CHANGE PAGE

Revision	Date	Editor	Description of Change
1	10/26/19	Brandon	Project Plan Document Created.
2	11/01/19	Daryle	Changed document format.
3	11/08/19	Brandon, Jella, Tom, Daryle	Added User Guide and Test Plan.
4	11/10/19	Daryle Urrea	Add revision form, Approval form, moved reference table to last page, updated table of contents.
5	11/15/19	Brandon, Jella, Tom, Daryle	Added Design Document.
6	11/20/19	Brandon, Jella, Tom, Daryle	Added Phase 1 Source Code and Testing Document.
7	11/21/19	Daryle	Changed R#12 Test requirement from “delete” to “disable” due to possible data constraints.
8	11/28/19	Brandon, Jella, Tom, Daryle	Added Phase 2 Source Code and Testing Document.
9	11/28/19	Brandon, Daryle	Changed words from Test requirement #1 from Welcome Page to Index Page. Changed words from #14 and #18 to match the code’s functionality.
10	11/29/19	Brandon, Daryle	Added retest as test #3. Test Cases numbering corrected (two #26 in prior document).
11	12/06/19	Tom, Daryle	Added phase 3 document and document review
12	12/12/19	Daryle	Added alternative design
13	12/13/19	Brandon, Jella, Tom	Reviewed and re-organized document layout. Added Summary of contributions
14	12/14/19	Brandon	Added conclusion. Finalized document for submission

Table of Contents

List of Figures

OVERVIEW

This SDP documentation will provide an overview of the overall planning and objectives of the development of the LogisTix Inventory Management System (IMS) software. In addition, this document will list timelines, illustrations, challenges, and expectations that will amplify the details of the development process.

SUMMARY OF INDIVIDUAL CONTRIBUTION

Jella An: A valuable member of team LogisTix as she contributed in multiple areas of the software development process. She led the front-end coding, developed the UI/ UX of the application, actively participated in weekly meetings and revisions of the software document and helped in the maintenance and monitoring of our GitHub repository.

Brandon Elliott: As the project lead, he is the core member of the team. His contribution is multi-dimension as he actively organizes the team's tasks and effectively balanced the workload. He led every week's meetings and revisions. He also helped in testing to assure a high-quality software is delivered.

Tommy Cress: He is an essential team member as he led the creation of test plan and the testing of the software. He also contributed by actively participating in weekly meetings and revisions. As the test lead, he was the main filter that ensured the software is fully tested.

Daryle Urrea: A valuable member of team LogisTix. As lead developer, he led the crafting of the backend of the software and quickly react to fix issues whenever a bug has been discovered. He is also the architect of the AWS cloud for deployment and actively participated in weekly meetings and revisions.

LOGISTIX

"YOU NAME IT, WE GOT IT!"

UMGC | CMSC 495 7982

CHAPTER 1 SOFTWARE PROJECT PLAN

Brandon Elliott | Jella An | Tom Cress | Daryle Urrea

CHAPTER 1 SOFTWARE PROJECT PLAN

1.1 PROJECT SUMMARY

1.1.1 Purpose, Scope, and Objectives

The purpose of the LogisTix IMS is to provide an organizational application to assist in an inventory stocking system. This system will give the ability to add new items to the database, track their quantity, track out of stock items, and other features in increasing the company's inventory system efficiency. The system will also have a user management system that is divided into 3 roles: admin, member, and guest. The number of functions a user can perform within the system will depend on the role.

1.1.2 Assumptions

It is assumed that the user has the basic knowledge of how to utilize a web-based application user interface and will have the ability to access the internet.

1.1.3 Project Deliverables

- LogisTix - Inventory Management System (Web-based Application)
- User Manual

1.1.4 Requirement Specification

1. Accessible through the internet.
2. Will have a login interface for registered member.
3. New Items are displayed on the index (last 8).
4. Contact form for guests to contact Logistix.
5. Once logged in, the user is authenticated and sent to the dashboard.
6. The dashboard will state welcome <Name> <time>.
7. The dashboard shows an overview of the inventory system.
8. The dashboard shows how many items are out of stock.
9. The dashboard will show total stocks overall.
10. There is a limit of 30 mins to which user can idle, the user will have to reauthenticate.
11. Users can click on a "logout" link to logout and will be sent to the index.
12. Admin can add/ disable regular members.
13. Admin can create, view, update, delete products.
14. Members can create, view, update products.
15. Guests can enter the site.
16. Guests can view the inventory.
17. Products will have Name, Manufacturers Name, Description, Qty, Image and status if out-of-stock.
18. Admin will need to change password first-time login
19. The site will be responsive that can be viewed in multiple media. Mobile phone, tablet, laptops, and pc.

1.1.5 System Specification

1. Cloud: AWS Cloud EC2 Instance
2. Remote Connection: Remote Desktop Client/ App
3. Localhost Server: XAMPP, WAMP, MAMP [Apache, MySql]
4. Programming Language: PHP, HTML, CSS, JavaScript
5. Local Code Editor: NetBeans, Eclipse, SublimeText
6. Documentation: Microsoft Word, Excel, Adobe Acrobat
7. Image Edit: Adobe Photoshop/ Illustrator [Atleast one member]
8. PC/ Mac Operating System: Windows 10, MacOS
9. Processor: 2.5 GHz Dual Core (Faster is better)
10. RAM: 4 GB (2018 At least 8 GB)
11. HDD: 50GB

1.2 DEFINITIONS

- GitHub: Software repository and collaboration tool.
- PHP (Hypertext Preprocessor): An open-source programming language that is widely used in web development.
- NetBeans: Is an integrated development environment (IDE) that is used in building software for multiple platform like web or desktop applications.
- Amazon Web Service (AWS): A cloud platform web service that offers flexible, reliable, scalable, easy-to-use and cost-effective cloud computing solutions.

1.3 PROJECT ORGANIZATION

The organization of the project is categorized into two divisions. The external division is the requirement and the internal division is the development team that will complete the project. As

shown by the image below. The requirement is acquired by the project lead and the project lead distributes the job requirements that accords to each member role.

1.3.1 Project Structure

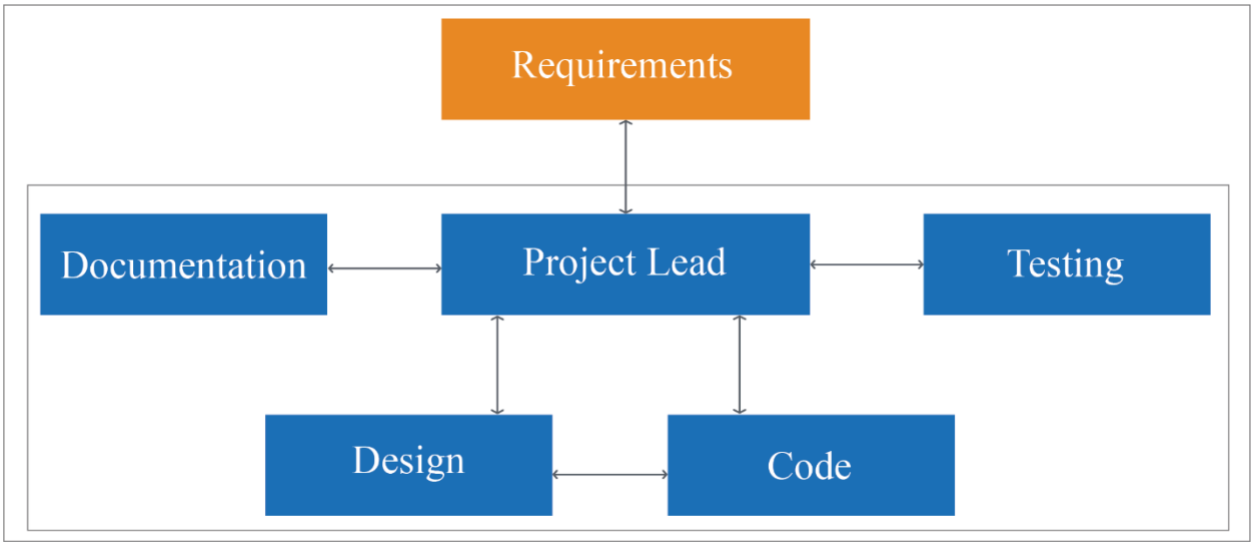

Figure 1-1: Team 1 Project Team Structure

1.3.2 Main Roles and Responsibilities

1.3.2.1 Project Lead [Brandon Elliott]

The project lead coordinates with all team members to ensure the deliverable is completed on time and in accordance with all requirements. Furthermore, the project lead organizes all group chat sessions, phase reports, and student-professor communications and finalizes documentation for system and requirements specifications. The project lead will also communicate clearly and concisely for any needs or concerns to the Professor and vice versa.

1.3.2.2 Design Lead [Jella An]

Responsible for the creative works within the software application. In charge of interface design that brings all the together. Assists with user guides and project plans to include formatting and

templating. Collaborates with the project lead to establish interconnectivity between the software interface and the inventory management system functions.

1.3.2.3 Coding Lead [Daryle Urrea]

Works to bring all of the web applications together to create a fully working application that meets requirements. Additionally, assists other developers to ensure codes used are within standard. Serves as liaison between the development team and other project workgroups.

1.3.2.4 Testing Lead [Tom Cress]

Creates, executes, and documents testing results of the application in accordance with the IEEE Standard for Software Test Documentation.

1.4 MANAGERIAL PROCESS PLAN

The managerial process plan emphasizes the work schedule and task management needed to meet the agreed-upon project goals.

1.4.1 Work Plan

The following section details both the work breakdown and project schedule for team 1.

1.4.1.1 Work Activities

The following table lists the breakdown of tasks according to its Software Development Life Cycle task.

Task name	Category	Subtask
Project Plan	Admin	Business analysis
Test Plan	Test	Test Design
Project Design	Design	Software Analysis/ DB Design/ Creative Design

Phase 1 Source	**Code**	**Login/user management/UI/Test**
Phase 2 Source	**Code**	**Inventory/ UI/ Test**
Phase 3 Source	**Code**	**Cloud migration/ Test**
Final Deliverable	**Code**	**Software Build/ Deployment/ Maintenance**

Figure 1-2 Work Breakdown Schedule

1.4.1.2 Schedule Allocation

The figure below illustrates the project schedule in a Gantt chart view. This chart highlights the dependency mapping between tasks. The chart shows the major project milestones along with delivery due dates for each project deliverable.

