Project Final Report
Due Date: 11:55 p.m. April 25, 2014
Team 1 – Interstellar Online Bank Project

1. Introduction – From Strategy to Necessity.

Online banking has evolved gradually from the early 1980's in New York with Citicorp and others using the videotext system[1]. Prior telephone banking, [2] developed in the early 1970's, began the revolution where ATM technology has transformed banking services. Further competitive pressure over the last decade and a half has transformed an online presence from a strategic advantage to an absolute necessity.

Growth of internet technology and the advancing sophistication of sites on the world-wide-web has arguably defined the look and feel of banking and carved a role for web development and online database management.

1.1 Project Overview and Statement of Proposal

Entry into the online banking market, as requested by a general financial services firm who's expanding their offerings to accommodate their customer's growing demands. The goal is for online services to act as a surrogate to the physical presence of traditional brick-and-mortar institutions.

Acceptance by consumers is envisioned by the growth of 24-hour services such as online shopping, the expansion of online financial services by insurance, stock and bond trading, and retirement management. The success of an online presence will be measured in customer growth, retention, and satisfaction rates, which subsequently advance profits.

Statement of Proposal: We propose to develop an online banking service; similar to one that would be required by a financial services firm entering the market with a business model built around online and electronic banking.

---

[1] Cronin, Mary J. (1997). Banking and Finance on the Internet, John Wiley and Sons. ISBN 0-471-29219-2 page 41 from *Banking and Finance on the Internet*, 2/4/2014 1:27:32 PM..

[2] "http://www.factmonster.com/ipka/A0801059.html." Fact Monster. © 2000–2013 Pearson Education, publishing as Fact Monster. 2/4/2014 1:27:48 PM

The online consumer banking system that is proposed will focus on customer use. It is assumed that hooks to greater offerings will be made to the prototype system that we will build; for example, loan services, tax software, and investing portfolios.

1.2 Project Scope and Objectives

The scope of the project, presented as phase one of the bank's strategic plan, is the consumer services of an online banking establishment. Ten core activities in such a system will be operationalized.

These **10 business functions** will define the requirements of the user interface and database. These functions can be seen in detail in Appendix A, which was derived from the following working list:

1) Access Control/Information Security
    a. Employees & Customer's logging in with their user name and secured password
2) New Customer Registration
    a. Adding the required and optional customer demographic information to grant access to their account information
3) New Account Creation
    a. Creating a new account number that a customer can then have access to
4) Update Existing Customer and Account Data
    a. Demographic updates and customer maintenance updates (last login, status, etc.)
5) Account Queries
    a. These are queries the bank generates as part of their administrative services
6) Prepare Monthly Statements
    a. These are queries the bank generates as part of their administrative services to their customers. This involves sending the results of these queries to the customers and has a date range (last month)
7) Track Customer Inquiries
    a. This is how the bank tracks incoming requests coming from customers, allowing them to know the status of requests that have come in and to reference previous requests made by a single customer
8) Customer Inquiries
    a. This is the customer's ability to submit requests to their bank. These could involve anything from customer service issues to requesting more checks

9) Customer Account Query
    a. These are ad hoc inquiries by the customer regarding their account information, which could involve transactional data.
10) Account Accessing
    a. Withdrawals, deposits, checks, and fund transfers
11) Bank Daily Processing
    a. Bank operation of processing transactions and posting updated balances.

2. Requirement Analysis

The online banking service has a list of business functions that are essential for the service to be usable to its users. These functionalities require data and constraints on the data to prevent the business functions from being contradicted or broken.

First, we have to understand what the users of this service need to be able to do. We referenced our personal experience as users of online banking as well as employees within the local banking industry. Based on this information we were able to come up with a reasonable list of expectations that an online-banking user has come to expect. This is how we determined our business functions.

Second, we detailed the data that was provided from these functions and worked to determine the source of each. This resulted in the attributes of our ER Diagram, which Appendix A outlines.

Third, the relationships and workflows between the entities were thought through in detail. This was to determine how our data was interacting. For instance, we had to know that customers need to have addresses and the customer is going to be the one entering these data points into the UI. This resulted, in part to, our database and UI constraints.

