Bank of Xanadu –
System Design

College Name: Edmonds Community College

Department: Computer Information Systems

Assignment Name: TA2 (Team Assignment #2)

Date Prepared: April 27, 2013

Team Name: Team Awesome

Team Members: Jon Raymond Melchizedek Day, Lindsy Carter, Brendan Jones, Joanna Tran, Michelle Lawson, Kemseng Sath

DATE: April 27, 2013
TO: Mr. Patrick Jay
Vice President & Manager
FROM: Team Awesome
Lindsy Carter, Jon Day, Joanna Tran,
Kemseng Sath, Michelle Lawson, Brendan Jones
SUBJECT: Contract Payment System – **System Design**

Attached is the System Design document for the new Automated Contractual Payment System created by Team Awesome for Bank of Xanadu. We would like to meet with you at Bank of Xanadu – Bellevue in Room 124 on Saturday, April 27, 2013 to discuss this System Design document with you.

Team Awesome looks forward to this meeting. If you have any questions please feel free to contact us.

Regards,

Lindsy Carter
Jon Day
Joanna Tran
Kemseng Sath
Michelle Lawson
Brendan Jones

BANK OF XANADU SYSTEM DESIGN

April 27, 2013

Presented by: Team Awesome

Lindsy Carter
Jon Day
Joanna Tran
Kemseng Sath
Brendan Jones
Michelle Lawson

Table of Contents

Introduction

The purpose of this document is to describe the design phase of the Bank of Xanadu's new automated contract payment system. To start, we will explain the architecture and design considerations as well as user navigation and design. We will also describe the procedures and coding standards in detail.

Architecture and Design Considerations

Architecture

Team Awesome has decided to use a Web-centric architecture to allow for the scalability of the system into Xanadus many branches across the world. This architecture will also be client/server to allow the thin client interface to be done using html and the database server to be controlled using php to do the majority of the work. The system will be a combination of online processing for accountants to work with what they need as they get it and batch processing for managers to print their monthly reports. A star network topology with a router will be used to connect the employees of one office with all the other offices around the world.

Assumptions

We are assuming that once the new system is tested we can easily implement in other branches. We also assume that the maintenance and security after implementation will be handled by the branch IT departments with training provided by us as well as on-going support. By using the online processing, the new system can easily accommodate more users and the distribution of data will remain the same quality.

A 'train the trainers' program will be instrumental in setting up later branches and the Bellevue branch will ahead of the curve. There should be one person hired internally to handle the functions and maintenance of the system.

Information/Data Model

The ERD (Appendix – 1) shows all the tables that will be built in the database along with the relationships between all the entities to ensure the system is built upon good information and responds as required. The Metadata Dictionary (Appendix – 2) includes all of the entities along with their associated attributes, definitions, domain and their referential integrity. A brief summary of the entities and attributes for the database is listed below.

BUYERS (Buyer_ID, Buyer_F_Name, Buyer_L_Name, Buyer_Phone, Buyer_Email)

CONTACTS (Contact_ID, Contact_F_Name, Contact_L_Name, Contact_Phone, Contact_Email, *Unit_ID*, *Programmer_ID*)

CONTRACTS (Contract_ID, Cont_Charge_Unit, Contract_Code, Contract_Rate, Contract_Start_Date, Contract_End_Date, Contract_Fee_Maximum, Contract_Desc, *Vendor_ID*, *Unit_ID*, *Contact_ID*, *Buyer_ID*, *Programmer_ID*)

DIVISION (Division_ID, Division_Code, Division_Name)

INVOICES (Invoice_ID, Invoice_Number, Invoice_Date, Invoice_Start_Date, Invoice_End_Date, Invoice_Due_Date, Invoice_Hours, Invoice_Desc, Invoice_Rate, Invoice_Total, Inv_Timesheet_Hours, Invoice_Status, Date_To_A/P, Invoice_Accrual, *Vendor_ID*, *Employee_ID*, *Contract_ID*)

PROGRAMMERS (Programmer_ID, Prog_F_Name, Prog_L_Name, *Vendor_ID*)

SYSTEM_USERS (Employee_ID, Emp_F_Name, Emp_L_Name, Emp_Phone, Emp_Email, Emp_User_Name, Emp_User_Password, Emp_User_Permissions)

UNITS (Unit_ID, Unit_Number, Unit_Location, *Division_ID*)

VENDORS (Vendor_ID, Vendor_Name, Vendor_Street, Vendor_City, Vendor_State, Vendor_Zip, Vendor_Phone, Vendor_Email, Vendor_Contact)

User Navigation Design (Storyboard)

User Navigation Design (Storyboard)

Team Awesome has been working hard to make sure the system is extremely user friendly. We have made a consistent interface, whether you are a manager or accountant, to ensure that the system is easy and has consistent help for all users. There are only 3 screens with navigation buttons to make sure everyone can do what he or she needs to do. A brief description of the navigation design can be seen below with a more information found in the Appendix.

Login Screen

This screen has a help box on the bottom with a login form in the middle of an iframe. A Xanadu with appropriate login privileges can login using their user name and password. This will then take the user to the appropriate screen for their job duties.

Accountant Screen

This screen has a help box on the top to describe all the navigation buttons and a help box on the bottom to describe all the parts of the form. The navigation main and sub buttons take the accountant to the appropriate forms for "New Contact", "New Division", "New Unit", "New Vendor", "Receive" (Contract), "Exception" (Contract), "Update" (Contract), "Receive" (Invoice), "Exception" (Invoice), "Update" (Invoice), "Inquiries", "Payment", and "Accruals".

Manager Screen

This screen has a help box on the top to describe all the navigation buttons and a help box on the bottom to describe all the parts of the form. The navigation buttons take the manager to the appropriate forms for "G/L Expense", "Accruals", "Prog Expense", "Fee Max", and "Contract Recap".

Inputs

Below are the input documents that will be required to produce the correct information from the new system:

Document	Description
Contract	At a minimum, a valid contract will contain the programmer name, hourly rate, fee maximum, and start and end dates.
Invoice	Invoice is sent from the programmer and will have all of the hours worked (with time sheet as back up) and hourly rate and an invoice total.
Time sheet	A time sheet will always be attached to its associated invoice as back up of hours worked in a particular period.

Outputs

Below are the output documents that are required to be accurate when the system functions properly:

Document	Description
Exception Memo	This memo is produced by the accountant to send back to the Buyer for any changes a contract or invoice needs
Data Entry Sheet	This sheet is produced by the Accountant once an invoice/timesheet is approved to instruct the accounts payable department on what to enter for payment
Invoice Report	Report that shows invoices paid to the programmer against the expense account. The accountant to balance each account on a regular monthly cycle also uses this.
Accrual Report	Report that shows all accrued invoices for so they can be processed in the current month and reversed in the next.
Expense Recap	This "Contract Programmers Monthly Expense Recap Report" shows an overview of the monthly expenses for each programmer. It is sorted by charge unit and

	sent to each bank division.
Contact Programmer Report	This report shows the actual amounts paid to the programmer against the contract fee maximum. It is sorted by charge unit and sent to each bank division.
Contract Recap	A report for each project manager to keep track of the programmers working for them and the contract information.

Procedures

A brief description of (1) how the users will use the new system and (2) a brief description of any code that will be used.

1. External Procedures: Buttons, links and other required actions will be designed to be intuitive, to maximize ease of use and efficiency.

At the Welcome screen, the user will enter their username and password. Once they are successfully authenticated and logged in, they will be directed to a screen with a menu of task choices.