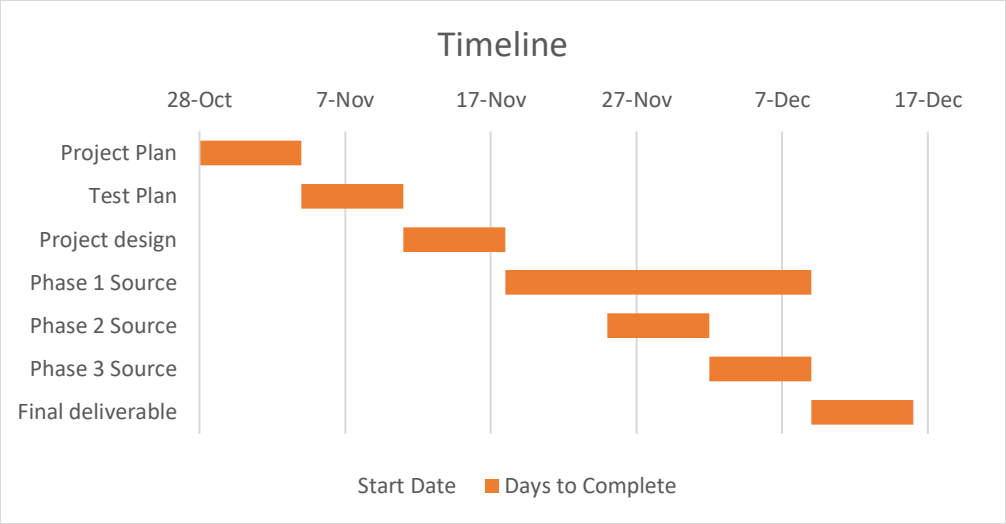

Figure 1-3 Project timeline

1.4.1.3 Milestone

Main Task/ Sub Task	Duration	Start Date	End Date	Task Owner
Project Plan	7	28-Oct-19	3-Nov-19	
Outline Contents & Review	7	28-Oct-19	3-Nov-19	Brandon
Requirement Specification	7	28-Oct-19	3-Nov-19	Jella
System Specification	7	28-Oct-19	3-Nov-19	Tom
Timeline, Illustrations & Formatting	7	28-Oct-19	3-Nov-19	Daryle
User Guide and Test Plan	7	4-Nov-19	10-Nov-19	
User Guide Writing	7	4-Nov-19	10-Nov-19	Daryle & Jella
Test Plan Writing	7	4-Nov-19	10-Nov-19	Tom & Brandon
User Guide Review	3	8-Nov-19	10-Nov-19	Tom & Brandon
Test Plan Review	3	8-Nov-19	10-Nov-19	Daryle & Jella
Design	7	11-Nov-19	17-Nov-19	
Data Structures	7	11-Nov-19	17-Nov-19	Tom
Back End: Classes, Functions	7	11-Nov-19	17-Nov-19	Daryle
Front End I/O formats, UI/UX	7	11-Nov-19	17-Nov-19	Jella
Documentation, Illustration, Assisting all Tasks	7	13-Nov-19	17-Nov-19	Brandon
Phase 1 Source	21	18-Nov-19	8-Dec-19	
Write code	7	18-Nov-19	24-Nov-19	Daryle & Jella
Test Code	7	18-Nov-19	24-Nov-19	Tom & Brandon

Phase 2 Source	7	25-Nov-19	1-Dec-19	
Write code	7	25-Nov-19	1-Dec-19	Daryle & Jella
Test Code	7	25-Nov-19	1-Dec-19	Tom & Brandon
Phase 3 Source	7	2-Dec-19	8-Dec-19	
Write code	7	2-Dec-19	8-Dec-19	Daryle & Jella
Test Code	7	2-Dec-19	8-Dec-19	Tom & Brandon
Final Deliverable/ Deployment	7	9-Dec-19	15-Dec-19	
Finalizing Code	7	9-Dec-19	14-Dec-19	Daryle & Jella
Testing Code	7	9-Dec-19	15-Dec-19	Tom & Brandon

Figure 1-4 Project Milestone

1.4.2 Risk Management Plan

This sub-clause of the SDP addresses the risks involved with the project. Its purpose is to identify, analyze, and prioritize risk factors. In addition, this sub-clause will layout specific risk mitigation plans. The probability of the risk occurring is graded on a 1 -5 scale with 1 being the lowest and 5 the highest probability of the risk occurring.

Risk ID	Description	Impact	Probability
1	All codes are not completed by the deadline	Inability to publish and deliver the software on-time	1
2	Back-end functions do not integrate smoothly with front-end	Unable to use the application or become vulnerable to exploits	2
3	Testing encounters problems	Slow down the process or inability to deliver on-time	2

Figure 1-5 Risk Management Matrix

In order to mitigate the risk of the project not meeting deadline group members will utilize on-line forums for communication in addition to a bi-weekly chat session. Secondly, the risk of the application not properly migrating smoothly into the cloud is mitigated by the use of GitHub for version control, testing the cloud environment as often. Software modules can be checked out of GitHub and worked on by all team members allowing for seamless software integration. Lastly, testing will begin early, and testers will be in constant communication with developers to quickly solve any issues which may arise.

1.5 TECHNICAL PROCESS PLAN

This sub-clause of the SDP will be used to explain the process model being utilized as well as the tools, techniques, and methods to be used in the development of the software.

1.5.1 Process Model

The process model that will be used by the group will be a traditional waterfall model for project management purposes. The developers' multi-skillset gives them the ability to utilize either a standard waterfall model or an Agile methodology. The use of these two methodologies will be flexible to the development of the software while maintaining an achievable set timeline for other tasks within the project.

1.5.2 Methods, Tools, and Techniques

In order to provide for a uniform standard across the Group 1 project team the following tools will be utilized during project development.

- Language - PHP, HTML, CSS, JavaScript
- Platform - LAMP | XAMMP | MAMP (Any of the options)
- Code Editor – Netbeans, Sublime Text, Notepad ++
- Code repository – GitHub
- Documentation - Microsoft Office | Adobe Acrobat
- Deployment – AWS EC2 Instance

Coding will be in PHP 7 + in order to utilize the up to date features of the language. GitHub will serve as the main repository for both version control and accessibility. All documentation will be done in Microsoft Office and Adobe Acrobat. And finally, deployment will be using cloud technology in AWS EC2 instance.

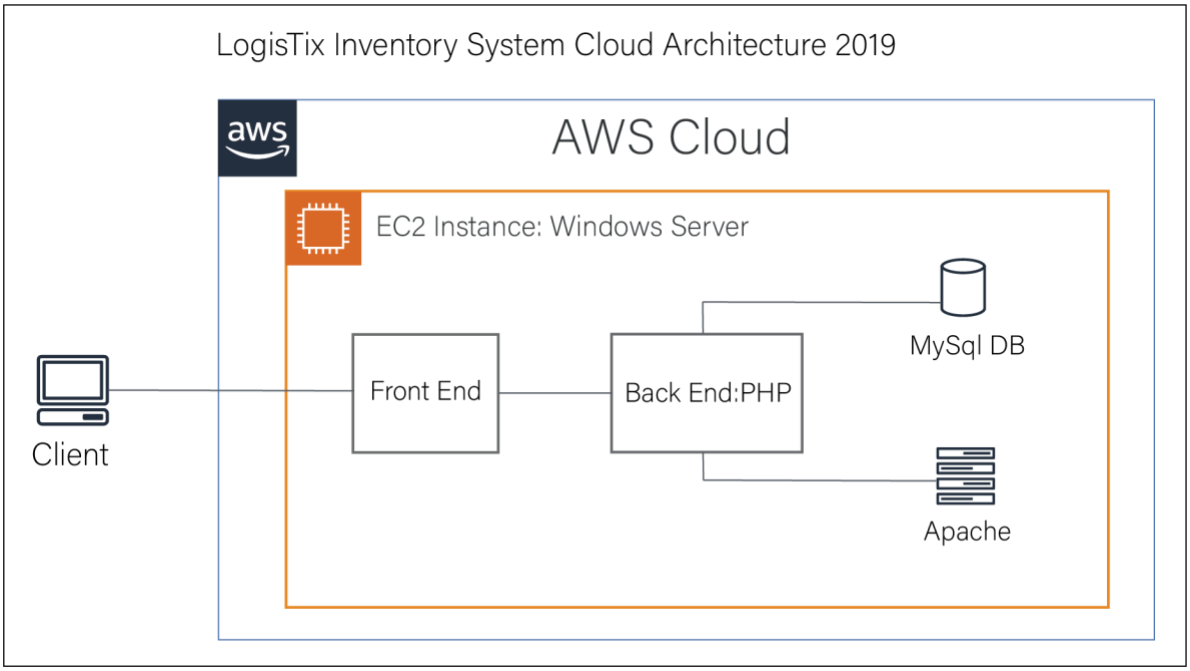

Figure 1-6 LogisTix Inventory System Cloud Architecture

1.6 APPROVALS

Name	Date
Brandon Elliott	
Jella An	
Tom Cress	
Daryle Urrea	

1.7 CONTRIBUTIONS

All coordination, reviews, and investigations are communicated to the team through Trello or Discord.

Brandon: Project lead, Test

- Reviewed and submitted documentation.
- Coordinated team communication.
- Worked with team leads to establish project requirements.
- Expanded project requirement use cases and documentation.
- Established/continued meeting/coordination timeline.

Jella: Developer, Design

- Reviewed programming languages to be used.
- Investigated alternatives to current development path forward.
- Investigated types of graphics to be used.
- Reviewed demo to determine design/layout of pages.
- Reviewed and contributed in the Project Plan document.

Tom: Test, Image Editing

- Reviewed use cases.
- Establishing formal test plan.
- Communicate with developers on images types and placement.
- Reviewed and contributed in the Project Plan document.

Daryle: Developer, Design

- Initial design.

- Program Demo/mock-up.
- Established development timeline for deliverables.
- Tied requirements to deliverables.
- Organized original meeting for establishing team timeline to assist in making future coordination organized.
- Reviewed and contributed in the Project Plan document.

LOGISTIX

"YOU NAME IT, WE GOT IT!"

UMGC | CMSC 495 7982

CHAPTER 2 USER GUIDE

Brandon Elliott | Jella An | Tom Cress | Daryle Urrea

CHAPTER 2 USER GUIDE

2.1 INTRODUCTION

This user guide will provide basic instruction on how to use LogisTix Inventory Management System. There are three types of users that can use the application. The admin, member, and guest. Depending on what type of access is used, the feature that is accessible to the user will vary.

2.2 SKILL REQUIRED

- Basic knowledge of internet navigation

2.3 DEFINITION

- IMS: Inventory Management System
- User Interface: The point where user interact with the application
- URL Address: Uniform Resource Locator is also known as web address which reference to specify the internet location of a site.

2.4 USING THE APPLICATION

2.4.1 Getting to the site

Upon release of the application, the users will be provided with official URL address (i.e. logistix.com). Once the user entered the correct URL, the browser will be sent to the index of the site.

2.4.2 Logging in (Click the login link)

Admin: The initial log of an admin will require the admin to change the password. By default, the username is Admin and password is Admin.

Member: A member will be created by the admin and will be assigned username and password. The member will have an option to change the password later.

Guest: Guest users will be able to login the system by clicking the Guest button on the index.

