# Business Intelligence Tool

**Project Plan**

**3/26/2017**

CMSC 495-7980 Group 5

Jay Herford, Daniel Park, Thomas Barton, Jason Boone

## Introduction

### Purpose

The purpose of this document is to describe the project idea and a high level overview of the project plan. The document will outline the technologies used in the project and the timeline of development in multiple phases.

### Project Description

The software is a business intelligence program targeted towards data analysts and business users. It is intended to help users analyze data and track trends within their company. The software will connect to a database to retrieve and present required data. The software will allow users to view specific tables and drill down on the column level. It will include some automated statistical analysis that is described further in the document. The business intelligence tool can also display data in graphical images including graphs and charts.

### Scope

The business intelligence software will include the following features:

* Three sample tables
* View full table data
* View column data
* Sort column data (ascending/descending)
* Filter data based on criteria:
  + Greater than
  + Less than
  + Equals
* Present data graphically with:
  + Pie Chart
  + Line Chart
  + Scatter Chart
  + Bar Chart

### 1.4 Overview

The document contains the following artifacts:

* Technology Specifications
* Architecture
* Project Implementation

## Technology Specifications

### Database

The business intelligence software will connect to MySQL hosted in Amazon Web Services. By using a cloud service, the data lying in the background may be accessed from any machine. A MySQL instance of AWS RDS will be created that will allow external connection. The specifications for the the MySQL relational database instance are below:

* MySQL version 5.7.16 (latest supported version)
* Instance class db.t2.micro (free tier version)
* 5GB Storage

The database will utilize the free tier version of AWS RDS. It will take on most of the default settings. The security group will be modified so that the default group can allow any external incoming traffic.

### Database Connection

The database will use JDBC connectivity for the software to establish connection with the database. The latest MySQL JDBC driver is version 5.1.41.

### Graphical User Interface

The graphical user interface will be created using Java FX. It will provide a layout with different buttons to utilize the functionality of the business intelligence software. The user may select different tables, columns within the tables, filter criteria options, and types of charts. The charts will be designed using JavaFX.

Sample GUI layout:

Table List

Table1  
Table2  
Table3

Table Data

Options

Column Select

Chart

Column Data

### User Requirements

In order to utilize this software, the user must have the following hardware/software configuration:

Software:

* Java 6.1 or later
* Internet connectivity (to connect to AWS hosted database)

Hardware:

* OS (Windows, Mac, or Linux)
* Monitor
* Keyboard
* Mouse

### Code and Additional Material

GitHub will be used to store the code related to the project. Additional supporting material will also be stored in GitHub.

## Architecture

The following diagram illustrates a high level interaction of the technologies utilized in this project:

Data

AWS RDS MySQL

Business Intelligence software GUI

User

SQL

The user can directly access the software’s graphical interface. From the user interface, the user can select the table, column, selection criteria, and type of chart he wants to retrieve. Once the desired options are selected, the software will then issue SQL queries to the database. Once the SQL queries finish executing, the results will be passed back to the software. The retrieved data will then be processed and formatted for presentation within the software.

## Specific Requirements

### 4.1 External Interfaces

Information is conveyed only to users via graphical user interface. Users select the table that they wish to report on, the criteria of the search, and the view that they wish to use. The results are displayed back to the user based on the selected criteria.

#### 4.1.1 User Interface

User application front end GUI – Users may access the system by launching the jar file from any currently supported windows operating system. The GUI provides access to all and displays requested information to the user.

#### 4.1.2 Hardware Interfaces

No hardware interfaces are required at this time. Future updates could include the ability to print or save results to file.

#### 4.1.3 Software Interface

The software will interface with a configured AWS RDS MySQL database for data. The application will use standard processes built into the development language to connect to SQL using SQL authentication with a preconfigured SQL administrator password.

#### 4.1.4 Communication Interface

No communication interfaces are required at this time. Future updates could include the ability to send results to an email address.

### 4.2 Scenario

Business user or data analyst will open the jar file which will take the user to the main graphical user interface to view specific tables and data. The user can specify which table and column he wishes to view. Based on the user selection, the software will return details of the specified data with reports of the data. Based on the returned results, the user may view data trends and conduct further analysis to provide business insight. The returned specifications are detailed in section 4.3 Use Cases.

