Transmittal Memo

Date: June 11, 2011
To: Patrick Jay, Manager, Vice President, Accounting Group at Bellevue
From: Robot Rebellion Regime
Subject: Transmittal of System Design Document

At your request we have prepared a System Design Document. Following the needs of the Bank of Xanadu's automated contractual payment system, we have designed this document to include the following information:

1) Architecture and Design Considerations
2) Information Models that include:
    i) Entity Relationship Diagram
    ii) Meta Data Dictionary
3) Procedures
4) Inputs
5) Outputs

This document should provide the necessary information that you can review at your leisure. If possible, we would like to request a meeting with you at the Bank of Xanadu Bellevue Branch, May 7th 2011 at 1:00 PM. If this time does not work within your schedule, please let us know and we can reschedule. Please invite Dave Spencer and anyone else you deem necessary to the successful outcome of the project to the meeting.

The following system analysts will be attending this meeting to review our material and answer any questions that you may have.

- Terry Gorle – Project Manager, backup to Chris Clemente
- Adam McCormack – Developer, backup to Terry Gorle
- Eric Nelson – User Manual Coordinator, backup to Adam McCormack
- Doug Vickers - Document Czar, backup to Eric Nelson
- Chris Clemente – Testing Coordinator, backup to Doug Vickers

We look forward to our meeting with you and getting your input on our progress thus far. Should you have any questions before then please feel free to contact us at RobotRebellionRegime@example.com or 206-555-1234

System Design Document

June 11, 2011

Robot Rebellion Regime

Adam McCormack, Terry Gorle, Doug Vickers, Chris Clemente, and Eric Nelson

Contents

Introduction 4
Architecture and Design Considerations 4
Information / Data Model 6
User Navigation Design 7
Input / Output 8
Procedures 9
Interface Design and Coding Standards 10
    External 10
    Internal 10
Appendix 11
    Data Dictionary 11
    Entity Relationship Diagram 15
    Use Case Diagram 16

Use Case Scenarios 17
User Navigation Storyboard 29
Input / Output Examples 30
Internal Procedures 33
Issues List 34
Future Enhancement List 35
Conclusion 36
Sign Off Page 36

Introduction

The Robot Rebellion Regime utilizes the Systems Development Life Cycle (SDLC) to build an information system. We have previously addressed the planning and analysis phases, and now we begin the design phase. The System Design Document is the primary deliverable of this phase and provides a comprehensive description of all documents for the system. These documents include:

- Architecture and Design Considerations, including hardware and software specifications
- Information/Data Model, including an Entity Relationship Diagram and Meta Data Dictionary in the Appendix on page 11.
- User Navigation Design, including screen design and screen shots, with a navigation chart in the Appendix on page 30
- Inputs, source documents or forms
- Outputs, primary reports or other outputs
- Various Procedures, including use cases
- Interface Design and Coding

Additional documentation, such as: Entity Relationship Diagram, Metadata Dictionary, Function Hierarchy Diagram, Source Document Examples, Data Input Forms, Report Formats, Output Examples, Use Cases, Internal Procedures, Design Standards Document, Issues List, and Future List.

Architecture and Design Considerations

The system architecture of the proposed design will consist of one centralized dedicated server, utilizing active RAID ghosting as backup for database integrity and security. Authenticated personnel will access the system at their desktop workstations via an isolated server-based network. Security levels will be implemented within the system to assure a hierarchy of database access from "read-only" to full administrative privileges based on job duties and responsibilities. In cases where two or more users have administrative permissions, data integrity demands only one user can actively edit data within a specific form at any given time. Permissions granted will follow strict guidelines.
The server will utilize Microsoft Windows Server 2008 Enterprise as its operating system. Microsoft Access 2010 will be the software application installed at workstations and utilized for this DBMS. The server hardware/software components are fully scalable to expand beyond the architecture capabilities designed into this initial closed-system DBMS proposal. All performance, integration, maintenance and security concerns have been considered and satisfied within this proposal.

Hardware Specifications:

- Dell PowerEdge R610 rack server
- (2) Intel Xeon X5667 3.06Ghz CPU's, 12M cache, Turbo, HT, 1333MHz Max Mem
- 16GB Memory (4x4GB), 1333MHz Dual Ranked RDIMM's for 2 CPU's, Advanced ECC
- (3) 500GB 7.2K RPM SATA 2.5in HotPlug Hard Drives
- (2) 146GB 10K RPM Serial-Attach SCSI 6Gbps 2.5in HotPlug Hard Drives
- RAID 1/RAID 5 for H700 Controller, Mixed SAS/SATA capability
- Broadcom 5709 Dual Port 1GbE NIC w/TOE iSCSI, PCIe-4
- DVD+/-RW, SATA Internal Drive
- Internal SD Module w/1GB SD Card

- Dell UPS, 1920 Watt, 2U Rack
- 3-year ProSupport and Mission Critical 4HR 7x24 Onsite Support / Proactive Systems Management
- 17” LCD Monitor, Keyboard, Optical Mouse, US

Software Specifications:

- Windows Server 2008 Enterprise x64, IIS7
- VMWare virtualization
- MySQL Server v5.x / PHP
- Microsoft Office Professional Suite

Microsoft Access 2010 Overview:

Access 2010 allows you track and report information with ease. Prebuilt applications can modify or adapt to changing business needs. Collect information through forms in e-mail or import data from external applications. Create and edit detailed reports that display sorted, filtered, and grouped information. Audit revision history, recover deleted information, set data access permissions, and back up your information at regular intervals. Access provides a comprehensive view of tables, forms, queries, and reports. Create custom groups to organize and see all the forms and reports related to a single table. Access can import Excel tables into a new datasheet, building all the fields and recognizing the data types automatically. Aside from being an extremely affordable solution, Access allows multi-value fields, which support and encourages the use of attributes having a list of values, rather than all attributes having a single value, providing complex data type support.

Information / Data Model

The information requirements include the acceptance of invoices, contracts, and the ability for them to be updated when necessary. The database application also must be able to provide for invoice and contract exceptions and extensions. There must also be the ability to update accruals. Finally, there will be the ability to create various reports, as detailed in the Input and Output section to follow. The entities include:

- Vendor
- Programmer

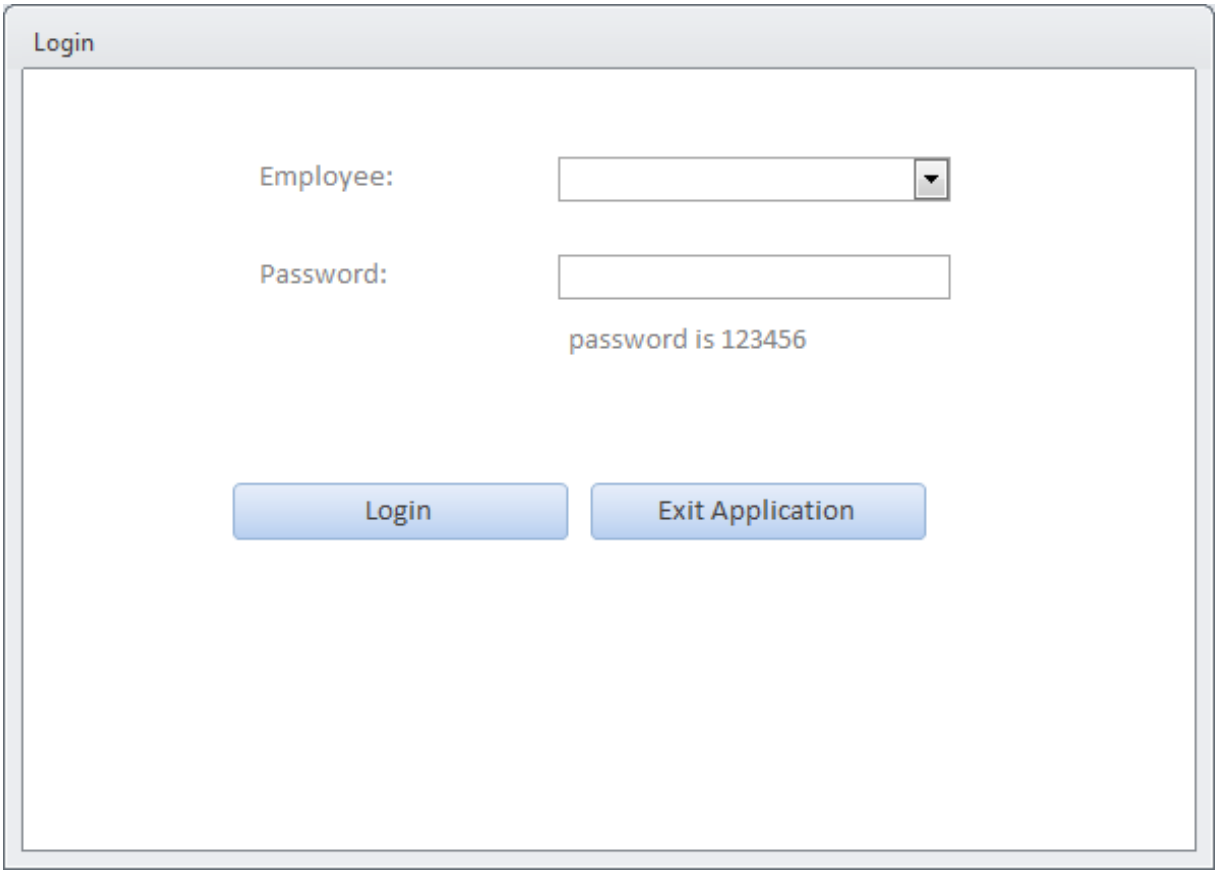
- Contract
- Unit
- Division
- Bank contact
- Buyer
- Invoice
- Accounting employee