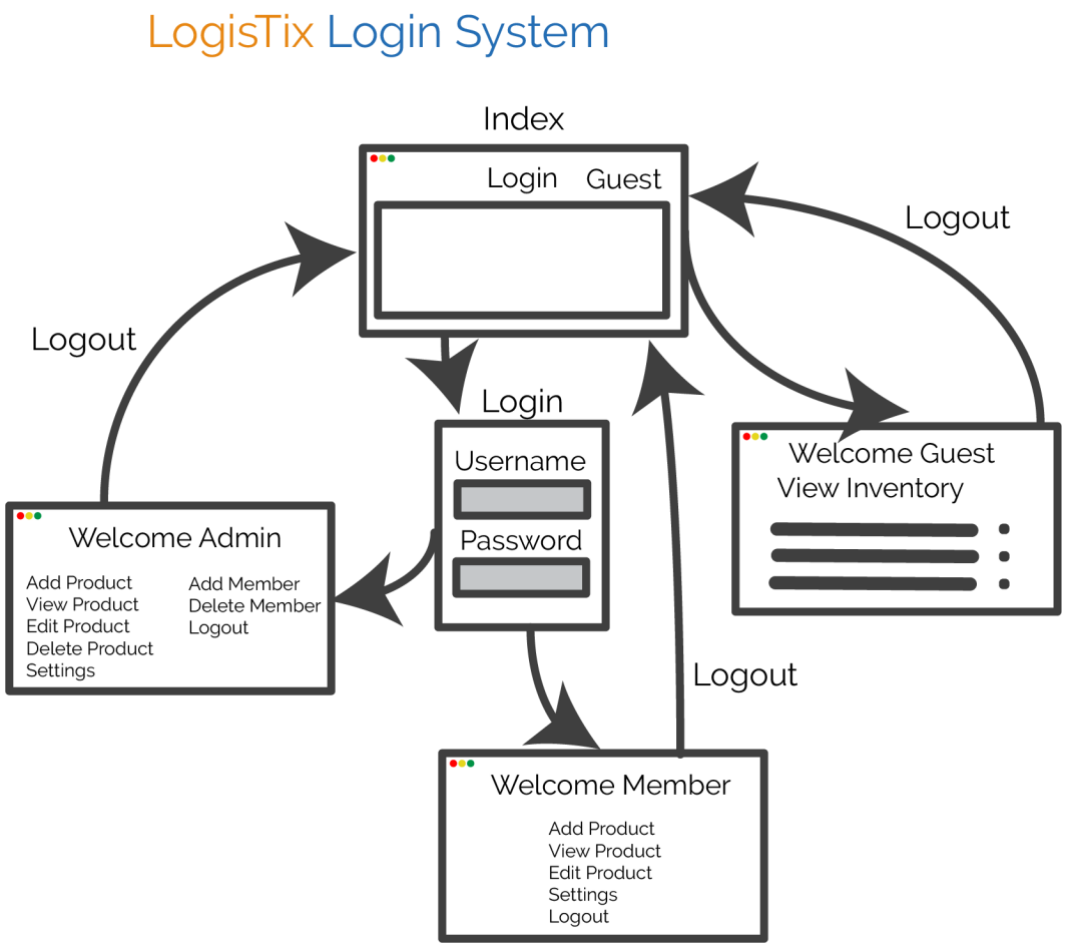

Figure 2-1 LogisTix Login System Storyboard

2.5 NAVIGATING THE DASHBOARD (Admin)

An admin access will have the highest type of access within the organization. Once logged in, the admin will be sent to the dashboard. The admin will be greeted, and the access type will be on display. In the dashboard, an overview of the inventory system is displayed.

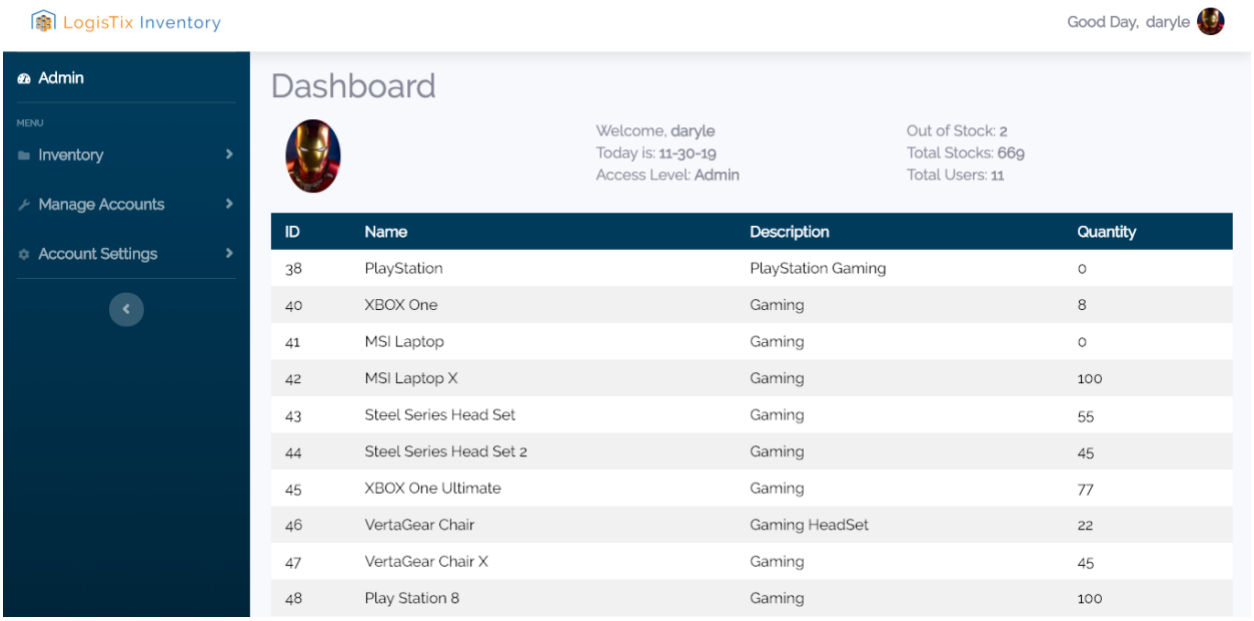

Figure 2-2 Admin Dashboard

2.5.1 Summary View

The summary view or the overview will show any inventory that is out of stock, total number of stocks, number of users, and summary list of inventories. There will also be a sidebar menu for managing products, managing members, and managing personal settings.

2.5.2 Inventory

2.5.2.1 Adding Products

To add a product on the inventory, an admin can either click on the add product on the dashboard of click on the inventory on the side navigation and then click the add product button.

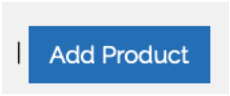

Figure 2-3 Add Product Button

Once the add product button is clicked, the user will then be prompted to fill in information fields for data storage of the product to be added. Fields are: Name of the product, Manufacturer, Description, Quantity, and Image. After the information field is filled, now the user can click “Add Product”. The user will be sent to the inventory view.

2.5.2.2 Viewing Products List

To view the product list, a user will simply go to inventory list.

2.5.2.3 Updating Products

To update the product, the user will click on the “edit” link while viewing the list of products. Once the “edit” link has been clicked, the user will be sent to a form where the user can edit information about the product.

2.5.2.4 Deleting Products

To delete a product, the user must click on the “delete” link. Only admin access will have the ability to delete a product.

2.5.3 Staff Management

2.5.3.1 Adding a Staff Member

Admins have the ability to add staff members. To add a member, the admin will click “Manage Staff” on the side navigation and then click “Add Staff”. The admin will be prompted to add staff member form to fill information about the new member. The information needed are: First Name, Last Name, Username, Email and Password.

2.5.3.2 Deleting

An admin will have the ability to disable members, in the event of someone leaving the company.

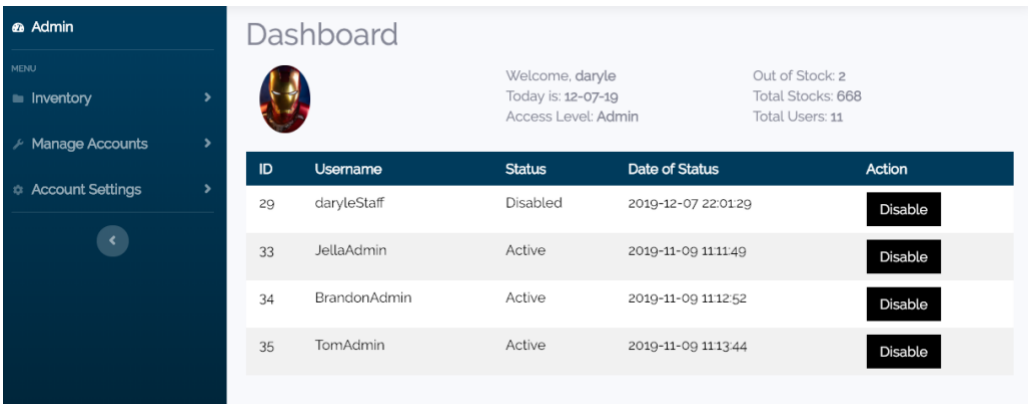

Figure 2-4 Disabling an account

2.5.4 Settings

The settings can be found on the dashboard side navigation. Once clicked, the user will have an option to change password.

2.6 NAVIGATING THE DASHBOARD (Staff)

A staff member will have a lower type of access within the organization. Once logged in, the staff will be sent to the dashboard. The staff will be greeted, and the access type will be on display. In the dashboard, an overview of the inventory system is also displayed.

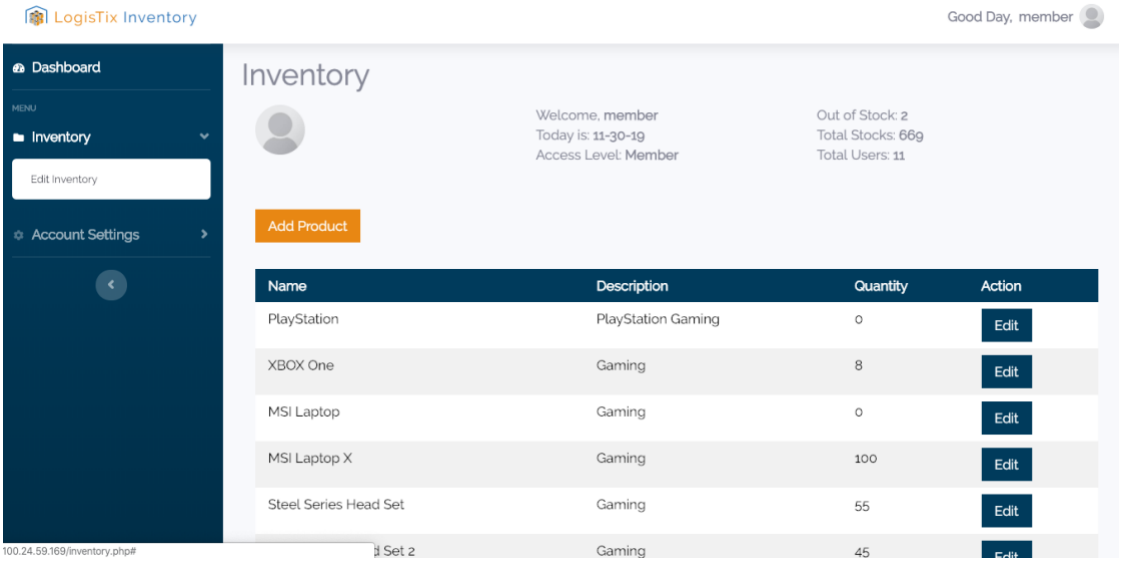

Figure 2-5 Member Dashboard

2.6.1 Summary View

The summary view or the overview will show any inventory that is out of stock, total number of stocks, number of users, and summary list of inventories. There will also be a sidebar menu for managing products and managing personal settings.

2.6.2 Inventory

2.6.2.1 Adding Products

To add a product on the inventory, a staff member can either click on the add product on the dashboard or click on the inventory on the side navigation and then click the add product button.

2.6.2.2 Viewing Products

To view the product list, a user will simply go to inventory list.

