ClassAxis Analysis

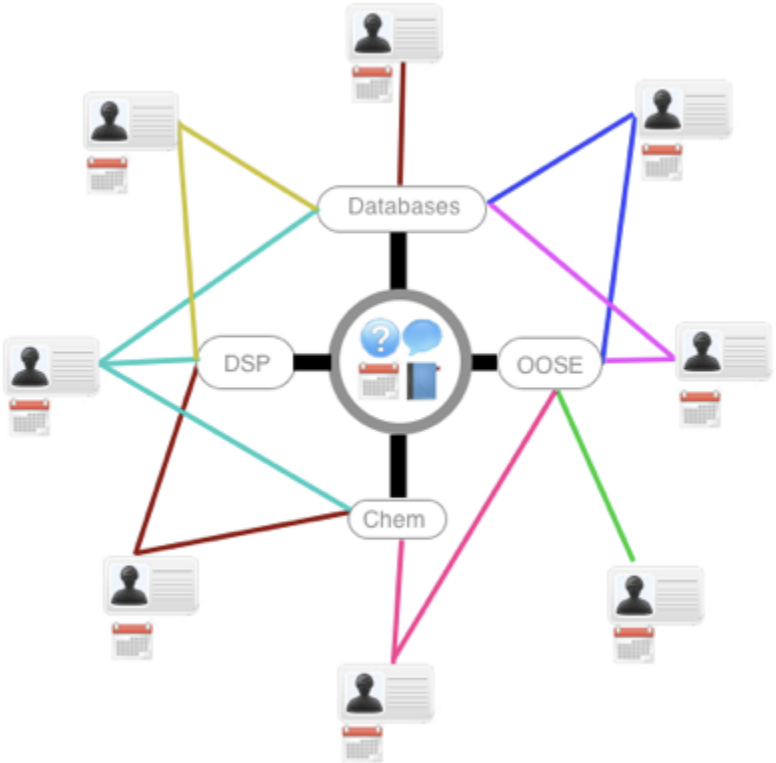
Marquette University, Department of Computer Engineering
COEN4610, Object-Oriented Software Engineering

Prepared for:
Dr. James Conigliaro

Prepared by:

Corey Bost	Johnny Fandel	Salome Marquez
Kellene Buchwald	Travis Huhn	Michael Santana
Dexter Del Frate	Andre Lancour	Sam Warmuth

December 10, 2010



Contents

1. Introduction

This document covers the Analysis of the Functional and Nonfunctional Requirements, Scenarios, Use Cases, Object models, Dynamic models, and User Interface models.

2. Current system

Currently, groups of people communicating have a difficult time with coordination. Although all information can currently be shared, it is not in one location. Groups today generally communicate through email, phone conversations, and perhaps an online medium such as forums. In this way, there are many places the group members have to check for new updates and information, increasing the chance of inconsistencies and problems of people missing deadlines.

3. Proposed system

ClassAxis will be the new tool groups can use for communication. It allows you to send messages to your group members, replacing the need for emails. It also has threaded discussions to debate ideas, figure out optimal strategy, and to communicate with other members. The event planning will allow all of the team members to see upcoming deadlines without having to remember which email has the deadline in it, ClassAxis will display it for you.

