

# Requirements Specification

## Redesign of the Software Engineering Site (R.O.S.E.S.)

### **Requested by:**

Dr. Timoth Lederman  
Professor  
Department of Computer Science  
Siena College

### **Delivered By:**

*Code Shark Solutions*

### **Prepared By:**

Kurt Greiner  
Daniel Rotondo  
Ryan Godfrey  
Rebecca Wilson  
Akeem Shirley  
Brittany Lintelman

November 3, 2010

# R.O.S.E.S.

## Requirements Specification Table of Contents

1.1	Product Overview and Summary .....	3
1.2	Development, Maintenance, and Operating Environment .....	3
1.3	UML Use Case Diagram .....	4
1.3.1	UML Use Case Legend .....	5
1.3.2	UML Use Case Diagram.....	6
1.4	Usage Case Narratives.....	7
1.4.1	Course Instructor Usage Case Narrative.....	7
1.4.2	Software Engineering Student Usage Case Narrative .....	7
1.4.3	Recommenders Usage Case Narrative .....	8
1.4.4	Family/Friend Usage Case Narrative .....	8
1.4.5	Alumnus/Alumna Usage Case Narrative.....	9
1.5	Data Flow Diagrams .....	9
1.5.1	Data Flow Diagram Legend .....	10
1.5.2	Context Diagram.....	11
1.5.3	Level 0 Diagram .....	12
1.5.4	Level 1 Diagram.....	13
1.6	Functional Requirements Inventory.....	14
1.6.1	The General User.....	14
1.6.2	The Course Instructor .....	15
1.7	Non-Functional Requirements Inventory.....	16
1.8	Exception Handling .....	16

1.9	Early Subsets and Implementation Priorities .....	17
1.10	Foreseeable Modifications and Enhancements.....	17
1.11	Testing Requirements and Acceptance Criteria.....	17
1.12	Guidelines .....	17

## ***1.1 Product Overview and Summary***

With the addition of Software Engineering to the computer science curriculum and advancements in website design, Dr. Timothy C. Lederman has decided that he wants the Software Engineering course website to be overhauled. The current site is inconsistent in its menu structure, lacks search capabilities and is not on par with other websites on an aesthetic level. The site needs these aspects to be updated so that users such as future employers, recommenders, students, course instructors, family and friends can use the website as needed. Each page will have a consistent main menu, each item on the main menu, once clicked, will display a submenu that will have options for the user to choose. A search feature will be added so that students and teams can be looked up with ease, allowing recommenders and future employers to gain some background about the student that they are referring or interviewing, respectively. The websites over all look and organization will be redone to accurately reflect the courses in-depth content and mission. These improvements will help to make the website more appropriate for the course it is representing.

## ***1.2 Development, Maintenance, and Operating Environment***

*Code Shark Solutions'* development environments are as follows:

Server:

Operating System:       CentOS (Linux) Release 5.2 (Final)

Server Name:       oraserv.cs.siena.edu

CPU Type:               x86\_64

Web Server: Apache Version 2.2.9  
PHP Version: 5.2.6  
Database: MySQL Version 5.0.45; Oracle Version 9.0.1

Macintosh Computer:

Operating System: Mac OS X 10.6.4  
Model: iMac 5,1  
Processor: Intel Core 2 Duo  
Speed: 2 GHz  
Memory: 1 GB  
L2 Cache: 4MB

Windows Computer:

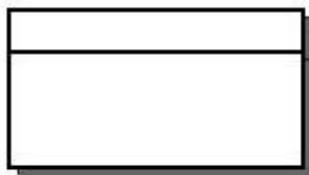
Operating System: Windows Vista Enterprise (6.0, Build 6002)  
Model: Dell OptiPlex 760  
Processor: Intel Core 2 Duo  
Speed: 2.93 GHz  
Memory: 3 GB

The website will be functional across all major browsers: Internet Explorer 8, Firefox 3.6, Google Chrome 7 and Safari 5.

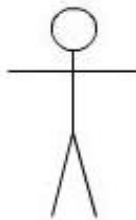
### ***1.3 UML Use Case Diagram***

UML Use Case Diagrams are used to show how different users will interact with a system. The users or 'Actors' interact with the system through 'Uses.' Lines are drawn between Uses and Actors to demonstrate a relationship.

