team unicorn

Edmonds Community College

Systems Design Document

Draft for Review
Prepared 4/25/14
Prepared by Team Unicorn:

Angela Hesse
Gregg Ratliff
Jim Ferguson
Kat Savitzky
Ryan Neff

team unicorn

Memorandum

To: Vice President Patrick Jay

From: Team Unicorn

Date: April 25th, 2014

Subject: System Design Document Review

The attached report contains the system design documentation for the implementation of an automated contractor payment system at the Bank of Xanadu. Team Unicorn would like to review the design document with you on Saturday May 3rd at the Bellevue Branch.

System Requirements Document

Bank of Xanadu

Contractor Account Tracking System Design Document

Prepared 4/24/14 Spring 2014

Prepared by Team Unicorn:

Angela Hesse
Gregg Ratliff
Jim Ferguson
Kat Savitzky
Ryan Neff

System Requirements Document

team unicorn

Table of Contents

Introduction

The purpose of this Systems Design Specification is to present the steps of the Systems Development Life Cycle (SDLC), which include design and implementation of the new Contractor Account Tracking System. The Design Phase sections discuss the User Interface Design, the Data Design and the Systems Architecture. Following the Design Phase, this document also covers the Implementation Phase. This document contains the following:

- Introduction
- System Architecture – This section discusses the type of hardware and network architecture to be used by the new system, as well as assumptions reached for designing and implementing this project.
- Information/Data Model – This includes the Entity Relationship Diagram (ERD) as well as meta-data dictionary.
- User Navigation Design (Storyboard) – The User Navigation Design both describe and show a graphical representation of the user interface and how the navigation in the interface will work.
- Inputs – This will be a list of the input source documents, as well as the descriptions for such documents. No forms needed to be redesigned for this project.
- Outputs – This section will contain a list and short descriptions of the reports that the system will provide. As of this writing, no reports have had to be redesigned for the new system.
- Procedures – This section contains the external and internal procedures, which describe the general flow of activities when the system is being used (external procedures) and any procedures, functions, SQL queries, VBA code, or pseudo code (internal procedures) in the new system.
- Interface Design and Coding Standards – The portion of the document will show the standards to be used and adhered to by all members of the project team. The guidelines are internal, as well as external and are to ensure consistency in the system.
    - External – Deals with screen fonts, navigation, button size, required command buttons, radio buttons, check-boxes and the like.
    - Internal – This deals with coding standards, as well as object naming conventions.

- Conclusion – The conclusion will summarize the document, as well as request an appointment with the primary users for a design walkthrough.
- Appendix – the Appendices will contain all of the supporting documentation pertaining to this project.

Architecture & Design Considerations

Architecture

The proposed Contractor Account Tracking System will be housed at the Bank of Xanadu, and will work in a client/server environment. It will consist of multiple workstations; the servers will also be built with an appropriate fault-tolerant architecture. The system will be connected via local network (LAN), with internal network connectivity supplied by switches. Routers will provide connectivity to other networks within the Bank, as well as to the Internet. For additional data security, the System's database should be backed up regularly to additional servers located on-site as well as in another location. The new system will be rolled out as a pilot deployment, as determined by Bank of Xanadu Management.

Assumptions

Team Unicorn assumes that there are sufficient desktop PCs, servers, as well as network hardware and software existing within the Bank of Xanadu to implement the new system. The new system is being designed with following considerations:

- Scalability – the system will be designed for optimum scalability.
- Extensibility – the system will be built to allow additional functionality to be added later, as needed.
- Usability – the system will be built with a user-friendly and intuitive interface, which will also promote user productivity and customer satisfaction.
- Multi-user capability – this system will be able to accommodate many users at once.
- Security requirements – appropriate multiple-level application and hardware security will be designed and implemented in the new system.
- Distribution of data – data will be resident on servers, which will be designed for both performance, as well as fault tolerance.
- Transaction volume – the new system is designed to accommodate a large transaction volume, and will also be adaptable for the Bank's needs, as required.

Information & Data Model

The information and data model for the proposed system for the Bank of Xanadu Contract Programmer Payment Management System consists of the following Entities:

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

CONTRACTS (Contract_ID, Cont_Number, Cont_Start_Date, Cont_End_Date, Cont_Fee_Max, Cont_Hourly_Rate, Cont_Description, Cont_Notes, *Vendor_ID, Unit_ID, Contact_ID, Employee_ID, Programmer_Id*)

DIVISIONS (Division_ID, Div_Name)

EMPLOYEES (Employee_ID, Emp_F_Name, Emp_L_Name, Emp_Contact, Emp_User_Name, Emp_Password, Emp_Auth_Level, Emp_Category)

INVOICES (Invoice_ID, Inv_Number, Inv_Date_Issued, Inv_Date_Received, Inv_Date_Paid, Inv_Terms, Inv_Amount, Inv_Rate, Inv_Hours, Inv_Start_Date, Inv_End_Date, Inv_Status, Inv_Accrual, Inv_Notes, *Vendor_ID, Employee_ID, Contract_ID*)

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

UNITS (Unit_ID, Unit_Number, *Division_ID*)

VENDORS (Vendor_ID, Ven_Name, Ven_Number, Ven_Address, Ven_Phone, Ven_Email)

To view the Entity Relationship Diagram (ERD), please see Appendix A. The Meta-Data Dictionary, which shows in greater detail the entities, attributes and keys in this ERD, as well as their domain and referential integrity constraints, can be found in Appendix B.

User Navigation Design

Bank of Xanadu's Automated Contractor Payment system is designed to be easily navigable, so the few tasks that are not completely automated can be quickly accessed & executed. Breadcrumb navigation will help the user remain spatially aware of their place in the system hierarchy as they navigate through sub-menus. The interface will also mirror common website navigation that most users are already familiar with. The navigation will be displayed using four tabs for the main working sections, and a fifth tab for logging in and out of the system. A detailed storyboard for the User Navigation design has been included in Appendix C. Bank of Xanadu's reporting system will be segmented into the following sections for ease of navigation:

Invoices Menu:

Under the Invoices menu, users will be able to View Previous Invoices, update existing invoices, and add new invoices.

Contracts:

Under the contracts menu, users will be able to create and renew contracts, update and add both vendor and programmer information

Reports Menu:

Under the reports menu users will be able to generate reports for contracts nearing their end date, accrual reports, invoice reports, and contract reports.

Maintenance Menu:

Under the maintenance menu, users will be able to update and enter new information for: Contacts, Vendors, Divisions, Units, Employees, and programmers.

User Logon:

Each user will be required to logon to the system, so that the appropriate features can be displayed according to their access credentials. menus.

Inputs & Outputs

Here is a list of the current input and output documents that are currently used in your system.

Input Documents:

- **Vendor Numbers** – A/P Sends vendor numbers to accounting be used on data entry sheet.
- **Contract** – Created by Buyers for Vendors and contains the specific information relating to contract programmer services.
- **Extension** – Extended by Buyers for Vendors and used to amend some part of the original contract.
- **Invoice** – Turned in by Vendor to the system for payment
- **Time Sheet** – Filled out by contract programmer to specify hours worked in a particular time period.

