Software Requirements Specification

Next-Gen Restaurant Application

December 4, 2016

Team Members:

- Sienna Guillory
- Dominic West
- Marilyn Monroe
- Orson Welles

GOTHAM CITY
UNIVERSITY

Gotham University - Astrobiology Department

ABLG 800: Human Beings and the Machines of Sunshine

Version History

Date	Document Version	Revision Description	Author(s)
September 4, 2016	2.0	Draft	Team
September 18, 2016	4.0	Added requirements, use cases, test cases, traceability matrix	Team
October 2, 2016	6.0	Refined requirements	Team
November 27, 2016	13.0	Refined requirements and test cases	Team
December 4, 2016	14.0	Document finalized	Team

Table of Contents

1. Introduction

1.1. Purpose: Mission Statement

Increase customer service through the efficient automation of consumer and staff interactions while collecting trend data that will offer stakeholders analytics to maximize operating efficiencies and reduce expenses.

1.2. Scope

The Next-Gen Restaurant Application (NRA) described within this specification will function as an automation tool to orchestrate customer seating and reservations in a restaurant environment. The design of the system shall provide restaurant specific coordination to optimize each consumer's dining experience while providing detailed analytics to help mature the business model and its processes. The system shall be responsible for satisfying the following operations:

1. Allow administrators to tailor a digital map rendering that represents a restaurant's table layout.
2. Maintain waiting queues of both walk-in customers and customers with reservations.
3. Provide notification to both consumers and host/hostess as tables become available.
4. Enable the system to accept data entry of orders and point-of-sales capabilities.
5. Offer kitchen staff with a display to coordinate order preparation.
6. Maintain order history
7. Provide workforce management features to optimize staff scheduling

9. Deliver reporting and analytics to help anticipate consumer volume, menu item consumption, and staff scheduling demand.

1.3. Definitions and Acronyms

- **PCI DSS:** Payment Card Industry Data Security Standard

1.4. References

- PCI Security Standards Council. (2016). *Payment Card Industry (PCI) Data Security Standard v3.2.* PCI Security Standards Council.

1.5. Regulations & Standards

1.5.1. The system shall meet PCI DSS requirements to adhere to merchant payment provider agreements.

2. Overall Descriptions

2.1. Product Perspective

We have decided to create a restaurant automation application that would allow a restaurant to operate a more efficient business. Restaurants tend to use a non-computerized system to conduct restaurant activities such as a pen/paper to keep track of orders and dry erase board to keep track of seat availability. Some restaurants may use a basic reservation system where a computer can keep track of seating availability and submit orders but does not have a way to automatically let customers know that their table is ready which results in a host or hostess to have to call out the customer’s name.

Our application will solve these issues and provide some additional functionality that would help restaurant owners make business decisions. Some features that the application would provide are: allow staff to schedule and coordinate their activities with each other, handle reservations, keep track of table status, and store or retrieve documents such as receipts. The application will also be able to leverage data analytics to anticipate traffic patterns and popularity of items on a menu.

2.2. Product Functions

- Allow customizing and organization of table map so the user can create a seating map that is reflective of the restaurants table layout
- Manage queues of customers and send out notifications to host/hostess and customer when a customer’s table is ready
- Allow placement of orders through system and the display of orders in a kitchen
- Generate receipts from orders and be able to store these receipts
- Tab support
- Staff scheduler support
- Keep track of reservations
- Analyze customer ordering habits in order to determine which items are popular or not
- Analyze traffic patterns in order to determine high/low traffic periods

2.3. Stakeholder Characteristics

NRA will have a wide range of users with varying levels of professional experience. Given that the system will need to support most ages that are employable, the system could have users as young as 15 and any maximum age. From management, to servers and hostesses, from the head chef to a kitchen helper, the level of education and computer literacy can be wide ranging. In addition, English may not be the primary language of every user. The following is an analysis of the different stakeholders to consider. This is then followed up by some rationale to identify the primary stakeholders and rank them in accordance to the system's impact on their role.

- Who is paying for the system?
  - Executive Team (Project Sponsors)
- Who is going to use the system?
  - Executive Team
  - Restaurant Management
  - Servers
  - Hosts/Hostesses
  - Kitchen Staff
  - Restaurant Management
  - Customer
- Who is going to judge the fitness of the system for use?
  - Executive Team
  - Restaurant Management
  - Servers
  - Hosts/Hostesses
  - Kitchen Staff
  - Restaurant Management
- What agencies (government) and entities (non-government) regulate any aspect of the system?
  - State Revenue Office (Sales Tax)
  - Payment Card Industry Security Standards Council
- What laws govern the construction, deployment, and operation of the system?

    - None

- Who involved in any aspect of the specification, design, construction, testing, maintenance, and retirement of the system?
    - Engineers
    - System Developers
    - System Designers
    - Quality Assurance Team
- Who will be negatively affected if the system is built?
    - Competitors
- Who else cares if this system exists or doesn't exist?
    - Vendors
    - Payment processing Provider
    - SMS Service Provider

Primary Stakeholder Identification and Ranking:

Stakeholder Class	Rank	Rationale
Executive Team (Project Sponsors)	3	● They are paying for the system and approving the high-level objectives
Customers	1	● Expects a high-level of customer service and order accuracy
Servers	2	● Expects order to be processed with a high degree of accuracy ● They are relying on the system to ease, not hinder, customer service so they do not have to deal with disgruntle customers
Hosts/Hostesses	4	● Wants an easy to use system to keep track of which tables are occupied
Kitchen Staff	6	● They want to know the order that come to them are accurate ● They want the system to assist in maintaining proper inventory
Restaurant Management	5	● They want accurate projections of shift volume in order to help properly schedule staff

2.4. Constraints

2.4.1. System shall append sales tax to every order

2.5. Assumptions and Dependencies

2.5.1. Payment process is dependent on external third-party payment processing service

3. Specific Requirements

3.1. Domain Requirements

3.1.1. The system will prevent a customer under the age of 21 from purchasing an alcoholic beverage.

3.1.2. The system will prevent all staff under the age of 21 from serving or handling alcoholic beverages.

3.2. System Requirements

3.3. Hardware

3.3.1. The system requires a touch screen display interface

3.4. Operating System

3.4.1. The system shall run on the Windows 10 operating system

3.5. Application Requirements

3.5.1. Point-of-Sale

3.5.1.1. The system shall allow staff to input orders using touch screen input.

3.5.1.2. The system shall present kitchen staff with a touch-based screen to complete order fulfillment

3.5.1.3. A status screen of all order item details for each table will be available to kitchen staff for all open orders.

3.5.1.4. The system shall allow checks to be split between any number of parties within a table's order.

3.5.1.5. The system shall account for gratuities made by customers on credit card transactions and associate those funds with the server so that payment is issued with the employee's payroll.

3.5.1.6. Order receipts shall be printed on with a maximum of 56 columns with a maximum print width of 2.83"

3.5.1.7. Customer credit card signature slip shall be printed on with a maximum of 56 columns with a maximum print width of 2.83"

3.5.1.8. Both a customer copy and a store copy of each credit card signature slip shall be printed for every credit card order.

3.5.1.9. The system shall allow customers to use the following forms of payment: cash, credit cards (Visa, MasterCard, American Express, Discover), and gift cards.

3.5.1.10. The system shall allow customers to keep a running tab on bar orders to be settled at the end of the customer visit.

