Inventory Management System

Bottom End Report

February 01, 2015

Prepared for Chandler Janzen

by

Craig Irvine

Keenan Johnstone

Justin Fraser

Table of Contents

[Change Log 2](#_Toc442075509)

[1. Purpose 2](#_Toc442075510)

[2. References 2](#_Toc442075511)

[3. Overall System Interconnects 3](#_Toc442075512)

[4. Front End 4](#_Toc442075513)

[4.1. ParseXMLResponse Script 4](#_Toc442075514)

[4.2. LoadBrowser Script 5](#_Toc442075515)

[4.3. LoadLog Script 6](#_Toc442075516)

[4.4. LoadItemData Script 7](#_Toc442075517)

[5. Backend 8](#_Toc442075518)

[5.1. IMSBase Class 8](#_Toc442075519)

[5.1.1. IMSBase->VerifyData 8](#_Toc442075520)

[5.1.2. IMSBase->GenerateXMLResponse 9](#_Toc442075521)

[5.2. IMSLog Class 10](#_Toc442075522)

[5.2.1. IMSLog->\_\_construct 10](#_Toc442075523)

[5.2.2. IMSLog->add\_log 11](#_Toc442075524)

[5.2.3. IMSLog->read\_log 12](#_Toc442075525)

[5.3. IMSSql Class 13](#_Toc442075526)

[5.3.1. IMSSql->\_\_construct 13](#_Toc442075527)

[5.3.2. IMSSql->connect 14](#_Toc442075528)

[5.3.3. IMSSql->command 15](#_Toc442075529)

[5.3.4. IMSSql->prepare 16](#_Toc442075530)

[5.3.5. IMSSql->exists 17](#_Toc442075531)

[5.4. Base Scripts 18](#_Toc442075532)

[5.4.1. CreateNewItem 18](#_Toc442075533)

[5.4.2. RetrieveBrowserData 19](#_Toc442075534)

[5.4.3. RetrieveItemData 20](#_Toc442075535)

[5.4.4. RetrieveLog 21](#_Toc442075536)

[5.4.5. AddNewClassData 22](#_Toc442075537)

[5.4.6. ModifyItem 23](#_Toc442075538)

[5.4.7. QueryAutocomplete 24](#_Toc442075539)

[5.4.8. GeneratePurchaseReport 25](#_Toc442075540)

[5.4.9. ReadOptions 26](#_Toc442075541)

[5.4.10. ModifyOption 27](#_Toc442075542)

[6. Database 28](#_Toc442075543)

# Change Log

|  |  |  |
| --- | --- | --- |
| Date | Version | Change |
| 01 February 2015 | 1 | Initial Submission |
|  |  |  |
|  |  |  |

# 

# Purpose

This report show in the details of all the modules required for Inventory Management System.

# 

# 

# References

<Statement of Work> EE495-201509-SOW-01A

Inventory Management System Specifications Version 3 (16 November 2015)

Inventory Management System Design Specifications Version 1 (19 November 2015)

# Overall System Interconnects

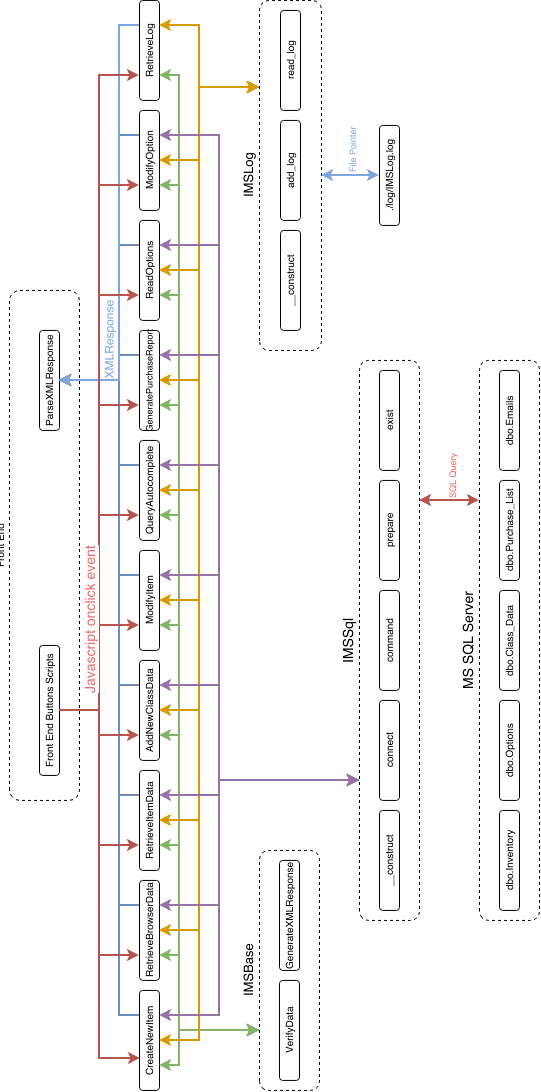


Figure 3-1 - System Interconnects

# Front End

## ParseXMLResponse Script

**Description:** Takes any XML response from the database and parses it into the appropriate format (such as the browser data into a table format).