Output Documents:

- **Invoice Reports** – The reports generated are used by the accountants to balance that vendor account at the end of each month.
- **Data Entry Sheet** – Attached to invoices that are sent to the A/P Department for payment.
- **Accrual Report** – Shows invoices that have been accrued, so that accruals can be processed and then reversed the following month.
- **Monthly Expense Recap** – This report shows the contract programmers invoice expenses paid each month. It is sent to each bank division and is then sorted by charge unit.
- **Contract Programmer Report, Fee Maximum vs. Actuals** – This report shows the contract fee maximum and the invoices that have been paid against each contract with the remaining fee max available. It is sent to each division and then sorted by charge unit.
- **Monthly Recap Report** – This report shows contract and invoice information for each Project Manager - it is used by them to keep track of the contract and programmers working for them.

All of these documents should be fully compatible with the new system that Team Unicorn is currently designing and building

Procedures

External Procedures

Using a clear interface users will be able to select the proper links and buttons they need to input specific data. After logging into the system through an opening screen the user will be directed to the first menu. This menu will contain Contracts, Invoices, Reports, Contact Maintenance and Log out.

When *contracts* has been selected the user will be taken to another menu which will contain options Create new contract, Contract renewal, return to main menu. Each option will be designed to simplify data entry and cut down on input errors making it more efficient.

When the user needs to input invoice info this selection will be taken to the invoice menu which will give the option to add a new invoice, update existing invoices, view previous invoices. This will be set up so that existing vendors and contracts are pulled up via a drop box menu. This will also check for errors in the amount and dates of the invoice generating an exception memo to be sent to the vendor requesting.

When *Maintenance* has been selected, the user will be taken to the Contact Maintenance menu, where the user can update or add programmer or vendor information, add new programmer, add a new vendor, or update vendor and programmer information.

When management has requested a report the user will select the *report* menu where they will be able to print out Reports for the contracts, invoices, accruals, as well as reports on contracts nearing their end date.

For information on how the features are used we have included a Use Case Diagram and Use Case scenarios. Following is a list of the Use case scenarios that you can find in the Appendix.

- UC001-Receive Contract
- UC002-Add new bank information
- UC003-Contract Exception
- UC004-Update Contract

- UC005-Receive Invoice
- UC006-Invoice Exception
- UC007-Update Invoice
- UC008-Invoice Status Inquiry
- UC009-Pay Invoice
- UC010-Accrue Invoice
- UC011-Run Accounting Reports
- UC012-Run Management Reports

Internal Procedures

Queries

Below are brief explanations of the queries currently identified.

Name	Invoice Status Inquiry
Description	Report for Vendor inquiry for payment status
User	Accountant
Tables	Contract, Invoices, Vendors,

Name	Accounting Reports
Description	This is the month-end reports for the accounting department.
User	Accountant
Tables	Contracts, Invoices, Vendor,

Name	Management Reports
Description	The month-end reports for the Bank Managers
User	Accountant
Tables	Contracts, Invoices, Accruals, Vendors, Programmers

Interface Design & Coding Standards

External Standards

For more detailed specifications, see Appendix F, design standards specifications.

Graphics: Text and colors will be used to identify the functions of various high-level command buttons. Iconography has been intentionally omitted to provide a streamlined interface free of visual clutter.

Colors: Default colors will be used for any command buttons, text boxes, list boxes, combo boxes, message boxes menus and scroll bars. If labels, option ("radio") buttons, check boxes and frames, colors will be matched with those of their backgrounds, in order to provide an overall consistent appearance. When forms are used, colors that help text and lines appear the most readable to users will be used.

Fonts: The base font will be Arial and the font size will range from the minimum of 12 to the maximum of 26. Any text requiring the users' immediate attention will be in a larger font size and will be in a bold, italic, or underlined style. Any combination of the three can be used, depending on the importance of the text.

Form and Button Size: This will depend upon the screen resolution of the given monitor of the user. The standards will be based on 800 x 600 and the 1024 x 768 resolutions, because these resolutions are the most common at this time. It is important that the sizes of forms and buttons be not too large for the 800 x 600 and not too small for the 1024 x 768 screen resolutions.

Command Buttons: Command buttons will be included in every module and will be macro-written. Such buttons are to perform a specific set of tasks. Command buttons will edit records, query requested data, print and view records, save records, exit a particular form without saving a record, and quit the application.

List Boxes: List boxes will be used to select data from a set of predetermined choices to be entered into a system as part of a record.

Internal Standards

Program Documentation: In every form where one or more procedures are written, a header must be included. This header will include the programmer name, project name and completion date. A descriptive comment will be written for every procedure to explain the purpose of the procedure. A short comment will be included for lines of code for the benefit of other programmers, so that the purpose and function of a particular line of code is clear.

Naming Convention for Code Variables and Objects: The “Hungarian” naming convention for variables and objects will be used. (Recommended by Microsoft.) Names start with a three-letter prefix, in lower case, followed by the actual name of the object or variable, starting with a capital letter for each word in the name. If the object or variable has more than one word, each word in the name will start with a capital letter, and there will be no spaces between multiple words making up that name, for example: “strLastName”, or “intStartDate”.
The prefixes for the following variables will be used and are as follows:

“int”: integers
“lng”: long integers
“sgl”: single floating points
“dbl”: double floating points
“str”: strings
“bln”: Boolean

Database Objects Naming Conventions: Objects in the database will have the same naming conventions as noted above.
The prefixes for common database objects are listed below:

“frm”: forms
“rpt”: reports
“tbl”: tables
“qry”: queries
“mnu”: menus
“lbl”: labels
“cbo”: combo boxes
“opt”: option buttons

Conclusion

This document includes all of the specifications and design plans for the new system currently being designed and built for the Bank of Xanadu by Team Unicorn. We would like to schedule an appointment with some representatives of your Bellevue office's accounting department to demonstrate our prototype and obtain any user suggestions that may improve the efficiency of the new system. If possible, we would like to schedule this appointment no later than the week of May 19, 2014.

