

Preliminary Design

R.O.S.E.S

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

December 8, 2010

Contents

1 Introduction	4
1.1 Product Overview	4
1.2 Development, Operation and Maintenance Environment	5
1.3 Usage Case Narratives	6
1.3.1 Course Instructor Usage Case Narrative	6
1.3.2 Software Engineering Student Usage Case Narrative	6
1.3.3 Recommenders Usage Case Narrative	6
1.3.4 Family/Friend Usage Case Narrative	7
1.3.5 Alumnus/Alumna Usage Case Narrative	7
1.4 Functional Requirements Inventory	7
1.4.1 The General User	8
1.5 Non-Functional Requirements Inventory	9
2 Preliminary Design	10
2.1 UML Use Case Diagram	10
2.1.1 UML Use Case Legend	10
2.1.2 Diagram	11
2.2 UML Deployment Diagram	12
2.2.1 Diagram	12
2.3 UML Activity Diagram	13
2.3.1 Activity Diagram Legend	13
2.3.2 Diagram	14
2.4 Website Map	15
2.4.1	15
2.4.2 Map	15
2.5 Data Flow Diagrams	16
2.5.1 Data Flow Diagram Legend	16
2.5.2 Context Diagram	17
2.5.3 Level 0 Diagram	18
2.5.4 Level 1 Diagrams	19
2.5.5 Level 2 Diagrams	20
2.6 Logical Data Dictionary	24
3 Prototype Screens	25
3.1 Home Page	25
3.2 Main Menu	26

3.3 Sub Menu	26
3.4 Footer Links	27
3.5 Software Engineering I – Course Description	27
3.6 Software Engineering I – Calendar	28
3.7 Software Engineering I – Grading.....	29
3.8 Software Engineering I – Syllabus	29
3.9 Software Engineering II – Course Description	30
3.10 Software Engineering II – Calendar	30
3.11 Software Engineering II – Grading.....	31
3.12 Software Engineering II – Syllabus	32
3.13 Software Engineering II – Speakers	33
3.14 Software Engineering II – Killer Robot.....	34
3.15 Search page	35
3.16 Results – Multiple Results	37
3.17 Results – Student Information.....	38
3.18 Results – Team Information	39
3.19 Results – Course Year Information	40
4 Test Plan	41
4.1 Overview and Strategy	41
4.2 Acceptance Test and Acceptance Criteria	41
4.3 Unit Tests.....	41
5 Appendices	42
5.1 Appendix A: Glossary.....	42
5.2 Appendix B: Timeline.....	44
5.2.1 Fall Semester	44
5.2.2 Spring Semester	45

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

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.

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

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.

1.4.1 The General User

The General User includes Students, Faculty, Family and Friends, Future Employers, Alumni and 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 student's information
- Will be able to view previous student's 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

1.5 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.

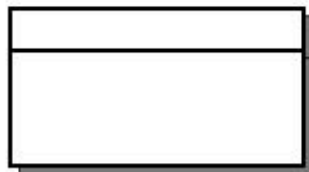
- The system must be aesthetically pleasing
- The system must be easily navigable
- The system must be easily maintainable
- The system must be easily modifiable
- The system must be stable

2 Preliminary Design

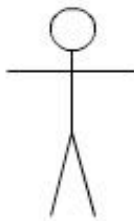
2.1 UML Use Case Diagram

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.

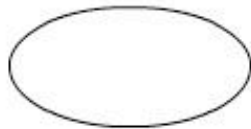
2.1.1 UML Use Case Legend



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.

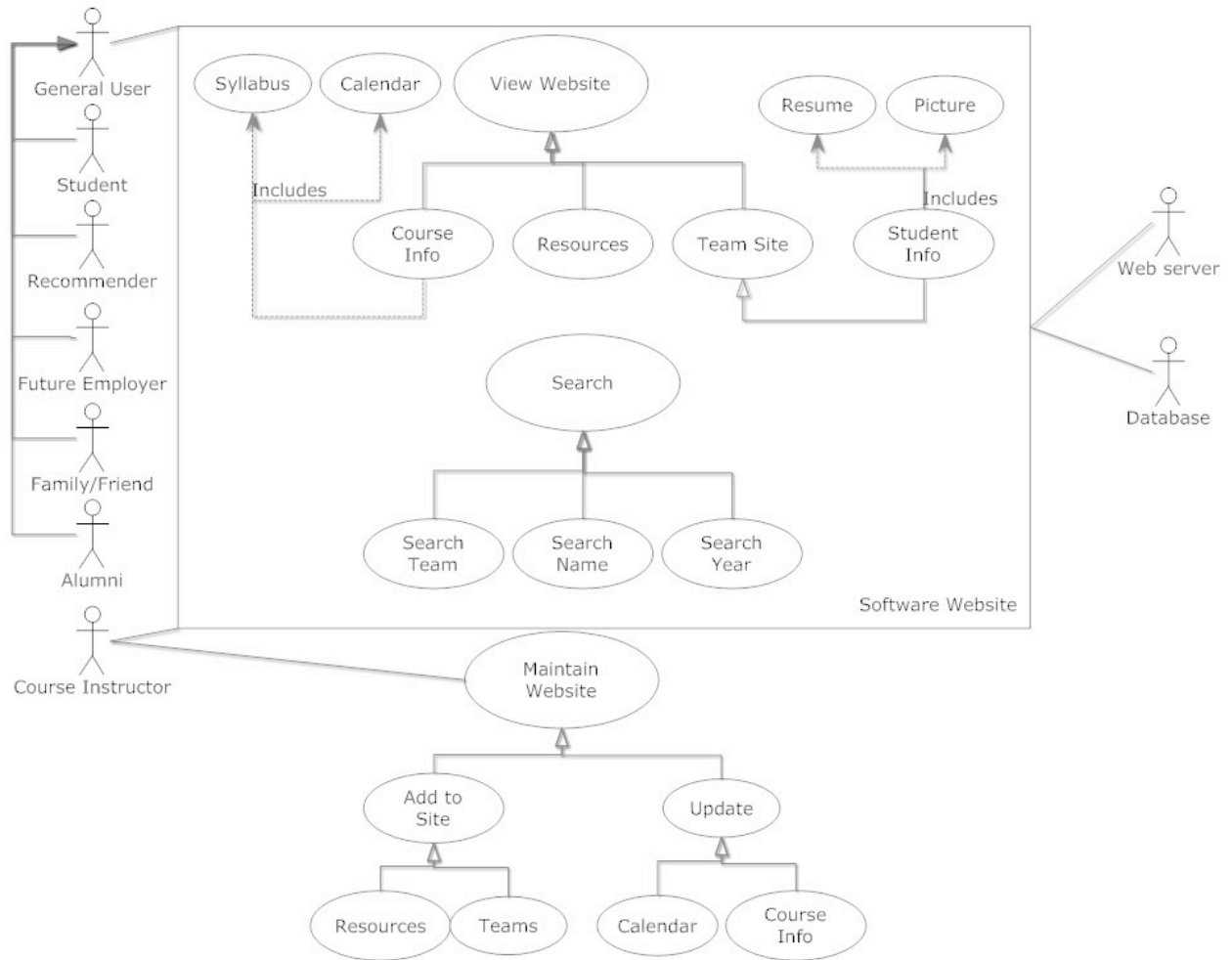


Participation Line- Lines that connect actors with uses showing their participation.



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

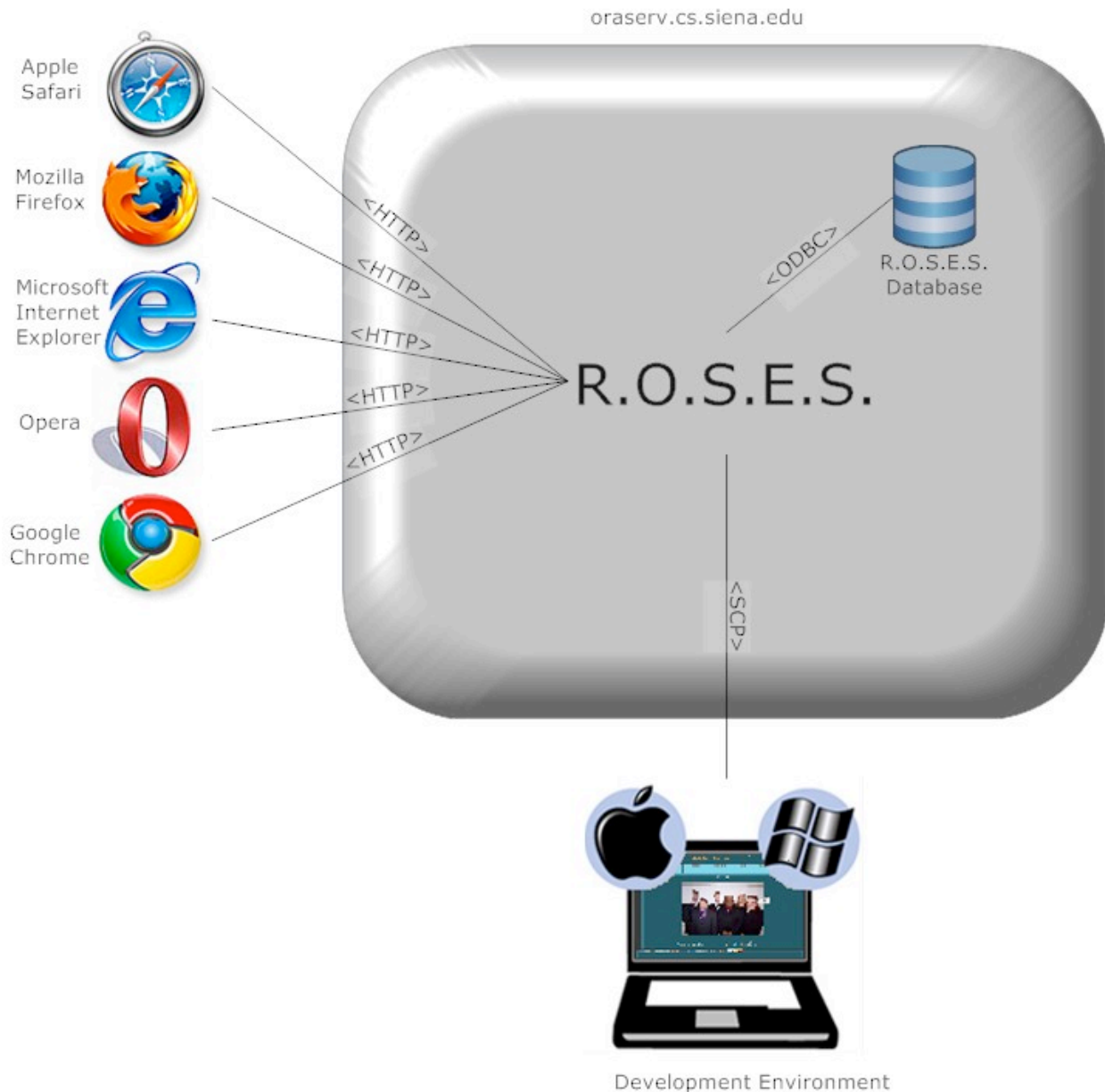
2.1.2 Diagram



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.

