9/30/2016

V-Menu Restaurant Management System

Building the Restaurant of Tomorrow Today

**School:** University of North Florida

**Course:** Software Engineering (CEN 4010)

**Instructor:** Dr. Douglas Leas

**Leader Developer:** Matthew Boyette

**Lead Designer:** David Hughes

**Developer:** William Mejia

**Developer:** Demetrius Myers

**Developer:** Slaven Popadic

Table of Contents

[Overview 2](#_Toc463042535)

[Project Team 2](#_Toc463042536)

[High-Level Use Case Diagram 3](#_Toc463042537)

[Use Case Summary 4](#_Toc463042538)

[Use Case: View Customer Reports 5](#_Toc463042539)

[Use Case: View Kitchen Status 6](#_Toc463042540)

[Use Case: View Manager Reports 7](#_Toc463042541)

[Use Case: View Menu 8](#_Toc463042542)

[Use Case: View Order Status 9](#_Toc463042543)

[Use Case: View Universal Status 10](#_Toc463042544)

[Use Case: View Wait Status 11](#_Toc463042545)

[Use Case: Bulk Data Import/Export 12](#_Toc463042546)

[Use Case: Contact the Owner 13](#_Toc463042547)

[Use Case: Create/Edit Events/Specials 14](#_Toc463042548)

[Use Case: Create/Edit Menu 15](#_Toc463042549)

[Use Case: Place/Confirm Order 16](#_Toc463042550)

[Use Case: Register Account 17](#_Toc463042551)

[Use Case: Search Restaurant Network 18](#_Toc463042552)

[Use Case: System Configuration 19](#_Toc463042553)

[Use Case: Cancel Order 20](#_Toc463042554)

[Use Case: Flag Order As Completed 21](#_Toc463042555)

[Use Case: Flag Order As Delivered 22](#_Toc463042556)

[Use Case: Request Help 23](#_Toc463042557)

[Non-functional Requirements 24](#_Toc463042558)

# Overview

V-Menu is a system designed to turn the day-to-day logistics of managing a restaurant into a problem solvable using distributed computing. By turning the individual human elements of the restaurant into computable problems, we hope to push the boundaries and take the next step toward a completely automated restaurant. Although a fully automated restaurant is still relatively infeasible with today’s level of off-the-shelf consumer technology, it won’t be long before advances in robotics make such an occurrence not only practical, but also commonplace. There are three primary factors of functionality that must be addressed before robotic employees become ready for mainstream adoption.

* Precision of movement. Tasks like cooking and waiting tables require precise movements, and the objects being moved are designed to be manipulated by a human hand.
* Collision avoidance. A restaurant is usually a bustling place with a great deal of movement (not only by employees, but also customers). Being able to avoid colliding with obstacles is critical.
* Communication interface. Computerized speech recognition still has some ways to go before a randomly chosen customer can reliably communicate their orders verbally to a robotic employee.

Many restaurants currently have similar systems in place already, but the feature-set and overall design is inconsistent. Further, many of these systems focus solely on patrons who want food delivered to an exterior location, or patrons who wish to pick up their food at the restaurant and then take it with them to dine elsewhere (usually a home or office).

