

RISK INFORMATION SHEET (QB questions)

1) Inexperienced Staff

Risk Information Sheet			
Risk id: P02-8-25	Date: 21/09/23	Prob: 75%	Impact: High
Description: This risk pertains to the potential challenges and issues that may arise due to team members lacking the necessary experience and expertise to effectively contribute to the project. Inexperienced staff may lead to delays, increased errors, and quality issues.			
Refinement / Context: <ol style="list-style-type: none">1) New hires have limited industry or project-specific experience.2) Rapid expansion of the team.3) Limited training or onboarding for new team members.4) High attrition rate5) Experienced employees leaving the organization			
Mitigation / monitoring: <ol style="list-style-type: none">1. Clearly define roles and responsibilities for each team member to avoid confusion and duplication of efforts.2. Provide targeted training and mentorship programs to help inexperienced staff acquire necessary skills.3. Ensure that the project is adequately staffed with a mix of experienced and inexperienced staff to provide guidance and knowledge transfer.4. Consider reallocating experienced team members to critical tasks.5. Encourage staff to document their work and share knowledge within the team to build a collective understanding.6. Maintain a succession plan to ensure a smooth transition of responsibilities.7. Conduct exit interviews with departing employees to gather knowledge and insights for knowledge transfer.			
Management / Contingency plan / Trigger: <ol style="list-style-type: none">1. Experienced members of the organisation should be kept with the less experienced ones for them to gain expertise, balance must be maintained within the team , roles defined and ensure smooth flow of the project tasks2.			
Current Status: 25/09/23: Mitigation steps initiated.			
Originator: Simona	Assigned: Simon		

2) Risk of budget and size estimation going wrong.

Risk Information Sheet			
Risk ID: P02-8-25	Date: 21/09/23	Prob: 75%	Impact: High
Description: The risk pertains to the possibility of inaccuracies in budget and size estimation for the project. Estimation errors can lead to financial discrepancies and impact project timelines and resource allocation. Inaccurate size estimation may result in an underestimation of the project's complexity, leading to resource shortages and schedule delays, or an overestimation, which may result in excess resources being allocated.			
Refinement / Context (Add points naming them sub condition 1, 2, 3): <ol style="list-style-type: none">1. The project requirements are not clearly defined, making it difficult to accurately estimate the scope of work and the resources required to complete the project.2. The unavailability of adequate data for estimation of project requirements and its complexity.3. Project managers with limited experience are more likely to make inaccurate estimates.4. Complex or evolving project requirements.5. External factors, such as changes in technology or market fluctuations, impact the accuracy of budget and size estimation.			
Mitigation / monitoring: <ol style="list-style-type: none">1. Expert Involvement : Engage experienced project managers and domain experts in the estimation process. Their knowledge and insights can help improve the accuracy of budget and size estimates.2. Estimation Tools: Utilize advanced estimation tools and software that incorporate industry best practices and data-driven models to enhance the precision of estimates.3. Historical Data Analysis: Analyze historical project data to identify patterns and trends. Historical data can provide valuable benchmarks for estimation.4. Continuously monitor project expenses and compare them against initial budget estimates.5. Develop a contingency budget to reserve funds to address unexpected cost increases in future.			
Management / Contingency plan / Trigger: The contingency plan should detail the steps to be taken if estimation errors are identified. This might involve reallocating resources, adjusting schedules, or securing additional funding as needed.			
Current Status: 25/09/23: Mitigation steps initiated.			
Originator:	Assigned:		

3) Reusability

Risk information sheet			
Risk ID: P02-4-32	Date: 5/9/09	Prob: 80%	Impact: high
Description: Only 70 percent of the software components scheduled for reuse will, in fact, be integrated into the application. The remaining functionality will have to be custom developed.			
Refinement/context: Subcondition 1: Certain reusable components were developed by a third party with no knowledge of internal design standards. Subcondition 2: The design standard for component interfaces has not been solidified and may not conform to certain existing reusable components. Subcondition 3: Certain reusable components have been implemented in a language that is not supported on the target environment.			
Mitigation/monitoring: 1. Contact third party to determine conformance with design standards. 2. Press for interface standards completion; consider component structure when deciding on interface protocol. 3. Check to determine number of components in subcondition 3 category; check to determine if language support can be acquired.			
Management/contingency plan/trigger: RE computed to be \$20,200. Allocate this amount within project contingency cost. Develop revised schedule assuming that 18 additional components will have to be custom built; allocate staff accordingly. Trigger: Mitigation steps unproductive as of 7/1/09.			
Current status: 5/12/09: Mitigation steps initiated.			
Originator: D. Gagne	Assigned: B. Laster		

4) Risk of High Staff Turnover

Risk Information Sheet			
Risk id: P02-8-25	Date: 21/09/23	Prob: 75%	Impact: High
Description: This risk pertains to the potential of a significant number of team members leaving the project or organization during its course. High staff turnover can disrupt project continuity, cause knowledge loss, impact team dynamics, and lead to delays.			
Refinement / Context: <ol style="list-style-type: none">1. Dissatisfaction among employees with work conditions or project direction.2. Inadequate work-life balance can force employees to leave the organization.3. Opportunities for career advancement elsewhere4. Lack of opportunities for professional development5. Market-driven factors, such as competitive job offers or industry trends.			
Mitigation / monitoring: <ol style="list-style-type: none">1. Conducting exit interviews to understand the reasons why employees are leaving and identify areas for improvement2. Collecting employee feedback surveys to gauge employee satisfaction and identify potential problems.3. Creating a positive work culture that promotes teamwork, collaboration, diversity, and inclusion4. Invest in ongoing training and development programs and support certifications and continuing education to enhance skills and knowledge.5. Regularly assess and adjust project timelines and resource allocation to accommodate potential staff turnover.6. Promote the long-term benefits of staying with the organization, such as stability and growth opportunities.			
Management / Contingency plan / Trigger: <ol style="list-style-type: none">1. Staff amenities and benefits have been improved , trainings initiated , Feedbacks noted and necessary improvisations are being worked upon			
Current Status: 25/09/23: Mitigation steps initiated.			
Originator:	Assigned:		

Function Point

13) Railway Reservation System

External Inputs (EI):

Login for passengers
Booking a ticket for passengers.
Modifying an existing reservation, including changes to passenger information, travel dates, and seats.
Canceling a reservation.
Providing feedback or complaints.

External Outputs (EO):

Generating a ticket for passengers.
Sending a booking confirmation via email or sms.
Printing a ticket receipt.
Emailing or texting a reservation cancellation confirmation.

External Inquiries (EQ): EG : What the user wants to know additionally

Checking seat availability for a specific train and date.
Searching for train schedules between two cities.
Inquiring about refund policies.
Checking luggage policies and fees.
Querying about onboard services and amenities.
Contacting customer support for assistance.

Internal Logical Files (ILF): DATABASE

Passenger records (individual and group).
 Reservation data, including booking history and payment records.
 Train schedules, including routes, availability, and pricing data.
 Staff records (for internal management).

External Interface Files (EIF): THIRD PARTY APPS

Central railway database for obtaining train schedules.
 External payment gateway for processing transactions.
 Integration with government identity verification systems (e.g., for ID checks).

Information Domain Value	Count	Weighting Factor	Total
External Input(EIs)	5	4	20
External Output(EOs)	4	5	20
External Inquiries (EQs)	6	4	24
Internal Logical Files(ILFs)	4	10	40
External Interface Files(ILFs)	3	7	21
Total Count			125

Complexity adjustment factor
CAF=0.65 + (0.01 * 42)

CAF=1.07

F.P= UFP *CAF

UFP = 125

FP = 125 * 1.07

FP = 133.75

32) online inventory management system

External Inputs (EIs):

Adding a new item to the inventory.

Updating item details (e.g., price, description).

Removing an item from the inventory.

Importing data from external sources (e.g., suppliers' catalogs).

Exporting inventory data for reporting.

External Outputs (EOs):

Generating inventory reports.

Producing purchase orders.

Sending low-stock alerts to staff.

Printing barcode labels.

Emailing item availability to customers.

External Inquiries (EQs):

Checking the current stock level for an item.

Searching for items based on specific criteria (e.g., category, location).

Querying item pricing and availability.

Viewing historical transaction records.

Tracking order statuses.

Internal Logical Files (ILFs):

Item master data (e.g., item details, stock levels).

Supplier information.

Customer information (if applicable).

Transaction history (e.g., sales, purchases).

User profiles and access rights.

External Interface Files (EIFs):

Supplier catalogs or price lists.

Integration with financial systems for payments.

Shipping and delivery tracking systems (if integrated).

Barcode scanning devices or systems.

Information Domain Value	Count	Weighting Factor	Total
External Input(EIs)	5	4	20
External Output(EOs)	5	5	25
External Inquiries (EQs)	5	4	20
Internal Logical Files(ILFs)	5	10	50
External Interface Files(ILFs)	4	7	28

Total Count			143
-------------	--	--	-----

Complexity adjustment factor
CAF=0.65 + (0.01 * 42)
CAF=1.07

$$F.P = UFP * CAF$$

$$UFP = 143$$

$$FP = 143 * 1.07$$

$$FP = 153.01$$

35. Perform function point estimation for a system like dream11.

External Inputs (EIs):

- Creating a new fantasy sports team.
- Modifying an existing team.
- Joining a contest or league.
- Editing user profile information.
- Depositing or withdrawing funds.
- Receiving notifications and alerts.

External Outputs (EOs):

- Displaying live scores and statistics during matches.
- Providing contest results and rankings.
- Sending contest winnings or rewards.
- Generating financial transaction history.
- Displaying recommendations and insights.

External Inquiries (EQs):

- Viewing player statistics and performance history.
- Checking upcoming match schedules.
- Exploring player availability and pricing.

Seeking customer support or FAQs.

Accessing terms and conditions.

Internal Logical Files (ILFs):

User account data (user profiles, preferences).

Player and team information.

Contest and league data (rules, schedules, prize details).

Transaction records (deposits, withdrawals, winnings).

Player performance statistics and scores.

External Interface Files (EIFs)

Real-time sports data feeds (for live scores and player statistics).

Payment gateways for handling financial transactions.

User authentication services (e.g., through Google or Facebook).

Notifications and push messaging services.

Player price and availability feeds from sports leagues.

Information Domain Value	Count	Weighting Factor	Total
External Input(EIs)	6	4	24
External Output(EOs)	5	5	25
External Inquiries (EQs)	5	4	20
Internal Logical Files(ILFs)	5	10	50
External Interface Files(ILFs)	5	7	35
Total Count			154

Complexity adjustment factor

$$CAF=0.65 + (0.01 * 42)$$

$$CAF=1.07$$

$$F.P = UFP * CAF$$

$$UFP = 154$$

$$FP = 154 * 1.07$$

$$FP = 164.78$$

17. Perform function point estimation for a system like magic bricks

External Inputs (EIs):

Posting a property listing (inputting property details, price, location, etc.).

Editing an existing property listing.

Inquiring about a property.

Subscribing to email alerts for property updates.

User registration and profile updates.

Reporting issues with listings.

External Outputs (EOs):

Displaying search results for property listings.

Showing property details and photos.

Sending email notifications for property updates.

Generating reports or summaries of property statistics.

Displaying advertisements.

External Inquiries (EQs):

Searching for properties based on various criteria (e.g., location, price, type).

Browsing property trends and market insights.

Accessing property buying or selling guides.

Frequently asked questions (FAQs) and support.

Internal Logical Files (ILFs):

Property listings and their details (e.g., property attributes, seller information).

User profiles and preferences.

Search history and saved property lists.

Advertisements data.

Property market data and trends.

External Interface Files (EIFs):

Geolocation services for property mapping.

Payment gateways for premium listings or subscriptions.
 External email services for notifications.

Information Domain Value	Count	Weighting Factor	Total
External Input(EIs)	6	4	24
External Output(EOs)	5	5	25
External Inquiries (EQs)	4	4	16
Internal Logical Files(ILFs)	5	10	50
External Interface Files(ILFs)	3	7	21
Total Count			136

Complexity adjustment factor

$$CAF = 0.65 + (0.01 * 42)$$

$$CAF = 1.07$$

$$F.P = UFP * CAF$$

$$UFP = 136$$

$$FP = 136 * 1.07$$

$$FP = 145.52$$

8. Perform function point estimation for online book store

External Inputs (EIs):

Searching for books by title, author, genre, or other criteria.
 Adding a book to the shopping cart.

Removing a book from the shopping cart.
Purchasing a book.
Writing a book review.
Updating user profile information.
Receiving and managing order confirmations and notifications.
Contacting customer support.

External Outputs (EOs):

Displaying search results for books.
Showing detailed book information and covers.
Generating order confirmations and receipts.
Sending email notifications for order status updates.
Displaying personalized book recommendations.
Showing advertisements and promotions.

External Inquiries (EQs):

Checking book availability.
Viewing book details and user reviews.
Accessing book recommendations and bestseller lists.
Browsing FAQs and customer support information.
Tracking order shipment status.

Internal Logical Files (ILFs):

Book catalog data (e.g., book details, availability, prices).
User profiles and preferences.
Order history and transaction records.
Book reviews and ratings.
Advertisements and promotional data.

External Interface Files (EIFs):

Integration with publishers' databases for book catalog data.
Payment gateways for processing transactions.
Shipping and delivery services for order tracking.
External email services for notifications.