Accountants will be taken to a page with options for "Bank Info", "Contracts", "Invoices", "Inquiries", "Payments", and "Accruals". When "Bank Info" is highlighted the accountant will be able to select additional options to create "New Vendor", "New Contact", "New Unit", and "New Division". When "Contracts" is highlighted the accountant will be able to select additional options for contracts including "Receive", "Exception", "Update". When "Invoices" is highlighted the accountant will be able to select additional options for invoices including "Receive", "Exception", "Update". When an option with no further suboptions is selected the accountant will be directed to the appropriate page to handle what is needed to be done.

Managers will be taken to a page with options for "G/L Expense", "Accruals", "Prog Expense", "Fee Max", and "Contract Recap". When an option is selected the manager will be directed to the appropriate page to generate the appropriate report.

Detailed information about each of these screens and their use, as well as Use Case Scenarios and diagrams, are contained in the Appendix. The Use Cases are:

UC001: "RECEIVE CONTRACT" Use Case Scenario
UC002: "ADD NEW BANK INFORMATION" Use Case Scenario
UC003: "CONTRACT EXCEPTION" Use Case Scenario
UC004: "UPDATE CONTRACT" Use Case Scenario
UC005: "RECEIVE INVOICE" Use Case Scenario
UC006: "INVOICE EXCEPTION" Use Case Scenario
UC007: "UPDATE INVOICE" Use Case Scenario
UC008: "INVOICE STATUS INQUIRY" Use Case Scenario
UC009: "PAY INVOICE" Use Case Scenario
UC010: "ACCRUE INVOICE" Use Case Scenario
UC011: " ACCOUNTING REPORTS" Use Case Scenario
UC012: "RUN MANAGEMENT REPORTS" Use Case Scenario

2. Internal Procedures: Some brief descriptions of some identified queries are below. More information on Internal Procedures can be found in the Appendix.

Enter Contact

- Description – This query enters a new contact into the database
- User – Accountant
- Tables - Contact

Enter Division

- Description – This query enters a new division into the database
- User – Accountant
- Tables - Division

Enter Unit

- Description – This query enters a new charge unit into the database
- User – Accountant
- Tables - Unit

Enter Vendor

- Description – This query enters a new vendor into the database
- User – Accountant
- Tables – Vendor

Enter Received Contract

- Description – This query enters a new contract into the database
- User – Accountant
- Tables - Contract

Enter Contract Exception

- Description – This query updates a contract in the database
- User – Accountant
- Tables - Contract

Enter Contract Update

- Description – This query updates a contact in the database
- User – Accountant
- Tables - Contract

Enter Received Invoice

- Description – This query enters a new invoice into the database
- User – Accountant
- Tables - Invoice

Enter Invoice Exception

- Description – This query updates an invoice in the database
- User – Accountant
- Tables - Invoice

Enter Invoice Update

- Description – This query updates an invoice into the database
- User – Accountant
- Tables - Invoice

Enter Invoice Inquiry

- Description – This query updates an invoice in the database
- User – Accountant
- Tables - Invoice

Enter Invoice Payment

- Description – This query updates an invoice in the database

- User – Accountant
- Tables - Invoice

Enter Accrual

- Description – This query updates an invoice in the database
- User – Accountant
- Tables - Invoice

Print Data Entry Sheet

- Description – This query prints out a data entry sheet for a/p
- User – Accountant
- Tables - Invoice

Print Exception Memo

- Description – This query prints out an exception memo for buyer
- User – Accountant
- Tables – Contract, Invoice

Print Invoice Inquiry Status

- Description – This query prints out a status email for buyer
- User – Accountant
- Tables - Invoice

Print G/L Report

- Description – This query prints out the General Ledger Expense report
- User – Manager
- Tables – Contract, Invoice, Programmer, Unit, Vendor

Print Accrual Report

- Description – This query prints out the Accrual report
- User – Manager
- Tables – Invoice, Programmer, Unit, Vender

Print Prog Expense Report

- Description – This query prints out the Contract Programmers Monthly Expense report
- User – Manager

- Tables – Division, Invoice, Programmer, Unit, Vender

Print Fee Max Report

- Description – This query prints out the Contract Programmer Report Fee Max Vs. Actual report
- User – Manager
- Tables – Contact, Contract, Division, Invoice, Programmer, Unit, Vender

Print Contract Recap Report

- Description – This query prints out the Monthly Contract Expense report
- User – Manager
- Tables –Contact, Contract, Invoice, Programmer, Unit, Vender

Interface Design and Coding Standards

External (screens & navigation): The default font for the site will be 14 point Verdana. General text will be centered. Form label text will be aligned to the left. There will be a help box at the top left of all screens to give more detailed information about the menus and submenus to aid the users in using the system. Navigation buttons will be blue and under the top help box. Submenu buttons of the navigation will be orange. Underneath the navigation will be an iframe that will display the different pages with the different forms for those pages. Drop down choice lists will be used in the forms. If something is entered incorrectly or not entered then that field will have a red box around it when submitted. Each form will have a red reset button and a green submit button. At the bottom of each form will be a help box to give more detailed information about what is to be entered into the form and what the buttons for the forms do. There will be another column near the forms if viewing something from the system is needed. More detailed information about the Interface Design can be found in the Appendix.

Internal (code & database design): Code will be commented as needed to clarify what the code is doing for code that is a function, condition, or loop. Whitespace and carriage returns will be used to make code easier to read. Code inside of other code will be tabbed in. Variable names will be short descriptive words added together using camel case and without underscores. Functions names will be camel case. SQL statements will be all uppercase except the field names that will use uppercase first letters and

underscore between words. Table names will be plural short descriptive names. Fields names will include the table name they are from and shortened if the name is too long. More detailed information about the Coding Standards can be found in the Appendix.

Conclusion

Team Awesome has now determined the architecture for Xanadus new Automated Contractual Payment System to be Web-centric, (thin) client/(database) server. A star network topology with a router will be ready to connect all of Xanadu's branches together when needed. Trainers will be trained and Xanadu's current IT staff will take care of security and maintenance for the system after deployed. The system design is now complete and Team Awesome would like to get started on developing the system to handle the navigation, inputs, outputs, and the internal and external procedures as defined. Team Awesome will use standards as defined to develop the needed system. At this time Team Awesome would like to request a meeting with the main users of the system for a "design walkthrough" to get feedback and approval on an interface prototype. Please accept our request to meet at Bank of Xanadu – Bellevue Saturday, May 11, 2013 at 1pm.

