Transmittal Memo

Date: June 11, 2011
To: Patrick Jay, Manager, Vice President, Accounting Group at Bellevue
From: Robot Rebellion Regime
Subject: System Testing/Test Plan
 This report is to clarify the testing procedures used on the system in development. The methods we will be using will be hands on testing by the development team and user acceptance testing. We will be looking at and testing each table and attribute in form view to ensure it will do what it is supposed to do accurately and efficiently. Fields will be tested with test data that is accurate and data that is inaccurate to test field data constraints and integrity. Valid and invalid characters will be used as appropriate.
The benefits of using the development team to test the system are:
- The development team has a vested interest in putting out a quality product.
- That any problems encountered by the development team can be addressed at the time of testing or shortly thereafter.

The benefits of testing end user acceptance are numerous. User acceptance is the key to long term success of any system put in place. Also, end users can suggest improvements or modifications to the system design while testing. It also provides the view of persons who are not on the development team, offering an objective viewpoint.
 Included in this report are:
- Chris Clemente – Testing Coordinator, backup to Doug Vickers
- Terry Gorle – Project Manager, backup to Chris Clemente

If you have if you have any questions, concerns or suggestions, please feel free to contact any member of the team at
RobotRebellionRegime@example.com or 206-555-1234

Test Plan

June 11, 2011

Robot Rebellion Regime

Adam McCormack, Terry Gorle, Doug Vickers, Chris Clemente and Eric Nelson

Contents

Introduction

This report will describe the testing methods and procedures used to check the system for accuracy, ease of use, aesthetic value, and to ensure it meets the business needs. The methods we will be using will be hands on testing by the development team and user acceptance testing. This will be performed in stages; the first of which is *alpha testing* using data that is made up. The second phase is *beta testing*, using real data. We will be examining and testing each table and attribute to ensure it will accept and retrieve data accurately and efficiently. Fields will be tested with test data that is accurate and inaccurate to test constraints and integrity. Valid and invalid characters will be used as appropriate, such as checking invoice and contract dates with inaccurate data to warn the user of the error and prompt for accurate data, or checking cost overruns by inputting more than a contractor is allowed on their Fee Maximum.

Other testing will be used for different reasons. The systems code will be developed and tested by the developers to ensure the desired result is achieved error free. Unit testing will tell us whether the module and program performs as specified. Any misinterpretation errors will be corrected at this time. Integration testing will ensure that each part of the program works efficiently with each other without errors. Graphical User Interfaces will be tested to ensure proper functionality. Data flow and exchange will also be tested.

Back to contents

Testing Approach

Unit Testing will first check the forms, reports and queries for accuracy and field/data constraints as appropriate. Real and false data will be used to evaluate whether the system responds to bad data in a way that is logical to the user, (Asks or prompts the user to input the correct data type or in the correct format). Tables and attributes will be tested for accuracy, data integrity, and data constraints. Table and attribute relations will also be looked at. All code will be tested by the programmers and any errors will be

identified and fixed. This testing is done after the developer has tested the underlying code and believes it to be error free.

Integration Testing will check whether all parts of the system work together error free. Any links and interfaces between program modules will be checked for proper function. Data flow, control, and exchange are tested and repaired if necessary. User interfaces, use case scenarios data flow, and system interfaces are tested in this section.

System testing will be done by the development team to ensure all parts of the system work together without error. This includes meeting business needs, security, and how well the system responds under a heavy load.

Acceptance Testing will be performed by the end users after system testing is completed. The system will be tested for completeness, accuracy, user acceptance, and that it fulfills all the business requirements. Generic data will be used to ensure and verify the accuracy of reports and queries. The system will be monitored during this testing by the development team, who will be looking for any errors, missed requirements, and for user acceptance. The overall benefits of testing this way will produce an error free system. The business will be impacted slightly as user testing is performed during business hours, but every effort will be made to keep this to a minimum.

Back to contents

Test Environment

Establishing the proper test environment and training procedures is critical to the success of implementing and operating within the database system.

Training Plan

The personnel required to access the database system will be trained in accordance with their authorization level to provide security and optimize the specific training and testing procedures.

RRR's database specialists will initially provide in-depth system training for administrative access to Patrick Jay and Dave Spencer. Training will follow for specific personnel within the Accounting, Contract, relevant departments, and IT Manager(s). Specific training will be provided in the following areas:

- System overview
- System Interface
- Login/Pass Authorizations
- Database navigation and functions
- Data entry and validations
- Operational processes and specialized business features

Training for Patrick Jay and Dave Spencer, providing the benefit of having a two bankers with admin privileges on site in the event one is absent, will be the most extensive, with an estimated training duration of two hours. RRR estimates that other personnel training will be completed in a 90-minute session. Patrick Jay will be responsible as ongoing primary support to branch personnel as well as training future personnel based on their permission levels and job duties.

Test Data

Test data will consist of using valid and invalid data in forms and fields to analyze proper validation, coding, and function. The database will undergo stress testing, function testing, and abnormal procedure testing. A special test database including erroneous data will be loaded beforehand, and user input of erroneous data in real time will also be tested and evaluated. Examples of the test data are included in the appendix on page 65.

Operational Environment

The hardware and software to be implemented and utilized within the database training and testing procedures will be configured for optimal performance, providing an identical environment to the branch location.

- **Hardware Components**
    a. System Test Team Workstations
    b. System Testing Server
    c. LAN Network Configuration

- **Software Components**
    a. Microsoft Windows Server 2008 Enterprise
    b. Microsoft Office 2010 Professional Suite
        i. Microsoft Access 2010 (primary DBMS application)
    c. Documentation Tools and Test Data
    d. Configuration Management Tools

At the time of testing and implementation of the system database, existing infrastructure duplicating the branch system within this project proposal will be utilized. No additional hardware or software will be provided, purchased, or evaluated by RRR analysts or the Bank of Xanadu for testing purposes. However, this system provides for future system expansion and modifications that may require additional hardware or software components to implement regional or international networking.

Back to contents

List of Test Cases

RRR has compiled a list of test cases for testing each screen of the system as well as navigation and interaction. Through this testing process, RRR will ensure the system will meet the system design specifications as well as the requirements of the Bank of Xanadu.

Test	Description	Condition Performed	Level	Objective
TC001	Create a new contract	New contract exists	Unit	The verified contract has been entered into the system and is ready to have valid invoices processed against it.
TC002	Add new bank info	The existance of a contract with new vendor, contact, unit, or division information.	Unit	The new vendor, contact, unit, or division information has been entered into the system.
TC003	A contract exception	A contract has been received that has missing or invalid information.	Unit	An error has been found and the incomplete or invalid contract has been returned to the Buyer
TC004	Update a contract	An updated contract has been received from the Buyer	Unit	A complete and valid contract has been updated in the system
TC005	Receive an invoice	An invoice has been received for contract programming services	Unit	The invoice has been verified for payment and entered into the system
TC006	Invoice exception	An invoice has been received that is either incomplete, invalid, or otherwise unpayable.	Unit	An error has been found and the incomplete, invalid, or otherwise unpayable invoice has been returned to the Buyer.
TC007	Update an invoice	A revised invoice and a contract extension has been received from the Buyer	Unit	The updated or new invoice has been entered into the system with the "Approved to Pay" status and the contract extension information has been entered into the system and the contract has been updated
TC008	Invoice status inquiry	The Vendor has sent an invoice for programming services and has not received payment according to the terms of the invoice	Unit	The Vendor has been updated on the current payment status of the invoice and, if applicable, a copy of the invoice has been sent with an exception memo to the Buyer for resolution

Test	Description	Condition Performed	Level	Objective
TC009	Pay invoice	An invoice is approved for payment and is ready to be sent to A/P to have a check cut.	Unit	The invoice has been sent to the A/P department with a Data Entry Sheet attached.
TC010	Accrue invoice	An invoice has been received and entered into the system with either a "Do Not Pay" status OR an invoice has been processed for payment after the cut-off date for the last A/P checkrun for the current month and before the 6th day of the following month.	Unit	The unpaid (no check cut) invoice has been designated as accrued.
TC011	Run accounting reports	It is time to generate the monthly Accounting department reports	System	The monthly Accounting department reports have are deliverable to the Accounting Manager
TC012	Run management reports	It is time to generate the monthly Bank Management reports.	System	The monthly Bank Management reports are ready tosend to the various bank units.
TC013	User log on	System booted up and ready	Unit	The user is able to log on successfully
TC014	Navigate from screen to screen	System is logged into	Unit	The user is able to navigate from screen to screen

Back to contents

Sample Test Data

Entity	Attribute	Valid data	Invalid data
Acc_Employee	Employee ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	First Name	John, Pete, Stanley, Paul, David, any valid alpha characters a-z	Jo9n, Pete!, Sta@ne(, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Last Name	Smith, Gonzales, Hill, Farrar, Wall, any valid alpha characters a-z	Sm8th, Gonz@les, Farr*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Phone Number	any valid combo of 1234567890, -, () in the format (xxx) xxx-xxxx	alpha characters a-z ~! @#$%^&*_+={}| []\<>?:";\, or any combination of these characters
	Acc_Group	any valid alpha characters a-z	*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Position	any valid alpha characters a-z	*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Acc_Employee_Passw ord	any valid alpha characters a-z and/or any valid combo of 1234567890	*(r,~!@#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Acc_Employee_Questi on	any valid alpha characters a-z and ?	*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Acc_Employee_Answe r	any valid alpha characters a-z and/or any valid combo of 1234567890	*(r,~!@#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Acc_Employee_Status	Locked or unlocked	Any character that does not spell locked or unlocked
	Invoice_ID	Manager, Employee, Terminated	Any character that does not spell Manager, Employee, or Terminated
Entity	**Attribute**	**Valid data**	**Invalid data**
Bank_Contact	Contact ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}|