3.5.1.11. The system shall pre-authorize credit card payment prior to extending a bar tab to a customer.

3.5.1.12. The system shall limit bar tabs from exceeding $300 without a manager's authorization.

3.5.2. Table Map Editing

3.5.2.1. The system shall allow authorized users to create a table map

3.5.2.2. The system shall allow free form placement of objects on table map

3.5.2.3. The system shall provide the creation of square and rectangle shapes

3.5.2.4. The system shall keep track of properties for each object

3.5.2.5. The system shall allow editing of each object’s properties

3.5.2.6. The system shall allow copying of objects in table map

3.5.2.7. The system shall allow deletion of objects in table map

3.5.2.8. The system shall allow the table map to be saved to a database

3.5.2.9. The system shall allow the table map to load a previously saved table map

3.5.2.10. The system shall track the occupancy of each table

3.5.3. Wait Queue

3.5.3.1. The system shall allow authorized users to add customers to a wait queue

3.5.3.2. The system shall keep track of Customer Name and Party size

3.5.3.3. The system can optionally keep track of customer’s phone number

3.5.3.4. The system shall send a notification over SMS when a customer is seated if a phone number is provided

3.5.3.5. The system shall provide a notification to host or hostess of next eligible customer for a table when a table is ready for seating

3.5.3.6. The system shall allow manual table assignment by host or hostess

3.5.3.7. The system shall allow removal of customer from wait queue

3.5.3.8. The system shall allow manual ordering of the wait queue

3.5.4. Data Analytics

3.5.4.1. The System database logging Protocol Shall Log Data for Customers.

3.5.4.2. System Ordered Items Protocol

3.5.4.3. System Customer Tracking Protocol

3.5.4.4. System Drinks Analysis Protocol

3.5.5. System Logging

3.5.5.1. The system shall log authentication successes and failures

3.5.5.2. The system shall log system failures which include syntax and runtime errors, connectivity problems, performance issues, third party service error messages and file system errors

3.5.5.3. The system shall store logs on the local file system

3.5.5.4. Each log entry shall include a timestamp with date and time, severity level, and event description

3.5.5.5. Log files shall be limited to one file per day

3.5.6. Reservation Management

3.5.6.1. Customers shall be able to make a reservation online.

3.5.6.2. Customers shall be able to view their own reservations after making them online.

3.5.6.3. Customers shall be able to call-in to make a reservation.

3.5.6.4. Customers shall be able to update the details of their reservation.

3.5.6.5. Customer shall be able to cancel their existing reservation at least 30 minutes before the previously assigned reservation time.

3.5.6.6. Customers shall be able to sign-up for and receive notifications for their existing or pending reservation.

3.5.6.7. The customer shall be able to Cancel an existing reservation.

3.6. Non-Functional Requirements

3.6.1. The system shall remain operational to serve customers in the absences of functioning internet access.

3.6.2. The system shall update the status of all pending reservations to cancelled when the reservation date has passed.

4. Accessibility

4.1. Access Restrictions

4.1.1. The user system shall be able to hear audio instructions for each functionality of the system.

4.2. Disabilities Accommodations

4.2.1. The system should not rely on any auditory cues to the user to elicit a response.

5. Security

5.1. Authentication

5.1.1. The system shall hash passwords

5.1.2. The system shall not store passwords in plain text

5.2. Authorization

5.2.1. The system shall allow administrators to assign role to users.

5.3. Payment Security

5.3.1. The system shall encrypt all stored payment card and bank account data

5.3.2. The system shall encrypt any transmission of payment card and bank account data traverses open, public networks.

7. Interfaces

7.1. User Interface

7.1.1. The system shall display currencies as USD

7.1.2. The user interface shall support touch screen input

7.1.3. The user interface shall provide a touch screen keyboard

7.1.4. The buttons on the user interface shall be optimized for touch screen layout

7.1.5. The response time after touching a button shall be no more than 1 second

7.2. Communication Interface

7.2.1. The system shall support the consumption of SOAP and RESTful web services.

7.2.2. The system shall enforce secured transport layer security (HTTPS) for external web requests

8. Test Plan

8.1. Test Cases

Test Case Index

Test Case ID	Name	Test Category	Priority
8.1.1	Table Order Entry Test	Functionality	High
8.1.2	Closeout Table Order (Credit Card) Test	Functionality	High
8.1.3	Closeout Table Order (Cash) Test	Functionality	Low
8.1.4	Append to Bar Tab Test	Functionality	Low
8.1.5	Close Bar Tab Test	Functionality	Low
8.1.6	Cancel Order Test	Functionality	High
8.1.7	Adding Table to Table Map Test	Functionality	Medium
8.1.8	Copying a Table while Editing Table Map Test	Functionality	Medium
8.1.9	Deleting a Table While Editing Table Map Test	Functionality	Medium
8.1.10	Save Table Map Test	Functionality	High
8.1.11	Load Table Map Test	Functionality	High
8.1.12	Toggle Table Occupancy Test	Functionality	Medium
8.1.13	Adding a Customer to Wait Queue Test	Functionality	High
8.1.14	Editing Table Properties Test	Functionality	High
8.1.15	Customer Receives Notification via SMS That Their Table is Ready Test	Functionality	High
8.1.16	Seating Customer Test	Functionality	High

8.1.17	Removing Customer from Wait Queue Test	Functionality	High
8.1.18	Moving Customer in Wait Queue Test	Functionality	High
8.1.19	Register for Reservation Notifications	Functionality	Medium
8.1.20	Track Customer Ordered Items	Functionality	High
8.1.21	Drinks Analysis System	Functionality	Medium
8.1.22	Customer Tracking System	Functionality	High
8.1.23	Customer Tracking System	Functionality	High
8.1.24	System Logging	System Maintainability	Medium
8.1.25	Authentication	Security	High
8.1.26	Add Reservation	Functionality	High
8.1.27	Update Existing Reservation	Functionality	High
8.1.28	Cancel an existing reservation	Functionality	Medium

8.1.1. Test Case: Table Order Entry

Test Case ID	8.1.1
Name	Table Entry Order Test
Description	Test order entry for a typical table order (non-bar or to-go order)
Execution Steps	1. Login to the system 2. Add a customer to the wait queue and seat them at a table 3. Select “ORDERS” from the top bar menu 4. Select “CREATE ORDER” from the Orders sub-menu 5. From the table drop down list, select the table assigned in step 2 6. Select “Dine In” as the Order Type 7. Select “Cab Filet Mignon” from the “Entree” menu category and click “Add to Order 8. Select “Surf & Turf” from the “Entree” menu category and click “Add to Order 9. Select “French Fries” from the “Side Order” menu category and click “Add to Order 10. ”Select “Broccoli” from the “Side Order” menu category and click “Add to Order” 11. Select “Iced Tea” from the “Beverages” menu category and click “Add to Order” 12. Select “Iced Tea” from the “Beverages” menu category and click “Add to Order” 13. Select “Iced Tea” from the “Beverages” menu category and click “Add to Order” 14. Right click on one of the Iced Teas in the Order Items and select “Remove Item” 15. Click “Submit Order” 16. View the order from the Edit Order screen to confirm the received order details reconcile with the order that was submitted.
Related Requirements	3.1.1, 3.1.2, 3.5.1.1
Related Use Cases	9.7, 9.12
Test Category	Functionality
Author	Orson Welles
Can be Automated?	Yes
Has been Automated?	Yes
Status	Executed

Remarks	Customer should already be added to the wait queue

8.1.2. Test Case: Closeout Table Order (Credit Card) Test

Test Case ID	8.1.2
Name	Closeout Table Order (Credit Card) Test
Description	Test case to closeout a table order by printing bill, processing a credit card payment, and issuing receipt.
Dependency	Test Case 8.1.1
Execution Steps	1. Login to system 2. Submit an order (see Test Case 8.1.1) 3. Select “ORDERS” from the top bar menu 4. Select “VIEW ORDERS” from the Orders sub-menu 5. Double click the Order Id of the order created in step 2 6. Click “Update: Preparing” to simulate the kitchen is in the process of fulfilling the order. 7. Click “Update: Ready for Pickup” to simulate the kitchen is ready for the server to pick up the order. 8. Click “Update: Served” to simulate the server has delivered the order to the customer. 9. Select “Pay Bill” 10. Select “Credit Card” radio button 11. Enter “4007000000027” in the Card Number textbox. 12. Enter “1018” in the Expiration (MMYY) textbox 13. Enter “123” in the Card Code textbox. 14. Enter “10.25” in the Gratuity textbox. 15. Click the “Accept” button 16. Click “Print Receipt” to send receipt to printer.
Related Requirements	2.5.1, 3.5.1.4, 3.5.1.5, 3.5.1.6, 3.5.1.7, 3.5.1.8, 3.5.1.9
Related Use Cases	9.8
Test Category	Functionality
Author	Orson Welles
Can be Automated?	Yes
Has been Automated?	Yes
Status	Executed

