Business Systems Solutions and Development

Bruce Norman Don Voges Fatma Sidibeh Marwan Beshir Stuart Crome

---

Date: April 29, 2011
To: Patrick Jay, Vice President and Accounting Manager
From: B.S.S.D
Subject: System Design Document

As a follow-up to the Project Plan, Business Systems Solutions and Development has completed a draft of the System Design Document for Bank of Xanadu's proposed automatic payment system. The attached draft provides our proposed design of the system and includes:

- architecture and design considerations
- information requirements
- an ERD of the data model and relevant Metadata
- a storyboard representation of the proposed database navigation
- a list of inputs source documents and examples of system outputs
- internal and external procedures
- interface design and coding standards

You will find all relevant source documents in the Appendices.

We will present our system design draft to you during our Saturday, April 30, 2011 appointment at 1:00 PM. This will give you the opportunity to request revisions and alterations before approving the system design for development. We are available before the meeting to answer any questions you may have.

System Design Document

Automatic Contractual Payment System

for

Bank of Xanadu

Bellevue, Washington

Prepared 04/29/2011

Business Systems Solutions and Development

Bruce Norman Donald Voges Fatma Sidibeh Marwan Beshir Stuart Crome

Table of Contents

Introduction

Business Systems Solutions and Development (BSSD) follows the Systems Development Life Cycle (SDLC) model. To satisfy the requirements of the Planning and Analysis Phases, BSSD has previously submitted a Preliminary Investigation Report and Systems Requirement Document to Bank of Xanadu for approval and progression to subsequent phases of the SDLC.

The Design Phase of the System Development Life Cycle creates a blueprint of the proposed system using the information requirements that were identified in the Analysis Phase. The System Design Document is the second deliverable in the Design Phase and it will provide an in depth review that establishes how the system will operate by documenting its overall architecture, information requirements, navigation, inputs, outputs, procedures, and design and coding standards.

Defining the architecture of the system in the Design Phase includes identifying the proposed system's hardware, software, and network environment. Determining nonfunctional requirements that the system must have, such as performance, security, and usability, allows the designer to select which architecture will perform best for the system. Hardware and software considerations can then be refined to choose those that will best support the system architecture.

The information requirements of the system were identified in the Systems Requirement Document that was presented to Bank of Xanadu on March 11, 2011. Based upon the information requirements from the Analysis Phase, BSSD has created a data model that represents the flow of the system's processes. The data model includes the elements of the system, how they relate to each other, and what information about them is maintained for later use. These elements, defined as entities, and their related information (attributes) have been normalized and documented in an Entity Relationship Diagram (ERD). (*see Appendix A*) The ERD identifies the relationships of the entities using crow's foot notation, which can also be used to communicate business rules. These entities and their attributes are further defined in the metadata dictionary. (*see Appendix B*)

The user interface design creates a physical means which allows the user to interact with the system to navigate, input data, and generate information outputs. Navigation screens allow the user to give instructions to the system, such as which page or form to go to, while forms allow the user to input data into the system. Reports allow the user to pull information from the system based on queries and organize the results in a defined manner or output. The user interface will have an effective layout that is consistent, aesthetically appealing,

and is easy to use. This will be accomplished by using design standards (*see Appendix I*) throughout the interface, such as interface templates, objects, actions, and icons. Consistent formatting and thoughtfully applied layouts and navigation will assist in reducing the amount of time and effort it takes a user in performing their tasks.

System inputs (*see Appendix D*) will include any documentation that data is taken from and entered into the system. These can include invoices, contracts, scanned documents, and vendor or employee documentation. Inputs can be entered into the system using text, numbers, or selection boxes such as radio buttons, check boxes, and dropdown lists. Inputs will use data integrity constraints to ensure accurate entry into the system. System outputs (*see Appendix E*) will include the reports and memos that are produced for the project managers, accounting department, buyers, and vendors.

Procedures such as use cases (*see Appendix G*) define activities performed by the system to create an output. The use cases for the system were identified in the Systems Requirement Document. They are used to create a detailed description of the processes in the system, what triggers them, the flow of events when they are triggered, and any exceptions or conditions that must be met before and after the event is triggered. Some procedures such as functions, pseudo code, and SQL queries may be identified prior to development of the of the system.

Architecture and Design Considerations

Architecture

The system will have a Client/Server architecture using a single database. We are planning to have the database and the system applications running on a network server. In our System Requirements Document, we proposed that we begin designing an automated payment system using Microsoft Access. We will be creating an ASP.NET Web application for user interaction and navigation within the system.

Assumptions

We have made the following assumptions as a basis for the design:

- **The system will be used locally.** The system will be used at the Bellevue branch of the Bank of Xanadu. The system will have one primary user plus several others to help with data inputs as needed. Since the system will have a limited number of users, scalability and volume of transaction considerations are minimal for this project.

- **Development tools will be available.** Development tools for the system will be stored on the server. For an ASP.NET application we will be using one of the Microsoft Visual Studio editions. The Visual Web Developer Express edition is available for a free download on the Web.

- **Enhancements can be added using the Visual Studio Integrated Development Environment (IDE).** There are resources for adding functionality or enhancements through the IDE without having to make major changes to the system infrastructure. The development environment will provide access to the database tables, form markups, and the application functions with their associated variables. Expansion of the system is also possible using this environment.

- **No new hired positions will be necessary.** The bulk of the programming work can be divided among two members of our team, with one member primarily focusing on database construction and functionality, while the other team member will focus on the Web application side of the project. We will be gathering information from programming colleagues and from information that is available online for clarification on technical issues.

Additional team members may be added to the programming work as needed.

- **The system will have multi-user capability**. The client/server architecture is ideal for multi-user applications. Transaction processing using ActiveX Data Objects (ADO) can help prevent transaction collisions. This is normally done with the use of transaction locks and rollbacks. ADO is a feature of the development environment.

- **The bank's security requirements can be met.** Security requirements can be met by network user administration using user logins and network firewalls. Users of the system will be given user names and passwords for access. The system will exist behind the bank's intranet firewall. A process for new user access approval should be established.

Information Data Model

The information requirements for the system include contractual terms and limitations, invoice and vendor information, accrual entries, generated reports, and exception memos. This information is required in order to successfully enter and process contracts and invoices to manage the contractual payment system.

The data model defined by BSSD contains ten entities that encompass the primary categories of the information used by the system. Each of the entities represents a table in the database and defines a different subject or object. These entities are populated by attributes that identify pertinent information or relevant characteristics of each entity. The ERD of the data model includes the following entities and their related attributes:

Acct_Employees (Acct_EmpID, Acct_EmpFirstName, Acct_EmpLastName, Acct_UserName, Acct_LoginID)

Bank_Contacts (Bank_ContactID, *Unit_ID*, Contact_FirstName, Contact_LastName, Contact_Title, Contact_Phone)

Buyers (Buyer_ID, Buyer_FirstName, Buyer_LastName, Buyer_Phone)

Contracts (Contract_ID, *Vendor_ID*, *Acct_EmpID*, *Buyer_ID*, *Unit_ID*, *Bank_ContactID*, *Programmer_ID*, Contract_StartDate, Contract_EndDate, Contract_HourlyRate, Contract_FeeMax, Contract_Description, Contract_Date)

Divisions (Division_ID, Division_Name)

Invoices (Invoice_ID, *Vendor_ID*, *Programmer_ID*, *Acct_EmpID*, *Contract_ID*, Invoice_Date, Invoice_StartDate, Invoice_EndDate, Invoice_HourlyRate, Invoice_Description, Invoice_Hours, Invoice_Total, Invoice_Terms, Invoice_GLAcct, Invoice_RecDate, Invoice_Accrued)

Memos (Memo_ID, *Invoice_ID*, *Acct_EmpID*, *Contract_ID*, Memo_Date, Memo_Type)

Programmers (Programmer_ID, *Vendor_ID*, Programmer_FirstName, Programmer_LastName, Programmer_Phone)

Units (Unit_ID, *Division_ID*, Unit_Name, Unit_Branch)

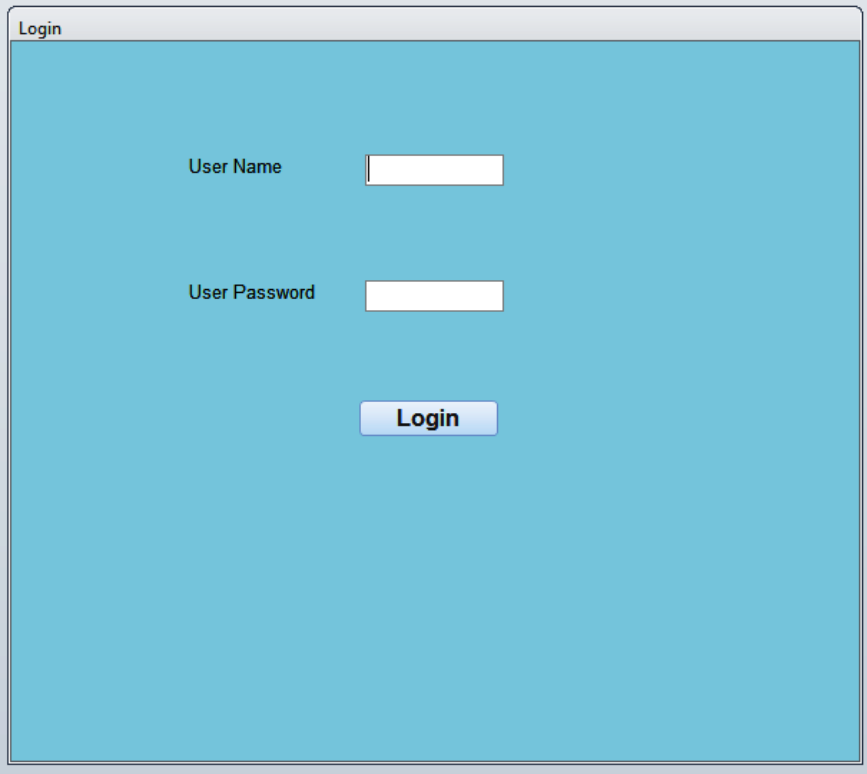
Vendors (Vendor_ID, Vendor_Name, Vendor_Address, Vendor_City, Vendor_State, Vendor_Zip, Vendor_Phone, Vendor_Email, Vendor_Contact, Vendor_Title)

The Entity Relationship Diagram and Metadata Dictionary in the appendices describe these entities, attributes, and their relationships in greater detail.

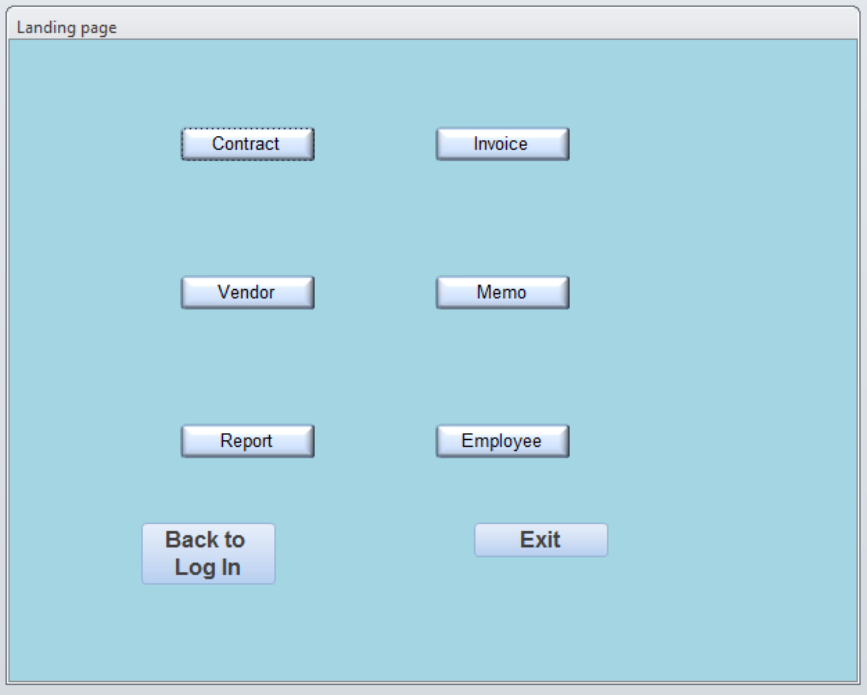
User navigation Design

BSSD has created simple examples of the proposed navigation screens to identify the information that will be required for navigation of the system and to give an example of how the input fields may appear on the screen. These are examples only, and may be revised by BSSD based upon feedback from relevant Bank of Xanadu employees.