Information Domain Value	Count	Weighting Factor	Total
External Input(EIs)	8	4	32
External	6	5	30

Output(EOs)			
External Inquiries (EQs)	5	4	20
Internal Logical Files(ILFs)	5	10	50
External Interface Files(ILFs)	4	7	28
Total Count			160

Complexity adjustment factor

$$CAF = 0.65 + (0.01 * 42)$$

$$CAF = 1.07$$

$$F.P = UFP * CAF$$

$$UFP = 160$$

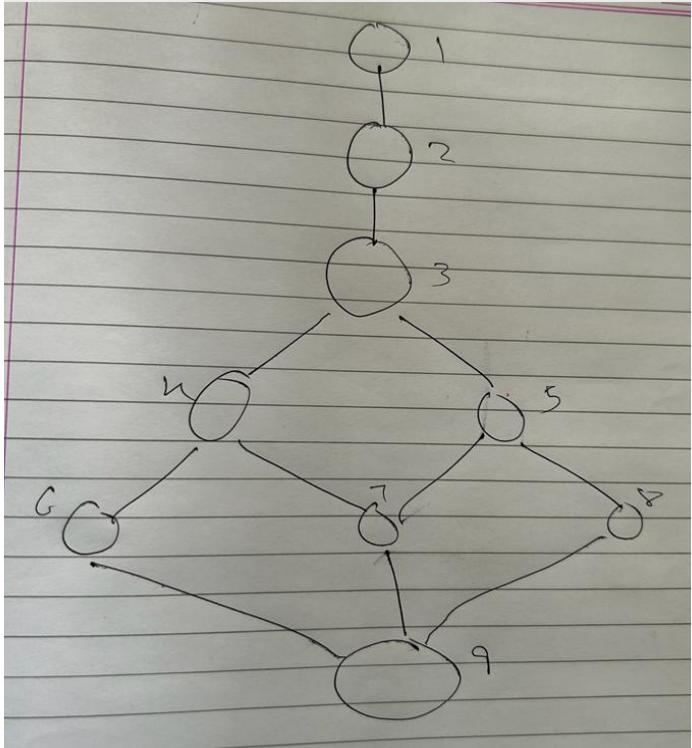
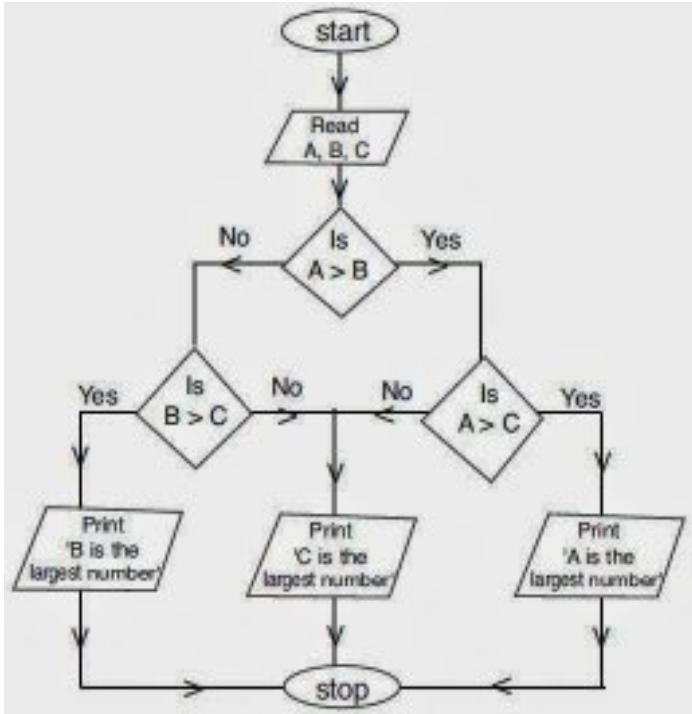
$$FP = 160 * 1.07$$

$$FP = 171.2$$

WHITEBOX AND BLACKBOX TESTING

* Data flow testing / control structure testing is not done *

11. Perform basis path testing and data flow testing on a program to find maximum between three numbers using if-else-if ladder



Paths \Rightarrow

1 - 2 - 3 - 4 - 5 - 6 - 9
1 - 2 - 3 - 4 - 7 - 9
1 - 2 - 3 - 5 - 7 - 9
1 - 2 - 3 - 5 - 8 - 9

Cycle Complexity $\Rightarrow V = E - N + 2$

$$= 11 - 9 + 2$$

$$= 4$$

\approx

Test Cases

Path 1: $n_2 > n_1 \& n_2 > n_3 \Rightarrow n_2$ is largest

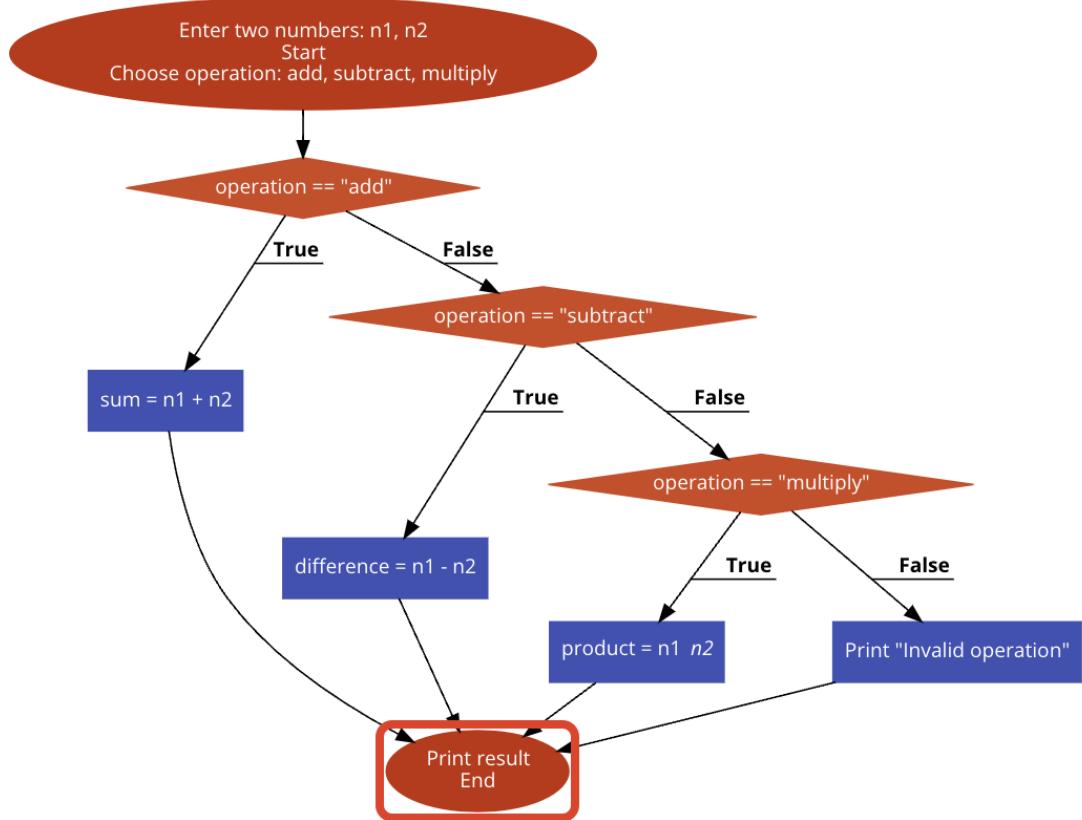
Path 2: $n_3 > n_2 \& n_3 > n_1 \Rightarrow n_3$ is largest

Path 3: $n_1 > n_2 \& n_1 > n_3 \Rightarrow n_1$ is largest

Path n: $n_1 > n_2 \& n_1 > n_3 \Rightarrow n_1$ is largest

28. Perform data flow and basis path testing for a program of arithmetic calculator which adds, subtracts and multiply 2 numbers.

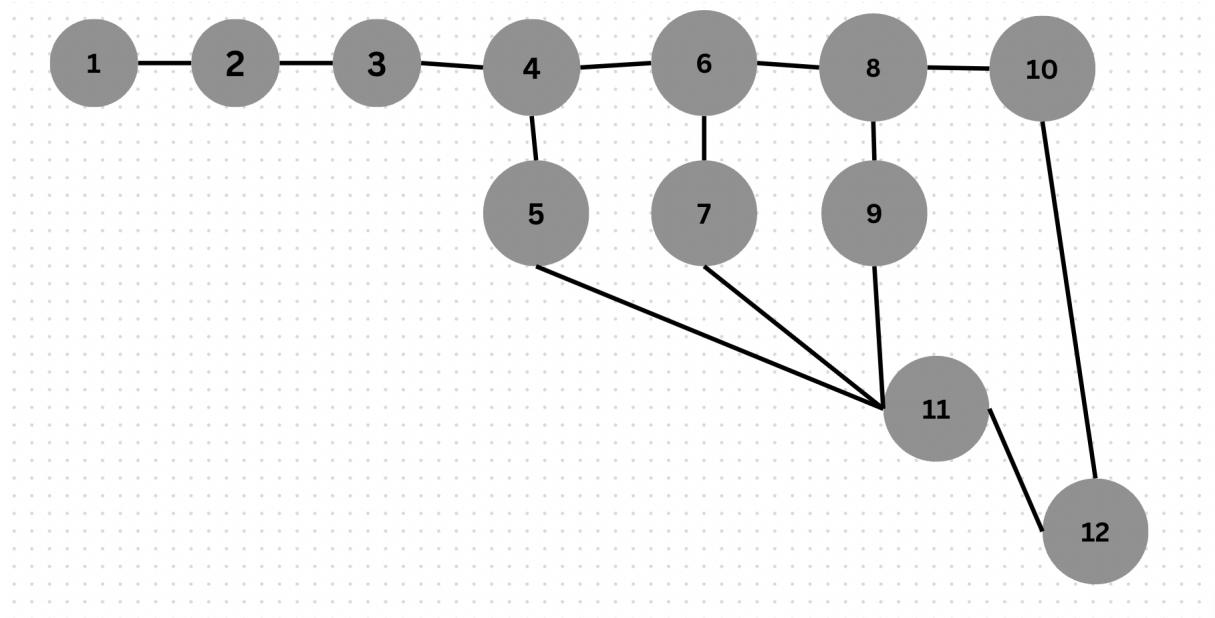
FlowChart



Steps

- Step 1- Start
- Step 2 - enter two numbers
- Step 3 - choose operation
- Step 4 - if Operation = add
- Step 5 - output = $n1 + n2$
- Step 6 - elseif Operation = subtract
- Step 7 - output = $n1 - n2$
- Step 8 - elseif operation = multiply
- Step 9 - output = $n1 * n2$
- Step 10 - invalid
- Step 11 - print "output"
- Step 12 - exit

DataFlow Graph



$$\begin{aligned} CC &= E - N + 2 \\ &= 14 - 12 + 2 \\ &= 4 \end{aligned}$$

There are 4 test cases

Path 1:

1->2->3->4->5->11->12

Test case: user asks for addition operation to be performed

Status : added both numbers

Output: n1 +n2

Path 2:

1->2->3->4->6->7->11->12

Test case: user asks for subtraction operation to be performed

Status : subtracted n1 from n2

Output: n1 +n2

Path 3:

1->2->3->4->5->6->8->9->11->12

Test case: user asks for multiplication operation to be performed

Status : multiply both numbers

Output: n1 +n2

Path 4:

1->2->3->4->5->6->8->10->12

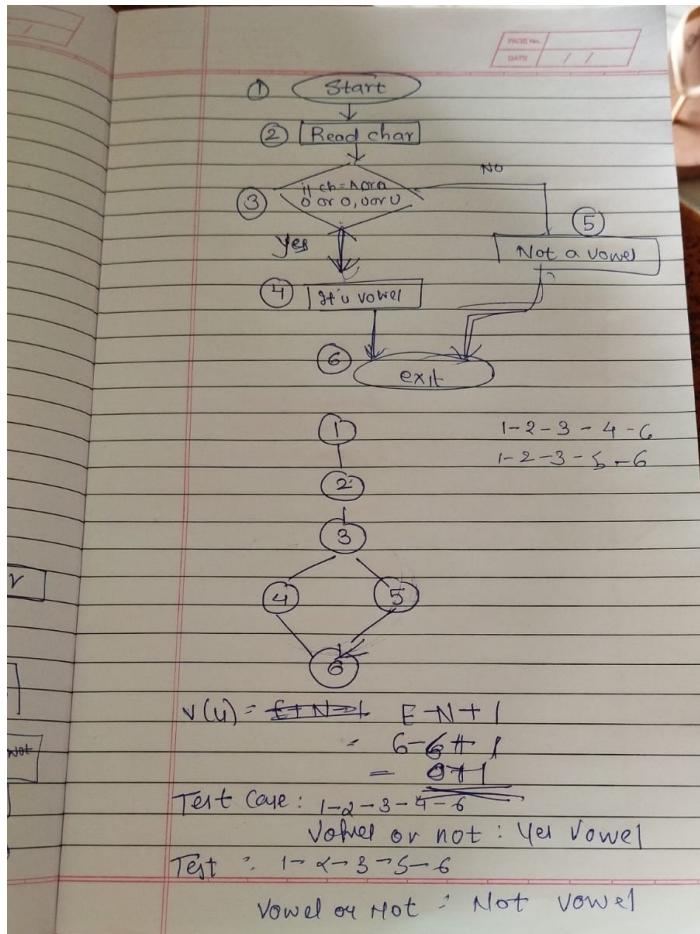
Test case: invalid operation

Status : no operation

Output: "invalid operation"

