

*SAPIENT PROJECT*  
**FUSION FOR THE FUTURE**

**REQUIREMENTS DOCUMENT**

# ***Contents***

## **1. Introduction**

### **1.1 General Information**

### **1.2 Overview**

### **1.3 Customers**

### **1.4 Goals of the Project**

### **1.5 Major Constraints**

### **1.6 Method**

### **1.7 Process**

## **2. System Functions and System Attributes**

### **2.1 System Functions**

### **2.2 System Attributes**

## **3. Use Cases**

## **4. Data Flow Diagrams**

### **4.1 Level 0 DFD**

### **4.2 Level 1 DFD**

## **5. Preliminary Users Guide**

### **5.1 Screen Shots**

### **5.2 Screen Navigation Diagram**

## **6. Project Planning (Part 1)**

### **6.1 Work Breakdown Schedule**

### **6.2 Work Input Summary**

## **7. Appendix**

### **7.1 Data Dictionary**

### **7.2 Correspondence with Sapient**

# **1. Introduction**

## **1.1 General Information**

The group is composed of six members: Denis Bourdon, Christine Lau, Leona Qi, Emily Tramel, Fredrick Wass, Drew Winther. All of the members of the group are current undergraduate students studying at UCLA. This project is run through an undergraduate course at UCLA called "Software Engineering" or CS130.

## **1.2 Overview**

Project Alpha is a World Wide Web based system that allows members of a project team to communicate more efficiently regardless of their physical location. This system will aid our customers in their efforts to maximize the productivity of their team members operating out of offices located throughout the world.

## **1.3 Customers**

Sapient is a business and technology consulting company that offers their clients technological solutions that reduce costs, improve business performance, and enhance the customer experience. Sapient focuses on delivering solutions that result in explicit, highly valuable business outcomes. By making use of their globally distributed delivery system, Sapient provides rapid, reliable results on a fixed-time, fixed-price basis. To maximize their effectiveness, Sapient project team members operate in a highly collaborative culture where efficient communication is critical.

## **1.4 Goals of the Project**

The primary goal of the project is to improve communication efficiency between

Sapient project team members located throughout the world by designing and implementing an Internet-based messaging system that facilitates both real-time and time-delayed communications. The system should be easy to use and learn, and it should have a logical, efficient interface. Additionally, access to the system must be restricted to Sapient team members, and organized by project team.

## **1.5 Major Constraints**

The two most critical constraints on the project are the time allocated to finish the project and the number of team members assigned. The project due date is set for the week of December 3, and will probably not be changed. Also, we have six members working on our team, with no foreseeable way of acquiring more. These two factors greatly limit the scope of additional features we can add beyond the core functionality. Finally, the system we are designing has certain restrictions regarding the environment it will operate in. The system must run on Microsoft Internet Information Server version 4.0 and use Active Server Pages. System data should be stored in a relational database compatible with Microsoft Access. Users accessing the system using Microsoft Internet Explorer 5.0 and above should have no compatibility issues (i.e. the system must support Internet Explorer browsers 5.0 and above). These technical constraints limit the tools and technologies we will use to design and implement the system.

## **1.6 Method**

We will use the unified modeling language (UML) designed by persons at the Rational Software Corporation. This is a form of object oriented modeling.

## **1.7 Process**

For our team organization we will use the democratic decentralized process. This is mainly because our group has only 6 members each with different abilities. The feature of this process is that there will be no permanent leader, our group decisions will be due to a consensus, and we'll have horizontal communication between our team members. Since there are only 6 persons each person will have a task in both the technical and documentation side of the engineering software. This group will also use the incremental model to analyze, design, code, and test the software. This is mainly due to our 10-week time constraint. Also, this enables us to split up the project into separate components, which can be built and designed independently.

## 2. System Functions and System Attributes

### 2.1 System Functions

#### 2.1.1 Login

<u>Ref #</u>	<u>Function</u>	<u>Category</u>
R1.1	Prompts user for their name and password	Evident
R1.2	Authenticates users provided information with that in the database i.e. checks login name and password.	Hidden
R1.3	Checks users status to see if they are a first time user.	Hidden
R1.4	Prompts user to enter a new permanent password to override their default password.	Evident
R1.5	Checks to see if the created password matches the value in the "confirm password" field.	Hidden
R1.6	If created password does NOT match the value of the "confirm password" field, then prompts users to enter their new password again.	Evident
R1.7	Updates database with users new login status and new password.	Hidden

#### 2.1.2 Administrator Access

<u>Ref #</u>	<u>Function</u>	<u>Category</u>
R2.1	System prompts user whether they are an administrator.	Evident
R2.2	Database verifies user's administrative status.	Hidden
R2.3	If user is granted administrative access then system displays all appropriate operation links.	Evident

### 2.1.3 User Access

<u>Ref #</u>	<u>Function</u>	<u>Category</u>
R3.1	Display all operation links.	Evident
R3.2	Allow user to update their personal information.	Evident
R3.3	Update the personal information in database with changes provided by the user.	Hidden
R3.4	Display's all of the user's project titles.	Evident

### 2.1.4 System Administrator Access

<u>Ref #</u>	<u>Function</u>	<u>Category</u>
R4.1	Display all of Sapient current projects.	Evident
R4.2	Display all team members of all the projects.	Evident
R4.3	Allow system administrator to update all the projects information and personal information in the database.	Evident
R4.4	Update system information.	Hidden

### 2.1.5 Project Access

<u>Ref #</u>	<u>Function</u>	<u>Category</u>
R5.1	Display list of projects available to users.	Evident
R5.2	Allow user to check on one of their projects to display the project's details.	Evident
R5.3	Within each project's page display personal information for all group members.	Evident

### 2.1.5 Message Board Access Function

<u>Ref #</u>	<u>Function</u>	<u>Category</u>
R5.1	Display message board message's title.	Evident
R5.2	Display option for new message topic.	Evident
R5.3	If a particular message is chosen then sever links to another page to display the content of the message.	Hidden
R5.4	Display content of the message and options for reply.	Evident
R5.5	If reply option is chosen, remain in same page and reply box displays on the bottom of the page.	Hidden
R5.6	Allow user to input reply to the message and submit to all the users in the newsgroup.	Evident
R5.7	Update the reply message in the database, and new sub-title appears on message board.	Hidden
R5.8	If new post option is chosen, then serer links to a new page.	Hidden
R5.9	Allow user to write new message and submit to all the users in the newsgroup.	Evident
R6.10	Update the new message in the database and new title appears on message board.	Hidden

