

Detailed Design

R.O.S.E.S

(Redesign Of the Software Engineering Site)

Requested by:

Dr. Timothy C. Lederman
Professor
Siena College
Department of Computer Science

Prepared by:

Code Shark Solutions

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

Table of Contents	5
1.1 Product Overview	5
1.2 Development, Operation and Maintenance Environment	6
1.3 Usage Case Narratives	7
1.3.1 Course Instructor Usage Case Narrative	7

1.3.2 Software Engineering Student Usage Case Narrative.....	
1.3.3 Recommenders Usage Case Narrative.....	7
1.3.4 Family/Friend Usage Case Narrative.....	7
1.3.5 Alumnus/Alumna Usage Case Narrative.....	8
1.4 Functional Requirements Inventory.....	9
1.5 Non-Functional Requirements Inventory.....	10
2 Detailed Design.....	11
2.1 UML Use Case Diagram.....	11
2.1.1 UML Use Case Legend.....	11
2.1.2 Diagram.....	12
2.2 UML Deployment Diagram.....	13
2.2.1 Diagram.....	13
2.3 UML Activity Diagram.....	14
2.3.1 Activity Diagram Legend.....	14
2.3.2 Diagram.....	15
2.4 Website Map.....	16
2.4.1 Legend.....	16
2.4.2 Map.....	16
2.5 Data Flow Diagrams.....	17
2.5.1 Data Flow Diagram Legend.....	17
2.5.2 Context Diagram.....	18
2.5.3 Level 0 Diagram.....	19
2.5.4 Level 1 Diagrams.....	20
2.5.5 Level 2 Diagrams.....	21
3 Data Stores.....	25
3.1 Logical Data Dictionary.....	25
3.2.1 Microsoft Access 2007 Relationships Diagram.....	30
3.2.2 Entity-Relationship (ER) Diagram.....	31
3.2.3 Database Relational Schema.....	31
4 Prototype Screens.....	33
4.1 Home Page.....	33
4.2 Main Menu.....	34
4.3 Sub Menu.....	35
4.4 Footer Links.....	36

4.5 Software Engineering I – Course Description.....	
4.6 Software Engineering I – Calendar.....	38
4.7 Software Engineering I – Grading.....	39
4.8 Software Engineering I – Syllabus.....	40
4.9 Software Engineering II – Course Description.....	40
4.10 Software Engineering II – Calendar.....	42
4.11 Software Engineering II – Grading.....	43
4.12 Software Engineering II – Syllabus.....	44
4.13 Software Engineering II – Speakers.....	45
4.14 Software Engineering II – Killer Robot.....	46
4.15 Search page.....	47
4.15 Search page cont.....	49
4.16 Results – Multiple Results.....	50
4.17 Results – Student Information.....	51
4.18 Results – Team Information.....	52
4.19 Results – Course Year Information.....	53
4.20 Results – Client Information.....	54
5 Test Plan.....	55
5.1 Testing Plan Identifier.....	55
5.2 Introduction.....	55
5.3 Test Items and Functional Requirements Inventory.....	56
5.4 Non-Functional Requirements.....	57
5.5 Exception Handling to Test.....	57
5.6 Testing Approach.....	57
5.7 Acceptance Test – Acceptance Criteria.....	59
5.8 Unit Tests.....	59
5.8.1 Directory.....	1
5.8.2 Unit 1.1.....	2
5.8.3 Unit 1.2.....	4
5.8.4 Unit 1.3.....	6
5.8.5 Unit 2.1.....	8
5.8.6 Unit 2.2.....	9
5.8.7 Unit 2.3.....	12
5.9 Integration Test.....	1

6 Appendices.....

6.1 Appendix A: Glossary.....2

6.2 Appendix B: Timeline.....4

6.2.1 Fall Semester.....4

6.2.2 Spring Semester.....5

1 Introduction

1.1 Product Overview

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, Operation and Maintenance 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
Virtual Server:	YES

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 Usage Case Narratives

1.3.1 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.

1.3.2 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.

1.3.3 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. This will benefit students by reminding the recommender who the student is and allows them to write a more personal recommendation.

1.3.4 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.

1.3.5 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.

1.4 Functional Requirements Inventory

View Website

- Will be able to view the Software Engineering course Homepage.
- Will be able to click links to this year's team pages.
- Will be able to click link to Siena College website.
- Will be able to click link to Siena Computer Science website.
- Will be able to click link to Siena School of Science website.
- Will be able to click link to Siena Career Center website.
- Will be able to view the Software Engineering I page.
- Will be able to view Software Engineering I Calendar page.
- Calendar will open to the current date.
- Will be able to select a date to display to.
- Will be able to view Software Engineering I Grading page.
- Will be able to view the Software Engineering I Syllabus page.
- Will be able to view the Software Engineering II page.
- Will be able to view the Software Engineering II Calendar page.
- Will be able to view the Software Engineering II Grading page.
- Will be able to view the Software Engineering II Syllabus page.
- Will be able to view the Software Engineering II Speakers page.
- Will be able to view the Software Engineering II Killer Robot page.
- Will be able to view the Search page.
- Will have a consistent menu system that will allow users to navigate to all pages.

Searching

- Will be able to search by student name.
- Will be able to search by team name.
- Will be able to search by course year.
- Will be able to select Student Name, Team Name, or Course Year from the displayed results.
- Will be able view page for a student.
- Will be able to view page for a software team.
- Will be able to view page for a team year.
- Will resize photos of teams and individuals to stay proportional given a static width.
- Will be able to print the result pages for Students and Teams
- Will be able to click link view all teams.
- Will be able to click link to view all course years.

1.5 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 are 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.

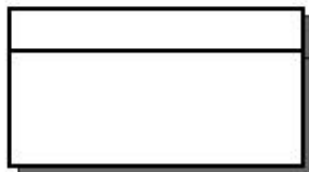
- The system must be aesthetically pleasing
- The system must be easily navigable
- The system must be easily maintainable
- The system must be easily modifiable
- The system must be editable

2 Detailed Design

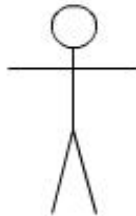
2.1 UML Use Case Diagram

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.

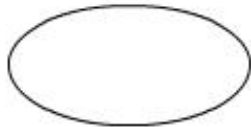
2.1.1 UML Use Case Legend



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.

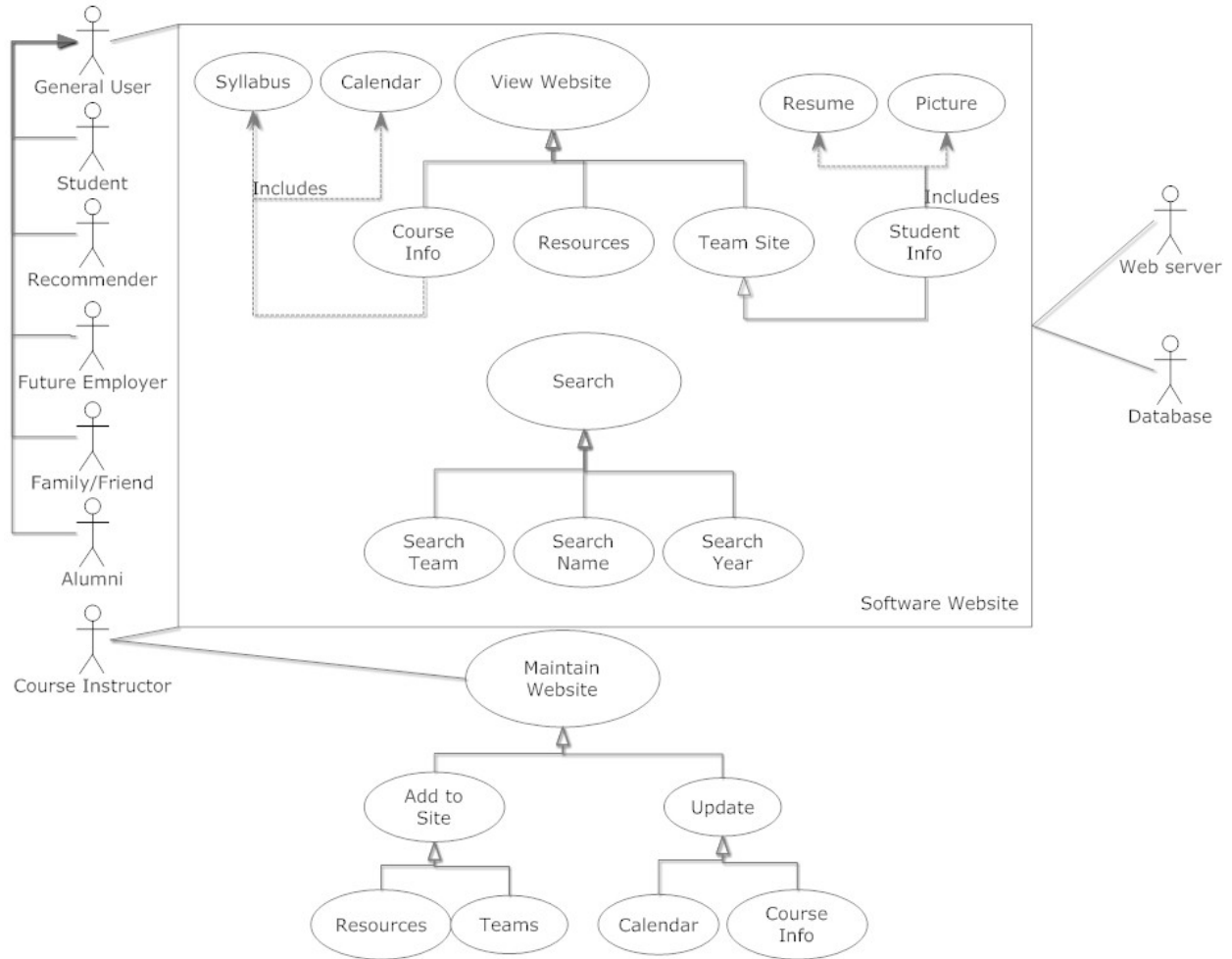


Participation Line- Lines that connect actors with uses showing their participation.



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

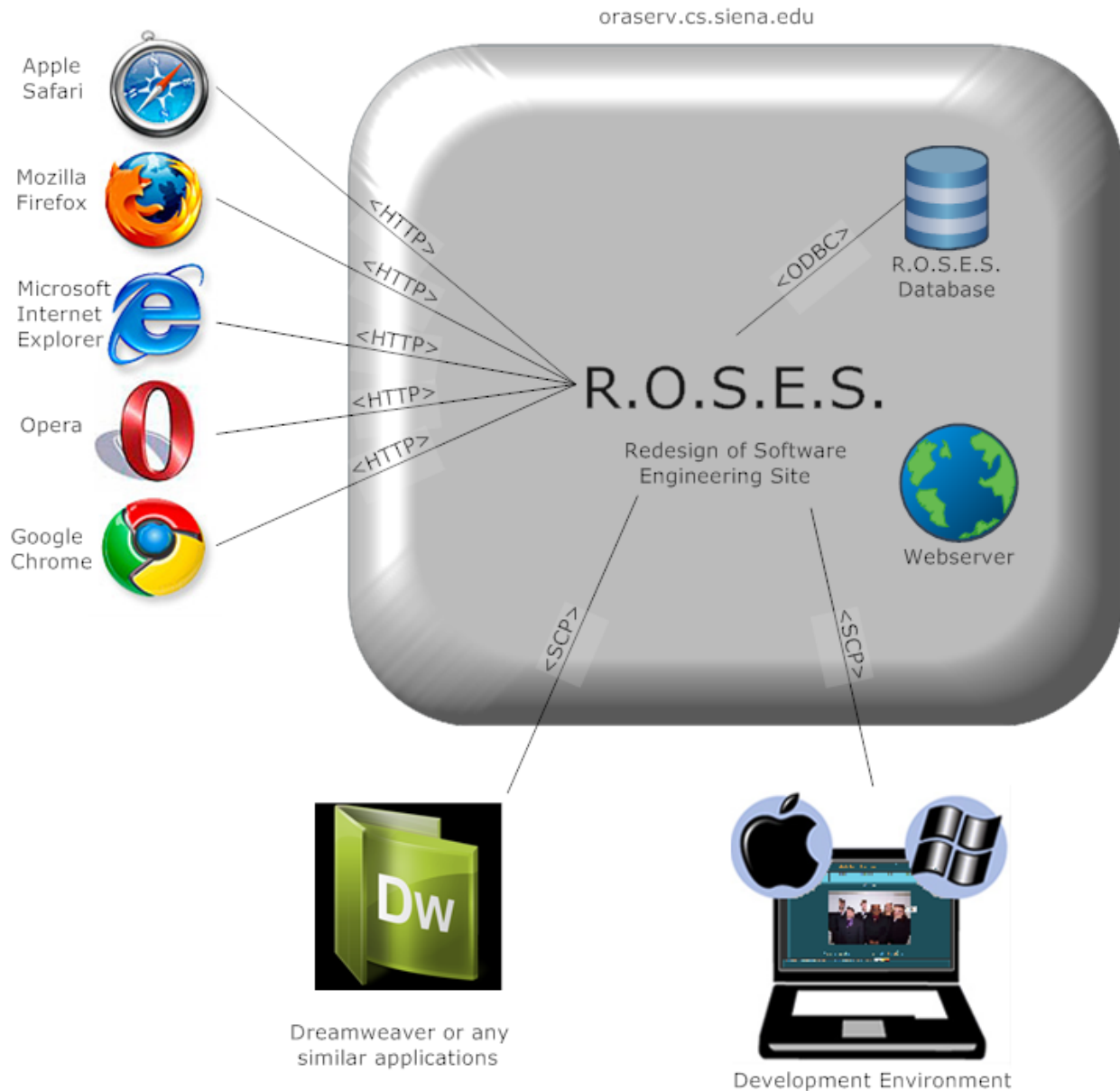
2.1.2 Diagram



2.2 UML Deployment Diagram

Deployment diagrams are used to show devices and execution environments for a system. The deployment diagram below shows the different browsers, development environment, and databases interacting with R.O.S.E.S. The diagram also shows the various protocols through which the interactions occur.

