Test Plan for Team TerraUser

The Web-based User Management Project

Michelle Harr
Naoko Tsunekawa
Daniel Wallace

6 May 2002

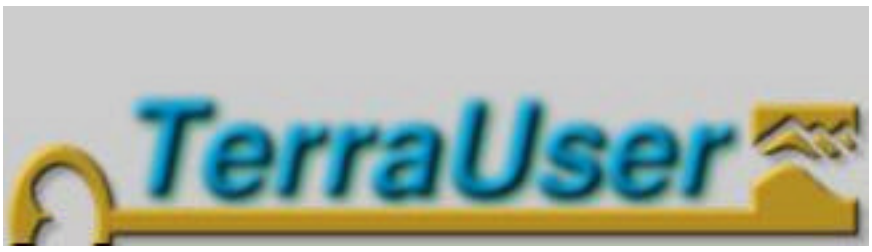

TABLE OF CONTENTS

This document is used to define the plan, scope, environment, roles and responsibilities for testing our system. This document includes the tests that we plan to perform, how to perform these tests, and when to perform these tests. When the results of these tests are collected this document should be updated to reflect the results.

1.1 Project Objectives

The business objectives of the *TerraUser* system are to provide a secure interface to our client's web-based applications, along with a way to manage and keep track of the users that are able to login and access these applications. Our client is Deborah Lee Soltesz from the U.S. Geological Survey; she works in the Astrology division as the web mistress.

1.2 System Description

The TerraUser system will force users to login before they can access our client's web-based software applications. The TerraUser system also allow the user to login to the TerraUser system and perform such tasks as changing their password, searching for application data that they are interested in, or set how they would like to view certain things on the application web pages (like font size, etc).

Besides the user side of TerraUser system there is an administration interface. This allows an administrator to be able to manage the users that have access to these web applications. The administrator functionality consists of such things like being able to add users, groups, set permissions, etc.

This document focuses most of its energy on discussion of testing issues. A more in depth discussion of the TerraUser system can be found in our Requirements Document located http://www.cet.nau.edu/~dw2/terrauser.

1.3 Plan Objectives

The objectives of the project Test Plan to have the tests, the test schedule, and the test process well defined, so when the testing phase occurs it is more complete. The plan is to try and do as much testing with the time and resources as are possible for the scope of this project. We will constantly need to make sure that the tests are updated.

1.4 Hardware Environment

The hardware requirements for the test environment include a network connection to World Wide Web, preferably a connection greater than or equal to a 56K modem. For the speed of a test machine we recommend a Pentium II 266MHz or greater. Testing can be completed on any platform, running any operating system, as long as there is a supported browser installed.

Our web server hardware consists of:

- AMD 4 Athlon(tm) processor, 1334 MHz
- 513608 kB RAM

- 658656 kB swap space
- Linux 7.3 professional version 2.4.10-4GB

1.5 Software Environment

The software requirements for the test environment include supported browsers (i.e. Netscape4.7 or higher, Internet explorer 5.0 or higher, and lynx). Any operating system or platform may be used as long as it has one of the supported browsers installed. We are not using any automated testing tools for this project.

The software that was installed on our server includes:

- SuSE LINUX 7.3,
- Tomcat (version 3.2.3),
- Apache (version 1.3.20),
- MySQL (version 3.32.44),
- java 2 sdk 1.3.1_02 and java 2 sdk ee 1.3.1, etc.

1.6 Test Documents

The test documents that have been created for this project are included in the appendix of this document. The test documents include, usability lab manual, acceptance tests, use cases, test cases, and a usability questioner. We will also summarize the results and include a report.

1.7 References

Some documents that define and trace the system requirements that are to be tested include our Requirements Specification Document along with our Requirements Traceability Matrix.

1.8 Outstanding Issues

At this state in the project planning we do not see any major issues or problems that are relevant to testing. We might have a problem finding relevant usability testing groups. We only have enough time in the schedule to perform one set of usability testing when we should have done at least two sets.

2. TEST SCOPE

2.1 Features To Be Tested

There are many pieces of functionality that need to be tested in the TerraUser system.

Basic User (Editor and Guest) functionality that will be tested includes:

- Change password (editor only)
- Start TerraData applications (editor only)
- Add/modify user's preference
- Search option
- Send e-mails

- Help documentation
- Log off

Administrator functionality that will be tested includes:

- Update user information
- Add/delete users
- Reset/expire user's password
- Add/delete/update teams
- View user log files
- Add/delete/modify user database fields
- Log user off
- Post message of the day (MOTD)
- Documentation on how to configure system

2.1 Features NOT To Be Tested

We don't have any major features or combination of features that will not be tested. We will also not be performing any performance testing, or load testing because of lack of time in the semester.

3. TEST PHASES

The test phases include: unit testing, integration testing, usability testing and acceptance testing. Unit testing is completed during implementation; regression tests are completed after integration; and acceptance tests are preformed after most of the bugs have been fixed. Following is a more in depth description of these different testing phases.

3.1 Definition

Unit testing consists of testing the functionality of and around a feature that has been implemented. The scope is limited to that feature. The designer usually preforms testing of the functionality of the feature. For example if designer A implements the login feature, then they would first test that they are able to connect to the database, then they would test weather or not their login web page could communicate and connect to the database. They would continue with testing whether the correct error message appeared or whether they were able to login.

Integration testing occurs when the whole system is integrated or put together. This includes running regression tests to try and cover as many possible situations as possible. By running all the tests that are outlined in the Test Cases we perform our regression tests.

Usability testing is the phase of testing where you actually sit different types of users in front of the system, ask them to perform certain tasks and evaluate the results to see if modification of the software is needed.

Acceptance testing is the last phase in the testing cycle. This is the last check before the software is delivered. It includes such things as the last check of the documentation, as well as a last check of the functionality. It ensures that there are no major problems left outstanding. The Acceptance testing checklist is at the end of this document. All group members participate in this phase of the testing.

3.2 Participants

There are a variety of participants that will be included in the testing phase. A majority of the testing is to be completed by the design team. The usability testing is where outside individuals get involved. The data for performing these tests can be found in the Use case or test case documents, along with the usability questioner, all of which are appended at the end of this document. To check and see if requirements are validated you can look at the Requirements Tractability Matrix.

Summary of test phase, participants, and descriptions are shown in Table 1.