_____ _____

Dave Spencer — Date
Accounting Manager

_____ _____

Patrick Jay — Date
Vice President

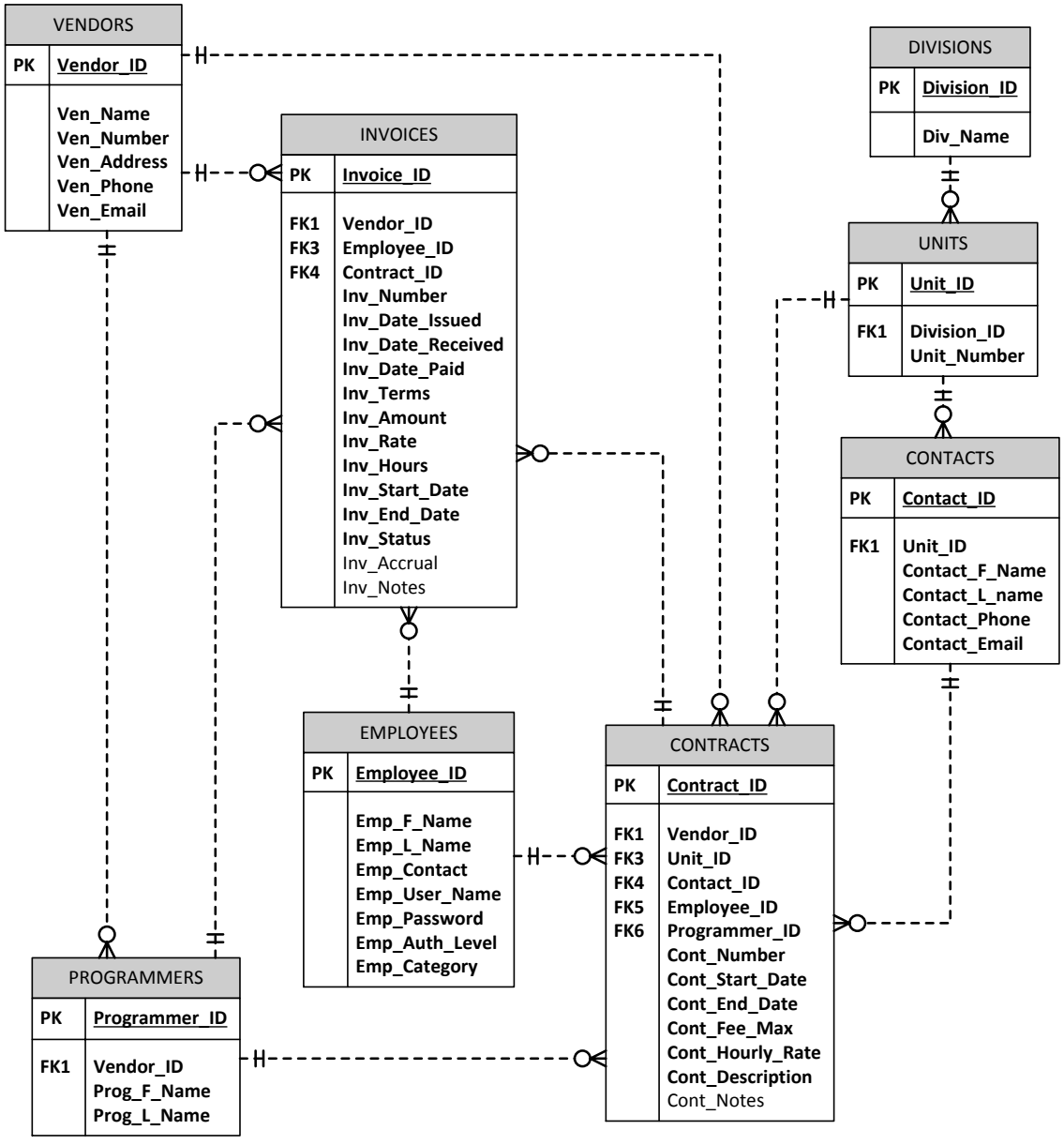
team unicorn

Appendices

Appendices Table of Contents

Appendix A: Entity Relationship Diagram

Bank of Xanadu ERD



Bank of Xanadu
Entity Relationship Diagram
Last Modified 4/19/14

Appendix B: MetaData Dictionary

Entity Name	Attribute Name	Definition	Domain Constraints	Referential Integrity Constraints
CONTACTS	Contact_ID	A unique identifier for each contact.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Unit_ID	The unique identifier for a bank unit.	Required. **FK** (UNITS)	A CONTACTS record cannot exist without a UNITS record.
	Contact_F_Name	The given/first name of the contact.	Required. Non-unique. Char (35)	
	Contact_L_Name	The surname/last name of the contact.	Required. Non-unique. Char (35)	
	Contact_Phone	The telephone number of the contact.	Required. Non-unique. Input Mask: Phone (###)-###-####	
	Contact_Email	The email address of the contact.	Required. Unique. Char (35)	
CONTRACTS	Contract_ID	The unique identifier for a contract.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Vendor_ID	The unique identifier for each vendor.	Required. **FK** (VENDORS)	A CONTRACTS record cannot exist without a VENDORS record.
	Unit_ID	The unique identifier for each bank unit.	Required. **FK** (UNITS)	A CONTRACTS record cannot exist without a UNITS record.

	Contact_ID	The unique identifier for each contact.	Required. **FK** (CONTACTS)	A CONTRACTS record cannot exist without a CONTACTS record.
	Employee_ID	The unique identifier for each employee.	Required. **FK** (EMPLOYEES)	A CONTRACTS record cannot exist without an EMPLOYEES record.
	Programmer_ID	The unique identifier for a contract programmer.	Required. **FK** (PROGRAMMERS)	A CONTRACTS record cannot exist without a PROGRAMMERS record.
	Cont_Number	The contract numbers used by the Bank of Xanadu.	Required. Unique. Char (10)	
	Cont_Start_Date	The date, on which a contract begins.	Required. Non-unique. Input Mask: Date mm/dd/yyyy	
	Cont_End_Date	The date, on which a contract ends.	Required. Non-unique. Input Mask: Date mm/dd/yyyy	
	Cont_Fee_Max	The maximum amount payable under a given contract.	Required. Non-unique. Numeric (10) Values in USD.	
	Cont_Hourly_Rate	The rate per hour earned by a contract programmer and specified in a contract.	Required. Non-unique. Numeric (6) Values in USD.	
	Cont_Description	The description about the contract.	Required. Non-unique. Char (255)	
	Cont_Notes	Any additional notes or remarks about the contract.	Optional notes field. Char (255)	

DIVISIONS	Division_ID	The unique identifier for each bank division.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Div_Name	The name of a bank division.	Required. Unique. Char (35)	
EMPLOYEES	Employee_ID	The unique identifier for each employee.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Emp_F_Name	The given/first name of an employee.	Required. Non-unique. Char (35)	
	Emp_L_Name	The surname/last name of an employee.	Required. Non-unique. Char (35)	
	Emp_Contact	The contact person for a particular employee.	Required. Non-unique. Char (35)	
	Emp_User_Name	The user name used by an employee to log on to this system.	Required. Unique. Char (50)	
	Emp_Password	The password used by an employee to be authenticated in this system.	Required. Unique. Char (50)	
	Emp_Auth_Level	The level of authorization possessed by an employee when using this system.	Required. Non-unique. Char (35). Valid values are: "Read-Only" or "Read-Write".	

