AN AJAX-BASED EVENT CALENDAR FOR A COURSE MANAGEMENT SYSTEM

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Kazi Suraiya Tulip
Spring 2012
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Kazi Suraiya Tulip:

An AJAX-Based Event Calendar for a Course Management System

Subrata Bhattacharjee, Chair
Department of Mechanical Engineering

Carl Eckberg
Department of Computer Science

Christopher Paolini
Computational Science Research Center

12/8/2011
Approval Date
Copyright © 2012

by

Kazi Suraiya Tulip

All Rights Reserved
DEDICATION

To my husband Galib for his unlimited support, my friend Sajia, and my family who gave the biggest support and encouragement in reaching my goal in life.
ABSTRACT OF THE THESIS

An AJAX-Based Event Calendar for a Course Management System
by
Kazi Suraiya Tulip
Master of Science in Computer Science
San Diego State University, 2012

ClassTA is a Course Management System based on AJAX technology that is used as a Web-based teaching assistant. ClassTA is a very secure, responsive, and scalable system, which allows it to scale to large deployments for thousands of students. Besides having the features like Quiz, Mail System, Roster, Help system, Calendar; it also has activity modules like Blog, and Friend finder which can help its users to build a very interactive learning community. The Drag and Drop feature also gives it the look and feel of a desktop application.

The existing calendar does not provide much functionality besides showing the date and printing. For example, in Mail, Blog, Create Class. In this work, a Calendar Management System named “Event Calendar” is developed to provide much needed functionalities of creating, maintaining and posting events. It allows instructors to add, edit, and delete events and share them with students. The events are categorized and color coded based on specific types which allow users to easily visualize the events by just seeing the calendar. Adding new events and finding existing ones are as easy and fast as a simple click on the specific date on the calendar.

The Software Development Methodology used to develop Event Calendar is AJAX. XHTML, CSS is used to build and maintain the appearance of the Webpage and JSON to fetch and exchange event objects between client and server sides. JavaScript is used to dynamically add and modify events, and Java to interact with the server. All the events created and modified are stored and fetched from tables in a MySQL server. In this thesis paper, a brief discussion of the history of computerized calendars is presented following the use and build of the Ajax-based Event Calendar feature for ClassTA.
TABLE OF CONTENTS

ABSTRACT ...............................................................................................................................v
LIST OF TABLES ................................................................................................................. viii
LIST OF FIGURES ........................................................................................................... ix
ACKNOWLEDGEMENTS ................................................................................................. xi

CHAPTER
1 INTRODUCTION .............................................................................................................1
  1.1 History of Course Management System .............................................................1
  1.2 Comparative Study between Blackboard and Moodle.................................3
  1.3 History of Electronic Calendar Software .......................................................4
2 CLASSTA AND IT’S FEATURES ...........................................................................6
  2.1 Account Management .................................................................................6
  2.2 Class Management ......................................................................................7
  2.3 Desktop .........................................................................................................8
  2.4 Roster ...........................................................................................................8
  2.5 Quiz Management .......................................................................................9
  2.6 Help Service ................................................................................................9
  2.7 Friend Finder ...............................................................................................11
3 TECHNOLOGIES USED .........................................................................................12
  3.1 AJAX .........................................................................................................12
    3.1.1 XHTML ..............................................................................................14
    3.1.2 CSS ..................................................................................................15
    3.1.3 JavaScript ........................................................................................16
    3.1.4 XMLHttpRequest ..........................................................................16
    3.1.5 DOM .................................................................................................17
    3.1.6 MySQL .............................................................................................18
  3.2 JSON ..........................................................................................................19
  3.3 NetBeans 6.7 .............................................................................................20
3.4 GUI ...........................................................................................................................................20

4 DESIGN AND IMPLEMENTATION OF CLASSTA CALENDAR MANAGEMENT SYSTEM ........................................................................................................................................23

4.1 Comparative Study of Calendar Management System .........................................................23
4.2 System Architecture ...........................................................................................................26
4.3 Database Architecture .......................................................................................................26
4.4 Layout ...................................................................................................................................28
  4.4.1 ClassTA Homepage ...........................................................................................................29
  4.4.2 Instruction Panel .............................................................................................................30
  4.4.3 Calendar .........................................................................................................................30
  4.4.4 Event Editor Panel ..........................................................................................................31
4.5 Features for Faculty ...........................................................................................................31
  4.5.1 Add Event .......................................................................................................................31
  4.5.2 Delete Event ...................................................................................................................33
  4.5.3 Edit Event .......................................................................................................................34
  4.5.4 View Event .....................................................................................................................37
  4.5.5 Starred Event ..................................................................................................................39
4.6 Features for Students .........................................................................................................39

5 FUTURE ENHANCEMENTS & CONCLUSION ........................................................................42

5.1 Future Enhancements ...........................................................................................................42
5.2 Conclusion .............................................................................................................................42

REFERENCES ...................................................................................................................................43
LIST OF TABLES

