



**Edmonds Community College
Computer Information Systems
System Design Specifications**

Prepared: April 30th 2015

Scrappers Systems Analysis and Development Group:

Jennifer Lam

Joshua Chong

Duane Samuels

Cody Smith

Memo

To: Patrick Jay, Vice President & Manager
From: Scrappers Systems Analysis Group
Date: April 23, 2015
Subject: Transmittal of System Design Specification

Scrappers Systems Analysis Group has attached the System Design Specification for the new Bank of Xanadu project of managing payments for contractor/programmers system for you to review.

The System Design Document contains the following main sections

- Architecture and Design Considerations
- Information/Data Model
- User Navigation Design (Storyboard)
- Inputs, Outputs, Procedures
- Interface Design and Coding Standards

We would like to arrange a date to review the document at the Bank of Xanadu branch in Bellevue, Washington on May 14, 2015 at 6:00pm. Please feel free to contact the Scrapper via email if you have any questions or concerns.

System Design Specifications
New Automated System for Managing Contract Payables

Bank of Xanadu
Accounting Department

Scrappers Systems Analysis Group:

Jennifer Lam

Joshua Chong

Duane Samuels

Cody Smith

Prepared: April 30th 2015

1 TABLE OF CONTENTS

- 2 Introduction 5
- 3 Architecture and Design Considerations 6
- 4 Information/Data Model..... 7
- 5 User Navigation Design 8
- 6 Inputs 10
- 7 Outputs 11
- 8 Procedures..... 12
- 9 Interface Design and Coding Standards..... 17
- 10 Conclusion 18
- 11 Appendix 19
 - 11.1 Appendix A: Entity Relationship Diagram..... 20
 - 11.2 Appendix B: Metadata Dictionary 21
 - 11.3 Appendix C: Navigation design..... 29
 - 11.4 Appendix D: Input..... 30
 - 11.5 Appendix E: Output..... 35
 - 11.6 Appendix F: Use Cases 41
 - 11.7 Appendix G: Design Standards Document 54
 - 11.8 Appendix H: Issues List..... 58
 - 11.9 Appendix I: Futures list..... 59

2 INTRODUCTION

Following Team Scrapper's Analysis and Planning phases, we are now ready to continue forward with the Systems Design phase of the project. The system design specifications document will contain all the information required to create the new invoicing system

In this document, you will find the following:

- Architecture and Design Considerations – Contains information on how the network will be structured and what resources are needed for the system to run
- Information/Data Model – Contains an overview of the data requirements of each entity
- User Navigation Design – Contains a summary of how users will interact and navigate through the system
- Inputs/Outputs – Contains a summary of source documents and their purpose in the system
- Procedures – Contains a summary of the general flow of events when a user is entering data into the system
- Interface Design and Coding Standards – Contains a summary of standard that the team will follow to ensure that the system has a consistent look and feel
- Appendix – the Appendix will contain detailed documentation for each of the previous listed sections

3 ARCHITECTURE AND DESIGN CONSIDERATIONS

The Bank of Xanadu faced many problems with the current Contract Management System (CMS). The problems not only included being prone to errors, the CMS needed to check to make sure the contract hours worked was correctly recorded, the maximum payable fee was declared and that the work dates are finished for the contractors to be paid correctly. In order to fulfill the requirements given by the Bank of Xanadu we, Team Scrappers (Scrappers), reached a conclusion to develop a centralized database to create a new CMS that would fix problems from the prior CMS and add new features the Bank of Xanadu needs.

We will be using Microsoft Access as our main source to create the centralized database. To insure security, a log in screen must be created for employees of the Accounting department to have access to the database. The data should also be set so that it cannot be deleted once it is entered into the CMS. Which brings up the next point, that the CMS will need a strong validation program or system that will look for errors while the data is being entered before it is submitted for completion.

Assumptions

The system design will proceed with the assumption that there will be no other type of programs used to build the database except Microsoft Access. We will also approach the prototype as a short-term project and it will turn into a long-term project when the whole system will be implemented. We will ensure security by adding log in with password to limit access to the database and aim for high usability as well as scalability. As of the current plan, the CMS will be able to handle hundreds of transactions per day but the number of transaction is subject to change. The CMS will have a multi-user environment, which is accessed by the accounting department of Bank of Xanadu Bellevue Branch.

4 INFORMATION/DATA MODEL

The following is a list of the data requirements that the new system needs for each input/output document, report and table. Each section symbolizes an important entity of the system and the attributes that describe them. For details about each entity and attribute, see the Entity Relationship Diagram and Metadata Dictionary located in the Appendix¹

Contacts (Contact ID, Unit ID, First Name, Last Name, Phone Number, Email)

Contracts (Contract ID, *Programmer ID*, *Unit ID*, *Contact ID*, *Vendor ID*, *Employee ID*, Number, Start Date, End Date, Hourly Rate, Fee Max, Description, Notes)

Divisions (Division ID, Name)

Employees (Employee ID, First Name, Last Name, Contact, User Name, Password, Authorization Level, Job Title)

Invoices (Invoice ID, *Vendor ID*, *Programmer ID*, *Employee ID*, Number, Issue Date, Date Received, Date Paid, Terms, Amount, Rate, Start Date, End Date, Status, Accrual, Notes)

Programmers (Programmer ID, First Name, Last Name)

Units (Unit ID, *Division ID*, Unit Number)

Vendors (Vendor ID, Name, Number, Address, Phone Number, Email)

¹ See Appendix A: Entity Relationship Diagram and Appendix B: Metadata Dictionary

5 USER NAVIGATION DESIGN

Scrappers' new Contract Payment System has been designed based on the current steps being done and requests made by the company. We have now incorporated these processes into a user-friendly interface to ease the processing of invoices for the accounting department. The navigation feature in Access will be the main UI structure for this program. A diagram of user interaction with the system can be found in the appendix². The program will be comprised of the following screens:

- **Logon**

The Logon screen contains the employee user name and password entry fields.

- **Main Menu**

It is topped by three area buttons for Contracts, Invoices, and Reports. In the very bottom area to the far right a button for logging off will be shown.

- **Main Body**

This area will change based upon the selection of the three top area buttons.

- **Main Body > Contracts**

Three sub-menu buttons: New Contract, Contract Exception, Update Contract. Each sub screen will have three common buttons: Save, Cancel, and Print.

- **Main Body > Contracts > New Contract**

It will contain all the entry fields for making a new contract record.

- **Main Body > Contracts > Contract Exception**

This area will contain the fields needed to make and print a contract exception report.

- **Main Body > Contracts > Update Contract**

Screen will be similar to New Contract screen with the exception of only being able to change the contract start/end dates, maximum fee amount, and the contract hourly rate. In addition the contract description and notes are available to edit as well.

- **Main Body > Invoices > New Invoice**

² See Appendix C: Navigation Design

This area will also include functions to handle creating a data entry sheet and making it available to the A/P department.

- **Main Body > Invoices > Invoice Exception**

An area for selecting the vendor which will auto-populate to the body of the report to be turned in. A drop-down list of reasons will be displayed for the user to pick from including a new input text box when Other is selected. Date and Action entry boxes will also be displayed here.

Note: this will need a new table or additional fields for the vendor table.

- **Main Body > Invoices > Update Invoice**

Same as New Invoice except user will only be able to edit/update the address, phone and email fields. A drop-down for selecting the vendor will be at the top to select the vendor.

- **Main Body > Invoices > Vendor Inquiry**

This area will provide a drop-down to select a vendor. In addition, the user can enter the date of getting the inquiry and an adjacent notes text box. A add button will be available for adding a new date and notes fields as the investigation progresses. A closing out inquiry area will include date and reason for closing fields.

- **Main Body > Invoices > Invoice Accrual**

It will have a drop-down to select the vendor. We will add the additional functions after the information get to us.

Note: We need more information on what to put in this area.

- **Main Body > Reports**

Buttons for generating different types of reports will be made available. More details to be provided at a later date

6 INPUTS

The following documents were input documents that were required by the previous systems. Team Scrapper used these source documents to help find information about an entity, its attributes and how it relates to another entity. We will use this information for the new system to help create forms and tables to be used the future. Examples of the source documents are located in the appendix³

Contracts – a sample of a new contract that comes from the buyer.

Contract Extension – a sample of an extension of an existing contract that could not be completed on time.

Invoices – a sample of a new invoice from the vendor.

Time Sheet – a sample of a new time sheet from the contractor that contains dates and hour worked

³ See Appendix D: Inputs

7 OUTPUTS

The following documents were output documents that were required by the previous systems. Team Scrapper used these source documents to help find information about an entity, its attributes and how it relates to another entity. We will use this information for the new system to help create forms and tables to be used the future. Examples of the source documents are located in the appendix⁴

Data Entry Sheet - The sheet is used by accounting when an invoice is sent to accounts payable, contains the vendor information, and amount to be paid

Exception Memo - This report is created when the accounting department finds an error in an invoice and states the reason behind the exception

Invoice Report – This report is sent to accounting and shows invoices that have been paid against the contract programmer expense account. It is used to balance accounts at the end of the month.

Accrual Report – This report is sent to accounting and shows all invoices that have been accrued. It is used to do accrual reversals on the next month.

Contract Programmers Monthly Expense Recap Report – This report is sent to each bank division and shows the contract programmers invoice expenses paid each month.

Contract Programmer Fee Maximum vs. Actual Report – This report is sent to each bank division and shows the contract fee maximum, invoices paid against each contract and the remaining fee maximum available.

Contract Recap Report – This report is sent to each Project Manager and shows the contract and invoice information for all contractors and programmers working for them.

Contract & Programmer (Programmer Name, Vendor Name, Project Manager ,Charge Unit (cost center) , Bank Division, Contract Start Date, Contract End Date, Programmer Hourly Rate, Fee Maximum (Contract budget), Project Description, Contract Number (Accounting dept.))

Problem Invoices to TAM (ID Number, Programmer, Company, Start Date, End Date, Invoice Number, Invoice Amount, Reason, Response, Remarks)

Contract Fee Max (ID Number, Programmer, Invoice Number, Date Paid, Begin Date, End Date, Rate, Total Hours, Total Invoice, Total To Date, Fee Max, Available Funds)

⁴ See Appendix E: Outputs

8 PROCEDURES

External

The procedure of the CMS is as follows.

First, the accounting office will receive the contract from the buyer/contractor. The accountant in charge of the contract will review the contract to make sure all data/input are correctly listed. If there is no bank information within the contract, the information will be added in. By any chance, if an error is found within the contract the contract will be filed for exception. The exception contract will then be returned to the buyer/contractor to fix errors. When there are no errors the contract will be updated and saved.

To pay the invoice the accountant will review the invoice before payment is made. If an error is spotted it will be filed for exception and will be sent to buyer/contractor to fix the errors. When errors are fixed or updates are made the invoice will be saved and updated to be saved. When all information is correct, the invoice will be sent to the Accounts Payable (A/P) department to approve for payment.

The accountant will make sure to review Invoice Status Inquiry for the status of each invoice and if they are payable or not. The accountant will make sure to accrue the invoice.

When the user first opens up the CMS it will prompt up a user selection with password input. When the user name and password is correct the screen will shift to the invoices menu. By pressing the "Log Off" button on the bottom right the user will be sent to the log in screen and by pressing the "Exit" button on the bottom right will close the application.

On the invoices menu the user can add new invoices by clicking on the "ADD NEW" button. This will open up a window with entry area for the user to input the necessary information to create an invoice. When there are errors detected, the box around the area will highlight red and the text will turn yellow. When everything is correct the user can press the save button on the bottom left to save the invoice. To print the selected invoice the user simply needs to press the print on the left side of the page. To update the invoice the user can press the update area of the page. After creating a new invoice or updating one the user then pressed all/refresh to make sure all updates are in place. The pending, approved, accrual and paid buttons only show each type of invoices as declared when being created or updated. The review button shows the invoice without being able to update any part of the invoice.

The contract page acts similar to the invoices as being able to create/update contracts. And sorts each type by active, inactive, invalid and extended. The user can also review each selected contracts.

The divisions, employees, programmers, units and vendors page holds information about the topics they hold.

Currently the reports section is not available but the purpose will be to show summary of all invoices and contracts on a single page.

Use Case 1 Receive Contract

Use Case 2 Add New Bank Information

Use Case 3 Contract Exception

Use Case 4 Update Contract

Use Case 5 Receive Invoice

Use Case 6 Invoice Exception

Use Case 7 Update Invoice

Use Case 8 Pay Invoice

Use Case 9 Invoice Status Inquiry

Use Case 10 Accrue Invoice

Internal

Here are short examples of use cases used in the new CMS. Complete information is included within the Appendix⁵.

Name	frmMain3
Description	This form holds all data from other forms and their query for easy navigation
User	Accountants
Tables	Contacts, Contracts, Divisions, Employees, Invoices, Programmers, Units, Vendors

Name	frmContacts
Description	This form holds data regarding information on Contacts
User	Accountants
Tables	Contacts

Name	frmContracts
Description	This form holds data regarding all information on Contracts
User	Accountants
Tables	Contacts

Name	frmDivisions
Description	This form holds data regarding information

⁵ See Appendix F: Use Cases & Diagram

	on Divisions
User	Accountants
Tables	Divisions

Name	frmEmployees
Description	This form holds data regarding information on Employees
User	Accountants
Tables	Employees

Name	frmInvoices
Description	This form holds data regarding information on Invoices
User	Accountants
Tables	Invoices

Name	frmProgrammers
Description	This form holds data regarding information on Programmers
User	Accountants
Tables	Programmers

Name	frmReports
Description	This form holds data regarding information on Reports
User	Accountants
Tables	Reports

Name	frmUnits
------	----------

Description	This form holds data regarding information on Units
User	Accountants
Tables	Units

Name	frmVendors
Description	This form holds data regarding information on Vendors
User	Accountants
Tables	Vendors

9 INTERFACE DESIGN AND CODING STANDARDS

External

This section provides a description of all the user interface features that will be on all screens. Scrappers will decide all menus and data entry screen with an approval from Bank of Xanadu. The Interface Design Standards will include all background screen colors to be silver (#cccccc). The text font will be Source Sans Pro, the color of the headline will be Navy blue (#2298cd) and the body text will be gray (#6f6f6f). The area of the form will determine the font size (for more detailed information, see the Appendix).

The buttons used on the main menus and on all forms will be gray background (#bfbfbf) and the size will be 130x45px. Command buttons will be located at the bottom of all forms. Data entry screen will use 11-point Verdana font. For data entry screen field will use radio buttons and drop down list to select a value (For more information, see the Appendix).

Internal

Scrappers' standard format for commenting code is required to use a descriptive block before each procedure and may include individual line for clarity. More specific guidelines are listed in the Design Standards Document in the Appendix⁶.

Scrappers will follow the Microsoft recommended standards naming convention for variables and objects in code (for more detailed information, see the Appendix). It is recommended to use camel case, although underscores are also allowed when writing variables and objects name. A prefix is required to indicate the type of value variables (e.g. string or integer) or objects (e.g. form or image).

⁶ See Appendix G: Design Standards Document

10 CONCLUSION

This concludes the system design specifications. The information in this document will provide enough information to finish the design phase of the new invoicing system. Following this section is the appendix that contains in depth specifications of design considerations and standards to be used, the information required by the system, how users will interact with the system, and examples of inputs, outputs, and reports. Also located in the appendix are sections detailing potential issues⁷ and possible future enhancements⁸.

Team scrappers' requests that we could schedule a meeting at the Bellevue branch on April 14th. At this meeting, we will demonstrate a prototype of the new invoicing system to be approved by management. After approval of the new system team scrappers will move forward with the project and begin the implementation and testing phases of the.

X

Dave Spencer

⁷ See Appendix H: Issues

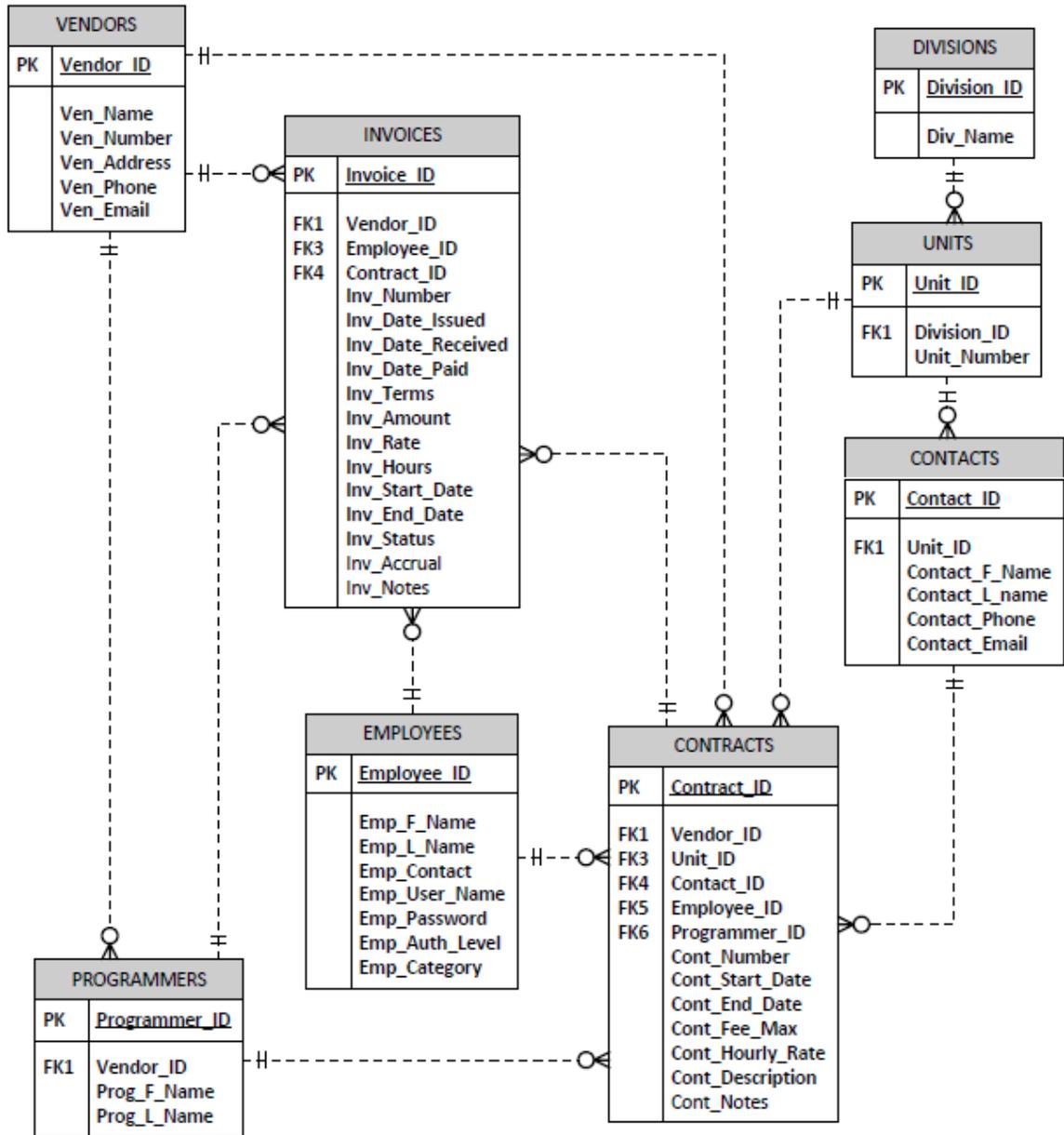
⁸ See Appendix I: Future list

11 APPENDIX

- A. Entity Relationship Diagram
- B. Metadata Dictionary
- C. Navigation Design
- D. Input
- E. Output
- F. Use Cases
- G. Design Standards Document
- H. Issues
- I. Futures List

11.1 APPENDIX A: ENTITY RELATIONSHIP DIAGRAM

Bank of Xanadu ERD



Bank of Xanadu
Entity Relationship Diagram

11.2 APPENDIX B: METADATA DICTIONARY

<i>Entity Name</i>	<i>Attribute Name</i>	<i>Definition</i>	<i>Domain Constraints</i>	<i>Referential integrity constraints</i>
Contacts	Contact_ID	The ID number assigned to a Contact	Required System Assigned Unique Numeric	Primary Key
	Unit_ID	The ID number assigned to a Charge Unit	Required Numeric Not Unique	Foreign Key (Unit) a Contact cannot exist without having a related Unit record
	Contact_F_Name	The First Name of the Contact	Required Not unique char	
	Contact_L_Name	The Last Name of the Contact	Required Not unique char	
	Contact_Phone	The Contacts Phone Number	Required Not unique Input Mask: (###)-###-####	
	Contact_Email	The Contacts Email Address	Required Unique Short Text	
Contracts	Contract_ID	The ID number assigned to a Contract	Required System Assigned Unique Numeric	Primary Key
	Vendor_ID	The ID number assigned to a Vendor	Required Not unique Numeric	Foreign Key (Vendor) a Contract cannot exist without having a related Vendor record
	Unit_ID	The ID number assigned to a Charge Unit	Required Not unique Numeric	Foreign Key (Unit) a Contract cannot exist

				without having a related Unit record
	Contact_ID	The ID number assigned to a Contact	Required Numeric Not Unique	Foreign Key (Contact) a Contract cannot exist without having a related Contact record
	Employee_ID	The ID number assigned to an Employee	Required Not unique Numeric	Foreign Key (Employee) a Contract cannot exist without having a related Employee record
	Programmer_ID	The ID number assigned to a Programmer	Required Not unique Numeric	Foreign Key (Programmer) a Contract cannot exist without having a related Programmer record
	Cont_Number	The Number assigned to a Contract	Required Unique System Assigned Numeric	
	Cont_Start_Date	The Date the Contract goes into effect	Required Non Unique Input Mask: ##-##-####	
	Cont_End_Date	The Date the Contract Expires	Required Non Unique Input Mask: ##-##-####	
	Cont_Fee_Max	The Maximum allowed payment on a single Contract	Required Non Unique Numeric	
	Cont_Hourly_Rate	The Hourly Rate at which	Required Non Unique	

		programmers are paid	Numeric	
	Cont_Description	A Description of the work to be done on a contract	Required Unique Char	
	Cont_Notes	An optional field describing any relevant information that could be useful	Optional Char	
	Cont_Status	The current state of contract during processing	Required Valid values: Active, inactive, invalid, extended	
Division	Division_ID	The ID number assigned to a Division	Required Unique System Assigned Numeric	Primary Key
	Div_Name	The Name of a division of the bank	Required Not unique Char	
Employees	Employee_ID	The ID number assigned to an Employee	Required Unique System Assigned Numeric	Primary Key
	Emp_F_Name	The First Name of an Employee	Required Not unique Char	
	Emp_L_Name	The Last Name of an Employee	Required Not unique Char	
	Emp_Contact	The Employees Person of Contact	Required Not unique Char	
	Emp_User_Name	The Username of	Required Unique	

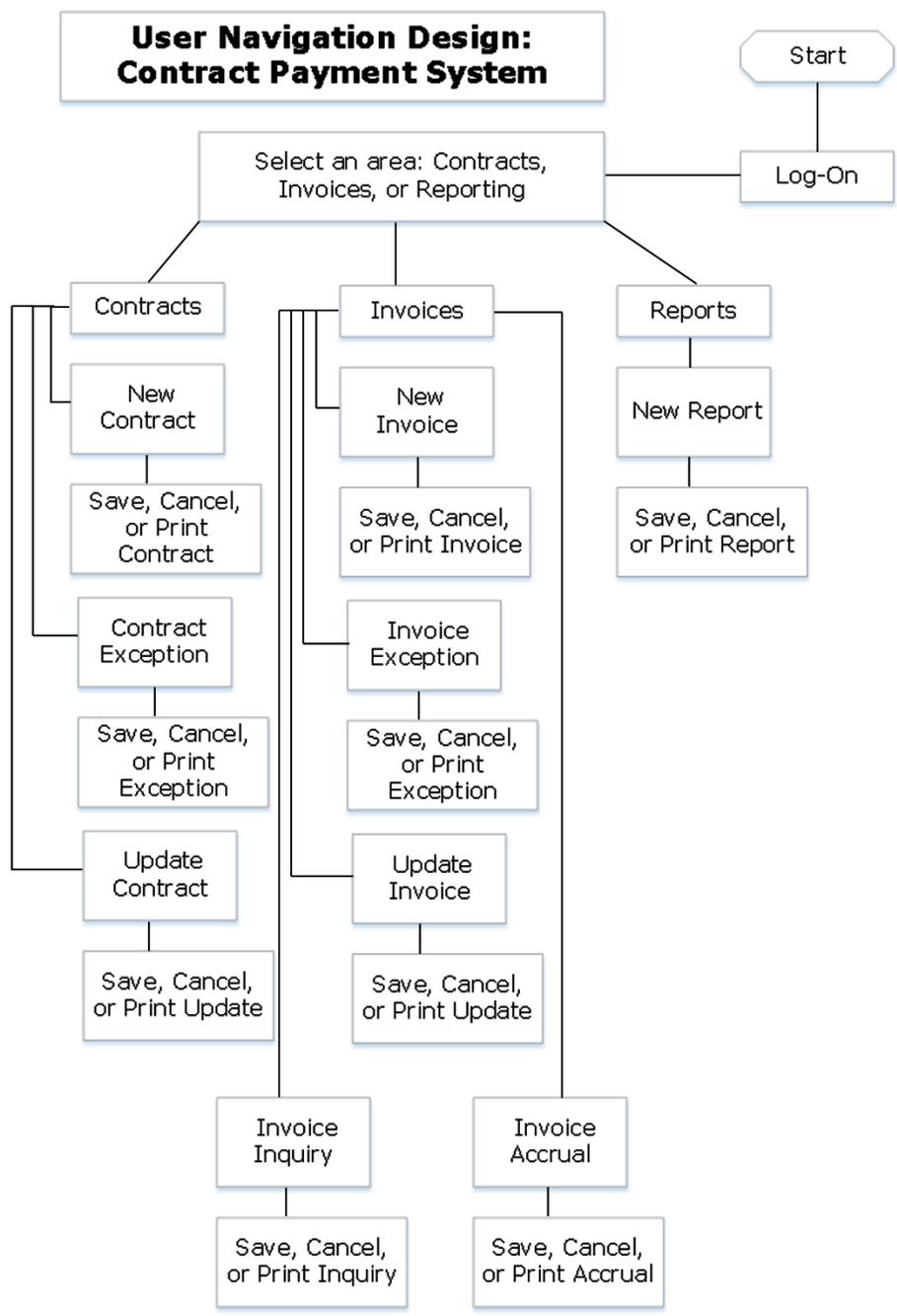
		an Employee to login to the system	Short Text	
	Emp_Password	The Password used by an Employees to login to the system	Required Unique Short text	
	Emp_Auth_Level	The Max level of Authorization that an Employee can access	Required Not unique Valid Values: ACCT, MGMT, A/P	
	Emp_Category	The Position of an Employee	Required Not unique Valid Values: ACCT, MGMT, A/P	
Invoices	Invoice_ID	The ID number assigned to an Invoice by the programmer who did the work	Required Unique System Assigned Numeric	Primary Key
	Vendor_ID	The ID number assigned to a Vendor	Required Not unique Numeric	Foreign Key (Vendor) an Invoice cannot exist without having a related Vendor record
	Employee_ID	The ID number assigned to an Employee	Required Not unique Numeric	Foreign Key (Employee) an Invoice cannot exist without having a related Employee record
	Contract_ID	The ID number given to a Contract	Required Not Unique Numeric	Foreign Key (Contract) an Invoice cannot exist without having

				a related Contract record
	Inv_Number	The Number that is assigned to an Invoice	Required Unique System Assigned	
	Inv_Date_Issued	The Date of when the Invoice is created	Required Not Unique Input Mask: ##-##-####	
	Inv_Date_Received	The Date that the Invoice is received by accounting	Required Not Unique Input Mask: ##-##-####	
	Inv_Date_Paid	The Date that the invoice was paid on	Required Not Unique Input Mask: ##-##-####	
	Inv_Terms	The Terms of how the invoice will be paid	Required Not Unique Valid Values: PIA, Net 7-90, EOM, COD, 1% 10 Net 30, ETC.	
	Inv_Amount	The Total Amount of an Invoice	Required Not Unique Numeric	
	Inv_Rate	The Rate at which the contractor is paid	Required Not Unique Numeric	
	Inv_Hours	The total amount of time in hours that the contractor worked	Required Not Unique Numeric	
	Inv_Start_Date	The first day that a Programmer started working	Required Not Unique Input Mask: ##-##-####	
	Inv_End_Date	The last day that the	Required Not Unique	

