

Software Plan

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

September 22, 2011

**C.A.C.T.U.S.
Software Plan**

Table of Contents

SYSTEM DEFINITION	1
Section 1: Problem Definition	1
Section 2: System Justification	1
Section 3: Goals for the System and on the Project	1
Section 4: Constraints on the System and Project.....	1
Section 5: Functions to be Provided (Hardware and Software/People)	2
Section 6: User Characteristics	2
Section 7: Development/Operating/Maintenance Environments.....	2
Section 8: Solution Strategy	3
Section 9: Priorities of System Feature	3
Section 10: System Acceptance Criteria	3
Section 11: Sources of Information	3
 PROJECT PLAN	 4
Section 1: Life-cycle Model	4
Section 2: Organizational Structure	5
Section 3: Preliminary Staffing and Resource Requirements	5
Section 4: Preliminary Development Schedule	6
Section 5: Project Monitoring and Control Mechanisms	6
Section 6: Tools and Techniques to be Used	6
Section 7: Programming Languages	6
Section 8: Testing Requirements	6
Section 9: Supporting Documents Required	6
Section 10: Manner of Demonstration and Delivery	7
Section 11: Sources of Information.....	7
 APPENDICES	 8
Section 1: Annotated Bibliography.....	8
Section 2: Team Resumes	9-13
Section 3: Glossary of Terms	14

SYSTEM DEFINITION

1.1 Problem Definition:

Programming contests have been becoming increasingly popular in the recent years. Schools, from secondary to collegiate, across the country participate in such competitions, whether it is at a local or national level. With more programming languages coming about in the last couple decades, these contests continue to grow in size and difficulty. As these competitions expand, several different programs have been developed in an attempt to better the way the contestants participate, and the way the judges score. Currently, the contests taking place at Siena College use the personal software developed by the system administrator. This system poses several problems and issues for those who use the software. Our client, Dr. Darren Lim, has requested a program that will enhance the capabilities of the contestants and judges in order to improve the programming competition overall. He would like a software system that encompasses all aspects of a typical computer science programming contest. These aspects include: the ability for teams to submit answers, ability to upload problems, ability to judge answers, ability to give feedbacks to teams, ability to run a scoreboard in real time, and some abilities to start and stop a contest and determining the ranking of teams.

1.2 System Justification:

The purpose of our software is to enhance the programming contests hosted at Siena College by pinpointing specific flaws within the current software. The new system will help to alleviate any unnecessary problems that occur throughout the competition. It should permit teams to submit their solutions through the system quickly and easily, resulting in a score being recorded on a constantly updated scoreboard. The software will also contain a permanent history of all past solutions and problems entered into the contest. In addition, it will improve the communication between contestant and judge, allowing the judge(s) to send messages to one team or many teams at once, while the contestants can send questions/responses through a “chat room”-like system. This software will improve the overall efficiency of Siena College’s programming contests.

1.3 Goals for the System and Project:

The goal for our project is to develop and implement an applet, with the possible option for making it a web-based system that allows users, or contestants, of a programming contest to easily submit their programs. The software must accept at least the Java programming language, but the team will consider allowing other languages. It will also enhance the communication between contestants and judges. The new system will improve the current contests in order to become a more reliable resource to be used within the Siena College Computer Science Department.

The team goals include broadening our knowledge of several different fields within computer science. Working in collaboration with one another is a very important goal we have set as well. We hope to successfully develop a product that will suffice our client's wishes and there for be successful as a software engineer.

1.4 Constraints on the System and on the Project:

The limitation of communication between judges and contestants is one constraint. This includes the specificities allowed in the messages sent from judges to contestants and vice versa. A second constraint is that if the system becomes web-based, we must take into consideration the reliability problems involving internet connection. Another constraint will be programming languages accepted by our software. Finally, the deadline for this project is April 2012.

1.5 Functions to be provided

- multiple teams will be able to upload responses/files onto servers
- judges will be able to review teams' solutions
- able to broadcast a message a message to all teams
- able to send messages to specific teams
- will have a database containing contest history, as well as information on past and current teams
- will have a scoreboard with real time score
- will save scores automatically

1.6 User Characteristics

This software would be the primary means for both contestants and judges of programming contests. It is an interface for the contestants to submit their solutions and get fast results, as well as allow judges to review those solutions at a more comfortable pace. However, the primary purpose of the software is to create an automated response to contestants' answers, furthering simplifying the role of the judges.

1.7 Development/Operating/Maintenance Environments

- **Development**

The development environment for this software will include all hardware and software available in the software engineering lab as well some of our own personal machines.

See section 2.3 for specs.

