SOFTWARE ENGINEERING

Def
Diagram
Example
Kiska topic h uska naam
How, Where, who
Insight of the question to make answer

Data dictionary

1. Data dictionary is a collection of metadata about the data in a database such as its meaning, type, and relationships.
2. Data dictionary ek documentation hota hai
3. ek database ya dataset mein data elements ke definitions, structure, aur properties ko store karte h aur represent karte h
4. Ex. oxford dictionary, Ek document ya tool hota h

Contents

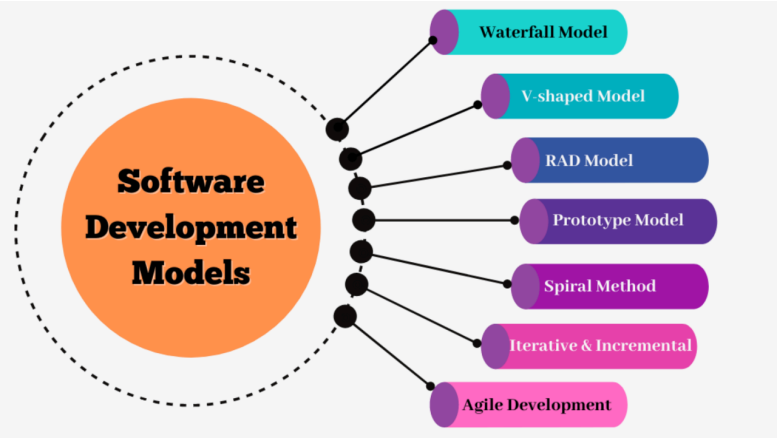
1. Data dictionary mein data element ke naam, data type, constraints, description aur data source jaisi information hoti hai
2. Data dictionary mein ek table hoti hai, jismein har row ek data element ko represent karta hai Aur Har column ek data element ke property ko represent karta hai.
3. Data dictionary ka use data ko organize karne, maintain karne aur understand karne ke liye hota hai.

Ex. of table

Name	Email	ID	Gender
Arthur	arthur@gmail.com	A1	M
Marie	marie@gmail.com	A2	F
Bob	bob@gmail.com	A2	M
Peter	peter@gmail.com	A3	M

Software development model

1. Software kese banta h, Sdm ko design, create, or test kiya jata h
2. Sdm ko step by step process se banaya jata h Easy to understand k liiye
3. Sdm mein user se Feedback liya jata h
4. Sdm banate wakt Low Risk, low cost Time saving, improve project quality, flexibility ka dhyaan rakkha jata h
5. Ex. jese recipe me steps hote h wese hi stepe hote h software ko banane k liye, windows 12 ka Software development model

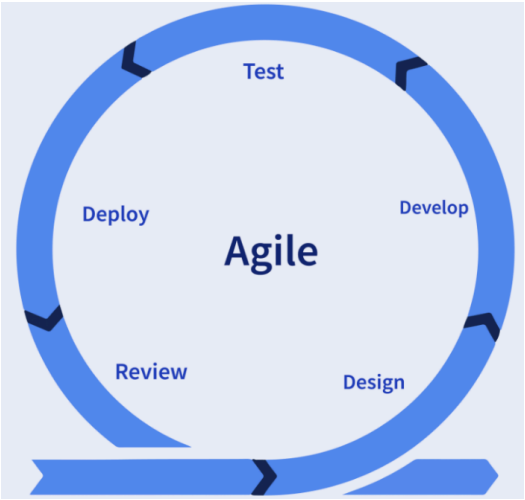


Which software development model is suitable for developing 'Online Examination System' ? Justify your selection. Also explain the selected model.

Iterative enhancement model, Agile Development

Agile Development model

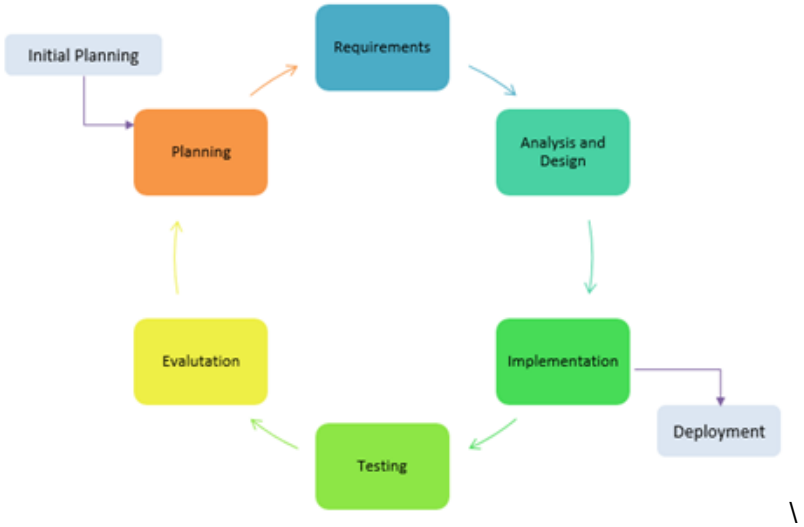
1. Software engineering me software development ka model h
2. software ko chhote-chhote hisson mein time par banaya jata hai
3. Step by step, Feedback, Easy to understand, Low Risk, low cost
   Time saving, improve project quality, flexibility
4. Ex. choti team jaldi se software features banati hai aur feedback le kar unhe behtar banati h



Iterative enhancement model

1. Software engineering me software development ka model h
2. software ko initial version se start karke chote-chote updates mein develop kiya jata h
3. Step by step, Feedback, Easy to understand, Low Risk, low cost
   Time saving, improve project quality, flexibility

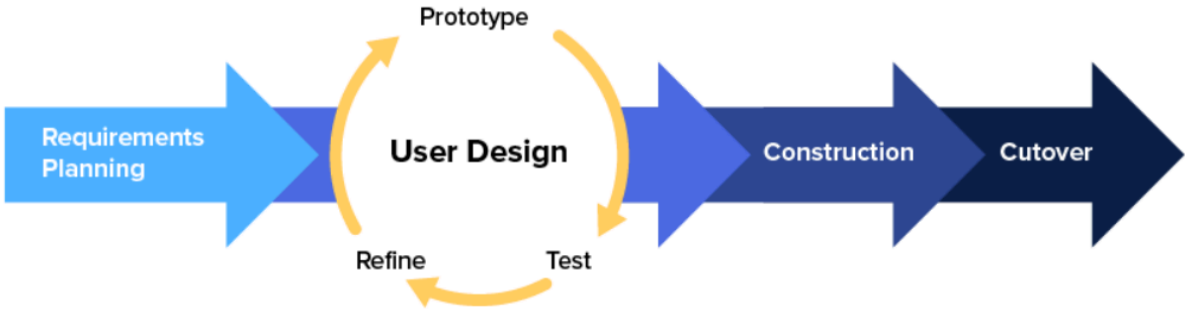
4. Ex. secaurity patch dekar secaurity aur features daalte h



\

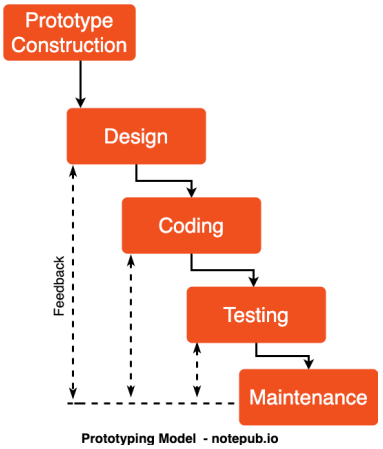
Rapid Application Development

1. RAD Rapid Application Development
2. Software engineering me software development ka model h
3. software ko jaldi banane aur improve kiya jata hai
   Taaki developement ko fast kiya jata h aur users se feedback jaldi liya jata h
4. Rad software ko jaldi aur efficiently banane ke liye prototyping aur iterative design ka use karta h
5. Step by step, Feedback, Easy to understand, Low Risk, low cost
   Time saving, improve project quality, flexibility
6. Ex. mobile banking apps ka development, jisme prototyping aur user feedback ka use karke jaldi se features implement kiye jaate hain.