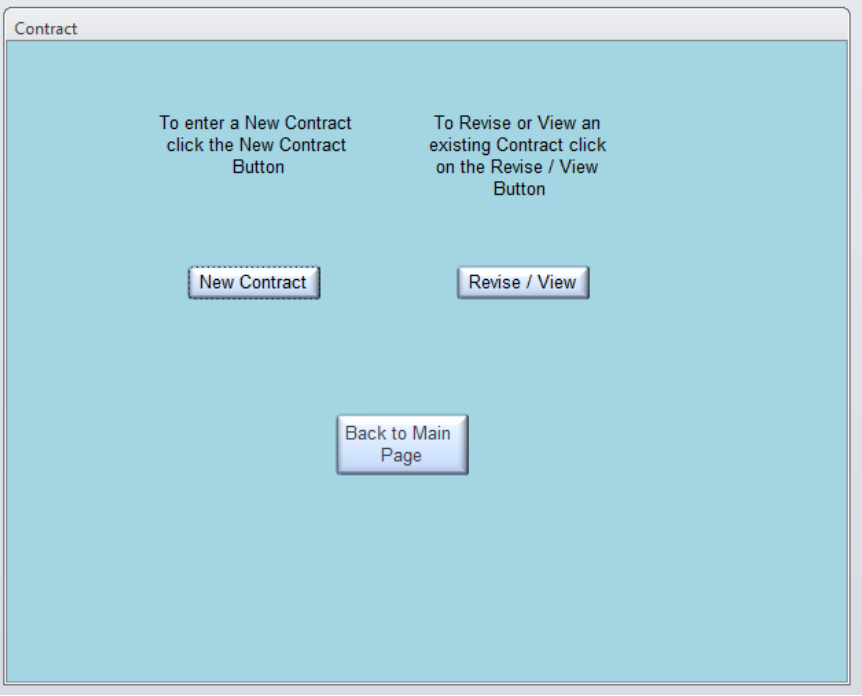
The initial screen will provide fields for employees to enter their user name and password to allow the user access to the system.



Successfully logging in will lead to the main landing page. This page will have six different buttons that provide access to the primary functions of the system. It will also provide the option to go return to the login screen or to exit the system. Choosing one of these selections will lead the user to the next screen in the site's navigation system.

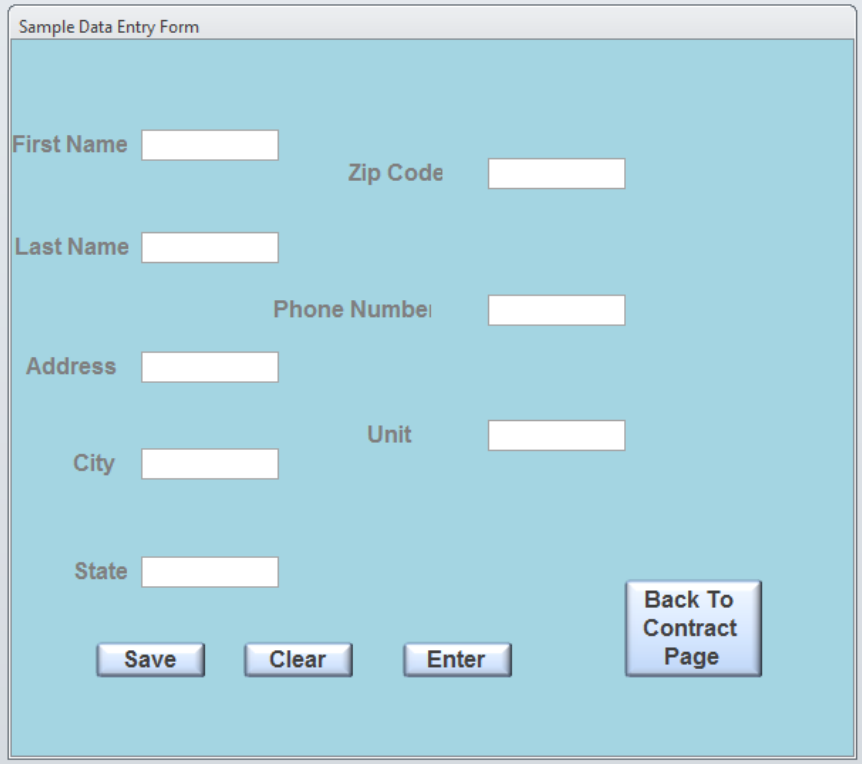


If the user selects the Contract button, it will lead to a screen that will include the following information and options:



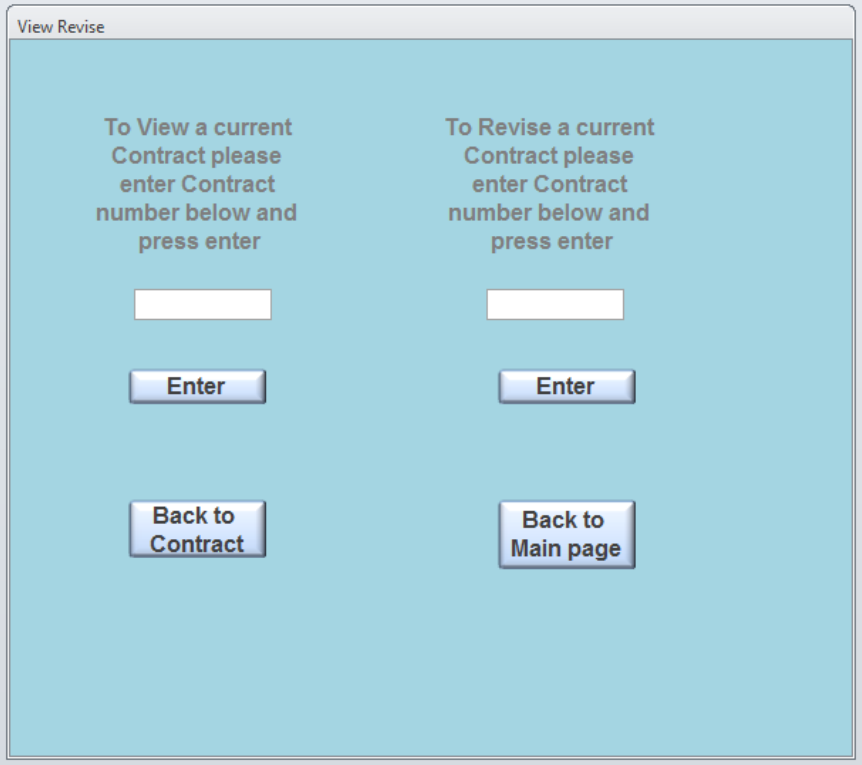
From here the user can go to a data entry screen for a new contract, a revise/view screen, or back to main landing page.

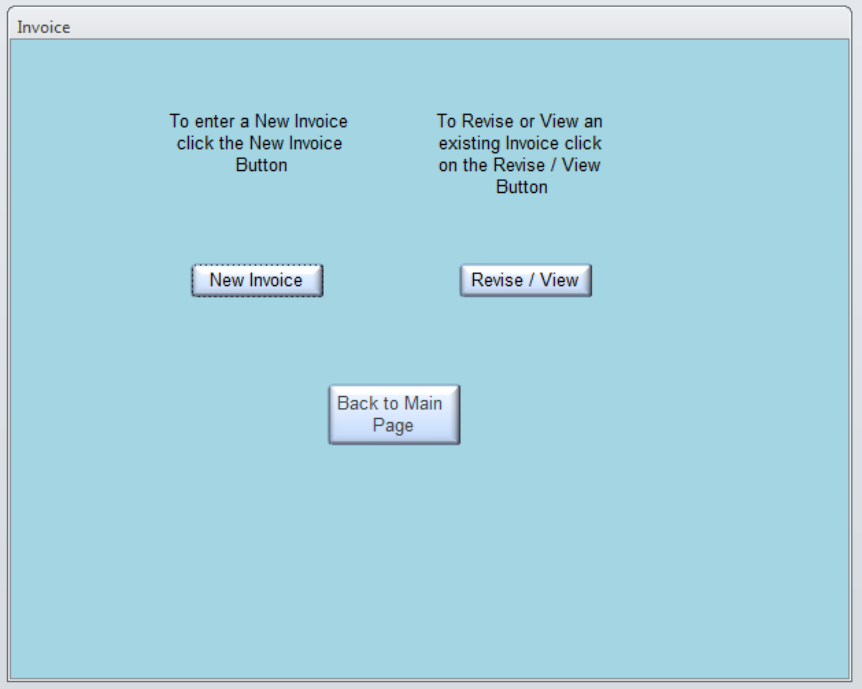
This is a sample data entry screen. Information related to the creation of a new contract will be entered into the system using a form similar to this. There may be additional screens to add further information regarding the creation of new contracts. Navigation to these screens will be placed at the bottom of the screen as required. Once again, this is an example of how the screens may be laid out and may be subject to change.



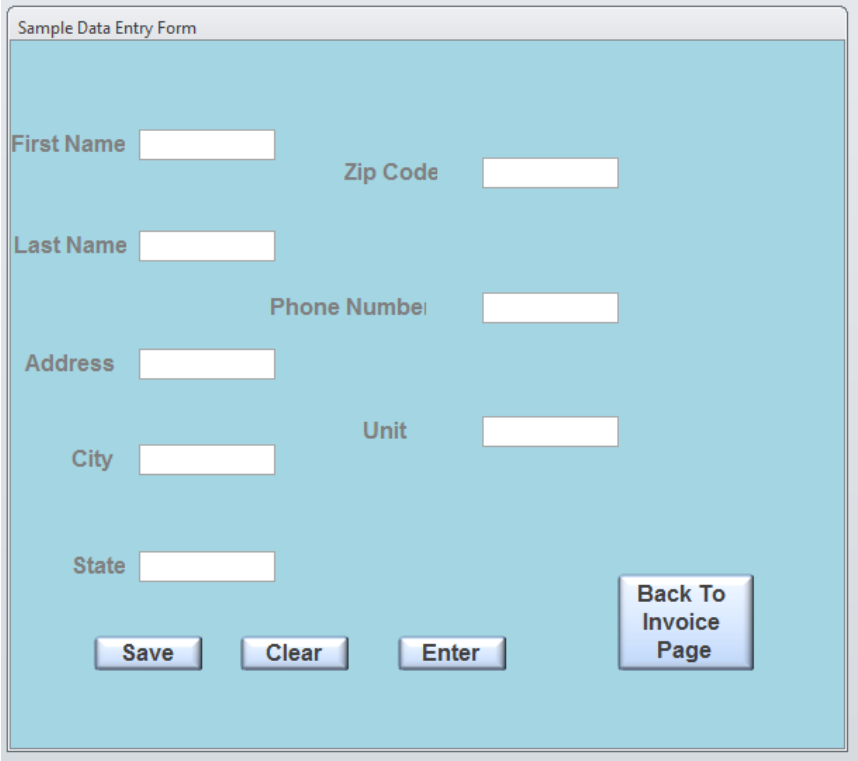
All of the screens will have backwards navigation to their respective options screens, and will allow the user to save or clear the information and to enter it into the system.