Table 1.1. Student Satisfaction Scores.......................................................................................3
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>ClassTA homepage</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>ClassTA sign-in screenshot</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>ClassTA, create class screenshot</td>
<td>8</td>
</tr>
<tr>
<td>2.4</td>
<td>ClassTA, joining new class</td>
<td>8</td>
</tr>
<tr>
<td>2.5</td>
<td>ClassTA desktop</td>
<td>9</td>
</tr>
<tr>
<td>2.6</td>
<td>ClassTA, roster</td>
<td>9</td>
</tr>
<tr>
<td>2.7</td>
<td>ClassTA gradable</td>
<td>10</td>
</tr>
<tr>
<td>2.8</td>
<td>ClassTA help</td>
<td>10</td>
</tr>
<tr>
<td>2.9</td>
<td>ClassTA friend finder</td>
<td>11</td>
</tr>
<tr>
<td>3.1</td>
<td>Classic web application model (left) compared with an AJAX model (right)</td>
<td>13</td>
</tr>
<tr>
<td>3.2</td>
<td>CSS property and value</td>
<td>15</td>
</tr>
<tr>
<td>3.3</td>
<td>Hierarchy of objects in DOM</td>
<td>18</td>
</tr>
<tr>
<td>3.4</td>
<td>JSON array representation</td>
<td>19</td>
</tr>
<tr>
<td>3.5</td>
<td>JSON object representation</td>
<td>20</td>
</tr>
<tr>
<td>3.6</td>
<td>Subversion commit</td>
<td>21</td>
</tr>
<tr>
<td>4.1</td>
<td>Moodle calendar icon on left, Blackboard calendar icon in middle. ClassTA</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>calendar icon on right</td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>ClassTA new event, Blackboard new event</td>
<td>24</td>
</tr>
<tr>
<td>4.3</td>
<td>ClassTA calendar instructions</td>
<td>25</td>
</tr>
<tr>
<td>4.4</td>
<td>Moodle old events</td>
<td>26</td>
</tr>
<tr>
<td>4.5</td>
<td>System architecture of event calendar</td>
<td>27</td>
</tr>
<tr>
<td>4.6</td>
<td>UML diagram for CalEvent</td>
<td>27</td>
</tr>
<tr>
<td>4.7</td>
<td>ClassTA calendar in homepage</td>
<td>29</td>
</tr>
<tr>
<td>4.8</td>
<td>Event calendar homepage</td>
<td>29</td>
</tr>
<tr>
<td>4.9</td>
<td>Event calendar instruction panel</td>
<td>30</td>
</tr>
<tr>
<td>4.10</td>
<td>Calendar inside event calendar homepage</td>
<td>30</td>
</tr>
<tr>
<td>4.11</td>
<td>Event calendar event panel</td>
<td>31</td>
</tr>
</tbody>
</table>
Figure 4.12. ClassTA event panel ................................................................. 32
Figure 4.13. Add new event button .............................................................. 32
Figure 4.14. Add old event message ............................................................ 33
Figure 4.15. Add old event message in panel ............................................. 33
Figure 4.16. Delete button inside edit window ......................................... 34
Figure 4.17. Edit button inside detail view window .................................. 34
Figure 4.18. Edit window ......................................................................... 35
Figure 4.19. Select class drop down menu .............................................. 35
Figure 4.20. Roster drop down menu ....................................................... 36
Figure 4.21. Type of event ....................................................................... 36
Figure 4.22. Create new event type .......................................................... 36
Figure 4.23. Edit window prompt .............................................................. 37
Figure 4.24. Edit and detail window ......................................................... 38
Figure 4.25. Disabled edit button for old events ...................................... 38
Figure 4.26. Detail view window ............................................................... 39
Figure 4.27. Highlighted event ................................................................. 39
Figure 4.28. Starred event ...................................................................... 40
Figure 4.29. Student event calendar ....................................................... 40
Figure 4.30. Student detail view ............................................................... 41
ACKNOWLEDGEMENTS

This dissertation would not have been completed without the encouragement and support of many individuals. On this occasion, I extend my heartiest gratitude to all of them. I would like to express my sincere gratitude to my thesis advisor, Professor Subrata Bhattacharjee for his motivation, and guidance. I am thankful to the members of my thesis committee: Professor Christopher Paolini and Professor Carl Eckberg for their time and support.
CHAPTER 1

INTRODUCTION

The Web has become one of the most integral parts of modern technologies. Many usage of it in modern life has made it an indispensible part of everyday life. Email, social networking, business, education and many other fields now rely upon this technology and numerous Web applications have evolved to meet these various needs. Online auctions, web mail, online retail sales, and wikis are just to name a few of the common and popular web applications. To add to the list, Course Management System (CMS) is also an integral part that is used in majority of the universities and is becoming an essential part for education. The evolution of information technologies has paved the way for Course Management System which is a component of e-learning. A Course Management System (CMS) is a software program or integrated platform that contains a series of Web based tools to support a number of activities and course management procedures [1].

1.1 HISTORY OF COURSE MANAGEMENT SYSTEM

According to the paper “Peer Comparison of Course/Learning Management Systems, Course Materials Life Cycle, and Related Costs” by Massachusetts Institute of Technology, Course Management System “provides the platform for the enterprise’s online learning environment by enabling the management, delivery and tracking of online and blended learning” [2]. CMS systems developed nowadays can be categorized into 4 categories; (a) commercial, e.g., Blackboard, (b) open-source, e.g., Segue, (c) developed "in-house" at a particular institution, or (d) community source such as Sakai. With the increase of its usage and added functionalities implemented with latest technology, Course Management Systems have become “ubiquitous” and have “evolved from simple HTML one-way communication tools” to a “multifunctional, enterprise level applications” on our campuses [3].

The history of Course Management System can be dated as early as 1960 when PLATO (Programmed Logic for Automatic Teaching Operations) was introduced [4]. Developed by University of Illinois, PLATO was the first generalized “computer assisted
instruction” system which functioned for four decades offering coursework to elementary university, local schools and university students [5]. For the last 40 years, CMS has evolved and became one of the crucial parts of education. Many institutions nowadays develop CMS systems. Blackboard, WebCT, Moodle, Desire2Learn, and Sakai Project are the few important Course Management Systems that are widely used.

CEO Michael Chasen and chairman Matthew Pittinsky founded Blackboard in 1997 which became a public company in 2004. As of December 2010, Blackboard is been used by over 9300 institutions in over 60 countries and considered “a leader in the development of Internet-based education software” [6]. The learning management system comprise four modules: learning system, course delivery and management, content management system, and a system to record and analyze student assessment results. In 1997, WebCT was developed by Murray Goldberg, a Computer Science faculty member of the University of British Columbia. In February 2006 WebCT was acquired by Blackboard. “WebCT's Vista and Campus Edition Course Management Systems continued to be offered following Blackboard's merger with WebCT; however as of April 2011 Blackboard had begun phasing these products out.” [7].

Moodle (Modular Object-Oriented Dynamic Learning Environment) was developed by Martin Dougiamas and is a free source learning software built to help educators create interactive online courses. It has continued to evolve since 1999 and the latest stable release came in Oct 10, 2011. Many of its features include assignment submission and grading, Wiki, online calendar, Moodle instant messages etc. [8]. Desire2Learn (also known as D2L) is also a provider of enterprise eLearning solutions and develops online Learning Management Systems (LMS) used for schools, colleges, universities, virtual schools, governments, associations, and organizations around the world and was founded in 1999 by President and CEO John Baker and Anil Sabharwal. [9]. Unlike the Course Management Systems mentioned above, the Sakai project was developed by collaboration among academic institutions, commercial organizations, and individuals. In 2004, January, Sakai Project was formed from several college and university projects. In addition to the course management features, Sakai is intended as a collaborative tool for research and group projects [10].
1.2 Comparative Study between Blackboard and Moodle

A comparative study has been performed among 135 students in Fall 2008 and Spring 2009 in University of South Florida. The two major Course Management Systems; Blackboard and Moodle, were compared by the participants who were randomly divided into two groups. Features that were compared are; Announcements, Course Documents, Assignment manager, Discussion Board, Collaboration Tools, Communications, and My Grades. One of the feature’s comparison results is explained below and after that the final cumulative result is given.