#### 2.1.6 Administrator Project-Maintenance Function

<u>Ref #</u>	<u>Function</u>	<u>Category</u>
R6.1	Allow administrator to create a new project with general description and group members.	Evident
R6.2	Update new project information in the database.	Hidden
R6.3	Allow the administrator to delete one of their projects.	Evident
R6.4	Update the project's information in the database	Hidden
R6.5	Allow the administrator to edit their project's general description.	Evident
R6.6	Update the project's description in the database	Hidden
R6.7	Edit Newsgroup Messages	Evident
R6.8	Update newsgroup Messages	Hidden

#### 2.1.7 Administrator-Maintenance of Team Member's

<u>Ref #</u>	<u>Function</u>	<u>Category</u>
R7.1	Allow the administrator to add a new user.	Evident
R7.2	Update the database with the new user information.	Hidden
R7.4	Allow the administrator to remove a user from a project.	Evident
R7.5	Confirm with administrator to remove the user from a project.	Evident
R7.6	Remove the user from that project not the data base	Hidden
R7.7	Allow the administrator to change team member's personal information.	Evident
R7.8	Update project information.	Hidden

#### 2.1.8 Logout

<u>Ref #</u>	<u>Function</u>	<u>Category</u>
8.1	Allow users to logout from the system	Evident
8.2	End the user session	Hidden



## 2.2 System Attributes

<b><u>Attribute</u></b>	<b><u>Detail and Constraints</u></b>	<b><u>Category</u></b>
Server-Operating System	Windows NT 4.0, Windows 98/ 2000 Server or Windows 2000 Advanced Server	Must
Web Server	Microsoft Personal Web Server or IIS 4.0 or greater	Must
Database	Microsoft Access	Must
Deployment Platform	ASP and HTML	Must
Browser Version	Internet Explorer 4.0 or higher	Must
Fast Response Time	Real-time efficient and fast queries	Want
Ease of use	The layout of the webpage is extensible, but yet simple and user friendly. The inputs are logical and straightforward.	Want
Simple Navigation	The links in the webpage allow the users to navigate from one page to another in a logical manner. The users should be able to drill down to the desired area within a few clicks.	Want
Real time Access	Newsgroup messages notification is immediate.	Want
Secure	A legal login and password are required to access the system. Only users with valid password are allowed to access the Administrator functions.	Must

### 3. Use Cases – High-level scenarios of user interaction with the software

#### 1.

Use Case	Actor enters their username and login information.
Actors	<b><u>User, Project Administrator, System Administrator</u></b>
Purpose	Grant authorized actors access to the system
Overview	The actor enters their name and password, if this information matches an account in the system database, the actor is granted access privileges according to their account status.
Type	Primary, Essential
Cross-references	R1.1 – R1.7
Preconditions	Actor has loaded the login page on their browser.

Actor's Action	System Response
1. Actor enters username and password, clicks "Sign In" or presses Enter button.	
	2. System searches for entered username in database, if found, checks if entered password is correct, and if username is a "first time user."
	3. Once authenticated, the system sends the actor the index page.

Alternative Courses:
Line 2: If entered username is not found, system returns page with error message "Username not found" with link back to login page.
Line 2: If entered password is incorrect, system returns page with error message "Invalid password" with link back to login page.
Line 2: If the database shows the entered username as being a first time user, system prompts actor to create a new permanent password and confirm it. Once confirmed, the system clears the first time flag and saves the new password in its database record for the user.

## 2.

Use Case	User views their personal information.
Actors	<b>User, System Administrator</b>
Purpose	Display actor's personal information on the screen.
Overview	The actor asks the system to display their personal information, at which time the system retrieves the information from its database, uses this information to generate the personal information page, and sends this page to the actor.
Type	Primary, Essential
Cross-references	R3.2 (this is the closest one, but doesn't match exactly)
Preconditions	The actor must be logged into the system. The actor must be at the index page of a project or be logged in as system administrator and have the master employee list page loaded.

Actor's Action	System Response
1. Actor clicks "My Information" link on project home page.	
	2. System retrieves relevant account information from database.
	3. System uses retrieved information to fill in fields of personal information page.
	4. System sends page to actor.

Alternative Courses:
Line 1: System administrator selects an employee's name from the master employee page.

### 3.

Use Case	User updates their personal information.
Actors	<b>User, System Admin</b>
Purpose	Allow the actor to make necessary changes to their personal information.
Overview	While viewing their personal information, the actor makes changes to the fields as necessary. When done, the actor sends the updated information to the system, which updates its database accordingly.
Type	Primary, Essential
Cross-references	R3.2, R3.3, R4.3 (Sys Admin section)
Preconditions	The actor must be logged in. The actor must be viewing their personal information page, or must be logged in as a system administrator and viewing a user's personal information page.

Actor's Action	System Response
1. Actor makes changes to personal information fields as they see fit.	
2. Actor clicks "Update" button.	
3. Actor's browser sends information in fields to system.	
	4. System overwrites current account information with updated information.
	5. System redisplayes personal information page with updated information.

Alternative Courses:
Line 2: If Actor clicks "Cancel" button, the system redisplayes the page with the original information.

**4.**

Use Case	User views project information.
Actors	<b>User</b>
Purpose	Display current project information on the screen.
Overview	While at a project home page, the actor asks the system to display the information page for the project. The system retrieves the project information from its database, generates the project information page using this data, and sends it to the actor.
Type	Primary, Essential
Cross-references	R4.2
Preconditions	The actor must be logged in and viewing a project home page.

Actor's Action	System Response
1. Actor clicks "Project Information" link on project home page.	
	2. System retrieves project information from database.
	3. System uses project information to generate project information page.
	4. System sends page to user.

**5.**

Use Case	User views team member's personal information.
Actors	<b>User</b>
Purpose	Display personal information of other team members on the screen.
Overview	While viewing the information page of a particular project, the actor asks the system to display the personal information of one of the other members of the project. The system retrieves the information from its database, uses this information to generate the personal information page, and sends this page to the actor.
Type	Primary, Essential
Cross-references	R4.3
Preconditions	The actor must be logged in. The actor must be viewing a project information page.

Actor's Action	System Response
1. Actor clicks a team member's name in the project information page.	
	2. System retrieves relevant account information from database.
	3. System uses retrieved information to fill in fields (non-editable) of personal information page.
	4. System sends page to user.

**6.**

Use Case	User views message board.
Actors	<b>User</b>
Purpose	Display the current discussion topics and posting options available.
Overview	The actor chooses to view the message board from the project home page. The system loads the list of current topics, and generates the message board page using this list. Once generated, the system sends the page to the actor.
Type	Secondary, Essential
Cross-references	R5.1, R5.2
Preconditions	The actor must be logged in and have a project home page loaded.

