

FINAL YEAR PROJECT

CloudKibo: Interactive Learning System

A Complete Video Conferencing Solution for Virtual Classrooms

Sojharo (sojharo.mangi@khi.iba.edu.pk)
Saba Channa (saba.channa@khi.iba.edu.pk)
Nadeem Hussain (nadeem.hussain@khi.iba.edu.pk)

Introduction:

CloudKibo is a complete video conferencing solution with vision to make virtual classroom environment possible. The system provides video and audio chat facilities to teachers and students. Teacher can remotely deliver lecture using the video broadcasting feature. Moreover, students can also hold video conferences virtually for group studies. Student may hold private video chat with a teacher too.

To make it more productive, system also allows to share computer screens during live call for delivering presentations. Moreover, system provides file sharing facility for teachers to provide handouts and for students to submit assignments. During the call, files can also be transferred in real-time. There is no size limit for transferring file.

Besides, users are also provided the facility to do instant messaging (online chat) parallel to video and audio chat facilities. Teachers and students will be able to make virtual classrooms and start collaborative learning. Beside Virtual School System, it is a full video communication tool in itself. In short, with CloudKibo, we can do:

- Video Call
- Audio Call
- File Transferring (up to any size you want)
- Screen Sharing
- Instant Messaging
- Enroll Courses and meet new study partners
- Online Group Study with CloudKibo Conference Call

CloudKibo is a *communication platform* which has better functionality than Skype on one hand and it is an *interactive learning system* on the other hand.

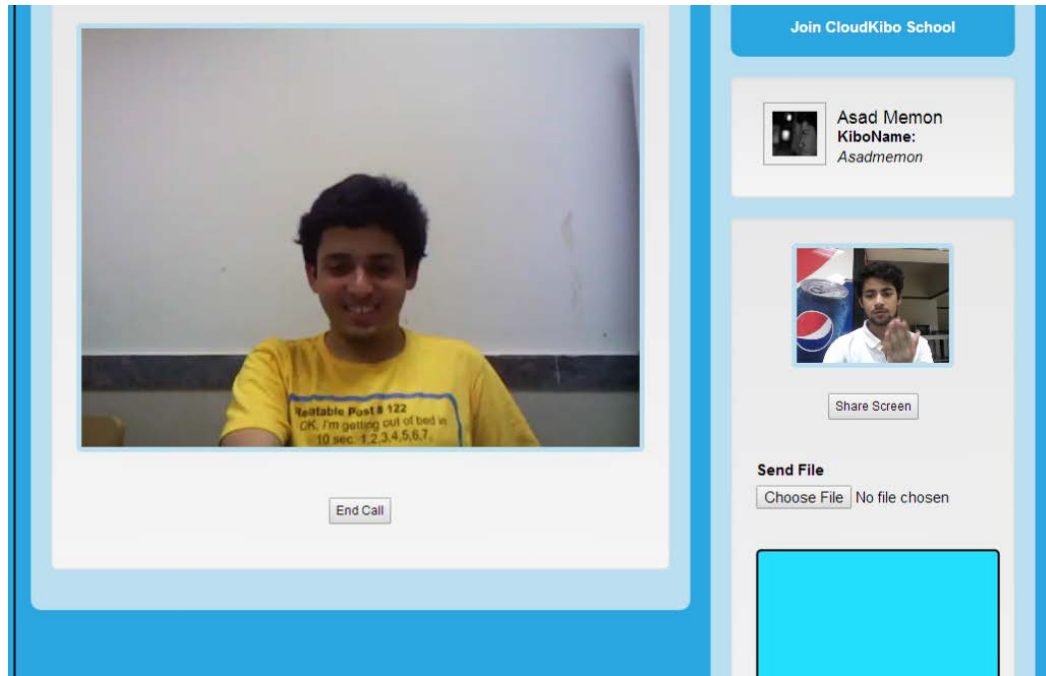


Figure 1 One-to-One Video Call with file transfer, Instant Messages and Screen sharing options visible

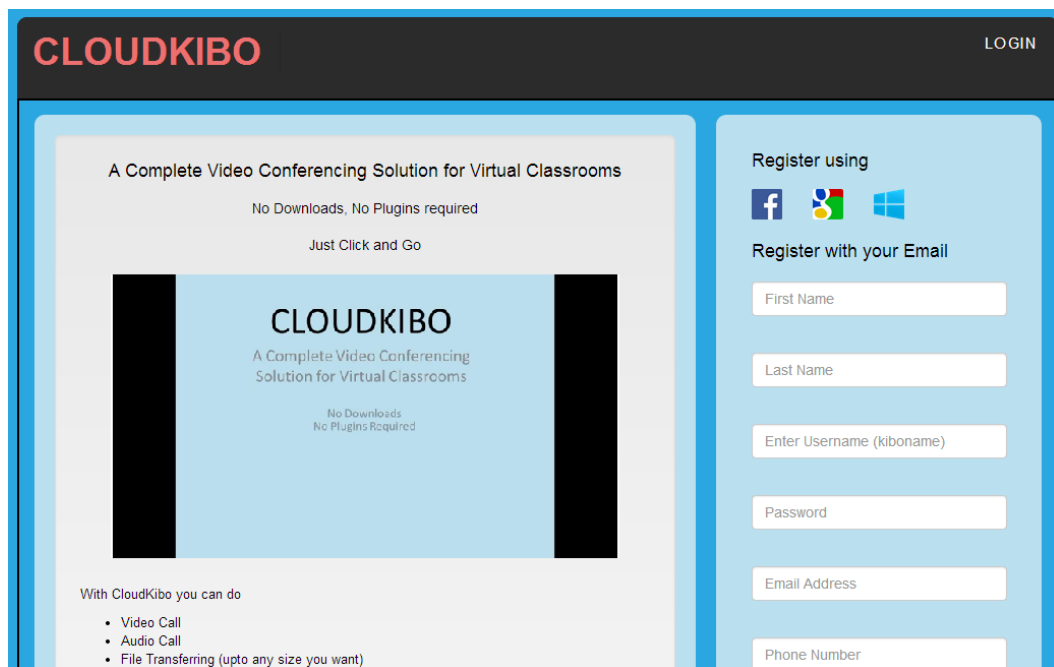


Figure 2 Welcome Page with Register and Login Forms, Introductory Video is also visible

High Level Design:

1. Rationale and sources of your project idea

World is going towards cloud and, sooner or later, all the service based systems will be based on cloud. Our project aims in future as real-time communication will have to be cloud-based. We approach this problem in a different way than other competitors. We won't require our users to download any plug-ins for this software to run. They will just login and start video calling.