Approval

__*Patrick Jay*, V.P. & Manager_____

_____

Manager Accountant

_____ _____

Manager Accountant

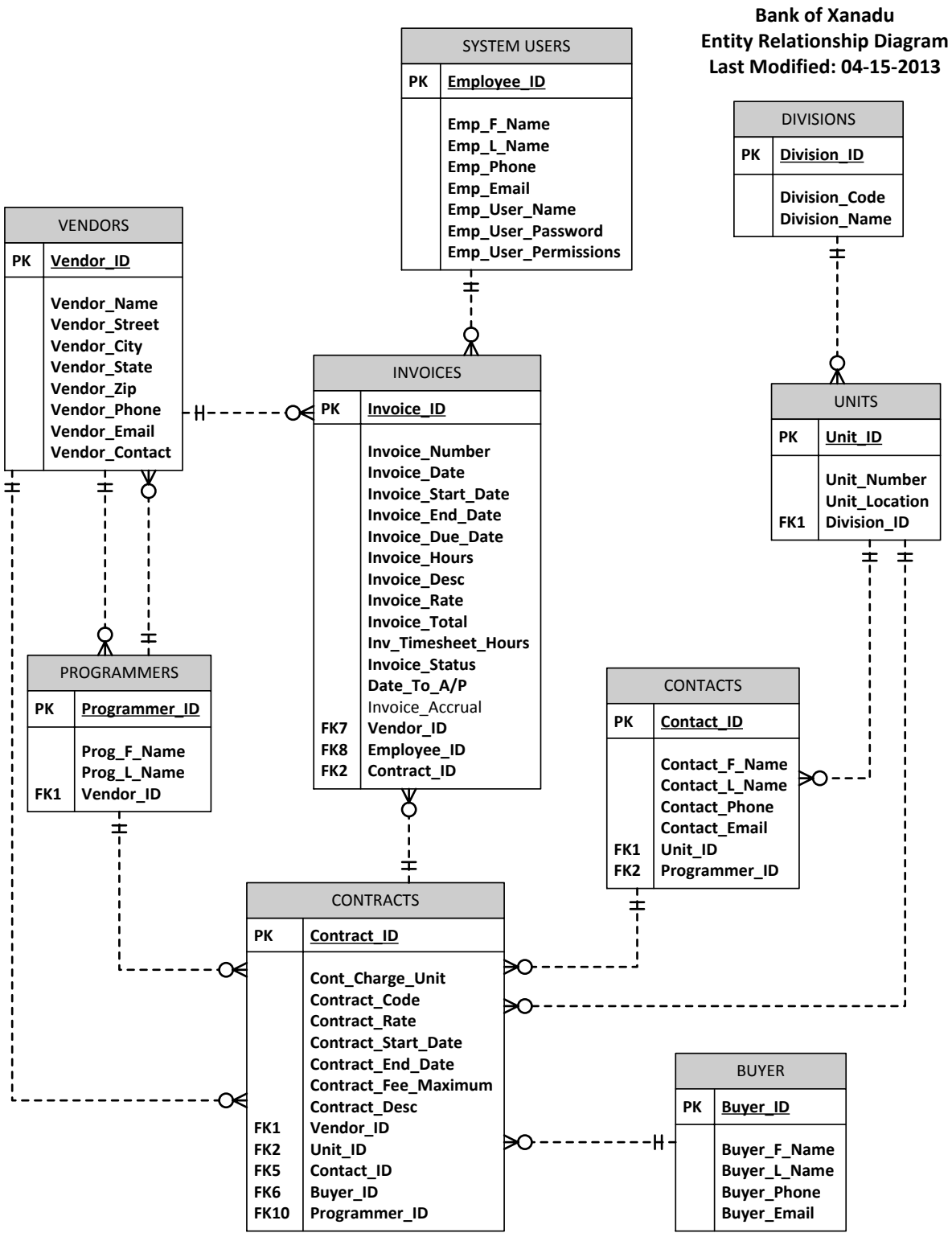
_____ _____

Manager Accountant

BANK OF XANADU
APPENDIX

Table of Contents

1. Entity Relationship Diagram (ERD)



2. MetaData Dictionary

Entity	Attribute	Definition	Domain	Referenti Integrity
BUYERS	Buyer_ID	A unique identifier for each buyer	System assigned Unique numeric(10)	PK Primary K
	Buyer_F_Name	First name of the buyer	Required. Non unique char(25)	
	Buyer_L_Name	Last name of the buyer	Required. Non unique char(25)	
	Buyer_Phone	Contact phone number for the buyer	Require. Non unique char (20) Input Mask: phone (###)###-####	
	Buyer_Email	Email address for the buyer	Required. Non unique char(40)	
CONTACTS	Contact_ID	A unique identifier for project manager	System assigned Unique numeric (10)	PK Primary K
	Contact_F_Name	First name of project manager	Required. Non unique char(25)	
	Contact_L_Name	Last name of project manager	Required. Non unique char(25)	
	Contact_Phone	Project manager phone number	Require. Non unique. Input Mask char (20) phone (###)###-####	
	Contact_Email	Project manager email	Required. Non unique char(40)	

		address		
	Unit_ID	A unique numbers use for identifying units in CONTRACTS	Required. Non unique FK UNITS Numeric(10)	A Unit_ID ca exist without related UNIT recorded
CONTRACTS	Contract_ID	A unique identifier for each contract	System assigned Unique numeric (10)	PK Primary K
	Cont_Charge	A unique identifier charge unit to the bank	Required. Non unique numeric (10)	
	Contract_Code	Assigned by the accounting department	Required. Non unique mix char and numeric(25)	
	Contract_Rate	Hourly rate for that contract	Required. Non unique numeric (25)	
	Cont_Start_Date	The date of contract begin	Require. Non unique. Input Mask: date mm/dd/yyyy	
	Cont_End_Date	The date of the contract end	Require. Non unique. Input Mask: date mm/dd/yyyy	
	Cont_Fee_Max	The maximum fee can be charge for that contract	Required. Non unique numeric (25)	
	Contract_Desc	The description of the contract	Required. Non unique. Text (100)	
	Vendor_ID	A unique numbers use for	Required. Unique FK VENDORS	A Vendor_ID can't exist without a

		identifying vendor in CONTRACTS	Numeric(10)	related VENDORS record
	Unit_ID	A unique numbers use for identifying unit in CONTRACTS	Required. Non unique FK UNITS Numeric(10)	A Unit_ID ca exist without related UNIT record
	Contact_ID	A unique numbers use for identifying contact in CONTRACTS	Required. Unique FK CONTACTS Numeric(10)	A Contact_I can't exist without a related CONTACTS record
	Programmer_ID	A unique numbers use for identifying programmer in CONTRACTS	Required. Unique FK PROGRAMMERS Numeric(10)	A Programmer_ can't exist without a related PROGRAMME record
	Buyer_ID	A unique numbers use for identifying buyer in CONTRACTS	Required. Unique FK BUYERS Numeric(10)	A Buyer_ID can't exist without a related BUYE record
DIVISIONS	Division_ID	A unique identifier for each Division	System assigned Unique numeric(10)	PK Primary K
	Division_Code	A unique Identifier for the Bank Division	Required. Non unique numeric(10)	
	Division_Name	Identifier the Bank Division name	Required. Non unique char(30)	
INVOICES	Invoice_ID	A unique identifier for	System assigned	PK Primary K

		each invoice	Unique numeric(10)
	Invoice_Number	The number assigned to an invoice	Required. Unique numeric (10)
	Invoice_Date	The date the invoice is generated	Require. Unique. Input Mask: date mm/dd/yyyy
	Invoice_Start_Date	The start of the dates worked that the invoice covers	Require. Non unique. Input Mask: date mm/dd/yyyy
	Invoice_End_Date	The last date the invoice covers	Require. Non unique. Input Mask: date mm/dd/yyyy
	Invoice_Due_Date	The date the invoice is due to be paid	Require. Non unique. Input Mask: date mm/dd/yyyy
	Invoice_Hours	A total hours for programer	Required. Non unique numeric(10)
	Invoice_Description	The description of the work performed on this billing cycle	Required. Non unique char(100)
	Invoice_Rate	Pay rate of pay for the programmer	Required. Non unique numeric(10)
	Invoice_Total	The total amount of hours times the hourly rate.	Required. Non unique numeric(10)
	Inv_Timesheet_Hours	The breakdown	Required. Non unique