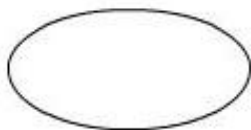
### 1.3.1 UML Use Case Legend



**System Boundary**- where users will interact between uses inside the system and actors outside.



**Actors**- Human or non-human users which interact with uses in the system



**Uses**- the activities that interact with actors outside the system.



**Includes Arrow**- Items that are included in a use.

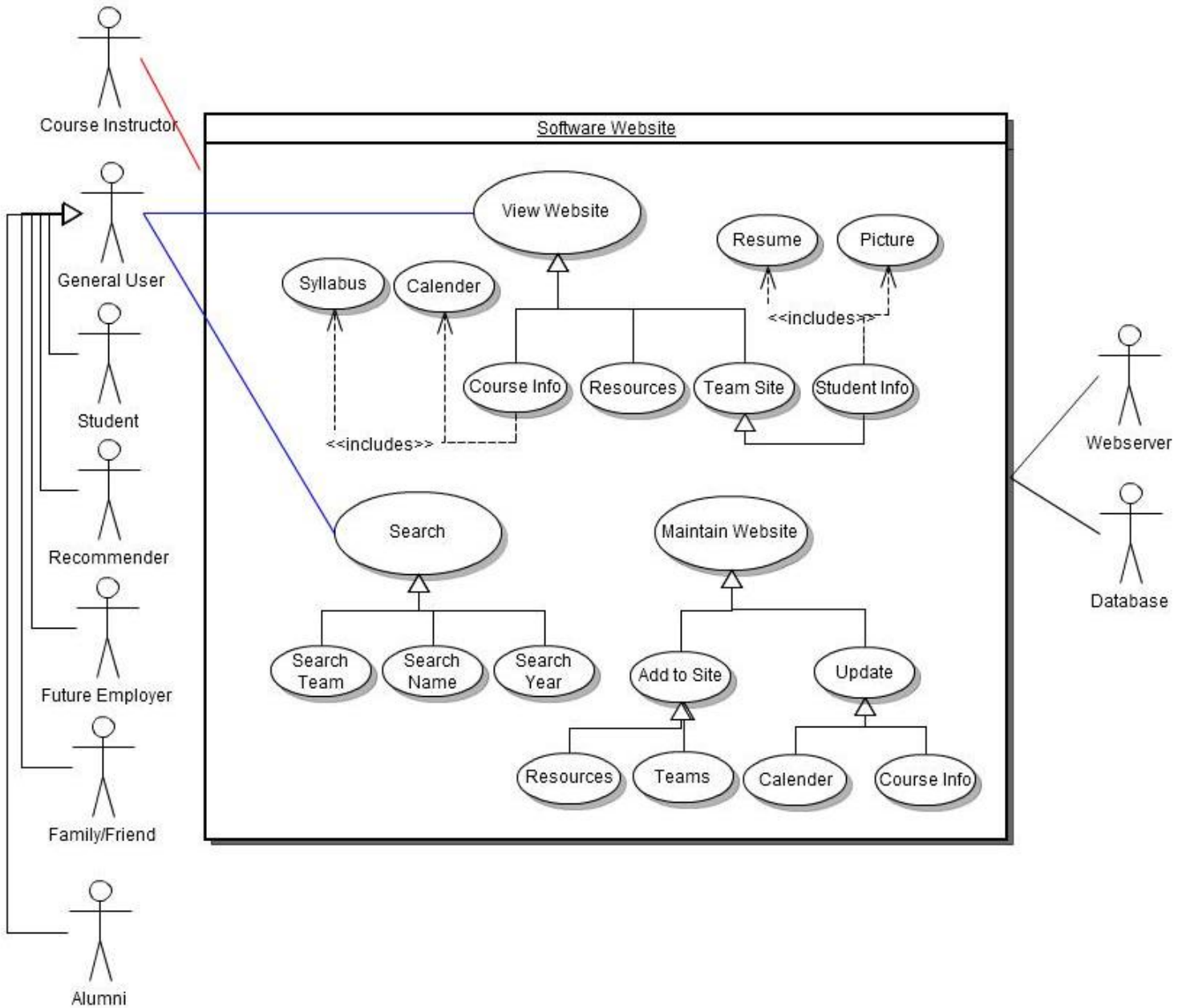


**Participation Line**- Lines that connect actors with uses showing their participation.



**Inherits Arrow**- Shows subuses of uses that may not necessarily be accessed when their parent use is accessed.

### 1.3.2 UML Use Case Diagram



## **1.4 Usage Case Narratives**

### **1.4.1 Course Instructor Usage Case Narrative**

The course instructor, as the administrator of the website and teacher of the Software Engineering course, is the main user of the website. The course instructor will be able to edit every aspect of the website, as well as add to any part of the website. The course instructor will also have the ability to edit the calendar as often as the course instructor would like. Since the course instructor not only runs the website, but also uses it in class as a reference point, the course instructor will be able to utilize the website in a similar manner to the other users, especially recommenders.

### **1.4.2 Software Engineering Student Usage Case Narrative**

The student enrolled in Software Engineering will have the ability to access the calendar, along with the syllabus and resource links for the course. The student enrolled in Software Engineering will be able to go back and forth between semesters without becoming confused and will have the ability to use the consistent menu to go back to the main page. The student enrolled in Software Engineering will be able to use the links provided on the website to go to certain websites of the college. There will be four external links for the Software Engineering student to utilize, which will consist of links to Siena College's main site, the Siena College Computer Science Department site, the Siena College School of Science site, and the Siena College Career Center site. These links will be very important for the Software Engineering student because the four sites are frequently used by all students at Siena College. Also, the student enrolled in Software Engineering will be able to navigate to, as well as use the search function to get to a team page, either the site of the Software Engineering student's team or the sites of previous teams. The student enrolled in Software Engineering will need to see previous teams' sites to use as examples. Lastly, the student enrolled in Software Engineering will be able to use the website to show the team's accomplishments by providing a link to the team website.

