Team A

Design Specification

CIS-470 – Senior Project



Table of Contents

[Scope 5](#_Toc449046056)

[System Description 5](#_Toc449046057)

[Major Software Functions 5](#_Toc449046058)

[Database Description 6](#_Toc449046059)

[Design Constraints and Limitations 6](#_Toc449046060)

[**Design Description** 7](#_Toc449046061)

[**Data Flow** 8](#_Toc449046062)

[Program Architecture 10](#_Toc449046063)

[Figure 2.1: Class Hierarchy – Elements & Relations 10](#_Toc449046064)

[Figure 2.2: Class Hierarchy – Logical & Presentation Layer(s) 11](#_Toc449046065)

[Component Interfaces 12](#_Toc449046066)

[Figure 2.3: Login Component 12](#_Toc449046067)

[Figure 2.4: Work Order Creation 12](#_Toc449046068)

[Figure 2.5: Input Customer Data 13](#_Toc449046069)

[Figure 2.6: Input Billing Data 13](#_Toc449046070)

[Figure 2.7: Input Shipping Data 14](#_Toc449046071)

[Figure 2.8: Inventory Validation 14](#_Toc449046072)

[Figure 2.9: Order Validation 15](#_Toc449046073)

[Figure 2.10: Submit Work Order 15](#_Toc449046074)

[Figure 2.11: Submit Quality Assurance Checklist 16](#_Toc449046075)

[Figure 2.12: Complete Work Order 16](#_Toc449046076)

[Detailed Design 17](#_Toc449046077)

[1.1 Login 17](#_Toc449046078)

[Processing Description 17](#_Toc449046079)

[Interface Description 17](#_Toc449046080)

[Pseudo-code 17](#_Toc449046081)

[Modules Uses 18](#_Toc449046082)

[1.2 Create Work Order 18](#_Toc449046083)

[Processing Description 18](#_Toc449046084)

[Interface Description 18](#_Toc449046085)

[Pseudo-code 18](#_Toc449046086)

[Modules Uses 19](#_Toc449046087)

[1.3 Create Customer 20](#_Toc449046088)

[Processing Description 20](#_Toc449046089)

[Interface Description 20](#_Toc449046090)

[Pseudo-code 20](#_Toc449046091)

[Modules Uses 21](#_Toc449046092)

[1.4 Billing Data Input 21](#_Toc449046093)

[Processing Description 21](#_Toc449046094)

[Interface Description 21](#_Toc449046095)

[Pseudo-code 21](#_Toc449046096)

[Modules Uses 23](#_Toc449046097)

[1.5 Shipping Data Input 23](#_Toc449046098)

[Processing Description 23](#_Toc449046099)

[Interface Description 23](#_Toc449046100)

[Pseudo-code 23](#_Toc449046101)

[Modules Uses 25](#_Toc449046102)

[1.6 Inventory Validation 25](#_Toc449046103)

[Processing Description 25](#_Toc449046104)

[Interface Description 25](#_Toc449046105)

[Pseudo-code 25](#_Toc449046106)

[Modules Uses 26](#_Toc449046107)

[1.7 Order Validation 26](#_Toc449046108)

[Processing Description 26](#_Toc449046109)

[Interface Description 27](#_Toc449046110)

[Pseudo-code 27](#_Toc449046111)

[Modules Uses 27](#_Toc449046112)

[1.8 Perform Work 27](#_Toc449046113)

[Processing Description 27](#_Toc449046114)

[Interface Description 28](#_Toc449046115)

[Pseudo-code 28](#_Toc449046116)

[Modules Uses 28](#_Toc449046117)

[1.9 Quality Assurance Check 28](#_Toc449046118)

[Processing Description 28](#_Toc449046119)

[Interface Description 29](#_Toc449046120)

[Pseudo-code 29](#_Toc449046121)

[Modules Uses 29](#_Toc449046122)

[1.10 Order Completion 30](#_Toc449046123)

[Processing Description 30](#_Toc449046124)

[Interface Description 30](#_Toc449046125)

[Pseudo-code 30](#_Toc449046126)

[Modules Uses 31](#_Toc449046127)

[2.0 Search Existing Customer 31](#_Toc449046128)

[Processing Description 31](#_Toc449046129)

[Interface Description 31](#_Toc449046130)

[Pseudo-code 31](#_Toc449046131)

[Modules Uses 32](#_Toc449046132)

[2.1 New Customer Creation 32](#_Toc449046133)

[Processing Description 32](#_Toc449046134)

[Interface Description 32](#_Toc449046135)

[Pseudo-code 32](#_Toc449046136)

[Modules Uses 34](#_Toc449046137)

[2.2 Modify Existing Customer 35](#_Toc449046138)

[Processing Description 35](#_Toc449046139)

[Interface Description 35](#_Toc449046140)

[Pseudo-code 35](#_Toc449046141)

[Modules Uses 36](#_Toc449046142)

[3.0 Search Inventory 36](#_Toc449046143)

[Processing Description 36](#_Toc449046144)

[Interface Description 36](#_Toc449046145)