Remarks	

8.1.3. Test Case: Closeout Table Order (Cash) Test

Test Case ID	8.1.3
Name	Closeout Table Order (Cash) Test
Description	Test case to closeout a table order by printing bill, accepting a cash payment, and issuing receipt.
Dependency	Test Case 8.1.1
Execution Steps	1. Login to system 2. Submit an order (see Test Case 8.1.1) 3. Select “ORDERS” from the top bar menu 4. Select “VIEW ORDERS” from the Orders sub-menu 5. Double click the Order Id of the order created in step 2 6. Click “Update: Preparing” to simulate the kitchen is in the process of fulfilling the order. 7. Click “Update: Ready for Pickup” to simulate the kitchen is ready for the serve to pick up the order. 8. Click “Update: Served” to simulate the server has delivered the order to the customer. 9. Select “Pay Bill” 10. Select “Cash” radio button 11. Click the “Accept” button 12. Click “Print Receipt” to send receipt to printer.
Related Requirements	2.5.1, 3.5.1.4, 3.5.1.5, 3.5.1.6, 3.5.1.7, 3.5.1.8, 3.5.1.9
Related Use Cases	9.8
Test Category	Functionality
Author	Orson Welles
Can be Automated?	Yes
Has been Automated?	No
Status	Executed
Remarks	

8.1.4. Test Case: Append Order to Bar Tab Test

Test Case ID	8.1.4
Name	Append Order to Bar Tab Test
Description	This test case will add additional menu item to a tab and reinitiate a credit card preauthorization if the current balance exceeds the existing pre-auth.
Dependency	Test Case 8.1.3
Execution Steps	1. Log into the system 2. From the "Bar Menu" select "View Open Tabs" 3. Select the tab for "John Smith" 4. Select "Add to Tab" 5. From the Beverage menu category, select "Spirits" 6. Select "Macallan Scotch 12Y Single Malt" 7. In the "Special Order" section, select "Neat" 8. Select "Complete Order"
Related Requirements	3.1.1, 3.1.2, 3.5.1.1, 3.5.1.5, 3.5.1.6, 3.5.1.7, 3.5.1.8, 3.5.1.9, **3.5.1.10, 3.5.1.11, 3.5.1.12**
Related Use Cases	9.10
Test Category	Functionality
Author	Orson Welles
Can be Automated?	No
Has been Automated?	No
Status	Executed
Remarks	

8.1.5. Test Case: Close Bar Tab Test

Test Case ID	8.1.5
Name	Close Bar Tab Test
Description	This test case validates
Dependency	Test Case 8.1.3, Test Case 8.1.4
Execution Steps	1. From the "Bar" menu, selects "View Open Tabs" 2. From the list, select "John Smith" 3. Select "Close Tab", verify the tab displayed in the tab details 4. Select "Complete Payment", the receipts will be printed, verify the details on the receipt match the tab details
Related Requirements	3.5.1.1, 3.5.1.5, 3.5.1.6, 3.5.1.7, 3.5.1.8, 3.5.1.9, 3.5.1.10, 3.5.1.11
Related Use Cases	9.11
Test Category	Functionality
Author	Orson Welles
Can be Automated?	No
Has been Automated?	No
Status	Executed
Remarks	

8.1.6. Test Case: Cancel Order Test

Test Case ID	8.1.6
Name	Cancel Order Test
Description	This test simulate the cancelling of an order after it has been created
Dependency	Test Case 8.1.1
Execution Steps	1. Login to the system 2. Select “ORDERS” from the top bar menu 3. Select “CREATE ORDER” from the Orders sub-menu 4. From the table drop down list, select the first available table in the list. 5. Select “Dine In” as the Order Type 6. Select “Cab Filet Mignon” from the “Entree” menu category and click “Add to Order 7. Click “Submit Order” 8. View the order from the Edit Order screen 9. Click the “Cancel Order” button. 10. When prompted to confirm the cancellation, click “Yes”.
Related Requirements	3.5.1.2
Related Use Cases	9.12
Test Category	Functionality
Author	Orson Welles
Can be Automated?	Yes
Has been Automated?	No
Status	Executed
Remarks	

8.1.7. Test Case: Adding Table to Table Map Test

Test Case ID	8.1.7
Name	Adding Table to Table Map Test
Description	This test is to verify the functionality of adding a table object to the table map while in edit mode via the GUI
Execution Steps	1. Log into system 2. Click on "Seating" tab 3. Click on "Modify" sub tab 4. In Table Map Editor, click on "Add Table" button 5. Verify that a green square with the name "Default" has been created 6. Left click and hold on created table and move mouse around 7. Verify that the object is moved around the screen with the mouse
Related Requirements	3.5.2.1, 3.5.2.2
Related Use Cases	9.1
Test Category	Functional
Author	Dominic West
Can be Automated?	Yes
Has been Automated?	Yes
Status	Test case passed
Remarks	

8.1.8. Test Case: Copying a Table while Editing Table Map Test

Test Case ID	8.1.8
Name	Copying a Table while Editing Table Map Test
Description	This test is to verify the functionality of copying a table object in the table map while in edit mode via the GUI
Execution Steps	1. Log into system 2. Click on “Seating” tab 3. Click on “Modify” sub tab 4. In Table Map Editor, click on “Add Table” button 5. Verify that a green square with the name “Default” has been created 6. Right click on created table 7. Set table name to “Test Table 1” 8. Set table type to “Table Large” 9. Set table capacity to “8” 10. Set table shape to “Rectangle” 11. Set table size to “Large” 12. Click on “Ok” 13. Left click on the table to select it 14. Click on “Copy Table” 15. Verify that a green rectangle with the name “Test Table 1” has been created 16. Right click on the new copy and verify that table name is “Test Table 1”, table type is “Table Large”, table capacity is “8”, table shape is “Rectangle”, and table size is “Large”
Related Requirements	3.5.2.4, 3.5.2.6
Related Use Cases	9.2
Test Category	Functional
Author	Dominic West
Can be Automated?	Partially
Has been Automated?	Yes

Status	Test case passed
Remarks	Can test the functions that create copies of objects without going through the GUI, this part can be automated

8.1.9. Test Case: Deleting a Table While Editing Table Map Test

Test Case ID	8.1.9
Name	Deleting a Table While Editing Table Map Test
Description	This test is to verify the functionality of deleting a table object in the table map while in edit mode via the GUI
Execution Steps	1. Log into system 2. Click on “Seating” tab 3. Click on “Modify” sub tab 4. In Table Map Editor, click on “Add Table” button 5. Verify that a green square with the name “Default” has been created 6. Left click on the table to select it 7. Click on “Delete Table” button 8. Verify that the table has been deleted from the screen
Related Requirements	3.5.2.7
Related Use Cases	9.3
Test Category	Functional
Author	Dominic West
Can be Automated?	Yes
Has been Automated?	Yes
Status	Test case passed
Remarks	