3.1. Overview

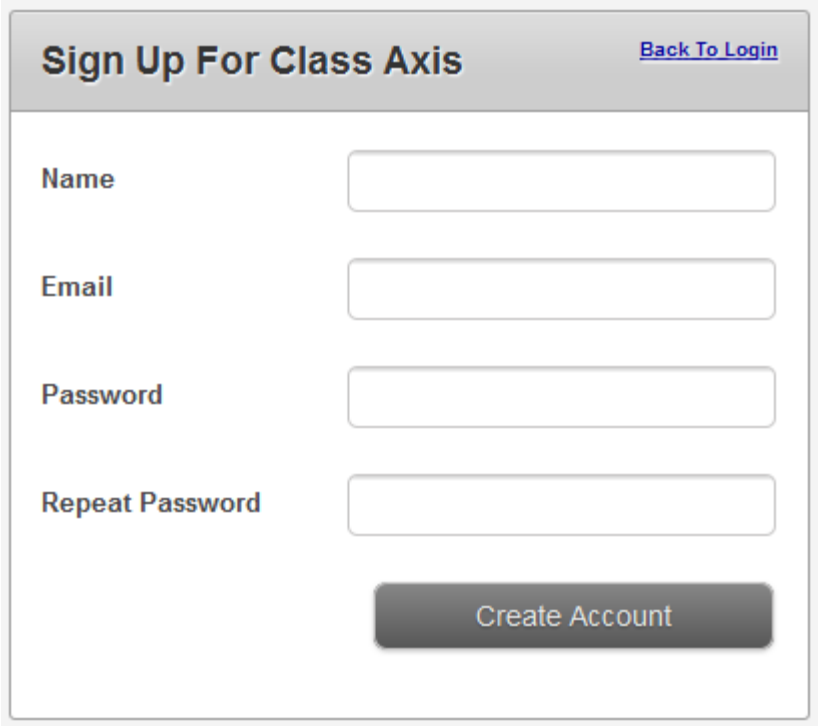
This section of the Analysis document includes an in-depth look at ClassAxis's Functional Requirements, Nonfunctional Requirements, and System models.

3.2. Functional Requirements

Classaxis will have several functions, including ones involved with accounts, home page, course page, event page, and topic page.

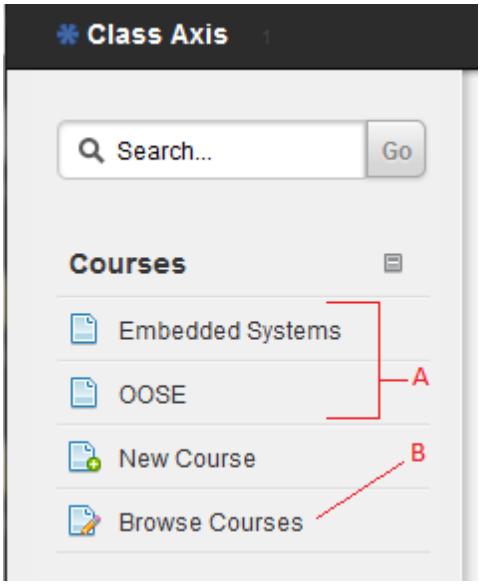
3.2.1 User account management

- The software shall require a name, password, and e-mail to create a user account.
- The software shall store passwords with SHA2 encryption, not as plain

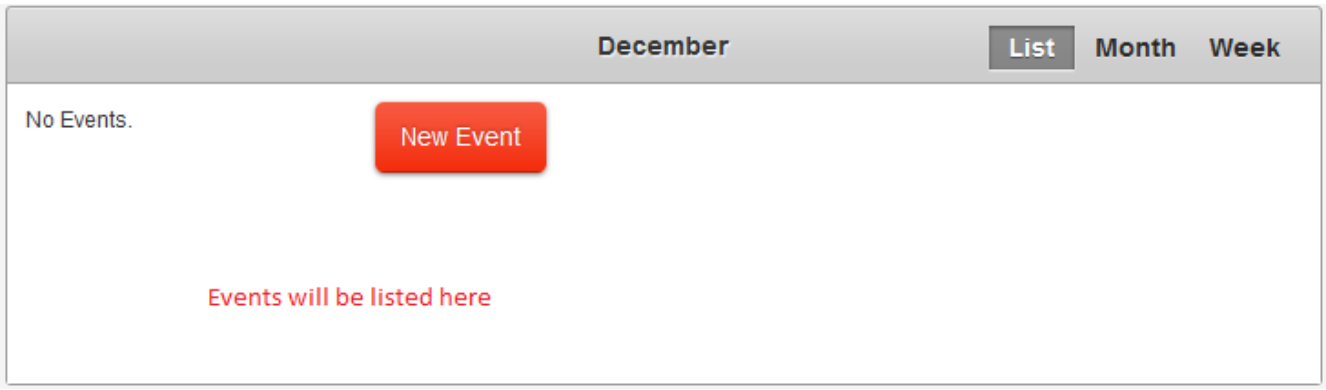
text. 

