**Software Requirements Specification**

**for**

Warehouse 1

**Prepared by Aaron Dyke, Jordan Lawrence, Michael Blocker**

**September 19th, 2017**

**Table of Contents**

**Table of Contents 2**

**1. Introduction 3**

1.1 Purpose 1

1.2 Document Conventions 3

1.3 Intended Audience and Reading Suggestions 3

1.4 Product Scope 3

1.5 References 3

**2. Overall Description 4**

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Classes and Characteristics 4

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 5

2.6 User Documentation 5

2.7 Assumptions and Dependencies 5

**3. External Interface Requirements 5**

3.1 User Interfaces 5

3.2 Hardware Interfaces 5

3.3 Software Interfaces 6

3.4 Communications Interfaces 6

**4. System Features 6**

4.1 Transferring Items Between Warehouses 6

4.2 Adding Inventory to a Warehouse 7

4.3 Input Sales Information 8

4.4 Generation Sales Invoices 9

**5. Other Nonfunctional Requirements 9**

5.1 Performance Requirements 9

5.2 Safety Requirements 10

5.3 Security Requirements 10

5.4 Software Quality Attributes 10

5.5 Business Rules 10

**6. Other Requirements 11**

6.1 Database Requirements 11

6.2 Security Requirements 11

**Appendix A: Glossary 11**

**Appendix B: Section Authors 11**

# **Introduction**

## **Purpose**

The purpose of this product is to control, document, and organize the day to day operations of a warehouse system that sells automotive parts to customers. This document will cover all aspects of this product including the website frontend/backend as well as the database attached to the website. Automotive products will be stored and moved between various warehouses and the products will be tracked and organized using the database in conjunction with the website we develop. Products will be sold to customers, capturing the transaction. Various sales reports will be generated.

## **Document Conventions**

The following terminology is used.

* shall - This indicates a hard requirement that shall be implemented.
* should - This indicates a soft requirement that may be implemented provided we have time.

## **Intended Audience and Reading Suggestions**

The intended audience of this document includes BGCVA representatives, project managers, developers, testers, and documentation writers. Section 2 of this document is the overall description in which a reader shall find a high-level view of the hardware, user, and functionality requirements. Section 3 describes the external interface requirements including user, hardware, software and communications interfaces. Section 4 encompasses system features including all major features that shall be implemented in the application. Section 5 includes nonfunctional requirements such as performance, safety, and security requirements. Lastly, section 6 includes other requirements not covered by the previous sections including database and security requirements. Project managers and Developers may want to traverse the document focusing on section 2, 3, 4, 5, 6 in that order to get a solid overview and look further at the functionality behind the requirements. documentation writers should approach the document focusing on sections 2, 3, 6, 5, 4 traversing the specifics of the documents to write about how the pieces of the product fit together and how they integrate to the user. Testers may want to read through the sections 2, 5, 6, 4, 3 in that order to get an idea of what the performance of each aspect should be before understanding what each piece of the software does specifically.

## **Product Scope**

The project will be a simplistic website interface to interact with a backend database that stores, transfers, and organizes the products held in BGCVA warehouse system. The objective of this migration to an online system is to improve organization and track business statistics. Meeting these objectives shall increase productivity and accountability within BGCVA.

# **Overall Description**

## **Product Perspective**

The Warehouse 1 website and database system is a standalone product that will replace the current spreadsheet/pen and paper system used by BGCVA. The product we are creating will be used to improve the current system by way of organization, user tracking, and business analytics.

## **Product Functions**

* Create/Read/Update/Delete user accounts
* Order products
* Approve/Deny order
* Transfer products
* Create/Read/Update/Delete products and attributes
* Create/Read/Update/Delete customers and attributes
* Create sales invoice
* Generate sales reports

## **User Classes and Characteristics**

The purpose of this product is to control, document, and organize The system actors anticipated are administrator, main warehouse manager, sales associate for each delivery van, and customer. The administrator is a superuser who shall have the ability to perform CRUD operations any type of account. It shall also have the ability to add, edit, and remove any products or tables from the database of product. Sales reports are also generated via the administrator user. The main warehouse managers will be used primarily to approve product transfer requests. They shall be able to perform CRUD operations for products in the product database. They shall also transfer products from the main warehouse product table into the delivery van product tables by approving requests for product transfers. The sales associate user, each of which has a delivery van, should have minimal privileges, only being allowed to request product transfers from the main warehouse and remove items from their own product table using the sales receipt feature. The customer user should be able to request items from delivery van users. The most important users of this product are the administrator, main warehouse manager, and delivery van. The customer user would benefit the productivity of the business but is not an integral part of this product.