### 4.3 Use Cases

The primary functions are elaborated into the following use cases.

* View Data as Table
  + Retrieve table list
  + Select table
  + Select filter criteria or leave criteria fields blank
  + Select table view
  + Display full table to user
* View Column Data from a Table
  + Retrieve table list
  + Select table
  + Select filter criteria or leave criteria fields blank
  + Select column view
  + Select column
  + Display column data to user
* View Statistics
  + Retrieve table list
  + Select table
  + Select filter criteria or leave criteria fields blank
  + Select view statistics
  + View average value
  + View percentage of null values
* View Data as Graph
  + Retrieve table list
  + Select table
  + Select filter criteria or leave criteria blank
  + Select graph view
  + Select Graph type from Pie Chart, Line Chart, Scatter Chart, Bar Chart
  + Display graph to user

### 4.4 Performance Requirements

The initial build of the software is designed with the intent of supporting a single user. If the application is scaled upwards to handle multiple users, user control will be implemented and performance metrics will need to be reevaluated to adjust to the new user demands accordingly. The 5GB limit imposed by using the specified free tier of SQL database (instance class db.t2.micro) will limit the total amount of data stored. If the software were to be moved into a full production environment, a sizing estimate would need to be performed in order to properly determine the amount of space required to store the data.

### 4.5 Logical Database Requirements

Three main reportable tables will be designed for each of the three phases. The intent is to develop tables based on standard business uses. These will represent reporting data pulled from other systems at a theoretical company. A lookup table will be used to fetch the list of all available tables for reporting. The initial design of these tables will be as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Table | Column | Data Type | Key |
|  |  |  |  |
| lookup | table\_id | integer | PK |
|  | Table\_name | varchar(50) |  |
|  |  |  |  |
| payroll | payroll\_id | integer | PK |
|  | check\_number | integer |  |
|  | department | varchar(10) |  |
|  | employee\_id | integer |  |
|  | date | datetime |  |
|  | amount | float |  |
|  |  |  |  |
| sales | transaction\_id | integer | PK |
|  | customer\_id | integer |  |
|  | employee\_id | integer |  |
|  | date | datetime |  |
|  | amount | float |  |
|  |  |  |  |
| expenses | expense\_id | integer | PK |
|  | department | varchar(10) |  |
|  | employee\_id | integer |  |
|  | date | datetime |  |
|  | amount | float |  |

### 4.6 Design Constraints

The design of the system is limited by budget. The hardware will meet suggested standards for the operating system and for SQL server but will not exceed these requirements. No licensing of third party programs or components is available.

## Software System Attributes

The design of this system imitates a business application but at this time does not need to meet the standards of a mission critical business application or financial system.

### 5.1 Reliability

The system does not have a reliability requirement at this time but should it be adapted for production use, this would be determined based on the requirements of the business implementing the solution.

### 5.2 Availability

The system does not have a availability requirement at this time but should it be adapted for production use, reliability would be limited to the availability of the AWS database which would in turn be determined by the tier of service selected during implementation.

### 5.3 Security

No live data will be stored in the solution at this time. Should the solution be adapted for production use, the terms of AWS RDS security would apply based on the purchased tier of service.

### 5.4 Maintainability

All maintenance will be automatic and no user intervention will be required to maintain the system. Should the solution see production use, backup failure and integrity alerts will be sent to a customer specified email to warn of errors based on AWS RDS settings specified at implementation.

### 5.5 Portability

Testing will be required prior to migration of the system to a newer operating system or version of SQL to assure that the system remains functional.

## Roles and Responsibilities

The project roles and responsibilities will be divided evenly between team members based on their respective areas of expertise. Due to the nature of the project where many pieces of the project are interlinked, even though on a high level responsibilities are defined, collaboration is required during all phases and aspects of the project.