8.1.10. Test Case: Save Table Map Test

Test Case ID	8.1.10
Name	Save Table Map Test
Description	This test is to test the save feature in table map system
Execution Steps	1. Log into system 2. Click on “Seating” tab 3. Click on “Modify” sub tab 4. In Table Map Editor, click on “Add Table” button 5. Move the first table in a desired location 6. In Table Map Editor, click on “Add Table” button 7. Move the second table in a desired location 8. In Table Map Editor, click on “Add Table” button 9. Move the third table in a desired location 10. Click on “Save Map” button 11. Click on “View” sub tab under “Seating” tab 12. Verify that the map is the same as what you have created
Related Requirements	3.5.2.8
Related Use Cases	9.1
Test Category	Functional
Author	Dominic West
Can be Automated?	Yes
Has been Automated?	Yes
Status	Test case passed
Remarks	

8.1.11. Test Case: Load Table Map Test

Test Case ID	8.1.11
Name	Load Table Map Test
Description	This test case is to test the load table map from storage function in table map system
Execution Steps	1. Log into system 2. Click on “Seating” tab 3. Click on “Modify” sub tab 4. In Table Map Editor, click on “Load Map” button 5. Verify that the map that has been loaded is the same that has been created in Save Table Map Test Case
Related Requirements	3.5.2.9
Related Use Cases	9.1
Test Category	Functional
Author	Dominic West
Can be Automated?	Yes
Has been Automated?	Yes
Status	Test case passed
Remarks	

8.1.12. Test Case: Toggle Table Occupancy Test

Test Case ID	8.1.12
Name	Dominic West
Description	This test is to verify that a table can be marked occupied or unoccupied
Execution Steps	1. Log into system 2. Click on “Seating” tab 3. Once table map is loaded in “View” tab, left click on a table 4. In pop-up window, Click on “Toggle Occupancy” 5. Verify that the table has turned red and the text turned white 6. Left click on the same table, and in the pop-up window, Click on “Toggle Occupancy” again 7. Verify that the table has turned green and the text turned black
Related Requirements	3.5.2.10
Related Use Cases	9.22
Test Category	Functional
Author	Dominic West
Can be Automated?	Yes
Has been Automated?	Yes
Status	Test case passed
Remarks	

8.1.13. Test Case: Adding a Customer to Wait Queue Test

Test Case ID	8.1.13
Name	Adding a Customer to Wait Queue Test
Description	This test to verify the functionality of adding customers to the wait queue
Execution Steps	1. Log into system 2. Click on “Wait Queue” tab 3. On new screen, click on “Add To Wait Queue” button 4. In customer name box, enter “John Tester” 5. In party size box, enter “2” 6. In phone number box, enter your phone number in format 123-456-7890 7. Click on “Ok” 8. Verify that the customer has been added to the list with the correct information entered and the column “Has Reservation” set to “False”
Related Requirements	3.5.3.1, 3.5.3.2, 3.5.3.3
Related Use Cases	9.4
Test Category	Functional
Author	Dominic West
Can be Automated?	Yes
Has been Automated?	Yes
Status	Test case passed
Remarks	

8.1.14. Test Case: Editing Table Properties

Test Case ID	8.1.14
Name	Editing Table Properties Test
Description	This test is to verify that the notification to the host/hostess screen displays when a customer is ready to be seated
Execution Steps	1. Log into system 2. Click on “Seating” tab 3. Click on “Modify” sub tab 4. In Table Map Editor, click on “Add Table” button 5. Verify that a green square with the name “Default” has been created 6. Right click on created table 7. Set table name to “Test Table 1” 8. Set table type to “Table Large” 9. Set table capacity to “8” 10. Set table shape to “Rectangle” 11. Set table size to “Large” 12. Click on “Ok” 13. Verify that the table name and table shape and table size has changed 14. Right click on created table 15. Verify that the table name, table type, and table capacity are the same
Related Requirements	3.5.2.3, 3.5.2.5
Related Use Cases	9.5
Test Category	Functional
Author	Dominic West
Can be Automated?	Yes
Has been Automated?	Yes
Status	Test case passed

Remarks	

8.1.15. Test Case: Customer Receives Notification via SMS That Their Table is Ready Test

Test Case ID	8.1.15
Name	Customer Receives Notification via SMS That Their Table is Ready Test
Description	This test is to verify the functionality of the SMS notification system if a customer provided a phone number
Execution Steps	1. Log into system 2. In wait queue screen, click on “Add Customer to Queue” 3. In the dialog box, enter “Jon Snow” as customer name and enter party size as 2 and phone number as your phone number and click “Ok” 4. In the wait queue list, select the row that contains customer “Jon Snow” 5. Click on button “Seat Customer” 6. Select “None” as table to be seated at 7. Make sure the “Send Text Message” box is set to “Yes” 8. Click “Ok” 9. Verify that the SMS has been sent
Related Requirements	3.5.3.4
Related Use Cases	9.6
Test Category	Functional
Author	Dominic West
Can be Automated?	Yes
Has been Automated?	Yes
Status	Test case passed
Remarks	

8.1.16. Test Case: Seating Customer Test

Test Case ID	8.1.16
Name	Seating Customer Test
Description	This test that the seat customer function correctly works
Execution Steps	1. Log into system 2. Click on “Wait Queue” tab 3. On wait queue screen, click on “Add To Wait Queue” button 4. In customer name box, enter “John Tester” 5. In party size box, enter “2” 6. Leave phone number box blank 7. Click on “Ok” 8. Click on the row that contains “John Tester” in the wait queue list 9. Click on “Seat Customer” 10. Verify that the customer name “John Tester” is displayed 11. Verify that Send Text Message box is set to “No” and is greyed out; unable to select yes 12. Select a table to seat the customer at 13. Click “Ok” 14. Click on “Seating” tab 15. Click on “View” tab if not selected already 16. Verify that the table chosen in step 12 is red with white text
Related Requirements	3.5.3.6
Related Use Cases	9.6
Test Category	Functional
Author	Dominic West
Can be Automated?	No
Has been Automated?	No
Status	Test case passed
Remarks	

8.1.17. Test Case: Removing Customer from Wait Queue Test

Test Case ID	8.1.17
Name	Removing Customer from Wait Queue Test
Description	This test verifies the ability to remove a customer from the wait queue
Execution Steps	1. Log into system 2. Click on “Wait Queue” tab 3. On wait queue screen, click on “Add To Wait Queue” button 4. In customer name box, enter “John Tester” 5. In party size box, enter “2” 6. Leave phone number box blank 7. Click on “Ok” 8. On wait queue screen, click on “Add To Wait Queue” button 9. In customer name box, enter “Abigail Tester” 10. In party size box, enter “4” 11. Enter phone number as “123-456-7890” 12. Click on “Ok” 13. Click on the row that contains “Jon Tester” 14. Click on “Quick Delete” 15. Verify that the row has been removed 16. Note the customer id for “Abigail Tester” 17. Click on “Remove from Queue” 18. Enter the customer id for “Abigail Tester” 19. Click on “Ok” 20. Verify that the row has been removed
Related Requirements	3.5.3.7
Related Use Cases	
Test Category	Functional
Author	Dominic West
Can be Automated?	Yes
Has been Automated?	Yes

Status	Test case passed
Remarks	

8.1.18. Test Case: Moving Customer in Wait Queue Test

Test Case ID	8.1.18
Name	Moving Customer in Wait Queue Test
Description	This test is to verify that a customer can be moved up and down in the wait queue
Execution Steps	1. Log into system 2. Click on “Wait Queue” tab 3. On wait queue screen, click on “Add To Wait Queue” button 4. In customer name box, enter “John Tester” 5. In party size box, enter “2” 6. Leave phone number box blank 7. Click on “Ok” 8. On wait queue screen, click on “Add To Wait Queue” button 9. In customer name box, enter “Abigail Tester” 10. In party size box, enter “4” 11. Enter phone number as “123-456-7890” 12. Click on “Ok” 13. On wait queue screen, click on “Add To Wait Queue” button 14. In customer name box, enter “Luna Tester” 15. In party size box, enter “8” 16. Leave phone number box blank 17. Click on “Ok” 18. Click on the row that contains “Luna Tester” 19. Click on “Move Up” button two times 20. Verify that the order from top to bottom is “Luna Tester”, “John Tester”, “Abigail Tester” 21. Click on the row that contains “John Tester” 22. Click on the “Move Down” button once 23. Verify that the order from top to bottom is now “Luna Tester”, “Abigail Tester”, “John Tester”
Related Requirements	3.5.3.8
Related Use Cases	9.21

Test Category	Functional
Author	Dominic West
Can be Automated?	Yes
Has been Automated?	Yes
Status	Test case passed
Remarks	A customer who has an existing reservation will be added to the wait queue from the reservation screen.

