



**HI-TEC ELECTRONIC SUPPLY LTD**

**M<sup>3</sup>CS Integrated Warehouse Management System**

**Final Report**

Prepared for:

**Nelson Eng, Instructor**

CSIS 3275 002 – Software Engineering

Development Team

**Matthew Lai**

**Manjot Sangha**

**Manveer Sidhu**

**Curtis Windsor**

**Sam Zhou**

## SUMMARY

Approximately 10 weeks ago, our development team set out to create a system to manage the warehouse operations of HI-TEC Electronic Supply Ltd., a fictional computer component supply company in the Lower Mainland. From the original project proposal:

*“The ultimate objective of this project is to create a comprehensive system for managing the movement of products in the warehouse. ”*

This document will serve as a confirmation of that goal; detailing the issues the development team encountered during the process as well as describing the current implementation of the system; allowing for comparison between it and the original proposed plan. It will also serve as a guideline for any future development and improvements to be made to the system.

## PROJECT DEVELOPMENT

At the beginning of this project, we defined separate roles and areas of responsibility for the development of the HITEC Warehouse Management System. From the original proposal we stated the roles as:

<b>PROJECT MANAGER:</b>	Matthew Lai
<b>WEBSITE DESIGN:</b>	Manjot Sangha and Manveer Sidhu
<b>SOFTWARE PROGRAMMING:</b>	Sam Zhou
<b>SERVER ADMINISTRATOR:</b>	Curtis Windsor

For the most part, this plan was followed. The only exception was that the Website module was developed entirely by Manjot, and Manveer instead moved over to the Software module. Our original prediction that most members of the team would contribute to the application was true as well, with Matt, Curtis, and Manveer contributing to the application in addition to Sam.

While the major aspects of the original system design were kept, there were some deviations:

- While not specified in the original design, the latest version of the application’s behaviour when handling a user with multi-form permissions became realized in the form of an admin-only menu bar that the admin user can utilize to switch forms.
- In the original proposal, the History form was stated to be a common form accessible by any user. In the latest version of the application, it is instead an admin-only form, as well as being the default form when any higher-tier user (multi-form access) logs in.
- In the original proposal, the Receiving and Shipping forms obeyed the same information-restriction scheme as the Inventory form. In the latest version of the application, this is not the case, with the user(s) only seeing a single ‘set’ of data.
- The original proposal stipulated that it would be possible for the user to add a completely new product to the database through the Receiving form. This function was removed from the latest version of the application due to time constraints. However, this feature could be easily added in the future.
- The original proposal called for our server to be hosted with AWS. However, the development team found that there were several issues preventing us from having a smooth implementation. In the end, we decision was made to simply host the database and website from a team member’s personal server. This allowed the team a greater amount of flexibility in that regard.

While we had estimated task completion dates and created a tentative schedule, in the end many of the 'deadlines' were not kept, with members working on and completing tasks as they had time. While this was not the most efficient process, due to the conflicting schedules of the development team as well as other factors, it was the one that was followed. We do not believe this affected the quality of the system in any major way as the development team proved to be agile enough to re-distribute the workload as need.

*See fig. II in the Appendix of this document for a comprehensive list of all technologies and tools used during the development of this system.*

## SYSTEM OVERVIEW

The current version of the system is essentially the same as the one proposed: a three module system composed of: 1) a server, 2) a website, and 3) an application. (Fig. II)

### 1) SERVER AND DATABASE

Running Linux, this server hosted our project's MySQL database and the actual website for the website module. Some details:

- a. Runs on DMZ
- b. LAMP Stack – Linux, Apache, MySQL, PHPmyadmin
- c. EMACS – for better text editing
- d. Changed permissions of /var/www to 777 and appended (insecure but effective)
- e. Config files altered - /etc/init.d/mysql.conf, /etc/init.d/php5.conf, etc/init.d/mysql/apache2.conf (various adjustments to make it easier for users to connect easier)
- f. Changed ports:
  - i. 8080 - HTTPS
  - ii. 2222 - SSH
  - iii. 21 - FTP
  - iv. 22 - SFTP
- g. OpenSSHserver - SSHkeys (public/private)

Our database which focused on the warehouse products contains 20 tables; each holding pertinent information about the movement of items in the system. Of course, some tables, such as the EMPLOYEE table, contain only the most basic information, since the assumption is that other systems exist alongside ours.