### **1.4.3 Recommenders Usage Case Narrative**

Recommenders will use the website mostly for references for Software Engineering students. Recommenders will be able to search for a specific Software Engineering student by the Software Engineering student's name, the name of the Software Engineering student's team, or the Software Engineering student's graduation year. When a recommender searches with the information the recommender provides, the recommender will be taken to a page with general information about the Software Engineering student being searched. The page will contain the resume, picture, class, and other background information of the Software Engineering student. Recommenders will need this information for references because then appropriate recommendations can be given by the recommenders. A URL will take recommenders to the Software Engineering website.

### **1.4.4 Family/Friend Usage Case Narrative**

The Software Engineering website is not just a tool that Software Engineering students use to keep up to date in the Software Engineering class; it is also a way to showcase talents and accomplishments of Software Engineering students. Therefore, it is important that family members and friends can easily find Software Engineering students on the website. Family members and friends will be able to search for a particular Software Engineering student by the Software Engineering student's name. A URL will take family members and friends to the Software Engineering website.



### **1.4.5 Alumnus/Alumna Usage Case Narrative**

As previously stated, the website is the means by which a Software Engineering student can display the Software Engineering student's achievements through the Software Engineering student's own website. Once Software Engineering students have become alumni, it may be so desired to see the achievements when alumni were Software Engineering students. Alumni will be able to do so by using the search function to find themselves. Since alumni may not remember team names or names of teammates, the search function will assist alumni in doing so. This ability to search will be for reminiscent reasons as well as professional reasons. Alumni may just want to revisit the past, or alumni may want to use the website as a way to showcase past accomplishments, either to prospective employers or the average person. Additionally, if one alumnus, the interviewee, is being interviewed by a person, the interviewer, who happens to be a fellow alumnus, the interviewee can go to the Software Engineering site and see what type of Software Engineering experience the interviewer had when the interviewer was enrolled in Software Engineering.

## ***1.5 Data Flow Diagrams***

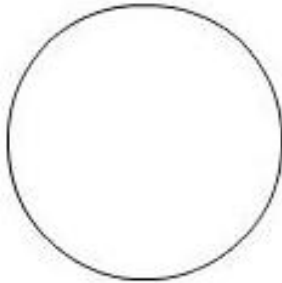
Data Flow Diagrams are created to show the movement of data throughout the system. They are used as a visual aid to help the reader to better understand how the system works. The context diagram is a general overview of the entire system. The level 0 diagram shows in more detail how each entity interacts with the system through a process. The level 1 diagrams show an in-depth look at each process within the system. In all diagrams arrows are drawn between entities, databases and processes to show the movement of data.