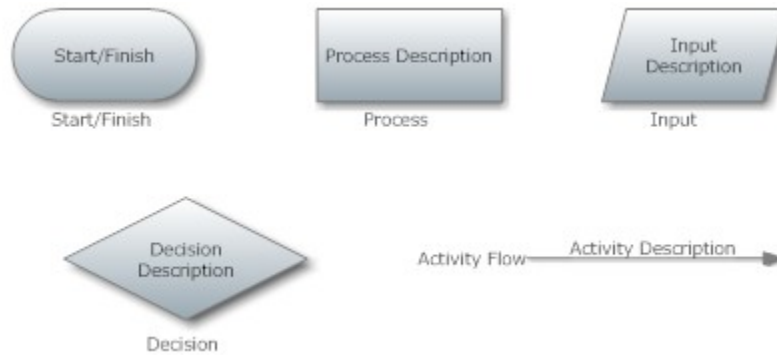
2.2.1 Diagram



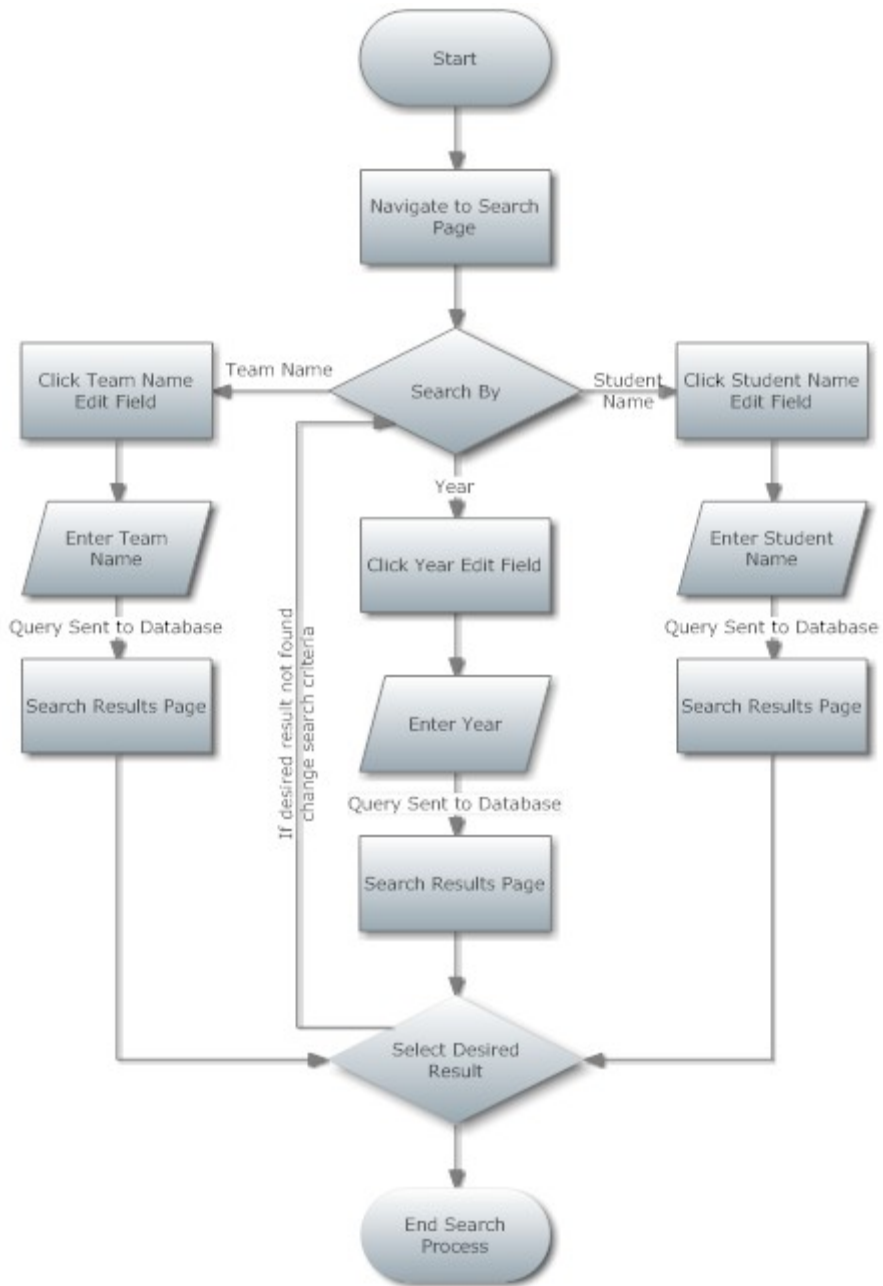
2.3 UML Activity Diagram

UML Activity Diagrams show the processes ‘Actors’ go through to interact with the system when performing a specific task. The diagrams show all possible decisions and actions that are available throughout the process.

2.3.1 Activity Diagram Legend



2.3.2 Diagram



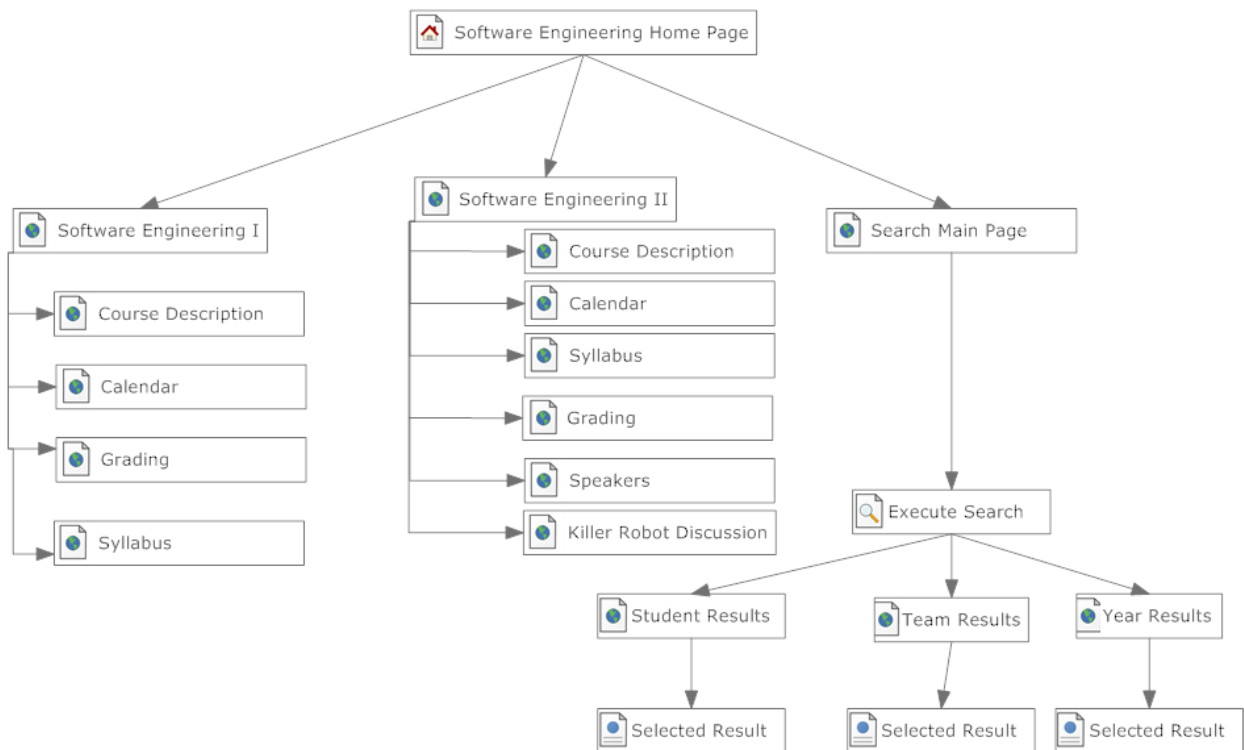
2.4 Website Map

The Website Map is a diagram of the website, showing its layout and how it is layered and linked.

2.4.1 Legend



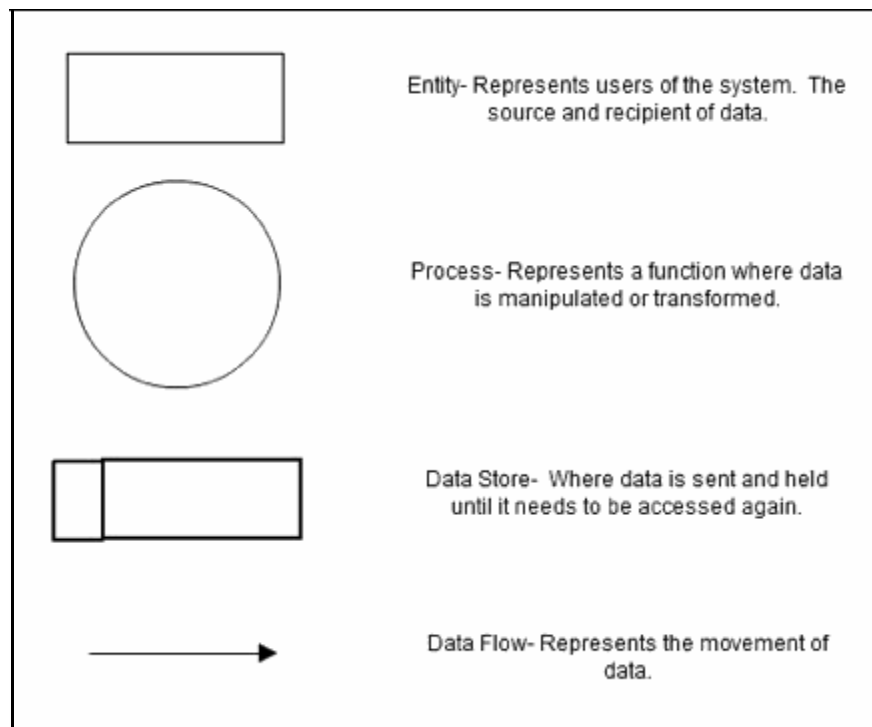
2.4.2 Map



2.5 Data Flow Diagrams

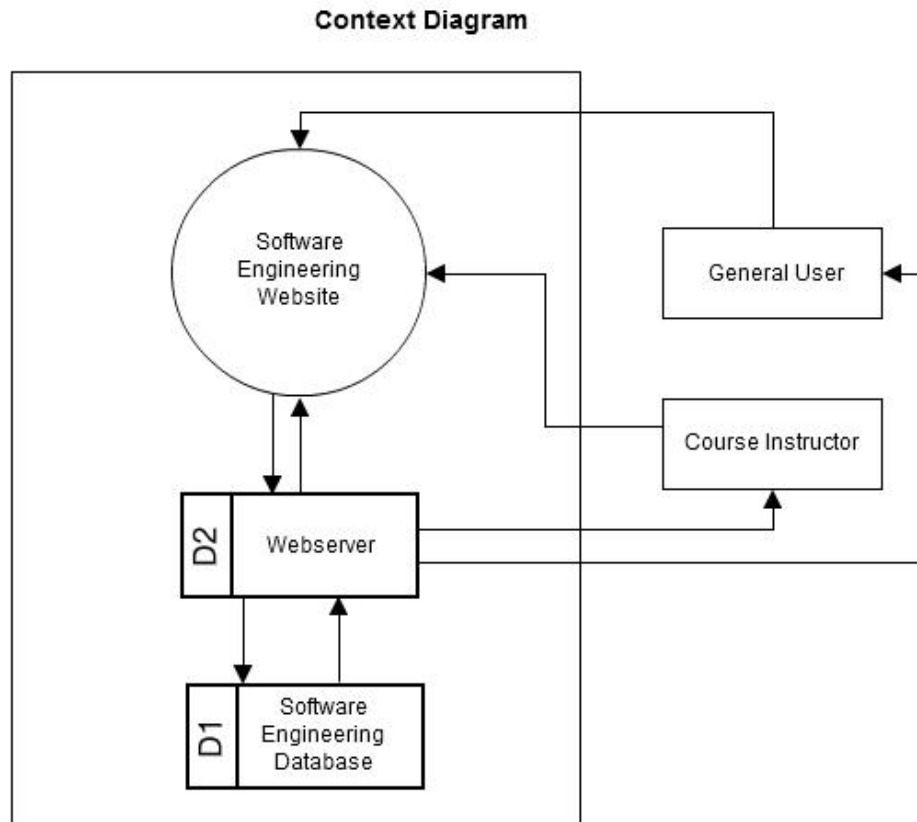
Data Flow Diagrams are created to show the movement of data throughout the system. They are used as a visual aid to help the reader to better understand how the system works. The context diagram is a general overview of the entire system. The level 0 diagram shows in more detail how each entity interacts 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.

2.5.1 Data Flow Diagram Legend



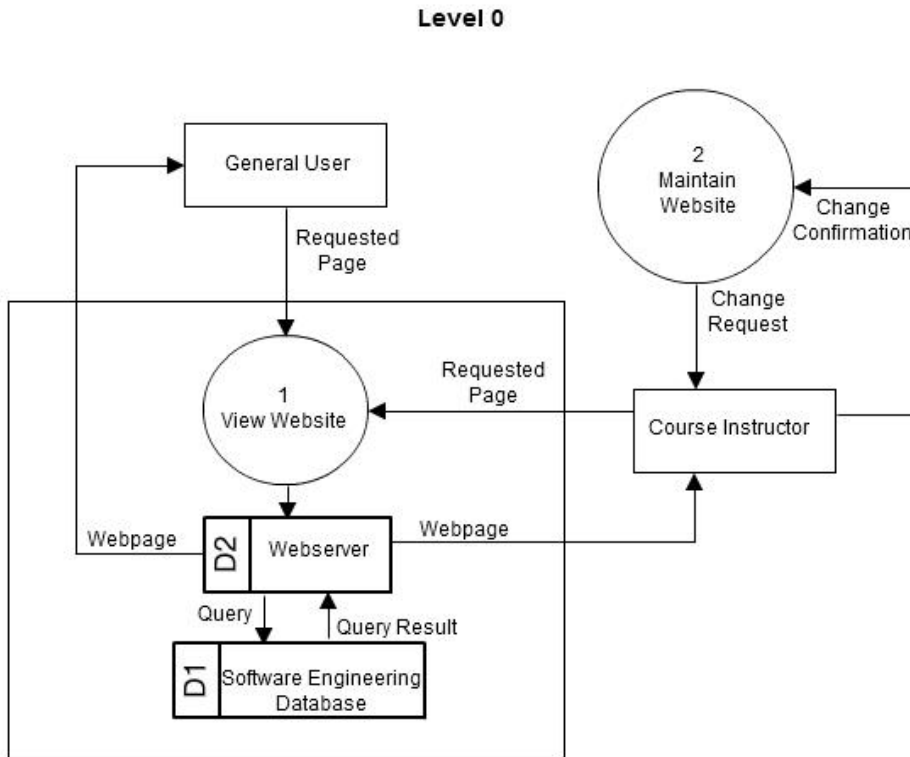
2.5.2 Context Diagram

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.



