ACMS

A System for Academic Course Management and Scheduling via the World Wide Web

Version 1.0.1

Specifications Document
DRAFT 1

Revised on February 10, 1997

1.0 Introduction

1.1 Document Overview

This document lays out the specifications for ACMS, a system for Academic Course Management and Scheduling via the World Wide Web.

1.2 Problem Statement

The following is excerpted from the Requirements Document:

Scheduling and calendaring information are scattered throughout the University. No centralized information source exists from which one individual or group of individuals within the community might see how one's own personal or group schedule comes together. An ideal sample of such information includes University holidays, course or student group schedules, and personal schedules.

Of particular importance is the demand for the use of electronic resources and teaching aids in the classroom. Through personal initiative, some University instructors have begun to take steps towards integrating computers and other technologies to the benefit of their students. The World Wide Web and campus file servers have proven invaluable as a central information source for students exposed to this technology.

For the remaining majority of the University's faculty and students, however, some means for easily accessing and utilizing electronic information resources needs to be developed. If a centralized data repository were to exist in the community for holding information such as course schedules and calendars, a facility should be provided for instructors who want to use this repository yet do not have the time to learn the details of emerging technologies. An ideal system would allow any given instructor the ability to create, maintain and publish a course syllabus via the World Wide Web, for example, without needing to learn the intricacies of WWW publishing.

If scheduling information such as course syllabi were then tied into a central data repository, people across campus would have quick and easy access to the information they need -- all from one source. Such would eliminate the scattering of information we currently see, and would greatly reduce the need for paper documents which still predominate in the world of information dissemination.

1.3 Solution Overview

The following is excerpted from the Requirements Document:

ACMS is a system which will provide the centralized data repository described in the Problem Statement. This repository will be an extensible, core information resource, and ACMS will provide a series of front-end applications necessary for the creation and maintenance of information bits and pieces within this repository. In particular, ACMS will entail the following:

- Providing all University offices, student and academic groups, and individual community members with a simple and secure means for creating and maintaining group or personal calendars and schedules. This would be a generic front-end application to the ACMS core repository, and would allow users the ability to publicize or privatize information on a group or individual basis.
- A “group” within ACMS can be viewed as a particular University office, department, course participants, or student organization, where each group consists of a series of individuals, each of whom might belong to more than one group.
- With academic courses as a particular focus group, ACMS will provide all faculty of the University with a simple and secure means for creating and maintaining standardized course websites for distributing course information. All schedule related information will be integrated into the ACMS data repository. This and other information includes syllabi, reading lists, laboratory or section times, class projects, handouts, lecture notes, and contact information.

ACMS will be designed for local, University access only over the local area electronic computer network.

1.4 Client and Target Users

The following is excerpted from the Requirements Document:

ACMS was conceived from discussions with Brown University’s Computing and Information Services (CIS) Department. ACMS is supported by the Brown University Webmaster, CIS Software Development Teams, and other related groups and individuals providing computing services to the University community.

Potential target users of ACMS will be instructors and students of the University. ACMS will be designed for use by all members and offices of the University.

1.5 Document References

The following are related documents of interest to this project:

- ACMS Project Overview
- ACMS Requirements Document (Draft 2)
- ACMS Specifications Document (Draft 1)
- *vCalendar* Specification
- *vCalendar* Whitepaper

During the development of ACMS, APIs will be written for all class interfaces, particularly those which pertain to security and transport standards. Whenever possible, javadoc generated API documents will be maintained; all Java source code should be written to conform to javadoc standards. APIs for server based C++ applications will be written in a similar fashion.

