Classroom teaching using Tablet PCs

Submitted in fulfillment of the requirements
of the degree of

Master of Technology

by

Neeraj Dhariwal
Roll No: 113050009

Under the guidance of:

Prof. D. B. Phatak

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai

2013
Abstract

Considering the increasing class sizes and rapidly improving technology, the use of handheld devices can add great value to classroom teaching. This project aims at introducing the use of tablet PCs in the classroom for the purpose of teaching and learning and to replace the traditional whiteboard based teaching system with this new system. This document describes a new classroom teaching system with the use of tablet PCs without using whiteboards. The idea is based on the growth of hand-held devices e.g. tablet PCs which can be used to make notes during a class by directly writing on screen with the help of a stylus pen. The system provides an alternative for both whiteboard and notebooks, as it facilitates the students to see what the teacher is teaching on their own tablet PCs at any point of time. At the same time, it allows them to make notes on their own screen on top of the teaching material received from the teacher. The project aims at enhancing the students’ ability to concentrate while attending classes by making them free from copying the contents of whiteboard in order to make overall classroom teaching more effective. It also attempts to reduce teachers’ overheads, e.g. taking attendance. It allows the teachers to monitor each student’s progress individually, and testing any students at any point of time by sharing his screen with the students.
Contents

1 Introduction 4

2 Related work 6
   2.1 Existing similar tools .......................... 7
       2.1.1 xCLASS ..................................... 7
       2.1.2 iTALC ....................................... 8
       2.1.3 Dokeos ...................................... 9
       2.1.4 Open-Sankore ................................ 9
       2.1.5 Virtual Network Computing (VNC) tools ........ 10
       2.2 Need for a new application ...................... 10

3 Architecture of the system 11
   3.1 Proposed features of the system .................. 11
       3.1.1 Registration and session management for teachers and students .. 11
       3.1.2 Teaching and note taking ....................... 11
       3.1.3 Monitoring ..................................... 12
       3.1.4 Sharing ......................................... 12
   3.2 System structure .................................. 13
       3.2.1 Client-server communication .................... 13
   3.3 Database design of the system .................... 15
       3.3.1 Description of the tables ...................... 15

4 Deployment of the system 19
   4.1 Writing on screen .................................. 20
       4.1.1 Scalable Vector Graphics (SVG) .................. 20
       4.1.2 HTML5 Canvas .................................. 20
   4.2 Capturing a browser tab ........................... 22
List of Figures

3.1 System structure .................................................. 13
3.2 Flow chart of a teacher ............................................. 14
3.3 Flow chart of a student ............................................. 17
3.4 Database design of the system ................................... 18
4.1 Block diagram of teacher’s screen broadcast .................. 27
Chapter 1

Introduction

With class sizes continuously increasing, the classical teaching system needs to be made adaptable to the current needs of teachers and students. In the current teaching system, teachers teach on a whiteboard and students record it in their notebooks. With students, coming from diverse academic backgrounds, effective communication between teachers and students is becoming very challenging. Teachers can’t always give personal attention to all the students and it becomes really difficult to track the progress of each student. In the classical teaching system that uses whiteboards, students get unengaged with the class and are busy in copying contents from the whiteboard most of the time. On the other hand, teachers find it difficult to monitor the activities of all the students in the class. At the time of class quizzes, the overhead in distributing, collecting and correcting the quiz papers of all the students is increasing continuously. This clearly indicates the need of a change in the education system in order to improve the in-class experience of teachers and students.

Technology is constantly improving and has made a great contribution in education by means of e-learning. But it has been used mainly in distant education programs. It’s also possible to introduce this technology in the field of classroom teaching by means of hand-held devices or PDAs. One reason for emphasizing hand-held devices is that laptops are not affordable to everyone. Furthermore, laptops with attractive slide making softwares can be used for teaching using presentations but they can’t provide user friendly note-taking features. Devices, e.g. Tablet PCs with a stylus pen can be extremely beneficial for making on the fly notes [1]. With tablet PCs being made available at low prices to educational institutes, the challenges pointed out in the first paragraph can be taken care of to a great extent. Students and teachers can carry their personal tablets in the classes and by setting up wireless networks within and across classrooms, teaching can be made much friendlier to both students and teachers.

This project aims at introducing teaching and learning using tablet PCs in classrooms. It
involves the development of a platform independent application, which can be used on tablet PCs to provide all the features available in the traditional whiteboard based teaching system with the goal of providing teachers and students a better environment for teaching and learning respectively. It should be noted that this system just needs initial costs. Once it is set up, it won’t require much expenses.