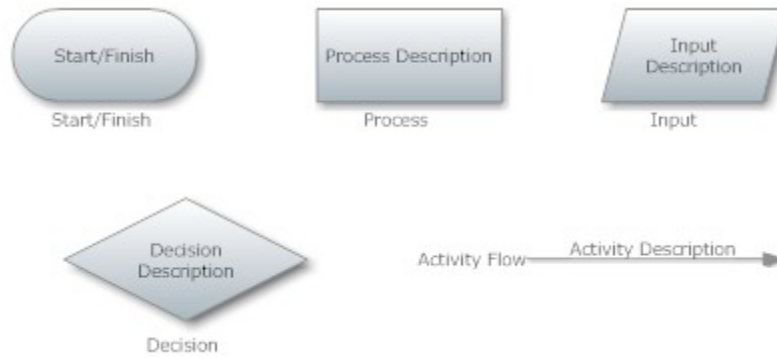
2.2.1 Diagram



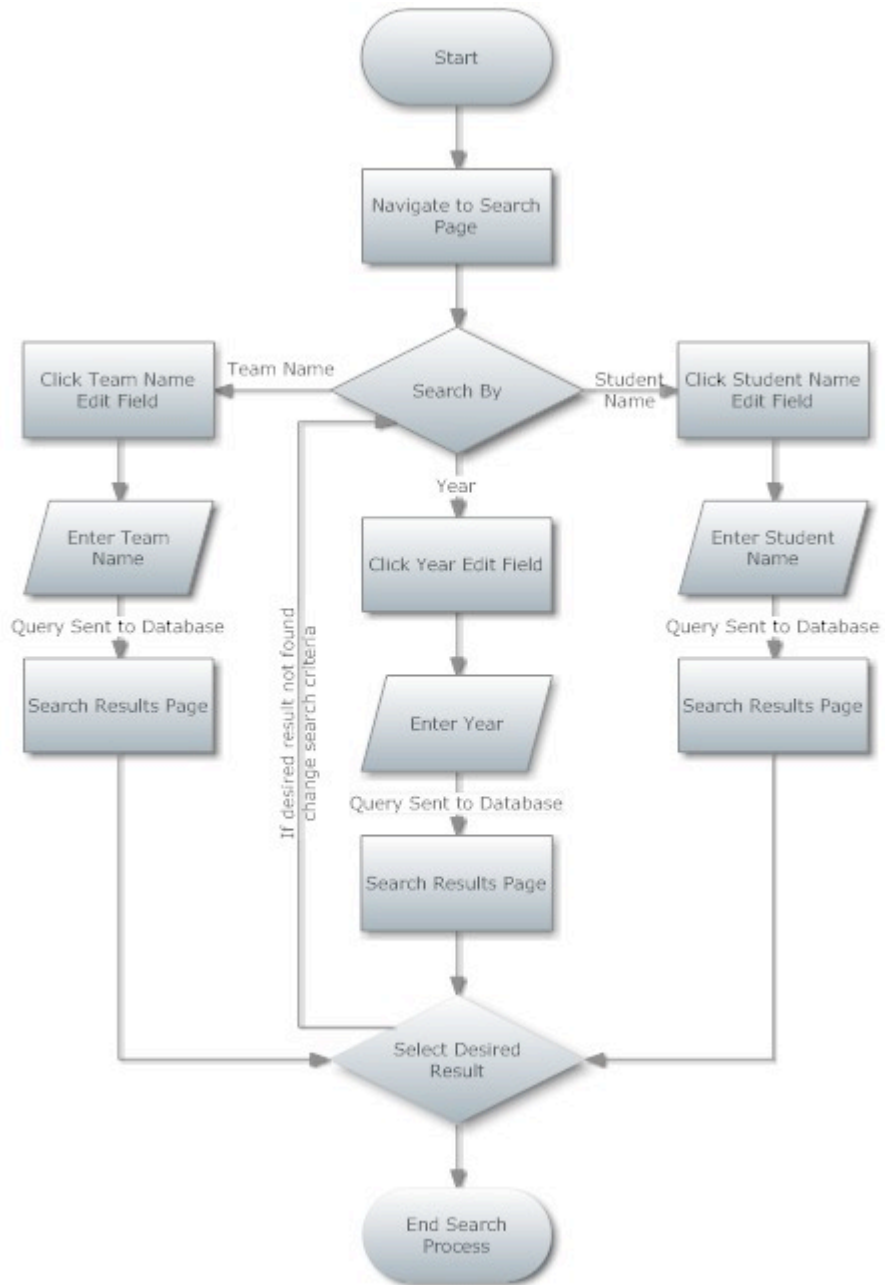
2.3 UML Activity Diagram

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



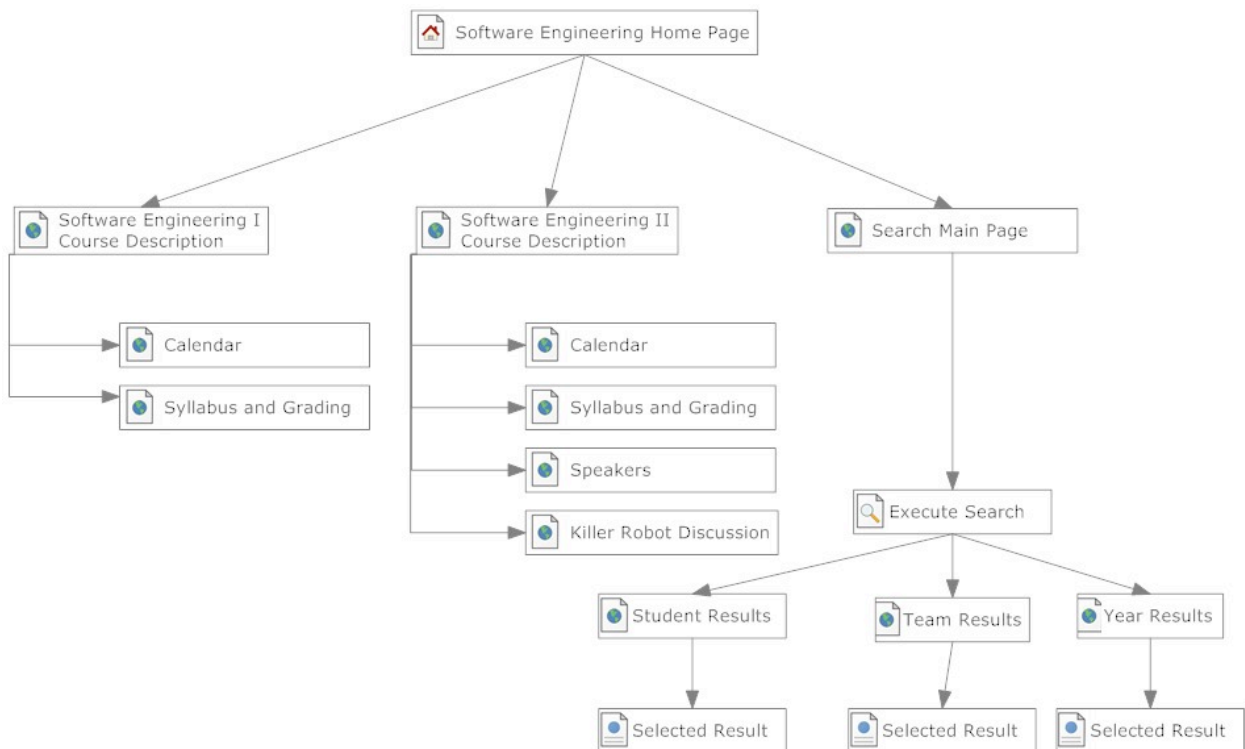
2.4 Website Map

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

2.4.1



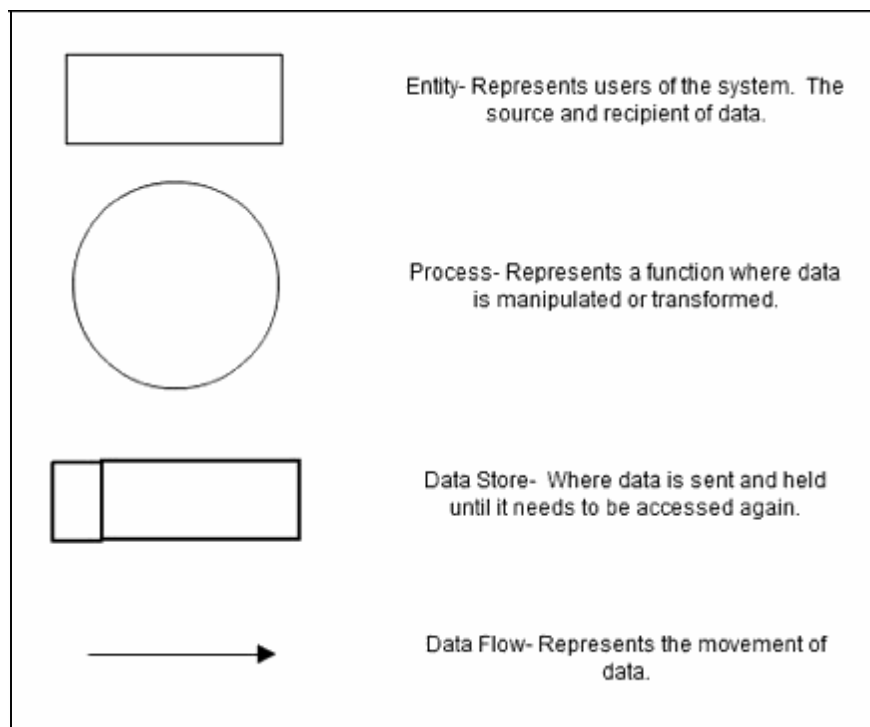
2.4.2 Map



2.5 Data Flow Diagrams

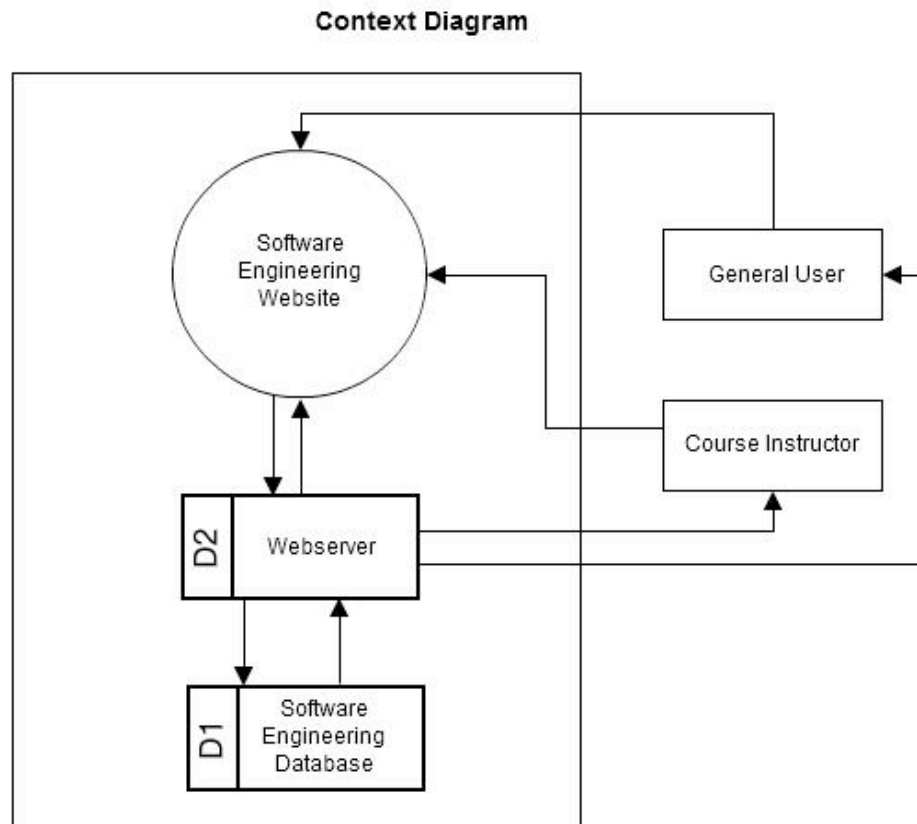
Data Flow Diagrams are created to show the movement of data throughout the system. They are used as a visual aid