

## **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

---

<b>Table of Contents</b> .....	<b>Pg. 3</b>
<b>Introduction</b> .....	<b>Pg. 4</b>
<b>Architecture and Design Considerations</b> .....	<b>Pg. 6</b>
Architecture.....	Pg. 6
Assumptions.....	Pg. 6
<b>Information/Data Model</b> .....	<b>Pg. 8</b>
<b>User Navigation Design (Storyboard)</b> .....	<b>Pg. 10</b>
<b>Inputs</b> .....	<b>Pg. 20</b>
<b>Outputs</b> .....	<b>Pg. 21</b>
<b>Procedures</b> .....	<b>Pg. 23</b>
External Procedures.....	Pg. 23
Internal Procedures.....	Pg. 25
<b>Interface Design and Coding Standards</b> .....	<b>Pg. 26</b>
External (Screens & Navigation).....	Pg. 26
Internal (Code & Database Design).....	Pg. 26
<b>Conclusion</b> .....	<b>Pg. 28</b>
<b>Appendices</b> .....	<b>Pg. 29</b>
Appendix A – Entity Relationship Diagram(ERD).....	Pg. 30
Appendix B – Metadata Dictionary.....	Pg. 31
Appendix C – Navigation Design.....	Pg. 42
Appendix D – System Inputs.....	Pg. 43
Appendix E – System Outputs.....	Pg. 47
Appendix F – Use Case Diagram.....	Pg. 53
Appendix G – Use Cases.....	Pg. 54
Appendix H – Internal Procedures.....	Pg. 75
Appendix I – Design Standards Document.....	Pg. 77
Appendix J – Issues List.....	Pg. 77
Appendix K – Future List.....	Pg. 77

## 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 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)

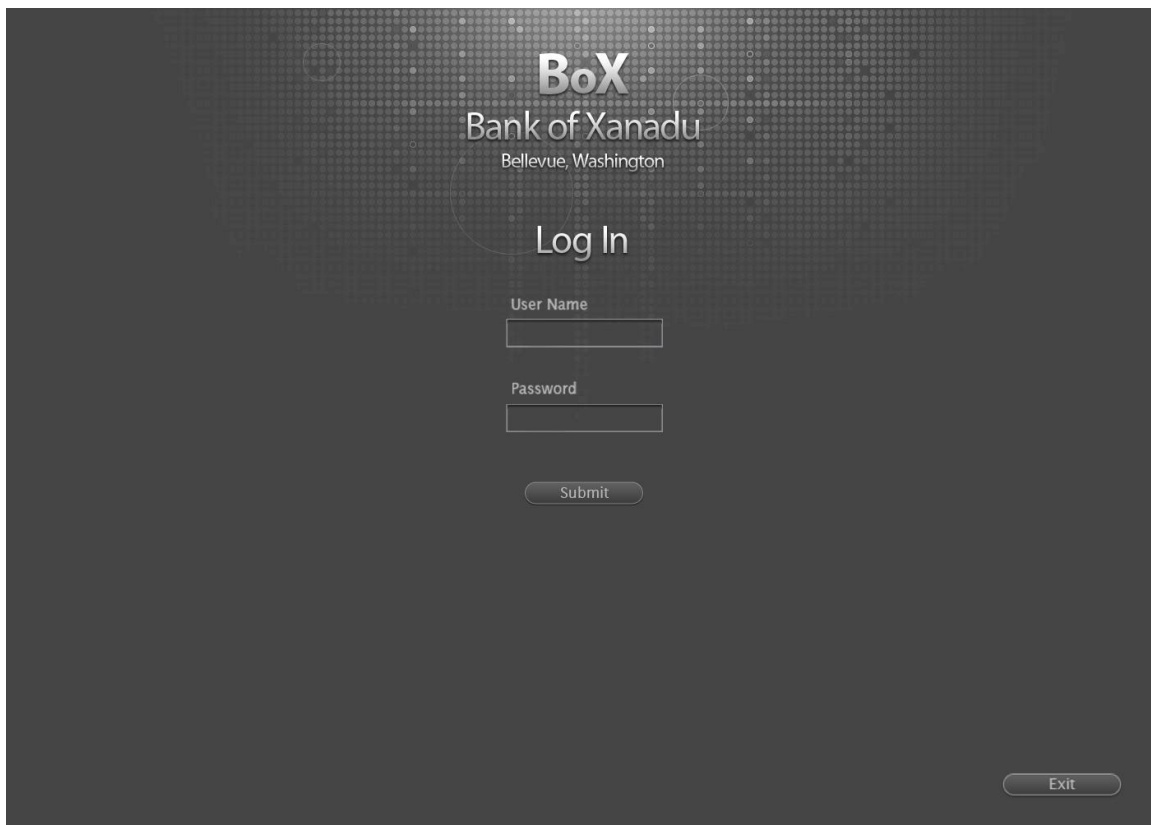
The Entity Relationship Diagram and Metadata Dictionary in the appendices describe these entities, attributes, and their relationships in greater detail.

## 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 login screen (see below) will provide fields for employee to enter their user name and password to allow them access to the system.



The image shows a login screen for the Bank of Xanadu. The background is dark with a grid of small white dots. At the top center, the logo 'BoX' is displayed in a large, bold, white font. Below it, the text 'Bank of Xanadu' is written in a smaller white font, followed by 'Bellevue, Washington' in an even smaller font. In the center, the words 'Log In' are written in a large, white, sans-serif font. Below this, there are two input fields: the first is labeled 'User Name' and the second is labeled 'Password'. Both fields are empty and have a light gray border. Below the password field is a 'Submit' button with a rounded rectangular shape and a light gray background. In the bottom right corner, there is an 'Exit' button with a rounded rectangular shape and a light gray background.

Successfully logging in will lead to the main landing page (see Appendix L, Item 1). This page will have six different buttons that provide access to the primary sections of the system. It will also provide the option to 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.

## From Home Page

---

**Contracts button:** leads to a screen (see Appendix L, Item 2) that includes three buttons/options to create a new contract, revise a contract, or view an existing contract. The navigation screen also includes a button to return to the Home page and another button to exit the system.

**Invoices button:** leads to a screen (see Appendix L, Item 6) that includes three buttons/options to create a new invoice, revise an invoice, or view an existing invoice. The navigation screen also includes a button to return to the Home page and another button to exit the system.

**Reports button:** leads to a screen (see Appendix L, Item 10) that includes three buttons/options to create a new report, revise a report, or view an existing report. The navigation screen also includes a button to return to the Home page and another button to exit the system.

**Memos button:** leads to a screen (see Appendix L, Item 14) that includes three buttons/options to create a new memo, revise a memo, or view an existing memo. The navigation screen also includes a button to return to the Home page and another button to exit the system.

**Vendors button:** leads to a screen (see Appendix L, Item 18) that includes three buttons/options to add a new vendor, revise a vendor's information, or view an existing vendor's information. The navigation screen also includes a button to return to the Home page and another button to exit the system.

**Employees button:** leads to a screen (see Appendix L, Item 22) that includes three buttons/options to create a new employee, revise an employee's information, or view an employee's information. The navigation screen also includes a button to return to the Home page and another button to exit the system.

## From Contracts Home Page

---

**New Contract button:** leads to a screen (see Appendix L, Item 3) with empty fields to enter data from the physical contractual document. There are five additional buttons on this screen as well. The Contract Page button returns the user to the Contract home page. The Home Page button returns the user to the system Home page. The Reset button resets the fields on the page. The Submit button enters the information in the fields into the database. The Exit Button exits the system

**Revise Contract button:** leads to a screen (see Appendix L, Item 4) with empty fields that will be populated with information from the database when the Contract ID is entered. These fields can be revised based upon information received from the Buyer to resolve discrepancies, errors, or revisions in the contractual data. This screen has the same five additional buttons that the New Contract page possesses.

**View Contract button:** leads to a screen (see Appendix L, Item 5) where the user can enter a Contract ID to pull and view its relevant information from the database. This screen also has a Contract Page button to lead back to the Contract home page, a Home Page button to return to the system's Home page, and an Exit button to exit the system.

## From Invoices Home Page

---

**New Invoice button:** leads to a screen (see Appendix L, Item 7) with empty fields to enter data from the received invoices. There are five additional buttons on this screen as well. The Invoice Page button returns the user to the Invoice home page. The Home Page button returns the user to the system Home page. The Reset button resets the fields on the page. The Submit button enters the information in the fields into the database. The Exit Button exits the system

**Revise Invoice button:** leads to a screen (see Appendix L, Item 8) with empty fields that will be populated with information from the database when the Invoice ID is entered. These fields can be revised based upon information received from the Buyer to resolve discrepancies, errors, or revisions in the invoice or contractual data. This screen has the same five additional buttons that the New Invoice page possesses.

**View Invoice button:** leads to a screen (see Appendix L, Item 9) where the user can enter an Invoice ID to pull and view its relevant information from the database. This screen also has an Invoice Page button to lead back to the Invoice home page, a Home Page button to return to the system's Home page, and an Exit button to exit the system.

## From Reports Home Page

---

**New Report button:** leads to a screen (see Appendix, Item 11) that has five separate buttons for each of the different types of reports that can be generated, i.e. Accrual Report, Expense Recap Report, Fee Max vs Actual Report, General Ledger Report, and Monthly Recap Report. Clicking any of these buttons will allow the user to create a new monthly report of that type. This screen also has a Report Page button to return the user to the Report Home page, a Home Page

button to return the user to the system's Home page, and an Exit button to exit the system.

**Revise Report button:** leads to a screen (see Appendix, Item 12) has the same five separate buttons for each of the different types of reports that can be generated, i.e. Accrual Report, Expense Recap Report, Fee Max vs Actual Report, General Ledger Report, and Monthly Recap Report. Clicking any of these buttons will allow the user to revise an existing monthly report of that type. This screen also has a Report Page button to return the user to the Report Home Page, a Home Page button to return the user to the system's Home page, and an Exit button to exit the system.