			[]\<>?:";\, or any combination of these characters
	Title	any valid alpha characters a-z	*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	First Name	John, Pete, Stanley, Paul, David, any valid alpha characters a-z	Jo9n, Pete!, Sta@ne(, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Last Name	Smith, Gonzales, Hill, Farrar, Wall, any valid alpha characters a-z	Sm8th, Gonz@les, Farr*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Phone Number	any valid combo of 1234567890, -, () in the format (xxx) xxx-xxxx	alpha characters a-z ~! @#$%^&*_+={}| []\<>?:";\, or any combination of these characters
	Email Address	john@gds.com, jd51@abc.org.ca, or any valid combo of 1234567890 and a-z and must include an @ symbol	Johngds.com, john@#gd.com* ~!#$%^&*()+={}][|?><, or any combo of these characters.
	Unit ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters

Entity	Attribute	Valid data	Invalid data
Division	Division ID	any valid combo of 1234567890	alpha characters a-z ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Division_Name	any valid alpha characters a-z	*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters

Entity	Attribute	Valid data	Invalid data
Contract	Contract ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters

	Field	Valid	Invalid
	Start Date	any valid combo of 1234567890 and / in the format x/x/xxxx or xx/xx/xxxx	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	End Date	any valid combo of 1234567890 and / in the format x/x/xxxx or xx/xx/xxxx	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Fee Max	any valid combo of 1234567890 with $, . in then format $xx,xxx.xx	alpha characters a-z ~!@# %^&*()_+={}|[]\<>?:";\ or any combination of these characters
	Description	any valid alpha characters a-z	*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Cont_Rate_Pay	any valid combo of 1234567890 with $, . in then format $xx,xxx.xx	alpha characters a-z ~!@# %^&*()_+={}|[]\<>?:";\ or any combination of these characters
	Contract_Status	any valid alpha characters a-z	*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Unit ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Vendor ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Programmer ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Acc_Employee_ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Contact_ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these

			characters
	Buyer_ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Division_ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters

Entity	Attribute	Valid data	Invalid data
Buyer	Buyer ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Buyer_FName	John, Pete, Stanley, Paul, David, any valid alpha characters a-z	Jo9n, Pete!, Sta@ne(, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Buyer_LName	Smith, Gonzales, Hill, Farrar, Wall, any valid alpha characters a-z	Sm8th, Gonz@les, Farr*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Contract ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Buyer_Email	john@gds.com, jd51@abc.org.ca, or any valid combo of 1234567890 and a-z and must include an @ symbol	Johngds.com, john@#gd.com* ~!#$%^&*()+={}][|?><, or any combo of these characters.

Entity	Attribute	Valid data	Invalid data
Invoice	Invoice ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Start Date	any valid combo of 1234567890 and / in the format x/x/xxxx or	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any

		xx/xx/xxxx	combination of these characters
	End Date	any valid combo of 1234567890 and / in the format x/x/xxxx or xx/xx/xxxx	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Invoice Date	any valid combo of 1234567890 and / in the format x/x/xxxx or xx/xx/xxxx	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Due Date	any valid combo of 1234567890 and / in the format x/x/xxxx or xx/xx/xxxx	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Invoice Terms	any valid alpha characters a-z and/or any valid combo of 1234567890 and /	*(r,~!@#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Hours Worked	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Work Description	any valid alpha characters a-z and/or any valid combo of 1234567890	*(r,~!@#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Invoice_Notes	any valid alpha characters a-z and/or any valid combo of 1234567890	*(r,~!@#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Invoice_Pay_Status	Paid or unpaid	Any character not paid or unpaid
	Invoice_Accrual	any valid combo of 1234567890 and / in the format x/x/xxxx or xx/xx/xxxx	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Invoice_Accrual_Reversed	any valid combo of 1234567890 and / in the format x/x/xxxx or xx/xx/xxxx	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Vendor _ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Programmer_ID	any valid combo of	alpha characters a-z ~!

		1234567890	@#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Contract ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Unit_ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Acc_Employee_ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Buyer_ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters

Entity	Attribute	Valid data	Invalid data
Unit	Unit_ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Unit_Name	any valid alpha characters a-z	*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Division ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters

Entity	Attribute	Valid data	Invalid data
Programmer	programmer ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	First Name	John, Pete, Stanley, Paul, David, any valid alpha	Jo9n, Pete!, Sta@ne(, 1234567890 ~!@#$

		characters a-z	%^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Last Name	Smith, Gonzales, Hill, Farrar, Wall, any valid alpha characters a-z	Sm8th, Gonz@les, Farr*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Phone Number	any valid combo of 1234567890, -, () in the format (xxx) xxx-xxxx	alpha characters a-z ~! @#$%^&*_+={}| []\<>?:";\, or any combination of these characters
	Address 1	any valid alpha characters a-z and/or any valid combo of 1234567890	*(r,~!@#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Address 2	any valid alpha characters a-z and/or any valid combo of 1234567890	*(r,~!@#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	City	any valid alpha characters a-z	*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	State	any valid alpha characters a-z	*(r, 1234567890 ~!@#$ %^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Zip Code	any valid combo of 1234567890 and – in the format xxxxx- or xxxxx-xxxx	*(r,~!@#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Email Address	john@gds.com, jd51@abc.org.ca, or any valid combo of 1234567890 and a-z and must include an @ symbol	Johngds.com, john@#gd.com* ~!#$%^&*()+={}|[|?><, or any combo of these characters.
	Notes	any valid alpha characters a-z and/or any valid combo of 1234567890	*(r,~!@#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Invoice_ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combo of these characters
Entity	**Attribute**	**Valid data**	**Invalid data**
Vendor	Vendor_ID	any valid combo of 1234567890	alpha characters a-z ~! @#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Name	any valid alpha characters a-z and/or any valid combo of 1234567890	*(r,~!@#$%^&*()_+={}| []\<>?:";\, or any combination of these characters
	Phone Number	any valid combo of	alpha characters a-z ~!

15

		1234567890, -, () in the format (xxx) xxx-xxxx	@#$%^&*_+={}|[]\<>?:";\, or any combination of these characters
	Address 1	any valid alpha characters a-z and/or any valid combo of 1234567890	*(r,~!@#$%^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Address 2	any valid alpha characters a-z and/or any valid combo of 1234567890	*(r,~!@#$%^&*()_+={}|[]\<>?:";\, or any combination of these characters
	City	any valid alpha characters a-z	*(r, 1234567890 ~!@#$%^&*()_+={}|[]\<>?:";\, or any combination of these characters
	State	any valid alpha characters a-z	*(r, 1234567890 ~!@#$%^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Zip Code	any valid combo of 1234567890 and – in the format xxxxx- or xxxxx-xxxx	*(r,~!@#$%^&*()_+={}|[]\<>?:";\, or any combination of these characters
	Email Address	john@gds.com, jd51@abc.org.ca, or any valid combo of 1234567890 and a-z and must include an @ symbol	Johngds.com, john@#gd.com* ~!#$%^&*()+={}][|?><, or any combo of these characters.
	Notes	any valid alpha characters a-z and/or any valid combo of 1234567890	*(r,~!@#$%^&*()_+={}|[]\<>?:";\, or any combination of these characters

Back to contents

Appendix

Sample Test Form Template

Scenario ID:	Test Type:	Tester:	Date:
Scenario Name:			
Background and Test Objective(s):			

				Actual Results	/Notes
Related Use Cases or Test Scenarios:					
Testing Notes:					

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01				☐ Pass ☐ Fail	
02				☐ Pass ☐ Fail	
03				☐ Pass ☐ Fail	
04				☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by:_____   Date Conducted:_____

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

   ☐   Passed: No further action required   (Date:_____,

   Initials:_____)

   ☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

   ☐   Errors Corrected   (Date:_____,

Initials:_____)

   ☐   Retested   (Date:_____,

Initials:_____)

   ☐   Other:

Back to contents
Sample Test Data for Use Cases

Accrue Invoice

Scenario ID: Test1	Test Type: Unit	Tester: Chris Clemente	Date: 5/8/2011
Scenario Name:	Accountant processes an invoice Accrual from trigger to completion.		
Background and Test Objective(s):	Dave Spencer and Patrick Jay are accountants and both will run this test. It verifies that correct data gets entered properly and incorrect data prompts the user to enter correct data with a message.		
Related Use Cases or Test Scenarios:	Pay Invoice (extends		
Testing Notes:	Refer to use case Accrue Invoice for step by step directions. Refer to the Excel spread sheet for the list of data used during this test.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	Run an Accrual	Try to run an accrual	Should be able to run it	X Pass ☐ Fail	
02	An incorrect Invoice ID is entered into the Accrual process.	See if the system recognizes the incorrect Invoice ID	Should prompt the user to enter the correct Invoice ID for the vendor	X Pass ☐ Fail	
03	Enter an incorrect date.	Enter a "Date Paid" that is not past the cut-off date for last A/P check run for the current month and after the 6th day of the following month	The result should be "Payment Status" is "Do Not Pay"	X Pass ☐ Fail	
04	Leave required fields empty.	Leave fields that need data empty.	System should give user a Required Data message.	X Pass ☐ Fail	

Testing Notes: None at this time, went smoothly.