to help the reader to better understand how the system works. The context diagram is a general overview of the entire system. The level 0 diagram shows in more detail how each entity interacts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



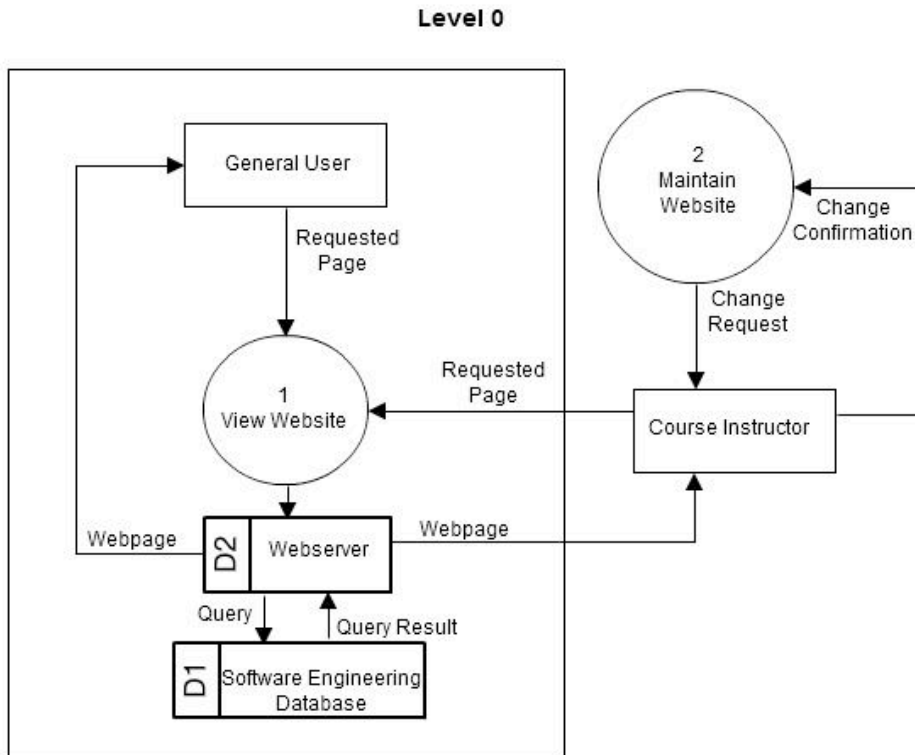
2.5.2 Context Diagram

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

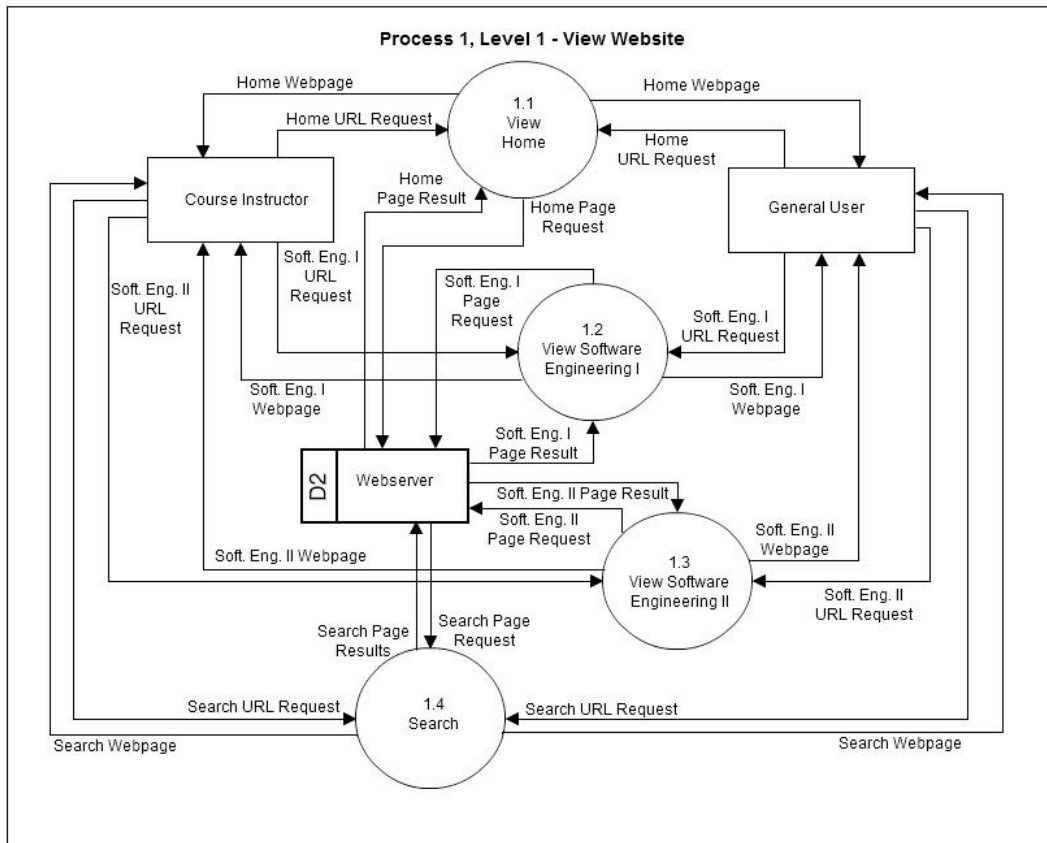
This diagram shows the major processes of the system.



2.5.4 Level 1 Diagrams

This Diagram shows the main use of the website.