2.5.3 Level 0 Diagram

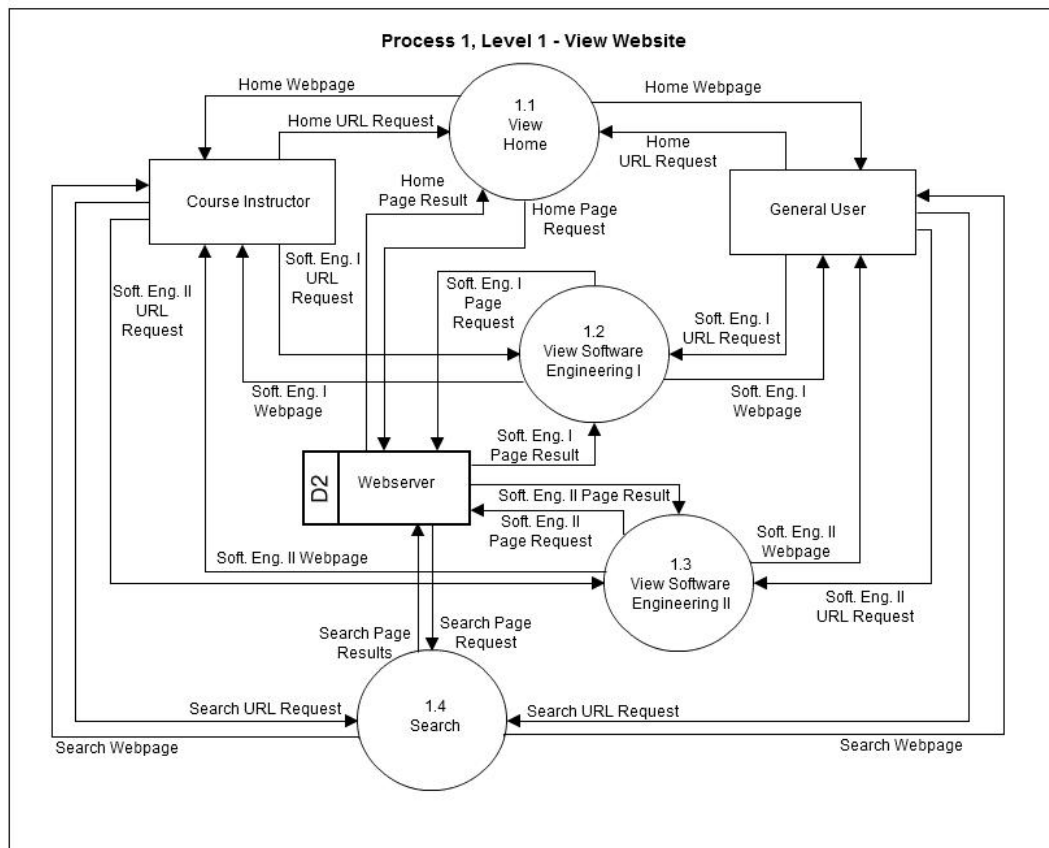
This diagram shows the major processes of the system.



2.5.4 Level 1 Diagrams

This Diagram shows the main use of the website.

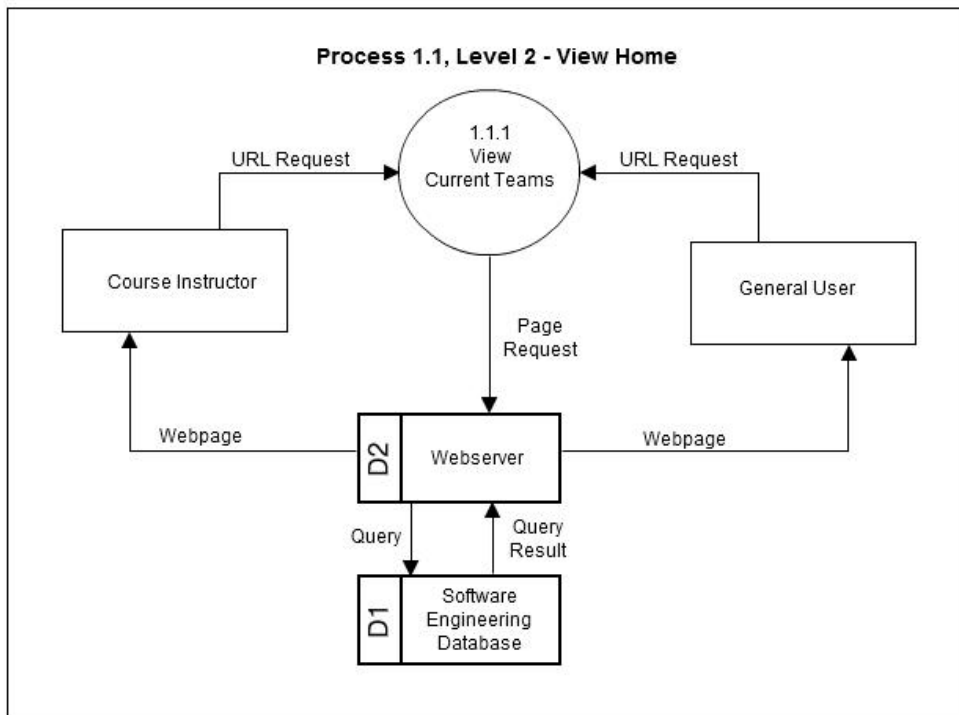
2.5.4.1 View Website



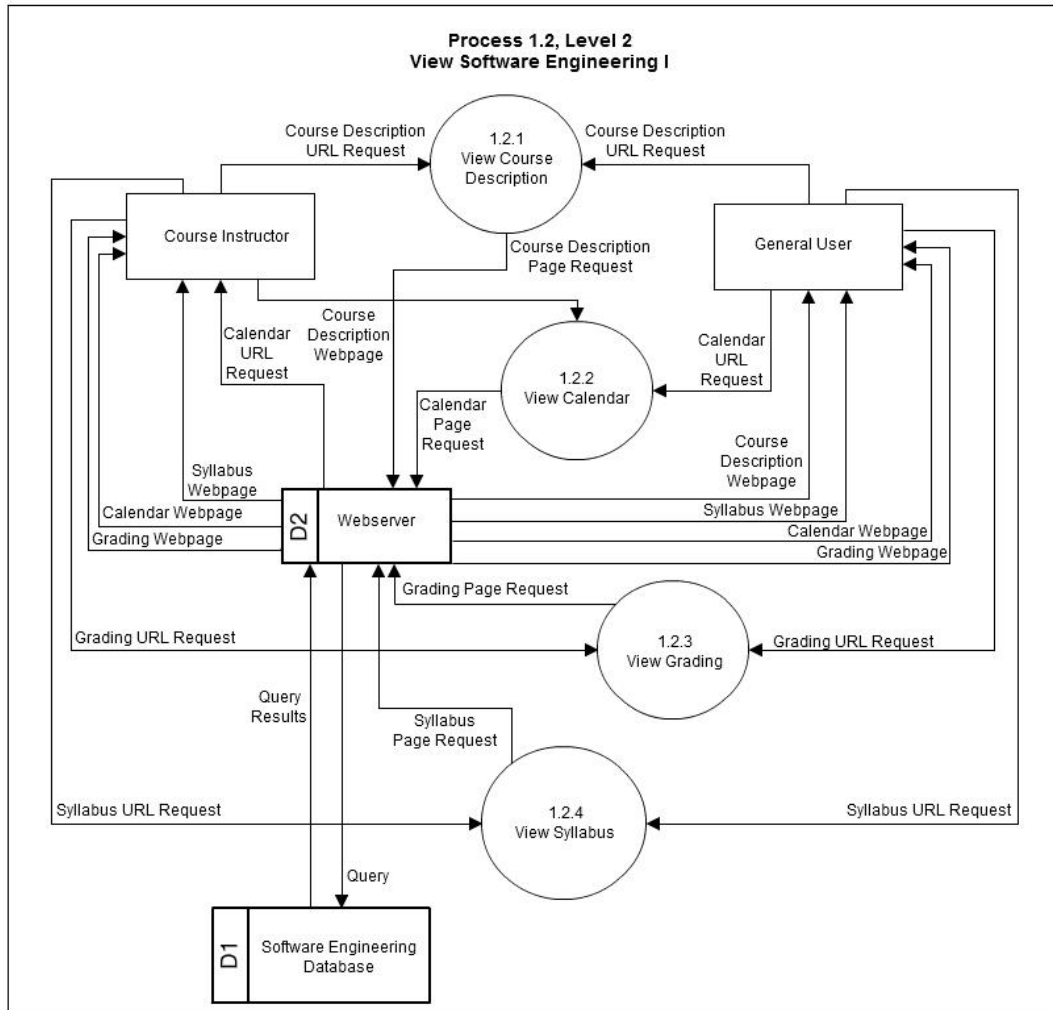
2.5.5 Level 2 Diagrams

These Diagrams break down the processes of the level one diagram.

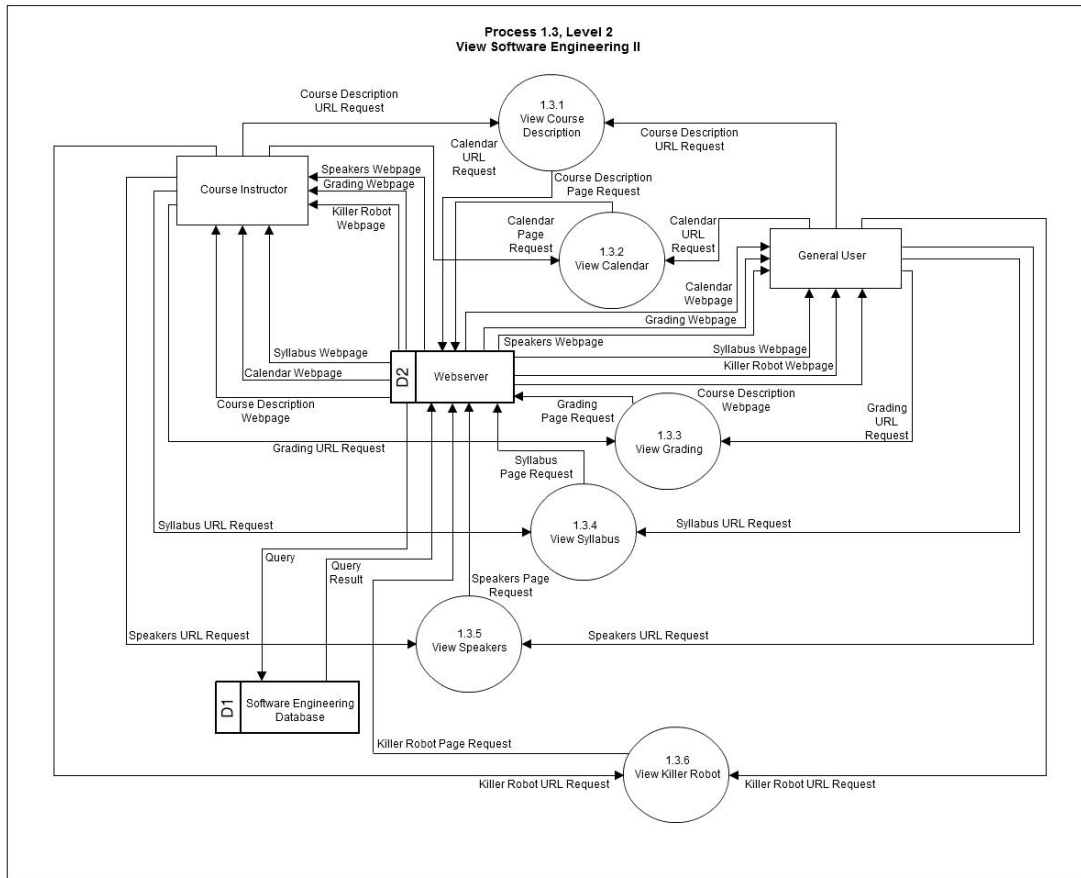
2.5.5.1 View Home



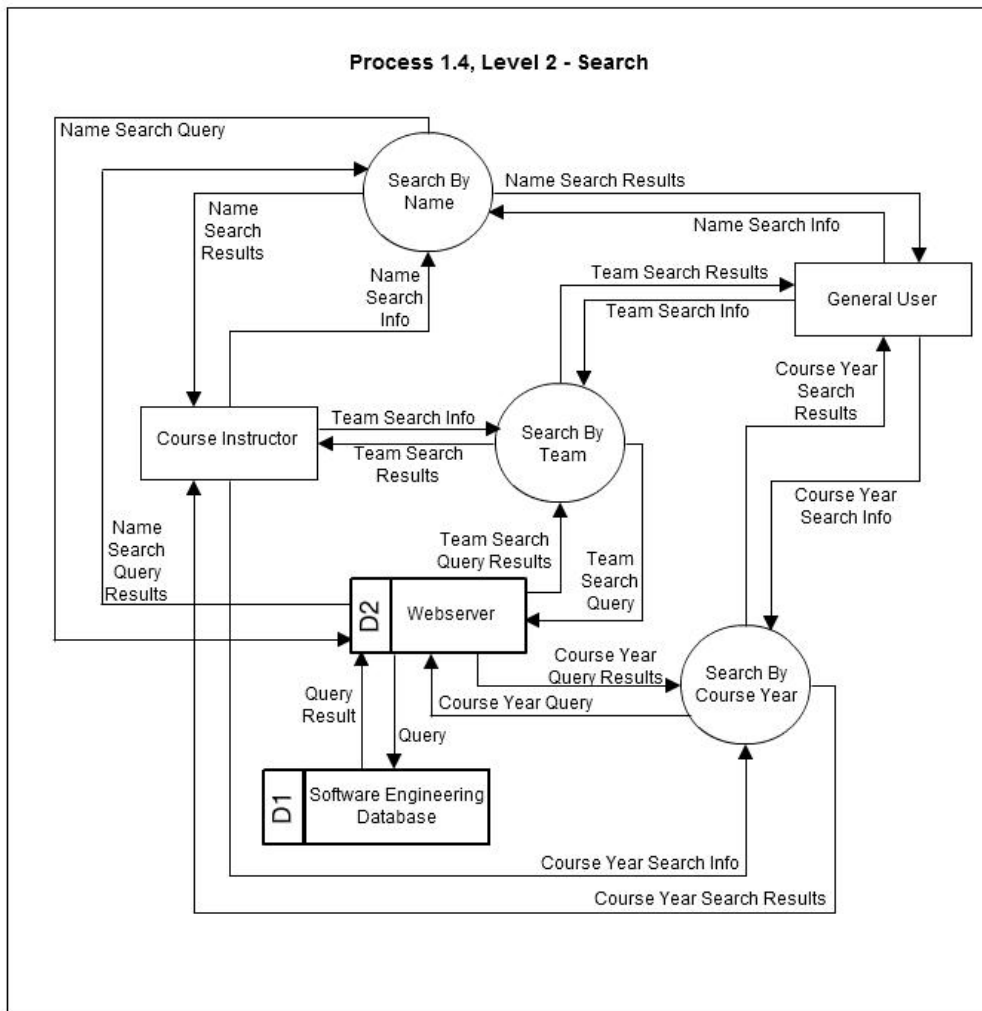
2.5.5.2 View Software Engineering I



2.5.5.3 View Software Engineering II



2.5.5.4 Search



3 Data Stores

3.1 Logical Data Dictionary

A Logical Data Dictionary is a list of all data entities and variables within a system. Any form of data that will be input, manipulated, or stored will be found in the data dictionary. The data dictionary contains the name of the data entity, what the entity stores, how it is used, the data type of that entity, the size of the entity, and what table it can be found in. A key is listed to better understand the entries in the data dictionary.