2. Logical structure

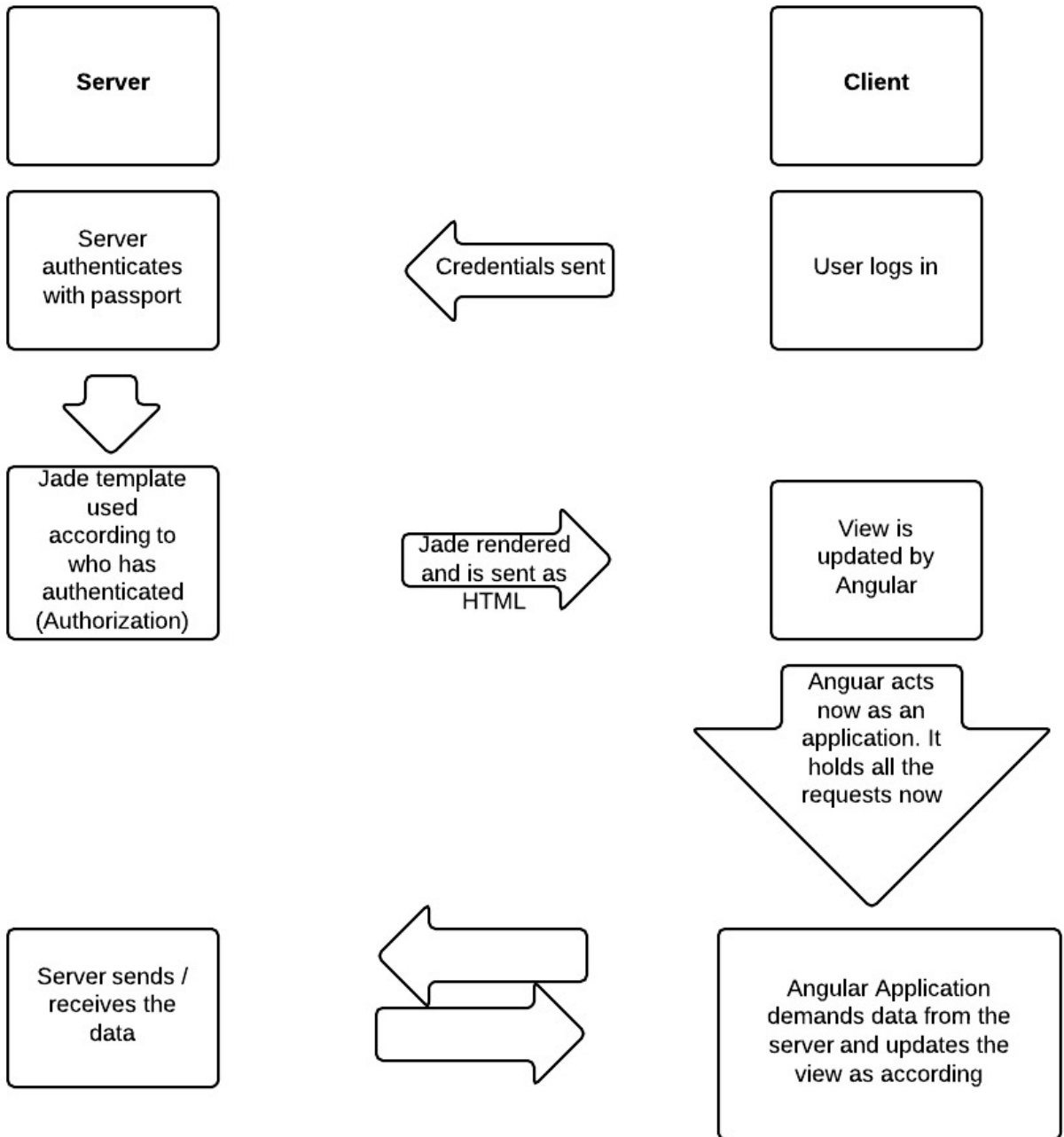
Table 1: CloudKibo Server Architecture

		Passport Google	Passport Facebook	Passport Windows	Passport Local	
	SendGrid	Passport				Passport-local-mongoose
Socket.IO	Express.JS (with Jade Templates)			Mongoose		
Node.js				MongoDB		

CloudKibo Application	ngAnimate	angularFileUpload	Socket.IO (Service)	KiboRTC (Service)	
	AngularJS				Adapter.JS

Browser	WebRTC	WebSockets
---------	--------	------------

Table 2: CloudKibo Client Architecture



Flow for Authentication (Server Side authentication)

3. Background

This software is built using the latest technologies like WebRTC, NodeJS, AngularJS, MongoDB, Socket.IO, Git, ExpressJS, Jade, SendGrid, OpenSSL, PhoneGap, Ionic and PassportJS. These are bleeding edge technologies and are soon going to replace PHP or ASP.Net and JQuery in future. Relational Databases sooner or later will be replaced by document-based schema-less No-Sql databases. With hybrid apps, we will be able to cover many mobile platforms with same code. It will be like code once and run on every platform.

All the communication will be **Server-less**. There will be no server-intervention during live call. Video will be sent directly to other peer. This is why we call it peer-to-peer calls. This way we ensure complete privacy for the users.

We have aimed for something big with this project. By creating a web-based communication platform, we have challenged all the existing communication platform technologies. While on other hand, we have created a whole online virtual school system (Interactive Learning System) which is unique and different from other such online learning systems in such a way that it is built on top of a communication platform.