## 1.5.1

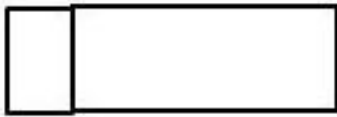
## Data Flow Diagram Legend



Entity- Represents users of the system. The source and recipient of data.



Process- Represents a function where data is manipulated or transformed.



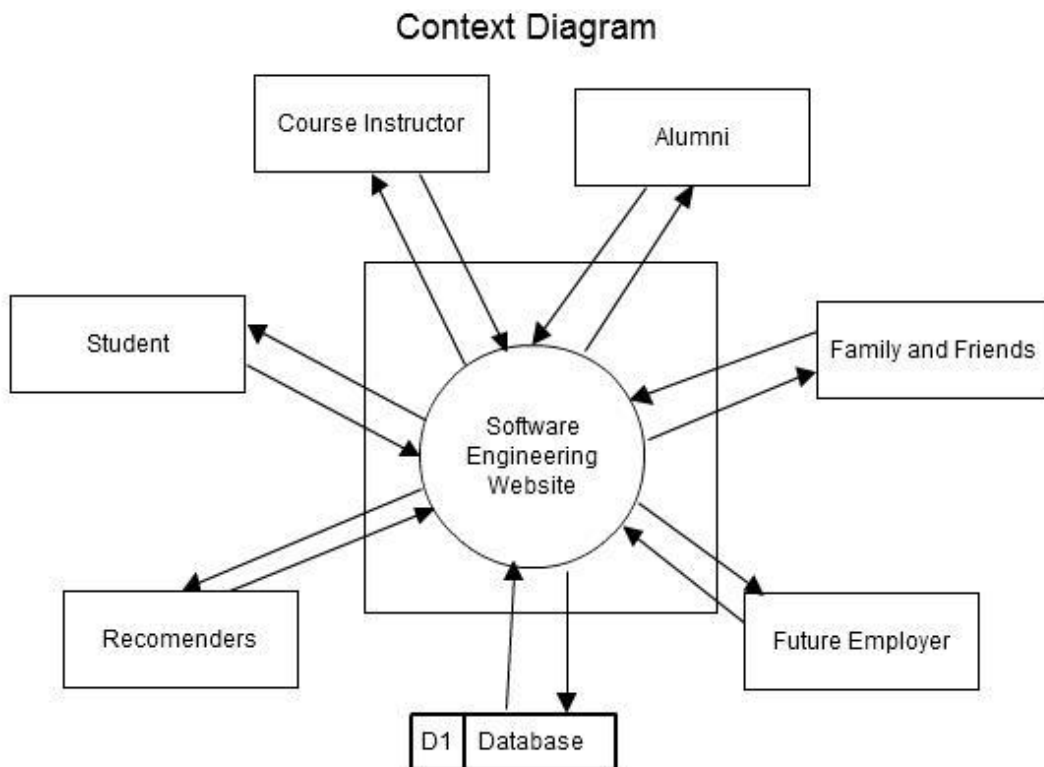
Data Store- Where data is sent and held until it needs to be accessed again.



Data Flow- Represents the movement of data.