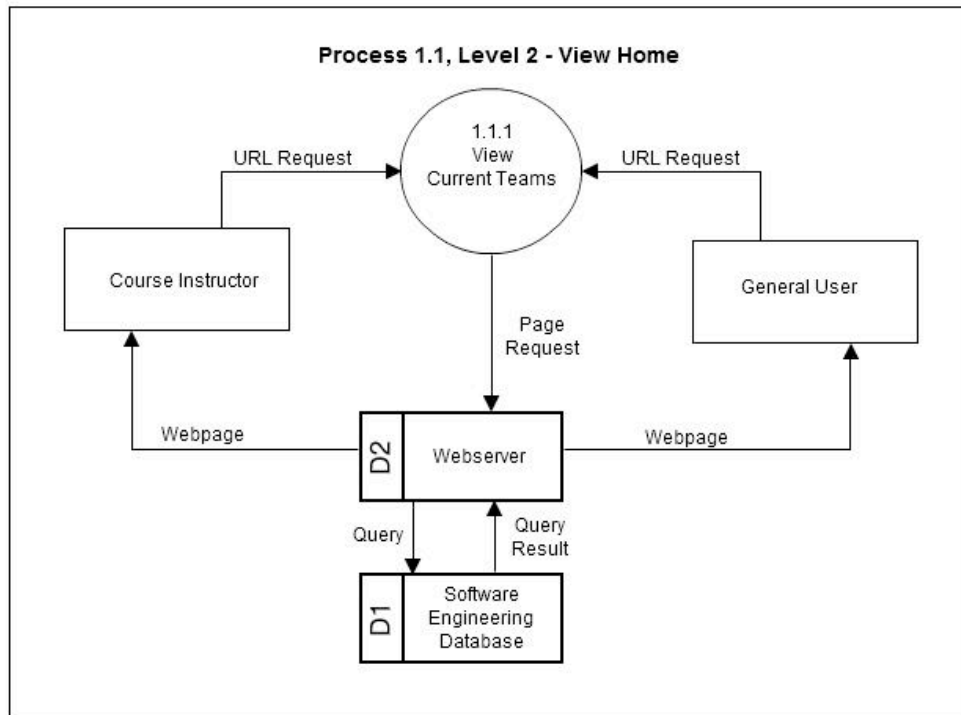
2.5.4.1 View Website



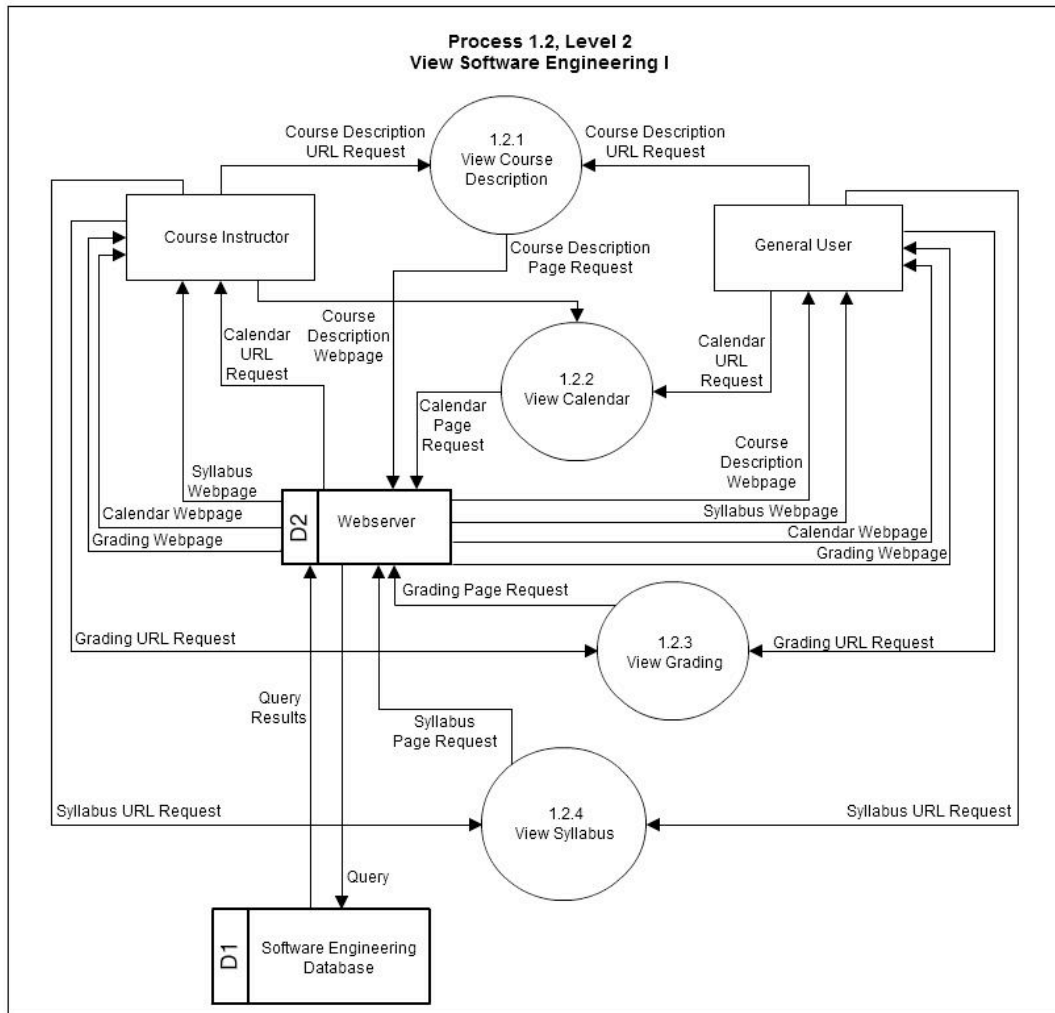
2.5.5 Level 2 Diagrams

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

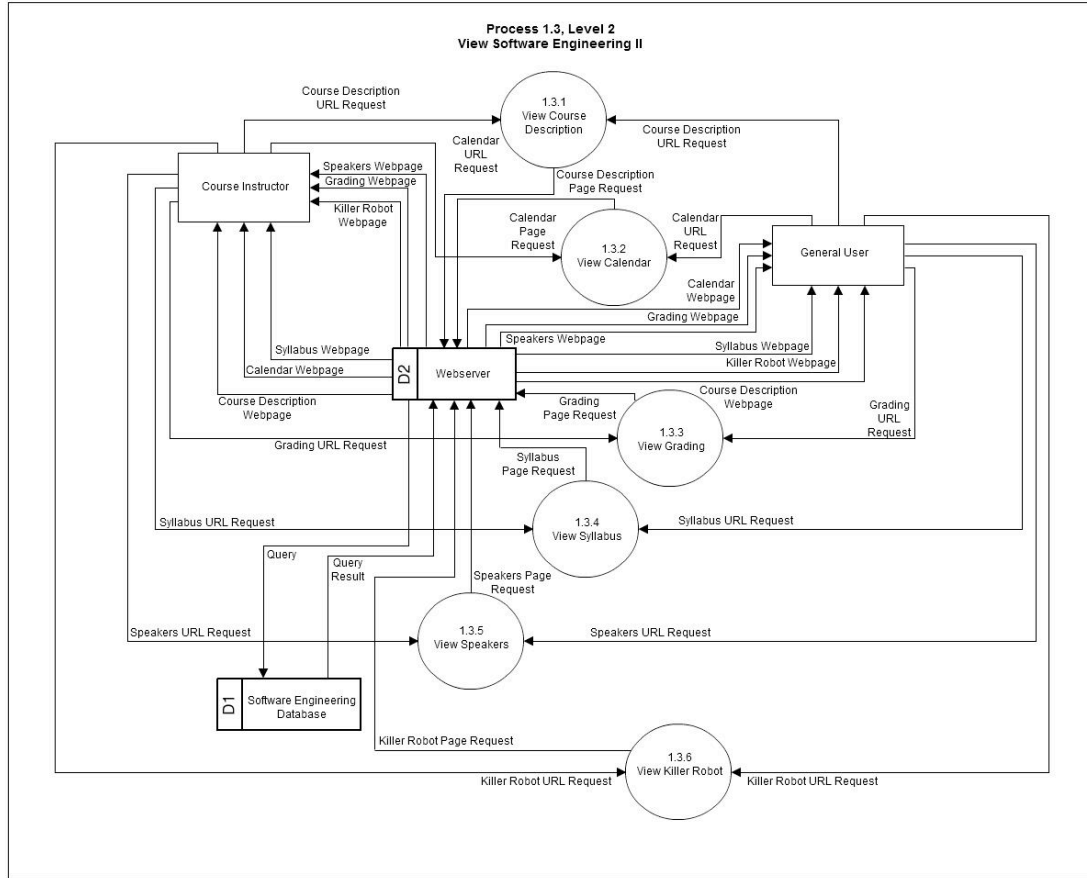
2.5.5.1 View Home



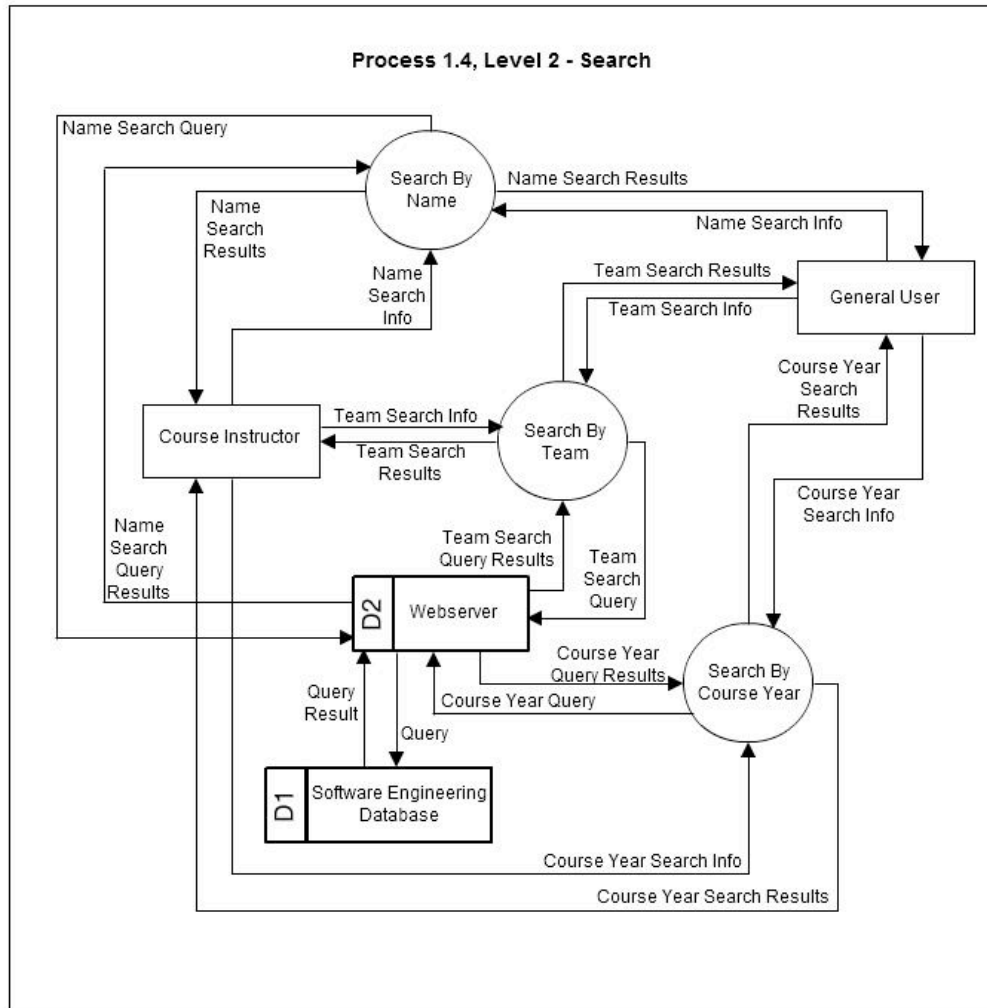
2.5.5.2 View Software Engineering I



2.5.5.3 View Software Engineering II



2.5.5.4 Search



2.6 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

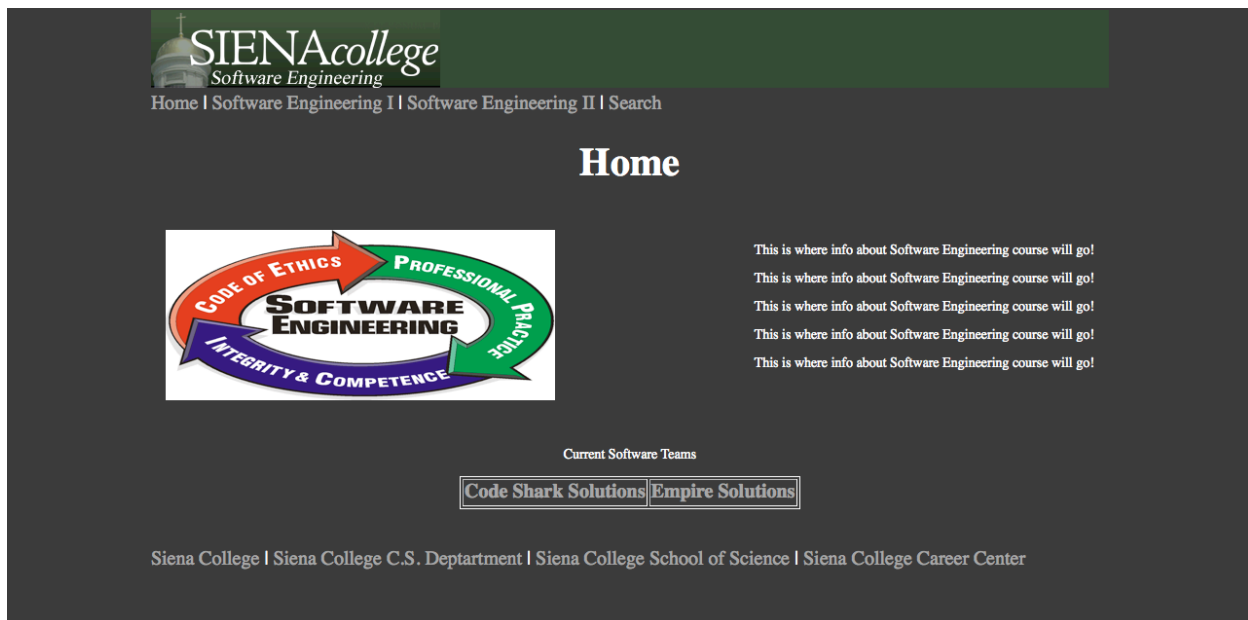
Notes- Any additional information that is useful in understanding the data entity

Please see next page for the Data Dictionary

3 Prototype Screens

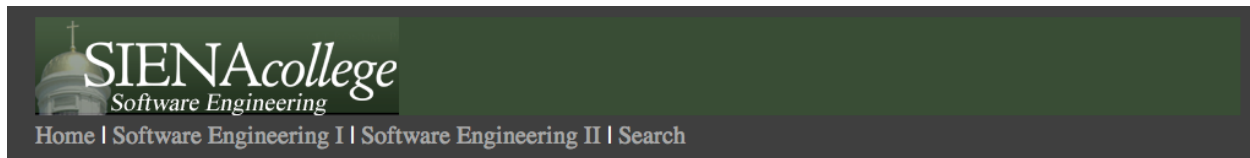
As we are getting closer to the actual development of the new website, we have designed a semi-functional prototype to help us and the client get a better understanding of what direction the project is heading in. Although the prototype is somewhat functional, all of the design aspects are subject to 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.

3.1 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.

3.2 Main Menu



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.

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](#)

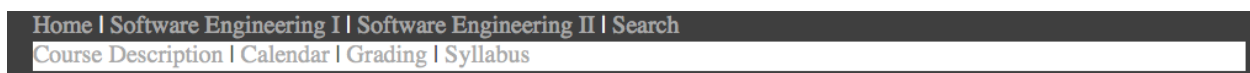
CSIS-410 & CSIS-415 Software Engineering I & II

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](#)

3.3 Sub Menu



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.

3.4 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.

3.5 Software Engineering I – Course Description



The screenshot shows a web page for Siena College. At the top left is the Siena College logo with the text "SIENAcollege Software Engineering". Below the logo is a navigation menu with links: "Home | Software Engineering I | Software Engineering II | Search | Course Description | Calendar | Grading | Syllabus". The main heading is "Software Engineering I - Course Description". Below this is the course title "CSIS-410. Software Engineering I (3 credits)" and a brief description: "Introduces the concepts of structured system analysis. Presents principles of software engineering including techniques for planning, specification, and system design. Specifications for an actual system will be developed. Prerequisite: CSIS-225. Recommended for all senior-level Computer Science majors". At the bottom of the page is the footer: "Siena College | Siena College C.S. Department | Siena College School of Science | Siena College Career Center".