KEY:

Table Name- The name of the table the data entity will be stored

Data Name- The name of the data entity

Comment- What the entity is used for

Applicable to- What page/functions the Data entity will be used in

Data Type- What type the data entity will be stored as(String, Int)

Data Size- How big the data entity is allowed to be

Good Example of Input- One example of how this data entity can be stored

Notes- Any additional information that is useful in understanding the data entity

Please see next page for the Data Dictionary

Table Name	Data Name	Applicable to	Data Type	Data Size	Good Example of Input	Notes
Team	teamName	Multi-Results, Search, Student/Team/Cours e-Year result profiles	String	5-50 Characters	Code Shark Solutions	
Team	acronym	Multi-Results, Search, Student/Team/Cours e-Year result profiles	String	2-15 Characters	C.S.S.	Can be left NULL
Team	siteLink	Multi-Results, Search, Student/Team/Cours e-Year result profiles	String	7-50 Characters	http://oraserv.cs.siena.edu/~perm_css/	Must have oraserv prefix: http://oraserv.cs.siena.edu/~perm_
Team	clientFirst	Multi-Results, Search, Student/Team/Cours e-Year result profiles	String	5-50 Characters	Timoth	
Team	clientLast	Multi-Results, Search, Student/Team/Cours e-Year result profiles	String	5-50 Characters	Lederman	
Team	fallYear	Multi-Results, Search, Student/Team/Cours e-Year result profiles	Int		2010	Year of Software Engineering I
Team	springYear	Multi-Results, Search, Student/Team/Cours e-Year result profiles	Int		2011	Year of Software Engineering II
Team	projectName	Multi-Results, Search	String		Redesign of the Software Engineering Site	
Team	projectAcronym	Multi-Results, Search	String	2-10 Characters	R.O.S.E.S.	Can be left NULL
Team	demo	Team Profile	String	1-100 Characters		Will be a string that will be extracted for use as a link on the webpage
Team	song	Team Profile	String	1-100 Characters		Will be a string that will be extracted for use as a link on the webpage

Team	teamPic	Team Profile	String	1-100 Characters		Will be a string that will be extracted for use as a link on the webpage
Student	firstName	Multi-Results, Search, Student/Team/Course-Year result profiles	String	5-50 Characters	Joseph	
Student	lastName	Multi-Results, Search, Student/Team/Course-Year result profiles	String	5-50 Characters	D'Avanzo	
Student	picture	Student search results, Student Profile	String	1-100 Characters		Will be a string that will be extracted for use as the user's image on the site
Student	resume	Student search results, Student Profile	String	1-100 Characters		Will be a URL linking to the student's resume
Student	teamLeaderFall	Student Profile	Boolean	1 Character	T	
Student	teamLeaderSpring	Student Profile	Boolean	1 Character	T	
Student	teamName	Student/Team search results, Student Profile	String	5-50 Characters	Code Shark Solutions	Will be a string that will be extracted for use as a file path to the user's resume
Speaker	firstName	Speaker Profile	String	5-50 Characters	Joseph	
Speaker	lastName	Speaker Profile	String	5-50 Characters	D'Avanzo	
Speaker	date	Speaker Profile	String	10-40 Characters	July 29th, 2010	Acceptable format for Date: Month Day, Year
Speaker	subject	Speaker Profile	String	1-100 Characters	PHP For Fun & Profit	Cannot be left NULL
Speaker	degree	Speaker Profile	String	10-50 Characters	B.S. Computer Science	Highest degree attained
Speaker	title1	Speaker Profile	String	10-200 Characters	Vice President Information Technology	Work related titles will always go first unless one isn't available and then education titles replace otherwise

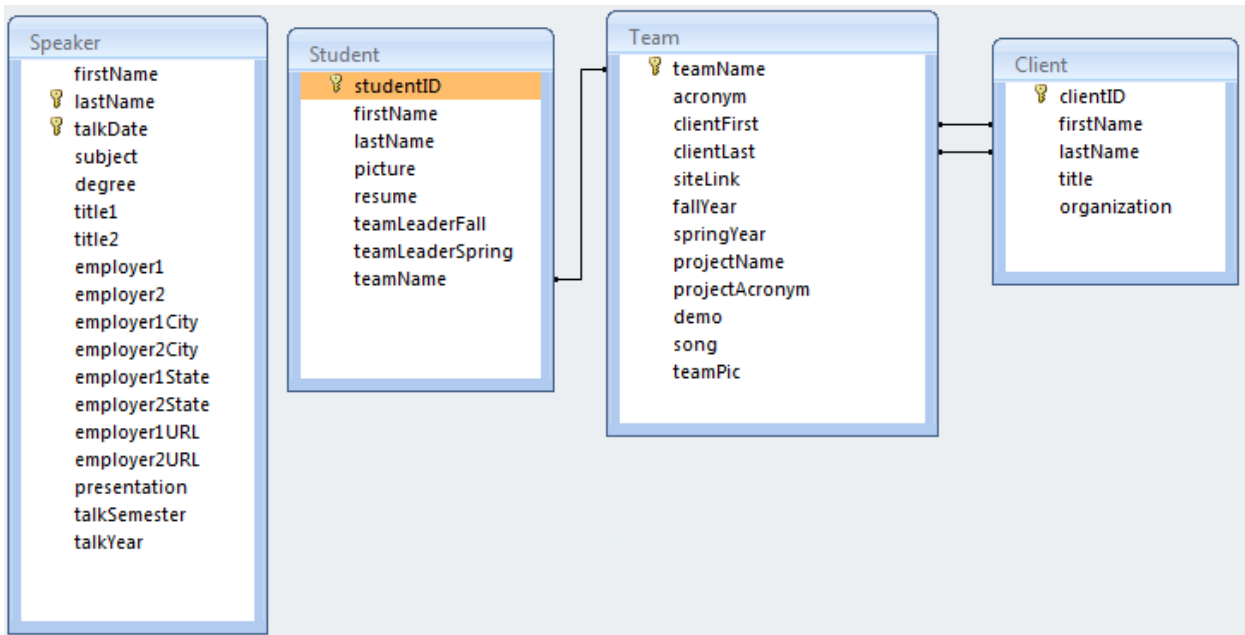
Speaker	title2	Speaker Profile	String	10-200 Characters	PhD student & Teacher's Assistant (Introduction to Java Programming)	Will be used in the case that speaker's have more than one titles. Two most important are listed.
Speaker	employer1	Speaker Profile	String	10-200 Characters		Will either be a string that is extracted as a link or an actual employer name
Speaker	employer2	Speaker Profile	String	10-200 Characters		Will either be a string that is extracted as a link or an actual employer name. Primarily for speakers who are current Ph.D students.
Speaker	employer1City	Speaker Profile	String	1-50 Characters	Atlanta	
Speaker	employer2City	Speaker Profile	String	1-50 Characters	Atlanta	
Speaker	employer1State	Speaker Profile	String	1-50 Characters	New York	
Speaker	employer2State	Speaker Profile	String	1-50 Characters	New York	
Speaker	employer1URL	Speaker Profile	String	1-50 Characters		Will be a string that is extracted as a link to the employer's URL if one exists.
Speaker	employer2URL	Speaker Profile	String	1-50 Characters		Will be a string that is extracted as a link to the employer's
Speaker	presentation	Speaker Profile	String	10-200 Characters		Will be a URL linking to the speaker's presentation
Speaker	semester	Speaker Profile	String	4-6 Characters	Fall	
Speaker	year	Speaker Profile	Int		2010	
Client	firstName	Client Profile	String	5-50 Characters	Timoth	
Client	lastName	Client Profile	String	5-50 Characters	Lederman	
Client	title	Client Profile	String	10-100 Characters	Professor of Computer Science	

Client	organization	Client Profile	String	10-100 Characters	Siena College	
--------	--------------	----------------	--------	----------------------	---------------	--

3.2 Logical Format of Data Files and Databases

3.2.1 Microsoft Access 2007 Relationships Diagram

The image provided below is a visual representation of the database tables, fields, and relationships between the tables and their fields. The fields that have a key to the left of them represent the primary keys of the table.

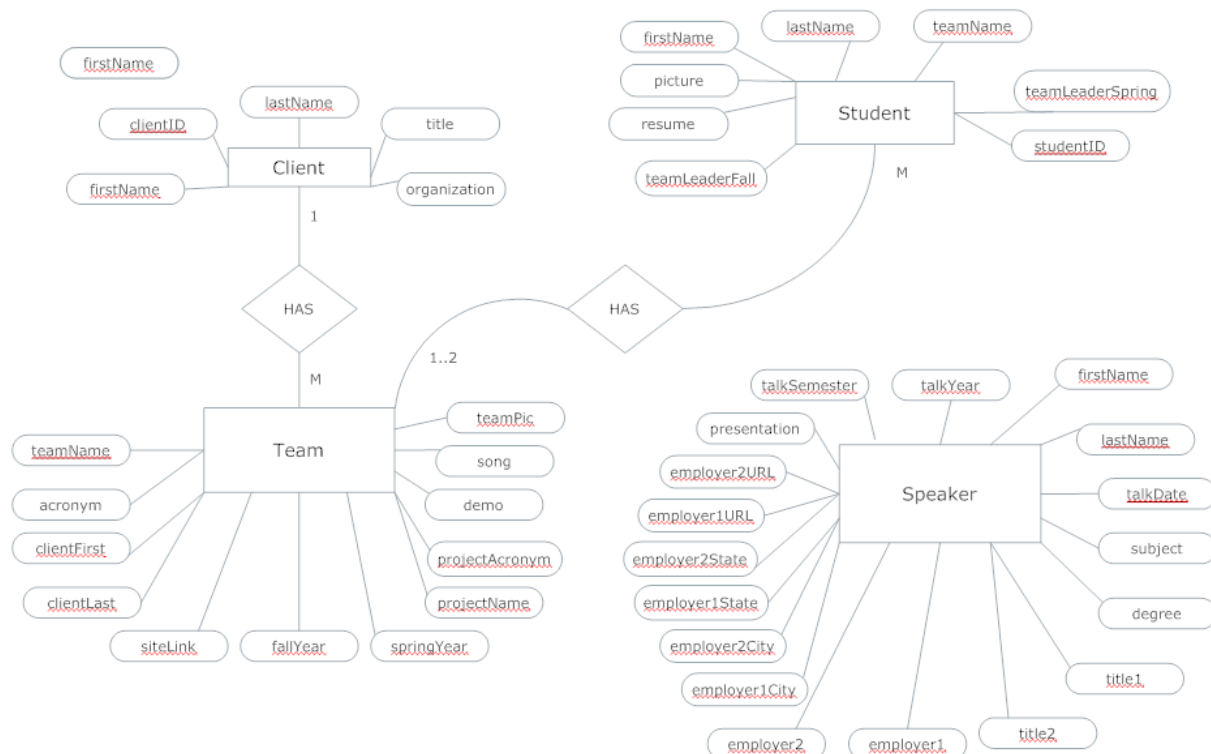


3.2.2 Entity-Relationship (ER) Diagram

AN ER diagram is a graphical representation of entities and their relationships to each other. An ER diagram consists of three different features: entities, relationships, and attributes. Entities are displayed as rectangles and represent a table in the database. Relationships are displayed as diamonds and represent the relationships between two entities. These relationships usually have a quantitative value associated with them:

- One-to-One: A relationship is defined as one-to-one if one instance of an entity (A) is associated with one other instance of another entity (B).
- One-to-Many: A relationship is defined as one-to-many if one instance of an entity (A) is associated with zero, one or many instances of another entity (B), but for one instance of entity B there is only one instance of entity A.
- Many-to-Many: A relationship is defined as many-to-many if one instance of an entity (A) is associated with one, zero, or many instances of another entity (B), and one instance of entity B is associated with one, zero, or many instances of entity A.

The last feature of ER diagrams are attributes which are displayed as ovals. Attributes represent data that is associated with an entity or a relationship.



3.2.3 Database Relational Schema

**Student(studentID, firstName, lastName, picture, resume, teamLeaderFall,
teamLeaderSpring, teamName)**

**Team(teamName, acronym, clientID, siteLink, fallYear, springYear,
projectName, projectAcronym, demo, song, teamPic)**

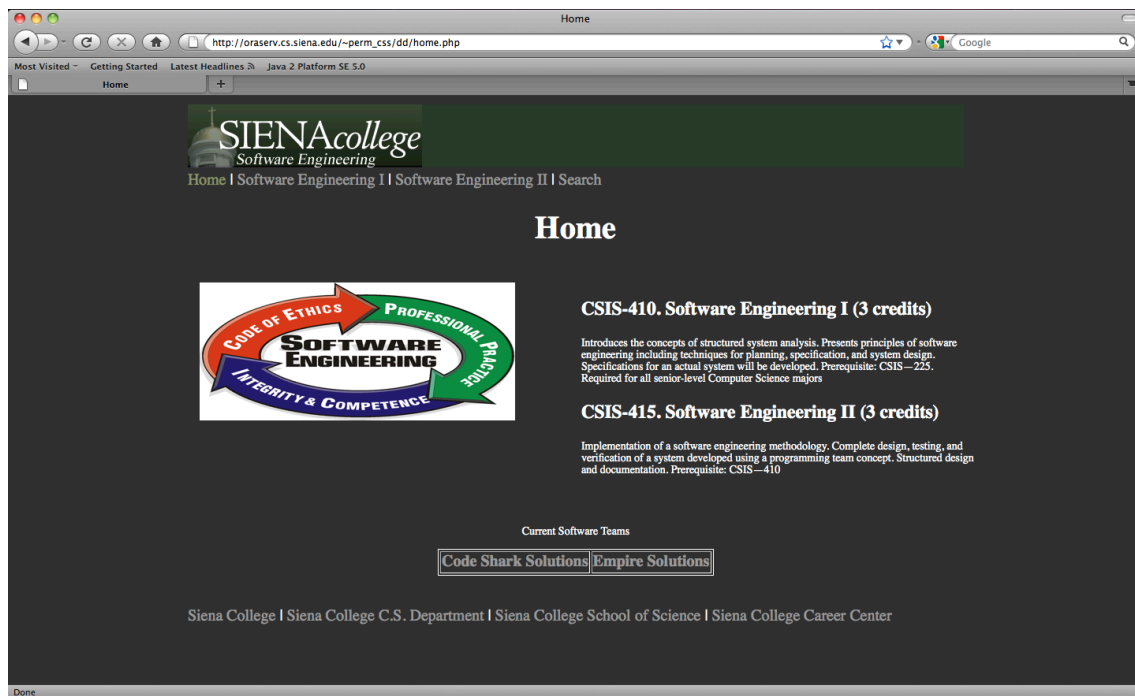
Client(clientID, firstName, lastName, title, organization)

**Speaker(firstName, lastName, talkDate, subject, degree, title1, title2, employer1,
employer2, employer1City, employer2City, employer1State, employer2State,
employer1URL, employer2URL, presentation, talkSemester, talkYear)**

4 Prototype Screens

As we are getting closer to the actual development of the new website, we have designed a semi-functional prototype to help us and the client get a better understanding of what direction the project is heading in. Although the prototype is somewhat functional, all of the design aspects are subject to change if the client desires. The main purpose of this is to get an idea of how the new Software Engineering site will be implemented. Some images of the current Software Engineering site are included to rationalize some of our decisions.

