Analytical

1. **Given:**
   Analytical Phase:

   a) Person days = 80, Team members: 5, Hourly Rate: $50, Cost of Analysis Phase $\Big\}$ = 80 days

   Design Phase:

   Person-days = 40

   $\boxed{\$60,000}$ $\Big\{$ * 8 hours/days, hours * 5 team mem

   Cost of design phase = 40 person-days * 8 hours/day * $50/hours * 5 team mem
   = $80,000

   Implementation Phase:

   Person-days = 70

   Cost for Implementation phase $\Big\}$ = 70 person-days * 8 hours/day * $50/hours * 5 team mem
   = $1,40,000

   Testing Phase:

   Person-days = 35

   Cost for testing phase = 35 person-days * 8 hours/day * $50/hours * 5 team mem
   = $70,000

   Maintenance Phase:

   Person-days per month = 25, Duration: 12 months

   Cost for Maintenance Phase = 25 person-days/month * 8 hours/days * $50/hr *
   5 team members * 12 months. = $6,00,000

   b) Calculate overall project cost, including maintenance phase:

   Overall projects cost = sum of all phases + overhead cost => $90,000

   c) If project delayed by 2 months in implementation phase, additional co

   Additional cost for delay = 2 months * 20 working days/month * 8 hour/day
 $50 hour * 5 team members = $80,000

   d) If project team decides to reduce maintenance phase to 6 Month onstead of 12 months, how much cost saving.

   Cost saving = (12 months - 6 months) * 25 person -days/month * 8 hours
   * $50/hr * 5 members = $~~30,0000~~ $3,00,000

2. **Scenarios:**

   1.) User 2.) Checkout System 3.) User profile system 4.) pricing system 5.) Pay
   6) Order System 7.) E-mail System 8.) Inventory System 9.) Shipping System

   Sequence:
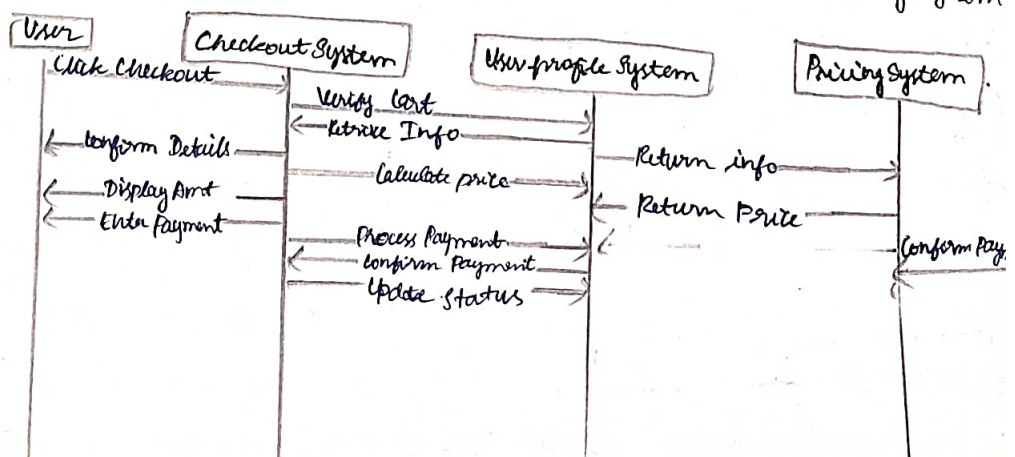
3. Implementation Time per complexity point = 2 hours
Testing time per scenario: 3 hours

a.) Calculate the total estimate time:

i.) Browse product:
Implementation time = 18×2 = 36

ii.) Add to cart: Implementation Time = 15×2 = 80

v.) View History:
Implementation Time = 20×2 = 40

b.) Calculate total Estimated Time:

i.) Browse Product:
Testing time = 5×3 = 15

iii.) Checkout:
Testing Time = 4×3 = 12

v.) View Order History:
Testing Time = 6×3 = 18

Total Testing time = 69 Hours

iii.) Checkout:
Implementation Time = 13×2 = 26

iv.) Manage Account:
Implementation Time = 17×2 = 34

Total Implementation Time = 166 hours

ii.) Add to cart:
Testing Time = 3×3 = 9

iv.) Manage Account:
Testing Time: 5×3 = 15

c.) Determine overall project time:

Overall Project Time = Total Implementation × total testing time
= 166 + 69
= 235 hours.