### Communication

* Create GitHub organization/repository: Jay Herford
* Contribute to GitHub repository: All

### Documentation

* Project Plan Lead: Daniel Park
* Project Design Lead: Jason Boone
* Test Plan Lead: Thomas Barton
* Editors and Contributors: All

### JavaFX GUI

* Implement Classes and Methods: Thomas Barton and Jay Herford
* Debug: All

### Database Schema

* Design Schema: Jason Boone and Daniel Park
* Debug: All

### Database Connection

* AWS RDS Creation: Daniel Park
* Test Connection: All

### 6.6 Data Retrieval

* Set up JDBC connection: Daniel Park
* Produce result set: Daniel Park, Jason Boone
* SQL development: Jason Boone, Daniel Park

## High Level Project Implementation Plan

The project will be implemented in 3 phases. Each phase will extend the capabilities of the previous phase.

### Phase 1 - POC

The first phase of the project is a proof of concept, which will focus on basic software functionality. The effort will focus on building the fundamental pieces of the software and getting each piece working togther. The requirements for the first phase of the project are as follows:

* Database
  + Create a MySQL instance of AWS RDS.
  + Create 1 table in the database and populate with data
  + Establish connection with the AWS hosted database in JDBC
* GUI
  + Create table selection interface
  + Create column selection interface
  + Create functioning data ascending/descending order buttons
  + Create 1 chart
* Code
  + Create SQL code that is dynamically created based on the user selection
  + Format retrieved data for presentation
* Test and Review

### Phase 2 – Expand Functionality

The second phase of the project will build on top of the first phase of the project to expand its capabilities.

* Database
  + - Create additional tables (as needed) in the database and populate with data
* GUI
  + - Improve on GUI design
    - Create filter function
    - Create 2 charts
* Code
  + - Create SQL code to support the newly added functionalities
    - Format retrieved data for presentation
    - Organize and clean up code
* Test and Review

### Phase 3 – Finalize

The final phase of the project will implement the remaining specifications of the software and master the software.

* Database
  + - Finalize tables in the database and validate data
* GUI
  + - Refine the GUI design
    - Create 1 chart
    - Create functions
* Code
  + - Create SQL code to support the newly added functionalities
    - Format retrieved data for presentation
    - Organize and clean up code
* Test and Review

## Detailed Project Implementation Schedule

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Due Date** | **Responsibility** |
| **Project Plan** | Mar 26 | Team Collaboration (Effort Lead: Daniel) |
|  |  |  |
| **Project Design** | Apr 9 | Team Collaboration (Effort Lead: Jason) |
|  |  |  |
| **Test Plan** | Apr 2 | Team Collaboration (Effort Lead: Thomas) |
|  |  |  |
| **Phase 1 Source** | Apr 16 |  |
| Database table 1 structure and test data | Apr 10 | Jason |
| Setup JDBC | Apr 10 | Daniel |
| Core UI | Apr 12 | Jay |
| Produce result set | Apr 12 | Daniel |
| Create SQL | Apr 12 | Jason |
| Display list of tables | Apr 12 | Thomas |
| Table display functionality with no parameters set | Apr 12 | Thomas, Jay |
| Group review | Apr 13 | Team |
| Final Revisions | Apr 15 | Team |
|  |  |  |
| **Phase 2 Source** | Apr 23 |  |
| Database table 2 structure and test data | Apr 17 | Daniel |
| Create SQL | Apr 19 | Jason |
| Produce result set | Apr 19 | Daniel |
| Table display functionality with parameters set | Apr 19 | Thomas, Jay |
| Column display functionality | Apr 19 | Jay |
| Group review | Apr 20 | Team |
| Final Revisions | Apr 22 | Team |
|  |  |  |
| **Phase 3 source** | Apr 30 |  |
| Database table 3 structure and test data | Apr 24 | Jason |
| Create SQL | Apr 24 | Jason |
| Produce result set | Apr 24 | Daniel |
| Graph display functionality | Apr 26 | Thomas, Jay |
| Group review | Apr 27 | Team |
| Final Revisions | Apr 29 | Team |
|  |  |  |
| **Final Submission** | May 7 |  |
| Full testing | May 1 | Team |
| Final revisions | May 6 | Team |