

# **Runtime Terror - Nozama**

## **Project Vision / Abstract:**

This software will benefit the user because it will allow for easy management of an online store's inventory, allowing the store to add/remove items, keep track of the total inventory, keep track of orders. The system will manage the store's inventory allowing the employees to focus on other tasks. The main stakeholders for this project are the customers of the store and the store's employees. The customers will only be able to purchase the items if they are in stock, and the employees will need to know if stock is getting low so that they can order more. The system will have a GUI that allows for the user to look up an item by an ID number and see its stock, a general description, and make a purchase. The software will also need to keep track of the items that the customer has added to their cart, to enable the purchase of multiple items.

## **Website Links:**

Website - <https://runtimeerrorbu.github.io/Nozama/>

GitHub - <https://github.com/RuntimeTerrorBU/Nozama>

GitHub Issue Tracking - <https://github.com/RuntimeTerrorBU/Nozama/issues>

## **Group Members w/ Timecard:**

Ashley Bickham:

- 42 hours worked
- 44 total commits

Joshua Hunter (Project Leader):

- 47 hours worked
- 48 total commits

Austin Lehman:

- 52 hours worked
- 96 total commits

Tyler Ross:

- 35 hours worked
- 29 total commits

**Total Commits: 211**

*\*All commits at: <https://github.com/RuntimeTerrorBU/Nozama/commits/gh-pages>*

## **Background story for introduction**

In the beginning of this project, we wanted to create a product site similar to that of Amazon or Etsy, with the outcome being that of an online inventory store. Rather than focus on one section of shopping, like clothes, food, or electronics, we wanted to create a store that allows for the user to search any type of item on our interface. We were faced with questions when trying to create this project, such as “What should the final result look like?” and “How do we go from a simple, drawn design to a more complex, fully functioning program?”

One of our biggest challenges we were hit with was the ignorance when it came to the amount of work we would have as we worked to complete each iteration. Each part in itself seems very simple and manageable, but with the growing code-line total, the complexity can quickly ramp up.

Toward the end of this project, we learned to work off of the work of each other more efficiently than when we began working together. We were successfully able to create a (very bare boned) store site in order to hold, sell, and restock the items held in our store.

## Requirements

	REQ 1	REQ 2	REQ 3	REQ 4	REQ 5	REQ 6	REQ 7	REQ 8	REQ 9	REQ 10	REQ 11	REQ 12	REQ 13	REQ 14	REQ 15	REQ 16	REQ 17
UC - 1 Make Purchase	✓		✓			✓	✓			✓							
UC - 2 Add Item to Cart	✓		✓	✓	✓		✓			✓				✓		✓	
UC - 3 Add Item to Wishlist	✓		✓	✓			✓			✓				✓	✓	✓	✓
UC - 4 Restock Item	✓		✓	✓			✓		✓								
UC - 5 Return Item	✓		✓			✓	✓				✓	✓	✓				
UC - 6 Make Account								✓									
UC - 7 Ship/Send Item	✓		✓				✓			✓	✓						
UC - 8 Add New Product	✓	✓	✓							✓							
UC - 9 Delete Product	✓	✓	✓														
UC - 10 View Product Availability	✓		✓	✓					✓	✓	✓			✓		✓	✓
UC - 11 Generate Report						✓	✓		✓							✓	
UC - 12 Remove Item from Cart	✓		✓		✓		✓										

**R1:** The system will keep a set of items that are sold by the store

**R2:** The store will be able to add and remove from the set of items

**R3:** Items will be associated with a unique ID to allow for easy lookup and management

**R4:** The system will keep track of the amount of each item that is available for sale

**R5:** Customers will be able to add and remove items they wish to purchase into a cart

**R6:** The system will be able to keep track of each order by its unique ID

**R7:** Each order will be stored in a database to allow for easy lookup

**R8:** Customers must be able to make accounts easily

**R9:** When stock is low, the system must notify an employee or manager

**R10:** Customers can rent items that allow it.

**R11:** Customers can return items that have been rented.

**R12:** If a customer's rental is late a fee should be sent to them.

**R13:** The customer should be able to return an item within 30 days.

**R14:** The customer should be able to search for items.

**R15:** The customer should be able to add items to a wishlist.

**R16:** The system should notify the customer when an item in their wishlist is in stock.

**R17:** The system should notify the customer when an item in their wishlist is on sale.

## Analysis

## **Design**

For this project, we wanted to create a working product with a similar look and feel as those of big name companies, like Amazon or Etsy. Our idea was to have different pages that were displayed with each main part of the product that the user needed to work with, and we designed Nozama to meet these ideas. We focused on creating more of an Object Orientated design, creating different classes that we would implement inside of our controllers and main view classes in order to display Nozama.

We made three main packages filled with our controller classes, our viewing classes, and our main object files. These were created to separate the different items and their functionalities, rather than placing everything on one common file, or fewer than what we created.

Our controller classes

## **Implementation**

To implement our design ideas, we created a starting page that would allow the user to login as a Customer, a Company Member, or create a login if they had never logged in once before. Once the user created their login (stating whether they were creating a login as a Customer or Company member), they could then login to their perspective sites.

If the user logged in under the Customer page, they were met with the Home Page of Nozama. You can see the items sold in the store, as well as the option to add an item to cart or wishlist. The customer can also search through the catalog of items or clear their search. They also have access to their Shopping Cart through the “Show Cart” button and their Wishlist through the “Show Wishlist” button. When opening the Shopping Cart, the Customer can see what items they have put in their cart, and choose to checkout if they would like to purchase said items. If they checkout, they enter their shipping and payment information (address, card information), and are given a receipt that they can download, as well as a shipping confirmation through Novama. When the Customer clicks to access their wishlist, they are met with another window that displays any items that they have placed into their wishlist, and they can edit the amount of that item they’d like to keep in their wishlist.