Test Phase	Responsible Party	Description (Level, areas, environment, etc.)
Unit Testing	Design team member who implemented that feature	Parallel to implementation phase, each function will be tested by the team members to check to see all the modules are functioning correctly.
Integration Testing	All design team members	All design team members should run through the Test cases at least once, if time becomes an issue the tests can be split up.
Usability Testing	Individuals will be recruited from outside the design team to represent all the different types of users.	Individuals are as follows: ▪ Scientists from USGS ▪ Administrators from USGS ▪ General public user (guest) ▪ Application developers Instructions will be given to each user types to follow. Design team members will observe how they behave.
Acceptance Testing	These tests will be divided among the design team members	This testing can be done by following the instruction, which can be found in Appendix A. All design team members must sign-off before goes out as a final product.

Table 1: Summary of Testing Phases

3.3 Schedule

The testing phase of this project is scheduled to start on Monday, April 8th and run through Monday, April 15th. This gives the team two weeks to complete the testing. Below is a detailed schedule of testing activities, along with associated responsibilities.

Test Activity	Start Date	Finish Date	Responsible	Dependencies
Unit Testing	03/06/2002	04/03/2002	Design Team	Check against Use Cases for functionality
Integration Testing	04/04/2002	04/07/2002	Design Team	Run through Test Cases
Usability Testing	04/08/2002	04/12/2002	Recruited Individuals	Individual follow the instruction given.
Acceptance Testing	04/08/02	04/19/02	Design Team	

Table 2: Schedule

4. FUNCTIONALITY TESTING

4.1 Approach

Full regression tests (test cases) will be run when the system is integrated. Test cases cover all the possibilities on the variety of functionalities of the product. If a modification in the code is made only testing around that feature is going to be made. Each member of the design team will run the test case script once. Acceptance tests will make sure all the functionality has been developed and integrated before the product is delivered to the client. We are not running any automated testing, because we do not have the time to write the scripts.

4.2 Test Data

The test data covers a variety of conditions, including the boundary cases. Test data is specified in the test case script. For example data: Null, case-sensitivity, 1, embedded spaces, limit.

4.3 Requirements Tracing

Tests and test results will be mapped to documented system requirements by documenting them in the requirements traceability matrix. We will also include a summary of the test results at the end of this document.

5. USABILITY TESTING

5.1 Approach

Usability tests will be run after integration of the system. We hope to get a wide variety of users. We are very flexible on the locations and space requirements for the test environment. Some of the sites where testing will occurs include: USGS, NAU labs, at home. At least two computers will be set for the testing for users interaction. Manual will be given to the test users to proceed. Team members will stand behind them to observe their behaviors. We ask users to either write down or talk loud how they think about the product. Team member will ask several questions regarding to the product and thank them for their participation.

5.2 Training

- Users should require less than fifteen minutes of training to use the interface.
- Administrators should require at most one hour of training to use the application.
- Application developers that want to use the interface should require less than one hour of training to interface (Reading the documentation should be enough.)

5.3 User Groups

To thoroughly test the TerraUser application we are going to have to do usability testing on different groups or categories of users:

1. Administrators of the system, whose task is to manage users.
2. Scientists, who use the system to edit, access, and manage their data.
3. The general public, who will use the system to view information that they have an interest in.
4. The Application developer, who wants to use the interface for secure access to their application.

APPENDIX

A. FINAL ACCEPTANCE TESTING

The Acceptance testing is kept short and sweet while exercising as much of the functionality as possible.

Today's Date: _____ Version Release Number: _____

_____1. A step-by-step execution of all test cases has been run and the results have been verified as correct (no steps are missing, and instructions are easy to follow).

_____2. Functionality of software has been documented. Screen shots in documentation have been checked and match actual screen shots exactly.

_____3. User documentation checked against the requirements of the software, and support platform and system requirements match.

_____4. Software has been checked to ensure that it gracefully handles erroneous data, such as out of range values.

_____5. Information and error messages have been reviewed: they are informative with correct spelling and grammar.

_____6. All intended web browsers have been tested:

	Version	Version	Version
Netscape			6.2
Internet Explorer			5.5

Tests checked on the lines above have been successfully completed. No severe programming errors remain.

_____ _____
Signature Date

_____ _____
Signature Date

_____ _____
Signature Date

Usability Lab Manual

Follow the steps listed below. Please write down any notes or comments that you have on this page.

User Type = Editor

1) Go to Login page (inside USGS http://192.168.253.11/jsp/servlet/Login or outside USGS http://terrauser.wr.usgs.gov/jsp/servlet/Login)
   Comments:

2) Login as an Editor (***User ID**:* ted ***Password:*** rocks)
   Comments:

3) Access an application (access **TerraData** application)
   Comments:

4) Update your preferences (**background:** light yellow, **font:** Arial, **font size:** 12 pt, **font color:** blue)
   Comments:

5) Perform a search using USGS search engine option
   Comments:

6) Email an administrator
   Comments:

7) Change your password (***Old Password:*** rocks ***New Password:*** garnets)
   Comments:

8) Logout
   Comments:

Usability Lab Manual

Follow the steps listed below. Please write down any notes or comments that you have on this page.

User Type = Guest

1) Go to Login page (inside USGS http://192.168.253.11/jsp/servlet/Login or outside USGS http://terrauser.wr.usgs.gov/jsp/servlet/Login)
   Comments:

2) Login as a Guest
   Comments:

3) Send an Email to administrator
   Comments:

4) Perform a search using USGS search engine option
   Comments:

5) Logout
   Comments:

Usability Lab Manual

Follow the steps listed below. Please write down any notes or comments that you have on this page.

User Type = Administrator

1) Go to Login page (inside USGS http://192.168.253.11/jsp/servlet/Login or outside USGS http://terrauser.wr.usgs.gov/jsp/servlet/Login)
   Comments:

2) Login as an Administrator (***User ID**:* admin01 ***Password:*** max)
   Comments:

3) Add a new user (***User ID**:* user01 ***Password:*** max ***First Name:*** user ***Last Name:*** last ***email:*** terrauser@cet.nau.edu)
   Comments:

4) Update an existing users information (***User ID**:* user01; update user's last name to 'test01')
   Comments:

5) Search for a user (***User Type**:* editor ***First Name:*** Michelle)
   Comments:

6) Create a Team (***Team Name**:* team01 ***Team Contact:*** Ted ***Team email**:* terrauser@cet.nau.edu)
   Comments:

7) Update a Teams Information (***Team Name**:* team01; update team contact to 'user01')
   Comments:

8) Add a user to a Team (***User ID**:* user01 ***Team Name**:* team01)
   Comments:

Usability Lab Manual cont.