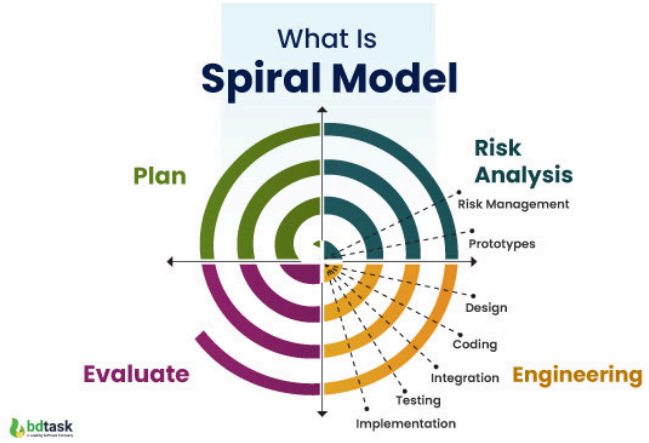
Prototype model

1. Software engineering me software development ka model h
2. software ka ek basic ya rough version jaldi se banaya jaata hai taaki feedback le kar usse behtar banaya ja sake
3. Step by step, Jaldi Visualization, Easy to understand, Low Risk, low cost
   Time saving, improve project quality, flexibility
   customer feedback ka importance hota h
4. Ex. rough copy



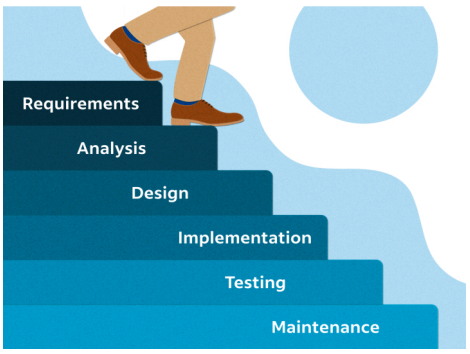
Spiral model

1. Software engineering me software development ka model h
2. Software ka development process multiple iterations (spirals) mein hota hai
3. Software ko baar baar chakkar lagakar develope kiya jata h
4. customer feedback, improve project quality, Jaldi Visualization, Easy to understand, Low Risk, low cost, Time saving, flexibility, step by step
5. Ex. Jese roj ek hi website par baar baar naye features daalte rhena time to time



Waterfall model

1. Software engineering me software development ka model h
2. Waterfall model mein software ka development process ek linear sequence mein hota h
3. Ek phase complete hone ke baad, next phase shuru hota hai aur peechhe lautna mushkil hota h Aur Har phase mein detail information hoti h
4. Step by step, Flexibility, Easy to understand, Low Risk, low cost
   Time saving, improve project quality, flexibility, customer feedback,
5. Ex. software ko starting se end tak banana ek linear sequence mein

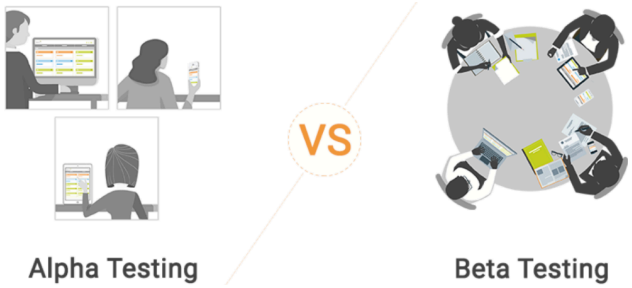


Alpha testing

1. alpha testing ek software development ka process hai
2. At me Software ko bananae k baad phele test hota h
3. At mein software ka phela version internal team ya limited users test karte h
4. internal team ya limited users se feedback lekar software bugs aur problems ko improve kiya jata h taaki beta version better ho
5. Ex. windows 12 ki alpha testing

Beta testing

1. Beta testing ek software development ka process hai
2. Beta testing mein software ka beta version limited public test karte hain
3. Software me selected Public se feedback lekar software k bugs aur problems ko fix kiya jata h, taaki stable version better ho.
4. Beta testing types - open beta (public ke liye), closed beta (limited users ke liye), aur private beta (specific users ke liye)
5. Ex. windows 11 k latest features use karne k liye beta version join karna



Explain unit testing and module testing with the help of suitable example for each.

Unit testing

1. Unit testing ek software testing technique hai
2. Ut me software k individual components ya functions ko alag-alag test kiya jata hai
3. Ut me software k individual components ki functionality sahi hai ya nahi, yeh check kiya jata h
4. Unit testing software development mein use hota hai

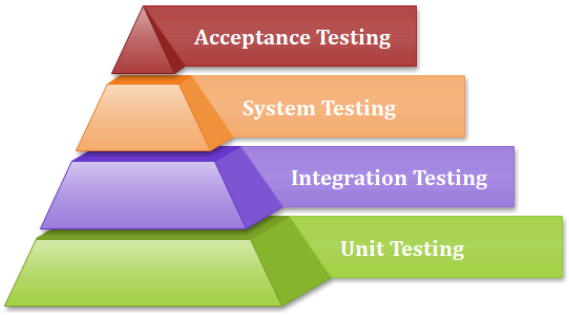
Module testing

1. Module testing ek software testing technique hai
2. Module testing mein software ke chhote-chhote hisson (modules) ko alag alag test kiya jata hai

3. Mt me softaware k modules ko dekha ja sake ki wo sahi kaam kar rahe hain ya nahi
4. Module testing software dev. Mein use hota hai
5. Ex. Ek team e-commerce app ke payment module ko test karti hai taaki transactions processing ko test kiya ja sake

Integration Testing

1. software testing technique hai
2. It me software k alag alag modules ko ek saath integrate karke test kiya jata hai
3. software ke alag-alag parts ko ek saath milaakar test karna, taaki yeh dekha ja sake ki woh sahi tarah se milkar kaam karte hain.
4. Ex. Online shopping app mein payment gateway ko cart system ke saath test karna, taaki dono modules sahi tarah se milkar kaam karein.



Coupling

1. modules ya components ek dusre se kitna tightly connected hain
2. High and low coupling

   high coupling: ka matlab hai zyada dependency

   low coupling: ka matlab hai kam dependency.
3. Tight and loose coupling

   Tight coupling: ka matlab hai do parts ek dusre se bohot zyada connected h agar ek part mein kuch change hota h to doosra part jyaada affect hota hai Aur update karna muskil hota h

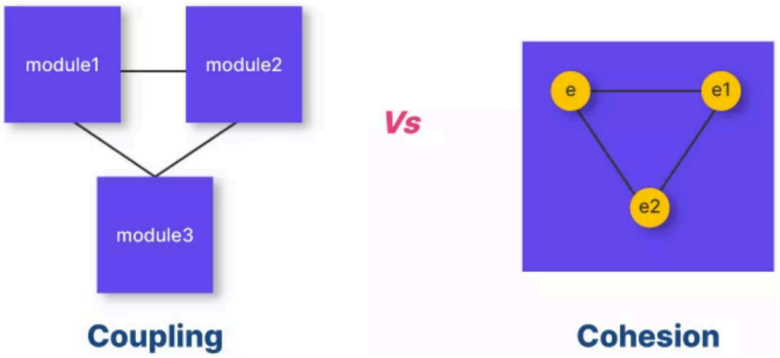
   Loose coupling: agar ek part mein kuch change hota h to doosra part kam affect hota hai Aur update karna asan hota h
4. Ex. remote TV se judta hai aur usko control karta hai, lekin dono alag cheezein hain

Cohesion

1. Cohesion me module ya component mein functions ya tasks kitni closely related h
2. ek module mein sabhi functions ek specific task ko complete karne ke liye grouped hote hain, jaise ki ek calculation perform karna
3. Cohesion me functions ek sequence mein execute hote hain, ek ke baad ek. Aur inka sequence ya order important nahi hota hai
4. Ex. ek pen, jisme nib, ink aur cap sab mil kar likhne ka kaam karte hain aur bina kisi ek part ke woh sahi se kaam nahi karega

Coupling aur cohesion mein farq

coupling ek module ko doosre modules ke saath kitna tightly connect kiya gaya hai
cohesion ek module ke andar ke elements kitna closely related hone ko dikhata hai

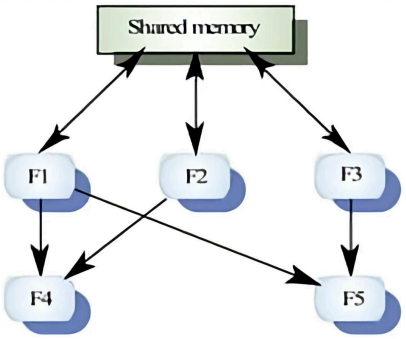


Ch Cohesion Types

1. **Functional Cohesion**: Jab ek module sirf ek kaam karta hai. Jaise ek function h jo sirf addition karta hai
2. **Sequential Cohesion**: Jab ek module ke tasks ek dusre ke baad aate hain Jaise data ko read karna aur phir usko process karna.
3. **Communicational Cohesion:** Jab ek module ke tasks same data ya resource par kaam karte hain. Jaise ek function jo user ke information ko read, update aur delete karta hai.
4. **Procedural Cohesion:** Jab ek module ke tasks ek specific order mein hote hain.
5. **Temporal Cohesion:** Jab ek module ke tasks ek specific waqt par ek sath execute hote hain

Function oriented design

1. (FOD) mein system ko alag-alag functions mein divide kiya jata hai jismein har function ek specific task perform karta hai.
2. Focus: FOD mein Functions par focus hota h
3. Hierarchical Breakdown: FOD me major functions ko sub-functions me divide karte h.
4. Reuse: FOD mein Functions ko reuse kiya ja sakta hai
5. Ex. ATM machine, jo alag functions ke through kaam karti hai jaise paisa nikalna
6. balance check karna, aur paisa jama karna har function apna ek specific kam karta h



Concepts:

1. **Function Decomposition**: System ko chote parts mein divide karte h Jisse har function specific task ko perform karta h
2. **Top-down Design**
3. **Modularity**
4. **Data Flow Diagrams (DFD)**
5. **Structured Analysis and Design**:
6. **Function Specification**:

7. **Abstraction or Information Hiding:** ek Function ke internal details dusre functions se chhupaye jaate hain, taki ek function ke changes dusre functions ko affect na kare
8. **Encapsulation**: Module ke andar ke details ko secure aur access par limit karte h

Advantages:

1. Divide aur Reuse, Easy to understand
2. Easy to Test: Functions ko alag se test karna easy hai.
3. Fast: Performance mein efficient hota hai.

Disadvantages:

1. Difficult in Large Systems: Bade systems mein handle karna difficult hota h
2. Data Security Kam
3. Difficult to Change
4. Real-World Representation: Real-world objects ko represent karna tough hai.