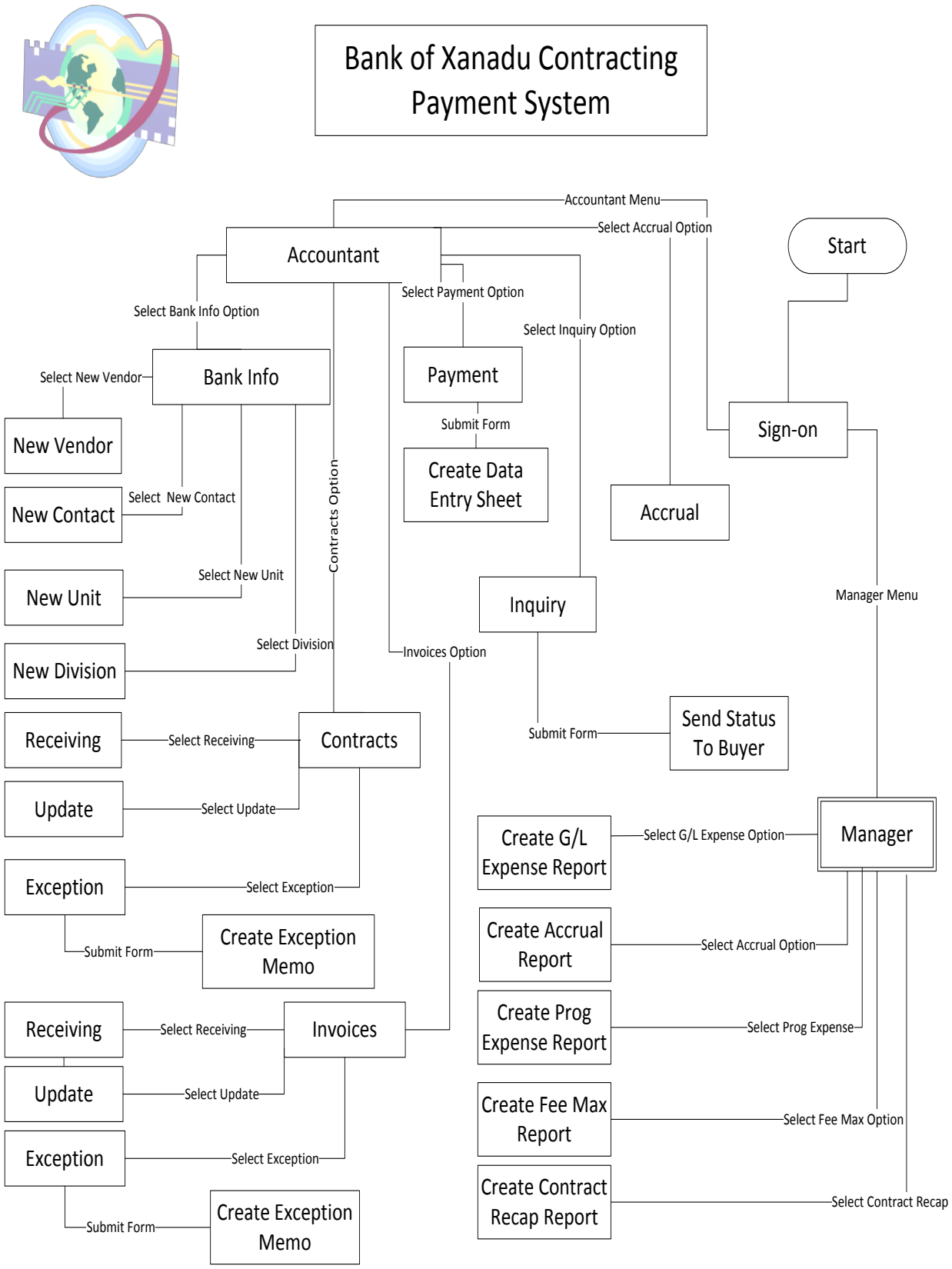
		of the hours spent working by a programmer in a billing cycle	numeric(10)	
	Invoice_Status	Payment status of an invoice. It could be paid or unpaid	Required. Non unique char(30)	
	Date_To_A/P	The date the invoice is sent to A/P	Require. Non unique. Input Mask: date mm/dd/yyyy	
	Invoice_Accrual	The amount accrued for the invoice if it is not paid immediately	Optional Non unique numeric (20)	
	Vendor_ID	A unique numbers use for identifying vendor	Required. Unique FK VENDORS Numeric(10)	A vendor_ID can't exist without a related VENDORS record
	Employee_ID	A unique numbers use for identifying Employee in INVOICES	Required. Unique FK SYSTEM USERS Numeric(10)	A Employee_ can't exist without a related SYST USERS recor
	Contract_ID	A unique numbers use for identifying contract in INVOICES	Required. Non unique FK CONTRACTS Numeric(10)	A Contract_I can't exist without a related CONTRACTS record
	Programmer_ID	A unique	Required. Non	A

		numbers use for identifying programmer in INVOICES	unique FK PROGRAMMERS Numeric(10)	Programmer_ can't exist without a related PROGRAMME record
PROGRAMMERS	Programmer_ID	A unique identifier the programmer	System assigned. Unique numeric(10)	PK Primary K
	Prog_F_Name	First name of the programmer	Required. Non unique char(25)	
	Prog_L_Name	Last name of the programmer	Required. Non unique char(25)	
	Vendor_ID	A unique numbers use for identifying vendor in INVOICES	Required. Unique FK VENDORS Numeric(10)	A vendor_ID can't exist without a related VENDORS record
SYSTEM USERS	Employee_ID	A unique identifier for system user	System assigned. Unique numeric(10)	PK Primary K
	Emp_F_Name	First name of the system user	Required. Non unique char(25)	
	Emp_L_Name	Last name of the system user	Required. Non unique char(25)	
	Emp_Phone	Contact phone number of the system user	Required. Non unique. Char (20) Input Mask: phone (###)###-####	
	Emp_Email	Email address of the system user	Required. Non unique char(40)	

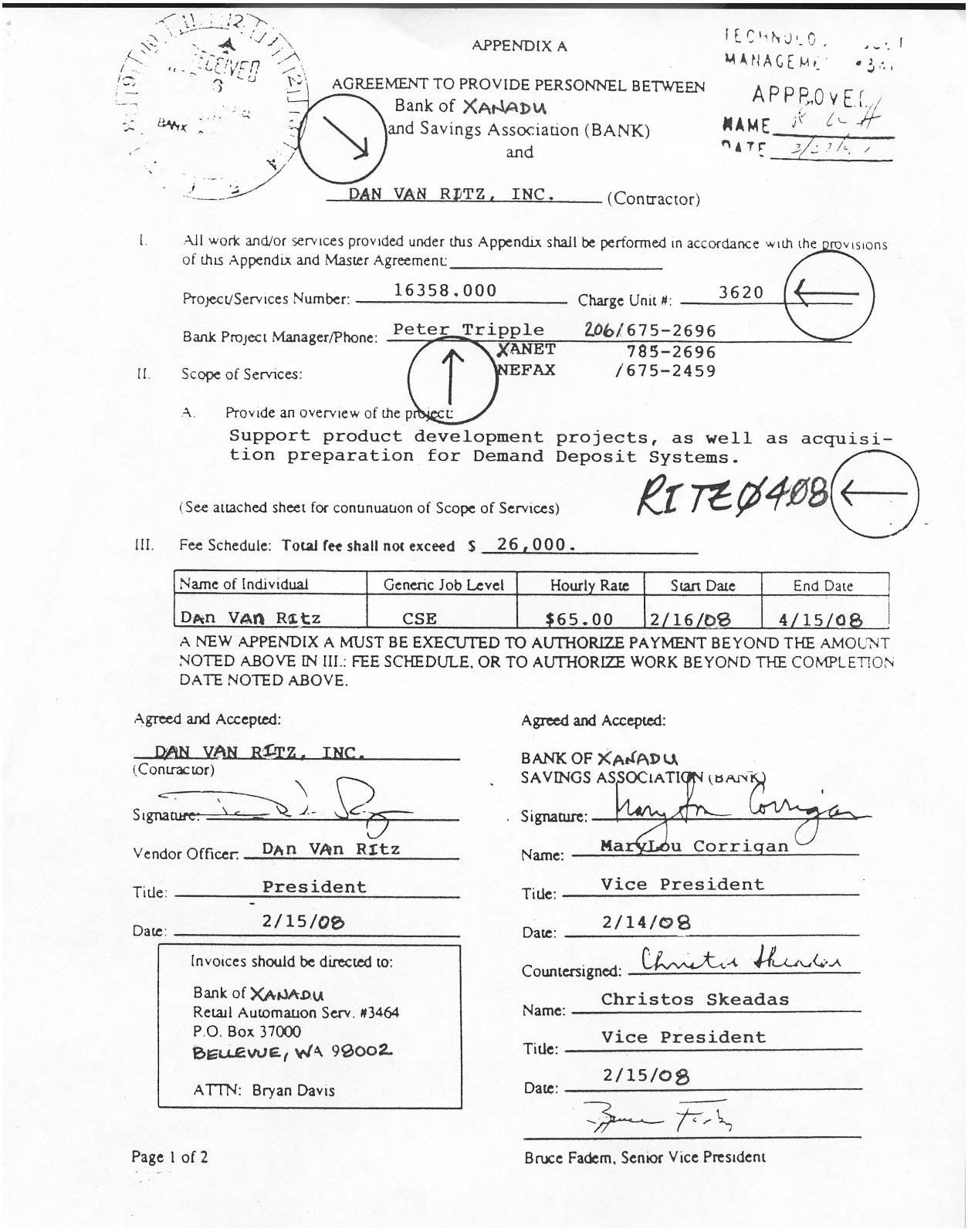
	Emp_User_Name	User ID for system user	Required. Unique cha(25)	
	Emp_Password	Password for system user	Required. Non unique mix char & numeric(25)	
	Emp_Permissions	Access permission for system user	Required. Non unique char(350)	
UNITS	Unit_ID	A unique identifier for each unit	System assigned unique numeric(10)	PK Primary K
	Unit_Number	A unique identifier for the bank cost center	Required. Non unique numeric(10)	
	Unit_Location	Location of the cost center	Required. Non unique char(35)	
	Division_ID	A unique number for identifying a division in UNITS	Required. Non unique FK DIVISIONS Numeric(10)	A Division_ID can't exist without a related DIVISIONS record
VENDORS	Vendor_ID	A unique identifier for each vendor	System assigned unique numeric(10)	PK Primary K
	Vendor_Name	Name of vendor	Required. Non unique char(30)	
	Vendor_Street	Primary address of the vendor	Required. Non unique numeric (35)	
	Vendor_City	City of the vendor address	Required. Non unique char(25)	
	Vendor_State	State of the vendor address	Required. Non unique char (4)Look up:	