- **Operating**

The operating environment for this software will be any Mac or PC that has internet capability and access our server.

- **Maintenance**

The maintenance environment will consist of all hardware and software used to create our software.

1.8 Solution Strategy

Our team will follow the Sequential Linear Model (Classic Waterfall Model) in order to fulfill all requirements and document our progress as requested by our client, Dr. Darren Lim. This model (seen in more detail in section 2.1) entails the following endeavors:

- Define and Obtain Requirements
- Development of Design
- Implementation/Coding of Project
- Testing the Solution
- Installation and Maintenance of System

1.9 Priorities of the System Feature

This software will bridge the communication gap between judges and contestants by creating an interface that supplement s existing systems. Some of the features it will include are: a real time scoreboard, some form of chat room or a message board where judges can either post announcements to teams collectively or individually, it will accept the Java programming language, judges will be able to bring up contestants' code and review it, and there will be a method of auto saving data in the event of a crash or a system failure. One of the primary features of this software will be the automated response to a team's solution. This will give programming contests more accurate results because a time stamp will be given with every correct solution, eliminating any uncertainties as to which team solved it first.

1.10 System Acceptance Criteria

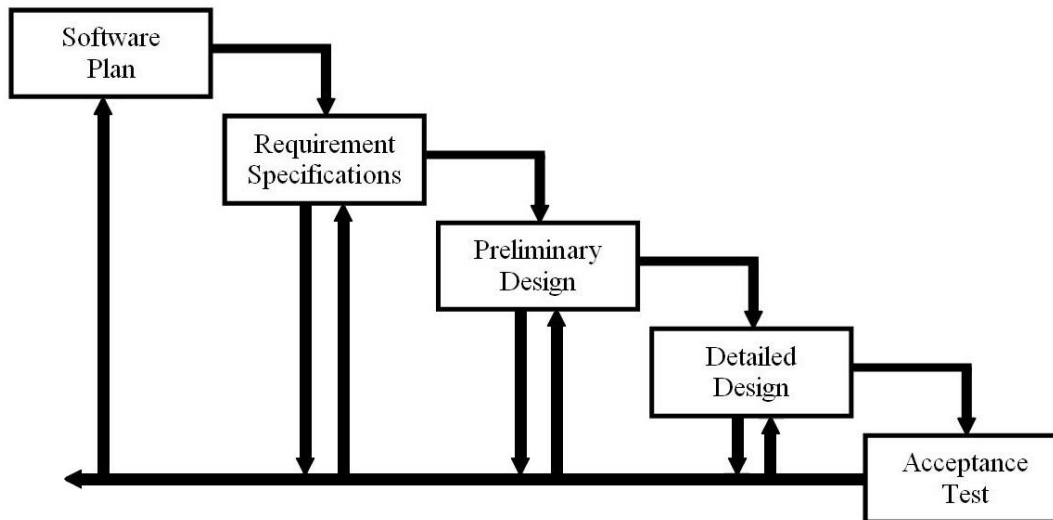
CACTUS will have the following:

- At least Java as a programming language for contestants to use
- Proper communication between Judges and contestants(Teams)
- Handle 30-40 contestants

1.11 Sources of Information

We obtained information from background research found on the internet and in articles, client meetings with Dr. Darren Lim, other experiences from programming contests, and documents or notes from class lectures given by Dr. Timothy Lederman.

2.1 Life-cycle Model: Waterfall Model (Linear Sequential)



Software Plan:

ExoNET Solutions will take a look at the problem during team meetings and client meetings, and then make a Problem Definition. Once the Problem Definition is appropriately made then ExoNET will create a solution.

Requirement Specification:

After client meetings and other use of resources, there should be more in depth specifications.

Preliminary Design:

ExoNET will use the specifications made in Requirement Specifications to create a basic design

Detailed Design:

This phase will start more into the spring semester of 2012. We will be able to modify our basic design into a detailed design.

Acceptance Test:

Most of the programming and testing will happen in this phase as we are getting ready to finish our program. This is the last phase unless there is a change needed that will cause ExoNET to move back a step

2.2 Organizational Structure

ExoNET Solutions consists of the following team members:

<u>Name</u>	<u>Email</u>	<u>Phone Number</u>
Paul Amodeo	pv22amod@siena.edu	(518) 439-3492
Thomas Delaney	tr20dela@siena.edu	(631) 782-5816
Stephanie Del Belso	sl19delb@siena.edu	(518) 258-7051
David Purcell	da14purc@siena.edu	(518) 364-8245
Marco Samaritoni	me10sama@siena.edu	(802) 377-8055

ExoNET Solutions is structured in the following manner:

Team Leader

David Purcell

Coordinates the team and ensures that work is properly distributed and done on time. Manages client meetings, team meetings, and tracks team progress and productivity.