Static object

1. Static object ek class ka instance hota hai
   jo class ke sabhi objects ke liye shared hota hai
   Ex. ek school ka clock ho sakta hai, jo sabhi classrooms mein ek hi time dikhata hai aur uska state sabhi jagah par same rehta hai.
2. **Static Variables**:
   ek baar initialize kiya jata h aur jab tak program chalta h tab tak memory mein rehte hain
   Een variables ko `static` keyword se declare kiya jaata h
3. **Static Methods**:
   class ka instance banaye bina directly call kiya jata hai
   een functions ko `static` keyword se declare kiya jaata h
4. **Static Classes**: class aur class k sare members ka instantiate nahi hota h kyuki static hote h
   classes ko `static keyword` se declare kiya jata h

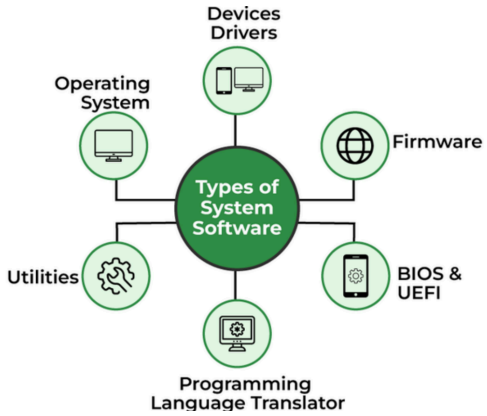
ch Static object specs

1. **Lifetime**: Static objects program start hote hi create hota h aur jab tak program chalta hai tab tak exist karta h
2. **Scope**: file ya class ke andar define kiya jata hai, fir wahi se access kiya jata hai.
3. **Initialization**: Static objects program shuru hone se pehle ek baar initialize hota h aur unhe default value milta hai
4. **Usage** program band hone ya restart hone ke baad bhi data waisa hi rahta h.

SOFTWARE SYSTEM

1. Software system ek group of interconnected softwares ya program hota h
   Jo milkar kaam karte h aur design kiya jata h
2. ss program k soft aur hard ke beech mein communication ko aasan banata hai.

3. System k andar preinstall software hote h jese calculator, radio. Camera aur application



*SEI-CMM

1. Software Engineering Institute Capability Maturity Model
2. SEI-CMM organizations ke software development processes ki quality ko behtar banane me help karta hai.
3. Development process ko maturity levels ke hisaab se classify kiya jata h taaki quality aur productivity badh sake
4. Ex. ek software company jo apne kaam ko behtar banane ke liye apne processes ko step-by-step improve karti hai

Initial (Level 1):
Organization k software development ki bnana shuru karti h bina planning aur rules k

Repeatable (Level 2):
Organizations k software development processes k phele k task ko use karti h

Defined (Level 3):
Organizations k software development process detail me declare aur understand kiya jata h

Managed (Level 4):
organizations ke software development processes k kaam ko manage aur problem solve kiya jata h

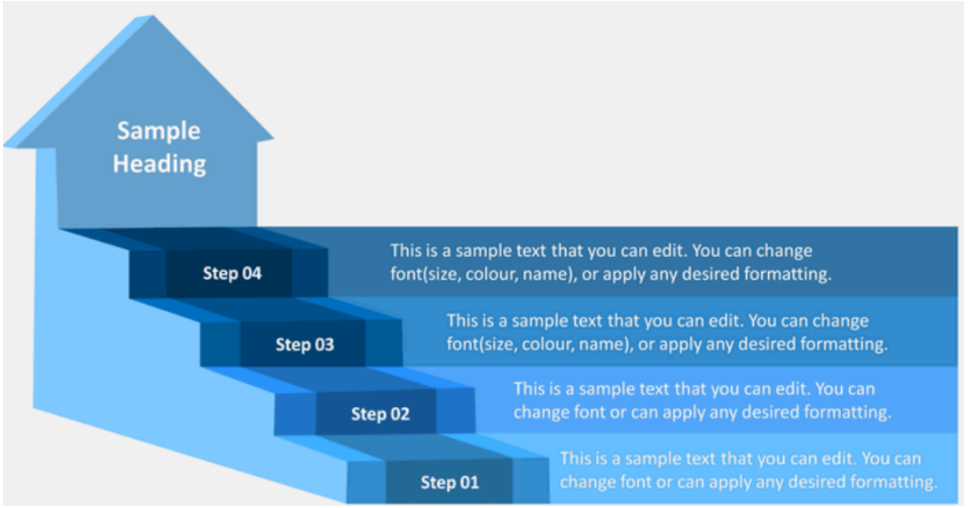
Optimizing (Level 5):
organizations ke software development processes ko better banane par focus hota h new idea aur improvments se

*Step-Wise

1. Step-Wise ek framework h jo badi mushkil problems ko chote-chote steps mein divide karta h aur solve karta h
2. Sw har step ko simple aur systematic banaya jata h
3. Sw har step ko oragnise, manage, sequence me kiya jata h
4. Step wise framework project management aur problem-solving me use hota h
5. Ex. ek recipe follow karna, jismein har step ko ek ke baad ek sahi tareeke se karte hue final dish banai jati hai

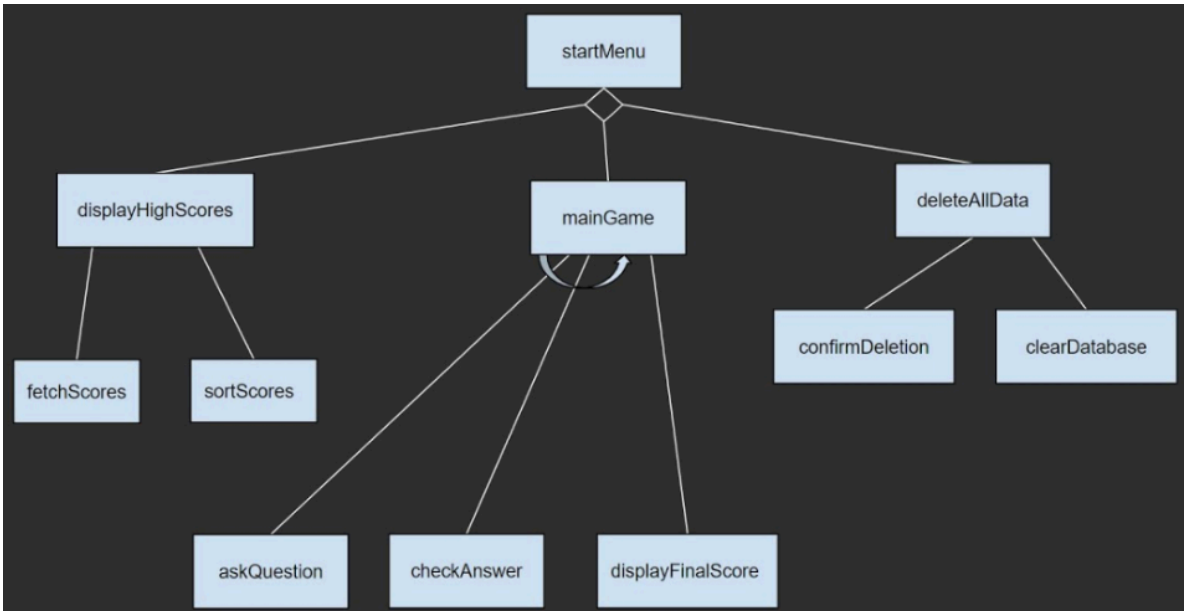
Yeh stages hote hain:

1. **Objectives Set Karo:** big problem ko step me karne k baad har step ka objective set hota h
2. **Requirements Samjho:** big problem k step ko samjha jata h jese kya requirements chahiye aur kya problems hain
3. **Solution Plan Karo**: big problem k step ka detailed plan banaya jata h
4. **Solution Implement Karo**: big problem k steps ko follow karke problem solve kiya jata h
5. **Test Aur Check Karo**: big problem k har step ko test aur check kiya jata h aur jaroorat k hisab se change kar sakte h
6. **Deploy Aur Maintain Karo:** big problem k har step ko check regular dekha jata h aur use karne k saath maintain kiya jata h



***Structure chart**

1. Sc ek visual representation hai
2. represent: Sc me system ya modules ko hierarchical structure me Represent kiya jata h
3. Hierarchical: Sc Modules ko hierarchical design me organize, manage kiya jata h h
4. Modularity: Sc me Sabse upar main module hota hai, aur uske niche sub-modules hote hain.
5. Ex. online shopping website ka structure chart dikhata hai ki user interface, product catalog, shopping cart, payment processing, aur order management jaise parts kaise ek dusre se judi hain, Modular Design dikhata h
   Example: https://www.geeksforgeeks.org/software-engineering-structure-charts/

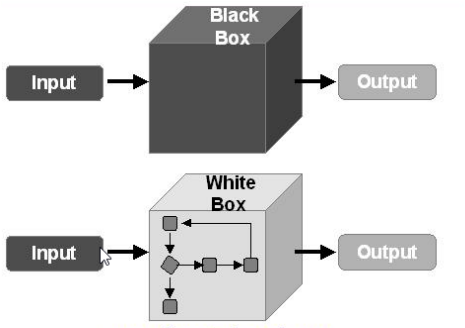


Black box testing

1. Black box testing me tester software ke sirf external functionalities ya structure ko test karta h
2. Bbt me Tester software ke inputs aur outputs ko check karta hai, bina internal workings or coding ko dekhe
3. Bbt me tester software k external bug aur error ko dhundta h aur fix karta h
4. Ex. ATM machine ko test karna, jahan user sirf input aur output check karta hai bina ye dekhe ke andar kaise kaam ho raha h