Tests Completed by: Chris Clemente_____   Date Conducted: 5/8/2011_____

Reviewed by:_____   Date Reviewed:_____

18

Corrective Actions:

    X   Passed: No further action required   (Date:__5/8/2011___, Initials: <u>CLC</u>

    ☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

    ☐   Errors Corrected   (Date:_____,

Initials:_____)

    ☐   Retested   (Date:_____,

Initials:_____)

    ☐   Other:

Contract Exception

Scenario ID: Test 3	Test Type: Unit	Tester: Chris Clemente	Date: 5/8/2011
Scenario Name:	Contract Exception		
Background and Test Objective(s):	This will test the steps for processing an exception memo to return an incomplete/invalid contract to the Buyer, from the time the incomplete/invalid contract is received until it has been returned to the Buyer.		
Related Use Cases or Test Scenarios:	Receive Contracts (Extends)		
Testing Notes:			

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	Enter a missing piece of information incorrectly in the system.	Use a 1 or nothing in a field that is missing information.	The system should prompt the user to enter either "Missing/Invalid or "zero" into the field.	X Pass ☐ Fail	
02	Enter non numeric data into the date and phone number fields.	Enter letter and symbols into a field requiring numeric data.	Fields should not accept data.	X Pass ☐ Fail	
03				☐ Pass ☐ Fail	
04				☐ Pass ☐ Fail	

Testing Notes:	

Tests Completed by: Chris Clemente_____   Date Conducted:_5/8/2011_____

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Invoice Exception

Scenario ID: Test 4	Test Type: Unit	Tester: Chris Clemente	Date: 5/8(2011
Scenario Name:	Invoice Exception		
Background and Test Objective(s):	This will test the steps for processing an invoice exception memo for an incomplete, invalid or non-payable invoice to the buyer, verifying that incorrectly entered information will be identified and error messages generated.		
Related Use Cases or Test Scenarios:	Receive Invoice (Extends)		
Testing Notes:	Refer to use-case (INVOICE EXCEPTION and list of data in the Excel spread sheet holding the test data.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	Enter an existing Invoice ID.	Enter ID for an invoice already entered in the system.	Should prompt the user that the ID already exists in the system and ask the use if they want to enter a new one.	X Pass ☐ Fail	

02	Leave required fields empty.	Do not put data in fields that cannot be empty.	A message prompting the user that the field cannot be left empty.	X Pass ☐ Fail	
03	Enter data that is inaccurate into the date/phone number fields.	Enter letters into fields requiring numeric data.	A message should prompt the user to enter numeric data only.	X Pass ☐ Fail	
04				☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by: Chris Clemente_____   Date Conducted: 5/8/2011_____

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Invoice Status Inquiry

Scenario ID: Test 5	Test Type: Unit	Tester: Chris Clemente	Date: 5/11/11
Scenario Name:	Responding to a Vendor inquiry invoice pay status		
Background and Test Objective(s):			
Related Use Cases or Test Scenarios:	None		

Testing Notes:		Refer to the list of test data in the excel spread sheet.			
Case	**Test Condition(s)**	**Test Activities**	**Expected Results/Notes**	**Actual Results**	**/Notes**
01	Check the Payment Status field.	Check the Payment Status Field to see the status currently.	If the invoice has a Do Not Pay status, check Invoice Notes to find the reason.	X Pass ☐ Fail	
02				☐ Pass ☐ Fail	
03				☐ Pass ☐ Fail	
04				☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by: Chris Clemente_____   Date Conducted:5/11/11_____

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Pay Invoice

Scenario ID: Test 6	Test Type: Unit	Tester: Chris Clemente	Date: 5/11/11

Scenario Name:	PAY INVOICE				
Background and Test Objective(s):	These tests check the accuracy of data entered into the Pay Invoice system.				
Related Use Cases or Test Scenarios:	None				
Testing Notes:	Refer to the list of data in the Excel spread sheet used for testing.				

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	Enter an incorrect date.	Enter a date in the date paid system in the future(not the current date)	A message saying "Please enter todays date" should appear	X Pass ☐ Fail	
02				☐ Pass ☐ Fail	
03				☐ Pass ☐ Fail	
04				☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by: Chris Clemente_____   Date Conducted: 5/11/11_____

Reviewed by:_____   Date Reviewed:_____

23

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Receive Contract

Scenario ID: Test 7	Test Type: Unit	Tester: Chris Clemente	Date: 5/11/11
Scenario Name:	RECEIVE CONTRACT		
Background and Test Objective(s):	These tests will check the accuracy of data entered into a new contract.		
Related Use Cases or Test Scenarios:	Contract Exception (extended by); Add New Contract Information (extended by); Update Contract (used by)		
Testing Notes:	Refer to use case Receive Contract and the data used for testing (in the Excel spread sheet)		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	Search for a none-existent Contract or Vendor/Contractor.	Search for a none-existent Vendor/Contractor to see if the system prompts user to enter a new Contract/Vendor/Contractor.	The system should prompt user "Would you like to enter a new Contract or Vendor/Contractor".	X Pass ☐ Fail	
02	Enter a Contract ID not on the contract.	Enter a fictitious ID number that does not exist.	The system should give the message " Contract ID not in system, would you like to enter a new contract?"	X Pass ☐ Fail	

03	Enter incorrect begin date.	Enter a date in the wrong format.	The system should prompt the user to enter the date format correctly.	X Pass ☐ Fail	
04	Enter a Project Manager name incorrectly.	Enter numeric and symbol values in the name field.	Message should tell the user that are not letters	☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by: <u>Chris Clemente</u>   Date Conducted: <u>5/11/11</u>

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

    ☐   Passed: No further action required   (Date:_____,

    Initials:_____)

    ☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

    ☐   Errors Corrected   (Date:_____,

Initials:_____)

    ☐   Retested   (Date:_____,

Initials:_____)

    ☐   Other:

Receive Invoice

Scenario ID: Test 8	Test Type: Unit	Tester: Chris Clemente	Date: 5/11/11
Scenario Name:	RECEIVE INVOICE		
Background and Test Objective(s):	These test check the accuracy and field constraints of the Receive Contract system.		
Related Use Cases or Test Scenarios:	Invoice Exception (extended by); Update Invoice (used by)		
Testing Notes:	Refer to use case Receive Invoice and the test data used in the Excel spread sheet.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes

01	Search for a none-existent Vendor/Contractor ID	Search for a none-existent Vendor/Contractor to check system prompts.	The system should prompt the user "Would you like to enter a new Vendor/Contractor?"	X Pass ☐ Fail	
02	Enter a Contract ID already in the system.	Enter a Contract ID that already exists.	The system should state "That Contract ID already exists, please enter another Contract ID number.	X Pass ☐ Fail	
03	Enter a billed rate incorrectly.	Enter a billed rate that is different from the "Hourly Fee" on the contract.	The system should prompt the user that the "Billed rate does not match the Hourly Fee on the contract"	X Pass ☐ Fail	
04	Run a Fee Maximum check with an amount over "Fee Maximum".	Run a check to ensure the "Fee Maximum" is not exceeded.	The system should tell the user that "fee Maximum" has been exceeded.	X Pass ☐ Fail	

Testing Notes:

Tests Completed by: Chris Clemente_____   Date Conducted: 5/11/11_____

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Run Accounting Reports

Scenario ID: Test 9	Test Type: Unit	Tester: Chris Clemente	Date: 5/11/11

Scenario Name:	RUN ACCOUNTING REPORTS				
Background and Test Objective(s):	Test the system for generating month-end reports.				
Related Use Cases or Test Scenarios:	None				
Testing Notes:	Refer the Excel spread sheet used for testing data.				