To revise or view the information entered for a contract, the user will need to provide a contract number and hit enter to bring up the required contract information. Simple onscreen notes or tooltips will be employed to inform the user of the information required in each field as necessary, or to inform the user what actions must be performed by them. A sample screen for viewing or revising a contract is provided below:

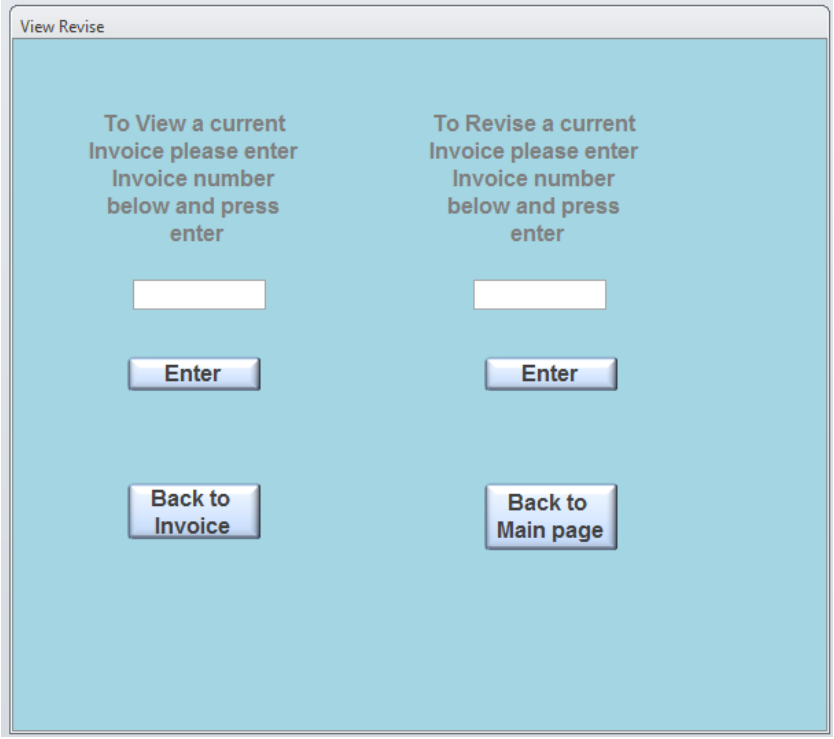


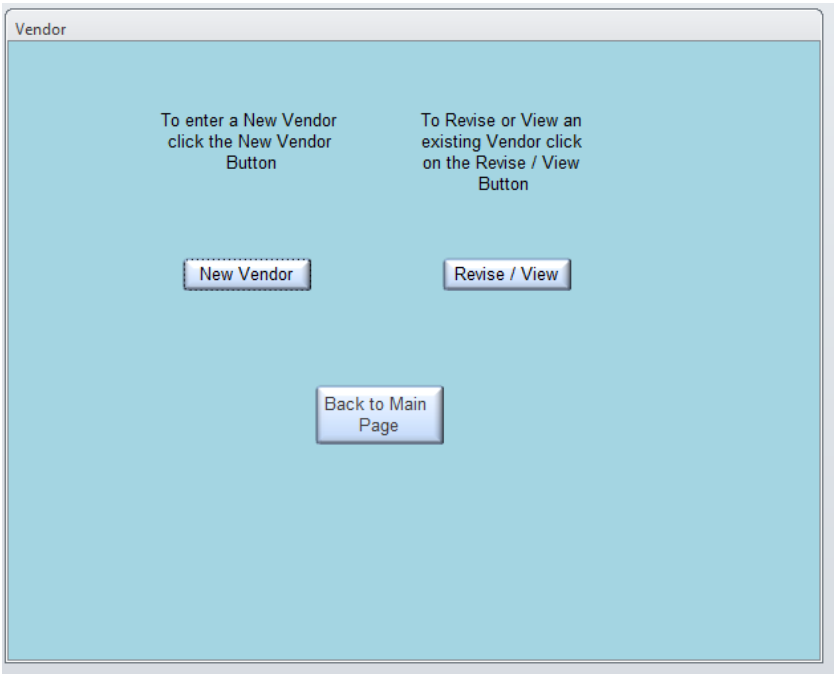


The Invoice screens are very similar to the Contract screen and will follow a standardized layout that includes similar forms and buttons for consistency and ease of navigation.

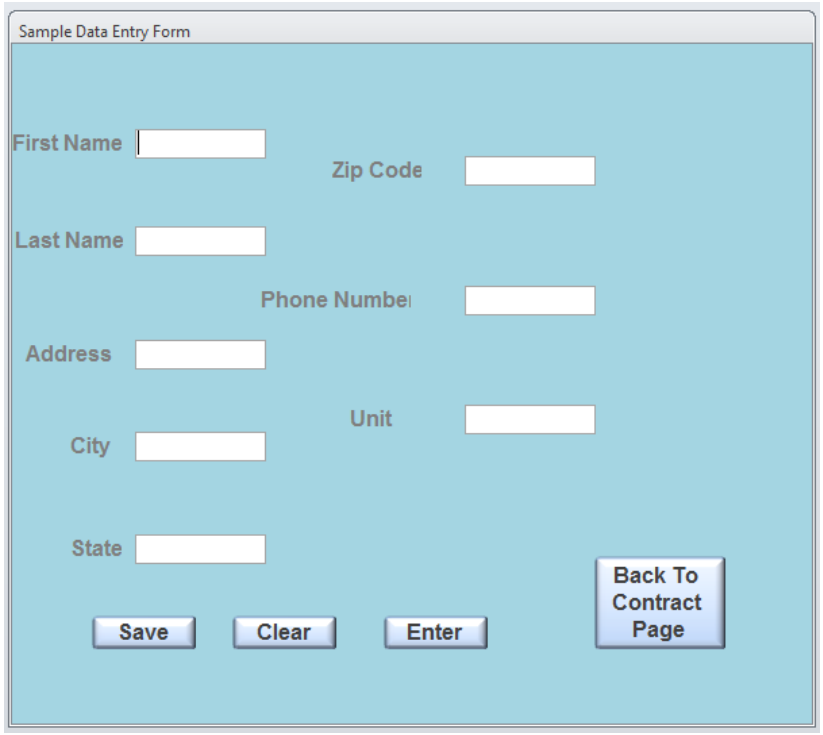


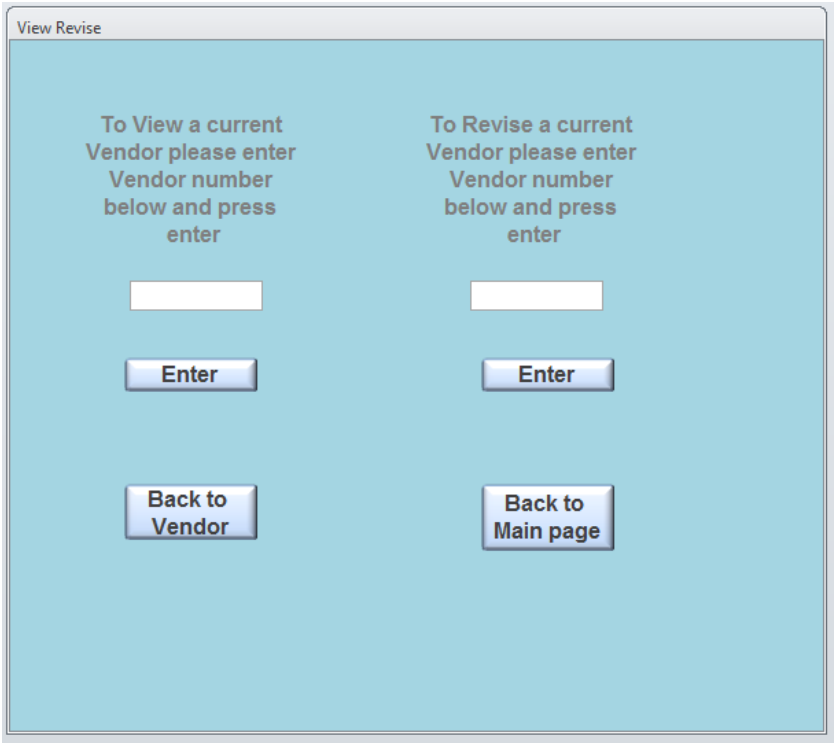
Once again, these are example forms that are subject to change based upon customer requests or suggestions.



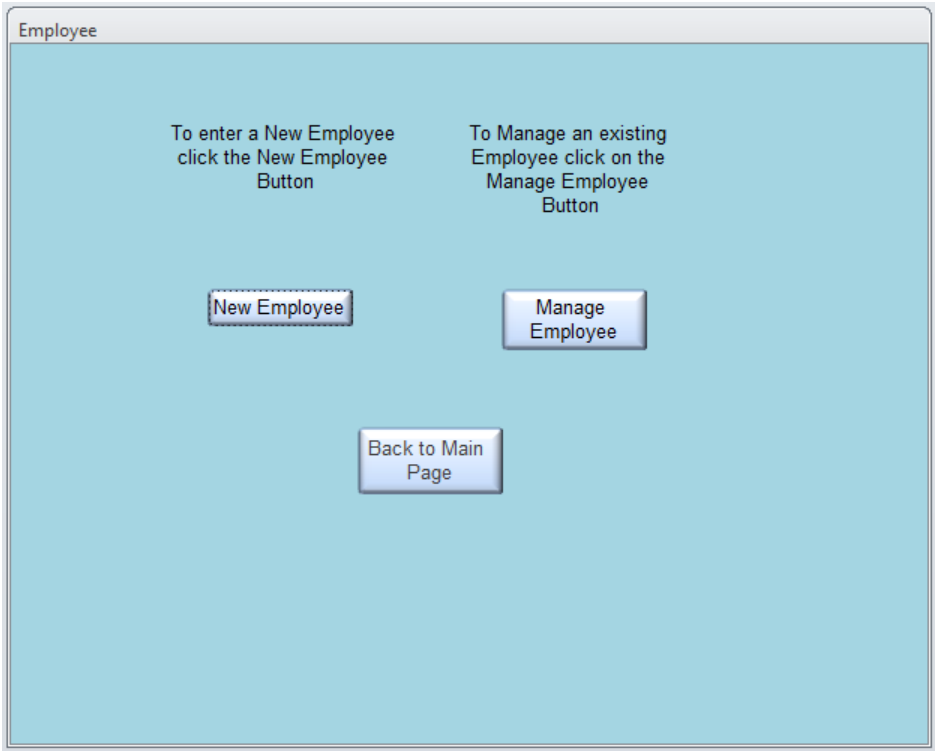


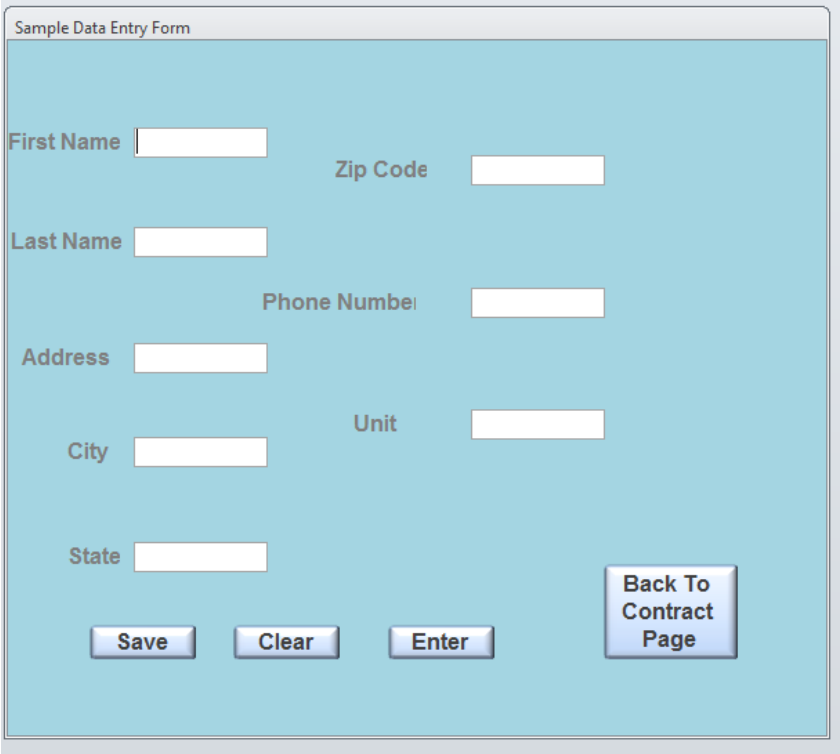
The Vendor screens will also follow the same layout and standards same of the Contract and Invoice screens.



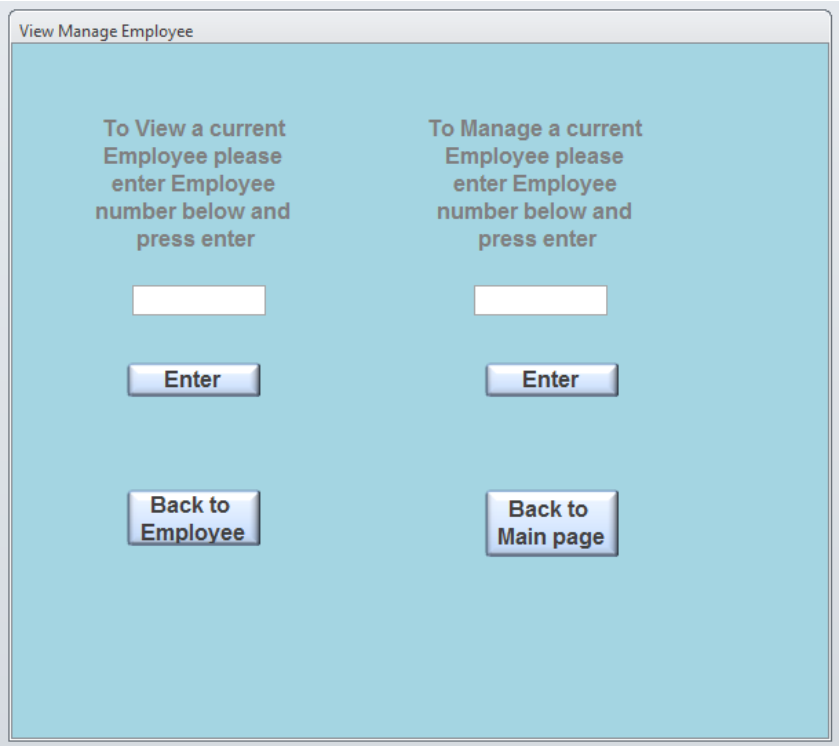


The Employees screens will follow the same layout as the previous screens and will allow the entry or revision of any information related to the employees.

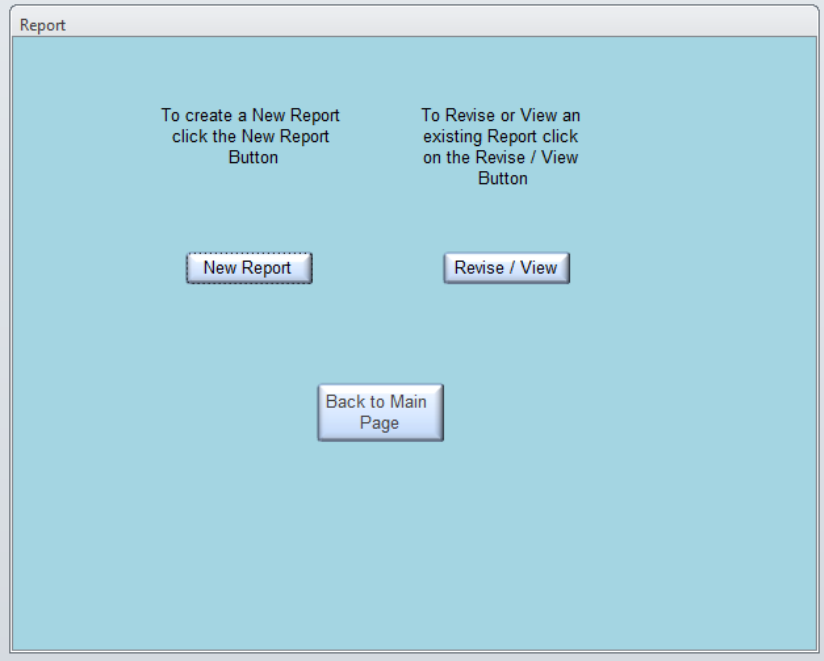


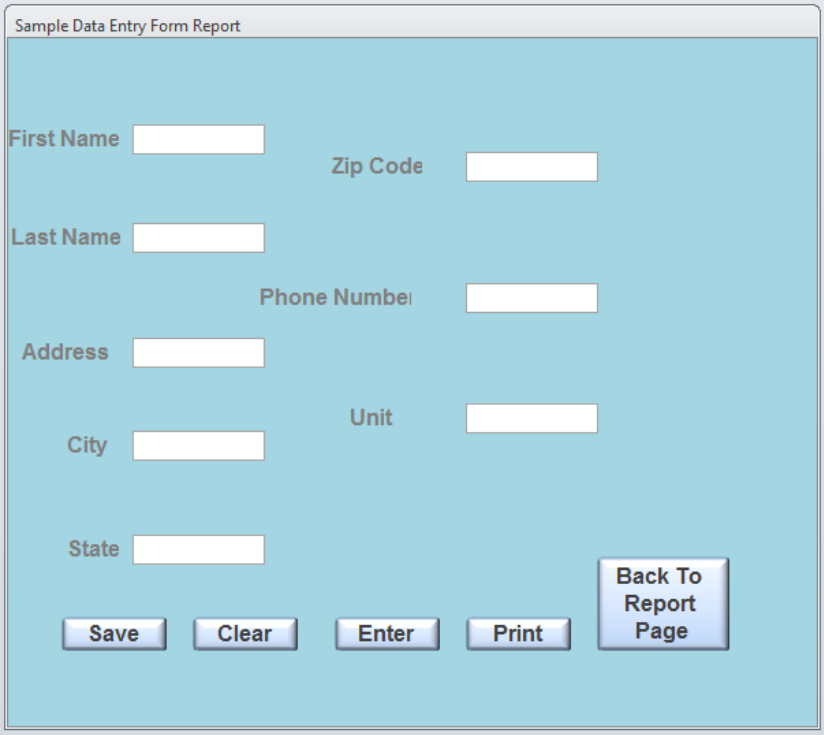


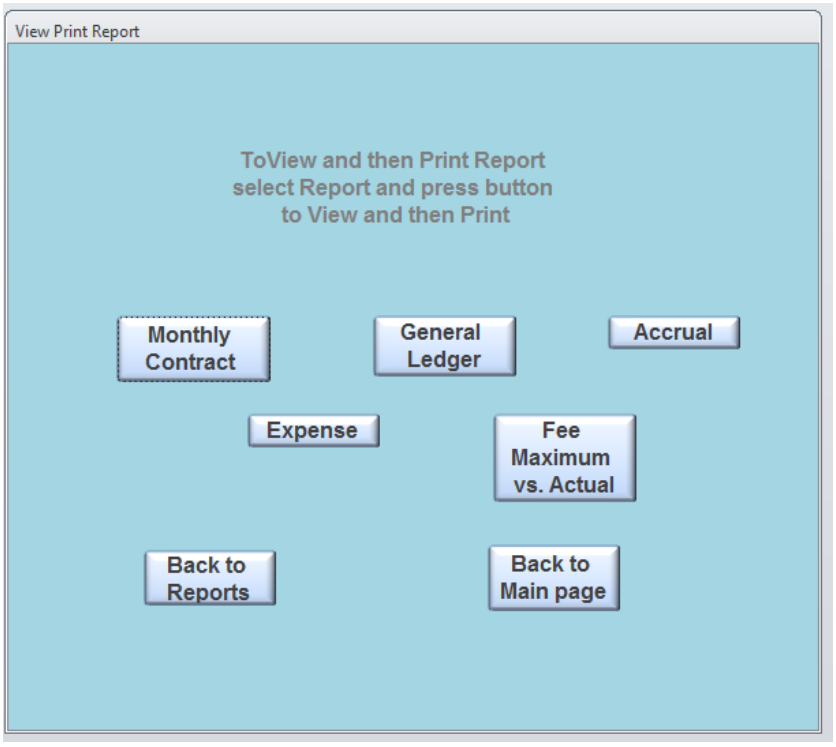
You can manage an employee from this section to change their access permissions for the system, denote a change in positions, etc.



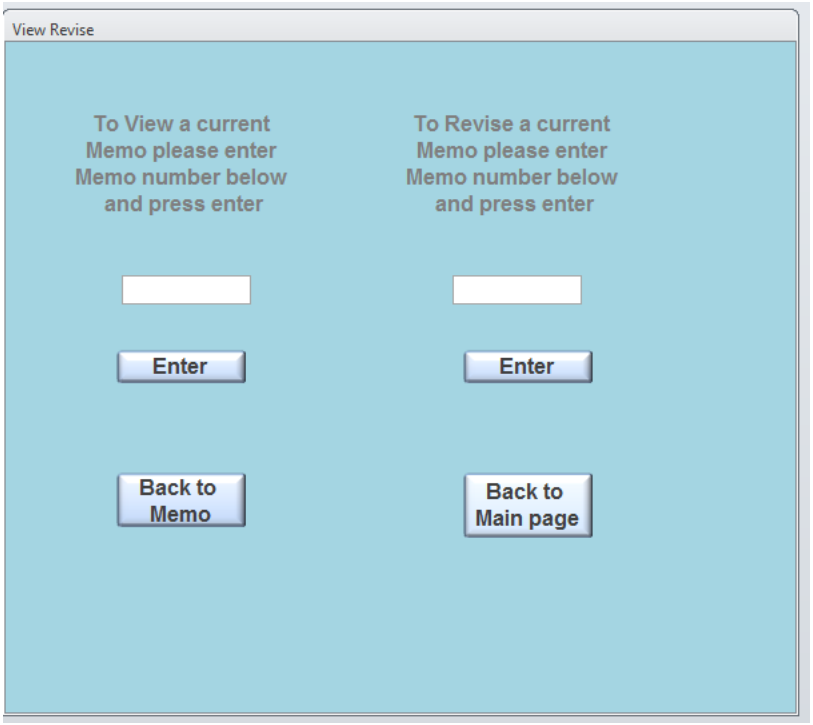
The Reports screen navigation will feature a similar overall design as that of the previous pages and will allow the user an option to select which type of form is to be created, or to view or revise existing reports.







The final screen option is for navigating through the Memo related screens. Again, these will follow the familiar format of previous pages and will allow the creation, viewing, and revision of memos.



Inputs

Bank of Xanadu currently relies upon manual data entry to input or capture the information required for the contractual payment system. This information is gathered from contracts, invoices, contractor timesheets, and related documents that contain required information needed to create the monthly reports produced by the system. These system inputs include:

- **Contracts**
  Contracts are agreements between Bank of Xanadu and various vendors who provide a service to the bank. Contracts will include information such as the vendor name, start and end dates, hourly rates, fee maximums, and a brief description of the work to be performed. The process begins when the buyer delivers a new signed contract to the accountant to be input into the system.

- **Invoices**
  Invoices are provided by the vendors or contractors as a bill for services rendered that includes information such as the start and end date, hourly rate, hours worked, payment terms, and vendor ID. The information on the invoices is compared with that entered from the contracts to verify the information is correct.

- **Programmer Timesheet**
  Programmers use timesheets to document the amount of time they provide services for a particular period of time for contracted work. The timesheet is delivered to the accountant for entry and payment. The entered information is compared with contractual terms to ensure accuracy and proper payment.

- **Employee Documentation**
  Employee documents can be used to create, revise, or update any employee related information. This information can include anything from new hires, employee promotions (title changes), a change in name (i.e. marriage/divorce), etc.

Outputs

System outputs are the results of a query, function, or document used to confirm or request information. Outputs of the system can include monthly reports, memos, responses, or data entry sheets. Reports are issued monthly and collect relevant data for Project Managers and Accounting employees that may denote the status of contracts under their direction. Bank of Xanadu currently uses Microsoft Word or Excel to generate reports, memos and invoices. BSSD will provide new forms for the following system outputs:

- **Data Entry Sheet**
  This is used to create a new vendor, bank contact, bank unit, or bank division record.

- **Contract / Invoice Exception Memo**
  This is a memo sent from the accountant to the buyer if there is invalid or incomplete information on a contract or invoice. The buyer is responsible for verifying and correcting the information, which is then returned to the accounting employee to be entered into the system. Contract and invoice errors must be corrected before they can be finalized and entered into the system.

- **Monthly Contract Recap Report**
  This is a monthly report sent to the Project Managers that recaps what has been paid for each contract that they manage.

- **Fee Maximum VS. Actual Report**
  These reports identify fee maximum, what's been paid, and percentage of what's been used.

- **Expense Recap Report**
  This report is sent to the bank units and details the expenses by division and unit.

- **Vendor Inquiry Response**
  This is sent to the vendor in response to an inquiry they made.

- **General Ledger Report**

Balances money charged to expense accounts for contract programmers that accounts payables cut checks for.

- **Accrual Report**
  This monthly report maintains a list of invoices that have been accrued from one month to the next, based upon what date the invoice was received.

Procedures

External

This section is a brief analysis of the approach used to develop an automated system considering the functional requirements, activities, and data required by the user through the processes of the system identified during the Analysis phase. We have used use case diagrams and a use case scenario (*see Appendix F for example)* to describe the functional requirement of this system and now we are using them as a reference for us in the Development phase.

A use case diagram is a graphical description of the major processes of the system that also shows the interaction between the different actors who interact with the system. In addition, the use case diagram describes the relationships between the different use cases.

The use case scenario is a written description for each use case. This approach states the normal flow of activities in each use case. Furthermore, it gives the exceptions for each activity. The use case scenario also includes the pre-conditions, the post conditions, information requirements, assumptions, and business rules needed for the use case.

BSSD has identified the following Use Cases:

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

General Flow of the Activities

Bank of Xanadu has a problem created from a shift in their corporate focus, thus all future programming will be handled by outside contractors working under very specific contractual terms. The bank needs an automated system to handle these contractual payments. Based upon previous interviews and our analysis of Bank of Xanadu;s needs, BSSD has identified the functional requirements of the new system for its design and development. The following is an outline of the system's flow of activities:

The process starts when the accountant receives a contract; he logs onto the system and navigates to the contract entry screen. He then selects the correct vendor. If the vendor doesn't exist, he navigates to the vendor entry screen and creates a new vendor record and returns to the contract entry screen. The accountant then selects the correct project manager (contact). If the project manager is not in the system, he navigates to the contact entry screen and creates a new contact record and returns to the contract entry screen. Next, he will repeat this same process for selecting the charge unit and corresponding bank division, creating new records if they are not in the system. Finally, the accountant enters the contract information, including the programmer(s) name(s), project start and end dates, hourly pay rate, fee maximum amount, and project description. After all required information has been entered into the system, a new contract record is created and the original hard-copy of the contract is filed for future reference.

When the accountant receives the invoice, he logs on to the system and searches for the contract ID which this invoice related to, and after that he compares the invoice details with the contract information and terms, such as: to make sure that the total amount is correct and the invoice period is related to the period which the project supposed to cover. If it is similar to its contract, he records the invoice information on the system, assigns it to the contract and gives it an identification number. But if it is not similar to the contract, the invoice cannot be paid and needs to be returned to the contract group, he determine the exception and send it back to the contract group to update the invoice with the new condition and return it back after the adjustment, and then, he sends it to the A/P group for payment. Finally, The A/P records the payment, write a check and mail it to the vendor.

The Usage of the System

As requested, the new system has to be automated in order to process the contractual information, incoming invoices, contract extensions and accruals in

quickly and effectively. This system should also calculate the maximum fee remaining in the contract. The system must have a database for all the information required for the process. This information may include vendors, programmers, divisions, charge information, problems with the invoices. These data may be stored in several ways discussed later on. This new system should also prepare reports for other stakeholders like the project manager and other division. For example, the reports include information about the programmers monthly expense recap and fee maximum with actual expenses. The new system should detect errors especially data entry errors and invoices problems. The process of the new system may also do some validation and verification according to specific numbers included in the contracts and the incoming invoices. This system is also expected to make an exception memo which includes problem with the invoices.

The new system will automate many of the calculations and functions that are currently being handled by the Excel system, such as processing incoming programming invoices, preparing accruals, determining whether the invoice falls within the time limitations, and calculating whether there is enough funding left on the contract to pay the invoice. The primary user of the new system will be Dave Spencer, who has indicated that he is proficient in the use of Excel's basic functions and has a limited knowledge of Microsoft Access. We feel that with minimal training any current or future users of the proposed system will be able to efficiently manage and maintain its database information. We are confident that from an operational standpoint this is the most viable solution. The new system will save a substantial amount of money in terms of man hours processing payments.

Internal

Internal procedures include any functions, procedures, or SQL queries that can be identified prior to the development of the system. (see Appendix H)

Interface Design and Coding Standards

External

All forms will use an Arial or Verdana font, primarily in the 10-point and 12-point sizes. Additional font sizes may be used, depending on the size of the portion of the form in which the text appears. Report and memo font sizes will be 10-point or larger. The background color of forms and screens will be light blue or gray. The goal will be to have the text contrast with the background in order to create a readable screen without causing visual discomfort to the reader.

Buttons will conform to standard heights (about 26 pixels) except when they contain more than one line of text. The width of buttons will be in accordance with the amount of text on the button. All forms except for the login form will contain a navigation button to the previous form or to the main page. Drop down choice lists will be used for defined choices the user must make.

Tool tips may be added to form elements as an aid to user navigation. The approach will be making sure the user has adequate information and guidance without adding clutter to the interface. Additional design elements and styles may be added to these standards with the approval of BSSD.

Internal

Naming Conventions for Database Objects

The database will use the same naming conventions for code objects and variables.

Naming conventions for specific objects in database:

- "rpt" for reports
- "qry" for queries
- "tbl" for tables
- "mcr" for macros
- "mdl" for modules

Naming conventions for the database:

- “frm” for forms
- “qry” for queries
- “rpt” for reports
- “mnu” for menus
- “txt” for text
- “lbl” for labels
- “lst” for list boxes
- “cbo” for combo boxes
- “opt” for option boxes
- “cmd” for command buttons
- “img” for images
- “fra” for frames
- “lin” for lines

There will also be input forms for all the tables in the data base that are used for maintenance only. These forms will use a standard color on all forms such as light blue. They will be designed with buttons for navigation in the forms that will include a new record, save, clear, next, back, and exit form button. The forms will not have scroll bars or other navigation buttons other than what is listed here unless requested by the client.

All table primary keys will use a unique numbering system for identifying the primary key.

Tables, reports, queries and attributes in the tables will use Acct_Employee (as an example) as the way they are named.

Conclusion

The System Design Document provides a blueprint for the automatic contractual payment system that will be developed in the Implementation Phase of the Systems Development Life Cycle. It establishes the design and development requirements for the system, including architectural considerations, system procedures, and interface design and coding standards. Related resources for this document such as use cases, metadata, an ERD, and sample outputs are located in the Appendix section.

Business Systems Solutions and Development is requesting an appointment to present the Prototype Demonstration/Walkthrough on May 3, 2011 at 1:00 PM. The prototype user interface will be presented to Bank of Xanadu employees to solicit feedback for suggested enhancements or changes and to seek approval to proceed with development of the system in the Implementation Phase of the SDLC.

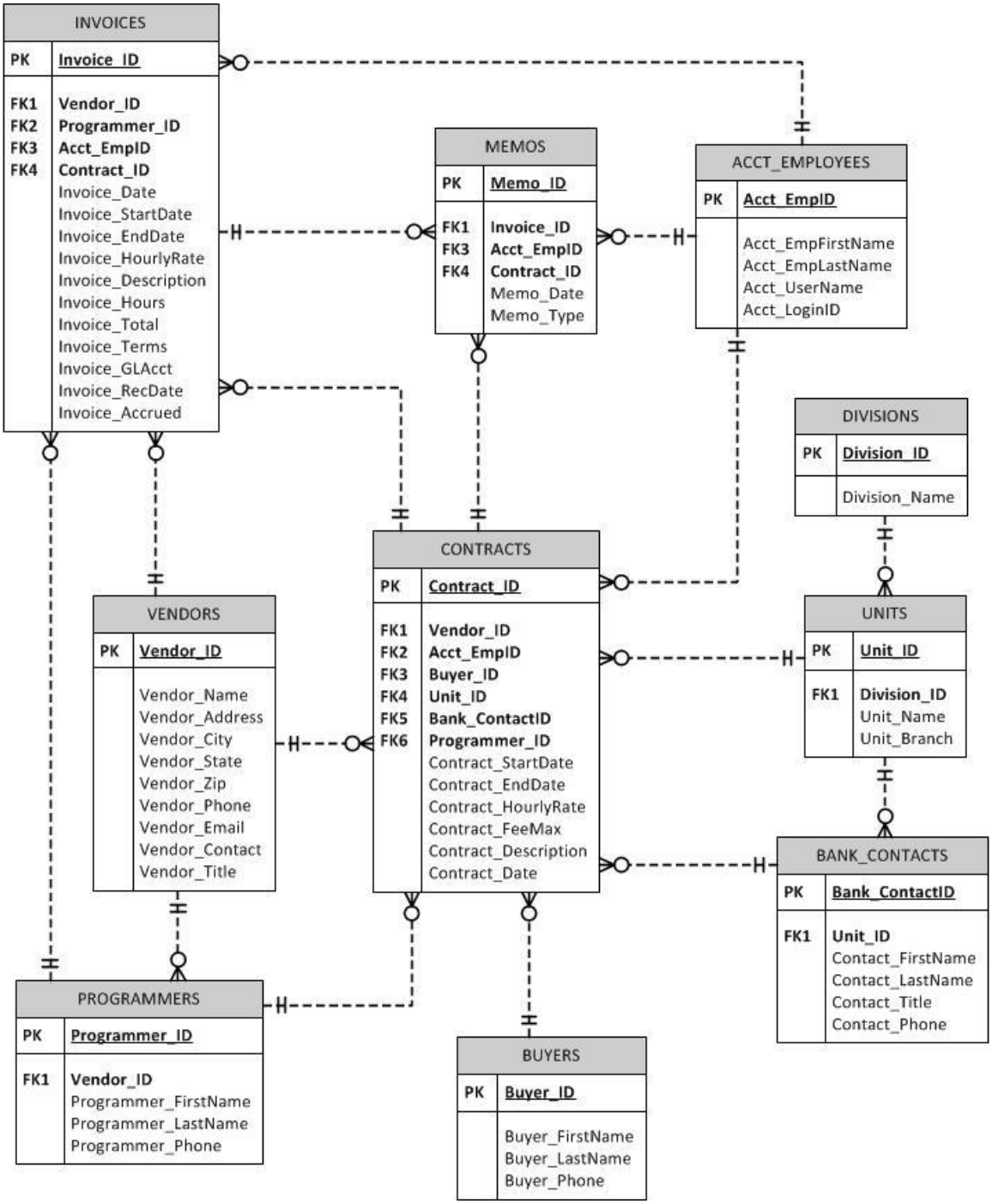
Signature

_____ _____

Patrick Jay, Vice President, Accounting Group Date
Bank of Xanadu

APPENDICES

Appendix A - Entity Relationship Diagram (ERD)



Appendix B - Metadata Dictionary

Entity Type:	**INVOICES**			
Definition:	The invoices are the documents provided by the vendor that serve as a bill for services rendered, products supplied, etc.			
Attribute	**Definition**	**Data Type Constraints**	**Data Value Constraints**	**Referential Constraints**
Invoice_ID	The unique identifier assigned by the company to each individual invoice	Char	Unique, Required	PK, Required
Vendor_ID	The unique identifier assigned by the company to each individual vendor	Char	Unique, Required	FK, Required, FK (Vendors), Required
Programmer_ID	The unique identifier assigned by the company to each individual programmer	Char	Unique, Required	FK, Required, FK (Programmers), Required
Acct_EmpID	The unique identifier assigned by the company to each individual accounting employee	Char	Unique, Required	FK, Required, FK (Acct_Employees), Required
Contract_ID	The unique identifier assigned by the company to each individual contract	Char	Unique, Required	FK, Required, FK (Contracts), Required
Invoice_Date	The date the invoice was created	Date		
Invoice_StartDate	The start date for the term that the invoice is billing	Date		
Invoice_EndDate	The end date for the term that the	Date		

	invoice is billing			
Invoice_HourlyRate	The hourly rate of the individual who has performed work	Numeric		
Invoice_Description	A description of the work performed, service provided, item purchased, etc	Text		
Invoice_Hours	Total hours of work performed being billed for on the invoice	Numeric		
Invoice_Total	The total amount due for the invoice	Numeric		
Invoice_Terms	The terms for payment of the invoice	Char		
Invoice_GLAcct	The general ledger account to which the invoice is billed	Char		
Invoice_RecDate	The date the invoice is received	Date		
Invoice_Accrued	Indicates whether the invoice has an accrual	Boolean		

Entity Type:	**VENDORS**			
Definition:	The vendors are contracted companies who provide goods and/or services to the Bank of Xanadu.			
Attribute	**Definition**	**Data Type Constraints**	**Data Value Constraints**	**Referential Constraints**
Vendor_ID	The unique identifier assigned by the company to each individual vendor	Char	Unique, Required	PK, Required
Vendor_Name	The company name of the vendor	Char		
Vendor_Address	The street address of the vendor	Char		
Vendor_City	The city that the vendor is located	Char		
Vendor_State	The state that the vendor is located	Char		
Vendor_Zip	The zip code that the vendor is located	Char		
Vendor_Phone	The vendor's phone number	Char		
Vendor_Email	The vendor's e-mail	Char		
Vendor_Contact	The name of the vendor employee	Char		
Vendor_Title	The title of the vendor employee	Char		

Entity Type:	**PROGRAMMERS**			
Definition:	The programmers work for vendor companies to provide programming services			
Attribute	**Definition**	**Data Type Constraints**	**Data Value Constraints**	**Referential Constraints**
Programmer_ID	The unique identifier assigned by the company to each individual programmer	Char	Unique, Required	PK, Required
Vendor_ID	The unique identifier assigned by the company to each individual vendor	Char	Unique, Required	FK, Required, FK (Vendors), Required
Programmer_FirstName	The first name of the programmer	Char		
Programmer_LastName	The last name of the programmer	Char		
Programmer_Phone	The phone number for the programmer	Char		

Entity Type:	**ACCT_EMPLOYEES**			
Definition:	Acct_employees are Bank of Xanadu accountants. They are the employees who manage the contractual payments system.			
Attribute	**Definition**	**Data Type Constraints**	**Data Value Constraints**	**Referential Constraints**
Acct_EmpID	The unique identifier assigned by the company to each individual accounting employee	Char	Unique, Required	PK, Required
Acct_EmpFirstName	The first name of the accounting employee	Char		
Acct_EmpLastName	The last name of the accounting employee	Char		
Acct_UserName	The user name of the accounting employee to access the system	Char		
Acct_LoginID	The login password for the accounting employee to access the system	Char		