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.

3.6 Software Engineering I – Calendar

COURSE CALENDAR FOR THE FALL 2010 SEMESTER

Month: Day:

EXAMPLE DATE	<i>EXAMPLE LECTURE:</i> <i>In-Class Movie:</i>	Syllabus, course schedule, speakers, discussions "The Forbidden Planet" (1956, 98 minutes). See the original <i>Robby the Robot</i> . Shown in class.
EXAMPLE DATE	<i>EXAMPLE LECTURE:</i> <i>In-Class Movie:</i>	Syllabus, course schedule, speakers, discussions "The Forbidden Planet" (1956, 98 minutes). See the original <i>Robby the Robot</i> . Shown in class.
EXAMPLE DATE	<i>EXAMPLE LECTURE:</i> <i>In-Class Movie:</i>	Syllabus, course schedule, speakers, discussions "The Forbidden Planet" (1956, 98 minutes). See the original <i>Robby the Robot</i> . Shown in class.
EXAMPLE DATE	<i>EXAMPLE LECTURE:</i> <i>In-Class Movie:</i>	Syllabus, course schedule, speakers, discussions "The Forbidden Planet" (1956, 98 minutes). See the original <i>Robby the Robot</i> . Shown in class.
EXAMPLE DATE	<i>EXAMPLE LECTURE:</i> <i>In-Class Movie:</i>	Syllabus, course schedule, speakers, discussions "The Forbidden Planet" (1956, 98 minutes). See the original <i>Robby the Robot</i> . Shown in class.
EXAMPLE DATE	<i>EXAMPLE LECTURE:</i> <i>In-Class Movie:</i>	Syllabus, course schedule, speakers, discussions "The Forbidden Planet" (1956, 98 minutes). See the original <i>Robby the Robot</i> . Shown in class.

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. 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

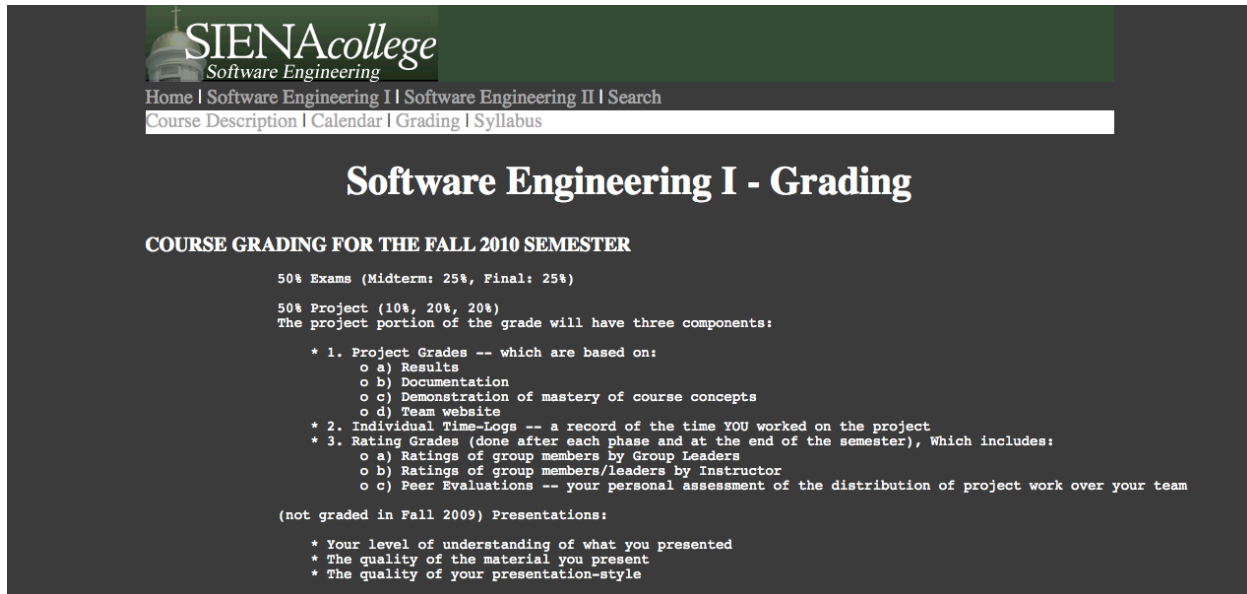


Course Description

Introduces the concepts of structured system analysis. Presents principles of software engineering including techniques for planning, specification, and system design. Specifications for an actual system will be developed. Prerequisite: CSIS-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) email/voicemail etiquette How to shake hands and make introductions Teams/Groups formed by team leaders & Dr. Lederman (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 Outlines 	<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

3.7 Software Engineering I – Grading



SIENAcollege
Software Engineering

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

Software Engineering I - Grading

COURSE GRADING FOR THE FALL 2010 SEMESTER

50% Exams (Midterm: 25%, Final: 25%)

50% Project (10%, 20%, 20%)
The project portion of the grade will have three components:

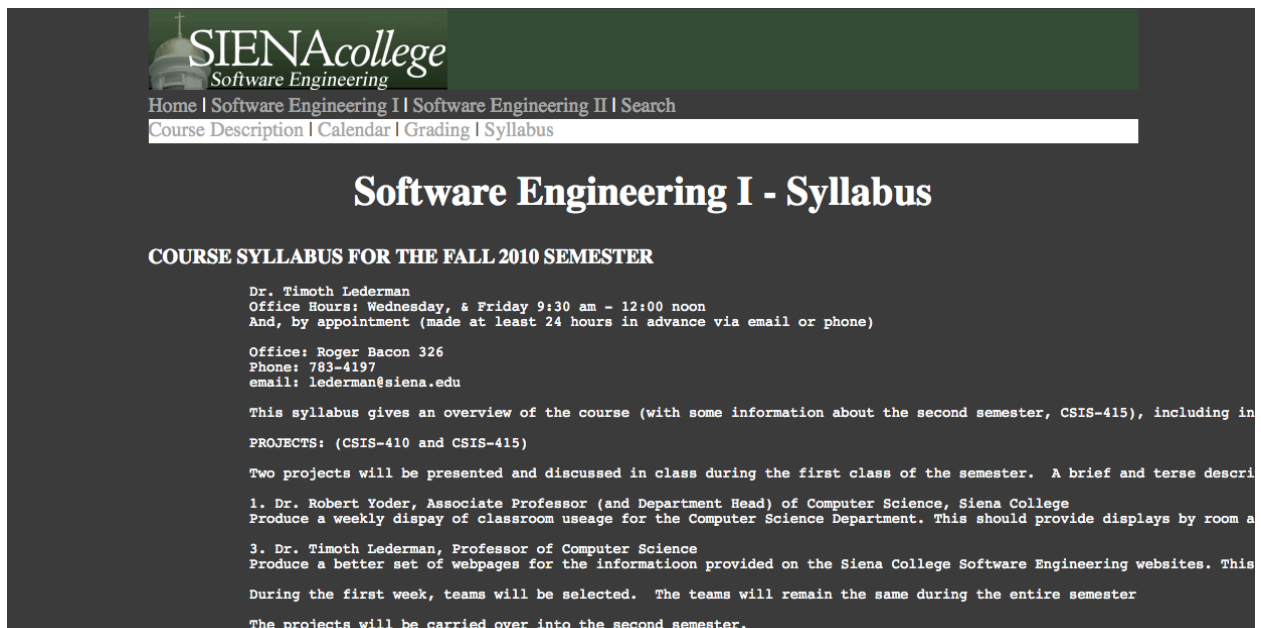
- * 1. Project Grades -- which are based on:
 - o a) Results
 - o b) Documentation
 - o c) Demonstration of mastery of course concepts
 - o d) Team website
- * 2. Individual Time-Logs -- a record of the time YOU worked on the project
- * 3. Rating Grades (done after each phase and at the end of the semester), Which includes:
 - o a) Ratings of group members by Group Leaders
 - o b) Ratings of group members/leaders by Instructor
 - o c) Peer Evaluations -- your personal assessment of the distribution of project work over your team

(not graded in Fall 2009) Presentations:

- * Your level of understanding of what you presented
- * The quality of the material you present
- * The quality of your presentation-style

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

3.8 Software Engineering I – Syllabus



SIENAcollege
Software Engineering

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

Software Engineering I - Syllabus

COURSE SYLLABUS FOR THE FALL 2010 SEMESTER

Dr. Timoth Lederman
Office Hours: Wednesday, & Friday 9:30 am - 12:00 noon
And, by appointment (made at least 24 hours in advance via email or phone)

Office: Roger Bacon 326
Phone: 783-4197
email: lederman@siena.edu

This syllabus gives an overview of the course (with some information about the second semester, CSIS-415), including in

PROJECTS: (CSIS-410 and CSIS-415)

Two projects will be presented and discussed in class during the first class of the semester. A brief and terse descri

1. Dr. Robert Yoder, Associate Professor (and Department Head) of Computer Science, Siena College
Produce a weekly display of classroom usage for the Computer Science Department. This should provide displays by room a

3. Dr. Timoth Lederman, Professor of Computer Science
Produce a better set of webpages for the information provided on the Siena College Software Engineering websites. This

During the first week, teams will be selected. The teams will remain the same during the entire semester

The projects will be carried over into the second semester.

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

3.9 Software Engineering II - Course Description

The screenshot shows the SIENA College website for the Software Engineering II course. The header includes the 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 - Course Description". Below it, the course is identified as "CSIS-415. Software Engineering II (3 credits)". A brief description follows: "Implementation of a software engineering methodology. Complete design, testing, and verification of a system developed using a programming team concept. Structured design and documentation. Prerequisite: CSIS-410". At the bottom, there are links to Siena College, the C.S. Department, the School of Science, and the Career Center.

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.

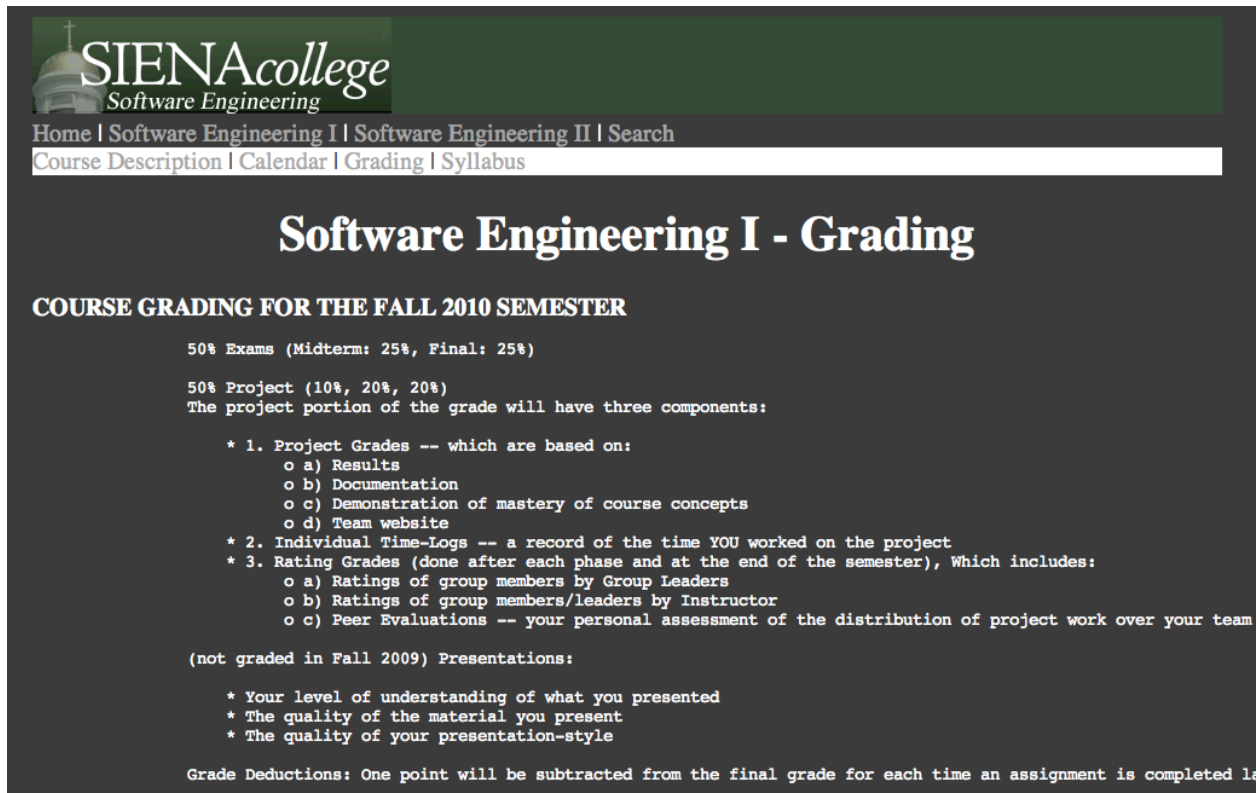
3.10 Software Engineering II - Calendar

The screenshot shows the "Software Engineering II - Calendar" page. It features a "CALENDAR OF EVENTS FOR THE SPRING 2010 SEMESTER" with a navigation bar for "Month: January" and "Day: 01", and a "Go to date" button. The calendar table lists events for Tuesday, January 19, Tuesday, January 26, and Tuesday, February 2.

Day	Date	Event
Tuesday	January 19	<p>Lecture: Syllabus, course schedule, speakers, discussions</p> <p>In-Class Movie: "The Forbidden Planet" (1956, 98 minutes). See the original <i>Robby the Robot</i>. Shown in class.</p>
Tuesday	January 26	<p>Presentation: Ken Swarner and Eric Crossman (Siena, 2003) Systems & Operations Managers, Siena College, Computer Science Department and School of Science "PHP For Fun & Profit"</p> <p>Lecture: Syllabus, course schedule, speakers, discussions, and NYS Department of Civil Service Cover letters, thank-you letters, graduate school references</p> <p>In-Class Movie: "The Case of the Killer Robot": Introduction by Dr. Lederman</p>
Tuesday	February 2	<p>Presentation: John Sawicki (Siena, 2005) Software Developer, Grid Operation Products, New York, Independent System Operator</p>

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.

3.11 Software Engineering II – Grading



The screenshot shows a web page for SIENA College Software Engineering. The header includes the college logo and navigation links: Home | Software Engineering I | Software Engineering II | Search | Course Description | Calendar | Grading | Syllabus. The main heading is "Software Engineering I - Grading". Below this, it states "COURSE GRADING FOR THE FALL 2010 SEMESTER". The grading breakdown is as follows:

- 50% Exams (Midterm: 25%, Final: 25%)
- 50% Project (10%, 20%, 20%)

The project portion of the grade will have three components:

1. Project Grades -- which are based on:
 - a) Results
 - b) Documentation
 - c) Demonstration of mastery of course concepts
 - d) Team website
2. Individual Time-Logs -- a record of the time YOU worked on the project
3. Rating Grades (done after each phase and at the end of the semester), Which includes:
 - a) Ratings of group members by Group Leaders
 - b) Ratings of group members/leaders by Instructor
 - c) Peer Evaluations -- your personal assessment of the distribution of project work over your team