			State=AL, AK, CA, etc.
	Vendor_Zip	Zip code of the vendor	Required. Non unique integer
	Vendor_Phone	Contact phone number for the vendor	Required. Non unique char (20) Input Mask: phone (###)###-####
	Vendor_Email	Email address of the vendor	Required. Non unique char(40)
	Vendor_Contact	Contact person name	Required. Non unique char(35)
	Vendor_Notes	Document or any information that related to the contract	Required. Non unique text

3. User Navigation Design (Storyboard)



4. Inputs



APPENDIX A

AGREEMENT TO PROVIDE PERSONNEL BETWEEN
Bank of XANADU
and Savings Association (BANK)
and
DAN VAN RITZ, INC. (Contractor)

TECHNOLO... MANAGEME...
APPROVED
NAME
DATE 3/27/...

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

Project/Services Number: 16358.000 Charge Unit #: 3620

Bank Project Manager/Phone: Peter Tripple 206/675-2696
XANET 785-2696
NEFAX /675-2459

II. Scope of Services:

A. Provide an overview of the project:

Support product development projects, as well as acquisition preparation for Demand Deposit Systems.

RITE0408

(See attached sheet for continuation of Scope of Services)

III. Fee Schedule: **Total fee shall not exceed** $ 26,000.

Name of Individual	Generic Job Level	Hourly Rate	Start Date	End Date
Dan Van Ritz	CSE	$65.00	2/16/08	4/15/08

A NEW APPENDIX A MUST BE EXECUTED TO AUTHORIZE PAYMENT 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:

Vendor Officer: Dan Van Ritz

Title: President

Date: 2/15/08

Invoices should be directed to:
Bank of XANADU
Retail Automation Serv. #3464
P.O. Box 37000
BELLEVUE, WA 98002
ATTN: Bryan Davis

Agreed and Accepted:

BANK OF XANADU
SAVINGS ASSOCIATION (BANK)

Signature:

Name: MaryLou Corrigan

Title: Vice President

Date: 2/14/08

Countersigned:

Name: Christos Skeadas

Title: Vice President

Date: 2/15/08

Bruce Fadem, Senior Vice President

Page 1 of 2

AGREEMENT TO PROVIDE PERSONNEL BETWEEN
Bank of XANADU
and Savings Association (BANK)
and

DAN VAN RITZ, INC. (Contractor)

II. Scope of Services - Continued:

B. List the specific tasks to be performed:

Complete systems design specifications.
Analyze and code in COBOL.
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.)

1. IBM 30XX, TSO/ISPF, OS JCL, VSAM.
2. Ability to analyze and code in COBOL.
3. Design, coding and testing skills.
4. Accounting systems background required, banking preferred. Deposit systems/prior acquisition experience a plus.
5. Prior BofA experience a plus.
6. Strong communications and documentation skills.
7. 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.

DAN VAN RITZ CONSULTING, INC.
5820 Stoneridge Mall Road Suite #
Pleasanton, WA 98506

08MAR 19 PM 1:24

INVOICE 100154

SALESPERSON	INVOICE DATE
DAn	3/18/08

TO
BANK OF XANADU
General Accounting #3707
P.O. Box 37000
BELLEVUE, WA 98002

INFORMATION
Master Agreement #90-3167
Project/Service #
Charge Unit #3620

ACCT#	DATE	PERIOD	TERMS	PURCHASE ORDER #
	3/18/08	3/1-3/15	Net 0	

HOURS	DESCRIPTION	UNIT PRICE	AMOUNT
88	Computer Consulting RT65	65.00	5720.00

RITE0408

RECEIVED
MAR 21 2008
Bank Gen. Acct.
#3707
BANK OF XANADU

APPROVED FOR PAYMENT
BY BSmith
UNIT # 3620

TOTAL 5720.00

Thank You

DAn VAn Ritz Consulting, Inc.

Contractor Time Sheet

Contractor Name: DAN VAN RITZ

Client Company: BANK OF XANADU

Period: From 3/1/08 To 3/15/08

Calender Days	Hours Worked	Calender Days	Hours Worked
1	8	16	
2		17	
3	8	18	
4	8	19	
5	8	20	
6	8	21	
7	8	22	
8		23	
9		24	
10	8	25	
11	8	26	
12	8	27	
13	8	28	
14	8	29	
15		30	
		31	

Total Hours: 88

Client Company Representative Acceptance: Carolyn Norris 3/19/08

Signature Date

Mail To: Dan Van Ritz Consulting, Inc.

MEMO TO: Rob Watt
TAM #3411

MEMO FROM: Del Billingsley
Vice President/Project Manager
Consumer Lending Division - Consumer Loan Services
Project Management & Technology Support #3454
Xnet 666-1464

COPY TO:

Mike DeVico	#3454
Jim Petersohn	#3761
Frank Smikoski	#3326
Kris Walunas	#3454

DATE: April 19, 2008

SUBJECT: Marathon Contract Extension - CPR PROJECT (#287)

The "Completion Date" on the Marathon Systems Consulting Service Agreement, Master Agreement #91-3664, has been extended to May 15, 2008 The Total Fees do not change; they will not exceed $77,000.

Please make note of this change in your files.

Thanks for your help and call me if any questions.