We could have developed our learning system on top of some other communication technology but we chose to develop our own communication platform. The significance of this platform is that it can be integrated in any other web application built with AngularJS. We have made our communication platform an Angularjs service (kind of JavaScript library) which can be used in any web application.

We are using latest technologies which have bright future. They are going to replace most of the traditional technologies soon. With our project, we want to introduce these technologies to Pakistan. We are of opinion that Pakistan cannot compete with the west by learning what was implemented by the west decades ago. Pakistan should become the hub of leading edge emerging technology. Our sole motivation behind this project was to learn these technologies and subsequently introduce them to Pakistan.

Here is the list of latest and emerging technologies which we used in CloudKibo:

WebRTC:

Web Real-Time Communication (WebRTC) is a free open project that supports browser-to-browser applications for voice calling, video chat and P2P file sharing without plugins by using simple JavaScript and HTML5. The W3C draft of WebRTC is a work in progress with advanced implementations in the Chrome and Firefox browsers.

Node.js:

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. (Source: nodejs.org)

AngularJs:

AngularJS is an open-source web application framework, maintained by Google and community, which assists with creating single-page applications, one-page web applications that only require HTML, CSS and JavaScript on the client side. Its goal is to augment web applications with model–view–controller (MVC) capability, in an effort to make both development and testing easier. (Source: Wikipedia)

MongoDB:

MongoDB is an open source cross-platform document-oriented NoSQL database system. It favors JSON-like documents with dynamic schemas making the integration of data easier and faster.

Socket.IO:

Socket.IO is a JavaScript library which aims to make real-time web applications possible in every browser and mobile device. It has two parts: a client-side library that runs in the browser, and a server-side library for node.js.

Git:

Git is a distributed version control system and source code management (SCM) system designed to handle every kind of project with efficiency and speed.

ExpressJS:

Express is a minimal and flexible node.js web application framework, providing a robust set of features for building single and multi-page, and hybrid web applications.

Jade:

Jade is a server side templating engine designed for node.js. It produces XML like documents (HTML, RSS etc.) which are then ready to be sent towards client.

SendGrid API:

SendGrid provides a cloud-based email delivery service that assists businesses with transactional email management. It is a complete email delivery system with insights and statistics.

OpenSSL:

OpenSSL is an open-source implementation of the Secure Sockets Layer and Transport Layer Security protocols.

PhoneGap:

PhoneGap is a free and open source framework that allows you to create mobile apps using standardized web APIs for the platforms like Android, iOS and Windows Phone.

PassportJS:

Passport is an authentication middleware for Node.js. It is extremely flexible and modular, Passport can be unobtrusively dropped in to any Express-based Web application. It also supports, besides username and password, authentication with Facebook, Twitter and many more.

4. Hardware / Software tradeoffs

No tradeoffs.

5. Relationship with available past projects or standards e.g. IEEE, ANSI, ISO and etc.

Similar type of products already exists in the market. This will be different from them as it will be using the latest technology which enables real time communication without having need of downloading or installing any separate plugin or software. All other web-based communication tools use plugins or flash. This system will use latest and emerging technology WebRTC which will change dynamics of communication in future. It gives more sense of privacy to the users as all the communication among the users is completely **server-less** i.e. once the handshaking is done by server among parties, the server's role ends there and all the communication is handled on the client side. Everything during communication will be then based on peer-to-peer model. Even, we won't be able to record the video call sessions. We won't be able to know which messages or files are transferred among parties.

6. Patents, copyright and trademarks

Copyrights and patents will be property of CloudKibo.

Software / Hardware Design:

1. Overview

User will have to make an account in order to make classrooms, in case the user is teacher, or to join classrooms, in case the user is student. Students will be able to join the class and attend the video lecture being delivered by a remote teacher (using WebRTC broadcast technology). Conference call feature will also be provided as a feature encouraging online group study. Students will be able to upload the files and share them among one another. For this, all the users will be given limited storage space to store the stuff online.

Teacher will be able to schedule the classes and give the online video lecture. Teacher may also use the online storage space to store files or handouts for the students. Besides file sharing, file transferring in real-time during a call session is most appealing feature. You can transfer file up to any size you want. For testing purpose, we successfully transferred a 7 GB file from one computer to

other in a live call. Screen sharing feature will also be provided during live session so that any user whether student or teacher will be able to give presentations or perform other tasks.

2. Program Details

- a. CloudKibo Communication
 - i. Video Conferencing (N-N)
 - ii. Video call (1-1)
 - iii. Screen sharing (1-1)
 - iv. Screen sharing (N-N)
 - v. Instant Message (1-1)
 - vi. Instant Message (N-N)
 - vii. File Transfer (1-1)
 - viii. File Transfer(N-N)
- b. CloudKibo School
 - i. Student Dashboard
 - ii. Teacher Dashboard
 - iii. Course Search
 - iv. Create Courses
 - v. My Courses
 - vi. File Management System (Online File Storage)
 - vii. Class Announcements
 - viii. News Updates
 - ix. Student Grading
 - x. Teacher Rating
 - xi. Display Course
 - xii. Course File Management System
 - xiii. User Profiles
 - xiv. Teacher Profile
 - xv. Student Profile

Database:

Following information will be maintained in the backend databases.