4.1 Home Page



The home page is the initial page the user will be directed to when first going to the site. There will be a small paragraph describing the course as a whole, along with some image that the client desires, and links to the current teams in Software Engineering. Descriptions of the menus on the site will be described elsewhere.

4.2 Main Menu

[Home](#) | [Software Engineering I](#) | [Software Engineering II](#) | [Search](#)

For our main menu, which will be located at the top left section of the screen, consistency is top priority. One of the major issues of the current site is the confusing navigation (shown below). This menu will be included on every page of the site. The items are links to the home page, Software Engineering I syllabus page, Software Engineering II syllabus page, and the search page. The currently displayed page will have its name in color.

Current menus:

[« Siena Software Engineering](#) by [Dr. Timothy C. Lederman](#)

CSIS-415 Software Engineering II

Siena College, Computer Science Department

[Calendar](#) [Syllabus/Grading](#) [Speakers](#) [Movies](#) [Killer Robot Discussions](#) [Siena Computer Science](#)

CALENDAR OF EVENTS FOR THE SPRING 2010 SEMESTER

[« Siena Software Engineering](#) by [Dr. Timothy C. Lederman](#)

CSIS-410 & CSIS-415 Software Engineering I & II

Siena College, Computer Science Department

[Home](#) [CSIS-410 Syllabus](#) [CSIS-415 Syllabus](#) [Course-Related Links](#) [Siena Computer Science](#) [Siena College](#)

[Course Descriptions](#) - [Team Websites](#) - [Historical Record of Projects/Teams](#)

4.3 Sub Menu



Home | **Software Engineering I** | Software Engineering II | Search
Course Description | Calendar | Grading | Syllabus

The sub menu is for additional pages for items on the main menu. Software Engineering I & II are the pages that will have a sub menu. A page for the course description, course calendar, grading, and syllabus will be on both Software Engineering I & II's sub menus. However, Software Engineering II's sub menu will also include links to a speaker's page and killer robot page. The currently displayed page will have its name in color.

4.4 Footer Links

Siena College | Siena College C.S. Department | Siena College School of Science | Siena College Career Center

As requested by our client, we will have a set of links on the bottom of each page. These items include a link to Siena College website, Siena College C.S. department website, Siena College School of Science website, and the Siena College Career Center website.

4.5 Software Engineering I – Course Description



This is the page where the course description of Software Engineering I will be. When a user clicks on the Software Engineering I click on the menu, this is where they'll be directed.

4.6 Software Engineering I – Calendar

Software Engineering I

http://oraserv.cs.siena.edu/~perm_css/dd/calendar1.php#November29

Course Description | Calendar | Grading | Syllabus

Software Engineering I - Calendar

COURSE CALENDAR FOR THE FALL 2010 SEMESTER

Month: Day:

November	29	<ul style="list-style-type: none"> Data Dictionary example from each team Team Meeting 	Provide (via email) sample entries from two units
December	1	Development vs Production Environments	
December	3	Review for Final Exam (25% of final grade)	
December	6	PPT discussion – Team Meeting	
December	7		Documents delivered BY NOON
December	8	<ul style="list-style-type: none"> Powerpoint Handouts are due for all teams ALL TEAMS Members and Leaders turn-in updated resume, appropriate rating sheets, 	<ul style="list-style-type: none"> Turn-in rating sheets - for only the Preliminary Design Turn in log - cumulative for entire semester Turn in attendance sheets for entire semester (Team

Siena College | Siena College C.S. Department | Siena College School of Science | Siena College Career Center

The calendar for the new website will be more polished than the current one (below). Instead of scrolling the entire page, the user can scroll within the page. In addition, the calendar will be implemented to display the current date in the window. If the current date is not an item on the calendar, the calendar will go to the closest next date. In addition, the calendar will start from the top when the date is past the last date on the calendar. The specifics on this are still in discussion. Lastly, we allow the user to input a date to go to, and the calendar will then display that date. Our client decided to not include the course related links page in the new Software Engineering site, so all those links will also be placed in the calendar.

Current calendar page:

CALENDAR OF EVENTS FOR THE FALL 2010 SEMESTER

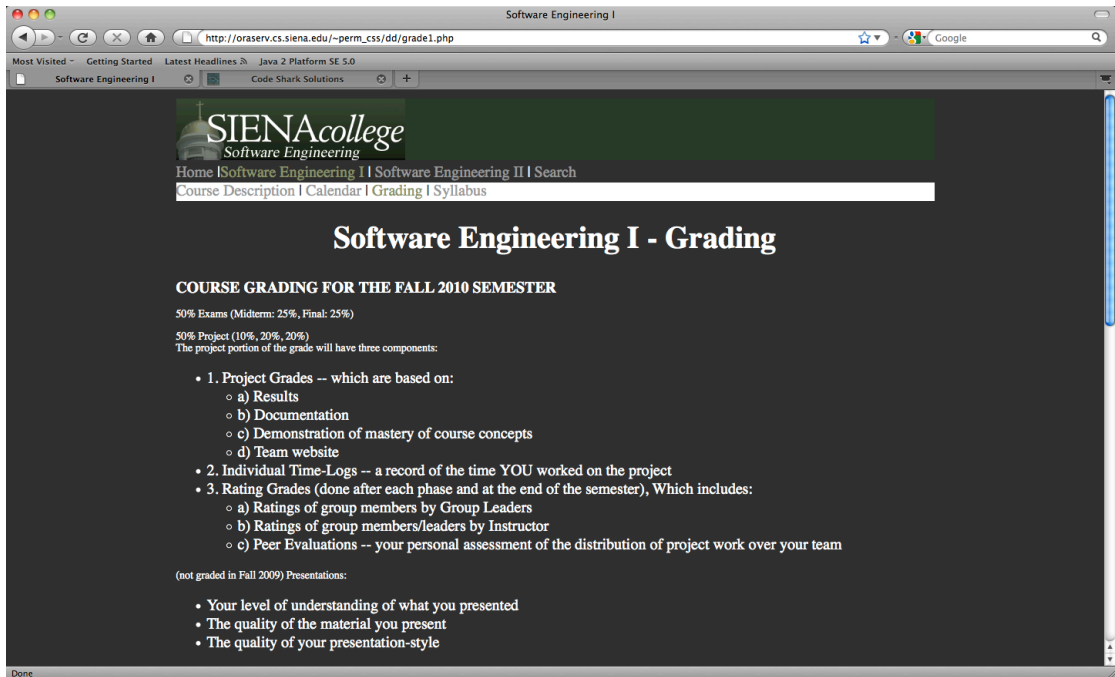


Course Description

Introduces the concepts of structural system analysis. Presents principles of software engineering including techniques for planning, specification, and system design. Specifications for an actual system will be developed. Prerequisite: CS25 – 225. Recommended for all senior-level Computer Science majors (Required for all major who matriculate Fall 2009 or later).

Month	Day	In-Class Activities	Assignments
September	8	Orientation	<ul style="list-style-type: none"> Send instructor e-mail nominations for team leaders
September	10	<ul style="list-style-type: none"> Vote on Team/Group Leaders (in class) Small/voice-mail etiquette Slur to shake hands and make introductions Teams/Groups formed by team leaders & Dr. Loferman (after class) 	<ul style="list-style-type: none"> Definition: Software Engineering E-mail etiquette Phone & Voicemail etiquette Get your resume ready
September	13	<ul style="list-style-type: none"> Discuss first meeting with client Discussion: Software Plan Outline 	<ul style="list-style-type: none"> Decide upon team member's "roles/titles" by Friday Look at previous Software Plans Various models for Software Engineering

4.7 Software Engineering I – Grading



This page will include all the information regarding the grading of Software Engineering I.

4.8 Software Engineering I – Syllabus



This page will act as the online version of the syllabus for Software Engineering I.

4.9 Software Engineering II – Course Description



This is the page where the course description of Software Engineering II will be. When a user clicks on the Software Engineering II click on the menu, this is where they'll be directed.

4.10 Software Engineering II – Calendar

Software Engineering II

http://oraserv.cs.siena.edu/~perm_css/dd/calendar2.php#February27

Most Visited - Getting Started Latest Headlines Java 2 Platform SE 5.0

Software Engineering II Code Shark Solutions

Course Description | Calendar | Grading | Syllabus | Speakers | Killer Robot |

Software Engineering II - Calendar

CALENDAR OF EVENTS FOR THE SPRING 2010 SEMESTER

Month: Day:

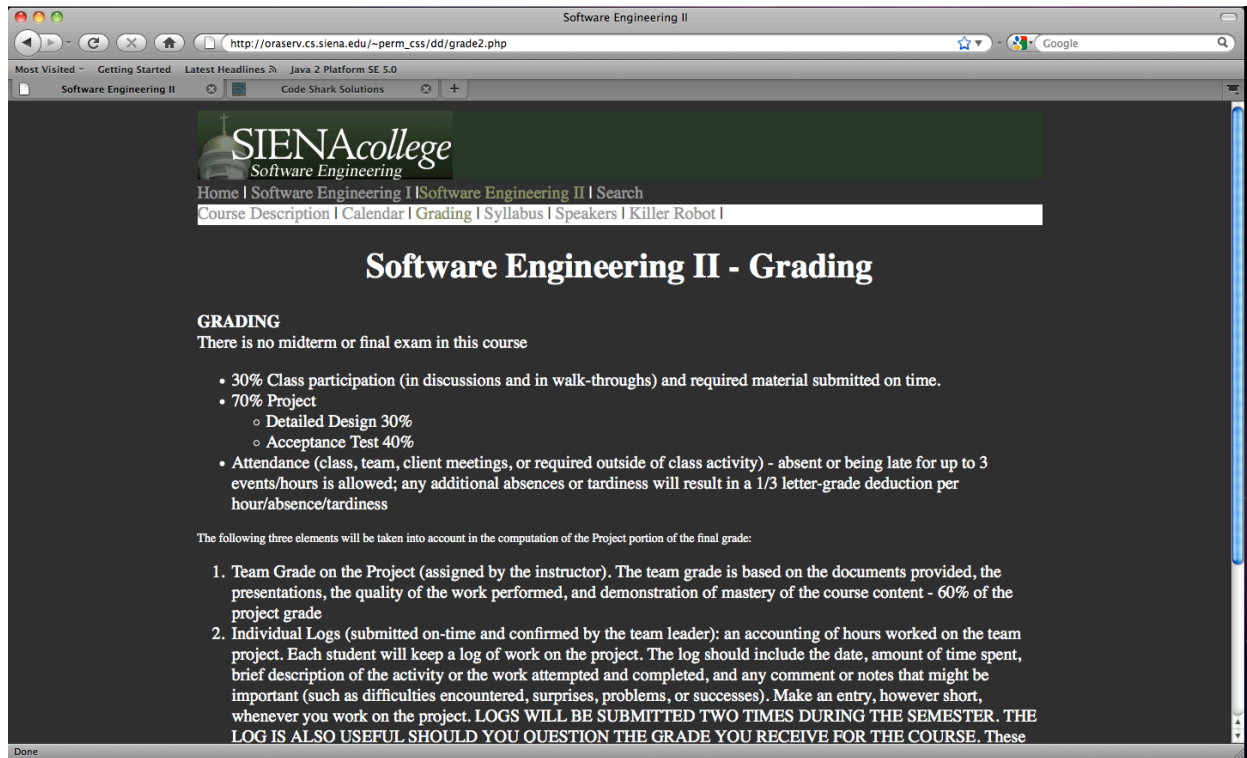
Tuesday, February 23
Winter Break

Tuesday March 1	<i>Presentations:</i>	<p>Jim Dzembo (Siena, 2007) Information Technology Specialist II - Programmer New York State Department of Taxation and Finance</p> <p>Mark Riley (Siena, 2007) IT Specialist 3 (formerly called Associate Programmer/Analyst) New York State Department of Motor Vehicles, Applications Development <i>"Employment at NYS Government Offices"</i></p>
	<i>Discussion:</i>	"The Case of the Killer Robot", led by 518 Interactive
	<i>Walk-Through:</i>	6. Silicon Technionics promised to deliver a safe robot
		Detailed Design questions; Project Walk Throughs (30 minutes)
Tuesday March 9	<i>Deliver:</i>	Monday by noon: Detailed Design Document
	<i>Presentations:</i>	Tuesday: Detailed Design Presentations

Siena College | Siena College C.S. Department | Siena College School of Science | Siena College Career Center

This is the calendar page for Software Engineering II. It will have the same functionality and design as the calendar page for Software Engineering I.

4.11 Software Engineering II – Grading



The screenshot shows a web browser window with the URL `http://oraserv.cs.siena.edu/~perm_css/dd/grade2.php`. The page title is "Software Engineering II". The browser's address bar shows "Google". The page content includes the Siena College logo and navigation links: Home | Software Engineering I | Software Engineering II | Search | Course Description | Calendar | Grading | Syllabus | Speakers | Killer Robot |. The main heading is "Software Engineering II - Grading".

GRADING
There is no midterm or final exam in this course.

