odoo

📌 Odoo Cafe POS

---

1. Project Overview

This hackathon project is a Restaurant POS (Point of Sale) system called **“Odoo Cafe POS”**, designed to handle:

- Restaurant table-based ordering (**Floor/Table view**)
- Fast billing & checkout
- Multiple payment methods (**Cash, Card/Bank, UPI QR**)
- **Kitchen Display** integration (send orders directly to kitchen)
- **Customer Display**
- POS backend configuration + reporting dashboards
- Optional additions: **Self/Online Ordering (token-based)**, and basic **Booking**

---

2. Goals & Scope

✅ Main Goal

Build a complete POS flow including backend setup and frontend ordering.

📌 Key Outcomes

- Cashier can open a POS session and start taking orders
- Orders can be paid using different payment methods
- Order details can be pushed to kitchen screen
- Dashboard/reporting shows sales and session details
- Customer-facing screen shows order/payment status

---

3. User Roles

🧑‍💼 POS User (Staff / Admin)

- Manages POS setup (products, payment methods, floors & tables)
- Opens session, creates orders, and sends them to Kitchen Display

- Completes payments (Cash / Digital / UPI QR)
- Views dashboard and reports
- Handles self/online orders received in POS

---

4. Modules / Features Breakdown

⚙️ A) POS Backend (Configuration Area)

🔐 A1) Authentication (Login / Signup)

- POS users can create an account using **Signup**
- Existing users can access the system using **Login**
- After login, the user can open POS session and access backend configuration

---

🛍️ A2) Product Management

1) General Info: Name, Category, Price, Unit, Tax, Product Description

2) Variants: Attribute (example: Pack), Values (6 / 12 items), Extra prices

---

💳 A3) Payment Method Setup

POS supports multiple payment types with enable/disable toggle:

1) Cash

- If enabled, becomes available during checkout

2) Digital (Bank/Card)

- Generic **“Digital”** category representing card and bank payments

3) QR Payment (UPI)

- Dedicated UPI method
- Requires UPI ID (example shown: `123@ybl.com`)
- System generates QR code at payment screen based on saved UPI ID

Notes:

- UPI QR appears on the Payment page
- Confirmation screen exists after QR scan

🏢 A4) Floor Plan Management

- Create floors (example: “Ground Floor”)
- Add/manage tables Back-end
- Table fields: Table Number, Seats, Active, Appointment Resource (optional)

🧾 A5) POS Terminal Setup + Sessions

POS Terminal is created from POS Settings and includes:

- Last open session
- Last closing sale amount

Button:

- Open Session → Opens POS terminal

📱 A6) Self Ordering (Optional)

- System generates a **token** for mobile/self ordering (linked to a table/session)
- Orders placed using this token automatically create an **Order Number**
- Order is sent directly to **Kitchen Display** for preparation

---

🍳 A7) Kitchen Display

- Receives order items after cashier sends the order
- Flow includes a Send button → pushes menu/order to Kitchen Display

---

📊 A8) Reporting & Dashboard

- Dashboard + reporting menu
- Export options: **PDF / XLS**

📌 Reporting Filters (Purpose)

- **Period:** Used to view sales/orders within a specific date range (today, week, custom range)
- **Session:** Filters reports by a specific POS session to analyze shift-wise sales
- **Responsible:** Filters data by staff/user responsible for the session or orders
- **Product:** Filters reporting based on a product to track best-selling or low-selling items

---

🧾 B) POS Frontend (Terminal Experience)

🧭 B1) POS Terminal – Top Menu

Top navigation contains:

- Table: Redirects to Table/Floor Plan view
- Register: Opens register screen

Actions:

- **Reload Data:** Refreshes POS data from backend (latest products/settings).
- **Go to Back-end:** Opens POS configuration/settings screen.
- **Close Register:** Ends the current POS session and closes the register.

🪑 B2) Floor View (Table View)

- Tables appear as selectable cards/buttons
- Example numbers: Table 3, Table 6 etc.
- Selecting a table starts order creation for that table

🛒 B3) Order Screen (Products + Cart)

- Pick products (Pizza, Pasta, Burger, Coffee, Water)
- Adjust quantities (+/-)
- View order lines with price totals
- Confirm and move to payment

💰 B4) Payment Screen

Payment screen includes:

- Total amount (example: **$580**)
- Payment methods list:
  - Cash
  - Digital / Card
  - UPI QR

After payment method selection:

- Validate payment
- Confirmation screen

📱 B5) UPI QR Payment Flow (Special Flow)

When UPI is selected:

- Show QR code screen
- Display:
  - Amount
  - “UPI QR” label
- Buttons:
  - Confirmed
  - Cancel

After confirmation:

- Payment confirmation screen appears
- Clicking anywhere dismisses it
- User returns to Floor View automatically

🖥️ B6) Customer Display

Separate display screen for customer view:

- Shows order info
- Shows payment status (paid/unpaid)
- Useful for transparency

👨‍🍳 B7) Kitchen Display

Kitchen Display shows:

- Only the products/categories that are configured to be sent to the kitchen
- Items list with quantity and item names
- Orders coming in real-time from POS “Send” action

🧾 Order Stages

- To Cook→ newly received orders
- Preparing → items currently being prepared
- Completed → ready orders

🖱️ Kitchen Actions

- Clicking a ticket/card moves the order to the next stage
- Clicking a product item marks it as prepared (strike-through)
- Ticket number is same as the Order number

---

5. Complete Flow (End-to-End)

- POS user signs up (first time) or logs in to access the system
- POS user configures POS (products, payment methods, floor/tables, displays)
- POS user opens session and selects a table
- Order is created either:
  - from POS manually, **or**
  - via mobile/self ordering using token (auto creates Order Number)
- Order is sent to Kitchen Display for preparation
- Kitchen updates order status (To Cook → Preparing → Completed)
- POS user completes payment (Cash / Digital / UPI QR)
- Reports are reviewed using filters (Period / Session / Responsible / Product)

---

6. Why This Hackathon Problem is Important

- **Real-world ERP workflow:** Shows how a complete restaurant flow works end-to-end (POS → Kitchen → Payment → Reports).
- **Business logic focus:** Teaches handling real operational problems like order lifecycle, table management, and status tracking not just UI.
- **Industry-ready system thinking:** Builds a production-like solution with real-time coordination, multi-payment support, and reporting insights.

🔗 Mockup Link

Mockup: https://link.excalidraw.com/l/65VNwvy7c4X/23T6FEnXS2l