8.1.19. Test Case: Register for Reservation Notifications Test

Test Case ID	8.1.19
Name	Register for Notifications
Description	The user should be able to register to receive notifications in the most convenient format selected by the customer.
Execution Steps	1. Log into system 2. Select Main Menu 3. Select ‘Notifications’ 4. Select Get Notifications 5. Select the method by which the notifications are to be sent 6. When selected, the system will display a confirmation screen 7. User will select OK to confirm selection.
Related Requirements	3.5.7.7
Related Use Cases	9.20
Test Category	Functional, Performance
Author	Sienna Guillory
Can be Automated?	Yes
Has been Automated?	No
Status	Test case passed
Remarks	

8.1.20. Test Case: Track Customer Ordered Items

Test Case ID	8.1.20
Name	Track Customer Ordered Items
Description	The user should be able to track ordered items by the customer for all the products that are currently being sold at the restaurant.
Execution Steps	1. Customer walks into restaurant and is seated (either bar or casual dining). 2. Customer orders Drinks. (logging occurs within the system by the Employee portal). 3. Customer orders main dinner (logging occurs within the system by the Employee portal.). 4. Customer pays for meal/drinks . (logging is confirmed and gets sent to database).
Related Requirements	Must have database and hardware in place for functionality to work.
Related Use Cases	9.21
Test Category	Functional
Author	Marilyn Monroe
Can be Automated?	Yes, partially system can automate the entry of the data but manual entry by the user is required to gain variables.
Has been Automated?	No
Status	Execution Pending
Remarks	

8.1.21. Test Case: Drinks Analysis System

Test Case ID	8.1.21
Name	Drinks Analysis System
Description	The user should be able to track Drink’s being ordered, both Alcoholic and Nonalcoholic. Also Age should be identified and recorded.
Execution Steps	1. First the customer enters the restaurant and chooses either to be seated at the bar or at casual dining. 2. Customer is greeted by employee and asks to take drinks orders. 3. Customer then orders a drink and if asked if they will ordering menu items (excludes appetizers). 4. Employee will check ID if drinks are alcoholic. 5. Customer then will order drinks and appetizers and the drinks system will log their entry. 6. Once they have paid for their entry , it will be sent to the database.
Related Requirements	Employee must have logging hardware and System must be in place for automation to process.
Related Use Cases	9.22
Test Category	Functional
Author	Marilyn Monroe
Can be Automated?	Yes, partially
Has been Automated?	No
Status	Execution Pending
Remarks	

8.1.22. Test Case: Customer Tracking System

Test Case ID	8.1.22
Name	Customer Tracking System
Description	The System should identify amount of traffic generated in the restaurant for any given period.
Execution Steps	1. Customer walks in to the restaurant and is greeted by the front service staff. 2. Customer asks to be seated and the employee asks if they want to be seated at the bar or casual dining. (this will get logged). 3. Customer is seated and orders meal. 4. Customer pays for meal. (This gets logged in to the system and sent to the database too). 5. Weather Customer sits or not , the visit is calculated and sent to logging.
Related Requirements	The employee must be able to accurately log the customer and their entire group and match them with a unique telephone number or email address.
Related Use Cases	9.23
Test Category	Functional, Performance
Author	Marilyn Monroe
Can be Automated?	No, Must be manually inputted.
Has been Automated?	No
Status	Execution Pending

Remarks	

8.1.23. Test Case: Customer Tracking System

Test Case ID	8.1.23
Name	Customer Tracking System
Description	The System should identify amount of traffic generated in the restaurant for any given period.
Execution Steps	1. Customer walks in and asks for time to be seated. 2. Employee responds with a wait period of 1 hour. 3. Customer decixdes not to dine in. 4. Employee logs customer's information, including group size, and send it in to the database as a "non-seater due to wait period".
Related Requirements	Employee Training
Related Use Cases	9.22
Test Category	Functional, Performance
Author	Marilyn Monroe
Can be Automated?	No, This is a manual process.
Has been Automated?	No
Status	Execution Pending
Remarks	

8.1.24. Test Case: System Logging

Test Case ID	8.1.24
Name	System Logging
Description	Testing the system logging functionality
Execution Steps	1. See if a log file is created for current day on file system 2. File does not exist for current day, create a new file 3. Try to create a file again, a file should not be made 4. Submit a logging request (1, CURRENT_TIMESTAMP, “User admin logged in successfully) 5. Submit a logging request (1, CURRENT_TIMESTAMP, “User admin failed to log in) 6. Submit a logging request (5, CURRENT_TIMESTAMP, “Object circleTable failed to create) 7. Submit a logging request (3, CURRENT_TIMESTAMP, “Lost connection to server) 8. Verify that log file has been created successfully and that entries have been logged
Related Requirements	3.5.5.1, 3.5.5.2, 3.5.5.3, 3.5.5.4, 3.5.5.5
Related Use Cases	9.24
Test Category	Functional
Author	Dominic West
Can be Automated?	Yes
Has been Automated?	No
Status	Execution Pending
Remarks	

8.1.25. Test Case: Authentication

Test Case ID	8.1.25
Name	Authentication
Description	This test simulate authenticating to the system
Dependency	None
Execution Steps	1. Start the application 2. Click“Cancel”from the login prompt 3. Re-start the application 4. Click“Login” 5. Click“Ok”to close error message 6. Enter “WrongUserName”in the UserName field 7. Click“Login” 8. Click“Ok”to close error message 9. Enter “WrongPassword” in the Password field 10. Click“Login” 11. Click“Ok”to close error message 12. Enter “Orson” in the UserName field (a correct user name) 13. Click“Login” 14. Click“Ok”to close error message 15. Enter “Orson” in the Password field (a correct password) 16. Click“Login”
Related Requirements	5.1.1, 5.1.2,
Related Use Cases	None Defined
Test Category	Security
Author	Orson Welles
Can be Automated?	Yes
Has been Automated?	No

Status	Executed
Remarks	

8.1.26 Test Case: Add a Reservation

Test Case ID	8.1.26
Name	Add a reservation
Description	The user should be able to add a new reservation.
Execution Steps	1. Log into system 2. Select Reservations 3. Select ‘Add Reservation’ 4. Enter the required fields for the reservation 5. Select Create Reservation
Related Requirements	3.5.7.1, 3.5.7.3
Related Use Cases	9.13
Test Category	Functional, Performance
Author	Sienna Guillory
Can be Automated?	Yes
Has been Automated?	No
Status	Test case passed
Remarks	

8.1.27 Test Case: Update existing Reservation

Test Case ID	8.1.27
Name	Update an existing reservation
Description	The user should be able to update an existing reservation.
Execution Steps	1. Log into system 2. Select Reservations 3. Select ‘Find Reservation’ 4. Enter customer name or phone number. 5. Select the Reservation 6. Click update reservation 7. Change the information
Related Requirements	3.5.7.4
Related Use Cases	9.14
Test Category	Functional, Performance
Author	Sienna Guillory
Can be Automated?	Yes
Has been Automated?	No
Status	Test case passed
Remarks	