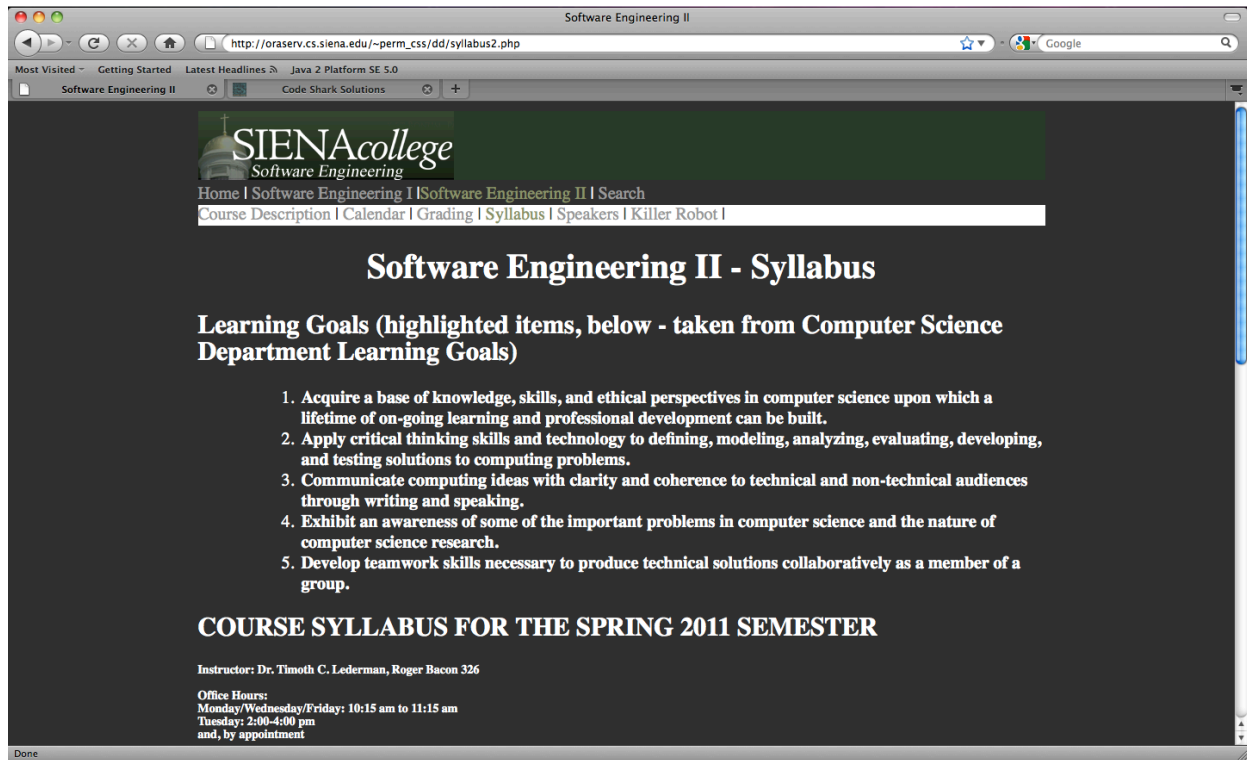
- 30% Class participation (in discussions and in walk-throughs) and required material submitted on time.
- 70% Project
 - Detailed Design 30%
 - Acceptance Test 40%
- Attendance (class, team, client meetings, or required outside of class activity) - absent or being late for up to 3 events/hours is allowed; any additional absences or tardiness will result in a 1/3 letter-grade deduction per hour/absence/tardiness

The following three elements will be taken into account in the computation of the Project portion of the final grade:

1. Team Grade on the Project (assigned by the instructor). The team grade is based on the documents provided, the presentations, the quality of the work performed, and demonstration of mastery of the course content - 60% of the project grade
2. Individual Logs (submitted on-time and confirmed by the team leader): an accounting of hours worked on the team project. Each student will keep a log of work on the project. The log should include the date, amount of time spent, brief description of the activity or the work attempted and completed, and any comment or notes that might be important (such as difficulties encountered, surprises, problems, or successes). Make an entry, however short, whenever you work on the project. LOGS WILL BE SUBMITTED TWO TIMES DURING THE SEMESTER. THE LOG IS ALSO USEFUL SHOULD YOU QUESTION THE GRADE YOU RECEIVE FOR THE COURSE. These

This is the grading page for Software Engineering II. It will have the same functionality and design as the grading page for Software Engineering I.

4.12 Software Engineering II – Syllabus



The screenshot shows a web browser window with the URL http://oraserv.cs.siena.edu/~perm_css/dd/syllabus2.php. The page features the Siena College logo and navigation links: Home, Software Engineering I, Software Engineering II, Search, Course Description, Calendar, Grading, Syllabus, Speakers, and Killer Robot. The main heading is "Software Engineering II - Syllabus". Below this, the text reads "Learning Goals (highlighted items, below - taken from Computer Science Department Learning Goals)". A list of five goals follows:

1. Acquire a base of knowledge, skills, and ethical perspectives in computer science upon which a lifetime of on-going learning and professional development can be built.
2. Apply critical thinking skills and technology to defining, modeling, analyzing, evaluating, developing, and testing solutions to computing problems.
3. Communicate computing ideas with clarity and coherence to technical and non-technical audiences through writing and speaking.
4. Exhibit an awareness of some of the important problems in computer science and the nature of computer science research.
5. Develop teamwork skills necessary to produce technical solutions collaboratively as a member of a group.

Below the list, the text reads "COURSE SYLLABUS FOR THE SPRING 2011 SEMESTER". The instructor is listed as "Instructor: Dr. Timothy C. Lederman, Roger Bacon 326". Office hours are provided: "Office Hours: Monday/Wednesday/Friday: 10:15 am to 11:15 am; Tuesday: 2:00-4:00 pm and, by appointment".

This is the syllabus page for Software Engineering II. It will have the same functionality and design as the syllabus page for Software Engineering I.

4.13 Software Engineering II – Speakers



Software Engineering II

http://oraserv.cs.siena.edu/~perm_css/dd/speakers.php

Most Visited - Getting Started Latest Headlines Java 2 Platform SE 5.0

Software Engineering II Code Shark Solutions

SIENAcollege
Software Engineering

Home | Software Engineering I | Software Engineering II | Search
Course Description | Calendar | Grading | Syllabus | Speakers | Killer Robot |

Software Engineering II - Speakers

Throughout the Spring semester, Siena College Computer Science Alumni and other professionals are scheduled to make presentations about how their organizations develop software. The speakers provide information on how they employ different models for Software Engineering, with emphasis on requirements, design, and testing. Speakers discuss techniques, policies, and procedures that are followed at their organization (or that are not being followed, or that have been followed but are no longer deployed). In addition, the speakers discuss their career paths, their continued education (graduate courses, graduate degrees, training courses, and professional certifications), and how they balance their personal and private lives.



Tuesday February 1	no presentation
Tuesday February 8	Snow Day Cancellation
Tuesday February 15	Das Nobel, BS (Siena, 2006), MS (???) "Database Optimization"
Tuesday February 22	Ken Swarner and Eric Crossman, BS (Siena 2003) Systems & Operations Managers, ITS, Siena College "PHP For Fun & Profit"

Done

This page is dedicated to all the speakers for the semester.

4.14 Software Engineering II – Killer Robot

Software Engineering II

http://oraserv.cs.siena.edu/~perm_css/dd/robot.php

Most Visited - Getting Started Latest Headlines Java 2 Platform SE 5.0


Software Engineering II Code Shark Solutions

SIENAcollege
Software Engineering

Home | Software Engineering I | Software Engineering II | Search
Course Description | Calendar | Grading | Syllabus | Speakers | Killer Robot

Software Engineering II - Kill Robot Discussion

"The Case of the Killer Robot" - Discussions for the Spring 2011 Semester
The Killer Robot Papers
Richard G. Epstein
West Chester University of PA, West Chester, PA 19383
epstein@golden.wcupa.edu
Copyright © 1989,1994 Richard G. Epstein



Permission is granted to copy this material for use in classroom instruction at a college or university. This material may not be copied for any other purpose without express written permission of the author.

This online archive contains the Killer Robot Papers consisting of seven newspaper articles, one journal article and one Sunday newspaper magazine interview. The Case of the Killer Robot book, published by John Wiley & Sons (Epstein 1997), contains a lot more material.

The persons and institutions involved in this scenario are entirely fictitious (except for the references to the venerable computer scientists: Ben Shneiderman and Jim Foley). Silicon Valley was chosen as the location for the accident because Silicon Valley is an icon of high technology. All of the persons, business entities and institutions named in Silicon Valley are purely fictitious. Silicon Valley University is purely fictitious and has nothing to do with any actual university in Silicon Valley or elsewhere. It is just a fictitious device. N.K.Taylor@hw.ac.uk

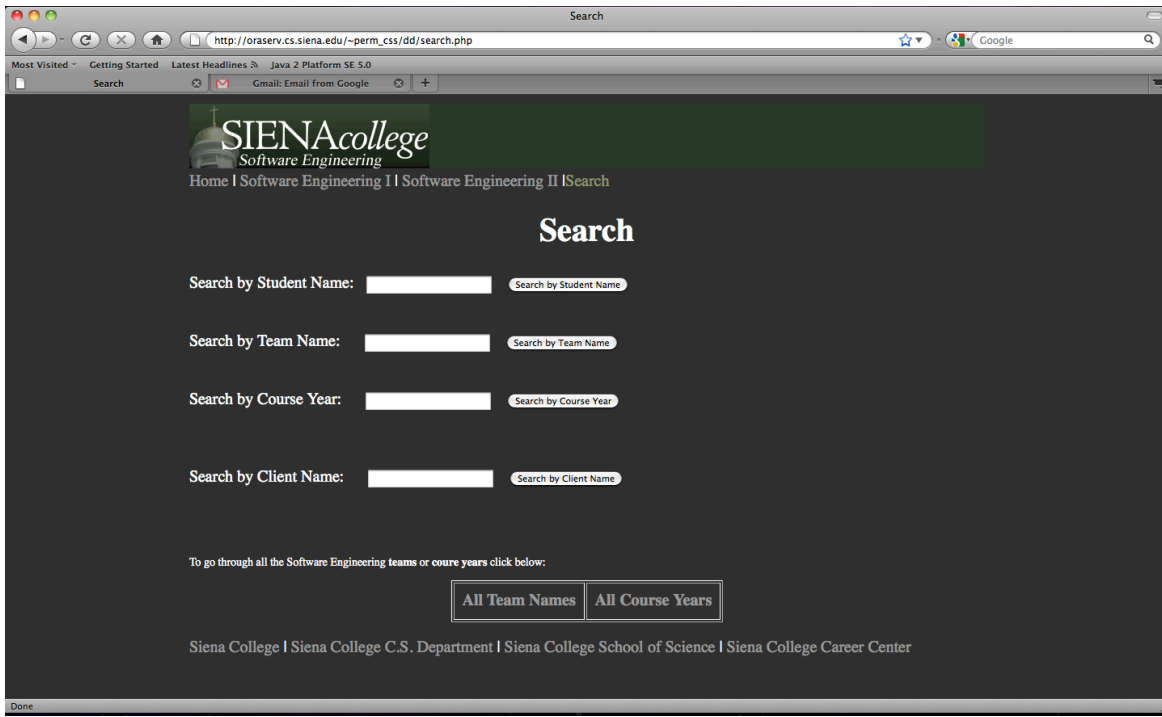
Questions for Killer Robot Discussions

Tuesday January 27	<ul style="list-style-type: none"> • Introduction, • Cast of Characters, • Words, Terms, & Allusions 	0. Dr. Lederman
-----------------------	---	------------------------

Done

This page is where all the dates of the killer robot discussion will be.

4.15 Search page



A new feature on the new Software Engineering website will be the search page. Here, the user can search by student name, team name, course year, or client. If the user wishes, they can view all team names and course years (images below) by clicking on their respective links.

4.15 Search page cont.

All course years page:

The screenshot shows a Mozilla Firefox browser window with the URL http://oraserv.cs.siena.edu/~perm_csu/dd/allyear.php. The page header includes the Siena College logo and navigation links: Home | Software Engineering I | Software Engineering II | Search. The main heading is "All Software Engineering Course Years" with a sub-heading "(for which we have records)". Below this is a table listing course years in a 5x6 grid.

1984-1985	1989-1990	1994-1995	1999-2000	2004-2005	2009-2010
1985-1986	1990-1991	1995-1996	2000-2001	2005-2006	2010-2011
1986-1987	1991-1992	1996-1997	2001-2002	2006-2007	
1987-1988	1992-1993	1997-1998	2002-2003	2007-2008	
1988-1989	1993-1994	1998-1999	2003-2004	2008-2009	

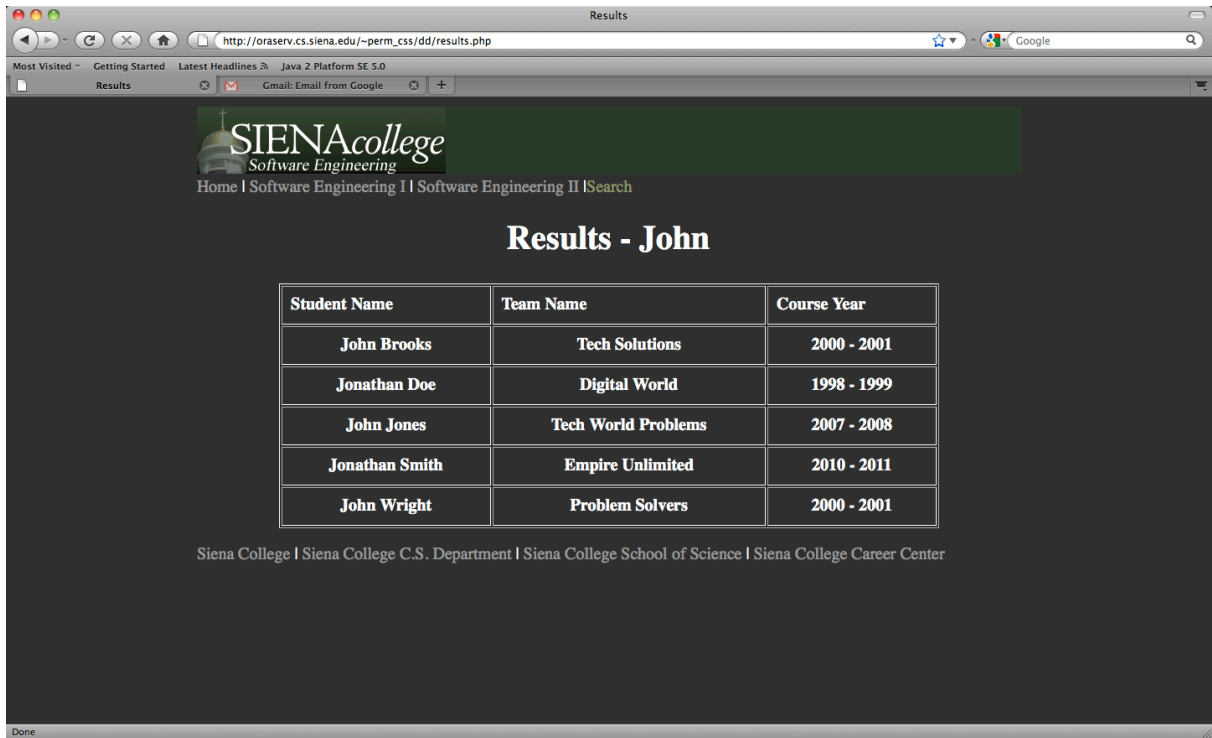
Footer text: Siena College | Siena College C.S. Department | Siena College School of Science | Siena College Career Center

4.15 Search page cont.

All teams page:



4.16 Results – Multiple Results



This page is where the user will be directed after attempting a search. The user will be given a table of students, teams, and course years that are close to what the user inputted. They will then be able to click on the results in the table to get more information on the selected link.

