Software Engineering
Assignment-4

Harshit Chawrasia
2301010210
BTech CSE Sec-D

Ans 1) Test Case Design for Payement Gatecoay

1. Test Case Design Techniques

Technique	field	Test Values	focus
• BVA	Amount	$0.99, $1.00, $1.01, $9999.00, $10000.00	Test boundary limits (min/max)
• ECT	Card No.	Valid 16-digit, Invalid 15-digit, Negative Amount	Test Representative data partitions

2. Decision Table (Payement Methods & Errors)

Rule	Card valid	fraud passed	Action
• R1	Y	Y	Process Payement
• R2	N	-	Decline (Invalid Card)
• R3	Y	N	Decline (fraudelent)

3. Cause-Effect Graphing

- Improvement: Graphically map all Input condition (causes) to system outcomes (Effects)
- Benefit: Systematically drives test cases for complex logical combinations (eg: card expiry AND currency error), ensuring complete test coverage for error handling and fraud detection.

Ans 2) Structural Testing for the Library Management System

1. Path Testing & CFG

Path testing executes all possible code paths based on a programs structure. It ensures thorough coverage of decision points.

- Simple CFG Paths:
  - → Path 1: Success (Book → User → Borrow)
  - → Path 2: Book Error (Invalid ID / Unavailable)
  - → Path 3: User Error (Invalid ID / Limit Exceeded)

2. Data Flow Testing (DFT)

DFT selects Test paths by tracking the definition (Def) and use (use) of variables. It aims to ~~ea~~ catch faults where data is used before its defined or where updates are missed.

- Key Variables: BookID, UserID, Book Status, Current Borrowed Count.
- Focus: Testing the Def-Use chain, eg ensuring current BorrowedCount is used in the limit check before its Defined (incremented) in the transition record.

3. Mutation Testing

Mutation Testing improves ~~that~~ test robustness by deliberately inserting small errors (mutants) into the code

- If the existing test suite fails against the mutant, the test is strong.
- If the test passes (mutant "survives"), it exposes a weakness in the test suite that needs ~~to~~ fixing.

Ans3) Integration & System Testing for Ride-Hailing Ride

1. Integration strategy
   - Strategy : Top down Integration
   - Method : Start with high-level modules (eg: UI/Booking) and integrate down to lower modules (GPS, Payement).
   - Goal : Verify correct data exchange across interfaces (eg. booking to GPS → Payement).

2. Alpha & Beta Testing

Test Type	→ who	→ where	→ Purpose
• Alpha	→ Internal Employees	→ Controlled (Dev site)	→ find major bugs and usability issues before public release
• Beta	→ Real External Users	→ Real Environment	→ Test performance, compatibility, and real world defects before final launch

3. Recommended Tools
   - functional : Appium / Selenium (Mobile UI automation), Postman / SoapUI (API testing).
   - Non-functional : JMeter / Load Runner (load / stress Testing), OWASP ZAP (Security Testing), Battery Historian (Performance / Battery usage).