The rest of the document is structured as follows: the next section describes the work already done in this field and some existing open source applications which are used for similar purpose along with improvements which need to be done, then the architecture of the system is described, then the implementation and deployment of the system is described and in the end, the experiments and testing is described.
Chapter 2

Related work

Because of the portability and low cost of wireless mobile devices, they are considered to have great potential of becoming the primarily used devices in both in-class and distant learning. In [2], three different experiments, testing the usability of handheld devices are shown, along with the observations. In [3], study conducted in a large Canadian school is presented concluding that touch screen devices, e.g. iPADS, tablet PCs can perform better than pen and paper as a teaching tool. In [4], an in-class teaching system, with the use of Wireless Local Area Network is proposed. The presented system allows teachers to effectively teach in a class as well as to monitor the activities of any student at any point of time. It also allows teachers to control students’ access to internet. It has also proposed solutions to some of the problems that commonly occur while introducing hand-held devices into classroom teaching. It has pointed out some security related issues and suggested solutions for the same. The presented system was implemented and tested in Purdue University, West Lafayette. A similar collaborative learning system is presented in [5]. The system makes use of a server machine, tablet PCs which are connected through Wireless LAN spread throughout a college campus and a client-server software for realizing connection between the tablet PCs over the Wireless LAN. This system allows students to dynamically take notes on their tablet PCs and teachers to monitor students’ activities without having to move physically. The study material is transmitted from teacher to students and vice versa through the server machine. The presented system was tested with tablets loaded with Windows XP tablet edition with screen size of 12 inch, which were connected to each other via 802.11 a/g/b Wireless LAN. Appropriate number of Access Points were set up in each class depending upon the no. of students in the class. These Access Points are then connected to the central server via hubs. The system was tested in a physical class having 40 students. A questionnaire was given to all the students in order to take a general opinion about the presented system and the response was encouraging. Most of the participants
found this system more effective than the traditional whiteboard based teaching system. In [6], a cross-platform, web based application involving the use of iPADs and tablet PCs is presented. The objective of this application was to increase interaction among students in a classroom by enabling students who are not in close proximity to solve problems in collaboration. The basic version of the presented application allowed students to solve a flowchart exercise in group. After logging-in, each student is assigned to a group and every group solves an exercise together. In order to identify different students correctly, each students is represented by a different color. HTML5 Canvas was used to enable the students with different drawing features, e.g. drawing an arrow. The presented system was tested in a class of 12 students. The class was divided into 2 groups having 6 students each and in the end, most of the students found the system quite satisfactory. In [7], an experiment of teaching Data Structures using animations with the use of tablet PCs equipped with stylus pen is presented, believing that animation is a very effective way of teaching Computer Science concepts. Tablet PCs can also be used effectively in distributed learning programs. In [8], a case study is shown which involves the use of tablet PCs in synchronous distributed learning environment. The system was tested with distributed learning programs offered by Georgia Institute of Technology and the results indicate that using tablet PCs in distributed learning has several advantages over other modes of distributed learning.

2.1 Existing similar tools

E-learning has now become a highly researched and experimented area in the field of Computer Science. There are several software tools which exist for e-learning. Some of these applications can also be used for in-class teaching. There are many software tools which have been developed for classroom teaching and classroom management. A few of such tools are described below.

2.1.1 xCLASS

xCLASS is a multimedia classroom management software developed by Sun Tech. group. It can be used in a computer laboratory or a classroom for the purpose of teaching and classroom management. It enables students to learn in collaboration with other students and the teacher, by using their computers in a networked classroom. It enables teacher to broadcast their screen to all the students, such that all the students can see anything which runs on the teacher’s screen. It also has support for different teaching modes. In monitor mode, the teacher can monitor the screen of any student that he wants to monitor. In sharing mode, the teacher can share any
student’s screen with all the other students. xCLASS also provides features such as group discussion, quiz, lesson recording and support for wireless network.

Unfortunately, xCLASS doesn’t provide any additional features for devices equipped with touch screen features. Hence teaching using already prepared documents or Power Point presentations is possible using xCLASS, but it cannot be used as an alternative to whiteboard and notebook. Also, it can only be used with devices having good level of computation power, good amount of storage and additional support for graphics and sound. Hence it’s very difficult to use xCLASS with low cost tablet PCs.[9]

2.1.2 iTALC

iTALC (Intelligent Teaching And Learning with Computers) is an open source software program which was designed specifically to be used in computer classes. Using iTALC, a teacher can monitor the computer screen of all the students and can access any student’s screen if the latter needs any assistance. It also provides a demo mode, in which the teacher can allow any student to demonstrate something. In this mode, the screen of the selected student is transmitted live to all the other students. In addition to it, iTALC offers a few other features, e.g. teacher can lock the screen of any student at any point of time.