4.17 Results – Student Information

Results - Brittany Lintelman

http://oraserv.cs.siena.edu/~perm_css/dd/resultname.php

Most Visited - Getting Started Latest Headlines Java 2 Platform SE 5.0

Results - Brittany Lintelman Code Shark Solutions

Print this page

2010 - 2011

Brittany Lintelman

Senior-Year Resume

Team Members:

Brittany Lintelman

Rebecca Wilson

Kurt Greiner

Ryan Godfrey

Akeem Shirley

Daniel Rotondo

Code Shark Solutions

Project: R.O.S.E.S.
(Redesign Of the Software Engineering Site)

Client: Dr. Lederman

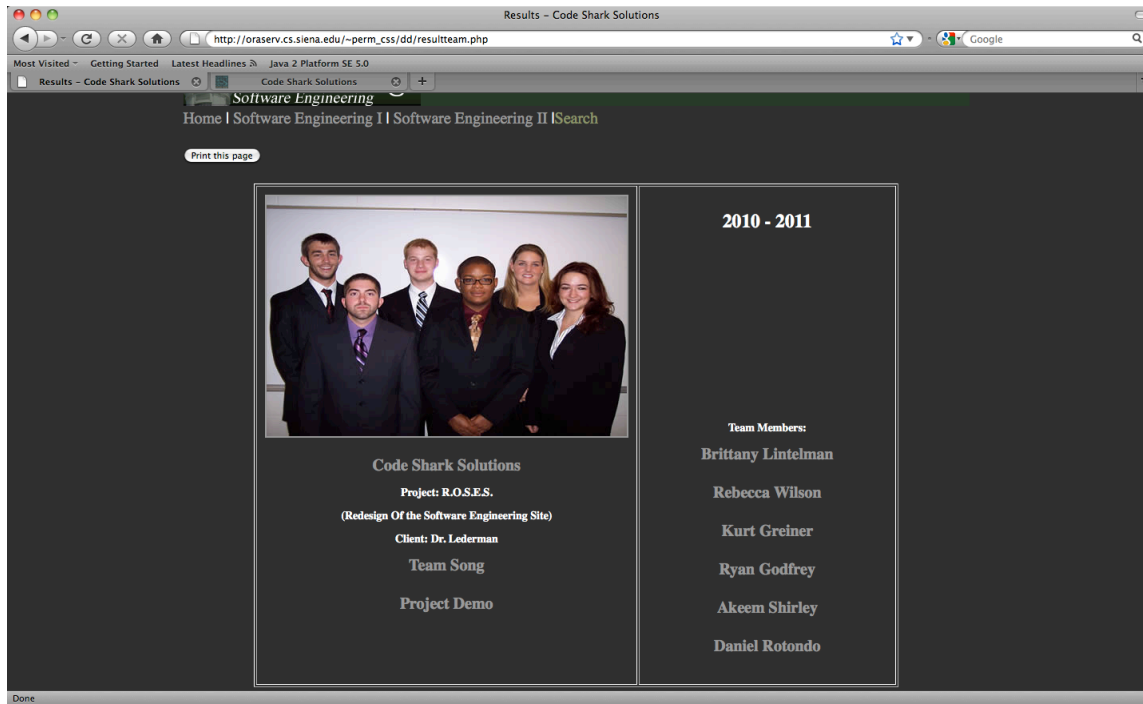
Team Song

Project Demo

Done

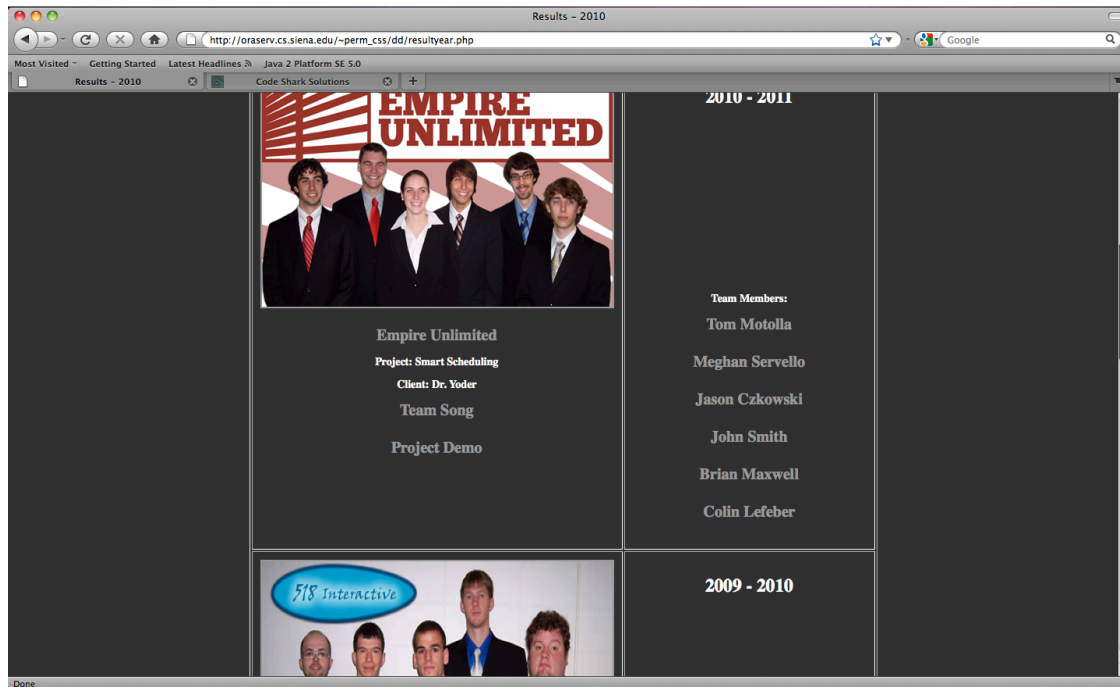
If the user selects a student name, the page above will appear. The information on the left side of the table will include the individuals' software engineering team name, team picture, team project name, the project name's acronym, client name, link to team website, and if they have it, links to their team song and project demo. On the right column will be the course years the individual was in Software Engineering, a picture of the individual, their first and last name, link to their resume, and links to all the other team members resumes along with the first and last names. A print button is also on the page to print the page.

4.18 Results – Team Information



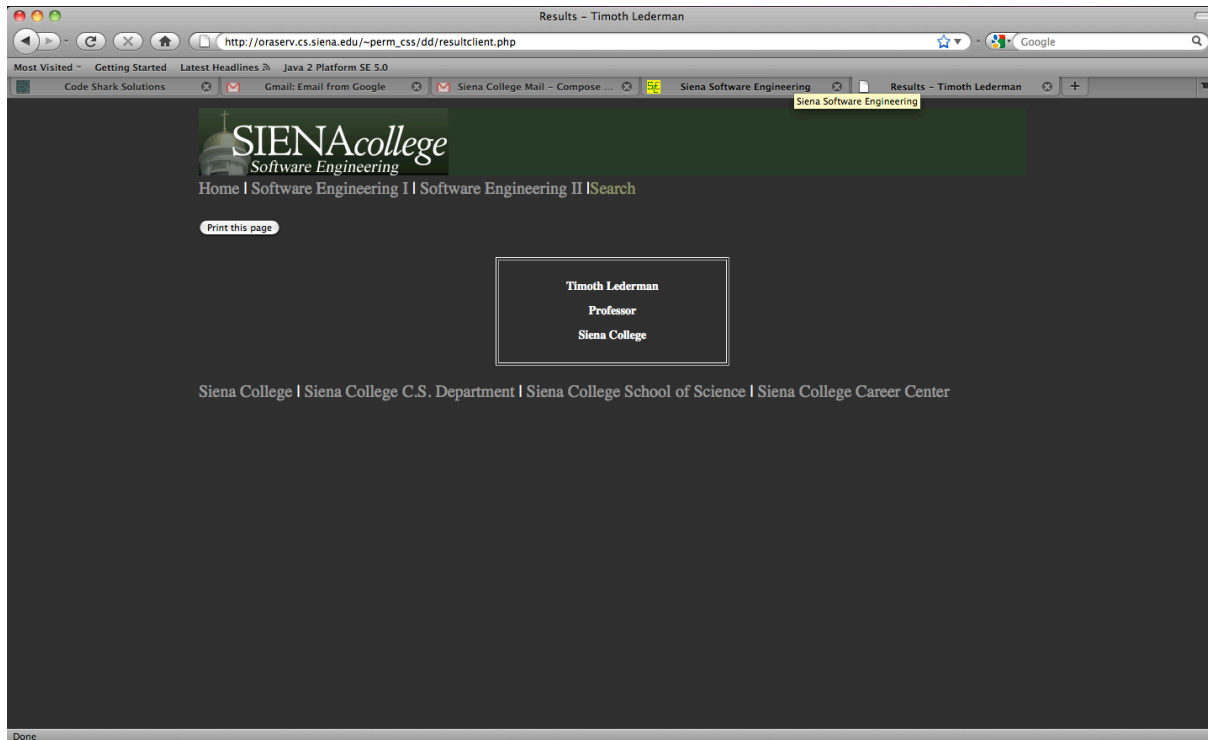
If the user decides to click on a team name on the results page, they will be directed to a page that looks like this. The information on the left side of the table will include the software engineering team name, team picture, team project name, the project name's acronym, client name, link to team website, and if they have it, links to their team song and project demo. On the right column will be the course years the team was in Software Engineering and links to all the team members' senior resumes along with the first and last names. A print button is also on the page to print the page.

4.19 Results – Course Year Information



If the user clicks on a course year on the results page, they will be directed to a page that look likes the image above. This page will be a table of teams that has all the information as the team information page. This page displays only the teams that existed during that specific year. For example, all the teams from Spring 2010 and Fall 2010 would be displayed if the user searched for course year “2010”.

4.20 Results – Client Information



If the user searches for a client, they will be directed to a page that looks like this. The specifics of what is to be displayed on this result page are still very much in discussion. On this page is the client's name, title, and organization are listed.

5 Test Plan

5.1 Testing Plan Identifier

For the Test Plan of R.O.S.E.S., we will have all the necessary testing requirements in order to make sure the application is complete and functional. As we progress to the completion of the project, the individual requirements will be fulfilled while still allowing some items to be changed or eliminated. The purpose of this document is to keep organized of these changes during the development of the application in order to assure the Acceptance Test fully tests the system.

5.2 Introduction

This document outlines and reviews the documents requirements. It also is used as a plan for testing those requirements in order to ensure each requirement is fully met and the system works as specified. This plan will cover each of the necessary testing of internal components. The test plan is broken up into multiple parts including the System Test, Integration/Regression Tests, Unit Tests, and the full Acceptance Test. The Unit Test will test each part of the system to make sure it runs as expected. After completing the Unit Test, the Integration Test will be preformed to in order to make sure that any data the Unit Test may have been altered was changed as expected and stored accordingly. The System Test includes all the Functional Requirements agreed upon with the client in the document. The final piece of the software to be tested is the Non-Functional Requirements. The Acceptance Test will make up of all these tests together.

5.3 Test Items and Functional Requirements Inventory

This section will outline all the items that can be physically tested, unlike the non-functional requirements which will be discussed later in the document. This section will be used as a guide in all tests to ensure that the requirements are received and tested properly. This section of testing will require the testing and development teams to fulfill the requirements specified by our client. The site will run on all major browsers including, Internet Explorer 7 and 9, 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. However, each of the 6 users (Student, Faculty, Family and Friends, Future Employer and Course Instructor), all have the same capabilities. So we will include all of the users under the General User.

The General User

Will be able to view the main website?

YES NO

Will be able to view the calendars?

YES NO

Will be able view the syllabi?

YES NO

Will be able to view the different courses information?

YES NO

Will be able to view the current and previous teams' websites?

YES NO

Will be able to previous and current student's information?

YES NO

Will be able to view previous and current student's picture?

YES NO

Will be able to view all tabs of groups they are members of?

YES NO

Will be able to view previous and current student's resume?

YES NO

Search Function

Will be able to search by team names?

YES NO

Will be able to search by team member names?

YES NO

Will be able to search by year of a team member?

YES NO

5.4 Non-Functional Requirements

Testing is required for the Non-Functional Requirements in addition to the Functional Requirements. Since these requirements are primarily what R.O.S.E.S. is to *be* rather than what it *does*, these can be difficult to test. The following are R.O.S.E.S. Non-Functional requirements, which specify how the system should be.

The system must be aesthetically pleasing. The design of the system must make sense stylistically, and must not interrupt from the content of the site due to poor styling.

The system must be easily navigable. The user interface must be logical and easy to navigate. The user should be able to get to anywhere on the site without any struggles.

The system must be easily maintainable. The Web Administrator of the system should be able to make changes to the system with little effort.

The system must be easily modifiable. The Web Administrator should be able to modify the style and content of the site easily. They should be able to clearly understand the HTML/PHP/JavaScript and be able to modify it based on how it was implemented, with little knowledge of the programming languages.

The system must be stable and secure. The system should be able to perform efficiently even under a high level of activity while maintaining a high level of performance. In addition, the system must be protected against internal and external attacks as best as possible.

5.5 Exception Handling to Test

Due to actions outside the control of the system, certain areas in R.O.S.E.S. may have errors occur. The system will be required to handle any of these types of exceptions.

On the search page, we will have the user fill out a form to search for a student, team, or course year. If a user puts alphabetical character inside of the course year input box or a too small value, an error message will occur to notify the user of their error.

1.6 Testing Approach

The method our team will use to test the new Software Engineering site(R.O.S.E.S.) will be to follow all guidelines outlined in this document to ensure all the functional and non-functional requirements are met. The testing will be a multi-step process that consists of activities for validating the software product, from the most primitive elements up to the fully integrated system. This area includes activities such as unit testing, integration testing, system testing, performance testing, and acceptance testing.

Our strategy consists of dividing the project up into separate modules, each with a distinct functionality. Each module has a Unit Test script that is followed, moving test case by test case to ensure that all parts of the module are thoroughly tested. In order for each unit to be considered 'passed' each test case in the unit must pass. Once each unit has passed testing, the units that interact with each other will be tested to make sure that they work together properly. The system is then tested as a whole to make sure all the functional and non-functional requirements have been met. When errors are found, Code Shark Solutions will come with a solution to the problem and then repeat the testing procedure to make sure the fix didn't create a problem elsewhere.

Testing will not finish until Code Shark Solutions has determined that all functional and non-functional requirements have been met.

1.7 Acceptance Test – Acceptance Criteria

Acceptance Testing is concerned with knowledge about validating the functional and non-functional requirements of a system. This is a useful way to go through the efforts required to validate the acceptability of the system. After it's complete, the test plan will help people not involved in the process to understand the 'why' and 'how' of our product validation. Our test plan documents the way we will authenticate and make certain that our system being tested meets the requirements and specifications.