### 2) WEBSITE

Having read-only access to our database, the website's purpose was to provide a method for our clients to view our inventory stock, as well as to give them our contact information (Fig. III). While being not terribly complex, the website module of this system fulfills its purpose of displaying information from the database while being in the 'proof-of-concept' nature of the current release. Future production versions are intended to be more complete.

At present, the main feature of the website module is its ability to display the products the company stocks, with the ability to search for specific products and filter by category. The website also displays whether the item is in-stock or not.

### 3) APPLICATION

Written in C#, the application for this system was designed to provide the employees of the company a streamlined and easy-to-use GUI to connect to the database in order to perform important tasks in the warehouse. The application has the following forms:

#### a. **Login Form**

- i. The form that appears when the program first boots up; the user is first prompted for their login details and after they are entered, the appropriate form based on the user's permission level will appear. The permission groups from lowest to highest are:
  1. Warehouse Employees
  2. Receiving Clerks
  3. Shipping Clerks
  4. Operations Managers
  5. Warehouse Managers
  6. Administrators

#### b. **History Form**

- i. The default form for managers and administrators, this form simply displays a list of all past shipments and orders that have passed through the system. Also from this form it is possible for those users to access the other forms.

#### c. **Inventory Form**

- i. The default form for regular warehouse employees, this simple form displays the physical location of the products inside the warehouse. The user is also able to look up a specific product by its SKU.
- ii. This form in particular followed the original information-access-restriction scheme stipulated in the proposal, as higher-tier users will see additional information about the product.

#### d. **Receiving Form**

- i. The default form for receiving clerks, this form deals with incoming products to the warehouse. It allows for the user to process the items, and then update the stock stored in the database.

#### e. **Shipping Form**

- i. The default form for shipping clerks, this form deals with outgoing products to our company's clients. It allows for the user to look up shipments and mark them as shipped when the items actually leave the warehouse.
- ii. This form also has the ability to create new shipments in the database and store them for later processing.

## **ANALYSIS OF THE PROJECT AND ITS DEVELOPMENT**

In retrospect, it is the opinion of the development team that the project progressed and concluded relatively smoothly. However, no project is perfect and ours is no exception. There were several areas and issues that could have been improved upon.

The first was the lack of a concrete timeline for task completion. While a tentative one was created, it was not followed and essentially ignored. While the application was completed within the given timeframe, 80% of the application module was constructed in the final two weeks, and required much more resources than originally estimated.

This can be attributed to the second issue, which was the temporary failure of the project management to properly control the time and manage the team. While the team delivered in the end, a great deal of stress could have been avoided if proper leadership and management were practiced earlier in the development process.

The third issue was the fact that the development team was formed later in the project than most, as the current line-up is the result of 3 other teams experiencing a member shake-up. This alone cost the project 4 weeks of work.

Essentially, most issues with the development of the system occurred or were related to the planning and organization, with few having to do with the actual development. The one exception to this was a major incident involving the Github repository which occurred late into the development of the application. Due to a severe miscommunication between team members, some code was overwritten in the repository and it became difficult to salvage. The error was only fixed after 3 hours of dedicated troubleshooting. Fortunately, no work was lost and the code blocks were successfully merged into a single stable version.

## **FUTURE VERSIONS AND FEATURES**

While the current version of the system is fully functioning with all major features implemented, there are several improvements that could be made and other functions that can be added to enhance the system.