After starting iTALC on his computer, a teacher can create a classroom and add computers to that classroom by their IP address and MAC address. Ideally, all the computers should be in the same Local Area Network (LAN), but Virtual Private Network (VPN) can be used to be virtually present on that LAN. Teacher can see the list of all the logged in users and export this list to a file. He can configure iTALC master interface according to his choice. He can change the interval in which updates are sent to all the connected computers. He can also change the color depth of the updates by changing the quality of demo mode. He can switch his role between teacher, administrator and supporter. In the default overview mode, he can monitor all the connected computers. He can turn the demo mode on or off at any point of time. He can lock or unlock all the client computers at any time. He can send teaching instructions in form of text messages to all the computers. He can also remote login to any connected computer. He can also power down all the connected computers.

Currently, iTALC is only available for Linux and Windows computers. It is not available for Android tablets. Moreover, it was particularly designed for teaching courses involving computer programming, but it doesn’t provide any particular feature for touch screen devices. Hence it can’t be completely used as an alternative to a whiteboard.[10]
2.1.3 Dokeos

Dokeos is a free e-learning and classroom management application which offers many learning and collaboration features. Teachers can create courses, publish study material and manage courses. Students can download the study material easily. Teachers can teach distant students by means of flash based video conferencing. Teachers can encourage collaborative learning by means of forums, group discussions and chats. Dokeos also allows audio recording and putting it on top of powerpoint slides. In addition Dokeos text editor provides an optional support for including mathematical equations in a document.

Dokeos is a very light web-application which is built using PHP and MYSQL. After installing it on a server machine, it can be accessed from any web browser, hence it can be run from any device, regardless of it’s operating system. Being a web application also gives it cross-platform operability, e.g. different devices using different operating systems can access it simultaneously.

Like most of the other tools, Dokeos also emphasizes mainly e-learning and classroom management instead of actual in-class teaching. It doesn’t provide any feature for making notes on touch screen devices, neither can it be used as a virtual whiteboard.[11]

2.1.4 Open-Sankore

Open-Sankore is an open source software designed for teaching using interactive whiteboards. It provides users with the features of drawing on screen using mouse or stylus pen, entering text, importing various types of data including flash based animation, picture, audio, video and even pdf documents. In addition, it provides many supportive features such as changing color and line width, virtual laser pointer etc. The basic idea behind Open-Sankore was to project the screen of a computer having the program installed, on a large interactive whiteboard, which can be shared with all the students of a class.

Open-Sankore can be an extremely useful software when used with an interactive whiteboard. But without it, it merely serves as an alternative to a notebook. Although using it on teacher’s computer and projecting the teacher’s computer screen on any screen can allow teacher to teach students while sitting at one place, without having to approach the actual whiteboard every once in a while, it would be great if teacher’s screen can be transmitted live to all the students, so that the students don’t have to switch their eyes from the whiteboard to their notebooks and vice versa. Moreover, Open-Sankore is available for Windows, Linux and Mac Operating Systems, it is not yet available for Android tablets.[12]
2.1.5 Virtual Network Computing (VNC) tools

Virtual Network Computing (VNC) is a technique used for graphically controlling a remote computer and sharing the screen of a remote computer. It frequently transmits the state of one computer screen to another using Remote Frame Buffer protocol[13] over a network. It also transmits the keyboard and mouse events that take place on the source computer.

There are different implementations available for VNC, e.g. RealVNC, TightVNC, LibVNC. These programs have two parts, a VNC Server and a VNC Viewer. VNC Server is installed on the computer whose screen is to be shared. VNC Viewer is installed on all the machines that want to share the screen. Now VNC Server is started on the source machine on a particular port and all the viewers can access the server’s screen on that port.[14]

VNC programs are available for Android and hence can easily be used on Android tablets for in-class teaching, but if it’s used in in-class teaching by installing a VNC Server on teacher’s tablet and VNC viewers on students’ tablets, then it will only allow students to see what the teacher is teaching at a particular point of time. The students won’t be able to make their own notes on top of that. Hence, it can be perfectly used as a whiteboard, but not as a notebook.