		Programmer worked on	Input Mask: ##-##-####	
	Inv_Status	The current state of the invoice during processing	Required Not Unique Valid Values: Pending Invoice, Approved, Accrual Needed, Paid, credit memo issued	
	Inv_Accrual	The Date that the invoice is Accrued	Optional Not Unique Input Mask: ##-##-####	
	Inv_Notes	An Notes field that could contain any relevant information needed by an invoice	Optional Char	
Programmers	Programmer_ID	The ID number assigned to a Programmer	Required Unique System Assigned Numeric	Primary Key
	Vendor_ID	The ID number assigned to a Vendor	Required Not unique Numeric	Foreign Key (Vendor) A Programmer cannot exist without having a related Vendor record
	Prog_F_Name	The Programmers First Name	Required Not unique Char	
	Prog_L_Name	The Programmers Last Name	Required Not unique Char	
Units	Unit_ID	The ID number given for a Charge	Required Unique System	Primary Key

		Unit	Assigned Numeric	
	Division_ID	The ID number assigned to a Division	Required Not unique Numeric	Foreign Key (Division) A Unit cannot exist without having a related Division record
	Unit_Number	The Number assigned to a Charge Unit	Required System Assigned Unique	
Vendors	Vendor_ID	The ID number assigned to a Vendor	Required Unique System Assigned Numeric	Primary Key
	Ven_Name	The Name of the Vendors Company	Required Not unique Char	
	Ven_Number	Number assigned to Vendor	Required System Assigned Unique Numeric	
	Ven_Address	The Address of the Vendor	Required Not Unique Short Text	
	Ven_Phone	The Phone Number to contact a Vendor	Required Not unique Input Mask: (###)-###- ####	
	Ven_Email	The Email to contact a Vendor	Required Unique Short Text	

11.3 APPENDIX C: NAVIGATION DESIGN



11.4 APPENDIX D: INPUT

Sample Contract

APPENDIX A

AGREEMENT TO PROVIDE PERSONNEL BETWEEN
Bank of Xanadu (BANK) and:

 DAN VAN RITZ, INC. (CONTRACTOR)

I. All work and/or services provided under this Appendix shall be performed in accordance with the provisions of this Appendix and Master Agreement: #90-3167

Project/Services Number 16358.000 Charge Unit #: 3620

Bank Project Manager/Phone: Peter Townsend 206-675-2696 

II. Scope of Services:
A. PROVIDE AN OVERVIEW OF THE PROJECT:
Support product development projects, as well as acquisition preparation for Demand Deposit Systems.

(See attached sheet for continuation of Scope of Services) RITZO415 

III. Fee Schedule: Total fee shall not exceed \$ 52,000.00 

Name of Individual	Generic Job Level	Hourly Rate	Start Date	End Date
Dan Van Ritz	CSE	\$65.00	12/16/14	4/15/15

A NEW APPENDIX A MUST BE EXECUTED TO AUTHORIZE MENT BEYOND THE AMOUNT NOTED ABOVE IN III. FEE SCHEDULE, OR TO AUTHORIZE WORK  BEYOND THE COMPLETION DATE NOTED ABOVE

Agreed and Accepted:
DAN VAN RITZ, INC. (Contractor)
Signature: Dan Van Ritz
Vendor Officer: DAN VAN RITZ
Title: President
Date: 12/15/14

Agreed and Accepted:
BANK OF XANADU (Bank)
Signature: Maryanne Kerrigan
Name: Maryanne Kerrigan
Title: Vice President
Date: 12/14/14

Countersigned: Charles Skeateas
Name: Charles Skeateas
Title: Vice President
Date: 12/15/14

Send Invoices to:
Bank of Xanadu
General Accounting #3707
P.O. Box 37000
Bellevue, WA 98002
Attn: Dave Spencer

Bryce Hazen
Bryce Hazen, Senior Vice President

AGREEMENT TO PROVIDE PERSONNEL BETWEEN
Bank of Xanadu (BANK) and:

_____ DAN VAN RITZ, INC. _____ (CONTRACTOR)

II. Scope of Services Continued:

B. LIST THE SPECIFIC TASKS TO BE PERFORMED:

Complete systems design specification
Analyze and code in C# (C sharp)
Perform unit, system, and integration testing
Provide installation support

C. LIST THE DELIVERABLES EXPECTED TO BE PRODUCED:

Detailed design specifications
Code
Test specifications
Unit testing, system testing
Conversion specifications
Installation specifications

D. LIST THE SPECIFIC TECHNICAL EXPERTISE REQUIRED (HARDWARE, OPERATING SYSTEMS, PROGRAMMING LANGUAGES, ETC.)

IBM 30XX, TSO/ISPF, OS JCL, VSAM
Ability to analyze and code in C# (C sharp)
Design, coding, and testing skills
Accounting systems background required, banking preferred.
Deposit systems/prior acquisition experience a plus
Prior Bank of Xanadu experience a plus
Strong communications and documentation skills
Team player with good interpersonal skills

E. LIST THE PERFORMANCE STANDARDS THAT WILL BE USED TO DETERMINE QUALITY OF WORK (E.G. SDP, DOCUMENTATION STANDARDS, TESTING STANDARDS, ETC.)

Adherence to project standards
Code reviews
SDP
Test plans and test result reviews

Sample Contract Extension

MEMO TO: Rob Watt
TAM #3411

MEMO FROM: Maryanne Kerrigan
Vice President
Consumer Banking Division – Demand Deposit Services
Project Management & Technology Support #3620
Xnet 666-1464

COPY TO: Peter Townsend #3620
Jim Herrington #3620
Frank Smolski #3620

DATE: April 15, 2015

SUBJECT: Van Ritz Contract Extension – Demand Deposit Project

The **Completion Date** ("End Date") on the Van Ritz Programming Service Agreement, master Agreement #90-3167, has been extended to May 15, 2015. The **Total Fees** ("Fee Schedule") do NOT change; they will not exceed \$52,000. 

Please make note of this change in your files.

Thanks for your help and call me if any questions.

RITZ0415 

Maryanne Kerrigan

Sample Invoice

Dan Van Ritz Consulting, Inc

5820 Stoneridge Road Suite 100
 Issaquah, WA 98506
 425-555-1212

INVOICE

INVOICE #100154

DATE: JANUARY 16, 2015

TO:
 Bank of Xanadu
 General Accounting #3707
 P.O. Box 37000
 Bellevue, WA 98002

FOR:
 Master Agreement #90-3167
 Charge Unit #3620

<u>PERIOD:</u>	1/1/15 – 1/15/15	<u>TERMS:</u>	ON RECEIPT
HOURS	DESCRIPTION	RATE	AMOUNT
88	Computer Programming/Consulting Services	\$85.00	\$5,720.00
	<i>RITZ0415</i>		
TOTAL			\$5,720.00

Make all checks payable to Dan Van Ritz Consulting, Inc

Thank you for your business!

Sample Time Sheet



TIME SHEET

PERIOD OF: 1/1/2015 – 1/15/2015

DAN VAN RITZ CONSULTING, INC.
5820 Stoneridge Road Suite 100
ISSAQUAH, WA 98506
425-555-1212

CONTRACTOR NAME: Dan Van Ritz	TITLE: Programmer/Consultant
CLIENT COMPANY: Bank Of Xanadu	SUPERVISOR: Peter Townsend

CALENDAR DAY	HOURS WORKED	CALENDAR DAY	HOURS WORKED
1	8	16	
2	8	17	
3		18	
4		19	
5	8	20	
6	8	21	
7	8	22	
8	8	23	
9	8	24	
10		25	
11		26	
12	8	27	
13	8	28	
14	8	29	
15	8	30	
		31	
TOTAL HOURS:	88		
EMPLOYEE SIGNATURE: <i>Dan Van Ritz</i>		DATE: 1/15/2015	
SUPERVISOR SIGNATURE: <i>Peter Townsend</i>		DATE: 1/16/2015	

11.5 APPENDIX E: OUTPUT

Data Entry Sheet

DATA ENTRY SHEET

Vendor Name: Donny Wicks Associates

Vendor Number: ZZ0002

Invoice Number: 329

Description: A. Peckham 12/16/14 to 12/31/14

Invoice Date: 01/02/15

Due Date: 01/17/15

Invoice Total: 3,600.00

G/L Account: 507613

P.O. Number: A. Peckham

Charge Unit: 9408

Processed by Dave Spencer 1/11/15

Exception Memo



Bank of Xanadu

Date: January 11, 2015

From: Dave Spencer, Accountant
Financial Controller's Division
Corporate General Accounting #3707

To: Rob Watt, Buyer
Technology Acquisition Management #3411

Classification: Internal

Subject: CONTRACTOR INVOICE PROBLEMS

Vendor:

I am unable to process the attached invoice(s) for the following reason(s):

<input type="checkbox"/>	No Contract on File
<input type="checkbox"/>	Dollar Amount Exceeds Contract Fee by \$
<input type="checkbox"/>	Invoice Period Outside of Contract Dates
<input type="checkbox"/>	No Time Sheet
<input type="checkbox"/>	No Invoice/Time Sheet Approval
<input type="checkbox"/>	Time Sheet & Invoice Discrepancy
<input type="checkbox"/>	Billed Rate Different from Contract Rate
<input type="checkbox"/>	Other:

Please provide the necessary information and return to me in unit #3707. Thanks you for your assistance in resolving these problems. If you have any questions, please call me at XanaduNet 785-1223.

Attachment included.

DATE	ACTION

	A	B	C	D	F	G	H	I	J	K	L	M	N	O
1	Invoices													
2														
3	ID Number	Programmer	Vendor	Charge	Invoice #	Date Paid	Begin Date	End Date	Rate	Total Hours	Total Invoice	Accrued	Memo	
4	Wilki0515	Wilkins, Peter	Donny Wicks Associates	9408	1001	12/21/14	12/01/14	12/15/14	59.00	64.0	3,776.00			
5										Total:	3,776.00			
6										Total for December:	3,776.00			
7														
8	Peckh0915	Peckham, Art	Donny Wicks Associates	9408	329	01/11/15	12/16/14	12/31/14	60.00	60.0	3,600.00	12/14		
9	Wilki0515	Wilkins, Peter	Donny Wicks Associates	9408	1002	01/11/15	12/16/14	12/31/14	59.00	66.0	3,894.00	12/14		
10	Brown0315	Brown, Lou	EDS Temps Inc	3072	509	01/11/15	12/17/14	12/31/14	25.00	70.0	1,750.00	12/14	Dec Exp	
11										Total:	9,244.00		13,020.00	
12														
13	Wilki0515	Wilkins, Peter	Donny Wicks Associates	9408	1003	01/25/15	01/02/15	01/15/15	59.00	85.0	5,015.00			
14	Brown0315	Brown, Lou	EDS Temps Inc	3072	510	01/25/15	01/02/15	01/15/15	25.00	68.0	1,700.00			
15										Total:	6,715.00			
16										Total for January:	15,959.00			
17														
18	Lehre1215	Lehrer, Philip	Beltam Systems Inc	3117	101	02/08/15	01/02/15	01/31/15	52.00	165.0	8,580.00	01/15		
19	Peckh0915	Peckham, Art	Donny Wicks Associates	9408	330	02/08/15	01/02/15	01/31/15	60.00	177.0	10,620.00	01/15		
20	Wilki0515	Wilkins, Peter	Donny Wicks Associates	9408	1004	02/08/15	01/16/15	01/31/15	59.00	82.0	4,838.00	01/15		
21	Brown0315	Brown, Lou	EDS Temps Inc	3072	511	02/08/15	01/16/15	01/31/15	25.00	70.0	1,750.00	01/15		
22	forti0615	Fortier, Brian	EDS Temps Inc	3072	3723	02/08/15	01/02/15	01/31/15	25.00	176.5	4,412.50	01/15	Jan Exp	
23										Total:	30,200.50		36,915.50	
24														
25	Brown0315	Brown, Lou	EDS Temps Inc	3072	512	02/22/15	02/01/15	02/15/15	25.00	68.0	1,700.00			
26	(start)													
27										Total:	1,700.00			
28										Total for February:	31,900.50			
29														
30														
31										Grand Total:	51,635.50			

Accrual Report

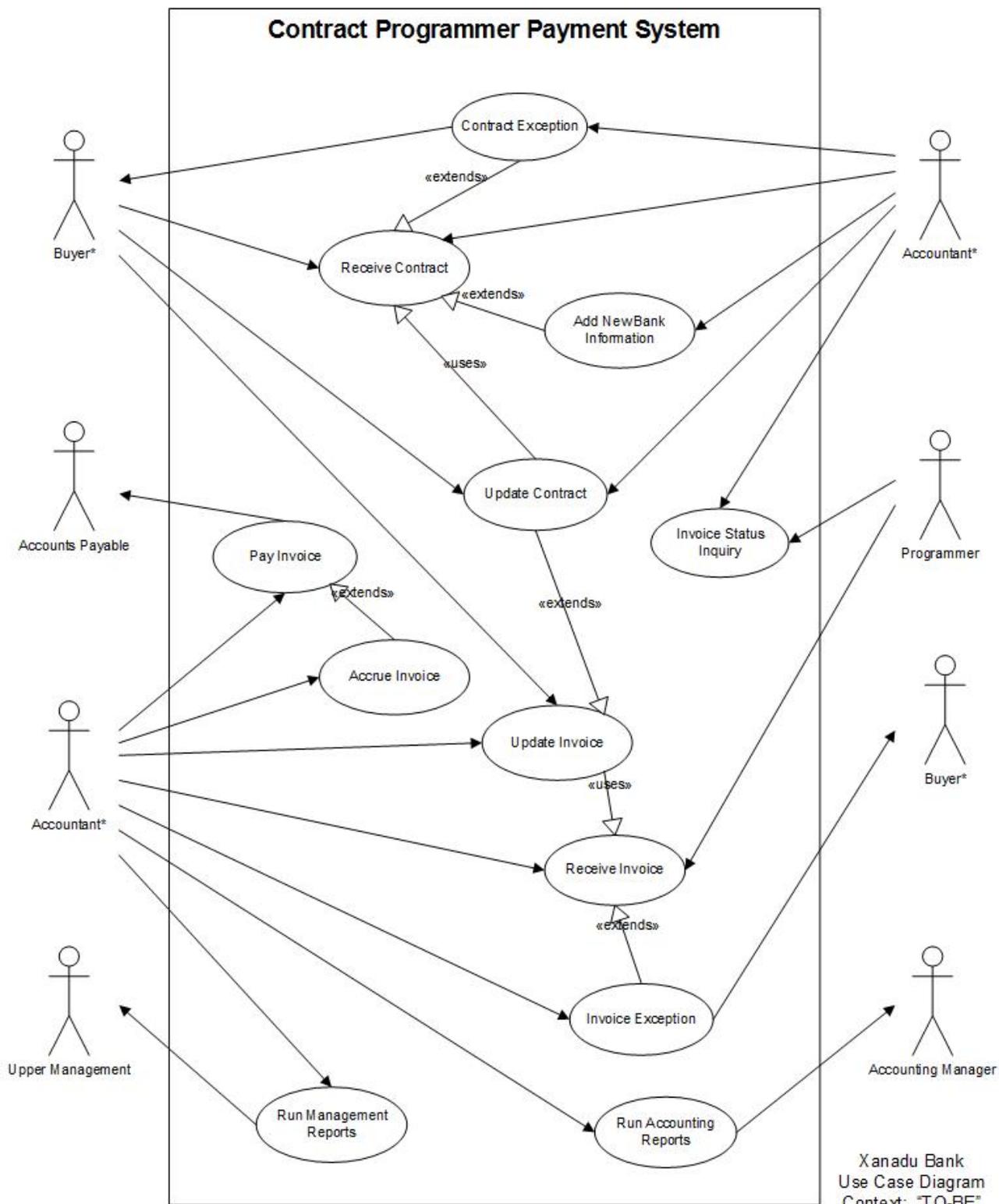
	B	C	D	F	L	M	O
1	Accruals						
2							
3	Programmer	Vendor	Charge	Invoice #	Total Invoice	Accrued	Reversed
4	Brown, Lou	EDS Temps Inc	3072	509	1,750.00	12/14	
5			3072	Total:	1,750.00		01/10/15
6	Peckham, Art	Donny Wicks Associates	9408	329	3,600.00	12/14	
7	Wilkins, Peter	Donny Wicks Associates	9408	1002	3,894.00	12/14	
8			9408	Total:	7,494.00		01/10/15
9			December 2014		9,244.00		
10							
11	Brown, Lou	EDS Temps Inc	3072	511	1,750.00	01/15	
12	Fortier, Brian	EDS Temps Inc	3072	3723	4,412.50	01/15	
13			3072	Total:	6,162.50		02/10/15
14	Lehrer, Philip	Beltam Systems Inc	3117	101	8,580.00	01/15	
15			3117	Total:	8,580.00		02/10/15
16	Peckham, Art	Donny Wicks Associates	9408	330	10,620.00	01/15	
17	Wilkins, Peter	Donny Wicks Associates	9408	1004	4,838.00	01/15	
18			9408	Total:	15,458.00		02/10/15
19			January 2015		30,200.50		
20							
21							
22			February 2015				
23							
24			Grand Total:		39,444.50		
25							
26							

Contract Programmers Monthly Expense Recap Report

	B	C	D	E	F	H	I	K	L	M
2	By Division and Unit									
3	January 2015									
4										
5	Programmer	Vendor	Division	Charge	Invoice #	Begin Date	End Date	Total Hours	Total Invoice	Accrued
6			Division: AMB					Total for Division:	8,580.00	
7	Lehrer, Philip	Beltam Systems Inc	AMB	3117	101	01/02/15	01/31/15	165.0	8,580.00	01/15
8				3117				Total for Charge Unit:	8,580.00	
9										
10			Division: CCR					Total for Division:	20,473.00	
11	Peckham, Art	Donny Wicks Associates	CCR	9408	330	01/02/15	01/31/15	177.0	10,620.00	01/15
12	Wilkins, Peter	Donny Wicks Associates	CCR	9408	1003	01/02/15	01/15/15	85.0	5,015.00	
13	Wilkins, Peter	Donny Wicks Associates	CCR	9408	1004	01/16/15	01/31/15	82.0	4,838.00	01/15
14				9408				Total for Charge Unit:	20,473.00	
15										
16			Division: NAB					Total for Division:	7,862.50	
17	Brown, Lou	EDS Temps Inc	NAB	3072	510	01/02/15	01/15/15	68.0	1,700.00	
18	Brown, Lou	EDS Temps Inc	NAB	3072	511	01/16/15	01/31/15	70.0	1,750.00	01/15
19	Fortier, Brian	EDS Temps Inc	NAB	3072	3723	01/02/15	01/31/15	176.5	4,412.50	01/15
20				3072				Total for Charge Unit:	7,862.50	
21										
22								Grand Total for January:	36,915.50	
23										
24	Contract Programmers Monthly Expense Recap Report									
25	By Division and Unit									
26	February 2015									
27										

Contract Programmer Fee Maximum vs. Actual Report

11.6 APPENDIX F: USE CASES



Xanadu Bank
 Use Case Diagram
 Context: "TO-BE"
 Last Update: 4/24/08

Use Case Name:	RECEIVE CONTRACT	ID: UC001
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps for processing a new contract, from the time that it is delivered by the buyer, until a new contract is verified and entered into the system.	
Trigger:	New contract is delivered to the accounting department	
Related Use Cases:	Contract Exception (extended by); Add New Contract Information (extended by); Update Contract (used by)	
Normal flow of events:	<p>This use case begins when the Buyer delivers a new contract to the Accountant.</p> <ol style="list-style-type: none"> 1) Manually review contract to ensure that all the information needed by the accounting department is on the contract. 2) Log onto the system and navigate to the "Enter Contract" screen. 3) Search for the correct Vendor (Contractor) Number and select it. 4) Enter all the required contract information (see Information Requirements below) into the system. Use appropriate "lookups" when applicable. 5) When finished entering all required information, SAVE the new contract record into the system. <p>This use case ends when the new contract is entered into the system.</p>	
Exception(s):	<ol style="list-style-type: none"> 1) If any required information is missing or invalid, an exception memo is created and sent to the buyer for resolution. 3) If the vendor is not listed, navigate to the "Create Vendor" screen and create a new vendor record. 4) If the contact (project manager), charge unit, or bank division is not listed in the appropriate lookup fields, a new record for that information will need to be created. 	
Pre-condition(s):	The existance of a new contract deliverd from the contract group	
Post-conditions(s)	The verified contract has been entered into the system and is ready to have valid invoices processed against it.	
Information Requirements:	Contract ID Programmer Vendor Begin Date End Date Charge Unit Bank Division Hourly Fee Fee Maximum Project Manager PM contact unit PM phone number Project Description	
Assumptions:	The accountant must refer to the corporate directory to verify the correct contact unit for the project manager.	
Business Rules:	<ol style="list-style-type: none"> 1) A contract is not considered valid if any of the required information is missing, and must be returned to the buyer for correction. 2) A contract can be for more than one programmer working for the same vendor. 3) A programmer may be working on more that one contract at a time 4) If the PM is not listed in the corporate directory the signing authority needs to be contacted to obtain that information. 	

Use Case Name:	CONTRACT EXCEPTION	ID: UC003
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps for processing a contract 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.	
Trigger:	An incomplete or invalid contract is received from the Buyer	
Related Use Cases:	Receive Contract (extends)	
Normal flow of events:	<p>This use case begins when the Buyer delivers a contract to the Accountant that is either incomplete or contains invalid information.</p> <ol style="list-style-type: none"> 1) A manual review of the contract determines that one of the required pieces of information required to enter a contract into the system is missing or invalid. 2) Enter the contract into the system with as much information as possible. 3) Enter "missing/invalid" or default to "zero" value in the field for the piece(s) of information that is missing or invalid 4) Enter the date and reason for the contract return in the "Contract Notes" field 5) SAVE the contract record into the system 6) Generate a return memo to the Buyer explaining the reason for the return 7) Attach the return memo to the contract and send it back to the Buyer <p>This use case ends when the incomplete/invalid contract has been returned to the Buyer.</p>	
Exception(s):	None	
Precondition(s):	A contract has been received that has missing or invalid information.	
Postconditions(s)	The incomplete or invalid contract has been returned to the Buyer	
Information Requirements:	(See "Receive Contract" use case UC001) Contract Notes	
Assumptions:	The Buyer will be able to supply the missing information or correct the invalid information	
Business Rules:	<ol style="list-style-type: none"> 1) A contract is not considered valid if any of the required information is missing and must be returned to the Buyer for correction 2) It is the Buyer's responsibility to correct any errors in the contract and return it to the Accountant who sent it back. 	

