CS 532 Software Engineering Test Plan

ICE TRACK System

Professor Leonard

March 15, 2016

Team ICE TRACK

Dustin Bonilla

Eric Ly

Ali Saadati

Mohammed Alshahrani

Anass Benothmane

Colan Kirk

**2. Introduction**

ICE TRACK shall be able to maintain the inventory of ice creams and generate reports on it, take in orders from a user in an acceptable format (provided by the client), submit the order to shipment tracking, and keep track of problems via a ticketing system. The inventory will allow users to add, delete or change status of items in inventory and log each interaction. The orders submitted shall check with the inventory for availability and then respond appropriately. Upon confirmed order, the shipment tracking will maintain the status of products during transportation to vendors, including anything that goes awry. Any troubles the system experiences or reports users submit are tracked by a ticketing system capable of sorting tickets accordingly.

Identifies the scope of the software application to be tested

This should be a paragraph or two that includes the project name and a high level description of what the software system/subsystem does and who it is being developed for.

**3. Software test environment**

Operating systems: Microsoft Windows 10 x64

Compiler: N/A, Ruby on Rails require no compiler, simply update the file and it will instantly update to the server.

Code Auditors: Sublime

Dynamic Test Analyzers: Dustin, Mohammed

Drivers: All default Windows 10 drivers and all drivers required for internet access.

Hardware: PC with Ethernet or WiFi adapter, mouse, keyboard, monitor.

Software: latest version of Google Chrome, Ruby tools to start/stop the server if needed.

Problem Reporting System:

1. Initiate Problem Report – Tester has identified a problem and will submit a ticket into our google drive in the Problem Report Folder. He should notify developer’s if he wishes immediate action or the developer will check at their own schedule (twice a week).
2. Problem Report Confirmation – The developer will attempt to confirm the problem and stay in contact with the tester. He should contact other testers to confirm the problem and isolate the issue.
3. Problem Report Action – The developer will mark the topic as “WIP” and will be added to a list of bugs the developer should fix, the developer along with his team mates will assign a priority number for the topic.
4. Problem Report Closure - It is the developer’s responsibility to work on problems and keep the database of problem reports updated. Upon resolving he will mark it as “FIXED” and for problem reports deemed insignificant, he will mark it as “CLOSED”.

**4. Test Descriptions**

***System Performance Requirements, all of Business Flow section B-0:*** Tests will ensure that all GUI requirements are satisfied, CRUD performance is optimized in SQL, data formats are either accepted or rejected with error messages displayed, and help messages are displayed along each module to assist the user. Pass/ fail criteria for GUI portions will determine if all interactive and graphical portions of the system are working properly. Pass/ fail for performance of the system will determine if all read/write/update/delete actions take less than 5 seconds to execute. Pass/ fail for password validation will determine if passwords are in proper format, secure, and functioning correctly. Pass/ fail criteria for proper data formatting will determine if all input data text field accept the proper characters and display a message when the wrong characters are entered. Pass/ fail criteria for help messages will determine that all screens have help features and all fields have an explanation of what they are there for.

***Inventory Management, B-1 to include IF-1.1, IF-1.2, IF-1.3, IF-1.4, IF-1.6, and IF-1.7:*** This is the foundation of the Inventory Management module. This tests the functionality of the database where the inventory is stored. Queries based on what products are available will rely on this data so it is essential to the entire system. The type of testing will be equivalence class testing and the pass/fail criteria will be that the data is actually CRUD-able. The prime constraint, which is present in most of the requirements, is the unfamiliarity of the programming language used for the project. Another constraint is an inability to test for all cases since there are no clear boundaries with this type of data.

***Inventory Management, B-1 to include IF-1.5:*** Test the ability of the system to store each user as separate entities and to record each user’s activity. Since it is a security issue, both white box and black box testing will be done to ensure proper functionality. Constraints may limit the amount of users to be able to access certain areas of the system at one time to provide up-to-date information and avoid and double entries.

***Shipment Tracking, B-2 to include IF-2.1 - IF-2.8:*** Tests the functionality of the various aspects of the Shipment Tracking system and will be closely tied to the Inventory Management module and a bit of Trouble Ticket. Testing to the functionality of the database will be done with black box testing to ensure proper functionality of all Creation, Reading, Updating, and Deleting of data. The UI will be tested using an intuitive approach where all interactive properties are tested for proper functionality. A pass/ fail test will determine that all UI buttons/ checkboxes/ etc. link to the proper objects. They will also determine that all database queries result in data being displayed in the correct form and that data is stored correctly.

***Shipment Tracking, B-2 to include IS-2:*** Tests the ability of the system to display data in the desired format and to send the data to a document. A pass/fail test will determine if the data is sent to a text document (DOCX/RTF). The assumption is the availability of a word-processing application to open the document and the constraint is the actual application used to open the document.

***Order Entry, B-3 to include IF-3.6:*** Tests that upon changes to a shipped order a charge is applied. This section will be directly connected to the Shipment Tracking system. Simple black box equivalence tests will verify that the system works properly. Pass/fail criteria will indicate that changes to a shipped item automatically generate additional charges to the customer. Assumptions are that charges have been determined by the user.

***Order Entry, B-3 to include IF-3.1, IF-3.2, IF-3.7, IF-3.8, UI-3.1, and UI-3.2:*** These are the main component of the Order Entry system. This will test the functionality of the GUI portion of the module and whether the underlying database structure for the order form is working properly. The GUI will be tested using object equivalence tests for all the text fields and a basic intuitive approach will be used to test buttons/ checkboxes/ etc. Databases will be white box tested to ensure the code is functional and satisfies performance restrictions. Assumptions are that the user has information about the customer and date the payments were received. A constraint on this section is the time complexity for data look-ups and creations.

***Order Entry, B-3 to include IF-3.3 - IF-3.5:*** This portion is directly linked to the Inventory Management system. Tests will determine if the ordered item is in stock and automatically reserve items available or automatically output an expected date and generate a trouble ticket. Black Box equivalence tests will determine if the correct output is given and white box tests will help test the specifics of the database queries. A pass/ fail will check if the system correctly identifies available items and what steps to take after that is determined. Constraints will be look-up times and creation times of data.

***Order Entry, B-3 to include UI-3.3 and UI-3.4:*** Test the ability of the system to print order invoices and customer status reports. White box testing will be implemented to ensure that proper databases are queried. Pass/ fail criteria will determine that the correct data is displayed and printed by the user. Assumptions are that the user has proper means to print/ display documents.

***Trouble Ticket, all of Business Flow section B-4:*** Test the ability of the system to store, edit, read and generate reports about data stored in a database for the Trouble Ticket module. The type of testing would be black box equivalence testing and white box for the sorting and report generation portions of the system. An intuitive approach will be used to test all GUI portions of the system. Constraints will be the use of word-processing software to display and save reports.

**5. Data Recording, Reduction, and analysis**

As most of our project is relaying information, the majority of our testing will be based on sending out various forms of information and checking to see if the information was both received properly, and is available for use. When a user fills out an order form, we test to see if all the data successfully made it to the database, as well as checking to see if the all the data sent was tied together (as one order). We must also check to see if this information is usable by the other parts. The inventory management subsection must check to see if the item name and quantity is in stock, and pass information to the shipment tracking subsection if and when the requested items can be shipped. Any error in sending or receiving this information says that the test failed and we have to work out the bugs.

**6. Test Conditions and Schedule**

*See Schedule v1.1*

**7. Updated Requirements Matrix**

*See Requirements Matrix v1.3*