2.6.2.3 Updating Products

To update the product, the user will click on the “edit” link while viewing the list of products. Once the “edit” link has been clicked, the user will be sent to a form where the user can edit information about the product.

2.6.3 Settings

The settings can be found on the dashboard side navigation. Once clicked, the user will have an option to change password.

2.7 NAVIGATING THE DASHBOARD (GUEST)

If a visitor wanted to check available inventory, the visitor may click “Continue as Guest” link.

2.7.1 Summary View

The summary view or the overview will show any inventory that is out of stock, total number of stocks, and summary list of inventories. There will also be a sidebar menu with a link for contacting the organization.

2.7.2 Inventory

2.7.2.1 Viewing Products List

To view the product list, a guest is directly sent to the inventory list upon entering the site.

2.8 LOGGING OUT

All users Admin, Staff member and guest will have an ability to logout by clicking the drop-down to see the link “logout” on the top right corner of the page.

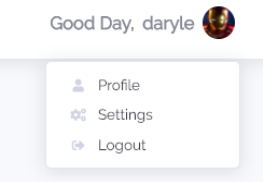

Figure 2-6 Logging out

2.9 CONTRIBUTION REPORT

All coordination, reviews, and investigations are communicated to the team through Trello or Discord.

Brandon: Project lead, Test

- Reviewed and finalized documentation for submission.
- Coordinated team communication.
- Hosted team meeting and provided inputs for this week's tasks.
- Worked with team leads to establish tasks in building User guide documentation.

Jella: Developer, Design

- Contributed in user guide.
- Participated in the meeting and provided inputs for this week's tasks.
- Reviewed user guide documentation for accuracy.

Tom: Test, Image Editing

- Contributed in user guide.
- Participated in the meeting and provided inputs for this week's tasks.
- Reviewed user guide documentation for accuracy.

Daryle: Developer, Design

- Contributed in user guide contents.
- Initiated and formatted user guide documentation.
- Participated in the meeting and provided inputs for this week's tasks.

LOGISTIX

"YOU NAME IT, WE GOT IT!"

UMGC | CMSC 495 7982

CHAPTER 3 TEST PLAN

Brandon Elliott | Jella An | Tom Cress | Daryle Urrea

CHAPTER 3 TEST PLAN

3.1 OBJECTIVES

This master test plan is designed to be the outline for any and all testing of the LogisTix IMS application. This test plan will cover the pages of the web application, the functionality of said pages, and the basic security of the website. Much of the testing will be conducted by the developers.

3.2 TASKS

All test will follow a test matrix that has been created by the entire LogisTix team. The matrix will include the requirements being tested, how they are being tested, the expected result, and if the test passed or failed. All results will be shared between the team member and professors as requested. Any failed tests that may occur during testing will be investigated and the solution documented and implemented. The entire test matrix will then be conducted again to ensure that the solution not only works but did not break any other aspect of the program. This process is repeated until all tests pass and have full documentation as to their pass/ fail history.

3.3 SCOPE

3.3.1 General

For this test plan, we will test many different functions of our application. Each of these tests will cover different pages of our application as each page will perform differently than the other. The following are the pages and the functionality tested.

I. Landing Page

a. Navigating to the landing page

b. Logging in on the landing page

c. Viewing the eight (8) newest items to the inventory

d. Using contact form link to reach out to the ‘management’

3.3.2 Tactics

Any interface being used will be tested by hand, by the developers upon first coding. The next step of testing, also by hand, will be conducted by the person designated to perform the testing.

Adding, removing, and editing users or items will tested through the developed interface. The result will be verified by the dashboard and by the database. Any database changes must be recognized by the database and the UI.

Navigating to the site will be tested by using multiple internet browsers. This includes FireFox, Chrome, and Internet Explorer.

3.4 TESTING STRATEGY

Testing will be conducted by the development team. Each member will be responsible for testing any code they submit. If that code changes the behavior of something already developed by someone else, all involved team members will participate in the user testing.

The lead tester will dictate any extra testing that needs to take place. There may be times when unexpected behavior occurs, and the lead tester must decide who will need to test the behavior of the program depending on each member’s skillset. The behavior of the program will be documented and sent to the tester by a written or verbal report.

Any bugs or issues discovered during the testing will also be documented. The team will use Trello to record the issue. In the event a test fails, those results will be documented in the test output and in Trello for easy team member access.

3.5 HARDWARE AND SOFTWARE REQUIREMENTS

1. Cloud: AWS EC2 Instance
2. Remote Connection: Remote Desktop Client/ App
3. Localhost Server: XAMPP, WAMP, MAMP [Apache, MySql]
4. Programming Language Understanding: PHP, HTML, CSS, JavaScript
5. Local Code Editor: NetBeans, Eclipse, SublimeText (Recommended)
6. Documentation Accessibility: Microsoft Word, Excel, Adobe Acrobat
7. Image Editor: Adobe Photoshop/ Adobe Illustrator
8. PC/Mac Operating System: Windows 10, MacOS
9. Processor: 2.5 GHz Dual Core
10. RAM: 4 GB
11. HDD: 50 GB

3.6 TEST SCHEDULE

Major testing will occur on major due dates. Small scale testing will continue throughout development. See below for the following major test dates.

Test 1	November 24, 2019
Test 2	December 1, 2019
Test 3	December 8, 2019
Test 4	December 15, 2019

Figure 3-1 Major Test Schedule

3.7 FEATURES TO BE TESTED

All testable features are outlines within the test matrix. The matrix will grow more features are added and bugs are discovered. The ability for the site to be accessed from a cell phone, personal computer (laptop or desktop), and tablet will also be tested.

Role Definition						
Role	Add product	View Product	Edit Product	Delete Product	Add Staff member	Delete Staff Member
Admin	X	X	X	X	X	X
Staff Member	X	X	X			
Guest		X				

Figure 3-2 Role Definition

3.8 FEATURES NOT TO BE TESTED

At this time, zoom options and other accessibility options will not be specifically tested. Website building best practices will be used to help account for accessibility.

3.9 TOOLS

GitHub: Code repository and version control.

Trello: Use for scheduling, tasking and bug tracking.

3.10 APPROVALS

Name	Date
Brandon Elliott	
Jella An	
Tom Cress	
Daryle Urrea	

3.11 TEST CASES

Test Case #	Requirement #	Test Description	Expected Result	Actual Result	Pass/Fail
Landing Page					
1	R1	User types in URL; Presses Enter.	URL takes User to Welcome Page.		
2	R2	User chooses Login.	Login Page appears.		
3	R3	The user view the new items in the index.	The 8 most recent items that were added to the inventory are listed are shown in the new item area.		
4	R4	The visitor uses the contact form and message the organization.	The admin will receive a message from the visitor through an email.		
Dashboard and Overview					
5	R5, R6, R7	Admin logs in. username: admin pass: admin	Will be sent to dashboard overview with a welcome message, the username and access level (Admin).		
6	R5, R6, R7	Staff Member logs in. username: membed pass: member	Will be sent to dashboard overview with a welcome message, the username and access level (Member).		
7	R5, R6, R7	User picks continue as Guest	Will be sent to dashboard overview with a welcome message and access level (Guest).		

Test Case #	Requirement #	Test Description	Expected Result	Actual Result	Pass/Fail
Dashboard and Overview					
8	R8, R9	Dashboard Overview Admin and Member	Shows out of stock status, total stocks, number of users and list of stocks		
9	R10	Any user will idle for 30 mins or more.	User is deauthenticate and reauthentication is required.		
10	R11	While logged in, user will click "Logout".	Session destroy and user is sent to index page.		
11	R12	Admin create add a staff member inputs: First name: Member Lastname: One Username: MemberOne Password: Member1	Tester verify the database info: First name: Member Lastname: One Username: MemberOne Password: Member1		
12	R12	Tester using admin account will delete a member	The deleted member is deleted from database and cannot login.		
13	R13, R17	Admin click add product: Name: Product 1 Manufacturer: Product Test Description: Test product description QTY: 99 Image: tester selected image	Product success message is shown. Database will have the new product info Name: Product 1 Manufacturer: Product Test Description: Test product description QTY: 99 Image: tester selected image		

Test Case #	Requirement #	Test Description	Expected Result	Actual Result	Pass/Fail
Dashboard and Overview					
14	R13	Admin click "Inventory" on side navigation.	Admin will be sent on the inventory page and view list of inventories.		
15	R13	Admin click edit on the inventory list and inputs information on the form to update: Name: Product Updated 1 Manufacturer: Product Test Updated Description: Test product Updated description QTY: 15 Image: blank	Product updated success message is shown. Database will have the new product info Name: Product Updated 1 Manufacturer: Product Test Updated Description: Test product Updated description QTY: 15 Image: blank		
16	R13	Admin click on delete while on inventor list page.	The product delete will be erased from the list and will be deleted from the database.		
17	R14, R17	Member click add product and provided input: Name: Member Product 1 Manufacturer: Member Product Test Description: Member Test Product description QTY: 89 Image: tester selected image	Product successfully added message is shown. Database will show product info: Name: Member Product 1 Manufacturer: Member Product Test Description: Member Test Product description QTY: 89 Image: tester selected image		

Test Case #	Requirement #	Test Description	Expected Result	Actual Result	Pass/Fail
Dashboard and Overview					
18	R14	Member click "Inventory" on side navigation.	Member will be sent on the inventory page and view list of inventories.		
19	R14	Member click edit on the inventory list and inputs information on the form to update: Name: Member Product Updated 1 Manufacturer: Member Product Test Updated Description: Member Test product Updated description QTY: 5 Image: blank	Update successful message shows and information of the product in the database is updated as Name: Member Product Updated 1 Manufacturer: Member Product Test Updated Description: Member Test product Updated description QTY: 5 Image: blank		
20	R15, R16	Guest click "Continue as Guest"	Guest is sent to guest page where inventory list can be viewed.		
21	R17	A tester uses admin or member and edit the products quantity to zero.	When a product qty is at zero, it will show on the dashboard that it is out of stock.		
22	R18	A tester will use an admin account for the first time.	The admin will be sent to change password page to change the password.		
23	R19	Tester view the site in a mobile phone.	The display adjusts to mobile view.		
24	R19	The tester views the site in an ipad.	The display adjusts to ipad view.		

Security					
25	Authentication Security	Tester uses wrong login info: username: hacker password: hacker	An invalid entry will show		
26	SQL Injection	SQL Injection: Tester will user sql injection code to login username: " or ""=" pass: " or ""="	An invalid entry will show		
27	Cross Site Scripting	Test will input: on the guest message box: <script>alert("TEST");</script>	The script is nullified by just displaying <script>alert("TEST");</script> as text in the message box.		
28	Password Encryption	Tester will verify that each users passwords format.	Each password should be in an encrypted format. Example format: $2y$10$XIykz7Sj6uqFkmDOop1F1uQC/NOi/VYbvo07InszrMEte0YxTu9su		
29	Path Traversal	Tester uses browser to go to homepage without logging in.	Tester is sent back to index.		
30	Path Traversal	Tester uses browser to go to Inventory page without logging in.	Tester is sent back to index.		
31	Path Traversal	Tester uses browser to go to add member page without logging in.	Tester is sent back to index.		