2.0 Functional Specifications

2.1 Authentication and Security

ACMS will provide secure user authentication and message encryption. Information maintained in the ACMS data repository will be accessible on a user/group basis, with a model similar to those found in most Unix systems, with the following high-level definitions:

- Users will always belong to at least one group of users. Users will most often belong to multiple groups. (For example, user A will belong to both the general community group AND a series of groups for each course in which user A is enrolled.)
- Groups will consist of at least two users. Groups may be created according to departments, student organizations, University offices, or individual academic courses.
- Every group will have at least one group administrator. This administrator may choose to give users (in the group) write access to the group's information repository.
- Users who have write access to a given group's information repository will be able to add or modify events for world and/or group accessibility.
- Each user, who will at the very least belong in a group defined by the entire University community, will be able to add and mark events in the ACMS repository as personal only.
- Each event in the ACMS data repository will have a user and group associated with it, similar to files in a multiuser filesystem. Items marked as personal will be associated specifically with one individual, using the broadest group as the item's group association.

- ACMS system administrators will have privileged access to all groups and users. ACMS sysadmins will be provided with a secure front-end application interface for creating, modifying, and deleting groups and users (see section 2.4).

ACMS will use MIT's Kerberos (Project Athena) for user authentication. This Kerberos implementation will initially adhere to Kerberos Version 4 for compatibility with the Kerberos authentication system already in place on campus.

End users will be required to present their Kerberos Principal name ("NetID") and associated Kerberos passwords in a trusted Java applet window. As per the spirit of Kerberos, passwords will not be sent across the network in any way, shape or form.

The Java Kerberos client will communicate with a server running on the same machine which served the applet to the end user. This server will relay all Kerberos requests to the network domain's Key-Distribution Service for authentication and ticket granting. The server will be a separate entity from the core ACMS server which handles event queries to the centralized data repository.

When a Kerberos Principal is authenticated, users will be automatically associated with the group or groups in which they belong. Any action taken by a user affecting the ACMS data repository will affect at most one group's repository.

ACMS users with Course Website creation and modification privileges will be given a choice of either Calendaring or Website Creation and Management upon a successful login.

2.2 Calendaring

The ACMS calendaring front-end application will be a specialized PIM (Personal Information Manager) used to create, modify, and delete individual or group schedule information maintained in the ACMS data repository. The ACMS calendaring client will provide the following functionality in order of priority:

1. Allow multiple views and filters of scheduled event data. Filters will be allowed on event types, descriptions, group associations, and time spans. Filters become important when the user attempts to create custom views of his or her schedule spanning the event data from multiple groups. For example, a student might want to view the schedules for all of his or her classes simultaneously, plus all general University calendar events such as holidays or registration deadlines. Events will be categorized by a series of types shown in Table 2.

2. Allow modification of event objects, including any of the following event item properties given in Table 1:

Table 1: Item Properties

Name	vCalendar Property Name	Description
Access	CLASSIFICATION	Defines who has access to see an item. PUBLIC: all members of the item’s Group can see the item. PRIVATE: only the Creator can see the item. CONFIDENTIAL: will not be used by ACMS. Althoug the vCalendar spec defaults to PUBLIC, ACMS will default to PRIVATE.
Creator	X-Creator	ACMS Extension: The Kerberos ID of the item’s creator.
Date/ Time Created	DCREATED	The date and time at which the item was created. Set by the server automatically.
Date/ Time Completed	DCOMPLETED	When a “TODO” style item is completed, this property is set.
Description	DESCRIP TION	The paragraph description of the item. Will appear in the item information panel if the item is selected.
Display Reminder	DALAR M	Functionality to allow users to be reminded of items by a dialog when ACMS is running.
Due Date/ Time	DUE	The end time for a “TODO” style item.
End Date/ Time	DTEND	The end date for an “EVENT” style item. Yes, that terminology is annoying, but that’s vCalendar’s fault.
Group	X-GROUP	ACMS Extension: The Group to which the item is accessible. The only legal options are groups to which the creator belongs.
Last Modified	LAST-MODIFIED	The date and time at which the item was last modified. Set by the server.
Mail Reminder	MALAR M	Functionality to allow users to be alerted of items by email.
Priority	PRIORITY	Required by the vCalendar spec. Will default to 0 (=undefined)
Start Date/ Time	DTSTAR T	The start time for an “EVENT” style item.
Status	STATUS	Required by the vCalendar spec. Will default to “NEEDS ACTION”