9) Remove a user from a Team (***User ID**:* user01 ***Team Name**:* team01)
   Comments:

10) Grant Team Access to an Application (***Application**:* TerraData ***Team**:* team01)
    Comments:

11) Remove team Access to Application (***Application**:* TerraData ***Team Name**:* team01)
    Comments:

12) Post a MOTD (Message of the Day)
    Comments:

13) Email a user
    Comments:

14) Backup db (name: backup_2002_04_20)
    Comments:

15) Delete a user (***User ID**:* user01)
    Comments:

16) Delete a Team (***Team Name**:* team01)
    Comments:

17) Logout
    Comments:

C. TEST CASES

For this product, the major test cases are:

- Invisible Application Login
- Application Login
- Basic User Navigation
- Administrator Navigation

Each case will be tested during the integration-testing phase. Detailed test cases as follows.

Test Case Number: 1 (Invisible Application Login)

Module: Invisible Interface Login Module
Functional Specification: User Authentication
Test Objective: To check whether the entered User name and Password is valid or invalid and if valid the application directs the user directly into the application, bypassing the TerraUser interface.
Assumptions: User is on the login page for a TerraWeb Application (ex. http://terraweb.wr.usgs.gov/TerraData/login.html)
Test Data: USER Name = User01 and PASSWORD = BOB

Try No	Steps	Data	Expected Results	Actual Results
1	Enter User Name, and press the LOGIN Button	User Name= User01	Should Display Error Message Box "Please Enter User name and Password"	
2	Enter Password and press the LOGIN Button	Password= BOB	Should Display Error Message Box "Please Enter User name and Password"	
3	Enter user Name and Password and press the LOGIN Button	USER = User01 AND Password = XYZ	Should Display Error Message Box "Please Enter User name and Password"	
4	Enter user Name and Password and press the LOGIN Button	USER = XYX AND Password = BOB	Should Display Error Message Box "Please Enter User name and Password"	
5	Enter user Name and Password and press the LOGIN Button	USER = XYZ AND Password = XYZ	Should Display Error Message Box "Please Enter User name and Passw ord"	
6	Enter user Name and Password and press the LOGIN Button	USER =" " AND Password = " "	Should Display Error Message Box "Please Enter User name and Password"	
7	Enter User Name and Password and press the LOGIN Button	USER = User01 AND Password = BOB	Should navigate user directly into the application.	

Table A-B1: Invisible Application Login Test Case

Test Case Number: 2 (Application Login)

Module: Interface Login Module
Functional Specification: User Authentication
Test Objective: To check whether the entered User name and Password is valid or invalid and if valid whether they match with the user type (Admin, Editor, or Guest) that is set.
Assumptions: User is on the login page for the TerraUser Application → outside USGS https://terrauser.wr.usgs.gov/jsp/servlet/Login, inside USGS http://192.168.253.11/jsp/servlet/Login
Test Data:

USER Name = user01, PASSWORD = bob, and User Type = Editor
USER Name = guest01, PASSWORD = guest01, and User Type = Guest
USER Name = admin01, PASSWORD = max, and User Type = Admin

Try No	Steps	Data	Expected Results	Actual Results
1	Enter User Name, set user type as 'Editor' and press the LOGIN Button	User Name= User01 User Type= Editor	Should Display Error Message Box "Please Enter User name and Password"	
2	Enter Password, set user type as 'Editor' and press the LOGIN Button	Password= BOB User Type= Editor	Should Display Error Message Box "Please Enter User name and Password"	
3	Enter user Name, Password, set user type as 'Editor' and press the LOGIN Button	USER = User01 AND Password = XYZ User Type= Editor	Should Display Error Message Box "Please Enter User name and Password"	
4	Enter user Name, Password, set user type as 'Editor' and press the LOGIN Button	USER = XYX AND Password = BOB User Type= Editor	Should Display Error Message Box "Please Enter User name and Password"	
5	Enter user Name, Password, set user type as 'Editor' and press the LOGIN Button	USER = XYZ AND Password = XYZ User Type= Editor	Should Display Error Message Box "Please Enter User name and Password"	
6	Enter user Name, Password, set user type as 'Editor' and press the LOGIN Button	USER =" " AND Password = " " User Type= Editor	Should Display Error Message Box "Please Enter User name and Password"	
7	Enter User Name, Password, set user type as 'Editor' and press the LOGIN Button	USER = User01 AND Password = BOB User Type= Editor	Should navigate to top level User Interface page.	
8	Enter User Name, Password, set user type as 'Admin' and press the LOGIN Button	USER = User01 AND Password = BOB User Type= Admin	Should Display Error Message Box "Please Enter User name and Password"	
9	Enter User Name, Password, set user type as 'Guest' and press the LOGIN Button	USER = Guest01 AND Password = KIM User Type= Guest	Should navigate to top level User Interface page.	
10	Enter User Name, Password, set user type as 'Admin' and press the LOGIN Button	USER = Admin01 AND Password = MAX User Type= Admin	Should navigate to top level Admin Interface page.	
11				Try a valid user name or password with embed spaces.
12				Try a username or password that is the max length

Table A-B2: Application Login Test Case

Test Case Number: 3 (Basic User Navigation)

Module: Basic User (Common & Extended Function Modules) Module
Functional Specification: User Navigation
Test Objective: To check whether the functions on the main user page function as expected.
Assumptions: User is already logged in to the TerraUser Application as an Editor or a Guest as their user type and is currently on the main user page.
Test Data: USER Name = user01, PASSWORD = bob, and User Type = Editor
USER Name = guest01, PASSWORD = guest01, and User Type = Guest

Try No	Steps	Data	Expected Results	Actual Results
1	Click on the 'Email' link	User01 (Editor)	Should be directed to the Email page (with session id).	
2	Click on the 'Email' link	Guest01 (Guest)	Should be directed to the Email page (with session id).	
3	Click on the 'Search' link	User01 (Editor)	Should be directed to the Email page (with session id).	
4	Click on the 'Search' link	Guest01 (Guest)	Should be directed to the Email page (with session id).	
5	Click on the 'Applications' link	User01 (Editor)	Should be directed to the Applications page (with session id).	
6	Click on the 'Applications' link	Guest01 (Guest)	Link should not be in the page.	
7	Click on 'Preferences' link	User01 (Editor)	Should be directed to the Preferences page (with session id).	
8	Click on 'Preferences' link	Guest01 (Guest)	Link should not be in the page.	
	Click on 'Change Password' link	User01 (Editor)	Should be directed to the Change Password page. (With session id)	
5	Click on 'Change Password' link	Guest01 (Guest)	Link should not be in the page.	
6	Click on 'Sign Off'	User01 and Guest01	Should sign off user, redirect to Terrauser Login page.	
7	Try to go to user main page without login in.		Error Message, redirect to login page	