One of the most used modules in online courses is Announcement (in Blackboard) or News (in Moodle) module. The main purpose of this feature is to help instructors keep students informed and updated about the course. In Blackboard, the homepage, by default, generates this module, while in Moodle this is automatically created for each course and for the front page. In both Blackboard and Moodle, only administrators and faculty have full access to post new events and edit/modify the existing ones. As shown in the Table 1.1 [6], the student’s satisfaction score is provided. SD stands for standard deviation.

<table>
<thead>
<tr>
<th>Table 1.1. Student Satisfaction Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Student Satisfaction Scores on Announcement Module for BlackBoard and Moodle</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>BlackBoard</td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>3.6</td>
</tr>
<tr>
<td>3.6</td>
</tr>
<tr>
<td>3.6</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

At the end of the study, the result indicated that students slightly preferred Moodle over Blackboard. Almost every module or function comparison that was made, Moodle was favored by course participants over Blackboard with the exception of the Discussion Board module [6].
1.3 HISTORY OF ELECTRONIC CALENDAR SOFTWARE

An electronic calendar provides users with an electronic version of a calendar which is an extension of many features provided for time and task management software. Electronic calendar is a standard feature for many PDAs (Personal Digital Assistants), EDAs (Enterprise Digital Assistants), and smart phones. The uses of electronic calendars have a long history in human-computer interaction literature. In 1982, Kelley and Chapanis interviewed 23 professionals and found that calendars were indispensable. In order to minimize the problems rose from using paper calendar and to get the most out of using calendars like, archiving, editing, and portable access; electronic calendar could be a great replacement [11]. Since mid-80’s, many products came to meet the need for electronic calendar and “transformed the process of scheduling, by easing the process of recording and providing the ability to share events” [12].

In 1979, Alfred B. Levine patented the invention of the electronic calendar which had a simple interface of storing future events, and also view, and delete them as need basis. The importance of electronic calendar rose with the introduction of groupware software. Groupware software aims towards allowing group of individuals to share and work in a collaborative environment. The first groupware application, WordPerfect Library, which was later named Novell GroupWise, allowed shared contact and calendar management between users. Later PDA’s included more functionality in electronic calendar of transferring events between other devices and also synchronizing with desktop-based groupware applications. [12].

Nowadays, capabilities of electronic calendars are very common as new Web-based services and hybrid devices like mobile phones are introduced. The synchronization between multiple devices and online storage systems allow users to access their calendar from anywhere. In most modern applications, a notable feature seen is resource sharing and social networking. These features are also incorporated in many modern electronic calendar applications. Many social calendar system like iCalendar, Google Calendar, and 30 Boxes embody Web 2.0 features, such as tagging, ‘hack ability’ and rich, AJAX interfaces [12].

One Course Management System developed at SDSU is ClassTA. It is Ajax based and is very fast and responsive. The drag and drop file management system, grade histogram, session synchronization, wikis, student group management all these features have made
ClassTA a very powerful and interactive teaching and learning tool. An existing calendar was used for ClassTA whose basic functionalities included viewing dates and inserting dates to Mail, Blog, and New Course, and also in other features. These functionalities were very limited and to extend its ability for users to create and share events a new Calendar Management System has been developed which is named “Event Calendar”. A brief description of ClassTA, the Calendar Management System, and several of their features will be explained in the following chapter.
CHAPTER 2

CLASSTA AND IT’S FEATURES

ClassTA is a Course Management System developed and used in SDSU to support thousands of students and teachers to make the teaching and learning experience easier and very interactive. Based on the four categories mentioned in the paper “Peer Comparison of Course/Learning Management Systems, Course Materials Life Cycle, and Related Costs”, ClassTA can be put in the category c; developed "in-house" at a particular institution [2]. Being Ajax-based, ClassTA can send data and also retrieve data from the back-end server asynchronously and without any interference to the current page; thus no disturbance is felt by the user. This stand out ability has given ClassTA users to fetch and send data faster than many other Web applications. In addition to providing the major features of a Course Management System like Quiz, Roster, Calendar, and Blog, ClassTA has allowed users to get familiarized with the system by providing Help service, participate more interactively by finding study mates using Friend Finder service, or even share questions, and concerns using the Blog system [13]. ClassTA has been able to stand out in the CMS world due to its rich features and easy to use functionalities. Figure 2.1 shows the ClassTA homepage. Brief descriptions of some of its features are presented in the following sub-sections.

2.1 ACCOUNT MANAGEMENT

As shown in Figure 2.2, each faculty and student of ClassTA will have a unique account. When the account is created for the first time user provides the username and password. In order to access, use, and view the features of ClassTA, having an account is mandatory.

After logging into ClassTA, a user can change or view user’s detailed information in the Settings screen. ClassTA also provides a password recovery system in case of a forgotten password. Each faculty will also have a sample student and TA account by in order to allow faculty to test before deploying a gradable which is the quiz management system of ClassTA.
2.2 CLASS MANAGEMENT

Faculty has the authority to create a class and maintain its content. Figure 2.3 shows the form where faculty can create a class giving it a name, a small description, and the end date for the class.

As shown in Figure 2.4, a new class contains new roster, new resource, and a class blog. The settings for the class can be changed from the class settings. After creating a class, the roster needs to be created and discussion blog needs to be activated. To access a class a student will need the add code from the faculty.
The ClassTA desktop is different depending on the type of user account. As shown in Figure 2.5, a faculty has a “Create new Class” icon and “A folder for archived classes” which will contain all the classes created by the faculty.

2.4 ROSTER

All student records registered under a class are stored under the class roster showing the name, grade, email address, and other information of each student as shown in Figure 2.6.
Using the Mail User Agent (MUA), professors can communicate with students. The grades and individual scores are also displayed in the roster. The histogram of the grades is stored in Roster, which gives an overview of grade statistics of the students in a roster.

### 2.5 Quiz Management

One of the main features of ClassTA is the Quiz Management System which includes a rich quiz builder interface and question bank management system. Quiz Management System allows instructors to access and reuse questions that were previously created. Different question and quiz types are supported by this system. Figure 2.7 shows the first step of Gradable – Quiz Management System where a faculty can create a new question or import existing ones from Question Bank.