Table 1: Item Properties

Name	vCalendar Property Name	Description
Style		There are two “Styles” of item. Those that have a start and an end time (what vCalendar calls “EVENT”s) and those that have only a due date (what vCalendar calls “TODO”s). Setting the style in ACMS will determine what type of vCalendar object is created.
Title	SUMMARY	The short name for an item. Will appear in calendar graphical views.
Type	CATEGORIES	The category/type of item this is. Legal values are those from the list below.
Unique Identifier	UID	A unique identifier for each item that may be useful for determining URLs and the like.
Parent		The parent event of a recurring event.
Recurrence Rule		Repitition rule for repeating events.
Modified Flag		Indicates if a repeating event has been modified from it’s parent. These events will not change when the user changes the parent event.
Location		Location for the event.

Table 2: Types (CATEGORIES)

Name	Description
Course.Lecture	Common meetings of a course .ACMS will require items of this Type to be of Style EVENT.
Course.Section	Section meetings of a course. ACMS will require items of this Type to be of Style EVENT.
Course.Deadline	Assignment due dates. ACMS will require items of this Type to be of Style TODO.
StudentGroup.Meeting	Rehearsals, organizational meetings, etc, for Student Groups. ACMS will require items of this Type to be of Style EVENT.
StudentGroup.Shindig	Public events, performances, parties, lectures, etc. ACMS will require items of this Type to be of Style EVENT.
StudentGroup.Deadline	Due dates for applications, submissions, etc. ACMS will require items of this Type to be of Style TODO.
Personal	Personal events, appointments, etc.

Table 2: Types (CATEGORIES)

Name	Description
University.Vacation	University vacation days. ACMS will require items of this Type to be of Style EVENT.
University.Deadline	University deadlines, such as for housing, course registration, meal plan, etc.ACMS will require items of this Type to be of Style TODO.

3. Allow easy, scalable access to a given time period's events. For example, to access a single day's events calendar, selecting a day in the image of a calendar month view will zoom to that day's events. Calendar views will scale in granularity to provide broader views of rarer events as selected by filtering.
4. Allow the addition of new events to the calendar. By default, a student's calendar, for example, will include all classes for which that student has registered. Other events of several, pre-defined generic types will be allowed, as specified above in Table 2.
5. Provide a means for repetition scheduling, as is available in most PIMs.
6. Provide event alerting for all or selected events which occur while the client is running, as per *vCalendar* specifications.
7. Provide email alerting for all or selected events which occur during a specified time period while the client is not running (that is, the user is not logged in).

2.2.1 Calendaring Inputs and Outputs

Login — Allows the user to login to the ACMS system and brings up his/her PIMC. This brings up the kerberos login dialog box.

Logout — Allows the user to logout of the system.

Save — Saves the current changes to the calendar.

Exit — Exits the PIMC.

Undo — Allows the user to undo previous changes. We will again allow multiple levels of undo.

Cut — Cuts the selected text and places it into a buffer.

Copy — Copies the selected text and places it into a buffer.

Search — Does a keyword search on the user's calendar.

Find Events
Searches the list of events on the database for specified events. This uses the event type and the user’s access to groups to find events the user has access to which match the specified type.

Add Event
Adds a selected event to the user’s personal calendar.

Add Meeting
Allows the user to specify an event which takes place on a weekly or daily basis. This will bring up a dialog box which will allow the user to specify the event and the intervals at which it will occur.

Remove Event
Deletes the selected event from the user’s calendar.

View by Month
Brings up the view of the calendar by month.

View by Day
Brings up the view of the calendar by day.

View by Week
Brings up the view of the calendar by week.