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	Enter an incorrect date range in the G/L Expense option begin date.	Enter a date far in advance and in the past of the correct date.	The system should prompt with a message saying "The date entered is invalid, please use correct date".	X Pass ☐ Fail	
02	Enter an incorrect date range in the Accruals option	Enter a date far in the future and past of the correct date.	The system should prompt with a message saying" The date entered is invalid, please use the correct date".	X Pass ☐ Fail	
03				☐ Pass ☐ Fail	
04				☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by: Chris Clemente_____   Date Conducted:_ 5/11/11_____

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

    ☐   Passed: No further action required   (Date:_____,

Initials:_____)

    ☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

    ☐   Errors Corrected   (Date:_____,

Initials:_____)

    ☐   Retested   (Date:_____,

Initials:_____)

    ☐   Other:

Run Management Reports

Scenario ID: Test 10	Test Type: Unit	Tester: Chris Clemente	Date: 5/11/11
Scenario Name:	RUN MANAGEMENT REPORTS		
Background and Test Objective(s):	These tests verify that inaccurate information cannot be entered into Management Reports.		
Related Use Cases or Test Scenarios:	None		
Testing Notes:	Refer to the test data in the Excel spread sheet used for storing the test data.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	Enter an incorrect date range in the Programmer Expense option.	Enter dates far in advance and in the past.	The system should prompt the user that "The dates entered are invalid; please enter a valid date range".	X Pass ☐ Fail	
02	Enter incorrect range in the Fee Maximum option.	Enter an amount over the "Fee Maximum".	The systems should prompt user saying "Fee Maximum has been exceeded".	☐ Pass ☐ Fail	
03	Enter an incorrect date range in the Contract Recap option.	Enter dates far in advance and in the past.	The system should prompt user saying "The dates entered are invalid, please enter a valid date range".	☐ Pass ☐ Fail	

04				☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by:_ Chris Clemente_____   Date Conducted:__5/11/11_____

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Update Contract

Scenario ID: Test 11	Test Type: Unit	Tester: Chris Clemente	Date: 5/11/11
Scenario Name:	UPDATE CONTRACT		
Background and Test Objective(s):	These tests ensure that incorrect data cannot be entered into a Contract Update.		
Related Use Cases or Test Scenarios:	Receive Contract (uses); Update Invoice (extends)		
Testing Notes:	Refer to use case Update Contract and the test data in the Excel spread sheet used for test data.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	Search for the contract in the system.	Ensure the contract exists.	Find the contract in question.	X Pass ☐ Fail	

02	Leave fields blank.	Leave fields blank that require data. (Contract ID; Vendor ID)	System should prompt the user to "Please enter valid ID"	X Pass ☐ Fail	
03	Enter letters in the date field.	Enter none numeric values in the date field.	The system should not allow anything but numeric dates formatted correctly.	X Pass ☐ Fail	
04				☐ Pass ☐ Fail	

Testing Notes:
Tests Completed by: <u>Chris Clemente_____</u>   Date Conducted: <u>5/11/11_____</u>
Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

    ☐   Passed: No further action required   (Date:_____,

    Initials:_____)

    ☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

    ☐   Errors Corrected   (Date:_____,

Initials:_____)

    ☐   Retested   (Date:_____,

Initials:_____)

    ☐   Other:

Update Invoice

Scenario ID: Test 12	Test Type: Unit	Tester: Chris Clemente	Date: 5/11/11
Scenario Name:	UPDATE INVOICE		
Background and Test Objective(s):	These tests check the accuracy of updating an invoice.		
Related Use Cases or Test Scenarios:	Receive Invoice (uses); Update Contract (extended by)		

Testing Notes:	Refer to use case Update Invoice				
Case	**Test Condition(s)**	**Test Activities**	**Expected Results/Notes**	**Actual Results**	**/Notes**
01	Search for the contract in the system.	Ensure the contract exists.	Find the contract in question.	X Pass ☐ Fail	
02	Leave fields blank.	Leave fields blank requiring data input. (Contract ID; Vendor ID)	System should prompt the user to "Please enter a valid ID"	X Pass ☐ Fail	
03	Enter letters in the date field.	Enter none numeric values in the date field.	The system should not allow anything but numeric dates formatted correctly.	X Pass ☐ Fail	
04				☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by: Chris Clemente_____   Date Conducted: 5/11/11_____

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Back to contents
Other Sample Test Forms

Scenario ID: UC1A	Test Type: Unit	Tester: David Hill	Date: 5/29/02

Scenario Name:	Nancy Numbscull creates a new project
Background and Test Objective(s):	Nancy Numbscull is a design lead that often enters incorrect data or forgets to enter required information. This set of test cases verifies that incorrectly entered data relating to the creation of a new project will be identified and appropriate error messages displayed.
Related Use Cases or Test Scenarios:	Create New Project (use-case), Update Project Information (use-case), Update Customer Information (use-case)
Testing Notes:	Refer to use-case (Create New Project) for specific step-by-step instructions. Refer to master list of test data for specific values to enter during the test activities.

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The project is already in the system.	Enter an existing project name or ID	The system should not allow another project with the same name or ID to be entered	☐ Pass ☐ Fail	
02	An incorrect customer ID is entered	Enter a customer ID or name that does not exist	The system should present a message "Customer does not exist. Would you like to add a new customer?"	☐ Pass ☐ Fail	
03	Incorrect data types entered	Enter non-numeric data into the phone number, project start date, and due date fields	Fields should not accept data	☐ Pass ☐ Fail	
04	Required fields left empty	Leave the Project Name field blank and attempt to save and exit the record	Error message should display: "Project Name is Required. Please Enter."	☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by:_____ Date Conducted:_____

Reviewed by:_____ Date Reviewed:_____

Corrective Actions:

    ☐   Passed: No further action required   (Date:_____, Initials:_____)

    ☐   Errors Submitted for Correction   (Date:_____, Initials:_____)

    ☐   Errors Corrected   (Date:_____, Initials:_____)

    ☐   Retested   (Date:_____, Initials:_____)

    ☐   Other:

Scenario ID: UC1B	Test Type: Unit	Tester: David Hill	Date: 05/29/02
Scenario Name:	Nancy Numbscull creates a new customer record		
Background and Test Objective(s):	Nancy Numbscull is a design lead that often enters incorrect data or forgets to enter required information. This set of test cases verifies that incorrectly entered data relating to the creation of a new customer record will be identified and appropriate error messages displayed.		
Related Use Cases or Test Scenarios:	Update Customer Information (use-case), Update Project Information (use-case), Create New Project (use-case)		
Testing Notes:	Refer to use-case (Update Customer Information) for specific step-by-step instructions. Refer to master list of test data for specific values to enter during the test activities.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	Is the customer name already in the database?	Enter a known current customer's name	The system should generate a message indicating that the customer already exists, and ask if the user really wants to create another customer record.	☐ Pass ☐ Fail	
02	Incorrect data types entered	Enter non-numeric data into the Phone Number field. Enter numeric data into the Country field.	Fields should not accept invalid data types	☐ Pass ☐ Fail	
03	Required fields left empty	Leave the Customer Name field blank and attempt to save and exit the record	Error message should display: "Customer Name Required"	☐ Pass ☐ Fail	
04	N/A	N/A	N/A	☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by:_____   Date Conducted:_____
Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

☐ Passed: No further action required (Date:_____,

Initials:_____)

☐ Errors Submitted for Correction (Date:_____,

Initials:_____)

☐ Errors Corrected (Date:_____,

Initials:_____)

☐ Retested (Date:_____,

Initials:_____)

☐ Other:

Scenario ID: UC1C	Test Type: Unit	Tester: David Hill	Date: 05/29/02
Scenario Name:	Brenda Brainless updates a project record		
Background and Test Objective(s):	Brenda Brainless is a project manager who often enters incorrect data into fields, or forgets to enter required information. This set of test cases verifies that incorrectly entered data relating to the updating of an existing project record will be identified and appropriate error messages displayed.		
Related Use Cases or Test Scenarios:	Update Project Information (use-case), Create New Project (use-case), Update Customer Information (use-case)		
Testing Notes:	Refer to use-case (Update Project Information) for specific step-by-step instructions. Refer to master list of test data for specific values to enter during the test activities.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	A Project Start Date is entered when there is no Contract Number	Enter a Project Start Date when the Contract Number field is blank	Error message should display: "You Must First Enter a Contract Number"	☐ Pass ☐ Fail	
02	A Project End Date is entered when there is no Project Start Date	Enter a Project End Date when the Project Start Date field is blank	Error message should display: "You Must First Enter a Project Start Date"	☐ Pass ☐ Fail	
03	Incorrect data types entered	Enter non-numeric data into the Project Start Date field. Enter non-numeric data into the Project End Date	Fields should not accept data	☐ Pass ☐ Fail	

| 04 | The Project Status is changed to "Active" without a Contract Number | Change the Project Status from "Pending" to "Active" when the Contract Number field is bland | Error message should display: "You Must First Enter a Contract Number" | ☐ Pass ☐ Fail | |

Testing Notes:

Tests Completed by:_____ Date Conducted:_____

Reviewed by:_____ Date Reviewed:_____

Corrective Actions:

☐ Passed: No further action required (Date:_____,

Initials:_____)

☐ Errors Submitted for Correction (Date:_____,

Initials:_____)

☐ Errors Corrected (Date:_____,

Initials:_____)

☐ Retested (Date:_____,

Initials:_____)

☐ Other:

Scenario ID: UC1D	Test Type: Unit	Tester: David Hill	Date: 5/29/02
Scenario Name:	Carla Careful creates a new project		
Background and Test Objective(s):	Carla Careful is a design lead that enters data in the system to create a new project record. This set of test cases verifies that the lookup fields in the Customer Information screen work correctly and display the appropriate data..		
Related Use Cases or Test Scenarios:	Create New Project (use-case), Update Project Information (use-case), Update Customer Information (use-case)		
Testing Notes:	Refer to use-case (Create New Project) for specific step-by-step instructions. Refer to master list of test data for specific values to enter during the test activities.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes

01	The user clicks on the drop down arrow in the lookup box for the Customer field	Press the drop down arrow in the lookup box of the Customer field	The system should display a list of all Customer records	☐ Pass ☐ Fail	
02	The user clicks on the drop down arrow in the lookup box for the Location field	Press the drop down arrow in the lookup box of the Location field	The system should display a list of all Location records	☐ Pass ☐ Fail	
03	The user clicks on the drop down arrow in the lookup box for the Project Status field	Press the drop down arrow in the lookup box of the Project Status field	The system should display a list of all Project Status catagories	☐ Pass ☐ Fail	
04	The user clicks on the drop down arrow in the lookup box for the Project Type field	Press the drop down arrow in the lookup box of the Project Type field	The system should display a list of all Project Type catagories	☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by:_____   Date Conducted:_____

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Scenario ID: NA1	Test Type: System	Tester: David Hill	Date: 05/29/02

Scenario Name:	Peter Perfection assigns a project team				
Background and Test Objective(s):	Peter Perfection is a design lead that needs accurate information on team member assignments in order to assign a project team. This set of test cases verifies that the design lead can navigate to the correct menu and generate a report that shows the team member assignments by project.				
Related Use Cases or Test Scenarios:	Assign Project Team (use-case)				
Testing Notes:	Refer to use-case (Assign Project Team) for specific step-by-step instructions.				