**View Report button:** leads to a screen (see Appendix L, Item 13) where the user can enter a Report ID to pull and view its relevant information from the database. This screen also has a Report Page button to return the user to the Report Home Page, a Home Page button to return the user to the system's Home page, and an Exit button to exit the system.

## From Memos Home Page

---

**New Memo button:** leads to a screen (see Appendix, Item 15) that has four separate buttons for each of the different types of memos that can be generated, i.e. Contract Exception Memo, Invoice Exception Memo, Accounts Payable Data Entry Sheet, and Vendor Inquiry Response. Clicking any of these buttons will allow the user to create a memo of that type. This screen also has a Memo Page button to return the user to the Memo Home page, a Home Page button to return the user to the system's Home page, and an Exit button to exit the system.

**Revise Memo button:** leads to a screen (see Appendix, Item 16) has the same four separate buttons for each of the different types of memos that can be generated, i.e. Contract Exception Memo, Invoice Exception Memo, Accounts Payable Data Entry Sheet, and Vendor Inquiry Response. Clicking any of these buttons will allow the user to revise an existing memo of that type. This screen also has a memo Page button to return the user to the Memo Home page, a Home Page button to return the user to the system's Home page, and an Exit button to exit the system.

**View Memo button:** leads to a screen (see Appendix L, Item 17) where the user can enter a Memo ID to pull and view its relevant information from the database. This screen also has an Invoice Page button to lead back to the Invoice home page, a Home Page button to return to the system's Home page, and an Exit button to exit the system. This screen also has a memo Page button to return the user to the Memo Home page, a Home Page button to return the user to the system's Home Page, and an Exit button to exit the system.

## From Vendor Records Home Page

---

**New Vendor button:** leads to a screen (see Appendix L, Item 19) with empty fields to enter data about new vendors. There are five additional buttons on this screen as well. The Vendor Page button returns the user to the Vendor Records home page. The Home Page button returns the user to the system's Home page. The Reset button resets the fields on the page. The Submit button enters the information in the fields into the database. The Exit Button exits the system

**Revise Vendor button:** leads to a screen (see Appendix L, Item 20) with empty fields that will be populated with information from the database when the Vendor ID is entered. These fields can be revised to correct discrepancies, errors, or revisions in the vendor data. This screen has the same five additional buttons that the New Vendor page possesses.

**View Vendor button:** leads to a screen (see Appendix L, Item 21) where the user can enter a Vendor ID to pull and view its relevant information from the database. This screen also has a Vendor Page button to lead back to the Vendor Records home page, a Home Page button to return to the system's Home page, and an Exit button to exit the system.

## From Employee Records Home Page

---

**New Employee button:** leads to a screen (see Appendix L, Item 23) with empty fields to enter data about new employees. There are five additional buttons on this screen as well. The Employee Page button returns the user to the Employee Records home page. The Home Page button returns the user to the system's Home page. The Reset button resets the fields on the page. The Submit button enters the information in the fields into the database. The Exit Button exits the system

**Revise Employee button:** leads to a screen (see Appendix L, Item 24) with empty fields that will be populated with information from the database when the Employee ID is entered. These fields can be revised to correct discrepancies, errors, or revisions in the employee data. This screen has the same five additional buttons that the New Employee page possesses.

**View Employee button:** leads to a screen (see Appendix L, Item 25) where the user can enter an Employee ID to pull and view its relevant information from the database. This screen also has an Employee Page button to lead back to the Employee Records home page, a Home Page button to return to the system's Home page, and an Exit button to exit the system.

## 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.

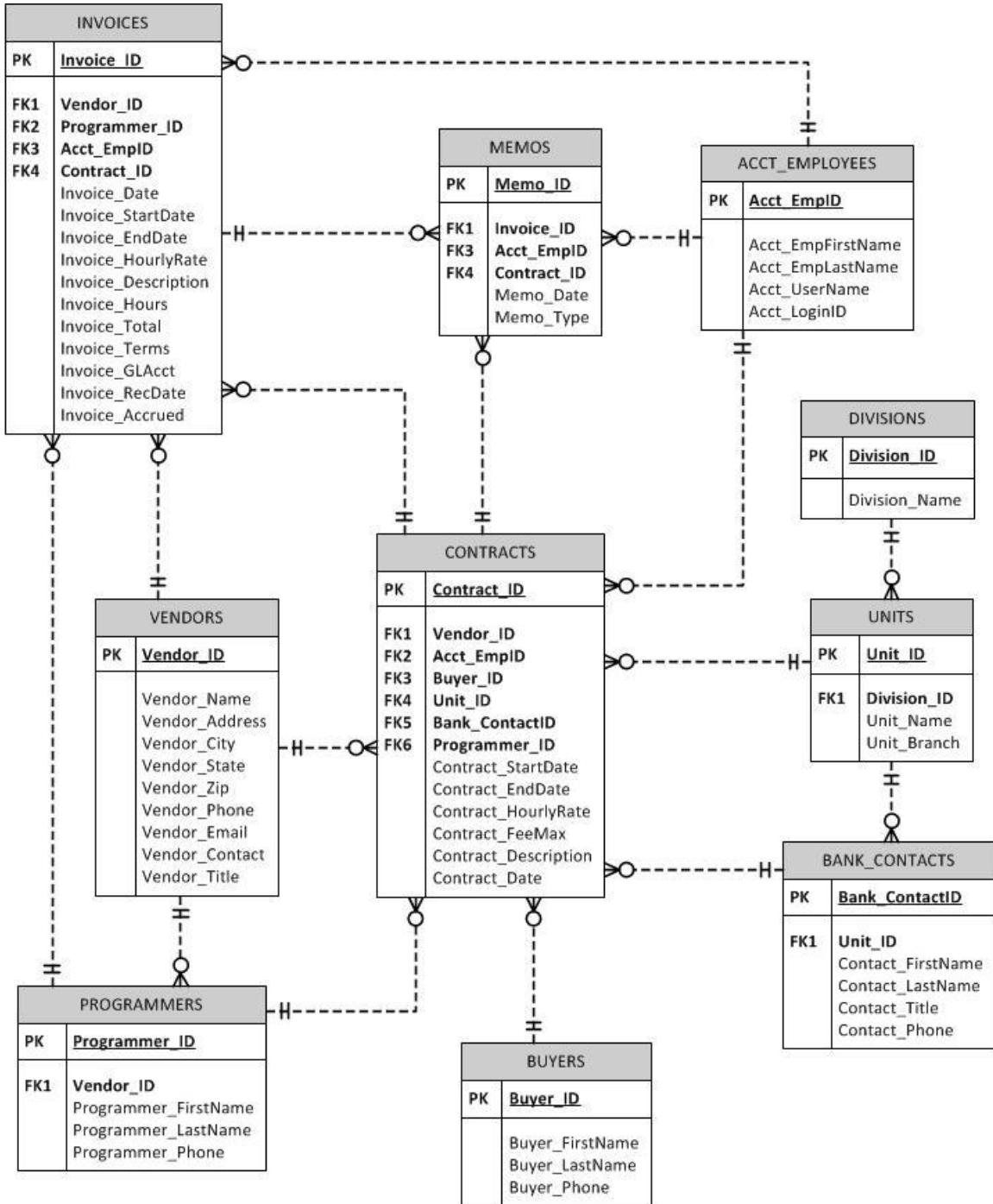
Signature

\_\_\_\_ *Patrick Jay, VP* \_\_\_\_\_ 5/6/11 \_\_\_\_\_  
Patrick Jay, Vice President, Accounting Group      Date  
Bank of Xanadu

## **APPENDICES**



## Appendix A - Entity Relationship Diagram (ERD)



## Appendix B - Metadata Dictionary

Entity Type: <b>INVOICES</b>				
<b>Definition:</b>	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_GLAacct	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		

<b>Entity Type: VENDORS</b>				
<b>Definition:</b>	The vendors are contracted companies who provide goods and/or services to the Bank of Xanadu.			
<b>Attribute</b>	<b>Definition</b>	<b>Data Type Constraints</b>	<b>Data Value Constraints</b>	<b>Referential Constraints</b>
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		

<b>Entity Type: PROGRAMMERS</b>				
<b>Definition:</b>	The programmers work for vendor companies to provide programming services			
<b>Attribute</b>	<b>Definition</b>	<b>Data Type Constraints</b>	<b>Data Value Constraints</b>	<b>Referential Constraints</b>
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		

<b>Entity Type: ACCT_EMPLOYEES</b>				
<b>Definition:</b>	Acct_employees are Bank of Xanadu accountants. They are the employees who manage the contractual payments system.			
<b>Attribute</b>	<b>Definition</b>	<b>Data Type Constraints</b>	<b>Data Value Constraints</b>	<b>Referential Constraints</b>
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		

<b>Entity Type: MEMOS</b>				
<b>Definition:</b>	Memos are notes between personnel who work for Bank of Xanadu. They can be passed between units, divisions, and bank branches.			
<b>Attribute</b>	<b>Definition</b>	<b>Data Type Constraints</b>	<b>Data Value Constraints</b>	<b>Referential Constraints</b>
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		

<b>Entity Type: CONTRACTS</b>				
<b>Definition:</b>	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.			
<b>Attribute</b>	<b>Definition</b>	<b>Data Type Constraints</b>	<b>Data Value Constraints</b>	<b>Referential Constraints</b>
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		

<b>Entity Type: DIVISIONS</b>				
<b>Definition:</b>	Divisions are the major departments of the Bank of Xanadu.			
<b>Attribute</b>	<b>Definition</b>	<b>Data Type Constraints</b>	<b>Data Value Constraints</b>	<b>Referential Constraints</b>
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		