Table A-B3: Basic User Navigation Test Cases

Test Case Number: 4 (Link to Applications)

Module: Links to Applications Module
Functional Specification: User Navigation
Test Objective: To check whether the functions on the Application page function as expected.
Assumptions: User is already logged in to the TerraUser Application as an Editor as their user type and is currently on the Application page.
Test Data: USER Name = user01, PASSWORD = bob → Team01
USER Name = user02, PASSWORD = kel → Team02
Team01 has an access to TerraData, Photo Archive, and MauiCam.
Team02 has an access to TerraData only.

Try No	Steps	Data	Expected Results	Actual Results
1	Click on the 'TerraData Application' link as User01	User01 (should have an access rights to see this in the link lists.)	Should be directed to the TerraData Application page. (Both User01 and User02 should have an access.)	
2	Click on the 'TerraData Application' link as User02	User02 (should have an access rights to see this in the link lists.)	Should be directed to the TerraData Application page. (Both User01 and User02 should have an access.)	
3	Click on the 'Photo Archive Application' link as User01	User01 (User should have an access rights to see this in the link lists)	Should be directed to the Photo Archive Application page. Only User01 should see this link.	
4	Click on the 'Photo Archive Application' link as User02	User02 (User should not have an access rights to see this in the link lists)	Should not see this link.	
5	Click on the 'MauiCam Application' link	User01 (User should have an access rights to see this in the link lists)	Should be directed to the MauiCam Application page. Only User01 should see this link.	
6	Click on the 'MauiCam Application' link	User02 (User should not have an access rights to see this in the link lists)	Should not see this link.	
7	Try to go to Application page without login in.		Error Message, redirect to login page.	

Table A-B4: Link to Applications Test Cases

Test Case Number: 5 (Setting up Preference)

Module: Preference Module
Functional Specification: User Navigation
Test Objective: To check whether the functions on the Preference page function as expected.
Assumptions: User is already logged in to the TerraUser Application as an Editor as their user type and is currently on the Preference page.
Test Data: USER Name = User01, PASSWORD = BOB, and User Type = Editor

Try No	Steps	Data	Expected Results	Actual Results
1	Select Background and click 'Change Preference'	User01	User's information in DB changes. Confirmation result of change (background) w ill be displayed.	
2	Select Font Type and click 'Change Preference'	User01	User's information in DB changes. Confirmation result of change (font type) will be displayed.	
3	Select Font Color and click 'Change Preference'	User01	User's information in DB changes. Confirmation result of change (font color) will be displayed.	
4	Select Font Size and click 'Change Preference'	User01	User's information in DB changes. Confirmation result of change (font size) will be displayed.	
5	Select Font Size and Font Color and click 'Change Preference'	User01	User's information in DB changes. Confirmation results of change (font size and font color) will be displayed.	
6	Select Background, Font Type, Font Color, and Font Size and Click 'Change Preference'	User01	User's information in DB changes. Confirmation results of all changes will be displayed.	
7	Try to go to Application page without login in.		Error Message, redirect to login page.	

Table A-B5: User Preference Test Cases

Test Case Number: 6 (Search)

Module: Search Module
Functional Specification: User Navigation
Test Objective: To check whether the functions on the Search page function as expected.
Assumptions: User is already logged in to the TerraUser Application as an Editor or Guest as their user type and is currently on the Search page.
Test Data: USER Name = User01, PASSWORD = BOB, and User Type = Editor
USER Name = Guest01, PASSWORD = KIM, and User Type = Guest

Try No	Steps	Data	Expected Results	Actual Results
1	Search with Keyword within TerraUser	User01	Search results from TerraUser should be displayed	
2	Search with Keyword within TerraUser	Guest01	Search results from TerraUser should be displayed	
3	Search with Keyword within Goggle	User01	Search results from Goggle should be displayed	
4	Search with Keyword within Goggle	Guest01	Search results from Goggle should be displayed	
5	Search with Keyword within TerraUser and Goggle	User01	Search results from TerraUser and Goggle should be displayed	
6	Search with Keyword within TerraUser and Goggle	Guest01	Search results from TerraUser and Goggle should be displayed	
7	Click 'Search' without any keyword	Guest01/User01	Error Message, redirect to search page.	
8	Try to go to Search page without login in.		Error Message, redirect to login page.	

Table A-B6: User Search Test Cases

Test Case Number: 7 (Email)

Module: Email Module
Functional Specification: User Navigation
Test Objective: To check whether the functions on the E-mail page function as expected.
Assumptions: User is already logged in to the TerraUser Application as an Editor or Guest as their user type and is currently on the E-mail page.
Test Data: USER Name = User01, PASSWORD = BOB, and User Type = Editor
USER Name = Guest01, PASSWORD = KIM, and User Type = Guest

Try No	Steps	Data	Expected Results	Actual Results
1	Send e-mail as editor with all fields completed correctly.	User01	'from' field should be filled in with the database information. Confirmation page will be displayed.	
2	Send e-mail as guest with all fields completed correctly.	Guest01	Confirmation page will be displayed.	
3	Send e-mail as editor without 'to' field.	User01	Error message, redirect to e-mail page.	
4	Send e-mail as guest without 'to' field.	Guest01	Error message, redirect to e-mail page.	
5	Send e-mail as guest without 'from' field.	Guest01	Error message, redirect to e-mail page.	
6	Try to go to E-mail page without login in.		Error Message, redirect to login page.	