### 2.6 Help Service

To accustom users of ClassTA with its rich functionalities and new features, the help system has been introduced which provides detailed information of a content requested by
user as shown in Figure 2.8. It can also be edited by distinguished user and the wiki enriched feature also allows its users to add images, videos or just simply edit or add contents in it by providing a rich user interface. All the functionalities give the help system a “wiki” enriched, dynamically generated, context sensitive help service provider for ClassTA. A new rating service is under development in Help Service that will allow the user to rate each page based on its usability.
2.7 FRIEND FINDER

The Friend Finder service feature is added to integrate the concept of social networking among students within a class in ClassTA. As shown in Figure 2.9, Friend Finder lets students fill up certain criteria in an interactive form to search for study mate or project mate within a class. The importance level of each criterion can also be set by the student. The view of Friend Finder page with search results for friends that matches certain criterion is shown.

![Figure 2.9. ClassTA friend finder.](image)
CHAPTER 3

TECHNOLOGIES USED

Most Course Management Systems are Web based using platforms like Microsoft .Net, Java/J2EE, or PHP and use database like MySQL, Oracle, or Microsoft SQL Server [7]. In earlier versions of Web applications, a synchronous page load was done when a user needs to wait after a request is being sent to the server to load a webpage. After the request is processed by the server, the requested data is sent back to the client and the entire page needed to be loaded to reflect the change. This entire process is considered very inefficient and time consuming. A group of Web technologies is used to make the communication with the server asynchronous and faster by not interfering with the current state of the page but only loading the specific content that needs to reflect the change. This entire process has made Web browsing much faster and easier. The term AJAX (Asynchronous JavaScript and XML) is used to represent this new technology. A brief description of the tools and technologies used in building ClassTA is as follows.

3.1 AJAX

In the book, “CSS, DHTML & AJAX”, J. Teague stated that AJAX use the collection of existing technologies like HTML, JavaScript, CSS, and DOM. The development of AJAX happened within the web development community over several years. AJAX is:

- Asynchronous – user can perform other functions while browser is waiting to get information back from the server.
- Technologies- Can be scripted with JavaScript. AJAX uses XHTML for content and CSS for presentation, DOM and JavaScript for dynamic content display.
- XML or JSON both data formats can be used to create and transfer content between server and webpage. [14]

In the article “AJAX: A New Approach to Web Applications”, J. James defined AJAX as:

- Standards-based presentation using XHTML and CSS;
- Dynamic display and interaction using the Document Object Model;
- Data interchange and manipulation using XML and XSLT;
Asynchronous data retrieval using XMLHttpRequest;
JavaScript binding everything together.

As mentioned above, AJAX is not a new kind of programming language, rather it introduces a new approach of using existing technologies to exchange data and update only the selective part that has been requested without updating an entire Webpage. In Figure 3.1 [15], instead of loading the entire webpage as seen in the classic web application model, the AJAX engine is introduced by the browser that renders the interface and also communicates with the server. In the classic model, a user action generates a HTTP request, but in the AJAX model a JavaScript call is made to a Web Service and the remote server handles the action itself if it does not require any response by server. In case of actions that require server response, such as retrieving new data, submitting data etc. the engine makes those requests asynchronously usually using XML thus making it faster for the end user [15].

Many big names in the field of web technology are using AJAX; for example Orkut, Gmail, Google Maps, Google Suggest, Google Groups all are powered by the AJAX technology. Many Flickr features and also Amazon’s A9.com search engine also use similar techniques [15]. A brief description of the technologies used for AJAX is as follows.

3.1.1 XHTML

XHTML, an acronym for "eXtensible Hypertext Markup Language" is a “family of XML markup language” that is a reformulation of HTML 4.0 as an XML 1.0 application. It is identical to HTML 4.0.1 and is a W3C recommendation as of January 2006 [16].

HTML has been merged and modularized as an XML (Extensible Markup Language) with its new name XHTML. XML compliant languages deliver information that can be processed by many different communication technologies that have emerged since digital communication revolution in Web. XHTML is HTML reformulated to adhere to the XML standard. Many features have been deprecated in XHTML; for example, the tag <center>. The World Wide Web Consortium (W3C) was formed to define the standards for HTML and later XHTML.

XHTML define the syntax and placement embedded directions that define how to display the contents of the document including text, images, and other support media. It can also provide hypertext link in order to connect one document with the another. XHTML supports features of HTML 4.01 and use rules of XML. The main focus of XHTML is the structure of the document; for example, section headers, structured lists, rules, titles etc. Some common features of XHTML are:

- Embedded tags; for example, the <i> tag following the word Hello tells the browser to display the text in italics. Most tags have a start and end tag.
- HTML skeleton: HTML document starts and ends with <html> and </html>. Except for the <html>, <head>, <body>, and <title> tags, the XHTML has few other required structured elements
- Comments: start tag <!--, and ending tag -->
- Text: The bulk of content that the user sees is the text.
- Multimedia: XHTML includes references to the multimedia elements via special tags which browser uses to load into document [16].
As of September 2009, XHTML5 is undergoing development as part of HTML5. First proposed by Opera Software, it is the fifth revision of HTML standard and aims to improve the language with “support for the latest multimedia while keeping it readable by users and compatible by computers”. After development it can be written in either HTML or XHTML thus making it a single markup language. Many features of HTML5 enable it to be a candidate for cross-platform mobile applications for smartphones and tablets [17].

3.1.2 CSS

CSS, an acronym for Cascading Style Sheets, allows defining the style properties of any element on the page. Earlier browsers supported CSS Level 1 and CSS-P; but CSS Level 2.1 is supported by most browsers. The standard for CSS is also defined by W3C. Although CSS works with XHTML, it is a separate code that enhances the abilities of XHTML by allowing redefining the way the existing tags display their content [14].

The three most common selectors or way to select the HTML to which a style applies are:

- HTML selector: The HTML elements name is used as a selector to redefine the tag.
- Class selector: Class selector that can be applied to any HTML tag.
- ID selector: Id’s are similar to classes, but can be assigned to any HTML tag, but only once for given page on a particular HTML tag.

As shown in Figure 3.2 [18], the selector is the HTML element that needs styling and each declaration consists of a property and a value. The property is the styles attribute that needs to be changed and each property has a value [18].

![CSS property and value](http://www.w3schools.com/css/css_intro.asp)

**Figure 3.2. CSS property and value. Source:** W3SCHOOLS, CSS introduction. W3schools, http://www.w3schools.com/css/css_intro.asp, accessed November 2011, n.d.

There are three places where CSS rules can be put:

- In an HTML tag
• In the head of a document
• In an external document [14].