Actor's Action	System Response
1. Actor clicks the "Message Board" link on project home page.	
	2. System generates message board page, using current message list retrieved from database.
	3. System sends page to actor.

**7.**

Use Case	User adds new topic to message board.
Actors	<b>User</b>
Purpose	Allow the actor to contribute new topic messages for group discussion.
Overview	While viewing the message list, the actor indicates that they wish to generate a new topic message for discussion. The system sends the actor the new topic page, which the actor fills in with the desired topic and message. When finished, the actor submits the message to the system, which adds it to the message database.
Type	Secondary, Essential
Cross-references	R5.7-R5.9
Preconditions	The actor must be logged in and have a project message board loaded.

Actor's Action	System Response
1. Actor clicks the "Post New Topic" link on the project message board.	
	2. System sends new topic page to actor.
3. Actor fills in the topic and message fields of the new topic page.	
4. Actor clicks "Send" button.	
5. Actor's browser sends contents of fields to system.	
	6. System adds actor's username and current system timestamp to message.
	7. System stores complete message in message in message database.
	8. System sends actor updated message list page.

Alternative Courses:
Line 4: If actor clicks "Clear" button, all message fields will be cleared. (We should probably also have a "Cancel" button that takes the user back to the message list).



**8.**

Use Case:	Response topic on message board
Actors:	Admin, System Admin, Project members
Purpose:	To respond message(s) posted on the board
Overview:	User click to view the desired topic and then click reply in the bottom of the page. Enter message and then submit to the newsgroup.
Type:	Primary, Essential
Cross References	Functions: R4.4, R4.5
Preconditioning:	User must have clicked on a particular message before entering the reply message.

Actor's Action	System Response
1. this use case begins when users loads the newsgroup page	
	2. The system display a lists of messages posted by the project members
3. Users click on a message and enter reply message.	
	4. The system verifies the correct format and detects any missing info in the message. Message is then updated in the database and will then posted on the message board.

Alternative Courses:
Line 4: If system detects incorrect format or missing info, display error message and take user back into the previous page where the message is displayed.

## 9.

Use Case:	User post a new topic to the message board
Actors:	<b><u>Admin, System Admin, Project members</u></b>
Purpose:	To post a new message on the board and board-cast it to the entire group
Overview:	User clicks on the "new post" icon in the bottom of the page. Then enter appropriate information and body message.
Type:	Primary, Essential
Cross References	Functions: R4.7, R4.8, R4.9
Preconditioning:	User must have clicked on the "new post" icon.

Actor's Action	System Response
1. this use case begins when users loads the newsgroup page	
	2. The system displays a list of messages posted by the project members. An "new post" icon locates at the bottom of the page
3. Users click on "new post"	
	4. The system opens a new page where the users have to input appropriate info and body message
5. Users enter appropriate info and body message.	
	6. The system verifies the correct format and detects any missing info in the message. Message is then updated in the database and will then posted on the message board.

Alternative Courses:
Line 6: If system detects incorrect format or missing info, display error message and take user back into the previous page where the message is displayed.

**10.**

Use Case:	<b><u>Admin views team members personal information</u></b>
Actors:	<b><u>Admin, System Admin</u></b>
Purpose:	To view team members personal information
Overview:	User allowed accessing team members' personal information and editing it if necessary.
Type:	Primary, Essential
Cross References	Functions: R6.9
Preconditioning:	User must have granted permission to access this page

Actor's Action	System Response
1. This use case begins when users login appropriately.	
	2. The system displays basic functionality and some higher-level functions particularly for this type of users.
3. Users click on view members info	
	4. The system opens a new page where a list of personal information of the members is displayed.

Alternative Courses:
Line 1: If system detects incorrect login information, it will take the users back to the login page.

**11.**

Use Case:	<b><u>Admin adds new field line for personal information</u></b>
Actors:	<b><u>Admin, System Admin</u></b>
Purpose:	To add new field lines for team members personal information
Overview:	User allowed accessing team members' personal information and editing it if necessary.
Type:	Primary, Essential
Cross References	Functions: R6.7, R6.8
Preconditioning:	User must have granted permission to access this page

Actor's Action	System Response
1. This use case begins when users login appropriately.	
	2. The system displays basic functionality and some higher-level functions particularly for this type of users.
3. Users click on edit members info	
	4. The system opens a new page where a list of personal information of the members is displayed.
5. Users add new field lines for the personal information	
	6. Update the database and the page will show the added new field lines.

Alternative Courses:
Line 1: If system detects incorrect login information, it will take the users back to the login page.

**12.**

Use Case:	<b><u>Admin updates Project info</u></b>
Actors:	<b><u>Admin, System Admin</u></b>
Purpose:	To update project information where all the project members can be seen.
Overview:	User can only update the project info when he/she is the admin.
Type:	Primary, Essential
Cross References	Functions: R5.1-5.6
Preconditioning:	User must have granted permission to access this page

Actor's Action	System Response
1. This use case begins when users login appropriately.	
	2. The system displays basic functionality and some higher-level functions particularly for this type of users.
3. Users click on edit project info	
	4. The system opens a new page where it displays the project info he/she supervises.
5. Users update project info	
	6. Update the database and the page will show the updated project info.

Alternative Courses:
Line 1: If system detects incorrect login information, it will take the users back to the login page.

**13.**

Use Case:	<b><u>Admin deletes a user</u></b>
Actors:	<b><u>Admin, System Admin</u></b>
Purpose:	To delete a user from a project.
Overview:	User can only delete a user from the personal information page where other project member can access. It will not delete any info store in the database.
Type:	Primary, Essential
Cross References	Functions: R6.4-6.6
Preconditioning:	User must have granted permission to access this page

Actor's Action	System Response
1. This use case begins when users login appropriately.	
	2. The system displays basic functionality and some higher-level functions particularly for this type of users.
3. Users click on view members info	
	4. The system opens a new page where it displays a list of personal info of the members that he/she supervises.
5. Users delete the users from the list	
	6. Database will NOT delete the info the particular individual. The system will show the updated page without the information of the "deleted person"

Alternative Courses:
Line 1: If system detects incorrect login information, it will take the users back to the login page.

**14.**

Use Case:	<b><u>Admin adds a user</u></b>
Actors:	<b><u>Admin, System Admin</u></b>
Purpose:	To add a user from a project.
Overview:	User adds a user to the project and input all the necessary personal info of the individual. (or import data from the individual)
Type:	Primary, Essential
Cross References	Functions: R6.1-6.2
Preconditioning:	User must have granted permission to access this page

Actor's Action	System Response
1. This use case begins when users login appropriately.	
	2. The system displays basic functionality and some higher-level functions particularly for this type of users.
3. Users click on view members info	
	4. The system opens a new page where it displays a list of personal info of the members that he/she supervises.
5. Users add a user to the project	
	6. Update the database with the added information. The system will show the page with the new member's information.