1. In the current release, the user credentials including the passwords are stored in the database in plain-text format for ease of testing. This obviously poses a huge security risk. In the final production release, we would insist on salting the passwords in order to protect the employee information.
2. The one feature in the application that was originally planned but was cut: the capability for the user to add a completely new product into the database. While not terribly complex from a coding standpoint, this feature was cut from the program during the aforementioned time-crunch in order to ensure that a working program was ready to be presented to the project stakeholders.
3. Currently in the application, the Shipping form has the capability for creating new shipments; adding products to a new shipment complete with destination and client information before saving it to the database for later processing. In a future release, we'd like to expand this capability to the Receiving form as well; giving the user the ability to create new orders to be sent out to suppliers in order to receive additional products.
4. Due to the prototype nature of the system, there is currently no method to correct data entry mistakes that occur on the application side while remaining in the application; meaning the development team had to log into PHPmyadmin to correct errors. Because inventory mistakes due to miscounts or theft

are about to occur, in the final release there will be another form for updating the inventory stock amounts. This new form would be restricted to higher-tier users for security reasons.

5. Finally, aesthetic improvements to the website could be added, as well as other content in order to create a fuller user experience. The current website was designed solely as a 'bare-bones' proof-of-concept module in the overall system.

## APPLICATION TECHNICAL MANUAL

This section will detail how to operate the application of the M3CS Integrated Warehouse Management System. Please refer to the Appendix of this document for images of the interfaces in question.

### Part I: **Main Login Form** (Fig. IV)

Here the user is prompted for two inputs: their employee ID and their password.

- Once the user enters their credentials and hits the 'Login' button underneath, the next form appropriate to their permission group will appear.
- Should their inputted credentials be incorrect, an error message will appear.

### Part II: **History Form** (Fig. V)

In this form the user is presented with two lists; one of the order history and the other of the shipment history.

- The user is able to click on any of the column headers, allowing them to sort it each list on the selected list.
- As the assumption is that only higher tier users have access to this interface; the user is able to use the 'Choose Form' menu in the upper left to select which form they wish to view.

### Part III: **Inventory Form** (Fig. VI)

The primary function of this interface is for the user to quickly look up the physical warehouse location of products in the database.

- When first loaded, all the products are displayed.
- The user is able to enter a specific product SKU in the upper left text box. Upon pressing the 'Search' button adjacent to it, the display will update to show the searched item. Should the user enter an incorrect input, an appropriate error message will appear.
- Pressing the 'Refresh' button in the upper right will bring the display back to the full list of products.
- Should the user logged in be a higher tier user, they will be able to view the product's Manufacturer and Price as well.

### Part IV: **Receiving Form** (Fig. VII)

The primary function of this interface is for the user (the receiving clerk) to be able to quickly process the items in a received order and update the product stock in the database.

- Upon loading the form, the user will input an order's identification number in the Order ID text box. Upon clicking the 'Order Lookup' button underneath, the products contained within the looked up order will appear in the top display. After the product details, there is a column named 'DAMAGED' which indicates whether the received item was damaged. The user can then mark them accordingly.
- Should the product be in good condition, selecting the product row and clicking the 'Add' button will add the selected item to the lower display. The original row in the top display will then turn green.
- When the user is finished, clicking the 'Execute Updates' button will add the item amounts to the product stock in the inventory. Damaged items are not added.

- All items in an order must be processed in order to execute the updates; if any items are not highlighted in green or marked as damaged, an appropriate message will appear.

#### Part V: **Shipping Form** (Fig. VIII)

The primary purpose of this interface is for the user (the shipping clerk) to be able to quickly processing outgoing shipments of our products to client locations.

- The left side display is for view the details of shipments that are already stored in the database. The Shipment ID text field is for the user to enter the identification number of a specific shipment in order to view its details. Upon clicking the 'Load Shipment' button, the display will update to show the shipment's details.
- If the user does not specify a shipment and leaves the Shipment ID field blank, then upon hitting the 'Load Shipment' button the full list of all shipments will be displayed, along with their ship status.
- When a shipment is marked as shipped, the appropriate inventory numbers in the database will decrease to reflect the updated stock. If the user attempts to ship a shipment that the company is unable to fulfill due to not having enough products, an appropriate message will appear.
- The right-side controls are for creating new shipments and adding them to the database. At first, the controls are disabled; the user must hit the 'New Shipment' button in order to begin the process.
- Using the Destination dropdown menu, the user selects which location the shipment is departing to.
- Using the Product SKU and Quantity text fields, the user indicates how many of what product they would like to add to the new shipment. When the user clicks the 'Add' button, the indicated product will be added to the list display.
- Selecting a product in the display and clicking the 'Remove' button will remove the product from the new shipment.
- Once the user is satisfied with their created shipment, they will hit the 'Create Shipment' button to save the details to the database. They can then view the details on the left-side display and mark it as shipped when the time comes.