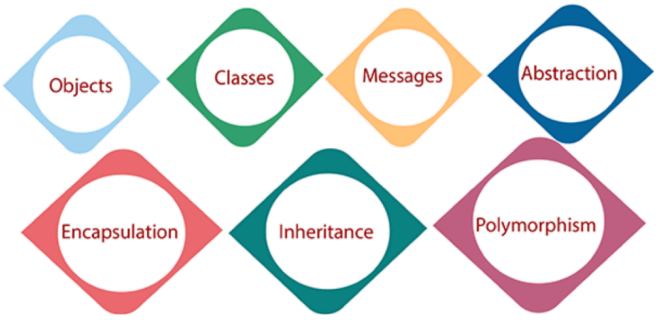
White box testing

1. White box testing mein tester software ke internal workings aur coding ko check karta h aur bugs aur errors ko dhundta hai.
2. Wbt me Tester software k internal chije jese code, logic, aur paths ko check karta h bina external working ko dheke
3. wbt me tester software k internal bug aur error ko dhundta h aur fix karta h
4. Ex. jab ek developer code ke andar jaa kar har line aur logic ko check karta hai, taake koi galti na ho



* Object-oriented design

1. Object-oriented design (OOD) ek software development ka process hai
2. Divide objects
   Ood me software ko chhote-chhote objects mein divide karke banaya jaata hai.
3. combination
   Ood me objects, data aur functions ka combination hote hain.
4. Useful function
   Ood me utility objects (ya helper classes) wo classes hoti hain jo useful functions provide karti hain
5. Ex. Car manufacturing mein "Car" ek class hai aur "Engine," "Wheel," aur "Door" uske objects hain, jo inheritance aur encapsulation ka use karke ek complete car banate hain



Criteria for Identifying Utility Objects

Utility objects or utility class

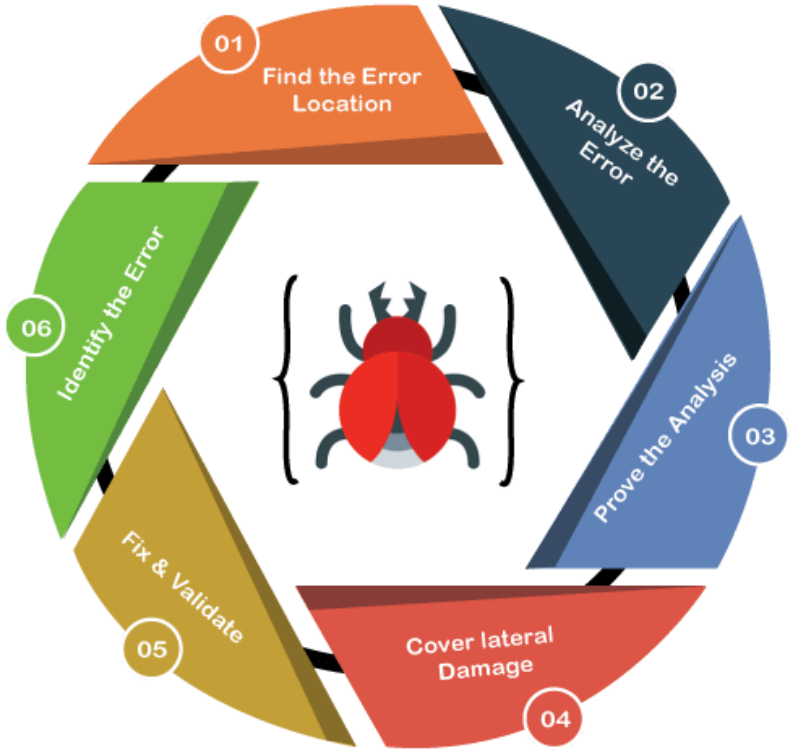
1. object-oriented design mein utility objects useful functions provide karte h
2. Uo me functions ka reuse aur simplify kiya jata h
3. Uo me code ko clean aur maintainable banaya jata h
4. Uo medata store nahi karte
5. Uo input par kaam karke result dete hain
6. CLINRS
    1. **Common Functionality**
       Utility objects Har jagah use hone wale common functions provide karte hain.
    2. **Static Methods**
       Utility objects Seedha class se access kiya jata hain, bina object banaye.
    3. **Reusability**
       Utility objects Baar-baar alag modules mein reuse hote h
    4. **No Side Effects**
       Utility objects Diye gaye data par bina change kiye kaam karta h

5. **Low Coupling**
   Utility objects mein Code dusri classes par kam depend karta hai.
6. **Immutability**
   Utility objects ko badla nahi ja sakta, jisse data safe rehta ha

EXPLAIN VARIOUS DEBUGGING STRATEGIES

1. Ds me code k errors ko identify, analyze, aur fix karne ke liye step by step methods aur tools ka use karte h
2. Ds code me galtiyan dhoondhne aur unhe theek karne k liye kaam aata h

Debugging Tools: IDEs ke debugging tools use karo jese vs code, pycharm,etc
Unit Tests: code k chote parts ko likho, test aur run karo taaki errors pakad sakein.
Code Reviews: Doston se code review karwao taaki unse errors mile
Rubber Duck Debugging: Apne code ko kisi ko explain karo
Documentation and Comments: code me comments add karo taaki samajhna asan ho.
Print Statements/Logging: Code mein print ya log statements daal ke check karo.
Breakpoints: Code ko pause karke line-by-line check karo
Divide and Conquer: Code ko chhote parts mein tod ke test karo.
Automated Testing: Tests likho jo errors automatically pakde
Static Code Analysis: Code ko tool se check karo taaki common errors mil sake
Binary Search: Code ke parts ko disable karke bug dhundo.
Pair Programming: Ek aur developer ke saath milke bug fix karo.
Refactoring: Complex code ko simplify karke bug samajho



Develop SRS for Online Study Center Allocation System (OSCAS) for students who apply for admission to a university. SRS should be in IEEE format. Make necessary assumptions. 25

Online study center allocation system ek system hai jo students ko unki pasand aur eligibility ke hisaab se study center deta hai.

SRS

Pdf: https://dspmuranchi.ac.in/pdf/Blog/srs_template-ieee.pdf

Example :

Online Admission System (OAS)

1. Introduction and Purpose
    - Purpose: This system helps students apply for admission to schools or colleges online. It makes the application process easier and faster.
        - Example: Students can fill out forms and upload documents from home instead of visiting the school.
2. Scope of the System
    - Scope: The system will allow:
        - Registration for users.
        - Filling out and submitting application forms.
        - Uploading required documents (like transcripts).
        - Paying application fees online.
        - Tracking application status.
        - Example: A student can apply for a program, upload their report card, and pay the fee all in one go.
3. Functional Requirements
    - User Registration: Students and administrators can create accounts and log in.
        - Example: A student creates an account using their email and password.
    - Application Form: Students can fill out their details and select the program they want to apply for.
        - Example: A student selects "Bachelor of Arts" and enters their grades.
    - Document Upload: Students can upload necessary documents as files.
        - Example: A student uploads a PDF of their high school diploma.
    - Application Tracking: Students can check the status of their application.
        - Example: A student sees "Application Submitted" on their dashboard.
    - Payment Processing: Students can pay their application fees online.
        - Example: A student pays a $30 fee using a credit card.
4. Non-Functional Requirements
    - Performance: The system should work smoothly even if many students are using it at the same time.
        - Example: The website should load in under 3 seconds.
    - Security: Protect student data with encryption to keep it safe.
        - Example: All information sent through the system should be encrypted.
    - Usability: The system should be easy to navigate and use.
        - Example: Clear instructions and help options should be available.

5. User Roles and Permissions
    - Roles:
        - Student: Can register, apply, upload documents, and track applications.
        - Administrator: Can manage applications and send notifications.
        - Example: Only admins can change an application status to “Accepted.”
6. System Architecture
    - Overview: The system has three main parts:
        - A website for users (front-end).
        - A server to handle requests (back-end).
        - A database to store all information.
        - Example: The website might use React for design and Node.js for the server.
7. Data Requirements
    - Data Types: The system will store user information, application details, and documents.
        - Example: User data includes name, email, and password.
8. Integration Requirements
    - Integration Needs: The system should work with payment processors for online payments.
        - Example: When a student pays the application fee, the system updates their application status right away.
9. User Interface Requirements
    - Design: The interface should be user-friendly and work on phones and tablets as well as computers.
        - Example: A mobile-friendly layout allows students to apply from their phones.
10. Testing and Validation
    - Testing Strategies:
        - Unit Testing: Check each part of the system works correctly.
        - Integration Testing: Ensure that payment processing and application submissions work together smoothly.
        - User Acceptance Testing (UAT): Have real users test the system to make sure it’s easy to use.
        - Example: A group of students tests the application process before it goes live

Online banking system

Introduction and Purpose:

- What it is: The online banking system lets customers access their bank accounts online.
- Example: Customers can check their account balance and make payments anytime, anywhere using the internet.

