Consolidated Data Warehouse System

Katherine Mayo & Malialosa Taupule

Final Project Report

CIS 341

Table of Contents

Project Overview

Two merging accounting firms are in need of a new and consolidated database system that is easy to use, easily accessible, and secure. Throughout this report we will distinguish the two firms by its relative sizes to one another: the smaller firm and the larger firm. The smaller firm currently utilizes a SQL database; whereas, the larger firm currently utilizes a data warehouse system. The issues that arise with this decentralized database system are insufficient security and limited data accessibility.

The newly improved system will utilize Cloud software, specifically Oracle's Big Data Cloud services, where all data from every source system associated with the firm is kept and maintained in one centralized area. The Cloud is the most cost efficient consolidation method, saving our client in areas like IT support, typically provided by the Cloud vendor, and hardware. Users of a cloud-based system are able to access data and other resources instantly. This eliminates wasted time spent offline data extractions that would originally occur if the systems remained separate. Because the cloud is hosted over the internet, users can interface the system on other platforms, specifically mobile. Cloud provides optimum security, providing high level security protocols to authorize users and data constraints to keep data incorruptible. The new system will use a dashboard-based interface that is compiled depending on the user type, which is determined at login. This personalized dashboard will provide relevant information relating to the user's job function, making the user's daily functions that much easier.

Systems Planning Phase

Current System's Analysis

Our client is an accounting firm looking to consolidate its database system. It is currently using two separate database systems: one SQL Server database organized in a proprietary schema, used by the smaller firm, and a data warehouse, used by the larger firm. Some issues that our client is looking to alleviate through a system change are:

- An outdated system
- A decentralized database
- A system vulnerable to data leaks and network sniffing due to insufficient security measures
- Data within associated offices are inaccessible by outside managers, making performance reporting inaccurate

Business Processes

Because our client is an accounting firm, much of the business processes revolve around financial operations. Our client is an accounting service provider to small businesses. It offers a variety of services, such as personalized financial consultation packages to financial documentation preparations. Illustrated below in *Table 1* is a few of our client's main business processes.

Table 1 - Business Process Table

Process	Objective	Input	Output	Tasks	Outcomes
Pricing & Invoicing	To determine service prices to be used to compile a running total of charges to be sent/billed to client	Converted from current SQL Server DB	1. Current pricing of services are made available online for clients to reference 2. Invoices are made available online for clients to reference in their personal billing account	1. Refer to currently used pricing model to determine optimal price for each service 2. For each service performed to a client, incur a charge for current period's invoice 3. At the end of the current period total charges incurred 4. Bill client with the invoice total	Pricing and invoices are kept up to date and are available for clients to access conveniently on their billing account. Client invoices is also stored in a central location accessible by any authorized employee within the firm.
Manage Client Financial Records	To store client's financial data used to assist in firm's provision of accounting services	Converted form current SLQ Server DB	Client's information and financial records can be shared electronically within the system to: 1. Assist firm's employees in providing personalized financial services to client 2. To be referenced to by client to assess his/her business' current financial status	1. Track client's transactions made this period: add, delete, or modify each transaction 2. Categorize each client's transactions made during this period as a revenue or an expense 3. At the end of each period, make sure all record reconciliation is complete	All client financial records are stored in a central location for auditors to process. Approved records are then processed and stored. Read-only reports of these records are made available for clients to reference at any time via their account portal.
Client Financial Auditing and Reporting	To perform overviews of all client financial records to ensure that each record is correctly inputted and fully compliant with current accounting regulations	Financial information currently stored within the new consolidated system	Financial records are accurate and are compliant with current accounting regulations.	1. Ensure records comply with laws and regulations 2. Analyze internal processes and accounting systems 3. Ensures necessary data is present and that all data is correct. 4. Based on the preceding tasks, if no discrepancies arise, approve record; otherwise, send back to accounting department for revision.	All records are kept up to date, accurate, and compliant to all current accounting regulations.

Use Cases

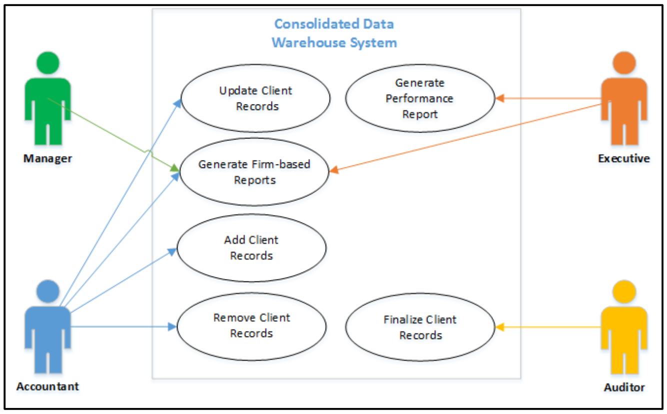


Figure 1 - New System's Use Case Diagram

As shown in the *Figure 1*, there are four main actors that we plan to interact with our new system: Managers, Accountants, Executives, and Auditors. An accountant's main role in the firm is to enter and modify existing client records. Manager's generate firm-based reports to make business-related decisions. Like managers, executives will be able to generate firm-based reports, to help asses a specific office's performance, as well as general performance reports which assesses the performance of the entire company. Lastly, an auditor's main role is to finalize client reports. This is to make sure that data entered by the accountants follow current regulations to avoid future penalties for the client. The following *Tables 2-4* illustrate a few of the diagrammed use cases above in more details:

Table 2 - Use Case: Generate Firm-Based Report

Name of Use Case	Generate Firm-based Reports
Actor	Manager, Executive, Accountant
Description	Managers at any firm will be able to generate their own firm's ad-hoc ("as the occasion requires") reports by simply specifying the type of data to be pulled via a form provided by the system.
Successful Completion	If manager, all of his/her firm's financial and client data is accessible to be pulled for reporting Manager specifies which data to be pulled via system form System pulls user-specified data from new database System outputs to screen the analytical data the user had specified
Alternative	If not manager, If an accountant, only client data is accessible to be pulled for reporting If an executive, all firm's financial data is accessible to be pulled for reporting Report is in read-only format System pulls user-specified data from data warehouse System outputs to screen the data in an organized fashion
Precondition	Upon access to the new database, the user is prompted to log in.
Post condition	User is identified as a manager and a report of the user-specified is generated
Assumptions	User login information is stored within the system