The following is to show the relationships that exist between the data points. This, along with the list of attributes, was used to determine the ER Diagram.

Employees and customers will have a unique (1-per) user to log into the bank's site and access the account information of the customer. The customer is able to have up to 3 accounts and the bank employee will be required to manage an unlimited number of accounts. The account will produce an unlimited number of transactions. These transactions are processed on a daily basis and *moved* to a journal. Transactions are only stored in one place, either as a transaction that hasn't been processed or

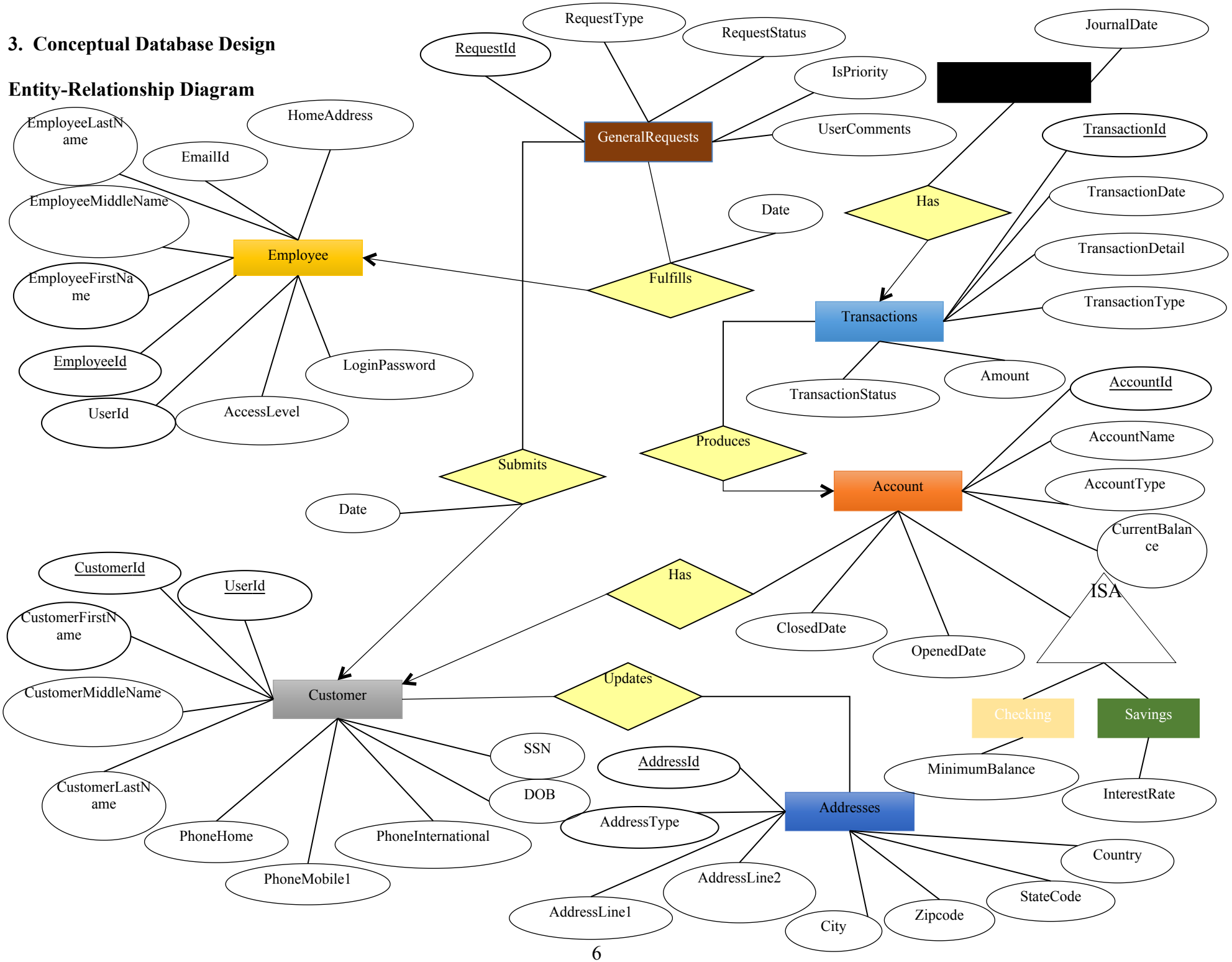
as a transaction that has been processed, with a journal date appended. Customers can have up to 3 addresses, one per address type (Business, Home, and Other); bank employees will only have their home address stored in this system. Customers will also be able to communicate to the bank through general requests; a customer can submit an unlimited number of requests to the bank. The bank employee will then be responsible for managing customer requests.