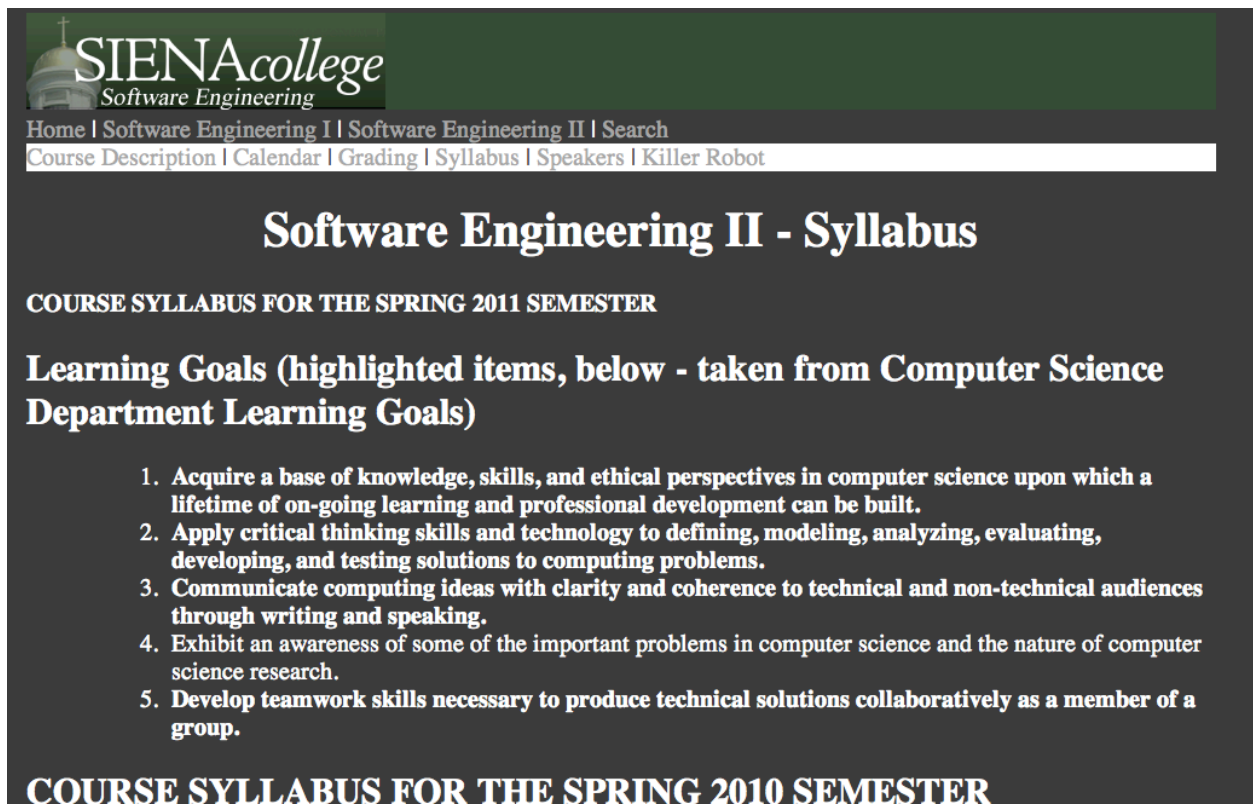
(not graded in Fall 2009) Presentations:

- * Your level of understanding of what you presented
- * The quality of the material you present
- * The quality of your presentation-style

Grade Deductions: One point will be subtracted from the final grade for each time an assignment is completed late.

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.

3.12 Software Engineering II – Syllabus



The screenshot shows the top portion of a web page for SIENA College's Software Engineering II syllabus. The header includes the college logo and navigation links. The main title is 'Software Engineering II - Syllabus'. Below it, the text reads 'COURSE SYLLABUS FOR THE SPRING 2011 SEMESTER'. The section is titled 'Learning Goals (highlighted items, below - taken from Computer Science Department Learning Goals)'. A list of five goals is provided, with the first three items highlighted in yellow in the original image. The bottom of the screenshot shows the text 'COURSE SYLLABUS FOR THE SPRING 2010 SEMESTER'.

SIENA College
Software Engineering

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

Software Engineering II - Syllabus

COURSE SYLLABUS FOR THE SPRING 2011 SEMESTER

Learning Goals (highlighted items, below - taken from Computer Science Department Learning Goals)

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.**

COURSE SYLLABUS FOR THE SPRING 2010 SEMESTER

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.

3.13 Software Engineering II – Speakers

SPEAKERS FOR THE SPRING 2010 SEMESTER

Full page layout dimensions

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 January 26	Ken Swarner and Eric Crossman (Siena, 2003) Systems & Operations Managers, Siena College, Computer Science Department and School of Science <i>"PHP For Fun & Profit"</i>
Tuesday February 2	John Sawicki (Siena, 2005) Software Developer, Grid Operation Products, New York Independent System Operator <i>"Real-World Application Development: Overview and Usage of Third Party Frameworks"</i>
Tuesday February 9	James Manico (Siena, 1997) Web Application Architect, The Manicode Group <i>"Stopping Cross Site Scripting with Contextual Output Encoding"</i>

This page is dedicated to all the speakers for the semester. The design of this page is still under discussion.

3.14 Software Engineering II – Killer Robot

Questions for Killer Robot Discussions

Tuesday January 26	<ul style="list-style-type: none"> • Introduction, • Cast of Characters, • Words, Terms, & Allusions 	0. Dr. Lederman
Tuesday February 2	<ol style="list-style-type: none"> 1. Silicon Valley programmer indicted for manslaughter 2. Developers of "Killer Robot" worked under enormous stress 	1. Digital Evolutions 2. Infinite Solutions
Tuesday February 9	<ol style="list-style-type: none"> 3. "Killer Robot" programmer was Prima Donna, co-workers claim 4. "Killer Robot" project mired in controversy right from start 	3. 518 Interactive 4. Digital Evolutions
Tuesday February 16	<ol style="list-style-type: none"> 5. The "Killer Robot" interface 	5. Infinite Solutions

This page is where all the dates of when speakers are coming will be. The design is still under discussion.

3.15 Search page

SIENAcollege
Software Engineering

Home | Software Engineering I | Software Engineering II | Search

Search

Search by Student Name: [Search by Student Name](#)

Search by Team Name: [Search by Team Name](#)

Search by Course Year: [Search by Course Year](#)

To go through all the Software Engineering teams or course years click below:

[All Team Names](#) [All Course Years](#)

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

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

All course years page:

SIENAcollege
Software Engineering

Home | Software Engineering I | Software Engineering II | Search

All Software Engineering Course Years

(for which we have records)

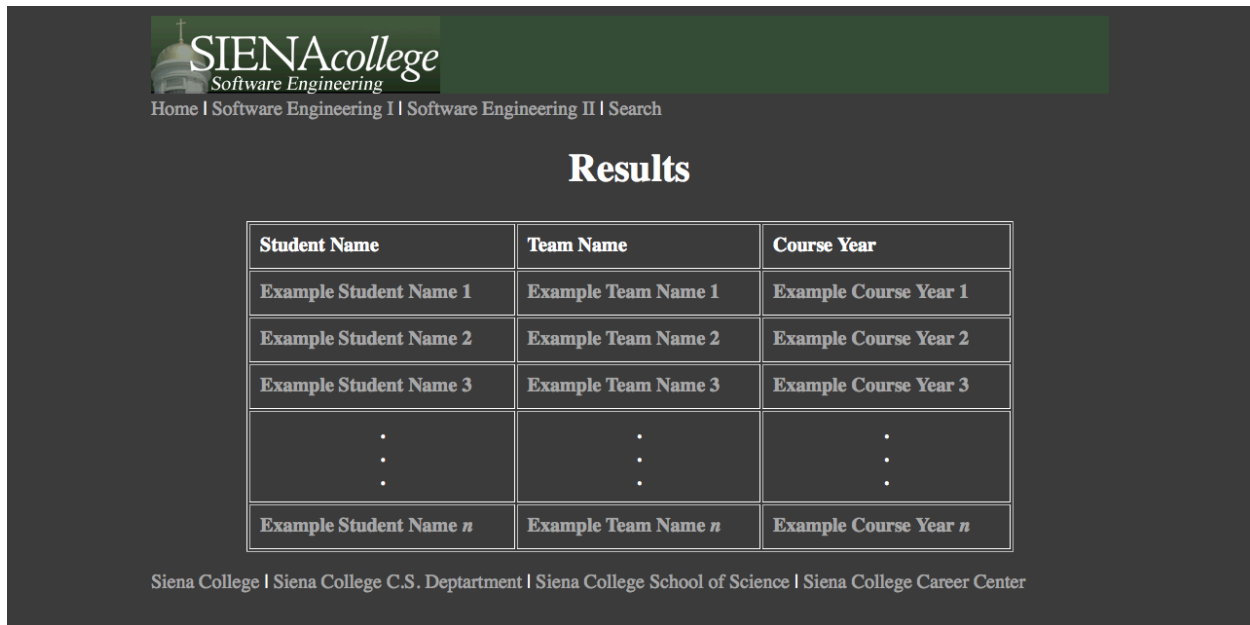
1984-1985	1989-1990	1994-1995	1999-2000	2004-2005	2009-2010
1985-1986	1990-1991	1995-1996	2000-2001	2005-2006	2010-2011
1986-1987	1991-1992	1996-1997	2001-2002	2006-2007	
1987-1988	1992-1993	1997-1998	2002-2003	2007-2008	
1988-1989	1993-1994	1998-1999	2003-2004	2008-2009	

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

All teams page:



3.16 Results – Multiple Results



SIENAcollege
Software Engineering
Home | Software Engineering I | Software Engineering II | Search

Results

Student Name	Team Name	Course Year
Example Student Name 1	Example Team Name 1	Example Course Year 1
Example Student Name 2	Example Team Name 2	Example Course Year 2
Example Student Name 3	Example Team Name 3	Example Course Year 3
⋮	⋮	⋮
Example Student Name n	Example Team Name n	Example Course Year n