8.1.28 Test Case: Cancel existing Reservation

Test Case ID	8.1.28
Name	Cancel an existing reservation
Description	The user should be able to cancel an existing reservation.
Execution Steps	1. Log into system 2. Select Find Reservations 3. Enter the appropriate search term to find a reservation 4. Select the reservation 5. Select Update Reservation 6. Change the status to Cancelled 7. Click OK
Related Requirements	3.5.7.4
Related Use Cases	9.15
Test Category	Functional, Performance
Author	Sienna Guillory
Can be Automated?	Yes
Has been Automated?	No
Status	Test case passed
Remarks	

9. Appendix A: Use Cases

Use Case Index

Use Case ID	Name	Primary Actor	Scope	Complexity[1]	Priority
9.1	Adding Table to Table Map	Restaurant Management	In	Complex	Medium
9.2	Copying a Table while Editing Table Map	Restaurant Management	In	Complex	Medium
9.3	Deleting a Table While Editing Table Map	Restaurant Management	In	Average	Medium
9.4	Adding a Customer to Wait Queue	Host/Hostess	In	Complex	High
9.5	Prompting Host/Hostess that a customer's table in queue is ready	Host/Hostess	In	Complex	High
9.6	Customer Receives Notification via SMS That Their Table is Ready	Customer	In	Simple	High
9.7	Table Order Entry	Server	In	Complex	High
9.8	Closeout Table Order	Server	In	Complex	High
9.9	Open Bar Tab	Bartender	In	Average	Low
9.10	Append Order to Bar Tab	Bartender	In	Complex	Low
9.11	Close Bar Tab	Bartender	In	Average	Low
9.12	Fulfill Order	Kitchen Staff	In	Complex	High
9.13	View Existing Reservation	Customer	In	Average	Medium

[1] Complexity is derived from the transactional complexity detailed within the Software Complexity Estimation Document

9.14	Update Existing Reservation	Customer	In	Average	Medium
9.15	Cancel Existing Reservation	Customer	In	Complex	Medium
9.16	Register for Notifications for Existing Reservation	Customer	In	Complex	Medium
9.17	Track Customer Ordered Items	Customer	In	Average	High
9.18	Customer Tracking System	Customer	In	Average	High
9.19	Drinks Analysis System	Customer	In	Average	Medium
9.20	System Logging	System	In	Simple	Medium
9.21	Manual Ordering of Wait Queue	Host/Hostess	In	Average	Low

9.1. Use Case: Adding Table to Table Map

Use Case Title:	Adding Table to Table Map
Author:	Dominic West
Primary Actor:	Restaurant Management
Level:	Actor Goal
Precondition:	User has appropriate access for admin duties
Minimal Guarantees:	N/A
Success Guarantees:	Map layout will be saved for immediate use in table tracker
Related Requirement:	3.5.2.1, 3.5.2.2, 3.5.2.3, 3.5.2.5, 3.5.2.8

Main Success Scenario:

1. User clicks on “Seating Modify”
2. System loads the table map editor
3. User clicks on “Add Table” button
4. System creates a table
5. User clicks “Save” to save changes to map
6. System saves changed to the table map

Extension:

4a: User Modifies properties of table

1. User right clicks on table
2. System displays a window with fields allowing user to change properties
3. User enters table name, table type, table capacity, table shape, table shape size
4. User Clicks “Ok”
5. System saves changes to the table object
6. Use case continues at step 5 in Main Success Scenario

3a User Loads Existing Table Map

1. User clicks on “Load” to load the table map
2. The system loads the previously saved table map
3. Use case continues at step 3 in Main Success Scenario

9.2. Use Case: Copying a Table while Editing Table Map

Use Case Title:	Copying a Table while Editing Table Map
Author:	Dominic West
Primary Actor:	Restaurant Management
Level:	Actor Goal
Precondition:	User has appropriate access for admin duties
Minimal Guarantees:	N/A
Success Guarantees:	Map layout will be saved for immediate use in table tracker
Related Requirement:	3.5.2.4, 3.5.2.6
Main Success Scenario: 1. User clicks on “Seating Modify” 2. System loads the table map editor 3. User clicks on “Load” to load the last saved table map 4. System loads the last saved table map 5. User clicks on a table they want to copy 6. User clicks on “Copy Table” button 7. System creates a copy of the table Extension: N/A	

9.3. Use Case: Deleting a Table While Editing Table Map

Use Case Title:	Deleting a Table While Editing Table Map
Author:	Dominic West
Primary Actor:	Restaurant Management
Level:	Actor Goal
Precondition:	User has appropriate access for admin duties
Minimal Guarantees:	N/A
Success Guarantees:	Map layout will be saved for immediate use in table tracker
Related Requirement:	3.5.2.7

Main Success Scenario:

1. User clicks on “Seating Modify”
2. System loads the table map editor
3. User clicks on “Load” to load the last saved table map
4. System loads the last saved table map
5. User clicks on a table they want to delete
6. User clicks on the “Delete Table” button
7. System deletes the table from the table map

Extension: N/A

9.4. Use Case: Adding a Customer to Wait Queue

Use Case Title:	Adding a Customer to Wait Queue
Author:	Dominic West
Primary Actor:	Host/Hostess
Level:	Actor Goal
Precondition:	User has appropriate authorization access
Minimal Guarantees:	N/A
Success Guarantees:	A Customer will be successfully added in a queue
Related Requirement:	3.5.3.1, 3.5.3.2, 3.5.3.3

Main Success Scenario:

1. User clicks on “Add to Queue”
2. System displays a new window with fields to add customer
3. User fills out required customer name and party size
4. User optionally fills out customer mobile phone number
5. User clicks on “Ok” once finished
6. System adds customer to wait queue and provides estimated wait time

Extension:

3a User fails to fill out customer info

1. User clicks “Ok” without filling out customer name and party size
2. System prompts user that customer name and party size is required to add a customer to a queue
3. User clicks “Ok” on prompt
4. User fills out customer name and party size
5. Use case continues at step 4 in Main Success Scenario

4a User fails to properly enter a phone number

1. User enters an incorrect phone number
2. System prompts user that they must enter a phone number in the format 123-456-7890
3. User enters a proper phone number
4. User case continues at step 5 in Main Success Scenario

9.5. Use Case: Editing Table Properties

Use Case Title:	Prompting Host/Hostess that a customer's table in queue is ready
Author:	Dominic West
Primary Actor:	Restaurant Management
Level:	Actor Goal
Precondition:	User has appropriate authorization access, User is already editing a table map
Minimal Guarantees:	N/A
Success Guarantees:	Restaurant Management will be able to edit the table properties for a selected table
Related Requirement:	3.5.2.3, 3.5.2.5

Main Success Scenario:

1. User right clicks on a table they want to change the properties of
2. System displays a pop up dialog with fields to edit Table Name, Type, Capacity, Shape, Shape Size
3. User enters the desired selections
4. User clicks "Ok"
5. The system saves the changes to the table object

Extension:

N/A

9.6. Use Case: Seat Customer from Wait Queue

Use Case Title:	Customer Receives Notification via SMS That Their Table is Ready
Author:	Dominic West
Primary Actor:	Host/Hostess, Customers
Level:	Actor Goal
Precondition:	User has SMS capable phone and gave host/hostess their phone number when put on wait list, Authorized user is using the system
Minimal Guarantees:	N/A
Success Guarantees:	A customer will be seated at a table
Related Requirement:	3.5.3.4, 3.5.3.6

Main Success Scenario:

1. User selects a customer to seat from the wait queue list
2. User clicks on “Seat Customer” button
3. System displays a pop up window showing customer name, a list of available tables to be seated at, and option to send a text message if a phone number is provided
4. User selects a table to be seated at from the list of available tables
5. User clicks on “Ok”
6. System removes customer from wait queue list, assigns customer to selected table, and selected table is marked as occupied in the seat map