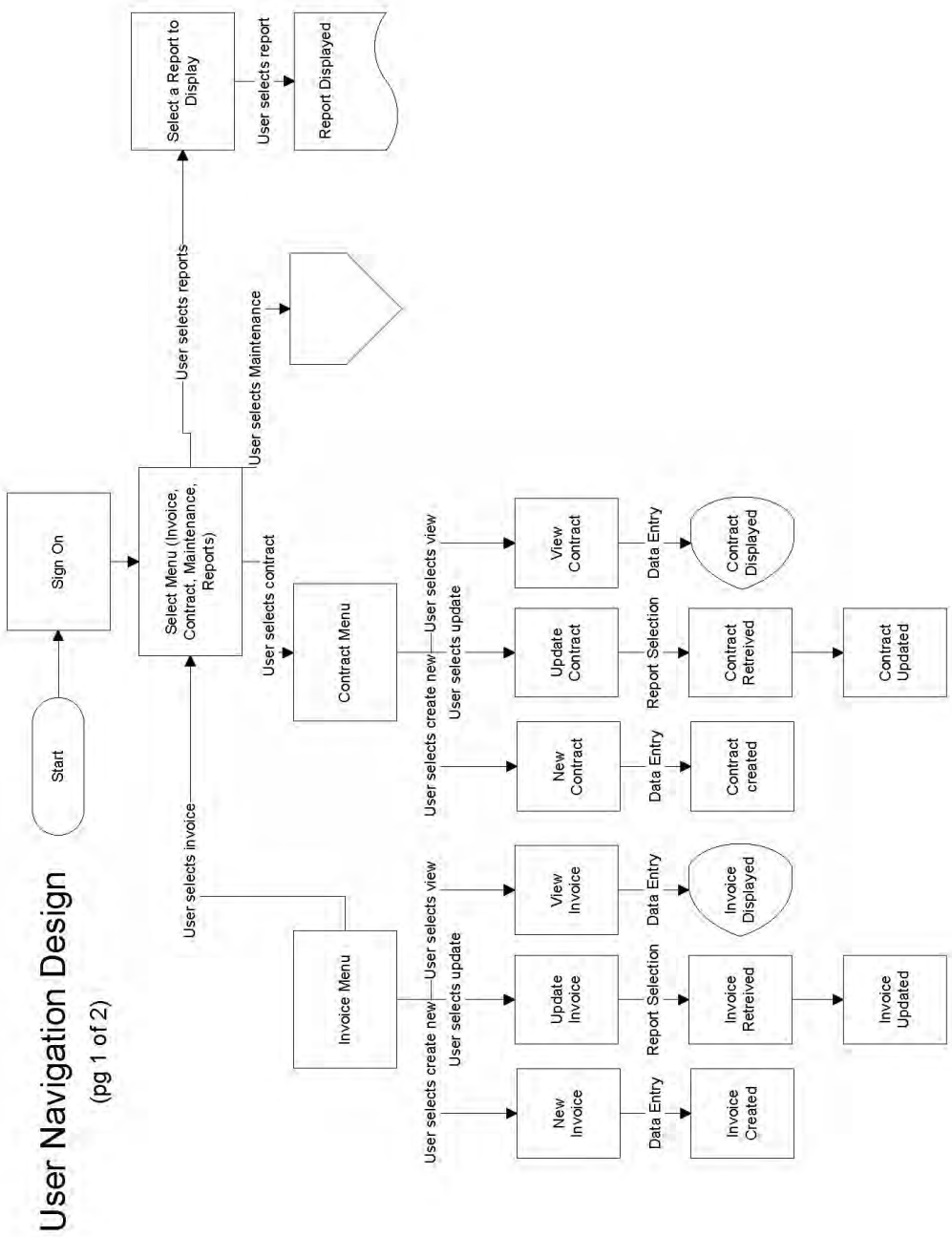
	Emp_Category	The class, job function or role of an employee accessing this system.	Required. Non-unique. Char (35) Valid values are: "Accounting Dept.", "Bank Management", "Project Manager", etc.	
INVOICES	Invoice_ID	The unique identifier of an invoice in this system.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Vendor_ID	The unique identifier for a vendor.	Required. **FK** (VENDORS)	An INVOICES record cannot exist without a VENDORS record.
	Employee_ID	The unique identifier for each employee.	Required. **FK** (EMPLOYEES)	An INVOICES record cannot exist without an EMPLOYEES record.
	Contract_ID	The unique identifier for each contact.	Required. **FK** (CONTRACTS)	An INVOICES record cannot exist without a CONTRACTS record.
	Inv_Number	The legacy number assigned to each invoice. Generated by the vendor prior to being received by Bank of Xanadu.	Required. Unique. Numeric (10)	
	Inv_Date_Issued	The date, on which an invoice was issued.	Required. Non-unique. Input Mask: Date mm/dd/yyyy	
	Inv_Date_Received	The date, on which an invoice was received at the Bank of Xanadu.	Required. Non-unique. Input Mask: Date mm/dd/yyyy	

