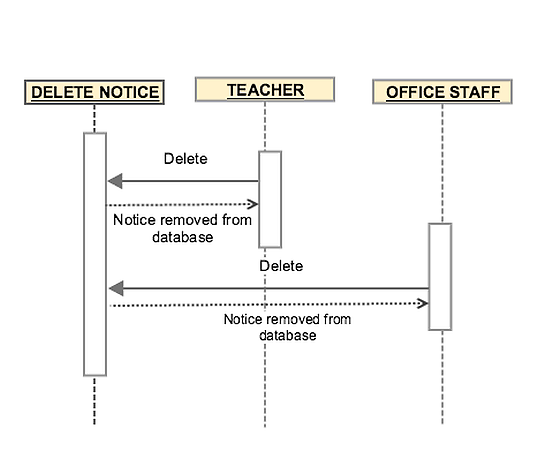


Fig.12.5

## 12.2 Class Diagram

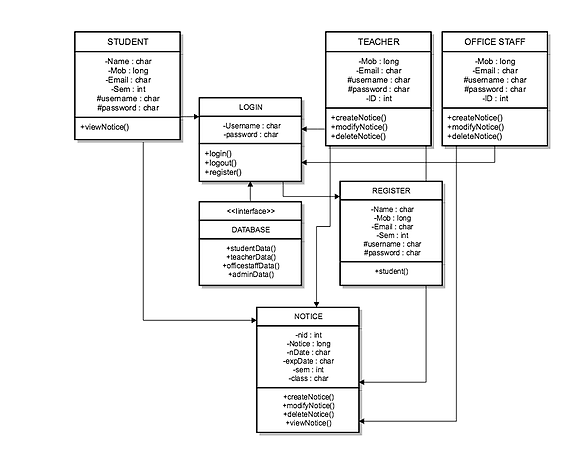


Fig.12.2

## A Class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, methods(operations), and the relationships among objects.The given class diagram contains 7 classes.

## STUDENT

## Class STUDENT has attributes Name, Mob, Email and 2 protected attributes username and password. Its methods are createNotice(), modifyNotice() and deleteNotice(). It is associated to classes LOGIN and NOTICE

## TEACHER

## Class TEACHER has attributes Mob, Email, Sem, ID and 2 protected attributes username and password. Its methods is viewnotice() alone. It is associated to classes LOGIN and NOTICE

## OFFICE STAFF

## Class OFFICE STAFF has attributes Mob, Email, Sem, ID and 2 protected attributes username and password. Its methods is viewnotice() alone. It is associated to classes LOGIN and NOTICE

## LOGIN

## Class LOGIN has 2 private attributes username and password. Its methods are login(), logout() and register(). It is associated to REGISTER.

## REGISTER

## Class REGISTER has attributes Name, Mob, Email and 2 protected attributes username and password. It has a method student(). It is associated with NOTICE.

## NOTICE

## Class NOTICE has attributes nID, nDate, Notice, expDate, sem and class.Its methods are createNotice(), modifyNotice(), deleteNotice() . and viewNotice().

## The class DATABASE is more like an interface and contains methods like studentData(), teacherData(), officestaffData() and adminData().

## 12.2.1 List of Classes

|  |  |  |  |
| --- | --- | --- | --- |
| Class No. | Class Name | Category (Boundary, Control or Entity) | Description |
| 1 | STUDENT | Entity | Students can login, view notice and send feedbacks whenever necessary. |
| 2 | LOGIN | Control | The authorised personnel, may it be a teacher or an office staff, registers on the website and logs in using the username and password |
| 3 | TEACHER | Entity | Teachers can create, modify and delete notices whenever necessary. |
| 4 | OFFICE STAFF | Entity | Office staff can create, modify and delete notices whenever needed. |
| 5 | DATABASE | Boundary | Database is used to store the various notices being created, and also the data like usernames and passwords of various users. |
| 6 | REGISTER | Control | The authorised personnel need to register on the website so as to get a username and password to make sure that only the authorised people can create or modify the notices. |
| 7 | NOTICE | Boundary | Notices contain various information that have to be passed onto the students regarding the submission dates or other related information. |

Table.12.2.1

## 12.2.2 List of Classes with Attributes and Operations

|  |  |  |  |
| --- | --- | --- | --- |
| Class no | Class name | Attribute | Operations |
| 1 | STUDENT | Name, Mob, Email, Semester, #username, #password | viewNotice() |
| 2 | LOGIN | Username, password | login(),logout(),register() |
| 3 | TEACHER | Name,Mob,Email, #username, #password, ID | createNotice(),  modifyNotice(), deleteNotice() |
| 4 | OFFICE STAFF | Name,Mob,Email, #username, #password, ID | createNotice(),  modifyNotice(), deleteNotice() |
| 5 | DATABASE |  | studentData(),teacherData(),officestaffData(), admindata() |
| 6 | REGISTER | Name, Mob, Email, Semester,#username, #password | Student(), |
| 7 | NOTICE | nid, Notice, nDate, Expdate, Semester, Class | createNotice(),modiftNotice(), deleteNotice(), viewNotice() |