3.2.3 Home screen

- There shall be a home screen that the user is sent to upon logging in.
- The home screen shall list the user's current courses. (A)
- The home screen shall allow the user to enroll in a new course by selecting one from a list of classes. (B)



- The home screen shall display the names, times, and locations of all current or future events that the user is attending.

- 

  The home screen shall provide links to 5 of the most recently updated threads that the user has subscribed to.
- The home screen shall allow the user to manage account settings, namely changing the password and e-mail address.
- The home screen shall allow the user to log out.

3.2.4 Course page

- Each course shall have a course home page.
- The course page shall allow the user to remove themselves from the course.
- The course page shall display the topics associated with the course.
- When the topic title is clicked, the topic page for that topic shall be displayed.
- The course page shall display the names, times, and locations of all events associated with this course.
- When the event name is clicked, the event page for that event shall be displayed.
- The course page shall have a means to navigate back to the user's home page.
- The home page shall have a mechanism for adding a new topic. This will require a subject and message from the user, along with optional tags and a means to subscribe to the new topic.
- The home page shall have a mechanism for adding an event to the course. This will require an event title, a time, and a location. There will be a way to optionally add a description and invite attendees from a list of students enrolled in that class. The user creating the event shall automatically be entered as attending. Invited users shall be sent an e-mail informing them of the event, but shall not automatically be marked as attending.

3.2.5 Event page

- There shall be a page for each event.
- The event page shall display the name, time, location, description, and course at the top of the page so that it is clearly visible.
- The event page shall display a list of all students attending the event.
- The event page shall display a list of all students invited to the event.
- The event page shall have a mechanism to set the user as attending the event, if the user is not currently attending the event.
- If the user is already attending the event, there shall be a mechanism for removing themselves from the

attending list.
- If the user is currently attending the event, the event page shall display a mechanism to invite other students enrolled in that course to the event.

3.2.6 Topic page

- Each topic shall have its own main page.
- The original post, along with all associated replies, shall be displayed on the topic page.
- Posts shall be grouped with their replies, and each level of reply shall be indented one more level.
- Each post, including the original topic post, shall have a Reply link that allows the user to compose a reply to the post whose Reply option was clicked.
- When creating a post, the only information the user enters shall be the content of the reply.
- The topic page shall provide a mechanism for a user to subscribe to the topic.
- When a topic is updated with a new reply, any users subscribed to that topic shall be alerted by e-mail.

3.3. Nonfunctional Requirements

The following subsections detail the nonfunctional requirements of ClassAxis.

3.3.1 Usability

- Users must be registered on Class Axis website to be able to use Class Axis
- User must be registered to a class to comment on a discussion, or post a new discussion

3.3.2 Reliability

- The password for a user must be secured and protected using SHA2 encryption.
- Privacy settings for user must be stored and preserved.
- A user's account and settings will be stored indefinitely.
- The system must have 99% up time.

3.3.3 Performance

- The system must respond in less than one second.
- The database must hold 16GB of data.

3.3.4 Supportability

- The system administrator must maintain the website daily.
- Only supports the English language.

3.3.5 Implementation

- ClassAxis must be compatible with Internet Explorer, Firefox, Chrome, and Safari web browsers.

3.3.6 Interface

- ClassAxis must be presented in the form of web pages.

3.3.7 Packaging

- ClassAxis will be installed on a webserver and will then be accessible to the server administrator and users with web browsers.

3.4. System models

The System models section of this document includes scenarios of how users would interact with the ClassAxis system.

3.4.1. Scenarios

3.4.1.1 Join ClassAxis

Scenario name	joinClassAxis
Participating actor references	user:User posts:Post
Flow of events	1. The user opens www.classaxis.com in a browser 2. The user clicks a link to register 3. The user enters a username, password, confirm password, and e-mail 4. If registration fails, user is sent back to the registration form and presented a message about what went wrong. 5. If registration succeeds, go to user home screen -- the home screen shows classes the user is attending, and allows quick access to posts being watched. 6. Also provides access to account settings, allowing the user to change password/email.