## **Operating Environment**

The operating environment includes a website and database which are both to be hosted by a cloud provider. The cloud provider is not specified by the client.

## **Design and Implementation Constraints**

The website and database shall be hosted by a cloud provider however, the client does not specify a preferred cloud provider. There are no limits on the language used to create the website frontend or backend. The database type was not specified allowing developers to implement the database of their choosing. Security of the database shall be critical as it will contain customer information. Transfers from the Main Warehouse database tables to the Delivery Vans tables shall include some sort of a lock as to not attempt to transfer the same to the two different Delivery Vans.

## **User Documentation**

User Documentation should include a tutorial with a write up on account creation, editing, and removal manuals for the administrator user. There needs to be documentation/a tutorial on the syntax of an order request file as well as a short write up of general use for the Warehouse manager and delivery van users.

## **Assumptions and Dependencies**

* Users of system have basic knowledge of computers
* Items within the warehouse system will not be moved using any other methods besides the product created.
* The product created will not have to interface/interact with other outside software

# **External Interface Requirements**

## **User Interfaces**

The User Interfaces should consist of a front facing website with a database backend. This website shall include an initial account login area, and an area for users to upload text files that, if properly formatted, will update items in stock, and transfer items between warehouses. The text file format may be created to fit coding needs, however a template file should be made available to all users. In addition to the file upload section, administrator accounts shall also have access to a section where they will be able to perform CRUD operations on user accounts, as well as create, and save invoices for clients, and in the future administrator accounts should be able to compute and share sales data for each salesman.

## **Hardware Interfaces**

The Hardware Interfaces for this project it should be assumed that the page will be accessed by any device that has access to the internet, including but not limited to smartphones, tablets, laptop computers, and desktop computers. Due to their constant mobility, salesmen will most likely be working from a smartphone or tablet device, while administrators and warehouse workers will most likely be accessing the database from a desktop or laptop computer. Therefore, the website should accommodate for mobile devices specifically where salesman have access and as such the file upload page should be easily accessible and useable by smaller devices. Any features used solely by administrators should be assumed to be accessed on a computer, and are not required to be smartphone compatible..

## **Software Interfaces**

The website, and subsequent back-end database, will be hosted on by a cloud provider such as google cloud, however any cloud provider may be chosen. Likewise the database back-end shall also be in the cloud, and according to the client any database may be used. The database software shall however be able to accommodate at minimum one thousand entries per table with the ability to add or remove columns and rows. This database, and website, should be protected from SQL Injection attacks and any sensitive information such as client information and account information should be stored in a hashed format in the database. The user accounts for the database should have permission limited only to the tables they need to access and only have permissions needed to perform their required tasks.

## **Communications Interfaces**

As stated above the website is required to accept .txt files from users and should execute queries to the database based on the user’s access privileges and database account privileges. The file should be transferred through FTP communication standards from the user’s device to the website. The database encryptions should be set up to be as strong as possible and can use methods such as MD5 hashing. To protect against SQL Injections, any forms on the site should be properly secured, especially any directly connected to the database. In addition, the database should safely run commands from the .txt file so as to prevent any unwanted attacks to occur from the file.

# **System Features**

*This section illustrates the functional requirements for the product by system features, the major services provided by the product. This is done by use case.*

## Transferring Items Between Warehouses

4.1.1 Description and Priority

*High Priority. The user shall be able to transfer items from one warehouse to another. This process will be done through the use of the website in conjunction with the underlying database. The process shall be secure and only available to those with sufficient permissions.*

4.1.2 Stimulus/Response Sequences

*The user will first be able to start the process of transferring inventory by navigating to the page on the website corresponding to the transferring of inventory. Here, the user will upload a document standardized for transferring inventory. After uploading, the document is read by the server. The server will then update the database with the list of changes provided by the document. The system will then notify the user of the success or failure of the operation.*

4.1.3 Functional Requirements

REQ-1: The system shall provide the user with a template for transferring inventory.