Table A-B7: User E-mail Test Cases

Test Case Number: 6 (Change Password)

Module: Change Password Module
Functional Specification: User Navigation
Test Objective: To check whether the functions on the Change Password page function as expected.
Assumptions: User is already logged in to the TerraUser Application as an Editor or Guest as their user type and is currently on the Search page.
Test Data: USER Name = User01, PASSWORD = BOB, and User Type = Editor

Try No	Steps	Data	Expected Results	Actual Results
1	Click on 'Change Password' with all fields completed correctly.	User01	Confirmation Message.	
2	Click on 'Change Password' without old password.	User01	Error message asking for old password, redirect to change password page.	
3	Click on 'Change Password' without new password	User01	Error message asking for new password, redirect to change password page.	
4	Click on 'Change Password' without retype new password	User01	Error message asking for retyping new password, redirect to change password page.	
5	Click on 'Change Password' with mismatched new password and retype new password.	User01	Error message, telling mismatch, redirect to change password page.	
6	Click on 'Change Password' with wrong old password	User01	Error message, telling mismatch, redirect to change password page.	
7	Click on 'Change Password' when the old password and the new password are same.	User01	Error message, telling need to be changed, redirect to change password page.	
8	Try to go to Change Password page without login in.		Error Message, redirect to login page.	

Table A-B6: User Search Test Cases

Test Case Number: 7 (Administrator Navigation)

Module: Administrator (User/Team management and Administrator task) Module
Functional Specification: Administrator Navigation
Test Objective: To check whether the functions on the main administrator page function as expected.
Assumptions: User is already logged in to the TerraUser Application as an Administrator and is currently on the main administrator page.
Test Data: USER Name = admin01, PASSWORD =max, and User Type = Admin

Try No	Steps	Data	Expected Results	Actual Results
1	Click on the 'Add Team' link under Team Management		Should be directed to the Add Team page	
2	Click on the 'Delete Team' link under Team Management		Should be directed to the Delete Team page	
3	Click on the 'Update Team' link under Team Management		Should be directed to the Update Team page	
4	Click on the 'Find User' link under User Management		Should be directed to the Find User page	
5	Click on the 'Add User' link under User Management		Should be directed to the Add User page	
6	Click on the 'Delete User' link under User Management		Should be directed to the Delete User page	
7	Click on the 'Update User' link under User Management		Should be directed to the Update User page	
8	Click on the ' Search' link under Tasks		Should be directed to the Search page	
9	Click on the ' Post MOTD' link under Tasks		Should be directed to the Post MOTD page	
10	Click on the ' View Logs' link under Tasks		Should be directed to the View Logs page	
11	Click on the ' Log off Users' link under Tasks		Should be directed to the Log off Users page	
12	Click on the ' Email' link under Tasks		Should be directed to the Email page	

Table A-B4: Administrator Navigation Test Cases

Use Case 1: Invisible Application Login

The following use case is used to define optimal paths for an Editor, or a Guest to get logged in to a specific application directly.

Use Case	***Invisible Application Login: Guest, Editor***
Description	This use case describes the interactions that take place when an Editor, or a Guest wants to get logged in directly to an application, in order to gain access to the system.
Scenario	Eddie (an Editor) is a scientist working on some DEM (Digital Elevation Model) data. He would like to use the TerraData application to access the metadata from the images that he had previously pulled from the system. Eddie types in the URL for the TerraData application. He must then enter his username and password in the provided area and hit the 'Login' button. The system will authenticate Eddie and put him directly into the TerraData application.
Actor(s)	Editor or Guest
Assumptions	We assume that Eddie has an active account.
Steps	1. Eddie enters top-level URL for application. 2. Eddie enters username and password. 3. Eddie presses the 'Login' button. 4. System will verify the information. 5. If any information is incorrect, the system will display correct error message and prompt Eddie to correct information. 6. Eddie should now be logged in to application.
Non-Functional	*Performance:* Should take less than 30 seconds to process after hitting 'Log in' *Reliability*: Users shouldn't be able to gain access to application without logging in. Once system is up and running Logins should be available when system is operational. *Frequency:* Must login each time want to access application, user logged out after thirty minuets of inactivity. *Fault Tolerance:* *Priority:*
Issues	

Table A-C1: Use Case Invisible Application Login

Use Case 2: TerraUser Login

The following use case is used to define optimal paths for an Administrator, an Editor, or a Guest to get logged in and gain access to the system.

Use Case	***TerraUser Login: Administrator, Guest, Editor***
Description	This use case describes the interactions that take place when an Administrator, an Editor, or a Guest wants to get logged in, in order to gain access to the system.
Scenario	Abigail is an Administrator. She wants to get logged into the TerraUser system so that she can add some users. Abigail types in the top-level URL for the TerraUser application. She must then enter her username and password in the provided area, select 'Administrator' as the user type and hit the 'login' button. The system will authenticate Abigail and put her directly into the TerraUser application.
Actor(s)	Administrator, Editor, or Guest
Assumptions	We assume that Abigail has an active account.
Steps	1. Abigail enters URL for login page in browser. 2. Abigail enters username and password. 3. Abigail selects 'appropriate user type 'Administrator' from the drop down menu labeled 'Type of User:' 4. Abigail presses the 'Login' button. 5. System will verify the information. 6. If any information is incorrect, the system will display correct error message and prompt Abigail to correct information. 7. If Abigail needs help they can click on the help link. 8. Abigail should now be logged in to appropriate interface.
Non-Functional	*Performance:* Should take less than 30 seconds to process after hitting 'Log in' *Reliability*: Users shouldn't be able to gain access to application without logging in. Once system is up and running Logins should be available when system is operational. *Frequency:* Must login each time want to access application, user logged out after thirty minuets of inactivity. *Fault Tolerance:* *Priority:*
Issues	

Table A-C2: Use Case TerraUser Login

Use Case 3: Changing Password

The following use case is used to define optimal paths for an Editor to be able to change their password.

Use Case	***Change Password: Editor***
Description	This use case describes the interactions that take place when an Editor wants to be able to change his/her password on the system.
Scenario	Eddie is an Editor. He wants to be able to use the Maui cam application. Eddie asks Abigail for access to this system. Abigail, the Administrator adds Eddie as a user to the system, gives him the URL for the TerraUser application, and asks him to login and change his password. Eddie Loges into the system, navigates to the preferences page. Eddie hits the 'Password' button, and changes his password.
Actor(s)	Editor
Assumptions	Precondition: Eddie has logged into the system.
Steps	1. Eddie clicks on the 'Preferences' page link. 2. Eddie clicks on the 'Password' button. 3. Eddie enters current password, and new password twice. 4. Eddie enters the 'update' button. 5. System verifies information, and displays appropriate message. 6. If there is an error in entering the data, the Eddie is prompted to reenter the data.
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	

Table A-C3: Use Case Change Password

Use Case 4: Access to Applications

The following use case is used to define optimal paths for an Editor to be able to access certain web-based applications that have been set by the Administrator. An administrator can give a whole group access to an application or individual users or a combination of both.