3.4.1.2 Add Topic

Scenario name	addTopic

Participating actor references	user:User topic:Topic
Flow of events	1. From the course home, user clicks the “New Topic” button. 2. A dialog appears, prompting user for a subject, message, tags, and a checkbox to subscribe to the topic. 3. The user clicks submit and the topic is immediately available to everyone in the course.

3.4.1.3 Post Reply

Scenario name	postReply
Participating actor references	user:User post:Post
Flow of events	1. Displayed along each post will be a Reply button. Clicking it brings up a text field in that window, along with a Submit button 2. The user enters their reply, then clicks Submit

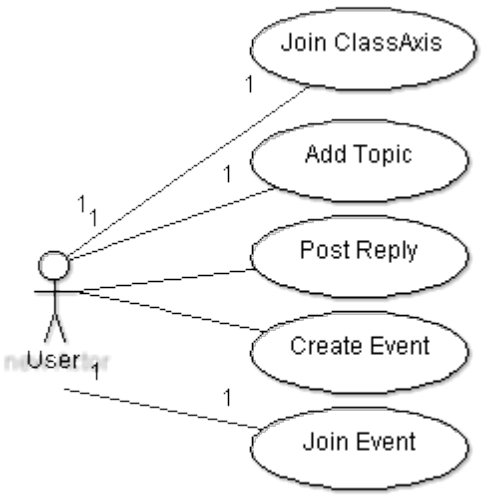
3.4.1.4 Create Event

Scenario name	createEvent
Participating actor references	user:User event:Event
Flow of events	1. Course home will have add event button. User clicks this. 2. Dialog appears, asking for event name, date, time, location, and options for repeating. User fills in all these, then clicks submit. 3. Event is displayed on course home page with Join Event button displayed to users not attending and Leave Event button displayed to users who are attending.

3.4.1.5 Join Event

Scenario name	joinEvent
Participating actor references	user:User event:Event
Flow of events	1. Click on Join Event button, click Yes if we have a confirm dialog.

3.4.2. Use Case model



3.4.2.1 Join ClassAxis

Scenario name	joinClassAxis
Participating actor	Initiated by user
Flow of events	1. The user opens www.classaxis.com in a web browser 2. The user clicks a link to register 3. The user enters a username, password, confirm password, and e-mail 4. If registration fails, user is sent back to the registration form and presented a message about what went wrong 5. If registration succeeds the browser goes to

	the user's home screen
Entry conditions	• user does not already have a ClassAxis account
Exit conditions	• user has successfully finished creating a new account
Quality requirements	• user is able to successfully create an account if they do not already have one

3.4.2.2 Add Topic

Scenario name	addTopic
Participating actor	Initiated by user
Flow of events	1. From the course home, user clicks the "New Topic" button. 2. A dialog appears, prompting user for a subject, message, tags, and a checkbox to subscribe to the topic. 3. The user clicks submit and the topic is immediately available to everyone in the course.
Entry conditions	• user clicks the "New Topic" button
Exit conditions	• user clicks "submit"

3.4.2.3 Reply to Topic

Scenario name	postReply
Participating actor	Initiated by user
Flow of events	1. Displayed along each post will be a Reply button. Clicking it brings up a text field in that window, along with a Submit button 2. The user enters their reply, then clicks Submit

Entry conditions	● user clicks “Reply” on an existing post
Exit conditions	● user clicks “submit” after finishing their reply