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user logs on to the system	Enter user logon id and password	The system should close the Logon Screen and open the Main Menu	☐ Pass ☐ Fail	
02	The user clicks on the Administration Menu button	Press the Administration Menu button	The system should close the Main Menu and open the Administration Menu	☐ Pass ☐ Fail	
03	The user clicks on the Reports Menu button	Press the Reports Menu button	The system should close the Administration Menu and open the Reports Menu	☐ Pass ☐ Fail	
04	The user clicks on the Team Member Availability Report button	Press the Team Member Availability Report button	The system should close the Reports Menu, run the Team Member Availability query, and display the Team Member Availability Report	☐ Pass ☐ Fail	
Testing Notes:					

Tests Completed by:_____ Date Conducted:_____

Reviewed by:_____ Date Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Scenario ID: UC3A	Test Type: Unit	Tester: David Hill	Date: 05/29/02
Scenario Name:	Martha Mucksitup assigns a project role		
Background and Test Objective(s):	Martha Mucksitup is a project manager who prefers to type data in lookup windows, doesn't always complete all the required information, and sometimes tries to create duplicate records. This set of test cases verifies that incorrectly entered data relating to the assignment of a project role will be identified and appropriate error messages displayed..		
Related Use Cases or Test Scenarios:	Assign Project Role (use-case), Update Team Member Project Assignment (use-case)		
Testing Notes:	Refer to use case (Assign Project Role) for specific step-by-step instructions. Refer to master list of test data for specific values to enter during the test activities.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	Incorrect data types entered	Enter numeric or special character data into the Employee field. Enter numeric or special character data into the Project Role field. Enter numeric or special character data into the Role Status field	Field should not accept data and generate an error message: "Invalid Entry. Please Enter a Character Between A and Z Only."	☐ Pass ☐ Fail	
02	Required fields left empty	Leave the Employee Name field blank and attempt to exit the record	Error message should display: "Employee Name required"	☐ Pass ☐ Fail	

| 03 | A specific project role already exists in the system | Create a project role by selecting an Employee, a Project Role, and a Role Status. Save the record and repeat the process with the exact same data selections. | Error message should display: "This Project Role Already Exists. Please Make Another Selection." | ☐ Pass
☐ Fail | |
| 04 | N/A | N/A | N/A | ☐ Pass
☐ Fail | |

Testing Notes:

Tests Completed by:_____   Date
Conducted:_____

Reviewed by:_____   Date
Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Scenario ID: UC3B	Test Type: Unit	Tester: David Hill	Date: 05/29/02
Scenario Name:	Steve Stressedout updates a team member project assignment		
Background and Test Objective(s):	Steve Stressedout is a project manager who tries to create records in the system with data that doesn't exist in lookup tables and tries to enter too many characters in comment fields. This set of test cases verifies that incorrectly entered data relating to the updating of a team member project assignment will be identified and appropriate error messages displayed..		
Related Use Cases or Test Scenarios:	Update Team Member Project Assignment (use-case), Assign Project Role (use-case)		
Testing Notes:	Refer to use-case (Update Team Member Project Assignment) for specific step-by-step instructions. Refer to master list of test data for specific values to enter during the test activities.		
Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes

01	A role is created for a non-existent employee	Enter a bogus Employee name	Error message should display: "Employee not found"	☐ Pass ☐ Fail	
02	An employee is assigned a non-existent Project Role	Enter a bogus Project Role	Error message should display: "Project role not found"	☐ Pass ☐ Fail	
03	An employee is assigned a non-existent Role Status	Enter a bogus Role Status	Error message should display: "Role status not found"	☐ Pass ☐ Fail	
04	Too many characters are entered in the Comments field	Hold down any key	When 257 characters have been entered, an error message should display: "Maximum field size of 256 characters has been exceeded."	☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by:_____ Date
Conducted:_____

Reviewed by:_____ Date
Reviewed:_____

Corrective Actions:

☐ Passed: No further action required   (Date:_____,

Initials:_____)

☐ Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐ Errors Corrected   (Date:_____,

Initials:_____)

☐ Retested   (Date:_____,

Initials:_____)

☐ Other:

Scenario ID: UC4A	Test Type: Unit	Tester: Pete Farrar	Date: 05/20/02

Scenario Name:	Charles Careless creates a new task assignment
Background and Test Objective(s):	Charles Careless is a project manager who often doesn't complete all the required fields, or tries to assign the same task to an employee more than one time. This set of test cases verifies that incorrectly entered data related to a new task assignment is identified and appropriate error messages are displayed
Related Use Cases or Test Scenarios:	Create Project Task Assignment (use-case); Update Project Task Assignments (use-case)
Testing Notes:	Refer to use-case (Create Project Task Assignment) for specific step-by-step instructions. Refer to the master list of test data for specific values to enter during the test activities.

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	Assign a task to a non-existent project role	Choose an Assigned Task with a blank project role	The system should not allow a task to be assigned to a non-existent project role	☐ Pass ☐ Fail	
02	A task start date is entered that is older than the current date	Enter a task start date older than the current date	The system should not allow a task start date that is earlier than the task assignment date.	☐ Pass ☐ Fail	
03	A task due date is entered that is older than the current date	Enter a task due date older than the current date	The system should not allow a task due date that is earlier than the task assignment date	☐ Pass ☐ Fail	
04	The task status is changed to other than "Pending" before the project start date	Change the project status from "Pending" for a task start date that is in the future	The system should not allow a task status other than "Pending" for a task that has not been started	☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by:_____   Date
Conducted:_____

Reviewed by:_____   Date
Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Scenario ID: UC4B	Test Type: Unit	Tester: Pete Farrar	Date: 05/20/02
Scenario Name:	Charles Careless updates existing task assignment information		
Background and Test Objective(s):	Charles Careless is a project manager who often enters the wrong type of data, or tries to assign an inappropriate task to a project role. This set of test cases verifies that incorrectly entered data related to an existing task assignment is identified and appropriate error messages are displayed		
Related Use Cases or Test Scenarios:	Update Project Task Assignments (use-case); Create Project Task Assignment (use-case)		
Testing Notes:	Refer to use-case (Update Project Task Assignments) for specific step-by-step instructions. Refer to the master list of test data for specific values to enter during the test activities.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The task of manager is assigned to a project role other than manager	Select the assigned task of manager for a project role other than manager	The system should not allow an assigned task of manger for a project role that isn't manager	☐ Pass ☐ Fail	
02	An incorrect data type is entered in a date field	Enter non-numeric data into both the start date and due date fields.	The system should not accept non-numeric data in these fields	☐ Pass ☐ Fail	
03	The task status is changed to "Pending" after the project start date	Change the project status "Pending" for a task start date that is in the past	The system should not allow a task status of "Pending" for a task that has been started	☐ Pass ☐ Fail	

| 04 | The task status is changed to "Completed" before the project start date | Change the project status to "Completed" for a task start date that is in the future | The system should not allow a task status of "Completed" for a task that has not been started | ☐ Pass
☐ Fail | |

Testing Notes:

Tests Completed by:_____   Date
Conducted:_____

Reviewed by:_____   Date
Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Scenario ID: UC5A	Test Type: Unit	Tester: Pete Farrar	Date: 05/20/02
Scenario Name:	Abbey Abnormal creates a weekly task status report		
Background and Test Objective(s):	Abbey Abnormal is a designer who frequently omits important data, tries to enter incorrect data formats, and creates improbable reporting statistics. This set of test cases verifies that incorrectly entered or omitted data is identified and appropriate error messages are displayed.		
Related Use Cases or Test Scenarios:	Enter Weekly Status Information (use-case); Update Task Status Information (use case)		
Testing Notes:	Refer to use-case (Enter Weekly Status Information) for specific step-by-step instructions. Refer to the master list of test data for specific values to enter during the test activities.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes

01	A Week Ending date is entered that is older than the current date	Enter a Week Ending date that is before the current date (in the past)	The system should not allow a Week Ending date that is before the current report date	☐ Pass ☐ Fail	
02	An incorrect data type is entered in the Weekly Billable Hours field	Enter non-numeric data into the Weekly Billable Hours field	The system should not accept non-numeric data in this field	☐ Pass ☐ Fail	
03	More than 100 hours are entered in the Weekly Billable Hours field	Enter any three digit number over 100	The systems should display a warning message: "You are reporting over 100 billable hours. Do you really want to do this?"	☐ Pass ☐ Fail	
04	Required fields are not entered	Leave the Weekly Billable Hours field blank and attempt to save and exit the record	An error message is displayed: "Billable Hours are Required, Please Enter."	☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by:_____   Date Conducted:_____

