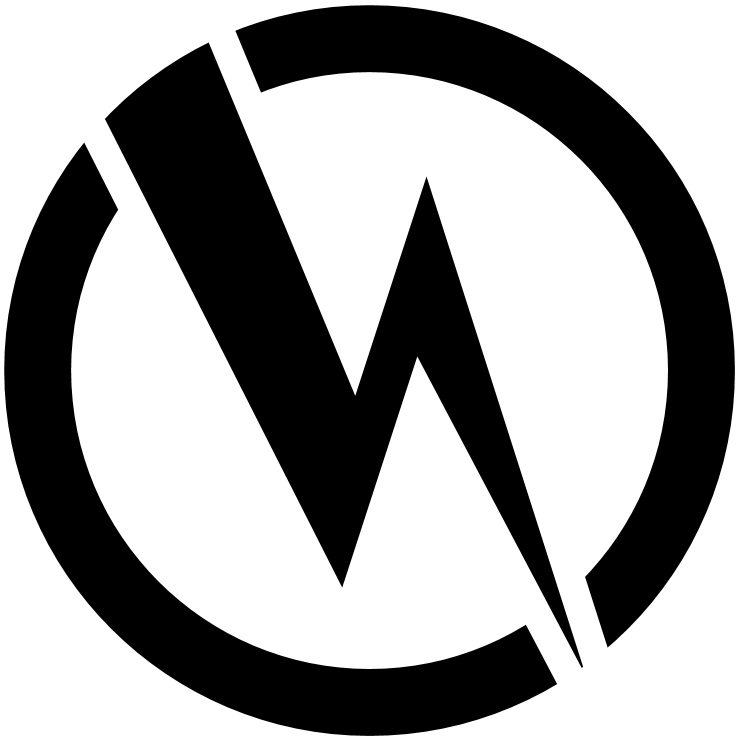
****

**HI-TEC ELECTRONIC SUPPLY LTD**

**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**

**TABLE OF CONTENTS**

Project Development Plan

System Overview

Application Technical Manual

Analysis of the Project and its Development

Future Versions and Features

Appendix

**PROJECT DEVELOPMENT PLAN**

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:

**PROJECT MANAGER:**  Matthew Lai

**WEBSITE DESIGN:** Manjot Sangha and Manveer Sidhu

**SOFTWARE PROGRAMMING:** Sam Zhou

**SERVER ADMINISTRATOR:** 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’ where 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.

**SYSTEM OVERVIEW**

The latest 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.

1. **SERVER AND DATABASE**

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

* 1. Runs on DMZ
  2. LAMP Stack – Linux, Apache, MySQL, PHPmyadmin
  3. EMACS – for better text editing
  4. Changed permissions of /var/www to 777 and appended (insecure but effective)
  5. 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
  6. Changed ports:
     1. 8080 - HTTPS
     2. 2222 - SSH
     3. 21 - FTP
     4. 22 - SFTP
  7. 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.

1. **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. The technologies used to develop this site include:

* 1. HTML
  2. PHP
  3. Bootstrap (CSS Framework)
  4. JavaScript

1. **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:

* 1. Login Form
     1. 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
  2. History Form
     1. 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 from it is possible for those users to access the other forms.
  3. Inventory Form
     1. 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.
     2. 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.
  4. Receiving Form
     1. 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.
  5. Shipping Form
     1. 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.
     2. This form also has the ability to create new shipments in the database and store them for later processing.

APPLICATION TECHNICAL MANUAL

**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

APPENDIX