Product Requirement Document for Odoo 18 Community Online Ordering System

Purpose & Overview

The purpose of this document is to outline the requirements for implementing an online ordering system using Odoo 18 Community. This system will allow website visitors to order food or products online, with an integration to an MQTT broker for sending details to a robot and receiving order status updates. The focus is on creating a seamless and secure ordering experience, enhancing customer satisfaction and operational efficiency.

Executive Summary

This document describes a project aimed at leveraging Odoo 18 Community's capabilities combined with MQTT protocol for real-time communication. The system will capture customer orders, transmit them to an IoT-enabled robot, and update customers with live status. Key functionalities include order processing, secure data transmission, and user interface enhancements for order tracking. The project aligns with strategic objectives of improving service delivery and operational capacity.

Stakeholders

- Customers
- Website Users
- Odoo Administrators
- Robot/IoT System
- MQTT Broker

Background and Rationale

The system operates on Odoo 18 Community with the website module and integrates with an MQTT broker. This setup is intended to automate the ordering process, improve response times, and provide a scalable solution for handling orders efficiently. The implementation acknowledges the growing trend towards automation in delivery systems and aims to capitalize on this by improving the customer experience and operational flow.

Product Goals and Objectives

- Enable customers to order food/products via the Odoo 18 website.
- Automatically send order details (product name, quantity) over MQTT to the topic brandname/orders.
- Receive order status messages from the topic brandname/status.
- Display order status in customer's "My Orders" page on the website.

User Personas & Use Cases

User Personas

- **End Customer:** Looking for a convenient online ordering experience with real-time updates.
- **Odoo Admin:** Manages website and monitors system performance, aiming for operational efficiency.

Use Cases

- **Order Placement:** Customer selects products and places an order through the website.
- **Status Tracking:** Customer tracks order status via the account page after placement.

Functional Requirements

- Capture order details from Odoo website checkout.
- Format and publish order data as JSON to MQTT topic brandname/orders.
- Subscribe to MQTT topic brandname/status to receive updates (statuses: accepted, preparing, ready, delivered).
- Store and display status in Odoo under the corresponding order.
- Provide order history and live tracking on customer’s account page.

Non-Functional Requirements

- Enable real-time communication with minimal delay.
- Ensure secure MQTT communication (utilizing authentication and TLS if possible).
- Scalable to handle multiple concurrent customer orders.
- Implement fault tolerance (retry publishing if MQTT broker is unavailable).

System Workflow / User Journey

The process begins with a customer placing an order on the Odoo website. The system then publishes order details via MQTT to an IoT-enabled robot. The robot updates the order status, which is received back via MQTT and displayed to the customer on their order tracking page. Steps include:

- Customer places an order on the website.
- Order details are published to MQTT broker topic brandname/orders.
- Robot updates order status, sending updates to topic brandname/status.
- Status updates are displayed on the customer's order page.

Sequence diagrams will be provided to further illustrate this process.

Assumptions & Constraints

- Odoo Community edition is used for the project.
- An external MQTT broker (such as EMQX or Mosquitto) handles the messaging.
- Internet connectivity is required for system operation.
- Basic robot integration is assumed without extensive customization.

Acceptance Criteria

- Orders are successfully sent to the robot within X seconds after placement.
- Status transitions are reflected in the Odoo system.
- Customers can view live order updates on their account page.

Design & UX Guidelines

User interfaces should be intuitive and consistent with Odoo's existing design standards. Emphasis should be placed on a simple order tracking interface and clear communication of status updates.

Technical Requirements & Integrations

- Implement MQTT protocol for message communication.
- Integrate secure authentication mechanisms for data exchange.
- Support JSON format for data transmission.

Risks & Mitigation Strategies

- **Risk:** MQTT broker failure.**Mitigation:**Implement retry logic and secondary broker setup.
- **Risk:** Internet connectivity issues. **Mitigation:** Ensure local caching of order data for resynchronization.

Timeline & Milestones

The project is divided into key phases: Requirement Gathering, System Development, Testing, and Deployment. Detailed milestones with dates should be defined during project planning.

Approval & Version History

Version	Date	Author	Notes
1.0	[Date]	[Author Name]	Initial draft