SDLC activity sheets

ACTIVITY SHEET 1 — Requirements Gathering (SRS Development)

Project Title: Shopping billing cart system

Team Members:

S. Hasini - 25RBAIM0060

M.Ramya sri - 25RBAIM0005

Y.Samitha- 25RBAIM0046

D.Sneha- 25RBAIM0043

Task 1: Stakeholder Interview

1. Identify your stakeholders

- Customers (users)
- Shop Owner / Store Manager
- Cashier / Billing Staff
- System Administrator
- Inventory Manager

2. Interview Questions (8–10)

1. What problems do you face with the current billing system?
2. What features do you expect in a shopping cart billing system?
3. Should the system support multiple payment methods?
4. Do you want automatic bill generation?
5. Should discounts and taxes be applied automatically?
6. How important is accuracy in billing calculations?
7. Do you need inventory updates after billing?
8. Should customers be able to view their bill summary?
9. What security features are required for the system?
10. Should reports be generated for daily or monthly sales?

Task 2: Extract Requirements

Requirement ID	Functional / Non-Functional	Requirement Description
R1	F	The system should allow users to add and remove products from the cart.
R2	F	The system should calculate the total bill including tax and discounts.
R3	NF	The system should provide fast and accurate

Requirement ID	Functional / Non-Functional	Requirement Description
		billing with minimal errors.

Task 3: User Stories

· As a **customer**, I want to add items to the cart, so that I can purchase multiple products.

· · As a **customer**, I want to view the total bill, so that I know how much to pay.

· · As a **cashier**, I want the system to generate bills automatically, so that billing is faster.

· · As a **shop owner**, I want sales reports, so that I can track business performance.

· · As a **system admin**, I want secure login access, so that data is protected

Deliverable: A mini SRS

Mini Software Requirement Specification (SRS)

1. Introduction

The Shopping Cart Billing System is designed to automate the billing process in retail stores. It helps customers add products to a cart and generates accurate bills efficiently.

2. Functional Requirements

- Add and remove items from the cart
- Calculate total price automatically
- Apply tax and discounts
- Generate and display bills
- Maintain product list

3. Non-Functional Requirements

- System should be user-friendly
- Fast response time
- Accurate calculations
- Secure access

4. Users

- Customers

ACTIVITY SHEET 2 – System Design (High-Level & Low-level)

Project: Shopping Billing Cart

Task 1: Create a System Architecture Diagram

Draw or attach:

- Components :
- User Interface (Web / Mobile App)
- Backend Server
- Database
- Payment Gateway (External System)
- Notification Service (External System)
- Data flow:
- User logs in/ sign up
- User selects products
- Products added to cart
- Backend calculates total bill
- Payment request sent to payment gateway
- Payment confirmation received
- Order stored in database
- Bill/receipt sent via notification
- External systems :
- Payment Gateway (UPI / Card / Wallet)
- Email / SMS Notification Service

Space for Diagram:

```
┌────────────────────┐
    │  User      │
    │ (Web/Mobile) │
    └─────────┬─────────┘
        │
        ▼
    ┌────────────────────┐
    │  Frontend  │
    │  (UI Layer)  │
```

```
└──────────┬──────────┘
      │ API Requests
      ▼
┌─────────────────────┐
│ Backend │
│ (Business │
│ Logic) │
└──────────┬──────────┘
      │
┌─────────────┴─────────────┐
│          │
▼          ▼
┌──────────────┐  ┌──────────────────┐
│ Database │  │ Payment │
│ (MySQL) │  │ Gateway │
└──────────────┘  │ (External) │
    └──────────────────┘
        │
        ▼
┌──────────────────────┐
│ Notification │
│ Service (SMS / │
│ Email) │
└──────────────────────┘
```

Task 2: Create UI Wireframes

Required screens (tick off as completed):

- Home Screen
- ┌──────────────────────────────┐
- │ Shopping Cart App │
- │-------------------------│
- │ [Browse Products] │
- │ [Login / Signup] │
- │ [View Cart] │
- └──────────────────────────────┘

- Login / Signup
- ┌──────────────────────────┐
- │ Login / Signup │
- │────────────│
- │ Email: [______] │
- │ Password: [______] │
- │ [Login] │
- │ [Signup] │
- └──────────────────────────┘
- Main Feature Screen
- ┌──────────────────────────┐
- │ Product List │
- │────────────│
- │ Product A ₹100 [Add] │
- │ Product B ₹200 [Add] │
- │ Product C ₹150 [Add] │
- │────────────│
- │ Cart Total: ₹450 │
- │ [Checkout] │
- └──────────────────────────┘
- Settings / Profile
- ┌──────────────────────────┐
- │ User Profile │
- │────────────│
- │ Name: │
- │ Email: │
- │ Order History │
- │ Logout │
- └──────────────────────────┘

Task 3: Database Design

Create an ERD.