Functional Requirements:

- User Registration and Login: Users can create accounts and log in securely.
    - Example: A user signs up with their email and password.
- Account Management: Users can see their account details.
    - Example: A user views their savings account balance.
- Money Transfers: Users can send money between accounts.
    - Example: A user transfers $50 to a friend’s account.
- Bill Payments: Users can pay their bills online.
    - Example: A user sets up automatic payments for their internet bill.
- Transaction History: Users can see their past transactions.

    - Example: A user looks at their transaction history for the last month.

Non-Functional Requirements:

- Performance: The system should work quickly, even with many users.
- Security: Users' personal information must be safe and encrypted.
- Usability: The site should be easy to use for everyone.
- Availability: The system should be online 99.9% of the time.

User Interface Requirements:

- Example: The site should have clear buttons and easy navigation.
    - Users should easily find options like "Transfer Money" and "View Statements."

System Architecture:

- Example: The system will have three parts:
    - User Interface: What users see and interact with online.
    - Application Logic: The rules that make things work (like transferring money).
    - Database: Where all user and transaction information is stored.

Data Management:

- Example: Information is stored in tables like:
    - Users: Contains names, emails, and passwords.
    - Accounts: Contains account types and balances.
    - Transactions: Contains details about money transfers.

User Roles and Permissions:

- Example:
    - Customer: Can view and manage their accounts.
    - Bank Staff: Can help customers and view account details.
    - Admin: Has full control over the system.

Error Handling and Logging:

- Example: If something goes wrong, the user sees a message like:
    - "Invalid login. Please try again."
- The system will keep a record of errors for troubleshooting.

Testing Requirements:

- Example: The system will be tested to ensure everything works:
    - Unit Testing: Check individual features like logging in.
    - User Testing: Real customers will try the system and give feedback.

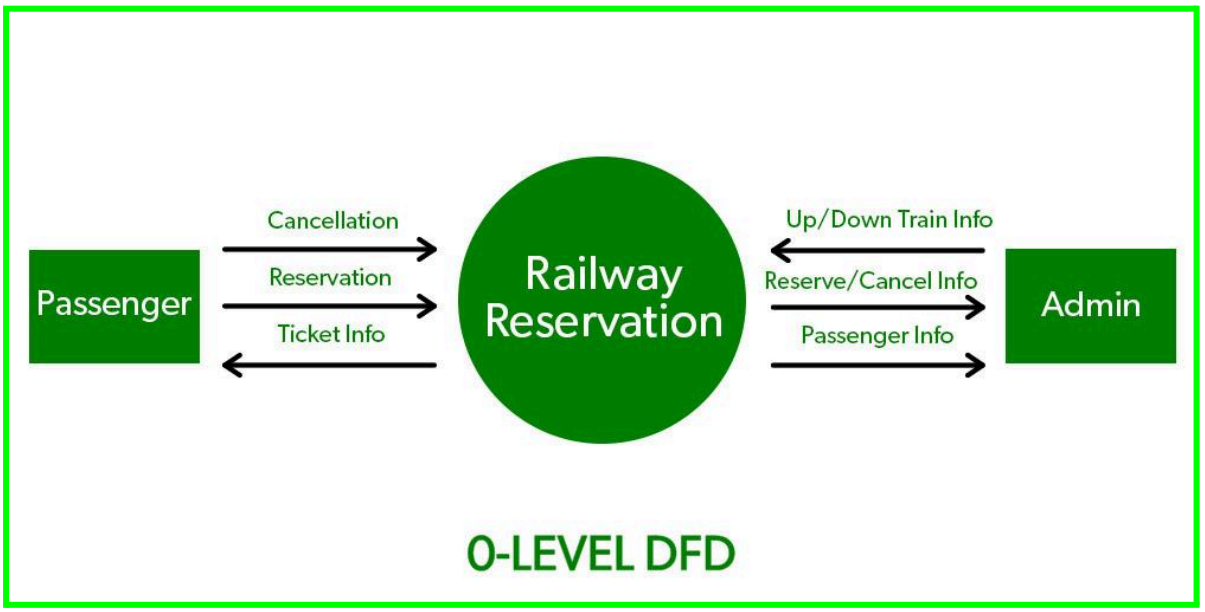
Maintenance and Support:

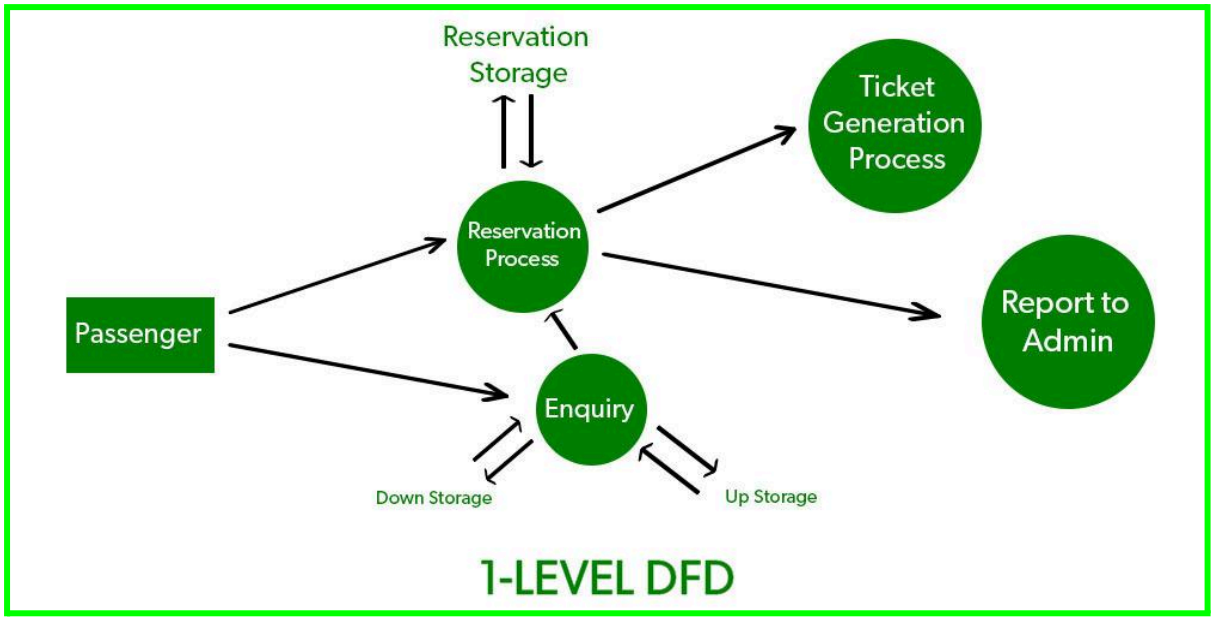
- Example: The system will be updated regularly for security and improvements.
- There will be a help desk for users to get assistance with issues.

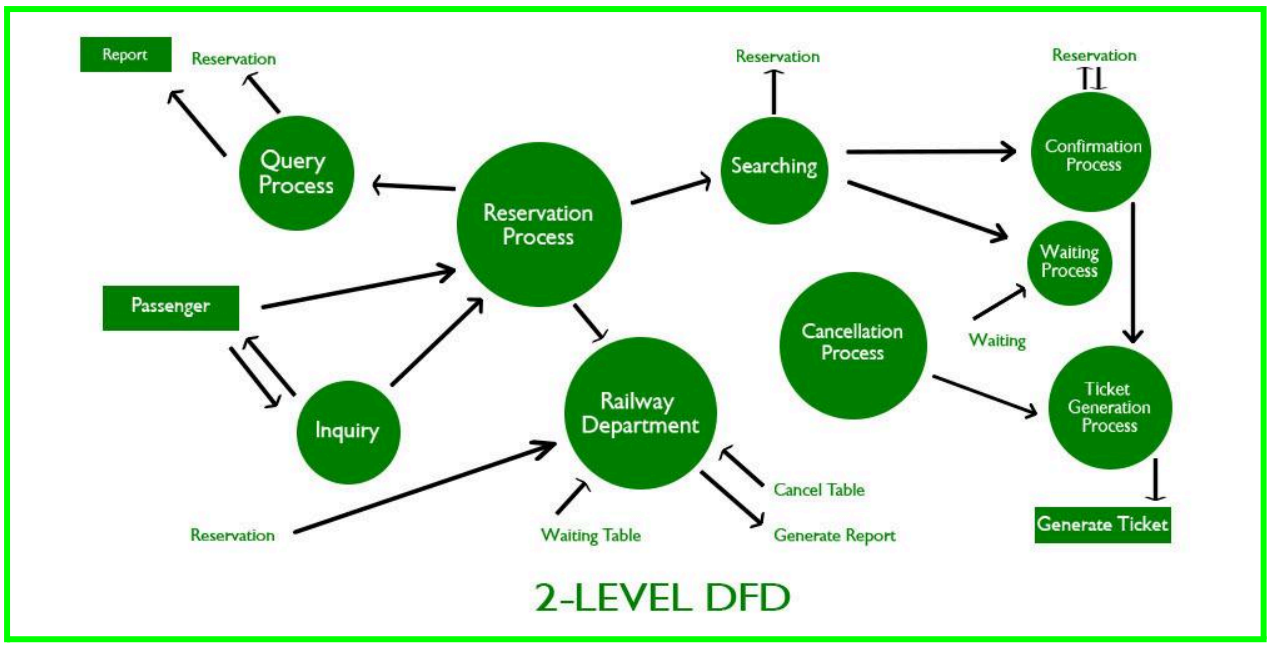
DFD

1. DFD data flow diagram visual representation h
2. System k andar data kese flow hota h
3. Dfd or data k movement ko dikhane k liye square, circle, line with arrow, dashed lines with arrow, one side open rectangle se dikhate h
4. Dfd use in the soft. Dev. when showing flow of the data