Entities with Attributes

Employee	**User**	**General Requests**
· First Name	· Access Level	· Request Type
· Middle Name	· User Role Description	· Request Status
· Last Name		· Request Priority
· Email Address		· User Comments
· Login Dates		
· Password		
Customer	**Addresses**	**Account**
· First Name	· Address Type	· Account Name
· Middle Name	· Address Line 1	· Account Type
· Last Name	· Address Line 2	· Current Balance
· Phone – Home, Mobile, International	· City	· Open Date
· Date of Birth	· State	· Closed Date
· Social Security Number	· Zip Code	
	· State Code	
	· Country	
Transactions	**Transactions Journal**	
· Transaction Type	· Transaction Type	
· Transaction Type	· Transaction Type	
· Transaction Detail	· Transaction Detail	
· Transaction Status	· Transaction Status	
· Transaction Date	· Transaction Date	
	· Journal Date	

3. Conceptual Database Design

Entity-Relationship Diagram



The Entity-Relationship Diagram is showing the *relationships* between the data required to implement this proposal, eight entities are displayed:

- User data for secure access control
- Employee information for secure access control
- Customer personal demographic information
- Addresses location and contact information
- Account information on each customer account
- Transactions information generally about all daily activities
- Transactions Journal for transaction storage receiving the daily posting from the daily activities for final reconciliations
- General Request log for various items (and/or services)

These entities produce following relationships:

- 2 has-a relationships with the customer entity
    - For account
    - For address
- Handles-a relationship with general request
    - For employee
- 1 transacts relationship with the account entity
    - For transaction
- 1 has-a relationships with transaction entity
    - For account
- 1 update-a relationships with transaction entity
    - For account
- 1 postings relationship with the general journal
    - For transaction
- 1 submits-a relationship with general request
    - For customer

3.1 ER Diagram Description

The Entity-Relationship Diagram shows the interaction between entities, which will eventually become tables. The attributes will typically become columns of the entity tables, if not they'll be separated into lookup/reference tables. It is assumed that each entity will have a primary key, which will be a unique integer that is created automatically as data is populated.

The relationships show one of three scenarios: one-to-one, one-to-many, and many-to-many. The arrows were used to display these relationships; when arrows are not used we are defining a many-to-many relationship. These relationships will result in either foreign key relationships or entity-relationship tables.

1) The flow of the ERD starts with a User, which can either be a Bank Employee or Customer. Either can exist without a User ID, but to gain access to the site they'll need to generate a User.
2) General Requests is an entity that will manage the communication/requests between a bank and customer, as well as between customer and bank.
3) Bank Employees will also be responsible for creating a customer's Account.
4) Once a Customer is linked to their Account they can use it to produce transactions.
5) These transactions will be end up in a Journal table and used to apply Credits and Debits to the Customer's Account
6) The option of including entities for the interbank clearing of funds[3] remains open as a business function which could still be implemented subject to decision making in progress.