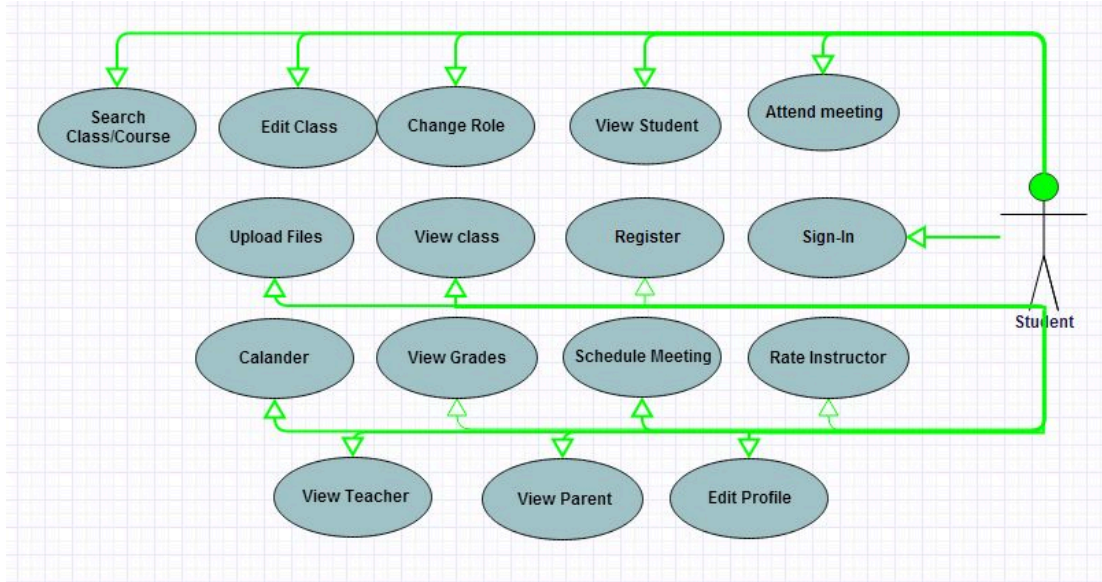
- Teacher related
 - Teacher's Profile
 - Classes or sessions teacher will be teaching
 - Teacher's available timings
 - Teacher's rating
- Student related
 - Student's profile
 - Classes student is enrolled in
 - Student Grading
- Classroom related
 - Teacher's information
 - No of students enrolled in the classroom
 - Information of the students enrolled

- Classroom timings
 - Class Announcements
 - Files stored in a class
- File Storage Statistics
- User local authentication
- User Facebook, Windows, Google authentication
- User Feedbacks
- News updates
- Call Statistics (not call recordings or data gathering from calls)

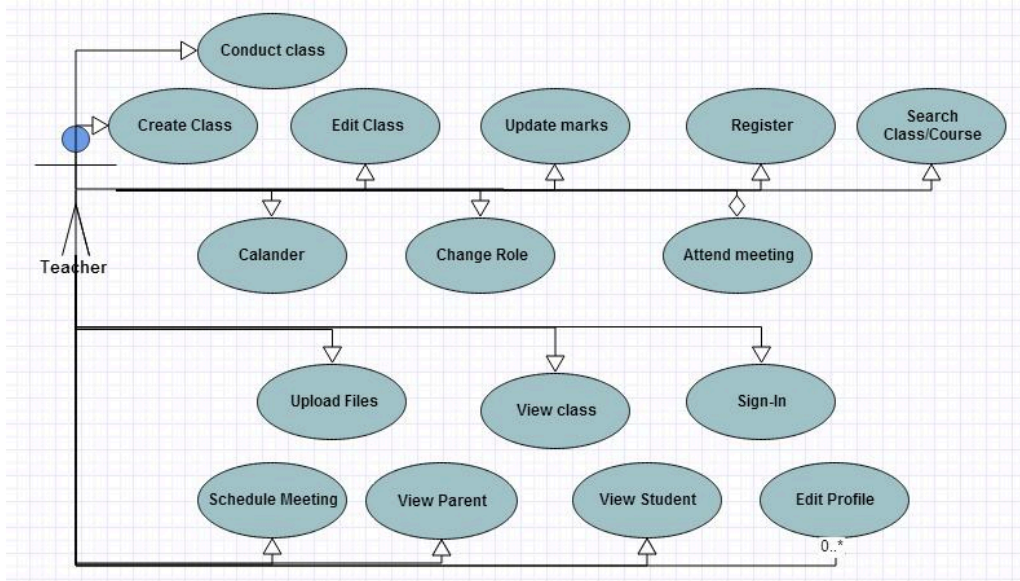
Use Case Diagram

Please include separate diagrams for each use cases (instructor, student, anonymous)

Student:



Teacher:



c. User interface

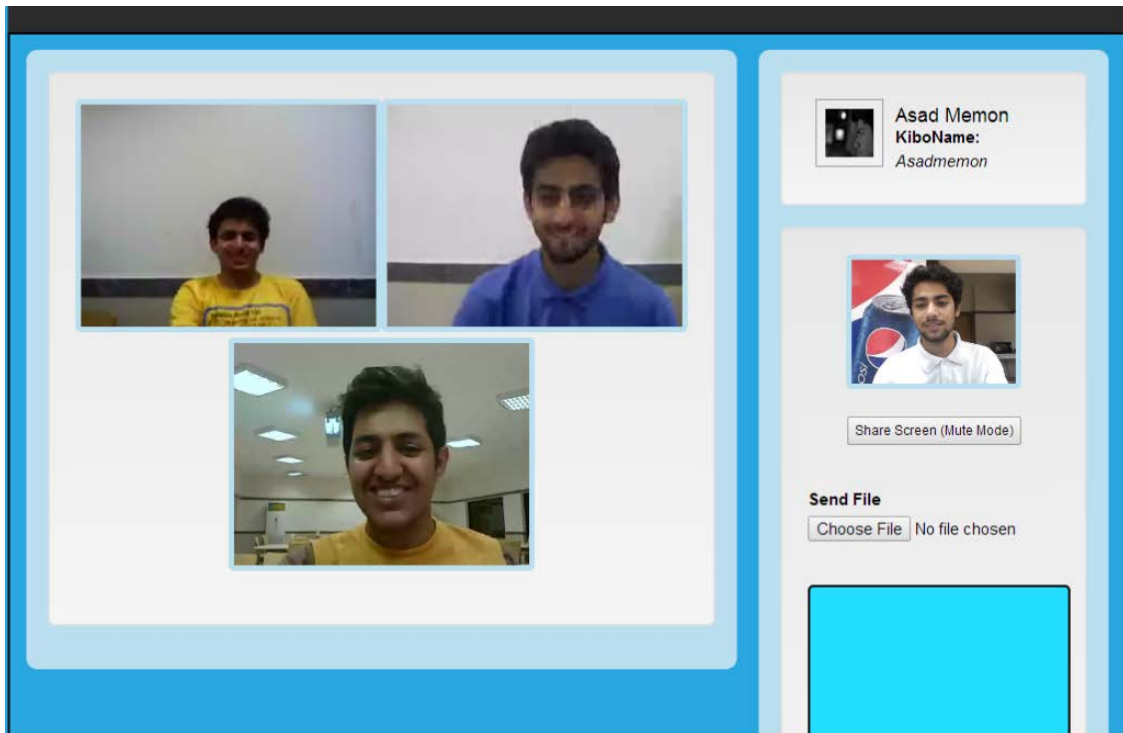


Figure 3 Conference Call, Screen sharing, file transferring and instant messaging features are visible