18. Perform data flow testing and basis path testing on a program to check whether entered character is vowel or not

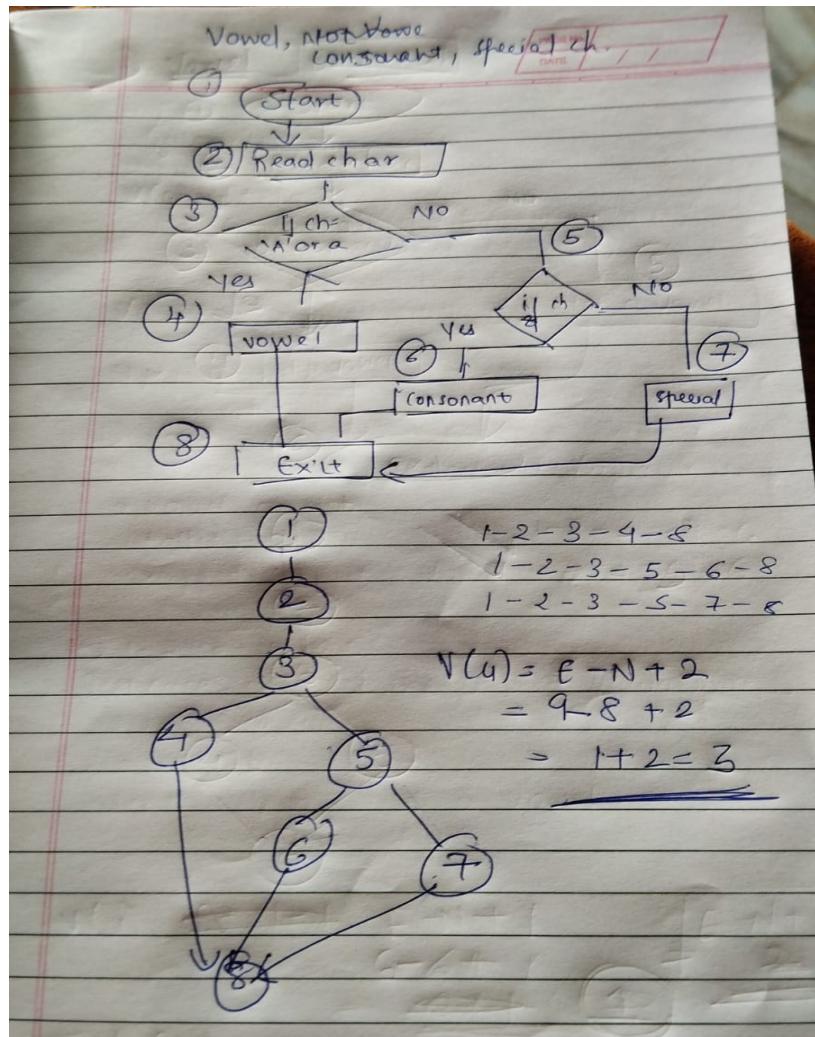
BASIC PATH



Cyclo Complexity above $\Rightarrow E-N + 2 = 6 - 6 + 2 = 2$

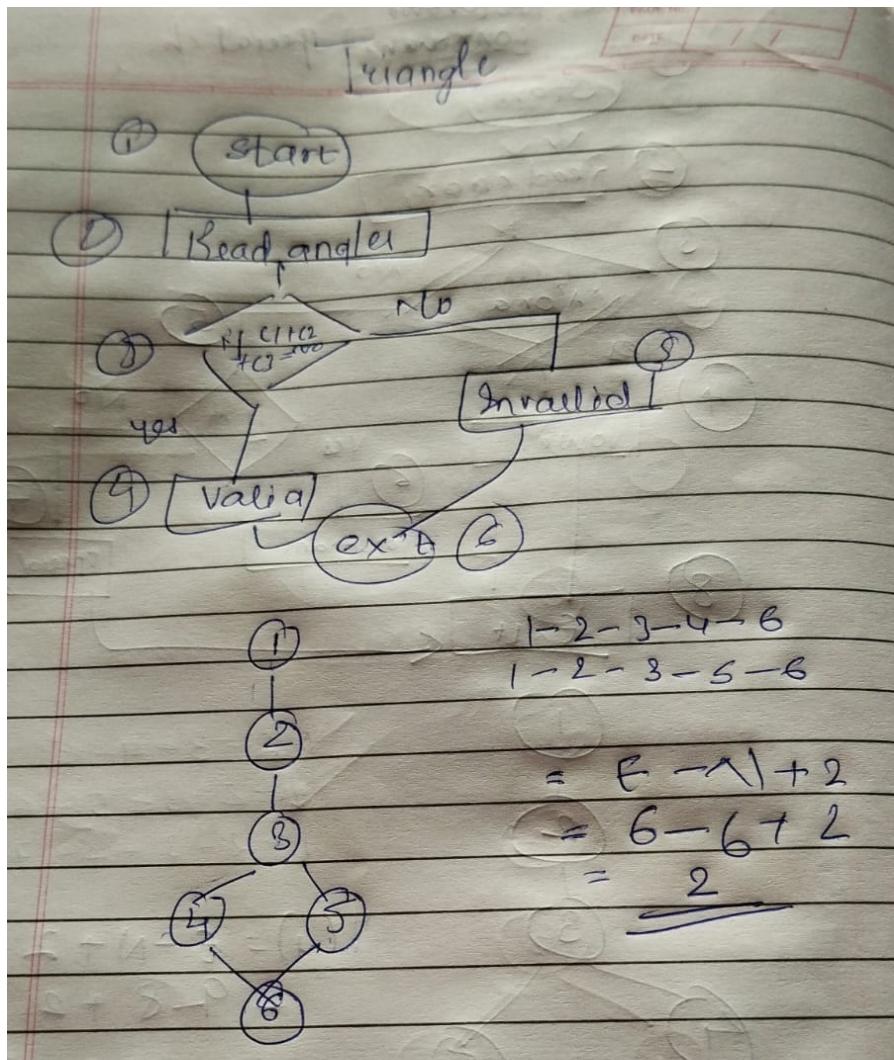
26. Perform basis path testing and control structure testing on a program to check whether entered character is vowel, consonant or special character

BASIC PATH



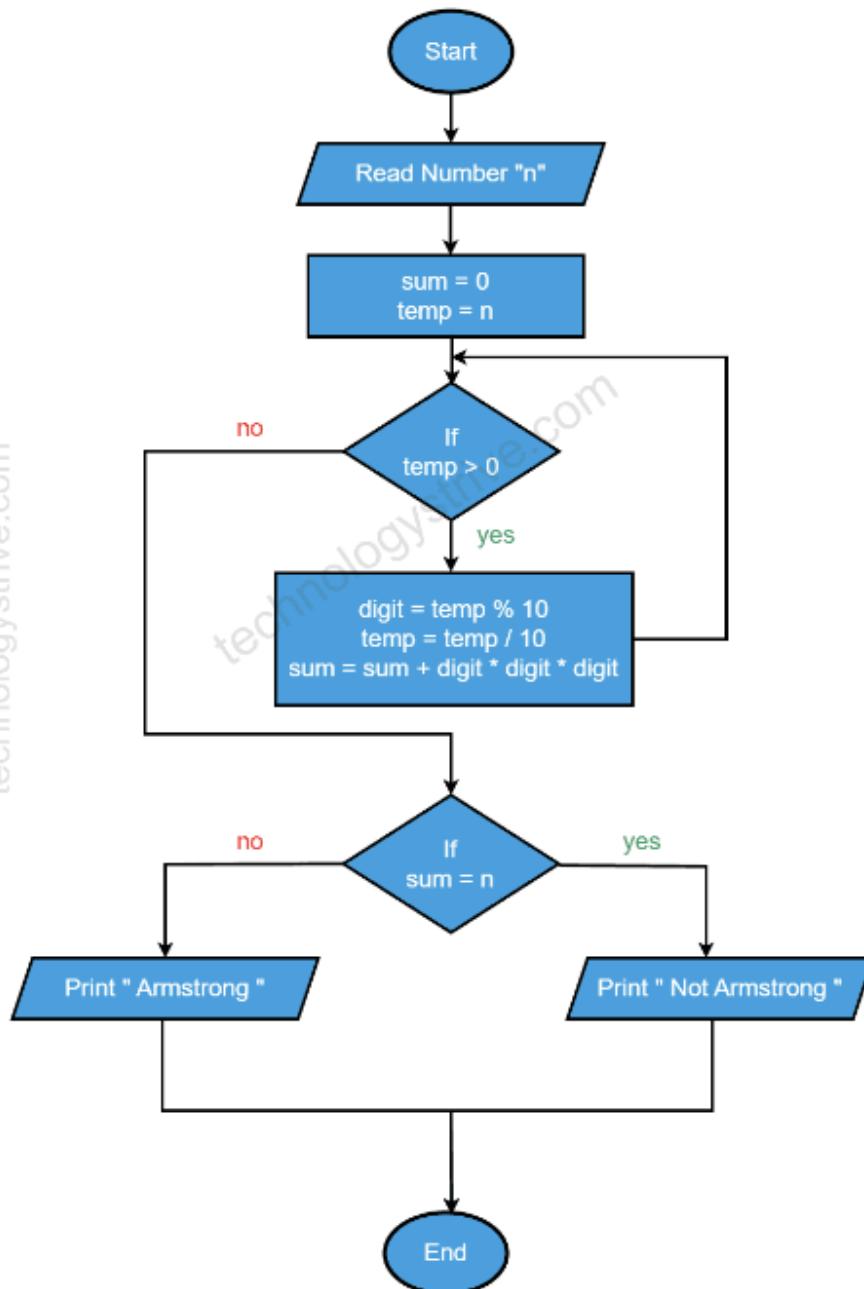
31. Perform basis path and data flow testing on a program to check whether triangle is valid or not if all angles of the triangle are given

BASIC PATH



41. Perform data flow testing and basis path testing for a program to check whether entered number is Armstrong or not

Flowchart

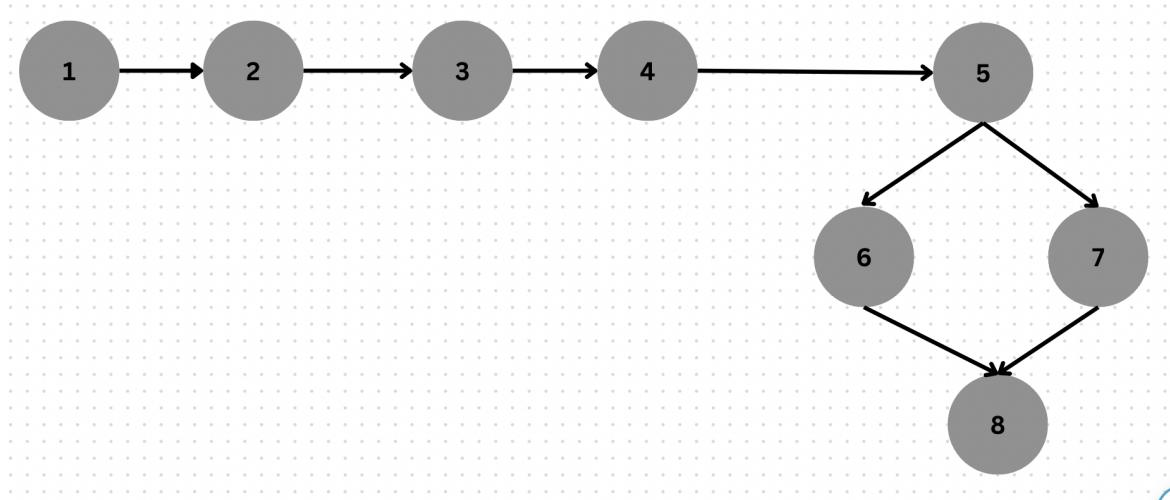


Steps

Step 1 - Start
Step 2 - input n
Step 3 - sum = 0 , temp = n

Step 4 - if temp >0
 armstrong logic
Step 5 - if sum = n
Step 6 - not armstrong
Step 7 - armstrong
Step 8 - end

DataGraph



$$\begin{aligned} CC &= E - N + 2 \\ &= 9 - 9 + 2 \\ &= 2 \end{aligned}$$

Number of test cases is 2

- Path 1:
1->2->3->4->5->6->8
Test case: number is not armstrong
Output: not armstrong
- Path 2:
1->2->3->4->5->7->8
Test case: number is armstrong
Output: armstrong

Cocomo Model

16. Use basic and intermediate Cocomo model for estimation of efforts and development time for online shopping system

Aim

To estimate project cost using COCOMO Model for online shopping system

Software Product	A _b	B _b	C _b	D _b
Organic	2.4	1.05	2.5	0.38
Semi- Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

$$E = a(KLOC)^b$$

Here E gives effort

BASIC COCOMO

Function Point of project : 100 (Assume)

for average LOC/FP for Java : 63

$$LOC = 63 \times 100 = 6300$$

$$KLOC = 6300/1000 = 6.3$$

Here as it is an Online Shopping System it needs to be an Embedded type as it has the highest complexity and experience. So we get:

$$a = 3.6, b = 1.20, c = 2.5, d = 0.32$$

$$E = 3.6(6.3)^{1.20} = 32.772$$

Effort for basic COCOMO model = 32.772

$$D(\text{Time}) = c(E)^d = 2.5(32.772)^{0.32} = 7.63$$

Time for basic COCOMO model = 7.63

For intermediate COCOMO model :

$$E = a(KLOC)^b * (EAF)$$

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very high	Extra high
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	..
AEXP	1.29	1.13	1.00	0.91	0.82	..
PCAP	1.42	1.17	1.00	0.86	0.70	..
VEXP	1.21	1.10	1.00	0.90
LEXP	1.14	1.07	1.00	0.95
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	..
TOOL	1.24	1.10	1.00	0.91	0.83	..
SCED	1.23	1.08	1.00	1.04	1.10	..