Use Case Name:	UPDATE CONTRACT	ID: UC004
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps for updating a contract, from the time that it is delivered by the Buyer, until the updated contract information has been entered into the system.	
Trigger:	An updated or revised contract is received from the Buyer	
Related Use Cases:	Receive Contract (uses); Update Invoice (extends)	
Normal flow of events:	<p>This use case begins when the Buyer delivers a corrected or updated contract to the Accountant.</p> <ol style="list-style-type: none"> 1) Manually review the contract to ensure all the information needed by the accounting department is on the contract 2) Search for the contract in the system 3) Change the fields that have new or revised values OR missing or zero values by entering the correct information from the updated contract 4) Enter the date returned and any additional information in the "Contract Notes" field 4) SAVE the updated contract record into the system <p>This use case ends when the contract has been correctly and completely updated in the system.</p>	
Exception(s):	None	
Precondition(s):	An updated contract has been received from the Buyer	
Postconditions(s)	A complete and valid contract has been updated in the system	
Information Requirements:	(see Receive Contract - UC001)	
Assumptions:	The system will be able to accept the updated contract information	
Business Rules:	The accountant must enter the updated contract information into the system and make a note of the date that the updated contract was returned by the Buyer	

Use Case Name:	RECEIVE INVOICE	ID: UC005
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps for processing a new invoice, from the time that it is received from the Programmer/Vendor, until the invoice has been entered into the system and approved for payment.	
Trigger:	An invoice for programming services is received from a Programmer/Vendor	
Related Use Cases:	Invoice Exception (extended by), Update Invoice (used by)	
Normal flow of events:	<p>This use case begins when a new invoice is received for programming services.</p> <ol style="list-style-type: none"> 1) Manually review the invoice to ensure that all the information needed by the accounting department is on the invoice (including an authorized time sheet) 2) Log onto the system and navigate to the "Enter Invoice" screen 3) Search for the correct Vendor (Contractor) number and select it 4) Search for the correct Contract ID and select it 5) Enter all the required invoice information (see Information Requirements below) into the system. Use appropriate "lookups" when applicable. 6) Run a system check to ensure that the dates for programming services fall within the date range specified on the contract 7) Run a system check to ensure that the billed rate is the same as the "Hourly Fee" on the contract 8) Run a system check to ensure that the dollar amount of the invoice does not exceed the "Fee Maximum" amount on the contract (must consider all previous invoices that have been paid against the contract fee maximum) 9) When finished entering all required information and validating that the invoice services fall between the valid contract dates, AND the billed rate matches the contract, AND the total dollar amount of the invoice has not exceeded the contract fee maximum, select "Approved to Pay" as the Payment Status. 10) SAVE the invoice record into the system <p>This use case ends when the invoice is entered into the system with an "Approved to Pay" status.</p>	
Exception(s):	<ol style="list-style-type: none"> 1) If any necessary information is missing from the invoice (including the time sheet), an exception memo is created and sent with the invoice to the Buyer for resolution. 3) If the Vendor is not listed (may indicate no valid contract), an exception memo is created and sent with the invoice to the Buyer for resolution (see next exception). 4) If the "Contract ID" cannot be located in the system (but there is a valid Vendor), an exception memo is created and sent with the invoice to the Buyer for resolution. 5) If the dates of service is outside those specified on the contract, an exception memo is created and sent with the invoice to the Buyer for resolution. 6) If the billed hourly rate for services does not match that specified on the contract, an exception memo is created and sent with the invoice to the Buyer for resolution. 7) If the total dollar amount of the invoice causes the maximum contract fee to be exceeded, an exception memo is created and sent with the invoice to the Buyer. 	
Precondition(s):	An invoice has been received for contract programming services	
Postcondition(s)	The invoice has been verified for payment and entered into the system	
Information Requirements:	Vendor Remit-to Address Contract ID Invoice Number Invoice Date Programmer Service Start Date Service End Date Hourly Rate	

Use Case Name:	ADD NEW BANK INFORMATION	ID: UC002
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps for creating a new vendor, bank contact, bank unit, or bank division record, from the time a contract is received with any of these new pieces of information, until a new record(s) is entered into the system.	
Trigger:	A contract is delivered to the accounting department with new vendor, contact, unit, or division information.	
Related Use Cases:	Receive Contract (extends)	
Normal flow of events:	<p>This use case begins when the Buyer delivers a contract with new vendor, contact, unit, or division information to the Accountant.</p> <ol style="list-style-type: none"> 1) Search for the correct Vendor (Contractor) Number and cannot find one. 2) Navigate to the "Create Vendor" screen. 3) Enter the required vendor name into the system. 4) Search for the correct Contact Person and cannot find one. 5) Navigate the the "Create Contact" screen. 6) Enter the required bank contact name into the system. 7) Search for the correct Charge Unit and cannot find one. 8) Navigate to the "Create Unit" screen. 9) Enter the required bank unit number into the system 10) Search for the correct Bank Division and cannot find one. 11) Navigate to the "Create Division" screen. 12) Enter the required bank division name into the system. 13) When finished entering any of the required information above, SAVE the new record into the system. <p>This use case ends when the new vendor, contact, unit, or division record is entered into the system.</p>	
Exception(s):	None.	
Pre-condition(s):	The existance of a contract with new vendor, contact, unit, or division information.	
Post-conditions(s)	The new vendor, contact, unit, or division information has been entered into the system..	
Information Requirements:	Vendor Name Contact Person (Project Manager) Charge Unit Bank Division	
Assumptions:	The accountant must refer to the corporate directory to verify the correct contact unit for the project manager.	
Business Rules:	1) In order to create a new contract record, valid vendor, contact, unit, and division information must be obtained and exist in the new system.	

Use Case Name:	INVOICE EXCEPTION	ID: UC006
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps for processing an invoice exception memo to return an incomplete, invalid, or unpayable invoice to the Buyer, from the time the incomplete, invalid, or unpayable invoice is received until it has been returned to the Buyer.	
Trigger:	An incomplete, invalid, or unpayable invoice is received from the Vendor	
Related Use Cases:	Receive Invoice (extends)	
Normal flow of events:	<p>This use case begins when the Vendor sends an invoice for programming services to the Accountant (should arrive via the Project Manager).</p> <ol style="list-style-type: none"> 1) A manual review of the invoice determines that one of the required pieces of information required to enter the invoice into the system for payment is missing (see Information Requirements below), OR, in the process of running a system check, it is determined that the invoice dates of service exceed those on the contract, or the "Hourly Fee" does not match with that of the contract, or payment of the invoice would cause the "Fee Maximum" amount of the contract to be exceeded 2) Enter the invoice into the system with as much information as possible 3) Enter "missing/invalid" or default to "zero" value in the field for the piece(s) of information that is missing or invalid 3) Enter the "Payment Status" as "Do Not Pay" 4) Enter into the "Invoice Notes" field the reason the invoice cannot be paid 5) SAVE the invoice into the system 6) Generate a return memo to the Buyer explaining the reason for the return 7) Attach the return memo to the invoice and send it back to the Buyer <p>This use case ends when the incomplete, invalid, or unpayable invoice has been returned to the Buyer.</p>	
Exception(s):	None	
Precondition(s):	An invoice has been received that is either incomplete, invalid, or otherwise unpayable.	
Postconditions(s)	The incomplete, invalid, or otherwise unpayable invoice has been returned to the Buyer.	
Information Requirements:	(See Receive Invoice UC005) Payment Status Invoice Notes	
Assumptions:	<ol style="list-style-type: none"> 1) The Buyer will be able to contact the Vendor and get a corrected invoice generated 2) The Buyer will be able to contact the Program Manager to get proper approval and charge information 3) The Buyer will be able to contact the appropriate parties and get a contract extension for either additional time period(s), and/or an adjustment to the "Hourly Fee", and/or an increase in the "Maximum Fee" amount 	
Business Rules:	<ol style="list-style-type: none"> 1) An invoice is not considered payable if any of the required information is missing and must be returned to the Buyer for resolution 2) It is the Buyer's responsibility to contact the Vendor to get a corrected invoice resent to the Bank that is able to be processed for payment 3) It is the Buyer's responsibility to contact the Project Manager if the invoice does not have the proper approval for payment 4) It is the Buyer's responsibility to contact the appropriate parties and generate a contract extension if the invoice service dates fall outside those of the original contract, OR the "Hourly Fee" does not match with that of the original contract, OR payment of the invoice would cause the "Fee Maximum" amount on the original contract to be exceeded 	

Use Case Name:	UPDATE INVOICE	ID: UC007
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps for updating an invoice, from the time that a revised or new invoice and/or contract extension is received from the Buyer, until the updated or new invoice and/or contract extension has been entered into the system.	
Trigger:	An updated or new invoice and/or a contract extension is received from the Buyer	
Related Use Cases:	Receive Invoice (uses); Update Contract (extended by)	
Normal flow of events:	<p>This use case begins when the Buyer delivers an updated or new invoice and/or a contract extension to the Accountant.</p> <ol style="list-style-type: none"> 1) Manually review the invoice to ensure all the information needed by the accounting department is on the invoice 2) If a contract extension is received, manually review it to ensure that all the information needed by the accounting department is on the contract extension 3) If a contract extension is received, search for the original contract in the system 4) Enter the new contract information in the system (see Information Requirements below and refer to UC004 for normal flow of events) (if applicable) 5) SAVE the updated contract record into the system (if applicable) 6) Search for the returned invoice in the system 7) Change the fields that have missing or zero values by entering the correct information from the updated or new invoice 8) Run a system check to ensure that the dates for programming services fall within the date range specified on the revised contract 6) Run a system check to ensure that the billed rate is the same as the "Hourly Fee" on the revised contract 7) Run a system check to ensure that the dollar amount of the revised or new invoice does not exceeded the "Fee Maximum" amount on the updated contract (must consider all previous invoices that have been paid against the contract fee maximum) 8) When finished entering all required information and validating that the revised or new invoice is able to be paid, change the "Payment Status" from "Do Not Pay" to "Approved for Payment" 9) Enter into the "Invoice Notes" field the date the invoice was returned 10) SAVE the invoice record into the system <p>This use case ends when the revised or new invoice and/or contract extension has been correctly and completely updated in the system.</p>	
Exception(s):	<ol style="list-style-type: none"> 7) If the invoice number has been changed (Vendor issued a newly numbered invoice), the old number must first be noted in the "Invoice Notes" field BEFORE overwriting the "Invoice Number" field (actually, the vendor should issue a credit memo for the original invoice that would be entered into the system to offset the original invoice, necessitating a new record for the new numbered invoice) 8) If the invoice still fails any of the system checks, follow the procedures under use case UC006 (Invoice Exception) to return the invoice back to the Buyer 	
Precondition(s):	<ol style="list-style-type: none"> 1) A revised invoice has been received from the Buyer 2) A contract extension has been received from the Buyer 	
Postconditions(s)	<ol style="list-style-type: none"> 1) The updated or new invoice has been entered into the system with the "Approved to Pay" status 2) The contract extension information has been entered into the system and the contract has been updated 	
Information Requirements:	(UC001) Begin Date End Date Hourly Fee Fee Maximum	

Use Case Name:	PAY INVOICE	ID: UC009
Primary Actor:	Accountant	
Brief Description:	This use describes the steps for sending an invoice that is approved for payment to the Accounts Payable (A/P) department, from the time the Accountant has approved the invoice for payment until the invoice has been sent to the A/P department.	
Trigger:	Invoice(s) are entered into the system with an "Approved to Pay" Payment Status	
Related Use Cases:	None	
Normal flow of events:	<p>This use case begins when the Accountant has set the invoice Payment Status to "Approved to Pay".</p> <ol style="list-style-type: none"> 1) If necessary, log onto the system and locate the invoice to be paid 2) Verify the Payment Status is "Approved to Pay" 3) Enter the current day's date in the "Date Paid" field 4) SAVE the invoice record 5) Generate and print out a Data Entry Sheet for the invoice 6) Attach the Data Entry Sheet to the Invoice and send both to the A/P department <p>This use case ends when the invoice has been updated with the "Date Paid" and sent to the A/P department to have a check issued.</p>	
Exception(s):	3) If the "Date Paid" is AFTER the cut-off for the last A/P check run for the CURRENT month AND before the 6 th day of the FOLLOWING month, the invoice will need to be accrued	
Precondition(s):	An invoice is approved for payment and is ready to be sent to A/P to have a check cut.	
Postconditions(s)	The invoice has been sent to the A/P department with a Data Entry Sheet attached.	
Information Requirements:	Vendor Name Vendor Number Invoice Number Description (the programmer's 1 st initial and full last name AND the dates of service covered by the invoice) Invoice Date Invoice Total G/L Account P.O. Number (the programmer's 1 st initial and full last name) Charge Unit Accountant's Name Date Paid (date invoice is sent to the A/P group)	
Assumptions:	All invoices received for services in the current month can be processed for payment and have a check cut by the A/P department before the end of the current month	
Business Rules:	<ol style="list-style-type: none"> 1) All invoices sent to the A/P department for payment must include a Data Entry Sheet with specific information (see Information Requirements above) 2) If an invoice cannot have a check cut for it BEFORE the end of the current time period (month), an accrual must be made so the expense dollars can be charged to the appropriate general ledger account to ensure the expense is realized in the appropriate period. 	

Use Case Name:	INVOICE STATUS INQUIRY	ID: UC008
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps for responding to a Vendor inquiry for invoice payment status, from the time the inquiry is received by the Accountant until the Accountant has responded to the Vendor with the requested invoice payment status.	
Trigger:	The Accountant receives an inquiry from the Vendor for payment status on an invoice	
Related Use Cases:	None	
Normal flow of events:	<p>This use case begins when the Accountant receives an invoice payment status inquiry from the Vendor.</p> <ol style="list-style-type: none"> 1) Log onto the system and search for the invoice in question (run a report or query) 2) Check the "Payment Status" field to determine the current status 3) If the invoice has a "Do Not Pay" status, check the "Invoice Notes" field to determine the reason the invoice cannot be currently processed for payment 4) If the Vendor is on the telephone, convey the details of the invoice payment status to the Vendor 5) If the invoice is not currently payable, explain the reason(s) for non-payment 6) If the Vendor has communicated via email, send an email through the system to the Vendor providing the invoice payment status or reason(s) for non-payment 7) Enter into the "Invoice Notes" field the date of the Vendor inquiry and the response that was provided in reply 8) SAVE the updated invoice record into the system <p>This use case ends when the Vendor has received the invoice payment status.</p>	
Exception(s):	<ol style="list-style-type: none"> 1) If the invoice has not been received and entered into the system, instruct the Vendor to resend or fax (preferred) a copy of the invoice in question directly to the Accounting department (Accountant), and then proceed with the steps to "Receive Invoice" (see UC005) and process an "Invoice Exception" (see UC006) (6 & 7) If the vendor has sent either a payment request letter or a duplicate copy of the original invoice, check the status of the invoice in question and either call (preferred) the Vendor or send them a letter through the US Postal Service 	
Precondition(s):	The Vendor has sent an invoice for programming services and has not received payment according to the terms of the invoice	
Postconditions(s)	<ol style="list-style-type: none"> 1) The Vendor has been updated on the current payment status of the invoice 2) If applicable, a copy of the invoice (see #1 under Precondition(s)) has been sent with an exception memo to the Buyer for resolution 	
Information Requirements:	Vendor Invoice Number Contract ID Payment Status Invoice Notes	
Assumptions:	<ol style="list-style-type: none"> 1) The Vendor will not send an invoice for programming services before a valid contract has been created for those services 2) The invoice in question will have been received by the Accounting department (Accountant) and entered into the system prior to the actual due date of the invoice 3) The Accountant will be able to satisfy the Vendor's inquiry with the information contained in and available in the system 	
Business Rules:	<ol style="list-style-type: none"> 1) An invoice is not payable unless a valid contract exists for the services billed on the invoice, and the invoice meets the constraints of that contract 2) Vendor inquiries must be resolved within a 24 hour timeframe 3) If a Vendor inquires about an invoice that is not currently in the Accounting system, the Accountant must request a copy of that invoice so it can be entered into the system and then sent to the Buyer for resolution 	

Use Case Name:	ACCRUE INVOICE	ID: UC010
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps for processing an invoice accrual, from the time the invoice is determined to need to be accrued, until the invoice has been designated as accrued in the system	
Trigger:	The invoice has been received and entered into the system after the cut-off for the last Accounts Payable (A/P) checkrun of the current month but before the 6 th day of the following month.	
Related Use Cases:	Pay Invoice (extends)	
Normal flow of events:	<p>This use case begins when an invoice has been entered into the system that either cannot be processed for payment OR is payable and cannot have a check cut in the current month.</p> <ol style="list-style-type: none"> 1) If necessary, log onto the system and locate the invoice that needs to be accrued 2) Verify that the date in the "Date Paid" field is past the cut-off date for the last A/P checkrun for the current month and before the 6th day of the following month OR the "Payment Status" is "Do Not Pay" 3) Enter the current month and year in the "Date Accrued" field 4) SAVE the invoice record in the system 5) Repeat the above 4 steps for ALL invoices that meet the criteria for accrual <p>This use case ends when an invoice has been designated as accrued in the system.</p>	
Exception(s):	None	
Precondition(s):	<ol style="list-style-type: none"> 1) An invoice has been received and entered into the system with either a "Do Not Pay" status OR 2) 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 	
Postconditions(s)	The unpaid (no check cut) invoice has been designated as accrued	
Information Requirements:	Programmer Vendor Charge Unit Invoice Number Invoice Total Date Accrued (month and year)	
Assumptions:	All invoices for services in the current time period will have been received by the 6 th day of the following time period	
Business Rules:	Any invoice that cannot have a check issued for it in the current time period (month) must be accrued so that the expense can be realized in the current period.	

Use Case Name:	RUN ACCOUNTING REPORTS	ID: UC011
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps to generate the Accounting department's month-end reports, from the time they are due until they have been printed out and delivered to the Accounting Manager.	
Trigger:	The deadline for the month-end Accounting department reports.	
Related Use Cases:	None	
Normal flow of events:	<p>This use case begins when the deadline due date for the "General Ledger Expense Report" and "Accrual Report" is reached.</p> <ol style="list-style-type: none"> 1) Log onto the system and navigate to the Reports Menu 2) Select the "G/L Expense" option 3) Enter the date range for the current reporting period 4) Select the PRINT REPORT option 5) Return to the Reports menu 6) Select the "Accruals" option 7) Enter the date range for the current reporting period 8) Select the PRINT REPORT option 9) Return to the Reports Menu OR exit to the Main Menu 10) Deliver both reports to the Accounting Manager <p>This use case ends when both the "General Ledger Expense Report" and "Accrual Report" have been delivered to the Accounting Manager.</p>	
Exception(s):	None	
Precondition(s):	It is time to generate the monthly Accounting department reports	
Postconditions(s):	The monthly Accounting department reports have been delivered to the Accounting Manager	
Information Requirements:	<p>(General Ledger Expense Report)</p> <p>Contract ID Programmer Vendor Charge Unit Invoice Number Date Paid Service Start Date Service End Date Hourly Fee Total Hours Worked Invoice Total Date Accrued Total G/L Expense (calculated)</p> <p>(Accrual Report)</p> <p>Programmer Vendor Charge Unit Invoice Number Invoice Total Date Accrued Total Accrued (calculated)</p>	
Assumptions:	There will actually be at least one invoice to be accrued for the current reporting period	
Business Rules:	The "General Ledger Expense Report" and "Accrual Report" are due to the Accounting Manager for auditing purposes on the 6 th business day of the month.	

Use Case Name:	RUN MANAGEMENT REPORTS	ID: UC012
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps to generate Bank Management's month-end reports, from the time they are due until they have been printed out and sent to the various requesting departments.	
Trigger:	The deadline for the month-end Bank Management reports.	
Related Use Cases:	None	
Normal flow of events:	<p>This use case begins when the deadline due date for the "Contract Programmer's Monthly Expense Recap Report", "Contract Programmer Report - Fee Maximum vs. Actuals", and "Monthly Contract Recap" is reached.</p> <ol style="list-style-type: none"> 1) Log onto the system and navigate to the Reports Menu 2) Select the "Programmer Expense" option 3) Enter the date range for the current reporting period 4) Select the PRINT REPORT option 5) Return to the Reports menu 6) Select the "Fee Maximum" option 7) Enter the date range for the current reporting period 8) Select the PRINT REPORT option 9) Return to the Reports Menu 10) Select the "Contract Recap" option 11) Enter the date range for the current reporting period 12) Select the PRINT REPORT option 13) Return to the Reports Menu OR exit to the Main Menu 14) Send a copy of each report to the appropriate bank requesting unit <p>This use case ends when the "Contract Programmer's Monthly Expense Recap Report", "Contract Programmer Report - Fee Maximum vs. Actuals", and "Monthly Contract Recap" have been sent to the appropriate bank requesting unit(s).</p>	
Exception(s):	None	
Precondition(s):	It is time to generate the monthly Bank Management reports.	
Postconditions(s)	The monthly Bank Management reports have been sent to the various bank units.	
Information Requirements:	<p>(Contract Programmer's Monthly Expense Recap Report)</p> <p>Programmer Vendor Bank Division Charge Unit Invoice Number Service Start Date Service End Date Total Hours Worked Invoice Total Date Accrued Total for Division (calculated) Total for Charge Unit (calculated) Grand Total (calculated)</p> <p>(Contract Programmer Report - fee Maximum vs. Actuals)</p> <p>Division Charge Unit Programmer Service Start Date Service End Date Hourly Rate Project Manager</p>	

Appendix G: Design Standards Document

Interface Design Guidelines

Splash Screen

Font

- Source Sans Pro
- 16pt

Title

- Automated Accounting System

Image

- Bank of Xanadu logo

Login Screen

Font

- Source Sans Pro
- 14pt

Title

- Username
- Password

Text Field

- Username
- Password

Buttons

- Submit

Main Menu

Font

- Source Sans Pro
- 16pt

Title

- Select a project
- Select an existing project
- Add a new project
- Select a report

Sub Menu

Font

- Source Sans Pro
- 16pt

Title

- Receive Contract
- Update Contract
- Receive Invoice
- Update Invoice
- Pay Invoice

Data Entry Screen

Font

- Verdana
- 16pt for Title
- 12pt for labels & buttons
- 11pt for data entry fields

Titles

- Vendor
- Employee
- Invoice
- Division
- Unit
- Contact
- Contract
- Programmers

Buttons

- Bottom of the form
- Size 130x45px

Drop down list

- Select Vendor
-

Label

- Save
- Close
- Ok
- Print
- Submit
- Exit

Coding Standards

Guidelines for Commenting Code

A Descriptive Blocks comment will be included above of every declaration procedure to explain the purpose and function of the code. May include an individual inline comment code at the end of the code for clarity.

At the beginning of each procedure will include the following information:

- Name of procedure (required)
- Name of developer (required)

- Date created (required)
- Date modified (optional)
- Description of procedure (required)
- Description of modification (optional)

Optional information is only required if procedures are modified.

Naming Convention for Code Variables and Objects

Scrappers will follow the guidelines:

- Naming conventions used for variables and objects will follow Microsoft recommended standards
- No spaces are allowed in variables and objects name. If spaces are required, use underscores (e.g. First_Name)
- Camel case is recommended for variable and objects names. (e.g. strFName)

Prefixes for Variables

Prefix	Control Type	Example
bln	Boolean	blnFound
byt	Byte	bytRasterData
cur	Currency	curRevenue
dtm	Date(Time)	dtmStart
dbl	Double	dblTolerance
int	Integer	intQuantity
lng	long	lngDistance
obj	Object	objCurrent
sng	Single	sngAverage
str	String	strFName

Prefixes for Objects

Prefix	Control Type	Example
chk	Check Box	chkReadOnly
cbo	Combo Box, drop-down list box	cboLanguage
cmd	Command Button	cmdExit
dtp	Date picker	dtpStartDate
frm	Form	frmEntry
fra	Frame	fraCategory
img	Image	imgLogo
lbl	Label	lblMessage
lst	List Box	lstState
mnu	Menu	mnuFile
opt	Option Button	optGender
txt	Text Box	txtFirstName

Prefixes for Database Objects

Prefix	Control Type	Example
tbl	Tables	tblCustomer
qry	Queries	qryAchiever
rpt	Reports	rptInvoice
mcr	Macros	mcrUpdateContract
mdl	Modules	mdlBilling

11.7 APPENDIX H: ISSUES LIST

The following is a list of problems and questions we have encountered while creating the new invoicing system. Management must decide how they will proceed with these issues by either using our proposed solution or by finding an alternative

1. Who should be used for emp_contact? For now we will use the next person up the ladder in management
2. Should cont_description be requires or not. The previous ERD had it as optional but the new one says required. For now we will set it as optional
3. Is there a preferred graphic style used by employees?
4. With the possibility of expansion to other branches of the bank, is there any additional info or requirements we should be aware of?

11.8 APPENDIX I: FUTURES LIST

This list contains possible enhancements to the system that are out of scope or would extend over time limitations. We believe that the following enhancements could greatly improve performance and user experience with the new invoicing system.

1. user preferences for color and font settings for better readability
2. add greater levels of security and per user read/write access
3. recovery and backup options
4. support for vendors to log in to the system and input invoices directly and check invoice status



**Edmonds Community College
Computer Information Systems
System Design Specifications**

Prepared: May 14th 2015

Scrappers Systems Analysis and Development Group:

Jennifer Lam

Joshua Chong

Duane Samuels

Cody Smith

Memo

To: Patrick Jay, Vice President & Manager
From: Scrappers Systems Analysis Group
Date: April 23, 2015
Subject: Implementation Plan

The Scrappers Systems analysis team has finished and attached the Implementation Plan for the Bank of Xanadu's new contract and invoice management system for dealing with the payment of programmers.

The implementation plan contains information on these sections-

- Implementation Approach
- Critical Success Factors
- Migration Plan
- User Training
- Readiness
- Ongoing Support and Maintenance

We would like to arrange a date to review the document at the Bank of Xanadu branch in Bellevue, Washington on May 14, 2015 at 6:00pm. Please feel free to contact the Scrapper via email if you have any questions or concerns.

System Design Specifications
New Automated System for Managing Contract Payables

Bank of Xanadu
Accounting Department

Scrappers Systems Analysis Group:

Jennifer Lam

Joshua Chong

Duane Samuels

Cody Smith

Prepared: May 14th 2015

1 TABLE OF CONTENTS

2	Introduction.....	5
3	Implementation Approach	6
4	Critical Success Factors.....	8
5	Migration Plan	9
6	User Training.....	10
7	Readiness.....	11
8	Ongoing Support and Maintenance.....	13

2 INTRODUCTION

Following the design phase of Team Scrappers new contract invoicing system, we will move forward with the project by creating an implementation plan. The implementation plan will contain information on what it will take to complete the project goals and ensure that all users of the old system are prepared to make the transition to the new one. The main sections of the implementation plan deliverable are:

Implementation Approach - this section explains to possible methods on how we can begin to implement the system in a live environment, along with the risks and benefits of each approach

Critical Success Factors – this section explains the requirements that need to be in place prior to implementation the system

Migration Plan – this section explains on how we plan to populate the new system with data

User Training – this section explains the method on how we will train employees to be ready to use the system

Readiness – this section explains the milestones that need to be met in user training before they are ready to use the new system

Ongoing Support and Maintenance – this section will explain the methods in which the system will maintained after it is implemented

3 IMPLEMENTATION APPROACH

The Implementation approach is the method in which users will transition from the older system to the newer. Team Scrappers has found four possible methods that could be used to implement the new contract invoicing system. We believe that a combination of the pilot and parallel approach will be the best solution for this instance, but all four methods will be explained with the possible benefits and risks of each. These four methods are:

Direct Cutover - this method requires that we completely abandon the old system and immediately replace it with the newer one

Benefits:

- Less employee overhead than other methods.
- Also the least expensive

Risks:

- no backup in case system is inadequate to needs
- highest risk

Parallel Operation – this method would require that we run both systems simultaneously until a time where we see that the new system is sufficient to be used without the support of the previous

Benefits:

- minimum risk
- if management decides the new system is lackluster; it can be removed without effecting business operations

Risks:

- Most expensive and most employee overhead due to both systems being run at the same time.