## APPENDIX

TECHNOLOGY NAME	MODULE
MySQL	Database
Linux	Server
Apache	Server
HTML	Website
JavaScript	Website
PHP	Website
Bootstrap	Website
C#	Application
DEVELOPMENT TOOL NAME	MODULE
MySQL Workbench 6.3 CE	Database
Mockaroo	Database
PHPmyadmin	Database
Adobe Dreamweaver	Website
Codelobster	Website
FileZilla	Website
Microsoft Visual Studio 2015	Application
Microsoft Visual Studio 2012 Express	Application
ReSharper	Application
Git	All
Github	All
SourceTree	All

Figure I: List of all technologies and development tools used:

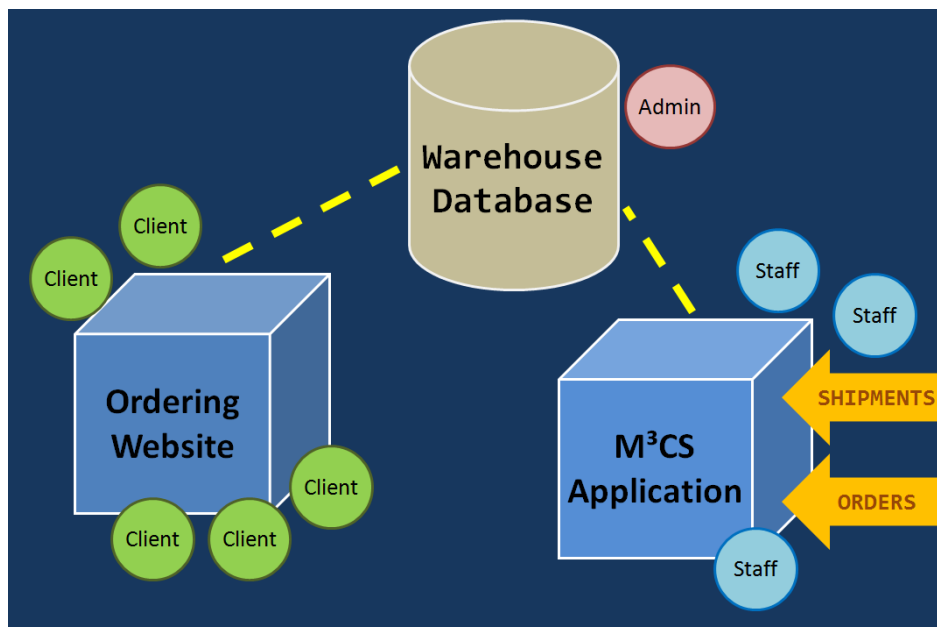


Figure II: System Overview



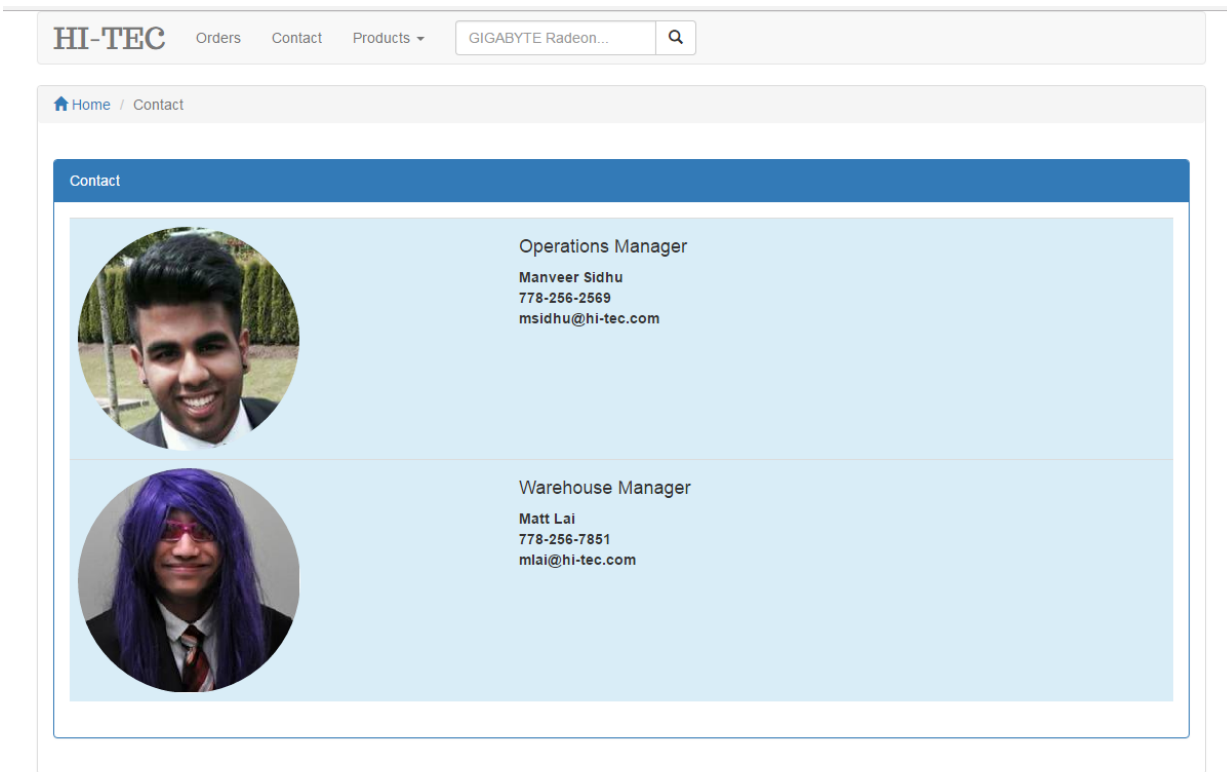


