Supermarket Self-Checkout System

ProLog-Project Documentation

Team Members & Responsibilities

Member	*Role*	*Key Contribution*
Hager Abo-Samra	Knowledge Base	Defined items, prices, tax rates ,loyalty tiers.
Anas El-gezawy	Core Operations	Implemented scan (with age-restriction triggers), remove, and stock management logic.
Shahd Ahmed	Produce Handling	Implemented weigh_produce for variable-weight items
Ebrahim El-safry	Security	Implemented verify_age to unlock the system for tobacco products.
Mohamed Ahmed	Financials	Implemented complex tax calculations, loyalty tier logic, and final receipt generation.
Mohamed Ashraf	Planner/Integration	Implemented show_basket for real-time system state monitoring.

1. Project Overview

This project implements a **Self-checkout system** using Prolog. The system manages the flow of a customer transaction from an **empty basket** to a **printed receipt**. It features intelligent state management that handles:

- **Inventory Control:** Real-time stock decrementing/restoring.
- **Regulatory Compliance:** Automatic blocking of transactions containing tobacco products until an admin verifies age.
- **Dynamic Pricing:** Calculation of variable-weight produce (kg) and category-specific tax rates.
- **Customer Loyalty:** A tiered reward system (Bronze/Silver/Gold) providing discounts.

2. State Representation

The system state is tracked using a dynamic 6-arity structure that evolves with every action:

state(Basket, Total, BlockedStatus, Loyalty, Stock, PaymentStatus)

- **Basket:** A list of item(Name, Price, Weight) representing the customer's cart.
- **Total:** The running financial total.
- **BlockedStatus:** none (Normal) or age_check_needed (Locks payment).
- **Loyalty:** The customer's card status (e.g., card_sara or no_card).
- **Stock:** A list of stock(Item, Count) tracking inventory levels.
- **PaymentStatus:** Tracks if the transaction is pending or completed.

3. Key Features & Logic

A. Inventory Management (Anas)

The system prevents selling items that are out of stock.

• Scanning: Decrements stock count by 1.

• Removing: Increments stock count by 1.

• Validation: Uses stock_check/3 to ensure Count > 0.

B. Age Restriction & Security (Ebrahim)

Certain items (Cigarettes, Rolling Tobacco) are flagged as age_restricted. If scanned, BlockedStatus becomes "age_check_needed". The verify_age(Admin) action is required to clear this block.

C. Variable-Weight Produce (Shahd)

Ensure the item is a variable-priced

Like Fruits are priced per KG using weigh_produce(Item, WeightInGrams).

D. Financials: Tax & Loyalty (Mohamed & Hager)

Tax Structure:

- • 0%: Fruits
  • 5%: Food
  • 10%: Snacks
  • 14%: Cleaning
  • 30%: Tobacco (High tax)

Loyalty Tiers:

- • Bronze (0-499 pts): 0% Discount
  • Silver (500-1999 pts): 2% Discount
  • Gold (2000+ pts): 5% Discount

4. Actions Reference

Action	Description
scan(Item)	Scans a barcoded item. Updates basket and stock. Triggers age check if tobacco.
remove(Item)	Removes an item from the basket and restores it to the shelf stock.
weigh_produce(Item, Grams)	Weighs produce (e.g., apple) and calculates price based on weight.
verify_age(AdminID)	Unlocks the system if an authorized admin confirms customer age.
calculate_total	Applies taxes and loyalty discounts to compute final amount.
pay(Method)	Finalizes transaction (Cash/Card/E-wallet). Only works if not blocked.
print_receipt	Outputs final receipt.
show_basket	Debugging tool to view current basket.

5. Usage Scenarios (Test Case)

Scenario A: Standard Grocery Run (Silver Card)

```
?- initial_state(S0),
  action(scan(milk), S0, S1),
  action(scan(bread), S1, S2),
  action(calculate_total, S2, S3),
  action(pay(e-wallet), S3, S4),
  action(print_receipt, S4, _).
```

6. Conclusion

The Supermarket Self-Checkout System successfully integrates the contributions of all 6 team members, handling conditional blocking, mathematical transformations, and state persistence.