Phased Operation – this method would require that we remove the old system and implement the new system in small sections at a time inside the bank. Any of the other three methods can be used for each phase of implementation.

Benefits:

- less expensive than the parallel method
- Only one section is changed at a time so if a problem comes up it is isolated and only in a small section of the database

Risks:

- more expensive than a direct cutover due t both systems being run
- Needs to be broken up in sections which can take longer than other methods

Pilot Operation – a pilot implementation is a method that implements the system into a single site of the bank to be run in parallel with the old system or a direct cutover. When it is found that performance is satisfactory at the site, it will be installed companywide

Benefits:

- a faster implementation time compared to a parallel

Risks:

- more expensive than the direct cutover

Recommendation

We believe that a combination of a pilot and parallel methods would be best fitting for this project. We have decided this from the fact that this is originally designed to be a pilot project that could be scaled up to be used in all of the bank of Xanadu's branches worldwide. We suggest that both systems be run in parallel at the pilot site in accounting of the Bellevue branch. By doing this we will lessen risks of the system being inadequate or users taking to the new system slower than desired. We would then remove the old system when business operations are up to speed and all previous contracts associated with the old system have expired.

4 CRITICAL SUCCESS FACTORS

There are three main areas that must be considered before the new system can be implemented into a live environment.

User Training/Readiness

Users must understand the potential to increase productivity and be willing to make the changes to the newer system

Users must be trained to such a point where they are capable of running the system without assistance and are able to train newer employees to use it.

System Support

The database has been installed on bank computer systems

New employees have been hired (as necessary) and trained to ease the workload on accounting

Network must be tested and secured to ensure confidentiality and availability

Business Impact-

Employees must perform their duties faster than with the previous system

The new system must require less interaction from accounting and automate previously manual processes when entering invoice and contract information

The system must be robust enough to handle hundreds of transactions per day

The system must lessen vendor inquiries due to lengthy invoice input time and inaccuracies while typing

5 MIGRATION PLAN

After reviewing the data tables within the current Contract Management System, we have decided that a migration of the current database to the new database will be processed manually. Essentially all data will be added into the new database via manual typing.

This may sound unreasonable when the amount of data is enormous. However, our knowledge of the current database system is the data is in Microsoft Excel format while the new database will be built using Microsoft Access. The conversion process does not occur naturally and using a third-party program to aid the transfer, it does not guarantee complete transfer of the data, excluding the possibility of having errors that can cause damage to the data itself.

For this reason, we are proposing the manual input for the new Contract Management System. Of course, this would mean possible hiring or paying overtime to increase the process speed for the large amount of data. This may possibly appear as loss of profit, however, on the other hand by creating a new database will find and fix errors from the current system, avoid confusion by trying to transfer Excel data to Access and other benefits that will exceed the cost of manual data input.

6 USER TRAINING

Scrappers will conduct an initial training at the Bellevue branch to the following employee for the pilot site in Access:

➤ Contract Group

Manager: Scott Sorenson

Rob Watt

Sam Esposito

Mark Martin

David Hart

Jagreet Kaur

Anthony Lewis

➤ Accounting Group

Manager: Patrick Jay

Dave Spencer

Tamisha Spencer

Misty Barber

➤ IT Group

Scrappers will conduct a train-the-trainer session prior to rollout. We have divided into two groups that will be receiving training on June 4, 2015 and June 11, 2015. The employees with the **BOLD** have been assigned to attend the first primary users training and the ***BOLD Italic*** will attend the secondary users training from each department.

Our goal is to train-the-trainer as much knowledge in using Access that they will be able to train in their small group. For training purpose we have created a training materials provided by the Scrappers. It is include a PowerPoint presentation with a PDF file and tutorial videos created by Scrappers to help ease the process of learning Access. We will also provide a completed end-user manual after the training session that they use as a reference. Scrappers will be available throughout the week to provide additional support before the launch.

7 READINESS

Scrapers will determine when the users are ready to “go live,” the following are the measurement of the ability to use the system.

All Bank of Xanadu users

- Ability to log into and out of the system
- Ability to navigate through the system
- Ability to enter project status information
- Ability to edit previously entered information
- Ability to preview or Print certain reports

All Contract Group users

- Ability to verify corrected contract record
- Ability to create a new contract record
- Ability to assign Contract Number
- Ability to select correct vendor
- Ability to enter contract information into the system
- Ability to save contract record

All Accounting Group users

- Ability to inquiry vendor information
- Ability to create a new invoice record
- Ability to enter invoice information into the system
- Ability to save invoice record
- Ability to create exception memo
- Ability to update invoice record into the system
- Ability to create a data entry sheet

- Ability to send pay invoice to AP
- Ability to accrual invoice

All Payables Group users

- Ability to verify timesheet is attached with the invoice
- Ability to verify pay invoice is correct
- Ability to access the report options on the system

8 ONGOING SUPPORT AND MAINTENANCE

After developing the new Contract Management System, there will be a need for support and maintenance to maintain or even increase the performance of the system. We, Team Scrappers, will like to continue our support by handling those operations.

We propose our payment to maintain the new system to be fifty dollars per hour. This amount may seem unreasonable at first glance but we are willing to negotiate and can explain the reason for this price.

We have all the necessary experience and knowledge for the new system since we are the ones who built the new system. Because of this crucial factor, there is no need to train additional employee or department just to support the system. It is possible that the initial payment for that option will seem profitable for the company. However, after hiring new employees they will need a long time of training before they can understand the system properly. Additionally there is no guarantee they will understand the system complete and possibly run into errors they cannot fix. Unlike hiring new employees, we Team Scrappers, do not need to be trained nor explore the system to maintain it. We will start our work as soon as the system is implemented and the cost will benefit the company on the long run than trying to manage the system from an outside source without any knowledge of the system.

9 Evaluation Criteria – Implementation Plan Draft

10Criteria	11Points	Given	Comments
<ul style="list-style-type: none"> ❑ Completed and submitted by the due date 	5	5	
<ul style="list-style-type: none"> ❑ User Manual contains content specified in assignment <ul style="list-style-type: none"> ○ Assignment Title Page ○ Cover Transmittal Memo ○ Report Title Page ○ Table of Contents ❑ Body of document discusses the following: <ul style="list-style-type: none"> ○ Introduction ○ Implementation Approach ○ Critical Success Factors ○ Migration Plan ○ User Training ○ Readiness ○ Ongoing Support and Maintenance 	25	24	See various comments.
<ul style="list-style-type: none"> ❑ Implementation plan demonstrates an understanding of the role and importance of implementation, as well as the various implementation methods and how they are used 	10	10	
<ul style="list-style-type: none"> ❑ Written in appropriate style for the audience ❑ Document organization and content demonstrate careful planning and thought and is easy to follow ❑ Well-written, free of spelling and grammar errors 	10	10	
Total	50	49	



**Edmonds Community College
Computer Information Systems
Test Plan**

Prepared: May 28th 2015

Scrappers Systems Analysis and Development Group:

Jennifer Lam

Joshua Chong

Duane Samuels

Cody Smith

Memo

To: Patrick Jay, Vice President & Manager
From: Scrappers Systems Analysis Group
Date: May 28, 2015
Subject: Test Plan

Scrappers Systems Analysis Group has created and attached the Test Plan for the bank of Xanadu new automated system for managing contract Payables. This Test Plan document contains the following sections

- Testing Approach
- Test Environment
- Test Scenarios
- Sample Test Data

We would also like to request a meeting for May 28th in the Bellevue branch of the bank of Xanadu. During this meeting, we will do a walkthrough of our Test Plan and discuss how we will move forward with the project.

Test Plan
New Automated System for Managing Contract Payables

Bank of Xanadu
Accounting Department

Scrappers Systems Analysis Group:

Jennifer Lam

Joshua Chong

Duane Samuels

Cody Smith

Prepared: May 28th 2015

1 TABLE OF CONTENTS

- 2 Introduction 5
- 3 Testing Approach 6
- 4 Test Environment..... 8
- 5 Test Scenarios10
- 6 Sample Test Data12
- 7 Appendix13
 - 7.1 Appendix-A: Test Forms.....13
 - 7.2 Appendix-B: Sample Test Data45

2 INTRODUCTION

This document will describe how Team Scrappers Systems Analysis Group will setup and carry out the software testing of the contract invoicing system. This stage is important to the success of the implementation of the system because it will ensure that when the system is in a live environment it will perform as required it will perform as required in the system specification.

This Test Plan document will explain the following:

- Testing Approach – explains the methods, benefits and risks of each type of test we will be using, and why the test is important to the functionality of the project
- Test Environment – this section contains the requirements that must be in place for the system testing to be successful, and explaining all the details of where and when the testing will occur and how it will happen, and who will conduct the testing
- Test Scenarios – this section contains detailed step-by-step scenarios of actions that the system will need to carry out on a regular basis
- Sample Test Data – this section will explain to users clearly, the valid and invalid values for each attribute of an entity In the database

3 TESTING APPROACH

The following is the steps we will follow

Blackbox Testing

This method requires us to make sure all the functions within the database is working correctly without viewing the code itself. Therefore, we must make sure all fields are accepting the right input and display the correct output. To achieve a reasonable result we must conduct multiple test to make sure the system is functioning correctly.

Whitebox Testing

Since this method requires us to test the internal structure or code of the system itself, we must make sure the other developers than the one who worked on the CMS will be conducting the test. We must include tests of making sure all path and naming convention of objects are correctly labeled so that the system will display the correct input and output when being operated.

Regression Testing

This phase of testing will primarily focus on finding bugs and fixing them. It is possible that during the blackbox and whitebox testing bugs will be discovered during these tests. To make sure bugs are fixed it might be recommended to use both testing method at the same time to make sure the bugs are found and fixed as soon as they are found. Since bugs can occur from simple typing error or even change of path connection, review the codes to check for corrections.

User acceptance testing

This testing explains itself. This will allow the user to be familiar to the new CMS and make sure that there is the least amount of bugs or errors possible. This phase will have a cooperated testing with the user to walk them through the system. They will review layout, system process and other system components to give us feedback. We will reflected on this feedback and fix components that can be fixed or explain why certain components must remain as is.

Code Walkthrough

This is simple having a group walkthrough to make sure all the team members are acquainted with the system. The developer and testers of the system will make sure everyone understands the program. Future team members will also be given walkthrough. Therefore developing a possible guideline or even manual will increase efficiency in the future.

4 TEST ENVIRONMENT

These are our general plans to conduct testing when system tests are needed.

Location

The two locations that will be used to conduct testing will be at Team Scrappers office or at the Bellevue branch of Bank of Xanadu. The reason for selecting our office as one of the testing location is because Team Scrappers Office has the system and experts that understand the system. Because of full access to the current version of the system and the developers, testing will run smoothly when conducted at Team Scrappers office. This will also increase efficiency since there won't be a large group occupying a large space at the Bellevue branch. However, testing may be conducted at the Bellevue branch if requested.

Time

Testing will occur at least once before June 19. Additional testing can be requested in the future if needed. Most test will run during after hours or weekends when the Bank of Xanadu is not busy or has the system running. However, emergency tests can occur during business hours when requested

Process

Our testing will primarily focus on the fact that the system is operating on live-server. Since live-servers are being tested, there are possibilities that the server may have errors during the tests.

People

When the test is being run at our office the developers, testers and additional employees will conduct white-box testing and other necessary testing to make sure the live-server functions correctly. Other components of the system will be tested to make sure the system runs smoothly. When the test is conducted at the Bellevue branch, we will conduct black box as well as user acceptance testing with the accountants, Xanadu IT department and our developers.

Readiness

To conduct our testing we must make sure several requirements are met. First the system must be correctly installed into the Xanadu database. This is simple yet crucial part of testing since without the database running the system, test cannot be conducted under any circumstances. Additionally all

Team Scrappers employees, Bank of Xanadu accountants and Xanadu IT department must be trained about the system to understand how the system works. This will ensure that all personnel involved in system testing will understand what is being tested and what is being altered if needed. Also numerous scenarios must be addressed and tested to make sure the system can handle different situations. This will increase system reliability and show areas to conduct possible bug fix.

5 TEST SCENARIOS

The following is a list of testing scenarios that will be used to ensure that users will be able to navigate through the system successfully and test that it can handle system processes like adding new contracts/invoices, handling their exceptions, adding new information, and running reports.

<i>ID</i>	<i>Scenarios</i>	<i>Condition/ Description</i>	<i>Level</i>	<i>Unit/Objective</i>
TS001	Accountant creates a new contract record	Navigate to the contracts menu and create a new contract	System	Verify the navigation while creating a new contract record
TS002	Accountant adds new bank information to the system	Enter new bank information	Unit	Add new bank information to the database
TS003	Accountant enters an invalid contract into the system that must be sent back to the buyer	Navigate to the contracts screen to add a new contract and create an exception memo	System	Verify navigation while creating a new contract that cannot be accepted
TS004	Accountant updates a contract previously recorded in the system	Modify the information in an existing contract	Unit	Update the information on an existing contract
TS005	Accountant creates a new invoice record	Navigate to the invoice menu and create a new invoice record	System	Verify navigation while creating a new invoice record
TS006	Accountant enters a invalid invoice into the system that must be sent back to the buyer	Navigate to the invoice menu and add a new invoice and create an exception menu	System	Verify navigation while creating a new invoice that cannot be accepted

TS007	Accountant updates an invoice previously recorded in the system	Modify the information on an existing invoice	Unit	Update the information on an existing invoice
TS008	Accountant needs to respond to a vendor, and discover the reason their invoice has not been paid	Navigate to the invoice screen and discover the status of an invoice	Unit	Verify the status of an invoice
TS009	Accountant needs to make an invoice record payable	Modify the status of the date paid field and the payment status field	Unit	Update the payment status and the date paid
TS010	Accountant needs to accrue an invoice that has not been made payable in the correct time period	Modify the payment status and the date accrued field in an invoice	Unit	Update the payment status and date accrued
TS011	Accountant needs to run end of the month reporting for the accounting department	Navigate to the reporting screen and run the accounting reports	System	Verify that the report runs and is printed
TS012	Accountant needs to run end of the month reporting for the bank management	Navigate to the reporting screen and run the management reports	System	Verify that the report runs and is printed

6 SAMPLE TEST DATA

Scrappers have created a comprehensive list of valid and invalid data that will assess all possible entry. One important reason of creating a sample test data is to display the samples input of each attribute. The role of test data helps the programmer code each attribute validation and will help alert the user when inputting an invalid data. See Appendix B for a full list of sample test data

7 APPENDIX

7.1 APPENDIX-A: TEST FORMS

Scenario ID: TS001		Test Type: System	Tester: Pete	Date: 5/14/15	
Scenario Name:		RECEIVE CONTRACT			
Background and Test Objective(s):		A new contract has been delivered to the accounting department and needs to be entered into the system. The objective of this test is to ensure that the data entry person is able to access the contract entry screen and successfully enter all required data elements into the system, creating a new contract record.			
Related Use Cases or Test Scenarios:		UC001 – RECEIVE CONTRACT			
Testing Notes:		Assume that the user is already logged onto the system – is an active user. Assume the user is at the main “menu”.			
Cas e	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	User needs to navigate to the contract entry screen.	Select the correct option to go to the contract entry screen.	The contract entry screen should appear.	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
02	The vendor needs to be selected	Click in the vendor field and select the correct vendor.	A drop down list of available vendors should appear and allow the user to select the appropriate one.	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

03	The programmer needs to be selected	Click in the programmer field and select the correct programmer.	A drop down list of available programmers should appear and allow the user to select the appropriate one. (Available list should only be for programmers associated to the selected vendor)	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
04	The contract number needs to be entered	Click in the contract number field and enter the correct contract number (EX: ROTZ0415)	The contract number field should accept the correct contract number (text format field)	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
05	The contract start date needs to be entered	Click in the start date field and enter the correct contract start date.	The start date field should accept the correct start date (date format field)	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
06	The contract end date needs to be entered	Click in the end date field and enter the correct contract end date.	The end date field should accept the correct end date (date format field)	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
07	The charge unit needs to be entered	Click in the charge unit field and select the correct charge unit.	A drop down list of available bank unit numbers should appear and allow the	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

			user to select the appropriate one.		
08	The hourly fee needs to be entered	Click in the hourly fee field and enter the correct hourly fee.	The hourly fee field should accept the correct hourly fee (currency format field)	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
09	The fee maximum needs to be entered	Click in the fee maximum field and enter the correct fee maximum.	The fee maximum field should accept the correct fee maximum (currency format field)	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
10	The project manager (contact) needs to be entered	Click in the contact field and select the correct contact.	A drop down list of available contacts should appear and allow the user to select the appropriate one.	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
11	The project description needs to be entered	Click in the project description field and enter the correct project description.	The contract description field should accept the correct contact description (notes format field)	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
12	The new contract record needs to be saved and recorded in the	Click on the "SAVE" option	The new contract record should be saved into the Contract table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

	database				
13	Need to verify that the new contract record was successfully created and saved in the Contract table	Navigate to the Contract table, open it, and search for the new contract record	The new contract record should be saved successfully in the Contract table	<input type="checkbox"/> Pass <input type="checkbox"/> 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: TS002		Test Type: Unit	Tester: Josh	Date: 5/18/15	
Scenario Name:		ADD NEW BANK INFORMATION			
Background and Test Objective(s):		A new contract has been received by the accountant and contains information on a new vendor, bank contact, bank unit, or bank division that needs to be entered into the system. The objective of this test is to ensure that the user can navigate to the new bank information screen and new records can be used for future contract/invoice input.			
Related Use Cases or Test Scenarios:		UC002 – ADD NEW BANK INFORMATION			
Testing Notes:		Assume that the user is already logged onto the system – is an active user. Assume the user is at the main “menu”.			
Cas e	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Result s	/Notes
01	User needs to navigate the system to the add new bank information screen	Select the correct option to enter the main screen	The main screen should appear	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
02	The new vendor information needs to be added	Click the button to enter vendor screen and “add new” to enter in the new vendor name	The vendor field should accept the new vendor name	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
	The new vendor information needs to be saved and recorded in the database	Click on the “save” button	The new vendor information should be saved in the vendor table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
03	The new programmer’s information needs to be added	Click the button to enter programmer screen and “add new” to enter in the new	The programmer field should accept the new programmer name	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

		programmer name			
	The new programmer information needs to be saved and recorded in the database	Click on the "save" button	The new programmer information should be saved in the programmer table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
04	The new bank contact information needs to be added	Click the button to enter contacts screen and "add new" to enter in the new vendor name	The bank contact person field should accept the new contact person's name	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
	The new contact information needs to be saved and recorded in the database	Click on the "save" button	The new information should be saved in the contact information table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
05	The new bank charge unit information needs to be added	Click the button to enter charge unit screen and "add new" to enter in the new charge unit	The bank charge unit field should accept the new bank unit	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
	The new charge unit information needs to be saved and recorded in the database	Click on the "save" button	The new charge unit information should be saved in the charge unit table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
06	The new bank division information needs to be added	Click the button to enter division screen and "add new" to enter in the new vendor name	The bank division field should accept the new bank division name	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
	The new bank division	Click on the "save" button	The new bank division	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

	information needs to be saved and recorded in the database		information should be saved in the divisions table		
07	The new project manager information needs to be added	Click the button to enter employees screen and "add new" to enter in the new project manager name	The project manager field should accept the new project manager	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
08	The new project manager information needs to be saved and recorded in the database	Click on the "save" button	The new project manager information should be saved in the employee table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
09	The user needs to verify that the new bank information was successfully created and saved in each table	Navigate to each data table to search and verify that the data has been saved	The new bank information should be saved in their tables	<input type="checkbox"/> Pass <input type="checkbox"/> 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: TS003		Test Type: System	Tester: Josh	Date: 5/18/15	
Scenario Name:		CONTRACT EXCEPTION			
Background and Test Objective(s):		A new contract that has been received contains missing or invalid information and is about to be entered into the system. The objective of this test is to ensure that when a user is entering a new contract, they can change the status of a contract to invalid and generate an exception memo			
Related Use Cases or Test Scenarios:		UC003 – CONTRACT EXCEPTION			
Testing Notes:		Assume that the user is already logged onto the system – is an active user. Assume the user is at the “new contract” menu Assume all required bank information has been saved – such as vendor name, programmers, charge unit, etc.			
Cas e	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Result s	/Notes
01	The user needs to enter the contract entry screen	Select the correct option to enter the contracts table	The contract screen should appear	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
02	The user needs to add all present contract information	Select the correct fields and enter the information on the invalid contract	The contract fields should accept the information given	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
03	The user need to set the status of a contract	Select the contract status field and set it to “invalid”	The status of the contract should accept the “invalid” option	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
04	The user needs to save the invalid contract	click the save button	The invalid contract information should be saved	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
05	The user	Enter the	The new invalid	<input type="checkbox"/> Pass	

	needs to verify that the invalid contract information has been saved	contracts table and locate the invalid contract	contract should be in the contracts table	<input type="checkbox"/> Fail	
06	The user needs to generate the exception memo to be sent to the buyer	Click the generate exception memo button in the contracts screen	A template for an exception memo should appear with a copy of the invalid contract	<input type="checkbox"/> Pass <input type="checkbox"/> 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: TS004		Test Type: Unit	Tester: Josh	Date: 5/18/15	
Scenario Name:		UPDATE CONTRACT			
Background and Test Objective(s):		A contract has been received that updates the information on a contract that is already recorded in the system. The objective of this test is to ensure that the user can navigate to the contract screen and edit the fields in an existing contract, overwriting the previous.			
Related Use Cases or Test Scenarios:		UC004 – UPDATE CONTRACT			
Testing Notes:		Assume that the user is already logged onto the system – is an active user. Assume the user is at the “Contracts” menu			
Cas e	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user must locate the contract in the contracts menu	Locate the contract to be edited by the contract ID	The contract is found	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
02	The user needs to enter the information of every field that was changed	Compare the information on the old contract and the new one to find the information that has been changed and update required fields	The user finds changed data and records it	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
03	The user needs to record the date returned and any additional information in the contract notes field	Click in the contract notes field and record the changes and date the section	The notes field is updated	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
04	The user needs to save	Click “save” button	The updated contract should	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

	the updated contract		be in the contracts table		
Testing Notes:					
Tests Completed by: _____ Date Conducted: _____					
Reviewed by: _____ Date Reviewed: _____					
<p>Corrective Actions:</p> <p><input type="checkbox"/> Passed: No further action required (Date: _____, Initials: _____)</p> <p><input type="checkbox"/> Errors Submitted for Correction (Date: _____, Initials: _____)</p> <p><input type="checkbox"/> Errors Corrected (Date: _____, Initials: _____)</p> <p><input type="checkbox"/> Retested (Date: _____, Initials: _____)</p> <p><input type="checkbox"/> Other:</p>					

Scenario ID: TS005	Test Type: System	Tester: Josh	Date: 5/18/15		
Scenario Name:	RECEIVE INVOICE				
Background and Test Objective(s):	A new invoice has been delivered to the accounting department and needs to be entered into the system. The objective of this test is to ensure that the data entry person is able to access the invoice entry screen and successfully enter all required data elements into the system, creating a new invoice record.				
Related Use Cases or Test Scenarios:	UC005 – INVOICE				
Testing Notes:	Assume that the user is already logged onto the system – is an active user. Assume the user is at the main “menu”				
Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	User needs to navigate to the invoice entry screen.	Select the option to go to the invoice entry screen	The invoice entry screen should appear	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
02	The vendor needs to be selected	Click the vendor field and select the correct vendor	A drop down list of available vendors should appear and allow the user to select the appropriate one.	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
03	The remit-to address needs to be entered	Click the vendor field and select the correct vendor	The remit-to address field should accept the correct	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

			address		
04	The contract ID needs to be selected	Click the contract ID filed and select the correct ID	A drop down list of available contract ID's should appear and allow the user to select the appropriate one.	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
05	The Invoice Number needs to be entered	Click in the invoice number field and enter the correct contract number	The invoice number field should accept the correct invoice number	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
06	The invoice date needs to be entered	Click in the date field and enter the correct invoice date.	The date field should accept the correct invoice date	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
07	The programme rs need to be entered	Click in the programmer field and select the correct programmer.	A drop down list of available programmers should appear and allow the user to select the appropriate one. (Available list should only be for programmers associated to the selected vendor)	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

08	The service start date needs to be entered	Click in the service start date field and enter the correct invoice service start date.	The start date field should accept the correct start date	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
09	The service end date needs to be entered	Click in the service end date field and enter the correct end date.	The end date field should accept the correct end date	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
10	The hourly rate needs to be entered	Click in the hourly rate field and enter the correct hourly rate	The hourly rate field should accept the correct hourly rate	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
11	The total hours worked needs to be entered	Click in the total hours worked field and enter the correct amount of hours worked	The total hours worked field should accept the correct amount of hours worked	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
12	The invoice total needs to be entered	Click in the invoice total field and enter the correct amount on the invoice	The invoice total field should accept the correct invoice total	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
13	The invoice terms needs to be entered	Click the invoice terms field and enter the correct invoice terms	The invoice terms field should accept the terms stated on the invoice	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
14	The approved	Click the approved by	The approved by field should	<input type="checkbox"/> Pass	

	by person needs to be entered	person and enter the name of the person who approved the invoice	accept the name of the person who approved the invoice	<input type="checkbox"/> Fail	
15	The charge unit needs to be entered	Click in the charge unit field and select the correct charge unit.	A drop down list of available bank unit numbers should appear and allow the user to select the appropriate one.	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
16	The payment status needs to be entered	Click in the payment status field and select the correct status	A drop down list of possible invoice statuses should appear and allow the user to select the appropriate one.	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
17	The new invoice record needs to be saved and recorded in the database	Click the "save" option	The new invoice record should be saved in the invoice table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
18	The user needs to verify that the invoice record was successfully created and	Navigate to the invoice table, open it, and search for the new invoice record	The new invoice record should be saved successfully in the invoice table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

	saved in the invoice tabled				
--	-----------------------------	--	--	--	--

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: TS006	Test Type: System	Tester: Josh	Date: 5/18/15
Scenario Name:	INVOICE EXCEPTION		
Background and Test Objective(s):	A new invoice that has been received contains missing or invalid information and is about to be entered into the system. The objective of this test is to ensure that when a user is entering a new invoice with invalid information, that the user can create an exception memo and set the payment status of the invoice to "do not pay"		
Related Use Cases or Test Scenarios:	UC006 – INVOICE EXCEPTION		
Testing Notes:	Assume that the user is already logged onto the system – is an active user.		