Back
Moves the calendar/date back. Brings up a dialog box which lets the user specify the number of days to go back.

Forward
Moves the calendar/date forward. Brings up a dialog box which lets the user specify the number of days to go back

Today
Sets the calendar/date to the current day.

New Event
Creates a new event. This will take the user to a blank event information panel into which he/she can type the information for the new event.

Delete Event
Deletes the selected event from the event list.

2.2.2 Personal Information Manager Errors

Login
Kerberos errors: incorrect password, username.

Exit
Unsaved changes. This will warn the user of unsaved changes and ask if he/she wants to save the changes before exiting the program.

Undo
Can’t undo. There are no changes to undo.

Cut
There is no text selected.

Copy
There is no text selected.

Paste
There is no text in the clipboard.

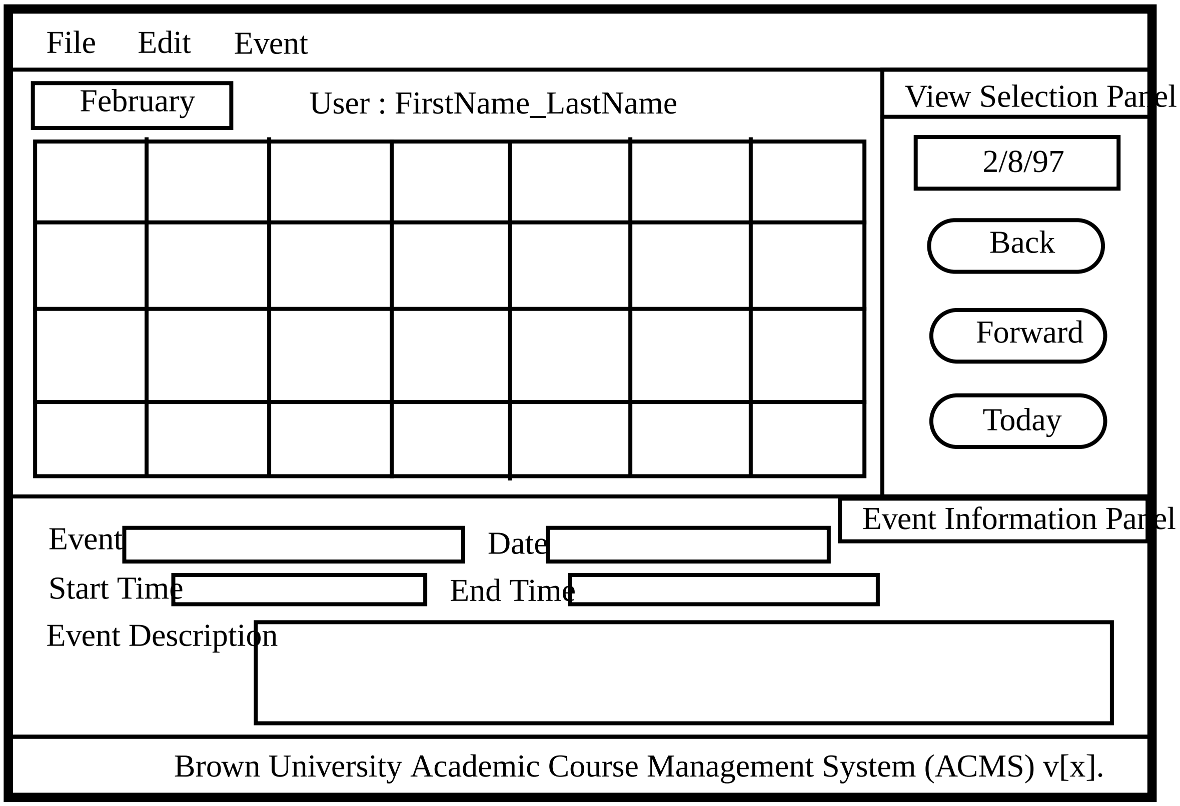
Search
The search didn't find anything.
No search string was entered. This will ask if you want to perform another search.