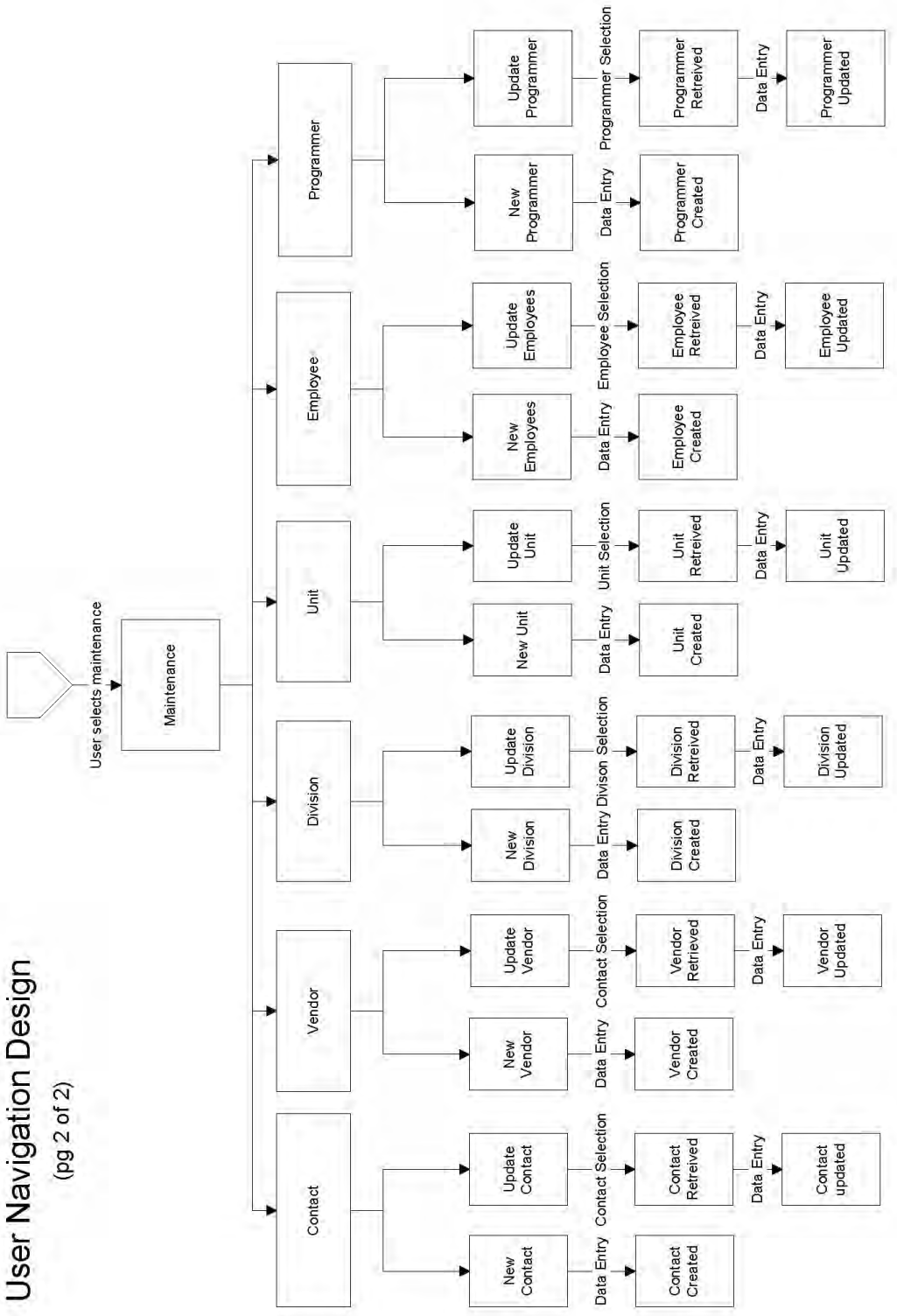
	Inv_Date_Paid	The date, on which the invoice was paid.	Required. Non-unique. Input Mask: Date mm/dd/yyyy	
	Inv_Terms	This field indicates whether an invoice is due upon receipt, or after a certain period of time.	Required. Non-unique. Valid values are: "Due On Receipt", "Net 30", or "2/10 Net 30".	
	Inv_Amount	The total dollar amount due on an invoice.	Required. Non-unique. Numeric (10) Values in USD.	
	Inv_Rate	The rate per hour indicated on an invoice.	Required. Non-unique. Numeric (5) Values in USD.	
	Inv_Hours	The number of hours billed on an invoice.	Required. Non-unique. Numeric (5)	
	Inv_Start_Date	The starting date of work billed on an invoice.	Required. Non-unique. Input Mask: Date mm/dd/yyyy	
	Inv_End_Date	The ending date of work billed on an invoice.	Required. Non-unique. Input Mask: Date mm/dd/yyyy	
	Inv_Status	Indicates whether an invoice is pending or paid.	Required. Non-unique. Valid values are: "Received", "Pending", "Paid", etc.	
	Inv_Accrual	Indicates if an invoice is to be accrued.	Optional. Non-Unique. Valid Values are "Yes" or "No"	
	Inv_Notes	Any additional notes or remarks about the invoice.	Optional. Notes field. Char (255)	

PROGRAMMERS	Programmer_ID	The unique identifier for a contract programmer	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Vendor_ID	The unique identifier of a vendor.	Required. **FK** (VENDORS)	A PROGRAMMERS record cannot exist without a VENDORS record.
	Prog_F_Name	The given/first name of the programmer.	Required. Non-unique. Char (35)	
	Prog_L_Name	The surname/last name of a programmer.	Required. Non-unique. Char (35)	
UNITS	Unit_ID	The unique identifier for each bank unit.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Division_ID	The unique identifier for each bank division.	Required. **FK** (DIVISIONS)	A UNITS record cannot exist without a DIVISIONS record.
	Unit_Number	Legacy unit numbers currently in use and assigned to bank units.	Required. Unique. Numeric (6)	
VENDORS	Vendor_ID	The unique identifier for a vendor.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Ven_Name	The name of the vendor.	Required. Unique. Char (35)	
	Ven_Number	The legacy vendor number currently in used by the Bank of Xanadu.	Required. Unique. Char (10)	
	Ven_Address	The street address of the vendor.	Required. Non-unique. Char 50)	

	Ven_Phone	The contact phone number of the vendor.	Required. Non-unique. Input Mask: Phone (###) -###-####	
	Ven_Email	The contact email address of the vendor.	Required. Unique. Char (35)	