3.1.3 JavaScript

JavaScript is an interpreted language rather than a compiled language which means as the browser runs the JavaScript it passes the script to an interpreter which converts the script into a machine code that is understandable to the computer. It first became available in Netscape Navigator 2 and was called LiveScript. After that Microsoft added Jscript which is their own version of JavaScript to Internet Explorer. Since then both Netscape and Microsoft released their newer version of the JavaScript and included them in their latest browsers [19].

One of the main reasons for choosing JavaScript is its widespread use and availability. The most common use of JavaScript is interacting with the users, fetching information from them, and validating their actions. There are three types of data in JavaScript:

• Numerical data: Integer numbers such as 145, and fractional numbers such as 1.234
• Text data: Another term for one or more characters is a string. JavaScript texts are enclosed inside double quote (""”) or single quote (‘’’). There are also some special characters: like, \b represents backspace, \n means new line, etc.
• Boolean data: The Boolean type has two different values: true and false.

Variables in JavaScript begin with the var keyword which allows the browser reserve memory for the value that would be stored in the variable. Each of the JavaScript objects (date and Array) have a collection of related properties and methods that are used to manipulate a certain kind of data. For example, the Array object consists of properties and methods to manipulate arrays [20].

3.1.4 XMLHttpRequest

All AJAX solutions require a method for passing data between the webpage in the client side and the files in the server side that are used to create the webpage. Earlier IFrames were used for this kind of back and forth communication between the client and the server. Nowadays, in many AJAX applications, the XMLHttpRequest object is used that can be used in JavaScript to retrieve data from a server. IFrames still require an entire page to be fetched from the server. The XMLHttpRequest object was created by Microsoft to be used with
Outlook as an ActiveX object. Since version 5 it has been part of the Windows version of Internet Explorer. Firefox, Safari, and Opera also support compatible version of XMLHttpRequest object. During mid-2006 W3C published a working draft that documents the common feature between XMLHttp of Microsoft and XMLHttpRequest.

DHTML + XMLHttpRequest + Server Files = Ajax
Filters data Transfer data between Provides data that is either for display client and server static or dynamically generated

XMLHttpRequest Calls should always be under asynchronous mode. The readyState reflects the status of current point in a call’s lifecycle. When object is born its 0, open() returns 1, and the point becomes 4 when the response in back. The user can work on other things while waiting for a response back. XMLHttpRequest fires `readystatechange` events. A callback function can be declared using the `onreadystatechange` field. The callback will then receive all state changes [14].

3.1.5 DOM

The W3C define the Document Object Model as “a platform and language neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page” [14]. The DOM has three levels as follows:

- Core DOM: standard model for structured document
- XML DOM: defines the “objects and properties of all document elements, and methods to access them”
- HTML DOM: a W3C standard that is platform and language independent and a standard object model and interface for HTML. The entire HTML document is considered as a document node and viewed as a tree structure called node-tree. Each HTML element is an element node. Text, attribute, and comments are considered text nodes, attribute nodes, and comment nodes respectively. All nodes can be accessed and manipulated from the tree by JavaScript or other programming language.

As shown in Figure 3.3 [21]:

- in a node tree, the top node is called the root
- every node, except the root, has exactly one parent node
- a node can have any number of children
a leaf is a node with no children
siblings are nodes with the same parent [21]

W3C realized the need to link scripting language to objects on a webpage. The browser manufacturers like Netscape and Internet Explorer introduced their own DOM and W3C released its standardized DOM in late 1998. The Netscape layer DOM lets write scripts to control elements created with the <layer> tag and elements created with CSS positioning. The Internet Explorer all DOM allows writing scripts that can access any element on the screen which includes CSS properties [14].

### 3.1.6 MySQL

MySQL is a relational database management system in where database stores data in separate tables rather than putting all the data in one big storeroom. This helps adding flexibility. The SQL stands for Structured Query Language. It is the most common structured language used to access database and is defined by the ANSI/ISO SQL Standard. MySQL software is open source and uses the GPL (GNU General Public License). MySQL Database Software is a client server system that consists of a multi-threaded SQL server supporting different backend, different client programs and libraries, administrative tools, and a wide range of APIs. MySQL was released in 1995 by MySQL AB. Some features unique to MySQL are its use of multiple storage engines, commit grouping, and unsigned Integer
values. MySQL supports MyISAM, InnoDB, BDB, and other storage engines. Some features of MySQL:

- Written in C and C++
- Works on many different platform
- Tested with different compilers
- Provides transactional and non-transactional storage engines
- API’s for C, C++, Java, Perl and many other languages
- A privilege and password system that is very flexible and secure and allows host based verification [22].

### 3.2 JSON

JSON (JavaScript object notation) is a lightweight data interchange format and are human readable and easy to parse and generated by machines. It is language independent and built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence. [23]

The basic types for JSON's are:

- Number
- String - a sequence of zero or more Unicode characters, wrapped in double quotes, using backslash escapes
- Boolean (true or false)
- Array - an ordered collection of values. An array begins with [ (left bracket) and ends with ] (right bracket). Values are separated by (comma). [23]

Figure 3.4 [23] and 3.5 [23] shows the JSON array representation and JSON object representation respectively.

![Figure 3.4. JSON array representation. Source: JSON. Introducing JSON. JSON, http://www.json.org/, accessed November 2011, n.d.](array.png)
3.3 NetBeans 6.7

It is based on a subset of the JavaScript, Standard ECMA-262 3rd Edition - December 1999. Yahoo and Google started offering some of its web services in JSON in 2005 and 2006 respectively [23].

Netbeans IDE is an integrated development environment for JAVA, JavaScript, PHP, Python, Groovy, C, C++ and others. It is written in JAVA and cross platform compatible with Windows, Mac OS, Linux, and Solaris. For JAVA development purposes, a JDK is needed. The platform allows applications to be developed in software components called modules.

Subversion can be integrated in Netbeans IDE which is an open source version control system. Figure 3.6 [24] shows the commit prompt for subversion. It provides features like

- Full version history is provided for renamed, moved or removed files.
- Commit operations are atomic, meaning that a collection of modifications either enter the repository completely or, in the event of connection failure, not at all.
- Versioning of project metadata is provided [24].

3.4 GUI

The software development life cycle has evolved over the year from traditional waterfall model to the structured approach and now the structured rapid prototyping approach which is derived from the structured development life cycle with few changes. The fundamental concepts of rapid prototyping can readily be applied to GUI development. This approach consists of rapid analysis, prototyping, designing, tuning, and testing.

