

Warehouse Management System

Project Plan Document

CPSC-430: Writing Assignment 2

Dr. Anewalt

09 October 2017

Warehouse Group 2: Samantha Miller, Jacques Troussard, Taylor Dohmen

Table of Contents

Table of Contents	2
Section 1: Introduction	3
1.1 Purpose	3
1.2 Scope	3
1.3 Document Conventions	3
1.4 Document Overview	3
Section 2: Project Description	4
2.1 Project Scope	4
2.2 User Characteristics	4
2.3 Functional Requirements	4
2.4 Constraints	9
Section 3: Project Schedule	11
3.1 Approach	11
3.2 Milestones and Deliverables	11
3.3 Work Breakdown Structure	12
3.4 Gantt Chart and Task Dependency Table	15
Appendix A. Glossary	18
Appendix B. System Documents	19
B.1 Product Import	19
B.2 Invoice Example	20
Appendix C. Document Contributions	21

Section 1: Introduction

1.1 Purpose

The purpose of this document is to describe and provide a detailed plan for development and execution of the Warehouse Management System (WMS) project in Computer Science (CPSC)-430. The audience of this document includes the writers' professor Dr. Karen Anewalt and BGCVA's representative, Gusty Cooper.

1.2 Scope

The scope of this project is to provide a software application that manages automobile inventory for a primary and mobile warehouses, including sales and invoices. Cooper represents the primary stakeholder in this project. The project will upload, display, and search products that BGCVA sells. This application intends to improve organization and track business statistics for the stakeholders.

1.3 Document Conventions

The document follows this terminology:

- “**shall**” indicates a hard requirement that shall be implemented.
- “**should**” indicates a soft requirement that may be implemented if time allows.

1.4 Document Overview

This document is organized as follows:

- Section 2 provides a general description of the system, including its overview, user characteristics, functional requirements, and constraints.
- Section 3 contains the project's schedule, including the approach, milestones deliverables, work breakdown structure, Gantt chart, and task dependency diagram.
- Appendix A contains a glossary of commonly used terms.
- Appendix B provides examples and explanations of project documents.
- Appendix C enumerates the document contributions by each member in the group.

Section 2: Project Description

2.1 Project Scope

WMS will be an application by which BGCVA employees can manage inventory and sales records. It will digitally store information about each warehouse, the products in each warehouse, product sales, and the employees. Using WMS, employees should be able to query product and sales data, generate sales reports and invoices, and update inventory. Inventory updates will consist of operations related to transferring items between warehouses, adding new product deliveries into the system, and selling items to customers. All databases will be hosted on a server in the cloud, and will be accessible via a browser interface. Only employees will be able to access the system, and will need to login with a unique username and password to use the application.

2.2 User Characteristics

This web application shall interface with the clients as an internal tool to track products, sales, and employee metrics. There shall be administrators who interface with the system to provide technical support, but have no need to engage with the product, unless other users need help. They may be able to update product fields and perform site maintenance as needed. They should have access to every feature on the site for testing, explanation, and future improvement. Warehouse managers who should also have full site access, except for administrator pages, for training and business information, including managing products and employees as well as their metrics. Client's employees are sales associates who move product and make sales using their mobile warehouse vans to local customers. Sales associates should not have access to the user accounts and metrics portions of the site. Administrators, managers, and sales associates should be able to access the "working" parts of the site, including searching for products, creating and displaying invoices, and importing product files to update the system. Customers are local automobile repair and supply shops who buy the products the sales associates offer. Customers and the general public may not login or create accounts to use the application because it remains an tool internal to the business and its associates.

2.3 Functional Requirements

2.3.1. Log into System

Description: A user accessing the site shall provide a email and password to use the application.

Main Flow:

1. User accesses site by Internet browser.

2. System displays login screen, prompting email and password fields, and a Login button.
3. User types email and password and clicks Login.
4. System verifies matching user information and displays the Home page of the site.

Alternate Flow A:

1. User accesses site page that requires login, like the Home or Import page.
2. The system redirects any user without a session to the Login page.

Alternate Flow B:

3. The user types either the email or password. The Login button is disabled until both fields are filled.

Alternate Flow C:

3. The user types a wrong email or password.
4. The system checks the provided information and returns a error message proclaiming the email or password was invalid.
5. Repeat until Main Flow 4 is met.

2.3.2. Log out of System

Description: A logged-in user shall exit their logged-in session on the site.

Main flow:

1. A logged-in user clicks the Logout button on the menu.
2. The system removes the user from the session and displays the Login page of the application.

2.3.3. Search for Product

Description: Any user shall search for products based on warehouse location, product name, or product number.

Main Flow:

1. A logged-in user navigates to the Search page via the menu bar.
2. The system displays the Search page, which offers options to search for products by warehouse location, product name, or product number, as well as Search and Reset buttons.
3. The user enters requirements in the form and clicks Search.
4. The system queries all related information and displays all product information in the table.

Alternate Flow A:

3. The user clicks the Reset button instead of Search.
4. The system clears the fields of the search boxes.

Alternate Flow B:

5. The user clicks the Reset button after searching.
6. The system clears the returned table and search fields.

2.3.4. Import Inventory File

Description: Any user shall upload a Comma Separated Values (CSV) file into the system which specifies what changes are being made to the warehouse inventory. The system shall then update the product database to account for the specified changes.

Main Flow:

1. User navigates to an inventory management page and selects the option to upload an inventory management file. (See Appendix B.1 for preliminary import configuration.)
2. System displays an Upload window including a File Explorer and Upload button.
3. User navigates the File Explorer, selects the desired file, and clicks the Upload button.
4. The system parses the file and updates the inventory database based on the file's specification. If the operation is time intensive, the system should display a progress indicator to the user.
5. The system displays a message informing the user that the operation was executed successfully.

Alternate Flow A:

4. The system parses the file and finds that it is improperly formatted.
5. The system displays a message informing the user that the file is improperly formatted. Return to Main Flow 1.

Alternate Flow B:

6. The system parses the file and finds that there is an invalid operation.
7. The System displays a message informing the user of bad operation. Return to Main Flow 1.

2.3.5. Create Invoice

Description: A sales associate shall select a subset of their mobile warehouse's inventory and create an invoice for the sale of those items.

Main Flow:

1. Sales associate navigates to a Create Invoice page.
2. System displays form with a field for entering customer information, a list of the products in the sales associate's mobile warehouse with areas to specify the sold quantity of each item, and a submit button.
3. Sales associate enters the customer's information, specifies the quantity of each item being sold, and selects submit.
4. System validates that the quantities are not larger than the quantity of the item in stock.
5. System generates invoice based on the price and quantities of the items sold and customer information given and saves the sale record in the database.
6. System updates inventory database to account for the sold items.
7. System displays message indicating that the sale was successfully documented.

Alternate Flow A:

4. System finds that a sale quantity is larger than the inventory quantity and displays a message indicating this error. Return to Main Flow 2.

2.3.6. Search for Invoice (by Sales Associate, Customer, Date)

Description: Any user should search for and review records of previous sales.

Main Flow:

1. User navigates to Sales page.
2. System displays a list of previous sales, sorted by date (most recent first), a search field, a drop-down menu to select search field, and a search button .
3. User selects what field they want to search by, enters a term into the search field, and selects search.
4. System queries the sales database to and displays a list of sales records that match the search terms.
5. User selects a particular search result.
6. System displays invoice and an option to download the invoice as a txt file for that particular sale.

2.3.7. Generate Sales Metrics (by associate between range of dates)

Description: A manager should generate a report detailing what each sales associate sold, given a date range.

Main Flow:

1. Manager navigates to Sales Metrics page.
2. System displays input fields for start date and end date and a generate report button.
3. Manager fills in the date fields and selects the Generate Report button.
4. System queries the sales database and compiles a report of metrics breakdown including total sales and sales per item each associate sold during the time period.
5. System displays a list of associates, their totals, and a sublist of items the quantity of each that the associate sold.

2.3.8. Create a User Account

Description: The administrator shall create any user accounts and the manager should create sale associate accounts, allowing unique sales associates to access their inventory stores.

Main Flow (Admin):

1. Administrator log into the postgresql database as an administrative user.
2. Administrator creates an inventory manager role with a preselected list of privileges allowing them to access and modify the 'users' table as well as all other tables necessary to facilitate the manager's functions.

Main Flow (Manager):

1. Manager logs into their account.
2. System compares credentials to SQL users table.
3. Manager navigates to manage users management page.
4. System queries users database and renders a table with users and check boxes
5. Manager selects 'Create User.'
6. System renders web form containing all the necessary fields required to insert a new user.
7. Manager finishes form and selects submit (reset simply clears form).

8. System receives post request from manager user, scrubs input and updates SQL users table.
9. System generates password and renders success page with username and password.

Alternate Flow (Manager):

9. System generates non-generic error and renders error page.

2.3.9. Update a User Account

Description: The manager should update the various attributes associated with any a user account of lesser privilege level.

1. Manager logs into their account.
2. System compares credentials to SQL user tables.
3. Manager navigates to user page.
4. System queries users database and renders a table with users and check boxes.
5. Manager selects user checkbox and clicks 'Update.'
6. System queries database for user and attributes.
7. System renders a web form preloaded with user attributes.
8. System logs date, time, manager id, and user's details.
9. Manager refills desired user attribute fields and selects 'Submit.'
10. System receives post request from manager user, scrubs input and updates SQL users table and appends changes to log.
11. System generates success page.

Alternate Flow:

10. System generates non-generic error and renders error page

2.3.10. Disable a User Account

Description: The manager should modify the access status to the system of any user account of lesser privilege level.

1. Manager logs into their account
2. System compares credentials to SQL user tables.
3. Manager navigates to user page.
4. System queries users database and and renders a table with users and check boxes.
5. Manager selects user checkbox and clicks 'Update.'
6. System queries database for user and attributes.
7. System renders a web form preloaded with user attributes.
8. System logs date, time, manager id, and user's details.
9. Manager toggles Disable/Enable radio button and selects submit.
10. System receives post request from manager user, scrubs input and updates SQL users table and appends changes to log.
11. System generates success page.

Alternate Flow:

10. System generates non-generic error and renders error page.

2.3.11. Update Product Attributes

Description: Inventory manager should access list of products queried from database and modify attributes individually.

Main Flow:

1. Manager logs into their account.
2. System compares credentials to SQL user tables.
3. Manager navigates to inventory management page.
4. System renders management page with link to modify item.
5. Manager click modify item link.
6. System renders page with part description field and submit button.
7. Manager enters item key terms (part number, name, etc.)
8. System queries database and renders table with possible matches.
9. Manager selects item and clicks on 'Modify.'
10. System renders item page, a preloaded web form with item attributes.
11. Manager modifies data and clicks submit.
12. System receives post request from manager user, scrubs input and updates SQL users table and appends changes to log.

Alternate Flow:

12. System generates non-generic error and renders error page.

2.4 Constraints

2.4.1 Access & connectivity

The software's classification as a web application is subject to some innate access and connectivity constraints, chiefly an internet connection. The entire system's operations are based upon a connection with high integrity. Since the application is hosted in the cloud, the very nature of its existence is dependent upon a connection. Furthermore the connection must continuously reliable because almost every function the application performs fetches data from a database over the internet.

2.4.2 Virtualized Components

Performance in this application is theoretically constrained by the capacities of the cloud provider, Google. The application itself and database will be hosted in the cloud. The entire system could be configured to grow its resources (computing/database capacities) if the main system detects it necessary, there still technically exists an upper threshold to these resources, but more realistically, however unlikely Google Cloud services are still vulnerable to physical as well as virtual complications.

2.4.3 Scalability

This system has virtually unlimited depth scalability. The database can be copied and moved into a larger system without many problems. However its breadth, that is appending new attributes to any existing table, is constrained by the database schema and is intentionally so to limit failures and increase reliability of the entire system.

Section 3: Project Schedule

3.1 Approach

The development team has constructed the project schedule by subdividing the requirements into milestones and deliverables. These divisions of smallest shippable code can be packaged to the production environment and presented to the stakeholders for testing and feedback. Once development and production environments have been established, the developers may begin working on the Product Import functionality and Search features. Since these features will be the most used, and other facets of the site require their functionality, their completion should come first. Then, there will be time for subsequent feedback, while other parts of the application are built.

Once the developers have created the core importing and searching functionalities, they may begin dependent tasks, including invoice generation and user account and session functionality for the application. After these milestones have been delivered, the medium priority requirements are available, such as Customer Management and Metrics pages. Once medium priority requirements have been addressed, the lowest priorities, including Invoice Search and more complex user account support may be implemented.

3.2 Milestones and Deliverables

The milestones and deliverables for the project are listed below.

1. Milestone: Initialize development and production environments.
2. Milestone: Define database schema.
3. Deliverable: Create the Import page.
 - a. Create file template to be read.
 - b. Create web page to process and display results.
 - c. Create Python script to read and validate input.
 - d. Create Python script to perform database operations.
4. Deliverable: Create Product Search page.
 - a. Create web page to search for and display results.
 - b. Create Python script to read and validate input.
 - c. Create Python script to perform database operations.
5. Deliverable: Create the Invoice page.
 - a. Create web page to process and save results.
 - b. Create Python script to read and validate input.
 - c. Create Python script to perform database operations.
 - d. Create downloadable text file.
6. Deliverable: Create basic user accounts.

- a. Implement user roles for session management in Python.
 - b. Login and logout functionality.
 - c. Create account generation for Administrator roles.
7. Deliverable: Create Customer Management page.
 - a. Create web page to create and update customers.
 - b. Create Python script to read and validate input.
 - c. Create Python script to perform database operations.
8. Deliverable: Create Metrics subsystem.
 - a. Create web page to search for and display results.
 - b. Create Python script to read and validate input.
 - c. Create Python script to perform database operations.
9. Deliverable: Create Invoice Search page.
 - a. Create web page to search for and display results.
 - b. Create Python script to read and validate input.
 - c. Create Python script to perform database operations.
10. Deliverable: Implement advanced user account management.
 - a. Update warehouse information.
 - b. Update user information.

3.3 Work Breakdown Structure

Work breakdown structure has been separated into six major milestones and several deliverables. Not all milestones are deliverable. Figure 3-1 provides a visual breakdown of this project.

Milestone 1: Project Management (Organization)

One week

The first milestone is essentially the outlined plan document and collection of notes necessary to begin development. The amount of work, that is the sort which produces tangible results, is very low. This is not to say that the collection of tasks under this milestone are insignificant, in fact it is the opposite. This milestone sets the tone and pace for the following milestones and deliverables. While the tangibles are a small outline for a document and a collection of notes this milestone is characterized by its frequent meetings and discussions. Several meetings between the development team and one with the client are needed to fine tune the requirements to ensure the program would be functional as well as it's delivery date be realistic. This fine tuning was not only targeting the bells and whistles but was more interested in consolidating the design to ultimately create a lightweight yet strong final product. After the initial meetings the development team will meet one final time and re-evaluate the outline before moving on to the next phase.

Deliverables

- Team meeting reports
- Project Plan outline

Milestone 2: Pre-development (Plan Document)**One week - 10 days**

The second milestone will focus on filling out the project plan outline designed in the previous milestone. Here the formalization of the scope will be written. Additionally the team will agree to the vocabulary/terminology to avoid any confusion moving forward. The bulk of this milestone resides in the completion of the requirement specifications. These specifications will outline the must have functionalities of the program, thereby dictating the bulk of the programming/design work. To aid in accomplishing this use cases are written up and discussed among the development team. Finally, now that the design and requirements have been outlined and described a formal schedule can be draw up.

Deliverables

- Team meeting reports
- Project Plan Document
 - Work breakdown structure
 - Gannt chart
 - Task dependency table

Milestone 3: Design**3 days**

The third milestone is estimated to have a very short schedule. This milestone will focus on the programs design both aesthetically and programmatically. Normally this milestone would demand more time however as per our client request our front end visuals are to be kept to a minimum. Additionally, during the previous two phases, unnecessary complications will be removed from the program streamlining the design process. During this phase a user interface will be wireframed, the database structure finalized, and some short A-B testing will be performed.

Deliverables

- Team meeting reports
- Final wireframe front end
- A-B testing reports
- Database schema

Milestone 4: Development**4 - 5 weeks**

The largest of all milestones also includes the most deliverables. As the milestones name implies the fourth milestone is concentrated with the actual code and implementation. The first step is to set up an agreed upon development environment. Once this and a git repository is set up the work can begin. Major sub phases will include front end, back end, and clean up.

Deliverables

- Team meeting reports
- Functional sections tied in with separate pages
 - Log in
 - Manager
 - Associate
 - Load text file and display inventory
 - Make sale / create invoice

Milestone 5: Testing**1 - 7 days**

The fifth milestone is the most uncertain in time of all the other milestones as the testing might reveal no bugs, minor bugs, or major flaws. In the case that no or only minor flaws are found during testing their fixes might be able to be accomplished on the spot without very much time investment, however if there are more drastic bugs found the entire team might have to go deeper testing to pinpoint the problem and ultimately rectify it.

Deliverables

- Team meeting reports
- Testing report

Milestone 6: Delivery**1 days**

The last milestone is concentrated on giving the final demonstration to the client, walking the client through setup and configuration options if any exist and answering any other questions.

Deliverables

- Software product

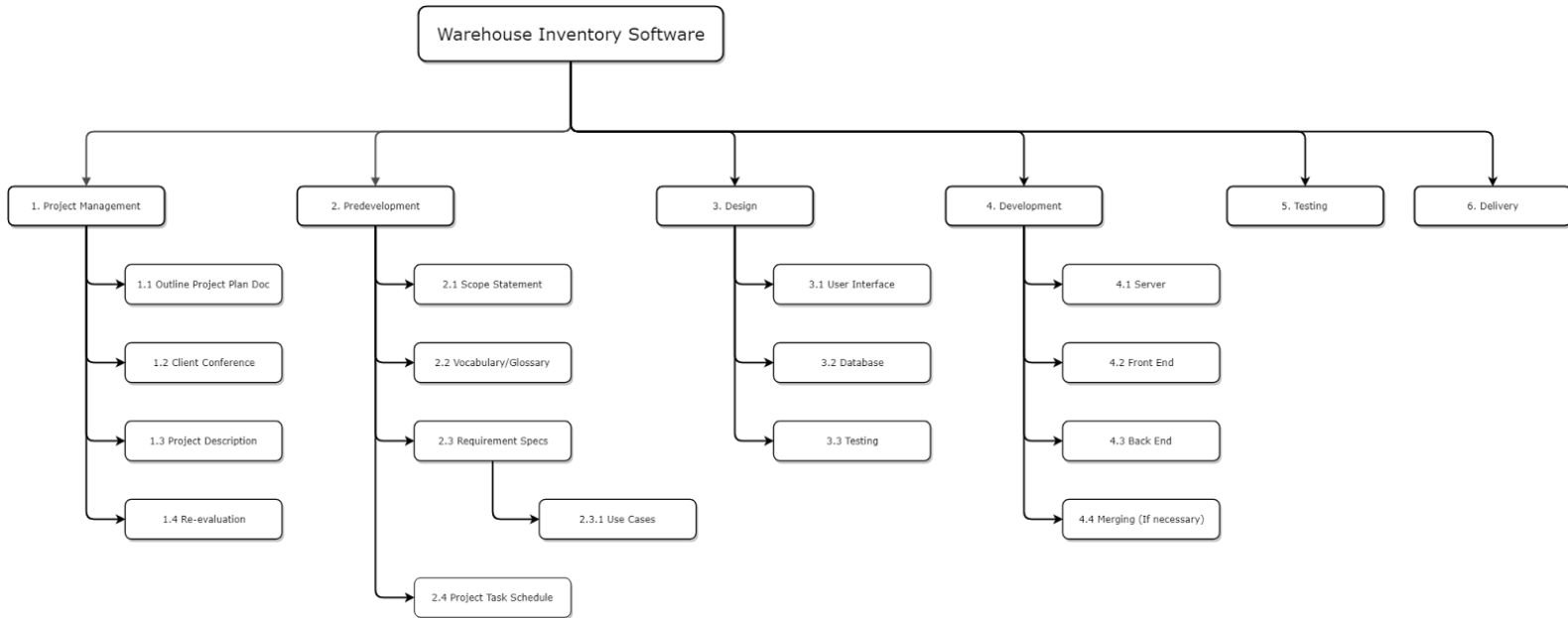


Figure 3-1. Work Breakdown Diagram

3.4 Gantt Chart and Task Dependency Table

The Gantt Chart (Figure 3-2) and Task Dependency Table (Figure 3-3) give a high-level overview of the project schedule and describe which tasks may be worked on concurrently. The plan begins with describing how data will be stored and organized within the system as well as creating environments for development and collaboration. Following those critical tasks, we must create subsystems for inputting and processing data in accordance with the structure we previously defined. As a final general phase of the project, we will create the parts of the system responsible for displaying information related to data already stored in the system such as generation of sales metrics, invoice records, product search results.

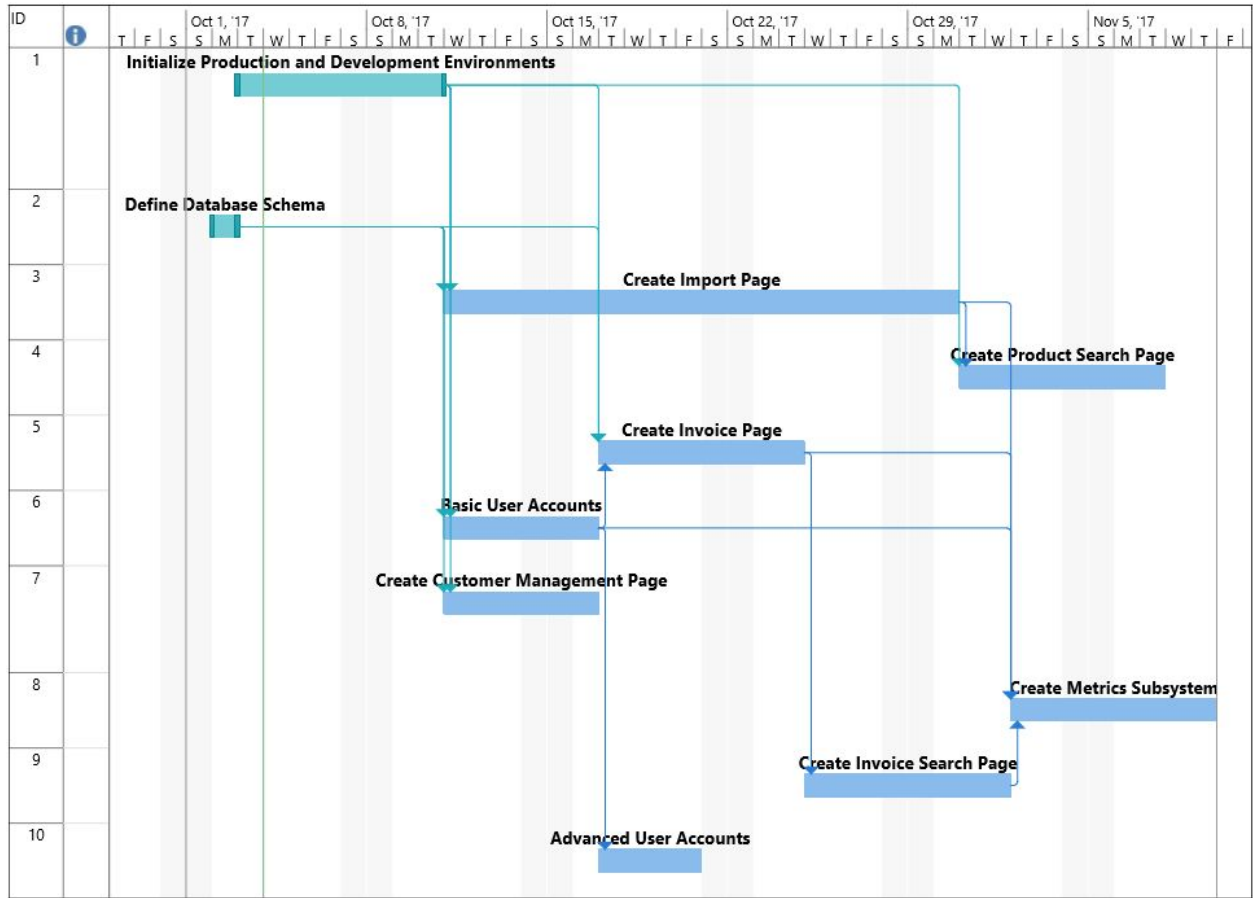


Figure 3-2. Gantt Chart

	A	B	C	D	E
1	Task Name ▼	Duration ▼	Start ▼	Finish ▼	Dependencies ▼
2	Initialize Production and Development Environments	6 days	Tue 10/3/17	Tue 10/10/17	
3	Define Database Schema	1 day	Mon 10/2/17	Mon 10/2/17	
4	Create Import Page	14 days	Wed 10/11/17	Mon 10/30/17	1,2
5	Create Product Search Page	6 days	Tue 10/31/17	Tue 11/7/17	3,1
6	Create Invoice Page	6 days	Tue 10/17/17	Tue 10/24/17	6,2,1
7	Basic User Accounts	4 days	Wed 10/11/17	Mon 10/16/17	2,1
8	Create Customer Management Page	4 days	Wed 10/11/17	Mon 10/16/17	2,1
9	Create Metrics Subsystem	6 days	Thu 11/2/17	Thu 11/9/17	3,5,9,6
10	Create Invoice Search Page	6 days	Wed 10/25/17	Wed 11/1/17	5
11	Advanced User Accounts	4 days	Tue 10/17/17	Fri 10/20/17	6

Figure 3-3. Task Dependency Table

Appendix A. Glossary

This appendix contains a list of abbreviations used in this document.

CPSC	Computer Science
CSV	Comma Separated Values
WMS	Warehouse Management System

Appendix B. System Documents

B.1 Product Import

The table below lists the columns and data types associated with a product import, which updates the quantities and locations tracked in the system.

Column	Data Type	Notes
From Warehouse	Integer	Should match a unique system number identifying the warehouse. If entering the main warehouse as new stock, this value is 0.
To Warehouse	Integer	Should match a unique system number identifying the warehouse.
Product Number	Integer	If new, this value is 0. If updating, number matches the product number in the system.
Quantity	Integer	The amount of product being moved from one warehouse to another.