Alternative Courses:
Line 4: If the admin input personal information in a wrong format, system will display error message and take the user back to the pervious page.
Line 1: If system detects incorrect login information, it will take the users back to the login page.

**15.**

Use Case	System Admin views all current projects.
Actors	<b><u>System Admin</u></b>
Purpose	Allow the actor to view all active Sapient projects using FFF.
Overview	The actor requests the master project listing from the system. The system retrieves the current project index page and sends it to the actor. The actor can then select an individual project information page using Use Case 12.
Type	Secondary, Non-Essential
Cross-references	R4.1 (Sys Admin Section)
Preconditions	The actor must be logged in as a System Administrator and have the System Administrator index page loaded.

Actor's Action	System Response
1. Actor clicks the "Master Project List" link on the System Administrator index page.	
	2. System retrieves master project list page and sends it to the actor.



**16.**

Use Case	System Admin views information for all current project team members.
Actors	<b><u>System Admin</u></b>
Purpose	Allow the actor to view information of all active Sapient employees with access to FFF.
Overview	The actor requests the master employee listing from the system. The system retrieves the current employee index page and sends it to the actor. The actor can then view an individual team member's information using Use Case 10.
Type	Secondary, Non-Essential
Cross-references	R4.2 (Sys Admin section)
Preconditions	The actor must be logged in as a System Administrator and have the System Administrator index page loaded.

Actor's Action	System Response
1. Actor clicks the "Master Employee List" link on the System Administrator index page.	
	2. System retrieves master employee list page and sends it to the actor.

**17.**

Use Case	System Admin adds a project.
Actors	<b><u>System Admin</u></b>
Purpose	Allow the actor to create a new project.
Overview	The actor chooses to add a project from the "Add/Remove Project" page. The system sends the actor the project creation page. The actor submits the information required to initialize the project. The system adds the project to its database.
Type	Secondary, Essential
Cross-references	R4.3 (Sys Admin section)
Preconditions	The actor must be logged in as a System Administrator and have the System Administrator index page loaded.

Actor's Action	System Response
1. Actor clicks the "Add/Remove Project" link on the System Administrator index page.	
	2. System sends Add/Remove Project page to actor.
3. Actor clicks the "Add Project" link.	
	4. System sends Add Project page to actor.
5. Actor fills in project information fields.	
6. Actor clicks "Submit" button.	
7. Actor's browser sends information to system.	
	8. System uses submitted information to add new project to project database.

Alternative Courses:
Line 8: System detects that actor omitted required information field(s).
System sends error message to actor, returns to line 4.

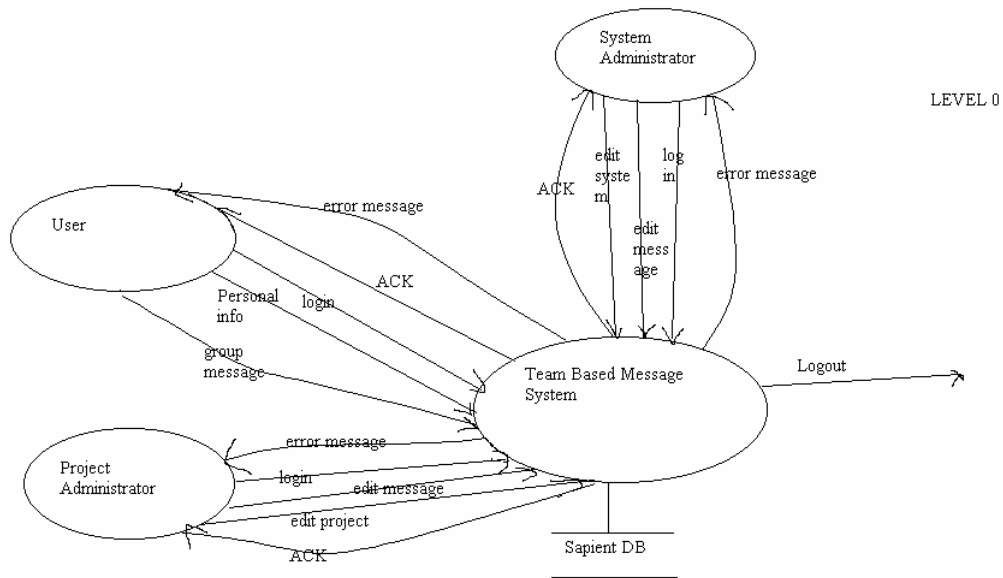
**18.**

Use Case	System Admin deletes a project.
Actors	<b><u>System Admin</u></b>
Purpose	Allow the actor to delete a project.
Overview	The actor chooses to delete a project from the "Add/Remove Project" page. The system retrieves the list of current projects and uses the list to generate the project deletion page. The system sends the actor the project deletion page. The actor selects the project to be deleted. The system removes the selected project from its database.
Type	Secondary, Essential
Cross-references	R4.3 (Sys Admin section)
Preconditions	The actor must be logged in as a System Administrator and have the System Administrator index page loaded.

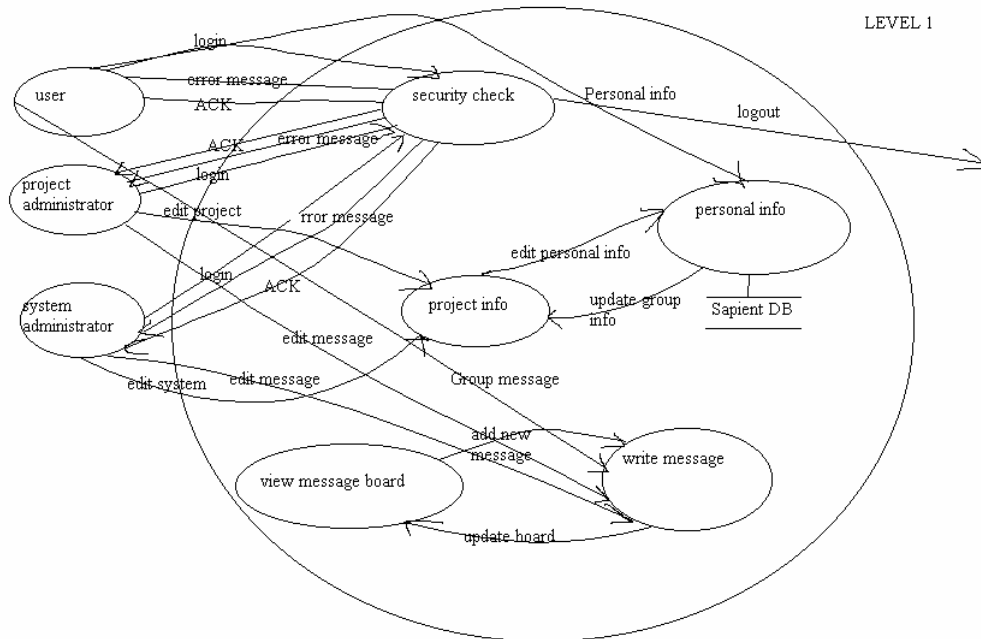
Actor's Action	System Response
1. Actor clicks the "Add/Remove Project" link on the System Administrator index page.	
	2. System sends Add/Remove Project page to actor.
3. Actor clicks the "Remove Project" link.	
	4. System retrieves current project listing.
	5. System uses listing to generate project deletion page.
	6. System sends project deletion page to actor.
7. Actor selects project to be deleted.	
	8. System removes selected project from project database.