The complete Entity Relationship Diagram can be found in the appendix on page 15. The ERD diagram illustrates the table structure, identifies the associated attributes, and displays the relationship pathways between entities. A Metadata Dictionary can also be found in the appendix on page 11. It provides further detail regarding various database table parameters.

User Navigation Design

The screen design will be a user-friendly and simple format that is highly navigable. There will be easy to identify buttons, labeled text boxes, and several drop down menus. A User Navigation Design storyboard is included in the appendix on page 29. An example of the screen design is:

Log in screen:



Input / Output

Data Input Forms:

The input forms will consist of the following:

- Vendor invoices
- Contracts
- Invoice update
- Contract update
- Memos / Note attachments
- Vendor information

- Programmer information

Although there will continue to be manual data entry and item selection for specific fields or drop down lists in certain forms, it is our goal that manual input will be minimized. Paper documentation can be scanned to PDF format for data backup storage and digital reference. Standardized format external hard copy invoice and contract forms will be utilized to maximize the efficiency of Optical Character Recognition (OCR) software to capture and automatically populate the associated database form fields therein, thereby eliminating a vast majority of the manual data entry used in the past. Form data can be proofread for accuracy more efficiently than manual entry. To minimize human error, input validation will be implemented within the form fields.

Data Output Forms:

While the database application will be able to send an output notification to Pay Invoice, as well as generate automated email notifications to vendors and bank personnel based on contract versus calendar events, the Data output forms will consist primarily of report generation.
The periodic reports include the following:

- Invoices Report
- Accruals Report
- Contract Programmers Monthly Expense Recap Report
- Contract Programmer Report
- Monthly Contract Recap Report
- Invoice exceptions
- Contract exceptions

Accountants, Bank Management, Project Managers, and the Contract Group access these reports as necessary for overview.

Procedures

The system will have with an intuitive design that will be user friendly. There will be a simple log in screen that leads to a page that the user will have options to choose from. That will lead to the specific screen that is organized to perform functions within the option chosen. For a more specific description of what each screen will entail, the following list of Use Cases is included:

- Receive Contract
- Add new Bank information
- Contract Exception

- Update Contract
- Receive Invoice
- Invoice Exception
- Update Invoice
- Invoice Status Inquiry
- Pay invoice
- Accrue Invoice
- Run Accounting reports
- Run Management Reports

There will also be written procedures that the bank staff will find easy to understand and follow.

Interface Design and Coding Standards

External (screens & navigation):

The font we will be using for the form labels, text boxes, buttons, etc. will be Calibri font size 11. The headers with the form title will be Calibri font size 20. The buttons will all be the same size no matter what screen the user is on. The command buttons will vary depending on the screen you are on. The only constant command button that will appear on every navigational screen, at the moment, will be the "logout" button. We will be using text boxes and drop downs for users to input information. We will be using the generic Microsoft Access color and scheme for all our pages. There will be a blue/gray header with the form title on the top along with the current date and time. The rest of the form will be white. You can find examples of interface design principles and standards on the web.

Internal (code and database design):

Guidelines for comments in the code:

- Comments should provide an outline of the function.
- They should explain what the code is doing without restating the code itself.

- They are necessary only when it is not obvious what the code is doing. To determine “obviousness”, ask yourself the question “If I had never seen this code before or even hadn't seen it for a week, how long would it take me to determine what this section does?” If the answer is anything longer than 1 minute, include a comment.

Guidelines for variables and objects in the code:

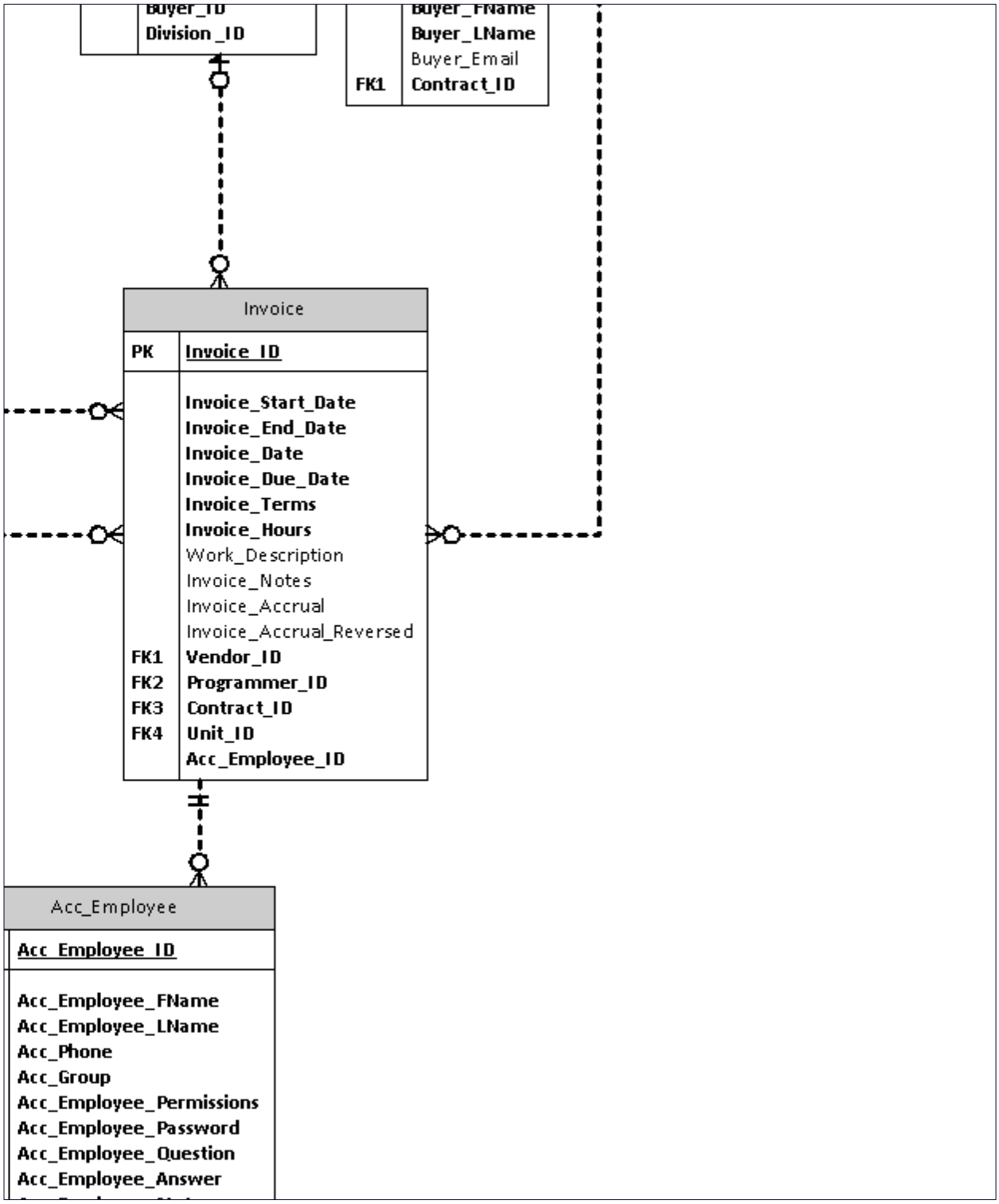
- Can include letters, numbers, or underscore characters
- Must begin with a letter

- Can't be a Visual Basic keyword. A keyword is a word that Visual Basic uses as part of its language. This includes predefined statements (such as If and Loop), functions (such as Len and Abs), and operators (such as Or and Mod).