Figure 6. Ratings for cost drivers under personnel attributes and project attributes

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	..
DATA	..	0.94	1.00	1.08	1.16	..
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	1.00	1.11	1.30	1.66
STOR	1.00	1.06	1.21	1.56

COST DRIVERS:

Personal attributes:

Analyst Capability (ACAP) , High = 0.86
Application Experience (AEXP), Nominal = 1.00
Software Engineering capability (PCAP), High = 0.86
Experience using VM (VEXP), Nominal = 1.00
Programming language experience (LEXP), High = 0.95

Project attributes:

Applications of Software Eng Methods (MODP), Nominal = 1.00
Applications of Software Tools (TOOL), High = 0.91
Required Development Schedule (SCED), Nominal = 1.00

Product attributes:

Required Software reliability (RELY), HIgh= 1.15
Database size (DATA),Very High = 1.16
Product Complexity (CPLX), Very High = 1.30

Computer attributes:

Execution time constraints (TIME), High = 1.11
Main storage constraints (STOR), High = 1.06
Virtual Machine Volatility (VIRT), Nominal = 1.00
Required turnaround time (TURN), High = 1.07

Calculating EAF:

EAF = ACAP x AEXP x PCAP x VEXP x LEXP x MODP x TOOL x SCED x RELY
x DATA x CPLX x TIME x STOR x VIRT x TURN

EAF = 1.39

Calculating E:

substituting values in the equation:

$$E = 32.772 \times 1.39$$

$$\mathbf{E = 45.55}$$

Effort for intermediate COCOMO model = 45.55

Time Required:

$$D=c(E)^d$$

$$D = 2.5(45.55)^{0.32}$$

$$\mathbf{D = 8.48}$$

19. Use basic and intermediate Cocomo model for estimation of efforts and development time for hospital management system.

To estimate project cost using COCOMO Model for hospital management system

Software Product	A _b	B _b	C _b	D _b
Organic	2.4	1.05	2.5	0.38
Semi- Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

BASIC COCOMO

Function Point of project : 100 (Assume)

for average LOC/FP for Java : 63

$$\text{LOC} = 63 \times 100 = 6300$$

$$\text{KLOC} = 6300/1000 = 6.3$$

Here as it is an Hospital management system it needs to be an Organic type as it has the lower complexity and experience. So we get:

$$a = 2.4, b = 1.05, c = 2.5, d = 0.38$$

$$E = 2.4(6.3)^{1.05} = 16.577$$

Effort for basic COCOMO model = 16.577

$$D(\text{Time}) = c(E)^d = 2.5(16.577)^{0.38} = 7.26$$

Time for basic COCOMO model = 7.26

For intermediate COCOMO model :

$$E = a(KLOC)^b * (EAF)$$

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very high	Extra high
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	..
AEXP	1.29	1.13	1.00	0.91	0.82	..
PCAP	1.42	1.17	1.00	0.86	0.70	..
VEXP	1.21	1.10	1.00	0.90
LEXP	1.14	1.07	1.00	0.95
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	..
TOOL	1.24	1.10	1.00	0.91	0.83	..
SCED	1.23	1.08	1.00	1.04	1.10	..

Figure 6. Ratings for cost drivers under personnel attributes and project attributes

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	..
DATA	..	0.94	1.00	1.08	1.16	..
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	1.00	1.11	1.30	1.66
STOR	1.00	1.06	1.21	1.56
VIRT	..	0.87	1.00	1.15	1.30	..
TURN	..	0.87	1.00	1.07	1.15	..

Figure 5. Ratings for cost drivers under product attributes and computer attributes

COST DR

Personal attributes:

Analyst Capability (ACAP) , Nominal = 1.00

Application Experience (AEXP), Nominal = 1.00

Software Engineering capability (PCAP), High = 0.86

Experience using VM (VEXP), Nominal = 1.00

Programming language experience (LEXP), High = 0.95

Project attributes:

Applications of Software Eng Methods (MODP), Nominal = 1.00

Applications of Software Tools (TOOL), High = 0.91

Required Development Schedule (SCED), Nominal = 1.00

Product attributes:

Required Software reliability (RELY), Very HIgh= 1.40

Database size (DATA),Very High = 1.16

Product Complexity (CPLX), High = 1.15

Computer attributes:

Execution time constraints (TIME), High = 1.11

Main storage constraints (STOR), Nominal = 1.00

Virtual Machine Volatility (VIRT), Nominal = 1.00

Required turnaround time (TURN), High = 1.07

Calculating EAF:

EAF = ACAP x AEXP x PCAP x VEXP x LEXP x MODP x TOOL x SCED x RELY
x DATA x CPLX x TIME x STOR x VIRT x TURN

EAF = 1.64

Calculating E:

substituting values in the equation:

$$E = 16.577 \times 1.64$$

$$\mathbf{E = 27.186}$$

Effort for intermediate COCOMO model = 27.186

Time Required:

$$D=c(E)^d$$

$$D= 2.5(27.186)^{0.38}$$

$$\mathbf{D= 8.76}$$

22. Use basic and intermediate Cocomo model for estimation of efforts and development time for online hotel management system

To estimate project cost using COCOMO Model for online hotel management system

Software Product	A _b	B _b	C _b	D _b
Organic	2.4	1.05	2.5	0.38
Semi- Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

$E = a(KLOC)^b$
Here E gives effort

BASIC COCOMO

Function Point of project : 100 (Assume)

for average LOC/FP for Java : 63

$$LOC = 63 \times 100 = 6300$$

$$KLOC = 6300/1000 = 6.3$$

Here as it is an online Hotel management system it needs to be an Organic type as it has the lower complexity and experience. So we get:

$$a = 2.4, b = 1.05, c = 2.5, d = 0.38$$

$$E = 2.4(6.3)^{1.05} = 16.577$$

Effort for basic COCOMO model = 16.577

$$D(\text{Time}) = c(E)^d = 2.5(16.577)^{0.38} = 7.26$$

Time for basic COCOMO model = 7.26

For intermediate COCOMO model :

$$E = a(KLOC)^b * (EAF)$$

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very high	Extra high
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	..
AEXP	1.29	1.13	1.00	0.91	0.82	..
PCAP	1.42	1.17	1.00	0.86	0.70	..
VEXP	1.21	1.10	1.00	0.90
LEXP	1.14	1.07	1.00	0.95
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	..
TOOL	1.24	1.10	1.00	0.91	0.83	..
SCED	1.23	1.08	1.00	1.04	1.10	..

Figure 6. Ratings for cost drivers under personnel attributes and project attributes

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	..
DATA	..	0.94	1.00	1.08	1.16	..
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	1.00	1.11	1.30	1.66
STOR	1.00	1.06	1.21	1.56
VIRT	..	0.87	1.00	1.15	1.30	..
TURN	..	0.87	1.00	1.07	1.15	..

Figure 5. Ratings for cost drivers under product attributes and computer attributes

COST DRIVERS:

Personal attributes:

Analyst Capability (ACAP) , High = 0.86

Application Experience (AEXP), Nominal = 1.00

Software Engineering capability (PCAP), High = 0.86

Experience using VM (VEXP), Nominal = 1.00

Programming language experience (LEXP), High = 0.95

Project attributes:

Applications of Software Eng Methods (MODP), Nominal = 1.00

Applications of Software Tools (TOOL), High = 0.91

Required Development Schedule (SCED), Nominal = 1.00

Product attributes:

Required Software reliability (RELY), High= 1.15

Database size (DATA), Very High = 1.16

Product Complexity (CPLX), High = 1.15

Computer attributes:

Execution time constraints (TIME), High = 1.11

Main storage constraints (STOR), Nominal = 1.00

Virtual Machine Volatility (VIRT), High = 1.15

Required turnaround time (TURN), High = 1.07

Calculating EAF:

$$\text{EAF} = \text{ACAP} \times \text{AEXP} \times \text{PCAP} \times \text{VEXP} \times \text{LEXP} \times \text{MODP} \times \text{TOOL} \times \text{SCED} \times \text{RELY} \\ \times \text{DATA} \times \text{CPLX} \times \text{TIME} \times \text{STOR} \times \text{VIRT} \times \text{TURN}$$

$$\text{EAF} = 1.33$$

Calculating E:

substituting values in the equation:

$$E = 16.577 \times 1.33$$

$$\mathbf{E = 22.047}$$

Effort for intermediate COCOMO model = 22.047

Time Required:

$$D=c(E)^d$$

$$D= 2.5(22.047)^{0.38}$$

$$\mathbf{D= 8.09}$$

23. Use basic and intermediate Cocomo model for estimation of efforts and development time for online banking systems.

To estimate project cost using COCOMO Model for online banking systems

Software Product	A _b	B _b	C _b	D _b
Organic	2.4	1.05	2.5	0.38
Semi- Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

$$E = a(KLOC)^b$$

Here E gives effort

BASIC COCOMO

Function Point of project : 100 (Assume)

for average LOC/FP for Java : 63

$$LOC = 63 \times 100 = 6300$$

$$KLOC = 6300/1000 = 6.3$$

Here as it is an online Hotel management system it needs to be an Embedded type as it has the high complexity and experience with large staff size. So we get:

$$a = 3.6, b = 1.20, c = 2.5, d = 0.32$$

$$E = 3.6(6.3)^{1.20} = 32.772$$

Effort for basic COCOMO model = 32.772

$$D(\text{Time}) = c(E)^d = 2.5(32.772)^{0.32} = 7.63$$

Time for basic COCOMO model = 7.63

For intermediate COCOMO model :

$$E = a(KLOC)^b * (EAF)$$

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very high	Extra high
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	..
AEXP	1.29	1.13	1.00	0.91	0.82	..
PCAP	1.42	1.17	1.00	0.86	0.70	..
VEXP	1.21	1.10	1.00	0.90
LEXP	1.14	1.07	1.00	0.95
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	..
TOOL	1.24	1.10	1.00	0.91	0.83	..
SCED	1.23	1.08	1.00	1.04	1.10	..

Figure 6. Ratings for cost drivers under personnel attributes and project attributes

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	..
DATA	..	0.94	1.00	1.08	1.16	..
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	1.00	1.11	1.30	1.66
STOR	1.00	1.06	1.21	1.56
VIRT	..	0.87	1.00	1.15	1.30	..
TURN	..	0.87	1.00	1.07	1.15	..

Figure 5. Ratings for cost drivers under product attributes and computer attributes

COST DRIVERS:

Personal attributes:

Analyst Capability (ACAP) ,Very High = 0.71

Application Experience (AEXP), Very High = 0.82

Software Engineering capability (PCAP), High = 0.86

Experience using VM (VEXP), High = 0.90

Programming language experience (LEXP), High = 0.95

Project attributes:

Applications of Software Eng Methods (MODP), High = 0.91

Applications of Software Tools (TOOL), High = 0.91

Required Development Schedule (SCED), Nominal = 1.00

Product attributes:

Required Software reliability (RELY), Very HIgh= 1.40

Database size (DATA),Very High = 1.16

Product Complexity (CPLX), High = 1.15

Computer attributes:

Execution time constraints (TIME),Very High = 1.30

Main storage constraints (STOR), Nominal = 1.00

Virtual Machine Volatility (VIRT), High = 1.15

Required turnaround time (TURN), High = 1.07

Calculating EAF:

EAF = ACAP x AEXP x PCAP x VEXP x LEXP x MODP x TOOL x SCED x RELY
x DATA x CPLX x TIME x STOR x VIRT x TURN

EAF = 1.05

Calculating E:

substituting values in the equation:

$$E = 32.772 \times 1.05$$

$$\mathbf{E = 34.41}$$

Effort for intermediate COCOMO model = 22.047

Time Required:

$$D=c(E)^d$$

$$D= 2.5(34.41)^{0.32}$$

$$\mathbf{D= 7.75}$$

34. Use basic and intermediate Cocomo model for estimation of efforts and development time for a system like magic bricks

Aim

To estimate project cost using COCOMO Model for Magic Bricks

Description The table for constants for Basic COCOMO model is as follows:

Software Product	A _b	B _b	C _b	D _b
Organic	2.4	1.05	2.5	0.38
Semi- Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Here E gives effort

BASIC COCOMO

Function Point of project : 100 (Assume)

for average LOC/FP for Java : 63

$$\text{LOC} = 63 \times 100 = 6300$$

$$\text{KLOC} = 6300/1000 = 6.3$$

Here as it is an Online book Store it needs to be an Embedded type as it has the highest complexity and experience. So we get:

$$a = 3.6, b = 1.20, c = 2.5, d = 0.32$$

$$E = 3.6(6.3)^{1.20} = 32.772619$$

Effort for basic COCOMO model = 32.772619

$$D(\text{Time}) = c(E)^d = 2.5(32.772619)^{0.32} = 7.636$$

Time for basic COCOMO model = 7.636

For intermediate COCOMO model

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very high	Extra high
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	..
AEXP	1.29	1.13	1.00	0.91	0.82	..
PCAP	1.42	1.17	1.00	0.86	0.70	..
VEXP	1.21	1.10	1.00	0.90
LEXP	1.14	1.07	1.00	0.95
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	..
TOOL	1.24	1.10	1.00	0.91	0.83	..
SCED	1.23	1.08	1.00	1.04	1.10	..

Figure 6. Ratings for cost drivers under personnel attributes and project attributes

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	..
DATA	..	0.94	1.00	1.08	1.16	..
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	1.00	1.11	1.30	1.66
STOR	1.00	1.06	1.21	1.56
VIRT	..	0.87	1.00	1.15	1.30	..
TURN	..	0.87	1.00	1.07	1.15	..