**Inputs:** An XML Response

**Return Value:** The appropriate table or formatted response from the server

**Designer:** Keenan Johnstone

**Development Time:** 3 Hours



Figure 4-1 - ParseXMLResponse Flowchart

## LoadBrowser Script

**Description:** Calls the RetrieveBrowserData.php Script to return an XML response and then passes that to the ParseXMLResponse Script

**Inputs:** None

**Return Value:** id=”browser”

**Designer:** Keenan Johnstone

**Development Time:** 3 Hours

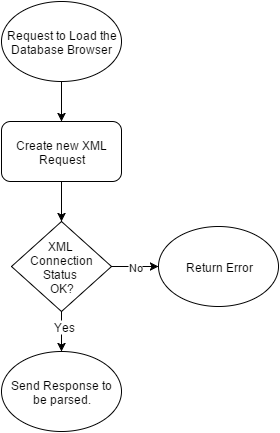


Figure 4-2 - LoadBrowser Flowchart

## 

## LoadLog Script

**Description:** Calls the RetrieveBrowserData.php Script to return an XML response and then passes that to the ParseXMLResponse Script

**Inputs:** None

**Return Value:** id=”log”

**Designer:** Keenan Johnstone

**Development Time:** 1 Hours

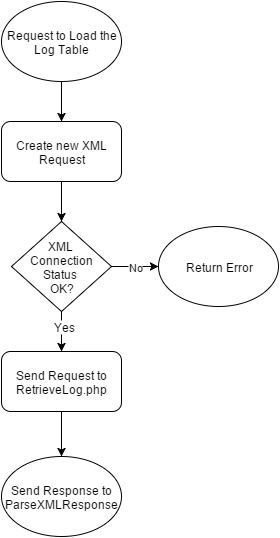


Figure 4-3 - LoadLog Flowchart

## 

## LoadItemData Script

**Description:** Calls the RetrieveItemData.php Script to return an XML response and then passes that to the ParseXMLResponse Script

**Inputs:** Item to be loaded

**Return Value:** id=”Itemdata”

**Designer:** Keenan Johnstone

**Development Time:** 1 Hour

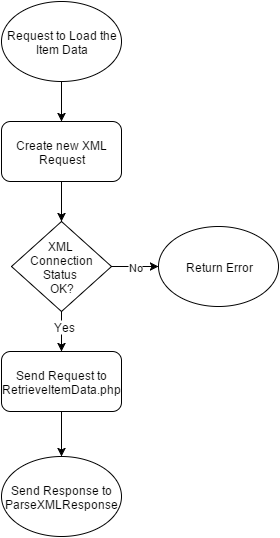


Figure 4-4 -LoadItemData Flowchart

# 

# Backend

## IMSBase Class

### IMSBase->VerifyData

**Description**: This function performs two checks on a data string passed to it to verify the proper data was passed from the front end.

**Inputs**: data - A string variable containing the data to check.

RegEx - A string variable containing a RegEx pattern to check the data against.

**Return** Value: Function will throw an Exception is either of the test fail otherwise it will return to calling function.

**Designer**: Craig Irvine

**Development Time**: 1 Hour

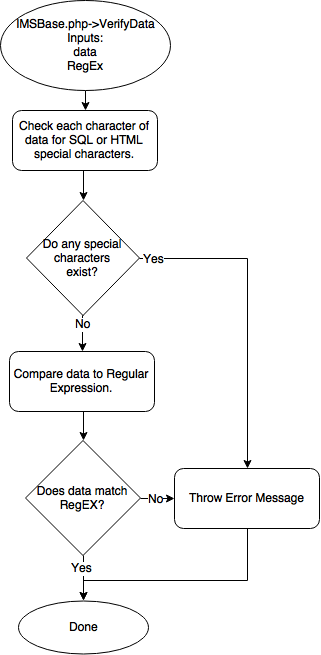


Figure 5-1 - IMSBase->VerifyData Flow Chart

### IMSBase->GenerateXMLResponse

**Description**: Generates a XML response document used for passing data from the backend scripts to the frontend GUI.

**Inputs**: Session ID - The ID for the calling front end session.

Status\_array - An array that contains the status of the response within two members.

query\_suggest\_array - (Optional) An array that contains a list of suggestions for use with

the autocomplete feature.

q\_access\_array - (Optional) An array that contains the information of a single item for use

in the quick access feature.

browser\_array - (Optional) An array that contains browser data generated from the

RetrieveBrowserData.php script.

log\_array - (Optional) An array that contains the log data generated from the RetrieveLog.php script.