Table 3 - Use Case: Generate Performance Reports

Name of Use Case	**Generate Performance Reports**
Actor	Executive
Description	Top executives who direct the firm will be able to generate performance based reports that focus on highlighting the overall health of their company. They will also be able to generate performance reports for specific firms, if desired.
Successful Completion	Executive specifies desired report type System generates reports by pulling specified data relevant to report type Reports are ready to read
Alternative	Executive specifies desired report type Report does not exist or data because data for report is not updated System will prompt "Does not exist" or "N/A" error message if report is unavailable Report will also show a "last updated" time stamp for Executive to assess whether or not the data is up-to-date
Precondition	Executive generates/accesses company-wide reports from database system.
Post condition	Necessary changes are made to improve company performance
Assumptions	New database system will generate performance reports

Table 4 - Use Case: Add Client Records

Name of Use Case	Add Client Records
Actor	Accountant
Description	Accountants can add new client records to the system.
Successful Completion	Accountant searches for client (to be inputted into system) System verifies that client does not preexist in DB Accountant is given the ability to add new client information (via completing a company-approved form)
Alternative	Accountant searches for client (to be inputted into system) System verifies that client does exist in DB Accountant is given the ability to update (rather than add) record
Precondition	Accountant logs into new database system and is granted access to client records
Post condition	Client financial and personal records are accurate and created/updated.
Assumptions	None

Budget Plan

Because our client’s larger firm houses a data warehouse. Hardware costs were not of concern with this project. It is the software costs that dominate, due to consolidating the databases via the Cloud. The pricings were generated from the Oracle’s Big Data Cloud Services pricing webpage along with other non-Oracle sources that provided reliable information pertaining to the small hardware costs pertaining to the implementation of the Cloud (Beal, "How Much Will a Server Cost?”) (Mor, "How Much Does It Cost...”).

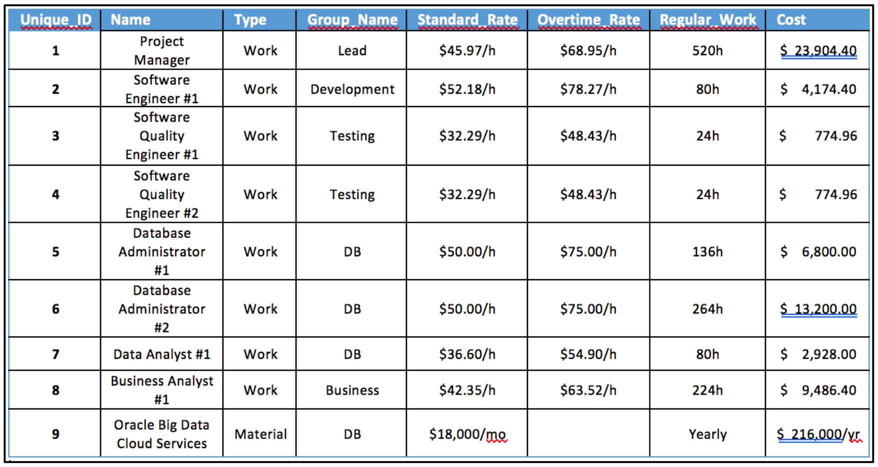
Table 5 - Our Project's 6 Month Budget Summary

Consolidated Data Warehouse Project 6 Month Budget Summary						
Description	**Priority**	**Apr-May**	**Jun-Jul**	**Aug-Sep**	**6 Month Totals**	**Ongoing Estimate**
OnGoing - Operations Training Detail by Area						
Operational Area	2	$6,000.00	$6,200.00	$6,400.00	**$18,600.00**	$4,000.00
Others		$-	$-	$-	**$-**	$-
Projects Summary						
Data Warehouse System	1	$167,660.00	$13,360.00	$13,360.00	**$194,380.00**	$24,400.00
Data Warehouse System - Evaluation/Audit	1	$-	$-	$20,200.00	**$20,200.00**	$20,000.00
OnGoing - Operations Detail						
Required Software	1	$2,000.00	$1,000.00	$1,000.00	**$2,000.00**	$-
Required Hardware	1	$1,000.00	$-	$-	**$1,000.00**	$-
Planned Software	2	$3,000.00	$-	$-	**$3,000.00**	$-
Planned Hardware	2	$1,000.00			**$1,000.00**	$-
Totals per year and Ongoing		**$180,660.00**	**$20,560.00**	**$40,960.00**	**$240,180.00**	**$48,400.00**