Figure 5. Ratings for cost drivers under product attributes and computer attributes

$$E = a(KLOC)^b * (EAF)$$

COST DRIVERS:

Personal attributes:

Analyst Capability (ACAP) , High = 0.86

Application Experience (AEXP), Nominal = 1.00

Software Engineering capability (PCAP), High = 0.86

Experience using VM (VEXP), High = 0.90

Programming language experience (LEXP), High = 0.95

Project attributes:

Applications of Software Eng Methods (MODP), high = 0.91
Applications of Software Tools (TOOL), High = 0.91
Required Development Schedule (SCED), High= 1.04

Product attributes:

Required Software reliability (RELY), HIgh= 1.15
Database size (DATA), High = 1.08
Product Complexity (CPLX), Very High = 1.30

Computer attributes:

Execution time constraints (TIME), High = 1.11
Main storage constraints (STOR), High = 1.06
Virtual Machine Volatility (VIRT), high = 1.15
Required turnaround time (TURN), High = 1.07

Calculating EAF:

EAF = ACAP x AEXP x PCAP x VEXP x LEXP x MODP x TOOL x SCED x RELY
x DATA x CPLX x TIME x STOR x VIRT x TURN)

EAF = 0.9361

Calculating E:

substituting values in the equation:

$$E = 32.772619 \times 0.9361$$

$$E = 30.678$$

Effort for intermediate COCOMO model = 30.678

Time Required:

$$D=c(E)^d$$

$$= 2.5(30.678)^{0.32} = 7.476$$

33. Use basic and intermediate Cocomo model for estimation of efforts and development time for system like dream11

Aim

To estimate project cost using COCOMO Model for DREAM 11

Description The table for constants for Basic COCOMO model is as follows:

Software Product	A_b	B_b	C_b	D_b
Organic	2.4	1.05	2.5	0.38
Semi- Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Here E gives effort

BASIC COCOMO

Function Point of project : 100 (Assume)

for average LOC/FP for Java : 63

$$\text{LOC} = 63 \times 100 = 6300$$

$$\text{KLOC} = 6300/1000 = 6.3$$

Here as it is an Online book Store it needs to be an Embedded type as it has the highest complexity and experience. So we get:

$$a = 3.6, b = 1.20, c = 2.5, d = 0.32$$

$$E = 3.6(6.3)^{1.20} = 32.772619$$

Effort for basic COCOMO model = 32.772619

$$D(\text{Time}) = c(E)^d = 2.5(32.772619)^{0.32} = 7.636$$

Time for basic COCOMO model = 7.636

For intermediate COCOMO model

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very high	Extra high
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	..
AEXP	1.29	1.13	1.00	0.91	0.82	..
PCAP	1.42	1.17	1.00	0.86	0.70	..
VEXP	1.21	1.10	1.00	0.90
LEXP	1.14	1.07	1.00	0.95
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	..
TOOL	1.24	1.10	1.00	0.91	0.83	..
SCED	1.23	1.08	1.00	1.04	1.10	..

Figure 6. Ratings for cost drivers under personnel attributes and project attributes

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	..
DATA	..	0.94	1.00	1.08	1.16	..
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	1.00	1.11	1.30	1.66
STOR	1.00	1.06	1.21	1.56
VIRT	..	0.87	1.00	1.15	1.30	..
TURN	..	0.87	1.00	1.07	1.15	..

Figure 5. Ratings for cost drivers under product attributes and computer attributes

$$E = a(KLOC)^b * (EAF)$$

COST DRIVERS:

Personal attributes:

Analyst Capability (ACAP) , High = 0.86

Application Experience (AEXP), Very High = 0.82

Software Engineering capability (PCAP), High = 0.86

Experience using VM (VEXP), High = 0.90

Programming language experience (LEXP), High = 0.95

Project attributes:

Applications of Software Eng Methods (MODP), high = 0.91
Applications of Software Tools (TOOL), High = 0.91
Required Development Schedule (SCED), High= 1.04

Product attributes:

Required Software reliability (RELY), HIgh= 1.15
Database size (DATA), High = 1.08
Product Complexity (CPLX), Very High = 1.30

Computer attributes:

Execution time constraints (TIME), High = 1.11
Main storage constraints (STOR), High = 1.06
Virtual Machine Volatility (VIRT), high = 1.15
Required turnaround time (TURN), High = 1.07

Calculating EAF:

EAF = ACAP x AEXP x PCAP x VEXP x LEXP x MODP x TOOL x SCED x RELY
x DATA x CPLX x TIME x STOR x VIRT x TURN)

EAF = 0.9139

Calculating E:

substituting values in the equation:

$$E = 32.772619 \times 0.9139$$

$$E = 29.955$$

Effort for intermediate COCOMO model = 29.955

Time Required:

$$D=c(E)^d$$

$$= 2.5(29.955)^{0.32} = 7.420$$

10. Use basic and intermediate Cocomo model for estimation of efforts and development time for online book store

Aim

To estimate project cost using COCOMO Model for Online Book Store

Software Product	A _b	B _b	C _b	D _b
Organic	2.4	1.05	2.5	0.38
Semi- Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Here E gives effort

BASIC COCOMO

Function Point of project : 100 (Assume)

for average LOC/FP for Java : 63

$$\text{LOC} = 63 \times 100 = 6300$$

$$\text{KLOC} = 6300/1000 = 6.3$$

Here as it is an Online book Store it needs to be an Organic type as it has the highest complexity and experience. So we get:

$$a = 2.4, b = 1.05, c = 2.5, d = 0.38$$

$$E = 2.4(6.3)^{1.05} = 16.577491$$

Effort for basic COCOMO model = 16.577491

$$D(\text{Time}) = c(E)^d = 2.5(16.577491)^{0.38} = 7.267033$$

Time for basic COCOMO model = 7.267033
 Average Staff Size : E/T = 16.577491/7.267033 = 2.28

For intermediate COCOMO model

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very high	Extra high
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	..
AEXP	1.29	1.13	1.00	0.91	0.82	..
PCAP	1.42	1.17	1.00	0.86	0.70	..
VEXP	1.21	1.10	1.00	0.90
LEXP	1.14	1.07	1.00	0.95
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	..
TOOL	1.24	1.10	1.00	0.91	0.83	..
SCED	1.23	1.08	1.00	1.04	1.10	..

Figure 6. Ratings for cost drivers under personnel attributes and project attributes

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	..
DATA	..	0.94	1.00	1.08	1.16	..
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	1.00	1.11	1.30	1.66
STOR	1.00	1.06	1.21	1.56
VIRT	..	0.87	1.00	1.15	1.30	..
TURN	..	0.87	1.00	1.07	1.15	..

Figure 5. Ratings for cost drivers under product attributes and computer attributes

$$E = a(KLOC)^b * (EAF)$$

COST DRIVERS:

Personal attributes:

Analyst Capability (ACAP) , Nominal = 1.00
Application Experience (AEXP), Nominal = 1.00
Software Engineering capability (PCAP), Nominal = 1.0
Experience using VM (VEXP), Nominal = 1.00
Programming language experience (LEXP), High = 0.95

Project attributes:

Applications of Software Eng Methods (MODP), high = 0.91
Applications of Software Tools (TOOL), High = 0.91
Required Development Schedule (SCED), Nominal = 1.00

Product attributes:

Required Software reliability (RELY), HIgh= 1.15
Database size (DATA), High = 1.08
Product Complexity (CPLX), Very High = 1.30

Computer attributes:

Execution time constraints (TIME), High = 1.11
Main storage constraints (STOR), High = 1.06
Virtual Machine Volatility (VIRT), Nominal = 1.00
Required turnaround time (TURN), High = 1.07

Calculating EAF:

EAF = ACAP x AEXP x PCAP x VEXP x LEXP x MODP x TOOL x SCED x RELY
x DATA x CPLX x TIME x STOR x VIRT x TURN)

EAF = 0.90618

Calculating E:

substituting values in the equation:

E = 16.577491x 0.90618

E = 15.022190

Effort for intermediate COCOMO model = 15.022190

Time Required:

D=c(E)d

D= 2.5(15.022190)0.38

D= 7.000

Average Staff Size : E/T = 15.022190/7 = 2.146

In COCOMO, projects are categorized into three types:

1. Organic
 2. Semidetached
 3. Embedded
- 1) **Organic** – A software project is said to be an organic type
- if the team size required is adequately small,
 - the problem is well understood and has been solved in the past
 - also, the team members have a nominal experience regarding the problem.
 - Examples of this type of projects are simple business systems, simple inventory management systems, and data processing systems.
- 2) **Semi-detached** – A software project is said to be a Semi-detached type if
- the development team consists of a mixture of experienced and inexperienced staff.
 - Team members may have finite experience in related systems but may be unfamiliar with some aspects of the system being developed
 - difficult to develop compared to the organic ones and require more experience and better guidance and creativity.
 - Eg: Compilers or different Embedded Systems can be considered of Semi-Detached type.

Embedded – A software project requiring the highest level of complexity, creativity, and experience

- Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced
- the software being developed is strongly coupled to complex hardware, or if the stringent regulations on the operational method exist.
- For Example: ATM, Air Traffic control.

$$E = a(KLOC)^b$$

$$time = c(Effort)^d$$

$$Person\ required = Effort/time$$

Software Requirements Specification (SRS) IEEE

Online Banking System

1. Introduction

1.1 Purpose

The Online Banking System is designed to provide a comprehensive platform for managing and conducting various financial transactions and services. This document outlines the scope, product functions, and user classes and characteristics of the system.

1.2. Scope

The Online Banking System aims to provide a secure and user-friendly online banking experience for customers. It enables users to perform a wide range of financial activities, including but not limited to:

It includes functions for account management, funds transfer, bill payment, transaction history, and alerts.

Customers can open various account types, transfer funds, pay bills, view transaction history, and receive notifications.

The system aims to streamline financial transactions and enhance customer experience.

2. Overall Description

2.1 Product Functions

1. Account Management

User Registration: New users can register by providing personal information and verifying their identity.

Account Creation: Users can create various types of accounts, such as savings, checking, and credit card accounts.

Account Information: Users can view account details, including balances and transaction history.

2. Funds Transfer

Internal Transfers: Users can transfer funds between their own accounts.

External Transfers: Users can initiate funds transfers to external accounts with proper authorization and verification.

Scheduled Transfers: Users can set up recurring transfers for regular payments.

3. Bill Payment

Pay Bills: Users can pay bills to registered billers or service providers.

Scheduled Payments: Users can schedule future bill payments and manage payment history.

4. Transaction History

Transaction Records: Users can view detailed transaction records, including date, time, amount, and transaction type.

5. Account Statements

Statement Generation: Users can generate account statements for a specified period.

Statement Download: Users can download account statements in PDF format.

2.2 User Classes and Characteristics

1. Registered Users

Registered users include individuals, businesses, and organizations.

They are required to create an account and authenticate themselves to access the system.

Registered users have varying levels of access based on their account type.

2. Administrators

Administrators are responsible for system maintenance, user management, and security.

They can add or remove users, manage accounts, and configure system settings.

3. Customer Support

Customer support agents have access to user information and transaction records to assist users with inquiries and issues.

Online Inventory Management System

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document outlines the requirements for the Online Inventory Management System. The system is designed to efficiently manage and track inventory for businesses, enabling users to streamline their inventory-related operations.

1.2 Scope

The Online Inventory Management System is intended to provide a comprehensive solution for businesses to manage their inventory. The scope of the system includes the following functionalities:

Inventory Management: Users can add, update, and delete products in their inventory.

Product Tracking: Users can track product quantities, locations, and supplier details.

Order Management: Users can create, process, and track purchase orders and sales orders.

Reporting: Users can generate inventory reports, including stock levels, sales reports, and supplier performance reports.

User Authentication: Secure access is provided with user roles and permissions.

2. Overall Description

2.1 Product functions

1. Inventory Management

Product Addition: Users can add new products to the inventory, specifying product details such as name, category, price, and quantity.

Product Updates: Users can edit product information and update quantities as products are received or sold.

Product Deletion: Users can remove products that are no longer in stock.

2. Product Tracking

Quantity Tracking: The system automatically updates product quantities as purchases and sales are recorded.

Location Tracking: Users can assign and track product locations within the inventory.

Supplier Information: Users can link products to specific suppliers, including contact details and pricing agreements.

3. Order Management

Purchase Orders: Users can create and manage purchase orders for replenishing stock.

Sales Orders: Users can create and track sales orders for managing customer sales.

Order Processing: The system updates inventory quantities and financial records upon order processing.

4. Reporting

Stock Levels: Users can generate reports to view current stock levels for all products.

Sales Reports: Users can access reports detailing sales performance and revenue.

Supplier Performance: Reports on supplier delivery times and pricing agreements are available.

2.2 User Classes and Characteristics

1. Administrators

Administrators have full system access, allowing them to manage users and system settings.

They can add, edit, or delete products, process orders, and generate reports.

Administrators have the authority to define user roles and permissions.

2. Inventory Managers

Inventory managers have access to inventory-related functions.

They can add, update, or delete products, track quantities, and manage orders.

Inventory managers do not have access to system-wide settings.

3. Sales Representatives

Sales representatives can access sales order management features to create and process sales orders.

They do not have access to inventory management or supplier-related functions.

Sales representatives can generate sales reports for their sales activities.

4. Suppliers

Suppliers have restricted access to view and update their product offerings.