- Rapid Analysis – Requirements and user task model are evolved iteratively as the user provides their input.
• Prototyping – In GUI application development, the initial prototype will produce a GUI with presentation and behavior, along with corresponding application functions.

• Design – The detailed design is derived from the user approved prototype. For GUI applications, a user interface prototype evolves in the design activity according to the result of rapid analysis.

• Tuning – It involves testing and making changes to the prototype if critical functions reveals performance shortfalls.

• Testing – Actual data and user actions are used to test the prototype throughout the life cycle as it iteratively evolves.

In the book “Object-Oriented GUI Application Development”, Geoff Lee has presented an integrated object oriented life cycle approach for a GUI application. The steps for GUI design are as follows:

• Preliminary Requirements Specification:
  • Objectives- Objective and the requirements needs to be specifies in this phase
  • Features – The features need to be specified which will affect overall software design and implementation effort
  • Constraints – It specifies the hardware and software environment, standard compliance, and other performance related issues. Estimating this helps lower development effort and cost.
• GUI Application Analysis – Four types of analysis is important for GUI application development: user requirement analysis, user analysis, user task analysis, and object-oriented analysis. Task analysis decomposes user tasks into goals and object oriented analysis constructs the object model for the application.

• GUI Design - GUI design phase involves designing and evaluating user interface, prioritizing and grouping of objects, and also visual cues and contextual messages are added to further refine the GUI.

• GUI Prototyping – Results of design activity guide the creation of prototype that contains user interface presentation, behavior, visual cues, and contextual messages.

• GUI Evaluation – As the users work with the prototype and evaluate the effectiveness of its GUI, the feedback leads to improvements of GUI.

• GUI Application integration – After being developed separately, the final integration of the GUI subsystem with the application will generate a fully functional GUI application [25].

A scientist and futurist Vannever Bush originated GUI in his classic 1945 article “As We May Think”. There he introduced “Memex”, an information administration tool that would allow data to be stored on microfilm and made easily accessible, linkable with hyperlinks and programmable. In 1963 Ivan Sutherland, a graduate student at MIT, developed a program for his Ph.D. dissertation called Sketchpad, which allowed the direct manipulation of graphic objects on a CRT screen using a light pen. Douglas Engelbart unveiled the first primitive mouse in 1950’s. PARC’s (Palo Alto Research Center) at Xerox developed the first usable GUI Alto computer in 1974. Steve Jobs, the co-founder of Apple Computer, was inspired by PARC innovations and the first commercially successful GUI was the Apple Macintosh. Microsoft’s first Operation system that incorporated GUI was developed in 1983. The high quality GUI was launched with Windows 95 in the year 1995 [26].
CHAPTER 4

DESIGN AND IMPLEMENTATION OF CLASSTA
CALENDAR MANAGEMENT SYSTEM

ClassTA has an existing calendar before event calendar was developed, which was useful to view dates and select dates from calendar when editing email, blog, or creating a new class. These limited features of the calendar were not of much help when users of ClassTA wanted to view if there are any existing events in calendar. This initiated the need of a Calendar Management System. The main purpose of the system is to provide ClassTA users a single place where all events for every class are viewable and based on user status editable. After providing a detailed analysis of the need and importance of Calendar Management System, the design and implementation are discussed in the following sections.

4.1 COMPARATIVE STUDY OF CALENDAR MANAGEMENT SYSTEM

In order to evaluate the Calendar Management System for ClassTA, some of its features are compared with the well-established, widely-used event calendars. Google, Blackboard, and Moodle Calendar features have been compared and contrasted with the newly build Calendar Management System for ClassTA.

1. Integrated: The Events in ClassTA are all integrated in one place and events from all courses can easily be viewed by a simple click on the calendar icon when the user logs in. Both the Blackboard and Moodle Calendar have the calendar integrated and events shown in one place too [27]. As shown in Figure 4.1 [27, 28], the homepage of ClassTA, Blackboard and Moodle have a calendar link that links to the event calendar page where users can view all the events for all courses and other occasions in one integrated place [28].

2. Same window: In Blackboard calendar, trying to insert a new event takes user to a new page leaving the original calendar and other information. User might wish to see information of other events and also copy some information from other event into the new event. As shown in Figure 4.2 [28], ClassTA event calendar allow user to stay in the same page to perform all activities. So user has the advantage to roam around the other events and other features needed without abandoning anything. Add, edit, delete occur in one window so user have view of others.


Moodle Calendar takes user to a new page when new event is created or event is clicked for detail information, it allows users to view other events in the calendar when user hover over the date, but it does not allow copying other events information into the new event which ClassTA event calendar does.
3. Color code like Google calendar, ClassTA calendar color code each event depending on its type and code information is shown next to the calendar as shown in Figure 4.3. Blackboard has same image set out throughout the calendar where events are marked. No events can be distinguished from the other as the images are same in the Blackboard calendar. ClassTA calendar not only color code the events differently based on its type, it also highlights today’s date and gray out weekends so user can easily view and understand what represents what.

![Figure 4.3. ClassTA calendar instructions.](image)

Moodle calendar also have color code for event types and, its color codes are in type Global, Group, Course, and User events. ClassTA calendar on the other hand differentiate events based on Quiz, Announcement, Class Cancellation etc. which are pretty much related to courses and based on faculty student communication.

4. Old and new events-In the event panel in ClassTA, the older events have a gray background which completely separate them from the new ones as shown in Figure 4.4 [27]. This gives users the ability to view and distinguish old and new events just by looking at them and the different colors. Moodle calendar gives user the option to view upcoming events by selecting from the dropdown menu. Google calendar gives user the option to dim past events.

5. Notification- Because ClassTA calendar functions are performed in the same page, user needs to get notified when a new event is created, and event has been modified or deleted. ClassTA calendar shows messages when an action has been performed and also warn user with a popup alert when some error has been done. Also, the event panel automatically highlights the last event the user has worked on by highlighting it which always keeps the user up to date on his actions. These makes ClassTA calendar very user friendly and responsive.

6. AJAX-being Ajax based, functions in ClassTA calendar can be performed in the same page without refreshing the entire page. So when user add, edit or delete a new event, he can still view the calendar without waiting for the previous action to be completed. This makes ClassTA calendar faster and very responsive. Other calendars require the loading a new page when a new event is added, but ClassTA allows the action to be performed in the same page and without disrupting the view and attention of the user from the current page.
4.2 SYSTEM ARCHITECTURE

The design of the workflow for Calendar Management System is simplified using the AJAX technology which uses HTML in addition to DOM for dynamic presentation and interaction in the very first Presentation layer. The system Architecture has been adapted from the architecture provided in codeproject website [29]. As shown in Figure 4.5 [29], the EventCalendar.js handles the client scripting for making it work. Information exchange amongst client and server is done using XMLHttpRequest residing in CalAuthenticator.js. CalAuthenticator.js is called to make a web service call and facilitate data transfer between browser and the database. The parameters received from the client side are sent back to CalManager.java which using SQL queries get and send data from the database to the front end. The JavaScript EventCalendar.js dynamically displays and handles the request from the user, and respond from the server.