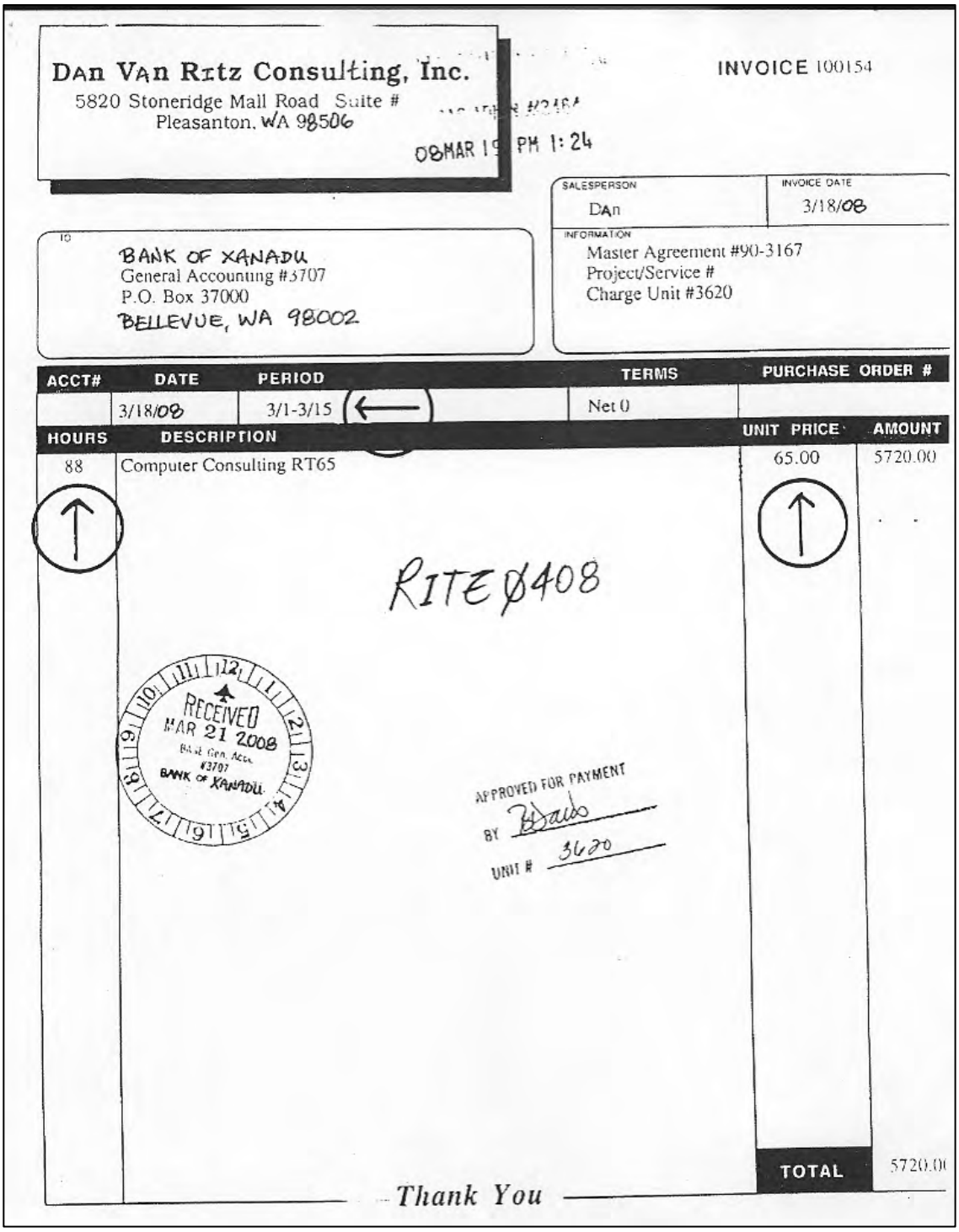
Appendix C: Navigation Design




User Navigation Design
(pg 2 of 2)
User selects maintenance
Maintenance
Contact
Vendor
Division
Unit
Employee
Programmer
New Contact
Update Contact
New Vendor
Update Vendor
New Division
Update Division
New Unit
Update Unit
New Employees
Update Employees
New Programmer
Update Programmer
Data Entry
Contact Selection
Data Entry
Contact Retreived
Data Entry
Contact updated
Contact Created
Data Entry
Vendor Retrieved
Data Entry
Vendor Updated
Vendor Created
Data Entry Divison Selection
Division Retreived
Data Entry
Division Updated
Division Created
Unit Selection
Unit Retreived
Data Entry
Unit Updated
Unit Created
Employee Selection
Employee Retreived
Data Entry
Employee Updated
Employee Created
Programmer Selection
Programmer Retreived
Data Entry
Programmer Updated
Programmer Created

Appendix D: Input

Invoice



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

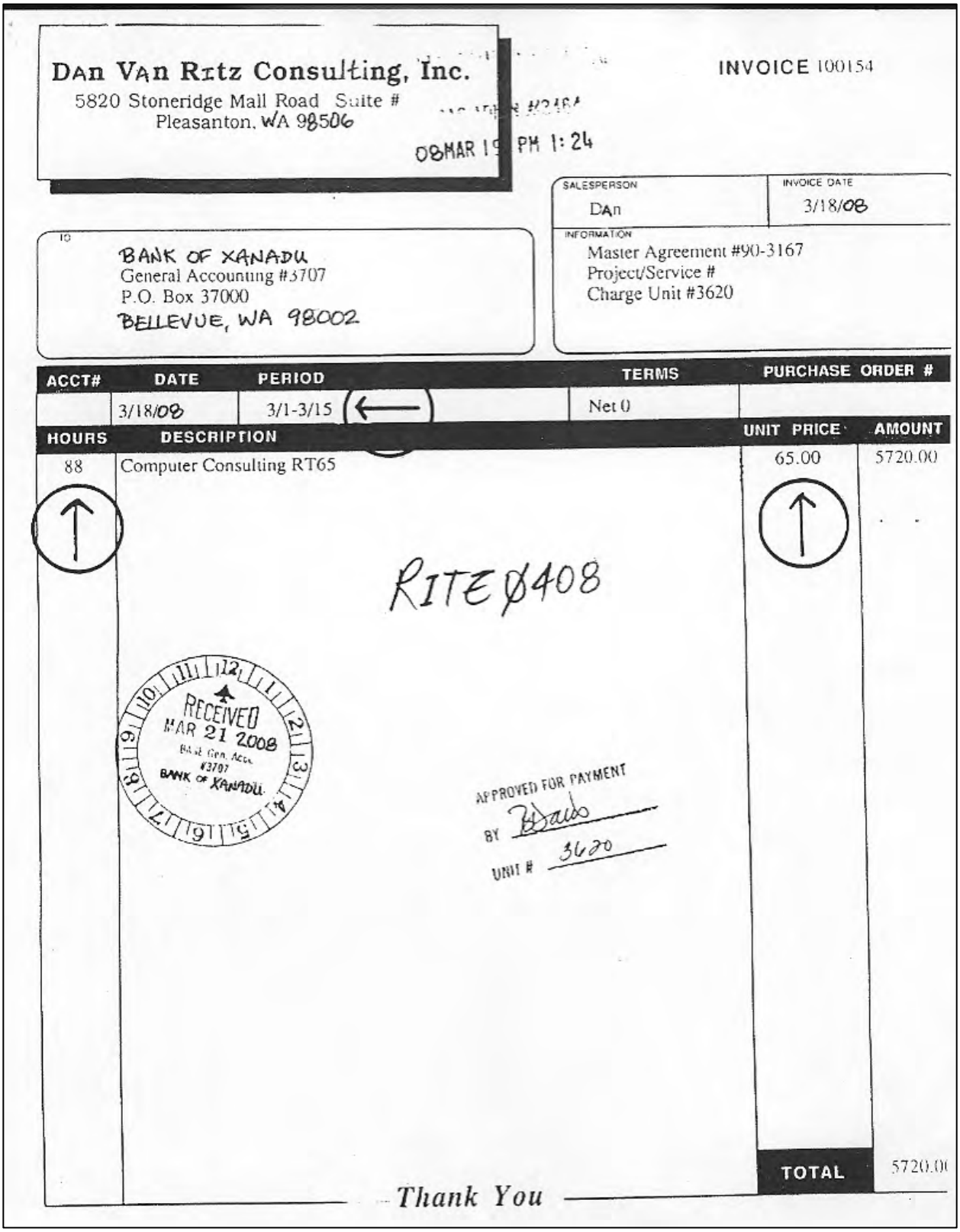
RITE Ø408

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

APPROVED FOR PAYMENT
BY BDavis
UNIT # 3620

TOTAL 5720.00

Thank You



Dan Van Rztz 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

RITE Ø408

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

APPROVED FOR PAYMENT
BY BDavis
UNIT # 3620

TOTAL 5720.00

Thank You

Exception Memo

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):

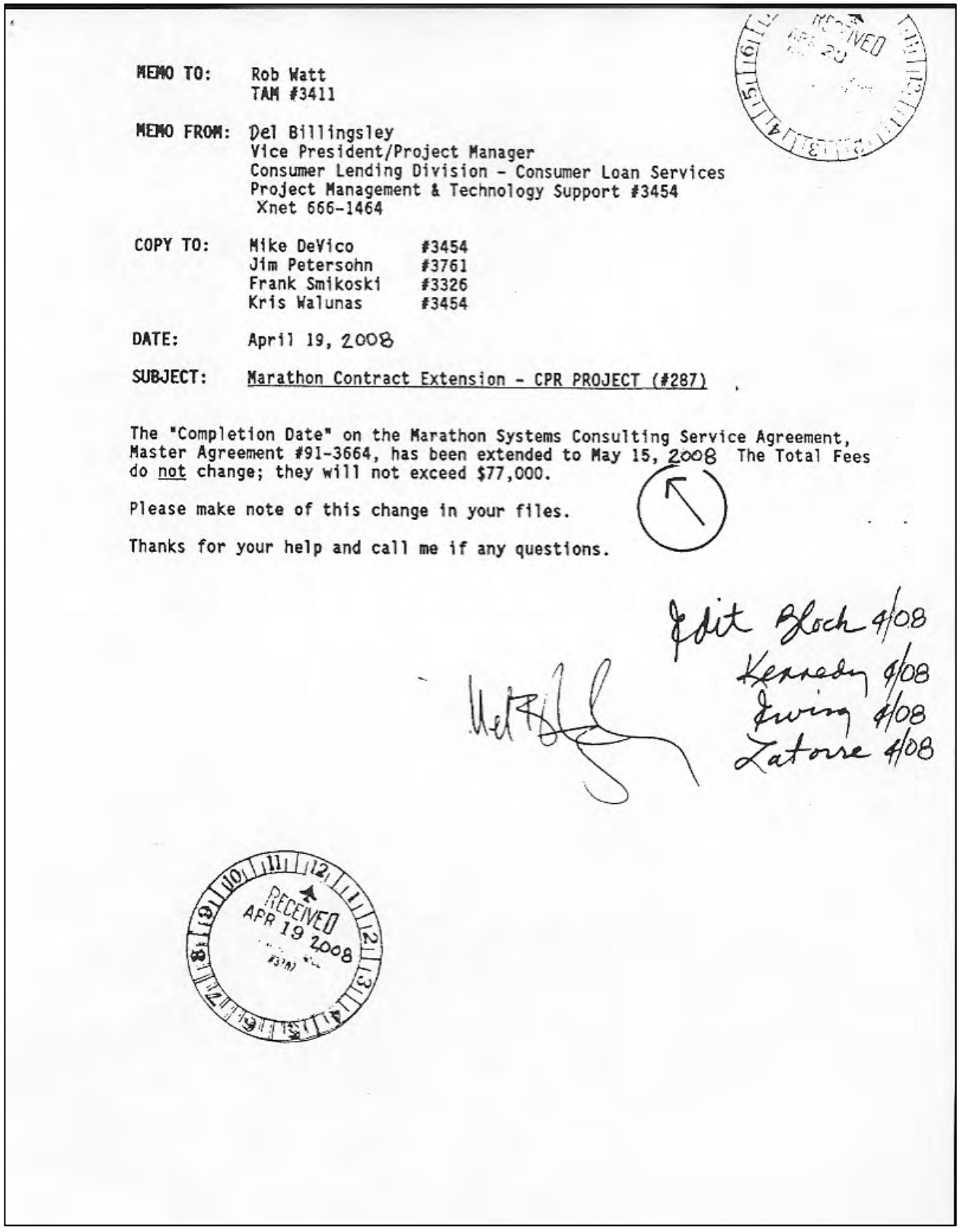
()	No Contract on File
()	Dollar Amount Exceeds Contract Fee by $
()	Invoice Period Outside of Contract Dates
()	No Time Sheet
()	No Invoice/Time Sheet Approval
()	Time Sheet & Invoice Discrepancy
()	Billed Rate Different from Contract Rate
()	Other:

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

Contract Extension



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 Bloch 4/08
Kennedy 4/08
Ewing 4/08
Latorre 4/08

RECEIVED APR 19 2008

Data Entry Sheet

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

Contract

APPENDIX A

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

APPROVED
NAME
DATE 2/14/08

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

APPENDIX A

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

APPROVED
NAME
DATE

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.

(See attached sheet for continuation of Scope of Services) RITZ0408

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

Appendix E: Output

Invoices Report

Invoices

ID Number	Programmer	Vendor	Charge	Division	Invoice #	Date Paid	Begin Date	End Date	Rate	Total Hours	Total Invoice	Accrued	Memo	
Wilki0508	Wilkins, Peter	Donny Wicks As	9408	CCR	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 As	9408	CCR	329	01/11/08	12/16/07	12/31/07	60.00	60.0	3,600.00	12/07		
Wilki0508	Wilkins, Peter	Donny Wicks As	9408	CCR	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	NAB	509	01/11/08	12/17/07	12/31/07	25.00	70.0	1,750.00	12/07		**Dec Exp**
										Total:	**9,244.00**			**13,020.00**
Wilki0508	Wilkins, Peter	Donny Wicks As	9408	CCR	1003	01/25/08	01/02/08	01/15/08	59.00	85.0	5,015.00			
Brown0391	Brown, Lou	EDS Temps Inc	3072	NAB	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**			
Lehre1208	Lehrer, Philip	Beltam Systems	3117	AMB	101	02/08/08	01/02/08	01/31/08	52.00	165.0	8,580.00	01/08		
Peckh0908	Peckham, Art	Donny Wicks As	9408	CCR	330	02/08/08	01/02/08	01/31/08	60.00	177.0	10,620.00	01/08		
Wilki0508	Wilkins, Peter	Donny Wicks As	9408	CCR	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	NAB	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	NAB	3723	02/08/08	01/02/08	01/31/08	25.00	176.5	4,412.50	01/08		**Jan Exp**
										Total:	**30,200.50**			**36,915.50**
Brown0391	Brown, Lou	EDS Temps Inc	3072	NAB	512	02/22/08	02/01/08	02/15/08	25.00	68.0	1,700.00			
(start)														
										Total:	**1,700.00**			
									Total for February:	**31,900.50**				
										Grand Total:	**51,635.50**			

Accruals Report

Accruals

ID Number	Programmer	Vendor	Charge	Division	Invoice #	Date Paid	Begin Date	End Date	Rate	Total Hours	Total Invoice	Accrued	Memo	Reversed
Peckh0908	Brown, Lou	EDS Temps Inc	3072	NAB	509	01/11/08	12/17/07	12/31/07	25.00	70.0	1,750.00	12/07		
			3072		**Total:**						**1,750.00**			**01/10/08**
Wilki0508	Peckham, Art	Donny Wicks Associates	9408	CCR	329	01/11/08	12/16/07	12/31/07	60.00	60.0	3,600.00	12/07		
Brown0391	Wilkins, Peter	Donny Wicks Associates	9408	CCR	1002	01/11/08	12/16/07	12/31/07	59.00	66.0	3,894.00	12/07		
			9408		**Total:**						**7,494.00**			**01/10/08**
					December 2007					**Total:**	**9,244.00**			
Lehre1208	Brown, Lou	EDS Temps Inc	3072	NAB	511	02/08/08	01/16/08	01/31/08	25.00	70.0	1,750.00	01/08		
Peckh0908	Fortier, Brian	EDS Temps Inc	3072	NAB	3723	02/08/08	01/02/08	01/31/08	25.00	176.5	4,412.50	01/08		
			3072		**Total:**						**6,162.50**			**02/10/08**
Wilki0508	Lehrer, Philip	Beltam Systems Inc	3117	AMB	101	02/08/08	01/02/08	01/31/08	52.00	165.0	8,580.00	01/08		
			3117		**Total:**						**8,580.00**			**02/10/08**
Brown0391	Peckham, Art	Donny Wicks Associates	9408	CCR	330	02/08/08	01/02/08	01/31/08	60.00	177.0	10,620.00	01/08		
forti0608	Wilkins, Peter	Donny Wicks Associates	9408	CCR	1004	02/08/08	01/16/08	01/31/08	59.00	82.0	4,838.00	01/08		
			9408		**Total:**						**15,458.00**			**02/10/08**
					January 2008					**Total:**	**30,200.50**			
					February 2008									
					Grand Total:					**Grand Total:**	**39,444.50**			

Monthly Expense Recap Report

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

ID Number	Programmer	Vendor	Division	Charge	Invoice #	Date Paid	Begin Date	End Date	Rate	Total Hours	Total Invoice	Accrued
		Division:	**AMB**							**Total for Division:**	**8,580.00**	
Wilki0508	Lehrer, Philip	Beltam Systems Inc	AMB	3117	101	02/08/08	01/02/08	01/31/08	52.00	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	02/08/08	01/02/08	01/31/08	60.00	177.0	10,620.00	01/08
	Wilkins, Peter	Donny Wicks Associates	CCR	9408	1003	01/25/08	01/02/08	01/15/08	59.00	85.0	5,015.00	
Brown0391	Wilkins, Peter	Donny Wicks Associates	CCR	9408	1004	02/08/08	01/16/08	01/31/08	59.00	82.0	4,838.00	01/08
				9408						**Total for Charge Unit:**	**20,473.00**	
		Division:	**NAB**							**Total for Division:**	**7,862.50**	
Lehre1208	Brown, Lou	EDS Temps Inc	NAB	3072	510	01/25/08	01/02/08	01/15/08	25.00	68.0	1,700.00	
Peckh0908	Brown, Lou	EDS Temps Inc	NAB	3072	511	02/08/08	01/16/08	01/31/08	25.00	70.0	1,750.00	01/08
	Fortier, Brian	EDS Temps Inc	NAB	3072	3723	02/08/08	01/02/08	01/31/08	25.00	176.5	4,412.50	01/08
				3072						**Total for Charge Unit:**	**7,862.50**	
Wilki0508												
Brown0391										**Grand Total for January:**	**36,915.50**	
forti0608												

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

Contract Programmers' Report Fee Maximum vs. Actuals

Contract Programmer Report
Fee Maximum vs. Actuals
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

team unicorn

Monthly Recap Report

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	

Appendix F: Use Cases

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.

Appendix G: Internal Procedures

(This section will be populated during the prototyping phase)

Appendix H: Design Standards Document

Bank of Xanadu Automated Vendor Payment System

Style Guide

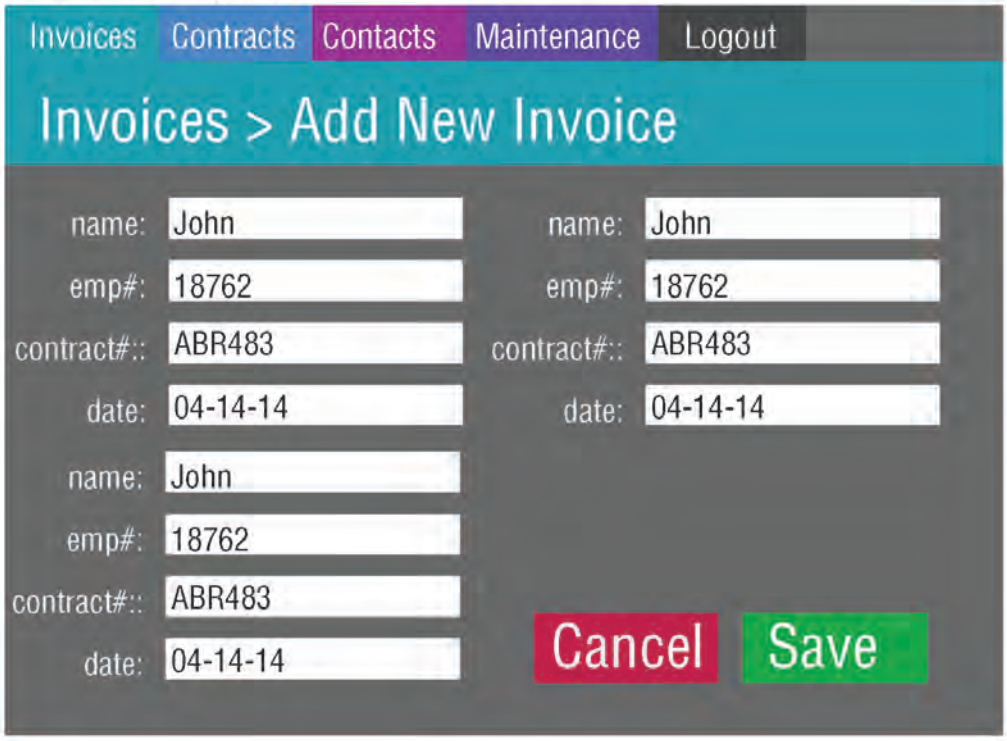
Color Pallete

01A0B1	4B86C6	983594	624EA0	09A64A	BF1F4C	484949	686868

Color Assignments

Invoices	Contracts	Contacts	Maintenance
Save	Cancel	Log In/Out	Background

Generic Layout



Bank of Xanadu Automated Vendor Payment System

Style Guide

Button Styles

All buttons and boxes will be solid colors, no gradients

All buttons will be white text on solid background, with no outline

All buttons and boxes will have square corners

All buttons and boxes will have no borders

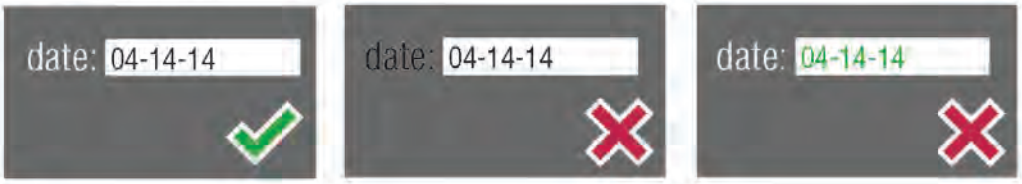
Bank of Xanadu Automated Vendor Payment System

Style Guide

Input Box Styles

All input boxes will be left aligned, all input box labels will be right aligned

All input boxes will be white with black text, all input box labels will be white text.



Appendix I: Issues List

None at this time.

Appendix J: Futures List

When Bank of Xanadu's needs expand beyond the pilot program, team Unicorn strongly recommends the construction of a DMZ server. The additional server construction will be outside-bank-facing, and is considered an additional project, outside the scope of the current project.