## 4. Data Flow Diagrams

### 4.1 Level 0 DFD




### 4.2 Level 1 DFD



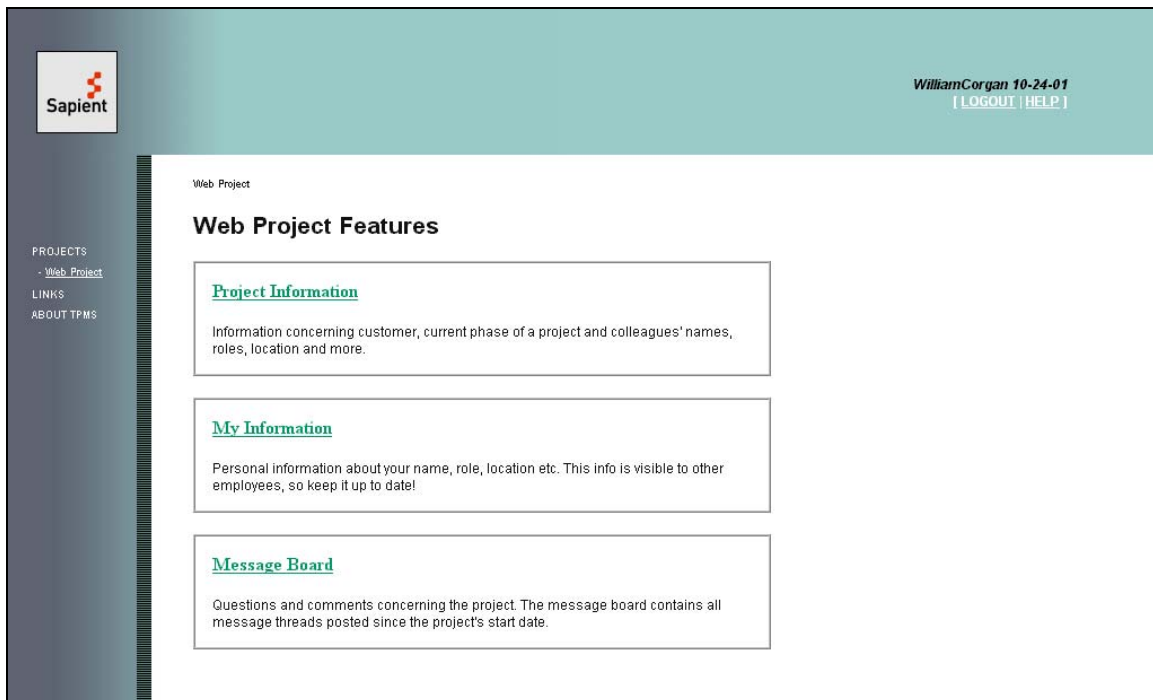
## 5. Preliminary Users Guide

### 5.1 Screen Shots



The screenshot shows a login interface for 'Sapien'. It features a logo in the top left corner. Below the logo, there are two input fields: 'Username:' and 'Password:'. To the right of the 'Password:' field is a 'Sign in' button. Below the button, there is a link that says 'Problems logging in? E-mail [admin@project.com](mailto:admin@project.com)'.

**Login** – Check users' username and password. If valid, user will be redirected to next page, otherwise an error prompt will be displayed and user is asked to reenter his/her data.



The screenshot shows the 'project\_main' page. It has a header with the 'Sapien' logo on the left and the user 'WilliamCorgan 10-24-01' with links for 'LOGOUT' and 'HELP' on the right. A left sidebar contains links for 'PROJECTS' (with 'Web Project' selected), 'LINKS', and 'ABOUT TPMS'. The main content area is titled 'Web Project' and 'Web Project Features'. It contains three sections: 'Project Information' (describing customer, phase, names, roles, location), 'My Information' (describing personal info visible to others), and 'Message Board' (describing questions and comments).

**project\_main** – An overview of the current projects features: project information / my information / message board.

**Sapien** WilliamCorgan 10-24-01 [LOGOUT] [HELP]

Web Project / Project Info

### Project Information

Customer:	Sapien
Project Name:	Instant Messaging Service
Project Phase:	Prototyping
Comments:	Get feedback from Sapien every week

### Team members

Name	Role	Office	Phone	Email	Status	Project Name	Project Phase	Start Date	Stop Date	Employment Status
<a href="#">Emily</a>	Eng	LA	123 456	emily@sapien.com	Vacation	Web Stuff	Hmmm	Today	Future	FT
<a href="#">Fred</a>	Eng	NY	123 456	fred@sapien.com	Vacation	Web Stuff	Hmmm	Today	Future	PT
<a href="#">Bill</a>	Boss	Seattle	xxx yyy	bill@microsoft.com	Relaxing	Don't know	Hmmm	Today	Future	PT
<a href="#">Fred</a>	Eng	NY	123 456	fred@sapien.com	Vacation	Web Stuff	Hmmm	Today	Future	PT

**project\_info** – Data specific to the project, e.g., the name of the customer, project name, project phase and information on the team members working on the project. The left column of the team-member table (Name) links to more information specific to that user. If user is currently logged in as admin, he/she has the possibility to edit and add user fields, and to remove users (e.g., if someone has left the project).


**Sapien** WilliamCorgan 10-24-01 [LOGOUT] [HELP]

Web Project / My Info

### Personal Information

First Name	<input type="text" value="Fredrik"/>
Last Name	<input type="text" value="Wass"/>
Role	<input type="text" value="Programmer"/>
Start Date	<input type="text" value="10/10/01"/>
Stop Date	<input type="text" value="12/25/01"/>
Current Office	<input type="text" value="Los Angeles"/>
Email	<input type="text" value="fwass@ucla.edu"/>
Phone	<input type="text" value="+ (310)-208 0710"/>
Project Phase	<input type="text" value="Working hard"/>
Employment Status	<input type="text" value="Part Time"/>

**my\_info** – The information that will be displayed about current user to other users. Only the current user and the admin can change this data.