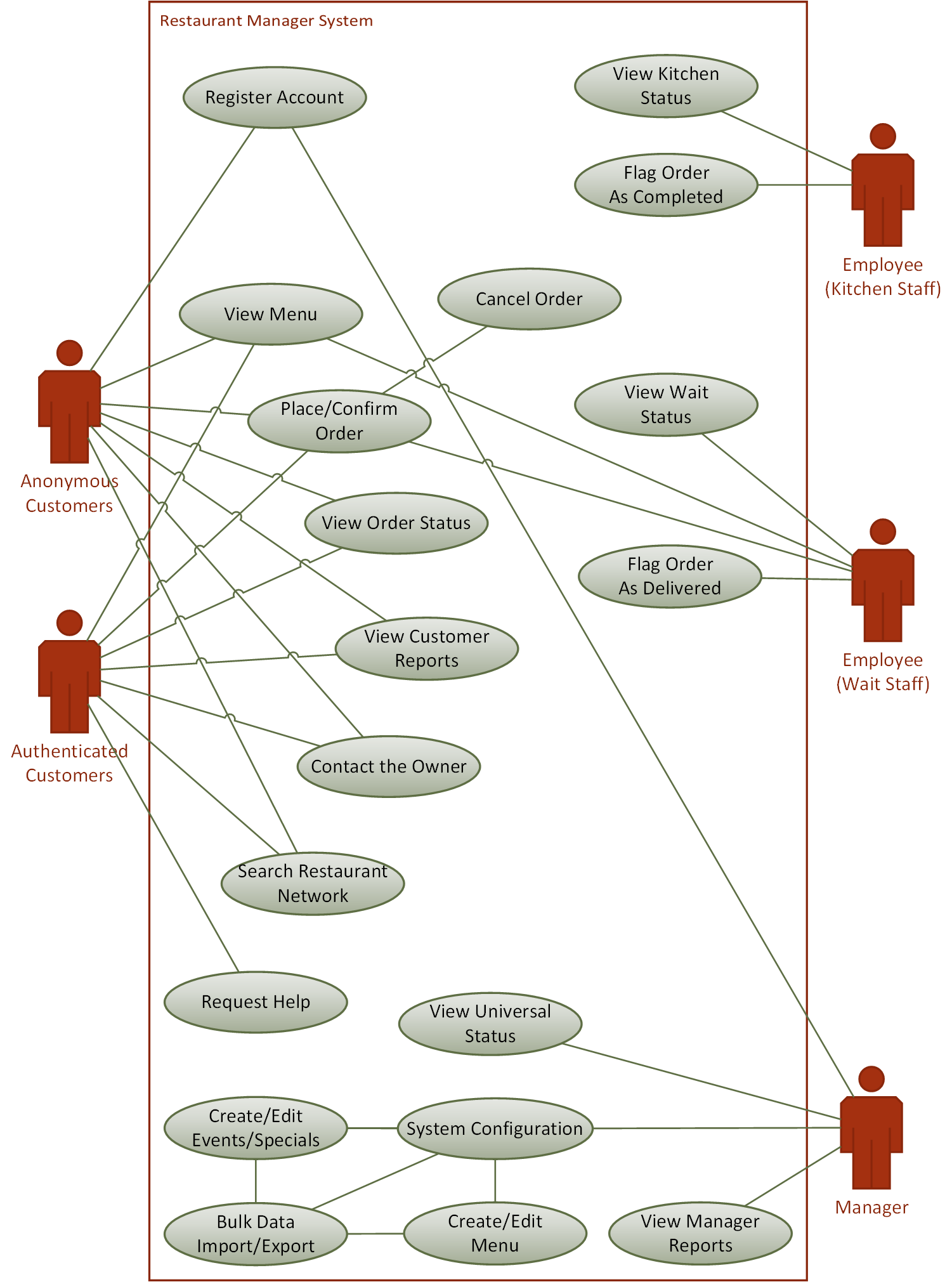
Our goal is two-fold: to serve the needs of those dining inside the restaurant, and to merge the functionality of existing systems for a consistent design and interface which can be easily customized or extended to suit each individual client.

# Project Team

* Matthew is the team leader and lead developer. The parts of this deliverable that he contributed are the title page, table of contents, and the overview (this page). He also participated in the presentation of our requirements to the class.
* David is the lead designer. The parts of this deliverable that he contributed are the individual use-case diagrams. He also participated in the presentation of our requirements to the class.
* William is a developer. The parts of this deliverable that he contributed are filling in use case details.
* Demetrius is a developer. The parts of this deliverable that he contributed are filling in use case details.
* Slaven is a developer. The parts of this deliverable that he contributed are filling in use case details. He also contributed significantly to the slides used in our presentation.