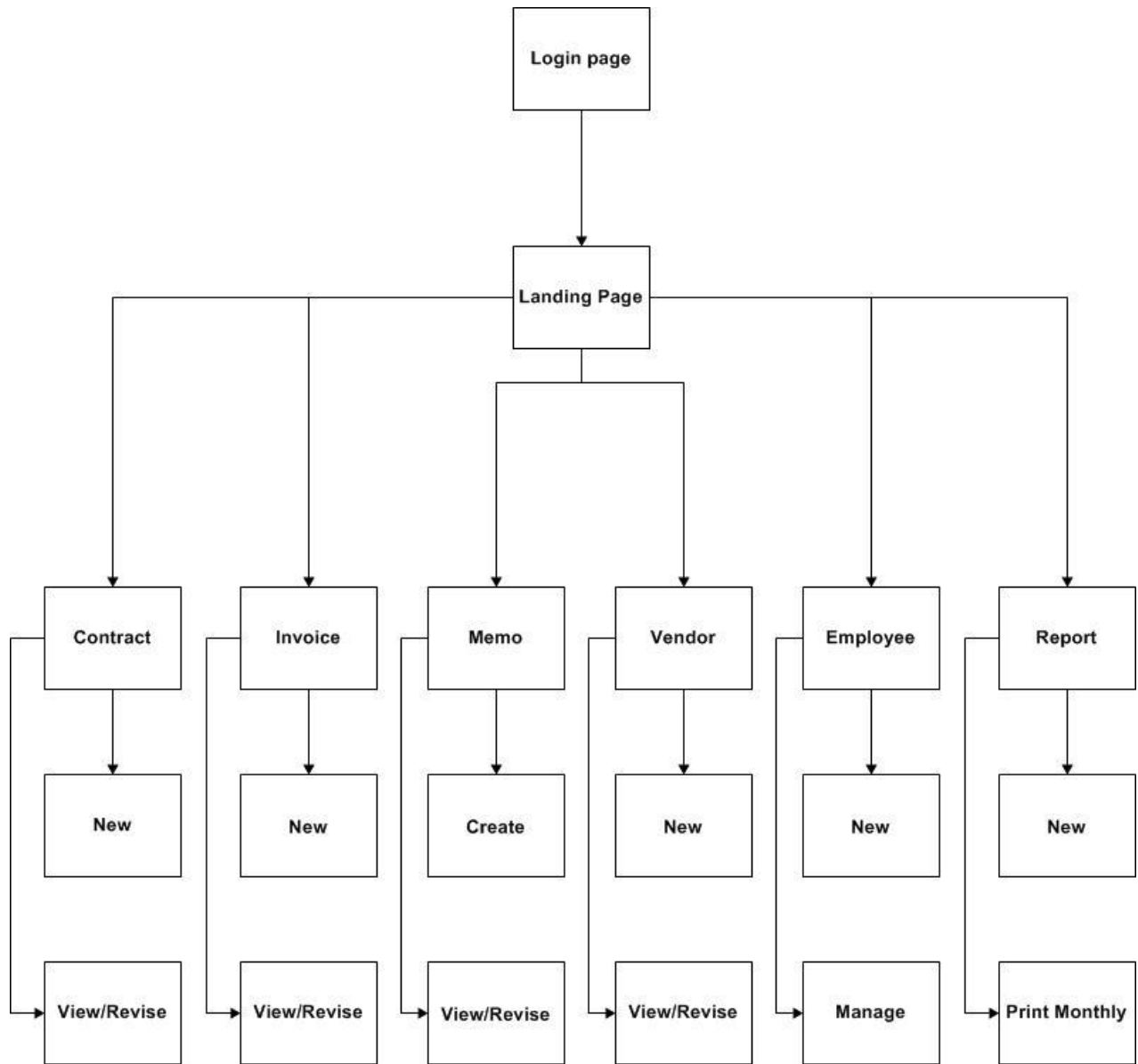
<b>Entity Type: UNITS</b>				
<b>Definition:</b>	Units are smaller organizations within the Bank of Xanadu hierarchy that are grouped under divisions			
<b>Attribute</b>	<b>Definition</b>	<b>Data Type Constraints</b>	<b>Data Value Constraints</b>	<b>Referential Constraints</b>
Unit_ID	The unique identifier 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		

<b>Entity Type: BANK_CONTACTS</b>				
<b>Definition:</b>	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			
<b>Attribute</b>	<b>Definition</b>	<b>Data Type Constraints</b>	<b>Data Value Constraints</b>	<b>Referential Constraints</b>
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		

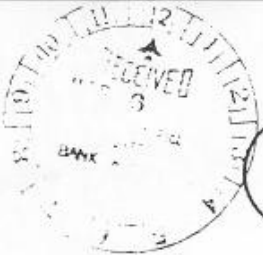
<b>Entity Type: BUYERS</b>				
<b>Definition:</b>	Buyers are Bank of Xanadu employees who negotiate the terms of contracts with vendors and are responsible for clarifying any mistakes or inconsistencies between vendor contracts and invoices received from them.			
<b>Attribute</b>	<b>Definition</b>	<b>Data Type Constraints</b>	<b>Data Value Constraints</b>	<b>Referential Constraints</b>
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)

TECHNOLOGY SUPPORT  
MANAGEMENT #301

APPROVED  
NAME R. W. H.  
DATE 2/15/08

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.**

(See attached sheet for continuation of Scope of Services) **RITE 408**

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

Name of Individual	Generic Job Level	Hourly Rate	Start Date	End Date
<u>DAN VAN RITZ</u>	<u>CSE</u>	<u>\$65.00</u>	<u>2/16/08</u>	<u>4/15/08</u>

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

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

DAN VAN RITZ, INC. (Contractor)

II. Scope of Services - Continued:

B. List the specific tasks to be performed:

Complete systems design specifications.  
Analyze and code in COBOL.  
Perform unit, system and integration testing.  
Provide installation support.

C. List the deliverables expected to be produced:

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

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

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

E. List the performance standards that will be used to determine quality of work (e.g., SDP, documentation standards, testing standards, etc.)

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

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

INVOICE 100154

08 MAR 19 PM 1:24

SALESPERSON Dan	INVOICE DATE 3/18/08
INFORMATION Master Agreement #90-3167 Project/Service # Charge Unit #3620	

BANK OF CANADA  
 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	
<p><i>RITE \$408</i></p> <p>RECEIVED                  MAR 21 2008                  BANK of CANADA</p> <p>APPROVED FOR PAYMENT                  BY <u>B. Bab</u>                  UNIT # <u>3620</u></p>				
<b>TOTAL</b>			5720.00	

Thank You



**DAN VAN RITZ Consulting, Inc.**

Contractor Time Sheet

Contractor Name: DAN VAN RITZ

Client Company: BANK OF CANADA

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

Calendar Days	Hours Worked	Calendar 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	8	30	
		31	

Total Hours: 88 ←

Client Company Representative Acceptance: Charlye Monix 3/19/08  
Signature Date



Mail To: Dan Van Ritz Consulting, Inc.

## Appendix E – System Outputs

B	C	D	F	L	M	O	P	Q	R	S	T	U	V	W	X	Y
<b>Accruals</b>																
3	<b>Programmer</b>	<b>Vendor</b>	<b>Charge Invoice #</b>	<b>Total Invoice</b>	<b>Accrued</b>	<b>Reversed</b>										
4	Brown, Lou	EDS Temps Inc	3072	1,750.00	12/07											
5			<b>3072</b>	<b>1,750.00</b>		<b>01/10/08</b>										
6	Peckham, Art	Donny Wicks Associates	9408	3,600.00	12/07											
7	Wilkins, Peter	Donny Wicks Associates	9408	3,894.00	12/07											
8			<b>9408</b>	<b>7,494.00</b>		<b>01/10/08</b>										
9			<b>December 2007</b>	<b>9,244.00</b>												
11	Brown, Lou	EDS Temps Inc	3072	1,750.00	01/08											
12	Fortier, Brian	EDS Temps Inc	3072	4,412.50	01/08											
13			<b>3072</b>	<b>6,162.50</b>		<b>02/10/08</b>										
14	Lehrer, Philip	Beitarn Systems Inc	3117	8,580.00	01/08											
15			<b>3117</b>	<b>8,580.00</b>		<b>02/10/08</b>										
16	Peckham, Art	Donny Wicks Associates	9408	10,620.00	01/08											
17	Wilkins, Peter	Donny Wicks Associates	9408	4,838.00	01/08											
18			<b>9408</b>	<b>15,458.00</b>		<b>02/10/08</b>										
19			<b>January 2008</b>	<b>30,200.50</b>												
22			<b>February 2008</b>													
24			<b>Grand Total:</b>	<b>39,444.50</b>												
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																
49																

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
7	Lehrer, Philip	AMB	3117	101	01/02/08	01/31/08	165.0	8,580.00	01/08
8		AMB	3117					8,580.00	
9									
10		CCR						20,473.00	
11	Peckham, Art	CCR	9408	330	01/02/08	01/31/08	177.0	10,620.00	01/08
12	Wilkins, Peter	CCR	9408	1003	01/02/08	01/15/08	85.0	5,015.00	
13	Wilkins, Peter	CCR	9408	1004	01/16/08	01/31/08	82.0	4,838.00	01/08
14			9408					20,473.00	
15									
16		NAB						7,862.50	
17	Brown, Lou	NAB	3072	510	01/02/08	01/15/08	66.0	1,700.00	
18	Brown, Lou	NAB	3072	511	01/16/08	01/31/08	70.0	1,750.00	01/08
19	Fortier, Brian	NAB	3072	3723	01/02/08	01/31/08	176.5	4,412.50	01/08
20			3072					7,862.50	
21									
22									
23								36,915.50	
24									
Contract Programmers Monthly Expense Recap Report									
By Division and Unit									
February 2008									
27									
28									
29									
30									
31									
32									
33									
34									
35									
36									
37									
38									
39									
40									
41									
42									
43									
44									
45									
46									
47									

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 Last Charged	Unit Appendix A Max
<b>DIVISION: NAB</b>										
<b>Unit Number: 3072</b>										
10 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
<b>DIVISION: CCR</b>										
<b>Unit Number: 9408</b>										
15 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
16 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
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
32										
33										
34										
35										
36										
37										
38										
39										
40										
41										
42										
43										
44										
45										
46										
47										
48										

Monthly Contract Recap As of December 31, 2007									
Invoice Number	Date Paid	Periods Paid	Hours	Dollar Total	Start Date	End Date	Rate/Hour	Company	Project
509	01/11/08	12/17/07 to 12/31/07	70	\$1,750.00	12/17/07	06/17/08	25.00	EDS Temps Inc	Tax System Assistance
Total of Hours & Invoice Dollars:				\$1,750.00					
				Total Charged to Contract:	\$1,750.00				
				Percent Used:	6%				
				Remaining Contract Dollars:	\$27,250.00				
Project Manager: Clark, Rudy      Unit: 3072 Programmer: Brown, Lou      Company: EDS Temps Inc Start Date: 12/17/07      End Date: 06/17/08      Rate/Hour: 25.00      Fee Max: 29,000.00      Charge To: 3072									



# 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):

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

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

Attachment included.