**WilliamCorgan 10-24-01**  
[\[ LOGOUT \]](#) [\[ HELP \]](#)

PROJECTS

- Web Project

LINKS

ABOUT TPMS

[Web Project / Message Board](#)


## Message Board

[\[ Post New Topic \]](#)

- » [Problem with server configuration](#) **Fred** 10/22/01 22:45
  - [A possible solution](#) **Billy** 10/22/01 22:45
- » [Client want us to add new features](#) **Bill Gates** 10/22/01 21:35
- » [Is there no standard anymore?](#) **Denis** 09/22/01 21:35
- » [Can you feel it coming in the air tonight?](#) **Christine** 10/22/01 21:35
- » [What shall we do with the drunken sailor?](#) **Christine** 10/22/01 21:35
- » [Problem with server configuration](#) **Fred** 10/22/01 22:45
- » [Client want us to add new features](#) **Bill Gates** 10/22/01 21:35
- » [Is there no standard anymore?](#) **Denis** 09/22/01 21:35
- » [Can you feel it coming in the air tonight?](#) **Christine** 10/22/01 21:35
- » [What shall we do with the drunken sailor?](#) **Christine** 10/22/01 21:35
- » [Problem with server configuration](#) **Fred** 10/22/01 22:45
- » [Client want us to add new features](#) **Bill Gates** 10/22/01 21:35
- » [Is there no standard anymore?](#) **Denis** 09/22/01 21:35

[1](#)
[2](#)
[3](#)
[4](#)
[5](#)
[6](#)
[Next](#)

**message\_board** – Shows the latest messages posted by team members. Clicking the subject heading links to the entire message, and also allows the user to reply.



WilliamCorgan 10-24-01

[\[ LOGOUT \]](#) [\[ HELP \]](#)

PROJECTS

- Web Project

LINKS

ABOUT TPMS

[Web Project / Message Board / Reply](#)

## Problem with server configuration

Posted by [Fred](#) 10-22-01 at 22:45

The Java™ programming language has a lot in common with every day life. Each day For action to take place in Java objects, blocks of code called methods are used. Methods tell an application what to do when buttons are clicked, menus are opened, and text is typed.

Methods are either predefined or are created from scratch to manipulate objects, or the data within those objects.

The Dive Log application you're going to create consists of many objects. These objects are based on classes that detail how the object is defined and how it behaves.

This lesson covers the basics of using predefined classes and creating, or designing, new objects with your own classes. In addition, you'll learn how to work with objects by calling predefined methods and writing new methods to get the application to do what you want it to do.

### Reply to Message

Message Title

Date

10/22/01

Name

Billy Corgan

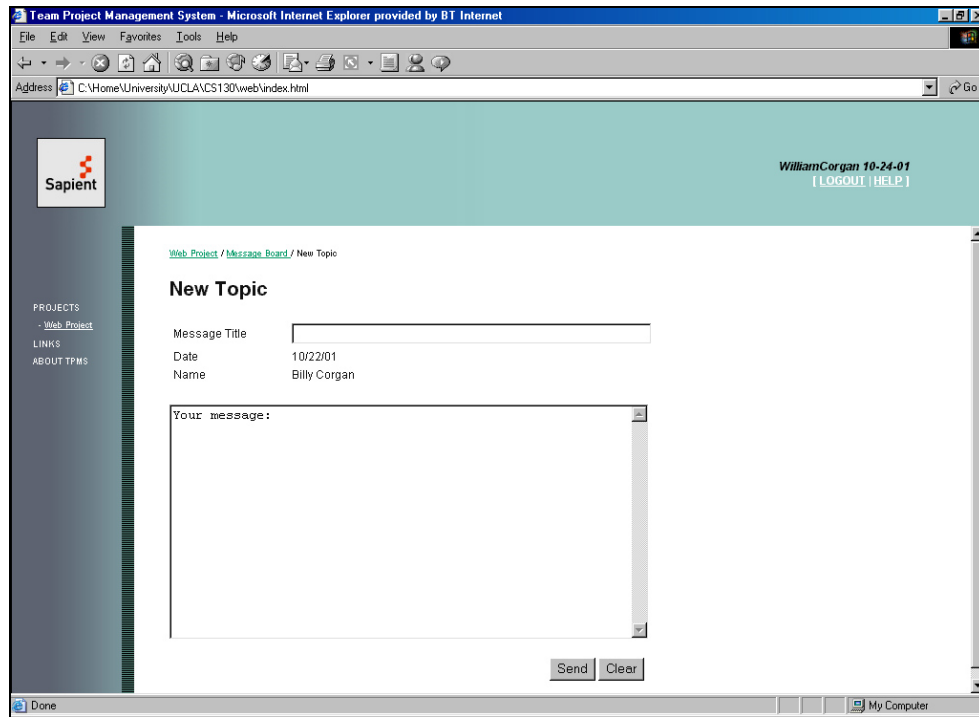
Your message:

Send

Clear

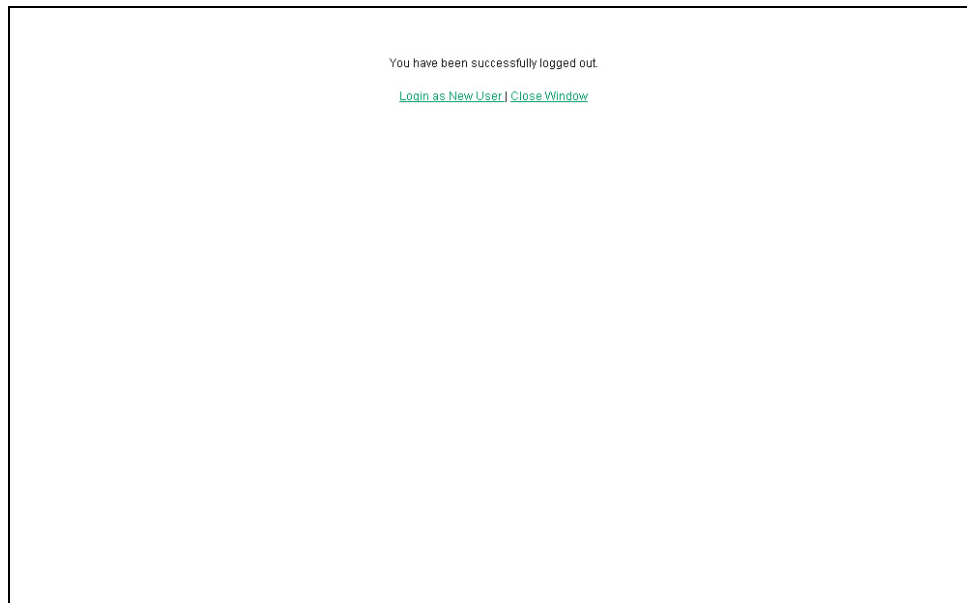
**message\_board\_reply** – Reply to message; this allows a user to reply to the message selected.