Cas e	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	User needs to navigate to the invoice entry screen.	Click the "new invoice" screen	The new invoice screen should appear	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
02	The user needs to enter all present information on the invoice	Select the correct fields and enter the information on the invalid invoice	The invoice fields should accept the information given	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
03	The user need to set the status of a contract	Select the contract status field and set it to "Do Not Pay"	The status of the invoice should accept the "Do Not Pay" option	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
04	The user needs to save the invalid invoice	click the "save" button	The information should be saved in the invoices table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
05	The user needs to verify that the invalid invoice information	Enter the invoices table and locate the invalid invoice	The new invalid invoice should be in the invoices table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

	has been saved				
06	The user needs to generate the exception memo to be sent to the buyer	Click the generate exception memo button	A template for an exception memo should appear with a copy of the invalid invoice	<input type="checkbox"/> Pass <input type="checkbox"/> 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: TS007	Test Type: Unit	Tester: Josh	Date: 5/18/15		
Scenario Name:	UPDATE INVOICE				
Background and Test Objective(s):	An invoice has been received that will update the information on an invoice that was previously recorded in the system. The objective of this test is to ensure that the user can navigate to the invoice screen and edit the status and notes field of an existing unpayable invoice to reference a new valid invoice ID number				
Related Use Cases or Test Scenarios:	UC007 – UPDATE INVOICE				
Testing Notes:	Assume that the user is already logged onto the system – is an active user. Assume the user is at the “Invoices” menu				
Cas e	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user needs to enter the new information in a new invoice	Click “add new” to create a new invoice record with the updated information	The fields should accept the data	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
02	The user needs to save the new invoice	Click the “save” button	The updated invoice should be in the invoice table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
03	The user needs to edit the old invoice	Locate the old invoice	The invoice should be found in the table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
04	The old invoice’s status needs to be changed	Click the status field and set it to “invalid w/ credit memo reference	The invoice status should accept the new info	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
05	The old invoice’s notes field needs to contain new invoice information	Click the notes field and record that the invoice is invalid and enter the new invoice number for reference	The notes field should accept the new information	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

06	The user needs to save the updated invoice	Click the "save" button	The updated invoice should be in the invoice table	<input type="checkbox"/> Pass <input type="checkbox"/> 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: TS008	Test Type: Unit	Tester: Josh	Date: 5/18/15
Scenario Name:	INVOICE STATUS INQUIRY		
Background and Test Objective(s):	A vendor has requested that they be informed on the payment status of an invoice that was received by accounting. The objective of this test is to ensure that a user can navigate to the invoice screen and locate the invoice and find the required information needed to report back to the vendor		
Related Use Cases or Test Scenarios:	UC008 – INVOICE STATUS QUERY		
Testing Notes:	Assume that the user is already logged onto the system – is an active user. Assume the user is at the main “menu”		

Cas e	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user must navigate to the invoices screen	Select the option to go to the invoice entry screen	The invoice entry screen should appear	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
02	The user must locate the invoice by the invoice ID	Scroll through the list until you find the matching Invoice ID	The user should find the invoice	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
03	The user must check the invoice status field and discover the reason the invoice has not been paid	Locate the Status field and Notes field	The user should find a “Do Not Pay” status and the in Notes field, it should have the reason for delay of payment	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
04	The user must record the inquiry incident	Input a log of the inquiry incident in the Notes field	The Notes field should accept the inquiry log	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
05	The user must save the	Click “save”	The invoice record should be	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

	updated record		saved successfully		
--	----------------	--	--------------------	--	--

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: TS009		Test Type: Unit	Tester: Josh	Date: 5/18/15	
Scenario Name:		PAY INVOICE			
Background and Test Objective(s):		A valid invoice is approved for payment and needs to be sent to accounts payable. The objective of this test is to ensure that users can navigate in the invoice screen to enter the information needed by accounts payable and create a data entry sheet.			
Related Use Cases or Test Scenarios:		UC009 – PAY INVOICE			
Testing Notes:		Assume that the user is already logged onto the system – is an active user. Assume the user is at the “Invoices” menu			
Cas e	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user needs to locate the invoice to be paid	Navigate to the invoice by the Invoice ID	The Invoice should be found	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
02	The user must verify the invoice status	Navigate to the Payment Status	It should say “Approved to Pay”	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
03	The Date paid needs to be entered	Click the Date Paid field an enter the current date	The Date Paid field should accept the date	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
04	The user must save the record	Click the “save” option	The invoice record should be saved in the invoice table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
05	The user must verify that the invoice changes have been saved	Navigate to the invoice table, open it, and search for the new invoice record	The new invoice record should be saved successfully in the invoice table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
06	The user must create a data entry sheet	Click the create data entry sheet	The data entry sheet template should appear		

		button			
--	--	--------	--	--	--

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: TS010		Test Type: Unit	Tester: Josh	Date: 5/18/15	
Scenario Name:		ACCRUE INVOICE			
Background and Test Objective(s):		An invoice is entered into the system and cannot be processed or is unable to be paid during that month. This test is to ensure that the user can navigate to the invoice that needs to be accrued and enter the required information			
Related Use Cases or Test Scenarios:		UC010 – ACCRUE INVOICE			
Testing Notes:		Assume that the user is already logged onto the system – is an active user. Assume the user is at the “Invoices” menu			
Cas e	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user must navigate to the invoice that must be accrued	Scroll to the invoice that must be accrued	The invoice is found	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
02	The user needs to enter the payment status	Locate the status field and change it to “accrued”	The field should accept the status	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
	The user must enter the Date Accrued	Click the “Date Accrued” field and enter the date	The field should accept the date given	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
03	The user must save the invoice record	Click “Save”	The invoice should contain the changes and be in the invoice table	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
04	The user must verify that the invoice record has been updated	Navigate to the invoice table, open it, and search for the invoice record	The new invoice record should be saved successfully in the invoice table	<input type="checkbox"/> Pass <input type="checkbox"/> 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: TS011	Test Type: System	Tester: Josh	Date: 5/18/15
Scenario Name:	RUN ACCOUNTING REPORTS		
Background and Test Objective(s):	It is the end of the month and accounting needs the "Accrual Report" and "General Ledger Expense Report". The objective of this test is to ensure that the user can navigate to the "Reports" menu and create the reports in the required date range.		
Related Use Cases or Test Scenarios:	UC011 – RUN ACCOUNTING REPORTS		
Testing Notes:	Assume that the user is already logged onto the system – is an active user.		

Cas e	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user must navigate to the reporting screen	Click the correct option to go to the reporting screen	The reporting screen should appear	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
02	The "Accrual Report" needs to be selected	Click the correct option to start the "Accrual Report"	The "Accrual Report" screen should appear	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
03	The date range needs to be entered	Select the correct date range for the current reporting period	The Date fields should accept the correct date range	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
04	The report must be printed	Click the option to print the report	The report should be sent to the printer	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
05	The "General Ledger Expense Report" must be selected	Click the correct option to start the "General Ledger Expense Report"	The "General Ledger Expense Report" screen should appear	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
06	The date range needs to be selected	Select the correct date range for the current reporting period	The Date fields should accept the correct date range	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
07	The report must be	Click the option to print the report	The report should be sent	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

	printed		to the printer		
Testing Notes:					
Tests Completed by: _____ Date _____					
Conducted: _____					
Reviewed by: _____ Date _____					
Reviewed: _____					
Corrective Actions:					
<input type="checkbox"/> Passed: No further action required (Date: _____, Initials: _____)					
<input type="checkbox"/> Errors Submitted for Correction (Date: _____, Initials: _____)					
<input type="checkbox"/> Errors Corrected (Date: _____, Initials: _____)					
<input type="checkbox"/> Retested (Date: _____, Initials: _____)					
<input type="checkbox"/> Other:					

Scenario ID: TS012	Test Type: System	Tester: Josh	Date: 5/18/15
Scenario Name:	RUN MANAGEMENT REPORTS		
Background and Test Objective(s):	It is the end of the month and management needs the "Contract Programmer's Monthly Expense Recap Report", "Contract Programmer Report - Fee Maximum vs. Actuals" and "Monthly Contract Recap". The objective of this test is to ensure that the user can navigate to the "Reports" menu and create the reports in the required date range.		
Related Use Cases or Test Scenarios:	UC012 - RUN MANAGEMENT REPORTS		
Testing Notes:	Assume that the user is already logged onto the system - is an active user.		

Case	Test Condition(s)	Test Activities	Expected Results/Notes	Actual Results	/Notes
01	The user must navigate to the reporting screen	Click the correct option to go to the reporting screen	The reporting screen should appear	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
02	The "Contract Programmer's Monthly Expense Recap Report" needs to be selected	Click the correct option to start the "Contract Programmer's Monthly Expense Recap Report"	The "Contract Programmer's Monthly Expense Recap Report" screen should appear	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
03	The date range needs to be entered	Select the correct date range for the current reporting period	The Date fields should accept the correct date range	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
04	The report must be printed	Click the option to print the report	The report should be sent to the printer	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
05	The "Contract Programmer Report - Fee Maximum vs. Actuals" needs	Click the correct option to start the "Contract Programmer Report - Fee	The "Contract Programmer Report - Fee Maximum vs. Actuals" screen	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

	to be selected	Maximum vs. Actuals"	should appear		
06	The date range needs to be entered	Select the correct date range for the current reporting period	The Date fields should accept the correct date range	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
07	The report must be printed	Click the option to print the report	The report should be sent to the printer	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
08	The "Monthly Contract Recap" needs to be selected	Click the correct option to start the "Monthly Contract Recap"	The "Monthly Contract Recap" screen should appear	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
09	The date range needs to be entered	Select the correct date range for the current reporting period	The Date fields should accept the correct date range	<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
10	The report must be printed	Click the option to print the report	The report should be sent to the printer	<input type="checkbox"/> Pass <input type="checkbox"/> 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:

7.2 APPENDIX-B: SAMPLE TEST DATA

Entity Name	Attribute Name	Valid Data	Invalid Data
Contacts			
	Contact_F_Name	Dan, Ron, any valid alpha characters a-z	D@n, R0n, 1234567890 ~!@#\$%^&*()_+= { } [] \ < > ? : " ; \ , or any combination of these characters
	Contact_L_Name	Ritz, any valid alpha characters a-z	Rit!, Sm*th, 1234567890 ~!@#\$%^&*()_+= { } [] \ < > ? : " ; \ , or any combination of these characters
	Contact_Phone	123-456-7890, or any valid combination of 1234567890	\$1234567890*, abcdefghijklmnopqrst uvwxyz~!@#\$%^&*()_+= { } [] \ < > ? : " ; \ , or any combo of these chars.
	Contact_Email	danRitz@mail.com, dRitz21@mail.com, or any valid combo of 1234567890 and a-z and must include an @ symbol	Dan#@mail.com ~!#\$%^&*()_+= { } [] \ < > ? : " ; \ , or any combo of these characters.
Contracts			
	Cont_Number	Ritz2015, or any contact last name and any valid combination of 123456789	Ritz#, Ritz@%, ~!#\$%^&*()_+= { } [] \ < > ? : " ; \ , or any combo of these characters.
	Cont_Start_Date	05-15-2015, or any valid date combination using 1234567890	5-15-15, May 15,2015, abcdefghijklmnopqrst uvwxyz~!@#\$%^&*()_+= { } [] \ < > ? : " ; \ , or any combo of these chars.
	Cont_End_Date	06-18-2015, or	6-18-15, June 18,

		any valid date combination using 1234567890	2015, abcdefghijklmnopqrstuvwxy~!@#%^&*()+={}[?><, or any combo of these chars.
	Cont_Fee_Max	\$50,000.00	Fifty thousand dollar, abcdefghijklmnopqrstuvwxy~!@#%^&*()+={}[?><, or any combo of these chars.
	Cont_Hourly_Rate	\$65.00, or any valid combination of 123456789	Sixty-five dollar, abcdefghijklmnopqrstuvwxy~!@#%^&*()+={}[?><, or any combo of these chars.
	Cont_Description	Any Text	none
	Cont_Notes	Any Text	none
Divisions			
	Div_Name	BAS, valid alpha characters a-z combination	BAS1, 1234567890~!@#%^&*()_+={ []\<>?:";\, or any combination of these characters
Employees			
	Emp_F_Name	Chris, Peter, any valid alpha characters a-z	CHR*s, P3t3r, 1234567890~!@#%^&*()_+={ []\<>?:";\, or any combination of these characters
	Emp_L_Name	Brown, Kaufman, any valid alpha characters a-z	Br0wn, K@ufman, 1234567890~!@#%^&*()_+={ []\<>?:";\, or any combination of these characters

	Emp_Contact	Chris Brown, Peter Van, any valid alpha characters a-z	CHR*s Br0wn, P3t3r V@n, 1234567890 ~!@#\$\$%^&*()_+= { } [] \ < > ? : " ; \ , or any combination of these characters
	Emp_User_Name	CBrown, any valid alpha characters a-z	C!Brown*, ~!@#\$\$%^&*()_+= { } [] \ < > ? : " ; \ , or any combination of these characters
	Emp_Password	S3cret1, any valid alpha characters a-z 1234567890 ~!@#\$\$%^&*()_+= ={ } [] \ < > ? : " ; \ , or any combination of these characters	Abc1, more than 6 characters
	Emp_Auth_Level	ACCT, MGMT, A/P, Accountant, Management, Account Payable, or any valid combination of a-z	Accountan1, M@nagement, 1234567890 ~!@#\$\$%^&*()_+= { } [] \ < > ? : " ; \ , or any combination of these characters
	Emp_Category	ACCT, MGMT, A/P, Accountant, Management, Account Payable, or any valid combination of a-z	Accountant2, Management1, 1234567890 ~!@#\$\$%^&*()_+= { } [] \ < > ? : " ; \ , or any combination of these characters
Invoices			
	Inv_Number	Ritz2015, or any contact last name and any valid combination of 123456789	Ritz#, Ritz@%, ~!#\$\$%^&*()+={ } [] ? > < , or any combo of these characters.
	Inv_Date_Issued	05-25-2015, or any valid date combination using 1234567890	5-25-15, May 26,2015, abcdefghijklmnopqrstuvwxy~!@#\$\$%^&*()+={ } [] ? > < , or

			any combo of these chars.
	Inv_Date_Received	05-26-2015, or any valid date combination using 1234567890	5-26-15, May 26,2015, abcdefghijklmnopqrstuvwxy~!@#\$%^&*()+={}[?><, or any combo of these chars.
	Inv_Date_Paid	05-28-2015, or any valid date combination using 1234567890	5-28-15, May 28,2015, abcdefghijklmnopqrstuvwxy~!@#\$%^&*()+={}[?><, or any combo of these chars.
	Inv_Terms	PIA, Net 7-90, EOM, COD, 1% 10 Net 30, ETC.	P1A, N3t 7-10, ~!@#\$%^&*()+={}[?><, or any combo of these chars.
	Inv_Amount	\$51,000.00, any valid combination using 1234567890	Fifty one thousand dollar, abcdefghijklmnopqrstuvwxy~!@#\$%^&*()+={}[?><, or any combo of these chars.
	Inv_Rate	65, any combo of 1234567890 up to 100	abcdefghijklmnopqrst uvwxy~!@#\$%^&*()+={}[?><, any combo of these characters, or any value over 100.
	Inv_Hours	18, 22, 40, up to 168, or any combo of 1234567890 up to 168	Eighteen, twenty-two, Forty, 1Hundred69, 169, 2002, 5/12/02, abcdefghijklmnopqrst uvwxy~!@#\$%^&*()+={}[?><, any combo of these characters, or any value over 168.

	Inv_Start_Date	05-25-2015, or any valid date combination using 1234567890	5-25-15, May 15,2015, abcdefghijklmnopqrstuvwxy~!@#%&*()+={}[!]?><, or any combo of these chars.
	Inv_End_Date	05-28-2015, or any valid date combination using 1234567890	5-28-15, May 15,2015, abcdefghijklmnopqrstuvwxy~!@#%&*()+={}[!]?><, or any combo of these chars.
	Inv_Status	Pending Invoice, Approved, Accrual Needed, Paid	P3nding, Inv0ice, Approv*d, Accru@l, Need3d, Pa!d, 1234567890 ~!@#%&*()+={}[!]?><?:"\;, or any combination of these characters
	Inv_Accrual	05-25-2015, or any valid date combination using 1234567890	5-25-15, May 15,2015, abcdefghijklmnopqrstuvwxy~!@#%&*()+={}[!]?><, or any combo of these chars.
	Inv_Notes	Any Text	None
Programmers			
	Prog_F_Name	Zach, Sue, any valid alpha characters a-z	Z@ch, Su3, 1234567890 ~!@#%&*()+={}[!]?><?:"\;, or any combination of these characters
	Prog_L_Name	Smith, Hill, any valid alpha characters a-z	Sm1th, Hi!!, 1234567890 ~!@#%&*()+={}[!]?><?:"\;, or any combination of these characters
Units			
	Unit_Number	3135, four digit	31a1,

		number, any valid combination of 123456789	abcdefghijklmnopqrs tuvwxyz~!@#%^&* ()+={}][?><, or any combo of these chars.
Vendors			
	Ven_Name	Redford Group, APEX, any valid alpha characters a- z	R3dford Gr0up, AP3X, 1234567890 ~!@#\$\$%^&*()_+= { } []\<>?:";\, or any combination of these characters
	Ven_Number	ZZ0023, Six digits, two alpha character a-z and any valid combination of 123456789	Z0235@, ~!@#\$\$%^&*()_+= { } []\<>?:";\, or any combination of these characters
	Ven_Address	20000 68th Ave W, or any valid combination of 1234567890 and a- z	2000 6*th a&e w, ~!@#\$\$%^&*()+= { }][?><, or any combo of these characters.
	Ven_Phone	123-456-7890, or any valid combination of 1234567890	\$1234567890*, abcdefghijklmnopqrs t uvwxyz~!@#\$\$%^& *()+={}][?><, or any combo of these chars.
	Ven_Email	danRitz@mail.com, dRitz21@mail.com, or any valid combo of 1234567890 and a-z and must include an @ symbol	Dan#@mail.com ~!#\$\$%^&*()+= { }][?><, or any combo of these characters.

Evaluation Criteria – Test Plan Draft

Team Name and Members:

Criteria	Points	Given	Comments
<ul style="list-style-type: none"> □ Completed and submitted on or before the due date 	5	5	
<ul style="list-style-type: none"> □ Test Plan contains content specified in assignment <ul style="list-style-type: none"> ○ Assignment Title Page ○ Transmittal Cover Memo ○ Report Title Page ○ Table of Contents □ Body of document discusses the following: <ul style="list-style-type: none"> ○ Introduction ○ Testing Approach ○ Test Environment ○ Test Scenarios ○ Appendix includes all testing forms (both new and completed) and list with all completed sample test data 	25	22	The “when” and various other comments throughout – mostly on test scenarios
<ul style="list-style-type: none"> □ Test plan demonstrates an understanding of the role and importance of testing, as well as the different testing methods and how they should be used 	10	10	

<ul style="list-style-type: none"> ❑ Written in appropriate style for the audience ❑ Document organization and content demonstrate careful planning and thought and is easy to follow ❑ Well-written, free of spelling and grammar errors 	10	10	
Total	50	47	Good job – needs some changes made.



**Edmonds Community College
Computer Information Systems
User Manual**

Prepared: June 4th 2015

Scrappers Systems Analysis and Development Group:

Jennifer Lam

Joshua Chong

Duane Samuels

Cody Smith

Memo

To: Patrick Jay, Vice President & Manager
From: Scrappers Systems Analysis Group
Date: June 4th, 2015
Subject: User Manual

Scrappers systems Analysis Group has created and attached a User Manual that will be used for training employees of the Bank of Xanadu for the use of the new automated system for managing contract payables. This test plan document contains the following sections

- System Overview
- Navigation
- Data Entry Screens & Step-by-Step Procedures
- Reports
- Where to Go for More Help

We would like to request a meeting with accounting on June 4th 2015 in the Bellevue branch. During this meeting, we will teach the employees of the bank on the proper use of the new system. Following the meeting each employee present should have enough knowledge of the system to be able to instruct a fellow employee on the use of the system, if needed.

User Manual
New Automated System for Managing Contract
Payables

Bank of Xanadu
Accounting Department

Scrappers Systems Analysis Group:

Jennifer Lam

Joshua Chong

Duane Samuels

Cody Smith

Prepared: June 4th 2015

1 TABLE OF CONTENTS

2	Introduction	5
3	System Overview.....	6
4	Navigation	7
5	Data Entry Screens & Step-by Step Procedures.....	8
6	Reports	44
7	Where to Go For More Help	45
8	Appendices	46

2 INTRODUCTION

This document has been created by Team Scrappers System Analysis to instruct users how to use the new Bank of Xanadu system for managing contract payables. This document will give a brief overview of the purpose of the new system, and how users will navigate through it. The user manual will contain pictures of every screen in the system that visually indicate each important button and field on the screen with a description of its purpose. The user manual will also contain detailed instructions on how a user navigates through the system to find each screen, how to create new records of invoices or contracts, and how reports are generated. If additional information is needed, users can refer to the "Where to Go for More Help" section of the user manual, where they can find information on how to contact us for additional support.

3 SYSTEM OVERVIEW

This new contract payables system has been created by Team Scrappers to enable processing of invoice payments in accordance to contractual time and fee limitations. This new system will replace the existing Excel spreadsheet system that is used by accountants to keep track of new invoices and contracts. The main features of the new contract payable system that will improve company productivity are:

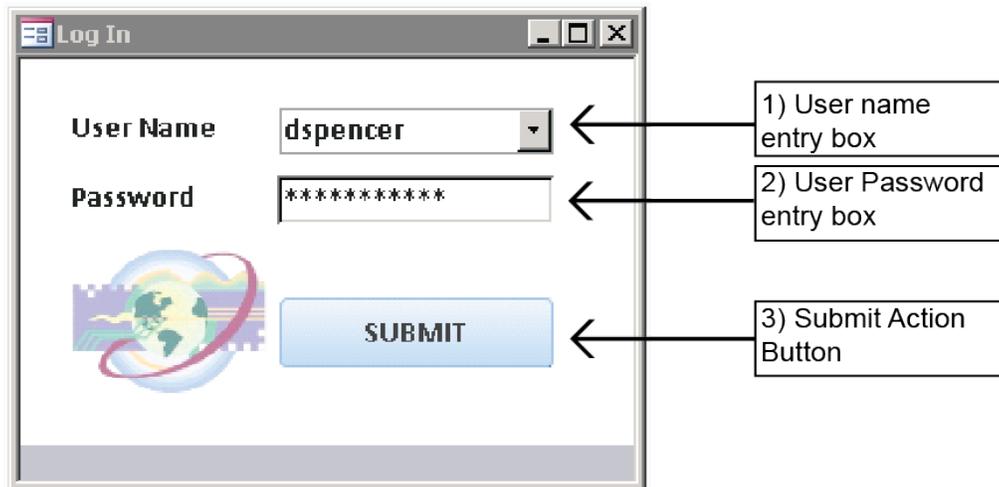
- Automated lookups to find if an invoice:
 - Is within the agrees upon contract start and end dates
 - Is using the correct hourly pay rate
 - Will go over the contracts funding limit
- Secure login system for the accounting department
- Easily navigable menus and tables

4 NAVIGATION

The method in which users will navigate through the new system will be simple and easy to learn. First, the user will open the database and be prompted to login by entering their unique username and password. The user will then see the main menu where they will find buttons along the top of the screen that will take them to the contract menu, invoice menu, and the reporting menu. From these screens, you can add, edit and sort the data in ways such as active or inactive contracts, pending or approved invoices, and create exceptions for the buyer.