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Scenario ID: NA2	Test Type: System	Tester: Pete Farrar	Date: 05/20/02
Scenario Name:	Randy Reporter verifies team status reports		
Background and Test Objective(s):	Randy Reporter is a project manager who verifies the weekly task status reports of his project team members. This set of test cases verifies that the project manager can navigate to the correct menu and generate a report that shows the team member weekly status report.		
Related Use Cases or Test Scenarios:	Verify Team Status Updates (use-case), Update Task Status Information (use-case), Run Project Status Reports (use-case)		
Testing Notes:	Refer to use-case (Verify Team Status Updates) for specific step-by-step instructions.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user logs on to the system	Enter user logon id and password	The system should close the Logon Screen and open the Main Menu	☐ Pass ☐ Fail	
02	The user clicks on the drop down arrow in the lookup box and selects the project they are managing for which they want a report	Press the drop down arrow in the lookup box and select the project for which the user is manger	The system should close the Main Menu and open the Manager Menu*	☐ Pass ☐ Fail	
03	The user clicks on the Review a Report button	Press the Review a Report button	The system should close the Manager Menu and open the Manager Report Menu	☐ Pass ☐ Fail	
04	The user clicks on the drop down arrow in the lookup box and selects the Task Status Summary Report	Press the drop down arrow in the lookup box and select the Task Status Summary Report	The system should close the Manager Report Menu, run the Task Status Summary query, and display the Task Status Summary Report	☐ Pass ☐ Fail	

Testing Notes: *The system should not open the Manager Menu on a project for which the user is not manager

Tests Completed by:_____   Date Conducted:_____

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

☐ Passed: No further action required (Date:_____, Initials:_____)

☐ Errors Submitted for Correction (Date:_____, Initials:_____)

☐ Errors Corrected (Date:_____, Initials:_____)

☐ Retested (Date:_____, Initials:_____)

☐ Other:

Scenario ID: UC5C	Test Type: Unit	Tester: Pete Farrar	Date: 05/20/02
Scenario Name:	Darren Duncecap needs to correct an error on his status report		
Background and Test Objective(s):	Darren Duncecap is a designer who makes mistakes when he creates his weekly task status reports. His project manager is continually requesting him to correct his errors. This set of test cases verifies that the employee is able to access his incorrect task status report and make the necessary corrections to his task status report.		
Related Use Cases or Test Scenarios:	Update Task Status Information (use-case); Enter Weekly Status Information (use-case)		
Testing Notes:	Refer to use-case (Update Task Status Information) for specific step-by-step instructions. Refer to the master list of test data for specific values to enter during the test activities.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user logs on to the system	Enter user logon id and password	The system should close the Logon Screen and open the Main Menu	☐ Pass ☐ Fail	
02	The user clicks on the drop down arrow in the lookup box and selects the project that has an incorrect Task Status Report	Press the drop down arrow in the lookup box and select the project with the incorrect Task Status Report	The system should close the Main Menu and open the Team Member Task Assignment Screen	☐ Pass ☐ Fail	

03	The user clicks the navigation button to display the task assignment for which the incorrect report was created	Click the back navigation button until the task assignment appears that needs to have the Task Status Report correction	The system should scroll through all the task assignment records for the chosen project	☐ Pass ☐ Fail
04	The user clicks the Create Weekly Report button	Click the Create Weekly Report button	The system should close the Team Member Task Assignment Screen and open the Create Team Member Report Screen displaying the Task Assignment previously selected	☐ Pass ☐ Fail
05	The user clicks the navigation button to display the team member report for which the incorrect data was entered	Click the back navigation button until the team member report appears that needs to be corrected	The system should scroll to the report created in the current week that needs to be corrected. (Prior reports should be 'read-only')	☐ Pass ☐ Fail
06	The user enters new hours in the Billable Hours field	Place the cursor in the Billable Hours, highlight the existing hours, and enter new hours	The system should allow new hours to be entered, overwriting the existing entry	☐ Pass ☐ Fail
07	The user clicks on the down arrow in the lookup box for the Task Status	Click on the down arrow in the lookup box for the Task Status and select a different status from that displayed	The system should allow a new task status to be selected	☐ Pass ☐ Fail
08	The user click in the Red Flags field	Place the cursor in the Red Flags field and enter new team member report notes	The system should allow new notes to be concatenated to the existing notes (but not overwrite them)	☐ Pass ☐ Fail
09	The user click in the Comments field	Place the cursor in the Comments field and enter new team member report notes	The system should allow new notes to be concatenated to the existing notes (but not overwrite them)	☐ Pass ☐ Fail
Testing Notes:				

Tests Completed by:_____   Date Conducted:_____
Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

    ☐   Passed: No further action required   (Date:_____,

    Initials:_____)

    ☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

    ☐   Errors Corrected   (Date:_____,

Initials:_____)

    ☐   Retested   (Date:_____,

Initials:_____)

    ☐   Other:

Scenario ID: UC5D	Test Type: Unit	Tester: Pete Farrar	Date: 05/20/02
Scenario Name:	Norma Normal creates a weekly task status report		
Background and Test Objective(s):	Norma Normal is a designer who enters weekly status reports. This set of test cases verifies that the user can navigate to the proper screen to enter their weekly task status report and that the data entered is accepted by the system.		
Related Use Cases or Test Scenarios:	Enter Weekly Status Information (use-case); Update Task Status Information (use case)		
Testing Notes:	Refer to use-case (Enter Weekly Status Information) for specific step-by-step instructions. Refer to the master list of test data for specific values to enter during the test activities.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user clicks on the drop down arrow in the lookup box and selects a project	Press the drop down arrow in the lookup box and select project	The system should close the Main Menu and open the Team Member Task Assignments screen displaying the selected project task	☐ Pass ☐ Fail	
02	The user clicks on the Create Weekly Report Button	Press the Create Weekly Report button	The system should close the Team Member Task Assignments screen	☐ Pass ☐ Fail	

			& open the Create Team Member Report screen, displaying the current date in the Reporting Date field and Friday's date of the current week		
03	The user enters the number of billable hours performed on the chosen task	Enter number of hours worked on the selected task	The systems should accept up to a three digit number	☐ Pass ☐ Fail	
04	The user clicks on the drop down arrow in the lookup box for the Task Status field	Press the drop down arrow in the lookup box for the Task Status field and select a task status	The system should display a list of all Task Status catagories, allow the user to chose one, and display the selection in the Task Status field	☐ Pass ☐ Fail	
05	The user moves the cursor to the Red Flags field	Press the Tab or click the mouse cursor in the Red Flags field	The system should make the Red Flags field the active field and allow data to be written in the field	☐ Pass ☐ Fail	
06	The user moves the cursor to the Comments field	Press the Tab or click the mouse cursor in the Comments field	The system should make the Comments field the active field and allow data to be written in the field	☐ Pass ☐ Fail	

Testing Notes: This test scenario assumes that the user has already logged on and is at the Main Menu

Tests Completed by:_____ Date Conducted:_____

Reviewed by:_____ Date Reviewed:_____

Corrective Actions:

☐ Passed: No further action required (Date:_____,

Initials:_____)

☐ Errors Submitted for Correction (Date:_____,

Initials:_____)

☐ Errors Corrected (Date:_____,

Initials:_____)

☐ Retested (Date:_____,

Initials:_____)

☐ Other:

Scenario ID: NA3	Test Type: System	Tester: Pete Farrar	Date: 05/20/02
Scenario Name:	Alice AdminAssistant runs weekly project summary report		
Background and Test Objective(s):	Alice Admin-Assistant is the administrative assistant to the owners who runs the weekly project summary report. This set of test cases verifies that the administrative assistant can navigate to the correct menu and generate a report that shows the weekly summary of the organizations projects.		
Related Use Cases or Test Scenarios:	Run Project Status Reports (use-case), Enter Weekly Status Information (use-case), Verify Team Status Updates (use-case)		
Testing Notes:	Refer to use-case (Run Project Status Reports) for specific step-by-step instructions.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user logs on to the system	Enter user logon id and password	The system should close the Logon Screen and open the Administration Report Menu	☐ Pass ☐ Fail	
02	The user clicks on the Weekly Project Summary Report button	Press the Weekly Project Summary Report button	The system should close the Administration Report Menu, run the Weekly Project Summary query, and display the Weekly Project Summary Report	☐ Pass ☐ Fail	

03	The user clicks on the print icon in the toolbar	Click the print icon (printer) in the toolbar at the top of the screen	The system should print the Weekly Project Summary Report displayed on the screen	☐ Pass ☐ Fail	
04	N/A	N/A	N/A	☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by:_____   Date
Conducted:_____

Reviewed by:_____   Date
Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Scenario ID: NA4	Test Type: Unit	Tester: Dave Hill	Date: 05/23/02
Scenario Name:	Harley Hacker logs on to the system		
Background and Test Objective(s):	Harley Hacker is a janitor who doesn't have access to the system. This test case verifies that the system will not allow invalid users to gain access..		
Related Use Cases or Test Scenarios:	Peter Perfection assigns a project team (test scenario), Randy Reporter verifies team status reports (test scenario), Darren Duncecap needs to correct an error on his status report (test scenario), Alice Admin-Assistant runs weekly project summary report (test scenario),		

Testing Notes:	None

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user logs on to the system	Enter a fictitous user logon id and password	The system should not allow access to unauthorized users, and display an error message: "Either your user name of password is incorrect. Please try again or contact an administrator."	☐ Pass ☐ Fail	
02	N/A	N/A	N/A	☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by:_____   Date
Conducted:_____

Reviewed by:_____   Date
Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Scenario ID: NA6	Test Type: System	Tester: Pete Farrar	Date: 05/20/02
Scenario Name:	Paula ProjectManager navigates through the system		

Background and Test Objective(s):	Paula ProjectManager is a designer who is being trained on the new system. This set of test cases verifies that a team member who is a project manager can navigate to the correct menus and data entry screens used by project managers.
Related Use Cases or Test Scenarios:	None.
Testing Notes:	This test scenario does not test the Exit System buttons, but rather only tests the buttons that navigate to other screens.