**message\_board\_new** - adds a new message.

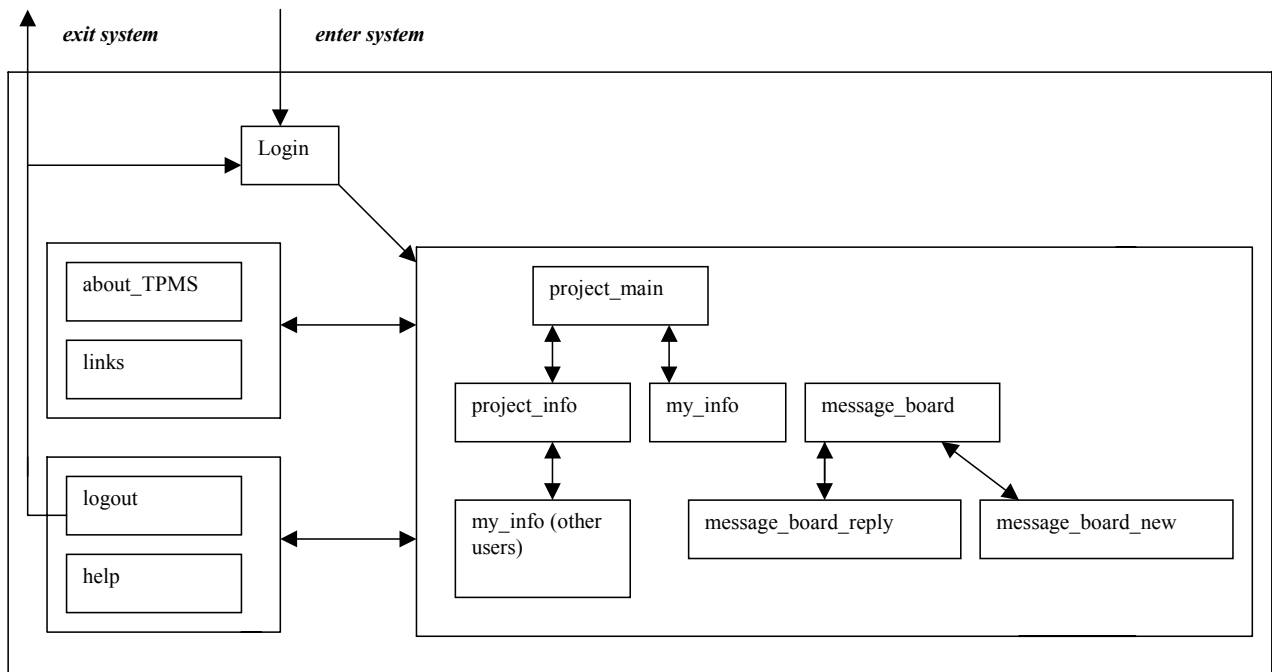
---



**logout** – If the user has been successfully logged out, the above window will be displayed. There are two options here, either to return to the log in window and log in as another user (if user is both registered as admin and an ordinary user), or the user can close the window.

## 5.2 Screen Navigation Diagram

Here is the navigation diagram of the web application:



## 6. Project Planning (Part 1 of the Project Plan)

### 6.1 Work Breakdown Structure

Work Task	Planned Start	Actual Start	Planned Complete	Actual Complete	Assigned Person	Time Allocated	Actual Time
<b>Requirements Analysis</b>							
Initial Meeting With Customer	10/3/01	10/3/01	10/3/01	10/3/01	All	2 hr	2 hr
Group meeting to discuss project goals	10/8/01	10/8/01	10/8/01	10/8/01	All	2 hr	1 hr
Create Concept Diagram	10/11/01	10/11/01	10/15/01	10/15/01	All	4 hrs	4 hrs
Define System Functions and System Attributes	10/11/00	10/11/00	10/25/00	10/23/00	Emily Fred Leona Denis	10 hrs	6 hrs
Document System Functions and System Attributes	10/14/01	10/15/01	10/25/01	10/24/01	Christine Emily	5 hrs	5 hrs
Develop Use Cases	10/11/00	10/14/00	10/15/00	10/17/00	Christine Drew	7 hrs	8 hrs
Develop Data Flow Diagrams	10/15/01	10/15/01	10/29/01	10/29/01	Leona Emily	5 hrs	3 hrs
Create Data base	10/9/01	10/9/01	10/15/01	10/11/01	Denis Leona	4 hrs	4 hrs
Document Part I of Requirements Document	10/15/01	10/15/01	10/20/01	10/17/01	Drew	3 hrs	2 hrs
Meeting to review screen shots and system functionality	10/22/01	10/22/01	10/22/01	10/22/01	Denis Fred	1 hr	1 hr
Meeting to Review and Refine: Screen Shots, Use Cases and System Functions	10/24/00		10/25/00	10/25/00	All	5 hrs	6 hrs
Assemble Requirements Document	10/11/00	10/14/00	10/31/00	10/31/01	Emily Christine	5 hrs	7 hrs
Create Screen Navigation Diagram	10/25/01	10/25/01	10/30/01	10/30/01	Fred	1 hrs	2 hrs
<b>Milestone:</b> Turn In Requirements Document	10/31/00	10/31/00	10/31/00	10/31/00	*	*	*
<b>Design</b>							
Design Web Pages	10/11/01	10/11/01	10/16/01	10/13/01	Emily Fred Leona	4 hrs	3 hrs
Design Data Base	10/15/01	10/13/01	10/22/01	10/20/01	Denis Leona	3 hrs	3 hrs
Design Task #3							

Work Task	Planned Start	Actual Start	Planned Complete	Actual Complete	Assigned Person	Time Allocated	Actual Time
<b>Implementation</b>							
Code Data base and ASP	10/15/01	10/15/01	11/5/01		Denis Leona	15 hrs	
Code Web Pages	10/11/01	10/14/01	10/24/01		Fred	10 hrs	
Set up Server	10/5/01	10/5/01	10/10/01	10/7/01	Denis	3 hrs	2 hrs
Integrate Sapient DB with our system	10/26/01		11/05/01		Denis Leona		
<b>Testing</b>							
Testing Tasks 1 - N							
<b>Test Plan</b>							
Test Plan Tasks 1 - N							
<b>Demo</b>							
Demo Tasks 1 - N							

## 6.2 Work Input Summary

We have allocated approximately 30 hours of work for each team member to complete this project. Our team has 6 team members, thus the total hours allocated for the project are 180 hours. This is a reasonable estimate, because we were allowed only 9 weeks to complete the project. Specifically, this gives us 20 hours of work per week for the project, 3 hours per person per week.