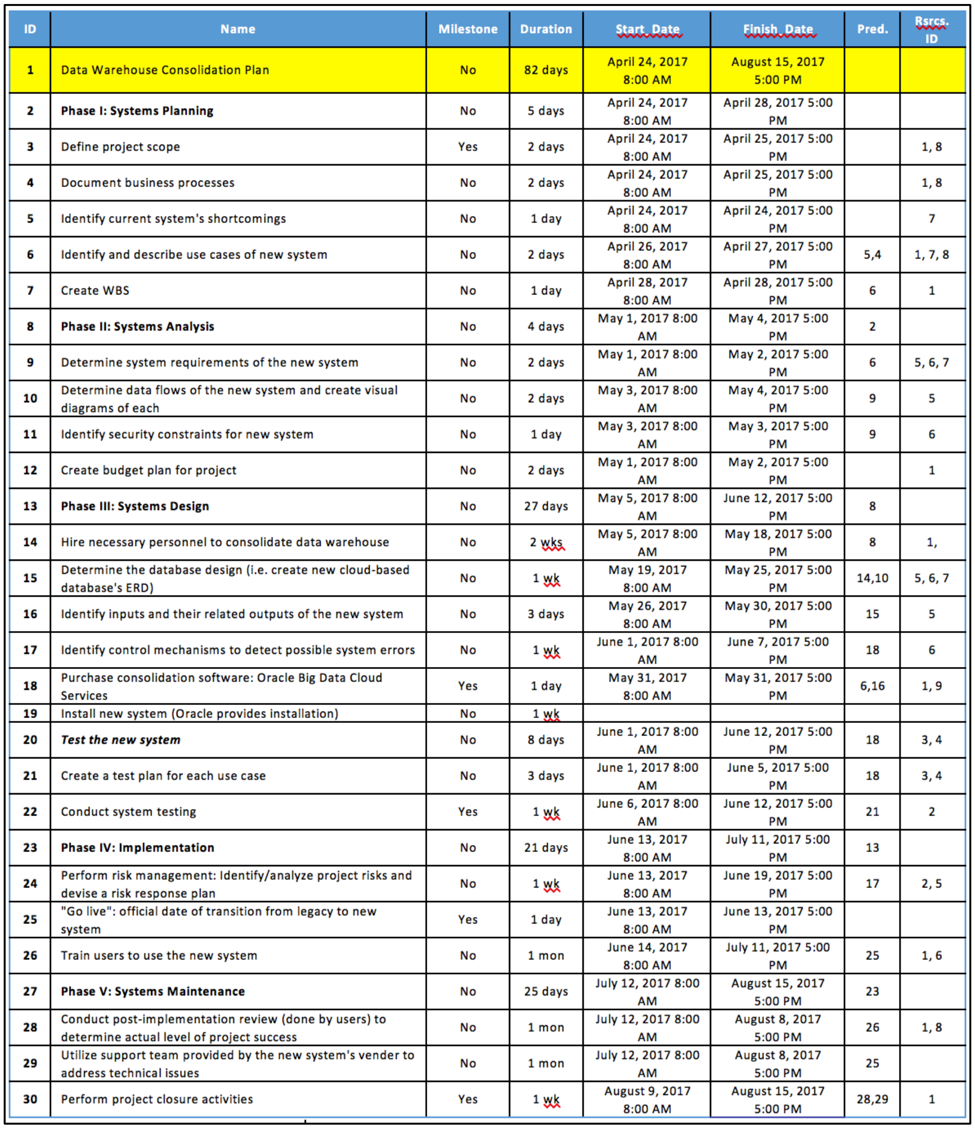
Project Plan

This project is expected to last 82 days, beginning on April 24, 2017 and ending August 9, 2017 if every task is completed on time. *Figure 2* shows the resources utilized for this project. The specific task breakdown of our project plan is shown in *Figure 3* below.

Figure 2 - Project Resources Table



Unique_ID	Name	Type	Group_Name	Standard_Rate	Overtime_Rate	Regular_Work	Cost
1	Project Manager	Work	Lead	$45.97/h	$68.95/h	520h	$ 23,904.40
2	Software Engineer #1	Work	Development	$52.18/h	$78.27/h	80h	$ 4,174.40
3	Software Quality Engineer #1	Work	Testing	$32.29/h	$48.43/h	24h	$ 774.96
4	Software Quality Engineer #2	Work	Testing	$32.29/h	$48.43/h	24h	$ 774.96
5	Database Administrator #1	Work	DB	$50.00/h	$75.00/h	136h	$ 6,800.00
6	Database Administrator #2	Work	DB	$50.00/h	$75.00/h	264h	$ 13,200.00
7	Data Analyst #1	Work	DB	$36.60/h	$54.90/h	80h	$ 2,928.00
8	Business Analyst #1	Work	Business	$42.35/h	$63.52/h	224h	$ 9,486.40
9	Oracle Big Data Cloud Services	Material	DB	$18,000/mo		Yearly	$ 216,000/yr



ID	Name	Milestone	Duration	Start_Date	Finish_Date	Pred.	Rsrcs. ID
1	Data Warehouse Consolidation Plan	No	82 days	April 24, 2017 8:00 AM	August 15, 2017 5:00 PM		
2	**Phase I: Systems Planning**	No	5 days	April 24, 2017 8:00 AM	April 28, 2017 5:00 PM		
3	Define project scope	Yes	2 days	April 24, 2017 8:00 AM	April 25, 2017 5:00 PM		1, 8
4	Document business processes	No	2 days	April 24, 2017 8:00 AM	April 25, 2017 5:00 PM		1, 8
5	Identify current system's shortcomings	No	1 day	April 24, 2017 8:00 AM	April 24, 2017 5:00 PM		7
6	Identify and describe use cases of new system	No	2 days	April 26, 2017 8:00 AM	April 27, 2017 5:00 PM	5,4	1, 7, 8
7	Create WBS	No	1 day	April 28, 2017 8:00 AM	April 28, 2017 5:00 PM	6	1
8	**Phase II: Systems Analysis**	No	4 days	May 1, 2017 8:00 AM	May 4, 2017 5:00 PM	2	
9	Determine system requirements of the new system	No	2 days	May 1, 2017 8:00 AM	May 2, 2017 5:00 PM	6	5, 6, 7
10	Determine data flows of the new system and create visual diagrams of each	No	2 days	May 3, 2017 8:00 AM	May 4, 2017 5:00 PM	9	5
11	Identify security constraints for new system	No	1 day	May 3, 2017 8:00 AM	May 3, 2017 5:00 PM	9	6
12	Create budget plan for project	No	2 days	May 1, 2017 8:00 AM	May 2, 2017 5:00 PM		1
13	**Phase III: Systems Design**	No	27 days	May 5, 2017 8:00 AM	June 12, 2017 5:00 PM	8	
14	Hire necessary personnel to consolidate data warehouse	No	2 wks	May 5, 2017 8:00 AM	May 18, 2017 5:00 PM	8	1,
15	Determine the database design (i.e. create new cloud-based database's ERD)	No	1 wk	May 19, 2017 8:00 AM	May 25, 2017 5:00 PM	14,10	5, 6, 7
16	Identify inputs and their related outputs of the new system	No	3 days	May 26, 2017 8:00 AM	May 30, 2017 5:00 PM	15	5
17	Identify control mechanisms to detect possible system errors	No	1 wk	June 1, 2017 8:00 AM	June 7, 2017 5:00 PM	18	6
18	Purchase consolidation software: Oracle Big Data Cloud Services	Yes	1 day	May 31, 2017 8:00 AM	May 31, 2017 5:00 PM	6,16	1, 9
19	Install new system (Oracle provides installation)	No	1 wk				
20	***Test the new system***	No	8 days	June 1, 2017 8:00 AM	June 12, 2017 5:00 PM	18	3, 4
21	Create a test plan for each use case	No	3 days	June 1, 2017 8:00 AM	June 5, 2017 5:00 PM	18	3, 4
22	Conduct system testing	Yes	1 wk	June 6, 2017 8:00 AM	June 12, 2017 5:00 PM	21	2
23	**Phase IV: Implementation**	No	21 days	June 13, 2017 8:00 AM	July 11, 2017 5:00 PM	13	
24	Perform risk management: Identify/analyze project risks and devise a risk response plan	No	1 wk	June 13, 2017 8:00 AM	June 19, 2017 5:00 PM	17	2, 5
25	"Go live": official date of transition from legacy to new system	Yes	1 day	June 13, 2017 8:00 AM	June 13, 2017 5:00 PM		
26	Train users to use the new system	No	1 mon	June 14, 2017 8:00 AM	July 11, 2017 5:00 PM	25	1, 6
27	**Phase V: Systems Maintenance**	No	25 days	July 12, 2017 8:00 AM	August 15, 2017 5:00 PM	23	
28	Conduct post-implementation review (done by users) to determine actual level of project success	No	1 mon	July 12, 2017 8:00 AM	August 8, 2017 5:00 PM	26	1, 8
29	Utilize support team provided by the new system's vender to address technical issues	No	1 mon	July 12, 2017 8:00 AM	August 8, 2017 5:00 PM	25	
30	Perform project closure activities	Yes	1 wk	August 9, 2017 8:00 AM	August 15, 2017 5:00 PM	28,29	1