5 DATA ENTRY SCREENS & STEP-BY STEP PROCEDURES

Login to the System



The image shows a screenshot of a web browser window titled "Log In". The window contains a login form with the following elements:

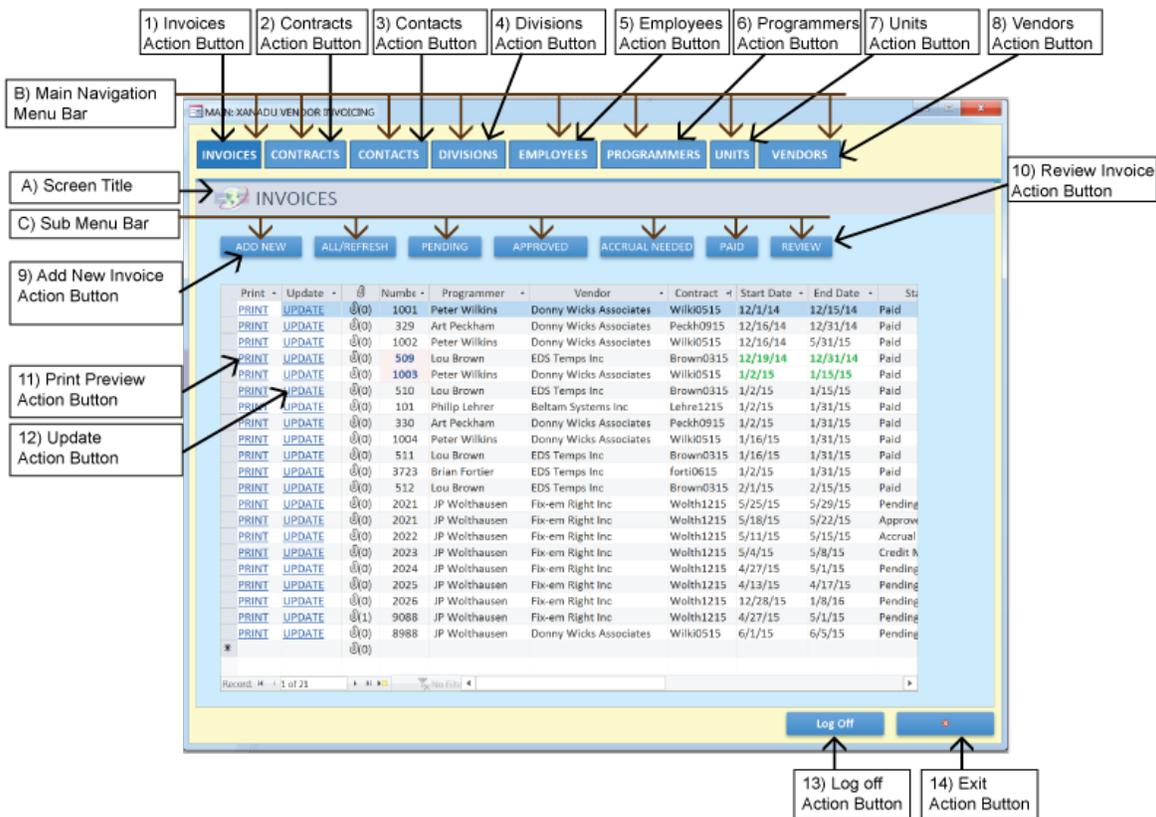
- User Name:** A text input field containing the text "dspencer".
- Password:** A text input field containing ten asterisks "*****".
- Submit Button:** A blue button labeled "SUBMIT".
- Decorative Element:** A graphic of a globe with a red and blue orbital path.

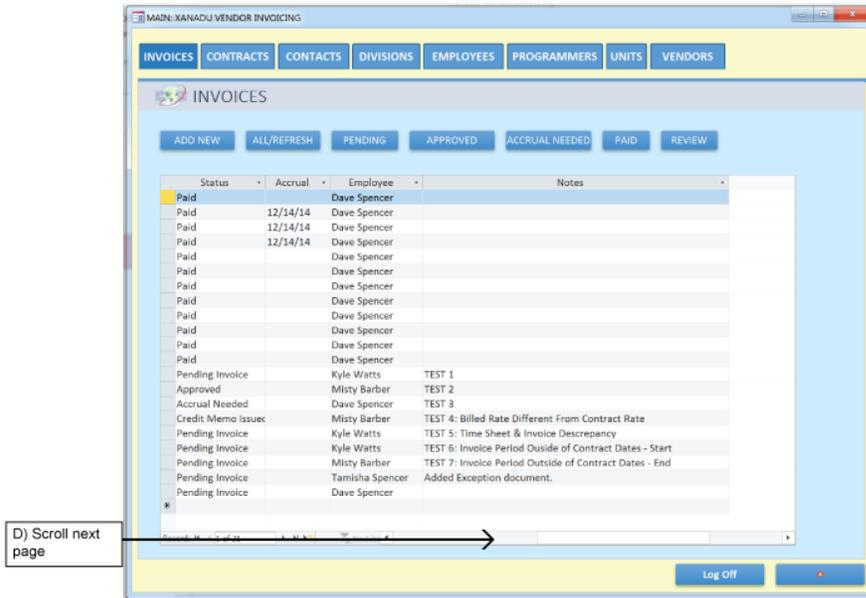
Three callout boxes with arrows point to the form elements:

- 1) User name entry box
- 2) User Password entry box
- 3) Submit Action Button

- 1) Enter the appropriate **User Name**
- 2) Enter the appropriate **Password**
- 3) Click the **Submit** button to access the system

Invoice Main Screen





A) **Screen Title** is the title of the current screen

B) **Main Navigation** is the interactive menu bar section of the screen

1) Click on the **Invoices** button to access the Invoice screen

2) Click on the **Contracts** button to access the Contract screen

3) Click on the **Contacts** button to access the Contact screen

4) Click on the **Divisions** button to access the division screen

5) Click on the **Employees** button to access the Employee screen

6) Click on the **Programmers** button to access the Programmer screen

7) Click on the **Units** button to access the Unit screen

8) Click on the **Vendors** button to access the Vendor screen

C) **Sub menu bar** is the secondary navigation of the selected screen

9) Click on the **Add New** Invoice button to access a new invoice popup screen

10) Click on the **Review** Invoice button to access a review invoice screen

- 11) Click on the **Print** button to access a print screen
 - 12) Click on the **Update** button to access a edit invoice screen
 - 13) Click on the **Log off** button to sign out
 - 14) Click on the **Exit** button to close the application
- D) Click on the **Scroll next page** of the current invoice screen to access the other page.

Add Invoice Screen

The screenshot shows the 'INVOICE: ADD NEW RECORD' screen with the following fields and callouts:

- 1) Screen Title: 'INVOICE: ADD NEW RECORD'
- 2) Invoice # entry box: Invoice #
- 3) Programmer Name drop-down box: Programmer
- 4) Vendor Name drop-down box: Vendor
- 5) Contract ID drop-down box: Contract
- 6) Date Issued entry box: Date Issued
- 7) Date Received entry box: Date Received
- 8) Start Date entry box: Inv Start Date
- 9) Employee name drop-down box: Employee
- 10) Term drop-box box: Terms
- 11) Hourly Rate entry box: Inv Hr Rate
- 12) End Date entry box: Inv End Date
- 13) Total Hours entry box: Total Hours
- 14) Invoice Amount entry box: Invoice Amount
- 15) Status drop-down box: Status (Pending Invoice)
- 16) Date Paid entry box: Date Paid
- 17) Note entry box: Invoice Notes
- 18) Save Action Button: Save
- 19) Go back Action Button: Back

Additional elements on the screen include:

- Invoice ID: [New]
- Unit Number
- Cont Start Date, Cont End Date, Contract Hr. Rate
- Max Amount, Remaining Max
- Contract Description
- Instructions: In 'Contract' title area, click on the downward triangle and then just select the correct contract name and press the 'OK' button.
- If the name is not listed: None have been paid.

Contract	Invoice	Amount	Status
Brown0315	512	\$1,700.00	Paid
Brown0315	511	\$1,750.00	Paid
Brown0315	510	\$1,700.00	Paid
Brown0315	509	\$1,750.00	Paid
fort10615	3723	\$4,412.50	Paid
Lehre1215	101	\$8,580.00	Paid
Total		\$51,635.50	

- 1) **Screen Title** is on top of the current screen
- 2) Enter the **Invoice #** in the entry box
- 3) Select from the drop-down list and select the appropriate **Programmer Name**
- 4) Select from the drop-down list the appropriate **Vendor Name**
- 5) Select from the drop-down list the appropriate **Contract ID**
- 6) Enter the **Date Issued** in the entry box.

- 7) Enter the **Date Received** in the entry box
- 8) Enter the **Start Date** in the entry box
- 9) Select from the drop-down list the appropriate **Employee Name**
- 10) Select from the drop-down list the appropriate **Term**
- 11) Enter the **Hourly Rate** in the entry box
- 12) Enter the **End Date** in the entry box
- 13) Enter the **Total Hours** in the entry box
- 14) Enter the **Invoice Amount** in the entry box
- 15) Click on the drop-down box and select the appropriate **Status**
- 16) Enter the **Date Paid** in the entry box
- 17) Enter a **Note** in the entry textbox
- 18) Click the **Save** button to save the invoice entry
- 19) Click the **Go Back** button to return to the invoice main screen

Review Invoice Screen

INVOICE DETAILED VIEW (read only)

Invoice ID: [] Invoice #: 1001

Programmer: Peter Wilkins

Vendor: Donny Wicks Associates

Contract: Wilki0515 Unit Number: 3580

Date Issued: 12/16/14 Date Received: 12/17/14

Cont Start Date: 12/1/14 Cont End Date: 5/30/15

Inv Start Date: 12/1/14 Inv End Date: 12/15/14

Terms: ETC Contract Hourly Rate: \$59.00 Hr x Rate = \$3,776.00

Total Hours: 64 Inv Hr Rate: \$59.00 Invoice Amount: \$3,776.00

Max Amount: \$48,000.00 Remaining Max: \$30,477.00

Date Paid: 12/21/14 Accrual Date: []

Employee: Dave Spencer Status: Paid

Contract Description: []

Invoice Notes: []

Attached Documents: []

USE NAVIGATION CONTROLS BELOW TO NAVIGATE RECORDS

Record 1 of 20

1) Review records Action Button

1) Preview all the records using the next record button.

Print Preview Invoice Screen

INVOICE PRINT VIEW
✕

Invoice ID

VENDOR INVOICE 1001

Programmer: Peter

Vendor: Donny Wicks Associates

Contract: Wilki0515 Unit Number: 3580 Terms: ETC

Date Issued: 12/16/14 Date Recieved: 12/17/14

Contract Start Date: 12/1/14 Contract End Date: 5/30/15

Invoice Start Date: 12/1/14 Invoice End Date: 12/15/14

Contract Hourly Rate: \$59.00 Total Hours: 64 Hr x Rate = \$3,776.00

Invoice Hourly Rate: \$59.00 Invoice Amount: \$3,776.00

Maximum Amount: \$48,000.00 Remaining Maximum: \$30,477.00

Date Paid: 12/21/14 Accrual Date:

Employee: Dave Status: Paid

Contract Description:

Invoice Notes:

Attached Documents

Record: 1 of 1
Filtered
Search

The print preview button display the document on the display screen.

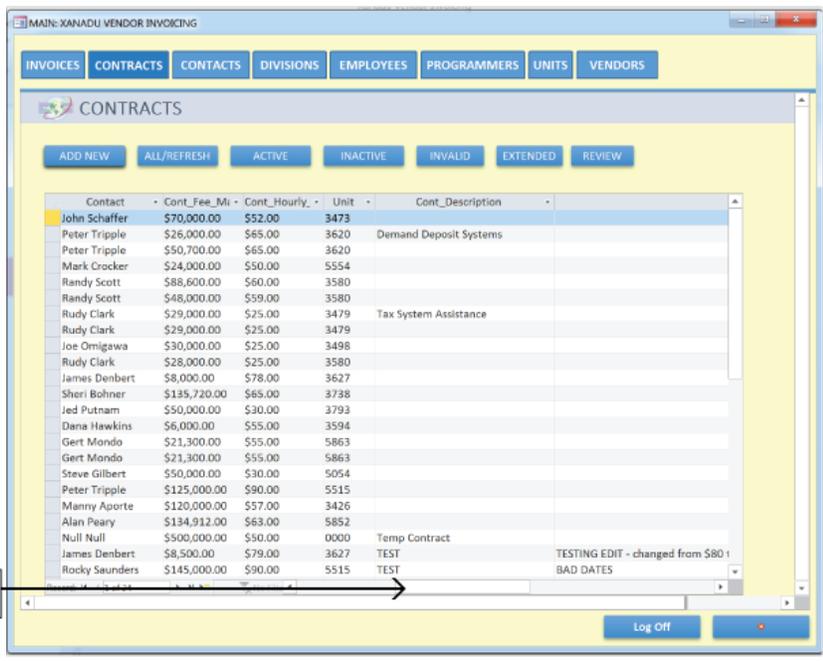
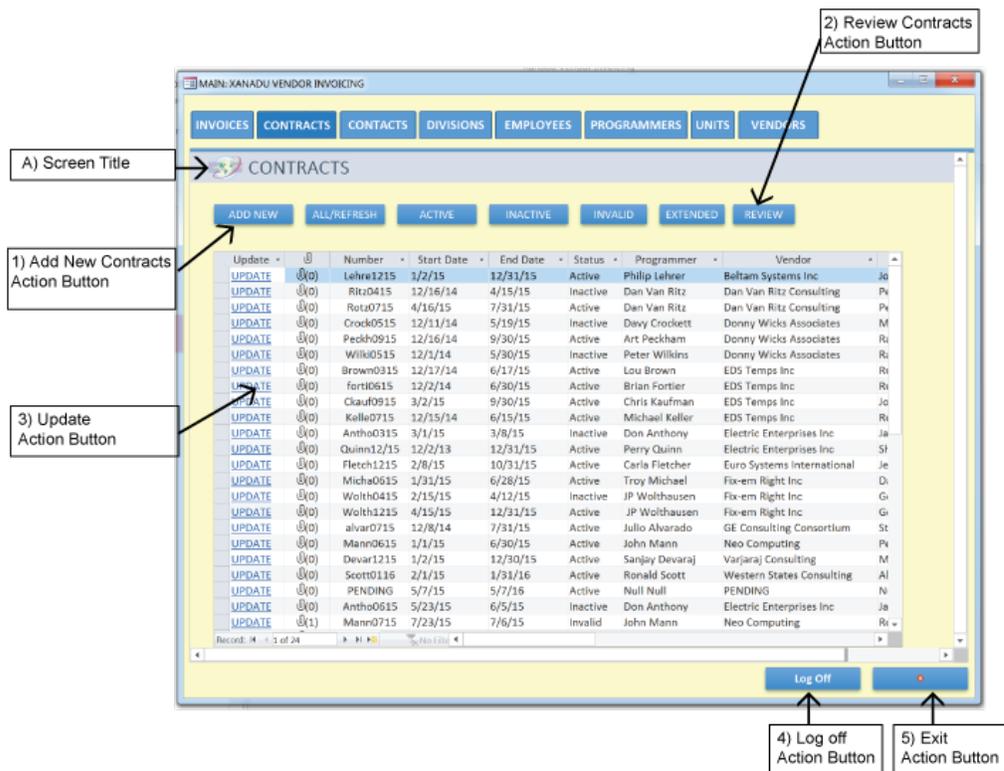
Update Invoice Screen

The screenshot shows a software window titled "INVOICE UPDATE NOTES" with a sub-header "INVOICE: NOTE UPDATE". The window contains a form with various fields for invoice details. At the bottom of the form, there are five callout boxes with arrows pointing to specific elements:

- 1) Document Action Button: Points to the "Timesheet or Exception Doc." button.
- 2) Note entry box: Points to the large yellow text area for "Invoice Notes".
- 3) Attach a file Action Button: Points to the "Attach Document" button.
- 4) Save Action Button: Points to the "Save" button at the bottom center.
- 5) Go back Action Button: Points to the "Go back" button at the bottom right.

- 1) Click on the **Timesheet or Exception Documents** button to access the documents screen.
- 2) Enter a Updated **Note** in the entry textbox
- 3) **Attach a Document** when creating an exception document.
- 4) Click the **Save** button to save the invoice entry
- 5) Click the **Go back** button to return to the Invoice main screen

Contracts Main Screen



A) **Screen Title** is on top of the current screen

- 1) Click on the **Add New** Contract button to access a new contracts popup screen
- 2) Click on the **Review** Contract button to access a review invoice screen
- 3) Click on the **Update** button to modify the contract screen
- 4) Click on the **Log off** button to sign out
- 5) Click on the **Exit** button to close the application

B. Click on the **Scroll next page** of the current contract screen to access the other page.

Add Contract Screen

The screenshot shows a software window titled 'CONTRACTS: ADD NEW RECORD'. The window contains several input fields and buttons. Numbered callouts identify the following elements:

- 1) Screen Title: 'CONTRACTS: ADD NEW RECORD'
- 2) Vendor Name drop-down list
- 3) Contact Name drop-down list
- 4) Programmer Name drop-down list
- 5) Start Date entry box
- 6) Max Fee Amount entry box
- 7) Description Text box
- 8) Notes text box
- 9) Contract # entry box
- 10) Unit drop down list
- 11) Employee Name drop down list
- 12) Hourly Rate entry box
- 13) End Date entry box
- 14) Save Action Button
- 15) Go Back Action Button

- 1) **Screen Title** is on top of the current screen
- 2) Select from the drop-down list the appropriate **Vendor Name**
- 3) Select from the drop-down list the appropriate **Contact Name**
- 4) Select from the drop-down list the appropriate **Programmer Name**
- 5) Enter the **Start Date** in the entry box
- 6) Enter the **Max Fee Amount** in the entry box
- 7) Enter the **Description** in the entry textbox
- 8) Enter the **Note** in the entry textbox
- 9) Enter the **Contract #** in the entry box
- 10) Select from the drop-down list the appropriate **Unit**
- 11) Select from the drop-down list the appropriate **Employee Name**
- 12) Enter the **Hourly Rate** in the entry box
- 13) Enter the **End Date** in the entry box

- 14) Click the **Save** button to save the contract entry
- 15) Click the **Go Back** button to return to the contract main screen

Review Contract Screen

CONTRACTS DETAILED VIEW

CONTRACTS: DETAILED VIEW (read only)

ID	2	Contract #	Lehre1215
Vendor	Beltam Systems Inc	Unit	3473
Contact	John Schaffer	Employee	Dave Spencer
Programmer	Philip Lehrer	Hourly Rate	\$52.00
Start Date	1/2/15	End Date	12/31/15
Maximum Fee	\$70,000.00		

Description

Notes

Attached Documents

USE NAVIGATION CONTROLS BELOW TO NAVIGATE RECORDS

Record: 14 of 24

1) Review records Action Button

1) Review all the records using the next record button.

Update Contract Screen

The screenshot shows a software window titled 'CONTRACTS NOTES' with a sub-header 'CONTRACTS: NOTE UPDATE'. The form contains the following fields and controls:

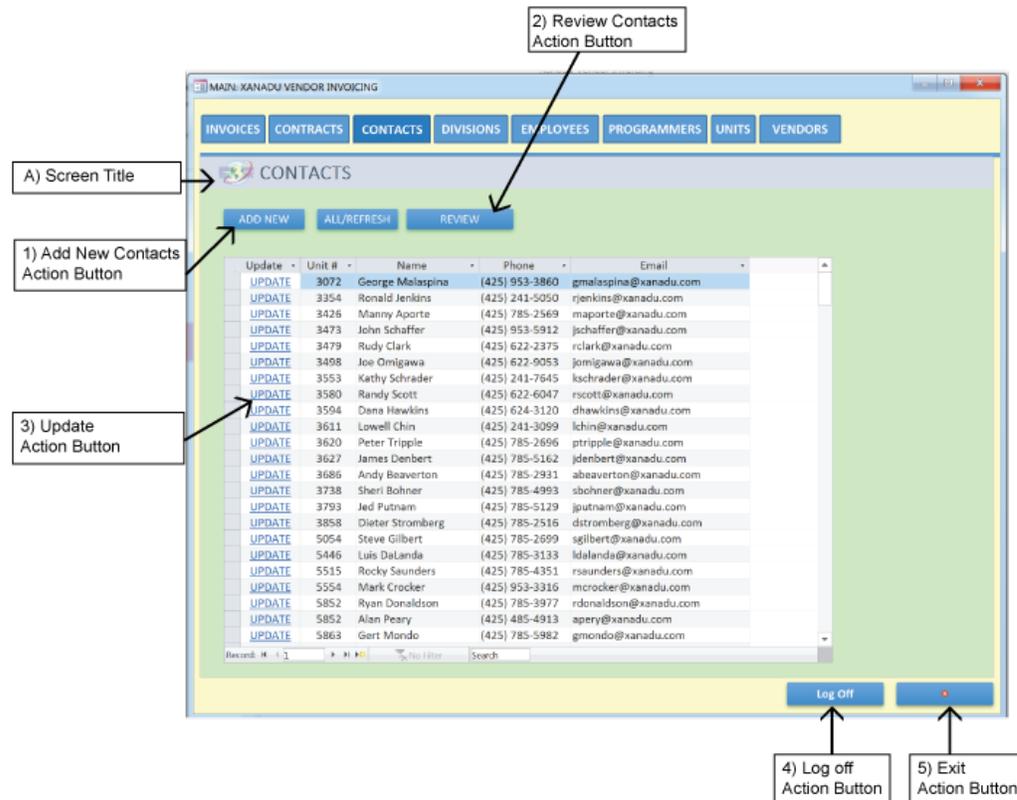
- ID:** 2
- Contract #:** Lehre1215
- Vendor:** Beltam Systems Inc
- Unit:** 3473
- Contact:** John Schaffer
- Employee:** Dave Spencer
- Programmer:** Philip Lehrer
- Hourly Rate:** \$52.00
- Start Date:** 1/2/15
- End Date:** 12/31/15
- Maximum Fee:** \$70,000.00
- Description:** Exception Document
- Notes:** A large yellow text area for entering notes.
- Buttons:** 'Exception Document' (orange), 'Attach Document' (yellow), 'Save' (blue), and 'Go back' (blue).

Callouts point to the following elements:

- 1) Document Action Button (points to the 'Exception Document' button)
- 2) Note entry box (points to the 'Notes' text area)
- 3) Attach a file Action Button (points to the 'Attach Document' button)
- 4) Save Action Button (points to the 'Save' button)
- 5) Go back Action Button (points to the 'Go back' button)

- 1) Click on the **Documents** button to access the documents screen.
- 2) Enter a **Note** in the entry textbox
- 3) **Attach a Document** when creating an exception document.
- 4) Click the **Save** button to save the contract entry
- 5) Click the **Go back** button to return to the contract main screen

Contacts Main Screen



A) **Screen Title** is on top of the current screen

- 1) Click on the **Add New** Contacts button to access a new contact popup screen
- 2) Click on the **Review** Contacts button to access a preview contact screen
- 3) Click on the **Update** Contacts button to access a modify contacts screen
- 4) Click on the **Log Off** to sign out
- 5) Click on the **Exit** button to close the application

Add Contact Screen

The screenshot shows a software window titled 'CONTACTS UPDATE'. The main title is 'CONTACTS: UPDATE'. The form contains the following fields and controls:

- Contact ID:** A text box containing the number '1'.
- Unit #:** A drop-down list showing '3072'. A callout box labeled '2) Unit drop down list' points to this field.
- First Name:** A text box containing 'George'.
- Last Name:** A text box containing 'Malaspina'.
- Phone:** A text box containing '(425) 953-3860'. A callout box labeled '3) Phone # entry box' points to this field.
- Email:** A text box containing 'gmalaspina@xanadu.com'. A callout box labeled '4) Email address entry box' points to this field.
- Buttons:** Two blue buttons are located at the bottom. The left button has a save icon and is labeled '5) Save Action Button'. The right button has a back arrow icon and is labeled '6) Go Back Action Button'.

A) **Screen Title** is on top of the current screen

1) Select from the drop-down list the appropriate **Unit**

2) Enter the **First Name** in the entry box.

3) Enter the **Last Name** in the entry box.

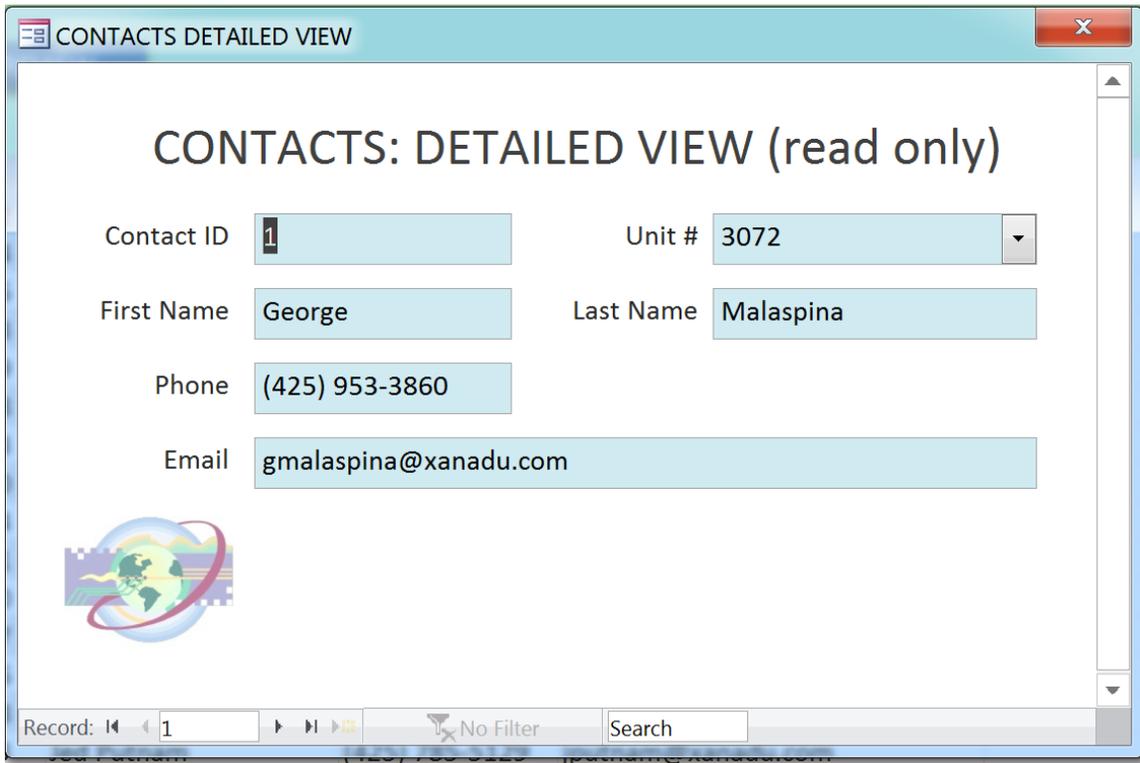
4) Enter the **Phone Number** in the entry box.

5) Enter the **Email Address** in the entry box.

6) Click the **Save** button to save the contact entry

7) Click the **Go Back** button to return to the contact main screen

Review Contact Screen



The screenshot shows a web application window titled "CONTACTS DETAILED VIEW". The main heading is "CONTACTS: DETAILED VIEW (read only)". The contact information is displayed in a form with the following fields:

Contact ID	1	Unit #	3072
First Name	George	Last Name	Malaspina
Phone	(425) 953-3860		
Email	gmalaspina@xanadu.com		

At the bottom left, there is a globe icon with a red orbital path. At the bottom of the window, there is a navigation bar with the following elements:

- Record: 1 (with navigation arrows)
- No Filter (with a filter icon)
- Search (with a search input field)

- 1) Preview the records of the contact.