They can track product availability, update pricing agreements, and receive purchase orders.

(SRS) for Online Food Delivery System

1. Introduction

1.1 Purpose

The purpose of this SRS is to define the scope, product functions, user classes, and their characteristics for the Online Food Delivery System, hereinafter referred to as "the system." This document will serve as a reference for the development team to design and build the system according to the specified requirements.

1.2 Scope

The Online Food Delivery System is a web-based platform that allows users to browse, order, and pay for food items from various restaurants. The system will enable restaurant owners to manage their menus and orders efficiently. Users will have the option to customize orders, track deliveries, and provide feedback. The system aims to provide a convenient and user-friendly experience for both customers and restaurant owners.

2. Overall Description

2.1 Product functions

1. Customer Functions

User Registration: Customers can create accounts by providing their personal information.

Browsing Restaurants: Users can browse a list of available restaurants and view their menus.

Order Placement: Customers can add food items to their cart, customize their orders, and place orders.

Payment: Users can pay for their orders using various payment methods, including credit cards and digital wallets.

Order Tracking: Customers can track the status and location of their food delivery in real-time.

Feedback and Ratings: Users can provide feedback and ratings for restaurants and delivery services.

Order History: Customers can view their order history and reorder previous meals.

2. Restaurant Owner Functions

Restaurant Registration: Owners can register their restaurants and provide menu details.
Menu Management: Restaurant owners can add, edit, and delete menu items and update pricing.

Order Management: Owners can view and manage incoming orders, including order confirmation and cancellation.

Restaurant Dashboard: Access to real-time data and analytics to help optimize operations.

Customer Communication: Owners can communicate with customers regarding orders and special requests.

2.2 User Classes and Characteristics

1. Customer User Class

Characteristics:

Customers are end-users who use the system to order food from restaurants.

They may vary in age, technical proficiency, and dietary preferences.

Customers require a user-friendly interface and a smooth ordering process.

They may have personal accounts to store their order history and preferences.

2. Restaurant Owner User Class

Characteristics:

Restaurant owners are the administrators of the system for their respective restaurants.

They are responsible for menu management and order processing.

They require access to a restaurant-specific dashboard and order management tools.

Restaurant owners may have varying levels of technical expertise.

(SRS) for HealthifyMe-like App

1. Introduction

1.1 Purpose

The purpose of this SRS is to define the scope, product functions, and user classes and their characteristics for a system similar to the HealthifyMe app, hereinafter referred to as "the system." This document will serve as a reference for the development team to design and build the system according to the specified requirements.

1.2 Scope

The HealthifyMe-like system is a mobile application and web platform designed to promote health and wellness by providing users with personalized fitness and nutrition guidance. The system enables users to set fitness goals, track their daily activities, and manage their diet through a user-friendly and interactive interface. It offers features such as meal tracking, workout tracking, and personalized coaching.

2 Overall Description

2.1 Product Functions

1. User Functions

User Registration: Users can create accounts by providing personal information, health goals, and dietary preferences.

Profile Management: Users can update their profiles, including weight, height, age, and fitness goals.

Calorie Tracking: Users can log daily food intake and receive real-time calorie consumption feedback.

Workout Tracking: Users can record their exercise routines and monitor progress.

Personalized Coaching: Users receive personalized fitness and nutrition recommendations based on their goals and activities.

Health Analytics: Access to health statistics, such as BMI, calorie intake, and activity levels.

Community Engagement: Users can interact with other users, share achievements, and seek support.

2. Coach Functions

Coach Registration: Certified coaches can join the system and offer coaching services to users.

Client Management: Coaches can manage their client list, review their progress, and provide tailored guidance.

In-App Messaging: Coaches can communicate with their clients through the app, sharing documents and exercise routines.

Goal Setting: Coaches can help users set personalized fitness and nutrition goals.

Reports and Progress Tracking: Coaches can access their clients' progress reports and adjust recommendations accordingly.

2.2 User Classes and Characteristics

1. User Class

Characteristics:

Users are individuals looking to improve their health and fitness.

They have varying fitness levels, dietary preferences, and health goals.

Users require an intuitive and user-friendly interface to track their diet and fitness activities.

They may have personal accounts to store their health data and receive personalized recommendations.

2. Coach Class

Characteristics:

Coaches are certified professionals in the health and wellness industry.

They assist users in achieving their fitness and nutrition goals.

Coaches need access to their client profiles, communication tools, and progress reports.

They may have varying levels of experience in using the app.

(SRS) for OLX-like System

1. Introduction

1.1 Purpose

The purpose of this SRS is to define the scope, product functions, and user classes and their characteristics for a system similar to OLX, hereinafter referred to as "the system." This document will serve as a reference for the development team to design and build the system according to the specified requirements.

1.2 Scope

The OLX-like system is a web-based and mobile platform that facilitates online classified advertisements for buying and selling goods. It provides a user-friendly interface for users to create listings, search for products, communicate with sellers, and transact securely. The system aims to connect buyers and sellers in a convenient and efficient manner.

2. Overall Description

2.1 Product Functions

1. User Functions

User Registration: Users can create accounts by providing personal information and contact details.

Product Listing: Users can create and publish advertisements for products they want to sell, including text, images, and pricing.

Search and Browse: Users can search for products based on various criteria, including category, location, and price.

Product Details: Users can view detailed information about products, including images, descriptions, and seller details.

Communication: Users can contact sellers directly through in-app messaging or provided contact information.

User Ratings and Reviews: Users can leave ratings and reviews for sellers and products.

Secure Transactions: Users can negotiate prices, agree on terms, and conduct transactions safely within the platform.

2. Seller Functions

Product Management: Sellers can edit, update, and delete their product listings.

Messaging: Sellers can respond to inquiries from potential buyers and arrange product viewings or transactions.

Transaction History: Sellers can view their transaction history, including sold items and earnings.

2.2 . User Classes and Characteristics

1. Buyer Class

Characteristics:

Buyers are individuals looking to purchase a variety of products.

They have diverse product preferences and may be searching for new or used items.

Buyers require an easy-to-use interface for product search, communication, and secure transactions.

They may have personal accounts to track their favorite listings and message sellers.

2. Seller Class

Characteristics:

Sellers are individuals or businesses offering products for sale.

They need an efficient listing management system to post and manage advertisements.

Sellers require communication tools to interact with potential buyers and complete transactions.

They may have varying levels of experience in online sales.

Library Management System

1. Introduction

1.1 Purpose

The purpose of this SRS is to define the scope, product functions, and user classes and their characteristics for a Library Management System, hereinafter referred to as "the system." This document will serve as a reference for the development team to design and build the system according to the specified requirements.

1.2 Scope

The Library Management System is a software application designed to automate and streamline the operations of a library. It encompasses functions for cataloging books, managing patrons, and facilitating efficient library processes. The system aims to provide a user-friendly and organized platform for librarians and patrons to access and manage library resources.

2. Overall Description

2.1 Product Functions

1. Librarian Functions

Cataloging: Librarians can add, edit, and remove book records in the library database, including title, author, publication details, and availability status.

Patron Management: Librarians can register patrons, issue library cards, and manage patron accounts.

Circulation: Librarians can handle book checkouts, returns, renewals, and reservations.

Overdue and Fines: The system can automatically calculate and collect fines for overdue books.

Inventory Management: Librarians can maintain an accurate inventory of books and track lost or damaged items.

Reports and Analytics: Generate reports on library usage, popular books, and overdue materials.

2. User Functions

Search and Browse: Patrons can search for books by title, author, category, or ISBN and view book availability.

Check Out: Patrons can borrow books, renew checkouts, and place holds on unavailable items.

Account Management: Patrons can manage their personal information, view borrowing history, and pay fines.

Recommendations: Patrons can provide book recommendations and feedback on library resources.

2.2 User Classes and Characteristics

1. Librarian Class

Characteristics:

Librarians are responsible for managing the library's resources and operations.

They have access to all system functions and require robust cataloging and patron management capabilities.

Librarians may vary in experience and technical proficiency.

2. user Class

Characteristics:

Patrons are individuals who use the library's resources.

They have limited system access and require user-friendly search and checkout functionalities.

Patrons may include students, faculty, or general community members.

(SRS) for Online Fitness Application

1. Introduction

1.1 Purpose

The purpose of this SRS is to define the scope, product functions, and user classes and their characteristics for an Online Fitness Application, similar to Dream11, hereinafter referred to as "the system." This document will serve as a reference for the development team to design and build the system according to the specified requirements.

1.2 Scope

The Online Fitness Application is a web and mobile platform designed to provide users with a virtual platform for participating in fitness challenges, tracking their workouts, and engaging in a social fitness community. Users can join challenges, create teams, compete with friends, and monitor their fitness progress. The system aims to inspire and motivate users to lead healthier and more active lives.

2. Overall Description

2.1 Product Functions

1. User Functions

User Registration: Users can create accounts by providing personal information, fitness goals, and preferences.

Challenge Creation: Users can create fitness challenges, set rules, and invite participants.

Team Formation: Users can form teams or join existing teams to participate in challenges.

Workout Tracking: Users can record and log their workouts, including duration, type, and intensity.

Competitions: Users can compete with other teams in fitness challenges, earning points for completed workouts.

Progress Tracking: Users can monitor their fitness progress, view statistics, and set new goals.

Community Interaction: Users can connect with other participants, provide support, and share achievements.

Achievements and Rewards: Users earn badges and rewards for reaching fitness milestones.

2. Challenge Creator Functions

Challenge Management: Creators can set challenge parameters, such as duration, rules, and scoring.

Invitations: Creators can invite participants, manage team formations, and oversee the challenge.

Progress Monitoring: Creators can track and verify participants' workout activities.

2.2. User Classes and Characteristics

1. Participant Class

Characteristics:

Participants are individuals looking to stay active and achieve their fitness goals.

They have varying fitness levels, workout preferences, and interests.

Participants require a user-friendly interface to join challenges, track workouts, and engage with the fitness community.

They may have personal accounts to store their fitness data and interact with other users.

2. Challenge Creator Class

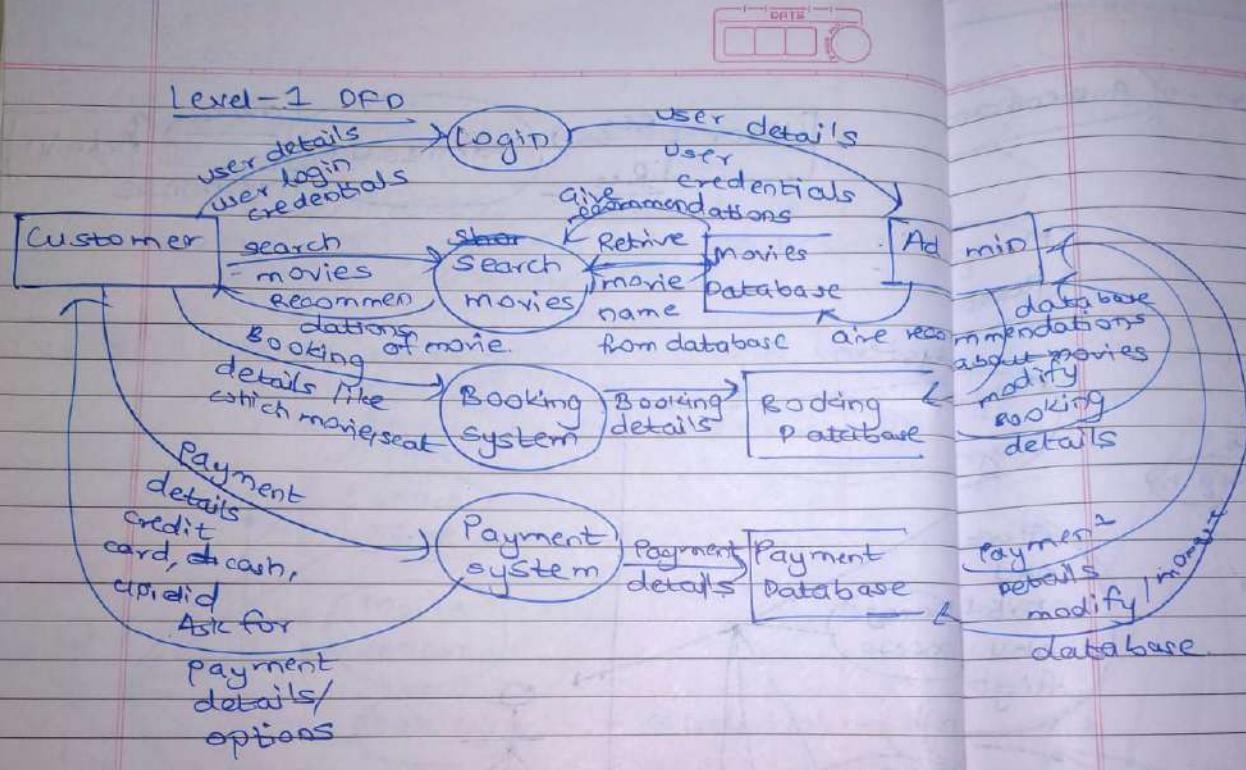
Characteristics:

Challenge creators are fitness enthusiasts who want to design and manage fitness challenges.