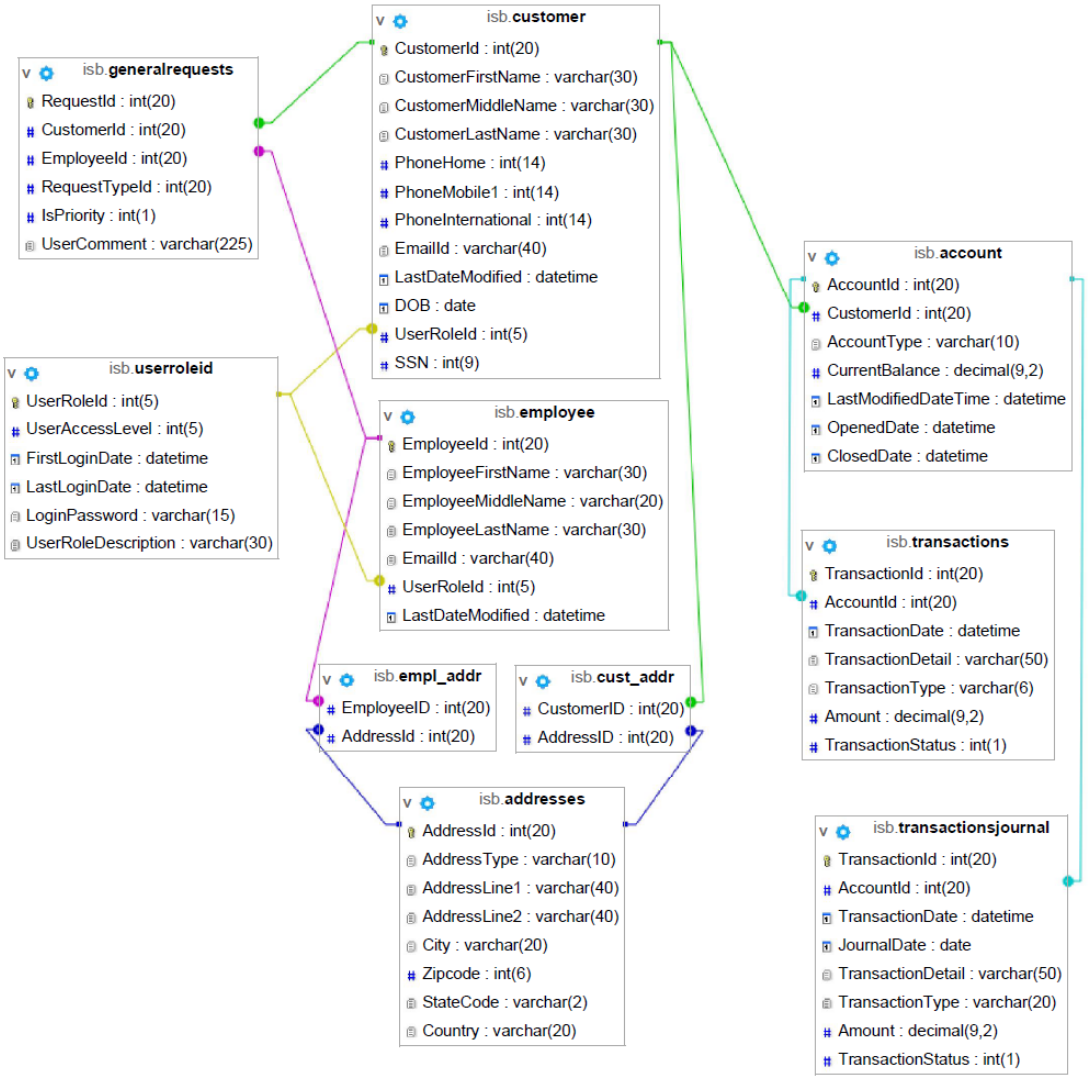
The constraints that are not shown in the ERD are listed in Appendix B.

[3] The Clearing House Payments Company, an organization owned by the same banks, was established in New York in 1853 for the purpose of processing transactions among banks. "*The Clearing House Interbank Payments System (CHIPS) A Recognition of Its 20th Anniversary*" https://www.theclearinghouse.org/about-tch/a-look-back, February 7, 2013 10:43pm

4. Database Diagram and Normalization

This section expresses the table structure via a database diagram; the diagram is for reference when reviewing the description of each table's normal form.