Additionally, every member of the team participated in constructing the high-level use case diagram during our team meetings.

# High-Level Use Case Diagram



# Use Case Summary



# Use Case: View Customer Reports

|  |  |
| --- | --- |
| Use Case Name | View Customer Reports |
| Description | Allows customers to view reports that are of interest to them. Things like the top ten most popular menu items and such. |
| Actors | * Customers (Authenticated and Anonymous) |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * Desired information is displayed. |
| Exceptions Paths | None. |
| Postconditions | None. |

# Use Case: View Kitchen Status

|  |  |
| --- | --- |
| Use Case Name | View Kitchen Status |
| Description | Allows kitchen staff to view all orders currently being prepared. |
| Actors | * Employees (Kitchen Staff) |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * Desired information is displayed. |
| Exceptions Paths | User does not possess required permissions. |
| Postconditions | None. |

# Use Case: View Manager Reports

|  |  |
| --- | --- |
| Use Case Name | View Manager Reports |
| Description | Allows managers to view reports that are of interest to them. Things like ranking employees by performance, financial reports, how many customers are using the system, average customer wait times, etcetera. |
| Actors | * Managers |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * Desired information is displayed. |
| Exceptions Paths | User does not possess required permissions. |
| Postconditions | None. |

# Use Case: View Menu

|  |  |
| --- | --- |
| Use Case Name | View Menu |
| Description | Allows customers and wait staff to view the items available for ordering. |
| Actors | * Customers (Authenticated and Anonymous) * Employees (Wait Staff) |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * Desired information is displayed. |
| Exceptions Paths | None. |
| Postconditions | None. |

# Use Case: View Order Status

|  |  |
| --- | --- |
| Use Case Name | View Order Status |
| Description | Allows customers to view the status of their order(s). |
| Actors | * Customers (Authenticated and Anonymous) |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * Desired information is displayed. |
| Exceptions Paths | User must have previously place an order in order to view its status. |
| Postconditions | None. |

# Use Case: View Universal Status

|  |  |
| --- | --- |
| Use Case Name | View Universal Status |
| Description | Allows managers to duplicate the functionality of both the View Kitchen Status and View Wait Status use cases. |
| Actors | * Managers |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * Desired information is displayed. |
| Exceptions Paths | User does not possess required permissions. |
| Postconditions | None. |

# Use Case: View Wait Status

|  |  |
| --- | --- |
| Use Case Name | View Wait Status |
| Description | Allows wait staff to view all of the tables to which they are currently assigned and those tables’ associated order(s). |
| Actors | * Employees (Wait Staff) |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * Desired information is displayed. |
| Exceptions Paths | User does not possess required permissions. |
| Postconditions | None. |

# Use Case: Bulk Data Import/Export

|  |  |
| --- | --- |
| Use Case Name | Bulk Data Import/Export |
| Description | Allows managers to import and export large quantities of configuration data. |
| Actors | * Managers |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * User enters any necessary data. * Action occurs. |
| Exceptions Paths | User does not possess required permissions. |
| Postconditions | None. |

# Use Case: Contact the Owner

|  |  |
| --- | --- |
| Use Case Name | Contact the Owner |
| Description | Allows customers to contact the owner of the restaurant. |
| Actors | * Customers (Authenticated and Anonymous) |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * User enters any necessary data. * Action occurs. |
| Exceptions Paths | None. |
| Postconditions | None. |

# Use Case: Create/Edit Events/Specials

|  |  |
| --- | --- |
| Use Case Name | Create/Edit Events/Specials |
| Description | Allows managers to configure specials and events based on time or date. Examples include things like Ladies’ Night, holiday discounts, last call, etcetera. |
| Actors | * Managers |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * User enters any necessary data. * Action occurs. |
| Exceptions Paths | User does not possess required permissions. |
| Postconditions | None. |