**Return Value**: A XML document is returned to sent to the frontend.

**Designer**: Craig Irvine

**Development Time**: 4 Hours

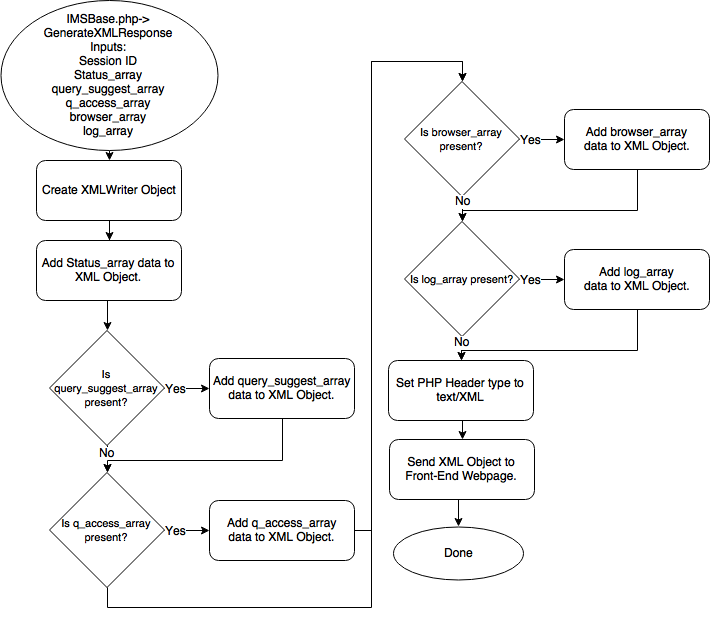


Figure 5-2 - IMSBase->GenerateXMLResponse Flowchart

## IMSLog Class

### IMSLog->\_\_construct

**Description**: Constructor for the IMSLog class. Sets the local file pointer to a passed or default location.

**Inputs**: Log File Location - The location where the log file is to be saved and read from.

**Return Value**: No return value.

**Designer**: Craig Irvine

**Development Time**: 1 Hour

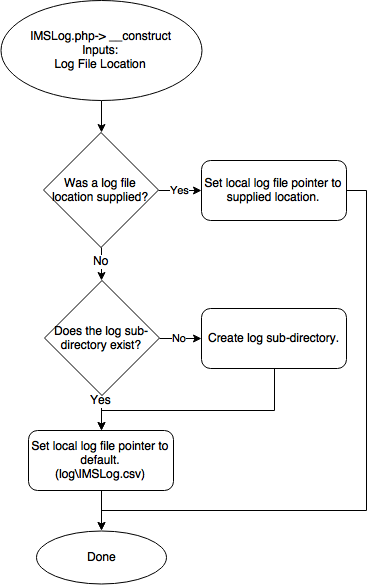


Figure 5-3 - IMSLog->\_\_construct Flowchart

### IMSLog->add\_log

**Description**: Adds a log entry to the log file.

**Inputs**: Session ID - The ID for the calling frontend session.

Log Level - The level of the log message, can be Information, Error, Warning, or Debug.

Message - The message to be logged.

**Returned Value**: None

**Designer**: Craig Irvine

**Development Time**: 1 Hour

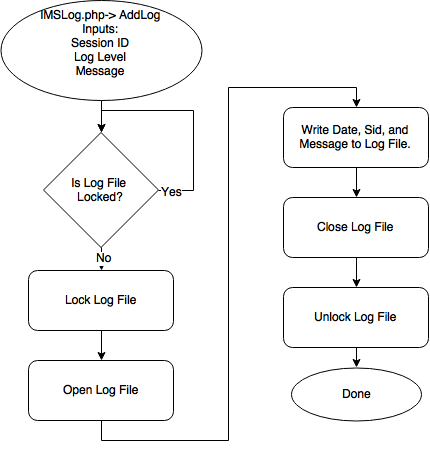


Figure 5-4 - IMSLog->add\_log Flowchart

### IMSLog->read\_log

**Description**: Reads the log file and retrieve entries related to the passed log level.

**Inputs**: Level Filter - A filter variable that set the log level to retrieve, can be Information, Error,

Warning, or Debug.