Figure 3 - Project's Work Breakdown Structure

Systems Analysis Phase

New System's Requirements

Illustrated in *Table 6* is our project team's specific requirements listing composed to aid our development team in choosing a proper consolidation method:

Table 6 - New System's Requirements Table

Req. ID	Requirement	Priority	Current Custom Solution	Details	Critical	Test ID
SR-1	**Data is accessible by all authorized employees regardless of physical location**	Must Have	There is no solution. A firm can only access the data stored in its local database.	- Consolidate the entire database system used by all firms through virtualization	H	TC-7
SR-2	**Data entered by any source system within the company is stored in a central location**	Must Have	Central data warehouse must perform an offline incremental data extraction for each source system in order to access data.	- Requires a central database accessible via online - Data warehouse has either direct or intermediate access to all source systems a part of the company - Data in data warehouse is organized using NoSQL	H	TC-8
SR-4	**System's user interface is very user friendly and requires none-little technical knowledge of back-end operations**	Must Have	To analyze firm and client data, the user must know SQL in order to query the firm's associated database.	- Online forms would need to be created so that user could enter data in a logical and understandable way (e.g. fill in different fields with valid input) - Input validation would need to be established for these forms - User interface will consist mainly of a dashboard; depending on the user type determines the information shown on the dashboard	H	TC-9
SR-5	**System will time-out if user is inactive for at least 5 consecutive minutes**	Nice to Have	Reminders are placed along all monitors to remind users to log out.	- A timer is initialized when the user signs in; every time the user is inactive, the timer begins. When the user interacts with the system, the timer is restarted.	L	TC-6
SR-6	**User authentication is done via system sign in**	Must Have	There currently is a user sign in to access data; however, user types are indistinguishable meaning all data is mutable by any employee logged in.	- To perform work-related duties, a user must first sign into the system with their employee ID and password. - This login process must very secure. - All user login information is stored in the system. - Unrecognized users are blocked from entering the system.	H	TC-2
SR-7	**System is also mobile-friendly**	Nice to Have	There is no current solution. The current database is only accessible via a desktop.	- Functionalities between the two platforms must differ when needed (e.g. accountants can only enter records via desktop) - System must be programmed to be compatible on both platforms.	M	TC-1

New System’s Processes, Data, and Logical Data Flows

Data flowing between the newly consolidated system is illustrated by *Figure 4***.** All system users interact with the new system via the application server provided by Oracle Big Data Cloud Services. The data is then processed by the cloud’s database and transferred to the firm’s data warehouse, which is local to the large firm. This gives all authorized users, validated by the Kerberos authentication process, the ability to access the data. This also allows all data, whether it originated from the small firm or another source system, to be stored in a central location. Business intelligence operations can then be performed on the data to generate company-wide reports.

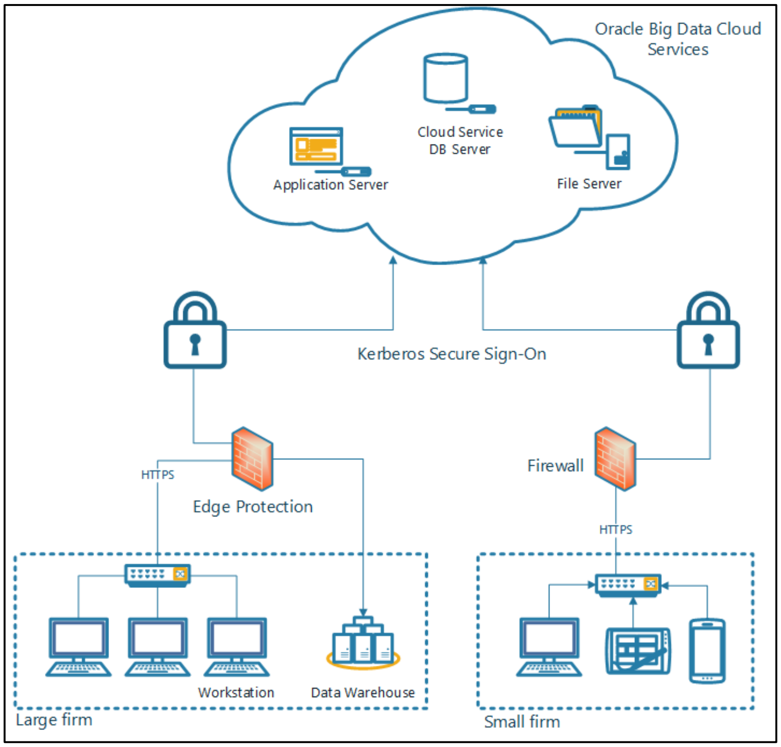


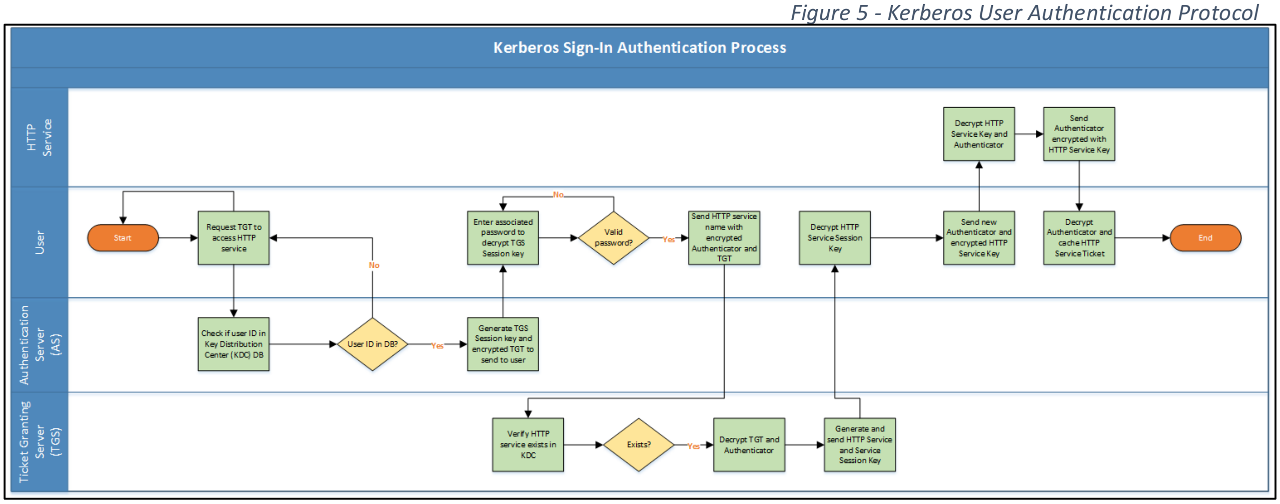
Figure 4 - New System's data flow facilitated by OBDCS