REQ-2: Instructions shall be provided by the system for how the template should be filled out by the user.

REQ-3: When uploaded, the document shall be checked for proper format. If the document is not in the proper format, an error message shall be sent to the user stating such and explaining where the error originated.

REQ-4: Each change noted by the document shall be made individually in the database. Any individual change that fails shall be noted to the user.

REQ-5: When the transfer succeeds, a copy of the operations made on the database shall be kept in a log.

## Adding Inventory to a Warehouse

4.2.1 Description and Priority

*High Priority. The user shall be able to add items to a warehouse that was not transferred from another warehouse. This process will be done through the use of the website in conjunction with the underlying database. The process shall be secure and only available to those with sufficient permissions.*

4.2.2 Stimulus/Response Sequences

*The user will first be able to start the process of adding inventory by navigating to the page on the website corresponding to adding inventory. Here, the user will upload a document standardized for adding inventory. After uploading, the document is read by the server. The server will then update the database with the list of changes provided by the document. The system will then notify the user of the success or failure of the operation.*

4.2.3 Functional Requirements

REQ-1: The system shall provide the user with a template for adding inventory.

REQ-2: Instructions shall be provided by the system for how the template should be filled out by the user.

REQ-3: When uploaded, the document shall be checked for proper format. If the document is not in the proper format, an error message shall be sent to the user stating such and explaining where the error originated.

REQ-4: Each change noted by the document shall be made individually in the database. Any individual change that fails shall be noted to the user.

REQ-5: When the additions succeeds, a copy of the operations made on the database shall be kept in a log.

## Input Sales Information

4.3.1 Description and Priority

*Medium Priority. The user will be a Sales Associate. When a Sales Associate sells items to a customer, they may take information on the customer. This information includes but is not limited to what the customer bought, how much they bought, and what they may have asked for that the Sales Associate did not have in possession. This process will be done through the use of the website in conjunction with the underlying database. The process shall be secure and only available to those with sufficient permissions.*

4.3.2 Stimulus/Response Sequences

*The Sales Associate will gather this information and input it into a Customer Sales Information template. That template will then be uploaded online. After uploading, the document is read by the server. The server will then update the database with the list of new information provided by the document. The system will then notify the user of the success or failure of the operation. How the customer information is stored in the database is TBD.*

4.3.3 Functional Requirements

REQ-1: The system shall provide the user with a template for entering customer information.

REQ-2: Instructions shall be provided by the system for how the template should be filled out by the user.

REQ-3: When uploaded, the document shall be checked for proper format. If the document is not in the proper format, an error message shall be sent to the user stating such and explaining where the error originated.

REQ-4: After the proper format is checked, the customer listed in the document will be looked up by the system. If the customer is not already in the system, then the system may prompt the user to input more information on the specific customer, or may ask the user to fill out a different form for adding a customer.

REQ-5: If the customer is found, the system will make each change for the customer individually to the database. An error will be displayed back to the user for any changes that failed in the database.

REQ-6: Multiple customers may be listed in each document. TBD

## Generating Sales Invoices

4.3.1 Description and Priority

*Medium Priority. The user will be a Sales Associate. When a Sales Associate sells items to a customer, they may sell the customer items without charging them. This sale includes what they bought and how many of that item they bought. The information, which is submitted via process 4.3, can then be generated into an invoice to bill the customer. This process will be done through the use of the website in conjunction with the underlying database. The process shall be secure and only available to those with sufficient permissions.*

4.3.2 Stimulus/Response Sequences

*The Sales Associate will look up information about a customer on a section of the website purposed for generating customer invoices. If the customer is found, the sales associate should be able to view the customer’s information including basic personal information and log of what they bought along with what they still owe.*

4.3.3 Functional Requirements

REQ-1: The system’s backend database shall keep information about customers including basic information and purchase logs.

REQ-2: Instructions shall be provided by the system for how the sales associate will generate the invoice.