[Pseudo-code 36](#_Toc449046146)

[Modules Uses 37](#_Toc449046147)

[3.1 Create New Inventory item 37](#_Toc449046148)

[Processing Description 37](#_Toc449046149)

[Interface Description 37](#_Toc449046150)

[Pseudo-code 37](#_Toc449046151)

[Modules Uses 39](#_Toc449046152)

[3.2 Modify Inventory item 39](#_Toc449046153)

[Processing Description 39](#_Toc449046154)

[Interface Description 39](#_Toc449046155)

[Pseudo-code 39](#_Toc449046156)

[Modules Uses 40](#_Toc449046157)

[4.0 Search Orders 41](#_Toc449046158)

[Processing Description 41](#_Toc449046159)

[Interface Description 41](#_Toc449046160)

[Pseudo-code 41](#_Toc449046161)

[Modules Uses 42](#_Toc449046162)

[4.1 Modify Order 42](#_Toc449046163)

[Processing Description 42](#_Toc449046164)

[Interface Description 42](#_Toc449046165)

[Pseudo-code 42](#_Toc449046166)

[Modules Uses 43](#_Toc449046167)

[5.0 Search permissions 43](#_Toc449046168)

[Processing Description 43](#_Toc449046169)

[Interface Description 43](#_Toc449046170)

[Pseudo-code 43](#_Toc449046171)

[Modules Uses 44](#_Toc449046172)

[5.1 Modify Permissions 44](#_Toc449046173)

[Processing Description 44](#_Toc449046174)

[Interface Description 44](#_Toc449046175)

[Pseudo-code 44](#_Toc449046176)

[Modules Uses 45](#_Toc449046177)

[5.2 Create New Permissions 46](#_Toc449046178)

[Processing Description 46](#_Toc449046179)

[Interface Description 46](#_Toc449046180)

[Pseudo-code 46](#_Toc449046181)

[Modules Uses 47](#_Toc449046182)

**Design Specification**

|  |  |
| --- | --- |
| Team: | Team A |
| Team Members: | John Boley, Justin Byrne, James Coltman, Marshal Gibson |
| Date: | 03/27/2016 |
| Project Title: | Williams Specialty Company - - Business Automation |
| Team Leader: | Justin Byrne |

# Scope

The function for this Design Specification is to serve as an overview and design pathway that directs (and outlines) each distinct aspect of the system design and architecture. This will lend insight and details regarding design, data, and interface components for the system. To ensure that each component satisfies the needs of the system model in addition to serving the demands of the user.

## System Description

Williams Specialty Company (WSC) wants to simplify (and improve) their workflow through the development (and implementation) of a proprietary Business Process Automation (BPA) application system. This system will include a host of business operations ranging from processing client and inventory data to generating vital business records and orders. Because this system is comprehensive in scope, the development should be approached as an overarching Business Process Management (BPM) strategy for WSC; maintaining and optimizing WSC’s core operational mechanisms such as business processes, reports, and data.

## Major Software Functions

The system is a Business Process Automation (BPA) application structure designed for employees to maintain, track, and process orders in an integrated environment. This portal allows employees the ability to enter new customer data, enter new order data, enter/retrieve work-in-progress status, client and inventory data processing operations, while supporting the fundamental business data structures, processes, and reports. Some of these technological functions include:

* Identities and Sets User Type
* Queries RDBMS for User Data
* Gets, Sets, and Processes Orders, Service Job(s), Catalog Items and Payment Data
* Generation Reports, Memos, and Notifications

Please see Figure 2.1: Class Hierarchy under Program Architecture, for a full Application Model.

## Database Description

The relational database employed by the Williams Specialty Company (WSC) application system, is Maria DB; a community-developed MySQL fork. This database will house various tables to support the core requirement(s) and operation(s) of this application, while providing a layer of data preservation through numerous InnoDB operations that encourage data stability and recovery. These operations include transaction commit and rollback, crash-recovery, row-level locking, clustered indexing, foreign key constraints, and more. The database schema for the WSC application systems will provide high-level users (or administrators) with a fast, reliable, and efficient means of managing user credentials, employee and customer records, orders, as well as inventory and production data.

## Design Constraints and Limitations

The WSC Business Automation Process (BPA) application system exhibits only a few minor design constraints (or limitations).

* **Unable to Transmit External Communications**

Internal messaging (or notifications) will be coordinated through application driven memos generated directly through the WSC – BPA application system; itself. External messages (or correspondence) via email or SMS will not be achievable through this system, and all interoffice notifications (memos) mitigated through this system will *only* be accessible through this application’s hosted server and/or internal network. Conclusively, the design of this application only allows for application communication(s) to be sent (and received) from within the application (itself), and outside (or external) correspondence is not available.