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

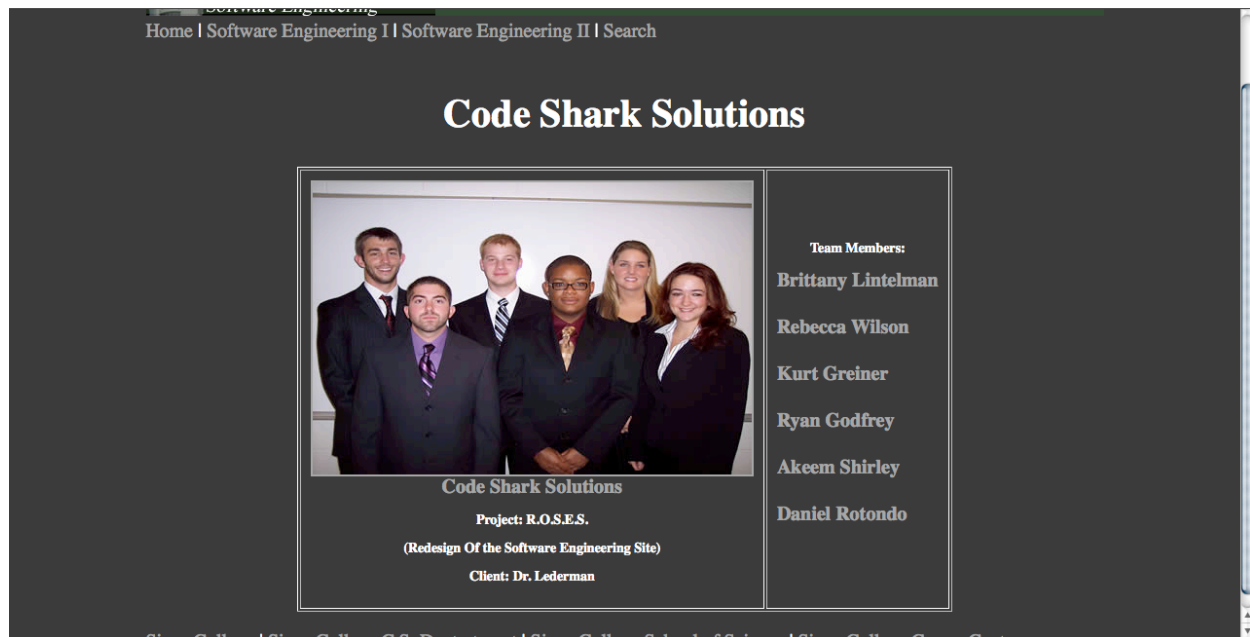
This page is where the user will be directed after attempting a search. The user will be given a table of students, teams, and course years that are close to what the user inputted. They will then be able to click on the results in the table to get more information on the selected link.

3.17 Results – Student Information

The screenshot displays a web page with a dark background. At the top, there is a navigation bar with the text "Home | Software Engineering I | Software Engineering II | Search". The main content is divided into two columns. The left column features the name "Brittany Lintelman" in bold, a portrait photo of a young woman with blonde hair, and a link labeled "Senior-Year Resume". The right column features the text "Team Name: Code Shark Solutions" above a group photo of six students in business attire. Below the photo, there are links for "Project: R.O.S.E.S. (Redesign Of the Software Engineering Site)", "Team Song", and "Project Demo". At the bottom of the page, a footer contains the text "Siena College | Siena College C.S. Department | Siena College School of Science | Siena College Career Center".

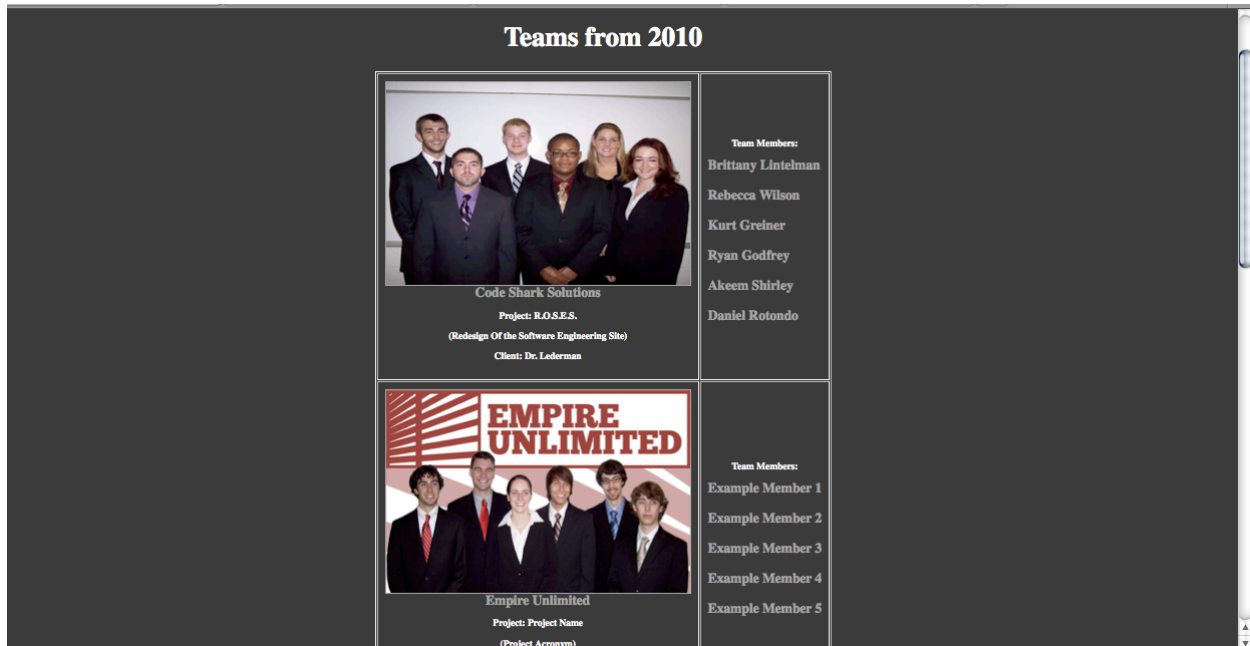
If the user selects a student name, the page above will appear. The information on this page will include the students first and last name, a link to their senior-year resume, their software engineering team's name, team picture, team project name, the project name's acronym, link to team website, and if they have it, links to their team song and project demo.

3.18 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 team name, link to team page, project name, project acronym, client name, and link to the team members resumes will all be included.

3.19 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 Test Plan

4.1 Overview and Strategy

A Test Plan documents the strategy that will be used to verify and ensure that a product meets its design specifications, as well as Functional and Non-functional requirements. 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.

4.2 Acceptance Test and Acceptance Criteria

Acceptance Testing is concerned with knowledge about validating the functional and non-functional requirements of a purchased or acquired system. This unit includes knowledge about techniques for using the contract, the statement of work, the software requirements specification, and the request for proposal to ensure that the delivered system meets all of the requirements (as perceived by the purchasing or acquiring organization).

The Acceptance Criteria will be determined by Code Shark Solutions. The project will be broken down into units which will be tested individually. The units will be tested thoroughly until they all pass individual unit testing. Once all the units pass testing they will be brought back together and will be tested as one system. In this document Code Shark Solutions has broken R.O.S.E.S. down into 2 different units.

4.3 Unit Tests

Please see next page for unit tests.

5 Appendices

5.1 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.

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. 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.

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.

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.

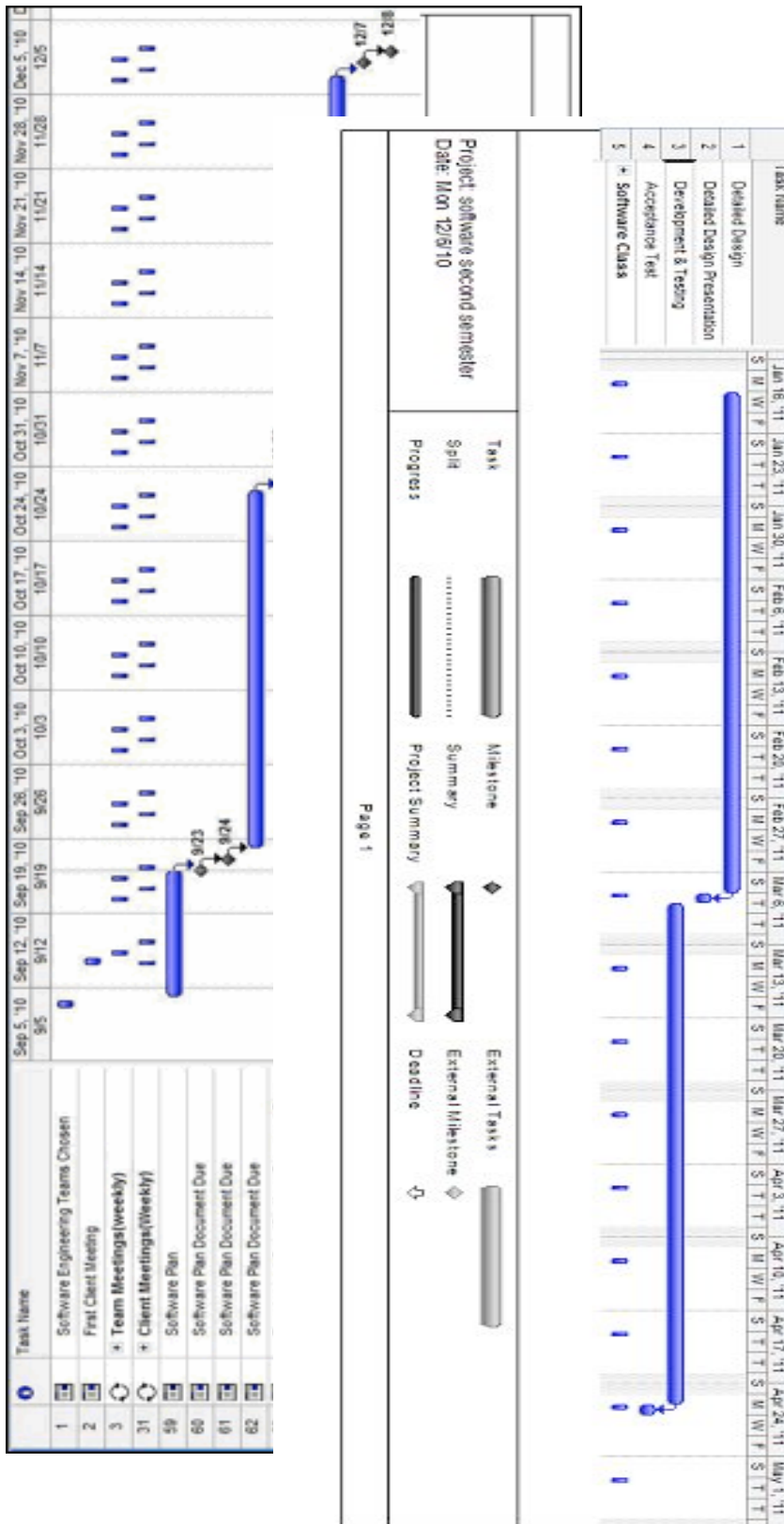
Use Case: A component of a UML Use Case Diagram which represents any process located within the System that is performable by an Actor.

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.

5.2 Appendix B: Timeline

5.2.1 Fall Semester



5.2.2 Spring Semester