Use Case	***Access to Applications: Editor, Guest***
Description	This use case describes the interactions that take place when an Editor wants to access other web-based applications that belong to the system.
Scenario	Eddie a scientist has editor privileges on the TerraData application. Eddie wants to gain access to the TerraData application through the TerraUser interface.
Actor(s)	Editor or Guest
Assumptions	Precondition: a valid Eddie has logged into the system.
Steps	1. Eddie clicks on the 'Application' link on main page 2. System takes editor to page with list of the applications they have access to. 3. Eddie clicks on application they want to use.
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	Implementation issues still awaiting resolution.

Table A-C4: Use Case Access to Applications

Use Case 5: Changing Preference

The following use case is used to define optimal paths for an Editor, or a Guest to be able to change their user preferences on the system (i.e. Change the look and feel of the application).

Use Case	***Change Preferences: Editor, Guest***
Description	This use case describes the interactions that take place when an Editor wants to be able to change his/her preferences on the system.
Scenario	Gus the guest is checking out the system to see how the interface works. He has a hard time reading some of the text on the page so he wants to changes his preferences to make the text bigger.
Actor(s)	Editor or Guest
Assumptions	Precondition: Gus has logged into the system. *Guest preferences will be reset to default upon logout. *Settings are saved for the Editor.
Steps	1. Gus clicks on the 'Preferences' link on the main page. 2. Gus Clicks on dropdown menus to change such things as color, font size, etc… 3. Gus clicks the update button at the bottom of the page. 4. Gus also has the choice to hit the 'default' button to restore default settings.
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	

Table A-C5: Use Case Change Preferences

Use Case 6: Search

The following use case is used to define optimal paths for an Editor, or a Guest to be able to perform a search on the system that will return matching results that they have ownership of, permission to access, or records that are marked public.

Use Case	***Search: Editor, Guest***
Description	This use case describes the interactions that take place when an Editor wants to search the system, the system will only return matching results that are owned by a group they belong to, or are marked public.
Scenario	Gus the guest is really excited to learn all he can about the geology of the Grand Canyon. He wants to do a search and see if he can find any information, so he runs a search.
Actor(s)	Editor or Guest
Assumptions	Precondition: Gus has logged into the system.
Steps	1. Gus clicks on the 'Search' link on the main page. 2. Gus types in search criteria, such as a key word. 3. System verifies the information. 4. System performs the search and returns results that match and are marked public in the database. 5. (For Editor search will also return records that they own or belong to their group). 6. User may click on the links to view data in greater detail.
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	

Table A-C6: Use Case Search

Use Case 7: E-mail

The following use case is used to define optimal paths for an Editor to be able to send an email to the Administrators of the system, to members of a group they belong to, or one or more individuals that are members of the same group that the Editor belongs to.

Use Case	***Email: Administrator, Guest, Editor***
Description	This use case describes the interactions that take place when an Administrator, an Editor, or a Guest wants to send an email through the system.
Scenario	Abigail the administrator would like to send out an email to the members of the DataCruncher group, informing them that they all have access to a new application.
Actor(s)	Administrator, Editor, or Guest
Assumptions	Precondition: Abigail has logged into the system. *An Administrator can send an email to an individual, member of a group they belong to, or to all the Users of the system. *An Editor can send email to an individual, members of a group they belong to, or the Administrator of the system. *A Guest can only send an email to the Administrator of the system.
Steps	1. Abigail clicks on the 'email' link on the main page. 2. Admin Abigail (also Editors) specifies recipients in 'to:' field 3. Abigail fills in the subject field, and their message in the main text box. 4. Abigail clicks on the 'mail' button. 5. System sends email message and displays appropriate response to the screen.
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	

Table A-C7: Use Case Email

Use Case 8: Add User

The following use case is used to define optimal paths for an Administrator to be able to add a new user to the system.

Use Case	***Add User: Administrator***
Description	This use case describes the interactions that take place when an Administrator wants to add a new user to the system.
Scenario	Abigail the administrator has receives a request from Molly that a new student intern has joined her group and needs to be added. Molly specifies the permissions that the student, Carla will require. Abigail adds Carla as a user and sends an email to Carla and Molly.
Actor(s)	Administrator
Assumptions	Precondition: Abigail has logged into the system.
Steps	1. Abigail clicks on 'Add User' Link on the administrator interface main page. 2. Abigail types in Carla's name, contact info, group, etc., etc… 3. Abigail will select 'Submit' button 4. System will verify the information. 5. If required information is missing, the system will prompt for correction. 6. System will notify if Carla's account has been successfully created. 7. If Abigail already exists, error message will be displayed. 8. If Abigail wants to create another user they may hit the 'reset' button.
Non-Functional	*Performance:* Should take less than 30 seconds to process after hitting 'submit' *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	Not decided if application will first do a check to see if user already exists and display search results??

Table A-C8: Use Case Add User

Use Case 9: Delete User

The following use case is used to define optimal paths for an Administrator to be able to delete an existing user from the system.

Use Case	***Delete User: Administrator***
Description	This use case describes the interactions that take place when an Administrator wants to delete an existing user from the system.
Scenario	Abigail the administrator receives and email from manager Molly that her top programmer Zed is leaving the Survey for Greener pastures. Molly sends an email to Abigail requesting that Zed's account be deleted. Abigail deletes Zed's account and sends an email to Molly.
Actor(s)	Administrator
Assumptions	Precondition: Abigail has logged into the system.
Steps	1. Abigail clicks on the 'User Search' link. 2. System displays the 'search' page. 3. Abigail enters search criteria into one of the following search fields (Zed's first name, Last Name, user ID, group, etc, etc…) 4. System displays a table on the 'search Results' page. 5. Abigail clicks on the 'delete' button next to the selected user. 6. Abigail is prompted if they really want to delete Zed 'YES or NO'. 7. System displays appropriate response to the previous question. 8. System should note in the log file which user got deleted.
Non-Functional	*Performance:* Should take less than 30 seconds to process after hitting 'yes' Should not return more than 100 search results, if more system will ask user to refine their search. *Reliability*: Should work every time, and only delete the user that is specified. *Frequency:* *Fault Tolerance:* *Priority:*
Issues	

Table A-C9: Use Case Delete User

Use Case 10: Update User Information

The following use case is used to define optimal paths for an Administrator to be able to update an existing user's information in the system.