Guidelines for database objects:

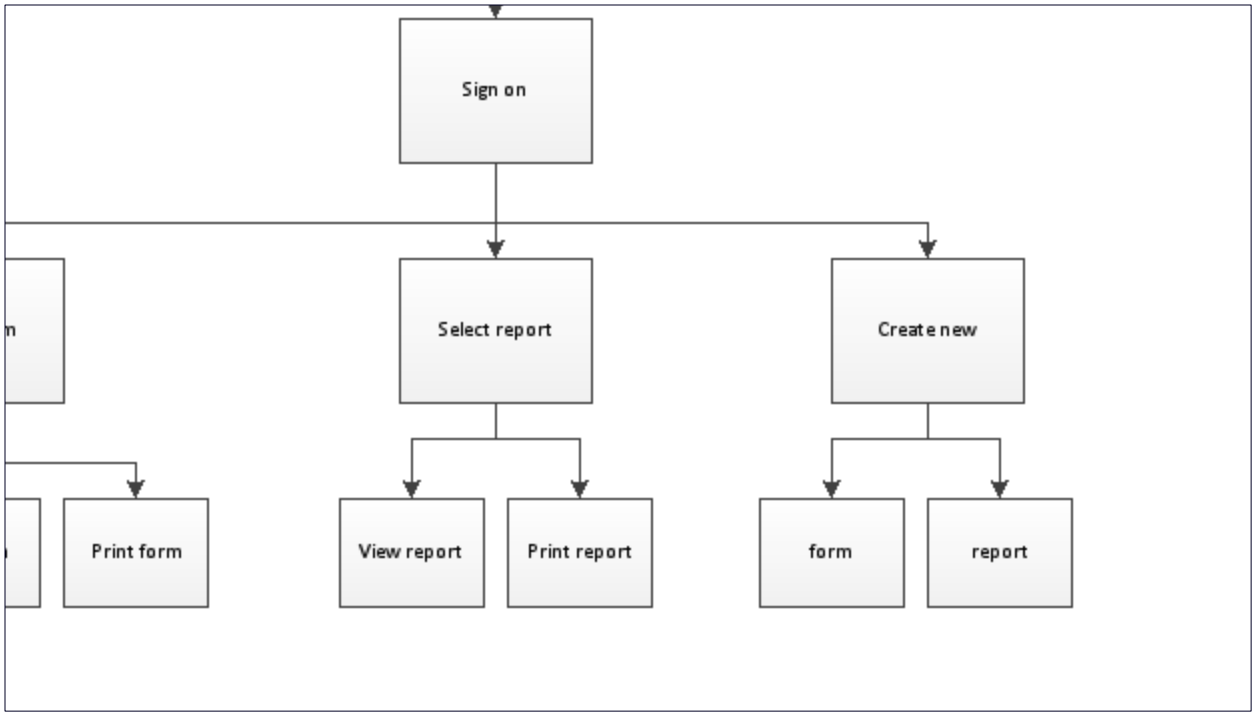
- Will be abbreviated (form will be frm, table will be tbl, etc.)
- They will be unique names so as to identify them when coding if needed
- Underscores will separate words (Contract_Exception)

Appendix



Entity Relationship Diagram

Navigation Storyboard



Data Dictionary

Entity Name	Attribute Name	Definition	Domain Constraints	Referential Integrity Constraints
tbl_Contract	Cont_ID	Unique identifier for Contract	Auto number. Primary key	PK (Primary Key)
	Cont_Start_Date	Contract's start date	Required. Input mask	
	Cont_End_Date	Contract's end date	Required. Input mask	
	Cont_Fee_Max	Contract's fee maximum	Required. Number	
	Cont_Description	Description of contract work	Not required. Memo	

	Cont_Rate_Pay	Rate of hourly pay for contract	Required. Currency	
	Contract_Status	The status of the contract.	Required. Lookup field.	
	Unit_ID	Charge unit's primary identifier	Required. Number. FK (tbl_Unit)	FK (Foreign Key)
	Vendor_ID	Unique identifier for Vendor	Required. Number. FK (tbl_Vendor)	FK (Foreign Key)
	Programmer_ ID	Unique identifier for Programmer	Required. Number. FK (tbl_Programmer)	FK (Foreign Key)
	Contact_ID	Unique identifier for bank contact	Required. Number. FK (tbl_Bank_Contact)	FK (Foreign Key)
	Acc_Employee_ID	Unique identifier for employee	Required. Number. FK (tbl_Acc_Employee)	FK (Foreign Key)
	Buyer_ID	Unique identifier for Buyer	Required. Number. FK (tbl_Buyer)	FK (Foreign Key)
	Division_ID	Unique identifier for the division.	Required. Number. FK (tbl_Division)	FK (Foreign Key)
tbl_Vendor	Vendor_ID	Unique identifier for Vendor	Auto number. Primary key	PK (Primary Key)
	Vendor_Number	Unique identifier for specific vendor	Required. Text. Char(40)	

	Vendor_Name	Vendor's name	Required. Char (40)	
	Vendor_Phone	Vendor's phone number	Required. Char (14) Input mask	
	Vendor_Address_1	Vendor's address	Required. Char (40)	
	Vendor_Address_2	Vendor's secondary address	Not required. Char (40)	
	Vendor_City	Vendor's city	Required. Char (20)	
	Vendor_State	Vendor's state	Required. Char (2)	
	Vendor_Zip	Vendor's zip code	Required. Char (10) Input mask	
	Vendor_Email	Vendor's email address	Not required. Char (60)	
	Vendor_Notes	Additional information for Vendor	Not required. Memo	
tbl_Programmer	Programmer_ID	Unique identifier for Programmer	Auto number. Primary key	PK (Primary Key)
	Programmer_FName	Programmer's first name	Required. Char (20)	
	Programmer_LName	Programmer's last name	Required. Char (40)	
	Programmer_Phone	Programmer's phone number	Required. Char (14) Input mask	
	Programmer_Address_1	Programmer's address	Required. Char (40)	
	Programmer_Address_2	Programmer's secondary address	Not required. Char (40)	
	Programmer_City	Programmer's city	Required. Char (20)	

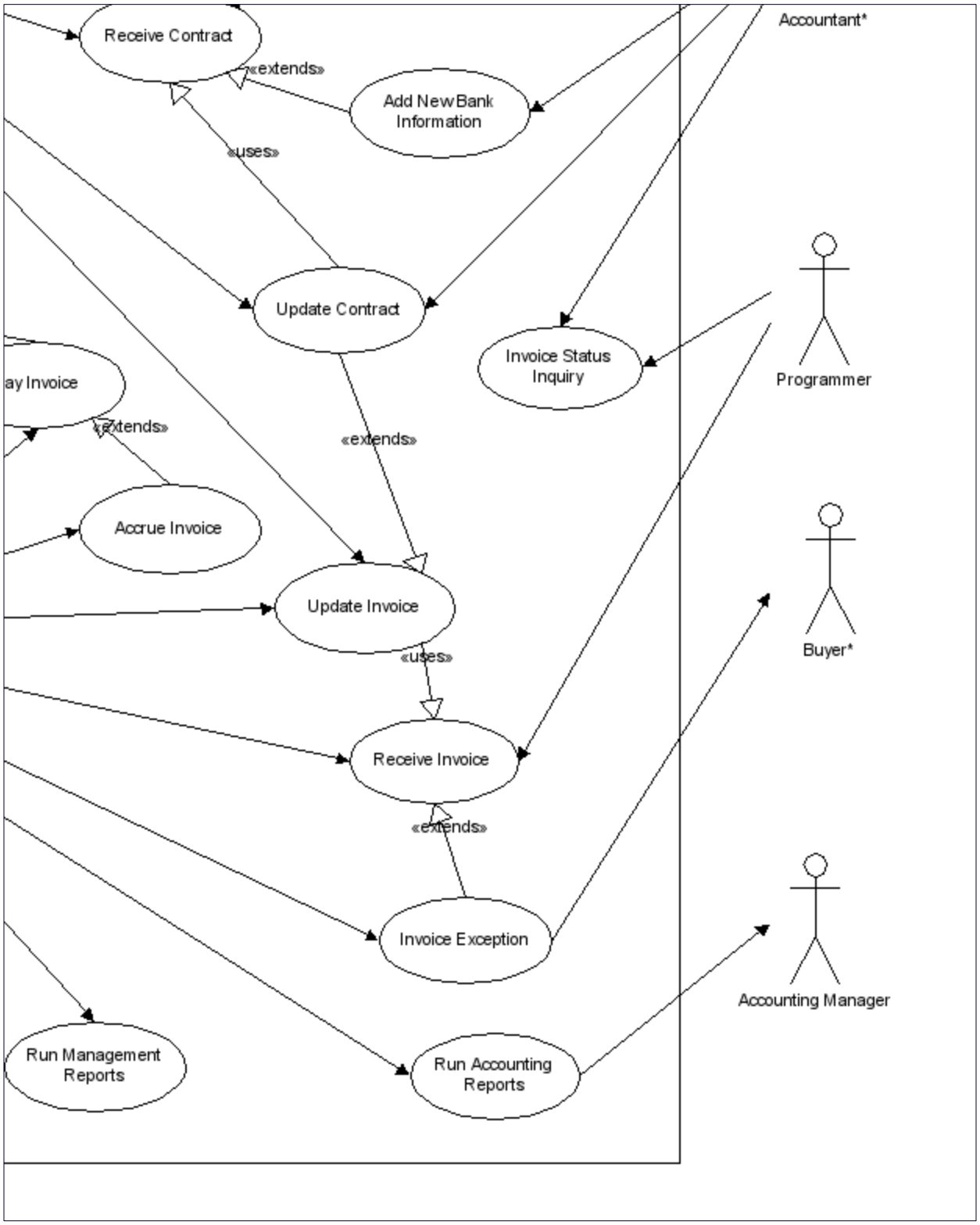
	Programmer_State	Programmer's state	Required. Char (2)	
	Programmer_Zip	Programmer's zip code	Required. Char (10) Input mask	
	Programmer_Email	Programmer's email address	Not required. Char (60)	
	Programmer_Notes	Additional notes for programmer	Not required. Memo	
	Vendor_ID	Unique identifier for Vendor	Not required. Number. FK (tbl_Vendor)	FK (Foreign Key)
tbl_Invoice	Invoice_ID	Unique identifier for Invoices	Auto number. Primary key	PK (Primary Key)
	Invoice_Start_Date	Date invoice was started	Required. Input mask	
	Invoice_End_Date	Date invoice ended	Required. Input mask	
	Invoice_Date	Date of Invoice	Required. Input mask	
	Invoice_Due_Date	Due date for invoice	Required. Input mask	
	Invoice_Terms	Terms of work for invoice	Not required. Memo	
	Invoice_Hours	Total hours of work	Required. Number	
	Work_Description	Description of work performed	Not required. Memo	
	Invoice_Notes	Any additional	Not required.	

		notes for an invoice	Memo	
	Invoice_Accrual	Date Invoice was accrued.	Not required. Input mask	
	Invoice_Accrual_Reversed	Date Invoice accrual was reversed	Not required. Input mask	
	Vendor_ID	Unique identifier for Vendor	Required. Number. FK (tbl_Vendor)	FK (Foreign Key)
	Programmer_ID	Unique identifier for Programmer	Required. Number. FK (Foreign Key)	FK (Foreign Key)
	Contract_ID	Unique identifier for Contract	Required. Number. FK (tbl_Contract)	FK (Foreign Key)
	Unit_ID	Unique identifier for unit	Required. Number. FK (tbl_Unit)	FK (Foreign Key)
	Acc_Employee_ID	Unique identifier for Employee	Required. Number. FK (tbl_Acc_Employee)	FK (Foreign Key)
tbl_Division	Division_ID	Unique identifier for Division	Auto number. Primary key	PK (Primary Key)
	Division_Name	Division's name	Required. Char (20)	
tbl_Unit	Unit_ID	Unique identifier for charge Unit	Auto number. Primary key	PK (Primary Key)

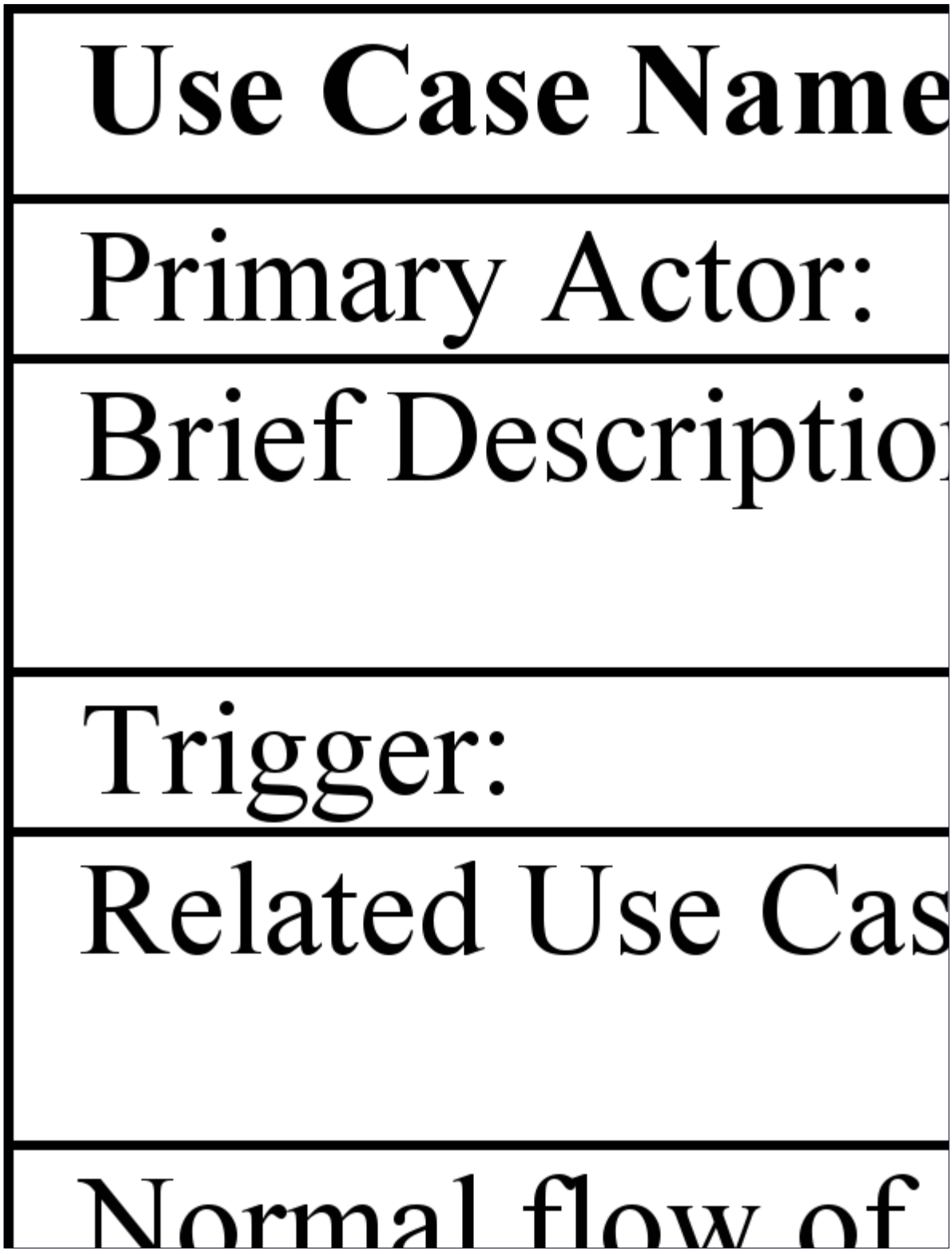
	Unit_Name	Name of unit.	Text. Required. Char (10)	
	Division_ID	Unique identifier for unit	Number. Required. FK (tbl_Unit)	FK (Foreign Key)
tbl_Bank_Contact	Contact_ID	Unique identifier for Bank Contact	Auto number. Primary key	PK (Primary Key)
	Contact_Title	Contact's title	Required. Char (20)	
	Contact_FName	Contact's first name	Required. Char (20)	
	Contact_LName	Contact's last name	Required. Char (40)	
	Contact_Phone	Contact's phone number	Required. Char (14) Input mask	
	Contact_Email	Contact's email address	Not required. Char (40)	
	Unit_ID	Unique identifier for Unit	Required. Number. Foreign key (tbl_Unit)	FK (Foreign Key)
tbl_Buyer	Buyer_ID	Unique identifier for Buyer	Auto number. Primary key	PK (Primary Key)
	Buyer_FName	First name of buyer	Required. Text char(40)	
	Buyer_LName	Last name of buyer	Required. Text char(40).	
	Buyer_Email	Email address of buyer	Not required. char(40).	
	Contract_ID	Unique identifier for Contract	Number. FK (tbl_Contract)	FK (Foreign Key)
tbl_Acc_Employee	Acc_Employee_ID	Unique identifier for Acc_Employe	Auto number. Primary key	PK (Primary Key)

		e	
	Acc_Employee_FName	Acc_Employee's first name	Required. Char (20)

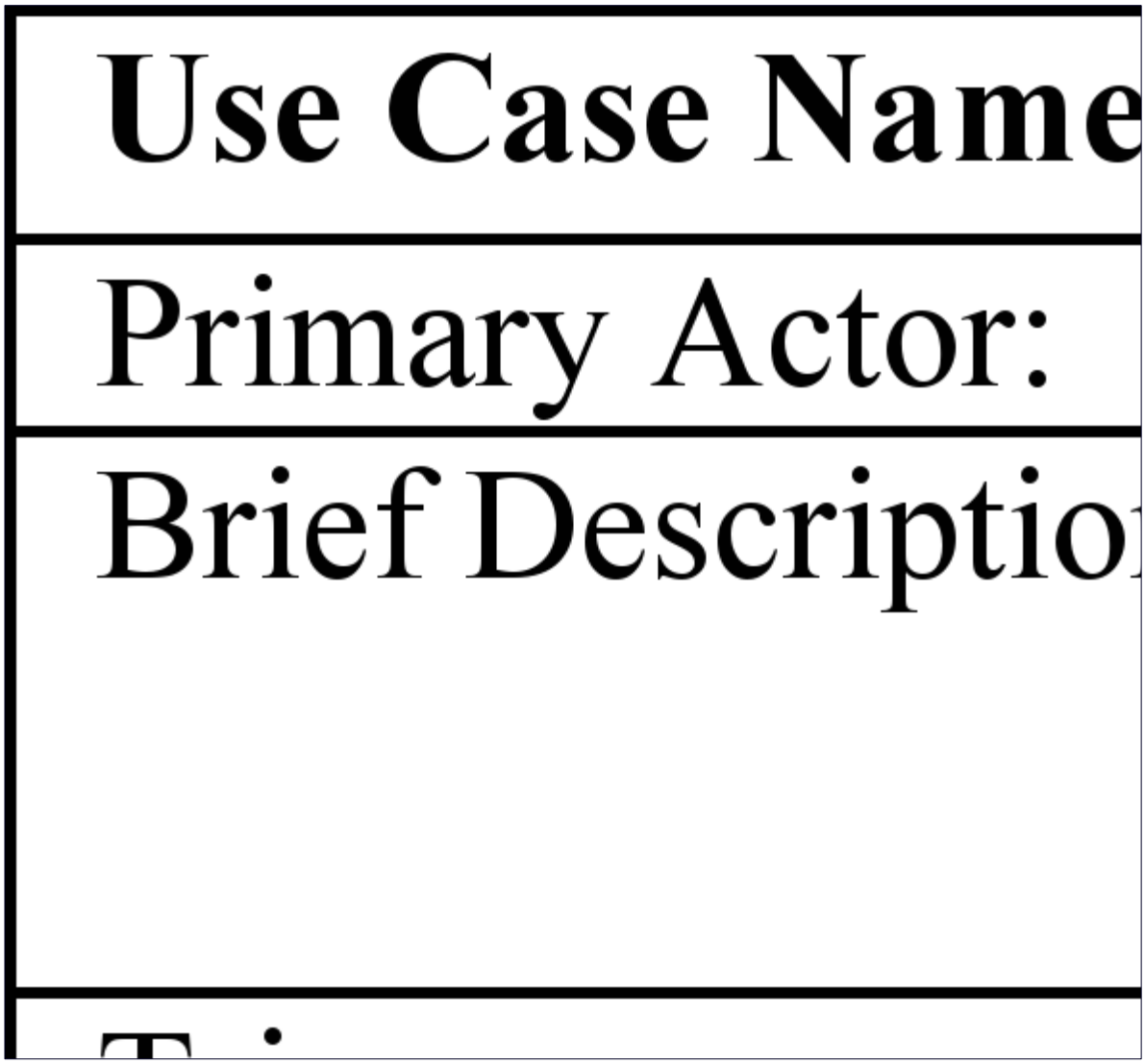
Use Case Diagram



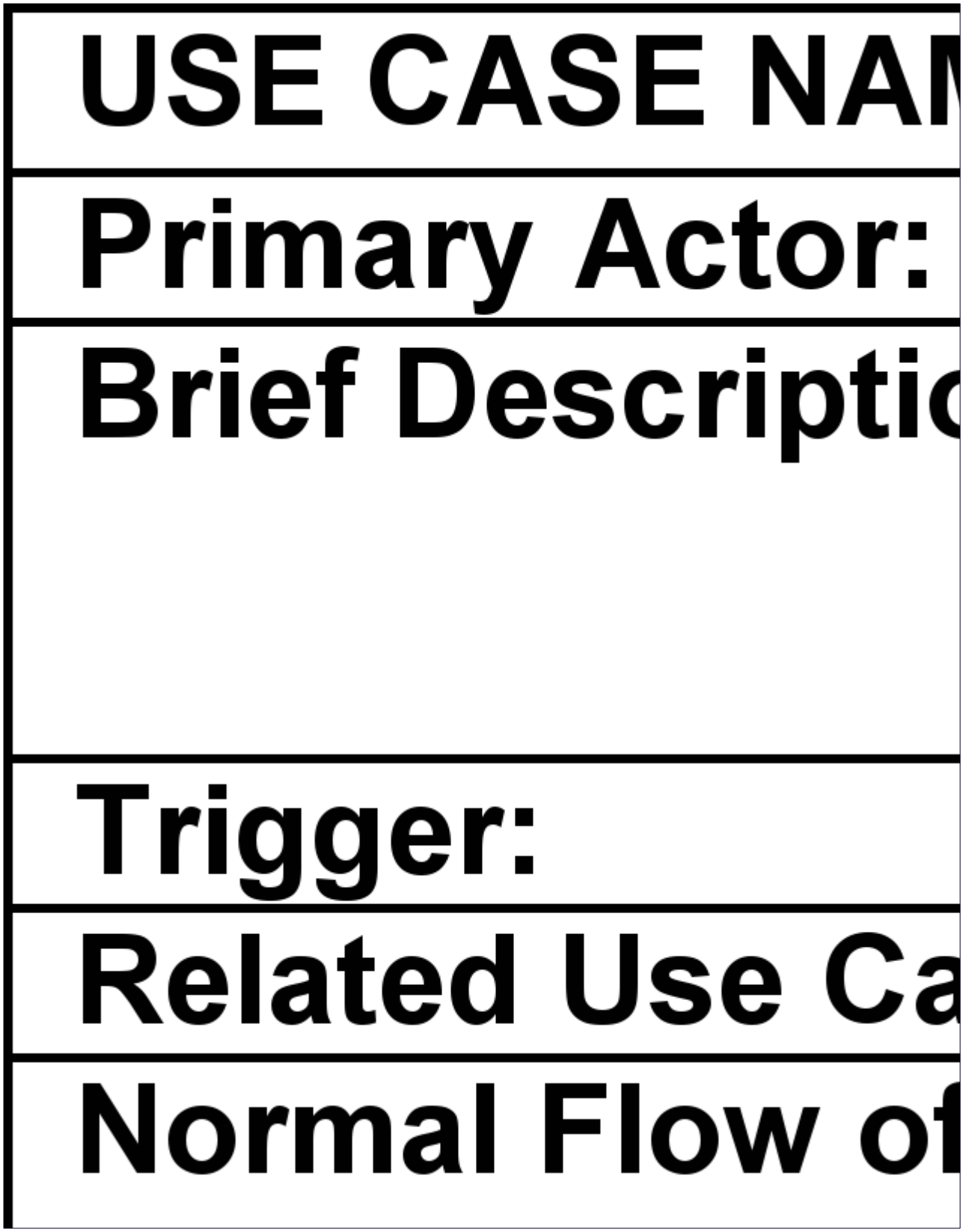
Use Case Scenarios



Use Case Name
Primary Actor:
Brief Description
Trigger:


Use Case Name
Primary Actor:
Brief Descriptio

Use Case Name
Primary Actor:
Brief Description


USE CASE NA
Primary Actor:
Brief Descriptio
Trigger:
Related Use Ca
Normal Flow of

Use Case Name
Primary Actor:
Brief Description
Trigger:
Related Use Cases:

Use Case Name
Primary Actor:
Brief Description
Trigger:
Related Use Cases:
Normal flow of

Use Case Name
Primary Actor:
Brief Description
Trigger:
Related Use Cases:
Normal flow of

Use Case Name
Primary Actor:
Brief Description
Trigger:
Related Use

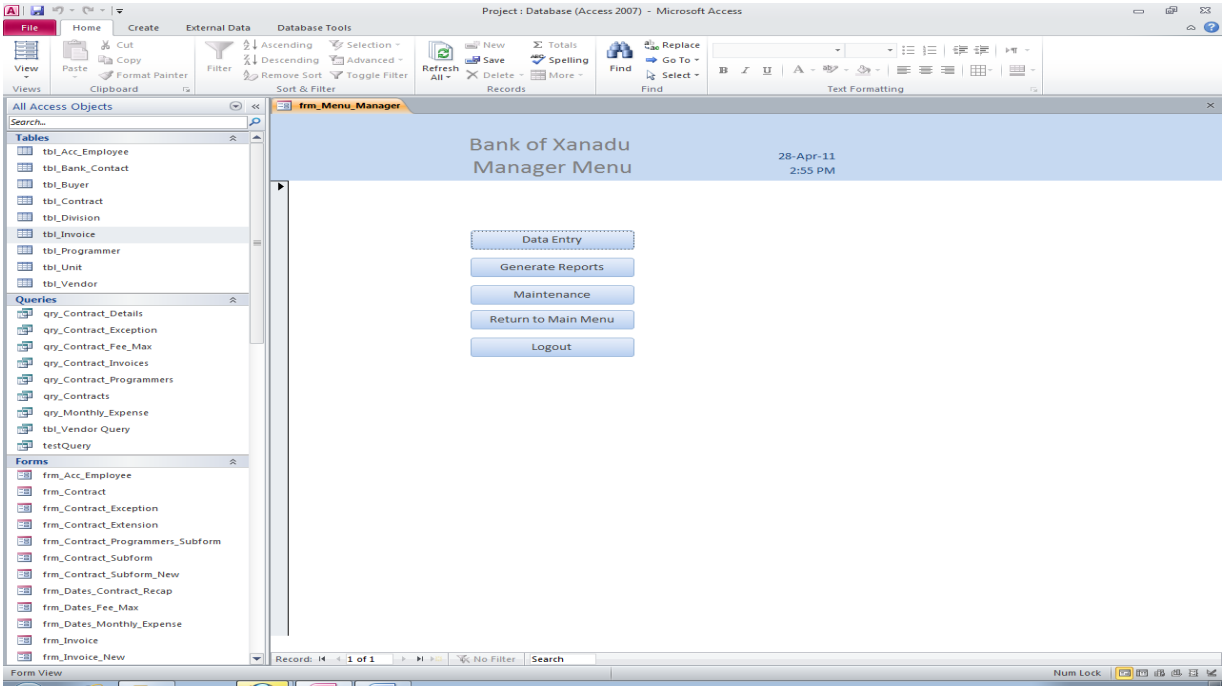
Use Case Name
Primary Actor:
Brief Description
Trigger:

Use Case Name
Primary Actor:
Brief Description
Trigger:
Related Use Cases:
Normal flow of

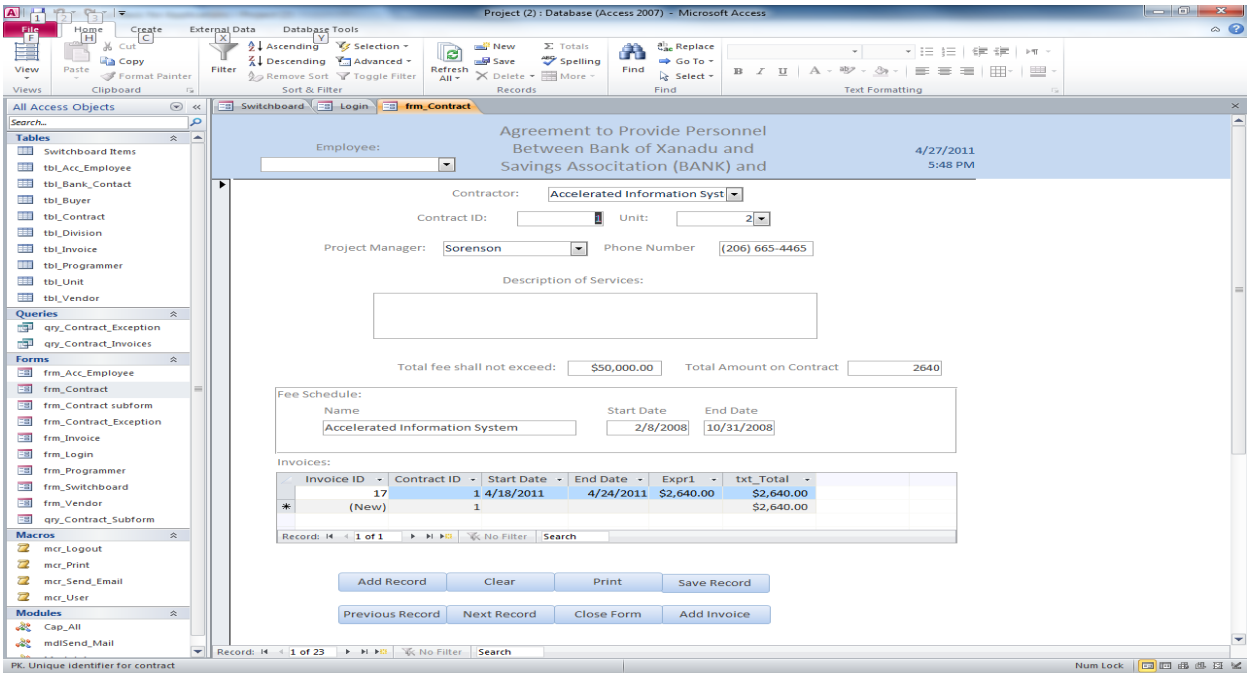
Use Case Name
Primary Actor:
Brief Description
Trigger:
Related Use Cases:
Normal flow of