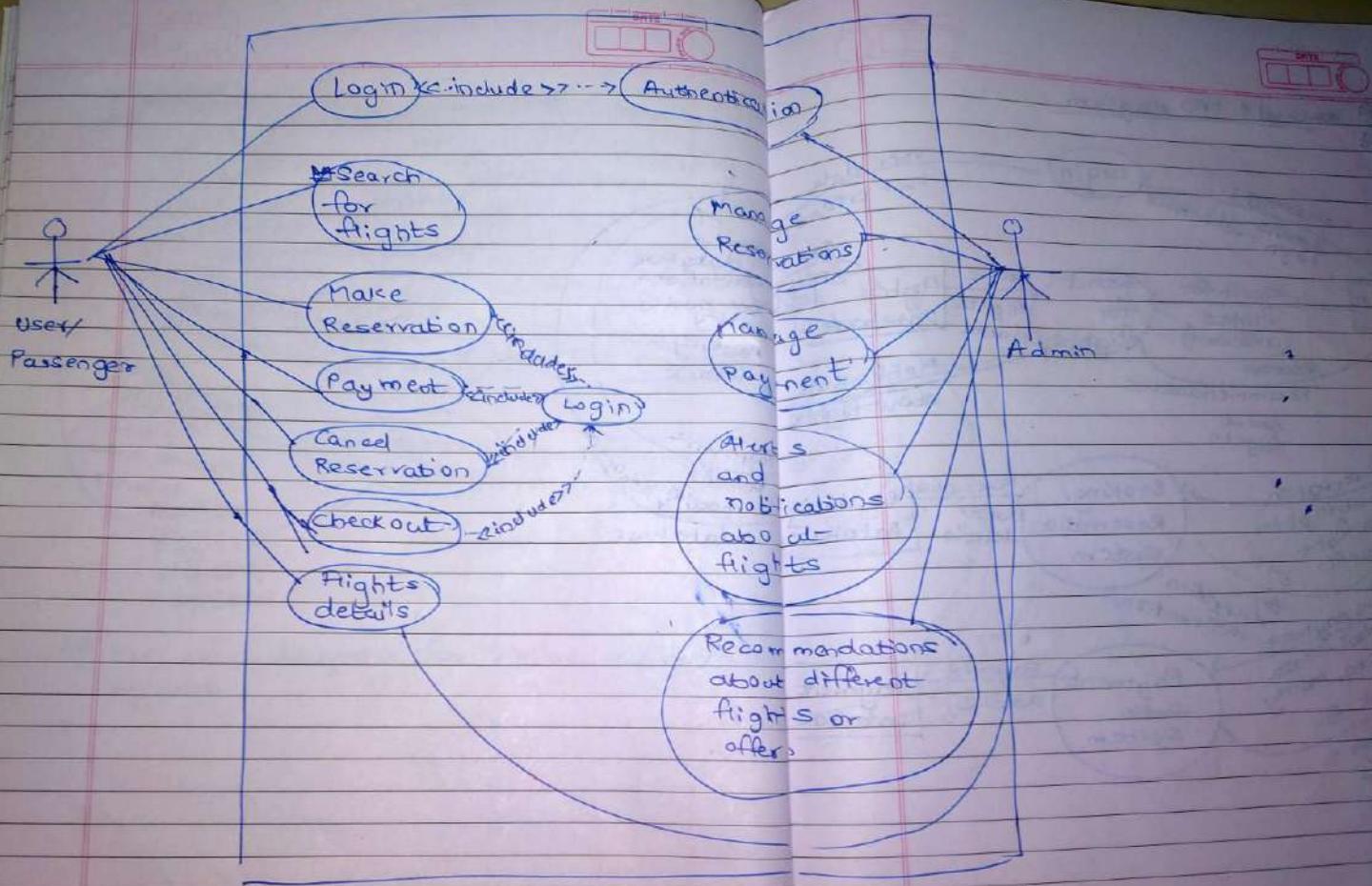
They require advanced functionality for challenge creation, monitoring, and team management.

Challenge creators may have expertise in fitness and event planning.

Level-1 DFD

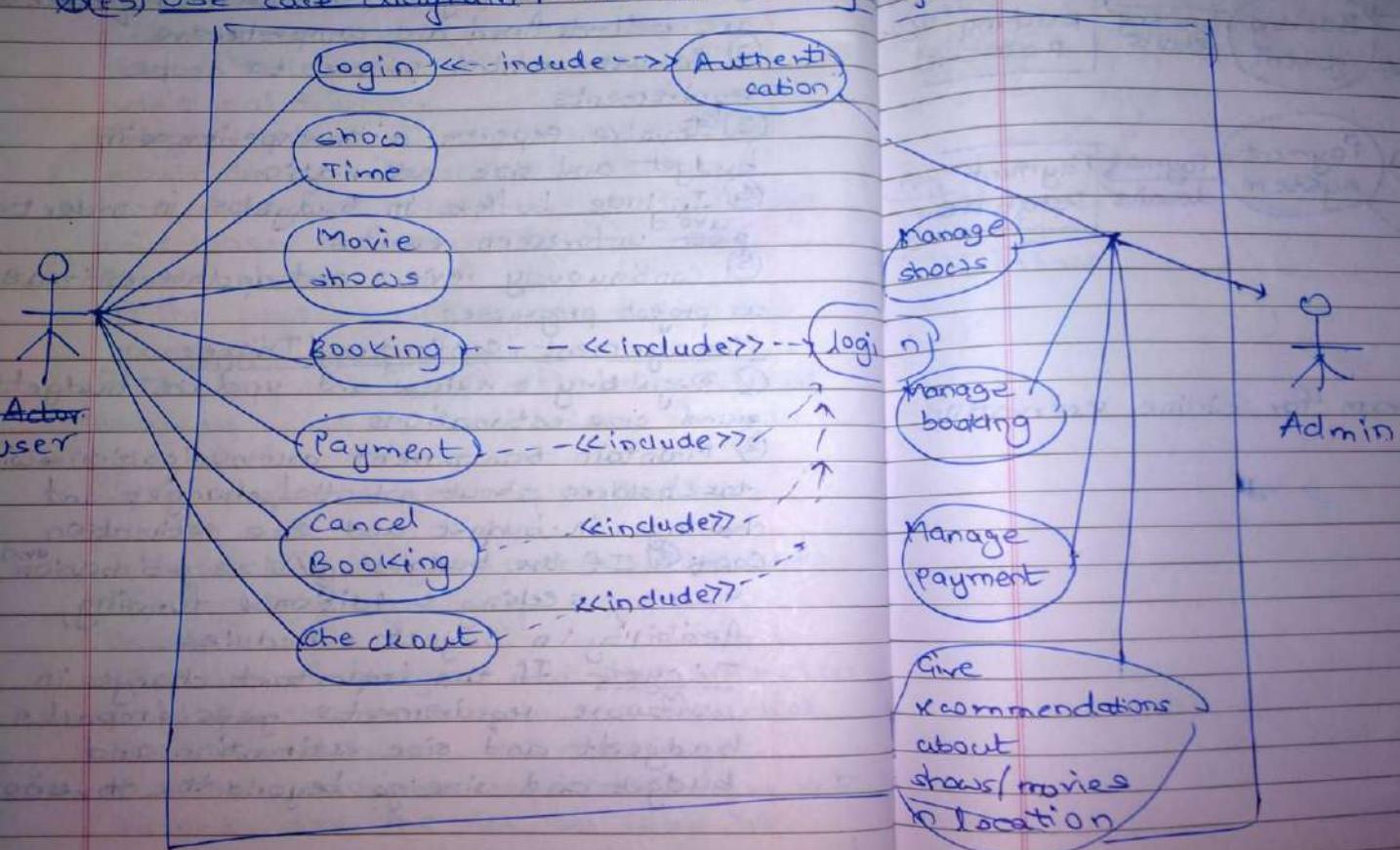


(7) Use case diagram for airline reservation system



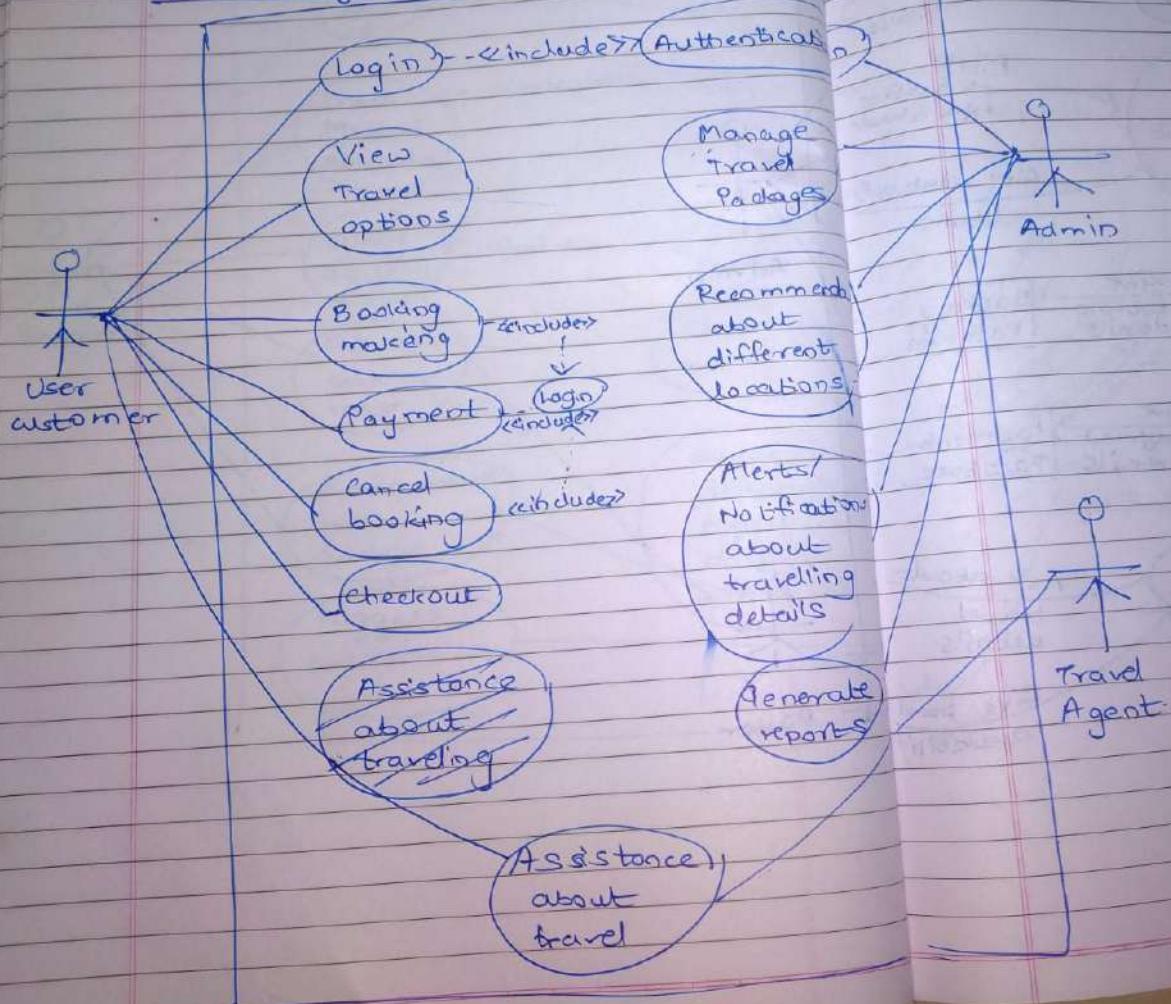
to minimize project estimation

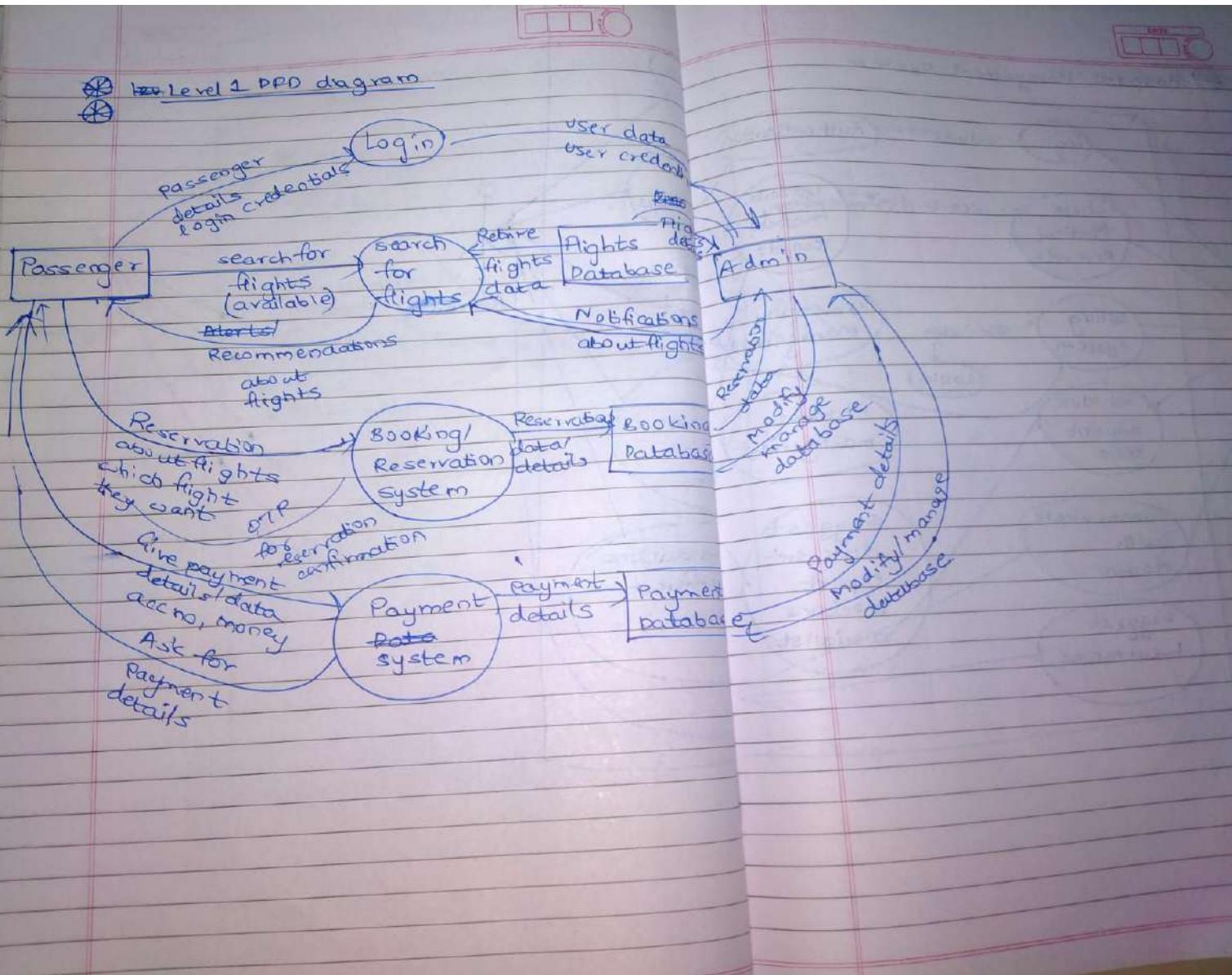
Q5) Use case diagram Movie ticket booking