Figure III: HI-TEC Website Contact Page

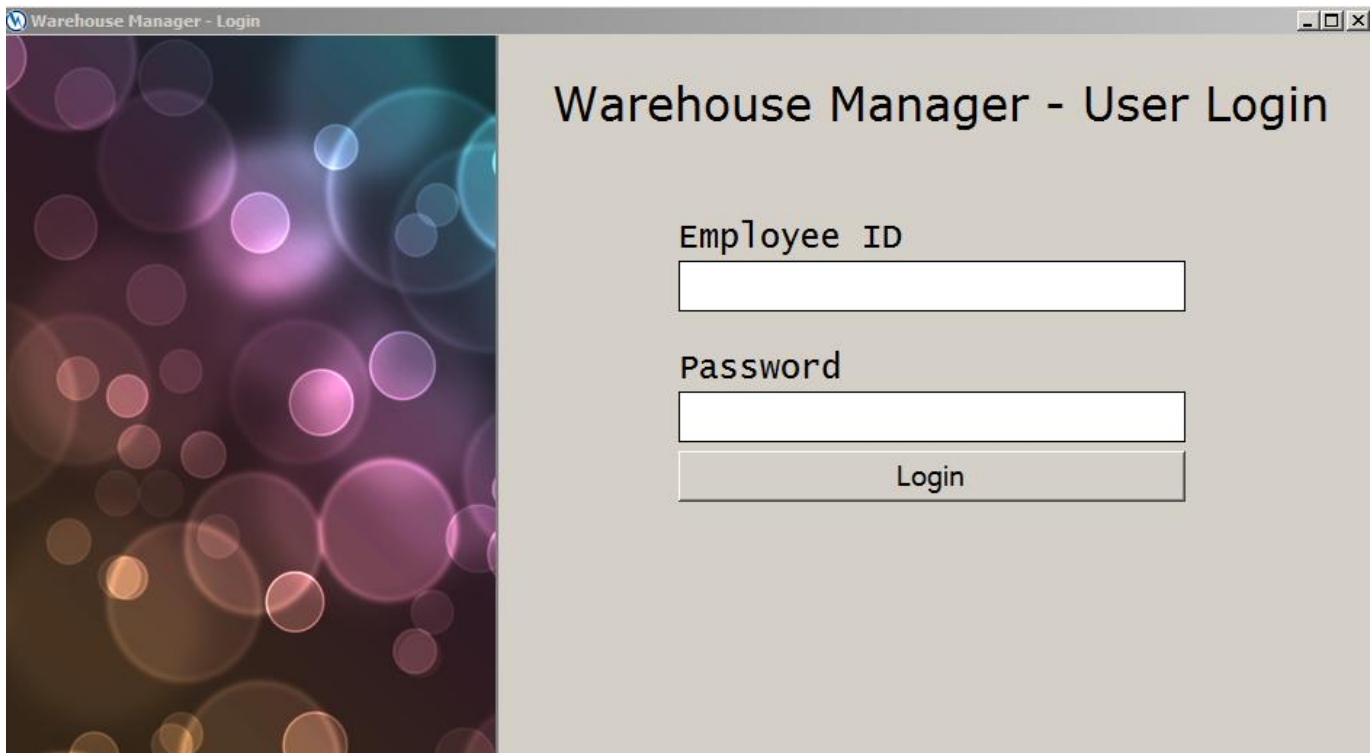


Figure IV: Main Login Form

**Warehouse Manager - History**

System Users Help

Choose Form ▾ **Shipment History** **Order History**

ID	Destination	Representative	Phone	Date	Stat.
3	Store 8	Sharon Sims	(154)371-0267	19/07/2016	✓
4	Store 3	Jason Rose	(680)212-8791	25/07/2016	✓
2	Store 4	Sean Ryan	(146)534-7673	01/08/2016	✓
5	Store 8	Sharon Sims	(154)371-0267	01/08/2016	✓
10	Store 13	Deborah Kennedy	(860)400-9218	08/08/2016	✓
13	Store 16	Lawrence Ruiz	(375)732-9065	08/08/2016	✓
1	Store 4	Sean Ryan	(146)534-7673	16/08/2016	✓
12	Store 20	Russell Burton	(580)533-4290	16/08/2016	✓
9	Store 9	Gloria Alvarez	(604)438-0748	19/08/2016	✓
15	Store 4	Sean Ryan	(146)534-7673	21/08/2016	✓
17	Store 10	Marie Carter	(783)464-1035	31/08/2016	✓
19	Store 17	Jeremy Lewis	(583)716-2756	13/09/2016	✓
20	Store 14	George Andrews	(646)947-8242	14/09/2016	✓
8	Store 15	Diane Simpson	(375)732-9065	17/09/2016	✓
11	Store 9	Gloria Alvarez	(604)438-0748	19/09/2016	✓
14	Store 16	Lawrence Ruiz	(375)732-9065	26/09/2016	✓
18	Store 12	Brandon Carr	(153)575-1258	02/10/2016	✓
7	Store 6	Keith Jacobs	(197)115-7452	11/10/2016	✓
6	Store 4	Sean Ryan	(146)534-7673	16/10/2016	✓
16	Store 18	Shawn Alvarez	(703)766-1583	25/10/2016	✓