If the user logged in under the Company page, they have access to the more behind-the-scenes parts of the Nozama store. The new options that the user is able to do include the ability to add a product to sell through Nozama to the database, delete a product from the database, restock an item, or generate a report of the orders that have been made through Nozama. When the user decides to add a product by clicking the “Add Product DB” button, they must input the name, description, price, ID, and quantity of the item they wish to add to Nozama. To delete a product from Nozama, the user must first select the item from the catalog, then press

the “Delete Product DB” button, which will delete the item from the view and the Nozama database. The company member can also restock an item in the store through the “Restock Item” button, and download a report of the ItemID’s and quantities of those items that have been sold through Nozama. The company member is also able to do any of the above actions available to the Customer.

Our design was made to work through different windows for different ways to move and work through Nozama. This design makes it easier for the user to access what they’re looking for within our project.

**evaluation/testing**



## **Conclusion**

Throughout this project, we were constantly faced with new bugs, issues, and ways to improve Nozama in order to make it more efficient for the user. Nozama allowed us to create a program that functioned as a working inventory management website, stretching our knowledge of Swing, JUnit testing, and GUI creation to its limit.

**Each section must use appropriate UML diagrams**

**Labeled and referenced**

## **User manual / guide**

### **USER MANUAL:**

#### **LOGIN PAGE:**

The first page the user is met with is the Log-In Page. They can select one of three options: Customer Login, Company Login, or Create Login. With the selection of Customer Login, the user must enter their username and password associated with Nozama. Only users with Company access are able to login through the Company Login button. If the user does not have any account, they can create one to be stored in Nozama when pressing the Create Login button.

#### **CUSTOMER PAGE:**

#### **COMPANY PAGE:**

**Roster of hours per team member**

**Results from git analysis (how many commits, each member commits, List of git contributions)**

**Link to issue tracking**

**Complete application with user interface and user input validation**

**Test documentation and JUnit testing**

**Suggested point redistribution**

**Code documentation (JavaDoc)**

**Runnable as JAR**

### Timecards report - Updated Gantt Chart for Iteration 3

## Runtime Terror Gantt Chart

CSI 3471

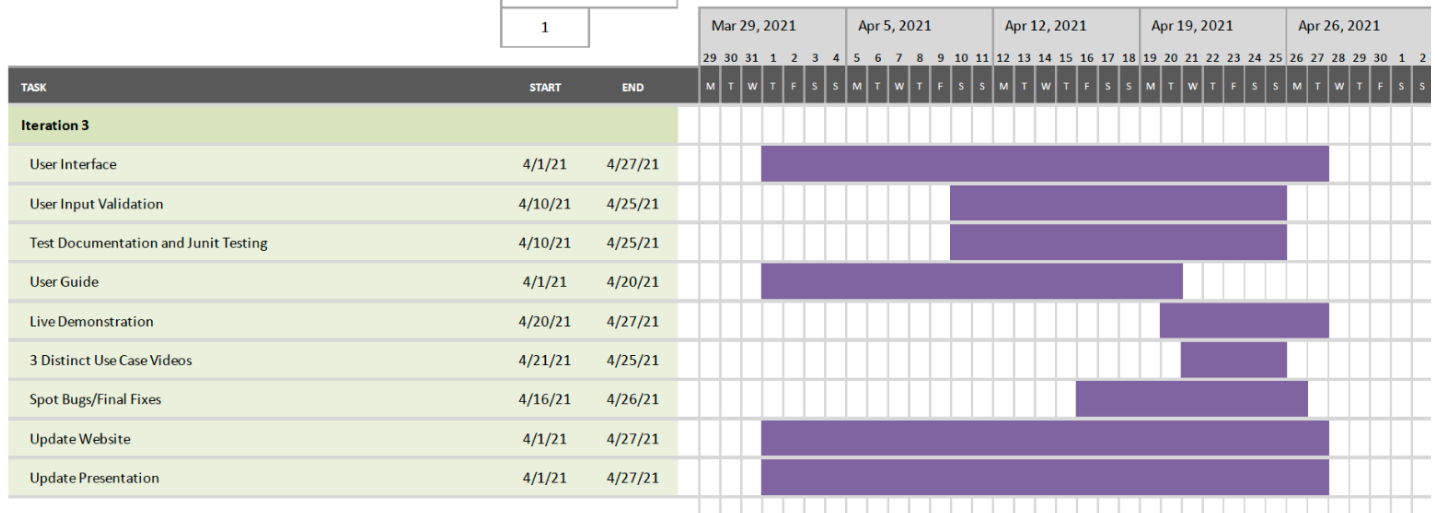
Ashley Bickham, Joshua Hunter, Austin Lehman, Tyler Ross

Thu, 4/1/2021

1

SIMPLE GANTT CHART by Vertex42.com

<https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html>



**Code quality matters (comments/formatting/coupling/cohesion)**

## Revised Iteration 2

Available for download at <https://runtimeerrorbu.github.io/Nozama/>

*\*Also included in Canvas submission*

### **Original Point Distribution: 4.4 / 5**

- From the original Iteration 2, we lost minor points due to issues within our diagrams and were missing detailed justification of our GRASP Patterns. We have fixed everything that was mentioned in the comments on our original submission, therefore we believe the suggested point redistribution should be full credit.

### **Suggested Point Redistribution: 5 / 5**

*Changes that were made:*

- Reply messages that were missing a dotted line were given a dotted line
- Changed the GRASP pattern to be more detailed and explained