Find Events
There were no matches to the search.
No search data was specified. This will ask if you wish to perform another search.

Add Event
No event was specified.
The specified event has already passed. (This will be a warning rather than an error).

Delete Event
Authentication error. The user does not have the privileges to delete/modify the given event.

2.2.3 Personal Information Manager Interface



2.3 Website Creation and Management

Specifically aimed towards University faculty, ACMS will provide a front-end application for the creation and maintenance of standardized course websites, as described previously in this document. This extension to the core, ACMS data repository will entail the following functionality in order of priority:

1. Offer an initial, preformatted website based on information from the Brown Online Course Announcement (BOCA). Information such as the course title, instructor, and meeting times will be pulled from BOCA, and a default but empty syllabus will be automatically generated from an internal calendaring system. Holidays and other variations in the calendar will be pulled from other sources within ACMS's centralized data repository such as the University calendar.
2. Allow the maintainer to modify events defined in the site calendar. The following is a minimal list of modification functions which ACMS will provide:
   a. Provide a list of all events which are modifiable upon selection.
   b. Allow modification of an individual event date.
   c. Allow modification of an individual event description.
   d. Allow modification of readings or projects due for a given event, if applicable.
   e. Allow modification of the instructor for each class, if applicable, defaulting to the instructor of the course as defined in BOCA.
   f. Allow for the addition of hypertext-linked lecture notes or other documents for each event (see item 7).
3. Provide recoverability functionality during the maintenance of each website, with all modifications to the site logged as they are performed.
4. Allow the maintainer the ability to revert all modifications at any time before a final commit of modifications to the repository. Final commits to the repository will not be reversible.
5. Generate a WYSIWYG display of the web pages as they are modified. This will be a separate display from the modifiable list of events described in item 2.
6. Allow the maintainer the capability of moving events with simple mouse movements or keystrokes.
7. Provide the maintainer with a means of uploading lecture notes documents, images, or other non-hypertext content files. The system will outline preformatted areas within the website for linking these objects in. (See item 2f.)
8. Display a map of the website on-demand.
9. Provide a means for placing website documents under access control lists(ACLs) on a group or individual basis.

10. Maintain a log of site access statistics viewable by the maintainer.

11. Alert the site maintainer of any outdated or invalid hypertext links as the site is verified on a regular basis.

2.3.1 Course Management Component

Open   Open a course file by typing the course number. To open a course for editing, one would have to be authenticated and have access to the course you request.

Close   Close a file which is currently open.

Save   Commits/Saves the changes that the user has made.

Login   Allows the user to log in to the course management component (or re-login as a different person). This will bring up a dialog box where the user will have to type in his kerberos login name and password.

Logout   Allows the user to logout of the system.

Exit   Quits the course management component.

Undo   Allows the user to undo previous changes. We will have multiple levels of undo.

Cut   Cuts the selected text and saves it into a clipboard so that it can be pasting back into the document.

Copy   Copies the selected text and saves it into a clipboard so that it can be pasted into the document.

Search   Allows the user to perform a keyword search. Brings up a dialog box where the user can type in a word to search on.

New Event   Creates a new event. This will take the user to a blank event information panel into which he/she can type the information for the new event.

Delete Event   Deletes the selected event from the event list.