4.3 DATABASE ARCHITECTURE

The CalEvent and CalEventS are the only objects used for the faculty and student Event Calendar respectively to represent the events for the system. Figure 4.6 showed the class representation of the event object. Both objects are identical except student only

Figure 4.6. UML diagram for CalEvent.

requires getStudentEvents() method in order to fetch all events for classes they have registered for.

The CalEvent class has 11 attributes and 3 major methods. iEventId is the primary key for the event. iOwnerId is the unique id that each user has in the database providing their user name and password for accessing their account. sEventTitle, sEventBody, sEventDate, sEventTypes, sClassName and sRosterName are all provided by the faculty through edit
window. sFacultyName is auto generated when faculty logs in and sEvenCreationTime is added to the database when a new event is created. modifiedTime changes each time user edits an existing event. Function getEvents fetch the events from the database by userID provided by the web service. The following SQL query is used to get all events for a unique userID from the database.

```
"select * from classta.calendar where ownerID = "+ownerID+"";
```

Calendar is the table in database in which all data for Event Calendar is stored.

addEvent() function adds a new event, saveEvent() modifies an existing event based on eventID from the database. SQL is used for adding new event to the database:

```
"insert into classta.calendar ( ownerID , rosterID , eventType , eventTitle , eventBody ,eventDate,lastName,creationTime,modifiedTime) values " + " ( " + ownerID + "," + sRosterID + "," + eventTypes + "," + eventTitle + "," + eventBody + "," + eventDate + "," + facultyName + "," + className + "" , NOW() , NOW())";
```

All the values are provided from the client side through the web service and creationTime and ModifiedTime are received from the function NOW() that provides the current date and time. To edit an existing event from the database, the web service gets the unique eventId from client side and supplies it to fetch the update the event from the database. deleteEvent() just deletes the event providing the eventId.

```
"delete from classta.calendar where eventID="+eventID.
```

To get all student events, a list of all rosters from client side are sent to the server and SQL query fetch those events matching the rosterIRL.

```
select * from classta.calendar where rosterIRL = "+rosterIRL.
```

### 4.4 Layout

The layout for the Event Calendar can be separated in 4 parts: the ClassTA homepage from where user access the system, the original Event Calendar system with 3 parts of Instruction Panel, Calendar, and Event Editor Panel.
4.4.1 ClassTA Homepage

As explained in previous sections there is a calendar icon in the homepage which gives user access to the Event calendar when user logs in. If user is not logged in, it only shows the calendar to view the dates. Figure 4.7 shows the calendar icon in ClassTA homepage and the view of calendar when user is not logged in.

Figure 4.7. ClassTA calendar in homepage.

If user is logged in, the clicking on the calendar icon will take the user to the Event Calendar page which has three parts in it, an instruction panel, the calendar, and an event panel. Figure 4.8 shows the three parts of Event Calendar. Each parts of the event calendar page is explained in the following sections.

Figure 4.8. Event calendar homepage.
4.4.2 Instruction Panel

The Instruction Panel gives user information about the basic functionalities of how Event Calendar works. As calendar is color coded based on the type of event, each representation of color code is also shown in the scrollable instruction panel next to the calendar so user can quickly reference the panel when needed as shown in Figure 4.9.

![Instruction Panel](image)

**Figure 4.9. Event calendar instruction panel.**

4.4.3 Calendar

The ClassTA calendar has the usual look of ClassTA homepage calendar but it is more interactive. As shown in Figure 4.10, The top arrow buttons takes user to the next or previous month or year and the bottom Today button brings user back to today’s date. The weekends are highlighted as gray to separate them from weekdays. The color coded dates indicate the dates that have events and clicking on them highlight the event(s) in the event panel. Also clicking on uncolored date adds a new event to the event panel. A detailed description of its use will be explained later.

![Calendar](image)

**Figure 4.10. Calendar inside event calendar homepage.**
4.4.4 Event Editor Panel

The event editor panel in bottom contains all the events ordered by event date. As shown in Figure 4.11, the table has 4 columns for Event Date, Class and Professor Name in the same column, type of Event and Event ID in same, and Event Detail in brief. The right most header contains a maximize/minimize button to maximize or minimize the editor panel. The event panel is scrollable when minimized. The events inside the panel have separate colors; gray ones represent events that are older than today and two version of blues show the dates that are newer. The events are clickable and on click it expands and shows the detail information of each event.

![Figure 4.11. Event calendar event panel.](image)

4.5 FEATURES FOR FACULTY

The major features for Event Calendar are; Add new event, Edit Event, Delete Event, View Event, and star events. Only the faculty is privileged to use all those features but students are only allowed to view the events and star them. Most of the features can be performed from both the calendar and also from the event panel. The faculty features are explained before the student feature. A detailed description of the features is as follows:

4.5.1 Add Event

Faculty has the authority to add new event on a specific date. From the calendar, the faculty needs to click on a date with no color code. That will create a new event in the specified position ordered by event date in the event panel. As shown in Figure 4.12, clicking on Nov 22 on the calendar highlights the date and adds a new event for that date in the editor
Figure 4.12. ClassTA event panel.

Panel before the event on Nov 23. The scroll bar automatically scrolls to the new event. The new event edit window is open for user to fill up and save to confirm its addition with detail information. The user can always delete or cancel the new event if does not wish to save it. A new event message is shown in the message panel indicating user of the action performed specifying the date.

Faculty can also add event from the event panel where an “Add New Event on this date” button allows user to add a new event on the specified date as shown in Figure 4.13. The button is located in the detail view window. The same procedure is followed in adding the new event as adding event from calendar.

Figure 4.13. Add new event button.
User is not allowed to create event on date that is older than today. To error proof that attempt, the user will be alerted as shown in Figure 4.14. Whenever user tries to create a new event from the calendar from a date that is older than today, a message in the message panel states “You cannot add events on previous dates.”

![Figure 4.14. Add old event message.](image)

Also the “Add new event” button is disabled inside the editor panel for older events and clicking on it will also show a message inside the detail window as shown in Figure 4.15.