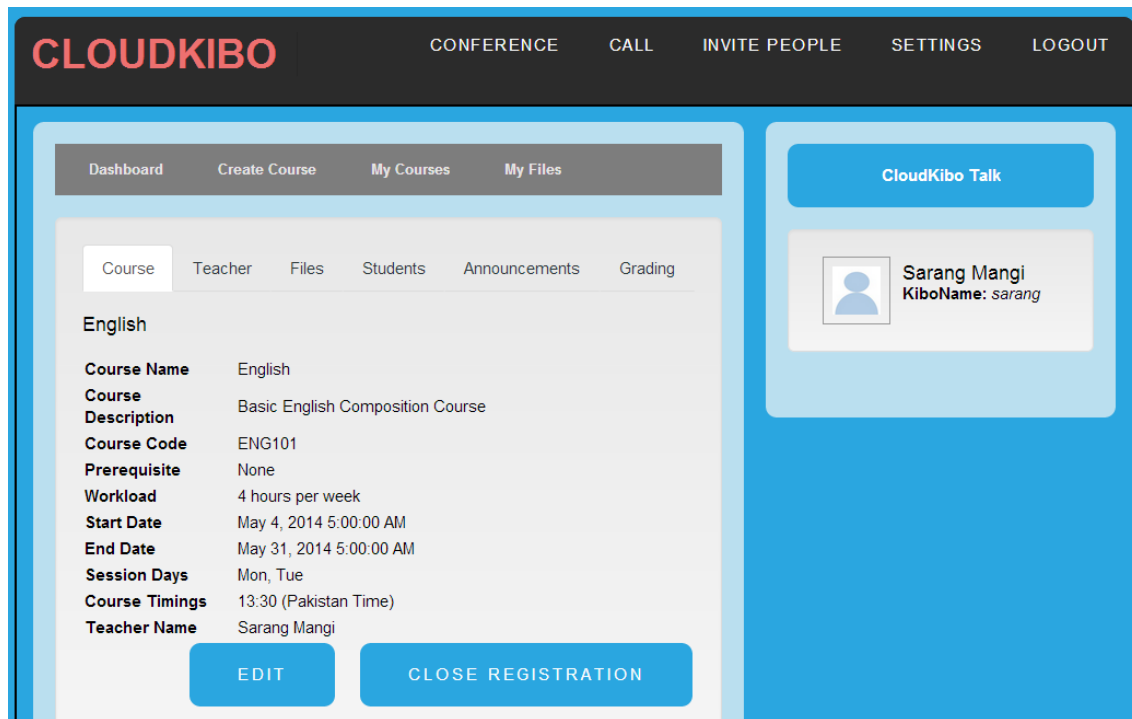


Figure 4 Course Display with File Management, Announcements, Students and Grading Options visible for teacher

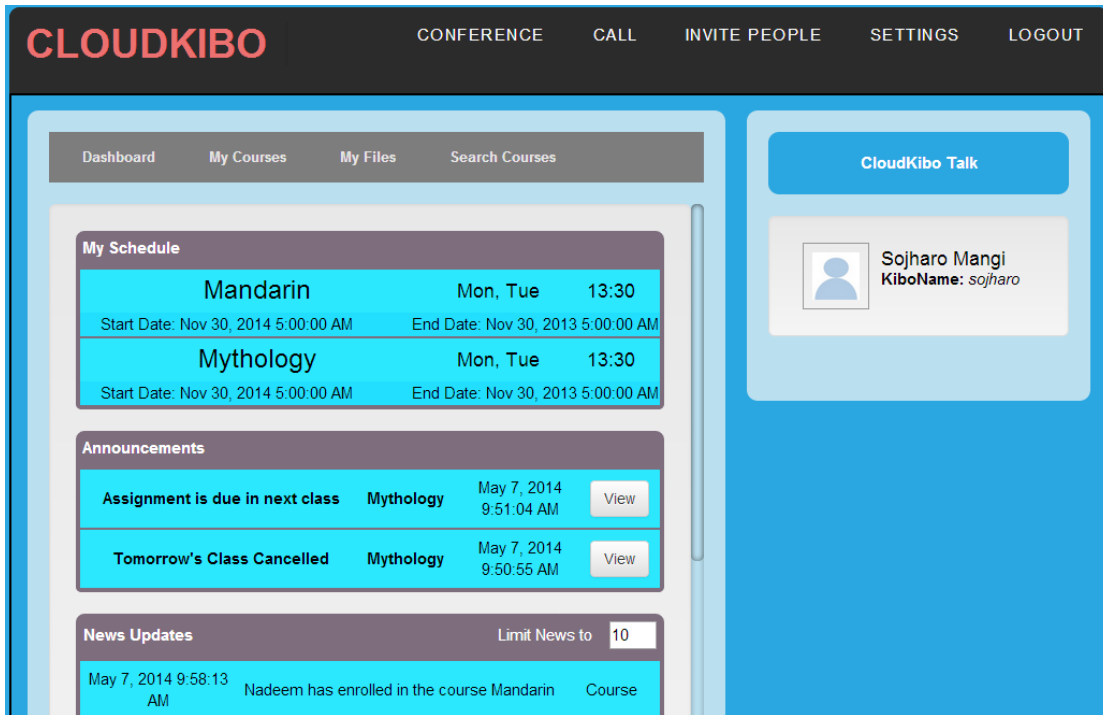


Figure 5 A typical dashboard for a student, Schedule, Announcements and News updates are visible

