Stag Bar
Bar Inventory System
Iteration 2
Version .3
Due: 4/26/2016

Abstract

The purpose of this project is too create a user friendly bar inventory system. The system will allow a inventory manager to keep track of their inventory and monitor internal theft. The system will allow a manager to create custom drinks, in order to avoid calculating how much spirits, and what spirits, was used for cocktails each time inventory is done.

Team

Thomas Andrikus

Gregory Dudar

Samuel McAdams

Anthony Whitaker

Table of Contents

1.0 High Level Goals

1) An admin user should be able to change user permission level (from guest to admin and vice versa).
2) An admin user should be able to delete another user.
3) An admin user should be able to create new mixed drinks
4) Edit the contents of mixed drinks
5) Retire/reinstate a mixed drink
6) The system should take the contents of each drink in to account when calculating inventory and discrepancies.
7) An admin user should be able to take current inventory and enter into the system.
8) An admin user should be able to enter in the sales of drinks.

9) An admin user should be able to enter if there are any deliveries
10) The system should be able to take into account inventory, sales, and deliveries and generate an accurate report of discrepancies.
11) The user should be able to view and print a report of inventory and discrepancies.
12) An admin user should be able to add and new alcohol into the system.
13) An admin user should be able to retire an alcohol, if it is no longer carried by the establishment.

1.1 Deployment System

Currently the system will be compiled and executed using a Java IDE and build file. For final release, the software will be opened or installed through the use of an JAR file. Upon execution the software will communicate with a cloud based database rather then forcing the user to download a database locally. Using a cloud-based database will be less overhead with for the consumer.

2.0 Work Items

1) Create UI. We will need UI that will take initial information from the user (username/password) if the system is being used for the first time. UI is also need to allow a user to change permission level and delete user. UI is needed to retire alcohol. UI is needed to allow a user to generate a report based upon a date range. Report UI should detail the alcohol name, the amount in inventory, the amount sold, and the amount delivered. The report UI should also show if there is a discrepancy between sales and what is currently in inventory (maybe highlight discrepancies?).

2) Need the system to detect if the user is using the system for the first time, and call the necessary database methods. This includes creating the database, adding the necessary tables to the database and creating a new user. Also need the system to make the necessary calculations to discover the discrepancies for the report.
3) Need multiple database methods to take/give information from the UI. These including adding mix drinks and their ingredients, updating mixed drinks ingredients, retrieving said mix drink names and respective ingredients, storing sales data, storing delivery data, storing inventory data, and the retrieving the necessary information properly to allow the system to calculate and generate a report.

2.2 Assignments

Thomas Andrikus: Work on the UI for the new user options (delete, change permission level). Also set up the listeners to allow the system to make the changes to the database. Work on SSD's for iteration 2 and use case models.

Greg Dudar: Create the UI for entering in a new alcohol group, also UI for retiring a mixed drink. Make sure the work properly with the system. Work vision statement, and domain model.

Sam McAdams: Create the necessary methods to interact with database. Create the Iteration plan for iteration 2 and security document.

Anthony Whitaker: Finish up the UI that Thomas and Greg are not working on. Work on writing JUNIT test's. Take the information from the database to properly calculate the data needed for the report. Work on Test plan and design document.

2.3 Testing Criteria

Thomas Andrikus: The interface for deleting user, and changing permission, should pop up 100% of the time. If there is any issues with them storing in the database that will be handled by Sam.

Greg Dudar: The menu for creating the custom alcohol group, and retiring alcohol UI should show up 100% of the time once the button is pressed. The UI should tell the system to store the new group in the database 100% of the time. If the System doesn't respond properly with database Anthony will work on it. If the database storage is the issue Sam will work on it.

Sam McAdams: There should be no errors in the SQL syntax, the logic should be correct 99% of the time. This is assuming that the user has entered valid data. Try to make the database entry 90% idiot proof.

Anthony Whitaker: The UI first time setup should show up only if it is the first time the user is using the system. Other UI components should show 100% of the time. It should also correctly interact with the system. If there are any issue with the database Sam will work on them.

Evaluation Criteria

Given the limited number nature of the iteration one use cases and the limited, amount of variables, all tests should be at least >= 99%. At this current stage calculating the data for the report is the most critical, it is the crux of our system. If a user cannot get accurate data when generating the report then the system is useless. Our project is becoming more robust and feature rich. At this current stage our system should be fully functional, and almost bug free. It will still need real

world testing, and user feedback however to be fully successful We should however be able to show the vision and potential for our software, and show that we have accomplished the goals of our system.