---

DATE	ACTION

## 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

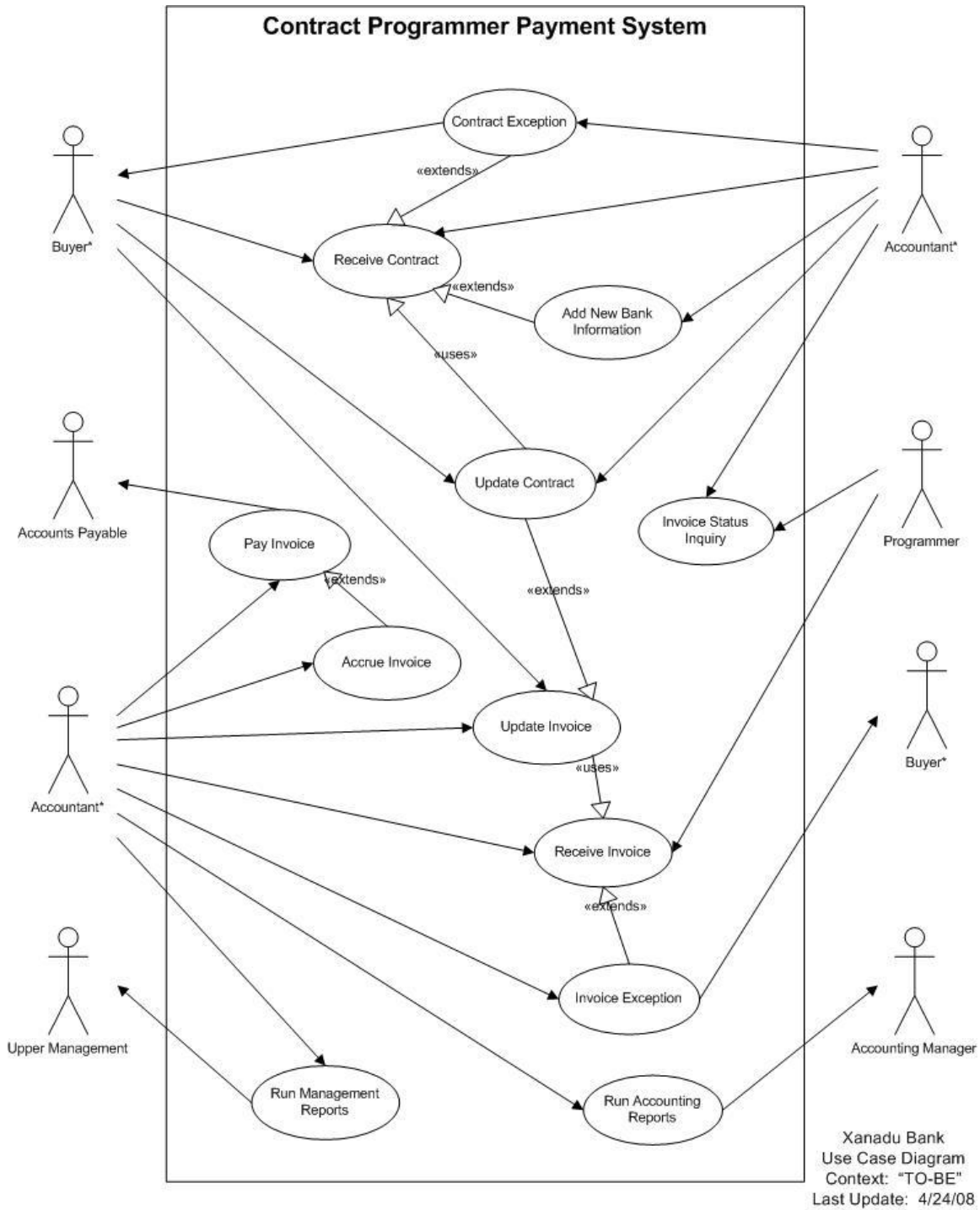
**P.O. Number:** A. Peckham

**Charge Unit:** 9408

Processed by Dave Spencer

1/11/08

## Appendix F - Use Case Diagram





## Appendix G - Use Case Scenarios

### UC001: Receive Contract

<b>Use Case Name:</b>	<b>RECEIVE CONTRACT</b>	ID: UC001
Primary Actor:	Accountant	
Brief Description:	This use case describes the steps for processing a new contract, from the time that it is delivered by the buyer, until a new contract is verified and entered into the system.	
Trigger:	New contract is delivered to the accounting department	
Related Use Cases:	Contract Exception (extended by); Add New Contract Information (extended by); Update Contract (used by)	
Normal flow of events:	<p>This use case begins when the Buyer delivers a new contract to the Accountant.</p> <ol style="list-style-type: none"> <li>1) Manually review contract to ensure that all the information needed by the accounting department is on the contract.</li> <li>2) Log onto the system and navigate to the "Enter Contract" screen.</li> <li>3) Search for the correct Vendor (Contractor) Number and select it.</li> <li>4) Enter all the required contract information (see Information Requirements below) into the system. Use appropriate "lookups" when applicable.</li> <li>5) When finished entering all required information, SAVE the new contract record into the system.</li> </ol> <p>This use case ends when the new contract is entered into the system.</p>	
Exception(s):	<ol style="list-style-type: none"> <li>1) If any required information is missing or invalid, an exception memo is created and sent to the buyer for resolution.</li> <li>3) If the vendor is not listed, navigate to the "Create Vendor" screen and create a new vendor record.</li> <li>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.</li> </ol>	
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:	<ol style="list-style-type: none"><li>1) A contract is not considered valid if any of the required information is missing, and must be returned to the buyer for correction.</li><li>2) A contract can be for more than one programmer working for the same vendor.</li><li>3) A programmer may be working on more that one contract at a time</li><li>4) If the PM is not listed in the corporate directory the signing authority needs to be contacted to obtain that information.</li></ol>

### UC002: Add New Bank Information

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

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

### UC004: Update Contract

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

<b>USE CASE NAME:</b>	<b>RECEIVE INVOICE</b>	<b>ID: UC005</b>
<b>Primary Actor:</b>	ACCOUNTANT	
<b>Brief Description:</b>	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.	
<b>Trigger:</b>	A new invoice is delivered by the vendor to the accountant.	
<b>Related Use Cases:</b>	Invoice Exception (extended by); Update Invoice (uses); Pay Invoice (uses); Accruals (uses); Accounting Report (uses); Management Report (uses)	
<b>Normal Flow of Events:</b>	<p>This use case starts when the vendor delivers a new invoice to the accountant.</p> <ol style="list-style-type: none"> <li>1) Accountant receives invoice.</li> <li>2) Accountant creates an ID number for the invoice.</li> <li>3) Accountant validates the invoice to ensure all information needed is provided.</li> <li>4) Accountant enters the invoice into the system, including all the required information (see information requirements below).</li> <li>5) Accountant saves the record.</li> <li>6) Accountant files hard copy of invoice.</li> </ol> <p>This use case ends when an invoice has been entered and saved in the system and has been filed.</p>	
<b>Exceptions:</b>	3) If any required information is missing or incomplete, the accountant will send the invoice to the buyer (see Invoice Exception use case).	
<b>Pre-condition(s):</b>	<p>There must be an existing contract with the vendor who sends the invoice.</p> <p>There must be a new invoice that needs to be entered into the system.</p> <p>The vendor who submits the invoice must be in the system.</p>	
<b>Post-condition(s):</b>	<p>A new invoice record has been created in the system and the original copy has been filed.</p> <p>The invoice has been approved for payment.</p>	
<b>Information Requirements:</b>	<p>ID Number</p> <p>Programmer</p> <p>Vendor</p> <p>Charge Unit</p> <p>Invoice Number</p> <p>Date Paid</p> <p>Begin Date</p> <p>End Date</p> <p>Rate</p>	