3.4.2.4 Create a New Event

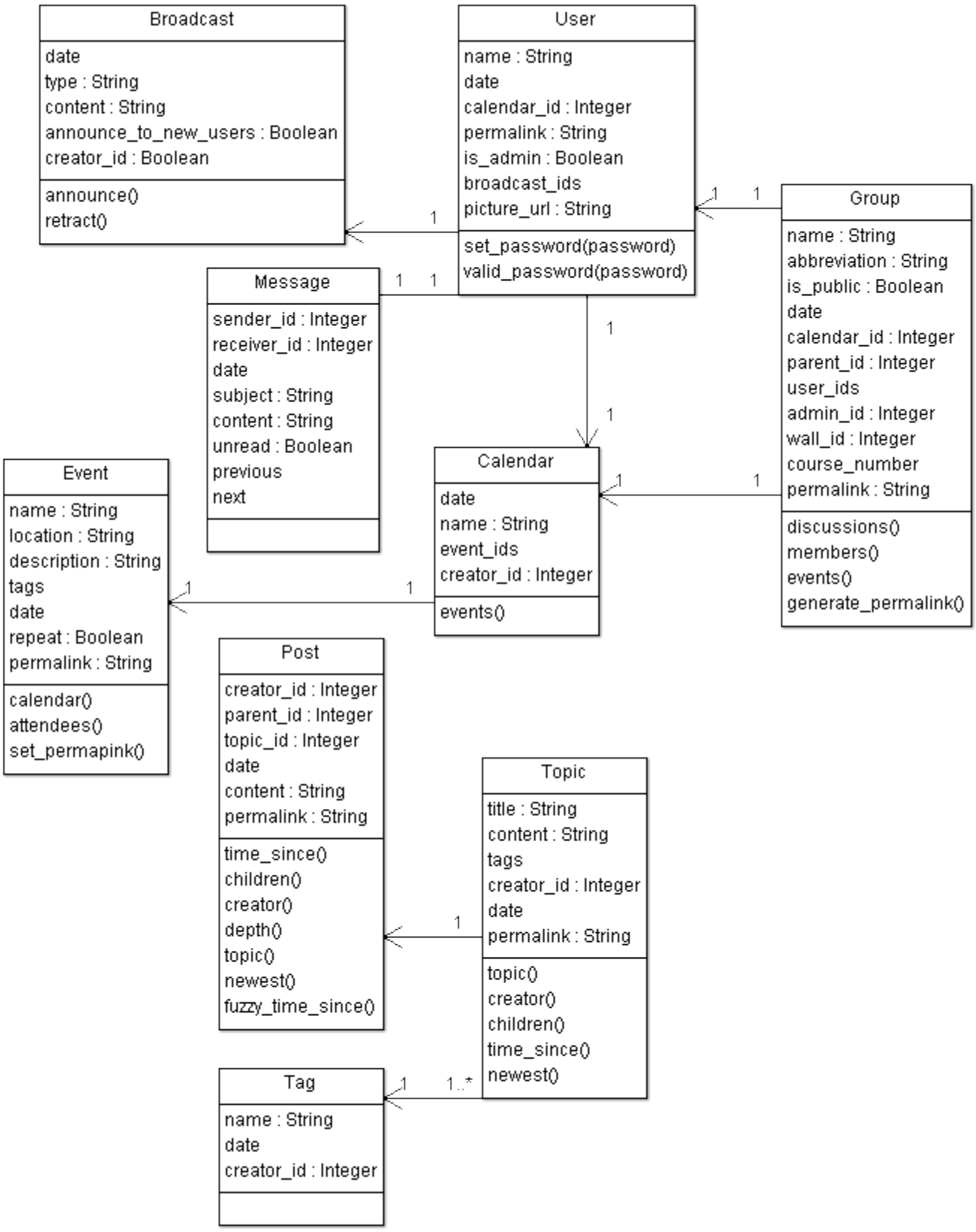
Scenario name	createEvent
Participating actor	Initiated by user
Flow of events	1. user clicks the add event button from the course home 2. Dialog box appears, asking for event name, date, time, location, and options for repeating. User fills in these fields and clicks submit. 3. Event is displayed on course home page
Entry conditions	● user clicks the add event button
Exit conditions	● user has successfully finished filling out the fields and clicks submit

3.4.2.1 Join Event

Scenario name	joinEvent
Participating actor	Initiated by user
Flow of events	1. user clicks on Join Event button on an event that they are interested in joining
Entry conditions	● user clicks on Join Event
Exit conditions	● user clicks yes in the confirm dialog