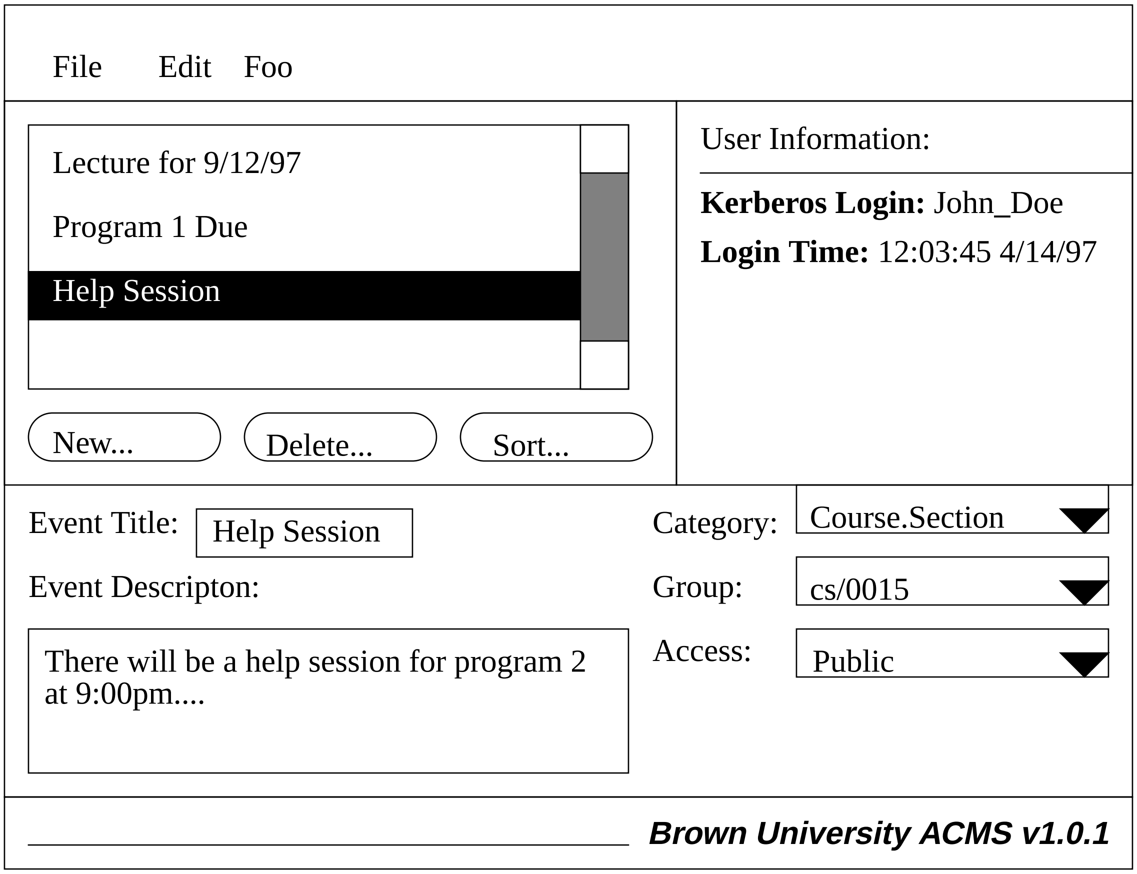
Sort   Forces the list of events in the event list panel to re-sort.

2.3.2 Course Management Errors

Open   Error finding the file. This will ask you to re-type the course number. Error in authentication. No access to file.

Close	Unsaved changes. This will give and error and bring up a dialog box which asks you if you would like to save the changes.
Login	Kerberos errors: incorrect password, username.
Exit	Unsaved changes. This will be the same error as the close error.
Undo	Can’t undo. There are no changes to undo.
Cut	There is no text selected.
Copy	There is no text selected.
Paste	There is no text in the clipboard.
Search	The search didn’t find anything. No search string was entered. This will ask if you want to perform another search.

2.3.3 User Interface Specifications for the Course Scheduling System



Lecture for 9/12/97

Program 1 Due

Help Session

User Information:

Kerberos Login: John_Doe

Login Time: 12:03:45 4/14/97

Event Title: Help Session

Event Descripton:

There will be a help session for program 2 at 9:00pm....

Category: Course.Section

Group: cs/0015

Access: Public

2.4 System Administration

ACMS will provide an interface to administrative functions to a select group of privileged users. These functions include:

1. Allow the sysadmin the ability to create, modify and delete groups.
2. Allow the sysadmin the ability to create and delete group administrators for a specific group.
3. Allow the sysadmin the ability to add, modify or delete users from any group. This includes the ability to modify any access rights for any group of users or individual user.
4. Allow the sysadmin the ability to add, modify or delete event information within any group's or individual's repository if necessary.
5. Provide status reports on the number of transactions performed by the core ACMS system on a regular basis.

6. Allow the sysadmin the capability of scheduling system downtimes for maintenance, including backups and upgrades. Administrative schedules would be ideally maintained as a separate group repository under ACMS.

2.5 Error Handling and Status Reporting

ACMS will alert the user of usage errors during runtime in a reasonable manner. ACMS will notify the users of system downtimes and maintenance. Problems encountered by ACMS users will be entered into a problem tracking system internal to ACMS developers.

2.6 Reliability and Recoverability

2.6.1 Client Failure

In order for this product to be useful to any clients, it must be able to recover from system problems as well as network problems. In order to make the system as extensible as possible in the realm of recoverability, the server should be able to maintain a list of the transactions which occur between each client and the server. In this way, uncommitted changes of any type in both the calendaring system and the scheduling system will be recoverable at a later time. Records of any client transactions would also make it possible to undo and redo a series of actions in case of user mistakes.

2.6.2 Server Failure

Ensure that in the case of server failure, all information stored there is recoverable. At worst, the data from a 24 hour period will be lost. Because the transactions for the individual users will be stored, it should be possible to recreate the missing data if the user desires it.

3.0 Look And Feel Specifications

It is important that all interfaces with which a user might be presented have a consistent look and feel to them. This should reflect a consistency in terminology, layout, means of inputting data, format of output, nature and tone of error messages, and so forth.

In the current state of the project, there are two different interfaces to the ACMS core. The first is the Course Management Component (CMC), the interface that professors will use to input the events associated with their courses and to create their course web pages from the ACMS boiler plates. The second component is the Personal Information Manager Component (PIMC), through which students, faculty, and staff will be able to keep track of and modify the various academic and university events that pertain to them.

3.1 System-Wide Look and Feel

3.1.1 Identification

Every window associated with the ACMS program should contain the phrase “ACMS v[x], where [x] is the current version number.

- The primary windows for ACMS components will have a banner across the bottom of the window reading “Brown University Academic Course Management System (ACMS) v[x].”
- Error, status, and other transient message windows will conclude their message with “(ACMS v[x]).”
- HTML pages generated by AMCS will contain a footer reading “<font size=”-1”>This page was generated by <a href=”http://www.brown.edu/cool_stuff/ACMS/”> Brown University ACMS v[x]</a>.</font>”

3.1.2 Interface

The goal of the ACMS interface is to make it easy for first-time users to navigate the system, while enabling experienced users to work quickly and effectively.

- Since ACMS should be available on multiple platforms (as it will be written in Java) the components will look as similar as possible across platforms, within the constraints of the Java Abstract Window Toolkit.
- Standard Cut-Copy-Paste functions should be available in all components.
- Identical functions across interface component will have identical names.
- Wherever possible, keyboard shortcuts will be available for actions accessible via a menu.

3.1.3 Alerts

Errors caused by user activity as well as system-status alerts (such as, “Sorry, the network is down.”) will be presented in application-modal windows. All error messages will use, as much as possible, plain-English phrasing. For example, “Could not connect to the server to confirm your changes. Please try again in a few minutes, or contact the administrator if the problem persists,” is preferable to “TCP Error, timed out waiting for response from server.” Consistent terminology will be used to refer to the components, actions, and features of the ACMS system.