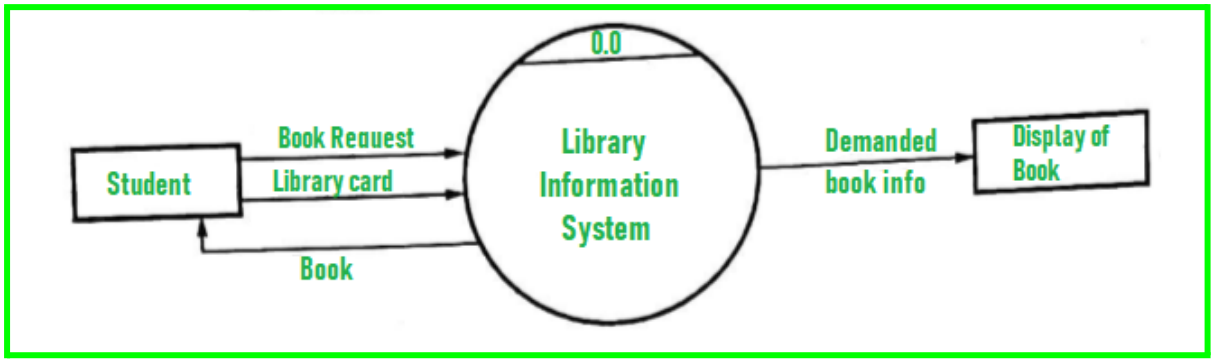
Railway Reservation System DFD



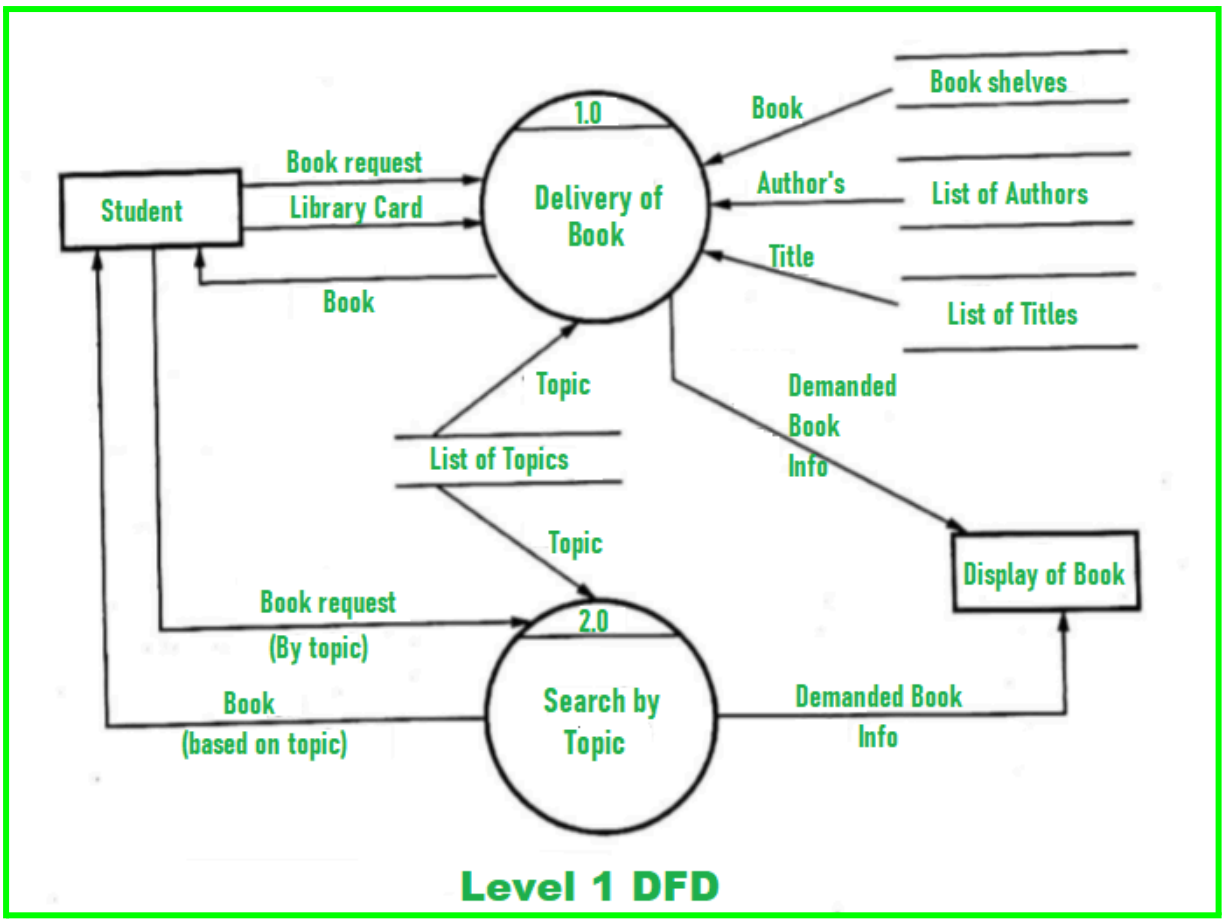




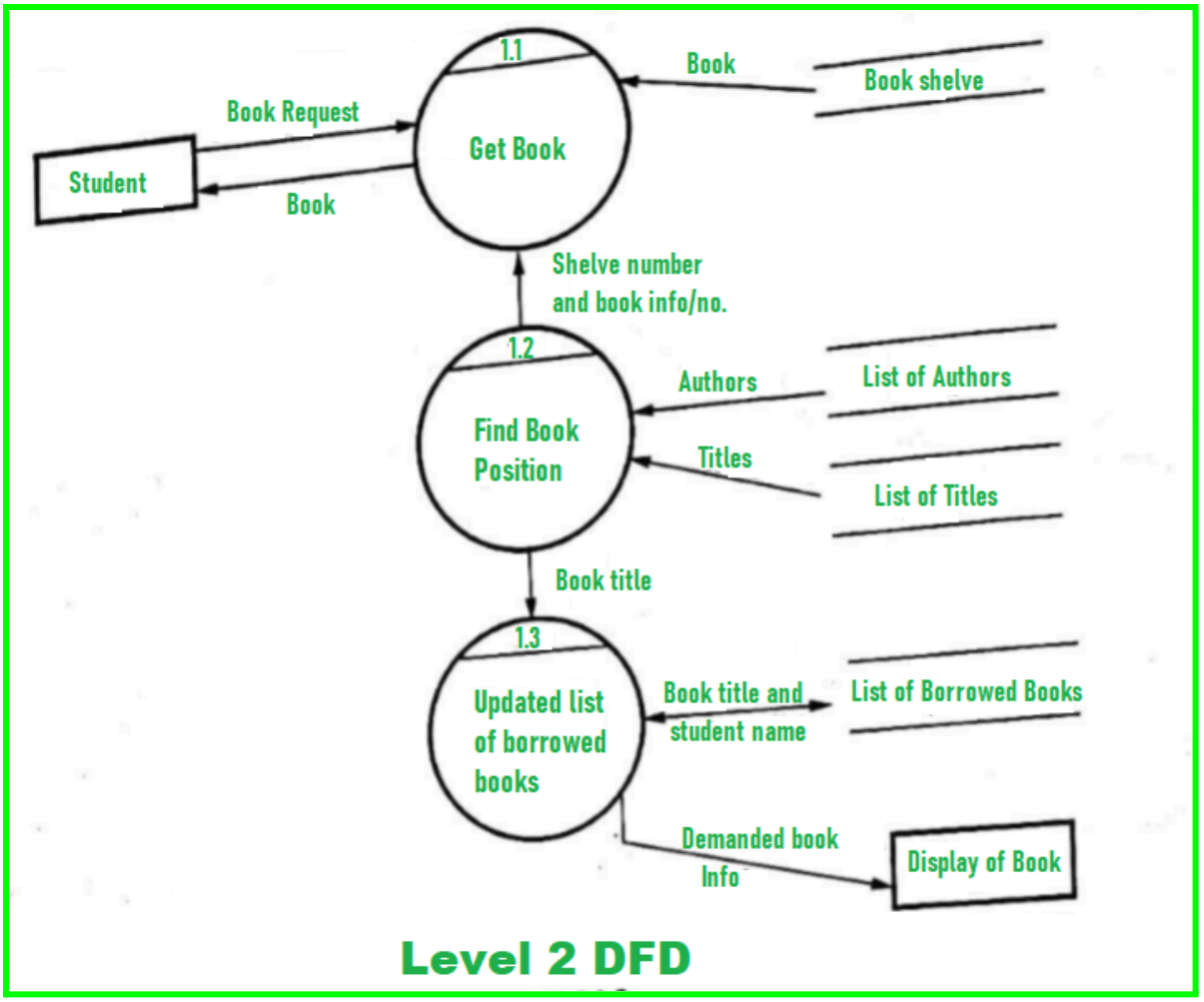
Library Management System DFD



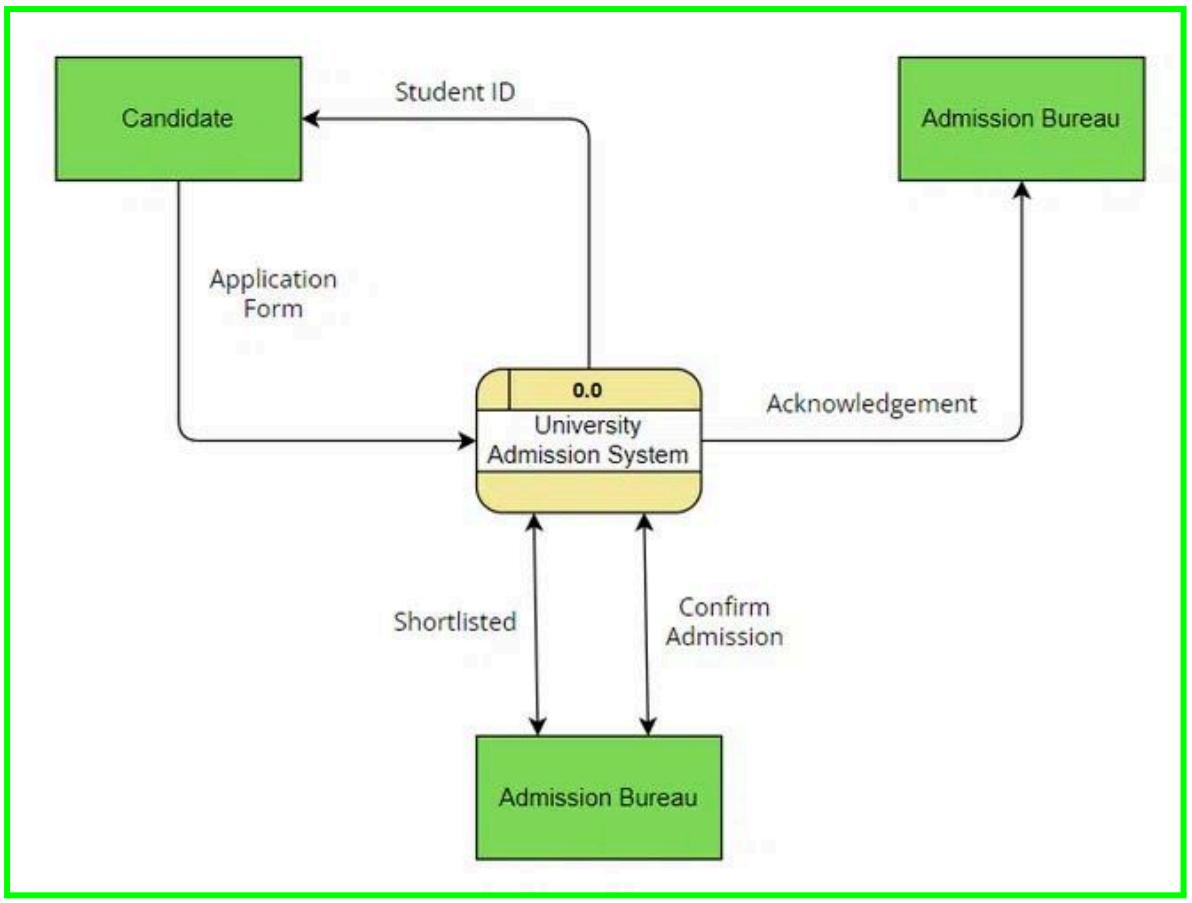
0 Level


Book shelves
1.0
Book
Book request
Student
Library Card
Delivery of Book
Author's
List of Authors
Title
List of Titles
Book
Topic
Demanded Book Info
List of Topics
Topic
Display of Book
Book request (By topic)
2.0
Book (based on topic)
Search by Topic
Demanded Book Info