Price	Double	The price per unit. This field is read only if the product is new stock entering the main warehouse (From Warehouse value of 0.)
Description	String	The description of the product. This field is read only if the product is new stock entering the main warehouse (From Warehouse value of 0.)

B.2 Invoice Example

Below captures the text file created from the system when a sale is made.

Business Name
123 Example Dr.
Ashland, VA 23005
<phone number>
<email>

To:
Business Name
123 Example Dr.
Ashland, VA 23005
Cc. Manager Name

<BG Distributive Group Company Name> Invoice

Invoice Number: ##
Date: mm-dd-yyyy

<u>Product Name</u>	<u>Qty</u>	<u>Price</u>	<u>Product Number</u>	<u>Total Amount</u>
Test 1	1	\$56.00	10100	\$56.00
Test 2	3	\$15.00	16534	\$45.00
Total				\$101

Appendix C. Document Contributions

This appendix lists each group member's document contributions.

- Samantha Miller
 - Wrote the Section 1: Introduction.
 - Wrote Section 2.2 User Characteristics and requirements 2.3.1 through 2.3.3.
 - Wrote Section 3.1 on the project schedule's approach. Discussed, cowrote, and styled Section 3.2, Milestones and Deliverables.
 - Created System Documents B.1 and B.2.
- Jacques Troussard
 - Wrote requirements 2.3.7 through 2.3.10.
 - Wrote Section 2.4 Constraints.
 - Discussed and cowrote Section 3.2, Milestones and Deliverables.
 - Created Work Breakdown graph.
 - Wrote breakdown explanation.
- Taylor Dohmen
 - Wrote section 2.1
 - Wrote requirements 2.3.4 through 2.3.9
 - Discussed and cowrote Section 3.2, Milestones and Deliverables.
 - Created Gantt Chart and Task Dependency Table and associated explanation in section 3.4