* **No Globalization**

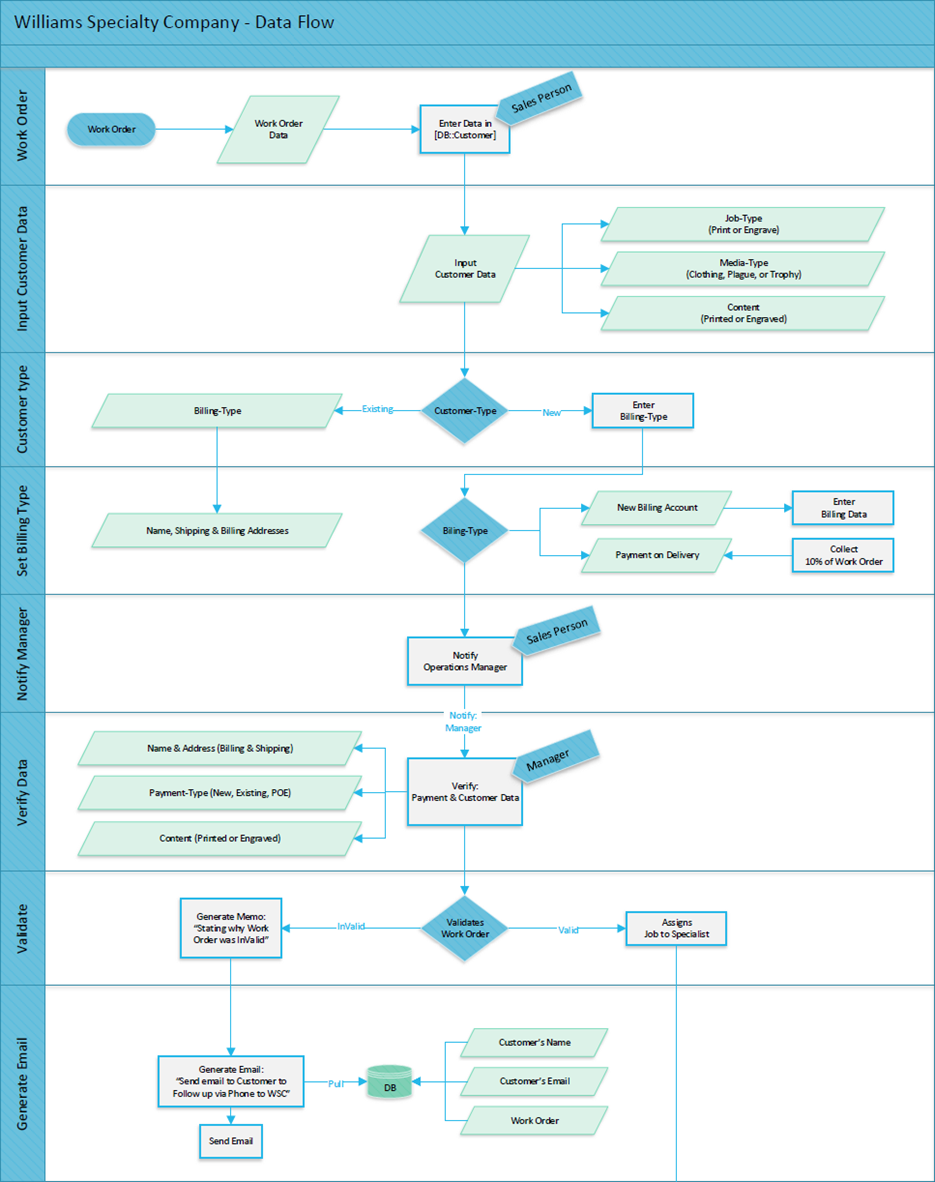
This application does not support any other known language(s), dialect(s), or speech than the U.S. English subset. Therefore, spoken-languages outside of this criterion will not refactor (or adjust) if it is requested to change its spoken-language context; therefore, it’s recommended for only English speaking users.

