Applying Systems Engineering Life Cycle to Café Lunch Receiving System

1. Objectives

- Map systems engineering lifecycle phases to a small C++ CLI café lunch system.
- Practice verification and validation planning with a real-world ordering example.

2. Theory / Background

The Systems Engineering Lifecycle (SELC) provides a structured approach to development. It ensures that requirements, design, implementation, verification, validation, and maintenance are followed in sequence. For the café lunch system, it guarantees that every step—from displaying the menu to billing and receipt generation—is carefully traced, tested, and meets user needs.

3. Software / Tools Required

- Diagramming Tool: Draw.io or EdrawMax
- Git: Version control
- C++ CLI Environment: Code::Blocks, g++, or Visual Studio

4. Lab Activities

Activity 1: Define System Boundary

Chosen System: Café Lunch Receiving System (C++ CLI project).

Inside Scope:
- Show café lunch menu
- Customer places order by item code and quantity
- System calculates bill (subtotal, total)
- Error handling for invalid inputs
- CLI input/output only

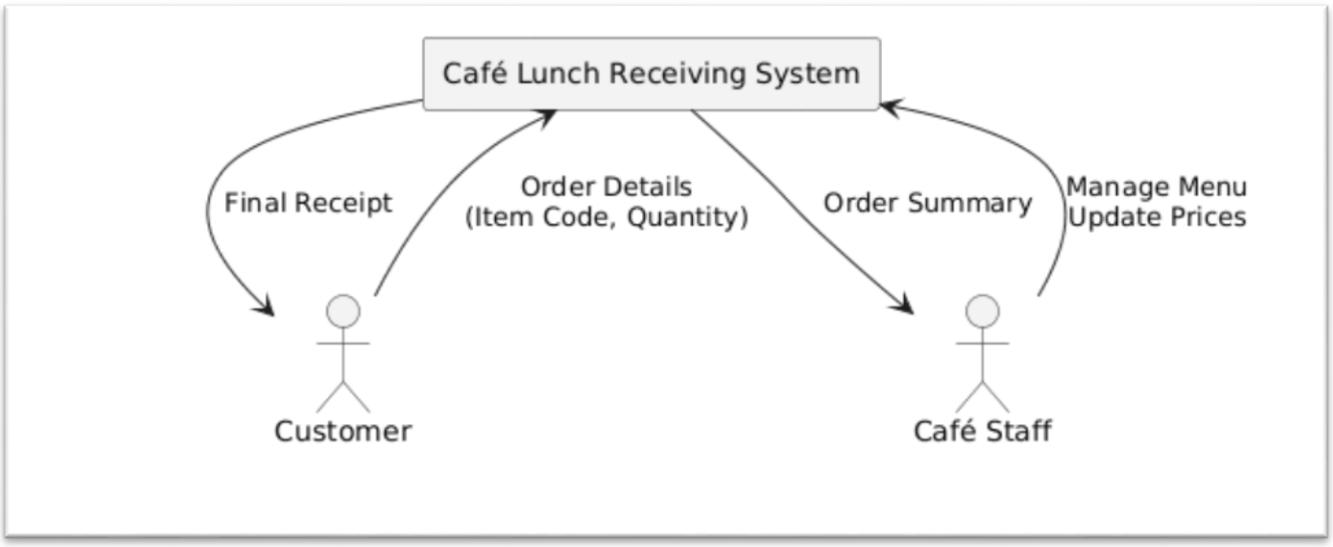
Outside Scope:
- GUI or web interface
- Online payments
- Database storage (use arrays/vectors in memory)

Stakeholders:
- Customers – place orders
- Café Staff – manage orders
- Instructor – evaluates project
- Developers – implement system

High-Level Goals:
- Provide a text-based menu

- Process orders quickly and accurately
- Show bill with totals
- Handle errors gracefully

CONTEXT DIAGRAM



Activity 2: System Requirements & Traceability Matrix (RTM)

High-Level Requirements:

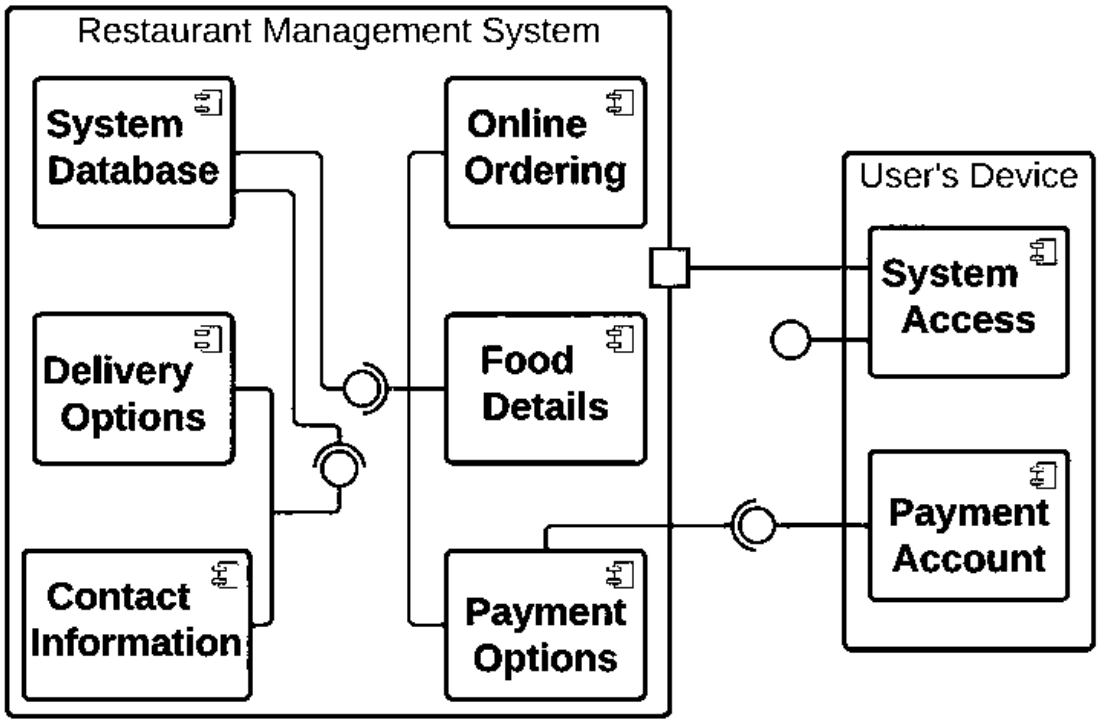
- R1: System shall display a lunch menu with item codes and prices.
- R2: System shall allow customers to order items using item codes.
- R3: System shall accept multiple items in one order.
- R4: System shall calculate total bill correctly.
- R5: System shall handle invalid item codes gracefully.
- R6: System shall handle invalid quantities gracefully.
- R7: System shall print a final receipt showing ordered items and totals.
- R8: System shall exit gracefully upon request.