Edit Block 4/08
Kennedy 4/08
Ewing 4/08
Latoure 4/08

5. Outputs

DATA ENTRY SHEET

Vendor Name:	Donny Wicks Associates
Vendor Number:	ZZ0002
Invoice Number:	329
Description:	A. Peckham 12/16/07 to 12/31/07
Invoice Date:	01/02/08
Due Date:	01/17/08
Invoice Total:	3,600.00
G/L Account:	507613
P.O. Number:	A. Peckham
Charge Unit:	9408

Processed by Dave Spencer 1/11/08

Bank of Xanadu

Date: February 11, 2008

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):

()	No Contract on File
()	Dollar Amount Exceeds Contract Fee by $
()	Invoice Period Outside of Contract Dates
()	No Time Sheet
()	No Invoice/Time Sheet Approval
()	Time Sheet & Invoice Discrepancy
()	Billed Rate Different from Contract Rate
()	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

Invoices

ID Number	Programmer	Vendor	Charge	Invoice #	Date Paid	Begin Date	End Date	Rate	Total Hours	Total Invoice	Accrued	Memo
Wilki0508	Wilkins, Peter	Donny Wicks Associates	9408	1001	12/21/07	12/01/07	12/15/07	59.00	64.0	3,776.00		
								Total:		**3,776.00**		
								Total for December:		**3,776.00**		
Peckh0908	Peckham, Art	Donny Wicks Associates	9408	329	01/11/08	12/16/07	12/31/07	60.00	60.0	3,600.00	12/07	
Wilki0508	Wilkins, Peter	Donny Wicks Associates	9408	1002	01/11/08	12/16/07	12/31/07	59.00	66.0	3,894.00	12/07	
Brown0391	Brown, Lou	EDS Temps Inc	3072	509	01/11/08	12/17/07	12/31/07	25.00	70.0	1,750.00	12/07	**Dec Exp 13,020.00**
								Total:		**9,244.00**		
Wilki0508	Wilkins, Peter	Donny Wicks Associates	9408	1003	01/25/08	01/02/08	01/15/08	59.00	85.0	5,015.00		
Brown0391	Brown, Lou	EDS Temps Inc	3072	510	01/25/08	01/02/08	01/15/08	25.00	68.0	1,700.00		
								Total:		**6,715.00**		
								Total for January:		**15,959.00**		
Lehre128	Lehrer, Philip	Beltam Systems Inc	3117	101	02/08/08	01/02/08	01/31/08	52.00	165.0	8,580.00	01/08	
Peckh0908	Peckham, Art	Donny Wicks Associates	9408	330	02/08/08	01/02/08	01/31/08	60.00	177.0	10,620.00	01/08	
Wilki0508	Wilkins, Peter	Donny Wicks Associates	9408	1004	02/08/08	01/16/08	01/31/08	59.00	82.0	4,838.00	01/08	
Brown0391	Brown, Lou	EDS Temps Inc	3072	511	02/08/08	01/16/08	01/31/08	25.00	70.0	1,750.00	01/08	
forti0608	Fortier, Brian	EDS Temps Inc	3072	3723	02/08/08	01/02/08	01/31/08	25.00	176.5	4,412.50	01/08	**Jan Exp 36,9**
								Total:		**30,20**		

										0.50	**15.50**
Brown0391 (*start)*	Brown, Lou	EDS Temps Inc	3072	512	02/22/08	02/01/08	02/15/08	25.00	68.0	1,700.00	
									Total:	**1,700.00**	
									Total for February:	**31,900.50**	
									Grand Total:	**51,635.50**	

Accruals

Programmer	Vendor	Charge	Invoice #	Total Invoice	Accrued	Reversed
Brown, Lou	EDS Temps Inc	3072	509	1,750.00	12/07	
		3072	**Total:**	**1,750.00**		**01/10/08**
Peckham, Art	Donny Wicks Associates	9408	329	3,600.00	12/07	
Wilkins, Peter	Donny Wicks Associates	9408	1002	3,894.00	12/07	
		9408	**Total:**	**7,494.00**		**01/10/08**
			December 2007	**9,244.00**		
Brown, Lou	EDS Temps Inc	3072	511	1,750.00	01/08	
Fortier, Brian	EDS Temps Inc	3072	3723	4,412.50	01/08	
		3072	**Total:**	**6,162.50**		**02/10/08**
Lehrer, Philip	Beltam Systems Inc	3117	101	8,580.00	01/08	
		3117	**Total:**	**8,580.00**		**02/10/08**
Peckham, Art	Donny Wicks Associates	9408	330	10,620.00	01/08	
Wilkins, Peter	Donny Wicks Associates	9408	1004	4,838.00	01/08	
		9408	**Total:**	**15,458.00**		**02/10/08**
			January 2008	**30,200.50**		
			February 2008			
			Grand Total:	**39,444.50**		

Contract Programmers Monthly Expense Recap Report
By Division and Unit
January 2008

Programmer	Vendor	Division	Charge	Invoice #	Begin Date	End Date	Total Hours	Total Invoice	Accrued
	Division:	**AMB**					**Total for Division:**	**8,580.00**	
Lehrer, Philip	Beltam Systems Inc	AMB	3117	101	01/02/08	01/31/08	165.0	8,580.00	01/08
			3117				**Total for Charge Unit:**	**8,580.00**	
	Division:	**CCR**					**Total for Division:**	**20,473.00**	
Peckham, Art	Donny Wicks Associates	CCR	9408	330	01/02/08	01/31/08	177.0	10,620.00	01/08
Wilkins, Peter	Donny Wicks Associates	CCR	9408	1003	01/02/08	01/15/08	85.0	5,015.00	
Wilkins, Peter	Donny Wicks Associates	CCR	9408	1004	01/16/08	01/31/08	82.0	4,838.00	01/08
			9408				**Total for Charge Unit:**	**20,473.00**	
	Division:	**NAB**					**Total for Division:**	**7,862.50**	
Brown, Lou	EDS Temps Inc	NAB	3072	510	01/02/08	01/15/08	68.0	1,700.00	
Brown, Lou	EDS Temps Inc	NAB	3072	511	01/16/08	01/31/08	70.0	1,750.00	01/08
Fortier, Brian	EDS Temps Inc	NAB	3072	3723	01/02/08	01/31/08	176.5	4,412.50	01/08
			3072				**Total for Charge Unit:**	**7,862.50**	
							Grand Total for January:	**36,915.50**	

Contract Programmers Monthly Expense Recap Report
By Division and Unit
February 2008

Contract Programmer Report
Fee Maximum vs. Acutals
December 2007

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
DIVISION:	**NAB**									
	Unit Number:	3072								
Brown, Lou	12/17/07	06/17/08	25.00	Clark, Rudy	622-2375	29,000.00	1,750.00	6%	01/11/08	27,250.00
DIVISION:	**CCR**									
	Unit Number:	9408								
Wilkins, Peter	12/01/07	05/30/08	59.00	Scott, Randy	622-6047	48,000.00	7,670.00	16%	01/11/08	40,330.00
Peckham, Art	12/16/07	09/30/08	60.00	Scott, Randy	622-6047	88,600.00	3,600.00	4%	01/11/08	85,000.00