**Return Value**: Passes the log entries to the IMSBase->GenerateXMLResponse function.

**Designer**: Craig Irvine

**Development Time**: 1 Hour

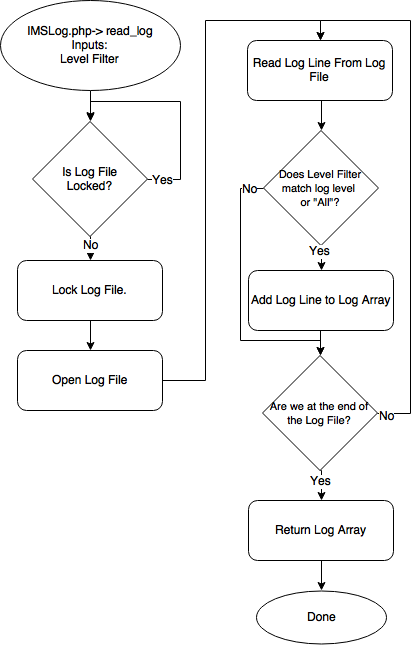


Figure 5-5 - IMSLog->read\_log Flowchart

## IMSSql Class

### IMSSql->\_\_construct

**Description**: A constructor for the IMSSql class. It sets local SQL server access variables then calls the connect function.

**Inputs**: server - The SQL server location

user - The username for logging into the SQL server.

password - The password for logging into the SQL server.

**Return Value**: The result of IMSSql->connect.

**Designer**: Craig Irvine

**Development Time**: 1 Hour

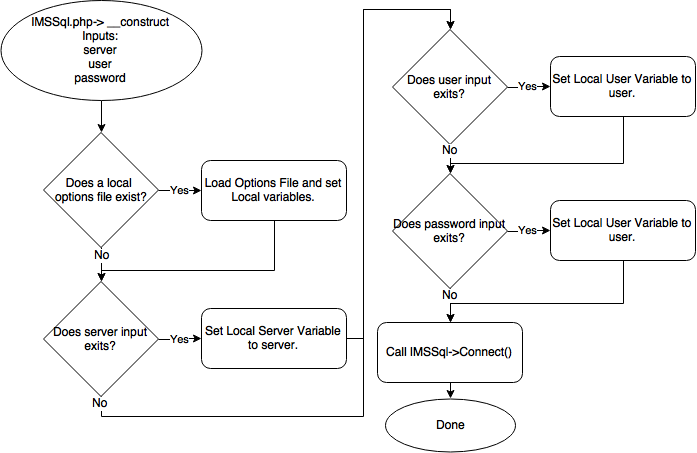


Figure 5-6 - IMSSql->\_\_construct Flowchart

### 

### 

### IMSSql->connect

**Description**: Create a connection to the SQL server using a PDO object.

**Inputs**: None

**Return Value**: PDOException if PDO creation was unsuccessful, otherwise nothing.

**Designer**: Craig Irvine

**Development Time:** 1 Hour

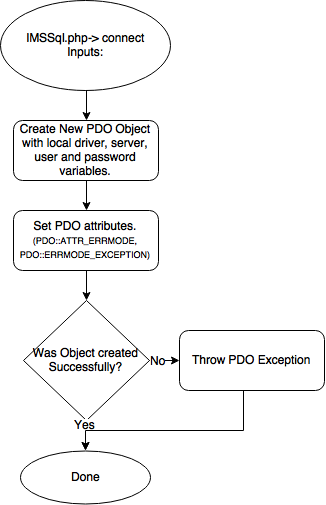


Figure 5-7 - IMSSql->connect Flowchart

### IMSSql->command

**Description**: Sends a SQL command to the SQL server to be executed.

**Inputs**: sql\_command: A SQL command string.

**Return Value**: A PDOException error is thrown if an error occurs, otherwise nothing.

**Designer**: Craig Irvine

**Development Time**: 0.5 Hour

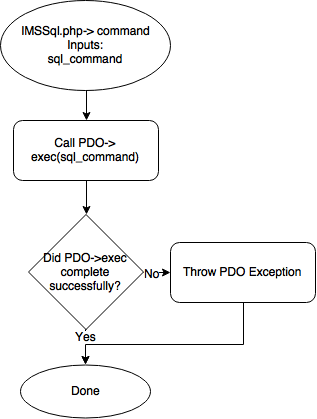


Figure 5-8 - IMSSql->command Flowchart

### IMSSql->prepare

**Description**: Creates a PDO::prepare statement and returns it to the calling script.

**Input**: SQLStatment - A string containing a SQL statement used to create the PDO::prepare object.

**Return Value**: A PDOExecption is thrown on error otherwise a PDO::prepare object is returned.

**Designer**: Craig Irvine

**Development Time**: 0.5 Hours

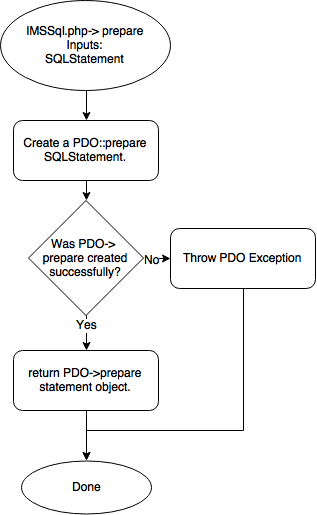


Figure 5-9 - IMSSql->prepare

### IMSSql->exists

**Description**: Checks to see if a part number already exists within the SQL database.

**Input**: partNumber - The item to look for in the database table.

table - The SQL table to look in.