Input / Output Examples

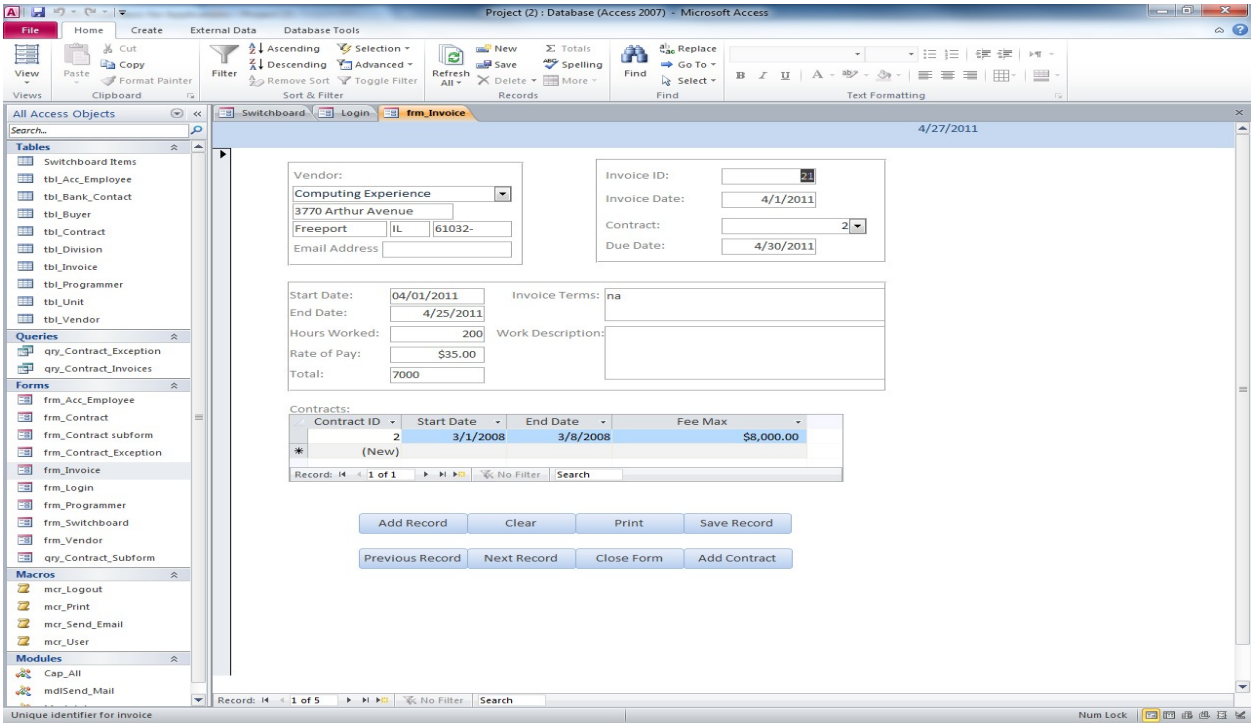
Navigation screen:



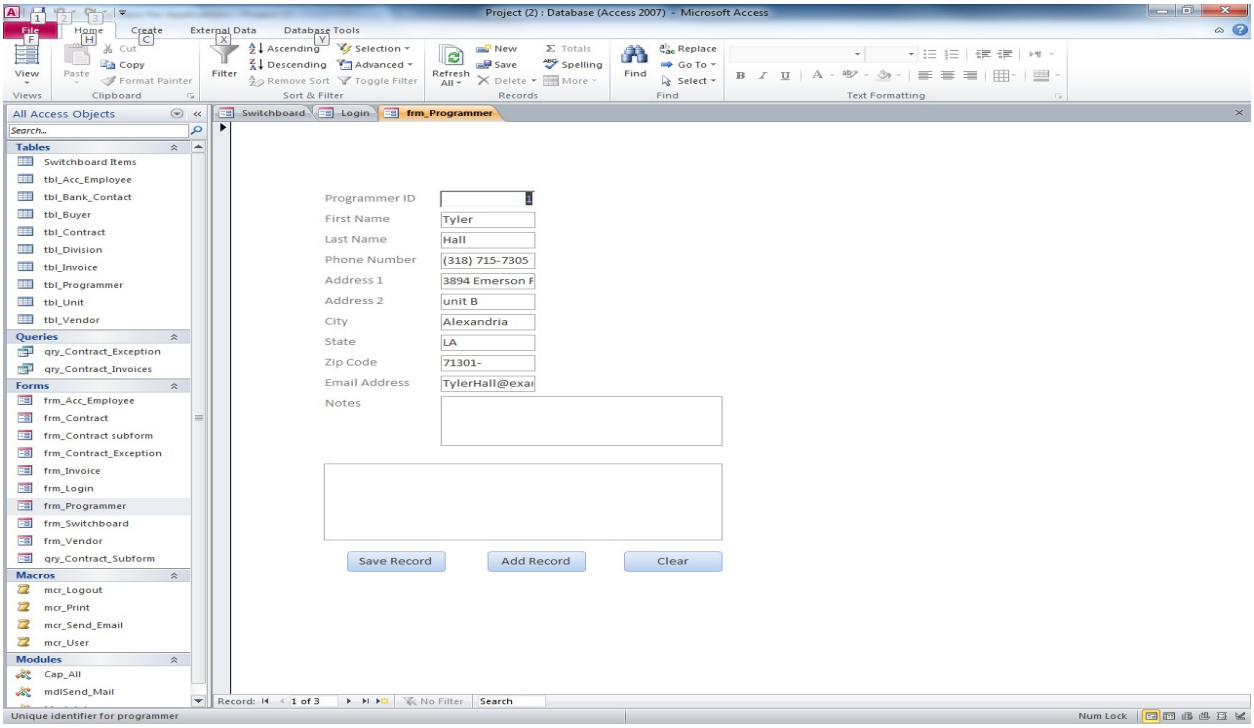
Contract screen:



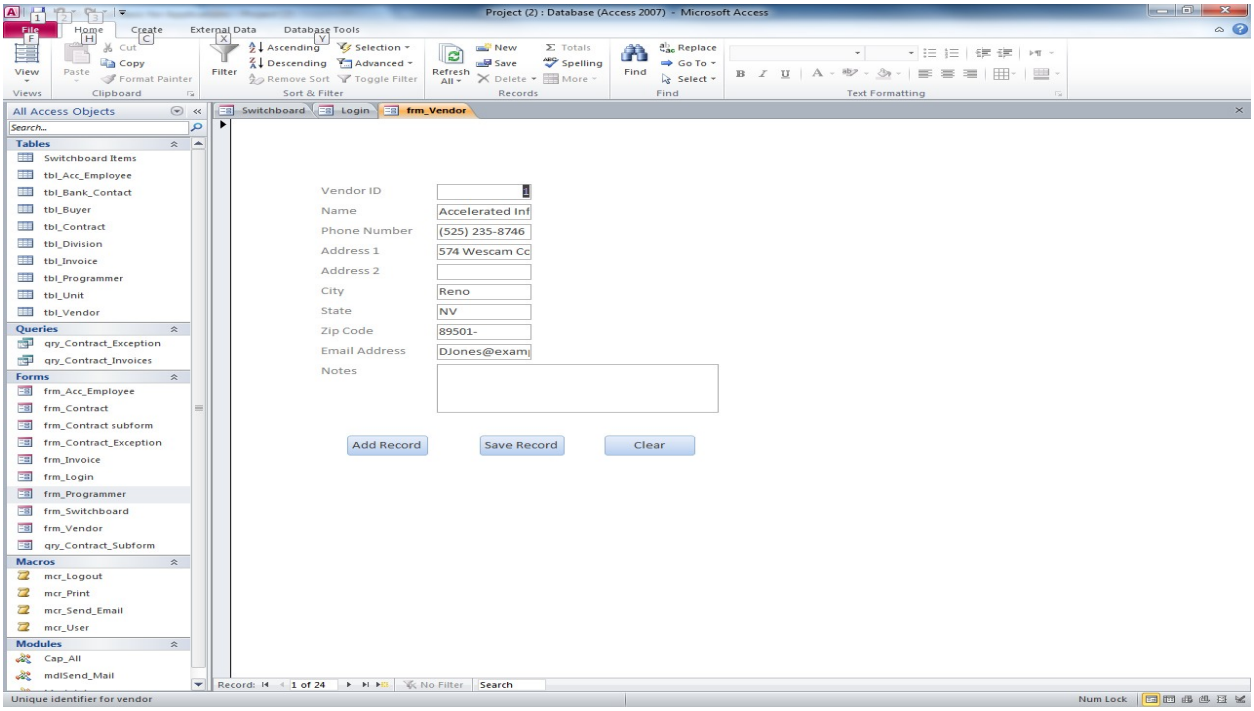
Invoice screen:



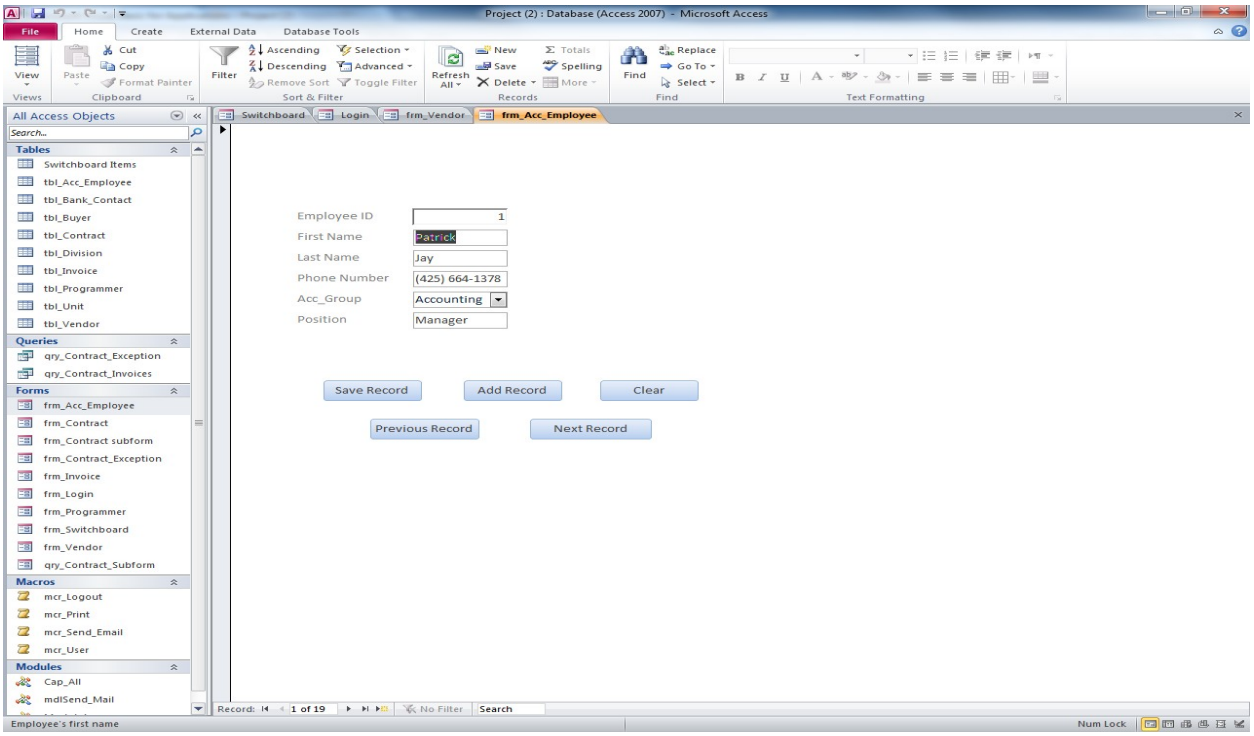
Programmer screen:



Vendor screen:



Employee screen:


Project (2) : Database (Access 2007) - Microsoft Access
File
Home
Create
External Data
Database Tools
All Access Objects
Tables
Switchboard Items
tbl_Acc_Employee
tbl_Bank_Contact
tbl_Buyer
tbl_Contract
tbl_Division
tbl_Invoice
tbl_Programmer
tbl_Unit
tbl_Vendor
Queries
qry_Contract_Exception
qry_Contract_Invoices
Forms
frm_Acc_Employee
frm_Contract
frm_Contract subform
frm_Contract_Exception
frm_Invoice
frm_Login
frm_Programmer
frm_Switchboard
frm_Vendor
qry_Contract_Subform
Macros
mcr_Logout
mcr_Print
mcr_Send_Email
mcr_User
Modules
Cap_All
mdlSend_Mail
Switchboard
Login
frm_Vendor
frm_Acc_Employee
Employee ID
1
First Name
Patrick
Last Name
Jay
Phone Number
(425) 664-1378
Acc_Group
Accounting
Position
Manager
Save Record
Add Record
Clear
Previous Record
Next Record
Record: 1 of 19
No Filter
Search
Employee's first name
Num Lock

Internal Procedures

- Reset the form to clear all text boxes and set focus to first tab stop.
- Before the user saves the form the system will check to make sure all required fields have been filled out.
- When printing, you can select:
  - Range of dates
  - All items
  - A single record
- Emailing a form directly from Access
- Capitalization of both letters in the state attribute
- Capitalization of first letter when referring to names
- When logging in the system will verify the users:
  - Name
  - Password
  - Permissions level

Issues List

We have no issues at this time.

Future Enhancement List

Should the Bank of Xanadu decide to expand the system to multiple branches we would set up a web front end using Access as the back end.

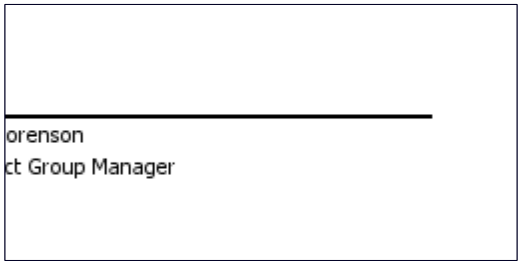
Conclusion

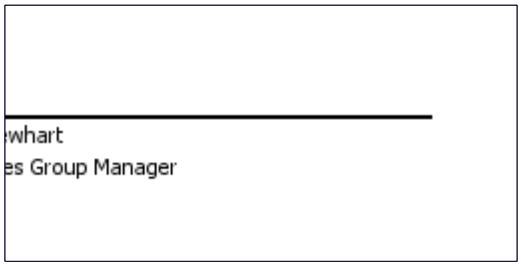
In the months since Robot Rebellion Regime entered into contract to perform an analysis of your existing Bank of Xanadu branch vendor payroll system, we believe we have successfully identified every transactional detail on which the continued success of your business relies. We have now put forth a proposal for the new database architecture. Our team is prepared to present a prototype of this database system that greatly improves the efficiency and security of your procedures, processes, technologies, and documents.

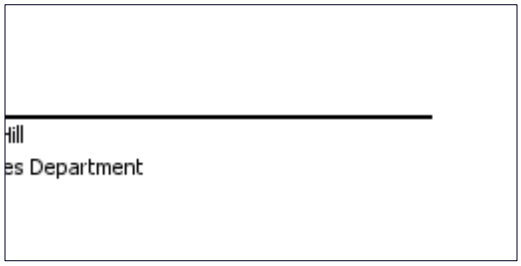
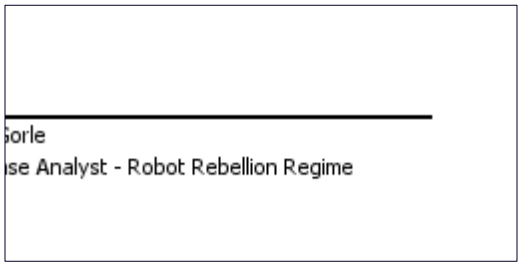
We would now like to formally request meeting for a system walkthrough on May 7, 2011, attended by the personnel tasked to access, administer, and approve this system.

It is in our best hopes that our recommended solution can easily be adapted to provide similar benefits throughout all the banking centers within the Bank of Xanadu worldwide.

Please sign the document below as confirmation of receipt and approval to







On behalf of Robot Rebellion Regime, we appreciate this opportunity to serve