**Core Java**

**2.1.1 Customer and Transaction Details Module**

**CREDIT CARD SYSTEM**

What would you like to do?

1. View Transactions by Zipcode

2. View Totals By Transaction Type

3. View Totals By State

4. View Customer Details By SSN

5. Edit a Customer's Info

6. View a Customer's Bill for a given Month

7. View a Customer's Transaction Between Two Dates

Enter your selection:

---------------------------------------------------------------------------------

Below are project details/Descriptions (also located in the **Eclipse\_Project.zip** file provided), and path that enables the user to be presented with the above “Credit Card System” processing choices.

**Runner (package)**

The **MainModule.java** is the controller for the entire application. This is where the decision to process request 1, 2, 3, 4, 5, 6 or 7 is handled. I’ve provided documentation within the code to help facilitate understanding of the flow.

**FILEPATH: \casestudy\src\com\cdw\runner**

* **MainModule.java**

**DAO (package)**

The DAO package contains java classes specific to their respective functions…

**FILEPATH: \casestudy\src\com\cdw\dao**

* **AbstractDAO.java – This class is setup to handle DB connection, accessing and processing**
* **CreditCardDAO.java – This class processes all DB queries related to Credit Card requests…**
* **CustomerDAO.java – This class processes all DB queries related to Customer requests…**
* **TransactionDAO.java – This class processes all DB queries related to Transaction requests..**

**Model (package)**

The Model package contains a class for each processing request.

**FILEPATH: \casestudy\src\com\cdw\model**

* CreditCardMonthlyBill.java – Contain the structure for CC Monthly Bill Data to be processed
* Customer.java – Contains the structure for all Customer info to be processed
* TransactionForAGivenDate.java – Contains the class structure to process this transaction
* TransactionForAGivenState.java - Contains the class structure to process this transaction
* TransactionForAGivenType.java - Contains the class structure to process this transaction
* TransactionsForAGivenDateRange.java - Contains the class structure required to process…

**Resources (package)**

The Resources package contains the files which can be accessed via the application for processing  
 while allowing for ease of modification in a central location.

**FILEPATH:** \**casestudy\src\com\cdw\resources**

* db.properties – is a file that contains the database connection information
* Queries.java – is a file that contains all of the DB queries the app needs to process requests

**CSV (package)**

The CSV package contains the class which handles the processing exports of transaction results to  
 CSV.

**FILEPATH:** \**casestudy\src\com\cdw\csvfile**

* GenerateCSVfile.java– is a class that processes all .csv exports..

**For example:**

Enter your selection: 1

Please enter Customer Zipcode: 39120

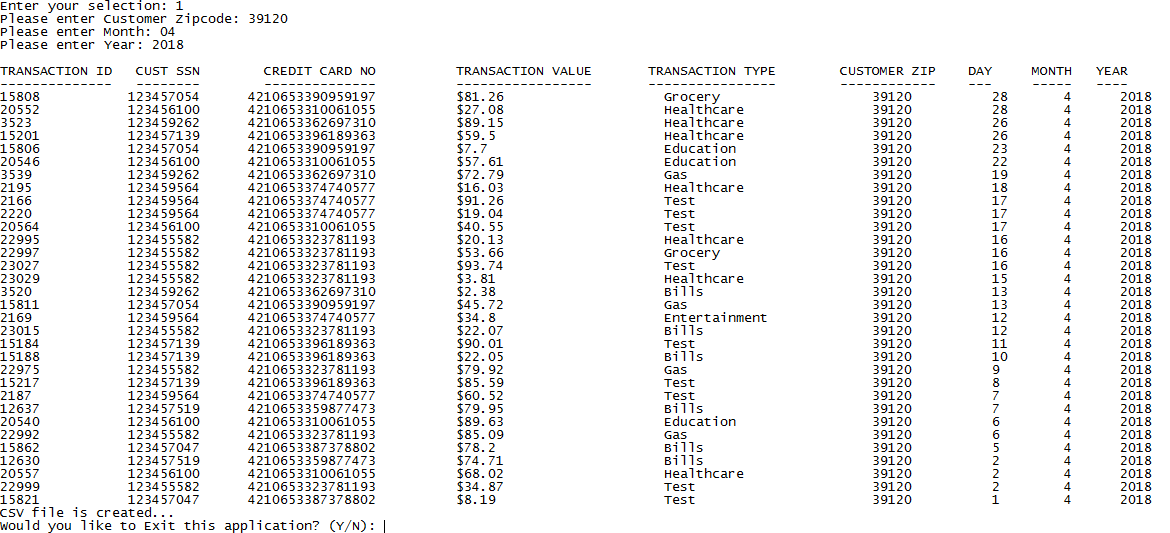
Please enter Month: 04

Please enter Year: 2018

Produces the following output file which will be provided with this document:

* TransactionsForAGivenDate.csv

Below is a screen shot of the output when ran via eclipse



**RDBMS/mySQL Description**:   
 **2.1.1 Customer and Transaction Details Module**

I’ve provided an EER Diagram and the exported “casestudy” database with the following  
 names in an **RDBMS.zip** file:

* EER\_Diagram\_casestudy.jpg
* casestudy\_cdw\_sapp\_branch.sql
* casestudy\_cdw\_sapp\_creditcard.sql
* casestudy\_cdw\_sapp\_customer.sql

The Java application SQL related info:

* Queries.java – is a file that contains all of the DB queries the app needs to process requests…  
  Please refer to this file I’ve provided along with this documentation to review all of the MySQL queries, and updates written for this application.

**Hadoop/hdfs/dataware housing**

**2.2.1 Data Extraction and Transportation with Sqoop**

Below are scoop jobs contained in individual .txt files that are zipped into one **Hadoop.zip** file:

* scoop\_job\_importBranchTable.txt
* scoop\_job\_import\_CreditCardTable.txt
* scoop\_job\_importCustomerTable.txt
* scoop\_job\_importTimeTable.txt
* scoop\_job\_updateBranchTable.txt
* scoop\_job\_appendCreditCardTable.txt
* scoop\_job\_updateCustomerTable.txt
* scoop\_job\_appendTimeTable.txt

**Hive and Partition**

**2.2.2 Data Loading Module**

Below are Hive scripts contained in individual hive files that are zipped into one **Hive.zip** file:

* Drop\_Branch\_Table.hive --- Staging Table
* Drop\_Branch\_Table\_Part.hive --- Partition Table
* Drop\_Credit\_Card\_Table.hive
* Drop\_Customer\_Table.hive
* Drop\_Time\_Table.hive
* CreateBranchTable.hive --- Staging Table
* CreateBranchTablePart.hive --- Dynamic Partition Table
* CreateCreditCardTable.hive
* CreateCreditCardTable.hive
* CreateTimeTable.hive
* Load\_data\_into\_branch.hive --- Load data from staging table to dynamic partition table

**Oozie (Sqoop and Hive)**

**2.2.3 Automating the Process with Oozie**

I’ve provided a Blueprint/Diagram in both MS Word & .pdfto better visualize my Oozie  
 workflow process:

* oozie workflow Diagram initial run.docx
* oozie workflow Diagram initial run.pdf

Below is a list of Oozie files required to execute the workflow process. I have combined these files into  
 one **oozie\_initial\_run.zip** file.  
  
 Note: Please refer to the above listed Blueprint/Diagram to make sense of the workflow…

* job.properties
* bundle.xml
* coordinator\_Branch.xml
* coordinator\_CreditCard.xml
* coordinator\_Customer.xml
* coordinator\_Time.xml
* workflow\_Branch.xml
* workflow\_CreditCard.xml
* workflow\_Customer.xml
* workflow\_Time.xml

**Oozie (Sqoop and Hive optimized)**

**2.2.4 Process Optimization Module**

I’ve provided a Blueprint/Diagram in both MS Word & .pdfto better visualize my Oozie  
 workflow process:

* oozie workflow Diagram optimized.docx
* oozie workflow Diagram optimized.pdf

Below is a list of Oozie files required to execute the workflow process. I have combined these files into  
 one **oozie\_optimized\_run.zip** file.  
  
 Note: Please refer to the above listed Blueprint/Diagram to make sense of the workflow…

* job\_optimized.properties
* bundle\_optimized.xml
* coordinator\_Update\_Branch.xml
* coordinator\_Append\_CreditCard.xml
* coordinator\_Update\_Customer.xml
* coordinator\_Append\_Time.xml
* workflow\_Update\_Branch.xml
* workflow\_Append\_CreditCard.xml
* workflow\_Update\_Customer.xml
* workflow\_Append\_Time.xml

**Visualization**

**2.2.5 Data Visualization**

I’ve provided a “**data\_visualization.zip**” file which contains the following:

* data\_visualization\_queries.docx
* Data\_Visual\_Hive\_\_trans\_type\_by\_Quarter.jpg
* Data\_Visual\_Hive\_top\_20\_zipcodes.jpg

**Pig**

* I’ve provided a **pig\_script.txt** file which contains my pig scripts and the outputs…