For the tasks that have been completed so far, we see from the work break down structure that we have allocated 68 hours total for work on the project. We have spent 59 hours so far on the project and our estimated ratio of time allocated to time required is 85%. This tells us that we are right on track to complete the project and are ahead of schedule at this point.

## 7. Appendix

### 7.1 Data Dictionary

- *DB*: Database
- *DFD*: Data Flow Diagram
- *Use Case*: Gives the different scenarios for interaction of an actor with the system FFF.
- *Microsoft Personal Web Server / Microsoft Internet Information Services*: Both are Microsoft Web Servers
- *Microsoft Access*: Database Management Software
- *ASP*: Active Server Pages used to create dynamic HTML pages
- *UML*: Uniform Modeling Language. A language used to design object-oriented models.
- *Democratic Decentralized Process*: Process model where there is no defined leader, the groups communicates horizontally with each other and good for short term projects. All decisions are made by a consensus.
- *User*: Sapient employee who is currently working on a particular software project.
- *Project Administrator*: Sapient employee who is the administrator of his project. Administrates his group's information in the system.
- *System Administrator*: Sapient employee who is the administrator over the whole system. This person is responsible for maintaining and updating the system.

## 7.2 Correspondence with Sapien

*Date: Thu, 11 Oct 2001 14:37:03 -0400*  
*From: Emily Tramel <emilyt@ucla.edu>*  
*Subject: UCLA CS 130*  
*To: bmeltz@sapien.com*

Hi Brock,

My name is Emily Tramel and I will be the contact person for group F. So far we've met and discussed the design of the product and are looking forward to working with you to create a useful product.

I just wanted to clarify with you regarding your request for Instant Messaging. Though in the end we would like to work this functionality into our product, we are requesting that initially we create a newsgroup type messaging service. This would be like the broadcast functionality you requested. The benefit of the newsgroup is that all past conversations will be on the server for the whole group to view whenever they need. In this way, though someone might not be directly involved in a conversation they will be able to view other conversations and see what their project partners are doing.

I look forward to communicating more with you throughout this quarter.

Sincerely yours,  
Emily Tramel

*Date: Thu, 11 Oct 2001 13:37:19 -0400*  
*From: System Administrator <postmaster@sapient.com>*  
*Subject: Undeliverable: UCLA CS 130*  
*To: emilyt@ucla.edu*

Your message

To: bmeltz@sapient.com  
Subject: UCLA CS 130  
Sent: Thu, 11 Oct 2001 14:37:03 -0400

**did not reach the following recipient(s):**

bmeltz@sapient.com on Thu, 11 Oct 2001 13:37:15 -0400  
The recipient name is not recognized  
The MTS-ID of the original message is: c=us;a=  
;p=sapient;l=SMCCIMS0101101117374VCQR25H  
MSEXCH:IMS:sapient:SAPIENT:SMCCIMS01 0 (000C05A6) Unknown Recipient

Date: Thu, 11 Oct 2001 14:37:03 -0400  
From: Emily Tramel <emilyt@ucla.edu>  
Subject: UCLA CS 130  
To: bmeltz@sapient.com  
Message-id: <200110111737.KAA17413@serval.noc.ucla.edu>  
MIME-version: 1.0  
X-Mailer: Internet Mail Service (5.5.2653.19)  
Content-type: text/plain; CHARSET=US-ASCII  
X-MS-Embedded-Report:

From: Emily Tramel [<mailto:emilyt@ucla.edu>]  
Sent: Monday, October 15, 2001 7:46 PM  
To: Brock Meltzer  
Subject: UCLA CS 130

Hi Brock Meltzer,

My name is Emily Tramel and I'll be the contact person for group F. So far we've designed the web page layouts and are working on some of the more technical issues. Before we continue working, I wanted to check with you regarding what you categorize to be required vs optional. One main optional requirements (according to the project description) that we've chosen to implement is the broadcast messaging. Please let me know what other features you require in your software.

Sincerely,  
Emily  
emilyt@ucla.edu

*Subject: RE: UCLA CS 130*  
*To: 'Emily Tramel' <emilyt@ucla.edu>*

Hi Emily,

Thanks for the email. To answer your question, the only required categories are italicized under "Core Business Requirements" (bottom of page 1) such as *Login and Identification* and *Team Member Details*. Optional Requirements are called out in the "Optional Business Requirements" (bottom of page 3) section and include *Broadcast Messaging*, *Instant Messaging*, *Multi-user Chat* and all other italicized categories. These optional requirements are there for your team if you feel like making your application more robust.

Sorry about the confusion with my email address. I believe that I am working with another group so if you know who the main contact is on that team, can you let them know about my email address?

Thanks and let me know if you have any further questions,

Brock

-----Original Message-----

**From:** Emily Tramel [<mailto:emilyt@ucla.edu>]

**Sent:** Thursday, October 25, 2001 1:01 AM

**To:** Brock Meltzer

**Subject:** CS130 at UCLA

<< File: sapient.zip >> << File: CS130 Doc.doc >> << File: ATT216157.txt >> Hi Brock,

Attached to this email is the screen shots of the web site that we have designed for you. The pages are not 100% complete, since some of the links are not working. I am requesting that you look at the attached .html files in the folder "sapient.zip" to make sure that my group has captured all your desired functionality. In order for you to better understand what functionality we will include in your site I have also attached a document titled "CS130 Doc.doc", this is a partial copy of our requirements document. Please take a look at both of these files to let us know if we are on the right track. We would appreciate any comments that you might have.

Thank you for your time.

Sincerely,  
Emily Tramel  
CS130 at UCLA  
[emilyt@ucla.edu](mailto:emilyt@ucla.edu)



Subject: RE: CS130 at UCLA  
To: 'Emily Tramel' <emilyt@ucla.edu>

Thanks for the email Emily! From the word document and html files, it looks like you guys are on the right track. I would suggest to make sure you are completely done with the functionality and technological requirements before you start designing so that you don't run into trouble later trying to implement everything. But, it seems like your team is on top of your work and are in good shape.

One thing I didn't read about much, but may already be implied in your work, is the ability to build in an automated import process where common "core" information can be entered into the application you are building. This can eliminate a lot of work from the user side if personal profiles are already set up and up to date.

I hope this helps and keep doing a great job!

Brock