Business Process Swim Chart

Kerberos Sign-In Authentication Process

Outlined below in *Figure 5* is the Kerberos user authentication process, used by Oracle’s Big Data Cloud Services software. In the Kerberos security protocol, a user must obtain a “ticket” by the Key Distribution Center (KDC) to be allowed to access the requested host (*Oracle Big Data Cloud Services*). Following the user’s attempt to sign-in is a long series of encrypted messages that are exchanged between the user and the KDC or HTTP service site to verify the user’s identity (*Oracle Big Data Cloud Services*).

Figure 5 - Kerberos User Authentication Protocol


Kerberos Sign-In Authentication Process
HTTP Service
User
Authentication Server (AS)
Ticket Granting Server (TGS)
Start
Request TGT to access HTTP service
Check if user ID in Key Distribution Center (KDC) DB
User ID in DB?
No
Yes
Generate TGS Session key and encrypted TGT to send to user
Enter associated password to decrypt TGS Session key
Valid password?
No
Yes
Send HTTP service name with encrypted Authenticator and TGT
Verify HTTP service exists in KDC
Exists?
Yes
Decrypt TGT and Authenticator
Generate and send HTTP Service and Service Session Key
Decrypt HTTP Service Session Key
Send new Authenticator and encrypted HTTP Service Key
Decrypt HTTP Service Key and Authenticator
Send Authenticator encrypted with HTTP Service Key
Decrypt Authenticator and cache HTTP Service Ticket
End

Security Constraints

Cloud systems are known for its optimum security. For the new system, Cloud will contain several security features such as least privilege, restricted access, and PCI compliance to ensure secure data storage and usage.

Restricted Access

User authorization is necessary upon Cloud access. Different user types will have different levels of authorization:

- **Executive**: Access to all records, including summaries and firm performance records
- **Manager**: Generates firm-based reports
- **Accountant**: Adds, edits, and removes client records; access to client financial reports
- **Client**: Access to financial reports (read only)

Each user is given a specific user ID that denotes his/her user type. Each user's personalized dashboard is dependent on the user type. If the user ID and/or password is invalid, the user is denied access into data system and will eventually be locked out of the system if too many attempts have been made. Once logged in, however, the user, no matter the type, will be granted respective access into the data system. Each login session is timed and if the user is inactive for more than five minutes, the system will automatically log out, to ensure confidentiality.

PCI Compliance

Because clients will be able to pay for their services online, the system must implement the Payment Card Industry Data Security Standard (PCI DSS). Because the firm will have access to cardholder information, it is necessary that sensitive data such as this is stored securely. To be PCI compliant, the new database system will build and maintain a secure network through methods such as: including firewalls and securing system passwords; encrypting data to ensure data is incorruptible; implementing strong access control measures to track and monitor resources and cardholder data; and running regular test security systems and processes ("What Is PCI Compliance?").

Systems Design Phase

Project's Development Process

Development Process

The project is estimated to have a 82-day duration. Prior to project planning, the accounting firm was able to articulate their expectations and requirements for the new database system well. This, therefore, made the project easier to plan and easier to section into phases - beginning with Phase I: System Planning, then moving to Phase II: Systems Analysis, Phase III: Systems Design, Phase IV: Implementation, and finally ending with Phase V: Systems Maintenance. Because the project is well defined and divided into phases, the waterfall method would be ideal to use due to its sequential. The waterfall method goes through each stage of the project and will not proceed until each stage is fully completed. At the end of each stage, a review is done to ensure the project is going in the right direction. Also, because the accounting firm's requirements simply desire to add new features and additions to their already existing database, little involvement is needed from the firm throughout the project's duration. With the waterfall methodology, projects usually move at a quicker pace because they do not need to constantly stop and check in with the client throughout each phase. This will allow the project team to carry out its job in a much more efficient manner (Lotz, "Waterfall vs. Agile: Which Methodology is Right for Your Project?").

Project Team

The project team will be divided into departments each focusing on a specific part of the project - known as the team-based approach. Each department is given a particular responsibility in each phase of the project with deadlines in place to keep the project on track. In this way, the team-based approach ensures the best coverage for all aspects of the project and maximizes the team's skills and resources, as well as overall productivity. Each department or group is required to submit a weekly report to the project manager, detailing their progress and accomplishments, as well as addressing any issues that they may be having. This allows the project manager to be aware of the team's overall progress allowing him/her to make any necessary changes to the project to ensure that the project is complete by its expected due date (Duggan, "The Advantages of Team-Based Structure").

New Database's Design

As the system had to provide online data processing and analytics, as well as a centralized system for all employees to use, we chose to consolidate the two systems by means of virtualization. This would be done through the purchase of the Oracle Big Data Cloud Services (OBDCS) package which is built around the Apache Hadoop and Apache Spark systems which stores its data in clusters. OBDCS provides a wide variety of benefits to the client ranging from a more expanded user functionality to faster and more efficient storage of data. One of the key features of this service is that it is optimized to run a diverse set of workloads structured under both the SQL and noSQL approach (*Oracle Big Data Cloud Services***)**. This would allow the system to be dynamically compatible to all data, having a flexible storage scheme common under the noSQL approach, as well as an interactive interface through the querying method provided by SQL.

To analyze the data, software automatically included with OBDCS, like Oracle's R Distribution software, work to aid the user by providing advanced analytics for both Hadoop and SQL workloads (*Oracle Big Data Cloud Services*). Oracle Big Data Spatial and Graph, another included software, provides even more functionality by providing data visualizations to enhance the user's understanding of the stored data (*Oracle Big Data Cloud Services*) Oracle interface provides a dashboard to allow users to view and analyze stored data.

Finally, to secure the data, OBDCS provides strong authentication process using Kerberos, ensuring that users who attempt to log in are who they claim to be (*Oracle Big Data Cloud Services*). Furthermore, the software provided is programmed to do data-at-rest and network encryptions to prevent "network sniffing" by unauthorized users attempting to capture the stored. To add another layer of security, OBDCS has Hadoop security in place to prevent the unauthorized querying of data.

Because data stored within a NoSQL database is both structured and non-structured, the diagrams to represent the NoSQL, or in the OBDCS' case, Hadoop schema, may vary (Katsov, "NoSQL"). As shown in *Figure 6*, the data is stored both either a normalized or aggregated form, depending on the data type. This figure models the user types stored within the system. Each user has a common attribute, an ID. However, each user type (i.e. Accountant, Manager, Executive, or Client), has its own attribute set. Because of NoSQL's "soft schema", a schema that allows one to form classes of entities with complex internal structures (nested entities), a single user could be diagramed using a single aggregate instance, which contains all attribute values of that instance (Katsov, "NoSQL").

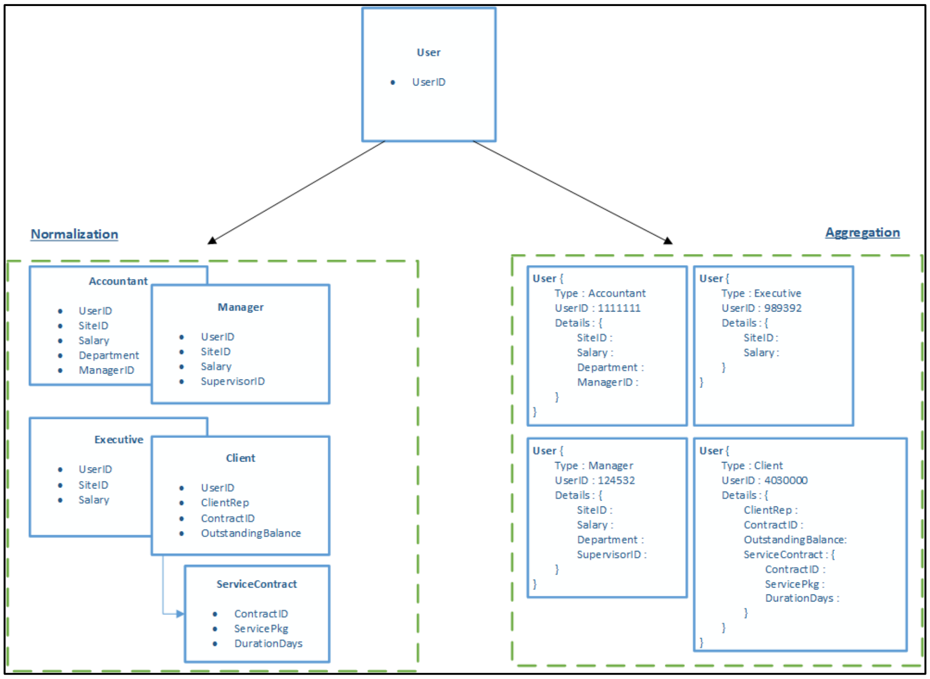
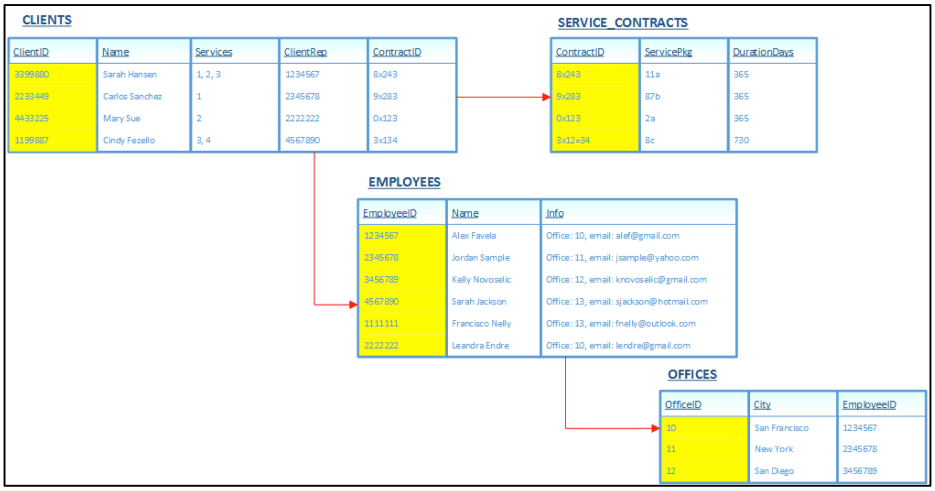

User
UserID
Normalization
Aggregation
Accountant
UserID
SiteID
Salary
Department
ManagerID
Manager
UserID
SiteID
Salary
SupervisorID
Executive
UserID
SiteID
Salary
Client
UserID
ClientRep
ContractID
OutstandingBalance
ServiceContract
ContractID
ServicePkg
DurationDays
User { Type : Accountant UserID : 1111111 Details : { SiteID : Salary : Department : ManagerID : } }
User { Type : Executive UserID : 989392 Details : { SiteID : Salary : } }
User { Type : Manager UserID : 124532 Details : { SiteID : Salary : Department : SupervisorID : } }
User { Type : Client UserID : 4030000 Details : { ClientRep : ContractID : OutstandingBalance: ServiceContract : { ContractID : ServicePkg : DurationDays : } } }

Figure 6 - New System's Schema using the NoSQL Aggregates Modeling Technique



CLIENTS

ClientID	Name	Services	ClientRep	ContractID
3399880	Sarah Hansen	1, 2, 3	1234567	8x243
2233449	Carlos Sanchez	1	2345678	9x283
4433225	Mary Sue	2	2222222	0x123
1199887	Cindy Fezello	3, 4	4567890	3x134

SERVICE_CONTRACTS

ContractID	ServicePkg	DurationDays
8x243	11a	365
9x283	87b	365
0x123	2a	365
3x12=34	8c	730