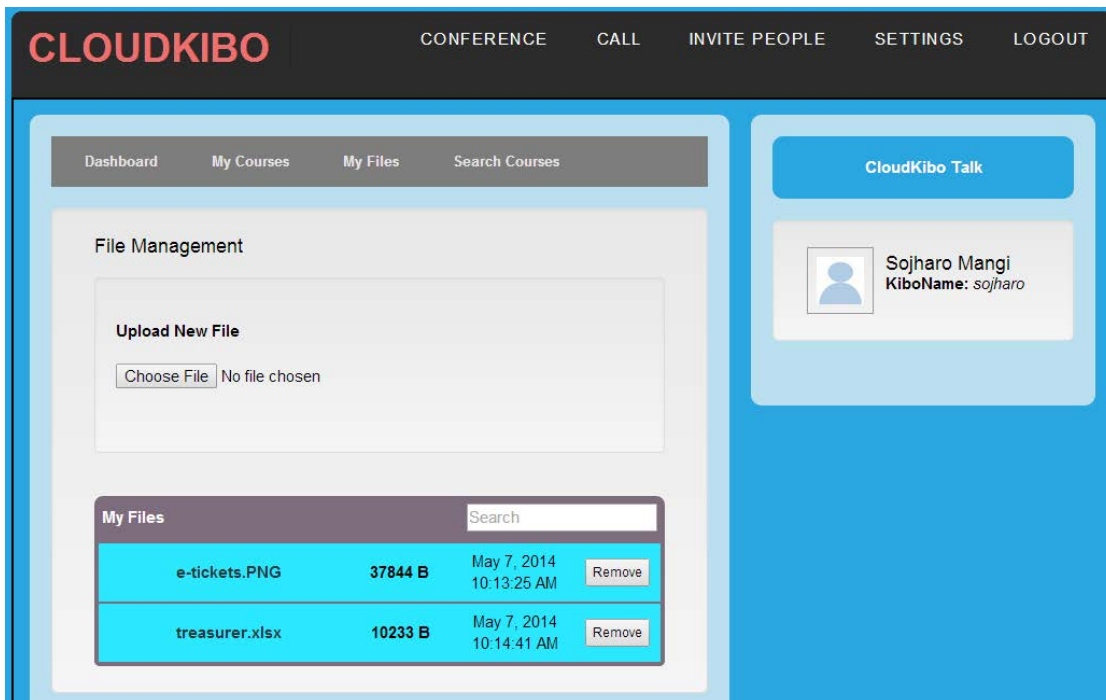


Figure 6 File management system for a user. User can upload and download personal documents here