Use Case	***Update User Information: Administrator***
Description	This use case describes the interactions that take place when an Administrator wants to update an existing user's information on the system.
Scenario	Abigail the administrator receives an email from Molly that her student Carla has graduated and promoted to a full time position within the group. Carla needs some more permissions changes. Abigail completes the request and sends an email back to Molly and Carla.
Actor(s)	Administrator
Assumptions	Precondition: Abigail has logged into the system.
Steps	1. Abigail clicks on the 'User Search' link. 2. System displays the 'Search' page. 3. Abigail enters search criteria into one of the following search fields (Carla's first name, Last Name, user ID, group, etc, etc…) 4. System displays a table on the 'Search Results'. 5. Abigail clicks on the 'update' button next to the selected user. 6. System displays 'Update User' page with the fields filled in with the known information. 7. Abigail modifies necessary fields. 8. Abigail clicks on the 'update' button located at the bottom of the page. 9. System will verify information. 10. If invalid information is entered the Abigail is prompted to correct this information. 11. System displays appropriate message to screen.
Non-Functional	*Performance:* Should take less than 30 seconds to process after hitting 'update' Should not return more than 100 search results, if more system will ask user to refine their search. *Reliability*: Should work every time, and only update the user's information that was specified. *Frequency:* *Fault Tolerance:* *Priority:*
Issues	

Table A-C10: Use Case Update User Information

Use Case 11: Password Reset/Expire

The following use case is used to define optimal paths for an Administrator to be able to reset a users password, or set the password to expire at some specified date.

Use Case	***Password Reset/Expire: Administrator***
Description	This use case describes the interactions that take place when an Administrator wants to reset a user's password, or have a user's password expire.
Scenario	It has been a while since Eddie (an Editor) the scientist has worked with the Terra applications and he has forgotten his password. Eddie asks Abigail the administrator to please reset his password. Abigail goes into the admin interface, resets Eddie's password, then notifies Eddie that he can logon with this new password and change it to a new one.
Actor(s)	Administrator
Assumptions	Precondition: Abigail has logged into the system.
Steps	1. Abigail clicks on the 'search' link on the main page. 2. Abigail performs search to find desired user. 3. System displays search results page. 4. Abigail Clicks on 'Password' button next to desired user's name. 5. Abigail is promotes to change the password, or have it expire. 6. System verifies information, and displays appropriate error message.
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	

Table A-C11: Use Case Password reset/expire

Use Case 12: Add Team

The following use case is used to define optimal paths for an Administrator to be able to add a team to the system.

Use Case	***Add Team: Administrator***
Description	This use case describes the interactions that take place when an Administrator wants to add a team to the system.
Scenario	Abigail the administrator has received a request via email that manager Molly needs to create a special group within her team to work an a special project. She includes in the email the new team name, the team members and special team information. Abigail will go into the administrator interface, add the team, then send an email to Molly confirming that the request was completed.
Actor(s)	Administrator
Assumptions	Precondition: Abigail has logged into the system.
Steps	1. Abigail clicks on the 'team' link on the main page. 2. Abigail clicks on 'Add Team' link. 3. Abigail prompted to enter the team information. 4. Abigail click the 'Add' button 5. The system verifies the information 6. If any information is incomplete the system prompts the Abigail to correct it. 7. System verifies information and displays correct message to screen.
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	

Table A-C12: Use Case Add Team

Use Case 13: Update Team Information

The following use case is used to define optimal paths for an Administrator to be able to update team information.

Use Case	***Update Team Information: Administrator***
Description	This use case describes the interactions that take place when an Administrator wants to update a team's information.
Scenario	Abigail the administrator has received an email from a manager named Frank, noting that frank has recently acquired two new members into his group. Frank would like his new group members added to his group. Abigail must go in and update the team information.
Actor(s)	Administrator
Assumptions	Precondition: Abigail has logged into the system.
Steps	1. Abigail clicks on the 'Team' link on the main page. 2. Abigail clicks on the team name, which is a link to the team info page. 3. Abigail updates information 4. Abigail clicks the 'update' button at the bottom of the page. 5. System verifies information. 6. System prompts Abigail to fill in missing or incorrect and displays correct message to screen.
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	

Table A-C13: Use Case Update Team Information

Use Case 14: Delete Team

The following use case is used to define optimal paths for an Administrator to be able to delete a team from the system.

Use Case	***Delete Team: Administrator***
Description	This use case describes the interactions that take place when an Administrator wants to delete a team from the system.
Scenario	Abigail the administrator is performing her weekly maintenance tasks on the system. She receives an email from Molly a manager of the wespt group, notifying Abigail that her group has been absorbed into other groups and no longer exists. Group wespt can be deleted.
Actor(s)	Administrator
Assumptions	Precondition: Abigail has logged into the system.
Steps	1. Abigail clicks on the 'team' link on the main page. 2. Abigail clicks on the 'delete' button to the right of the team name. 3. Abigail is prompted with a 'yes' or 'no' question do they really want to do this. 4. System processes request and displays appropriate response on screen.
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	

Table A-C14: Use Case Delete Team

Use Case 15: Log Off Users

The following use case is used to define optimal paths for an Administrator to be able to log off all currently connected users and display an appropriate message when the users try to log back in.

Use Case	***Log off Users: Administrator***
Description	This use case describes the interactions that take place when an Administrator wants to log off all currently connected users from the system.
Scenario	Abigail the administrator would like to log out all the users that are currently logged in so that she can perform some maintenance on the system.
Actor(s)	Administrator
Assumptions	Precondition: Abigail has logged into the system.
Steps	1. Abigail clicks on 'Active Users/Logs' Link on the administrator interface main page. 2. System shows a dynamic page displaying users currently logged in. 3. User clicks on the 'Log users out' button 4. System kicks all users off except Abigail, and displays appropriate message to screen. Page is dynamic and should update with the number of users logged in.
Non-Functional	*Performance:* All users should be logged out within 30 seconds. *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	We will need to consider blocking users from logging in again, until lock released?? And what about a message to be posted on the login page also.

Table A-C15: Use Case Log off Users

Use Case 16: Post MOTD (Message of the Day)

The following use case is used to define optimal paths for an Administrator to be able to post a Message of the Day that appears on the Login page.

