Product Line:	[redacted]
Release:	OH DISH App Version 1.0
Product Manager:	[redacted]
Author:	Sanyogita Herwathe

Distribution List
Feature Product Manager: [redacted]
Project Manager: [redacted]
Test Coordinator: [redacted]
Others: ---

Table of Contents

Revision History

Revision

Date (yyyy-mm-dd)	Author	Version	Description
2025-11-20	Sanyogita H	1.0	First Draft

Review

Date (yyyy-mm-dd)	Author	Version	Reviewed By
2025-11-20	Sanyogita H	1.0	

Related Documentation

OH DISH Application Functional requirements: Functional Design_OhDish_ver 1.0

OH DISH Application Internal Design document: InternalDesign_OhDish_ver 1.0

The scope of this document is to provide testing framework to fully test OH-DISH software. The test cases will be created and used to test an application as a part of a ad-hoc, functional (smoke, unit, system, integration, data integrity) and regression test.

1. Introduction

1.1. Functional Overview (Description of the product)

OH-DISH is a web-based platform designed to centralize and streamline all operations for restaurants, merchants, and food service businesses. The application consolidates customer activity, orders, menu settings, and business operations into a single system. Its core functional components include:

(Currently Within scope as Tester)

1. User & Role Management

A unified module that maintains detailed information about system users and their assigned roles. It enables the business to:

- Store users and their roles
- Manage internal users (as per roles created by merchant in the system)
- Users with different access levels based on (Active/ Blocked/ Expired/ Deleted/ Pending for approval/Suspended)
- Create and Assign Roles as per their assigned permissions to view-create-edit-delete varied fields

2. POS (Point of Sale)

A built-in POS system lets staff:

- Create walk-in or in-house orders
- Process payments
- Manage order tickets
- Connect with receipt printers

3. Order Management

A central area for handling all customer orders across various stages. Users can:

- View and manage **new**, **processing**, **ready**, **completed**, or **canceled** orders
- Monitor order flow in real time
- Synchronize POS and online orders
- Ensure fast and accurate fulfillment

(**Other Systems**)

4. Dashboard displays quick stats such as:

- Total orders
- Cancellations
- Refunds
- Sales performance

5. Menu & Item Configuration

This module allows merchants to configure all food-related data, including:

- Food categories
- Menu items
- Attributes (size, addons, variations)
- Order types (delivery, pickup, dine-in)

Users can add, edit, delete, or view menu items at any time with instant updates to the system.

6. Table Booking / Reservations

Allows businesses to:

- Accept online or in-store table reservations
- Manage booking schedules
- View upcoming reservations
- Confirm or cancel bookings

7. Campaigns & Marketing

Tools for growing customer engagement:

- Promotions and discount creation
- Coupon management
- Communication and messaging tools (email/SMS)

8. Reports & Analytics

A comprehensive reporting section for business insights, including:

- Sales overview and daily performance
- Top customers
- Order trends
- Exportable reports for accounting or analysis

9. System Configuration & Settings

A section enabling management of:

- Payment gateways
- Printers
- Images & media
- Taxes
- SMS settings
- Account details

Objectives of OH-DISH

The main goals of the platform are to:

1. Centralize Merchant Operations

- Combine orders, customer data, menu settings, and POS activity into one system

2. Improve Efficiency

- Provide fast order processing
- Simplify menu management
- Reduce operational mistakes

3. Support Multi-User Access

- Allow multiple staff members to work simultaneously
- Maintain secure and organized user permissions

4. Build a Scalable Restaurant Management Ecosystem

- Support high traffic and future business growth

1.2. Risks

Risks associated with this software are enlisted below
1.Risk of data breaches due to the exclusive use of open-source frameworks, which may include security vulnerabilities if not regularly patched or monitored.
2.Risk of project delays caused by extensive compatibility testing required when integrating multiple open-source components.
3.Risk that the software may not scale effectively under high concurrency, resulting in decreased performance or incompatibility during system scale-ups.
4. Risk of authentication weaknesses, such as insufficient login validation or unlimited login attempts, potentially leading to unauthorized access or data breaches.