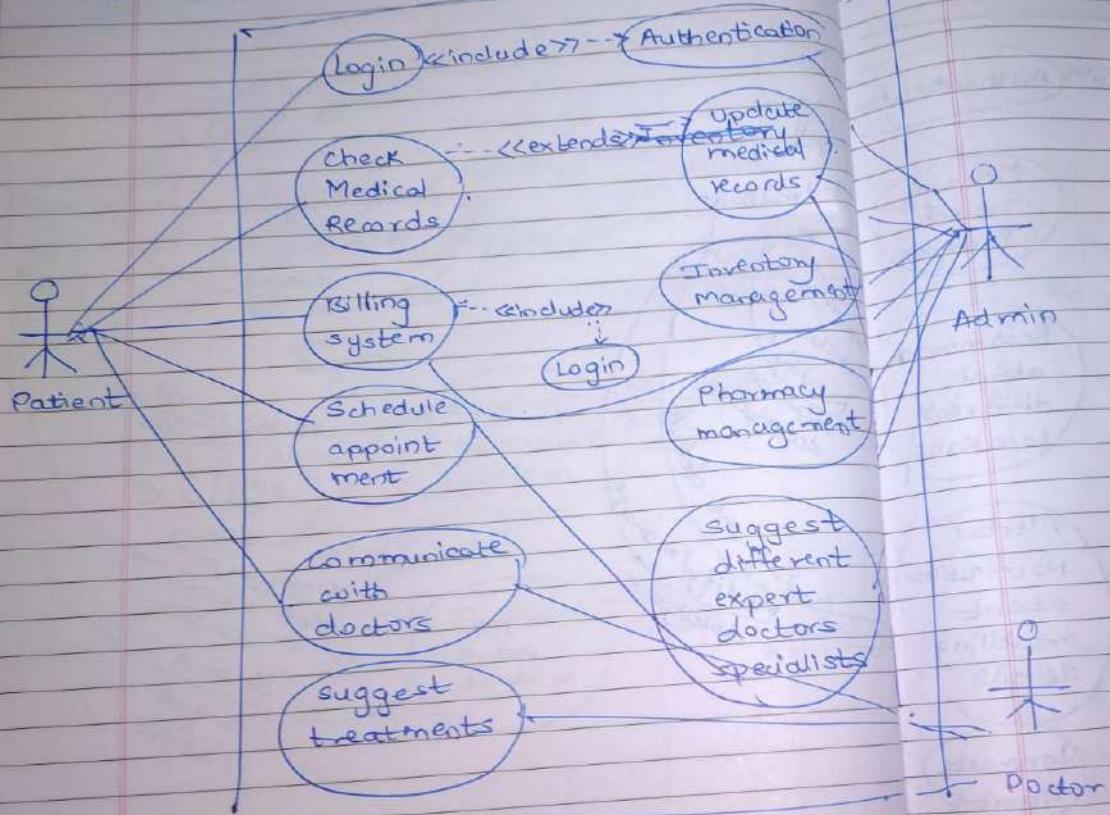
Tours and travels booking system

Use case diagram

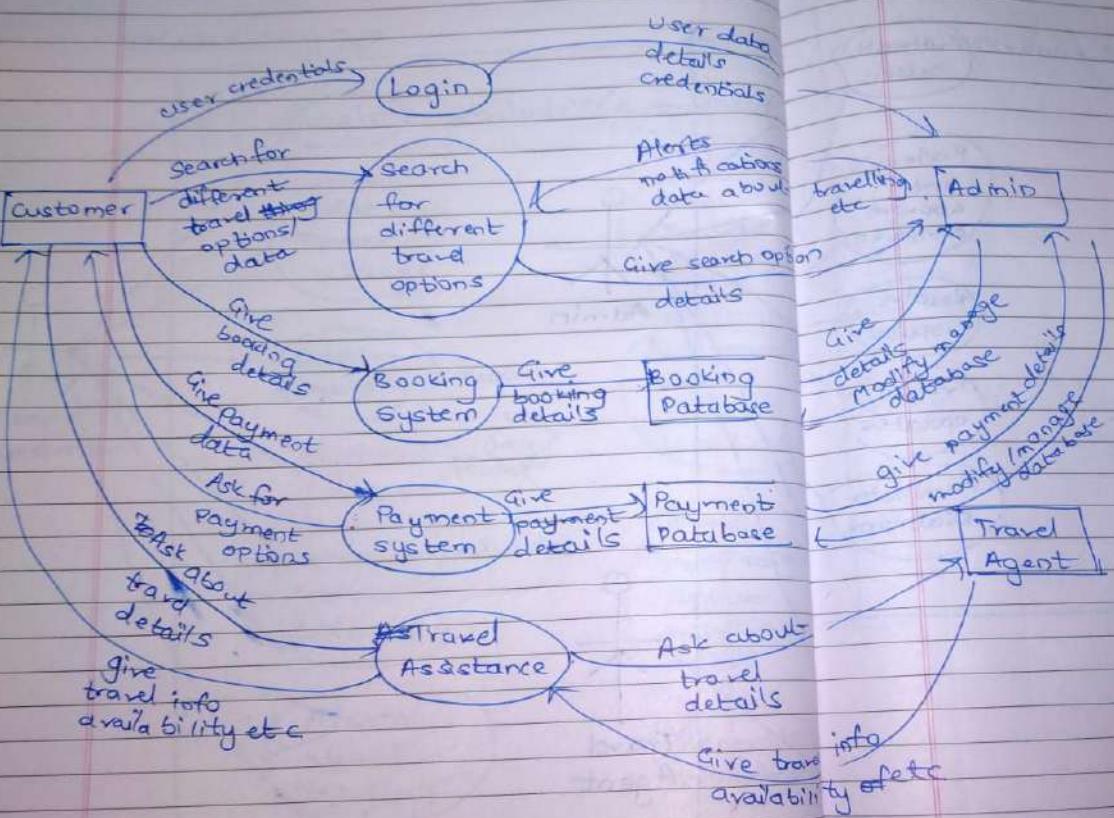




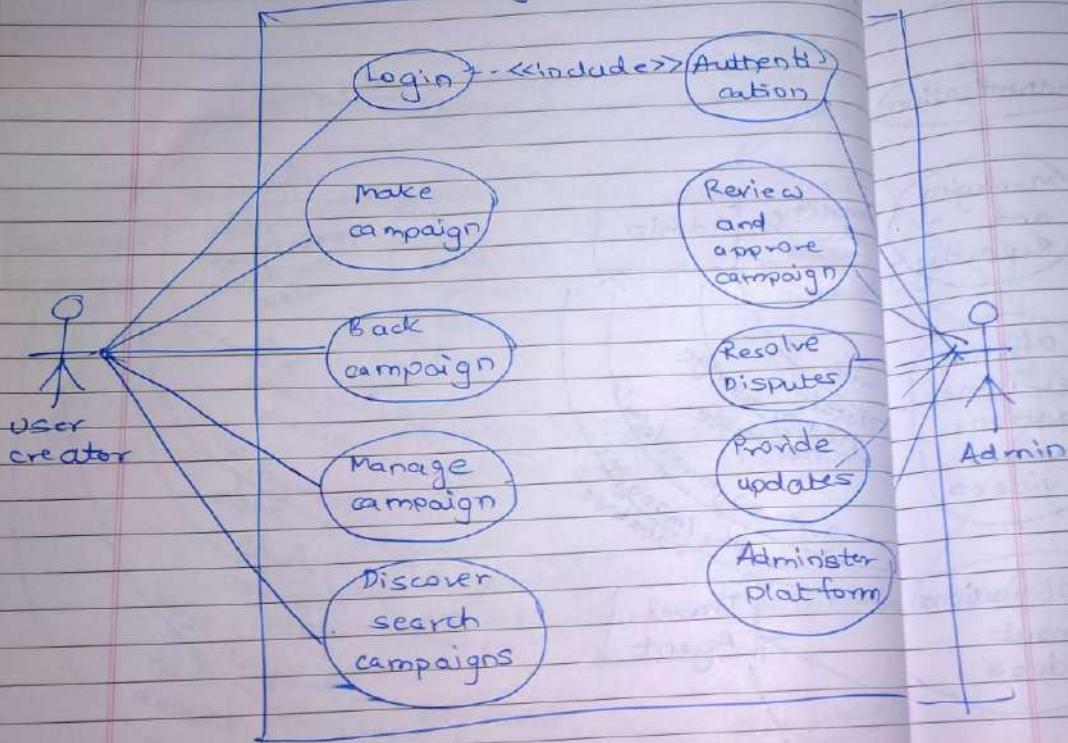
Hospital Management System



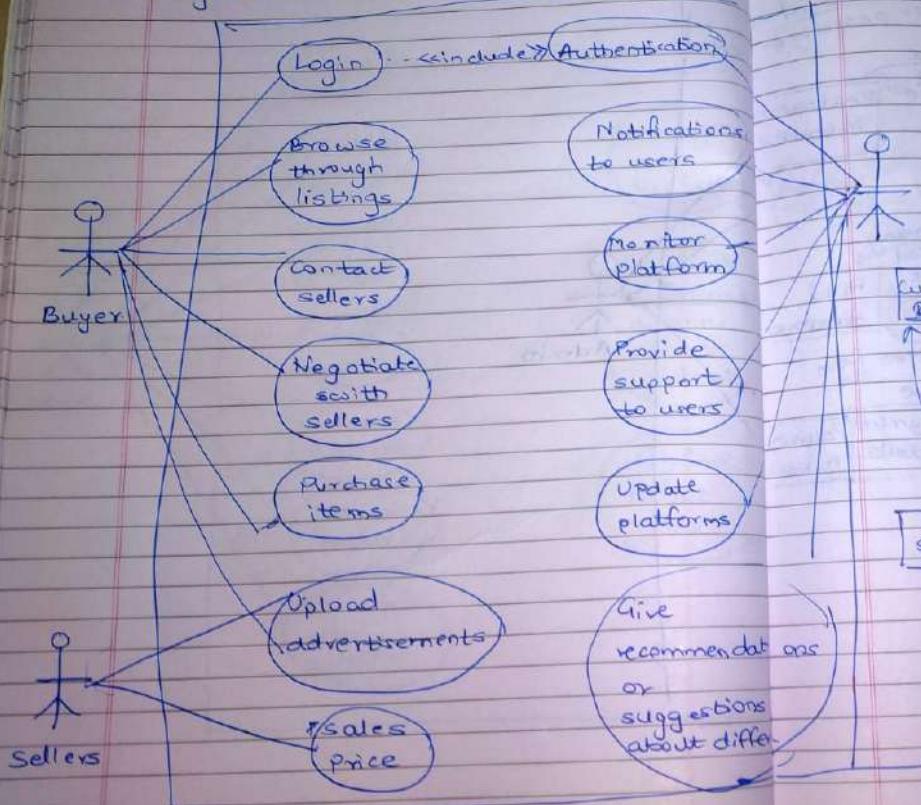
Level 4 DDFD



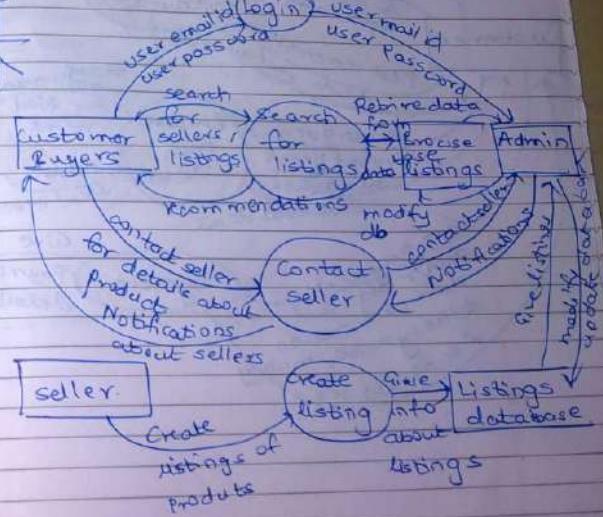
Online crowdfunding platform



Q) Draw a level 1 DFD and use-case diagram for a system like OLA



 World Levels DFD



use case diagram for tiktok