Use Case	***Post MOTD (Message of the day): Administrator***
Description	This use case describes the interactions that take place when an Administrator wants to post a MOTD (Message of the Day) that will appear on the Login page.
Scenario	Abigail the administrator would like to notify users that the system will be off line for maintenance over the weekend. She will do this by posting a message on the login page.
Actor(s)	Administrator
Assumptions	Precondition: Abigail has logged into the system. Whatever text or characters are in the text box when the post is made will appear as the MOTD on the login page.
Steps	1. Abigail clicks on 'MOTD' link on the Administrator Interface main page. 2. Abigail enters a MOTD in the text box, or modifies message that is in this field. 3. Abigail clicks on the 'update MOTD' button. 4. System displays appropriate message.
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	

Table A-C16: Use Case Post MOTD

Use Case 17: View Active Users and Logs

The following use case is used to define optimal paths for an Administrator to be able to view which users are currently logged in and view the log files.

Use Case	***View Active Users and Logs: Administrator***
Description	This use case describes the interactions that take place when an Administrator wants to view how many users are logged in, or view the log files.
Scenario	Abigail the administrator would like to see how many users are currently logged on to the system. Then she would like to view the web logs.
Actor(s)	Administrator
Assumptions	Precondition: Abigail has logged into the system.
Steps	1. Abigail clicks on the 'System Status' link on the main page. 2. System displays dynamic page that shows the number of users that are logged in. 3. To view the log files, Abigail clicks on the 'Log' button. 4. System displays log file on screen, or appropriate message.
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	

Table A-C17: Use Case View Active Users and Logs

Use Case 18: Administrator Search

The following use case is used to define optimal paths for an Administrator to be able to search the database for user and team information.

Use Case	***Administrator Search***
Description	This use case describes the interactions that take place when an Administrator wants to search for users, or groups and have the system return matching results.
Scenario	Abigail the administrator would like to search to see if a user currently exists in the database, so she can update their settings.
Actor(s)	Administrator
Assumptions	Precondition: Abigail has logged into the system.
Steps	1. Abigail clicks on the 'Search' link located on main page of Admin interface. 2. Abigail enters search criteria. 3. Abigail clicks the 'Search' button. 4. System verifies information and prompts Abigail for corrections if necessary. 5. System displays results page.
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	Should the number of results that it returns limit the search???

Table A-C18: Use Case Admin Search

Use Case 19: Add Information Fields

The following use case is used to define optimal paths for an Administrator to be able to add new fields or information to the user database.

Use Case	***Add Information Fields***
Description	This use case describes the interactions that take place when an Administrator wants to add user information fields.
Scenario	Abigail the Administrator receives an email request from Dana the application developer for the Photo Archive. Dana would like the TerraUser application to store information on whether the user exited the photo Archive using view A, B, or C. This information will be kept track of in the TerraUser database, and used in the future for other application designs. Abigail navigates to the Add information pave, fills out the form and hits 'submit'. Abigail then sends an email to Dana with the application communication details.
Actor(s)	Administrator
Assumptions	Precondition: Abigail has logged into the system.
Steps	1. Abigail clicks on the 'Add Info' link on the main page. 2. ??? Issues here that have not yet been resolved by design team!!!
Non-Functional	*Performance:* *Reliability*: *Frequency:* *Fault Tolerance:* *Priority:*
Issues	Implementation issues still awaiting resolution.

Table A-C19: Use Case Add Information Fields

A. Overall

			0	1	2	3	4	5		N/A
1	Overall system satisfactory	terrible	☐	☐	☐	☐	☐	☐	wonderful	☐
2	Easy of use - simplicity	difficult	☐	☐	☐	☐	☐	☐	easy	☐
3	Powerful / effective	inadequate	☐	☐	☐	☐	☐	☐	adequate	☐
4	Flexibility	rigid	☐	☐	☐	☐	☐	☐	flexible	☐
5	The system has all functions and capabilities you expect it to have.	frustrating	☐	☐	☐	☐	☐	☐	satisfying	

B. Screen

			0	1	2	3	4	5		N/A
1	Reading characters on the screen	hard	☐	☐	☐	☐	☐	☐	easy	☐
2	Highlighting simplified task	not all	☐	☐	☐	☐	☐	☐	very much	☐
3	Organization of information	confusing	☐	☐	☐	☐	☐	☐	very clear	☐
4	Sequence of screens	confusing	☐	☐	☐	☐	☐	☐	very clear	☐
5	Consistent design throughout the interface	inconsistent	☐	☐	☐	☐	☐	☐	consistent	

C. Adapting to the User

			0	1	2	3	4	5		N/A
1	Use of terms throughout system	inconsistent	☐	☐	☐	☐	☐	☐	consistent	☐
2	Terminology related to task	never	☐	☐	☐	☐	☐	☐	always	☐
3	Positioning messages on screen	inconsistent	☐	☐	☐	☐	☐	☐	consistent	☐
4	Prompts for input	confusing	☐	☐	☐	☐	☐	☐	very clear	☐
5	Feel comfortable using the system	uncomfortable	☐	☐	☐	☐	☐	☐	comfortable	☐
6	Easy to find the information you needed	difficult	☐	☐	☐	☐	☐	☐	easy	☐
7	Easy to find what is not possible	never	☐	☐	☐	☐	☐	☐	always	☐

D. Feedback and Errors

			0	1	2	3	4	5		N/A
1	Computer informs about its progress	never	☐	☐	☐	☐	☐	☐	always	☐
2	Confirmation messages	never	☐	☐	☐	☐	☐	☐	always	☐
3	Clear error messages	never	☐	☐	☐	☐	☐	☐	always	☐
4	Timely feedback	never	☐	☐	☐	☐	☐	☐	always	☐

E. Learning

			0	1	2	3	4	5		N/A
1	Learning to use the system	difficult	☐	☐	☐	☐	☐	☐	easy	☐
2	Performing tasks is straightforward	never	☐	☐	☐	☐	☐	☐	always	☐
3	Help message on the screen	unhelpful	☐	☐	☐	☐	☐	☐	helpful	☐

F. System Capabilities

			0	1	2	3	4	5		N/A
1	System speed	too slow	☐	☐	☐	☐	☐	☐	fast enough	☐
2	System reliability	unreliable	☐	☐	☐	☐	☐	☐	reliable	☐
3	Correcting your mistakes	difficult	☐	☐	☐	☐	☐	☐	easy	☐
4	Designed for all levels of users	never	☐	☐	☐	☐	☐	☐	always	☐

G. Aspects

List the most **negative** aspect(s):

1. _____
2. _____
3. _____

List the most **positive** aspect(s):

1. _____
2. _____
3. _____