EMPLOYEES

EmployeeID	Name	Info
1234567	Alex Favela	Office: 10, email: alef@gmail.com
2345678	Jordan Sample	Office: 11, email: jsample@yahoo.com
3456789	Kelly Novoselic	Office: 12, email: knovoselic@gmail.com
4567890	Sarah Jackson	Office: 13, email: sjackson@hotmail.com
1111111	Francisco Nelly	Office: 13, email: fnelly@outlook.com
2222222	Leandra Endre	Office: 10, email: lendre@gmail.com

OFFICES

OfficeID	City	EmployeeID
10	San Francisco	1234567
11	New York	2345678
12	San Diego	3456789

Figure 7 - New System's Schema Using the NoSQL Index Table Technique

Another way to diagram our new system's schema is also shown in *Figure 7* which highlights an important aspect of our database: file indexing. Because our database extracts all data from each source system and stores each data cluster in its raw form, our database can store complex files, such as PDFs. For example, our database will be able to store service contracts in PDFs simply by attaching an index before stored into the contract repository. Each contract file would have its own data attributes like service package, associated client, length of contract, etc. When the system needs to access a service contract, it will search for its memory location using the index value (Katsov, "NoSQL"). The system would then pull and transform the data into the necessary format.

Object and Methods Design

Objects & Attributes:

Class SourceSystem This class is used to create objects to represent the source systems (i.e. databases) local to the company's associated remote offices. The data stored within each of these systems will be processed via ETL to integrate into the company's centralized data warehouse. The Cloud will have its own system ID for sites that input their data directly to the cloud.

Attributes Name, Address, SystemID, ClientRecords, SiteRecords

Class Client: This class is used to create instances of clients served by a specific firm. These objects will be used to associate a client with its own financial records maintained by the employees who serve them. The employees who serve them can use these records to create specialized service plans for each client.

Attributes SiteID, ClientID, Name, Address, Industry, ClientRecords

Class OfficeSite: This class is used to create instances of the different sites owned by the firm. Each site will either store its data in a source system (data will be extracted via ETL) or directly to the Cloud. Also, each firm will have its own set of financial records and client records.

Attributes SiteID, SourceSystem, Address

Class ClientRecord: This class is used to create instances of client records to be stored into the system. Each record will keep track of the current service(s) the client has purchased from the firm, the amount owed for each service, and the amount paid for current services.

Attributes RecordID, LastModified, CurrentServices, AmountOwed, AmountPaid

Class SiteRecord: Similar to ClientRecord, this class is used to create instances of a site's financial records to be stored into the system. These records are used by the middle and top-level managers to assess the performance of each site relative to the company's other sites. To differentiate between the different financial records, each record will have a record type (e.g. Accounts Receivable, Accounts Payable, etc.).

Attributes RecordID, LastModified, RecordType, EmployeeID (ID of accountant who entered record)

Class Data: This class is used to create instances of the different data clusters processed by the Cloud and stored both in the Cloud and the data warehouse. An important attribute of this class is its index used by the system to locate each data cluster. Along with its index, each data cluster will also have an ID to identify the site that inputted it (either by direct input or extraction) into the system.

Attributes Index, SourceID

Methods:

Description: This method will be used by the system to extract data from the source system to be loaded.

Precondition: srcID is a valid identifier of the source system used for extraction

Postcondition: Data from source system is returned to be further processed and loaded into the system.

Method 1: ExtractData(SourceSystem srcID) -- this method extracts data from the SQL Server DB to eventually be loaded into the data warehouse → I can write down the specifics to this method

```
IF srcID is not valid,
        PRINT error message
        EXIT abnormally
ELSE,
        EXTRACT data
        ASSIGN data's sourceID to site's ID
        ASSIGN data container to store extracted data
        RETURN data container
```

Description: This method will be used by the system to load extracted data into the source system.
Precondition: None
Post condition: Data is now stored in system. Transformation is not done until the data is needed for use (i.e. via the Schema on Read approach).

Method 3: LoadData(vector<Data> extractedRawData)

```
IF extractedRawData is NULL
            PRINT error message
            EXIT abnormally
      ELSE
            WHILE there is more data to load
                  SEARCH system for next free index
                  ASSIGN free index to raw data
                  STORE raw data at free index
```

Description: This method will be used by the system to transform the data into the proper format so that it could be used by the system.
Precondition: dataIndices cannot be NULL; this vector is used by the system to locate the data stored in the system to be transformed.
Post condition: Data is transformed into the necessary format.

Method 2: TransformData(vector<Data> dataIndices)

```
IF dataIndices is NULL
            PRINT error message
            EXIT abnormally
      ELSE
            WHILE there are more data indices
                  INDEX system to data's index
                  READ data
                  TRANSFORM data
```

Description: This method will be used to generate an ad-hoc report of a client's financial records stored in the system for either the client's use or client's financial consultant's use.
Precondition: clientID is an existing client
reportType is a valid identifier for the type of report to be generated
Post condition: A report is generated and outputted to screen.

Method 4: GenerateClientReport(long clientID, int reportType)

```
IF clientID is invalid OR reportType is invalid
        PRINT error message
        EXIT abnormally
    ELSE
        ASSIGN calculations to be done based on reportType
        FOR each relevant calculation
            SEARCH system for necessary operands (i.e. relevant client data)
            STORE operands
            CALCULATE report results using operands
            STORE report results into temporary container
        FOR each temporary container
            OUTPUT stored report result in a user-friendly format
```

Description: This method is used to generate an ad-hoc report of a site's financial records stored in the system to be reviewed by management.

Precondition: siteID is a valid ID
reportType is a valid identifier for the type of report to be generated

Post condition: A report is generated and outputted to the screen.

Method 5: GenerateSiteReport(long siteID, int reportType)

```
IF siteID is invalid OR reportType is invalid
        PRINT error message
        EXIT abnormally
    ELSE
        ASSIGN calculations to be done based on reportType
        FOR each relevant calculation
            SEARCH system for necessary operands (i.e. necessary site data)
            STORE operands
            CALCULATE report results using operands
            STORE report results into temporary container
        FOR each temporary container
            OUTPUT stored report result in a user-friendly format
```

Controls

To ensure the accuracy and security of the data, the new data system will include control mechanisms to detect and prevent future errors from occurring. Especially dealing with client's private information and monetary records, precision is essential to not only the company's performance but for the client's security as well. The following methods will be some of the control mechanisms used by the database to prevent such misusage.

Error Catching

To prevent any discrepancies in the financial records, the system will be able to detect numerical errors such as excessive digits/decimals, invalid characters, etc, to ensure no incorrect/unnecessary data will be stored into the database. Preventing such errors will lead to less inaccuracies in the financial records and will take up unnecessary storage within the database.