Activity 3: High-Level Design

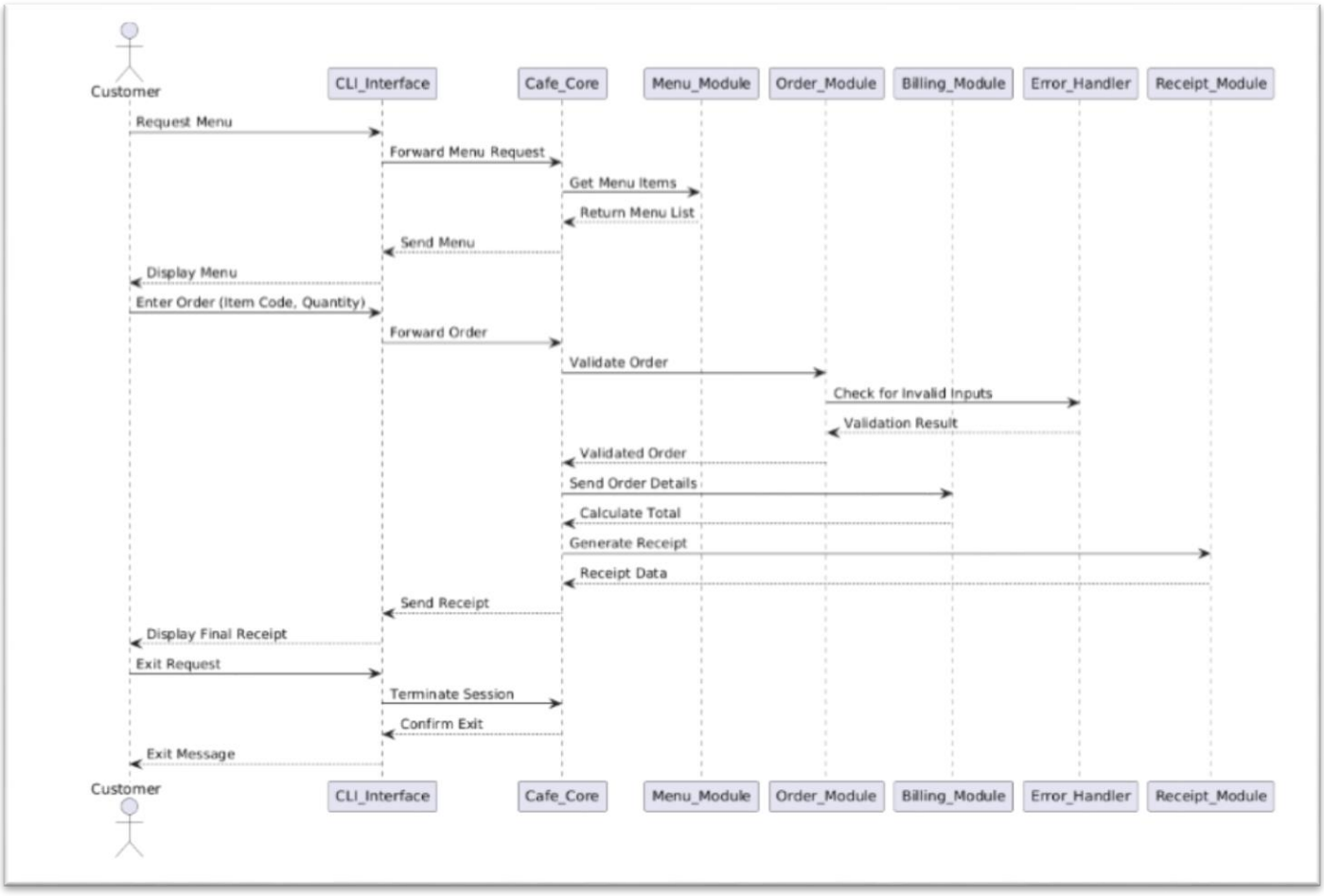
Modules & Interfaces:
- Menu Module → displays food items and prices
- Order Module → accepts item codes & quantities
- Billing Module → calculates totals
- Error Handling Module → invalid inputs
- Receipt Module → displays final receipt

Data Formats:
- Input: Item code (int), quantity (int)
- Output: Receipt with item names, quantities, totals
- Storage: Temporary in-memory arrays/vectors

COMPONENT DIAGRAM



SEQUENCE DIAGRAM



Activity 4: Verification & Validation (V&V) Planning

Req ID	Verification Method	Validation Method	Acceptance Criteria	Test Type
R1 – Show menu	Code inspection	Functional Test	Menu displays correctly	Black-box
R2 – Order items	Unit test	Functional Test	Accepts valid codes	Black-box
R3 – Multiple items	Unit test	Functional Test	Accepts >1 item	Black-box
R4 – Bill total	Unit test	Functional Test	Correct totals	Black-box
R5 – Invalid codes	Test	Negative Test	Error message displayed	Black-box
R6 – Invalid quantity	Test	Negative Test	Error message displayed	Black-box
R7 – Receipt printing	Demo	Usability Test	Clear receipt shown	Black-box
R8 – Exit system	Demo	Functional Test	System exits gracefully	Black-box

Activity 5: Implementation Planning

RACI Chart:

Module / Task	Responsible	Accountable	Consulted	Informed
Menu Module	Student A	Instructor	Student B	All
Order Module	Student B	Instructor	Student A	All
Billing Module	Student C	Instructor	Student B	All
Error Handling	Student D	Instructor	Student C	All
Receipt Module	Student A & B	Instructor	Student C	All
Integration & Testing	Student C & D	Instructor	All	All
Documentation	Student A & D	Instructor	Student B	All

Integration Checklist

1. 1. Verify menu displays correctly
2. 2. Order input works with multiple items
3. 3. Billing calculations are correct
4. 4. Invalid input handled
5. 5. Receipt generated
6. 6. End-to-end test (customer order → receipt)
7. 7. Exit functionality

5. Discussion / Analysis

The system boundary clarified café ordering scope and avoided unnecessary features. The RTM ensured no requirements were missed. Design diagrams improved understanding of module interactions. V&V planning reduced risks of errors in billing and order handling.

6. Conclusion

The Café Lunch Receiving System successfully applied SELC phases. Requirements, design, implementation planning, and V&V were aligned, resulting in a structured and reliable small-scale project.