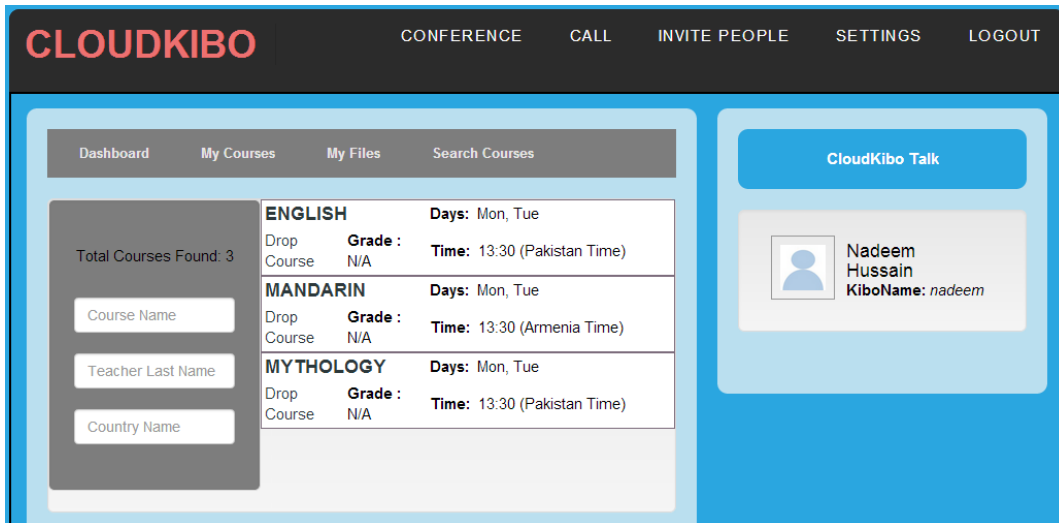


Figure 7 Courses enrolled by student

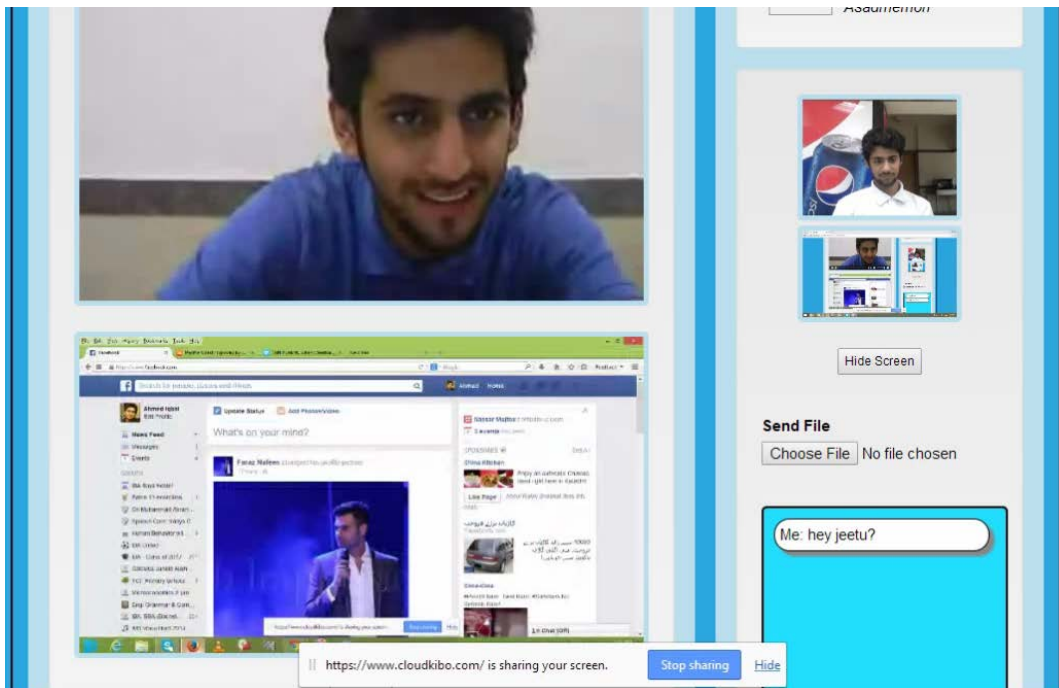


Figure 8 Screen Sharing during one-to-one call, instant messages box is also visible

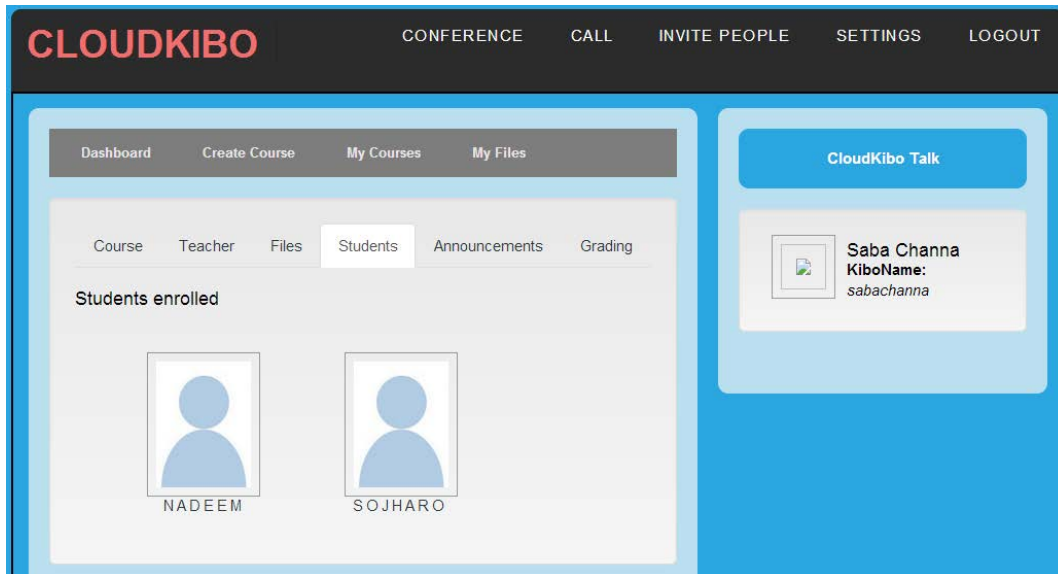


Figure 9 List of Students enrolled in a course

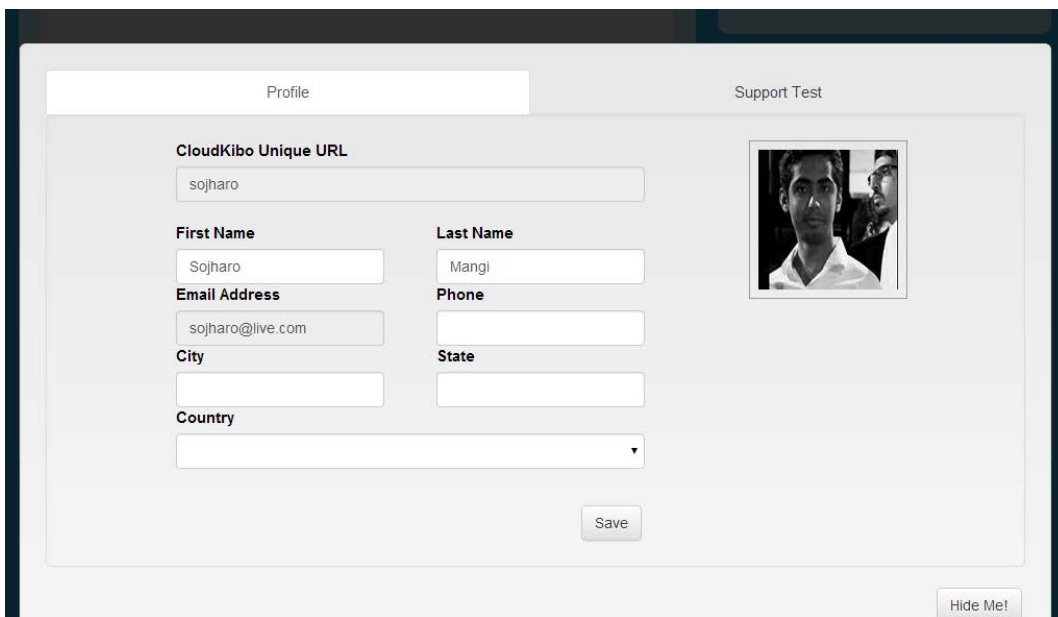


Figure 10 User Settings

d. Errors

- No audio is transmitted when screen is shared by a user in Conference Call. It is scheduled to be resolved soon by creating a separate channel for audio transmit. It is a minor fix.

e. Trails and tests

Trials have remained successful and also challenging at the same time. Testing has remained the most important task in the whole development. It has remained successful too.

3. Hardware Details

Not Applicable.

Results:

We have been able to do multi-party video calls in which users can share their screen to multiple users at a time. They can send files to multiple users and also send messages to each other. File transferring feature is faster than other file transferring features i.e. one provided by Skype. Transfer of video stream and computer stream is also faster. Reason for all this speed is our server-less peer-to-peer technology. Video does not go to server from user A in order to reach user B as is the case with Skype or Google Hangouts.

