# **Test Plan Report**

Ice Track

Neapolitan Solutions

Wednesday, October 20th, 2021

## 2. Introduction

This document describes the test policy for Neapolitan Solutions' Ice Track software product developed for Tom and Adam's Ice Cream Company. Ice Track is a comprehensive web solution to shipment tracking and inventory management relying on four subsystems in a multiuser environment with an emphasis on responsiveness, multi-client synchronization, and ease reversal of changes.

Ice Track's principal subsystems are: order entry, inventory management, shipment tracking, and trouble ticket management. These subsystem's respective tasks include the automation of orders from enterprise customers, providing an interface for product inventory update, storage of relevant information pertaining existing orders, and a reporting system for errors a user may experience at any step of the software operation.

## 

## 3. Software Test Environment

Requirements for white box testing:

1. Hardware Items:

* 4 Gigabytes (GB) ram minimum.
* 320 Kilobytes (KB) hardware disk space minimum.

1. System Software Items:

* Node.js v10.19.0 or later:
  + Windows 10 21H1 build 19043 or later
    - 64-bit or 32-bit
  + Mac OS 11 Big Sur or later
    - 64-bit
  + Linux 5.13.0-7614 or later
    - 64-bit
    - ARMv7 or ARMv8
* Any Terminal or IDE capable of local hosting on ports using Node.js. (Visual Studio Code)

We are using JIRA as our problem reporting system.

Requirements for black box testing:

1. System Software Items:

* Any internet browser with an internet connection:
  + Google Chrome 95.0.4638.50 or later
  + Mozilla Firefox 60.9.0 or later
  + Safari 15 or later
  + Microsoft Edge 94.0.992.50 or later

## 

## 4. Test Description

1. **Inventory Tests:** the objective of these tests would be to confirm that the inventory system functions as intended and can be edited by authorized users. The requirements tested would be 3.1.1 to 3.1.11 as listed in our requirements matrix. The types of testing necessary would be Database Testing. The pass/fail criteria of the database testing would show that the data in the inventory system is intact and not corrupted, and that only authorized users have access to *editing* the current inventory (all users should be able to *view* the current inventory).
2. **Order Placement System Tests:** the objective of these tests would be to ensure the order placement system functions as intended and interacts with the inventory system correctly when a customer places an order. The requirements tested would be 2.1.1 to 2.2.3. The types of testing necessary include Security-Relevant Testing, Database Testing, and Comparison Testing. The pass/fail criteria of the Security-Relevant Testing would be that the login portal for customers is secure and that no users would be able to access subsystems that are not intended for them to access. The pass/fail criteria of the Database testing would show that placed orders affect the inventory as intended i.e. placed the ordered items in the pending inventory so that those items cannot be ordered a 2nd time. Some Assumptions we might have are that there are no concurrent orders being placed, i.e. two customers ordering the same product at the same time. Our constraint for these tests would be that they to not occur concurrently.
3. **Shipment Tracking Tests:** the objective of these tests are to ensure proper functionality of the shipment tracking system, so that orders are never lost in the shipping process, and that customers can always have access to information about their order. In addition, if any problems arise with the shipment, they are reported automatically to the trouble ticket system. The requirements tested in these tests are 4.1.1 to 4.2.5. The types of testing necessary for this subsystem are Comparison Testing and Security-relevant testing. The pass/fail condition for the comparison test would be that the customer always receives the current, up-to-date tracking information whenever requested. The pass/fail condition for the Security-Relevant test would be that users/customers would only be able to access shipment tracking information relevant to their own orders, and not other orders. For these tests, we can assume that we would not want real shipments to be used when testing this software, so we would constrain these tests to a simulated environment.
4. **Trouble Ticket Management Tests:** The objective of these tests are to ensure that problems that arise are recorded and documented properly so that they can be corrected in a timely manner. The requirements tested would be 5.1.1 to 5.2.2. The types of testing necessary for this subsystem include performance testing, specifically stress testing, as well as Comparison Testing and Security-Relevant Testing. The pass/fail criteria for the stress performance tests would be that the system can handle many trouble tickets placed in a short amount of time, such as when a customer is particularly frustrated. The pass/fail criteria for the Comparison tests would be that if a user were to enter the same trouble ticket twice, only one would be recorded. The pass/fail criteria for security-relevant testing would be that regular customers’ user accounts cannot see things that an administrator could see, such as the notes associated with a trouble ticket. (for example, an admin might deem a certain ticket low-priority, and if a customer saw this information they would be upset that their issues aren’t being prioritized).

## 5. Data Recording, Reduction, and Analysis

* GUI:
  + Is the GUI buggy or does it display what is needed
  + Does it look nice and user friendly
  + Are there spelling errors or other mistakes
  + Data:
    - Does not really return specific data, just makes sure it renders correctly
* Security
  + Does the log in/sign up system work
  + Are you able to access pages without being logged in
  + Are you able to work around or spoof the login process
  + Are you able to take over someone’s account
  + Data:
    - If user is not logged in, return false and redirect page
    - If user is not allowed on a certain page, deny and redirect page
* Roles
  + Are pages or elements displayed differently based on user roles
    - Attempt to go to pages roles should not have permission for; redirect if not authorized
  + Is information protected from certain roles
    - Attempt to access information not available to a role
    - Make sure no compromising data is returned or accessible
  + Does each role have the ability to do what it is meant to do
    - Test each role’s functionality
    - Make sure the information queried is returned correctly
  + Is each role defined well
    - Return specific permissions and make sure the permissions are accurate
* Auto Logout
  + Does it log out after a defined period of time of idling?
    - Let the program sit for that defined time without interaction to test this
    - Test to make sure that the timer works by returning or logging intervals
* Processing of names or words
  + Try different combinations of names/words with hyphens and characters to make sure it does not break the program, but accurately gets that information
* Printing
  + Is the information printable and easily readable?
  + Is the information being printed accurate and formatted correctly?
* Reporting Requirements
  + Are the reports viewable before printing?
  + Are the reports accurate and formatted correctly?
* Online help
  + Is the FAQ and documentation readable and easy to understand?
    - Have others not involved with the development of the program read this and help make sure it is easy to follow
* Screens
  + Do the screens and other menus work?
    - Test screen sizes and window size, and make sure everything that is supposed to be clickable is
* Getting information
  + Does gathering information from the database return the correct data? Is it usable?
* Error testing
  + Test each function with wrong information/invalid information to make sure that the error is caught.
    - Ex. Leave required fields blank, put in wrong information, make sure injection does not work, etc.
    - Test hyphenated and spaced names in fields, as well as special characters
* Function Testing
  + Make sure functions do exactly what they are supposed to do.
* Runtime
  + Test the time it takes to run each function and to load the data
* Boolean for function success
  + Return a boolean value if the function ran correctly.
* User data returned
  + User Display Name
  + User ID
* Correct Database data returned using example ID
  + Return tickets
  + Return users
  + Return roles
  + Return Orders

## 

## 6. Test Conditions and Schedule

[See updated schedule here](https://docs.google.com/spreadsheets/u/3/d/1QeU2sPi6dW89G62PEPwvEtTfkLfFjAGA5nVIOWRuGJE/edit)

## 

## 7. Updated Requirements Matrix

[NS-262150-191021-0531.pdf](https://drive.google.com/file/d/1s-oeqEPN7gM6FG_V4nU2S7wwILoYIIps/view?usp=sharing)