ID	Supplier	Contact	Email	Phone	Date
5	Bednar, Purdy and Rempel	Kevin Willis	kwillis4@csmonitor.com	(717)395-8025	06/07/2016
1	Collier, Quigley and Weber	Edward Davis	edavis2@studiopress.com	(376)269-7822	08/07/2016
16	Stamm Group	Debra Adams	dadams8@myspace.com	(197)115-7452	12/07/2016
14	Rau-Daniel	Paul Diaz	pdiaz5@wordpress.com	(382)371-6216	20/07/2016
9	Stoltenberg, Bernier and Blick	Clarence Rivera	crivera1@alibaba.com	(140)233-4202	28/07/2016
10	Bednar, Purdy and Rempel	Kevin Willis	kwillis4@csmonitor.com	(717)395-8025	05/08/2016
11	Ortiz-Kub	Alice Thomas	athomas3@virginia.edu	(703)766-1583	06/08/2016
3	Ortiz-Kub	Alice Thomas	athomas3@virginia.edu	(703)766-1583	15/08/2016
15	Hemann LLC	Justin Rogers	jrogers9@360.cn	(914)311-8373	20/08/2016
18	Towne-Auer	Debra Bryant	dbryant0@furl.net	(871)440-0104	22/08/2016
8	Stoltenberg, Bernier and Blick	Clarence Rivera	crivera1@alibaba.com	(140)233-4202	04/09/2016
6	Stamm Group	Debra Adams	dadams8@myspace.com	(197)115-7452	04/09/2016
12	Ortiz-Kub	Alice Thomas	athomas3@virginia.edu	(703)766-1583	05/09/2016
2	Towne-Auer	Debra Bryant	dbryant0@furl.net	(871)440-0104	25/09/2016
4	Bednar, Purdy and Rempel	Kevin Willis	kwillis4@csmonitor.com	(717)395-8025	25/09/2016
13	Rau-Daniel	Paul Diaz	pdiaz5@wordpress.com	(382)371-6216	01/10/2016
19	Shanahan and Sons	Shawn Sims	ssims7@abc.net.au	(264)306-8677	01/10/2016
7	Stamm Group	Debra Adams	dadams8@myspace.com	(197)115-7452	06/10/2016
17	Shanahan and Sons	Shawn Sims	ssims7@abc.net.au	(264)306-8677	12/10/2016
20	Hemann LLC	Justin Rogers	jrogers9@360.cn	(914)311-8373	15/10/2016

Nelson Eng Connected

Figure V: History Form

**Warehouse Manager - Inventory**

System Users Help

Search Refresh

	SKU	PRODUCTNAME	STOCK	SHELFNUM	AIULENUM	SECTIONNAME
▶	1	2x4GB Nature Valley DDR3 RAM	70	5	103	CENTRAL
	2	2x8GB KitKat DDR3 RAM	0	1	104	CENTRAL
	3	2x8GB Coffee Crisp DDR4 RAM	21	4	101	CENTRAL
	4	Enterprise 4-Slot Motherboard	137	1	103	CENTRAL
	5	Nimitz 4-Slot Motherboard	64	3	103	CENTRAL
	6	Kitty Hawk 2-Slot Motherboard	20	1	101	CENTRAL
	7	ATD Senna 6GB Video Card	331	1	101	CENTRAL
	8	GTZ Vettel 8GB Video Card	253	4	103	CENTRAL
	9	GTZ Clarkson 8GB Video Card	99	5	101	CENTRAL
	10	Quad-core 4.0GHz Socrates Processor	179	4	104	CENTRAL
	11	Quad-core 3.5GHz Plato Processor	336	2	101	CENTRAL
	12	Quad-core 4.0GHz Nietzsche Processor	340	5	104	CENTRAL
	13	Hummingbird 120GB SSD	161	6	101	CENTRAL
	14	Pelican 2TB HDD	372	2	103	CENTRAL
	15	Falcon 500GB SSD	25	1	101	CENTRAL
	16	Franklin 650W PSU	238	3	104	CENTRAL
	17	Faraday 450W PSU	208	1	104	CENTRAL
	18	Tesla 650W PSU	117	2	103	CENTRAL

Matt Lai Connected

Figure VI: Inventory Form

Warehouse Manager - Receiving

System Users Help

order ID

Order Lookup

Add

Execute Updates

Manveer Sidhu Connected

### Figure VII: Receiving Form

[illegible]

Figure VIII: Shipping Form