

### Introduction

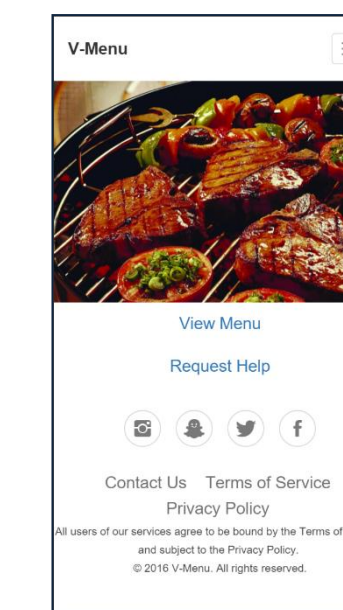
Have you ever wanted to order your food from the restaurant before you got there? Have you ever been in a situation where the waiter didn't give you what you ordered? With V-Menu, these annoyances are a thing of the past.

With the convenience of V-Menu, you can now order your food on the go and arrive at the restaurant with food nearly ready to serve. The application's OpenMenu integration means you can also send your dining preferences exactly as you like them without fear of miscommunication. V-Menu gives you the flexible futuristic dining experience you didn't know you were missing!

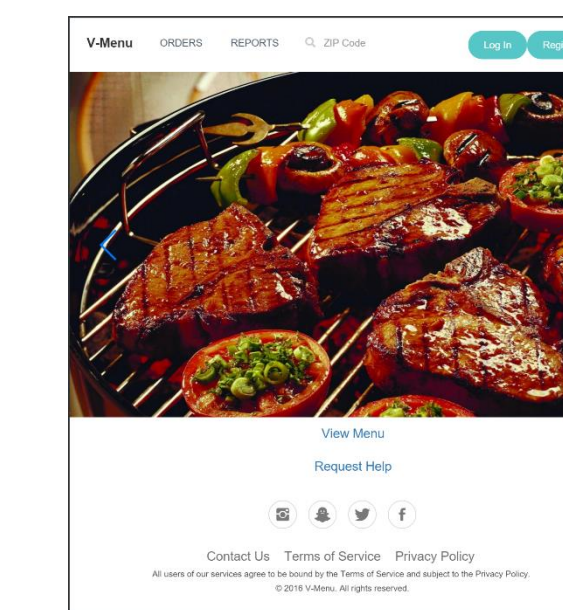
### Methods and Materials

V-Menu is a web application that utilizes a standard client-server configuration over the HTTP and HTTPS protocols. It is built on top of a standard LAMP stack, and is integrated with the OpenMenu service. The core of V-Menu was built using the Drupal content management system (CMS) framework. The layout was designed using Bootstrap Studio.

- Drupal 8.2.x
- Ubuntu Server
- Apache
- MariaDB/MySQL
- PHP
- JQuery
- Bootstrap
- Bootstrap Studio
- MySQL Workbench
- Microsoft Visual Studio



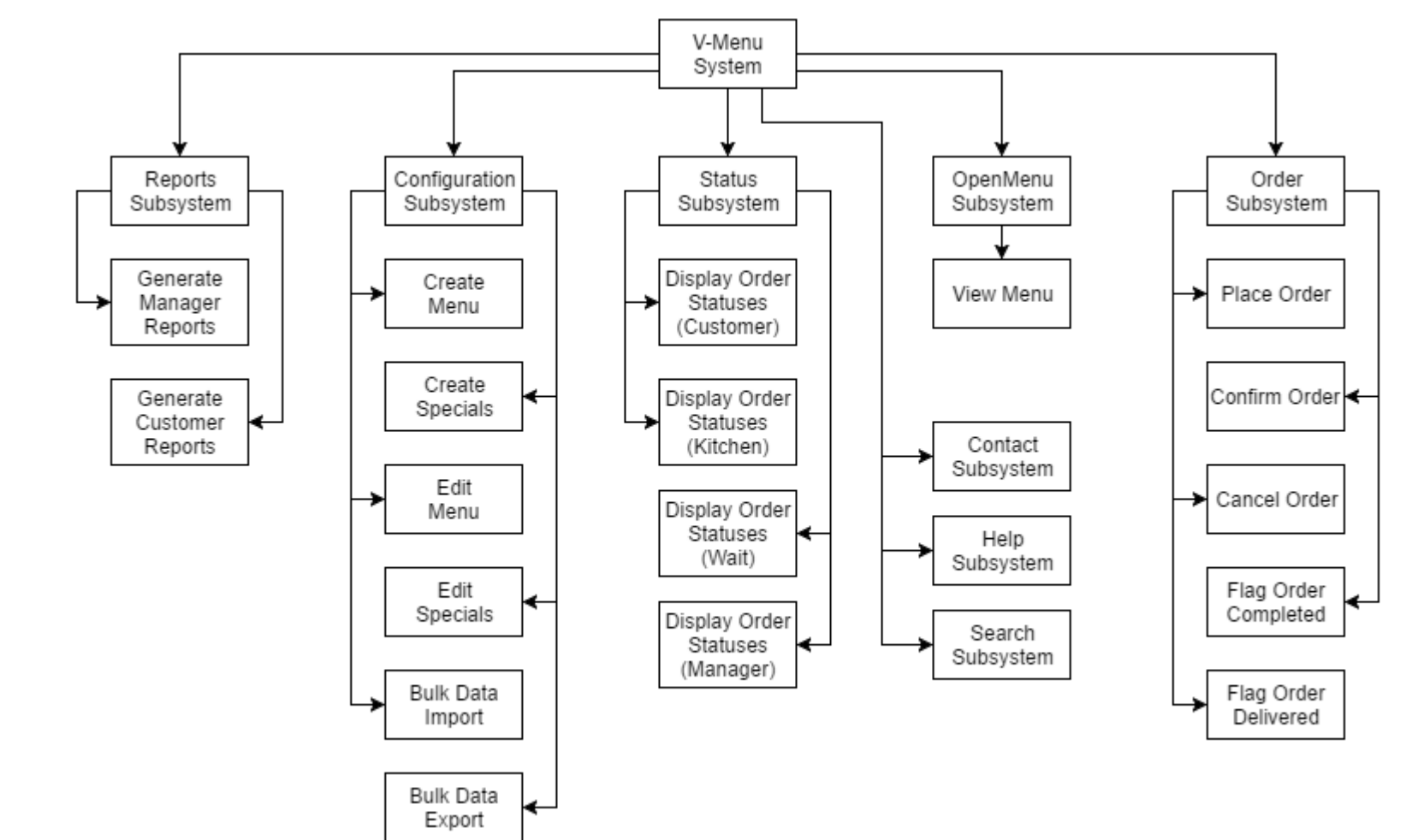
**Figure 1.** V-Menu main page layout. Smartphone form factor.