Results: Please refer to section 5.4 COMPLETED TEST CASES

3.12 CONTRIBUTION REPORT

All coordination, reviews, and investigations are communicated to the team through Trello or Discord.

Brandon: Project lead, Test

- Reviewed and submitted documentation.
- Coordinated team communication.
- Hosted team meeting and provided inputs for this week's tasks.
- Worked with team leads to establish tasks in building Test Plan documentation.
- Initiate the creation of Test Plan.

Jella: Developer, Design

- Contributed in test plan contents.
- Participated in the meeting and provided inputs for this week's tasks.
- Reviewed test plan for accuracy and according to requirements.

Tom: Test, Image Editing

- Contributed in test plan contents.
- Participated in the meeting and provided inputs for this week's tasks.
- Reviewed test plan for accuracy and according to requirements.

Daryle: Developer, Design

- Contributed in test plan contents.
- Formatted test plan documentation.
- Participated in the meeting and provided inputs for this week's tasks.
- Reviewed test guide documentation for accuracy and according to requirements.

LOGISTIX

"YOU NAME IT, WE GOT IT!"

UMGC | CMSC 495 7982

CHAPTER 4 - DESIGN

Brandon Elliott | Jella An | Tom Cress | Daryle Urrea

CHAPTER 4 DESIGN

4. 1 OVERVIEW

The purpose of the LogisTix IMS is to provide an organizational application to assist in inventory stocking system. This document details the design portion of LogisTix IMS software development plan. Design consist of multiple are working and interacting with each other to form a secured, efficient and elegant application for users to experience.

4.2 CLASS AND FUNCTIONS

There are two fundamental classes that are being used in the LogisTix IMS application. The userClass and the Database Class. The user class constructs the user's attributes and it's associated to functions.php. The database class construct login credentials for the MySQL database and it's associated with dbConnect. Functions.php is associated to both dbConnect and userClass. The class diagram below shows the association of functions and classes of the LogisTix IMS application.

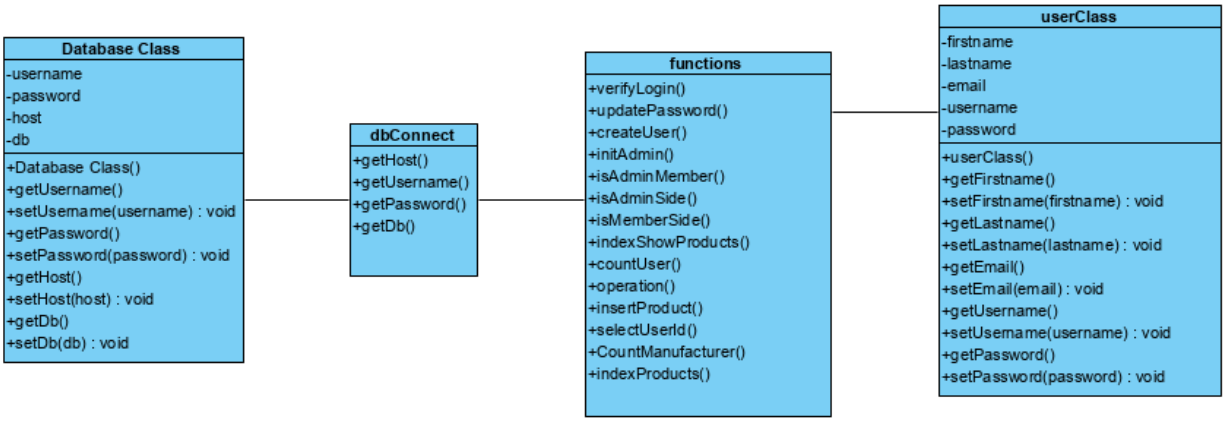

Figure 4-1 Class Diagram

4.3 DATA AND DATA FILES STRUCTURES

LogisTix IMS uses a MySQL database with three tables. The users table stores information about the user's identity and access level. An admin will have more privileges than regular staff and will require a password reset upon first successful login. This table uses integers, varchar, tinyint, and datetime. The next table is the products table and this table stores data about a specific product that is added to the inventory. This table uses integers, varchar, int, and timestamp. The third table is the manufacturer table to store data of the manufacturers. This table uses int and varchar.

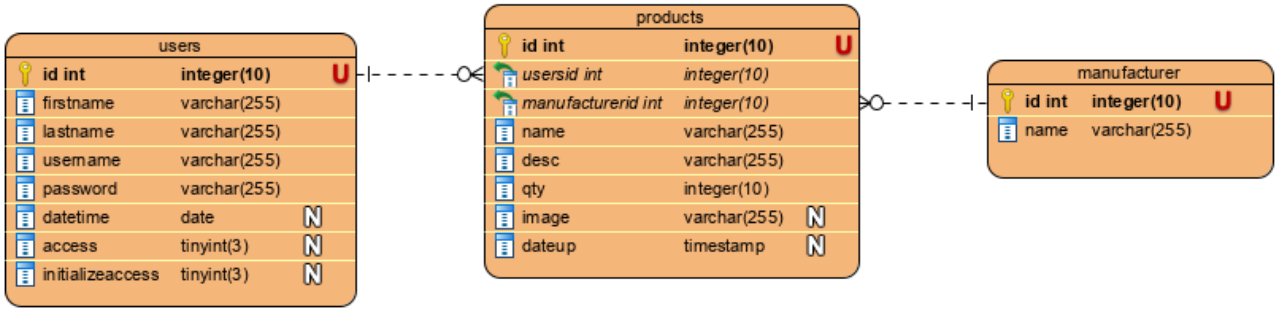

Figure 4-2 LogisTix ERD

4.4 SEQUENCE

The sequence of the design details the sequencing of the operation of the LogisTix IMS application. The user will attempt to login using credentials and once the user is authenticated, the user will be sent to user dashboard. If the credentials are not valid, the user will be sent back to index.

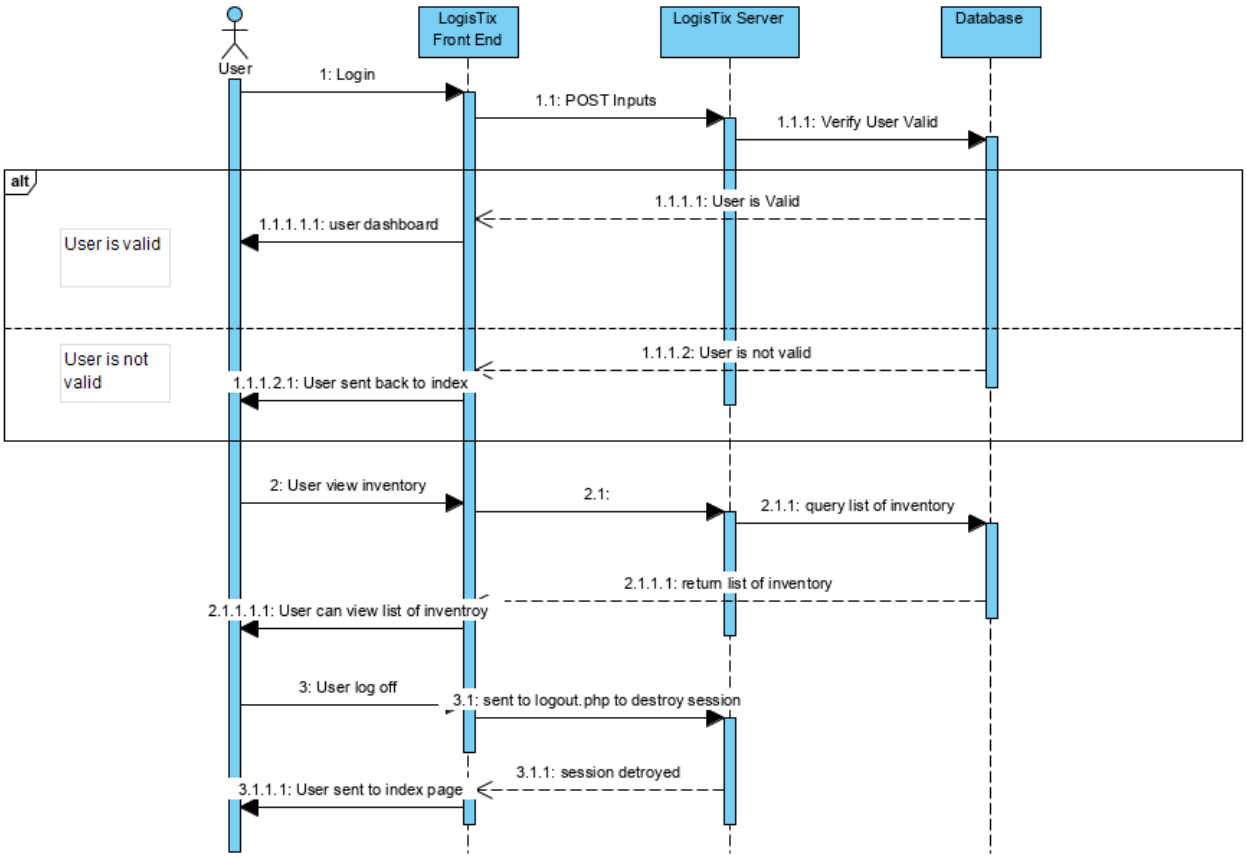

Figure 4-3 Sequence Diagram

4.5 INTERFACE AND FIELDS

LogisTix uses several types of interfaces and fields to provide a better user experience. The contact form uses text fields where users can provide input that will directly send email to the organization. The index of the site shows a styled table for a creative look and it also shows the last few items that were recently added. The login system uses a pop-up modal that will provide text and password fields for authorized users. The system also has a registration form that uses a text field and a password field in creating new users.