System Administrator

Thomas Delaney

Monitors and maintains the computer software on the lab station. Updates existing software and installs new software when necessary.

Document Analyst

Stephanie Del Belso

Manages all team documents and meeting notes for both internal records and client deliverables. Leads the design and distribution of team documents.

Web Master

Marco Samaritoni

Designs and maintains the team website in the aspects of both content and style.

Assistant Web Master

Paul Amodeo

Designs and maintains the team website in the aspects of both content and style.

ExoNET Solutions provides a collaborative working environment. Team members may call upon each other for assistance if need be. This ensures a timely work flow and fosters creative solutions. Important decisions will be made by the entire group and ruled by majority vote. The Team Leader has the final word in the case of a tie.

2.3 Development Environment

ExoNET's development environment is as follows:

SE Lab hardware/software specs:

- Dell ACP x86-based PC
- 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:

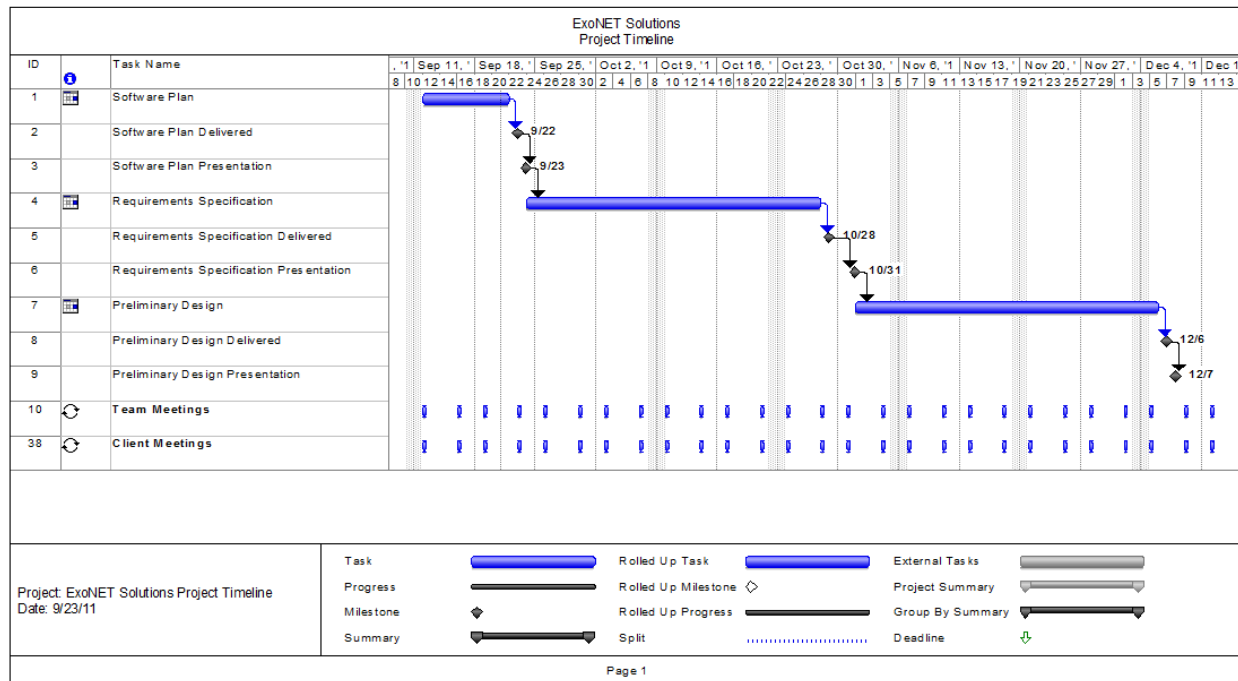
- HDD: 600GB
- Display Adapter: NVIDIA GeForce GTX 285
- DVD/CD ROM:
TSSTcorp CDDVDW SH-S22A SCSI CdRom Device (DVD/CD burner)
ZGFKPUJ 12JKXIR SCSI CdRom Device (DVD/CD burner)
- Logitech HID-Compliant Keyboard
- Logitech HID-Compliant G5 Laser Mouse
- HP 2159 Series Wide LCD Monitor
- Processor: Intel 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

2.4 Preliminary Development Schedule

The following image depicts the timeline that ExoNET Solutions will be using to ensure a smooth and timely progress of the project. Important dates, team meetings, client meetings, and due dates for documents and presentations are included to provide a clear sense of direction.