Risks associated with this software are enlisted below
5. Risk of unauthorized access to the centralized database if access controls are not properly enforced, potentially causing data contamination, modification, or loss.
6. Risk of poor user interface design making the system difficult to navigate, which may lead to user errors, duplicated entries, or accidental data loss.
7.Risk of inadequate data backup processes, insufficient security audits, or lack of versioning controls, leading to data duplication, corruption, or loss.
8. Risk that the application may not support heavy user load, resulting in system slowdowns or outages during peak usage.
9.Risk of degraded user experience due to slow performance from certain open-source libraries or underlying framework limitations.
10.Risk of integration failures either between open-source components or within the system itself, especially when handling different role-based user processes.

1)

2)

1.3. Test Schedule

Total Test Planning time: (See Outline format in Test Coverage section)

Test Area	Test Object	Test Low-Level Object	Test Method	No. of Test Cases#
Users	All Users	View all user roles	Verification, Functional UI	25
		Add User	Functional Validation	21
		Edit User	Functional Data Integrity	21
		Delete User	Functional Permission	4
		Role-based Access	Security Testing, Permission	6
Orders	New Orders	View New Orders List	UI Functional	5
		Order Detail View	Functional	16
	Orders Processing	Change Order Status: New → Processing	Workflow Integration	12

Test Area	Test Object	Test Low-Level Object	Test Method	No. of Test Cases#
	Orders Ready	Change Order Status: Processing → Ready	Workflow Integration	12
	Completed Orders	Mark Order as Completed	Functional	12
	Scheduled Orders	View scheduled items	Functional	3
	All Orders	Search, Filter, Sort	Functional UI	6
POS	Create Order	Add Items to Order	Functional UI	15
		Modify Item Quantities	Functional	10
		Apply Discounts/Notes	Functional	2
		Payment Processing	Integration Functional	5
	Order History	View historical orders	UI Functional	4
Total				68

1.4. Total Resource Estimates

Metric	Value
Total Test Cases (TC)	179
Total Testing Time (1.3* TC)	233 hours
Work Hours Per Day	2-3 hours
Total Conduct Time	77-116 days
Additional Buffer Time must be setup as per Team discussions	
Total Resources Required for testing sections within scope (Users, Roles, Orders and POS)	1 person

1.5. Test Strategy

The test strategy outlines the overall approach and methodology used to validate the functionality, performance, security, and usability of the Oh-Dish application. The objective is to ensure the system meets all business and technical requirements and is ready for production deployment.

The primary purpose of these tests is to uncover the system's limitations and measure its full capabilities. A list of the test objectives follows below:

- Verify that all functional requirements are implemented correctly.
- Validate data integrity and business rules across the Oh-Dish workflows.
- Ensure usability and accessibility of functional and UI elements on major browsers.
- Confirm application performance under normal and peak loads.
- Identify and resolve defects prior to release.

1.6. Smoke Test

The smoke test will be performed daily on each new build to define if it stable enough for further testing

1.7. System Test

The System tests will focus on the behavior of the OH-DISH. User scenarios will be executed against the system as well as screen mapping and error message testing. Overall, the system tests will test the integrated system and verify that it meets the requirements defined in the requirements document.

1.8. Documentation Test

Tests will be conducted to check the accuracy of the user documentation. These tests will ensure that no features are missing, and the contents can be easily understood.

1.9. Bug Reporting

All bugs should be reported and tracked using the GitLab System. Priorities should be assigned as discussed in the Wiki section of Gitlab or via Team communications.

1.10. Entry Criteria for Test Execution

- All Functional Design documents have been approved and baseline is frozen.
- All Test cases and other test supporting documents have been completed and approved

- All test data requirements have been identified, and test data has been mapped to the test cases
- Test environments have been setup as per the application requirements
- Test management tool has been setup with required accesses to all testers
- Unit Testing is 100%.