Table 2.1 shows a comparative summary of all the tools mentioned above.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Can be used as a whiteboard?</th>
<th>Can be used as a notebook?</th>
<th>Provides drawing features?</th>
<th>Available for Android?</th>
</tr>
</thead>
<tbody>
<tr>
<td>xCLASS</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>iTALC</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Dokeos</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Open-Sankore</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>RealVNC</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 2.1: Comparison of existing tools

2.2 Need for a new application

All the tools described in the previous section fulfil some of the requirements of an effective in-class teaching and learning system, but none of them can be completely used as an in-class teaching tool. Some of them don’t support on-screen writing e.g. iTALC, VNC, while some provide the feature of on-screen writing but not the feature of transmitting the teaching data to other computers, e.g. Open-Sankore. Since, this project aims at developing a complete in-class teaching system, that supresses the need to use whiteboard and notebook, a platform independent system which looks like a combination of Open-Sankore and VNC is needed.
Chapter 3

Architecture of the system

3.1 Proposed features of the system

As stated in the earlier chapters, the project aims at producing a cross-platform in-class teaching and classroom management system that can serve both as a whiteboard and a notebook. According to it, some of the important features of the proposed system are described below.

3.1.1 Registration and session management for teachers and students

The system should keep the record of all the teachers, students and courses. Hence, it should allow registration of teachers and students and enrollment of students in different courses. In running state, the system should keep track of the present students at any point of time. For this, the system needs to maintain up-to-date session information for both teachers and students. Also, the system should automatically record the attendance of any student who logs in and enters a particular course so as to reduce teacher’s burden of taking attendance. The system should make sure that no student registers himself more than once in a single lecture. The system should enable any student to check his attendance in any course. It should allow teachers to check the attendance of all the students enrolled in any particular course, so that he can inform the students having less attendance than expected.

3.1.2 Teaching and note taking

This is the most important feature of the proposed system. It should present every teacher and student with an interface having features of basic drawing using stylus pen on a tablet PC. The system should also enable them to change the color and line width of drawing as they may want
to highlight something. As the teacher starts teaching (solving a problem) on his tablet (instead of whiteboard), teacher’s screen should be transmitted live to all the present students. Now all the students should be able to see the problem being solved on their own screens instead of the whiteboard. Moreover, they should be able to make their own notes on top of the live feed that they receive from the teacher. In that way, teacher won’t have to keep standing for the whole duration of a lecture, students won’t have to switch their eyes from whiteboard to their notebooks and they will have better organized class notes. After a page gets full, users should be able to save it as a file for future and a new blank page should appear. Users should also be able to load a previously saved page, if they want to solve or change an earlier solved problem.

3.1.3 Monitoring

In real world classes, many times a teacher gives the students a problem to solve. Students solve it and the teacher walks across the classroom to track the progress of any student he wants. In an effective teaching system, this should be automated. The teacher should be able to switch the mode from Teach to Monitor. In the monitor mode, the teacher should be presented with a list of all the students present in the class at that point of time. After the teacher enters the user id of the student he wants to monitor, the screen of that student should be visible on the tablet of teacher. The teacher should also be able to write comments or suggestions on the student’s screen from his own screen. This will reduce the teacher’s efforts in tracking the progress of every single student.

3.1.4 Sharing

This is another important feature of the proposed system. In real classes, teacher sometimes asks a student to solve a problem on the whiteboard. This should also be automated. In the proposed system, the teacher should be able to switch the mode to Share. After switching the mode, the teacher should again be presented with a list of students present in the class. After the teacher enters the user id of a student, the screen of that student should be visible on the tablets of all the students as well as the teacher. The teacher should also be able to write comments or suggestions on the screen of the selected student. The difference between monitor mode and share mode is that in the former, only teacher can view the selected student’s screen while in the latter, every present member of the class can view it.
3.2 System structure

The system is based on the simple client-server architecture. There is a central server on which the entire program and database are kept. Teacher and each student have a tablet. There are different clients for teacher and student. Clients are able to connect to the server via a wireless network set up in the classroom. The system structure is shown in figure 3.1.

![System structure diagram](image)

Figure 3.1: System structure

3.2.1 Client-server communication

There are different messages exchanged between clients and server over the period of a lecture. These message are as follows.

- At the beginning of a lecture, clients log in to the server. At this time, clients send their authentication information, i.e. username and password. After authenticating the clients, server sends them their respective home pages.

- Clients send the id of the course that they want to enter to the server. The server, after verifying the course id, sends them the current page of that course.