# Use Case: Create/Edit Menu

|  |  |
| --- | --- |
| Use Case Name | Create/Edit Menu |
| Description | Allows managers to create or edit individual menu items. |
| Actors | * Managers |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * User enters any necessary data. * Action occurs. |
| Exceptions Paths | User does not possess required permissions. |
| Postconditions | None. |

# Use Case: Place/Confirm Order

|  |  |
| --- | --- |
| Use Case Name | Place/Confirm Order |
| Description | Allows customers and wait staff to place and confirm an order. Customers must pay when confirming their orders. Wait Staff may elect until their customers are finished prior to entering payment. |
| Actors | * Customers (Authenticated and Anonymous) * Employees (Wait Staff) |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * User enters any necessary data. * Action occurs. |
| Exceptions Paths | Cannot place an empty order (must have selected at least one menu item). |
| Postconditions | None. |

# Use Case: Register Account

|  |  |
| --- | --- |
| Use Case Name | Register Account |
| Description | Allows anonymous customers to register an account to become authenticated customers. Also allows managers to create accounts for employees. |
| Actors | * Anonymous Customers * Managers |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * User enters any necessary data. * Action occurs. |
| Exceptions Paths | User does not possess required permissions. |
| Postconditions | None. |

# Use Case: Search Restaurant Network

|  |  |
| --- | --- |
| Use Case Name | Search Restaurant Network |
| Description | Allows customers to search for other restaurants in the nearby area (based on postal codes) that are also using the V-Menu system. |
| Actors | * Customers (Authenticated and Anonymous) |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * User enters any necessary data. * Action occurs. |
| Exceptions Paths | None. |
| Postconditions | None. |

# Use Case: System Configuration

|  |  |
| --- | --- |
| Use Case Name | System Configuration |
| Description | Allows managers to configure the V-Menu system. |
| Actors | * Managers |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * User enters any necessary data. * Action occurs. |
| Exceptions Paths | User does not possess required permissions. |
| Postconditions | None. |

# Use Case: Cancel Order

|  |  |
| --- | --- |
| Use Case Name | Cancel Order |
| Description | Allows customers and wait staff to cancel orders. |
| Actors | * Customers (Authenticated and Anonymous) * Employees (Wait Staff) |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * Action occurs. |
| Exceptions Paths | User must have previously place an order in order to cancel it. |
| Postconditions | None. |

# Use Case: Flag Order As Completed

|  |  |
| --- | --- |
| Use Case Name | Flag Order As Completed |
| Description | Allows kitchen staff to indicate that an order has been completed. |
| Actors | * Employees (Kitchen Staff) |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * Action occurs. |
| Exceptions Paths | User does not possess required permissions. |
| Postconditions | None. |

# Use Case: Flag Order As Delivered

|  |  |
| --- | --- |
| Use Case Name | Flag Order As Delivered |
| Description | Allows wait staff to indicate that an order has been delivered. |
| Actors | * Employees (Wait Staff) |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * Action occurs. |
| Exceptions Paths | User does not possess required permissions. |
| Postconditions | None. |

# Use Case: Request Help

|  |  |
| --- | --- |
| Use Case Name | Request Help |
| Description | Allows customer to request help from the wait staff. |
| Actors | * Authenticated Customers |
| Triggers | Initiated by user action. |
| Assumptions | None. |
| Basic Course of Events | * User activates the appropriate GUI widget. * Action occurs. |
| Exceptions Paths | User does not possess required permissions. |
| Postconditions | None. |

# Non-functional Requirements

These are our project’s non-functional requirements.

* Popular browser support (Chrome, Firefox, Internet Explorer/Edge, and Safari) across multiple platforms.
* Ability to keep track of multiple orders concurrently (minimum 1000).
* Ability to keep track of multiple users concurrently (minimum 100).
* Support for customized username, password, and security policies.
* Order data retention for up to three calendar years.