Database Diagram - for Referencing Normalization Explanation



4.1 Normal Form Explanation – by table

Account:

This table was normalized in the following manner in order to make it of the third normal form:

- It is of first normal form since each attribute contains only one value, and the value of each attribute is in the proper domain of that attribute; for example, AccountType contains AccountType information only, and only allows one AccountType.
- It is of the second normal form because it is of the first normal form and none of the non-prime key attributes are functionally dependent on a part of a candidate key. In this case the candidate key is the primary key. The fields in this table can assume any value irrespective of each other's value.
- It is of the third normal form because it is of the first and second normal form and every non-prime attribute of the table account is directly dependent on every super key. In the case of this table we only have one super key and it is the declared primary key.

Addresses:

This table was normalized in the following manner to make it of the third normal form:

- It is of first normal form since each attribute contains only one value, and the value of each attribute is in the proper domain of that attribute; for example, StateCode contains a two character state code, and only allows one state code.
- It is of the second normal form because it is of the first normal form and none of the non-prime key attributes are functionally dependent on a part of a candidate key. In this case the candidate key is the primary key AddressId. The fields in this table can assume any value irrespective of each other's value.
- It is of the third normal form because it is of the first and second normal form and every non-prime attribute of the table account is directly dependent on every super key. In the case of this table we only have one super key and it is the declared primary key.

Customer:

This table was normalized in the following manner to make it of the third normal form:

- It is of first normal form since each attribute contains only one value, and the value of each attribute is in the proper domain of that attribute; for example, the design is such that the name

fields contain only one individual name and they always contain individual names and no other information.

- It is of the second normal form because it is of the first normal form and none of the non-prime key attributes are functionally dependent on a part of a candidate key. In this case the candidate key is the primary key CustomerId. The fields in this table can assume any value irrespective of each other's value.
- It is of the third normal form because it is of the first and second normal form and every non-prime attribute of the table account is directly dependent on every super key. In the case of this table we only have one super key and it is the declared primary key.

Employee:

This table was normalized in the following manner to make it of the third normal form:

- It is of first normal form since each attribute contains only one value, and the value of each attribute is in the proper domain of that attribute; for example, the EmployeeLastName field will not hold anything but a single Employee last name by design.
- It is of the second normal form because it is of the first normal form and none of the non-prime key attributes are functionally dependent on a part of a candidate key. In this case the candidate key is the primary key (EmployeeId). The fields in this table can assume any value irrespective of each other's value.
- It is of the third normal form because it is of the first and second normal form and every non-prime attribute of the table account is directly dependent on every super key. In the case of this table we only have one super key and it is the declared primary key.

GeneralRequests:

This table was normalized in the following manner to make it of the third normal form:

- It is of first normal form since each attribute contains only one value, and the value of each attribute is in the proper domain of that attribute; for example, every record in this table can contain only one CustomerId value in this field.
- It is of the second normal form because it is of the first normal form and none of the non-prime key attributes are functionally dependent on a part of a candidate key. In this case the candidate key is the primary key (RequestId). The fields in this table can assume any value irrespective of each other's value.

- It is of the third normal form because it is of the first and second normal form and every non-prime attribute of the table account is directly dependent on every super key. In the case of this table we only have one super key and it is the declared primary key.

Transactions:

This table was normalized in the following manner to make it of the third normal form:

- It is of first normal form since each attribute contains only one value, and the value of each attribute is in the proper domain of that attribute; for example, TransactionDate contains a date value and only allows one date value at a time.
- It is of the second normal form because it is of the first normal form and none of the non-prime key attributes are functionally dependent on a part of a candidate key. In this case the candidate key is the primary key. The fields in this table can assume any value irrespective of each other's value.
- It is of the third normal form because it is of the first and second normal form and every non-prime attribute of the table account is directly dependent on every super key. In the case of this table we only have one super key and it is the declared primary key.

TransactionsJournal:

This table was normalized in the following manner to make it of the third normal form:

- It is of first normal form since each attribute contains only one value, and the value of each attribute is in the proper domain of that attribute; for example, the AccountId contains account id information only, and only allows one account id per field.
- It is of the second normal form because it is of the first normal form and none of the non-prime key attributes are functionally dependent on a part of a candidate key. In this case the candidate key is the primary key which is the TransactionId. The fields in this table can assume any value irrespective of the other fields.
- It is of the third normal form because it is of the first and second normal form and every non-prime attribute of the table account is directly dependent on every super key. In the case of this table we only have one super key and it is the declared primary key.