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user logs on to the system	Enter user logon id and password	The system should close the Logon Screen and open the Main Menu. The Administration Menu button should be grayed out.	☐ Pass ☐ Fail	
02	The user clicks on the drop down arrow in the lookup box and selects a project that has a project role of manager selected	Press the drop down arrow in the lookup box and select a project with a project role of manager	The system should close the Main Menu and open the Manager Menu	☐ Pass ☐ Fail	
03	The user clicks on the Create a Report button	Press the Create a Report button	The system should close the Manager Menu & open the Create Manager Report screen	☐ Pass ☐ Fail	
04	The user clicks on the drop down arrow in the lookup box and selects a project role	Press the drop down arrow in the lookup box and select a project role	The system should add a new field titled Select a Task.	☐ Pass ☐ Fail	
05	The user clicks on the drop down arrow in the lookup box and selects a task	Press the drop down arrow in the lookup box and select a task	The system should close the Create Manager Report screen & open the Create Manager Task Status Report screen	☐ Pass ☐ Fail	
06	The user clicks on the Select a Different Task button	Press the Select a Different Task button	The system should close the Create Manager Task Status Report screen and open the Create Manager Report Menu	☐ Pass ☐ Fail	

07	The user clicks on the Return to Manager Menu button	Press the Return to Manager Menu button	The system should close the Create Manager Report Menu screen and open the Manager Menu	☐ Pass ☐ Fail
08	The user clicks on the Review a Report button	Press the Review a Report button	The system should close the Manager Menu and open the Manager Report Menu	☐ Pass ☐ Fail
09	The user clicks on the Return to Manager Menu button	Press the Return to Manager Menu button	The system should close the Manager Report Menu and open the Manager Menu	☐ Pass ☐ Fail
10	The user clicks the Return to Main Menu button	Press the Return to Main Menu button	The system should close the Manager Menu and open the Main Menu	☐ Pass ☐ Fail
11	The user clicks the Project Information button	Press the Project Information button	The system should close the Main Menu and open the Manger Select Project Menu with the Add a New Project button grayed out	☐ Pass ☐ Fail
12	The user clicks on the drop down arrow in the lookup box and selects a project	Press the drop down arrow in the lookup box and select a project	The system should close the Manager Select Project Menu and open the Project Information screen	☐ Pass ☐ Fail
13	The user clicks the Add Team Member button	Press the Add Team Member button	The system should close the Project Information screen and open the Employee Role screen	☐ Pass ☐ Fail
14	The user clicks the Return to Project Info button	Press the Return to Project Info button	The system should close the Employee Role screen and open the Project Information screen	☐ Pass ☐ Fail
15	The user clicks the Return to Project Menu button	Press the Return to Project Menu button	The system should close the Project Information screen and open the Manager Select Project screen	☐ Pass ☐ Fail

16	The user clicks the Return to Main Menu button	Press the Return to Main Menu button	The system should close the Manager Select Project screen and open the Main Menu	☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by:_____   Date Conducted:_____

Reviewed by:_____   Date Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Scenario ID: NA7	Test Type: System	Tester: Dave Hill	Date: 05/23/02
Scenario Name:	Sam SysAdmin navigates through the system utilities		
Background and Test Objective(s):	Sam SysAdmin is a systems administrator who is being trained to maintain data in the system lookup/reference tables. This set of test cases verifies that a systems administrator can navigate to the correct menus and data entry screens used by system administrators.		
Related Use Cases or Test Scenarios:	None.		
Testing Notes:	This test scenario does not test the Exit System buttons, but rather only tests the buttons that navigate to other screens.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes

01	The user logs on to the system	Enter user logon id and password	The system should close the Logon Screen and open the Main Menu. The down arrow in the lookup box should be disabled and the Project Information button should be grayed out	☐ Pass ☐ Fail
02	The user clicks on the Administration Menu button	Press the Administration Menu button	The system should close the Main Menu and open the Maintenance Menu	☐ Pass ☐ Fail
03	The user clicks on the Customer Information button	Press the Customer Information button	The system should close the Maintenance Menu and open the Customer Information screen	☐ Pass ☐ Fail
04	The user clicks on the Return to Menu button	Press the Return to Menu button	The system should close the Customer Information screen and open the Maintenance Menu	☐ Pass ☐ Fail
05	The user clicks on the Project Task button	Press the Project Task button	The system should close the Maintenance Menu and open the Project Task screen	☐ Pass ☐ Fail
06	The user clicks on the Return to Menu button	Press the Return to Menu button	The system should close the Project Task screen and open the Maintenance Menu	☐ Pass ☐ Fail
07	The user clicks on the Employee Information button	Press the Employee Information button	The system should close the Maintenance Menu and open the Employee Information screen	☐ Pass ☐ Fail
08	The user clicks on the Return to Menu button	Press the Return to Menu button	The system should close the Employee Information screen and open the Maintenance Menu	☐ Pass ☐ Fail
09	The user clicks on the Project Role button	Press the Project Role button	The system should close the Maintenance Menu and open the Project Role screen	☐ Pass ☐ Fail

10	The user clicks on the Return to Menu button	Press the Return to Menu button	The system should close the Project Role screen and open the Maintenance Menu	☐ Pass ☐ Fail
11	The user clicks on the Hub Location button	Press the Hub Location button	The system should close the Maintenance Menu and open the Hub Location screen	☐ Pass ☐ Fail
12	The user clicks on the Return to Menu button	Press the Return to Menu button	The system should close the Hub Location screen and open the Maintenance Menu	☐ Pass ☐ Fail
13	The user clicks on the Project Phase button	Press the Project Phase button	The system should close the Maintenance Menu and open the Project Phase screen	☐ Pass ☐ Fail
14	The user clicks on the Return to Menu button	Press the Return to Menu button	The system should close the Project Phase screen & open the Maintenance Menu	☐ Pass ☐ Fail
15	The user clicks on the State/Province button	Press the State/Province button	The system should close the Maintenance Menu and open the State/Province screen	☐ Pass ☐ Fail
16	The user clicks on the Return to Menu button	Press the Return to Menu button	The system should close State/Province screen & open the Maintenance Menu	☐ Pass ☐ Fail
17	The user clicks on the Task Status Name button	Press the Task Status Name button	The system should close the Maintenance Menu & open the Task Status Name screen	☐ Pass ☐ Fail
18	The user clicks on the Return to Main Menu button	Press the Return to Main Menu button	The system should close the Maintenance Menu and open the Main Menu	☐ Pass ☐ Fail

Testing Notes:

Tests Completed by:_____   Date Conducted:_____

Reviewed by:_____	Date Reviewed:_____

Corrective Actions:

☐   Passed: No further action required   (Date:_____,

Initials:_____)

☐   Errors Submitted for Correction   (Date:_____,

Initials:_____)

☐   Errors Corrected   (Date:_____,

Initials:_____)

☐   Retested   (Date:_____,

Initials:_____)

☐   Other:

Scenario ID: NA5	Test Type: System	Tester: Pete Farrar	Date: 05/20/02
Scenario Name:	Terry TeamMember navigates through the system		
Background and Test Objective(s):	Terry TeamMember is a developer who is being trained on the new system. This set of test cases verifies that a team member who is not a project manager of any project can navigate to the correct menus and data entry screens used by team members who are not project managers.		
Related Use Cases or Test Scenarios:	None.		
Testing Notes:	This test scenario does not test the Exit System buttons, but rather only tests the buttons that navigate to other screens.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user logs on to the system	Enter user logon id and password	The system should close the Logon Screen and open the Main Menu. The Project Information & Administration Menu buttons should be grayed out.	☐ Pass ☐ Fail	
02	The user clicks on the drop down arrow in the lookup box and	Press the drop down arrow in the lookup box and select project	The system should close the Main Menu and open the Team Member Task	☐ Pass ☐ Fail	

	selects a project		Assignments screen		
03	The user clicks on the Create Weekly Report Button	Press the Create Weekly Report button	The system should close the Team Member Task Assignments screen & open the Create Team Member Report screen	☐ Pass ☐ Fail	
04	The user clicks on the Return to Task Menu button	Press the Return to Task Menu button	The system should close the Team Member Report screen and open the Team Member Task Assignments screen.	☐ Pass ☐ Fail	
05	The user clicks on the Select a Different Project button	Press the Select a Different Project button	The system should close the Team Member Task Assignments screen & open the Main Menu	☐ Pass ☐ Fail	