WebRTC, technology behind CloudKibo, is fully supported by Google Chrome. We came to know that Mozilla FireFox does not support screen sharing yet. It will support it soon in the future. Microsoft is anti-webrtc and Internet Explorer does not support WebRTC on purpose to save Skype.

In real-world, peer-to-peer call using WebRTC will not be simple as many devices are behind proxy servers or several layers of NAT. Some anti-virus software often block certain ports or protocols. CloudKibo uses ICE framework to overcome complexities of real-world networking. ICE first tries to make a connection using the host address obtained from a device's operating system and network card; if that fails (which it will for devices behind NATs) ICE obtains an external address using a STUN server, and if that fails, traffic is routed via a TURN relay server.

In other words:

- A STUN server is used to get an external network address.
- TURN servers are used to relay traffic if direct (peer to peer) connection fails. (Reference : <http://www.html5rocks.com/en/tutorials/webrtc/infrastructure/>)

Conclusions:

WebRTC is a giant leap towards new age of real time communications. It is so powerful that communication can be made internal and seamless part of the web applications. We have tried to make it possible by making communication platform an internal part of a Virtual School System. It has been successful and is better than any communication technology out there whether it is web-based or desktop-based.

CloudKibo can be spanned across devices and platforms. We have developed a prototype of Android Version of CloudKibo. Similarly, Iphone and Windows phone versions can also be developed for CloudKibo. The universal nature of WebRTC is such that we can even go beyond the browsers and applications and enable our system CloudKibo to support VoIP and Telephone Calls and messaging.

WebRTC is a promising technology and it is going to change the dynamics of communication models and infrastructure in future. CloudKibo is an endeavor to use, almost, all of the powerful features of this technology in order to prove its power. We have decided to continue this project and make it replace existing communication technologies and software.

Appendix:

Appendix 1: Equations

Not applicable.

Appendix 2: Code

Code is the property of CloudKibo.

Appendix 3: Schematic of your hardware

No hardware used. We have a cloud-based server purchased from DigitalOcean.

Appendix 4: Software/parts list

- a. CloudKibo Communication
 - i. Video Conferencing (N-N)
 - ii. Video call (1-1)
 - iii. Screen sharing (1-1)
 - iv. Screen sharing (N-N)
 - v. Instant Message (1-1)
 - vi. Instant Message (N-N)
 - vii. File Transfer (1-1)
 - viii. File Transfer(N-N)
- b. CloudKibo School
 - i. Student Dashboard
 - ii. Teacher Dashboard
 - iii. Course Search
 - iv. Create Courses
 - v. My Courses
 - vi. File Management System (Online File Storage)
 - vii. Class Announcements
 - viii. News Updates
 - ix. Student Grading
 - x. Teacher Rating
 - xi. Display Course
 - xii. Course File Management System
 - xiii. User Profiles
 - xiv. Teacher Profile
 - xv. Student Profile

Appendix 5: Work distribution

Work Distribution according to Asana Schedule

Nadeem Hussain's Tasks

- Make Administration backdoor login
- Implementation of User/Password authentication
- Implementation of Federate Authentication (Facebook)
- File storage statistics module/constraints
- Update profile / view / create profile
- Register a course, view schedule module
- Calender module
- Lost Password solutions and Email validation
- Android Version

Saba Channa's Tasks

- Designing Forms
- Implementation of Database
- Development of Workspace
- create / edit class modules
- Implement rating and grading systems
- Course search module
- File upload storage service
- Android Version

Sojharo's Tasks

- Upload on the real-world server
- Video/Audio/Text counselling (1-1)
- Conduct lecture / attend lecture (1-N)
- Group meeting (N-N) video/audio/text
- Real time file sharing / screen sharing
- Manage Instant Messages
- Other facilities & views for anonymous
- Android Version

Appendix 6: Project timeline

- Designing Forms
- Implementation of Database
- Implementation of User/Password authentication
- Implementation of Federate Authentication (Facebook)

- File upload storage service
- File storage statistics module/constraints
- Upload on the real-world server
- Development of Workspace
- Make Administration backdoor login
- Update profile / view / create profile
- Implement rating and grading systems
- Course search module
- Register a course, view schedule module
- Lost Password solutions and Email validation
- Calender module
- Video/Audio/Text counselling (1-1)
- Manage Instant Messages
- create / edit class modules
- Conduct lecture / attend lecture (1-N)
- Group meeting (N-N) video/audio/text
- Real time file sharing / screen sharing
- Other facilities & views for anonymous
- Android Version

Created with Microsoft Office OneNote 2007
One place for all your notes and information

References:

Books:

- Mastering Web Applications Development with AngularJs (Darwin, Kozlowski)
- AngularJS directives (Alex Wanston)
- AngularJS (Oreilly)
- MongoDB Crud Guide (Available on official website)
- NodeJs in Action (Mike Cantelon)
- Professional NodeJs (Pedro Teixeira)
- Smashing NodeJs (Guluirmo Rauch)
- Node Beginner Book (Manuel Keisling)
- Socket.IO: Real-Time Web Communication (Rohit Rai)
- The Definitive Guide to HTML5 WebSockets (Salim, Wang & Moskovits)3
- Getting Started with WebRTC (Rob Manson)
- *WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web, Third Edition*

Inspirations for code and designs:

- Learning of new bleeding edge technologies

- WebRTC, NodeJs, Express, Angular Js, Socket.IO
- Cloud Deployment, MongoDB, Jade
- GitHub, WireFrame, Asana
- SendGrid, SSL Security, PassportJS
- PhoneGap and Ionic Hybrid Mobile Apps framework
- Dealing with Open Source Packages

Papers:

None

Datasheets:

None

Vendor:

None

Background sites:

- www.webrtc.org/
- nodejs.org/
- expresjs.com
- angularjs.org
- socket.io
- www.github.com
- www.mongodb.org
- www.balsamiq.com
- www.asana.com
- www.digitaloceans.com

Acknowledgements:

We would like to thank Dr. Shakeel Khoja, our supervisor, and Jawaid Ekram, our client, for their support and trust in us.