Table.12.2.2

# Data Model

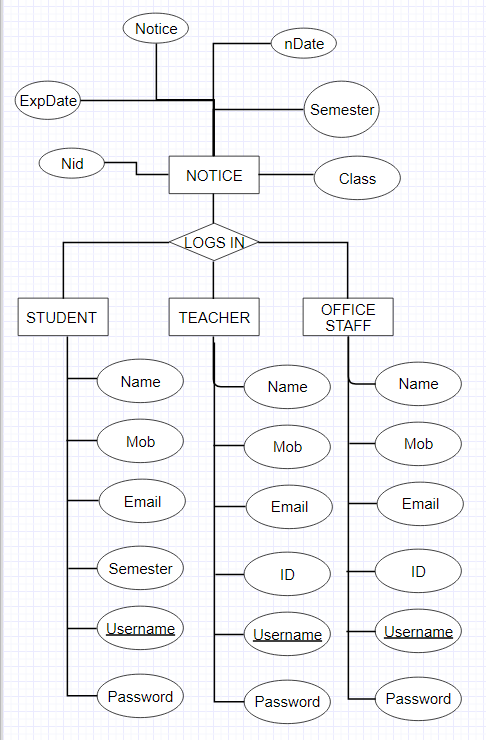


Fig 13

# Limitations

The power used by a LCD screen is high. There must be an individual that should maintain the server and the databases.LCD displays can get affected by incoming sunlight making it difficult to read. The students have to wait sometime for the next notice to appear as the notice that they wish to read may not be current one being displayed.

For LCD displays,

* The color and contrast from various viewing angles is inconsistent.
* Motion blur is common.
* With brightness from backlighting, imagery may appear flat.
* Pixel-based display may be stuck on screen.
* Fixed resolution.
* Imagery not as good with analog interface

# Conclusion

Wireless operations allow services, such as long-range communications, that are impossible or impractical to implement with the use of wires. It provides fast transfer of information and It is cheaper to install and maintain. This paper provides an efficient way of displaying messages on Notice Board digitally It also provides user authentication in order to avoid any misuse of the system.

Remote control is the most popular gadget nowadays. Right from the intense creativity of remotely controlling laser chip makers to the highly destructive remotely ignitable bombs, from the pins to the planes, remote control is occupying a omnipresence state and it is also enhancing its scope and domains.

**Bibliography**

<https://www.faceplusplus.com/>

<https://www.technologyreview.com/s/603494/10-breakthrough-technologies-2017-paying-with-your-face/>

<https://techcrunch.com/2015/05/14/faceplusplus/>

<http://www.sedao.co.uk/digital-signage-products/education/INFOrm.htm>

https://blog.ted.com/the-future-of-facial-recognition-7-fascinating-facts/

Understanding the Face Image Format Standards

Paul Griffin, Ph.D.

Chief Technology Officer

Identix

April 2005

http://www.lcd.net/advantages-and-disadvantages-of-lcd-technology.html

**Appendix**

**Table Name:**  Student

**Table Description:** To store student details and feedback

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Data Type & Size** | **Description** |
| Name | VARCHAR2(50) | Name of user |
| Mob | Number(15) | Mobile number |
| Email | VARCHAR2(50) | Email Id |
| Semester | Number | Semester |
| **Username** | **VARCHAR2(50)** | **Unique user name** |
| Password | VARCHAR2(50) | User set password |

**Primary Key:** Username

**Table Name:**  Teacher

**Table Description:** To store teacher details.

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Data Type & Size** | **Description** |
| Name | VARCHAR2(50) | Name of user |
| Mob | Number(15) | Mobile number |
| Email | VARCHAR2(50) | Email Id |
| ID | Number | College ID no |
| **Username** | **VARCHAR2(50)** | **Unique user name** |
| Password | VARCHAR2(50) | User set password |

**Primary Key:** Username

**Table Name:**  OfficeStaff

**Table Description:** To store teacher details.

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Data Type & Size** | **Description** |
| Name | VARCHAR2(50) | Name of user |
| Mob | Number(15) | Mobile number |
| Email | VARCHAR2(50) | Email Id |
| ID | VARCHAR2(50) | College ID no |
| **Username** | **VARCHAR2(50)** | **Unique user name** |
| Password | VARCHAR2(50) | User set password |

**Primary Key:** Username

**Table Name:**  Notice

**Table Description:** To store notice details.

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Data Type & Size** | **Description** |
| Notice | VARCHAR2(50) | Notice name |
| Class | Number(15) | Class to which notice is issued |
| Semester | VARCHAR2(50) | Semester |
| **Nid** | **VARCHAR2(50)** | **Notice ID** |
| nDATE | DATE | Date of notice |
| ExpDate | DATE | Expiry date of notice |

**Primary Key:** Nid