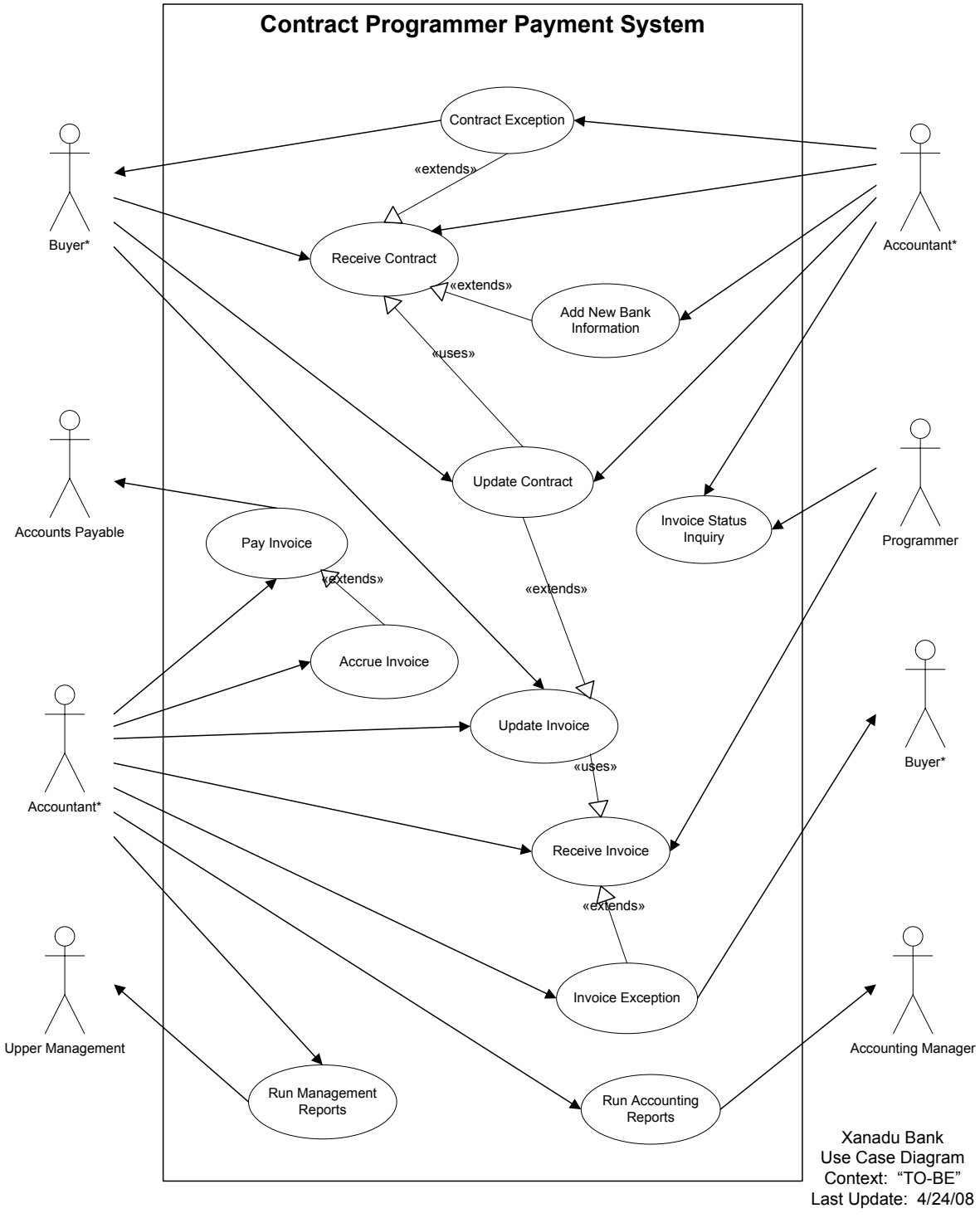
Monthly Contract Recap
As of December 31, 2007

Project Manager:	Clark, Rudy	**Unit:**	3072						
Programmer:	Brown, Lou			**Company:**	EDS Temps Inc		**Project:**	Tax System Assistance	
Start Date:	12/17/07	**End Date:**	06/17/08	**Rate/Hour:**	25.00		**Fee Max:**	29,000.00	**Charge To:** 3072

Invoice Number	**Date Paid**	**Periods Paid**			**Hours**	**Dollar Total**
509	01/11/08	12/17/07	to	12/31/07	70	1,750.00
				Total of Hours & Invoice Dollars:	70	$1,750.00

Total Charged to Contract:	$1,750.00
Percent Used:	6%
Remaining Contract Dollars:	$27,250.00

6. USE CASES/USE CASE DIAGRAM



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:	This use case begins when the Buyer delivers a new contract to the Accountant. 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. This use case ends when the new contract is entered into the system.	
Exception(s):	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:	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:	**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:	This use case begins when the Buyer delivers a contract with new vendor, contact, unit, or division information to the Accountant. 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. This use case ends when the new vendor, contact, unit, or division record is entered into the system.
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:	**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:	This use case begins when the Buyer delivers a contract to the Accountant that is either incomplete or contains invalid information. 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 This use case ends when the incomplete/invalid contract has been returned to the Buyer.	

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:	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:	This use case begins when the Buyer delivers a corrected or updated contract to the Accountant. 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 This use case ends when the contract has been correctly and completely updated in the system.
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:	This use case begins when a new invoice is received for programming services. 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 exceeded 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 This use case ends when the invoice is entered into the system with an "Approved to Pay" status.
Exception(s):	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
Postconditions(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 Total Hours Worked Invoice Total Invoice Terms Approved By Charge Unit Payment Status
Assumptions:	1) A valid contract has already been entered into the system for the services specified on the invoice and the invoice is payable against that contract. 2) If the Vendor is not in the system, then there would also not be a valid contract either. 3) The Vendor will not send an invoice for programming services without first signing a contract with the Bank 4) The Vendor will not send an invalid or incorrect invoice that violated the terms of the contract. 5) The proper invoice approval process has been followed by the Project Manager.
Business Rules:	1) To be considered a valid invoice, it must contain all the information required by both the Buyer and Accounting department (including both signed and approved invoice and time sheet(s). 2) The Bank must not authorize programming services without first executing a valid contract for such services. 3) The project manager is responsible for verifying that the information on the invoice is correct and that a valid time sheet has been included with the invoice.

	4) The project manager must approve the invoice and sign-off authorizing payment, including the proper bank unit that is to be charged. 5) An invoice cannot be paid if it exceeds the terms of the contract, including the dates of service and hourly rate. 6) Payment of invoice does not exceed the contractual fee maximum amount after all the previous invoice totals have been tallied and deducted from the fee maximum.

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 invoide 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:	This use case begins when the Vendor sends an invoice for programming services to the Accountant (should arrive via the Project Manager). 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 This use case ends when the invomplete, invalid, or unpayable invoice has been returned to the Buyer.
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:	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:	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 contact, 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:	This use case begins when the Buyer delivers an updated or new invoice and/or a contract extension to the Accountant. 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

	This use case ends when the revised or new invoice and/or contract extension has been correctly and completely updated in the system.
Exception(s):	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):	1) A revised invoice has been received from the Buyer 2) A contract extension has been received from the Buyer
Postconditions(s)	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 (see UC005 "Receive Invoice" for applicable data) Payment Status Invoice Notes
Assumptions:	1) The Buyer will have taken the necessary steps to ensure the new information provided on the invoice and/or contract

	extension is sufficient to allow the invoice to be payable 2) If the vendor changes the invoice number, they will either issue a credit memo to cancel the previous unpayable invoice OR they will somehow notify the Accounting department (Accountant) with instructions to modify/change the previous invoice number and ignore the fact that it existed
Business Rules:	1) The Accountant must enter the updated invoice and/or contract information into the system and make a note of the date that the updated contract and/or invoice was returned by the Buyer. 2) If the invoice number needs to be changed, the Accountant must document fully the circumstances surrounding the change and/or document that a credit memo was received to offset the original invoice. 3) The Accountant must re-check the updated invoice against the contract limitations before approving it for payment