REQ-3: The user shall be able to input some basic information about a customer (TBD) and be able to view that customer’s purchase information

REQ-4: The user shall be given an option to generate an invoice for a customer they looked up in the form of a text file.

REQ-5: This text file shall be downloadable for printing purposes.

# **Other Nonfunctional Requirements**

## **Performance Requirements**

For Performance Requirements, the backend database shall be able to support large tables, as well as multiple users accessing and editing these tables in real time. It should be assumed that multiple users can, and will, be logged into the same table, and editing information simultaneously. The reason for this assumption is that a main warehouse can be doing multiple exchanges simultaneously, with other smaller warehouses making sales at the same time, and it is possible for deliveries to the main warehouse to occur all at the same time. For the front end website, it should be assumed that multiple users will also be logged in and working with the site itself as well. A cloud platform should be chosen to be able to support all users and their work, while maximizing the uptime of the site and system.

## **Safety Requirements**

With this system, there exists a chance of possible loss, damage, or harm of customer personal data as well as any loss of time that can occur if products are misplaced or tables are deleted. Most accidents, such as accidental deletion of rows, can be mitigated by limiting the permissions each user account has in each databases. To mitigate against higher level attacks, such as SQL Injection, and data theft, the database should not store any customer information in plain text. All sensitive information should be stored in a hashed format. The database should also be coded to prevent any form of SQL injection.

## **Security Requirements**

As stated in section 5.2 above, the database should be coded as to prevent accidental deletions from occurring, while also coding the inputs so that attacks such as SQL Injection is not possible on the database. To protect the customer, and worker information, any sensitive information should be hashed using an MD5 hashing algorithm to prevent any leak of information in plaintext format. Finally the databases should limit not only who can access each database, but also the access levels they have in each database.

## **Software Quality Attributes**

The software quality should prioritize usability, reliability, and availability when shipped. The goal of the project should be to make the front-facing website clean and easy to navigate, with the back-end being easy to learn and edit. Both the front-end and back-end should attempt to maximize uptime during regular business hours. Portability and maintainability should be prioritized next to allow for software updates and the ability to access pages via mobile devices.

## **Business Rules**

5.5.1 Salesmen:

* An account made for a salesman will be the lowest account with access to the warehouses and databases. The salesman account should be able to access and query information of other warehouses, and have the ability to edit their specific warehouse.

5.5.2 Warehouse Worker / Supervisor:

* An account made for a warehouse workers and supervisors will have the ability to access, and edit information in their warehouse and any mobile warehouse.

5.5.3 Office worker:

* An account made for any office worker who requires the ability to access generate and print invoices for customers, order additional products for each warehouse, and track the sales of each salesman.

5.5.4 Administrator

* The administrator account should have the ability to edit, query, and access all warehouses and have the ability to also perform all functions of a office worker such as generate invoices, track salesman sales, and order product.

# **Other Requirements**

## Database Requirements

* + 1. Database used for the project shall be constructed in First Normal Form.
    2. Database shall be available online hosted through a cloud-based system.
    3. Must be accessible by the user application.
    4. Must be able to add, edit, and view information in the database.

## Security Requirements

* + 1. Database roles shall be limited to only the permissions befitting the role. Example: Sales Associates shall only be able to input and view customer information.
    2. Database and server shall be protected against malicious code. Information uploaded online shall be protected against SQL injection and any form of database manipulation through non-standard means.
    3. Different roles in the system shall be protected through the use of secure usernames and passwords. This includes the hashing of all passwords stored for users.

**Appendix A: Glossary**

**CRUD:**  Create, Read, update, and Delete

**Cloud-based:** a term that refers to applications, services or resources made available to users on demand via the Internet from a cloud computing provider's servers.

**MD5 Hash:** a cryptographic function that is used for encrypting and authenticating a message.

**SQL Injection:** a code injection technique used on SQL databases.

**Appendix B: Section Authors**

1. Introduction Aaron Dyke
2. Overall Description Aaron Dyke
3. External Interface Requirements Jordan Lawrence
4. System Features Michael Bloecker
5. Other Nonfunctional Requirements Jordan Lawrence
6. Other Requirements Michael Bloecker
7. Appendix A Aaron Dyke
8. Appendix B Aaron Dyke