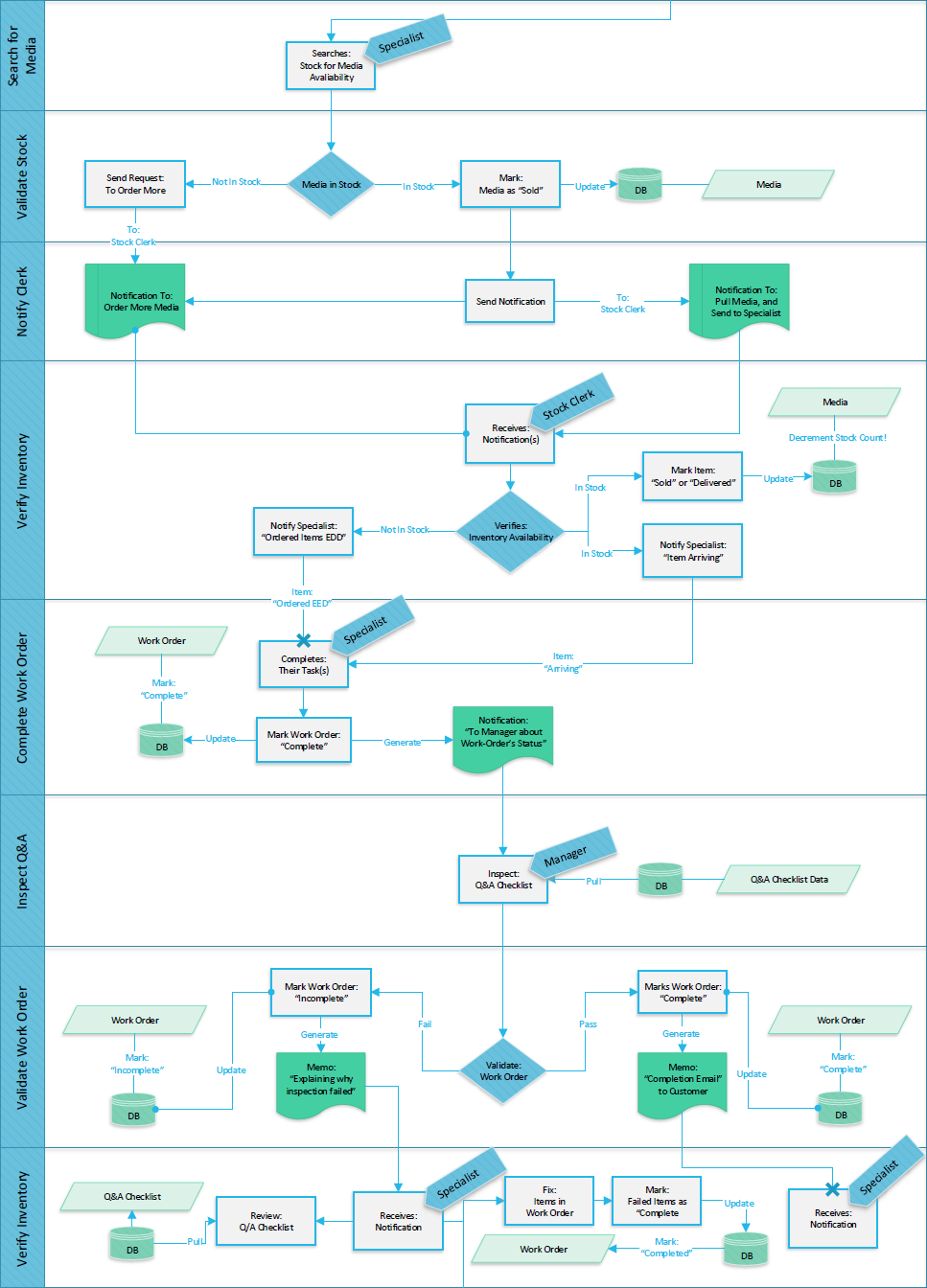
**Design Description**

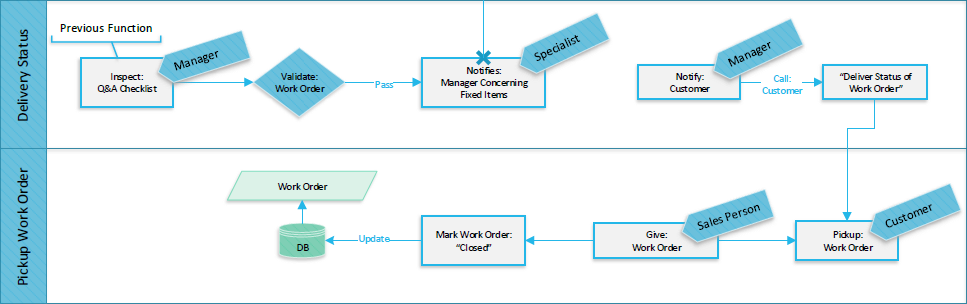
Data Description



**Data Flow**







## Program Architecture

Figure 2.1: Class Hierarchy – Elements & Relations



Figure 2.2: Class Hierarchy – Logical & Presentation Layer(s)



## Component Interfaces

Figure 2.3: Login Component



Figure 2.4: Work Order Creation



Figure 2.5: Input Customer Data



Figure 2.6: Input Billing Data



Figure 2.7: Input Shipping Data



Figure 2.8: Inventory Validation



Figure 2.9: Order Validation



Figure 2.10: Submit Work Order



Figure 2.11: Submit Quality Assurance Checklist



Figure 2.12: Complete Work Order



# Detailed Design

## 1.1 Login

### Processing Description

This process takes user-entered credentials and checks them against the database then either redirects bad credentials to the login in screen or grants access to those with approved credentials

### Interface Description

An html form with 2 text boxes labeled User Id and Password with a submit button.

### Pseudo-code

Login Button Click

Verify whether UserID is empty, then

IF UserID is empty

Show error message "No Employee ID Available"

ELSE

Capture userID to validate user

EndIF

Verify whether Password is empty, then

IF Password is empty

Show error message "No Password Available"

ELSE

Capture Password to validate user

EndIF

<<Security Area>>

Check userID and password against existing records in the credentials table in the Williams database

Securely pass credentials to validation page, then

Get user’s password from the Williams DB ‘credentials’ table; using the captured UserID

Validate Password (HASH) against user’s password from the Williams DB

IF Password (HASH) does not match, then

Return error message and redirect user to logout page

ELSE

Grant user access to administration page

EndIf

End Process

### Modules Uses

This Module is used to log into the WSC Business automation system.

## 1.2 Create Work Order

### Processing Description

This Process takes user input for creating a new work order for Williams Specialty Company, checks to make sure all required fields are filled and then redirects the user to the next stage of order entry.

### Interface Description

An html form with text boxes and dropdown menus for the various user input with a submit button.

### Pseudo-code

Order Submit Button Click

Start to validate Order Entries

Verify a jobtype has been selected from UI dropdown

If jobtype is empty

Show error message: "No Job Type selected"

ELSE

Capture jobtype for order storage

EndIf

Verify a mediatype has been selected from UI dropdown

If mediatype is empty

Show error message: "No Media selected"

ELSE

Capture mediatype for order storage

EndIf

Verify a whether description is empty

If description is empty

Show error message: "No Description entered"

ELSE

Capture description for order storage

EndIf

Pass order data into the orders table

IF successful

Show success message: "Order successfully entered."

Redirect user to Customer info form

ELSE

Show error message: "There was an error saving your order:

End Process

### Modules Uses

This module is used for WSC sales people to create a new work order.

## 1.3 Create Customer

### Processing Description

This Process takes user input for creating a customer record for Williams Specialty Company, checks to make sure all required fields are filled and then redirects the user to the next stage of order entry.

### Interface Description

An html form with text boxes for the various user input with a submit button.

### Pseudo-code

Customer Submit Button Click

Start to validate Customer Entries

Verify Customer Name is not empty

If CustomerName is empty

show error message: "No Customer Name entered"

ELSE

Capture CustomerName for order storage

EndIf

Verify whether Phone number is not empty

If phone is empty

Show error message: "No Phone Number entered"

ELSE

Capture phone for order storage

EndIf

Verify a whether email address is empty

If email is empty

Show error message: "No Email Address entered"

ELSE

Capture email for order storage

EndIf

Pass customer data into the orders table

IF successful

Show success message: "Customer successfully entered."

Redirect user to Billing information form

ELSE

Show error message: "There was an error saving your order:

End Process

### Modules Uses

This module is used for WSC sales people to create a new work order, specifically creating a customer entry.

## 1.4 Billing Data Input

### Processing Description

This Process takes user input for creating a new order in regards to billing address for Williams Specialty Company, checks to make sure all required fields are filled and then redirects the user to the next stage of order entry.

### Interface Description

An html form with text boxes and dropdown menus for the various user input with a submit button.

### Pseudo-code

Billing Submit Button Click

Start to validate Billing Entries

Verify Street Address is not empty

If billingAddress is empty

Show error message: "No Address Entered"

ELSE

Capture billingAddress for Billing storage

EndIf

Verify whether City is not empty

If billingCity is empty

Show error message: "No City entered"

ELSE

Capture billingCity for order storage

EndIf

Verify a state has been selected from UI dropdown

If billState is empty

Show error message: "No State selected"

ELSE

Capture email for order storage

EndIf

Verify whether Zip is not empty

If billingZip is empty

Show error message: "No Zip Code entered"

ELSE

Capture billingZip for order storage

EndIf

Pass billing data into the payments table

IF successful

Show success message: "Billing information successfully entered."

Redirect user to Shipping information form

ELSE

Show error message: "There was an error saving your information:

End Process

### Modules Uses

This module is used for WSC sales people to create a new work order, specifically inputting billing data.

## 1.5 Shipping Data Input

### Processing Description

This Process takes user input for creating a new order in regards to shipping address for Williams Specialty Company, checks to make sure all required fields are filled and then redirects the user to the next stage of order entry.

### Interface Description

An html form with text boxes and dropdown menus for the various user input with a submit button, with a check box to indicate that the same address is being used for both shipping and billing.

### Pseudo-code

Shipping Submit Button Click

Start to validate Shipping Entries

Verify Street Address is not empty

If shippingAddress is empty

Show error message: "No Address Entered"

ELSE

Capture shippingAddress for shipping storage

EndIf

Verify whether City is not empty

If shippingCity is empty

Show error message: "No City entered"

ELSE

Capture shippingCity for shipping storage

EndIf

Verify a state has been selected from UI dropdown

If shipState is empty

Show error message: "No State selected"

ELSE

Capture email for order storage

EndIf

Verify whether Zip is not empty

If shipZip is empty

Show error message: "No Zip Code entered"

ELSE

Capture shipZip for order storage

EndIf

Pass Shipping data into the customer table

IF successful

Show success message: "Shipping information successfully entered."

ELSE

Show error message: "There was an error saving your information:

End Process

### Modules Uses

This module is used for WSC sales people to create a new work order, specifically inputting shipping data.

## 1.6 Inventory Validation

### Processing Description

This process checks for a valid order ID then checks the WSC database Inventory table to see if the item is in stock.

### Interface Description

An html form with a text box for the order ID number user input with a submit button.

### Pseudo-code

Start to validate Inventory

Verify Order Number is not empty

IF orderNumber is empty

Show error message: "No Order Number Entered"

ELSE

Capture orderNumber for inventory verification

EndIf

Validate Order Number

Check Order number against the Order table in the Williams database

IF orderNumber does not exist in DB

Show error message: "Order Number invalid"

ELSE

Prompt User for Item Number and Quantity

Get itemNumber and quantity from UI

Validate Item number

Check itemNumber against inventory database

IF itemNumber does not exist in db

Show error message: "Invalid Item number or out of stock"

ELSE

Check quantity against inventory table in database

ENDIF

IF quantity is less than quantity table record

Show error message: "Invalid Item number or Out of stock"

ELSE

Show message: "Inventory Validated, Notifying specialists to begin work"

redirect user to Notifications form

ENDIF

End Process

### Modules Uses

This module is used by stock clerks to make sure there is inventory available for a new order to be created.

## 1.7 Order Validation

### Processing Description

This process is used by the Operations Manager to validate a new order coming into the WSC from a sales person. This takes an orderID input from the Manager and then marks the order as valid.

### Interface Description

An html form with a text box for the order ID number user input with a submit button.

### Pseudo-code

Start to validate order

Verify Order Number is not empty

IF orderNumber is empty

Show error message: "No Order Number Entered"

ELSE

Check Order number against orders database

ENDIF

IF orderNumber does not exist

Show error message: "Invalid Order Number Entered"

ELSE

Mark Order as Valid

EndIf

End Process

### Modules Uses

This module is used to validate a new order.

## 1.8 Perform Work

### Processing Description

This process is used by the Printing/Engraving Specialists to indicate an order has finished and indicates to Managers that QA tests are ready to be run. This takes an orderID input from the Specialist and then marks the order as ready for QA.

### Interface Description

An html form with a text box for the order ID number user input with a submit button.

### Pseudo-code

Start to Perform work on order

Verify Order Number is not empty

IF orderNumber is empty

Show error message: "No Order Number Entered"

ELSE

Check Order number against orders database

ENDIF

IF orderNumber does not exist

Show error message: "Invalid Order Number Entered"

ELSE

Mark Order as WorkComplete

EndIf

End Process

### Modules Uses

This module is used to mark an order as work-complete.

## 1.9 Quality Assurance Check

### Processing Description

This process checks for a valid order ID then allows the Manager to mark the order as complete-ready for deliver OR Needs Re-work and sends the appropriate memo.

### Interface Description

An html form with a text box for the order ID number and a check box to indicate pass/failure user input with a submit button.

### Pseudo-code

Start to Perform QA on order

Verify Order Number is not empty

IF orderNumber is empty

Show error message: "No Order Number Entered"

ELSE

Check Order number against orders database

ENDIF

IF orderNumber does not exist

Show error message: "Invalid Order Number Entered"

ELSE

Prompt user for Pass/Fail on Order

EndIf

IF passFail has Failed Inspection

Send Memo Message "Re-Work Needed"

ELSE

Send Memo Message "Order Complete, contact customer for order delivery"

EndIF

End Process

### Modules Uses

This module is used by managers to indicate a pass/fail of a QA check.

## 1.10 Order Completion

### Processing Description

This process checks for a valid order ID then allows the Manager to mark the order as closed-order out for delivery.

### Interface Description

An html form with a text box for the order ID number for user input with a submit button.

### Pseudo-code

Start to Complete order

Verify Order Number is not empty

IF orderNumber is empty

Show error message: "No Order Number Entered"

ELSE

Check Order number against orders database

ENDIF

IF orderNumber does not exist

Show error message: "Invalid Order Number Entered"

ELSE

Prompt user to mark complete Order Yes/No

EndIf

IF complete is marked NO Inspection

Redirect to Admin Page

ELSE

Send Memo Message "Order Closed"

EndIF

End Process

### Modules Uses

This module is used by managers or sales people to indicate an order is closed.

## 2.0 Search Existing Customer

### Processing Description

This process allows users to search customer records. It takes a “search by” option via dropdown menu – Customer ID, Last Name, and Phone Number. Then takes input from a text box to query the database for and returns the row(s) to the UI.

### Interface Description

A drop down menu for “search by”, a text box for search terms, and a submit button.

### Pseudo-code

Start Customer Search

Validate User entries

Verify a Customer Search By has been selected from a Dropdown menu

If cusSearchBy is empty

Show error message: "Please Select a Customer Search Option"

ELSE

Capture cusSearchBy for database SELECT

EndIf

Verify a Customer Search term has been entered

If cusSearchTerm is empty

Show error message: "No Search Term entered"

ELSE

Capture cusSearchTerm for database SELECT

EndIf

Search Customer Table for user data

IF No rows returned

Return message: "Customer does not Exist"

IF row returned

Return: Customer Details

### Modules Uses

This module allows users to access Customer data outside of an order creation process.

## 2.1 New Customer Creation

### Processing Description

This Process takes user input for creating a customer record for Williams Specialty Company, checks to make sure all required fields are filled and then Inserts the data into the database

### Interface Description

An html form with text boxes for the various user input with a submit button.

### Pseudo-code

Start Customer Management

Button Click: Create Customer

Verify Customer Name is not empty

If CustomerName is empty

Show error message: "No Customer Name entered"

ELSE

Capture CustomerName for new customer

EndIf

Verify whether Phone number is not empty

If phone is empty

Show error message: "No Phone Number entered"

ELSE

Capture phone for new customer

EndIf

Verify a whether email address is empty

If email is empty

Show error message: "No Email Address entered"

ELSE

Capture email for new customer

EndIf

Verify Street Address is not empty

If shippingAddress is empty

Show error message: "No Street Address Entered"

ELSE

Capture shippingAddress for new customer

EndIf

Verify whether City is not empty

If shippingCity is empty

Show error message: "No City entered"

ELSE

Capture shippingCity for new customer

EndIf

Verify a state has been selected from UI dropdown

If shipState is empty

Show error message: "No State selected"

ELSE

Capture email for new customer

EndIf

Verify whether Zip is not empty

If shipZip is empty

Show error message: "No Zip Code entered"

ELSE

Capture shipZip for new customer

EndIf

INSERT customer data into the customer table

IF successful

Return message: "Customer Created."

ELSE

Return error message: "There was an error creating a new Customer:

EndIF

End Process

### Modules Uses

This module is used for WSC employees to create a new customer record.

## 2.2 Modify Existing Customer

### Processing Description

This process allows a user to modify a field in a row in the customer table by entering a customer ID and selecting a field from a dropdown menu, and then entering a new value for it. The user submit button attempts to update the database.

### Interface Description

An html form with a text box for customer ID, a drop down menu for field, and another text box for change.

### Pseudo-code

Start Customer Modify

Validate User entries

Verify a Customer Id has been entered

If cusID is empty

Show error message: "Please Enter a Customer ID"

ELSE

Capture cusID for database UPDATE

EndIf

Verify a Customer column to modify has been selected from a Dropdown menu

If cusCol is empty

Show error message: "Please Select a Field you would like to modify"

ELSE

Capture cusCol for database UPDATE

EndIf

Update Customer Table for row cusID with VALUE cusCol

IF No rows modified

Return: "Invalid Data"

ELSE

Return: "Customer Modified"

EndIF

End Process

### Modules Uses

This module allows a user to make a change to a customer’s record.

## 3.0 Search Inventory

### Processing Description

This process allows users to search Inventory item records. It takes a “search by” option via dropdown menu. Then takes input from a text box to query the database for, then returns the row or rows.

### Interface Description

A drop down menu for “search by”, a text box for search terms, and a submit button.

### Pseudo-code

Start Inventory Management

Button Click: Search Inventory

Verify an Inventory Search by has been selected from a Dropdown menu

If invSearchBy is empty

Show error message: "Please Select an Inventory Search Option"

ELSE

Capture invSearchBy for database SELECT

EndIf

Verify an Inventory Search term has been entered

If invSearchTerm is empty

Show error message: "No Search Term entered"

ELSE

Capture invSearchTerm for database SELECT

EndIf

Search Inventory Table for item data

IF No rows returned

Return message: "Item does not Exist"

ELSE

Return: Item(s) Details

ENDIF

End Process

### Modules Uses

This module allows users to access Inventory Data.

## 3.1 Create New Inventory item

### Processing Description

This Process takes user input for creating a new inventory item for Williams Specialty Company, checks to make sure all required fields are filled and then Inserts the data into the database

### Interface Description

An html form with text boxes for the various user input with a submit button.

### Pseudo-code

Start Inventory Management

Button Click: Add Inventory

Verify Inventory Type is not empty

If invType is empty

Show error message: "No Inventory Type entered"

ELSE

Capture invType for new inventory item

EndIf

Verify whether Item Description is not empty

If description is empty

Show error message: "No Item Description entered"

ELSE

Capture description for new inventory item

EndIf

Verify a whether Item Cost is empty

If cost is empty

Show error message: "No Item Cost entered"

ELSE

Capture cost for new inventory item

EndIf

Verify Item Quantity is not empty

If quantity is empty

Show error message: "No Item Quantity entered"

ELSE

Capture quantity for new inventory item

EndIf

INSERT Inventory data into the inventory table

IF successful

Return: "New Item."

ELSE

Return error message: "There was an error adding a new Item"

EndIF

End Process

### Modules Uses

This module is used for WSC employees to create a new kind of inventory item.

## 3.2 Modify Inventory item

### Processing Description

This process allows a user to modify a field in a row in the Inventory table by entering an Inventory ID and selecting a field from a dropdown menu, and then entering a new value for it. The submit button attempts to update the database.

### Interface Description

An html form with a text box for Inventory ID, a drop down menu for field, and another text box for change.

### Pseudo-code

Start Customer Modify

Validate User entries

Verify an Inventory Id has been entered

If invID is empty

Show error message: "Please Enter an Inventory Id"

ELSE

Capture invID for database UPDATE

EndIf

Verify an Inventory column to modify has been selected from a Dropdown menu

If invCol is empty

Show error message: "Please Select a Field you would like to modify"

ELSE

Capture invCol for database UPDATE

EndIf

Update Inventory Table for row invID with VALUE invCol

IF No rows modified

Return: "Invalid Data"

ELSE

Return: "Inventory Modified"

EndIF

End Process

### Modules Uses

This module allows a user to make a change to an Inventory record.

## 4.0 Search Orders

### Processing Description

This process allows users to search order records. It takes a “search by” option via dropdown menu. Then takes input from a text box to query the database, then returns the row or rows.

### Interface Description

A drop down menu for “search by”, a text box for search terms, and a submit button.

### Pseudo-code

Start Order Management

Button Click: Order Search

Verify an Order Search by has been selected from a Dropdown menu

If ordSearchBy is empty

Show error message: "Please Select an Order Search Option"

ELSE

Capture ordSearchBy for database SELECT

EndIf

Verify an Order Search term has been entered

If ordSearchTerm is empty

Show error message: "No Search Term entered"

ELSE

Capture ordSearchTerm for database SELECT

EndIf

Search Order Table for orders data

IF No rows returned

Return message: "Order does not Exist"

ELSE

Return: Order(s) Details

ENDIF

End Process

### Modules Uses

This module allows users to access Orders data.

## 4.1 Modify Order

### Processing Description

This process allows a user to modify a field in a row in the Order table by entering an Order ID and selecting a field from a dropdown menu, and then entering a new value for it. The submit button attempts to update the database.

### Interface Description

An html form with a text box for Order ID, a drop down menu for field, and another text box for change.

### Pseudo-code

Start Order Modify

Validate User entries

Verify an Order Id has been entered

If ordID is empty

Show error message: "Please Enter an Order Id"

ELSE

Capture ordID for database UPDATE

EndIf

Verify a Order column to modify has been selected from a Dropdown menu

If ordCol is empty

Show error message: "Please Select a Field you would like to modify"

ELSE

Capture ordCol for database UPDATE

EndIf

Update Order Table for row ordID with VALUE ordCol

IF No rows modified

Return: "Invalid Data"

ELSE

Return: "Order Modified"

EndIF

End Process

### Modules Uses

This module allows a user to make a change to an order record.

## 5.0 Search permissions

### Processing Description

This process allows a Manager to search permissions records. Then takes input employee ID from a text box to query the database for, then returns the row or rows.

### Interface Description

a text box for search terms, and a submit button.

### Pseudo-code

Start Permissions Management

Button Click: Search Existing Permissions

Verify an Employee ID has been entered

If empID is empty

Show error message: "Please Enter an Employee ID"

ELSE

Capture empID for database SELECT

EndIf

Search Permissions Table for employee data

IF No rows returned

Return message: "Employee Permissions do not Exist"

ELSE

Return: Employee Permissions

ENDIF

End Process

### Modules Uses

This module allows users to see permissions Data.

## 5.1 Modify Permissions

### Processing Description

This process allows a Manager to modify a field in a row in the employee table by entering an employee ID and selecting a field from a dropdown menu, and then entering a new value for it. The submit button attempts to update the database.

### Interface Description

An html form with a text box for Employee ID, a drop down menu for field, and another text box for change.

### Pseudo-code

Start Permissions Management

Button Click: Edit Permissions

Validate User entries

Verify an Employee Id has been entered

If empID is empty

Show error message: "Please Enter an Employee Id"

ELSE

Capture empID for database UPDATE

EndIf

Verify a Permissions column to modify has been selected from a Dropdown menu

If prmCol is empty

Show error message: "Please Select a Field you would like to modify"

ELSE

Capture prmCol for database UPDATE

EndIf

Update Permissions Table for row empID with VALUE prmCol

IF No rows modified

Return: "Invalid Employee"

ELSE

Return: "Permissions Modified"

EndIF

End Process

### Modules Uses

This module allows a manager to make a change what an employee has access to or not.

## 5.2 Create New Permissions

### Processing Description

This process allows a Manager to create a new employee and give them application permissions based on job title by entering their information into text boxes and by selecting drop down options and submit it to the database.

### Interface Description

An html form with text boxes and drop down menus for employee information, and a submit button.

### Pseudo-code

Start Permissions Management

Button Click: Create New Permissions

Validate User entries

Verify an Employee ID has been entered

If empID is empty

Show error message: "Please Enter an Employee ID"

ELSE

Capture emp for database INSERT and database UPDATE

EndIf

Verify a Password has been entered

If password is empty

Show error message: "No password entered"

ELSE

Capture password for database INSERT

EndIf

Verify a Job Title has been entered

If jobTitle is empty

Show error message: "No Job Title entered"

ELSE

Capture jobTitle for database UPDATE

EndIf

INSERT Permissions Data into credentials table

IF not successful

Return error message: "There was an error adding a new permission"

ELSE

UPDATE Employee Table Row EmpID with VALUE jobType

IF not successful

Return error message: "There was an error updating Employee Job Title"

ELSE

Return: "Employee Permission Created"

EndIF

ENDIF

End Process

### Modules Uses

This module allows a manager to create a new employee and decide what they have access to according to their job title.