## 1.5.2 Context Diagram

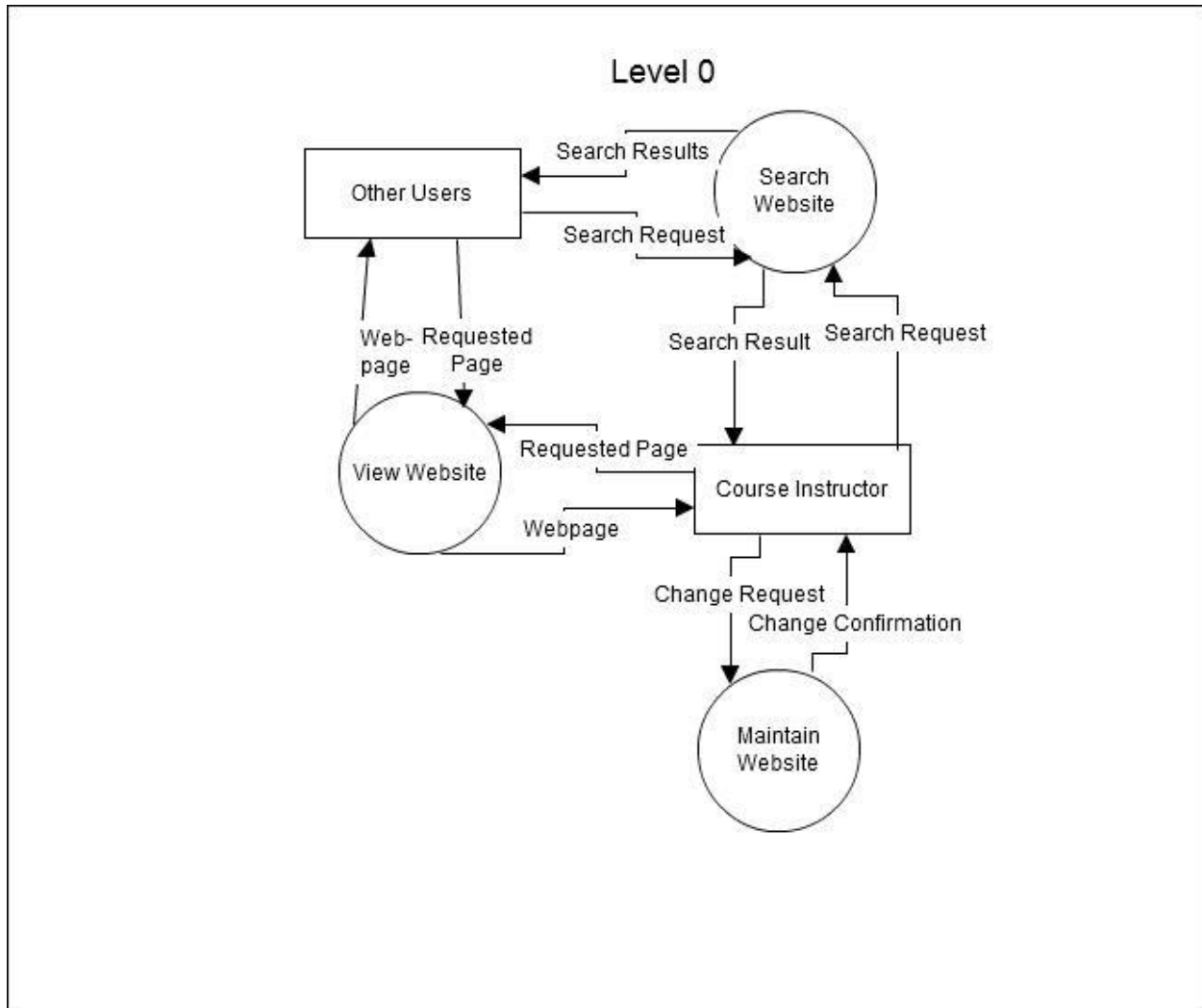
This is the highest level and most general representation of data flow in our system. It shows interactions between users, databases, and software



engineering website.