	<p>Total Hours  Total Invoice  Accrued  Memo  Description of Charges</p>
<b>Assumptions:</b>	<ol style="list-style-type: none"> <li>1) The vendor will deliver a complete invoice with all the required information.</li> <li>2) The information on the invoice is valid and accurate.</li> <li>3) The system will be able to accept all the required information on the invoice.</li> <li>4) The vendor is already in the system (entered with contract information).</li> </ol>
<b>Business Rules:</b>	<ol style="list-style-type: none"> <li>1) The vendor will deliver the invoice to the accountant.</li> <li>2) All invoice hard copies are filed for future reference.</li> <li>3) All invoices must be entered into the system.</li> <li>4) All invoices are manually assigned a unique ID number.</li> <li>5) Each invoice can have only one contractor name on it.</li> <li>6) Each invoice must have all the required information needed (see Information Requirements above).</li> <li>7) All invoices will be time stamped on their received dates.</li> <li>8) Invoice Totals will be subtracted from the Fee Maximum.</li> <li>9) Approval signature and Charge Unit will be included on Invoices approved for payment.</li> <li>10) Invoice information will be matched to contract stipulations to validate negotiated terms, rates, and limitations.</li> </ol>



### UC006: Invoice Exception

<b>Use Case Name:</b>	<b>INVOICE EXCEPTION</b>	ID: UC006
<b>Primary Actor:</b>	Accountant	
<b>Brief Description:</b>	This use case describes the steps for processing an invoice exception memo to return an incomplete, invalid, or unpayable invoice to the Buyer, from the time the incomplete, invalid, or unpayable invoice is received until it has been returned to the Buyer.	
<b>Trigger:</b>	An incomplete, invalid, or unpayable invoice is received from the Vendor	
<b>Related Use Cases:</b>	Receive Invoice (extends)	
<b>Normal flow of events:</b>	<p>This use case begins when the Vendor sends an invoice for programming services to the Accountant (should arrive via the Project Manager).</p> <ol style="list-style-type: none"> <li>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</li> <li>2) Enter the invoice into the system with as much information as possible</li> <li>3) Enter "missing/invalid" or default to "zero" value in the field for the piece(s) of information that is missing or invalid</li> <li>3) Enter the "Payment Status" as "Do Not Pay"</li> <li>4) Enter into the "Invoice Notes" field the reason the invoice cannot be paid</li> <li>5) SAVE the invoice into the system</li> <li>6) Generate a return memo to the Buyer explaining the reason for the return</li> <li>7) Attach the return memo to the invoice and send it back to the Buyer</li> </ol> <p>This use case ends when the invomplete, invalid, or unpayable invoice has been returned to the Buyer.</p>	
<b>Exception(s):</b>	None	
<b>Precondition(s):</b>	An invoice has been received that is either incomplete, invalid, or otherwise unpayable.	
<b>Postconditions(s):</b>	The incomplete, invalid, or otherwise unpayable invoice has been returned to the Buyer.	
<b>Information</b>	(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 contract, OR the "Hourly Fee" does not match with that of the original contract, OR payment of the invoice would cause the "Fee Maximum" amount on the original contract to be exceeded

### UC007: Update Invoice

<b>Use Case Name:</b>	<b>UPDATE INVOICE</b>	ID: UC007
<b>Primary Actor:</b>	Accountant	
<b>Brief Description:</b>	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.	
<b>Trigger:</b>	An updated or new invoice and/or a contract extension is received from the Buyer	
<b>Related Use Cases:</b>	Receive Invoice (uses); Update Contract (extended by)	
<b>Normal flow of events:</b>	<p>This use case begins when the Buyer delivers an updated or new invoice and/or a contract extension to the Accountant.</p> <ol style="list-style-type: none"> <li>1) Manually review the invoice to ensure all the information needed by the accounting department is on the invoice</li> <li>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</li> <li>3) If a contract extension is received, search for the original contract in the system</li> <li>4) Enter the new contract information in the system (see Information Requirements below and refer to UC004 for normal flow of events) (if applicable)</li> <li>5) SAVE the updated contract record into the system (if applicable)</li> <li>6) Search for the returned invoice in the system</li> <li>7) Change the fields that have missing or zero values by entering the correct information from the updated or new invoice</li> <li>8) Run a system check to ensure that the dates for programming services fall within the date range specified on the revised contract</li> <li>6) Run a system check to ensure that the billed rate is the same as the "Hourly Fee" on the revised contract</li> <li>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)</li> <li>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"</li> <li>9) Enter into the "Invoice Notes" field the date the invoice was returned</li> <li>10) SAVE the invoice record into the system</li> </ol> <p>This use case ends when the revised or new invoice and/or contract extension has been correctly and completely updated in the system.</p>	

Exception(s):	<p>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)</p> <p>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</p>
Precondition(s):	<p>1) A revised invoice has been received from the Buyer</p> <p>2) A contract extension has been received from the Buyer</p>
Postconditions(s)	<p>1) The updated or new invoice has been entered into the system with the "Approved to Pay" status</p> <p>2) The contract extension information has been entered into the system and the contract has been updated</p>
Information Requirements:	<p>(UC001)</p> <p>Begin Date</p> <p>End Date</p> <p>Hourly Fee</p> <p>Fee Maximum</p> <p>(see UC005 "Receive Invoice" for applicable data)</p> <p>Payment Status</p> <p>Invoice Notes</p>
Assumptions:	<p>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</p> <p>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</p>
Business Rules:	<p>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.</p> <p>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.</p> <p>3) The Accountant must re-check the updated invoice against the contract limitations before approving it for payment</p>

### UC008: Invoice Status Inquiry

<b>Use Case Name:</b>	<b>INVOICE STATUS INQUIRY</b>	ID: UC008
<b>Primary Actor:</b>	Accountant	
<b>Brief Description:</b>	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.	
<b>Trigger:</b>	The Accountant receives an inquiry from the Vendor for payment status on an invoice	
<b>Related Use Cases:</b>	None	
<b>Normal flow of events:</b>	<p>This use case begins when the Accountant receives an invoice payment status inquiry from the Vendor.</p> <ol style="list-style-type: none"> <li>1) Log onto the system and search for the invoice in question (run a report or query)</li> <li>2) Check the "Payment Status" field to determine the current status</li> <li>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</li> <li>4) If the Vendor is on the telephone, convey the details of the invoice payment status to the Vendor</li> <li>5) If the invoice is not currently payable, explain the reason(s) for non-payment</li> <li>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</li> <li>7) Enter into the "Invoice Notes" field the date of the Vendor inquiry and the response that was provided in reply</li> <li>8) SAVE the updated invoice record into the system</li> </ol> <p>This use case ends when the Vendor has received the invoice payment status.</p>	
<b>Exception(s):</b>	<ol style="list-style-type: none"> <li>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)</li> <li>(6 &amp; 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</li> </ol>	
<b>Precondition(s):</b>	The Vendor has sent an invoice for programming services and has not received payment according to the terms of the invoice	
<b>Postconditions(s):</b>	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

<b>Use Case Name:</b>	<b>PAY INVOICE</b>	ID: UC009
<b>Primary Actor:</b>	Accountant	
<b>Brief Description:</b>	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.	
<b>Trigger:</b>	Invoice(s) are entered into the system with an "Approved to Pay" Payment Status	
<b>Related Use Cases:</b>	None	
<b>Normal flow of events:</b>	<p>This use case begins when the Accountant has set the invoice Payment Status to "Approved to Pay".</p> <ol style="list-style-type: none"> <li>1) If necessary, log onto the system and locate the invoice to be paid</li> <li>2) Verify the Payment Status is "Approved to Pay"</li> <li>3) Enter the current day's date in the "Date Paid" field</li> <li>4) SAVE the invoice record</li> <li>5) Generate and print out a Data Entry Sheet for the invoice</li> <li>6) Attach the Data Entry Sheet to the Invoice and send both to the A/P department</li> </ol> <p>This use case ends when the invoice has been updated with the "Date Paid" and sent to the A/P department to have a check issued.</p>	
<b>Exception(s):</b>	3) If the "Date Paid" is AFTER the cut-off for the last A/P check run for the CURRENT month AND before the 6 <sup>th</sup> day of the FOLLOWING month, the invoice will need to be accrued	
<b>Precondition(s):</b>	An invoice is approved for payment and is ready to be sent to A/P to have a check cut.	
<b>Postconditions(s)</b>	The invoice has been sent to the A/P department with a Data Entry Sheet attached.	
<b>Information Requirements:</b>	<p>Vendor Name                  Vendor Number                  Invoice Number                  Description (the programmer's 1<sup>st</sup> initial and full last name AND the dates of service covered by the invoice)                  Invoice Date                  Invoice Total                  G/L Account                  P.O. Number (the programmer's 1<sup>st</sup> initial and full last name)                  Charge Unit                  Accountant's Name                  Date Paid (date invoice is sent to the A/P group)</p>	
<b>Assumptions:</b>	All invoices received for services in the current month can be	

	processed for payment and have a check cut by the A/P department before the end of the current month
Business Rules:	<ol style="list-style-type: none"><li>1) All invoices sent to the A/P department for payment must include a Data Entry Sheet with specific information (see Information Requirements above)</li><li>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.</li></ol>



### UC010: Accrue Invoice

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

<b>Use Case Name:</b>	<b>RUN ACCOUNTING REPORTS</b>	ID: UC011
<b>Primary Actor:</b>	Accountant	
<b>Brief Description:</b>	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.	
<b>Trigger:</b>	The deadline for the month-end Accounting department reports.	
<b>Related Use Cases:</b>	None	
<b>Normal flow of events:</b>	<p>This use case begins when the deadline due date for the "General Ledger Expense Report" and "Accrual Report" is reached.</p> <ol style="list-style-type: none"> <li>1) Log onto the system and navigate to the Reports Menu</li> <li>2) Select the "G/L Expense" option</li> <li>3) Enter the date range for the current reporting period</li> <li>4) Select the PRINT REPORT option</li> <li>5) Return to the Reports menu</li> <li>6) Select the "Accruals" option</li> <li>7) Enter the date range for the current reporting period</li> <li>8) Select the PRINT REPORT option</li> <li>9) Return to the Reports Menu OR exit to the Main Menu</li> <li>10) Deliver both reports to the Accounting Manager</li> </ol> <p>This use case ends when both the "General Ledger Expense Report" and "Accrual Report" have been delivered to the Accounting Manager.</p>	
<b>Exception(s):</b>	None	
<b>Precondition(s):</b>	It is time to generate the monthly Accounting department reports	
<b>Postconditions(s):</b>	The monthly Accounting department reports have been delivered to the Accounting Manager	
<b>Information Requirements:</b>	(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 6 <sup>th</sup> business day of the month.