Entity Type:	**MEMOS**			
Definition:	Memos are notes between personnel who work for Bank of Xanadu. They can be passed between units, divisions, and bank branches.			
Attribute	**Definition**	**Data Type Constraints**	**Data Value Constraints**	**Referential Constraints**
Memo_ID	The unique identifier assigned by the company to each individual memo	Char	Unique, Required	PK, Required
Invoice_ID	The unique identifier assigned by the company to each individual invoice	Char	Unique, Required	FK, Required, FK (Invoices), Required
Acct_EmpID	The unique identifier assigned by the company to each individual accounting employee	Char	Unique, Required	FK, Required, FK (Acct_Employees), Required
Contract_ID	The unique identifier assigned by the company to each individual contract	Char	Unique, Required	FK, Required, FK (Contracts), Required
Memo_Date	The date the memo was created	Date		
Memo_Type	The type of memo (Contract Exception, Invoice Exception, Accounting)	Char		

Entity Type:	**CONTRACTS**			
Definition:	Contracts are agreements between Bank of Xanadu and vendors that define the terms of the agreement, including hourly rates, start and end dates, fee maximum, and a description of the services to be provided.			
Attribute	**Definition**	**Data Type Constraints**	**Data Value Constraints**	**Referential Constraints**
Contract_ID	The unique identifier assigned by the company to each individual contract	Char	Unique, Required	PK, Required
Vendor_ID	The unique identifier assigned by the company to each individual vendor	Char	Unique, Required	FK, Required, FK (Vendors), Required
Acct_EmpID	The unique identifier assigned by the company to each individual accounting employee	Char	Unique, Required	FK, Required, FK (Acct_Employees), Required
Buyer_ID	The unique identifier assigned by the company to each individual buyer	Char	Unique, Required	FK, Required, FK (Buyers), Required
Unit_ID	The unique identifier assigned by the company to each unit	Char	Unique, Required	FK, Required, FK (Units), Required
Bank_ContactID	The unique identifier assigned by the company to each bank contact	Char	Unique, Required	FK, Required, FK (Bank_Contacts), Required
Programmer_ID	The unique identifier assigned by the company to each individual programmer	Char	Unique, Required	FK, Required, FK (Programmers), Required
Contract_StartDate	The date the contract starts	Date		
Contract_EndDate	The date the	Date		

	contract ends			
Contract_HourlyRate	The hourly rate to be paid to the vendor, contractor, etc.	Char		
Contract_FeeMax	The maximum amount that can be charged by the vendor to the contract	Char		
Contract_Description	A description of the work to be performed, service provided, item purchased, etc	Text		
Contract_Date	The date the contract is finalized with all required signatures	Date		

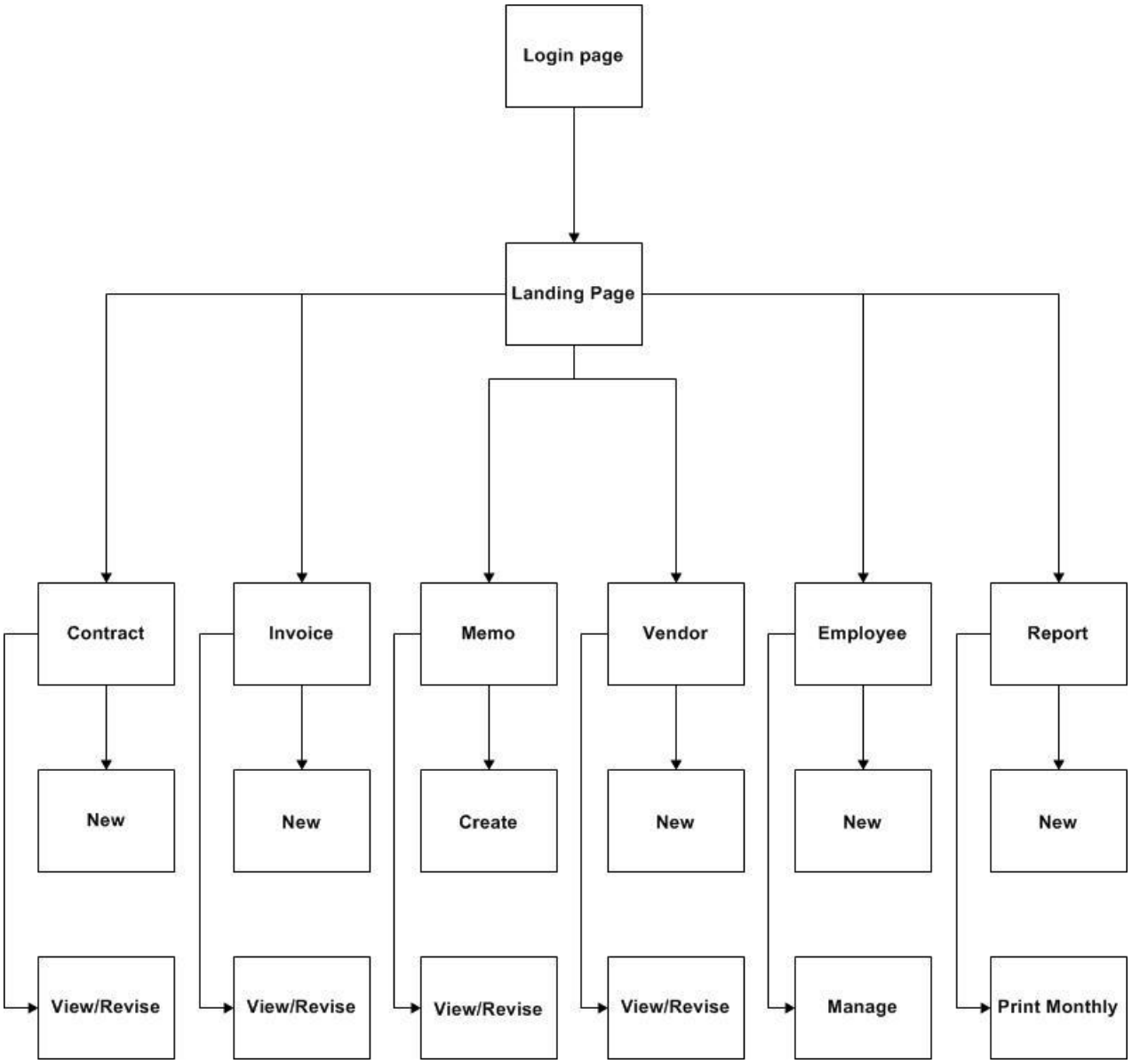
Entity Type:	**DIVISIONS**			
Definition:	Divisions are the major departments of the Bank of Xanadu.			
Attribute	**Definition**	**Data Type Constraints**	**Data Value Constraints**	**Referential Constraints**
Division_ID	The unique identifier assigned by the company to each individual division	Char	Unique, Required	PK, Required
Division_Name	The name of the division	Char		

Entity Type:	**UNITS**			
Definition:	Units are smaller organizations within the Bank of Xanadu hierarchy that are grouped under divisions			
Attribute	**Definition**	**Data Type Constraints**	**Data Value Constraints**	**Referential Constraints**
Unit_ID	The unique identifier assigned by the company to each individual unit	Char	Unique, Required	PK, Required
Division_ID	The unique identifier assigned by the company to each individual division	Char	Unique, Required	FK, Required, FK (Divisions), Required
Unit_Name	The name of the unit	Char		
Unit_Branch	The location of the unit	Char		

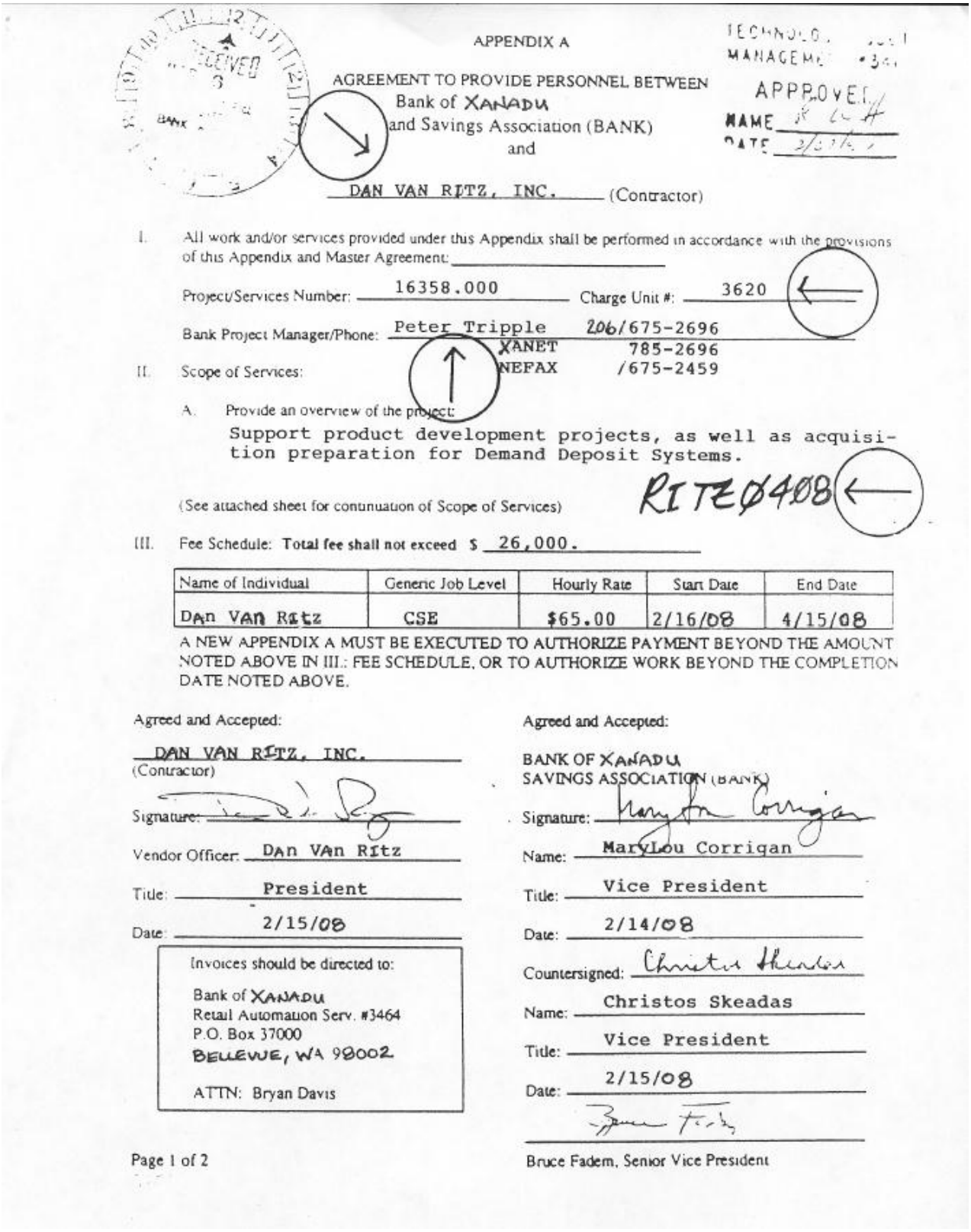
Entity Type:	**BANK_CONTACTS**			
Definition:	Bank contacts are the Bank of Xanadu's employees who are not accountants or buyers. Bank contacts can include project managers, department heads, vice presidents, etc			
Attribute	**Definition**	**Data Type Constraints**	**Data Value Constraints**	**Referential Constraints**
Bank_ContactID	The unique identifier assigned by the company to each individual bank employee who is not a buyer or accounting employee	Char	Unique, Required	PK, Required
Unit_ID	The unique identifier assigned by the company to each individual unit	Char	Unique, Required	FK, Required, FK (Units), Required
Contact_FirstName	The first name the bank contact	Char		
Contact_LastName	The last name of the bank employee	Char		
Contact_Title	The position title of the bank employee	Char		
Contact_Phone	The phone number of the bank employee	Char		

Entity Type:	**BUYERS**			
Definition:	Buyers are Bank of Xanadu employees who negotiate the terms of contracts with vendors and are responsible for clarifying any mistakes or inconsistencies between vendor contracts and invoices received from them.			
Attribute	**Definition**	**Data Type Constraints**	**Data Value Constraints**	**Referential Constraints**
Buyer_ID	The unique identifier assigned by the company to each individual buyer	Char	Unique, Required	PK, Required
Buyer_FirstName	The first name of the buyer	Char		
Buyer_LastName	The last name of the buyer	Char		
Buyer_Phone	The phone number of the buyer	Char		

Appendix C - Navigation Design



Appendix D – System Inputs



APPENDIX A

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

APPROVED
NAME
DATE 2/13/08

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.

RITZ0408

(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.

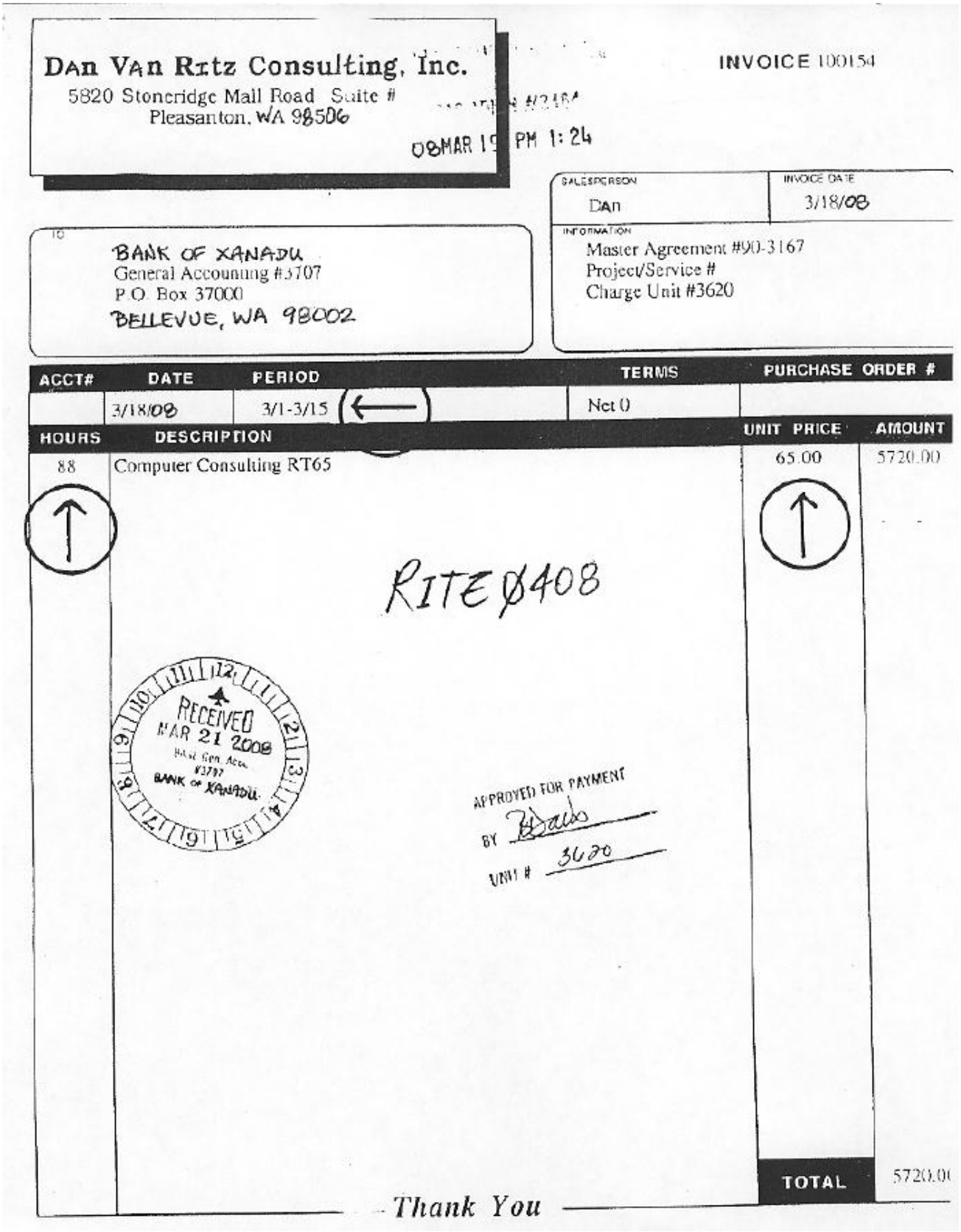
D. List the specific technical expertise required (hardware, operating systems, programming languages, etc.)

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.

Page 2 of 2



Dan Van Rıtz Consulting, Inc.
5820 Stoneridge Mall Road Suite #
Pleasanton, WA 98506

08MAR 19 PM 1:24

INVOICE 100154

SALESPERSON	INVOICE DATE
Dan	3/18/08

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

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

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
Bkg Gen Acct
#3707
BANK OF XANADU

APPROVED FOR PAYMENT
BY
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	

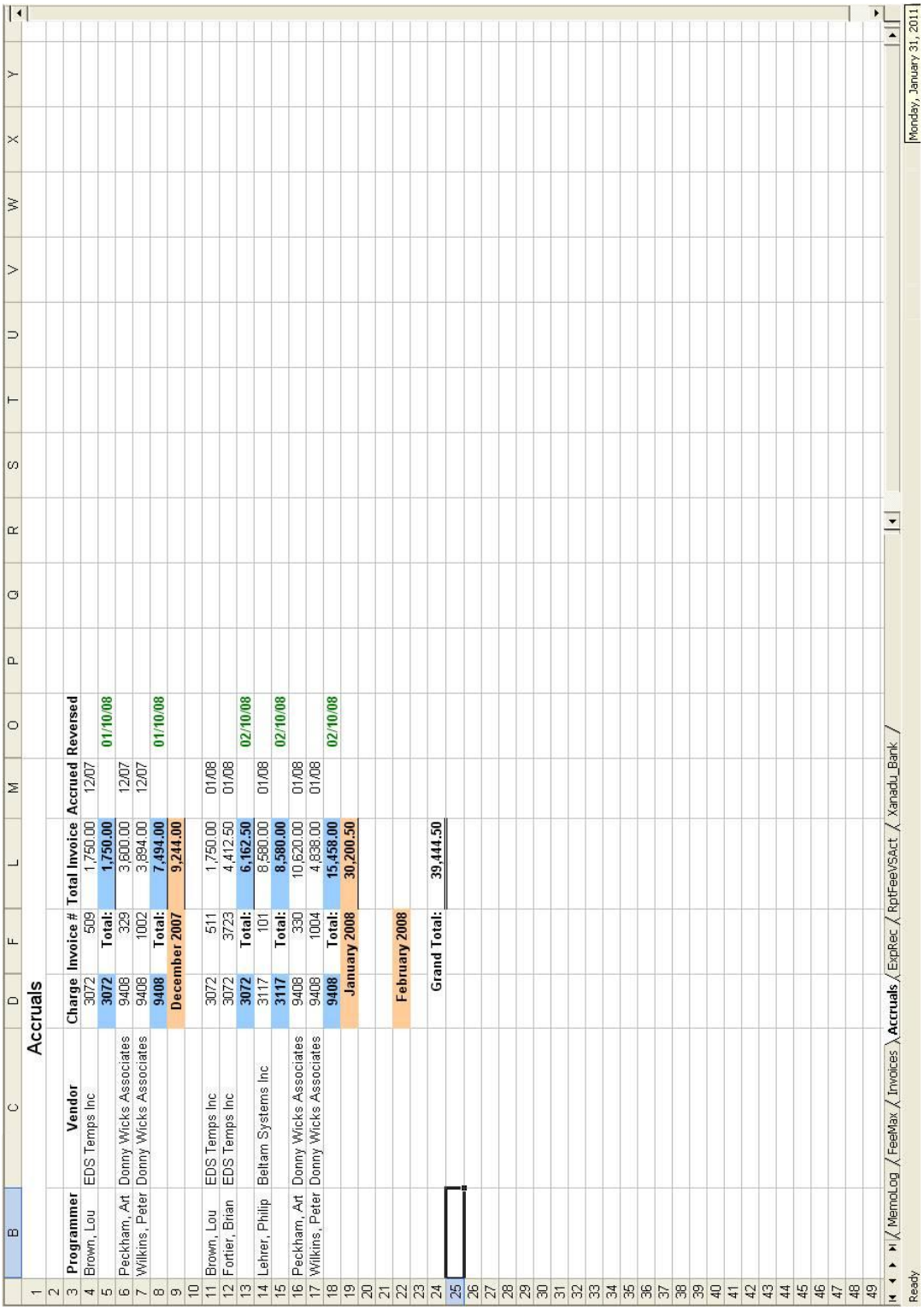
Total Hours: 88

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

Signature Date

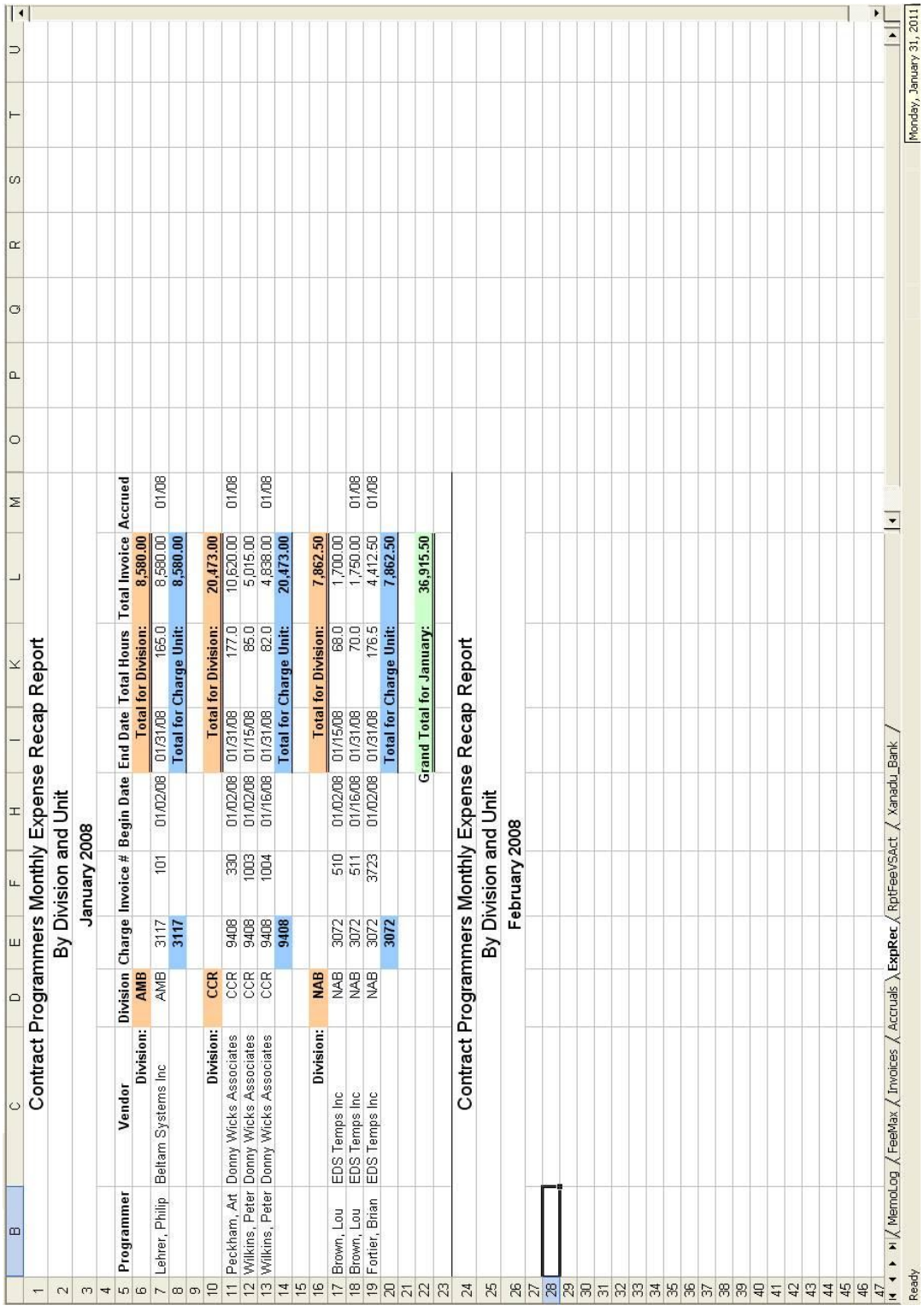
Mail To: Dan Van Ritz Consulting, Inc.

Appendix E – System Outputs



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

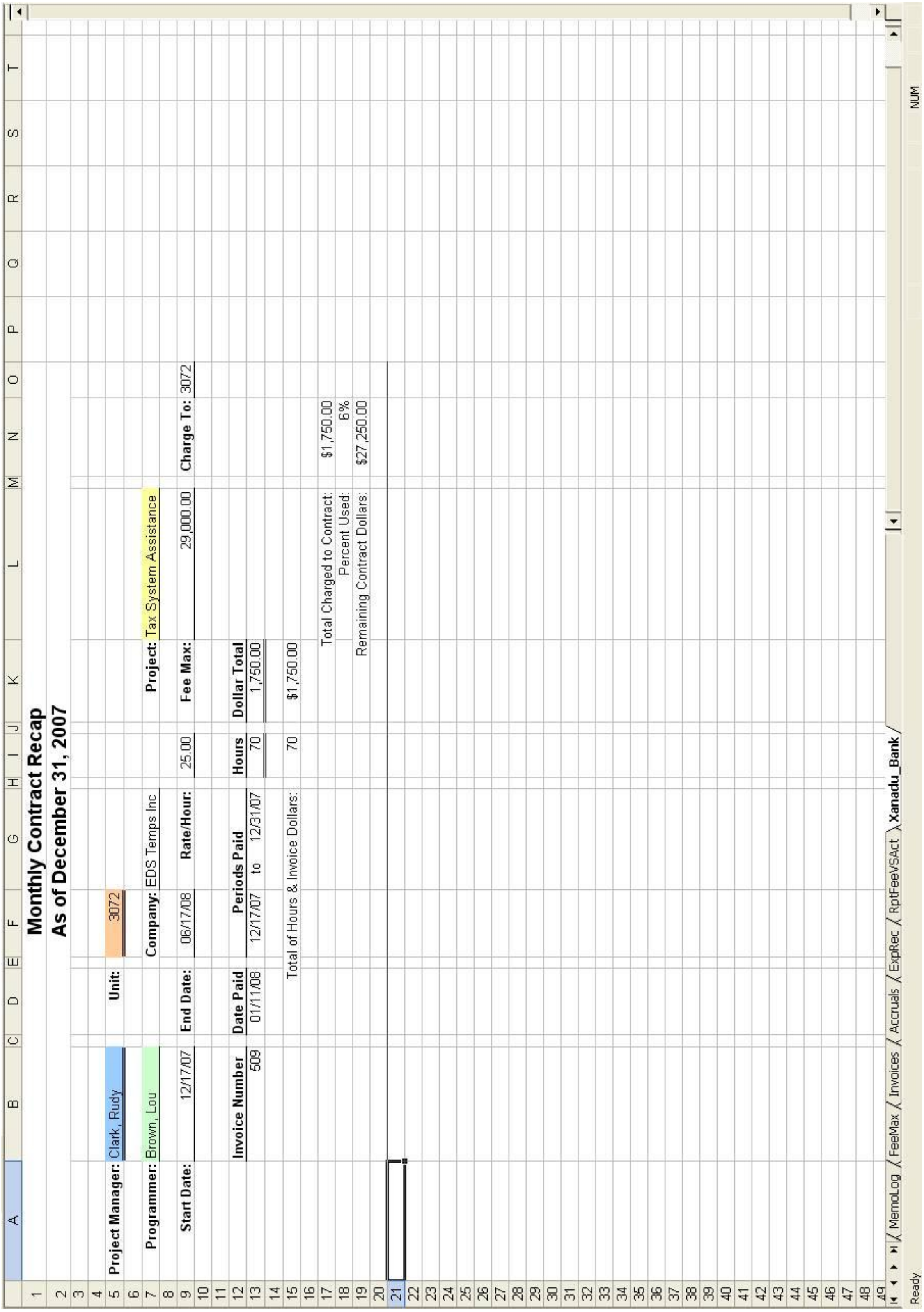
February 2008

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

MemoLog / FeeMax / Invoices / Accruals / ExpRec / RptFeeVSAct / **Xanadu_Bank**

Ready NUM

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.

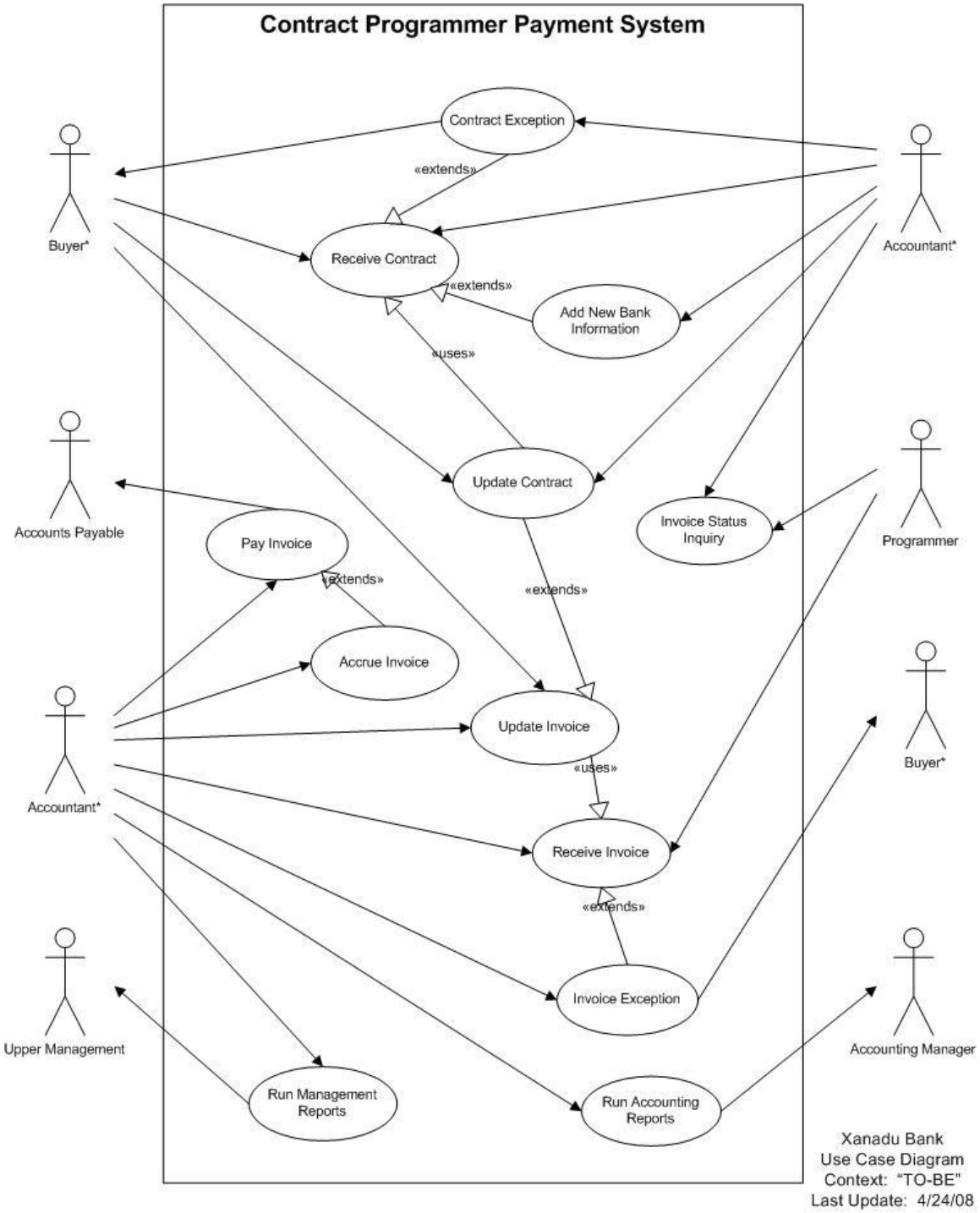
DATE	ACTION

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

Appendix F - Use Case Diagram



Appendix G - Use Case Scenarios

UC001: Receive Contract

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 existence of a new contract delivered 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.

UC002: Add New Bank 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 existence 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.

UC003: Contract Exception

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.	

UC004: Update Contract

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	

UC005: Receive Invoice

USE CASE NAME:	RECEIVE INVOICE	ID: UC005
Primary Actor:	ACCOUNTANT	
Brief Description:	This use case describes the steps for RECEIVE INVOICE, from the time the invoice is delivered to the accountant until the invoice has been validated and entered into the system and filed away.	
Trigger:	A new invoice is delivered by the vendor to the accountant.	
Related Use Cases:	Invoice Exception (extended by); Update Invoice (uses); Pay Invoice (uses); Accruals (uses); Accounting Report (uses); Management Report (uses)	
Normal Flow of Events:	This use case starts when the vendor delivers a new invoice to the accountant. 1) Accountant receives invoice. 2) Accountant creates an ID number for the invoice. 3) Accountant validates the invoice to ensure all information needed is provided. 4) Accountant enters the invoice into the system, including all the required information (see information requirements below). 5) Accountant saves the record. 6) Accountant files hard copy of invoice. This use case ends when an invoice has been entered and saved in the system and has been filed.	
Exceptions:	3) If any required information is missing or incomplete, the accountant will send the invoice to the buyer (see Invoice Exception use case).	
Pre-condition(s):	There must be an existing contract with the vendor who sends the invoice. There must be a new invoice that needs to be entered into the system. The vendor who submits the invoice must be in the system.	
Post-condition(s):	A new invoice record has been created in the system and the original copy has been filed. The invoice has been approved for payment.	
Information Requirements:	ID Number Programmer Vendor Charge Unit Invoice Number Date Paid Begin Date End Date Rate	

	Total Hours Total Invoice Accrued Memo Description of Charges
Assumptions:	1) The vendor will deliver a complete invoice with all the required information. 2) The information on the invoice is valid and accurate. 3) The system will be able to accept all the required information on the invoice. 4) The vendor is already in the system (entered with contract information).
Business Rules:	1) The vendor will deliver the invoice to the accountant. 2) All invoice hard copies are filed for future reference. 3) All invoices must be entered into the system. 4) All invoices are manually assigned a unique ID number. 5) Each invoice can have only one contractor name on it. 6) Each invoice must have all the required information needed (see Information Requirements above). 7) All invoices will be time stamped on their received dates. 8) Invoice Totals will be subtracted from the Fee Maximum. 9) Approval signature and Charge Unit will be included on Invoices approved for payment. 10) Invoice information will be matched to contract stipulations to validate negotiated terms, rates, and limitations.

UC006: Invoice Exception

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	(See Receive Invoice UC005)	

Requirements:	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

UC007: Update Invoice

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

UC008: Invoice Status Inquiry

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

UC009: Pay Invoice

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 accrual must be made so the expense dollars can be charged to the appropriate general ledger account to ensure the expense is realized in the appropriate period.

UC010: Accrue Invoice

Use Case Name:	**ACCRUE INVOICE**	ID: UC010
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps for processing an invoice accrual, from the time the invoice is determined to need to be accrued, until the invoice has been designated as accrued in the system	
Trigger:	The invoice has been received and entered into the system after the cut-off for the last Accounts Payable (A/P) check run 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 payable 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 check run 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.	

UC011: Run Accounting Reports

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 Accounting 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.

UC012: Run Management Reports

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 H - Internal Procedures

Pseudocode is a brief explanation of the programming logic we anticipate will be needed for the system's main functions. Below is pseudocode for the Login page, the Main page, the Contract page and the Invoice page.

Login page

Function: Login button clicked
  If user name and password equal to system user name, system password:
    Open main page.
  Else, User Message: "You have entered an invalid user name and/or password. Please try again."

Main page

Function: Contract button clicked
  Open contract page

Function: Vendor button clicked
  Open Vendor page

Function: Report button clicked
  Open Report page

Function: Invoice button clicked
  Open Invoice page

Function: Memo button clicked
  Open Memo page

Function: Employee button clicked
  Open Employee page

Function: Back to Log In button clicked
  Open/refresh Login page

Contract page

Function: New Contract button clicked
  Open New Contract page

Function: Revise/View button clicked
  Open Existing Contracts page

Function: Back to Main Page button clicked
  Open Main Page

Invoice page

Function: New Invoice button clicked
Open New Invoice page

Function: Revise/View button clicked
Open Existing Invoices page

Function: Back to Main Page button clicked
Open Main Page

Appendix I - Design Standards Document

External

These standards will be defined after meeting with Bank of Xanadu employees on April 30, 2011.

Internal

These standards will be defined after meeting with Bank of Xanadu employees on April 30, 2011.

Appendix J - Issues List

None at this time

Appendix K - Future List

Scaling system for international access

The scope of the current project defined in the “System Requirements Document Automatic Contractual Payment System for Bank of Xanadu, Bellevue, Washington” is for the bank’s Bellevue branch exclusively. In the future, the bank may want to expand the area of the system to additional branches. This would require a more thorough analysis of the multi-user capability and scalability of the system. This analysis is currently not within the scope and timeframe of the project.