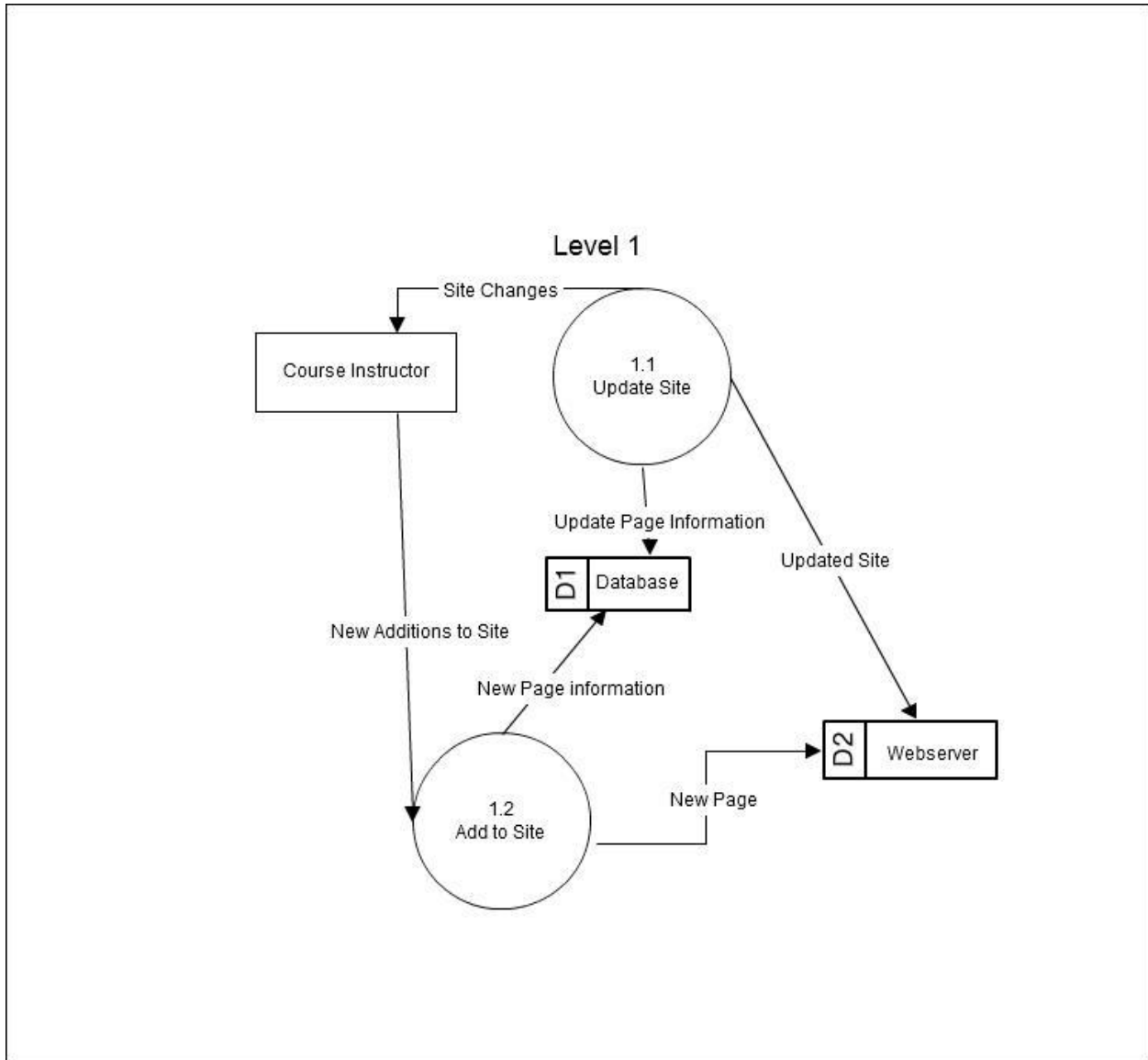
### 1.5.3 Level 0 Diagram

This diagram shows the major processes of the system.



### 1.5.4 Level 1 Diagram

This diagram shows the data flow of one of the main uses, maintaining the website.



## **1.6 Functional Requirements Inventory**

Functional Requirements Inventory is a complete, detailed list of all system functions agreed upon by the client and the development team.

This site will run on all major browsers including, Internet Explorer 7 and 8, Mozilla Firefox, Safari and Google Chrome.

The following is a list of the functional requirements for each user. The requirements are grouped by user case and there are 6 different users for R.O.S.E.S.

### **1.6.1 The General User**

The General User includes Students, Faculty, Family and Friends, Future Employers, and Alumni.

#### View Website

- Will be able to view the main website
- Will be able to view the calendar
- Will be able to view the syllabus
- Will be able to view the different courses information
- Will be able to view the current and previous teams websites
- Will be able to view previous students information
- Will be able to view previous students picture
- Will be able to view previous students resume

#### Search Function

- Will be able to search by team names
- Will be able to search by team member names

- Will be able to search by year of the team of member

## 1.6.2 **The Course Instructor**

### View Website

- Will be able to view the main website
- Will be able to view the calendar
- Will be able to view the syllabus
- Will be able to view the different courses information
- Will be able to view the current and previous teams websites
- Will be able to view previous students information
- Will be able to view previous students picture
- Will be able to view previous students resume

### Search Function

- Will be able to search by team names
- Will be able to search by team member names
- Will be able to search by year of the team of member

### Maintain Website

- Will be able to update the calendar for each course
- Will be able to update the course information for each course
- Will be able to add future teams website and information
- Will be able to add resources to the site for each course

## 1.7 Non-Functional Requirements Inventory

Non-functional requirements are requirements that have a specific criteria used to critique the operation of a system. Some of the criteria typically used is user interface, aesthetics, and more. These requirements explain what R.O.S.E.S is to *be* instead of what it *does*. Below is the list of our non-functional requirements inventory.