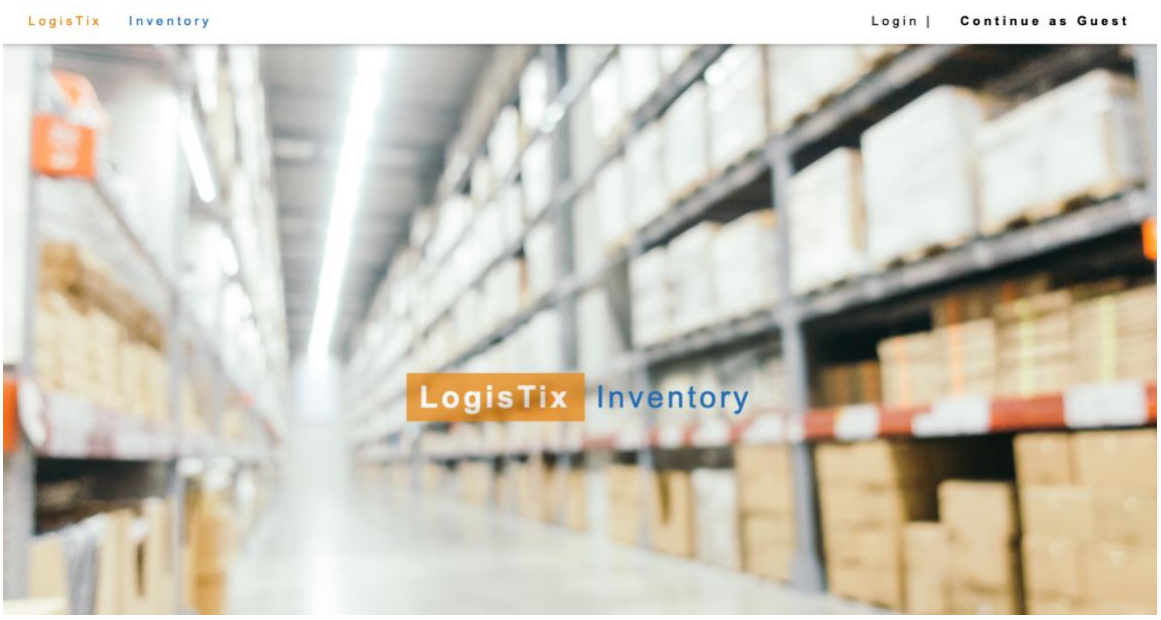

Figure 4-4 LogisTix IMS index

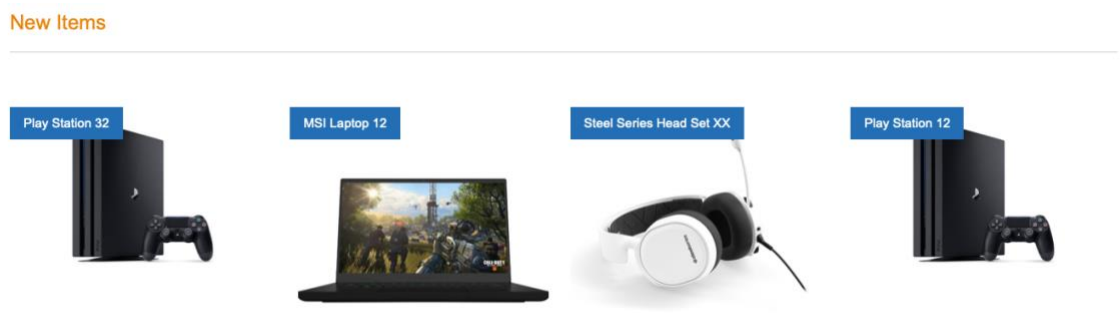

Figure 4-5 New Items

Figure 4-6 Contact Form

Figure 4-7 Login Form

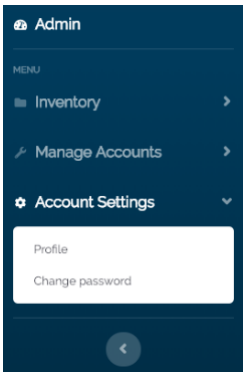

Figure 4-8 Side navigation dashboard

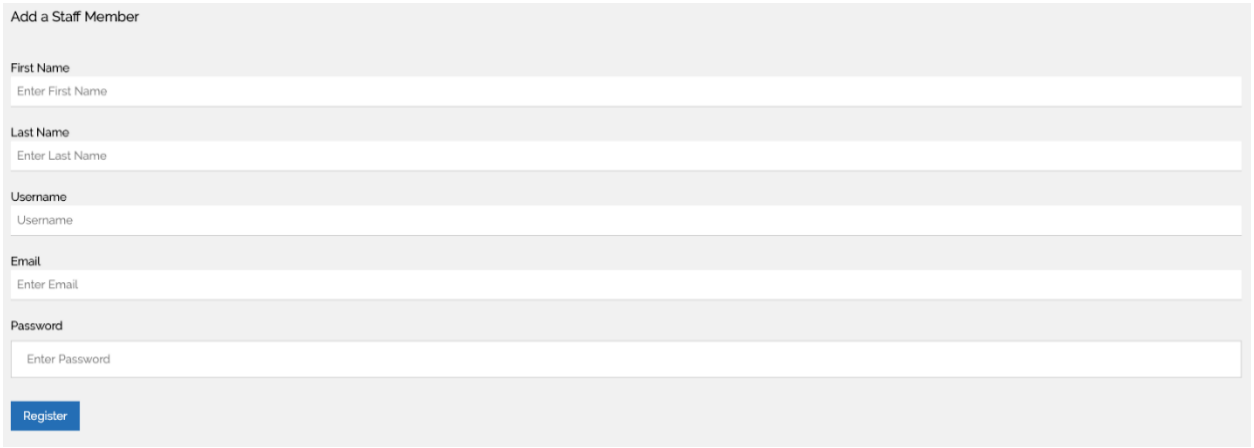

Figure 4-9 New member registration

4.6 USE CASE DIAGRAM

This use case diagram shows the LogisTix IMS logic side per use case. A guest can continue as guest and view the inventory, A member can view and edit inventory. The admin has the highest level of access and can edit and view inventory, and add and edit staff.

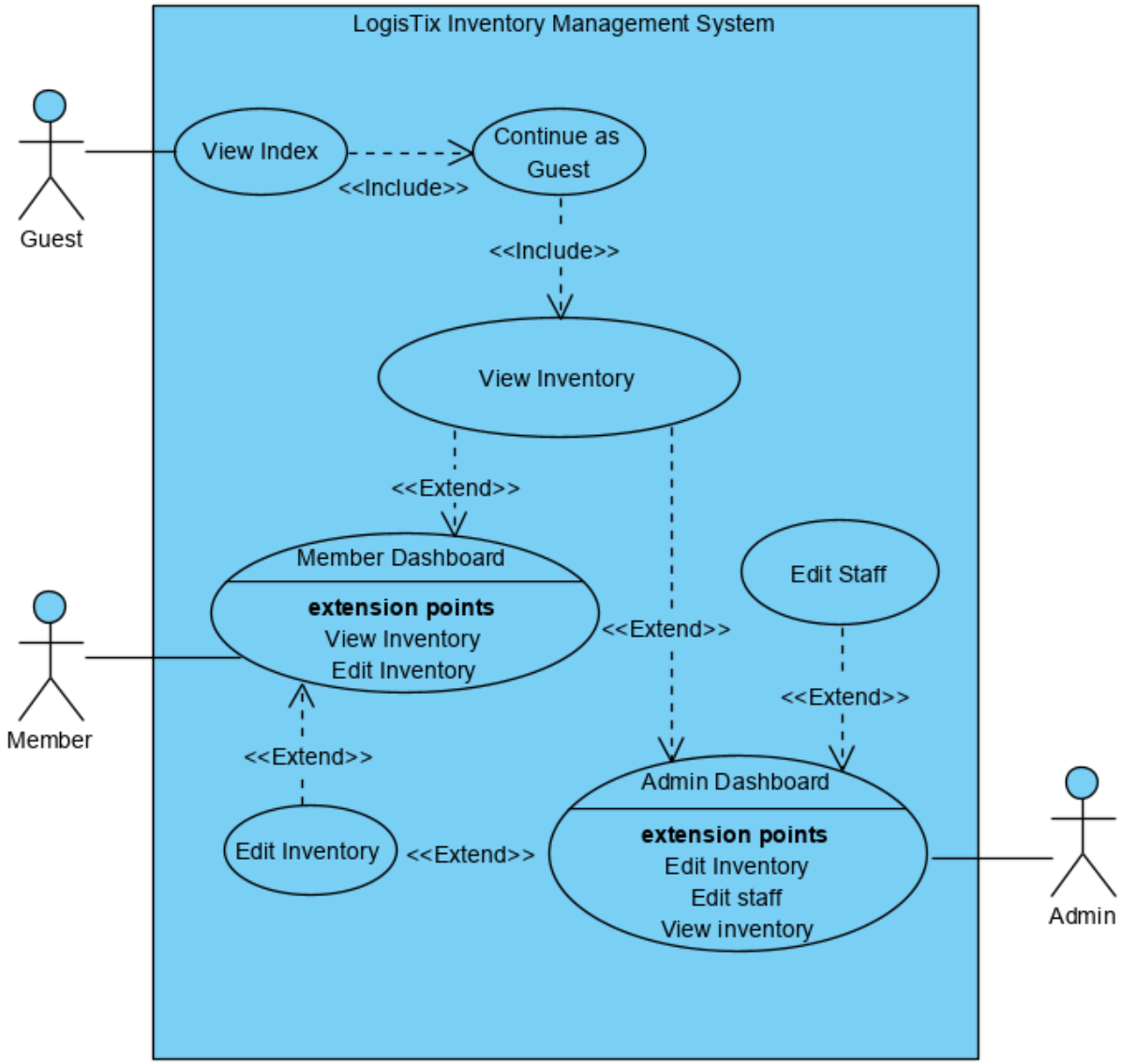

Figure 4-10 Use case diagram

4.7 PSEUDO CODE

The pseudo code will show a raw operation of functions logic within the LogisTix IMS application.

```
//connecting to database

algorithm connectdb() {
        // Get the DBParameters
        getDbparms();

         // Try to connect
         new mysqli($mydbparms->getHost(), $mydbparms->getUsername(),
                        $mydbparms->getPassword(),$mydbparms->getDb());

           if $mysqli->connect_error)

             die('Connect Error (' . $mysqli->connect_errno . ') '
                  . $mysqli->connect_error);

          return $mysqli;
        }

//provide a link to login modal

include ('includes/LoginModal.php');


//this shows the latest 8 products

algorithm showProducts(){
        connectdb();
   SELECT *, Timestamp(dateup) from products ORDER BY dateup DESC limit 8";
   $result = $mysqli->query($sql);
   if num_rows===1
        fetch_array(MYSQLI_ASSOC);
   /* Fetch the results of the query */
   while $row = fetch_assoc()
   $row["PImage"];
   $row["PName"];
        echo $name;
        echo $image;
}}
}
```

```
//login verification

algorithm verifyLogin($username, $password){

    connectdb();
    "SELECT * from users WHERE userName='$username'"
    $mysqli->query

    if num_rows === 1

        etch_array(MYSQLI_ASSOC);
        password_verify($password, $row['passWord']);
        if ($authcheck){
        session_start();

        $_SESSION['uname'] = $username;

        echo "<script>window.open('home.php','_self')</script>";


    }else{
        echo "<br>";
        echo "invalid entry";
        echo "<br>";
        header('Refresh: 1; URL = index.php');
     }
}

//verification if first time user admin

algorithm initAdmin(){
    connectdb();

    $_SESSION['uname'];

          SELECT initAccess, access from users WHERE userName='$user'
    query

    if num_rows===1

        fetch_array(MYSQLI_ASSOC);

        $initAdmin = $row['initAccess'];
        $access = $row['access'];
```

```
  if ($access == 1){

     if($initAdmin == 0){

        goto   'changePassword.php'
   }}}

   //to find the username

   echo $_SESSION['uname'];

   //to find if user is admin or admin to be greeted as such

         algorithm isAdminMember(){

         connectdb();

   $user=$_SESSION['uname'];

         SELECT initAccess, access from users WHERE userName='$user'
   query

   if num_rows===1

      fetch_array(MYSQLI_ASSOC);
      $access = $row['access'];

      if ($access == 1){
         echo "Admin";

      }  if($access == 0){
         echo "Member";
   }
}

//side navigation will vary display depending on what type of role

   algorithm isAdminSide(){
      $mysqli = connectdb();
      $access = "";
      $user=$_SESSION['uname'];
      $sql = "SELECT initAccess, access from users WHERE userName='$user'";
      $result = $mysqli->query($sql);

      if num_rows===1
```

```
        fetch_array(MYSQLI_ASSOC);
        $access = $row['access'];

        if ($access == 1){
            isAdminSideDisplay();

        if $access == 0{
            isMemberSideDisplay();
        }
    }

//sidenav for admin

algorithm isAdminSideDisplay (){
            echo "Overview"
            echo "Inventory"
            echo "Add Staff"
            echo "Change Password"
}

//sidenav for member

algorithm isAdminSideDisplay (){
            echo "Overview"
            echo "Inventory"
            echo "Change Password"
}

//sidenav for logout

 session_destroy();

 echo "Thanks for using LogisTix Inventory Management System"

 redirect (index.php);
```

4. 8 ALTERNATIVE DESIGN

- LOGIN: Implementation of 2FA will be beneficial for the site as it will add to security.
- LOGIN: Use of limitation of fail logs will prevent brute force attempt. After 3 fail logs, the system should disable the account and flags the administrator for further investigation.
- Backend: Implementation of frameworks like Laravel or symfony to help in a more efficient development.
- Front-end: Using bootstrap will allow us to create a more pleasing UI.
- Deployment: using different AWS service in the event the original architecture did not work. EC2 (Linux AMI) + RDS).

