Stag Bar
Bar Inventory System
Iteration 2
Version .2
Due: 3/31/2016

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) The User should be able to change their password.
2) An admin user should be able to delete another user or themselves.
3) An admin user should be able to create a new guest user.
4) An admin user should be able to create custom groups for alcohol (IE. Craft Beer, Domestic Beer).
5) A user should be able to enter in inventory for alcohol.

1.1 Deployment System

Currently the system will be compiled and executed using a Java IDE and build file. The build file will compile and run the software. For final release, the software will be opened or installed through the use of an executable or JAR file (TBA which will be used). 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) Need a method, to create custom database schema, to populate the database with the necessary tables, and enter in the first user, upon starting the application for the first time.
2) Create UI. We will need UI for creating custom groups and adding inventory. Also will need UI added to the user options. These options will include changing password, deleting user, creating guest user, and changing permission level (guest/admin).

3) Need the system to be an intermediate between the user and the database. In the pervious iteration we had the user directly manipulating the database. Now we will have the system stand in between the user and database, to select, insert and modify data. The system will also maintain the connection for the user, using a master connection, rather then creating a custom connection login for the user.
4) Will need methods to allow the system to communicate with the database. We will methods to enter new alcohol into alcohol table, retrieve the alcohol from the table. Methods will also be needed to add a new user, with their permission level and password, into the user table. Will also need methods to remove users from the user table upon deletion and to update the users password in the user table. Since the design has changed former methods such for creating user, and getting connection will need to be modified to work with the new system design.

2.2 Assignments

Thomas Andrikus: Work on the UI for the new user options (delete, change password, etc). Also set up the listeners to allow the system to make the changes to the database. Work on Design Document for iteration 2.

Greg Dudar: Create the UI for entering in a new alcohol group, also make sure the work properly with the system. Work on Use Case model and update vision statement for iteration 2.

Sam McAdams: Create database and the necessary methods to interact with database. Also modify the former methods to work with the new system design. Create the Iteration plan for iteration 2.

Anthony Whitaker: Create the UI for entering in new inventory. Also set up the system to work in the way in which was discussed (using singleton design pattern). Work on SSD's for iteration 2.

2.3 Testing Criteria

Thomas Andrikus: The interface for deleting user, and changing password, 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 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. For this iteration we will assume data is valid and work on "idiot proofing" the entered data in iteration 3.

Anthony Whitaker: The UI for the entering of new inventory should pop 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 creating entering a new alcohol group and entering in new inventory is the most critical. If a user cannot create a custom group or enter in inventory then any future use cases will not be able to be implemented. Our project is becoming more robust and feature rich. We still have to work on doing the most important feature of showing discrepancies in sales VS. inventory. However it would be impossible to do this with out the current features we are implementing. We should however be able to show the vision and potential for our software, and garner a favorable response from the viewers of our demo.