Order System   Email System   Inventory   Shipping System.

mail

← Update
Inventory   ← Notify
Shipping.
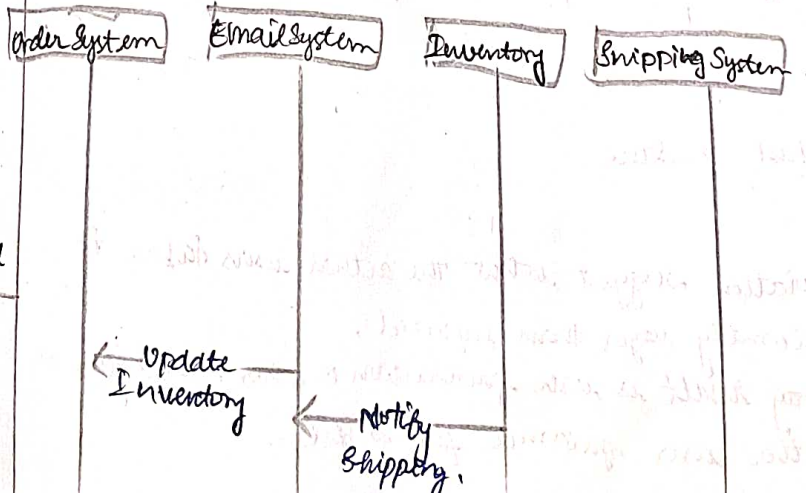
A° Calculate total

Ↄ make
Ↄ Model
Ↄ year
Ↄ weight
} Total no. of attributes for vechile is (4)

Ↄ make
Ↄ model
Ↄ Year
Ↄ No. of door
Ↄ weight
Ↄ truck_size
} Total no. of attributes for cars is (6)

Ↄ make
Ↄ model
Ↄ year
Ↄ weight
Ↄ type of handle
Ↄ frame material
} Total no of attributes is (6)

Ↄ make
Ↄ model
Ↄ year
Ↄ weight
Ↄ Large-capacity
Ↄ No. of Axles
} Total no. of attributes for truck : 6

b.)

The independence Aniom states that the design should maintain the independence of components melting each subclass should have attributes specific to it and not be influenced by others subclass.

Contain common attributes Bike - Type of handle sea and frame material sp car - no. of doors & trunksize specific Truck - Large capacity and no of clas specific. Each subclass has its own specific attributes in addition of inheriting comma attributes from vechile class.

(.)

vechile : 4 attributes   B: 6 attributes (4+2)   total : 22 attributes.
car : 6 attributes   Truck : 6 attributes (4+2)

Suggestion to minimize content

To adhere to the information the information aniom content should be mini while maintaing the designs functionality. Given the attributes are well-distribu

a)

$Z = \dfrac{X - \mu}{a}$ Task A: $z_A = \dfrac{6-5}{1} = 1$, Task B = $\dfrac{12-10}{2} = 1$, Task c: $\dfrac{25-20}{3}$

b.)

Task c has the highest z-score.

Potential

Ↄ Task c is higher deviation suggest what the actual users take to complete this is significantly longer than expected.
Ↄ The increased theme may result is user frustruction or error, suggestion e eat for redesign or better user guidence for taskC.

z score for less than 5 mins:

$$z = \frac{5-6}{1} = -1$$

Probability ≈ 0.1587

z score for less than 10 mins

$$z = \frac{10-12}{2} = -1$$

Probability ≈ 0.1587

z-score for less than 20 mins:

$$z = \frac{20-25}{3} = -1.67$$

Probability $= 0.0475$

To estimate the total effort required to compare complete the developement and quality associative phase of life project, we need to break down slow problem

1. Calculate total no of methods

Total Methods $= 5 \times 4 = 20$

2.) Calculate total lines of code.