**Return Value**: Is the part number exists within the table return true, else return false.

**Designer**: Craig Irvine

**Development Time**: 1 Hour

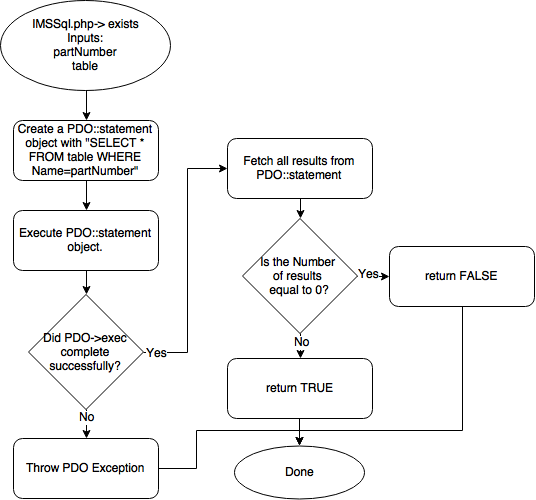


Figure 5-10 - IMSSql->exists Flowchart

## Base Scripts

### CreateNewItem

**Description**: Create a new item in the SQL database in the dbo.Inventory table.

**Inputs**: SID - The ID for the calling frontend session.

PartNumber - The part number of the item being created.

**Designer**: Craig Irvine

**Development Time**: 2 Hours

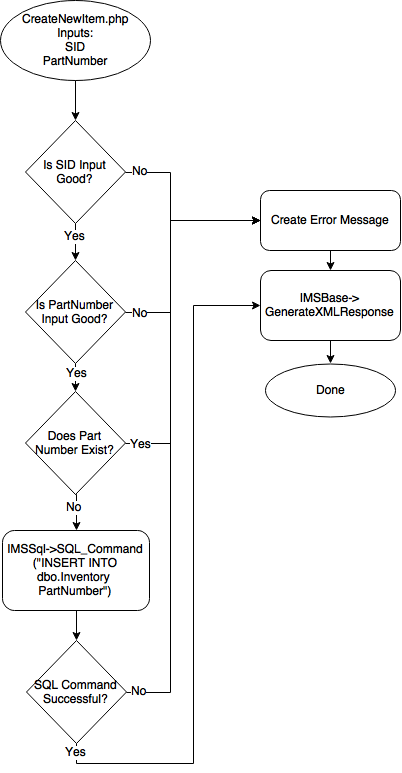


Figure 5-11 - CreateNewItem.php Flowchart

### RetrieveBrowserData

**Description**: Retrieves data for the browser second of the GUI with the ability to sort and filter the data.

**Inputs**: SID - The ID for the calling frontend session.

SortColumn - A string containing the name of column to sort.

SortDirection - A string containing ASC or DESC used to specify the sort direction.

Filter - A string containing a value to filter the data by.