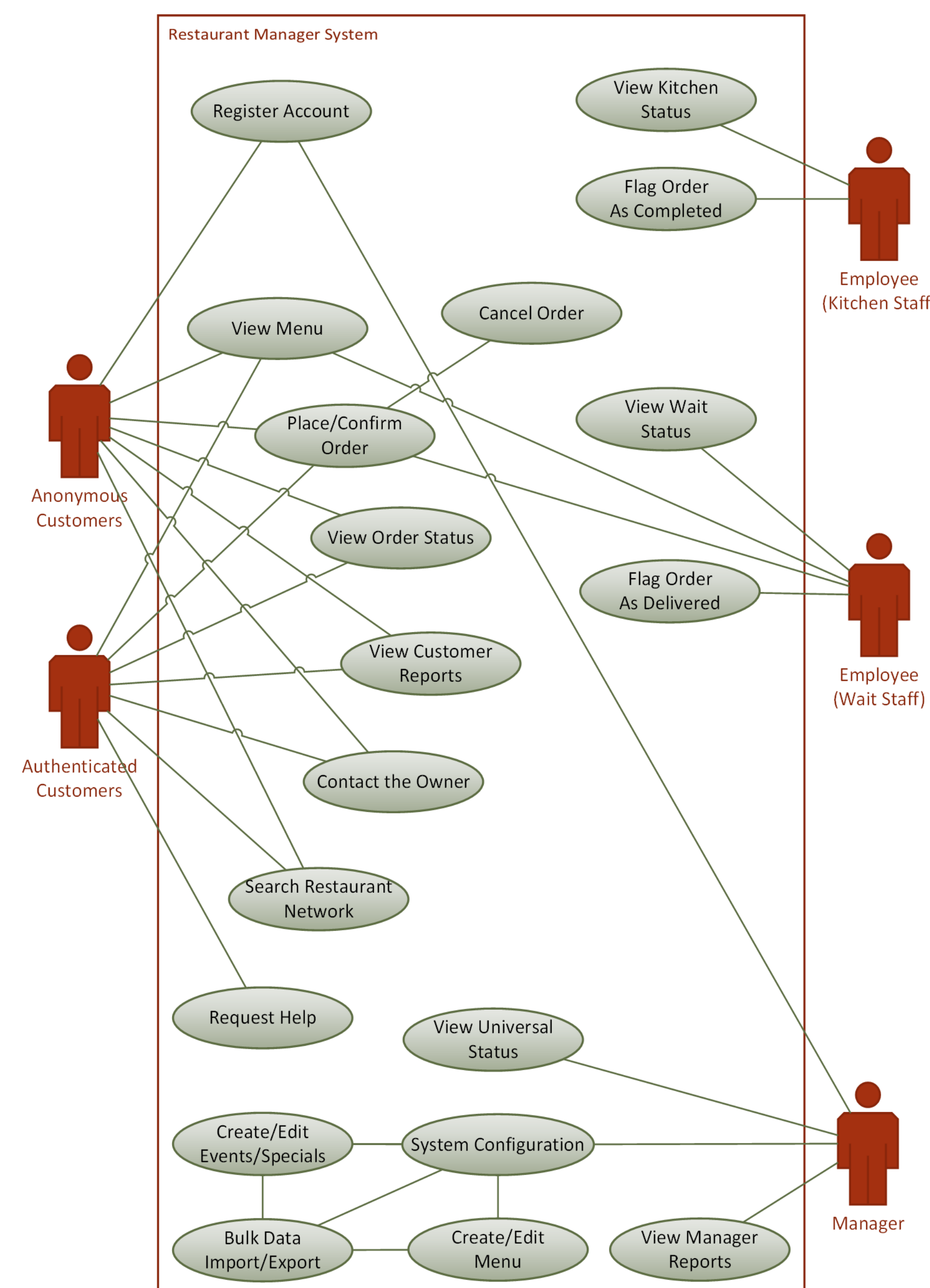
**Figure 2.** V-Menu main page layout. Tablet form factor.

### Functional Decomposition Diagram

Our team chose to perform functional decomposition for our application since it most accurately mirrored our high-level use case diagram.



### High-Level Use Case Diagram



### Objectives

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.

### Conclusions

V-Menu is the future of digital restaurant management. Our product provides a convenient way for the customer to get what they want in real-time without the hassle of waiting in line, while simultaneously arming management with advanced operational data metrics for maximal business decision making assistance. We have gone to great lengths to package simplicity and reliability all in one application.

### Server Architecture Diagram

Server Architecture - Layered