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:	This use case begins when the Accountant receives an invoice payment status inquiry from the Vendor. 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 This use case ends when the Vendor has received the invoice payment status.
Exception(s):	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)	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:	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:	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:	**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:	This use case begins when the Accountant has set the invoice Payment Status to "Approved to Pay". 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 This use case ends when the invoice has been has been updated with the "Date Paid" and sent to the A/P department to have a check issued.
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 6th 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 1st 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 1st 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:	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 accural 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:	**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 accred 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 6th day of the following month.
Related Use Cases:	Pay Invoice (extends)
Normal flow of events:	This use case begins when an invoice has been entered into the system that either cannot be processed for payment OR is payble and cannot have a check cut in the current month. 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 This use case ends when an invoice has been designated as accrued in the system.
Exception(s):	None
Precondition(s):	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 6th 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:	This use case begins when the deadline due date for the "General Ledger Expense Report" and "Accrual Report" is reached. 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 Acounting Manager This use case ends when both the "General Ledger Expense Report" and "Accrual Report" have been delivered to the Accounting Manager.
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:	(General Ledger Expense Report) 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) (Accrual Report) Programmer Vendor Charge Unit Invoice Number Invoice Total Date Accrued Total Accrued (calculated)
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 6th 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:	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. 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 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).
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:	(Contract Programmer's Monthly Expense Recap Report) 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) (Contract Programmer Report - fee Maximum vs. Actuals) Division Charge Unit Programmer Service Start Date Service End Date Hourly Rate Project Manager

	PM Phone Number Fee Maximum Total Charged to Contract (calculated) Percent Used (calculated) Date Last Charged (calculated) Under/Over Contract Fee Max (calculated) (Monthly Contract Recap) Project Manager PM Contact Unit Programmer Vendor Begin Date (contract) End Date (contract) Hourly Fee Project Description Fee Maximum Charge Unit Invoice Number Date Paid Service Start Date Service End Date Total Hours Worked Invoice Total Total Charged to Contract (calculated) Percent Used (calculated)

	Remaining Contract Dollars (calculated)
Assumptions:	There will actually be at least one invoice paid to the contract programmer G/L account 507613 in the current reporting period
Business Rules:	The "Contract Programmer's Monthly Expense Recap Report", "Contract Programmer Report - Fee Maximum vs. Actuals", and "Monthly Contract Recap" are due to be sent to the various bank requesting units by the 11th business day of the month.

7. Internal Procedures

INSERT INTO BUYERS (ALL BUYER FIELD NAMES)

VALUES (NEW BUYER INFO);

INSERT INTO CONTACTS (ALL CONTACT FIELD NAMES)

VALUES (NEW CONTACT INFO);

INSERT INTO CONTRACTS (ALL CONTRACT FIELD NAMES)

VALUES (NEW CONTRACT INFO);

INSERT INTO DIVISIONS (ALL DIVISION FIELD NAMES)

VALUES (NEW DIVISION INFO);

INSERT INTO INVOICES (ALL INVOICE FIELD NAMES)

VALUES (NEW INVOICE INFO);

```
INSERT INTO PROGRAMMERS (ALL PROGRAMMER FIELD NAMES)
        VALUES (NEW PROGRAMMER INFO);

INSERT INTO SYSTEM_USERS (ALL SYSTEM_USER FIELD NAMES)
        VALUES (NEW SYSTEM_USER INFO);

INSERT INTO UNITS (ALL UNIT FIELD NAMES)
        VALUES (NEW UNIT INFO);

INSERT INTO VENDORS (ALL VENDORS FIELD NAMES)
        VALUES (NEW VENDORS INFO);

UPDATE     CONTRACTS
SET        FORM VALUES
WHERE      CONTRACT FIELDS EQUAL THE RECORD TO BE CHANGE;

UPDATE     INVOICES
SET        FORM VALUES
WHERE      INVOICE FIELDS EQUAL THE RECORD TO BE CHANGE;

SELECT     CONCAT(Contact_Fname, ` `, Contact_Lname)
FROM       CONTACTS;
```

```
SELECT    Division_ID
FROM      DIVISIONS;

SELECT    Unit_Number
FROM      UNITS;


SELECT    Vendor_ID
FROM      VENDORS;

SELECT    Contract_Code
FROM      CONTRACTS;

SELECT    COMPLETE CONTRACT RECORD
FROM      CONTRACTS
WHERE     Contract_Code = formContractCode;

SELECT    Contract_Fee_Max, Contract_Start_Date,
          Contract_End_Date, Contract_Rate
FROM      CONTRACTS
WHERE     Contract_ID = formContractId;
```

```
SELECT    Invoice_Number
FROM      INVOICES;

SELECT    COMPLETE INVOICE INFO
FROM      INVOICES
WHERE     Invoice_Number = formInvoiceNumber;

SELECT    INFO FOR THE GL EXPENSE REPORT
FROM      THE TABLES THAT HAVE THE INFO
WHERE     IT IS IN A SPECIFIC DATE PERIOD;

SELECT    INFO FOR THE ACCRUAL REPORT
FROM      THE TABLES THAT HAVE THE INFO
WHERE     IT IS IN A SPECIFIC DATE PERIOD;

SELECT    INFO FOR THE PROG EXPENSE REPORT
FROM      THE TABLES THAT HAVE THE INFO
WHERE     IT IS IN A SPECIFIC DATE PERIOD;

SELECT    INFO FOR THE FEE MAX REPORT
FROM      THE TABLES THAT HAVE THE INFO
WHERE     IT IS IN A SPECIFIC DATE PERIOD;
```

SELECT INFO FOR THE CONTRACT RECAP REPORT

FROM THE TABLES THAT HAVE THE INFO

WHERE IT IS IN A SPECIFIC DATE PERIOD;

8. Design Standards Document

Interface Design

Login Screen

Font: centered Verdana 14 point for help boxes

Font: left-aligned Verdana 10 point for forms

All buttons are 120px width and 30px height

Form borders have green border

All borders other borders are black

All borders have rounded corners

Help box at top left and on the bottom of the iframe

All forms will be in the iframe

The form title is shown using h1 tags

Form buttons animate to oval on hover

Submit buttons are green and reset buttons are red

Bottom help box shows help when hovering on a form button

Bottom help box shows help when hovering on a form field

Accountant Screen

Font: centered Verdana 14 point for help boxes and navigation buttons

Font: left-aligned Verdana 10 point for forms

All buttons are 120px width and 30px height

Form borders have green border

All borders other borders are black

All borders have rounded corners

Help box at top left and on the bottom of the iframe

All forms will be in the iframe

The form title is shown using h1 tags

Form buttons animate to oval on hover

Navigate buttons have a shadow under them

Navigation main buttons are blue and sub buttons are orange

Navigation main buttons animate to ovals and move down on hover

Navigation sub buttons animate to ovals on hover

Submit buttons are green and reset buttons are red

Submit and reset buttons animate to ovals on hover

Bottom help box shows help when hovering on a form button

Bottom help box shows help when hovering on a form field

Manager Screen

Font: centered Verdana 14 point for help boxes and navigation buttons

Font: left-aligned Verdana 10 point for forms

All buttons are 120px width and 30px height

Form borders have green border

All borders other borders are black

All borders have rounded corners

Help box at top left and on the bottom of the iframe

All forms will be in the iframe

The form title is shown using h1 tags

Form buttons animate to oval on hover

Navigate buttons have a shadow under them

Navigation buttons are blue

Navigation buttons animate to ovals and move down on hover

Submit buttons are green and reset buttons are red

Submit and reset buttons animate to ovals on hover

Bottom help box shows help when hovering on a form button

Bottom help box shows help when hovering on a form field

Coding Standards

Commenting Code

Commenting will be used for all functions, conditions, and loops. The coding will consist of:

- Author name and revised author name if applicable
- Date and revised date if applicable
- Description of the function and a revised description if applicable

Whitespace

Whitespace and carriage returns will be used for readability purposes

Naming Conventions

Functions and variables will use camelcase with short descriptive words.

Variables will use shorthand 3-letter prefix to show the type of data.

SQL statements will use all caps for everything except for field names.

SQL statements will use underscores with short descriptive words.

9. Issues List

- Is Xanadu's IT Department going to handle security and networking, or should we hire somebody?
- Does Xanadu want our team to train everyone or just train the trainers?

10. Futures List

- Customizable user interface
- Changing functionality to update database structure and system interface based on what is needed/wanted by the users