$LOC = 20 \times 25 = 500$

3) Estimate developement:

Effort $=$ Total LOC $\times 1.4$

$= 700$ person-hours

4.) Estimate quality assurance effort

$= 0.20 \times 70$

$= 140$ person hours

5.) Calculate total effort

$= 700 + 140$

$= 840$ person-hours

To Components:

1) Client Interface   4) Authentication Service
2) Transaction manager   5) Notification Service.
3.) Account db

Interface.

1) Client Interface   4.) Authentication Service
2.) Transaction Manager   3.) Account db
5.) Notification service.

D pre

1.) Client Interface → Transaction Manager
2.) transaction Manager → Account db
3.) Transaction manager → Authentication Service
4.) Transaction manager → Notification Service.

deployment:

1) Client Server (Hosist client interface and Transaction Manager)
2.) Database Server (Hosist account db)
3.) Authenotify Server (Hosist Authentication)

Connection:

1.) Client Server → Db Server
2.) Client Server → Authorify Server.

Client Server: Hosts client Interface and transaction manager compor

Hosts account db component.

Hosts the Authentication service And sunotification Service.

8.) a)

1.) Library - Book (A library consists many books)
2.) Library - Member (A library consists many member)
3.) Library - Loan (A library mage as many loans.
4.) Book loan (A book can be loaded multiple times)
5.) Member Loan (A member loan can have multiple loans)

b.)

1.) Library - Book : Library 1 : Book : 0
2.) Library - Member : Library : 1 Member : 0
3.) Library - loan : Library : 1 Loan : 0
4.) Book - Loan : Book : 1 Loan : 0
5.) Member loan : Member : 1 Loan : 0

(-)

1.) Library and Book : Composition
The library own books and responsible for its life cycle

3.) Library and Loan : Aggregation
The library manage loan but loans independence of library life cycle

5.) Member Loan : Association.
A member can have multiple loans

2.) Library Member : Aggregation
The library holds members but members independence of library.

4.) Book and Loan : Association
A book can be learned multiple times and records simple reference

9.)

Product Table :

↳ Product Id (int , primary key)
↳ Name (String)
↳ Description (String)
↳ Price (Float)
↳ Quantity Stock (int)

Customer Table :

↳ Customer ID (int , primary key)
↳ First Name (String)
↳ Last name (String)
↳ Email (String)
↳ Phone number (String)

Order Table :

↳ Order Id (int , primary Key)
↳ Order Date (date)
↳ Customer Id (int)
↳ Total Amount (float)

4.) Order Item Table :

↳ Order Item Id (primary key int)
↳ Order Id (int , forgin Key)
↳ Product Id (int)
↳ Quantity : (int)
↳ Item price (int)

(b)

1.) Product Table : 5   2.) Customer Table : 5   3.) Order Table : 4
1.) Order Item : 5   Total = 19 Attributes.

Product Table : 1500 products 2.) Customer Table : 150 customers 3.)
Order Item Table : Total : 1500 × 5 = 7500.   Total : 7500

Given: Total no. of classes : 500   Avg no. of methods per class : 10

Total no. of test method : no. of classes × Avg no. of methods per class = 5000

Given : 85% method coverage. Test case per method to achieved 85% coverage

Total no. of test case = Total no. of methods × Test case per method = 50

Given : 0.5 defects per method.

Expected no. of defects : Total no. of methods × Defects per method = 2500

Given : Defect detection efficiency (DDE) = 70%.

Effected no. of defects and fixed. = Expected no. of defects × DDE
= 1750

Given : Expected no. of defects before testing expected number of defects detected and fixed

= 2500 - 1750 = 750.

ANALYTICAL.

Requirement Analysis :

↳ Functional Requirements

↳ Student Enrollment

• Add new student

• Update

• Drop

• Generate

↳ Grade Management :

• Record Grades

• Update Grades

• Generate

• Calculate CGPA

↳ Course Registration

• Add new courses

• Update

• Drop

• Update

↳ User Management

• Admin Login

• Student Login

• Faculty Login

Non-Functional Requirements

↳ Performance   ↳ Usability

↳ Security   ↳ Scalability

↳ Reliability   ↳ Compliance