- In Teach mode, teacher continuously (e.g. once in every 250 ms) sends the teaching data to the server, which is then sent to all the students by the server.
• As soon as a student enters a course, he continuously (e.g. once in every 250 ms) sends a request to the server to know the current mode of operation. The server queries the database and sends the current mode (i.e. Teach, Monitor or Share)

• When the teacher switches to Monitor mode, the teacher stops sending teaching data, instead the selected student sends data (his screen) to the server and the server then sends it to the teacher.

• When the teacher switches to Share mode, the selected student sends data (i.e. his screen) to the server which the server then sends to all the other connected clients.

• At the end of the lecture, all the clients log out of the server and the communication terminates.

Figures 3.2 and 3.3 show the typical work-flow of a teacher and a student respectively.

![Flow chart of a teacher](image)

**Figure 3.2: Flow chart of a teacher**
3.3 Database design of the system

The records of all the teachers, students and courses as well as their mutual relationships are stored in the database. Figure 3.4 shows the database design of the system.

3.3.1 Description of the tables

**Teacher**

This table contains the Name, user id and password of all the teachers. Teacher’s user id is the primary key of this table. For security, passwords are not directly stored in the table. Instead, the md5 hash of the password is stored.

**Student**

This table contains the Name, user id and password of all the students. Student’s user id is the primary key of this table. Similar to table ‘Teacher’, in this table also md5 hash of the password is stored.

**Course**

In this table, the details of all the courses are stored. Each tuple of this table contains Course id, Course name, id of the teacher who teaches the course and the total number of lectures held of that course. Teacher’s id is a foreign key which refers to the primary key of table ‘Teacher’. Here maintaining referential integrity is very important, hence whenever a teacher is deregistered, in all the corresponding tuples of table ‘Course’, teacher’s id is set null.

**A table for each course**

There is a separate table for each course. Each tuple in this table contains the id of a student enrolled in that course, name of the student and the total number of lectures of the course attended by that student. The total number of lectures attended gets updated whenever the student enters a lecture of that course, making sure that no student can update his attendance by more than one in a single lecture.

**A table for each student and teacher**

There is a separate table for each student. This table contains the Course id and course name of all the courses the student is enrolled in. Similarly there is a separate table for each teacher.
which contains the Course id and name of all the courses that teacher teaches.

**Session info**

This table contains the details of all the clients who are present in the classroom at any point of time. Each tuple of this table includes user type (i.e. teacher, student), user id and name of the user. This information is used when the teacher wants to see a list of all the students present in the classroom for switching the mode to *Monitor* or *Share*. Whenever a teacher or student enter a course, his entry is recorded in this table.

**Current**

This table contains the information of course which is being taught in the class currently, the current mode of teaching, the id of student whom the teacher want’s to monitor (if the *Monitor* mode is active) and the id of student whom the teacher wants to share with (if the *Share* mode is active). Whenever a teacher enters a course or changes the teaching mode, he makes necessary changes in this table. All the students continuously check this table for the current teaching mode and act accordingly.
Figure 3.3: Flow chart of a student
### Figure 3.4: Database design of the system

<table>
<thead>
<tr>
<th>Teacher</th>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Id</td>
<td>*Id</td>
</tr>
<tr>
<td>*Name</td>
<td>*Name</td>
</tr>
<tr>
<td>*password</td>
<td>*password</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Course id</td>
</tr>
<tr>
<td>*Course name</td>
</tr>
<tr>
<td>*Teacher's id</td>
</tr>
<tr>
<td>*Total no. of lectures</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CS-101</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Student_id</td>
</tr>
<tr>
<td>*Student name</td>
</tr>
<tr>
<td>*Attendance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>abc</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Course_id</td>
</tr>
<tr>
<td>*Course name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>xyz</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Course_id</td>
</tr>
<tr>
<td>*Course name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Session_info</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>*user_type</td>
<td>*Course</td>
</tr>
<tr>
<td>*user_id</td>
<td>*Mode</td>
</tr>
<tr>
<td>*user_name</td>
<td>*Monitor</td>
</tr>
</tbody>
</table>

For each course

For each student

For each teacher
Chapter 4

Deployment of the system

As mentioned in the earlier chapters, the aim of this project is to develop an application which is platform independent. With such a constraint, an obvious choice is to develop a web application, which just needs a compatible web browser to run, irrespective of the device’s platform. The web server can be set up on the central server machine with adequate processing power and disk space. All the clients (teachers and students) can login to the web server from their individual tablets equipped with the recent version of a compatible mobile web browser.

The major part of the proposed teaching application is the live transmission of teacher’s screen to all the students. There are many VNC applications, discussed in chapter 2, but those applications transmit the entire screen of one machine to another one. This includes taking care of the position of mouse pointer and keystrokes. In the proposed application, transmitting the entire screen is not necessary. Instead, transmitting only the browser tab suffices so that the total data sent over the network can be reduced.