2.5 Project Monitoring and Control Mechanisms

To guarantee a clear understanding of the problem and work to be done, our team will meet twice a week for both client and team meetings. Having two client meetings every week gives our team time to communicate questions and discuss answers critical to the project's success. ExoNET Solutions team members will come together at least twice a week to discuss the state of the project and assign individual and group work. Our team will communicate frequently through email and use client feedback and project milestones to guide our progress.

2.6: Tools and Techniques to be Used

We will be using Adobe Dreamweaver, Oracle and Microsoft Office software. Our team will be using Dreamweaver and Kompozer for web design. For Integrated Development Environments, we will be using Bluej, Netbeans, and Eclipse. We will be running Oracle scripts for our DBMS. Management of the system will be performed through an applet and possibly the internet and Oracle. We will be using techniques learned from our Software Engineering course as the basis for the development of our project. We will also be using many of the programming strategies and methods gained from prior Computer Science courses to aid in the design and overall functionality of our software.

2.7: Programming Languages

In order to develop the unified contest system, we will be using Java, JQuery, CSS, HTML, Ruby, PHP, and SQL.

2.8: Testing Requirements

We will be using various techniques to compare the usefulness and reliability of our project compared to similar applications. We will collaborate weekly to test the viability of our software, including presentations to our client, Darren Lim.

2.9: Supporting Documents Required

We will be presenting our Software Plan (Problem Definition and Project Plan) on Friday, September 23rd, 2011. Documentation for the Software Plan will be finished on Thursday, September 22nd. The Software Requirements Specifications and Preliminary design will be documented in October and December of this year, respectively.

Additional documentation on the Detailed Design will be provided to the client in the Spring of 2012

2.10: Manner of Demonstration and Delivery

We will give a number of presentations during the development of this project. These presentations will consist of verbal descriptions of power point slides made by the team that will demonstrate the progress of our project. Date of presentations to the clients and the delivery of documents to the client are as follows:

- 1) Project Definition/Project Plan – September 22, 2011
- 2) Project Definition/Project Plan Presentation – September 23, 2011
- 3) Software Requirements Specifications Document– October 28, 2011
- 4) Software Requirements Specifications Presentation– October 31, 2011
- 5) Preliminary Design Document – December 6, 2011
- 6) Preliminary Design Presentation – December 7, 2011

2.11: Sources of Information

The major source of information for this section of the project resulted from meetings with our client, Dr. Darren Lim. Other sources of information include Dr. Lederman's Software Engineering lectures, as well as previous Software Engineering teams' projects and documents. Lastly, we will be using each team members' knowledge and skills from past experiences.

APPENDICES

Annotated Bibliography

Arefin, A. S., Rahman, M. A., Sharna, S. A., Mahmud, S, & Kaykobad, M. (2005). Secured Programming Contest System with Online and Real-time Judgment Capability. *8th International Conference on Computer and Information Technology*, 5 pgs.

This work presents an online judge system to be used in programming contests. Such a system must provide a secure execution environment that can execute arbitrary code submitted by any programmer, without compromising confidential information or system stability. This system has been implemented in C#.NET language, but has been projected with extensibility for new programming languages in mind. It has the capability of arranging simultaneously a contest in a real-time environment and online programming contest.

Bentiba, A., Zemerly, M. J., Mansoori, M. A. (2010). Smart Web Based Programming Contests Management Tool. *Computational Intelligence and Modern Heuristics*, 241-250.

Because of PC²'s limitations and weaknesses, These scholars from Khalifa University of Science decided to design an alternative to this system. They call it Wide Area Contests System (WACS) to allow people from different cities or countries to participate in different contests not only programming and language contests.

Conlon, M. P. (2005). RockTest: A Programming Contest Management System. *Consortium for Computing Sciences in Colleges*, 27-35.

RockTest is a software system to facilitate the activities of contestants and judges in computer programming competitions. RockTest was developed out of experience running programming competitions for high school students as a recruiting tool for the Computer Science Department. It has been proven reliable and has received enthusiastic reviews from the competitors, judges, and contest administrators at two universities who have used it.

Other sites referred to:

<http://www.ecs.csus.edu/pc2/>

Information about PC² for background research

<http://domjudge.sourceforge.net/>

An automated judge system to run programming contests

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.

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 – 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.

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 – Internet 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) – Internet browser designed by Microsoft.

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.

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.

JOLT - Java Open Language Toolkit definition project

A project aimed at providing a freely available and redistributable implementation of Sun Microsystems's Java language and tools.

Java - a high-level, object-oriented computer programming language used especially to create interactive applications running over the Internet.

C++ - (pronounced "cee 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.

SQL -

structured query language: a computer programming language used for database management