1.11. Exit Criteria for Test Execution

- All Test scripts/cases need to be run at least once
- All defects irrespective of the severity has to be reported to the stakeholders
- All defects with severity Critical, High, Medium need to be fixed and retested successfully.
  (NOTE. All minor/cosmetic defects can be deferred to be fixed in later stages)
- All defects need to be moved to terminal stages such as Closed, Non-Defect, and Deferred. No defect has to be in any stage other than the terminal stages.
- All regression defects fixed and retested
- All User Acceptance Tests are approved (signed-off) and based of requirements obtained from business users
- Passed to a planned ratio of the tests should be at least 95%
- All Functional Design documents have been signed

1.12. Tools

The following testing tools will be utilized during the project:

- Git Lab (Issue reporting and Management)
- Other tools to be discussed / TBD

1.13. Environment Requirements

1.13.1. Workstations

The QA and testing team will utilize a diverse hardware and OS setup to simulate a range of end-user environments and browser configurations:

- **Hardware Specifications (Minimum)**
  - Processor: 3.5 GHz (quad-core or better)
  - Memory: 8 GB RAM
  - Storage: 512 GB HDD/SSD
  - Network: Wired or wireless LAN; access to shared test servers

- Peripherals: Network-attached printer for test reports and letter printing

- **Test Machines**
  - 1 IBM-compatible PC running **Windows 10**

This mix of platforms will ensure broad compatibility and identify any platform-specific issues early in the testing lifecycle.

1.14. Test Coverage Outline

The following areas will to be tested:

I. Areas of Independent Testing

1. **Users**
   - **Users**
     1. Add Users
     2. Update Users
     3. Delete Users
   - **Roles**
     1. Add Roles
     2. Edit Roles
     3. Delete Roles
2. **POS**
   - Create an Order
     1. New
        1. Select Items (from Menu)
        2. Customer information
        3. Kitchen (Tracking preparing orders)
        4. Payment Method
        5. Mode of delivery (Takeout/ DineIn/ Pickup/Delivery)
        6. Promo/Discount/ Tips/Points
        7. Reset/ Hold Bill
        8. Book a Table
        9. Register Customer Requests
   - Order History

3. **Orders**
   - New Orders
   - Order Processing
   - Order Ready
   - Completed

- Scheduled
- All Orders

II. Objects and Their Properties

1. **User**
   - First name
   - Last name
   - Email address
   - Contact Number
   - Username
   - Status
   - Role
   - Password
   - Password Confirmation
2. **Roles**
   - Number#
   - Name
   - Access
   - Actions (Edit/ Delete)
   - Add New
   - Permissions
3. **POS**
   - Create an Order
     1. New
        1. Select Items (from Menu)
        2. Customer information
        3. Kitchen (Tracking preparing orders)
        4. Payment Method
        5. Mode of delivery (Takeout/ DineIn/ Pickup/Delivery)
        6. Promo/Discount/ Tips/Points
        7. Reset/ Hold Bill
        8. Book a Table
        9. Register Customer Requests
   - Order History
4. **Orders**
   - New Orders (Filters for (“Order Type”, “Payment status”, “Sort”) and New Order Results)

- Order Processing (Filters for (“Order Type”, “Payment status”, “Sort”) and “Order Processed” Results)
- Order Ready (Filters for (“Order Type”, “Payment status”, “Sort”) and “Order Ready” Results)
- Completed (Filters for (“Order Type”, “Payment status”, “Sort”) and “Completed” Results)
- Scheduled (Filters for (“Order Type”, “Payment status”, “Sort”) and “Scheduled” Results)
- All Orders (Filters for (“Order Type”, “Payment status”, “Sort”) and “Scheduled” Results)

III. Methods

1. **AD-Hoc Testing**
2. **Functional Testing**
   - End-to-End Scenario Testing for Core Modules

(Note: Other Testings will be conducted as required and as requested)

3. **UI Testing**
4. **Regression Testing**
5. **Integration Testing**
6. **System Testing**
7. **Smoke Testing**
8. **Security Testing**
9. **Performance Testing**
10. **Compatibility Testing**
11. **Documentation Testing**

1.15. Test Cases

See attached test Case File for this section

References