Wait-Less

Group 20 - Hongcheng Wu, Spancer Guo, Jiajie Lin, Zachary Flebbe

Wait-Less is an application created to optimize a restaurant's efficiency by facilitating worker communication with an easy to user interface. Through the creation of different modes (manager, host and server), essential restaurant tasks are separated among employees and information is distributed accordingly. All information is stored in a database to be accessed across multiple devices. Privileged information (such as daily sales/statistics) is protected and only accessible if login type is correct. The manager can generate employee schedules and check restaurant statistics; the manager will also be able to operate as a host or server situationally. The host can generate a wait list for customers on location and a reservation list for call-ins. These lists include the customers name, party size and time of arrival (or current time) based on reservation/wait classification; this information can be sorted by time or party size. Once a party is ready to be seated, the table is assigned to a server and removed from the list. The server can access their assigned tables, order/remove food items, add comments to orders and check out a table.

Unit Testing

Main Testing:

The main testing method ensures that the 3 different user modes (manager, host, server) are accessible via the main menu. To access a specific mode, the user's login information must be present on the database coupled with the proper login type. If the information is not valid, the user is once again prompted to enter their information. The manager's login information should also be able to access all 3 modes if the manager needs to assist other employees. Testing will require that a valid manager login can successfully access all 3 modes, a valid host login must access the host mode, and a valid server login must access the server mode. Once the information is validated, the user should be redirected to the proper mode.

Manager Mode:

Manager mode testing requires that revenue statistics be correct and that a generated schedule follows time constraints (00:00 – 23:59). Since revenue values are stored on the database, several orders' totals are added to the day's revenue and the total is printed. Although menu and item prices are pre-set, the daily revenue must not accept negative table totals. This is to ensure that revenue cannot be manipulated by lower level employees. The schedule is generated via a drag and drop GUI, so all accessible time values for a shift are within the constraints of a normal day (00:00 – 23:59). However, the shift must have a valid name (> 1 character), otherwise an error message is thrown; this is checked before a shift is added to the schedule. To test this constraint, a shift is entered with 0 characters in the employee name. If this test is correct, the shift will not be entered into the schedule and the system will prompt the user with an error message. When testing the revenue total, we must attempt to close a tab with a negative total. This total will be subjected to a simple constraint (tab >=0), which upon failure will not be accepted by the database. In this event, the manager will be notified. Zero values are accepted in the event that a meal is compensated, or restaurant credit is used.

Host Mode:

Host mode testing requires that the two types of lists (waitlist/reservation) can successfully add and remove customers in real time. When adding an item to the list, the system will ensure that all the required fields are filled in the correct format. This includes the customer's name, the party size, current date and a timestamp. The name must be > 0 characters, the party size must be an integer >= 1, the date must be in the correct format (MM/DD/YYYY) and the time must be within the correct restraints (00:00 – 23:59). If these requirements are not met, the list item will not be added, and the host will be prompted to enter or correct the invalid information. To test this, adding a list item will be attempted in several scenarios: without a name, without a party size, without a date, without a timestamp, a party size of 0, a time outside of the constraint (i.e. 24:01) and an incorrectly formatted date (i.e. noninteger; 0.4/8.9/20RF). Each of these cases should result in an error prompt being shown. When removing a list item, the user is prompted to enter a customer's information; when this information is entered, the system will attempt to find the customer on the corresponding list. If found, the customer will be removed from the list and the list updated accordingly. Otherwise, the system will prompt the host that no such customer is found and nothing will be removed from the list.

Server Mode:

Server mode requires that a server have the ability to open a new table, add and remove items from a tab, and close a table/calculate total. When opening or closing a table, the restaurant GUI must update accordingly for all servers. To open a table, the server is required to press a button below a valid (green) table. If the table is red, the server will be unable to open the table as it is already occupied. Upon successfully opening a table, the table is turned red when returning to the GUI. When testing, the button below a red table should redirect the server to the table's tab and the button below a green table will prompt the user to place an order. When a table is selected, the restaurant's menu is displayed on the left hand side in the form of buttons; to add an item, the server presses the corresponding menu item which is then added to the tab on the right. To remove an item, the item is selected on the tab and the 'remove' button is pressed. Testing will require pressing several menu items and ensuring they are added to the tab, and then removing an item and ensuring it is removed. After pressing the 'total' button, the total should reflect the total of all of the items on the tab and be sent to the database.

Database:

Information for a Wait-Less restaurant is held on an SQL database on the local server. The information stored includes a list of employee names/passwords and total revenue for every day the restaurant is open. The database must have a successful connection/disconnection, username and password must be validated on login, and the total revenue must be returned to the manager. To ensure the connection is successful, the system will attempt to connect to the server. If an exception is thrown, the connection is unsuccessful and an error is thrown. The same is done when disconnecting from the database. When validating a login, the database is searched for a username and matching password and returns a boolean value of true or false.