4. 9 CONTRIBUTION REPORT

All coordination, reviews, and investigations are communicated to the team through Trello or Discord.

Brandon: Project lead, Test

- Reviewed, finalized and submitted documentation.
- Coordinated team communication.
- Contributed in pseudo code.
- Hosted team meeting and provided inputs for this week's tasks.
- Worked with team leads to establish tasks in building Test Plan documentation.

Jella: Developer, Design

- Contributed in classes and functions.
- Contributed in Interface and Fields.
- Contributed in creative design: UI/ UX revision.
- Participated in the meeting and provided inputs for this week's tasks.
- Reviewed design document for accuracy and according to requirements.

Tom: Test, Image Editing

- Contributed in data and data file structure.
- Contributed in use case diagram.
- Contributed in sequencing diagram.
- Participated in the meeting and provided inputs for this week's tasks.
- Reviewed design document for accuracy and according to requirements.

Daryle: Developer, Design

- Contributed in sequencing diagram.
- Contributed in pseudo code.
- Formatted design documentation.
- Participated in the meeting and provided inputs for this week's tasks.
- Reviewed design document for accuracy and according to requirements.

LOGISTIX

"YOU NAME IT, WE GOT IT!"

UMGC | CMSC 495 7982

CHAPTER 5 SOURCE CODE (PHASE 1, 2 and 3)

Brandon Elliott | Jella An | Tom Cress | Daryle Urrea

CHAPTER 5 SOURCE CODE (PHASE 1, 2 and 3)

5.1 SUMMARY REPORT:

- Requirement Specifications Completed: (19/19) **100%**
- Test Result: [# of Test Cases: **31/31**] [Pass: **31/31**] [Grade: **100%**]
- Cloud migration and testing [**Successful**]
- Source code documentation
- Issues: Synchronization with members, source code versioning, document folder can get messy, back-end integration with front-end, bugs will always be present.
- On schedule with milestone: **YES**

5.2 DEVELOPMENT HISTORY (SOURCE AND TESTING)

5.2.1 Phase 1 (18 – 24 November)

- **18 to 20 November:** 11 of 19 Requirement Specifications completed: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 and 11. ***(Refer to Chapter 1: Specification Requirements)***
- **20 November:** [Test #1] 18 of 31 Test Cases Covered: 1, 2, 4, 5, 6, 7, 9, 10, 11, 12, 20, 22, 25, 26, 27, 28, 30, 31. ***(Refer to Chapter 3 Test Cases)***
- **Test #1 Result:**

  18 out of 31 Test cases were tested and 18 passed with the grade of **100%**. Although all the test cases that were tested passed, there are few bugs discovered during the tests.

  Bugs Discovered:

1. [Problem] Image size on the page for products are inconsistent while displaying the latest products.

   [Solution] *This is due to the inconsistencies of the image ratio and it's* ***fixed*** *by applying* ***style =" object-fit: cover"*** *on the img src.*

2. [Problem] After a contact email has been sent, there are codes visible on the background.
   [Solution] This is due to debugger is on in the PHP Mailer extension. To solve the problem is simply turn the code off ($mail->SMTPDebug = 0;).

5.2.2 Phase 2 (25 Nov – 01 Dec)

- **25 to 27 November: 8 of 19** Requirement Specifications completed: 7, 8, 9, 13, 14, 16, 17, 19 ***(Refer to Chapter 1: Specification Requirements).***
- **28 November: [Test #2]** 13 of 31 Test Cases Covered: 3, 8, 13, 14, 15, 16, 17, 18, 19, 21, 23, 24, 30. ***(Refer to Chapter 3 Test Cases)***
- **Test #2 Result:**

  13 out of 31 Test cases were tested and 9 passed with the grade of 69%.

- **Test #3 (Re-test) Result:**

  4 out of 31 test cases were tested and 4 passed with the grade of 100%

  Bugs Discovered:

  1. [Problem] #13 Failed due to no successful message after successfully added a product.
     [Solution: Fixed] dev team created alarm function to show upon a successful operation.
  2. [Problem] #14 When an admin clicks "Inventory" it does not send the admin to inventory but drops another option to select "Edit inventory" then the admin will be sent to inventory.php.

[Solution: **Fixed**] This is due to that “edit inventory” is a new adjustment to the side navigation. The link that was originally performing the function is rendered null and replaced by a drop down.

3. [Problem] #23 Unable to see login on mobile view.
   [Solution: **Fixed**] This is due to CSS code that’s requiring it to disappear when the size is small. To fix this, just remove the code from the class element.
4. [Problem] #19 #21 As a member, the tester could not edit product. Admin account passed.
   [Solution: **Fixed**] This is due to the function for a member is not properly configured. To solve this, developer ensure that the member have a function to edit products.
5. [Problem] #8 Dashboard does not show # of users per requirement specification.
   [Solution: **Fixed**] This is due to function not properly linked to the page. To solve this, the developers ensured that the code is properly called and link to the page.
6. [Problem] Profile pictures are not consistent throughout the site. Some shows different pictures on different pages.
   [Solution: **Fixed**] This is due to functions to call user’s image is mixed with a defaulted image. To solve this problem, the developers ensured that all profile pictures are using the function to call the user’s image.

5.2.3 Phase 3 (02 – 08 December)

- 02 December: Double check database settings and performed code walkthrough.
- 02 December: Fully migrated software to cloud server.
- **Test #4:** 02 – 05 December: Retest all requirements with a grade of 100%.
- 02 December to current: Monitoring the software for security and functionality.
- 05 December: Changed the admin access account for security.

5.3 COMPLETED REQUIREMENT SPECIFICATIONS

1. Accessible through the internet.
2. Will have a login interface for registered member.
3. New Items are displayed on the index (last 8).
4. Contact form for guest to contact Logistix.
5. Once logged in, the user is authenticated and sent to dashboard.
6. The dashboard will state welcome <Name> <time>.
7. The dashboard shows overview of the inventory system.
8. The dashboard shows how many items are out of stock.
9. The dashboard will show total stocks overall.
10. There is a limit of 30 mins to which user can idle, the user will have to reauthenticate.
11. User can click on "logout" link to logout and will be sent to index.
12. Admin can add/ disable regular member.
13. Admin can create, view, update, delete products.
14. Member can create, view, update products.
15. Guest can enter the site.
16. Guest can view the inventory.
17. Products will have Name, Manufacturers Name, Description, Qty, Image and status if out-of-stock.
18. Admin will need to change password first time login
19. The site will be responsive that can be viewed in multiple media. Mobile phone, tablet, laptops and pc.

5.4 COMPLETED TEST CASES

Test Case #	Requirement #	Test Description	Expected Result	Actual Result	Pass/Fail
Landing Page					
1	R1	User types in URL; Presses Enter.	URL takes User to Index Page.	URL takes User to Index Page.	Pass
2	R2	User chooses Login.	Login Page appears.	Login Page appears.	Pass
3	R3	The user view the new items in the index.	The 8 most recent items that were added to the inventory are listed are shown in the new item area.	The 8 most recent items that were added to the inventory are listed are shown in the new item area.	Pass
4	R4	The visitor uses the contact form and message the organization.	The admin will receive a message from the visitor through an email.	The admin will receive a message from the visitor through an email.	Pass
Dashboard and Overview					
5	R5, R6, R7	Admin logs in. username: admin pass: admin	Will be sent to dashboard overview with a welcome message, the username and access level (Admin).	Will be sent to dashboard overview with a welcome message, the username and access level (Admin).	Pass
6	R5, R6, R7	Staff Member logs in. username: membed pass: member	Will be sent to dashboard overview with a welcome message, the username and access level (Member).	Will be sent to dashboard overview with a welcome message, the username and access level (Member).	Pass
7	R5, R6, R7	User picks continue as Guest	Will be sent to dashboard overview with a welcome message and access level (Guest).	Will be sent to dashboard overview with a welcome message and access level (Guest).	Pass

Test Case #	Requirement #	Test Description	Expected Result	Actual Result	Pass/Fail
Dashboard and Overview					
8	R8, R9	Dashboard Overview Admin and Member	Shows out of stock status, total stocks, number of users and list of stocks	Shows out of stock status, total stocks, number of users and list of stocks	Pass
9	R10	Any user will idle for 30 mins or more.	User is deauthenticate and reauthentication is required.	User is deauthenticate and reauthentication is required.	Pass
10	R11	While logged in, user will click "Logout".	Session destroy and user is sent to index page.	Session destroy and user is sent to index page.	Pass
11	R12	Admin create add a staff member inputs: First name: Member Lastname: One Username: MemberOne Password: Member1	Tester verify the database info: First name: Member Lastname: One Username: MemberOne Password: Member1	Tester verify the database info: First name: Member Lastname: One Username: MemberOne Password: Member1	Pass
12	R12	Tester using admin account will delete a member	The deleted member is deleted from database and cannot login.	The deleted member is deleted from database and cannot login.	Pass
13	R13, R17	Admin click add product: Name: Product 1 Manufacturer: Product Test Description: Test product description QTY: 99 Image: tester selected image	Product success message is shown. Database will have the new product info Name: Product 1 Manufacturer: Product Test Description: Test product description QTY: 99 Image: tester selected image	Product success message is shown. Database will have the new product info Name: Product 1 Manufacturer: Product Test Description: Test product description QTY: 99 Image: tester selected image	Pass

Test Case #	Requirement #	Test Description	Expected Result	Actual Result	Pass/Fail
Dashboard and Overview					
14	R13	Member click "Inventory" on side navigation. Drops down menu and click on “Edit Inventory	Admin will be sent on the inventory page and view list of inventories.	Admin will be sent on the inventory page and view list of inventories.	Pass
15	R13	Admin click edit on the inventory list and inputs information on the form to update: Name: Product Updated 1 Manufacturer: Product Test Updated Description: Test product Updated description QTY: 15 Image: blank	Product updated success message is shown. Database will have the new product info Name: Product Updated 1 Manufacturer: Product Test Updated Description: Test product Updated description QTY: 15 Image: blank	Product updated success message is shown. Database will have the new product info Name: Product Updated 1 Manufacturer: Product Test Updated Description: Test product Updated description QTY: 15 Image: blank	Pass
16	R13	Admin click on delete while on inventor list page.	The product delete will be erased from the list and will be deleted from the database.	The product delete will be erased from the list and will be deleted from the database.	Pass
17	R14, R17	Member click add product and provided input: Name: Member Product 1 Manufacturer: Member Product Test Description: Member Test Product description QTY: 89 Image: tester selected image	Product successfully added message is shown. Database will show product info: Name: Member Product 1 Manufacturer: Member Product Test Description: Member Test Product description QTY: 89 Image: tester selected image	Product successfully added message is shown. Database will show product info: Name: Member Product 1 Manufacturer: Member Product Test Description: Member Test Product description QTY: 89 Image: tester selected image	Pass