**Return Value**: Returns an XML document with the results of the query.

**Designer:** Craig Irvine

**Development Time**: 2 Hour

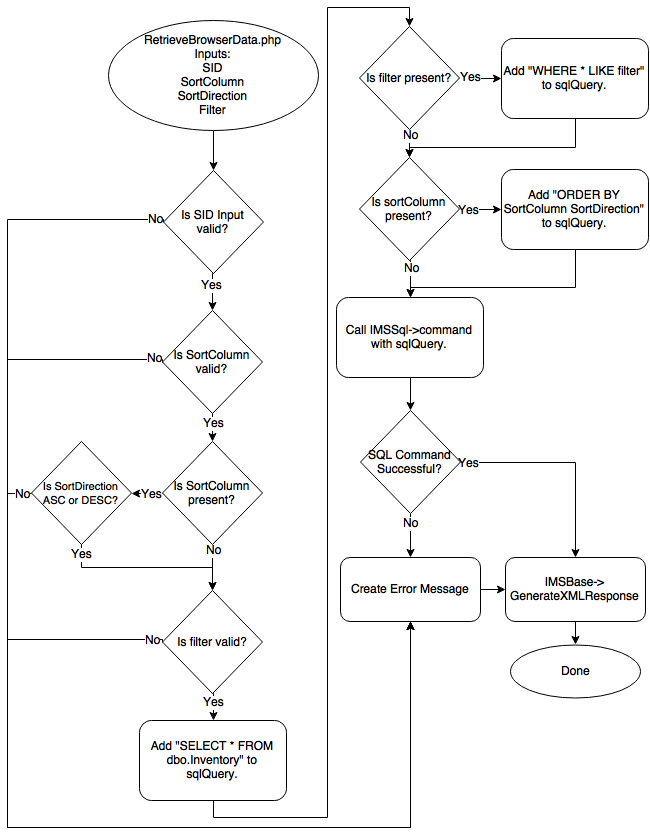


Figure 5-12 - RetrieveBrowserData.php Flowchart

### RetrieveItemData

**Description**: Retrieves data for a single item.

**Input**: SID - The ID for the calling frontend session.

PartNumber - A string containing the part number to retrieve data for.

**Return Value**: Returns an XML document with the results of the query.

**Designer:** Craig Irvine

**Development Time**: 1 Hours

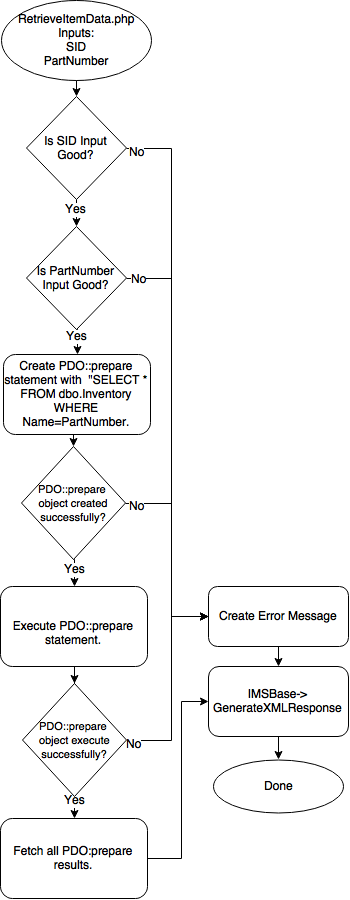


Figure 5-13 - RetrieveItemData.php Flowchart

### RetrieveLog

**Description**: Retrieve log entries for a specified log level.

**Inputs**: SID - The ID for the calling frontend session.

LogLevel - A string containing the log level requested, can be Information, Error, Warning,

or Debug.

**Return Value**: Returns an XML document with the results of the query.

**Designer:** Craig Irvine

**Development Time**: 1 Hour

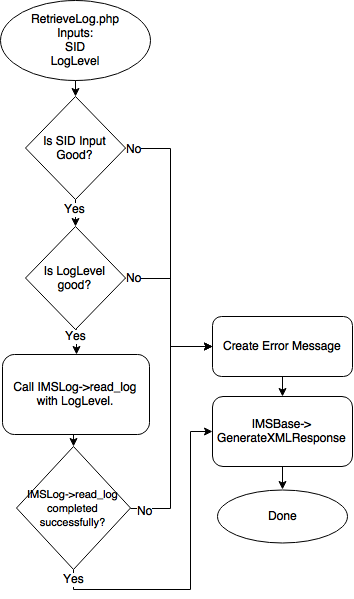


Figure 5-14 - RetrieveLog.php Flowchart

### AddNewClassData

**Description**: Add a new Record into the ClassData table.

**Inputs**:SID - The ID for the calling frontend session.

Class - A string containing the name of the class.

PartNumber - A string containing the part number needed.

Quantity - A positive interger contain the number of required parts.

Date - A string containing the date the parts are required by.

**Return Value**: Returns an XML document with status of the operation.

**Designer:** Craig Irvine

**Development Time**: 1 Hour

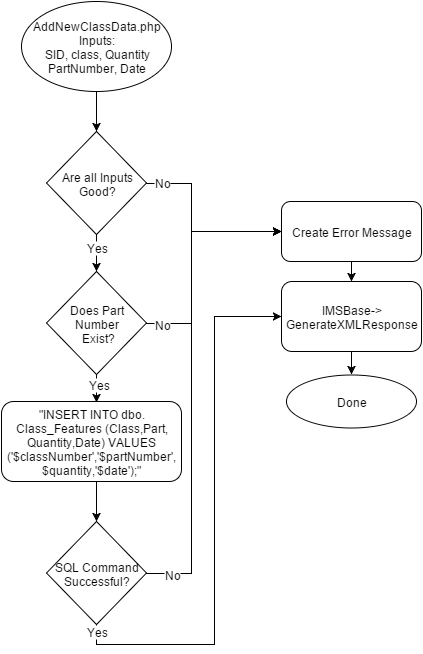


Figure 4-16 - AddNewClassData.php Flowchart

### 

### ModifyItem

**Description**: Modifies a single column value for a specified part number record.

**Inputs**:SID - The ID for the calling frontend session.

PartNumber - A string containing the part number needed.

field - A string containing the column to modify.

value - A string containing the value to modify to.