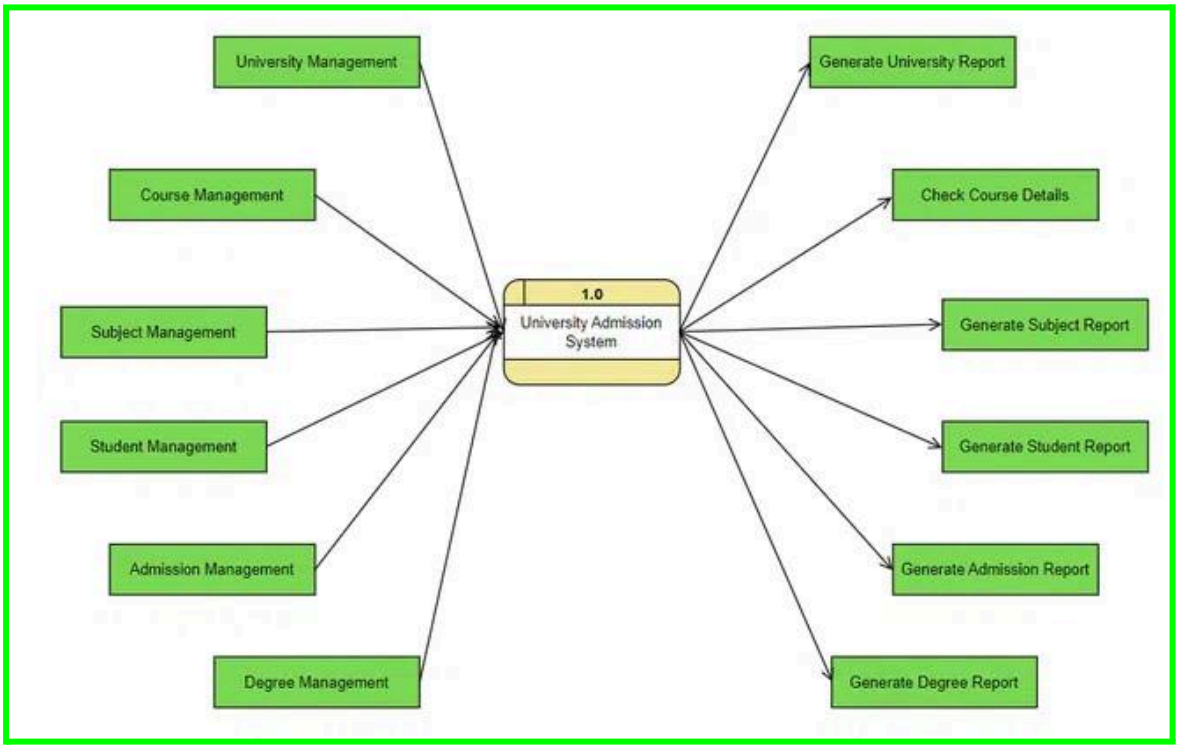
Level 1 DFD



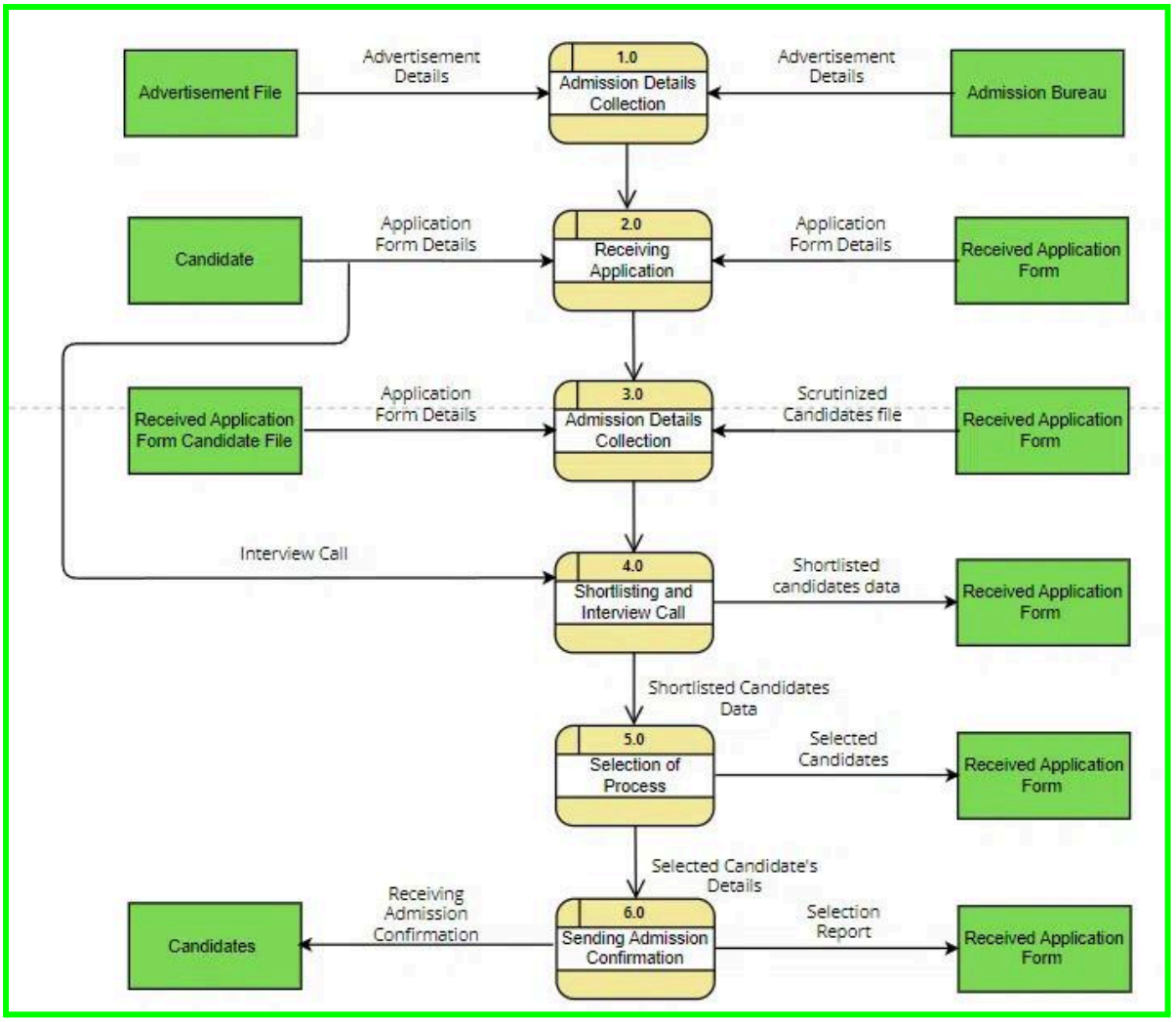
University Admission system DFD