Another important part of the application is the teacher’s and students’ ability to write on screen with the help of a stylus pen so that tablets can be used as whiteboard and notebooks.

The entire application can be subdivided into smaller subproblems, which are listed below.

1. Writing on screen at teacher’s tablet.
2. Capturing teacher’s browser tab.
3. Sending the captured browser tab to all the students.
4. Rendering the received data (teacher’s tab) on the students’ tablets such that it is possible to write on top of it.

The following sections describe the solutions to all the subproblems listed above.
4.1 Writing on screen

For on-screen writing, teachers and students need the ability to draw random graphics on browser screen with the help of a stylus. There are mainly two ways of drawing graphics on a browser tab, which are described below.

4.1.1 Scalable Vector Graphics (SVG)

Scalable Vector Graphics (SVG) is an XML based vector graphics format, which can be used to draw graphics on web. The graphics drawn using SVG are resolution independent, hence it’s well suited in cross-platform environments where resolution may vary. SVG graphics define geometric primitives via Document Object Model (DOM) elements. Every SVG element is appended to the DOM of the web page and can be manipulated by JavaScript and CSS. Every element in an SVG graphic can be controlled using the JavaScript SVG DOM API. But since SVG has to maintain a separate reference for each element it adds to the DOM, its rendering becomes very slow when drawing complex graphics.

4.1.2 HTML5 Canvas

HTML5 Canvas is an HTML element that can be put anywhere in a web document. Everything drawn on a canvas is interpreted as a collection of pixels and hence the entire canvas is considered a single element of the web page. Any shape can be drawn on a web page using HTML5 canvas without much performance degradation. Hence, canvas is considered a better option as compared to SVG graphics for drawing complex graphics. But Canvas creates raster images which are resolution dependent, so whenever we try to increase the resolution of an image drawn on a canvas, the performance degrades. No references are kept for the shapes drawn on a canvas as the entire canvas is treated as a single object instead of multiple objects, hence individual elements of a canvas image can not be manipulated. Canvas drawings can be downloaded as .jpeg or .png images, which makes it easier to transmit it over a network.

Similar to other tags in HTML, e.g. `<img>`, `<p>`, the tag for including a canvas in a web page is `<canvas>`. The styling of canvas can be done in the `<canvas>` tag itself (as shown in the example below) or using a css style-sheet.

```html
<canvas id="mycanvas" width="400" height="400" style="background-color:#FF0000">

Canvas offers the programmers a drawing context which can be used to draw on individual pixels inside the canvas. To take the drawing context, the following javascript code is used.