## UC012: Run Management Reports

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

	<p>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)</p>
<p>Assumptions:</p>	<p>There will actually be at least one invoice paid to the contract programmer G/L account 507613 in the current reporting period</p>
<p>Business Rules:</p>	<p>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 11<sup>th</sup> business day of the month.</p>

## 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**

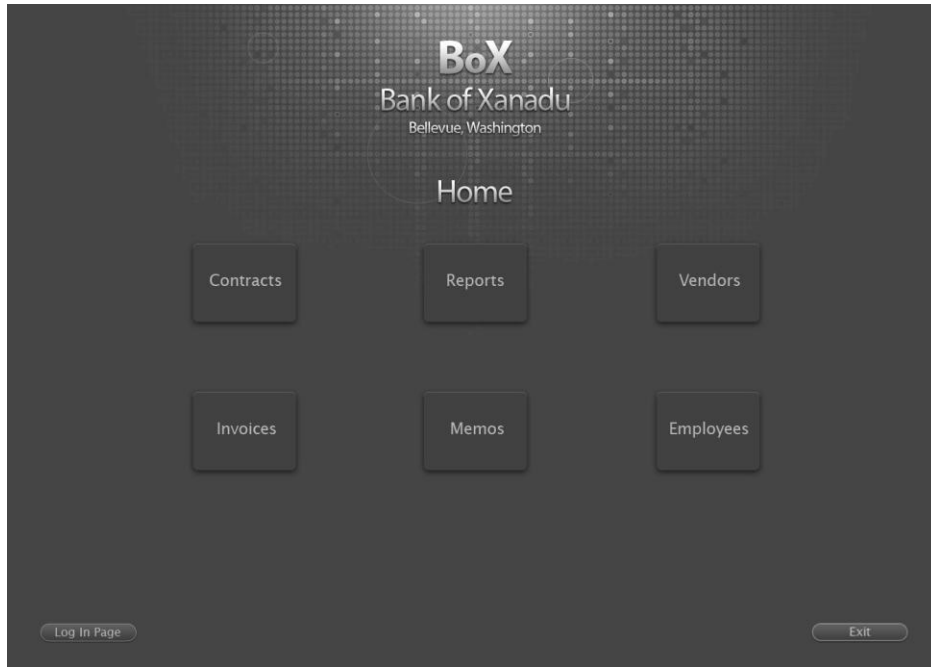
The scope of the current project defined in the “System Requirements Document Automatic Contractual Payment System for Bank of Xanadu, Bellevue, Washington” is for the bank’s Bellevue branch exclusively. In the future, the bank may want to expand the area of the system to additional branches. This would 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.



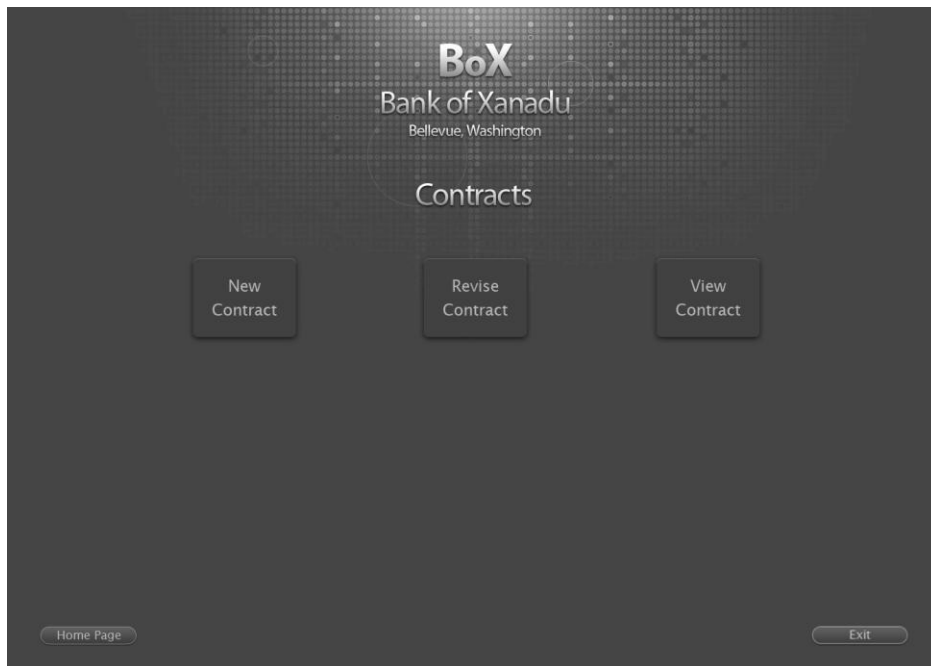
## Appendix L – Navigation Screens

---

### Item 1: Home Page



### Item 2: Contracts



**Item 3: New Contract**

The screenshot shows a web interface for the 'BoX Bank of Xanadu' system. The header includes the logo 'BoX Bank of Xanadu' and the location 'Bellevue, Washington'. The main title is 'New Contract'. The form contains the following fields:

Contract ID	Contract Date	Unit ID	Bank Contact ID
Vendor ID	Buyer ID	Programmer ID	Acct Employee ID
Contract Start Date	Contract End Date	Contract Hourly Rate	Contract Fee Max
Contract Description		Contract Status	

At the bottom of the form, there are navigation buttons: 'Contract Page', 'Home Page', 'Reset', 'Submit', and 'Exit'.

**Item 4: Revise Contract**

The screenshot shows a web interface for the 'BoX Bank of Xanadu' system. The header includes the logo 'BoX Bank of Xanadu' and the location 'Bellevue, Washington'. The main title is 'Revise Contract'. The form contains the following fields:

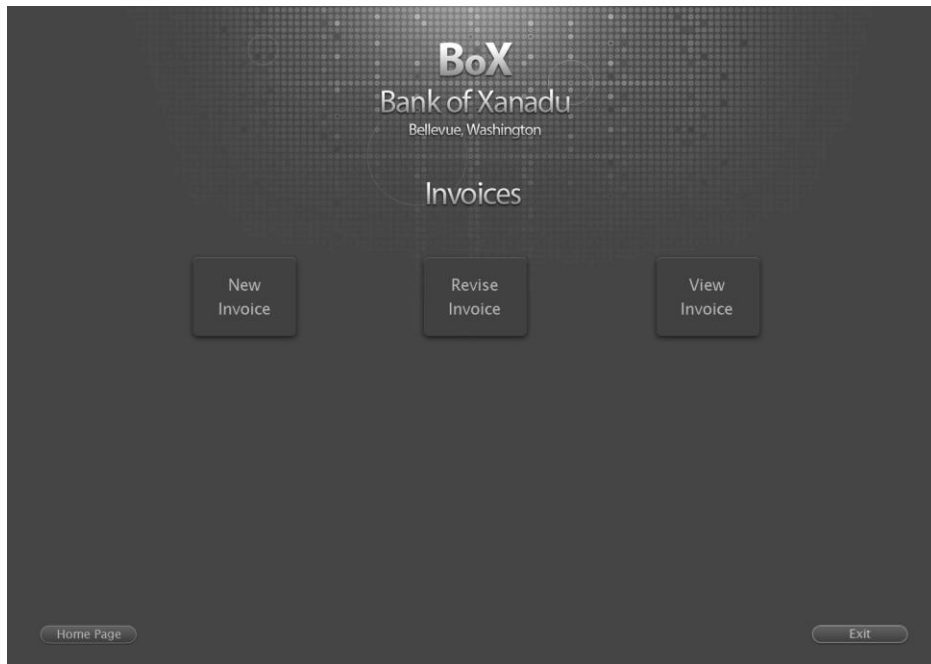
Contract ID	Contract Date	Unit ID	Bank Contact ID
Vendor ID	Buyer ID	Programmer ID	Acct Employee ID
Contract Start Date	Contract End Date	Contract Hourly Rate	Contract Fee Max
Contract Description		Contract Status	

At the bottom of the form, there are navigation buttons: 'Contract Page', 'Home Page', 'Reset', 'Submit', and 'Exit'.

### Item 5: View Contract



### Item 6: Invoices



**Item 7: New Invoice**

The screenshot shows a web form titled "New Invoice" for the "BoX Bank of Xanadu" in Bellevue, Washington. The form is set against a dark background with a grid of light dots. It contains the following fields and controls:

- Invoice ID:
- Invoice Date:
- Invoice Received:
- Invoice Accrued:  No  Yes
- Acct Employee ID:
- Contract ID:
- Programmer ID:
- Vendor ID:
- Invoice Start Date:
- Invoice End Date:
- General Ledger Acct:
- Invoice Terms:
- Invoice Hours:
- Invoice Hourly Rate:
- Invoice Total:
- Fee Max Balance:
- Invoice Description:
- Invoice Status:

Navigation and action buttons are located at the bottom:

- Invoice Page (button)
- Home Page (button)
- Reset (button)
- Submit (button)
- Exit (button)

**Item 8: Revise Invoice**

The screenshot shows a web form titled "Revise Invoice" for the "BoX Bank of Xanadu" in Bellevue, Washington. The form is set against a dark background with a grid of light dots. It contains the following fields and controls:

- Invoice ID:
- Invoice Date:
- Invoice Received:
- Invoice Accrued:  No  Yes
- Acct Employee ID:
- Contract ID:
- Programmer ID:
- Vendor ID:
- Invoice Start Date:
- Invoice End Date:
- General Ledger Acct:
- Invoice Terms:
- Invoice Hours:
- Invoice Hourly Rate:
- Invoice Total:
- Fee Max Balance:
- Invoice Description:
- Invoice Status:

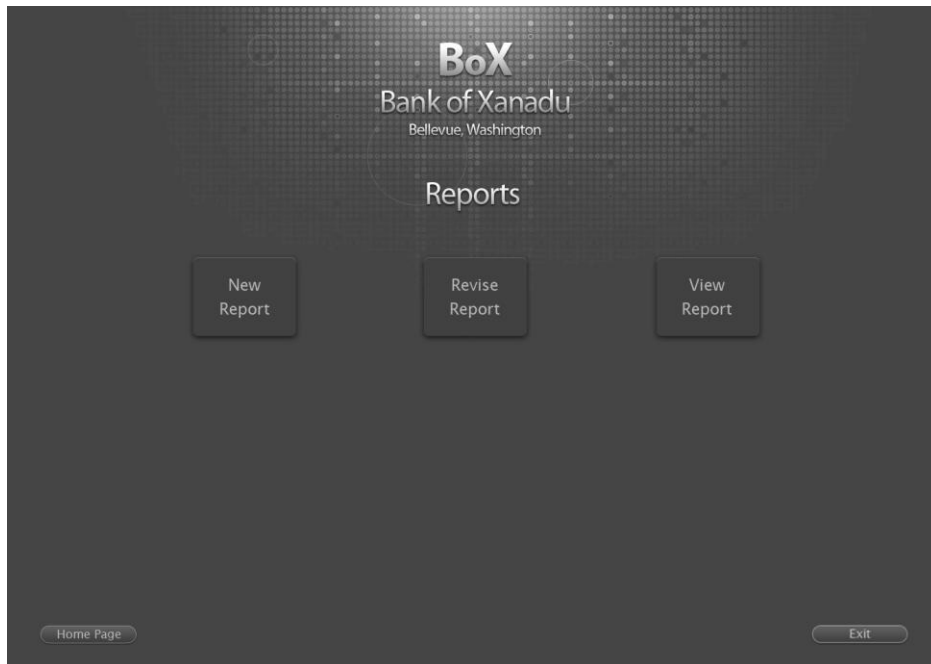
Navigation and action buttons are located at the bottom:

- Invoice Page (button)
- Home Page (button)
- Reset (button)
- Submit (button)
- Exit (button)