User:

This table was normalized in the following manner to make it of the third normal form:

- It is of first normal form since each attribute contains only one value, and the value of each attribute is in the proper domain of that attribute; for example, the UserAccessLevel consistently maintains only one access level and no other type of information.
- It is of the second normal form because it is of the first normal form and none of the non-prime key attributes are functionally dependent on a part of a candidate key. In this case the candidate key is the primary key. The fields in this table can assume any value irrespective of each other's value.
- It is of the third normal form because it is of the first and second normal form and every non-prime attribute of the table account is directly dependent on every super key. In the case of this table we only have one super key and it is the declared primary key.

Cust_Addr:

This table was normalized in the following manner to make it of the third normal form:

- It is of first normal form since each attribute contains only one value, and the value of each attribute is in the proper domain of that attribute; for example, the CustomerId field holds only one integer value.
- It is of the second normal form because it is of the first normal form and none of the non-prime key attributes are functionally dependent on a part of a candidate key. In this case the candidate key is the primary key. The fields in this table can assume any value irrespective of each other's value.
- It is of the third normal form because it is of the first and second normal form and every non-prime attribute of the table account is directly dependent on every super key. In the case of this table we only have one super key. The primary key would be composed of the combination CustomerId and AddressId in order to enforce uniqueness and eliminate redundancy.

Empl_Addr:

This table was normalized in the following manner to make it of the third normal form:

- It is of first normal form since each attribute contains only one value, and the value of each attribute is in the proper domain of that attribute; for example, the user the EmployeeID field holds only one integer value.
- It is of the second normal form because it is of the first normal form and none of the non-prime key attributes are functionally dependent on a part of a candidate key. In this case the candidate key is the primary key. The fields in this table can assume any value irrespective of each other's value.
- It is of the third normal form because it is of the first and second normal form and every non-prime attribute of the table account is directly dependent on every super key. In the case of this table we only have one super key. The primary key would be composed of the combination EmployeeID and AddressId in order to enforce uniqueness and eliminate redundancy.

4.2 SQL Statements for Creating Entities and Relationships

Due April 6th – Separate Document

4.3 SQL Statements for 10 business functions

Due April 6th – Separate Document

5. Team Dynamics

5.1 Project Status

The team has been meeting each Saturday for three hours at J. Murrey Atkins Library. We've used our in-person meeting time to review the progress made, troubleshoot conflicting approaches, and list the deliverables for the next meeting. Countless discussions via chat and video have made the development of our project moving smoothly. Collaboration is streamlined by quickly troubleshooting through concerns of team members as they arise; little lag exists in getting answers to continue development.

Prior to the first version of the report we had focused on the business functions and how they'd be implemented into a database, via the ER Diagram. Since the first version of the report was posted the group has been unofficially placed into three subgroups. All of the group members collaborate outside of their subgroup, but within their group they've been tasked with meeting one of the following deliverables:

Business Rule Documentation

- This group is heavily involved in the creation of this document and working towards the presentation at the end of the semester. Our goal is to interpret the requirements of the project and see that they are met by the database and user interface. The core group members with project management experience are ensuring that we stay on track and everyone has input into the final product.

Database Implementation

- This group had the most collaboration across the team since it's the subject of the class. Most people on the team had input towards the design of the database and at a minimum understood why the design choices were made. The most experienced members wrote the SQL code, which was quality checked by members of both the user interface and business rule groups. As business rules and the user interface changes this team will be responsible for implementing change requests.

User Interface

- We had the opportunity to leverage group members with experience in PHP, JavaScript, and Python that is being used to implement the user interface of the database. This subgroup initially collaborated to determine how the business rules would be implemented on the website. More

recently their goal is to take those designs and implement them with the database. Finalizing the UI screens and linking it to the database will be the most active task of our group until the presentation.