The Acceptance Criteria will be determined by the functional and non-functional requirements inventory listed in our other documents. What the system can do and what is testable is considered our functional requirements while the non-functional requirements are things our system will be. At the completion of this project, Code Shark Solutions will be able to determine which of the requirements were met and those that were not met. Please keep in mind that these requirements are still subject to change according to our client's preferences and more information is obtained.

R.O.S.E.S. will be tested on both Windows and Mac operating systems and with the four major browsers Internet Explorer 7 and 8, Mozilla Firefox, Safari and Google Chrome. Code Shark Solutions will determine the testing requirements. We will use three different types of tests: a full system test, individual unit tests, and then integrated/regression tests. Our results will be documented in the Acceptance Test document later on in the development process. Code Shark Solutions will use these results to figure out if all the requirements were met.

5.8 Unit Tests

The Unit Test will test each of the programs to make sure it runs as expected. It is concerned with knowledge about testing a program unit to determine that it is free of data, logic, or standards errors.

Please see next page for unit results

5.8.1 Directory

System Test - Test Results for All Unit Tests						
Code Shark Solutions						
R.O.S.E.S.						
Dr. Timoth Lederman						
Directory of Unit Tests						
Pass/Fail Status	Unit Number	Unit Test Name	Date Last Tested	Comments or brief description	Integrated with these units	
F	0%	1.1		Test Links on the homepage		
F	0%	1.2		Test Links on Software Engineering I Page		
F	0%	1.3		Test Links on Software Engineering II Page		
F	0%	2.1		Search By Student Name		
#REF!	0%	2.2		Search By Team Name		
#REF!	0%	2.3		Search By Course Year		

5.8.2 Unit 1.1

Test Links on Homepage

Unit 1.1

Test all links on the Homepage to make sure that they go to the right page

Test Cases										
Pass/ Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
	1.001	Navigate to Software Engineering Home page	N/A	Open your internet browser and enter the course URL into the address	Your browsers default page	Software Engineering Homepage				
	1.002	Navigate to "This Years Teams" Pages	N/A	Click on links for "This Year's Teams"	Software Main Homepage	Desired Team Homepage				
	1.003	Repeat 1.002 until all team links are tested	N/A							
	1.004	Test link to "Siena College" website	N/A	Click on the "Siena College Website"	Software Main Homepage	Siena College Website				
	1.005	Test link to "Siena College CS Dept"	N/A	Click on the "CS Dept" link	Software Main Homepage	CS Dept Website				
	1.006	Test link to the "School of Science" website	N/A	Click on the "Siena College School of Science" link	Software Main Homepage	School of Science Website				
	1.007	Test link for " Career Center" website	N/A	Click on the "Career Center"	Software Main Homepage	Career Center Website				
	1.008	Test "Home Link"	N/A	Click on the "Home" tab in the main menu	Software Main Homepage	Software Main Homepage				
	1.009	Test "Software Engineering I" link	N/A	Click on the "Software Engineering I" tab	Software Main Homepage	Software I Main Page		Navigate Back to "Software Main Homepage" after test		

	1.010	Test "Software Engineering II" link	N/A	Click on the "Software Engineering II" tab	Software Main Homepage	Software II Main Page		Navigate Back to "Software Main Homepage" after test		
	1.011	Test "Seach" link	N/A	Click on the "Search" tab	Software Main Homepage	Search Main Page		Navigate to Software Engineering I page to begin Unit 1.2		
	1.012	Repeat Unit 1.1 for all browsers (IE, Firefox, Chrome, Safari and Opera								
F	= Unit Summary			0% passing		0 passed		Date of last test =		
	11	tests				0 failed				

5.8.3 Unit 1.2

Test Links on Software Engineering I Page

Unit 1.2

Test all links on the Software Engineering I page to make sure that they go to the right page

Test Cases										
Pass/ Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
	1.001	Navigate to Software Engineering I Main Page								
	1.002	Test "Siena College" website link.	N/A	Click on "Siena College" link.	Software Engineering I Main Page	Siena College Website				
	1.003	Test link to "Siena CS Dept" website	N/A	Click on "Siena College CS Dept" link	Software Engineering I Main Page	Siena College Computer Science Website				
	1.004	Test link to "Siena College School of Science" website	N/A	Click on "Siena College School of Science" link	Software Engineering I Main Page	Siena College School of Science Website				
	1.005	Test link to "Career Center" website	N/A	Click on "Career Center" link	Software Engineering I Main Page	Siena College Career Center Website				
	1.006	Test link to Software Engineering I Calendar	N/A	Click on the "Calendar" menu item	Software Engineering I Main Page	Calendar Page				
	1.007	Test to make sure "Go to Date" button works	N/A	Select a Month and Day and Click button	Software Engineering I Calendar page open to current date	Calendar is now displaying the selected date				

	1.008	Test link to Software Engineering I Grading page	N/A	Click on the "Grading" item on the submenu	Software Engineering I main page	Software Engineering I Grading page				
	1.009	Test link to Software Engineering I Syllabus page	N/A	Click on the "Syllabus" menu item on the submenu	Software Engineering I Main Page	Software Engineering I Syllabus Page				
	1.010	Repeat Unit 1.1 for all browsers (IE, Firefox, Chrome, Safari and Opera								
F	= Unit Summary		0%	passing	0	passed	Date of last test =			
	10	tests			0	failed				

5.8.4 Unit 1.3

Test Links on Software Engineering II Page

Unit 1.3

Test all links on the Software Engineering II page to make sure that they go to the right page

Test Cases										
Pass/ Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
	1.001	Navigate to Software Engineering II main page								
	1.002	Test "Siena College" website link.	N/A	Click on "Siena College" link.	Software Engineering II Main Page	Siena College Website				
	1.003	Test link to "Siena CS Dept" website	N/A	Click on "Siena College CS Dept" link	Software Engineering II Main Page	Siena College Computer Science Website				
	1.004	Test link to "Siena College School of Science" website	N/A	Click on "Siena College School of Science" link	Software Engineering II Main Page	Siena College School of Science Website				
	1.005	Test link to "Career Center" website	N/A	Click on "Career Center" link	Software Engineering II Main Page	Siena College Career Center Website				
	1.006	Test link to Software Engineering II Calendar	N/A	Click on the "Calendar" item in the submenu	Software Engineering II Main Page	Software Engineering II Calendar Page				

	1.007	Test to make sure "Go to Date" button works	N/A	Select a Month and Day and Click button	Software Engineering II Calendar page open to current date	Calendar is now displaying the selected date				
	1.008	Test Syllabus Menu Item	N/A	Click on the "Syllabus" item in the submenu	Software Engineering II Main Page	Software Engineering II Syllabus Page				
	1.009	Test Speakers Menu item	N/A	Click on the "Speakers" item in the submenu	Software Engineering II Main Page	Software Engineering II Speakers Page				
	1.010	Test Killer Robot Menu item	N/A	Click on the "Killer Robot" item in the submenu	Software Engineering II Main Page	Software Engineering II Killer Robot Page				
	1.011	Repeat Unit 1.1 for all browsers (IE, Firefox, Chrome, Safari and Opera								
F	= Unit Summary		0% passing		0 passed		Date of last test =			
	11	tests			0 failed					

5.8.5 Unit 2.1

Search By Student Name										
Unit 2.1										
To search for a particular student by their name										
Test										
Pass/ Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
	2.1.0	Navigate to Software Engineering Site	N/A	Open a web browser and go to the site	Browsers default page	Software Engineering Site				
	2.2.0	Navigate to Search page	N/A	Click on the "Seach" menu item in the main menu	Software Engineering Home Page	Search Page				
	2.3.0	Click in the Student Name edit field	N/A	Move cursor to the search criteria box	Empty Search Page fields	Cursor is in the Student Name box				
	2.4.1	Type in a name	N/A	Cursor must be in Student Name box	Empty Student Name box	Student Name box has input				
	2.5.0	Click the Search Button	N/A	Student Name Box must have input	Search Page	Results Page				
	2.6.0	Choose Correct Student from the List of Results	N/A	On the Results Page	Results Page	Student's Information Page				
	2.7.0	Click on the links to view different information about the student	N/A	Be on the Information Page	Information Page	Desired information is displayed				
F	= Unit Summary		0%	passing	0	passed	Date of last test =			
	7	tests			0	failed				

5.8.6 Unit 2.2

Search By Team Name										
Unit 2.2										
To search for a particular team by their name										
Test										
Pass/ Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
	2.1.0	Navigate to Software Engineering Site	N/A	Open a web browser and go to the site	Browsers default page	Software Engineering Site				
	2.2.0	Navigate to Search page	N/A	Click on the "Seach" menu item in the main menu	Software Engineering Home Page	Search Page				
	2.3.0	Click in the Team Name edit field	N/A	Move cursor to the search criteria box	Empty Search Page fields	Cursor is in the Team Name box				
	2.4.1	Type in a name	N/A	Cursor must be in Student Name box	Empty Student Name box	Team Name box has input				
	2.5.0	Click the Search Button	N/A	Team Name Box must have input	Search Page	Results Page				
	2.6.0	Choose Correct Team from the List of Results	N/A	On the Results Page	Results Page	Team's Information Page				
	2.7.0	Click on the links to view different information about the team	N/A	Be on the Information Page	Information Page	Desired information is displayed				
F	= Unit Summary		0% passing			0 passed		Date of last test =		
	7	tests				0 failed				

5.8.7 Unit 2.3

Search By Course Year										
Unit 2.3										
To search for Teams in a particular Course Year										
Test										
Pass/ Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
	2.1.0	Navigate to Software Engineering Site	N/A	Open a web browser and go to the site	Browsers default page	Software Engineering Site				
	2.2.0	Navigate to Search page	N/A	Click on the "Seach" menu item in the main menu	Software Engineering Home Page	Search Page				
	2.3.0	Click in the Year edit field	N/A	Move cursor to the search criteria box	Empty Search Page fields	Cursor is in the Year Search box				
	2.4.1	Type in a year	N/A	Cursor must be in Student Name box	Empty Student Name box	Team Name box has input				
	2.5.0	Click the Search Button	N/A	Team Name Box must have input	Search Page	Results Page				
	2.6.0	Choose Correct Team from the List of Results	N/A	On the Results Page	Results Page	Team's Information Page				
	2.7.0	Click on the links to view different information about the team	N/A	Be on the Information Page	Information Page	Desired information is displayed				
F	= Unit Summary		0% passing		0 passed		Date of last test =			
	7	tests			0 failed					

5.9 Integration Test

Integration Testing is concerned with knowledge about validating that software components which have been unit tested separately. It checks that the units interact correctly when they are put together to perform a higher order function. This testing unit also includes knowledge about dependency checking for calls, data, and processes, and about interface checking in terms of range, type compatibility, representation, number and order of parameters, and method of transfer.

Unit Number	Integrated with these units
1	2
2	1

6 Appendices

6.1 Appendix A: 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.

Apache Server: Known as Apache, it is web server software notable for playing a key role in the initial growth of the World Wide Web.

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

Database Relational Schema: refers to the organization of data to create a blueprint of how a database will be constructed

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 an object-oriented scripting language that operates on the user's computer rather than on the hosting server.

Killer Robot Discussion: A page with links to access the Killer Robot papers, which are used to generate in-class discussion between the Software Engineering students.

MySQL: MySQL is 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 (25 Requirements Specification *Code Shark Solutions*).

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.

Primary Key: uniquely identifies each record in the table. Primary keys may consist of a single attribute or multiple attributes in combination.

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.

R.O.S.E.S.: Acronym for the project name: **Redesign of the Software Engineering Site**

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.

Speakers: A page consisting of a list of speakers that come to the Software Engineering class and talk to the students about certain topics.

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.

Unit Testing – a method by which individual units of source code are tested to determine if they are fit for use. A unit is the smallest testable part of an application.

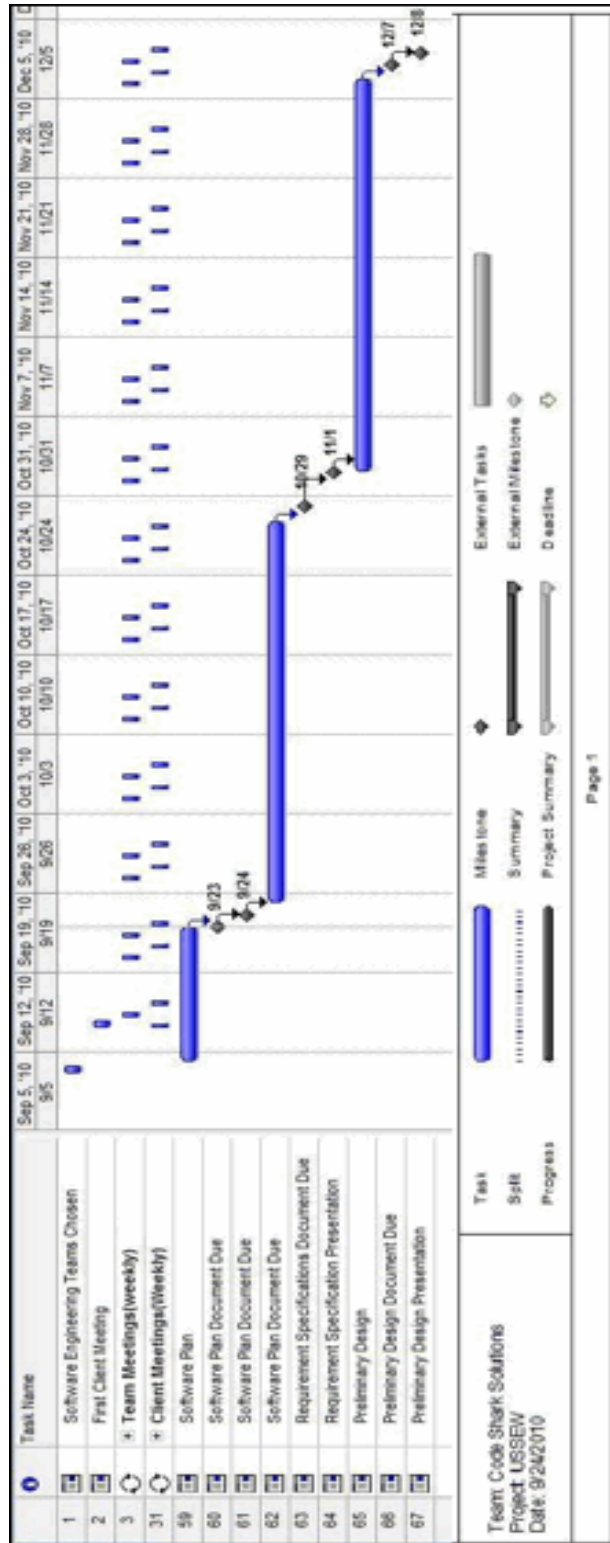
Use Case: A component of a UML Use Case Diagram which represents any process located within the System that is performable by an Actor.

Usage 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.

6.2 Appendix B: Timeline

6.2.1 Fall Semester



6.2.2 Spring Semester