**Return Value**: Returns an XML document with status of the operation.

**Designer:** Craig Irvine

**Development Time**: 1 Hour

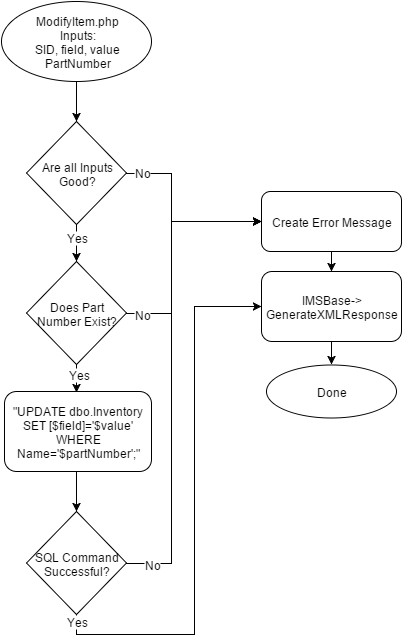


Figure 5-16 - ModifyItem.php Flowchart

### QueryAutocomplete

**Description**: Generates a list of item records that partially match a passed record.

**Inputs**:SID - The ID for the calling frontend session.

filter - A string containing a partial search value.

**Return Value**: Returns an XML document with a list of record that match the partial search value.

**Designer:** Craig Irvine

**Development Time**: 1 Hour

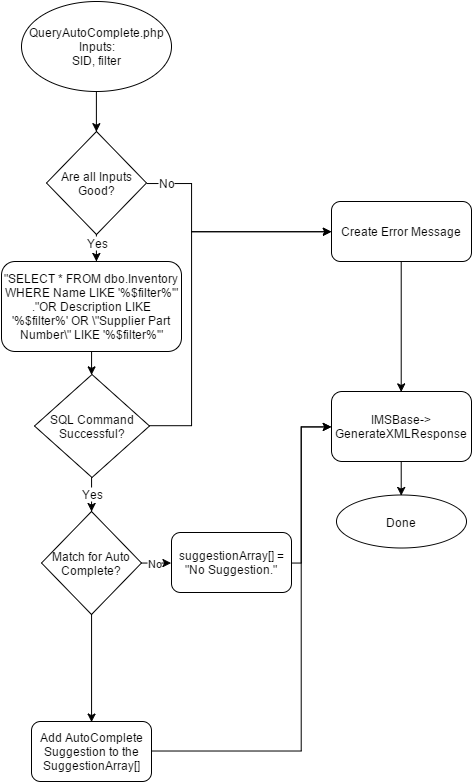


Figure 5-17 - QueryAutoComplete.php Flowchart

### GeneratePurchaseReport

**Description:** Populate Purchase List table with inventory items whose quantity is below threshold.

**Inputs:** SID -- the ID for the calling frontend session.

**Returns:** Returns an XML document with results of the query.

**Designer:** Justin Fraser

**Development Time:** 2 Hours

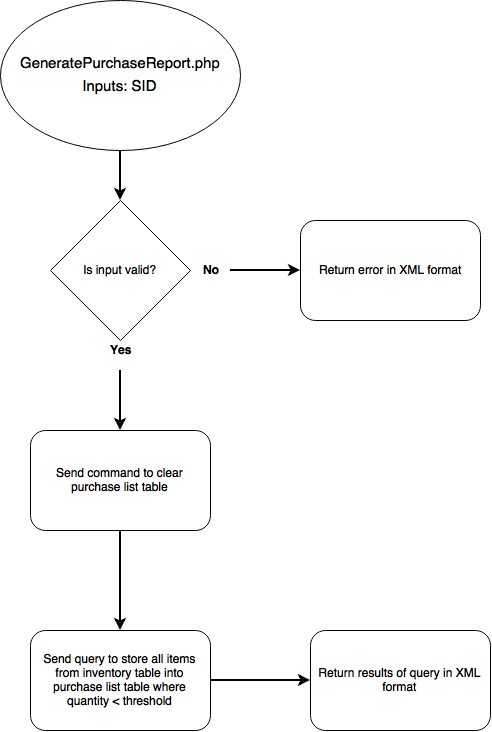


Figure 5-18 - GeneratePurchaseReport Flowchart

