Requirements Specification

Requested by:
Dr. Darren Lim
Associate Professor of Computer Science
Siena College
Computer Science Department

Competitive Algorithm Calculation Testing in a Unified System

(C.A.C.T.U.S.)

ExoNET Solutions

Prepared by: Paul Amodeo, Web Master
Tom Delaney, System Admin.
Stephanie Del Belso, Document Analyst
David Purcell, Team Leader
Marco Samaritoni, Web Master

October 29, 2011

C.A.C.T.U.S.
Requirements Specification

C.A.C.T.U.S.
Requirements Specification

Table of Contents

1.1 Product Overview and Summary

The size and complexity of the programming contests held at Siena College has brought about a need for a simple yet powerful solution. Dr Lim has requested that a system be created to prepare and run programming contests simply and easily. The current contest system lacks the ability to keep a precise record of exactly when a team (consisting 1 to many contestants) submits a solution to a problem, the ability for teams to communicate electronically with judges (i.e. uploading problem solutions, checking ambiguities in the instructions, among any other form of legal correspondence dictated by the administrators of the contest). Also, in the current system, a scoreboard must be updated manually on a whiteboard, and judges lack the ability to run alternate arguments (parameters) on a submission, due to the lack of a way to submit contestants' code. Our product will solve these problems by:

- Keeping an electronic timestamp of all submissions made to judges for consideration
- Embedding a compiler into our system so that code may be submitted to judges, and also so the judges can run different data sets (arguments) against the contestants' code
- A digital scoreboard will be created so that all parties involved in the contest can see all teams' solved problems, and the timestamps for correct problem submissions
- All submissions will be stored in a database; also, contestants' work will be saved periodically, so any system or software failure does not end in a complete loss of information for C.A.C.T.U.S..
- A chat window will be available within C.A.C.T.U.S., so that teams and judges can communicate with each other for ambiguities within contest problem sets, and any other pertinent information that can be shared within a given contest.

1.2 Development, Operating, and Maintenance Environments

Our product will be developed and maintained with a variety of resources, from the computers in the Software Engineering lab and Roger Bacon 330, to our personal machines.

SE Lab hardware/software specs:

- Dell ACP x86-based PC
- Intel ® Core™ 2 Duo CPU E7500 @ 2.93 Ghz
- Operating System: Windows Vista Enterprise
- Memory: 305.1 GB of total space 258.6 GB free space
- Ram: 4.00 GB
- Network Adapters: Intel(r) 82567LM-3 Gigabit Network Connection
- Display Chip: Intel (R) 4 series Internal Chipset 2.93 GHz
- Browsers: Mozilla FireFox 4.0.1; Internet Explorer 9; Google Chrome; Macromedia Flash, Macromedia Dreamweaver

Marco's hardware/software specs:

- HDD: 600GB
- Display Adapter: NVIDIA GeForce GTX 285
- DVD/CD ROM:

TSSTcorp CDDVDW SH-S22A SCSI CdRom Device (DVD/CD burner)
ZGFKPUJ 12JKXIR SCSI CdRom Device (DVD/CD burner)

- Logitech HID-Compliant Keyboard
- Logitech HID-Compliant G5 Laser Mouse
- HP 2159 Series Wide LCD Monitor
- Processor: Intel Core i7 CPU 920 @ 2.67 GHz
- Audio: SoundMAX Integrated Digital HD Audio
- Software: Bluej, Netbeans, Microsoft Office, Google Chrome, KompoZer

David's hardware/software specs:

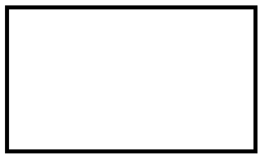
- Operating System: Windows 7 Home Premium 64-bit (6.1, Build 7601) System Model: H55M-S2V
- Processor: Intel(R) Core(TM) i3 CPU
- Memory: 4096MB RAM
- Speed: 4 CPUs @ 3.2GHz
- Gimp 2.6
- Paint.NET 3.5.8
- Audacity 1.3

- Netbeans 7.0.1
- Notepad++ 5.9.3
- Google Chrome 14.0.835.163
- Mozilla Firefox 6.0.2
- WinSCP 4.2.9
- PuTTY 0.60

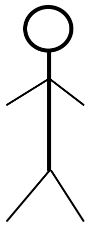
1.3 UML Case Diagram

UML Use Case Diagrams are made to display the users and functionality of a system, in addition to explaining how they all interact with one another. These relationships are explained through the use of lines drawn to and from various actors in the diagram. This diagram represents the User Case Narratives; it is meant to function as a quick reference guide to the User Case Narratives, not as full description of them. Putting our User Case Narratives into a picture gives an overview of our solution, making it easier for anyone who views the document to attain a simple understanding of what our program is going to do.

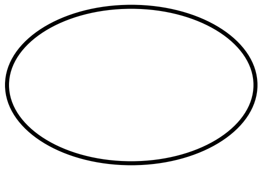
1.3.1 UML Diagram Key



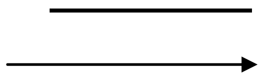
The System Boundary - defines the interactions between the system and the actors (human or non-human) outside the system.



Actors - are users, whether they are human or non-human; they interact with the system and/or each other.



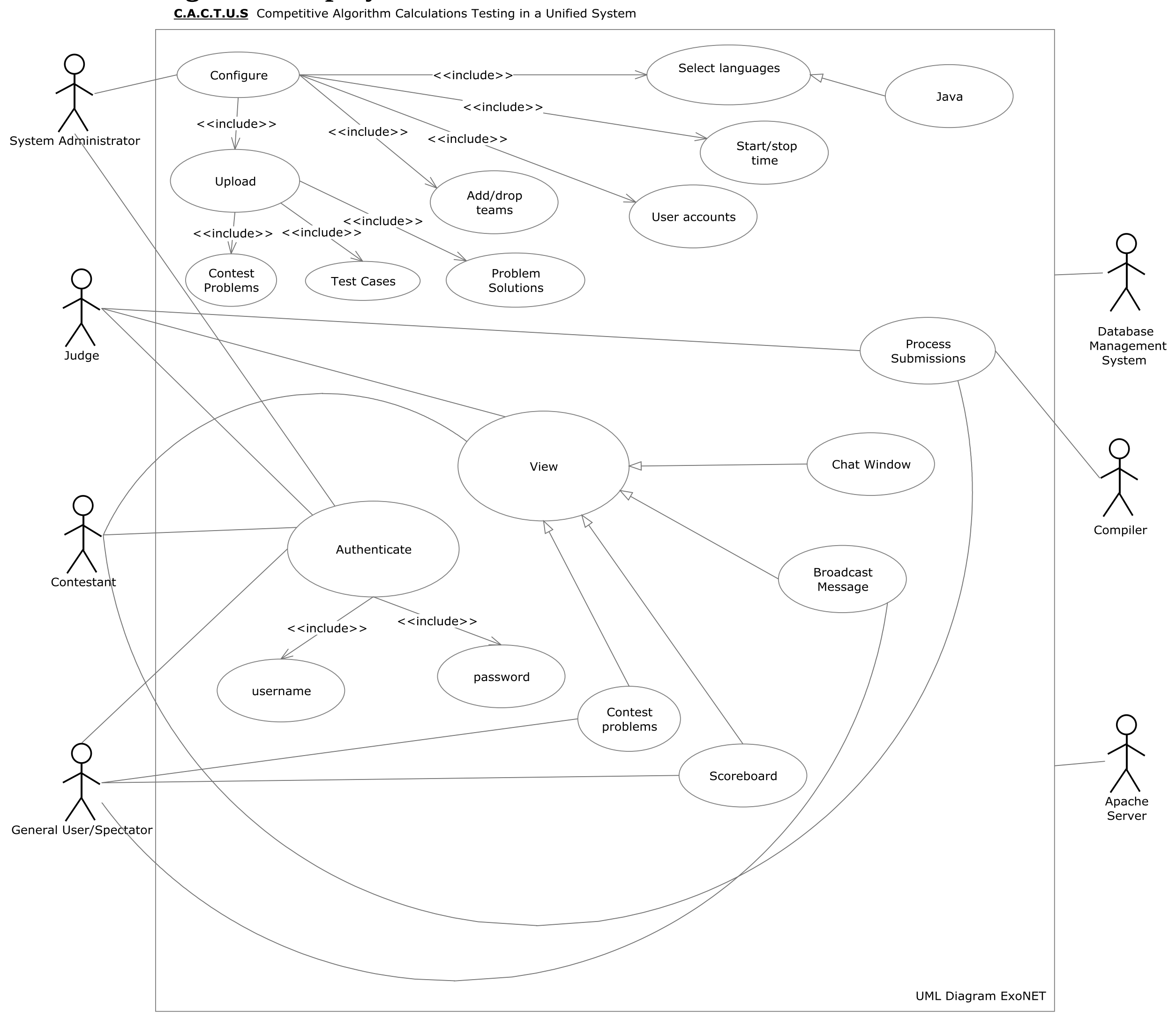
Uses - are activities that interact with actors.



Participations - connect actors and uses to show participation. These lines may connect uses to sub-uses, and actors to uses.

1.3.2 UML Diagram

Our UML Diagram is displayed below.



1.4 User Case Narratives

The following User Case Narratives describe how users will interact with ExoNET Solutions' Competitive Algorithm Calculations Testing in a Unified System (C.A.C.T.U.S.) system. The narratives provide insight into each user's behavior within C.A.C.T.U.S. and allow ExoNET Solutions to discover user requirements. There are four types of users within C.A.C.T.U.S. The System Administrator configures the programming contest environment and has extensive privileges. Judges answer questions and monitor the scoreboard during the programming contest. Contestants participate in a programming contest and make Problem Submissions to contest problems. Spectators can view public information about a running programming contest.

System Administrator:

The System Administrator will be able to access C.A.C.T.U.S. by using a specific username and password that will allow them into an administrative account. Within the administrative account, the System Administrator is prompted to input certain information in order to fully configure the contest environment. This information includes the uploading of contest-problems and problem solutions (including test cases), selection of languages (Java and possibly others), editing start and stop times of the competition, adding and dropping of Contestants, and designating the time for freezing the scoreboard during the competition. The System Administrator is also responsible for establishing usernames and passwords for the Contestant accounts as well as Judge accounts and Spectator accounts.

Judge:

Each Judge is given a Judge account with a username and password by the System Administrator. A Judge is allocated one or more Contestants to supervise for the duration of the contest.

When a Judge logs into C.A.C.T.U.S. using his or her Judge account, he or she will have access to a chat window and the contest scoreboard. A Judge can use the chat window to

send text-based messages to other Judges and Contestants.
The chat window messages can be sent to all Judges and Contestants, or a subset of the Contestants that the Judge was assigned to supervise. A Judge may receive problem submissions from Contestants along with C.A.C.T.U.S.'s analysis of the submissions. The Judge may then review the problem submissions and can decide to approve or reject the submission. A Judge will be able to take over for another Judge that needs to take a leave of absence for any period of time. In this case, the Judge who is taking over will supervise all of the leaving Judge's Contestants.

Contestant:

Each Contestant is given a Contestant account with a unique username and password assigned by the System Administrator. After logging into C.A.C.T.U.S. with the assigned Contestant account, the Contestant will be able to familiarize themselves with C.A.C.T.U.S. and get comfortable with the system before a contest is started. Once a contest has begun, a Contestant will have the ability to: submit problem submissions to C.A.C.T.U.S. in the form of source code, message their supervising Judge using a chat window, view the scoreboard and contest problems.

Spectator:

Each Spectator is given a Spectator account with a username and password assigned by the System Administrator. A Spectator will be able to view the scoreboard at any time during a programming contest. A Spectator will be able to see every Contestant's name, score, and standing. A Spectator will be able to view the contest problems being used in a running programming contest. A Spectator will not be able to communicate with Contestants and Judges. A Spectator will be able to watch a programming contest unfold without interfering with the teams.

1.5 Data Flow Diagrams

Data Flow Diagrams (DFD) are a visual representation of data that is moved, manipulated, and stored by a particular system. Each Data Flow Diagram examines a particular system component at a specific level of detail. The lowest level diagram is the Context Diagram, which presents the system (C.A.C.T.U.S.) and its users. The second lowest level diagram is the Level 0 diagram, which presents all users and their major interactions with the system. Higher level Data Flow Diagrams with increased detail can be constructed by decomposing lower level diagrams. By creating Data Flow Diagrams, ExoNET Solutions gains an increased understanding of both the general structure of the system and (with higher level diagrams) the eventual implementation. Data Flow Diagrams can be easily reviewed by both developers and clients, insuring common understanding and allowing for the discovery of additional requirements.

1.5.1 Data Flow Diagram Legend

The following legend should be used to interpret the Data Flow Diagrams in sections 1.5.2 to sections 1.5.4. ExoNET Solutions' Data Flow Diagrams have no system boundary and they do not follow some of the standard Data Flow Diagram rules. ExoNET Solutions' diagrams do not follow standard conventions for the sake of clarity and readability.

▭	**External Entity (Data Source/Sink)** – External Entities represent sources of data into the system or destinations of data sent by the system.
⬭	**Process** – Processes are system functions that transform or manipulate data. Processes receive and send out data.
→	**Data Flow** – Data flows represent the movement of data.
◫	**Data Store** – Data stores contain persistent data that does not usually move. Data can be saved to and retrieved from Data Stores.

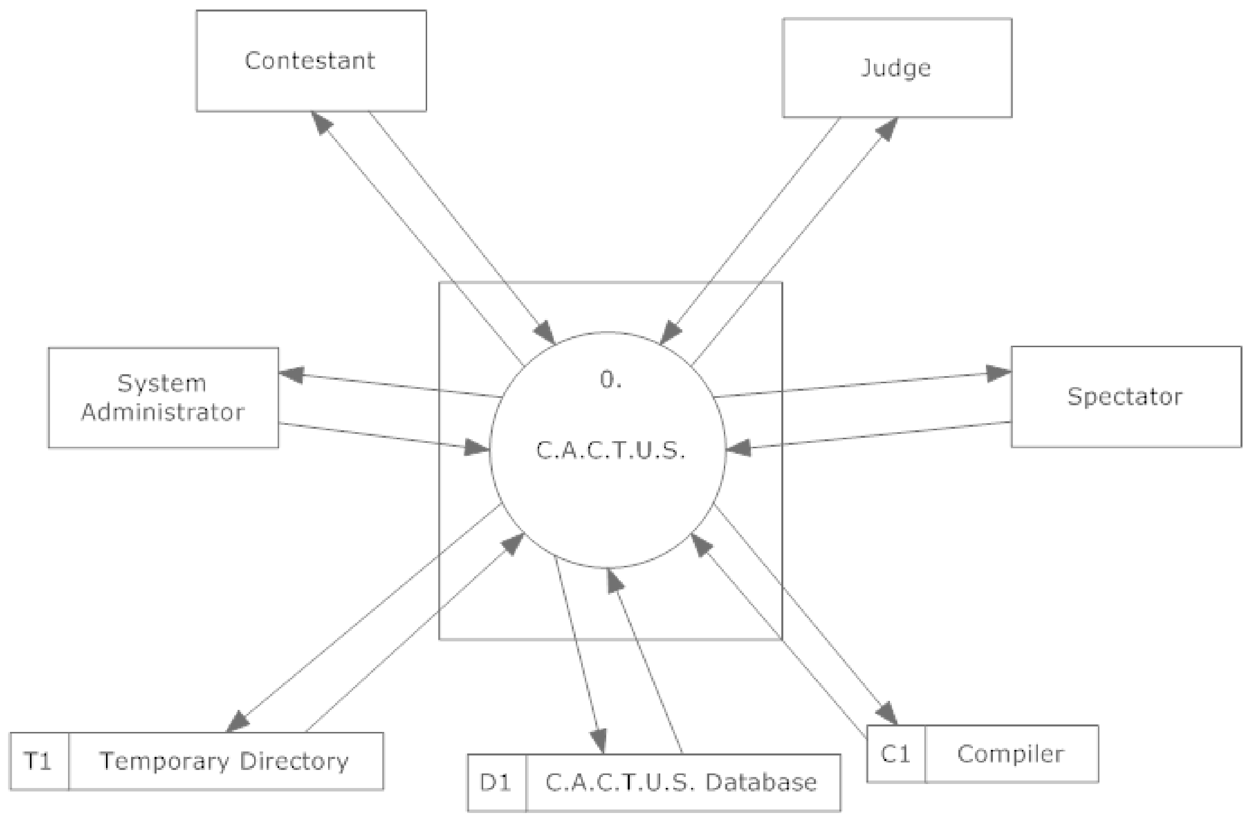
1.5.2 Context Diagram

The Context Diagram describes the major users (data syncs or sources) that interact with the system, namely C.A.C.T.U.S. This includes a temporary directory, database, and a compiler as well. This diagram is the highest level of all Data Flow diagrams. It shows the input and output of the system in relation to all external users.

ExoNET Solutions

Project C.A.C.T.U.S.

Context Diagram



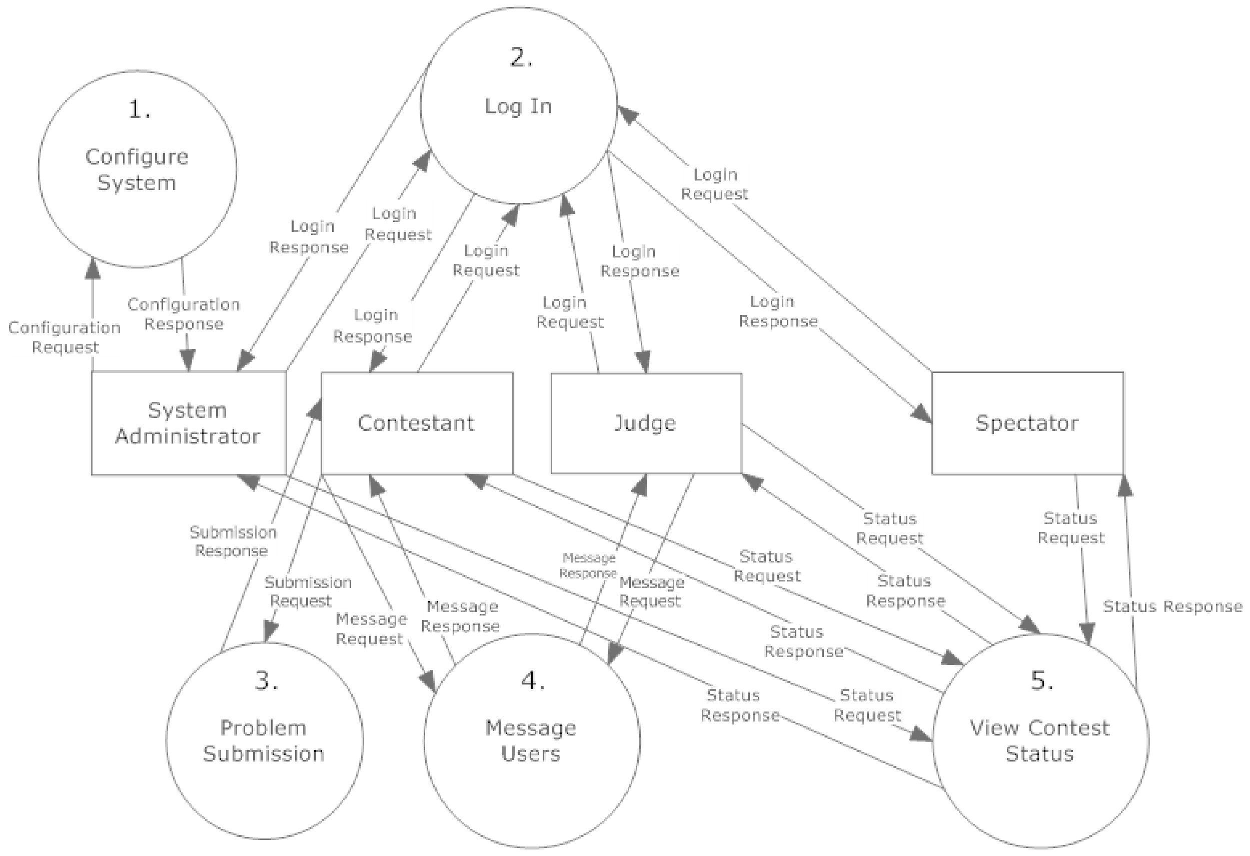
1.5.3 Level 0 Diagram

The Level 0 Diagram is an expansion of the Context Diagram. It shows the major processes (numbers 1-5) that are present within C.A.C.T.U.S. These processes have data flow inputs and outputs to related users describing the type of data involved. There is a request of data and a response of data for every interaction between process and user.

ExoNET Solutions

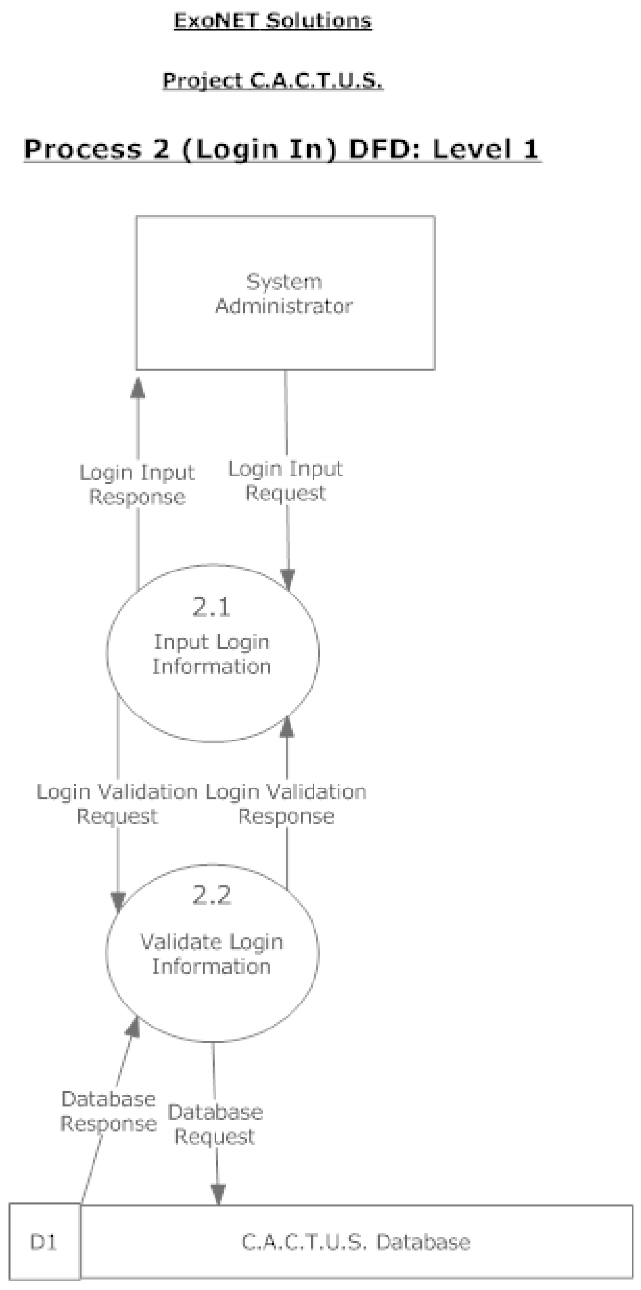
Project C.A.C.T.U.S.

Level 0 Data Flow Diagram



1.5.4 Process 2 (Log In): Level 1 Diagram

Process 2: Level 1 Diagram is an expansion of a specific process in the Level 0 Diagram. In this case, it expands Process number 2: Log In. This specific diagram describes the Log In of the System Administrator user. It shows that two sub-processes are present within Process 2, and makes it clear that communication with a database is needed to validate the login information.



1.6 Functional Requirements Inventory

This is a list of functional requirements. A functional requirement defines a function of a software system or its components. This list is subject to change as ExoNET goes further into the project.

C.A.C.T.U.S (Competitive Algorithm Calculations Testing in a Unified System)-

- Responsiveness(specific ability of a functional unit to complete assigned tasks within a given time)
- Scalability (must be able to operate 30 teams)
- Efficiency (must get responses back within a reasonable time/if infinite loop-time limit exceeded)
    - Time limit exceeded – specific time should be determined per contest (decided by system administrator).
- Robustness (if C.A.C.T.U.S. fails or crashes, users should be able to recover successfully)
    - Saving option
- System must be secure (user's should not be able to access other user accounts)
- C.A.C.T.U.S. will be viewable on multiple web browsers
    - Web browsers include: Mozilla Firefox, Internet Explorer, Google Chrome
- C.A.C.T.U.S will compile programs during the contest
- C.A.C.T.U.S will maintain a scoreboard during the contest.
    - Scoreboard will list all Contestants in decreasing order of standing (determined by # of problems solved and total time)
    - For every team, Scoreboard will show the time-stamp of every solved problem and list the total time for all solved problems
    - Scoreboard will be able to be "frozen" after a set period of time

System Administrator-

- Securely log in and out of C.A.C.T.U.S.
- Upload problem solutions, contest problems and test cases

- Add/drop Contestants during contest creation
- Delete and create new Judge, Contestant, Spectator accounts
  - Administrator will initialize usernames and passwords for each user
- Change start and stop time for a contest
  - Administrator will set freeze time for scoreboard
- Selecting Languages
  - Java will always be a selection choice for the contest

<u>Judge-</u>

- Securely log in and out of C.A.C.T.U.S.
- View contest problems
  - Judge will be able to view test cases
- View Scoreboard
- View chat window
  - Judge will be able to send broadcast messages to all Contestants
  - Judge will be able to send specific messages to their assigned Contestants
- Process submissions
  - Judge will be able to ‘accept’ or ‘decline’ submissions sent to them by their specified Contestants

<u>Contestant-</u>

- Securely log in and log out of C.A.C.T.U.S.
- View contest problems
- View Scoreboard
- View broadcasted messages and chat window
  - Contestants will be able to send messages to their assigned Judge
- Submit and save submissions of source code
- Compile their code

<u>Spectator-</u>

- Securely log in and log out of C.A.C.T.U.S.
- View contest problems
- View Scoreboard
- View only broadcast messages

1.7 Non-Functional Requirements Inventory

This is a list of non-functional requirements. This list is contained with requirements of how our system should be. A non-functional requirement is described to being a quality. This is subject to change as ExoNET goes further into the project.

<u>General System Features-</u>

- Must be easily used (user friendly) – not easily measureable at this time.
- Stability – not easily measureable at this time.

1.8 Exception Handling

The ultimate goal of project C.A.C.T.U.S. is to create a working system that achieves all of the goals outlined in the Problem Definition of the Software Plan. As part of the system goals, project C.A.C.T.U.S. will include fail-safes that, in the event of an exception error, will handle it without the whole system failing. At this stage of the project, the potential threats to the system are still unknown, however, one of the main issues that will require handling will focus on the loss of data due to an unintentional or unexpected exit of the software. Such an event could be due to error on the part of the system or the user. In either case, C.A.C.T.U.S. will be designed so that the backup of data will be automatic so that it may be recovered in the event of an error.

1.9 Early Subsets and Implementation Priorities

The important implementation components of C.A.C.T.U.S. include:

- A user friendly interface
- A secure login for all users
- The ability to submit problems and solutions
- The ability to open a chat between judges and contestants
- The ability for judges to review solutions
- The ability for the program to be updated and maintained

1.10 Foreseeable Modifications and Enhancements

Given that project C.A.C.T.U.S. is projected to be a modified version of the current system used for running programming competitions, there are no major foreseeable modifications to be made. Should Dr. Lim determine it, or us, that further enhancements are made on the system proposed in the Software Plan for project C.A.C.T.U.S., those enhancements will be made accordingly.

1.11 Acceptance Criteria and Testing Requirements

Each module within C.A.C.T.U.S. will be tested individually through a series of unit tests performed on the system. System tests will then be done in order to check to see that the individual modules work together as a whole. Finally the acceptance test will take place to check to see which functional requirements have been met, and which have not. Since Non Functional Requirements cannot be definitively measured, ExoNET Solutions will not be able to apply any formal tests. However, a sound judgment will be made by both ExoNET team members and Dr. Lim to ascertain the effectiveness of the Non-Functional Requirements.

C.A.C.T.U.S. will be rigorously tested on computers running Windows 7 operating systems and have the web browsers: Mozilla Firefox 4, Internet Explorer 9, and

Google Chrome. Testing protocol will be designed by ExoNET Solutions using the Functional Requirements Inventory to develop the Acceptance Test.

1.12 Design Hints and Guidelines

Since it is still very early in the development phase of this project, there are not many design hints and guidelines known at this time. One thing that has been decided by ExoNET Solutions is that C.A.C.T.U.S. will be implemented as an applet imbedded within the web. Navigation for the users shall be implemented within C.A.C.T.U.S. entirely in each user's interface.

Appendices

Appendix 1 Sources of Information

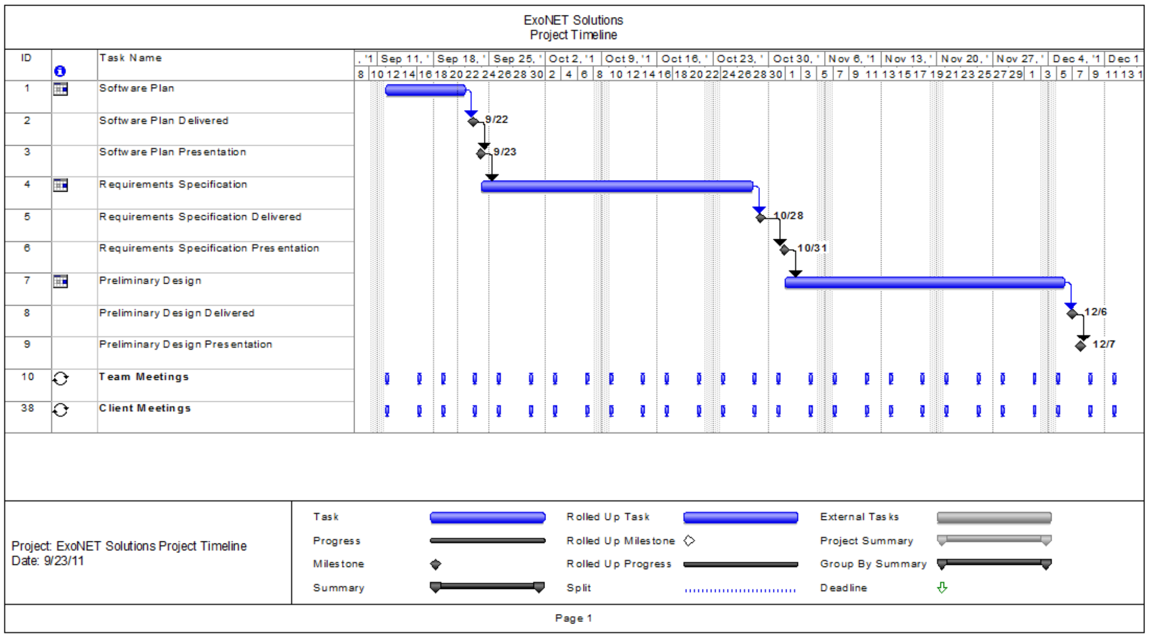
Information found within this Requirement Specification document has been obtained through meetings with our client, Dr. Darren Lim. Information was also obtained through Dr. Lederman's Software Engineering lectures. Information has also been collected from various internet resources, as well as requirement specification documents from previous years. Lastly, we use each team members' knowledge and skills from past experiences.

Appendix 2 Glossary of Terms

Apache HTTP Server (Web Server) - Referred to as Apache, it is web server software notable for playing a key role in the initial growth of the World Wide Web.
C++ - (pronounced "see plus plus") is a statically typed, free-form, multi-paradigm, compiled, general-purpose programming language. It is regarded as an intermediate-level language, as it comprises a combination of both high-level and low-level language features.
CACTUS - Java Open Language Toolkit definition project -
A project aimed at providing integrated system for computer programming contests hosted at Siena College.
Cascading Style Sheets (CSS) - A style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language.
Chrome – An Internet browser designed by Google.
Conflict – When an activity can't be scheduled due to room use, weekend, and one resource being currently in use.
Constraint – When the client specifies that a certain resource must be used, or the project has to be done in a certain way, using certain specifications.
Data Flow Diagram - A graphical representation of the "flow" of data through an information system
Database - An organized collection of data for one or more uses, typically in digital form.
Dreamweaver – A web development application.
Dropbox - A Web-based file hosting service operated by Dropbox, Inc. which uses cloud computing to enable users to store and share files and folders with others across the Internet using file synchronization.
Firefox – AnInternet browser designed by Mozilla.

Gantt Chart - A type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project.
HTML (HyperText Markup Language) - The predominant markup language for web pages. It is written in the form of HTML elements consisting of "tags" surrounded by angle brackets within the web page content. It is the building blocks of all basic websites.
HTTP (Hypertext Transfer Protocol) - a protocol used to transfer hypertext requests and information between servers and browsers.
Internet - A global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide. It is a *network of networks* that consists of millions of private, public, academic, business, and government networks, of local to global scope, that are linked by a broad array of electronic and optical networking technologies.
Internet Explorer (IE) – An Internet browser designed by Microsoft.
Java - a high-level, object-oriented computer programming language used especially to create interactive applications running over the Internet.
JavaScript - An implementation of the ECMAScript language standard and is typically used to enable programmatic access to computational objects within a host environment.
MySQL - A relational database management system that runs as a server providing multi-user access to a number of databases.
PHP (PHP Hypertext Preprocessor) - A widely used, general-purpose “server side” scripting language that was originally designed for web development to produce dynamic web pages.
Ruby™ - A Proprietary, dynamic, reflective, general purpose object-oriented programming language that combines syntax inspired by Perl with Smalltalk-like features.
Spiral Model - A software development process combining which elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts.
SQL- structured query language: a computer programming language used for database management
UML Use Case Diagram - a type of behavioral diagram to present a graphical overview of the functionality provided by a system.
Waterfall Model (Classic) - The Classic Waterfall Model is a sequential software development model in which development is seen as flowing steadily downwards (similar to a waterfall) through the phases of requirements analysis, design, implementation, testing, integration, and maintenance
WinZip - A proprietary file archiver and compressor for Microsoft Windows,
XHTML (eXtensible Hypertext Markup Language) - A family of XML markup languages that mirror or extend versions of the widely used Hypertext Markup Language (HTML), the language in which web pages are written.
XML (Extensible Markup Language) - A set of rules for encoding documents in machine-readable form. To create a tagging scheme that allows elements of a document to be marked according to their content rather than their format.

Appendix 3 Gantt Chart (Timeline)



ExoNET Solutions
Project Timeline

ID	Task Name
1	Software Plan
2	Software Plan Delivered
3	Software Plan Presentation
4	Requirements Specification
5	Requirements Specification Delivered
6	Requirements Specification Presentation
7	Preliminary Design
8	Preliminary Design Delivered
9	Preliminary Design Presentation
10	**Team Meetings**
38	**Client Meetings**

Project: ExoNET Solutions Project Timeline
Date: 9/23/11

Task		Rolled Up Task		External Tasks	
Progress		Rolled Up Milestone	◇	Project Summary	
Milestone	◆	Rolled Up Progress		Group By Summary	
Summary		Split		Deadline	⇩

Page 1