![Figure 4.15. Add old event message in panel.](image)

### 4.5.2 Delete Event

Faculty is also allowed to delete an existing event from both the calendar and event editor panel. As shown in Figure 4.16, in the panel the delete button is located inside the edit window. If any of the required fields are not empty, user will get an alert asking if the event really needs to get delete. Only a confirmation will allow user to delete it. If user clicks on other date on calendar or any event in the panel and the edit window fields are empty, user will not be prompted of the deletion and the event will automatically be deleted without asking users permission. User can also delete event from calendar and same rules also apply
here. When user creates an event from calendar the calendar date is highlighted. Clicking on
the same date or another date will prompt user with permission to delete the event if the
fields are not empty or just delete it if the fields are empty.

If a new event is created, delete, cancel, or close any of these buttons will delete the
event from the list as no modifications have been done.

4.5.3 Edit Event

The edit window allows user to perform editing an existing event or insert new
information for new event. Figure 4.17 shows the edit button residing inside detail view
window.

As shown in Figure 4.18, the edit window header shows the date of the event that is
being edited. The save and delete buttons saves and deletes the event respectively. Close
button simply closes the edit window and refresh the editor panel in its original state. The
Cancel button takes the user from edit view to detail view. The required fields are marked
Figure 4.18. Edit window.

with red asterisk. The small calendar icon opens up a mini calendar beside detail body. Selecting a date automatically inserts the text of the date inside the detail body field.

Figure 4.19 shows the list of classes for the faculty. Faculty can choose one of the classes where he or she wants to post a particular event. Also he or she can choose the private event that no one else but the faculty can view.

Figure 4.19. Select class drop down menu.
When a faculty member chooses a class, a list from a roster is also shown in the drop down list for Roster. From the roster list, faculty can choose for which specific roster he wants the event to get posted. The roster list is shown in Figure 4.20.

![Figure 4.20. Roster drop down menu.](image)

Faculty members can also choose the type of event from the Select Type drop down list shown in Figure 4.21. As mentioned already, each type has a unique color listed in the instruction panel. The default type is Announcement.

![Figure 4.21. Type of event.](image)

Faculty can also create a new type by selecting the “Create Type” option from the drop down menu which will create another text field for user to insert his own type of event. Figure 4.21 and 4.22 shows the list of event types and text field for creating a new type respectively.

![Figure 4.22. Create new event type.](image)
In order to error proof a user’s interaction with events, a user will only be allowed to open one edit window. Trying to open another edit window or creating a new one will prompt the user to close the existing open edit window. Figure 4.23 shows the open edit window and the alert user gets trying to open another edit window without closing it.

![Figure 4.23. Edit window prompt.](image)

When an edit window of an event is open, user can view one detail view of an event for reference. The color coded date in the calendar will not open any detail view. User can only view the details by clicking the events from the editor panel. Trying to open one detail view window while another detail view is open will cause the previous detail view to close. Figure 4.24 shows an edit window on the top and detail view of another event in the bottom from which user can get reference.

Edit button is always disabled for older events same as the Add new event button located inside the detail view window. Clicking on Edit button on old dates will show a message inside the detail window stating old events cannot be edited as shown in Figure 4.25.

### 4.5.4 View Event

Both faculty and student have the option of viewing event from both the calendar and from the panel. Each color coded date in the calendar represents a specific type of event and clicking on one will show the detail view of the event in the event panel. The scroll bar will scroll to the position the event is which helps user to view the detail information easily.
Figure 4.24. Edit and detail window.

Figure 4.25. Disabled edit button for old events.

Figure 4.26 shows the selected color coded date and detail view of the event in the panel. User can also view the detail information of an event by selecting any event from the window. If a date has multiple events, all of the events detail window will be shown in the panel. Clicking the same date will close the detail window.

As shown in Figure 4.26, all the fields of detail view window look same as the edit window except the fields are not editable. The creation date of the event is shown on the right top corner of the window. The Professor’s name is also shown for the students.
When user closes a detail view window or an edit view window, the event gets highlighted. It helps the user identify the last event he has worked on. Figure 4.27 shows the highlighted event that the user has last worked on.

![Figure 4.27. Highlighted event.](image)

### 4.5.5 Starred Event

Both faculty and students are capable of adding a star for special events. Stars help users to easily identify the specially marked events in the current window and also when the user logs back in. The stars are located in the left side of each event as shown in Figure 4.28.

![Figure 4.27. Highlighted event.](image)

#### 4.6 FEATURES FOR STUDENTS

In ClassTA Calendar, students are allowed to view events and star mark the important events. The layout differs as it does not contain the instruction panel as in faculty calendar as shown in Figure 4.29.
Students can view events from calendar and also from the event panel. The detail view window only shows information like date, detail, class, professor’s name, and event type as shown in Figure 4.30. It does not contain any edit or add new event button as students are not allowed to perform these functions.

Students get events from the classes they have registered for. Faculty can select a specific class and roster while creating an event and student can view the events when the faculty has posted under the specific class roster. If students have new events, the event calendar tab will automatically open whenever the student logs in.
Figure 4.30. Student detail view.
CHAPTER 5

FUTURE ENHANCEMENTS & CONCLUSION

5.1 FUTURE ENHANCEMENTS

By comparing the features that exists in other leading Event Calendars like Google Calendar, Moodle Calendar, and Blackboard calendar; some new enhancements can be added to make the current version more useful to the user.

- Users should be able to insert new events from other modules without going inside the event calendar. For example, if the user is inside Blog, a link can be added there which can allow the user to insert a new event which will synchronize with the original event calendar.
- Students can also have the privilege to add new events for personal use. They can view these events as a reminder. Add feature will come with edit and delete the existing private events for students.
- Users can also have the option of showing only the starred events. By selecting this option both faculty and student can concentrate on only the starred events they marked as important.

5.2 CONCLUSION

The development of the new Calendar Management System has expanded and enhanced functionalities and usage than the existing older version. With the advancement of technology, many people now rely on electronic calendar as a means of event set up and reminder of new upcoming events. The new Calendar Management System has the capability to keep both faculty and student up-to-date about the events. Any user can still view the old calendar without logging in just for the purpose of view date. A user can operate the calendar without having prior knowledge, but the instruction panel still gives some basic guideline about the functionalities. The new Calendar Management System can be a very useful feature for faculty student interaction. The new calendar is AJAX based, thus, it is very fast and responsive and the enhanced GUI can give the user a very easy to use feel.
REFERENCES