Update Contact Screen

The screenshot shows a software window titled "CONTACTS UPDATE". The window contains the following fields and controls:

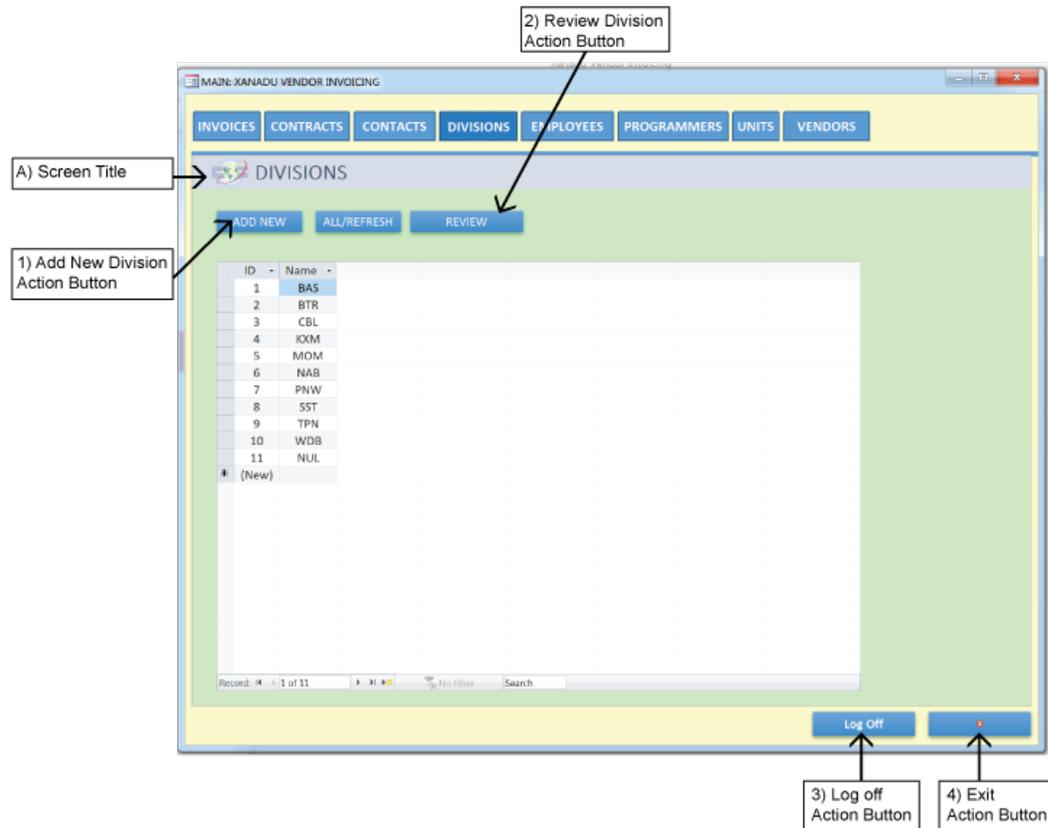
- Contact ID:** An empty text entry box.
- Unit #:** A dropdown menu currently showing "3072".
- First Name:** A text entry box containing "George".
- Last Name:** A text entry box containing "Malaspina".
- Phone:** A text entry box containing "(425) 953-3860".
- Email:** A text entry box containing "gmalaspina@xanadu.com".
- Buttons:** Two blue buttons at the bottom. The left button has a floppy disk icon and is labeled "4) Save Action Button". The right button has a left-pointing arrow icon and is labeled "5) Go Back Action Button".

Five callout boxes with arrows point to specific elements:

- "1) Unit drop down list" points to the Unit # dropdown.
- "2) Phone # entry box" points to the Phone text box.
- "3) Email address entry box" points to the Email text box.
- "4) Save Action Button" points to the left button.
- "5) Go Back Action Button" points to the right button.

- 1) Enter the updated **Unit** in the entry box.
- 2) Enter the updated **Phone #** in the entry box.
- 3) Enter the updated **Email Address** in the entry box
- 4) Click the **Save** button to save the contact entry
- 5) Click the **Go back** button to return to the contact main screen

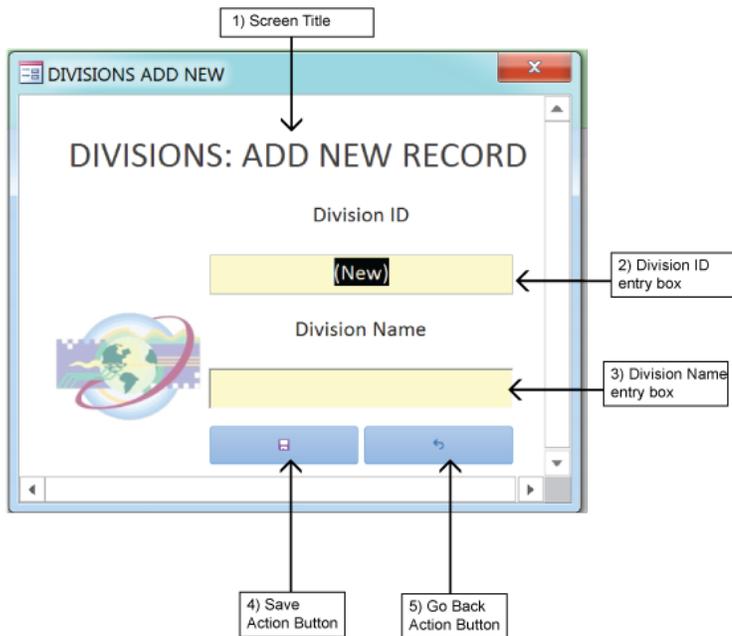
Division Main Screen



A) **Screen Title** is on top of the current screen

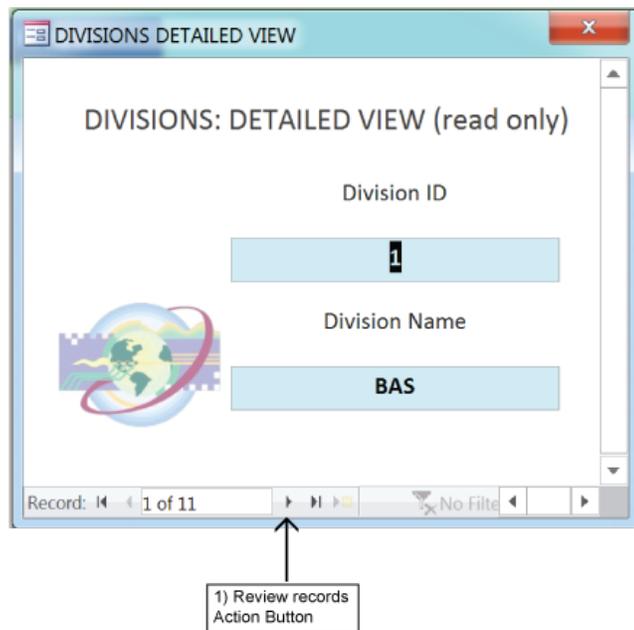
- 1) Click on the **Add New** Division button to access a new division popup screen
- 2) Click on the **Review** Division button to access a preview division screen
- 3) Click on the **Log Off** to sign out
- 4) Click on the **Exit** button to close the application

Add New Division Screen



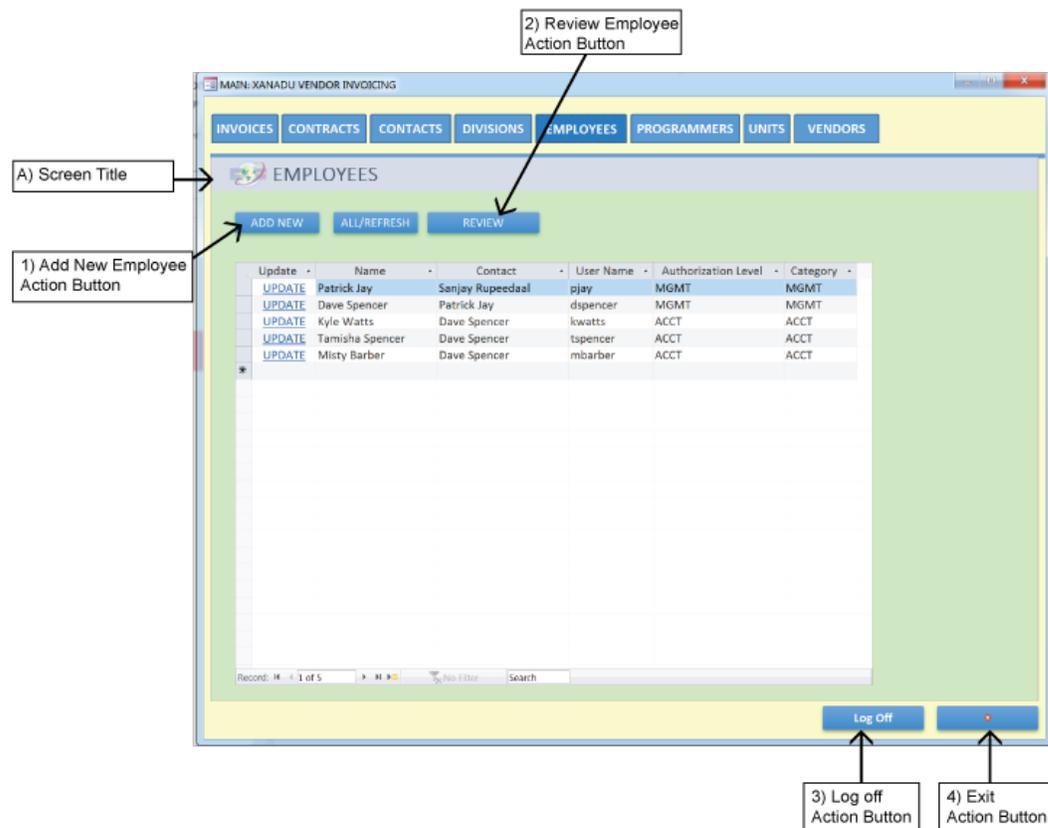
- 1) **Screen Title** is on top of the current screen
- 2) Enter the Division ID in the entry box
- 3) Enter the Division Name in the entry box
- 4) Click the **Save** button to save the division entry
- 5) Click the **Go Back** button to return to the division main screen

Review Division Screen



1) Preview all the records using the next record button.

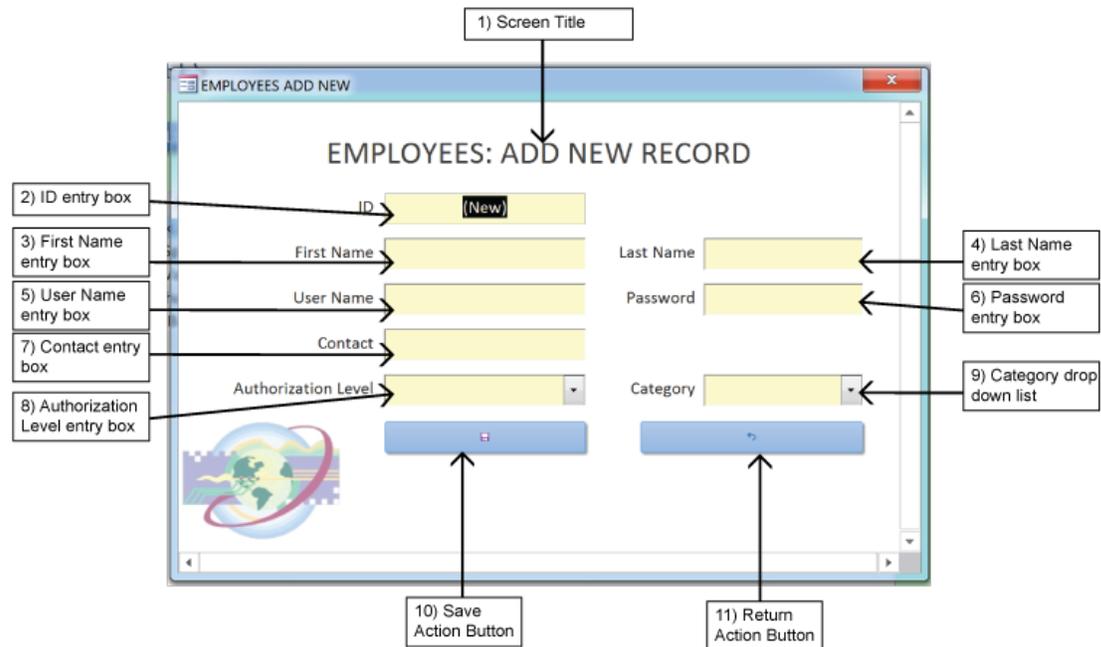
Employee Main Screen



A) **Screen Title** is on top of the current screen

- 1) Click on the **Add New** Employee button to access a new employee popup screen
- 2) Click on the **Review** Employee button to access a preview employee screen
- 3) Click on the **Log Off** to sign out
- 4) Click on the **Exit** button to close the application

Add New Employee Screen



- 1) **Screen Title** is on top of the current screen
- 2) Enter the **ID** in the entry box
- 3) Enter the **First Name** in the entry box
- 4) Enter the **Last Name** in the entry box
- 5) Enter the **User Name** in the entry box
- 6) Enter the **Password** in the entry box
- 7) Enter the **Contact** in the entry box
- 8) Select from the drop-down list the appropriate **Authorization Level**
- 9) Select from the drop-down list the appropriate **Category**
- 10) Click the **Save** button to save the employee entry
- 11) Click the **Go Back** button to return to the employee main screen

Review Employee Screen

EMPLOYEES DETAILED VIEW

EMPLOYEES: DETAILED VIEW

ID	<input type="text" value="1"/>		
First Name	<input type="text" value="Patrick"/>	Last Name	<input type="text" value="Jay"/>
User Name	<input type="text" value="pjay"/>	Password	<input type="text" value="*****"/>
Contact	<input type="text" value="Sanjay Rupeedaal"/>		
Authorization Level	<input type="text" value="MGMT"/>	Category	<input type="text" value="MGMT"/>



Record: 1 | No Filter | Search

1) Preview the records of the employee.

Update Employee Screen

The screenshot shows a web application window titled "EMPLOYEES UPDATE". The main heading is "EMPLOYEES: UPDATE". The form contains the following fields and controls:

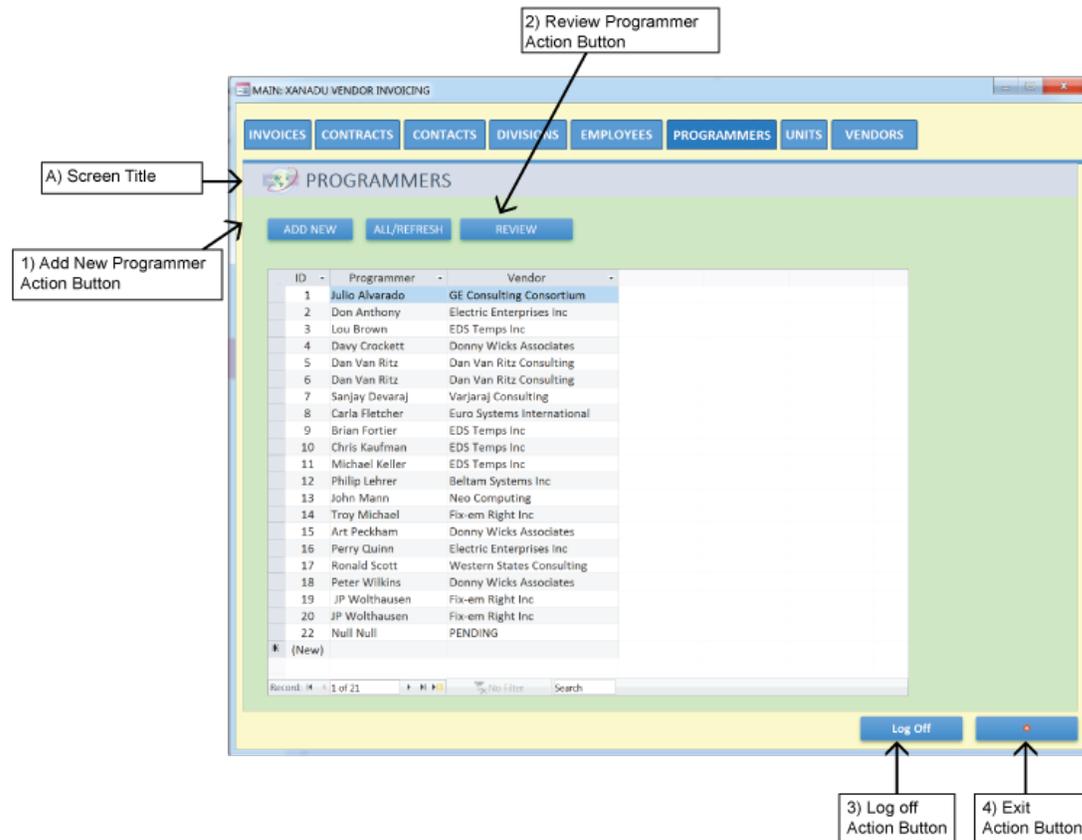
- ID: A text input field with the value "1".
- First Name: A text input field with the value "Patrick".
- Last Name: A text input field with the value "Jay".
- User Name: A text input field with the value "pjay".
- Password: A password input field with the value "*****".
- Contact: A text input field with the value "Sanjay Rupeedaal".
- Authorization Level: A drop-down menu with the value "MGMT".
- Category: A drop-down menu with the value "MGMT".
- Save Action Button: A blue button with a white "S" icon.
- Return Action Button: A blue button with a white "B" icon.

Annotations with arrows point to the following elements:

- 1) Password entry box
- 2) Contact entry box
- 3) Authorization Level entry box
- 4) Category drop down list
- 5) Save Action Button
- 6) Return Action Button

- 1) Enter the updated Password in the entry box
- 2) Enter the updated Contact in the entry box
- 3) Selected the updated Authorization Level from the drop-down list.
- 4) Selected the updated Category from the drop-down list.
- 5) Click the **Save** button to save the employee entry
- 6) Click the **Go Back** button to return to the employee main screen

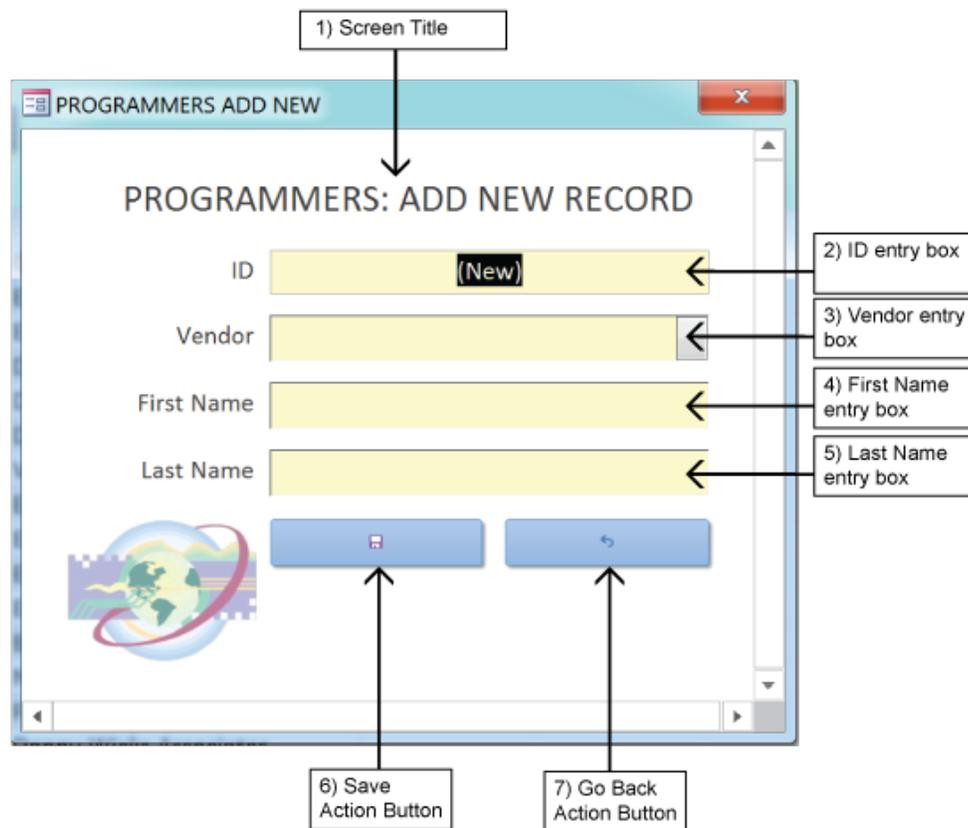
Programmer Main Screen



A) **Screen Title** is on top of the current screen

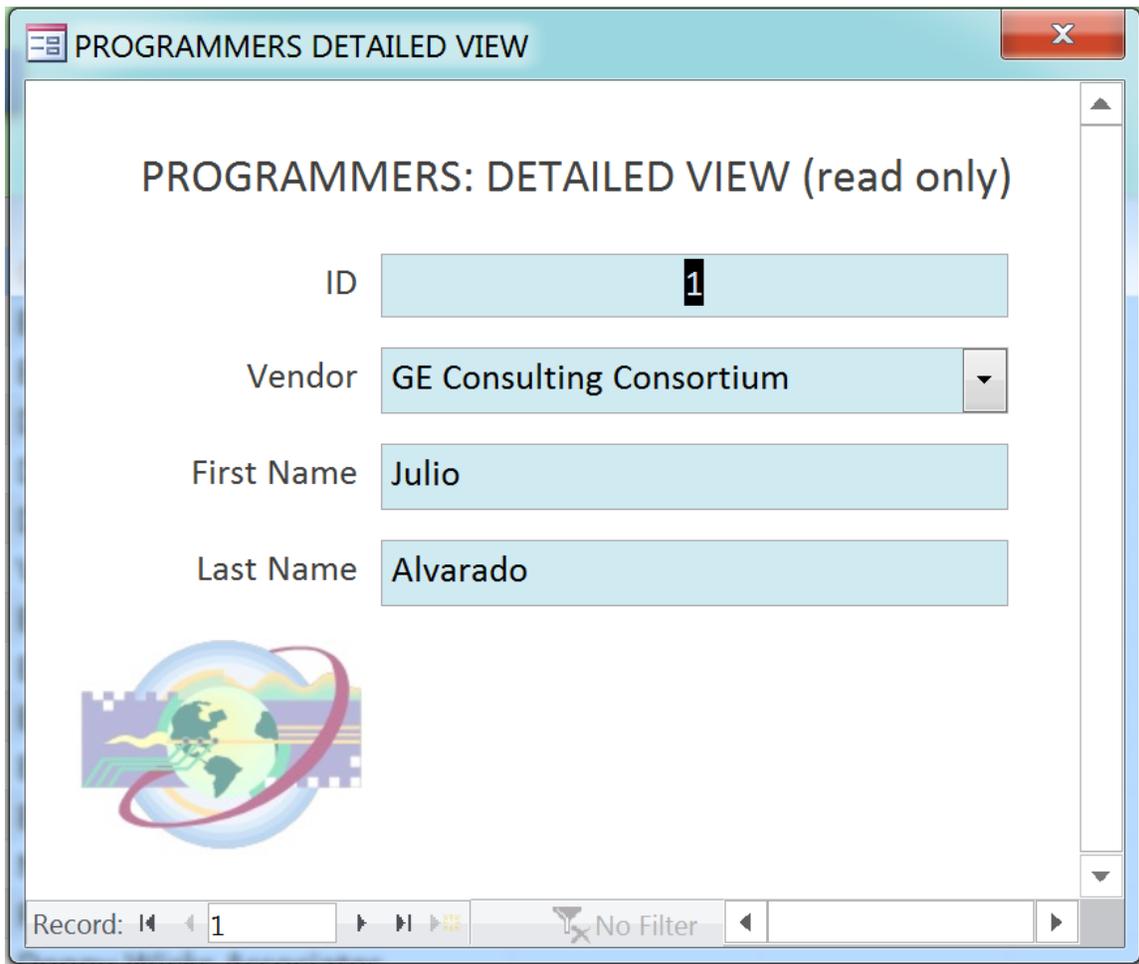
- 1) Click on the **Add New** Programmer button to access a new programmer popup screen
- 2) Click on the **Review** Programmer button to access a preview programmer screen
- 3) Click on the **Log Off** to sign out
- 4) Click on the **Exit** button to close the application

Add New Programmer Screen



- 1) **Screen Title** is on top of the current screen
- 2) Enter the **ID** in the entry box
- 3) Select from the drop-down list the appropriate **Vendor**
- 4) Enter the **First Name** in the entry box
- 5) Enter the **Last Name** in the entry box
- 6) Click the **Save** button to save the programmer entry
- 7) Click the **Go Back** button to return to the programmer main screen

Review Programmer Screen



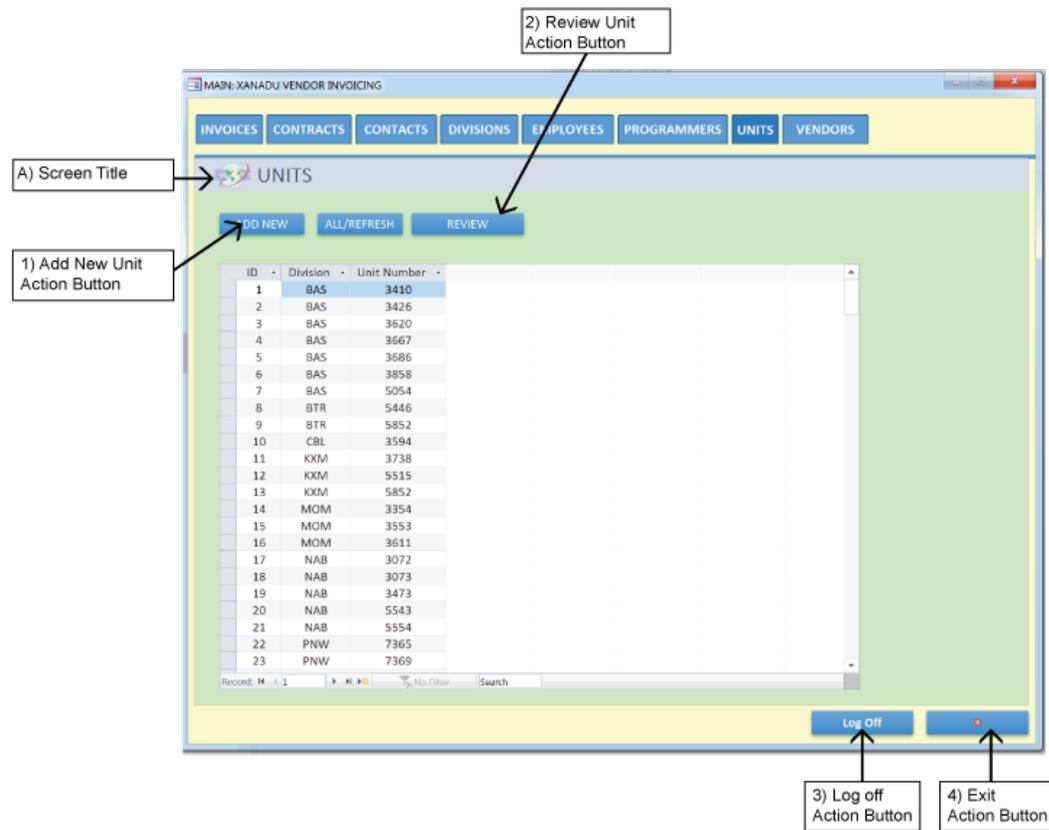
The screenshot shows a software window titled "PROGRAMMERS DETAILED VIEW" with a close button in the top right corner. The main content area is titled "PROGRAMMERS: DETAILED VIEW (read only)". Below the title, there are four input fields:

- ID: 1
- Vendor: GE Consulting Consortium (dropdown menu)
- First Name: Julio
- Last Name: Alvarado

At the bottom left of the form area is a logo featuring a globe with a red orbital path and a city skyline. At the bottom of the window is a navigation bar with the following elements from left to right: "Record:", a left arrow, a text box containing "1", a right arrow, a double right arrow, a refresh icon, a filter icon, the text "No Filter", a left arrow, a text box, a right arrow, and a double right arrow.