Extension:

4a SMS Notification to Customer

1. User checks “Yes” for Send Test Message
2. User clicks on “Ok”
3. The system sends a text message to phone listed in the customer’s profile
4. Use case continues in Main Success Scenario step 6

9.7. Use Case: Table Order Entry

Use Case Title:	Table Order Entry
Author:	Orson Welles
Primary Actor:	Server
Level:	User Goal
Precondition:	The user is logged into the system with access to ordering subsystem.
Minimal Guarantees:	If there is a failure, the error and trace information is logged
Success Guarantees:	The order is entered into the system and submitted to the kitchen for fulfillment
Related Requirements:	3.1.1, 3.1.2, 3.5.1.1

Main Success Scenario:

1. Select the table for the order
2. Select “New Order” from the menu
3. Select the number of people at the table
4. Select first customer in the order from the seating chart
5. Select “Add new item”
6. Select the menu category for the item to be ordered
7. Select the menu item to add to the order
8. Return to step 5 until all order items have been entered.
9. Return to step 4 for all subsequent customers for this order
10. Select “Submit Order” button
11. Order status is updated from “New” (Default) to “Submitted”

Extension:

7a: Enter any special order details
7b: Customer age verification if alcoholic beverage is ordered
7c: Server age verification if alcoholic beverage is ordered

9.8. Use Case: Closeout Table Order

Use Case Title:	Closeout Table Order
Author:	Orson Welles
Primary Actor:	Server
Level:	User Goal
Precondition:	The server is logged in, an order has been placed, the order was fulfilled, and the customer is created to close out their check
Minimal Guarantees:	If there is a failure, the error and trace information is logged
Success Guarantees:	The order is totaled, a check is issued, and payment is processed.
Related Requirement:	1.5.1, 3.5.1.4, 3.5.1.5, 3.5.1.6, 3.5.1.7, 3.5.1.8, 3.5.1.9

Main Success Scenario:

1. Select the table for the order from the table layout screen
2. Select “Create Bill” from the context menu associated with the table
3. After reviewing the bill, selects “Print” and the bill is printed and delivered to the customer’s table.
4. The bill is collected from the table and the table is selected from the table layout screen
5. Select “Pay Bill” from the context menu associated with the table, a list of generated bills will be displayed
6. Select the appropriate bill
7. A list of available forms of payment is displayed, select the appropriate form of payment
8. Select “Process Payment
9. Select “Complete Payment”, the receipts will be printed
10. Order Status is updated from “Served” to “Paid”

Extension:

2a: If the customer wishes to slip the bill the Server selects “Split Bill”
- 2a1: From the table seat chart, the server selects which customers will be included in the first bill.
- 2a2: Selects “Finalize Bill”
- 2a3: If all the seats have not yet been assigned to a bill, an additional bill is created and server returns to step 2a1

7a: If the form of payment select is Credit Card
- 7a1: Enter the credit card number, expiration date, and CVV2 code
- 7a2: Select “Process Credit Card”

7b: If the form of payment select is Gift Card
- 7b1: Enter the gift card number and security code

8a: The cash drawer will open if credit card was not selected, deposit the cash into the drawer

9.9. Use Case: Open Bar Tab

Use Case Title:	Open Bar Tab
Author:	Orson Welles
Primary Actor:	Bartender
Level:	User Goal
Precondition:	The user is logged into the system with access to bartending subsystem.
Minimal Guarantees:	If there is a failure, the error and trace information is logged
Success Guarantees:	Customer information is saved to the system, payment is pre-authorized, and is opened so bar order can be appended to it.
Related Requirement:	3.5.1.1, 3.5.1.5, 3.5.1.6, 3.5.1.7, 3.5.1.8, 3.5.1.9, **3.5.1.10, 3.5.1.11**

Main Success Scenario:

1. From the “Bar” menu, selects “New Tab”
2. Enter Customer name and ID Information
3. Select “Save”
4. Selects “Add Credit Card” information
5. Enters credit card number, expiration date, and CVV2 code
6. Select “Save”
7. Select “Pre-Authorize” to process a Pre-Auth transaction on the credit card

Extension:

None

9.10. Use Case: Append Order to Bar Tab

Use Case Title:	Append Order to Bar Tab
Author:	Orson Welles
Primary Actor:	Bartender
Level:	User Goal
Precondition:	The user is logged in, has security access role allowing access to bartending functionality, a tab has been created, and the tab status is “Open”
Minimal Guarantees:	If there is a failure, the error and trace information is logged
Success Guarantees:	An order is associated with an open tab and the tab balance is updated to reflect the additional order items
Related Requirement:	3.1.1, 3.1.2, 3.5.1.1, 3.5.1.5, 3.5.1.6, 3.5.1.7, 3.5.1.8, 3.5.1.9, **3.5.1.10, 3.5.1.11, 3.5.1.12**

Main Success Scenario:

1. From the “Bar” menu, selects “View Open Tabs”
2. From the list, select the tab by customer’s name that should be appended with additional order items.
3. Select “Add to Tab”
4. Select the item category of the menu item to be added
5. Select the menu item to be added
6. Select “Add Item”
7. Return to step 4 for any additional items that should be added.
8. Select “Complete Order”

Extension:

5a: If item customizations are required
5a1: In the Special Order section, specify any special order customization for the selected menu item
8a: If the total order exceeds the existing credit card pre-authorization
8a1: An error is displayed: “Balance exceeds what has been authorized”
8a2: User is prompted if they would like to create a new pre-authorization to cover the existing amount
8a2a: If the user selects “Yes”, a pre-auth transaction is submitted
8a2b: If the user selects “No”, the order is cancelled

9.11. Use Case: Close Bar Tab

Use Case Title:	Close Bar Tab
Author:	Orson Welles
Primary Actor:	Bartender
Level:	User Goal
Precondition:	The user is logged in, has security access role allowing access to bartending functionality, a tab has been created, the tab status is “Open”, and menu items have been added to the tab
Minimal Guarantees:	If there is a failure, the error and trace information is logged
Success Guarantees:	The total balance of the tab is charged to the customer’s credit card and the Tab Status is updated to “Closed”
Related Requirement:	3.5.1.1, 3.5.1.5, 3.5.1.6, 3.5.1.7, 3.5.1.8, 3.5.1.9, **3.5.1.10, 3.5.1.11**

Main Success Scenario:

1. From the “Bar” menu, selects “View Open Tabs”
2. From the list, select the tab by customer’s name of the tab to close.
3. Select “Close Tab” Tab details are displayed
4. Select “Complete Payment”, the receipts will be printed

Extension:

None

9.12. Use Case: Fulfill Order

Use Case Title:	Fulfill Order
Author:	Orson Welles
Primary Actor:	Kitchen Staff
Level:	User Goal
Precondition:	The user is logged in, has security access role allowing access to the food preparation functionality, an order has been created, and the Order Status is “Submitted”
Minimal Guarantees:	If there is a failure, the error and trace information is logged
Success Guarantees:	Success Guarantees Here
Related Requirement:	3.5.1.2

Main Success Scenario:

1. User is alerted by a flashing new submitted order on the “Orders” screen
2. Select the order to be prepared
3. Select “Prepare”
4. Order Status is updated from “Submitted” to “Preparing”
5. When order preparation is complete, select “Ready for Pickup”
6. Order Status is updated from “Preparing” to “Ready for Pickup”
7. The Server picks up the order and selects “Served”
8. Order Status is updated from “Ready for Pickup” to “Served”

Extension:

None

9.13. Use Case: Register for Notifications for Existing Reservation

Use Case Title:	Register for Existing Reservations Notification
Author:	Sienna Guillory
Primary Actor:	Customer
Level:	Customer
Precondition:	User is able to access the system and their account using a Mobile or Desktop application or have access to a mobile device.
Minimal Guarantees:	User will be able to view their existing reservation.
Success Guarantees:	User will be able to view and get notifications for their reservation.
Related Requirement:	3.5.7.6

Main Success Scenario:

1. User will log into application
2. User will select the ‘main menu’ to view list of options to choose from
3. User will select ‘Notify Me when Reservation Details Change’
4. User will view a list of reservations and times
5. User will be able to select the reservation to be notified of.
6. The system will allow the user to confirm that they would like to be notified.
7. Upon confirmation the user will be asked to select the medium through which they would like to be notified.
8. The user will select the appropriate medium from the list.
9. If applicable, the system will ask for SMS phone number if the user has selected SMS but no phone number exists in the system or the user would like to use a different phone number for notifications.
10. The system will have the user confirm that the phone number is correct.
11. The user will then be sent a confirmation code.
12. Confirmation code will then be entered by the user when prompted.
13. System will verify code and accept the entered phone number.