Some tasks were dependent on the completion of others, making it essential for timelines to be implemented; this lead us to use Microsoft Project. At a high level we are currently working through the below deliverables. Each week we meet, this list is being updated to determine the next week's deliverables.

Task Name
Functions
Updated List of Functions
SQL Functions
Update DDL according to ERD
Making changes to ERD as needed
UI Mockup/SiteMap
Implementing UI Mockups
Domain setup / Server / Web hosting
Turning functions into UI Tabs
Merging Documentation
Rewrite requirement analysis
Normalization Documentation
Presentation
Create slide deck
Schedule Practice Runs
Designate speakers to topics

5.2 Team #1 Members

Pierre Arbajjan

Michael Burke

Jinfeng Du

David Hardister

Jacob Highfill

Priyanka Kochhar

James Lu

Naveen Mysore

Shelby Setzer

David York

Appendix A: Business Functions and Data Requirement to Support

Business Functions to be Covered by Project		Data Requirements	Entities Derived
Function Groups	*Business Functions(Activities)*	*(Data Dependencies)*	
1.Service Administrative	1. Access Control / Information Security	User (Access) Information	Access Control Plus Employee or Customer
(Bank staff intervention required or for internal bank functions)	2. New Customer registration	Customer information	Customer
		Employee Information	Employee
	3. New Account Creation	Customer Information	Customer
		Account Information	Customer
	4a. Upgrade existing customer data	Customer Information	Customer
	4b. Upgrade existing account data	Account information	Account
	5. Accounts Queries	Account Information	Account
			Transactions
		Transaction Information	Customer
			long-term Transactions
	6. Prepare monthly statements	Customer Information	Customer
		Account Information	Account
		Transaction Information	Long Term Transactions
	7. Track Customer inquiries	Message or Service Record Information	Contact Record History
		Customer Information	Customer
		Employee Information	Employee
2. Customer Originated or Completed Activities	8. Customer Inquiries (General)	Message or Service Record Information	Contact Record History
	9. Customer Account Query: balances, history	Account Information Transaction Information	Account Long Term Transactions Plus/Minus Day Transaction
	10. Account Accessing (a) Withdrawals (query sufficient Funds)	Account Information Transaction Information	Account Days Transactions
	(b) Deposits		Account Day Transaction
	(c) Checks (query sufficient funds)		Account Day Transaction
	(d) Fund transfers: inter-account		Account Day Transaction

	inter-bank		Long Term Transactions
3. Bank Operations Accounting / Business Records Maintenance	11. Bank daily processing – (internal) transaction postings to bank (daily) journal	Daily Transactions Activity Accounting Database (Gen_Journal)	Daily Transaction Activity Long-term Transactions (Gen_Journal)

Appendix B: DATA CONTRAINTS NOT ON ERD

Entity	Constraint
Customer Addresses	One CustomerId can only have one AddressId per AddressTypeId
Customer Account	AccountId must exist in the Account Table
Customer Account	CustomerId must exist in the Customer Table
Transactions	One AccountId per TransactionId
Transactions	AccountId must exist in the Account Table
Transactions	Status must be either Pending or Processed
Users	One CustomerId per UserId
Users	One EmployeeId per UserId
Users	EmployeeId must exist in the Employee Table and cannot have a position = 'Former Employee'
Journal Transactions	IsProcessed must equal either 1 or 0
Journal Transactions	AccountId must exist in the Account Table
Journal Transactions	JournalDate must be < Current DateTime
Employee	Employee must have correct delegation to open new accounts (i.e. New Employee in Training may not be freely open or close accounts)
Customer	Our Online Banking System should concentrate on Personal Customers therefore no Business Customers at this stage
Address	Data Collection should be limited to Physical and Mailing Address only
Address	No duplicate addresses under the same address type
Transaction	Status could also be something like denied if the transaction was not allowed to go through for some reason, unless this is a sub-category in processed
Customer Date of Birth	Must be a past date
Customer Date of Birth	Must have been from at least x years ago
General Requests	Priority Status must be either urgent, standard, or low
Account	Currency must be greater than or equal to zero. Any transactions attempting to go below zero are rejected at the customer’s expense (no negative balance)