Permission Overrides

In some cases, the system may incorrectly detect an error or the accountant may need to enter an amount that is over the usual limit--in which would require approval from the firm's management team before proceeding. Either a managerial override will be prompted and/or an alert will be sent to management for them to assess the error more closely. Requiring managerial permission and/or alerts will minimize the risk of error as well as accountant fraud/unethical data manipulation.

Login Validation

The system will prompt error message if the user id/password entered during login is invalid or does not match any of the existing ones in the database. In addition to the system message, the number of attempts made to login are recorded and user will be locked out if too many unsuccessful attempts have been made. Login validation will prevent unauthorized users from accessing exclusive data and will restrict what each user can do to/with the data depending on their user type.

Checksums

Checksums are a popular control method used for validation, to assess the integrity of the system and/or its files, and to detect errors related to the transmission/storage of data. The checksum is a string of numbers and letters that is unique to each file/system. A checksum calculator is then used to generate another checksum for the file/system and the two are compared. If they are the same, the file/system is valid, and if not, it is invalid and may have been tampered with. Checksums ensure that data is intact, corrupt-free and has not been tampered with; therefore ensuring the security of the data.

Payment Validation

For a more secure and reliable payment, the system will be able to detect payment errors such as invalid card number/cardholder information. Ensuring that card payments are able to process successfully will prevent budget shortages and will also ensure that the firm is correctly being paid for their services.

Test Plan

ID	Subject	Passed	If test didn't pass, why?	Test Description
TC-1	Database Accessibility	Y	N/A	Test if database is accessible on various mobile devices
TC-2	User Log In	Y	N/A	Test if log-in proceeds to different dashboards depending on user type
TC-3	Accountant Option to Add/Remove/Update Record	Y	N/A	Test if Accountant dashboard allows accountants to add/remove/update client records
TC-4	Manager & Executive Option to Generate Reports	Y	N/A	Test if Manager & Executive dashboard allows management team and executive to generate client/company reports
TC-5	Auditor Option to Review Records	Y	N/A	Test if Auditor dashboard allows auditor to review and finalize records
TC-6	System Time Out	Y	N/A	Test if system will time out after five minutes of inactivity and/or if user tries too many times to log in

Project Report Summary

In conclusion, our goal was to create a more secure and efficient system for the growing accounting firm. The new system will offer more features to make the lives of the accounting team's easier, by providing mobile-ready capabilities as well as a more cost efficient and user friendly data system. Now, the accounting team is able to access all necessary data at their convenience, and will not have to worry about constantly having to remove less-used data in order to conserve memory space and hardware expenses. Also, both HR and financial records as well as the records from both larger and smaller accounting firms will now all be available on one system.

To implement this plan, the project team will use a team-based approach to divide and conquer the project, while keeping the accounting firm's needs and desires as the focal point of the project's duration. Because the project will be completed in a short amount of time, about three months, it is important that the team works efficiently and stays on schedule with the project plan. This will require lots of involvement from the management team and overall team work.

Works Cited

Beal, Vangie. "How Much Will a Server Cost?" *Webopedia*. N.p., 31 Mar. 2011. Web. <http://www.webopedia.com/DidYouKnow/Hardware_Software/how_much_will_a_server_cost.html>.

Chevalier, Max, Arlind Kopliku, Ronan Tournier, and Olivier Teste. *How Can We Implement a Multidimensional Data Warehouse Using NoSQL*. N.p.: Research Gate, 29 Apr. 2016. PDF<ttps://www.researchgate.net/profile/Olivier_Teste/publication/301621653_How_Can_We_Implement_a_Multidimensional_Data_Warehouse_Using_NoSQL/links/572318c008aef9c00b7e951b.pdf>.

Duggan, Tara. "The Advantages of Team-Based Structure." *Chron*. N.p., 26 Oct. 2016. Web. <http://smallbusiness.chron.com/advantages-teambased-structure-55624.html>.

Fisher, Tim. "What is a Checksum? Definition, Examples and More." Lifewire. N.p., 16 Oct. 2016. Web. <https://www.lifewire.com/what-does-checksum-mean-2625825>.

Hirt, Allan. *A Guide to Consolidating SQL Data Warehouse*. N.p.: SQL Server Magazine, May 2009. PDF. <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwiCtafm9NzTAhUTHGMKHdTCC9YQFggnMAA&url=https%3A%2F%2Fwww.platformmodernization.org%2Funisys%2FLists%2FSuccessStories%2FAttachments%2F1%2FGuide_to_Consolidating_SQL_Server_DW.pdf&usg=AFQjCNE1-sB4T2Qppg5_Np_iAouNK8VDZQ&sig2=o8genO0ynfelJJJfUaKybQ>.

Katsov, Ilya. "NoSQL Data Modeling Techniques." *Highly Scalable Blog*. Ilya Katsov, 16 Oct. 2012. Web. 19 May 2017.

Kimball, Ralph, and Margy Ross. *The Data Warehouse Toolkit*. N.p.: Wiley Computer Publishing, 2002. PDF.<http://www.dsc.ufcg.edu.br/~sampaio/Livros/alph%20Kimball.%20The%20Data%20Warehouse%20Toolkit..%20The%20Complete%20Guide%20to%20Dimensional%20Modelling%20(Wiley,2002)(ISBN%200471200247)(449s).pdf>.

Lotz, Mary. "Waterfall vs. Agile: Which Methodology is Right for Your Project?" *Segue Technologies*. N.p., 5 July 2013. Web. <http://www.seguetech.com/waterfall-vs-agile-methodology/>.

Mor, Yoav. "How Much Does It Cost to Run Relational Database Options on AWS." Cloudyn. N.p., 20 Aug. 2015. Web. <https://www.cloudyn.com/blog/how-much-does-it-cost-to-run-relational-database-options-rds-on-aws/>.

Oracle Big Data Cloud Service. N.p.: Oracle, 2015. PDF. <https://cloud.oracle.com/_downloads/Datasheet_BigData_1/BDCS_Datasheet.pdf>.

YouTube. (2015 August, 10). Comprehensive, Secure, Dedicated, Big Data Cloud Service [Video file]. Retrieved from https://youtu.be/84_Ty4GREds.

"What is PCI Compliance?" Online Tech. N.p., n.d. Web. <http://www.onlinetech.com/resources/references/what-is-pci-compliance>.