- The system must be aesthetically pleasing
- The system must be easily navigable
- The system must be easily maintainable
- The system must be easily modifiable
- The system must be stable

## 1.8 Exception Handling

Currently it is not clear of what the exact types of exceptions that *Code Shark Solutions* will need to deal with for R.O.S.E.S. Once *Code Shark Solutions* gathers more information and moves on to the next phase, the exceptions will be much more precise. One type of exception that might need to be dealt with is how to handle a server crash. Another exception would be if a user tries to hack the system, using either SQL, JavaScript injection, or other ways, in attempts to gain access to the database. This would be handled with good security practices that ensure information sent by the user through the system is both valid and safe. Handling incorrect input entered by a user will also be an exception. To handle this we will allow only valid input and notify user of their error. In order to prevent a user from gaining access to private folders by entering a relative URL, measures will be put in place to redirect the user away from the data.



## **1.9 Early Subsets and Implementation Priorities**

The important components of R.O.S.E.S. are:

- The ability to run on all major browsers
- The ability to have a consistent navigation
- The ability to have a friendly user interface
- The ability to have a secure login for course instructor
- The ability for the site to be edited by the course instructor
- The ability to have a search function
- The ability to have all Software Engineering team websites working properly

## **1.10 Foreseeable Modifications and Enhancements**

The newest addition to the Siena College Software Engineering Website will be its search functionality. To begin with it will be a very basic search function that will take in exact words or phrases to locate users, teams, or projects. In the near future it is very well possible that it will be expanded to include dealing with non-exact search hits and also providing possibilities based on what the user input.

## **1.11 Testing Requirements and Acceptance Criteria**

Currently *Code Shark Solutions* does not have a finalized set of Testing Requirements and Acceptance Criteria. As the project progress through the Preliminary Design and Detailed Design phases a more formal document will be created which will include both the Functional and Non-Functional Requirements which are listed in this document in phases 1.7 and 1.8, respectively. Upon completion of the project, *Code Shark Solutions* will go through the document to ensure the system meets all declared requirements.

## **1.12 Guidelines**

The following guidelines have been declared for the R.O.S.E.S. project:

- There must be fixed anchors implemented with scripts that determine the current date for the calendar implementation.

As the client declares more guidelines for the implementation of the project, they will be added to the formal documentation.

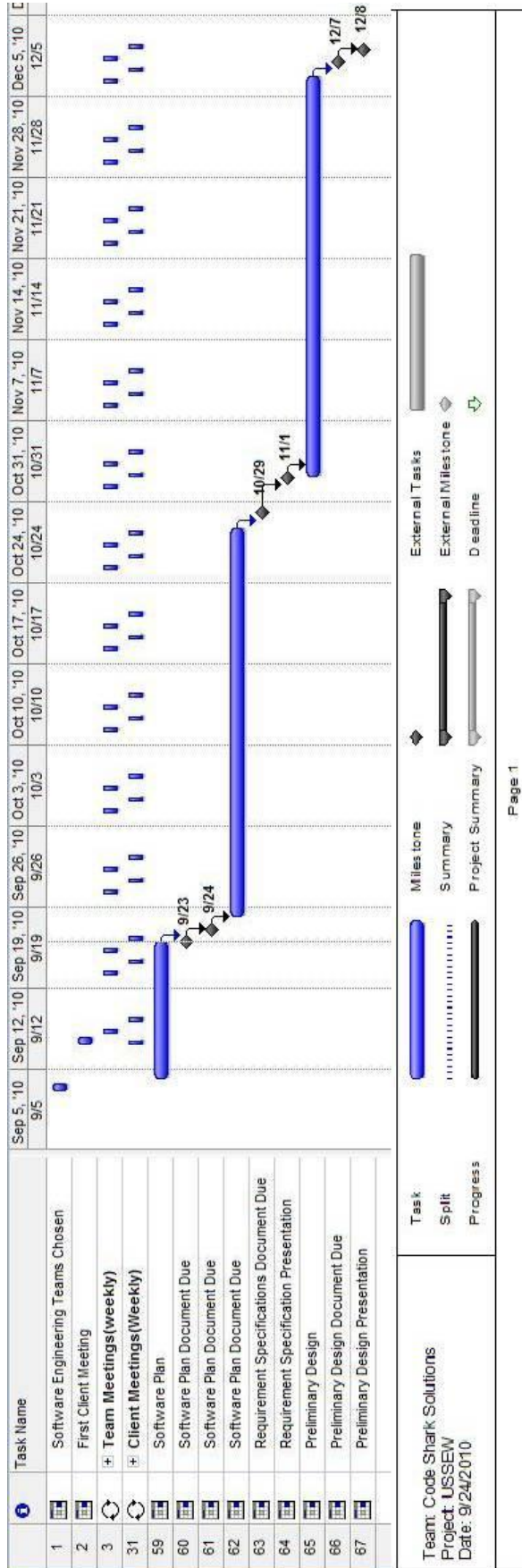
## **Appendix A: Sources of Information**