Testing Notes:

Tests Completed by:_____ Date Conducted:_____

Reviewed by:_____ Date Reviewed:_____

Corrective Actions:

☐ Passed: No further action required (Date:_____,

Initials:_____)

☐ Errors Submitted for Correction (Date:_____,

Initials:_____)

☐ Errors Corrected (Date:_____,

Initials:_____)

☐ Retested (Date:_____,

Initials:_____)

☐ Other:

Actual Sample Test Data Used
Buyer Table

Contact_ID	Contact_Title	Contact_FName	Contact_LName	Contact_Phone	Contact_Email	Unit_ID
1	Project Manager	Greg	Sorenson	2066654465	Sorenson@example.com	2
2	Project Manager	Emily	Dickenson	4256654497	Dickens@example.com	2
3	Project Manager	Fred	Flinstone	4256658897	Fred@example.com	2
4	Project Manager	Jerry	Lewis	6657794878	Lewis@example.com	3
5	Project Manager	Mary	Hart	6198746754	Hart@example.com	4
7	Project Lead	George	Romero	2065551324	Romero@example.com	3

Contract Table

Buyer_ID	Buyer_FName	Buyer_LName	Contract_ID	Buyer_Email
1	Harry	Gorgio		Gharry@example.com
2	Jeff	Marting		Marting@example.com
3	Sara	Forn		Sforn@example.com
4	Abel	Williams		Awilliams@example.com

Division Table

Division_ID	Division_Name
1	AMB
2	CCR
3	NAB
4	ORF

Invoice Table

Invoice_ID	Invoice_Start_Date	Invoice_End_Date	Invoice_Date	Invoice_Due_Date	Invoice_Terms	Invoice_Hours
6	4/20/2012	4/25/2012	4/13/2011	4/27/2012	na	30
7	11/18/2011	11/25/2011	4/25/2011	11/28/2012	n/a	40
17	4/18/2011	4/24/2008	4/25/2008	4/30/2008	n/a	80

20	4/1/2008	4/1/2008	4/10/2008	4/25/2009		40
21	4/1/2011	4/25/2011	4/1/2011	4/30/2011	na	200
23	8/1/2008	10/1/2008	8/1/2008	10/15/2008	n/a	200
24	1/1/2008	6/1/2008	1/1/2008	6/10/2008	n/a	200
25	3/1/2008	4/1/2008	2/27/2008	4/10/2008	n/a	120
26	8/1/2008	9/15/2008	7/26/2008	9/20/2008	n/a	150
28	8/1/2011	9/1/2011	5/1/2011	9/16/2011	na	200
47	5/1/2008	5/25/2008	5/2/2008	5/31/2008	n/a	50
48	5/2/2008	5/25/2008	5/1/2008	5/30/2008	n/a	25
78	8/1/2008	8/26/2008	7/30/2008	8/31/2008	n/a	80
79	3/1/2008	3/8/2008	3/1/2008	3/10/2008	n/a	25
80	8/1/2008	8/10/2008	8/1/2008	8/16/2008	n/a	25

Invoice Table (cont.)

Work_Description	Invoice_Notes	Invoice_Pay_Status	Invoice_Accrual	Invoice_Accrual_Reversed	Vendor_ID
clean up		Unpaid			2
security consulting		Unpaid			6
basic services		Paid	5/6/2008	5/13/2008	1
services		Paid			2
basic services		Unpaid			14
Designer software		Paid			16
Designing and installing security monitoring software		Paid			14
General cleanup		Paid			15
general services		Paid			3
general services		Unpaid			2
general services		Paid			3
general computing services		Paid	5/27/2008		3
general services		Paid	8/31/2008		10
cleanup		Paid			10
cleanup		Paid			10

Invoice Table (cont.)

Vendor_ID	Programmer_ID	Contract_ID	Unit_ID	Acc_Employee_ID	Buyer_ID

2	1	33	1		1
6	2	33	1		2
1	3	1	2		3
2	4	2	3		4
14	5	14	3		1
16	2	4	4		2
14	2	3	4		3
15	3	5	2		4
3	4	10	3		1
2	4	9	2		2
3	5	6	2		3
3	6	7	2		4
10	6	8	4		2
10	6	1	3		2
10	6	1	3		1
13	2	1	2		

Programmer Table

Programmer_ID	Programmer_FName	Programmer_LName	Programmer_Phone	Programmer_Address_1	Programmer_Address_2
1	Tyler	Hall	3187157305	3894 Emerson Road	unit B
2	Luther	Watkins	6238759829	2269 Cambridge Drive	
3	Michelle	Ashby	8057561183	13 Par Drive	
4	Jerry	Karl	8411523322	14365 Ashberry	
5	Mary	Lamb	4256651544	123 Fake St	ap 8
6	Harry	Henderson	2069994465	123 201st place	
7	Jerry	Only	2065551324	321 Georgia St	

Programmer Table (cont.)

Programmer_City	Programmer_State	Programmer_Zip	Programmer_Email	Programmer_Rate_Pay	Programmer_Notes	Vendor_ID
Alexandria	LA	71301-	TylerHall@example.com	50		1
Youngtown	AZ	85363-	WatkinsLuther@example.com	50		2
San Luis	CA	93401-	Asby@example	25		3

Obispo			.com				
San Fransisco	CA	98045	Karl@example.com	15			7
Seattle	WA	98114	Lamb@example.com	40			8
Shoreline	WA	98043	Henderson@example.com	45			10
Shoreline	WA	98036-	OnlyJerry@exampl.com	55			4

Unit Table

Unit_ID	Unit_Name	Division_ID
1	DDRM	1
2	VHCD	2
3	VCHM	3
4	OPRT	4

Vendor Table

Vendor_ID	Vendor_Name	Vendor_Phone	Vendor_Address_1	Vendor_Address_2	Vendor_City
1	Donny Wicks Associates	(525) 235-8746	574 Wescam Court		Reno
2	APEX	(201) 546-1348	2336 Pine Street		Portland
3	Beltam Systems Inc	(818) 425-6413	194 Layman Avenue		Raeford
4	Computer Helpers	(425) 664-5896	1161 Peaceful Lane		Cleveland
5	Computing Experience	(401) 254-7877	3770 Arthur Avenue		Freeport
6	Integrated Info Systems	(425) 555-1234	2129 Drummond Street		Paterson
7	David Monroe Consulting	(664) 254-1346	4525 Oxford Court		Columbus
8	Donny Wicks Associates	(720) 123-6541	1986 Maple Street		Los Angeles
9	EDS Temps Inc	(425) 789-6458	2117 Don Jackson Lane		Hamburg
10	Electric Enterprises Inc	(206) 555-1456	4601 Steele Street		Lombard
11	Euro Systems International	(201) 256-3456	1240 Poe Lane		Kansas City
12	Fix-em Right Inc	(858) 997-4465	1266 Sunny Day Drive		Los Angeles
13	Electric	(250) 546-	3628 Frederick		El Paso

	Enterprises Inc	7155	Street		
14	Integrated Info Systems	(845) 134-5464	382 Red Maple Drive		El Monte
15	International Consultants	(660) 213-8544	4555 Moore Avenue		Keller
16	Neo Computing	(760) 254-8898	3024 Bagwell Avenue		Ocala
17	NW Consulting	(775) 124-8975	2692 C Street		East Bridgewate e
18	PN Associates	(619) 554-7844	87 Cemetery Street		Salinas
19	Programmers are US	(330) 225-2264	4848 Cameron Road		Jamestow n
20	Quality Computer Pros	(554) 664-4878	3158 Mesa Drive		Las Vegas
21	R N Andersen & Associates	(546) 197-6452	4621 Turnpike Drive		Wadley
22	Redford Group	(807) 135-5444	697 Elk City Road		Indianapo lis
23	Varjaraj Consulting	(888) 256-1254	1672 Bagwell Avenue		Ocala
24	Western States Consulting	(654) 123-4567	3301 Lochmere Lane		Wallingfor d

Vendor Table (cont.)

Vendor_Sta te	Vendor_Zi p	Vendor_Email	Vendor_Not es
NV	89501-	DJones@example.com	
PA	97205-	Leeson@example.com	
nc	28376-	MoisesMCastillo@example.com	
OH	44114-	JBarnard@example.com	
IL	61032-	Cexperience@example.com	
NJ	07501-	KatieTMarshall@example.com	
MS	39705-	EllenDPerez@example.com	
ca	90017-	TamiDVance@example.com	
MI	48139-	EDS@example.com	
Il	60148-	EEI@example.com	
KS	64106-	EuroSI@example.com # MichelleLDyson@example.com	
CA	90017-	FER@example.com	
TX	79927-	Hinojosa@example.com	
CA	91731-	Bowen@example.com	
TX	76248-	Consultants@example.com	
Fl	34471-	Mward@example.com	
MA	02333-	Steele@example.com	

CA	93901-	PNA@example.com # RosemaryJPrice@example.com	
NY	14701-	Gonzalez@example.com	
NV	89101-	QCPros@example.com # AnneJBell@example.com	
AL	36276-	Villegas@example.com	
IN	46254-	Redford@example.com	
fl	34471-	MichaelGPatrick@example.com	
CT	06492-	Axelson@example.com	

Back to contents