Entity	**Attributes**
1. User	user_id(PK),name,email,password,phone
2.Product	product_id,(PK),product_name,price,stock_quantity
3. Cart	cart_id (PK),user_id (FK),total_amount
4.Cart_Item	cart_item_id(PK),cart_id(FK),product_id(FK),quantity,price
5. Order	order_id(PK),user_id(FK),total_amount,payment_status,order_date
6. Payment	payment_id(PK),order_id(FK),payment_method,payment_status,transaction_id

Deliverable: Architecture Diagram + UI Wireframes + ERD

- Cashiers
- Shop Owner
- Administrator

ACTIVITY SHEET 3 — Development Phase

Task 1: Break Work Into Tasks

Development Backlog

Task ID	Feature	Description	Assignee	Status
T1	Product Management	Add product list with price and quantity	Hasini	Done
T2	Cart Operations	Add and remove items from the shopping cart	Ramya	In Progress
T3	Billing Calculation	Calculate total bill with tax and discount	Samitha	To Do

Task 2: Code Walkthrough Notes

What features were implemented today?

- ✓ Product list creation
- ✓ Add items to cart
- ✓ Display selected items

What blockers did you face?

- Minor errors in calculating total price
- Handling invalid user inputs

Solutions attempted / next steps:

- Used conditional checks to validate inputs
- Will implement tax and discount calculation next
- Plan to generate final bill output

Deliverable: Updated backlog + coded feature demo

```
# Shopping Cart Billing System

products = {
    "Rice": 60,
    "Sugar": 45,
    "Milk": 30,
    "Oil": 120
}

cart = {}

def add_item(item, quantity):
    if item in products:
        cart[item] = cart.get(item, 0) + quantity
        print(item, "added to cart")
    else:
        print("Item not available")

def remove_item(item):
    if item in cart:
        del cart[item]
        print(item, "removed from cart")
    else:
        print("Item not in cart")

def calculate_bill():
    total = 0
    for item, qty in cart.items():
        total += products[item] * qty
tax = total * 0.05
    discount = total * 0.10 if total > 500 else 0
    final_amount = total + tax - discount

    print("\nBill Summary")
    print("Total Amount:", total)
    print("Tax (5%):", tax)
    print("Discount:", discount)
    print("Final Amount:", final_amount)

# Demo Execution
add_item("Rice", 2)
add_item("Milk", 3)
remove_item("Milk")
calculate_bill()
```

ACTIVITY SHEET 4 — Testing Phase

Task 1: Create Test Cases

Test Case ID	Description	Steps	Expected Result	Actual Result	Status (P/F)
TC1	Add product to cart	1. select a product 2. Click add to cart	Product added to cart with correct price	Product added successfully	P
TC2	Update product quality	1. go to cart 2. Increse quantity	Total price updates correctly	Total updated correctly	P
TC3	Checkout process	1. click checkout 2. Enter details 3. Confirm order	Bill generated successfully	Error while generating bill	F

Task 2: Bug Report

Bug ID	Description	Severity	Steps to Reproduce	Screenshot	Status
B1	Bill not generated on chckoout	High	1. add product 2. Click checkout 3. Confirm payment		Open
B2	Cart total incorrect for multiple items	medium	1. add 2 different items 2. Increse quantity		Fixed

Task 3: Test Summary

- Total test cases: 3
- Passed: 2
- Failed: 1
- Major issues found:
- 1. Checkout bill generation failure
- 2.Incorrect cart total calculation (fixed)

ACTIVITY SHEET 5 — Deployment & Release Notes

Task 1: Deployment Checklist

Code merged

Database configured

Environment variables set

Build successful

Final testing done

Task 2: Release Notes

Features Included:

Shopping cart functionality with add/remove/update items.

Secure billing system with multiple payment gateway integration (credit/debit cards, UPI, PayPal).

Order summary and invoice generation.

User-friendly checkout flow with address and shopping details.

Basic admin dashboard for monitoring transactions.

Known Issues:

Limited support for international currencies.

No discount/coupon system in this release

Next Update Goals:

Add coupon/discount code support.

Expand currency and tax calculation for global users.

Enhance reporting and analytics for admins.

Improve mobile responsiveness and accessibility.

Deliverable: Deployment & Release Document:

This document confirms successful deployment of the Shopping Cart Billing System v1.0. It records completed deployment steps and provides release notes for stakeholders, ensuring transparency and clarity for future updates.

ACTIVITY SHEET 6 — Maintenance & Reflection

Task 1: Fix & Patch

Record any issues and fixes

Patch ID	Issue	Root Cause	Fix Implemented	Status
P1	Incorrect total bill calculation	Logic error in price and quantity multiplication	Corrected calculation logic and added input validation	Done

Task 2: Team Retrospective

What worked well?

- Clear requirement gathering helped in smooth development
- Modular coding made debugging easier
- Billing calculations worked accurately after fixes

What needs improvement?

- Better testing before final implementation
- Improved user interface for easier interaction

What will we change next time?

- Perform unit testing at every stage
- Plan tasks more effectively
- Add more validation and error handling