Test Case #	Requirement #	Test Description	Expected Result	Actual Result	Pass/Fail
Dashboard and Overview					
18	R14	Member click "Inventory" on side navigation. Drops down menu and click on “Edit Inventory	Member will be sent on the inventory page and view list of inventories.	Member will be sent on the inventory page and view list of inventories.	Pass
19	R14	Member click edit on the inventory list and inputs information on the form to update: Name: Member Product Updated 1 Manufacturer: Member Product Test Updated Description: Member Test product Updated description QTY: 5 Image: blank	Update successful message shows and information of the product in the database is updated as Name: Member Product Updated 1 Manufacturer: Member Product Test Updated Description: Member Test product Updated description QTY: 5 Image: blank	Update successful message shows and information of the product in the database is updated as Name: Member Product Updated 1 Manufacturer: Member Product Test Updated Description: Member Test product Updated description QTY: 5 Image: blank	Pass
20	R15, R16	Guest click "Continue as Guest"	Guest is sent to guest page where inventory list can be viewed.	Guest is sent to guest page where inventory list can be viewed.	Pass
21	R17	A tester uses admin or member and edit the products quantity to zero.	When a product qty is at zero, it will show on the dashboard that it is out of stock.	When a product qty is at zero, it will show on the dashboard that it is out of stock.	Pass
22	R18	A tester will use an admin account for the first time.	The admin will be sent to change password page to change the password.	The admin will be sent to change password page to change the password.	Pass
23	R19	Tester view the site in a mobile phone.	The display adjusts to mobile view.	The display adjusts to mobile view.	Pass
24	R19	The tester views the site in an ipad.	The display adjusts to ipad view.	The display adjusts to ipad view.	Pass

Security					
25	Authentication Security	Tester uses wrong login info: username: hacker password: hacker	An invalid entry will show	An invalid entry will show	Pass
26	SQL Injection	SQL Injection: Tester will user sql injection code to login username: " or ""=" pass: " or ""="	An invalid entry will show	An invalid entry will show	Pass
27	Cross Site Scripting	Test will input: on the guest message box: <script>alert("TEST");</script>	The script is nullified by just displaying <script>alert("TEST");</script> as text in the message box.	Script does not show on the email.	Pass
28	Password Encryption	Tester will verify that each users passwords format.	Each password should be in an encrypted format. Example format: $2y$10$XIykz7Sj6uqFkmDOop1F1uQC/NOi/VYbvo07InszrMEte0YxTu9su	Each password should be in an encrypted format. Example format: $2y$10$XIykz7Sj6uqFkmDOop1F1uQC/NOi/VYbvo07InszrMEte0YxTu9su	Pass
29	Path Traversal	Tester uses browser to go to homepage without logging in.	Tester is sent back to index.	Tester is sent back to index.	Pass
30	Path Traversal	Tester uses browser to go to Inventory page without logging in.	Tester is sent back to index.	Tester is sent back to index.	Pass
31	Path Traversal	Tester uses browser to go to add member page without logging in.	Tester is sent back to index.	Tester is sent back to index.	Pass

5.6 ISSUES/ SOLUTIONS

5.6.1 Synchronization between team members: This issue is expected given the model and displacement between team members. Considering the distance, time zone and other life's work, this is a problem. To solve this issue, an *effective communication and scheduled meetings* can put everyone in the same page. Updating the virtual office on what is currently being done should be practiced.

5.6.2 Document folder can get messy: Every member of the team has their own individual tasks and the folder where everyone uploads their files will get confusing. *Organizing the folders, having a standard file naming system and cleaning the folder as you go* will alleviate a lot of headaches in the future.

5.6.3 Back-end integration with front-end: This can be situated by *testing as often*. When a function is integrated with the front-end, it is better to test it right away than finding out later that it does not work and could cause a major setback.

5.6.4 Source Code Versioning: Using version control can be difficult when everyone is pushing their updated codes to the repository and to minimize this issue, the *person merging the code should communicated to the owner of the code being merged to have a good understanding of the merge*.

5.6.5 Bugs: Bugs will always exist in a source and the only way to beat it is to test early and test often. This will give the development and early awareness on which bug exist and they will have time to fix it

5.7 CONCLUSION:

5.7.1. Lessons Learned:

a. **Communication:** Since the project is done on a virtual environment, communication will be more challenging. Every member of the team should do their best in participating in the virtual office or chat to be in the same page. This will prevent duplication of work and allow everyone to participate more efficiently.

b. **Skills:** Since everyone is new to each other, we have no idea who can effectively complete which task. By strategically assigning roles based on their confidence and starting a culture that every task is open to everyone, we effectively completed all the software requirements in timely manner.

c. **Bugs:** During testing, there are numerous bugs that are discovered within the software and we tackled the issue by actively fixing the bugs while test is being conducted. Bugs were fixed in the order of importance. This means that bugs preventing a requirement from being met were fixed first, then those which could prevent basic functionality, and finally any bugs that were leftover.

5.7.2. Design Strengths:

a. **Flexibility:** The source code is flexible enough that any modification to the program can be implemented with relative ease.

b. **Easy to understand:** The structure of the source is easy to understand, and comments are included. This will allow anyone with coding experience to not only understand the code, but contribute to it as well.

c. **UI/UX:** The software application has a pleasing design that give users better experience and view. The design can also be easily adjusted to suit the deployment environment.

5.7.3. Limitations:

a. **Security:** The application can use another security enhancement like 2FA and captcha functionality.

b. **View:** The guest view can use enhancement by being able to select an individual product to view description. This would be a relatively simple addition, but outside the scope of the original requirements.
c. **Alerts:** The application could be expanded to add automatic e-mail alerts when a product runs out of stock.

5.7.4. Future Improvements:

a. **2FA:** To improve the security of the system, the team can add 2FA.
b. **Event log view:** To further and easily track the operation of the system, an event log display on administrator's dashboard will be beneficial.
c. **Individual product view:** To add to the user experience, an individual product view with the product's details should be an option.
d. **Reactivate account:** Administrator should have a reactivation option to reactivate a disabled account.

5.7 CONTRIBUTION REPORT (Phase 1, 2 and 3)

All coordination, reviews, and investigations are communicated to the team through Trello or Discord.

Brandon: Project lead, Test

- Reviewed and finalized documentation for submission.
- Coordinated team communication.
- Hosted team meetings and provided inputs on every week's tasks.
- Perform testing in accordance with the test case chart.
- Perform post deployment testing and monitoring of the application

Jella: Developer, Design

- Contributed in Phase 1, 2 and 3 documents.
- Participated in the meetings and provided inputs on every week's tasks.
- Reviewed the current project document for accuracy.
- Wrote source code.
- Maintain GitHub repository.
- Perform post deployment testing and monitoring of the application

Tom: Test, Image Editing

- Contributed in Phase 1, 2 and 3 documents.
- Participated in the meetings and provided inputs on every week's tasks.
- Reviewed the current project document for accuracy.
- Perform testing in accordance with test cases chart.
- Perform post deployment testing and monitoring of the application

Daryle: Developer, Design

- Contributed in Phase 1, 2 and 3 documents.
- Participated in the meetings and provided inputs on every week's tasks.
- Reviewed the current project document for accuracy.
- Wrote source code.

- Maintain GitHub repository.
- Created AWS cloud architecture and deployed the application.

5.8 REFERENCES

IEEE. *IEEE Std 1058-1998 IEEE Recommended Practice for Software Design Descriptions.* IEEE Computer Society, 1998.

PHP. (n.d.). PHP: Documentation. Retrieved from https://www.php.net/docs.php

XAMPP. (n.d.). XAMPP Installers and Downloads for Apache Friends. Retrieved from https://www.apachefriends.org/index.html

AWS. (n.d.). Amazon EC2. Retrieved from https://aws.amazon.com/ec2

SoftwareTestingMentor. (2013, February 12). Importance of metrics and measurement in SDLC. Retrieved from https://www.softwaretestingmentor.com/importance-of-metrics-and-measurement-in-sdlc/

OWASP. (n.d.). The OWASP Testing Framework. Retrieved from https://www.owasp.org/index.php/The_OWASP_Testing_Framework#Phase_1.1:_Define_a_SDLC

RedHat. (2019). CVE-2019-0220. Retrieved from https://access.redhat.com/security/cve/cve-2019-0220

OWASP. (n.d.). Review webpage comments and metadata for information leakage (OTG-INFO-005). Retrieved from https://www.owasp.org/index.php/Review_webpage_comments_and_metadata_for_information_leakage_(OTG-INFO-005)

AffinitySecurityService. (2018, January 8). What Is Information Leakage ? Retrieved from https://affinity-it-security.com/what-is-information-leakage/

Apache. (n.d.). httpd 2.4 vulnerabilities. Retrieved from https://httpd.apache.org/security/vulnerabilities_24.html

OWASP. (n.d.). OWASP Testing Guide v4 Table of Contents. Retrieved from https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents

Testing for Reflected Cross site scripting (OTG-INPVAL-001). (n.d.). Retrieved from https://www.owasp.org/index.php/Testing_for_Reflected_Cross_site_scripting_(OTG-INPVAL-001)

OWASP. (n.d.). Testing for Stored Cross site scripting (OTG-INPVAL-002). Retrieved from https://www.owasp.org/index.php/Testing_for_Stored_Cross_site_scripting_(OTG-INPVAL-002)

OWASP. (n.d.). Testing for SQL Injection (OTG-INPVAL-005). Retrieved from https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OTG-INPVAL-005)

Klariti. (2018, October 22). User Guide Tutorial. Retrieved from http://www.klariti.com/technical-writing/User-Guides-Tutorial.shtml

Hodgson, P. (2007, June 4). Tips for writing user manuals. Retrieved from https://www.userfocus.co.uk/articles/usermanuals.html

Agile Modeling. (n.d.). Core Practices for Agile/Lean Documentation. Retrieved from http://www.agilemodeling.com/essays/agileDocumentationBestPractices.htm

Klariti. (2018, October 22). User Guide Tutorial. Retrieved from http://www.klariti.com/technical-writing/User-Guides-Tutorial.shtml

Hodgson, P. (2007, June 4). Tips for writing user manuals. Retrieved from https://www.userfocus.co.uk/articles/usermanuals.html

Agile Modeling. (n.d.). Core Practices for Agile/Lean Documentation. Retrieved from http://www.agilemodeling.com/essays/agileDocumentationBestPractice

Dragon Point Software. (2019, May 1). What's the Design Phase of a Software Development Project? Retrieved from http://www.dragonpoint.com/whats-the-design-phase-of-a-software-development-project/

UML Diagram. (n.d.). Retrieved from https://www.smartdraw.com/uml-diagram/

Visual Paradigm. (n.d.). UML Class Diagram Tutorial. Retrieved from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutoria