
Software Requirements Specification

For

Colorado **S**ki **R**acing **M**anagement **S**ystem

In satisfaction of the course requirements in SYST3020 and SYST3510

Prepared by Hirokazu Inoue

University of Colorado at Boulder

Leeds School of Business

Information Systems

First Publication on 11/10/2005

Table of Contents

Software Requirements Specification i
 Colorado Ski Racing Management System
 Prepared by Hirokazu Inoue
 University of Colorado at Boulder
 First Publication: 11/10/2005

Table of Contents ii

Revision History ii

1. Introduction.....1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 2

1.5 References 2

2. Overall Description.....2

2.1 Product Perspective 2

2.2 Product Features 4

2.3 User Classes and Characteristics 8

2.4 Operating Environment 8

2.5 Design and Implementation Constraints..... 9

2.6 User Documentation 9

2.7 Assumptions and Dependencies 9

3. System Features9

3.1 Prospective Athletes Recruitment..... 10

3.2 Team Roster..... 10

3.3 Content Management..... 11

4. External Interface Requirements17

4.1 User Interfaces 17

5. Other Nonfunctional Requirements.....19

5.1 Performance Requirements..... 19

5.2 Safety Requirements..... 19

5.3 Security Requirements..... 19

5.4 Software Quality Attributes..... 19

6. Other Requirements20

Appendix A: Glossary20

Appendix B: Issues List.....20

Revision History

Name	Date	Reason For Changes	Version
2nd Publication	11/18/05	Chapter 4 revised and Chapter 6 completed.	1.1
3 rd Publication	12/03/05	Minor misspelling revised.	2.0

1. Introduction

1.1 Purpose

This software proposal for the Colorado Ski Racing Management System (CSRMS) was created based on consultations with Paul Rozsypal, the Program Director and Head Coach of the USCSA Division Ski Team at the University of Colorado at Boulder. The requirement list in this proposal covers a new management system that will be implemented as a single subsystem of the current team website. The website resides at <http://www.colorado.edu/Athletics/coloradoskiracing/>, and has been designed and managed by the same developer. As an official proposal document of the management system, this is the first version. This document will give an explanation of the system from both a business point of view and a technical point of view.

1.2 Document Conventions

Chapter index is written using **this font** along with its chapter number. The section index is written using **this font** along with its section number. A paragraph is written in this font.

1.3 Intended Audience and Reading Suggestions

This document was prepared for all the associated courses' faculties, for future developers who may wish to further develop this software, and for the Program Director and Head Coach of the USCSA Division Ski Team at the University of Colorado at Boulder. Since this document will contain some technical aspects of this software, separate general usage documentation will be provided for the general users (team athletes and administrators) for simplicity. This document is comprised of 6 chapters including the appendix. The important chapters of this document are chapters 2, 3, and 4. Chapter 2 explains the business model of this software; chapters 3 and 4 explain the system features from a technical point of view.

1.4 Project Scope

CSRMS will implement an organized form of access to the future athlete recruiting information, current athlete registration information, event participation record information, training schedule information, team roster information, and team financial accounting information. Unlike the current paper-based documents used to manage the above information, this software will allow both the athletes and the Program Director to update certain information that is on file. Hence, both users including athletes and the Program Director are able to reference and use up-to-date information to better handle a larger number of athletes.

1.5 References

Current team website: <http://www.colorado.edu/Athletics/coloradoskiracing/>, as of 11/10/2005, developed and managed by Hirokazu Inoue.

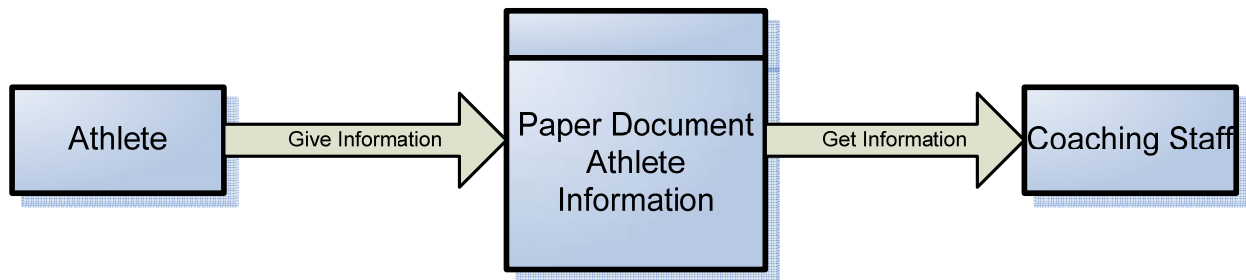
2. Overall Description

2.1 Product Perspective

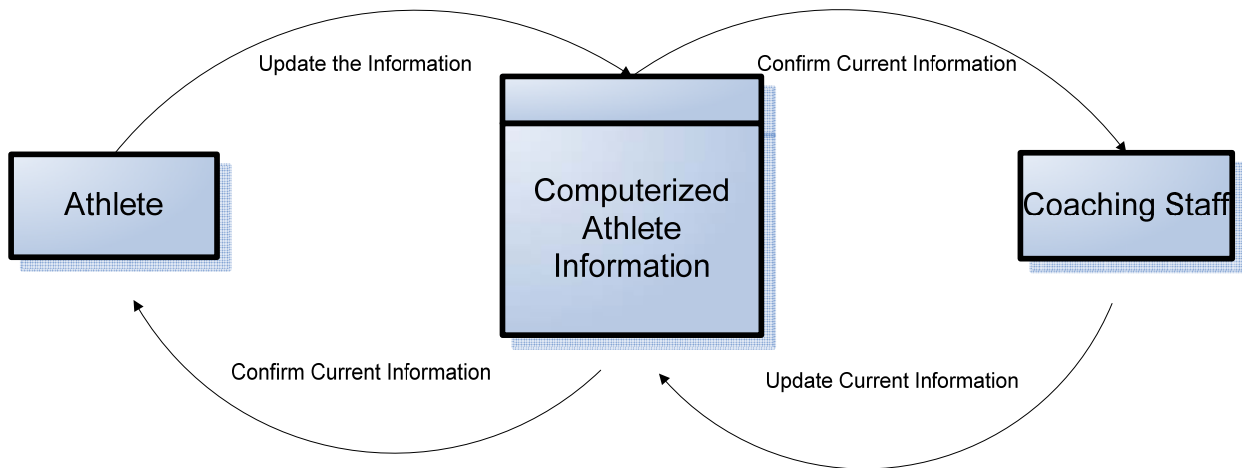
Although ITS and the University of Colorado System at this university have implemented a network based on a large scale campus wide information management system to store and manage information from nearly 40,000 students and faculties, there is no such system on a smaller scale designed for student groups and organizations. Some examples of this large scale, campus-wide management system are the Student Information Systems (SIS), CU Connect, WebCT, and PLUS; some are not developed within the university but purchased from outside sources. While student groups and organizations are generally formed by a smaller number of people, some of these groups have enough members that they need an organized information management format. The USCSA Division Ski Team at the University of Colorado at Boulder, which is registered as a student organization under the name of Colorado Ski Racing, is one of the student organizations with a growing number of people (one of the largest USCSA ski racing programs in the United States). This season, nearly 50 athletes –in addition to a coaching staff— are involved in this group, while there is little organization in terms of accessing timely updated information of the athletes after they

finish registering most of their information on paper. The Colorado Ski Racing Management System evolved after observing some of the content implemented by the previously mentioned management systems used on campus. CSRMS will ensure an organized and secure communication of basic, but necessary and important, information between team athletes and the coaching staff.

Paper Document



CSRMS



2.2 Product Features

1.0: Athlete Recruitment

Prospective athletes will have a form to enter their information. This page will not be access controlled by a username and password. All general public users who are prospective team athletes can enter their contact information. The information collected on this form is passed to a database and stored for review by the Program Director.

2.0: Team Roster

The team roster is provided for any team member and general public users to view. The roster will be dynamically created and updated by referencing the stored athlete information in the database. This page will show only the relevant information for the page to serve as a team roster. Even if the database will have an extensive amount of the athlete's information, the dynamically created team roster will pick only the athlete's name, phone number, and e-mail address. These are collected at the time when an athlete uses (c.f. 3.2.1).

3.0: Team Management

3.0.1: User Login

Only registered athletes and the ski team staff will go through the login procedure. The login form will process username and password information and retrieve its role (either athlete or administrator). Beyond this page, any unauthorized access will be strictly filtered since athlete's information contains sensitive personal material such as their home address and insurance information. Once a login is granted, an athlete or an administrator will be taken to the appropriate page. As a security feature, the IP address and the date of challenge will be stored into the database when a person fails to login.

3.1: User Registration

The user registration feature is provided to register an athlete if an athlete does not already have access to the system. Although the administrator has access to the feature of adding either athlete or administrator, this page will only self-serve athletes for their user registration. This way, an administrator will not need to add users manually for each athlete. Before an athlete proceeds to this user registration, there will be an authentication to register by entering a secret key that is provided by the Program Director after an athlete has paid

the team fee. This user registration form will accept a username, password, and e-mail that are chosen to be registered by the athletes.

3.2: Main Menu (Athlete Mode)

Once an athlete is logged in with their username and password, he/she is taken to a page which contains the following items on the menu:

3.2.1: Update Personal Information

An athlete can update his/her own information including contact information, permanent address, emergency contact, and insurance information.

3.2.2: Payment Status

An athlete can check his/her own payment status. This information is shown based on his/her event participation and team fee payment record that is controlled by the Program Director.

3.2.3: Event Participation

An athlete can indicate his/her participation in the displayed event. Currently displayed events on this page are controlled by the Program Director.

3.2.4: Your Account

An athlete can change his/her password for the login and e-mail address under the user registration. This information will be used for an account information reminder in case an athlete forgets his/her user information.

3.3: Main Menu (Administrator Mode)

When a user registered as an administrator logs in, the user is taken to this page. This page contains tools to control most of the necessary variables and attributes entered into the database.

3.3.1: Prospective Athlete Management

An administrator can add, edit, and delete a list of the prospective athlete information. This is created when a prospective athlete enters his/her information (c.f. feature 1.0). In order to centralize the list of prospective athletes, an administrator can also add prospective athlete information here. The basic view of this list is formed with the athlete's first name, last name, phone number, and e-mail address. There are an additional two buttons next to the list of each prospective athlete: one is a delete button, which deletes a prospective athlete's information; the

other is a detail button, which shows the rest of the information (contact address, current USSA, and FIS points) that were entered by a prospective athlete.

3.3.2: Current Athlete Management

3.3.2.1: Athlete Profile

This is one of the primary tools to manage the current athletes featured in this software. This will show the current athlete's profile (not limited to their user login information) in a very similar manner with feature 3.3.1 (athlete's first name, last name, phone number, and e-mail address). There will be one button located on the side of the list for each athlete (detail). The detail view will allow an administrator to view an athlete's detailed information such as insurance, permanent address, and emergency contact information. Each user created by feature 3.1 will have their profile displayed on this list. In order to delete or add a new athlete, an administrator has to go through the delete a user account; an administrator cannot *only* erase athlete's profile.

3.3.2.2: Search Listing

(* this page can be implemented in c.f. 3.3.2.1. together)

This allows an administrator to retrieve a specified athletes' information on the list. For example, in the case that the Program Director may need to list all the athletes' USSA ID along with their first name and last name for certain events, this feature will only show USSA ID and an athlete's first name and last name, removing all other unnecessary information from view. All attributes from the database table can be chosen as a criterion for the search.

3.3.3: Event Management

The event management function allows administrators to edit the event information that athletes will attend. This information includes the name of the event, date, and cost. At the same time, this function will provide administrators the ability to view the athlete's intention to attend, which was input by athletes from their individual accounts.

3.3.4: Financial Management

There will be 2 separate parts of the financial management functions implemented. One is the team fee and second is the event fee. Each part will have a separate view; titles and attributes of the events are derived from Event Management (3.3.3). This form also allows administrators to edit how much an athlete has paid. Information entered here will be submitted into each athlete's Payment Status (3.2.2).

3.3.5: Roster Management

Roster management has a function to modify the roster on the main team website. Along with this function, this form allows control of which athletes are to be shown or not to be shown on the roster (the default is that all athletes are shown on the team roster).

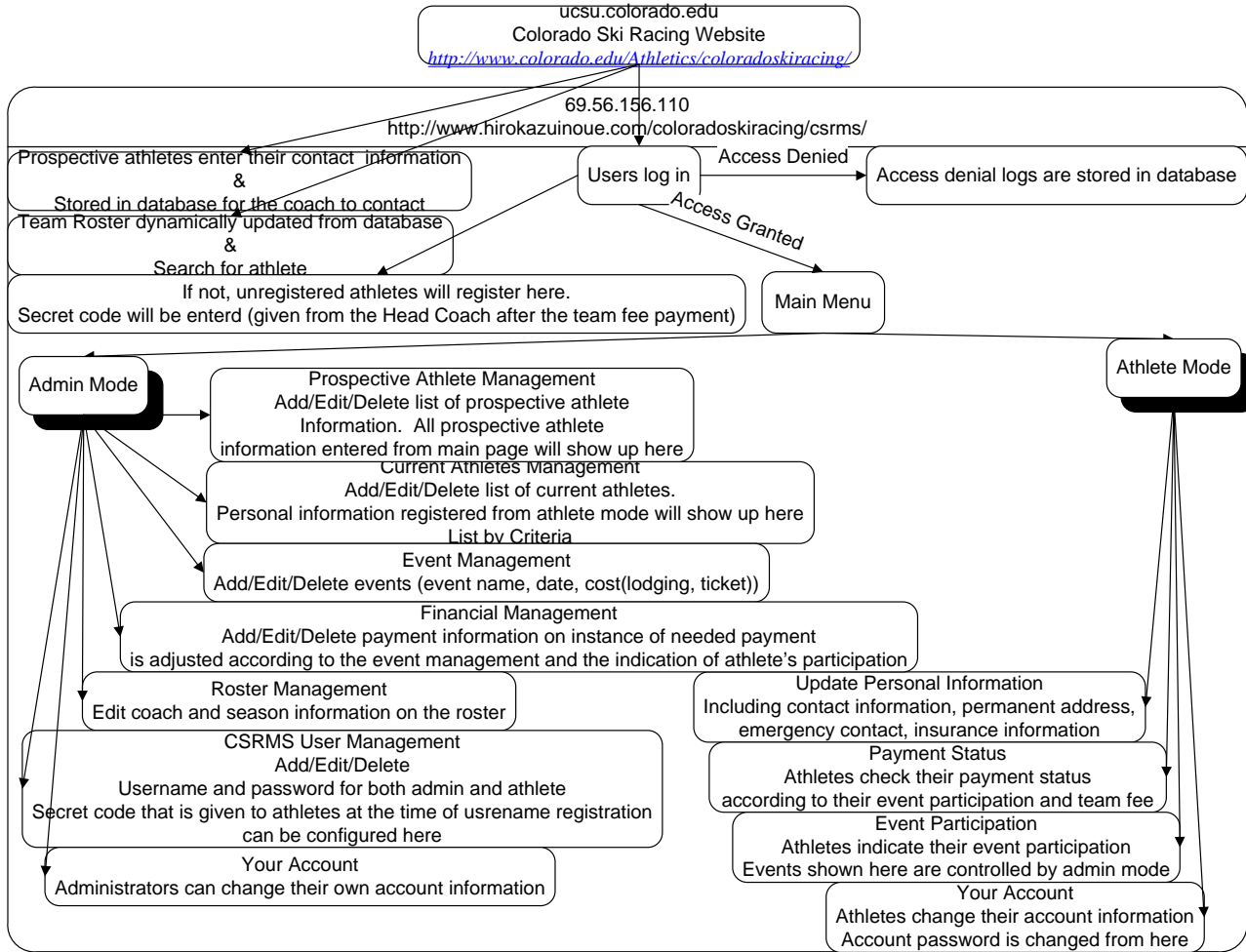
3.3.6: CSRMS User Management

This user management function will allow administrators to add/edit/delete both the administrator accounts and the athlete accounts. If an administrator deletes the user registration, then an athlete's user profile, event participation record, and financial record will be deleted as well. There will be a warning before processing.

3.3.7: Your Account

An administrator can change his/her password for the login and e-mail address under the user registration. This information will be used for an account information reminder in case an administrator forgets his/her user information.

Following is a diagram of product features and flows.



2.3 User Classes and Characteristics

In order to perform the features described in section 2.2, there are two user classes required. They will be an athlete class and an administrator class. All athletes will have an athlete class account; only the Program Director and the site administrator will have access to the administrative console. However, an administrator can be added from their console as needed. Each access class is referred to a certain access level throughout the process. When a login is performed, each user class is taken to their appropriate menu.

2.4 Operating Environment

The CSRMS will run based on Macromedia ColdFusion web technology; CSRMS requires ColdFusion MX version 6 or higher to be installed on the hosting server. The developer will use his

own account at CFDynamics for the development of CSRMS and its operation (currently under www.hirokazuinoue.com). However, this hosting server can be changed at any time as long as there is a sufficient ColdFusion hosting environment.

2.5 Design and Implementation Constraints

It is ideal to have hosting under the domain name of this ski team with a unique domain (Example: www.csrms.net). However, the costs that will incur with the domain registration and hosting plan prevent this operation to be on its own dedicated domain. The ucsu.colorado.edu server, where the current team website resides, does not have ColdFusion installed; this prevents the implementation to be done under the Colorado.EDU domain. However, since the development and implementation is done by a personally paid, outside campus server, there are little constraints that can be caused by the use of the server at the University of Colorado.

In order to match the color scheme of the entire team web site, the same style sheet, and a set of background images as the current team website are used for the overall image design of CSRMS.

2.6 User Documentation

All user documentation links will be provided on the website at the following address. <http://69.56.156.110/HIROKAZUINOUE/coloradoskiracing/csrms/> All user documentations will be in PDF format. Following is the expected amount of user documentation:

1. Administrator's manual for CSRMS
2. Athlete's Manual for CSRMS

Along with the above, any document published for development purposes will be available through the same web site.

2.7 Assumptions and Dependencies

This proposal was made under the assumption of a functional Macromedia ColdFusion hosting environment and a development environment with Macromedia Dreamweaver 8 with local ColdFusion hosting. Although section 2.4 explains that the version MX 6 or above is required, specific versions have not been tested.

3. System Features

3.1 Prospective Athletes Recruitment

Description and Priority

As the feature was described in section 2, this allows prospective athletes to enter their information; the information will be stored in the database for review. For simplicity, this can be implemented at a lower priority.

Priority: Low

Stimulus/Response Sequences

A prospective athlete enters his/her information → Stored inside the database

Functional Requirements

Besides the described features, an implementation of exception handling when prospective athletes do not enter necessary information and the standardized input format for telephone numbers are needed in order to increase the data integrity of the database.

3.2 Team Roster

Description and Priority

The team roster pulls and lists an athlete's profile from the database; this is a first name, last name, phone number, and e-mail address. The information is meant to be used for the team's communication list; a search athlete function should be implemented. This does not have a complex system structure and therefore the priority is low.

Priority: Low

Stimulus/Response Sequences

Roster page opens → Execute query to look up all athletes' name, phone number, and e-mail.

A user enters search keyword → Execute query to look up specific athlete by criteria

Functional Requirements

Exception handling for empty search (when a user enters empty search criteria) is required.

3.3 Team Management

Description and Priority

This is the most essential and complex part of CSRMS. The description of the system is as described in the feature. This function involves the most planning for the systematic logic, so the priority is set high.

Priority: High

Stimulus/Response Sequences

Each number corresponds to the feature number used in section 2.2.

3.0: Team Management

3.01: User Login

A user enters username and password and presses Login button →

Login Success:

Login Success will save USER_ID in the session for later use in the features available through the main menu. The purpose of this is to re-recognize the user's information later. The user is taken to either an administrative console or an athlete console depending on their assigned class.

Login Fail:

Login Fail will first save the client's IP address and its occurrence date in the session. When a user is taken to the login fail notification page, the above information is stored in the database for the administrator's reference. Lastly, this notification page clears the session (both IP address and occurrence date) in the server.

3.1: User Registration

A user presses a button to register from the login main page → a user is then prompted for a secret key entry →

Success:

A user is prompted to further enter their information including username, a password chosen by the user, and his/her e-mail address. Once the register button is pressed after the entry, a notification of successful registration is shown to the user. At the same time, this will send a confirmation e-mail with a username and password to the e-mail address entered into the registration. All users registered through this form are granted an athlete access level.

Fail:

A user is directly taken to the access denial page. This page will take the IP address information and its occurrence date; these are stored in the database.

3.2: Main Menu (Athlete Mode)

When a user registered with an access level of athlete successfully logs in, he/she is taken to this page.

3.2.1: Update Personal Information

An athlete is provided with a form to enter their information: First Name, Last Name, Age, Sex, Year in School, USSA ID, FIS ID, Year in This Team, Home Mountain, Address, City, State, ZIP, Phone Number, E-mail, Parent's First Name, Parent's Last Name,

Permanent Address, Permanent City, Permanent State, Permanent ZIP, Permanent Phone Number, Emergency Contact Person's First Name, Emergency Contact Person's Last Name, Emergency Contact Person's Phone Number, Emergency Contact Person's Relationship, Insurance Provider Name, Insurance Policy Number, Insurance Group Number, and Insurance Contact Phone Number. When an athlete opens this form for the first time after creating a new username and password, a code to initialize the form with matched database entry of the username and ATHLETE_ID (which is internally controlled) is loaded since the above information is stored into several different tables. First time entry and second time update will share the same form. For the first time entry, all values inside the form will be empty except that the database controlled username is loaded during this session.

3.2.2: Payment Status

An athlete is provided with a form that shows the status of their payment. These are the Team Fees and Event Fees. Each payment status is recognized by related database attributes (ATHLETE_ID and USER_ID) that are automatically given by the time of the account creation. Actual accounting information is stored inside a separate table. For example, the EVENTPARTICIPATION table that has many to many relationships between the EVENT table (contains event information such as name, date, and cost) and the ATHLETE table (contains basic athlete information) by EVENT_ID and ATHLETE_ID. This form only allows a viewing of the payment status. There is no function to provide an online payment.

3.2.3: Event Participation

An athlete is provided with a tool to indicate their intent to participate in events such as training scheduled outside the regular days involving cost and ski-racing events that require reservations. An athlete will enter their intent to attend events by choosing from two option buttons (will participate, will NOT participate). The information here is stored into an individual record that is related to the ATHLETE table in the database. Once an indication is accepted by the system, it is stored into the database and the user is not allowed to change his/her record without

contacting the Program Director. When participation is indicated for an event, the cost information will go into the accounting system, which is managed by the Program Director.