### Item 9: View Invoice



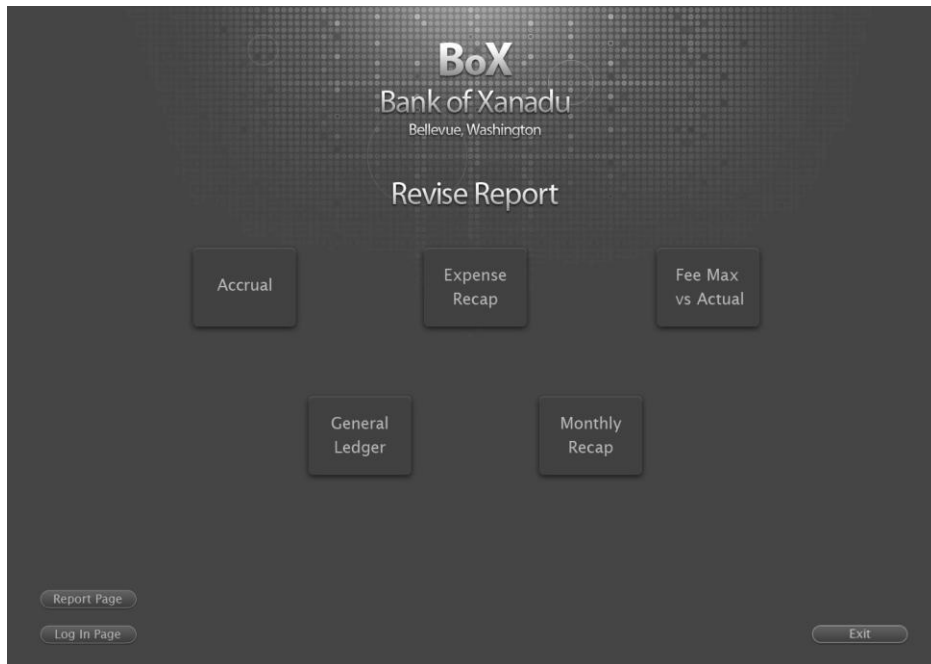
### Item 10: Reports



**Item 11: New Report**



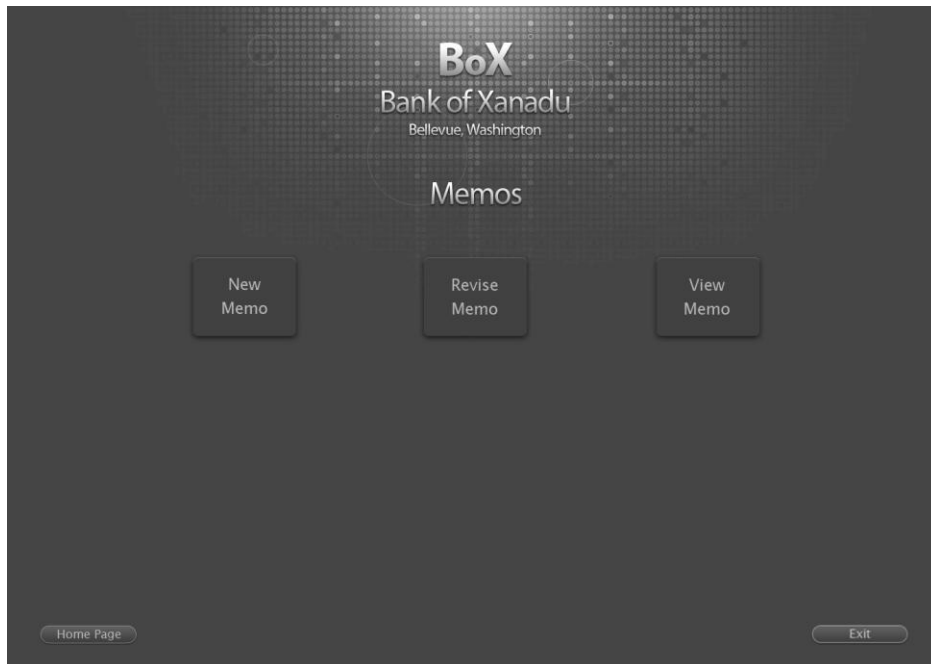
**Item 12: Revise Report**



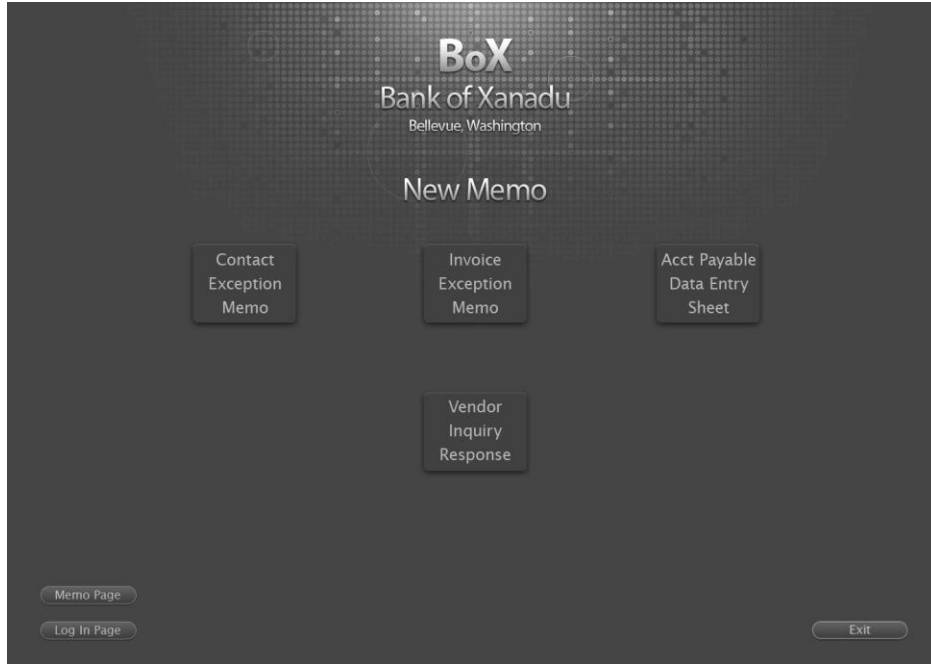
### Item 13: View Report



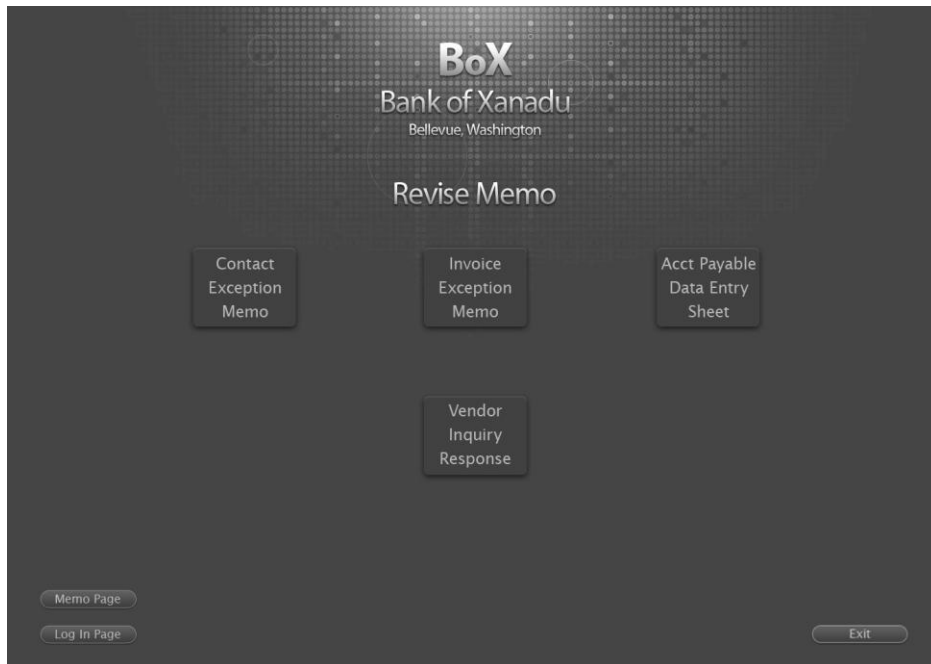
### Item 14: Memos



**Item 15: New Memo**



**Item 16: Revise Memo**

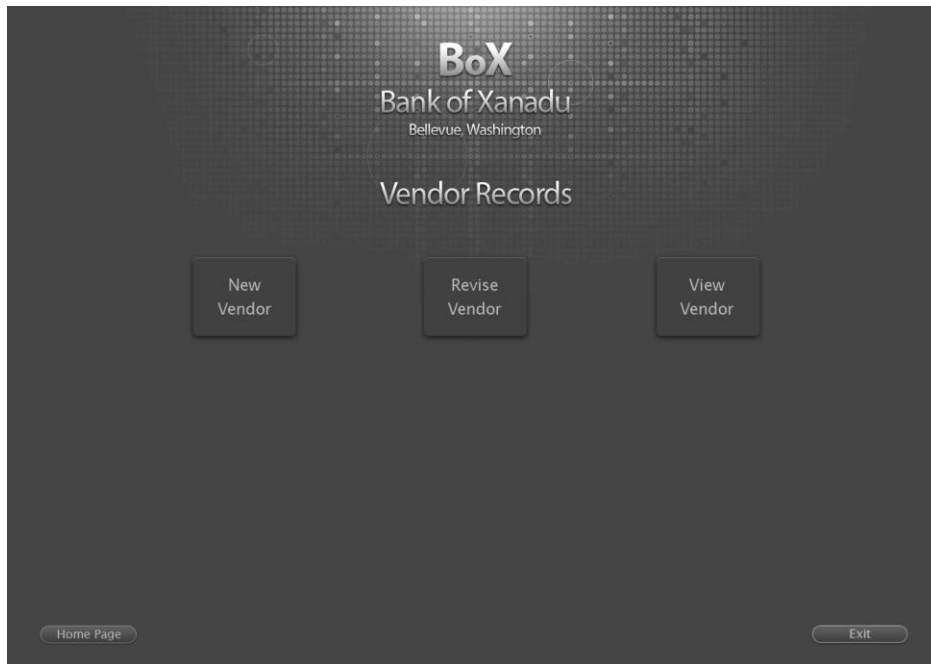




**Item 17: View Memo**



**Item 18: Vendors**



**Item 19: New Vendor**

The screenshot shows a web form titled "New Vendor" for the "BoX Bank of Xanadu" in Bellevue, Washington. The form is set against a dark background with a grid of light dots. It contains the following fields:

- Vendor ID
- Vendor Name
- Vendor Address
- Vendor City
- Vendor State
- Vendor Zip
- Vendor Phone
- Vendor E-Mail
- Vendor Contact
- Vendor Title

At the bottom of the form, there are four buttons: "Vendor Page", "Home Page", "Reset", "Submit", and "Exit".

**Item 20: Revise Vendor**

The screenshot shows a web form titled "Revise Vendor" for the "BoX Bank of Xanadu" in Bellevue, Washington. The form is set against a dark background with a grid of light dots. It contains the following fields:

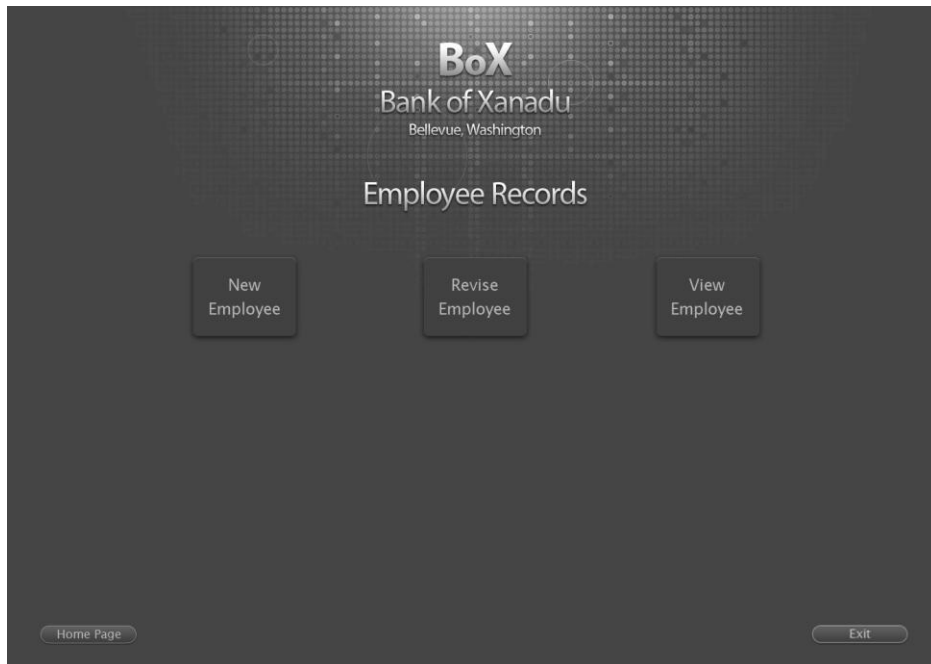
- Vendor ID
- Vendor Name
- Vendor Address
- Vendor City
- Vendor State
- Vendor Zip
- Vendor Phone
- Vendor E-Mail
- Vendor Contact
- Vendor Title

At the bottom of the form, there are four buttons: "Vendor Page", "Home Page", "Reset", "Submit", and "Exit".

**Item 21: View Vendor**



**Item 22: Employees**



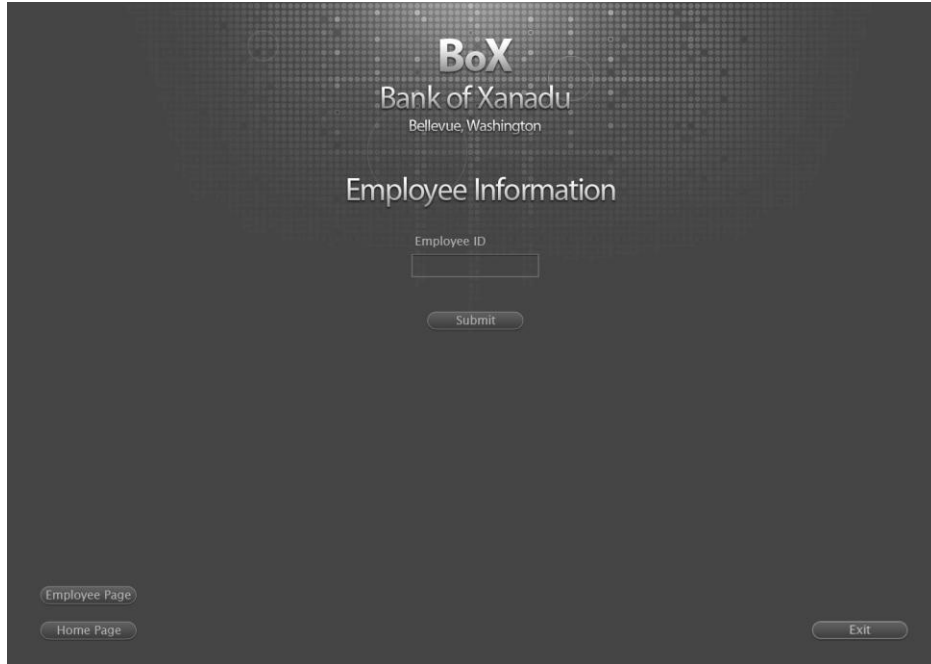
### Item 23: New Employee

The screenshot shows a web form titled "New Employee" for the "BoX Bank of Xanadu" system. The form is set against a dark grey background with a subtle grid pattern. At the top center, the logo "BoX" is displayed in a large, bold, white font, with "Bank of Xanadu" and "Bellevue, Washington" in smaller white text below it. The main title "New Employee" is centered in a white font. The form consists of eight input fields arranged in two rows of four. The first row contains fields for "Employee ID", "Employee First Name", "Employee Last Name", and "Employee Unit". The second row contains fields for "Employee User Name", "Employee Login ID", "Employee Phone", and "Employee Title". At the bottom of the form, there are four buttons: "Employee Page" (on the left), "Home Page" (below it), "Reset" (centered), "Submit" (centered), and "Exit" (on the right).

### Item 24: Revise Employee

The screenshot shows a web form titled "Revise Employee" for the "BoX Bank of Xanadu" system. The form is set against a dark grey background with a subtle grid pattern. At the top center, the logo "BoX" is displayed in a large, bold, white font, with "Bank of Xanadu" and "Bellevue, Washington" in smaller white text below it. The main title "Revise Employee" is centered in a white font. The form consists of eight input fields arranged in two rows of four. The first row contains fields for "Employee ID", "Employee First Name", "Employee Last Name", and "Employee Unit". The second row contains fields for "Employee User Name", "Employee Login ID", "Employee Phone", and "Employee Title". At the bottom of the form, there are four buttons: "Employee Page" (on the left), "Home Page" (below it), "Reset" (centered), "Submit" (centered), and "Exit" (on the right).

**Item 25: View Employee**



**Item 26: Generic View**