```javascript
var canvas = document.getElementById("mycanvas");
```
var context = canvas.getContext("2d");

After taking the context, various shapes, e.g. line, rectangle, ellipse, circle, bezier curves etc. can be drawn on the canvas using javascript functions. draw color and line width can be changed. Any curve or any random shape can be drawn on the canvas using the following JavaScript methods.

- **context.beginPath():** This method is called whenever we want to initiate drawing a new path.
- **context.lineTo(x,y):** This method adds a new point having coordinates x,y to the current path and connects this point to the previous point on the path using a straight line.
- **context.moveTo(x,y):** This method moves the context to a new point having coordinates x,y without joining it to the previous point on the path.
- **context.stroke():** This method actually renders the drawn path using the current stroke styles, i.e. stroke color and line width.

**Drawing with pen on HTML5 Canvas in touch screen devices**

After taking the 2d context of a Canvas, any shape can be drawn on it. A Stylus pen can be used to draw random shapes on the canvas in devices with touch-screen support, while mouse can be used in the remaining devices. But since drawing with mouse or pen is entirely dynamic, it is quite different as compared to drawing any static figure on the canvas, e.g. a rectangle.

While drawing with pen, we need to monitor the status of the touch and the position of the pen at any point of time. For this purpose, there are Events in javascript. There are different events denoting different actions of keyboard, mouse and touch. In this project the following touch events take place[15].

- **touchstart:** On the occurrence of this event, a flag named isStart will be set, which will be used by other event listeners to figure out that the touch is currently active. Along with that, a path will begin, which may be drawn on the canvas.

- **touchmove:** This is the event that is actually responsible for drawing anything on screen. It examines the flag isStart, and if it’s set, it draws the required shape on the canvas. Also, in case of touch supportive devices, usually screen moves when a touchmove event occurs, but since in the project, touchmove is used as a command to draw a shape on screen, the default scrolling of screen has to be prevented.
- **touchend**: This event indicates that the touch has now been released and there is no need to draw anything on screen. Basically it unsets the *isStart* flag.

There are other events, e.g. *mouseup*, *mousedown* and *mousemove* which are used to draw on screen using mouse.

### 4.2 Capturing a browser tab

The next important step in broadcasting teacher’s screen to all the students is to capture it and convert it in a form which can be transmittable over a network. There are mainly two ways to achieve this which are described below.

#### 4.2.1 Creating a clone of the Document Object Model (DOM)

One way to capture any browser tab is to create a clone of the DOM of the web page and then convert it into a form which can be transmitted over a network. Hence it includes two steps which are as follows.

1. **Creating the clone**: The JavaScript method `document.documentElement.cloneNode(true)` duplicates the entire document tree and returns some HTML code reflecting the cloned web page.

2. **Converting the clone to a file**: Since the clone of a web page contains raw HTML code, it need to be converted to a file before to make it transferable over a network. JavaScript `Blob` method can be used to achieve this. The syntax is shown below.
   ```javascript
   var blob = new Blob([clone.outerHTML], type: 'text/html');
   ```
   The above method creates a new HTML file and appends the content of the clone to that file.

Though being a pretty straight forward way of capturing a browser tab, the above methods has a few drawbacks. It can only capture static content of a web page, not the input supplied by the user, i.e. if a web page contains a text box, this method will capture the text box, but not the text entered by the user in the text box. Similarly if a web page contains a canvas, this method can not capture anything drawn inside the canvas, which makes this method inadequate to be used with this project. Moreover, the frame rate achieved by this method is very low. Hence, it doesn’t give a smooth live transmission on the receiving end.
4.2.2 Capturing the Canvas:

As described in section 4.1.2, HTML5 Canvas renders anything drawn on it as a collection of pixels, it can be converted into an image. The following steps need to be followed to achieve this.

1. **canvas.toDataURL()**: This is a common JavaScript method to convert the entire content of a canvas to a DataURL. DataURL is a Uniform Resource Identifier (URI) scheme which allows internal objects, e.g. images in a web page in the form of external objects. It allows fetching all the objects in a web page in a single HTTP request rather than making individual request for each of the objects. Images are always inserted in web pages in form of DataURL. The `canvas.toDataURL()` method returns a string that represents an encoded URL containing the captured content of the canvas. This string can either be shown as text in a text box or shown as an image in another window.

2. **Converting DataURL to an image file**: Since DataURL is just a string representation of an image, it can be used to display the content of the canvas in a new window, but it can not be used to transfer it over a network, hence the DataURL needs to be converted into an image file. The following steps need to be followed to achieve this.

   (a) DataURL contains a header that denotes the file type. This header needs to be removed as it is not required in an image file.

   (b) DataURL is a string which is represented in ASCII string format by translating binary string to base-64 encoded string. Hence, before converting it to an image file, it needs to be decoded in binary form. JavaScript method `window.atob()` is used to achieve this.

   (c) The binary string resulting from `window.atob()` is a sequence of bits. This sequence is now converted to an array of unsigned 8 bit integers.

   (d) The array of unsigned integers created in the previous step is now appended to a Binary Large Object (BLOB). The JavaScript syntax for creating a BLOB and appending the array to it is shown below.

   ```javascript
   var bb = new WebKitBlobBuilder();
   bb.append(uintarray.buffer);
   ```

   where `uintarray` in the array of unsigned integers created earlier.

The output of the last step is a BLOB which can be sent over a network.
4.3 Transmission of the captured browser tab

After capturing the teacher’s browser tab (canvas), the next subproblem is to broadcast that captured frame to all the students. Initially, when all the clients log in to the server, a stateless HTTP connection is established between clients and the server. This connection is used for the transfer of HTML code between clients and server, but when real time communication is needed, this connection should not be used. Real time communication between client and server can be facilitated by one of the following methods.

4.3.1 HTTP polling

HTTP polling is the simplest and oldest method of continuous communication between client and server. It defines a polling interval. The client sends an HTTP request to the server once in the polling interval. The server responds with a new message if there is one, and with a blank message otherwise. As the polling interval expires, the client again sends a request and the sequence of request response messages continues. Hence, whenever updates are to be sent from the server to client with a fixed frequency, HTTP polling can be used.

The major problem with HTTP polling is that since it’s simple HTTP communication, it adds a HTTP header with each request and response. Hence there is a bulk of overhead involved in real time communication resulting in high network traffic and latency. Moreover, as the frequency of data transmission increases, the overhead further increases. Hence, HTTP polling is not considered a good option for real time data communication.

4.3.2 Long polling

In simple HTTP polling, the server sends an empty message in response to the client’s request if it doesn’t have any new message to send. This unnecessarily increases network traffic without any advantage. Long polling is an enhancement over polling which takes care of this. In long polling, the server doesn’t send an empty response to the client. Instead, it puts the client’s request on hold until a new message is available.

The problem with polling in real time situations remains intact with long polling. Hence, it also should not be used in real time communication with high frequency of updates.

4.3.3 Server sent events

In the previous two methods, the client needs to keep requesting the server to send updates with a predecided frequency. Server sent event is a server push technology, which allows the server
to push data to the client without requiring the client to make the initial request. Server sent events can be used to push frequent updates from the server to the client.

All the three methods described here can only be used to send updates from server to client. But in this project, sending real time data from teacher to student is required. Since teacher and student both are clients, a methodology to first send data from one client to server and then from server to all the other clients is needed.

### 4.3.4 WebSocket

WebSocket provides a substantial enhancement over other techniques in real time communication. WebSocket provides a full duplex bidirectional communication channel over a single TCP connection. A socket can be started at the server at a particular port. Then clients can connect to that socket and it can be programmed to receive or send data to the connected clients.

A WebSocket connection has two parts: handshake and data transfer. In the first part, handshake messages are exchanged between client and the WebSocket server for establishing a connection. At this time, the URL of the WebSocket server along with the port on which it is running, is specified. After that, data transfer can take place. The WebSocket server can be programmed to receive data and broadcast it to all the connected clients except the sender.

In [16], a comparative study done over a network spanning four countries between HTTP polling, long polling and WebSocket is shown. Results clearly indicate that WebSocket outperforms both polling and long polling in terms of latency by a significant margin.

Although, WebSocket is a recent development in the field of internet communication, the recent versions of all the popular web browser support it. The Default Android browser doesn’t support it yet, but latest mobile versions of Google Chrome, Firefox and Opera browsers support it. There are many server side implementations available for WebSocket in different languages. Some of them are *ws* in node.js, *pywebsocket* in python and *libwebsocket* in c/c++.

To use WebSocket, first of all, a suitable version of WebSocket server needs to be installed in the server machine and it needs to be set to listen to requests on a specific port. Now a connection request needs to be made to the server by each client who wants to connect to the socket. The syntax for doing it is shown below.[17]