3.2.4: Your Account

An athlete is provided with a form to update their information. This form refers to the currently stored user information displayed in the textbox. An athlete is allowed to modify their password and e-mail address only. Once the update button is pressed, the information inside the textbox will overwrite the existing information in the database.

3.3: Main Menu (Administrator Mode)

When a user registered with an athlete access level successfully logs in, he/she is taken to this page. The same file called `main_menu_left.cfm` will have both the Athlete Mode and Administrator Mode, but they are filtered by what is inside the access level session stored at the time of the login.

3.3.1: Prospective Athlete Management

An administrator is first provided with a view of the prospective athlete list when he/she enters this page. People on the list can only be added by function 1.0 or through this management console. When the Program Director does not need the information anymore, an entry can be deleted. When the Program Director needs to add someone to the list, an entry can be made by the add function provided on this page. When the Program Director decides to view/modify the prospective athlete information stored (address and USSA Points), an entry can be modified by the detail view. All functions will be implemented inside one file.

3.3.2: Current Athlete Management

This page can possibly lead to building two separate pages: listing view with the detail/edit button and search the list view of athletes by listing criteria. However,

it is ideal to come up with a page that already has the second function, (c.f. 3.3.2.2) built into it.

3.3.2.1: Athlete Profile (3.3.2.2 included)

An administrator is provided with a list view of the athletes' names, phone numbers, and e-mail addresses (these are retrieved from query). Along with this list, there will be a button to view detail/edit the rest of the information provided by athletes. The form will also provide different views such as name and USSA ID, or name and emergency contact information. The user should be able to modify this view dynamically. There will be a search function implemented on this page. A user can also specify the criteria of the search and only the matching data will be displayed along with a view detail/edit button. This page should not have an Add/Remove User feature since the data model of this system should avoid a user with no profile or a profile with no user as much as possible. Using the form described above, feature 3.3.2.2 can be included on this one page.

3.3.3: Event Management

An administrator is provided with a tool to add/edit/delete events. When an event is constructed, the following data needs to be specified: name of the event, place of the event, date of the event, departure time, total cost, and a note explaining the detail of the cost, address of lodging, etc. When a create button is pressed, a database entry with the above information is made. An administrator can create as many events as they need.

Secondly, an administrator is provided with a summary listing of all the current athletes' intent on attending a certain event. They are updated dynamically as athletes enter their intent for each event by using 3.2.3. An administrator may modify this information when asked by an athlete to do so. Only an administrator may modify the participation information once entered by an athlete.

3.3.4: Financial Management

The Financial Management System on CSRMS depends on 3.3.3 and 3.2.3. This will list the payment and needed payment for each athlete for every event. The data retrieval will be done from the event cost and the team fee associated with each athlete. When an athlete intends on attending, the system will indicate the cost automatically; this can only be corrected by an administrator.

3.3.5: Roster Management

The team roster has two parts to be managed: namely, the index information of the roster (year of season, head coach, assistant coach, team captain's information), the list of athletes, and choosing whether or not to show a particular athlete on the roster (the default is to show them on the roster). However, if an athlete chooses his/her information not to be listed on the roster, an administrator can remove his/her information from the roster without affecting the rest of the management system features. An athlete needs to directly contact the Head Coach in writing, if preferred. This is done by entering the value of either 0 (not show) or 1(show) into the ATHLETE table, which contains the athlete's information; retrieving/modifying it would be done from the menu provided here.

3.3.6: CSRMS User Management

This will add/edit/remove the information stored in the USER table (login information for each user): username, password, and e-mail address. Notice that this information is different from the profile of the users. In addition to adding an athlete, this allows administrators to add another user with an administrative privilege to this system.

3.3.7: Your Account

An administrator can access his/her own login information (username, password, and e-mail address) and modify them. This is done by a simple retrieval of the database in the USER table and an update of the values in this table.

Functional Requirements

Certain exception handlings are required; besides submitting an empty text, exceptions will include a first record when it is entered into the table using the update query mode. A first time input should be taken to the initialization page in the background, then back to the update page. Since users are interacting with the same information regardless of if it is for the first time or updating for the second time, only one form will be used. Initialization of the data entry code is one of the most important functions to be implemented in this software besides handling the empty data submitted by the users from various pages.

In order to avoid users linking directly to the pages with sensitive data, the login and access level check is done even if a user skips the login page. This code will be implemented inside the Application.cfm; files for the athlete and administrator mode are stored in separate folders. This way, the code does not need to be implemented for every page.

4. External Interface Requirements

4.1 User Interfaces

A user types text into the Text Box.



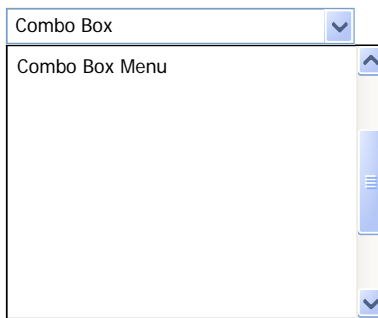
A user clicks on Radio Button. If multiple choices exist, a user may choose only one option.



A user clicks on Check Box. If multiple choices exist, a user may choose as many as applicable.



A user clicks the down arrow on the right side of the Combo Box. The Combo Box Menu shows up. A user may choose only one option under this Combo Box Menu. Once clicked, the choice is entered into the Combo Box.



A user slides the bar and clicks on the up or down arrow to move along the list of options and texts



A user clicks the button to execute a command. The function of the command is often written on the button.



* Note that the design of Text Box, Radio Button, Check Box, Combo Box, Slide Bar, and Button may differ by the version of the operating system, browser, and theme used by the user.

Yellow Button: Used to identify functions that a user may want to pay attention to.

Red Button: Used to identify some critical functions that affect the registration record of all your information on the CSRMS system. Whenever clicking the red button, users must pay attention to the action they are taking.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

CSRMS uses an Access database. Backup of this file should be done at least once a day. As an optimization for query performance, a tool from Access should be used, then put back the file into the server after the optimization. This should increase not only the speed of the query, but also improve the size of the mdb file.

5.2 Safety Requirements

Although a user is less likely to use this software for an extended amount of time, excessive use may harm the user's health.

5.3 Security Requirements

Users will input personally sensitive information into this system. Some should be encrypted with certain algorithm when stored into the database and decrypted when read by certain forms. Some available methods of algorithms in ColdFusion are Backward-Compatible Algorithm, Block Encryption Algorithms, and Password-Based Encryption Algorithms. As the use of Backward-Compatible Algorithms is the least secure option, this method should be avoided as much as possible. The documentation regarding which method was used should not be published publicly. For reporting purposes, the method used will be commented on within the same cfm file.

5.4 Software Quality Attributes

The quality of this software will be represented by the user friendliness of the user interface, the security of the data, the flexibility of manipulating data such as the real-time change of events, and the accuracy of financial records for each athlete depending on the changing events.

6. Other Requirements

Appendix A: Glossary

Should a reader of this proposal encounter difficulty understanding any technical terms, refer to the <http://www.webopedia.com/>.

Appendix B: Issues List

Although an extensive amount of exception handlings was taken into consideration when designing and programming CSRMS, a user may experience some technical difficulties associated with an application error. Should this happen, any comments are to be directly sent to either Hirokazu.Inoue@Colorado.EDU or hirokazu.inoue@gmail.com. Any support of this software will be done by the Web Administrator appointed by the Program Director of Colorado Ski Racing. If the future Web Administrator finds an error on this application and is not able to contact the developer by e-mail, a user may contact to the following address.

Hirokazu Inoue
2305-1 Ohaza Nakagawa
Kitano-machi Kurume-shi
Fukuoka-ken, 830-1101
Japan
Telephone: +81-942-78-2052