3.4.3. Object model

3.4.3.1. Class diagrams


Broadcast
date
type : String
content : String
announce_to_new_users : Boolean
creator_id : Boolean
announce()
retract()
User
name : String
date
calendar_id : Integer
permalink : String
is_admin : Boolean
broadcast_ids
picture_url : String
set_password(password)
valid_password(password)
Group
name : String
abbreviation : String
is_public : Boolean
date
calendar_id : Integer
parent_id : Integer
user_ids
admin_id : Integer
wall_id : Integer
course_number
permalink : String
discussions()
members()
events()
generate_permalink()
Message
sender_id : Integer
receiver_id : Integer
date
subject : String
content : String
unread : Boolean
previous
next
Calendar
date
name : String
event_ids
creator_id : Integer
events()
Event
name : String
location : String
description : String
tags
date
repeat : Boolean
permalink : String
calendar()
attendees()
set_permapink()
Post
creator_id : Integer
parent_id : Integer
topic_id : Integer
date
content : String
permalink : String
time_since()
children()
creator()
depth()
topic()
newest()
fuzzy_time_since()
Topic
title : String
content : String
tags
creator_id : Integer
date
permalink : String
topic()
creator()
children()
time_since()
newest()
Tag
name : String
date
creator_id : Integer
1
1..*

3.4.5. User interface, navigational paths, and screen mock-ups

3.4.5.1 Sign-Up Screen

Figure 1 displays the mock-up for the ClassAxis sign-up page. As can be seen, the sign-up screen will take in the name and e-mail for the user. The user will then be required to create a password for their account. The last field requires the user to reenter his or her password to confirm. An account will be created once the user fills in all the fields and clicks the “Sign-Up” button on the bottom of the page.

3.4.5.2 Logged-In Home Screen

Figure 2 displays the mock-up for the ClassAxis logged-in home page. As be seen, the top of the screen displays the user’s name. Three different sections will be diplayed: recent headlines, upcoming events, and discussions. The headlines section displays the most recent recent events that have occurred within ClassAxis. The Upcoming Events section displays events on a calendar that the user has been invited to and is attending. The final section displays the discussions that the user has created.

The sidebar on the left of the page is described in section 3.4.5.8: ClassAxis Sidebar.

3.4.5.3 ClassAxis Sidebar

Figure 3 displays the mock-up for the ClassAxis sidebar. As can be seen, the sidebar contains the following menu items: Courses, Groups, Events, and Discussions. Each menu item can expand and minimize by clicking on the plus and minus symbols respectively. Expanding the courses menu will display all the current courses the user is enrolled in. Likewise, expanding the groups menu will display all groups the user is a part of. Expanding the events menu will display all the events that the user has been invited to. Expanding the discussions section will display discussions that the user has created. The user can also select “Browse” under each menu item to view all the items of the respective menu item.

3.4.5.4 ClassAxis Course Homepage

Figure 4 displays the mock-up for the ClassAxis Course Homepage. As can be seen, the top of the screen displays the name of the course. The course homepage consists of the three sections: Events, Discussions, and Members. The Events section displays the calendar events associated with the course. The Discussions section displays discussion topics associated with the course. Users can create new events and discussions by clicking on the respective buttons. The members section lists all of the ClassAxis users that have joined the course.

3.4.5.5 ClassAxis Group Homepage

Figure 5 displays the mock-up for the ClassAxis groups homepage. The page is very similar to the course homepage. There are three sections: Events, Discussions, and Members. The Events section displays the calendar events associated with the group. The Discussions section displays discussion topics associated with the group. Users can create new events and discussions by clicking on the respective buttons. The members section lists all of the ClassAxis users that have joined the group.

3.4.5.6 ClassAxis Discussions Homepage

Figure 6 displays the mock-up for the ClassAxis groups homepage. As can be seen, the page displays all the discussion threads that have been posted within ClassAxis. For each discussion that is posted, the title of the discussion, who posted it, and the message of the first post is displayed. To view the full thread of the discussion, the user can click the “View Discussion” button.