- 1) Preview the records of the programmer.

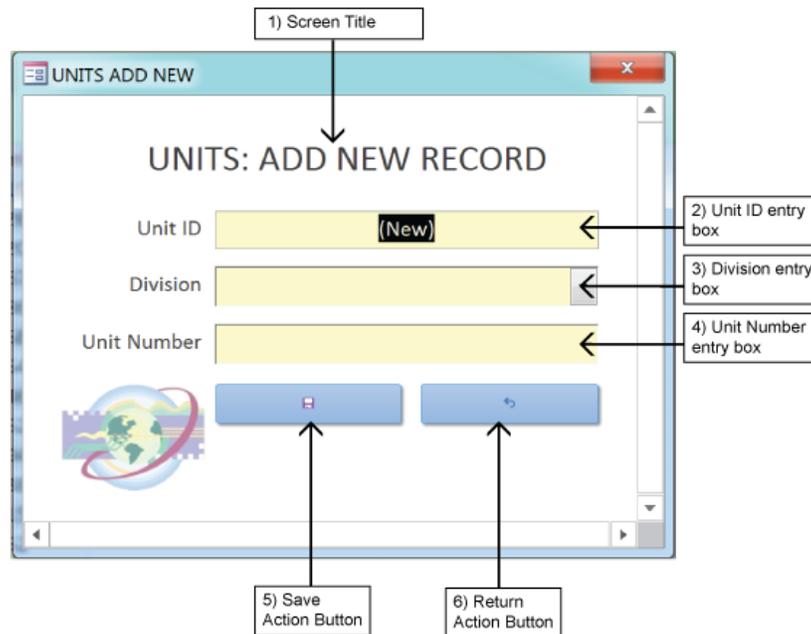
Unit Main Screen



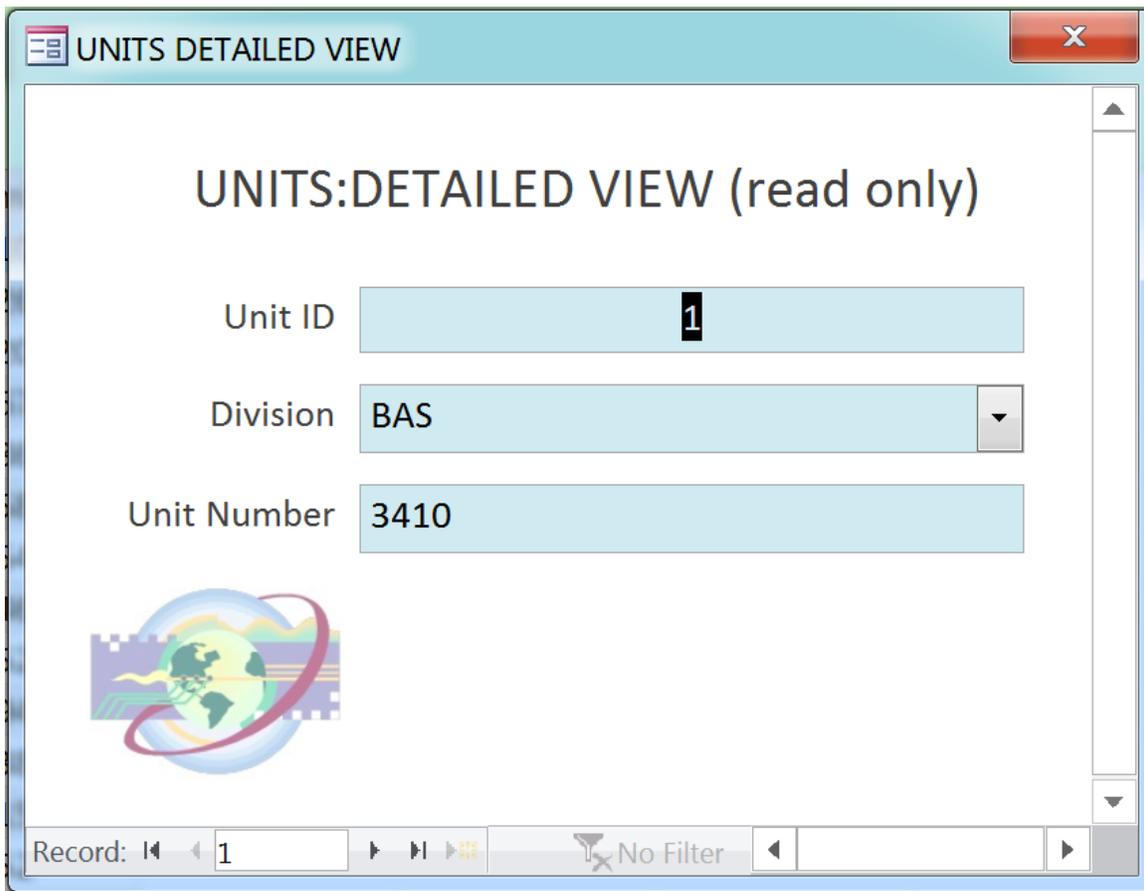
A) **Screen Title** is on top of the current screen

- 1) Click on the **Add New** Unit button to access a new unit popup screen
- 2) Click on the **Review** Unit button to access a preview unit screen
- 3) Click on the **Log Off** to sign out
- 4) Click on the **Exit** button to close the application

Add New Unit Screen



- 1) **Screen Title** is on top of the current screen
- 2) Enter the **Unit ID** in the entry box
- 3) Select from the drop-down list the appropriate **Division**
- 4) Enter the **Unit Number** in the entry box
- 5) Click the **Save** button to save the unit entry
- 6) Click the **Go Back** button to return to the unit main screen

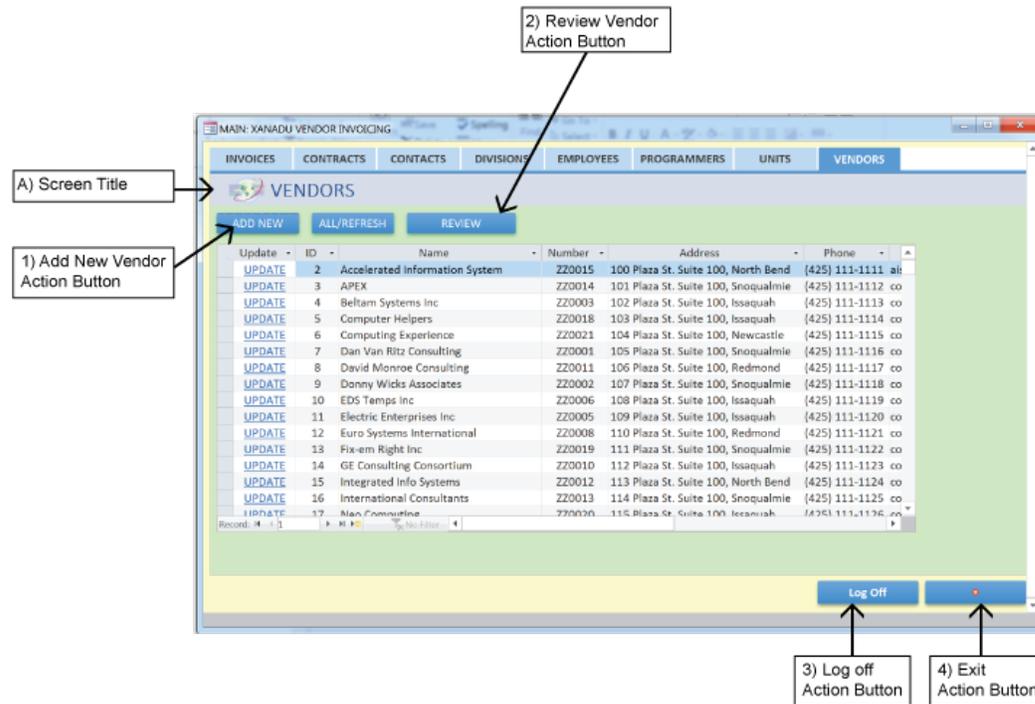
Review Unit Screen

The screenshot shows a software window titled "UNITS DETAILED VIEW" with a close button (X) in the top right corner. The main content area displays "UNITS:DETAILED VIEW (read only)" in large text. Below this, there are three input fields: "Unit ID" with the value "1", "Division" with a dropdown menu showing "BAS", and "Unit Number" with the value "3410". To the left of these fields is a graphic of a globe with a red orbital ring. At the bottom, there is a navigation bar with "Record: 1" and a "No Filter" button.

Unit ID	1
Division	BAS
Unit Number	3410

1) Preview the records of the unit.

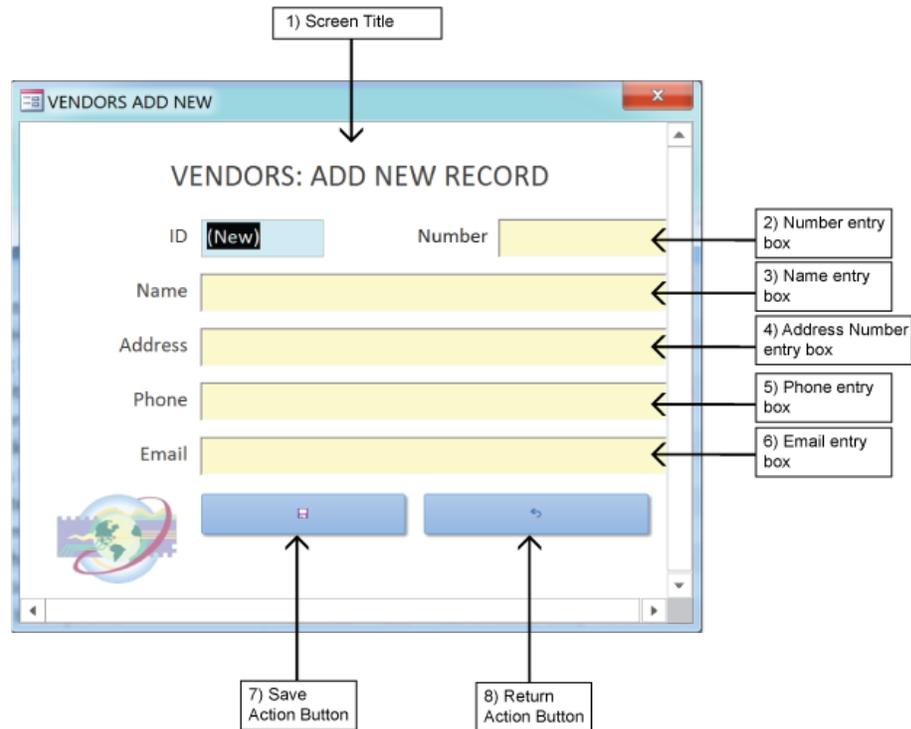
Vendor Main Screen



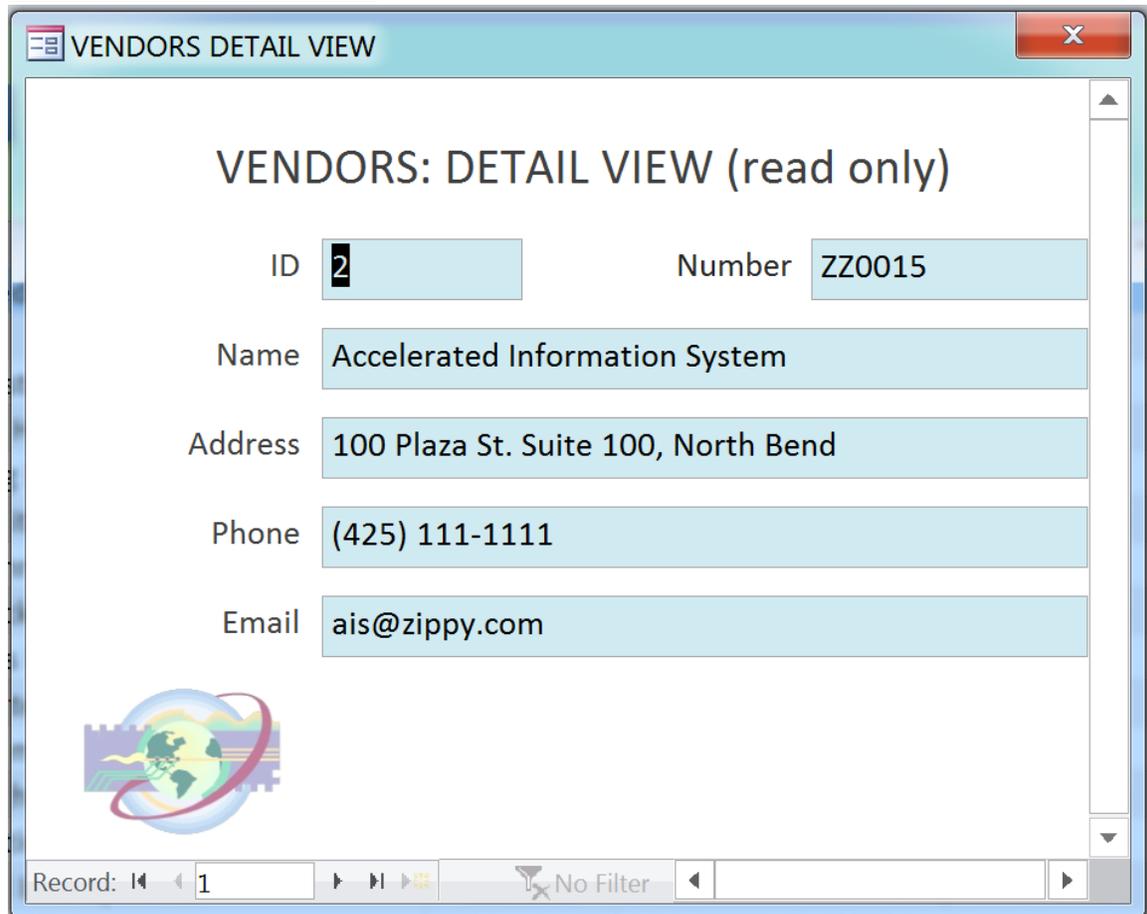
A) **Screen Title** is on top of the current screen

- 1) Click on the **Add New** Vendor button to access a new vendor popup screen
- 2) Click on the **Review** Vendor button to access a preview vendor screen
- 3) Click on the **Log Off** to sign out
- 4) Click on the **Exit** button to close the application

Add New Vendor Screen



- 1) **Screen Title** is on top of the current screen
- 2) Enter the **Number** in the entry box
- 3) Enter the **Name** in the entry box
- 4) Enter the **Address** in the entry box
- 5) Enter the **Phone Number** in the entry box
- 6) Enter the **Email Address** in the entry box
- 7) Click the **Save** button to save the vendor entry
- 8) Click the **Go Back** button to return to the vendor main screen

Review Vendor Screen

VENDORS DETAIL VIEW

VENDORS: DETAIL VIEW (read only)

ID Number

Name

Address

Phone

Email



Record: No Filter

1) Preview the records of the vendor.

Update Vendor Screen

The screenshot shows a software window titled "VENDORS UPDATE" with a sub-header "VENDORS: UPDATE RECORD". The form contains the following fields and buttons:

- ID: 2
- Number: ZZ0015
- Name: Accelerated Information System
- Address: 100 Plaza St. Suite 100, North Bend (highlighted in yellow)
- Phone: (425) 111-1111 (highlighted in yellow)
- Email: ais@zippy.com (highlighted in yellow)
- 4) Save Action Button (blue button with a floppy disk icon)
- 5) Return Action Button (blue button with a left-pointing arrow icon)

Callouts on the right side of the screen point to the Address, Phone, and Email entry boxes. Callouts at the bottom point to the Save and Return action buttons.

-
- 1) Enter the Updated **Address** in the entry box
 - 2) Enter the Updated **Phone Number** in the entry box
 - 3) Enter the Updated **Email Address** in the entry box
 - 4) Click the **Save** button to save the vendor entry
 - 5) Click the **Go Back** button to return to the vendor main screen

6 REPORTS

Creating Invoice & Contracts

1. Click Add new
2. A form will pop-up. Enter the information the form requests to create invoice/contracts.
3. Save form by clicking on the save button.

Reviewing Invoice & Contracts

1. Find the information you are looking for.
2. Click on pending, approved, accrual needed or paid to check the reports for each section
3. Select the report needed and view the information

7 WHERE TO GO FOR MORE HELP

It is possible that the system will run into errors. In these events, it is possible that the IT department of Bank of Xanadu will not be able to solve the problem. In cases of emergency, please contact Duane Samuels, the developer of the program who should be able to answer and fix the problems you are experiencing. In the case that Mr. Samuels is missing, contact Joshua Chong, the system tester. If you need to leave a message, call the main office number provided below.

Duane Samuels: duanegsam@scrappers.com

Joshua Chong: joshuamcho@scrappers.com

Phone: 1-425-123-4567

8 APPENDICES

Sample Reports:

	A	B	C	D	F	G	H	I	J	K	L	M	N	O
1	Invoices													
2														
3	ID Number	Programmer	Vendor	Charge	Invoice #	Date Paid	Begin Date	End Date	Rate	Total Hours	Total Invoice	Accrued	Memo	
4	Wilki0515	Wilkins, Peter	Donny Wicks Associates	9408	1001	12/21/14	12/01/14	12/15/14	59.00	64.0	3,776.00			
5										Total:	3,776.00			
6										Total for December:	3,776.00			
7														
8	Peckh0915	Peckham, Art	Donny Wicks Associates	9408	329	01/11/15	12/16/14	12/31/14	60.00	60.0	3,600.00	12/14		
9	Wilki0515	Wilkins, Peter	Donny Wicks Associates	9408	1002	01/11/15	12/16/14	12/31/14	59.00	66.0	3,894.00	12/14		
10	Brown0315	Brown, Lou	EDS Temps Inc	3072	509	01/11/15	12/17/14	12/31/14	25.00	70.0	1,750.00	12/14	Dec Exp	
11										Total:	9,244.00		13,020.00	
12														
13	Wilki0515	Wilkins, Peter	Donny Wicks Associates	9408	1003	01/25/15	01/02/15	01/15/15	59.00	85.0	5,015.00			
14	Brown0315	Brown, Lou	EDS Temps Inc	3072	510	01/25/15	01/02/15	01/15/15	25.00	68.0	1,700.00			
15										Total:	6,715.00			
16										Total for January:	15,959.00			
17														
18	Lehre1215	Lehrer, Philip	Beltam Systems Inc	3117	101	02/08/15	01/02/15	01/31/15	52.00	165.0	8,580.00	01/15		
19	Peckh0915	Peckham, Art	Donny Wicks Associates	9408	330	02/08/15	01/02/15	01/31/15	60.00	177.0	10,620.00	01/15		
20	Wilki0515	Wilkins, Peter	Donny Wicks Associates	9408	1004	02/08/15	01/16/15	01/31/15	59.00	82.0	4,838.00	01/15		
21	Brown0315	Brown, Lou	EDS Temps Inc	3072	511	02/08/15	01/16/15	01/31/15	25.00	70.0	1,750.00	01/15		
22	forti0615	Fortier, Brian	EDS Temps Inc	3072	3723	02/08/15	01/02/15	01/31/15	25.00	176.5	4,412.50	01/15	Jan Exp	
23										Total:	30,200.50		36,915.50	
24														
25	Brown0315	Brown, Lou	EDS Temps Inc	3072	512	02/22/15	02/01/15	02/15/15	25.00	68.0	1,700.00			
26	(start)													
27										Total:	1,700.00			
28										Total for February:	31,900.50			
29														
30														
31										Grand Total:	51,635.50			

	B	C	D	F	L	M	O
1	Accruals						
2							
3	Programmer	Vendor	Charge	Invoice #	Total Invoice	Accrued	Reversed
4	Brown, Lou	EDS Temps Inc	3072	509	1,750.00	12/14	
5			3072	Total:	1,750.00		01/10/15
6	Peckham, Art	Donny Wicks Associates	9408	329	3,600.00	12/14	
7	Wilkins, Peter	Donny Wicks Associates	9408	1002	3,894.00	12/14	
8			9408	Total:	7,494.00		01/10/15
9			December 2014		9,244.00		
10							
11	Brown, Lou	EDS Temps Inc	3072	511	1,750.00	01/15	
12	Fortier, Brian	EDS Temps Inc	3072	3723	4,412.50	01/15	
13			3072	Total:	6,162.50		02/10/15
14	Lehrer, Philip	Beltam Systems Inc	3117	101	8,580.00	01/15	
15			3117	Total:	8,580.00		02/10/15
16	Peckham, Art	Donny Wicks Associates	9408	330	10,620.00	01/15	
17	Wilkins, Peter	Donny Wicks Associates	9408	1004	4,838.00	01/15	
18			9408	Total:	15,458.00		02/10/15
19			January 2015		30,200.50		
20							
21							
22			February 2015				
23							
24			Grand Total:		39,444.50		
25							
26							

	B	C	D	E	F	H	I	K	L	M
2	By Division and Unit									
3	January 2015									
4										
5	Programmer	Vendor	Division	Charge	Invoice #	Begin Date	End Date	Total Hours	Total Invoice	Accrued
6			Division: AMB					Total for Division:	8,580.00	
7	Lehrer, Philip	Beltam Systems Inc	AMB	3117	101	01/02/15	01/31/15	165.0	8,580.00	01/15
8				3117				Total for Charge Unit:	8,580.00	
9										
10			Division: CCR					Total for Division:	20,473.00	
11	Peckham, Art	Donny Wicks Associates	CCR	9408	330	01/02/15	01/31/15	177.0	10,620.00	01/15
12	Wilkins, Peter	Donny Wicks Associates	CCR	9408	1003	01/02/15	01/15/15	85.0	5,015.00	
13	Wilkins, Peter	Donny Wicks Associates	CCR	9408	1004	01/16/15	01/31/15	82.0	4,838.00	01/15
14				9408				Total for Charge Unit:	20,473.00	
15										
16			Division: NAB					Total for Division:	7,862.50	
17	Brown, Lou	EDS Temps Inc	NAB	3072	510	01/02/15	01/15/15	68.0	1,700.00	
18	Brown, Lou	EDS Temps Inc	NAB	3072	511	01/16/15	01/31/15	70.0	1,750.00	01/15
19	Fortier, Brian	EDS Temps Inc	NAB	3072	3723	01/02/15	01/31/15	176.5	4,412.50	01/15
20				3072				Total for Charge Unit:	7,862.50	
21										
22								Grand Total for January:	36,915.50	
23										
24	Contract Programmers Monthly Expense Recap Report									
25	By Division and Unit									
26	February 2015									

	A	B	C	D	E	F	G	H	I	J	K
1	Contract Programmer Report										
2	Fee Maximum vs. Actuals										
3	December 2014										
4											
5	Programmer	Begin Date	End Date	\$/Hour	Contact Person	Phone	Appendix A Fee Max	Total Charged to Appendix A	Percent Used	Date Unit Last Charged	Under/Over Appendix A Max
6											
7	DIVISION: NAB										
8	Unit Number: 3072										
9											
10	Brown, Lou	12/17/14	06/17/15	25.00	Clark, Rudy	622-2375	29,000.00	1,750.00	6%	01/11/15	27,250.00
11											
12	DIVISION: CCR										
13	Unit Number: 9408										
14											
15	Wilkins, Peter	12/01/14	05/30/15	59.00	Scott, Randy	622-6047	48,000.00	7,670.00	16%	01/11/15	40,330.00
16	Peckham, Art	12/16/14	09/30/15	60.00	Scott, Randy	622-6047	88,600.00	3,600.00	4%	01/11/15	85,000.00

Evaluation Criteria – User Manual Draft

Criteria	Points	Given	Comments
<ul style="list-style-type: none"> ❑ Completed and submitted on or before the due date 	5		
<ul style="list-style-type: none"> ❑ User Manual contains content specified in assignment <ul style="list-style-type: none"> ○ Cover Memo ○ Title Page ○ Table of Contents ❑ Body of document discusses/includes the following: <ul style="list-style-type: none"> ○ Introduction ○ System Overview ○ Step-by-step Procedures ○ Navigation ○ Entry Screens ○ Reports ○ Where to go for more help ○ Appendices 	25		
<ul style="list-style-type: none"> ❑ User Manual demonstrates an understanding of the role and importance of a help system, as well as the various methods by which one can be implemented 	10		
<ul style="list-style-type: none"> ❑ Written in appropriate style for the audience ❑ Document organization and content demonstrate careful planning and thought and is easy to follow ❑ Well-written, free of spelling and grammar errors 	10		
Total	50		