```javascript
var ws = new WebSocket("ip address or DNS name of the server:port no. on which socket is running");
```

After the connection is made, a message can be sent to the socket by the following syntax.

```javascript
ws.onopen = function(){ws.send("message")};
```

Similarly, received message can be checked using the following syntax.
Although, the procedure described above is not enough to broadcast the data received on socket. For that, the server needs to be programmed appropriately.

### 4.4 Rendering the received data on the student’s tablets

The next subproblem is to render the received image data on each student’s tablet in such a way that the student can make his own notes on top of the teaching data received. As explained in section 4.1.2, HTML5 Canvas can again be used to make notes on tablet. Since the data received from the WebSocket in eventually image data, it can be converted into a form which can be rendered on a canvas. The process of accomplishing this is explained in the following steps.

1. The data received from the socket is in form of a binary image file. It cannot be directly displayed on a web page, since images are attached with a web page in dataURI format. So, the received image file (BLOb) needs to be converted into a dataURI. The JavaScript methods `window.URL.createObjectURL()` and `window.webkitURL.createObjectURL()` can be used to accomplish that.

2. The created object URL can now be drawn on top of a canvas. Because of that, it’s also possible to write on top of the received teaching data, since it is being rendered on top of a canvas, which is writable.

Sections 4.1 to 4.4 explain the entire process of teaching and broadcasting the teacher’s screen live to all the students. The process of screen broadcast is also summarized in figure 4.1
Figure 4.1: Block diagram of teacher’s screen broadcast
Chapter 5

Experiments and results

I have conducted some experiments on the amount of data that the system sends over network over time, the delay in capturing teacher’s screen over time and the average delay in rendering the received data on student’s tablets over time. I’ll put the results in this chapter as soon as I plot the graphs.
Bibliography


[9] xCLASS.
   http://www.suntechgroup.com/XCLASS/index.html

[10] iTALC.
    http://italc.sourceforge.net/documentation.php

    http://dev.dokeos.com/documentation/

    http://open-sankore.org/en


    http://www.html5rocks.com/en/tutorials/websockets/basics/