3.1.4 Description of the Course Management Component

The window for the Course Management Component (CMC) will consist of 5 elements: a menu bar and 4 panels. The bottom panel consists of the identifying banner mentioned above. Clicking on the banner brings up an “About...” window. The panels are as follows:

- The Event List Panel (ELP): The Event List Panel contains a scrolling text box. Each line in the box is the name of an event. Clicking (and thereby highlighting) an event in this list causes the description of the event to appear in the Event Info Panel (described below). Below the scrolling box are three buttons: “New,” which is used to create a new event; “Delete,” which deletes the selected event; and “Sort,” which forces the list of events to resort itself.
- The User Information Panel (UIP): This panel lists information about the current user, none of which is editable. It displays the Kerberos ID of the logged-in user, and other helpful information (to be determined). At the bottom of the panel are two buttons, “Login” and “Logout” to login and logout of Kerberos.
- The Event Information Panel (EIP): This panel contains the detailed editable view of the event which is selected in the ELP. It consists of the following:
    - A text field for the event TITLE.
    - A large text field for the event DESCRIPTION.
    - Two specialized “time” text-fields for the start and end time.
    - Pop-up menus (Choices) for GROUP, ACCESS, STYLE, and TYPE.
    - A checkbox for GROUP_WRITABLE.
    - Some means for specifying whether an event repeats (REPETITION).

3.1.5 Description of the Personal Information Manager Component

The window for the PIMC will consist of 5 elements: a menu bar and 4 panels. The bottom panel consists of the identifying banner mentioned above. Clicking on the banner brings up an “About...” window. The panels are as follows:

- The View Selection Panel (VSP): This panel consists of buttons which are used to select a view for the Calendar Display Panel (described below). The buttons are “Month,” “Week”, and “Day.”
- The Calendar Display Panel (CDP): Displays a calendar view based on the granularity selected in the VSP. In “Month” view, displays a monthly calendar in grid form. In “Week” view, displays 7 columns for the days of the week, with times form 8:00am to 8:00pm down the side. “Day” view is just like “Week” only for one day. Duh. There are also three buttons: “Back,” “Today,” and “Forward,” which control the date displayed by the calendar.
- EIP: Just like above. The information is only editable if the user has write permission to the event in question. If the user tries to modify an event for which s/he does not have write permission, s/he will receive a polite but firm reprimand.

4.0 Hardware and Operating System Specifications

ACMS client software will require a Java capable environment and WWW browser. ACMS shall be written under the assumption that all client machines will be Java capable.

A sample compatible system would include a WWW browser supporting the latest release of the Java Virtual Machine and runtime interpreter, as ACMS front-end applications will be written using version 1.1 of the Java development kit.

ACMS back-end systems will consist of a set of one or more Unix based servers for the WWW server and database cores. Server related details and implementation can be found in the Specifications Document.

5.0 General Implementation Requirements

The following items provide a summary of the implementation requirements for ACMS. Though they do not affect the front-end user explicitly, these are listed here for completeness:

1. The user interfaces will be written in Java for increased platform independence. Server side objects will be written in Java or C++. All CGI scripts, if any, will be written in Perl. All Java code will be written to conform with the javadoc utility standards, and APIs for all objects will be generated as specified in Section 1.5 of this document.
2. All event data transported between clients and server will be done using the vCalendar specification. See the vCalendar specifications and whitepaper for more information.
3. All time and date references within ACMS will adhere to ISO 8601 standards.
4. ACMS will be written to support any character set registered with the Internet Assigned Numbers Authority (IANA), with all vCalendar packets defaulting to the ASCII character set. ACMS will consequently support any language as specified in the vCalendar specification.
5. The ACMS database server will conform to recommended transport practices for vCalendar data to allow for enhanced compatibility with other vCalendar enabled systems such as Netscape Calendar.
6. The authentication modules will implement Kerberos Version 4 using a Java interface. The Java Kerberos client will be written to be expandable to newer versions of MIT's Kerberos package or other Generic Security Service APIs if possible. An API will be developed documenting the Java interfaces to Kerberos Version 4.