### ReadOptions

**Description:** Read IMS options.

**Inputs:** SID -- the ID for the calling frontend session.

**Returns:** Returns an XML document with results of the query.

**Designer:** Justin Fraser

**Development Time:** 2 Hours

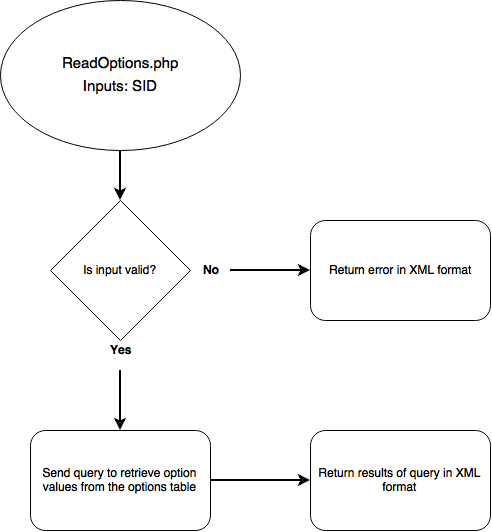


Figure 5-19 ReadOptions Flowchart

### ModifyOption

**Description:** Modify IMS options.

**Inputs:** SID -- the ID for the calling frontend session.

Option – the option to modify.

Data – value of the new option.

**Returns:** Returns an XML document with results of the query.

**Designer:** Justin Fraser

**Development Time:** 2 Hours

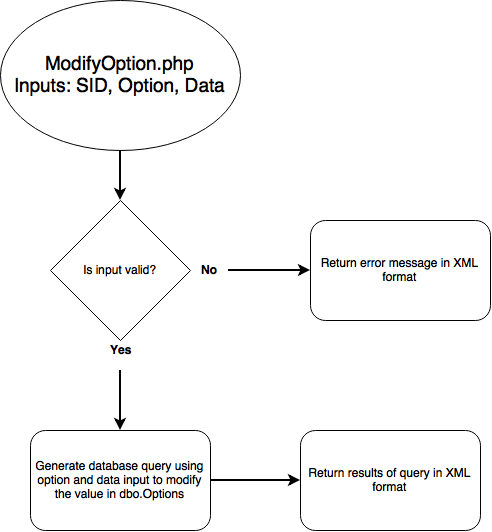


Figure 5-20 - ModifyOption Flowchart

# Database

**Description:** Back end interacts with the database which holds five tables of data. Currently, there are no relationships or dependencies between the tables in the database.

**Inputs:** queries or commands issued by the back end in the form of PHP scripts.

**Outputs:** rows of data on a successful query or an error message on an unsuccessful query to be handled by the back end.

**Designer:** Justin Fraser

**Development Time:** 20 Hours

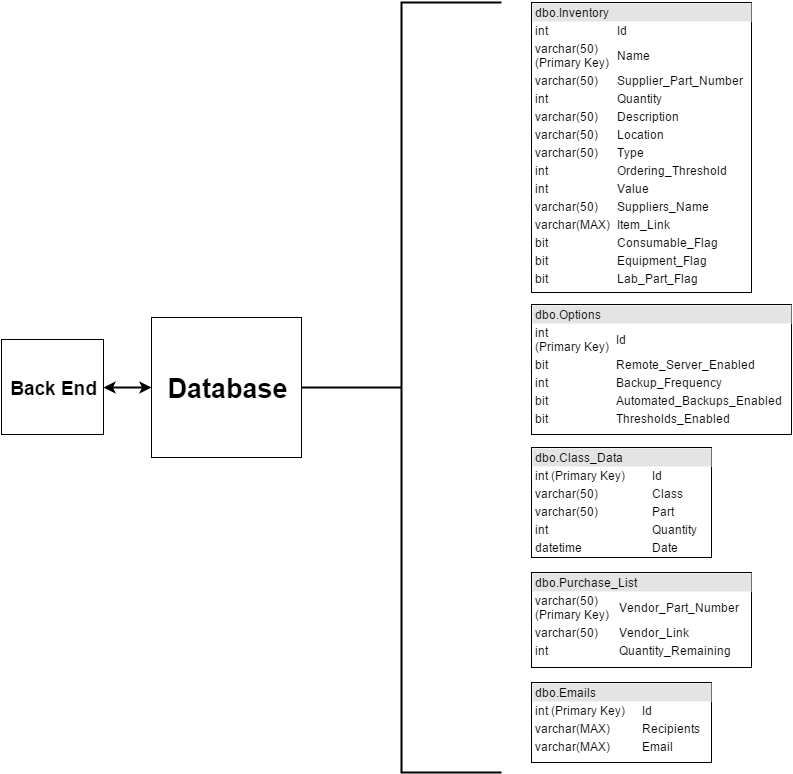


Figure 6-1 - Database layout