Extension: None

9.14. Use Case Track Customer Ordered Items

Use Case Title:	Track Customer Ordered Items
Author:	Marilyn Monroe
Primary Actor:	Computer System - Customers
Level:	Owner Goal
Precondition:	The User/Employee is logged in and is able to place a name with a order.
Minimal Guarantees:	Must have a log of the user along with their unique identifiers (email or phone number), and the employee id of the entry.
Success Guarantees:	There must be a way to backtrace if their is a cancellation in order or if there is a low rating.
Related Requirement:	3.5.4.1

Main Success Scenario:

1. Step 1 - The Employee seats the customer(s) in their table.
2. Step 2 - The Employee asks for drinks, and then logs them into the system.
3. Step 3 - The Employee asks for their order , and then breaks it down further by appetizer, meal, and dessert.
4. Step 4 - Once the Employee takes the bill and process the payment, the entire order is sent to the database for logging.
5. Step 5 - A survey is sent to the customer regarding the food quality.

Extension:

None

Remarks: The requirement is to identify the most ordered items on the menu and identify satisfaction ratings to understand what items should or should not be removed from the menu.

9.15. Use Case: Customer Tracking System

Use Case Title:	Customer Tracking System
Author:	Marilyn Monroe
Primary Actor:	Customers
Level:	Owner Goal
Precondition:	Must have a way to identify Customers who visit the Restaurant and Customers who decide to Dine after arriving.
Minimal Guarantees:	System must be able to track how many people approach the front of the restaurant to be seated.
Success Guarantees:	System must catch all who were actually seated vs who ju st inquired.
Related Requirement:	3.5.4.2

Main Success Scenario:

1. Step 1 - Customers will arrive at the restaurant and inquire for seating. The Employee must log the customers total count as visited.
2. Step 2 - If said customers decide to be seated and ARE seated, then they must log the customers (email/phone number) and total count of groups that were seated.
3. Step 3 - This information must be submitted after the entire meal is done and the payment has been processed.
4. Step 4- If they decide to leave before payment is processed or decide not to consume, then there must be a way to log the information with a flag that states they decided to leave.

Extension:

None.

Remarks: The requirement is identify how many Customers ACTUALLY visited the restaurant.

9.16. Use Case: Drinks Analysis System

Use Case Title:	Drinks Analysis System
Author:	Marilyn Monroe
Primary Actor:	Customers ordering Drinks
Level:	Owner Goal
Precondition:	Must be able to house drinks information.
Minimal Guarantees:	Have a way to log various drinks by code and frequency.
Success Guarantees:	None.
Related Requirement:	3.5.4.3

Main Success Scenario:

1. Step 1 - Customer walks into restaurant and decides to sit either by the bar or casual dining (this must also be captured in the previous Use Case).
2. Step 2 - Customer decides to order alcoholic drink while other members of his/her group decide non-alcoholic.
3. Step 3 - Customer pays for total drinks.
4. Step 4 - Logs system for what drinks were ordered.

Extension:

None.

Remarks: The requirement is to identify how many Alcoholic drinks are being sold vs Non-Alcoholic. Also to understand ordering more supplies.

9.17. Use Case: System Logging

Use Case Title:	System Logging
Author:	Dominic West
Primary Actor:	System
Level:	System Goal
Precondition:	None
Minimal Guarantees:	A new log file will be created if one has not for the current day
Success Guarantees:	Event written to log file
Related Requirement:	3.5.5.1, 3.5.5.2, 3.5.5.3, 3.5.5.4, 3.5.5.5

Main Success Scenario:

1. System submits a logging request to log subsystem
2. Log subsystem writes request to system log file with specified severity level, timestamp at the moment log entry is written, and the specified description

Extension:

1a. Log file does not exist
1. System creates a new log file in format YYYY_MM_DD_SYSTEM_LOG.txt
2. Use case continues at step 2

Remarks: The requirement is to identify how many Alcoholic drinks are being sold vs Non-Alcoholic. Also to understand ordering more supplies.

9.18. Use Case: Add Reservation

Use Case Title:	Add Reservation
Author:	Sienna Guillory
Primary Actor:	Customer
Level:	Customer Goal
Precondition:	None
Minimal Guarantees:	The customer must be able to create a reservation.
Success Guarantees:	The customer will get first priority when in wait queue.
Related Requirement:	3.5.7.1, 3.5.7.3

Main Success Scenario:

1. Customer submits a reservation request
2. Reservation information is saved and displayed to the customer.

Extension:

Remarks: This requirement is to allow the convenience of making a reservation for advance addition to the wait queue upon arrival in the restaurant.

9.19. Use Case: Update Reservation

Use Case Title:	Update Reservation
Author:	Sienna Guillory
Primary Actor:	Customer
Level:	Customer Goal
Precondition:	A reservation must already exist.
Minimal Guarantees:	The customer must be able to update a reservation.
Success Guarantees:	The customer will not have to show up to be served.
Related Requirement:	3.5.7.4, 3.5.7.5

Main Success Scenario:

1. Customer performs a search to find the existing reservation.
2. Once the reservation is identified, customer will select it.
3. Customer will click Update Reservation
4. Customer will then change the reservation detail they wish to change
3. Reservation information is saved and displayed to the customer.

Extension:

Remarks: This requirement is to allow the convenience of updating a reservation for advance addition to the wait queue upon arrival in the restaurant.

9.20. Use Case: Cancel a Reservation

Use Case Title:	Cancel a Reservation
Author:	Sienna Guillory
Primary Actor:	User
Level:	User Goal
Precondition:	A reservation must already exist.
Minimal Guarantees:	The customer must be able to update a reservation.
Success Guarantees:	The customer will not have to show up to be served.
Related Requirement:	3.5.7.7

Main Success Scenario:

1. User performs a search to find the existing reservation.
2. Once the reservation is identified, User will select it.
3. User will click Update Reservation
4. User will then change the reservation status to Cancelled.
5. Reservation information is saved and displayed to the User.

Extension:

Remarks: This requirement is to allow the convenience of updating a reservation the was cancelled by the customer.

9.21. Use Case: Manual Ordering of Wait Queue

Use Case Title:	Manual Ordering of Wait Queue
Author:	Dominic West
Primary Actor:	Host/Hostess
Level:	User Goal
Precondition:	User has appropriate access, There are already customers in the wait queue
Minimal Guarantees:	The order of wait queue will be preserved
Success Guarantees:	The order of the wait queue has been successfully changed
Related Requirement:	3.5.3.8

Main Success Scenario:

1. User selects a row that contains the customer they want to move
2. User selects “Move Up” button
3. The system moves the customer up in the queue by 1 and preserves the order in the rest of the queue

Extension:

2a Moving Down

1. User selects “Move Down” button
2. The system moves the customer down in the queue by 1 and preserves the order in the rest of the queue