All information contained in this document was obtained through meetings with the client, Dr. Lederman, as well as class lectures with Dr. Lederman and documents of past Software Engineering teams.

## **Appendix B: Timeline**

The tentative development schedule is outlined by the Timeline (Gantt Chart) on the following page. Changes are made as needed by agreement between *Code Shark Solutions* and Dr. Lederman.

# Appendix of Terms



# C: Glossary

**Actor:** An entity in UML Use Case Diagrams and UML Activity Diagrams. It represents the human and non-human external entities (outside the system boundary) that interact with the system.

**CSS (Cascading Style Sheets):** CSS is used alongside HTML to add aesthetic value to a website.

**Data Flow Diagram (DFD):** a pictorial representation of the flow of data in a Software System which is comprised of varying levels of detail.

**Data Flows:** A component of a Data Flow Diagram (DFD) that represents the movement of data from an External Entity to a Process or Data Store, and vice versa.

**Data Stores:** A component of a Data Flow Diagram(DFD) that represents any location in which information or data is stored.

**External Entities:** A component of a Data Flow Diagram that represents any human or non-human user of a Software System.

**Functional Requirements Inventory:** Define what the system will be able to do and what is testable about the system.

**Hardware:** The physical parts of a computer, such as the hard drive and the CPU.

**HTML (HyperText Markup Language):** HTML is a scripting language used to design the structural layout of a website.

**JavaScript:** JavaScript is a object oriented scripting language that operates on the user's computer rather than on the hosting server.

**MySQL:** MySQL is an open source relational database management software based on the SQL vocabulary which can be employed in combination with most server-side languages and can be used to access information in databases.



**Non-Functional Requirements:** Specifies how a product is supposed to *be*, compared to functional requirements that describe what the product *does*. Such examples are the user interface, aesthetics, accessibility, maintainability, security, etc. Non-functional are difficult, if not impossible to quantifiably test.

**PHP (PHP Hypertext Preprocessor):** PHP is a “server side” programming language that is used to create in depth functionality on websites. PHP can also communicate with servers and databases.

**Processes:** A component of a Data Flow Diagram that represents any scenario or action within a Software System.

**Relationship:** A component of a UML Use Case Diagram which represents the interactions between the Actor and the System.

**Screen Resolution:** The screen resolution is the number of pixels displayed on the screen, it is usually given in the form Width x Height where width and height are the number of pixels across and down the screen.

**Software:** The programs installed on the computer, such as Microsoft Office and Adobe Fireworks.

**System:** A component of UML Use Case Diagram which represents the Software System.

**UML (Unified Modeling Language) Use Case Diagram:** A general pictorial explanation of the basic processes of a Software System used by Software Development Teams.

**Use Case:** A component of a UML Use Case Diagram which represents any process located within the System that is performable by an Actor.

**User Case Narrative:** an explanation of the functions and abilities users have for a specific Software System.

**Waterfall Model:** A basic software development strategy that clearly labels each phase of the software engineering process. The strategy follows consecutively the following steps: Requirements Specification, Design, Construction, Verification, and Maintenance.