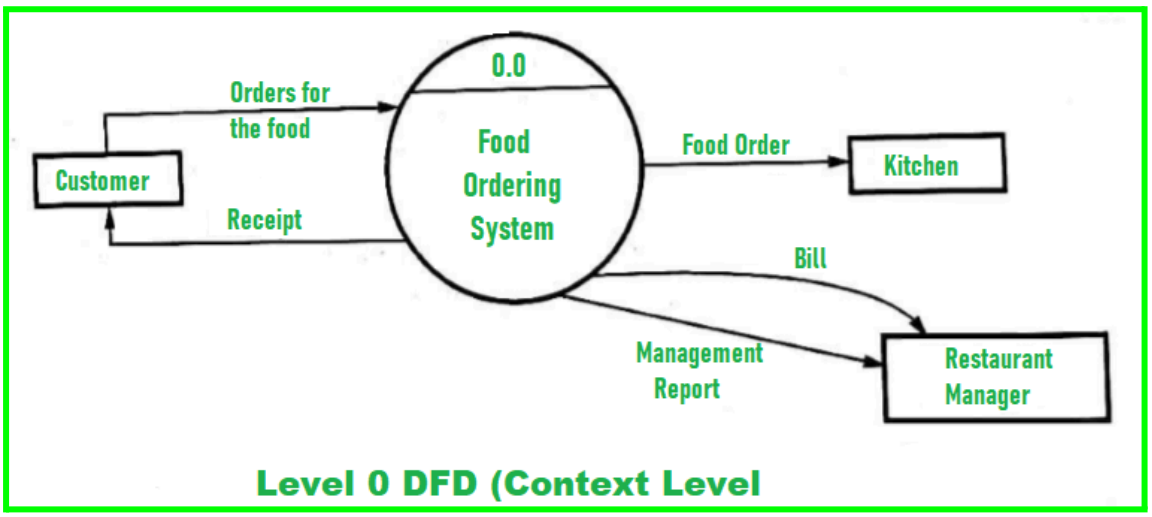
Level 0

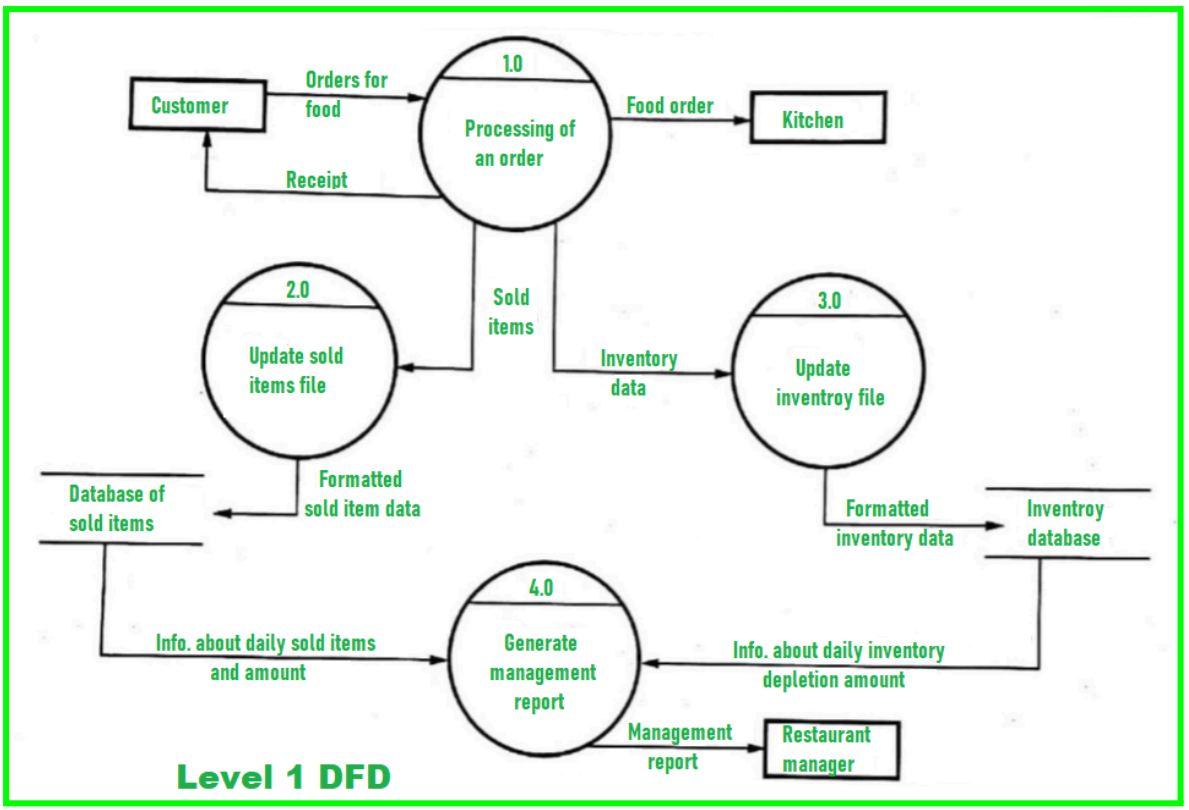


Level 1

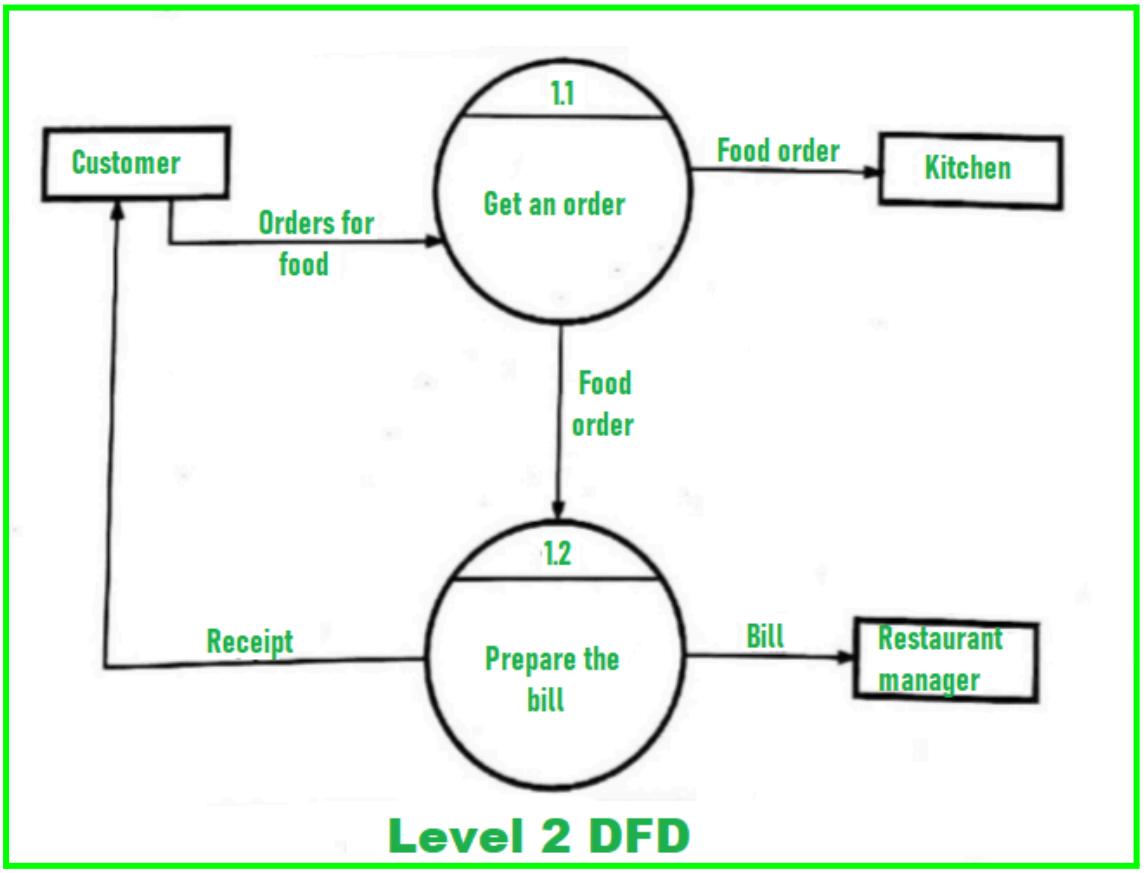


Food Ordering system DFD

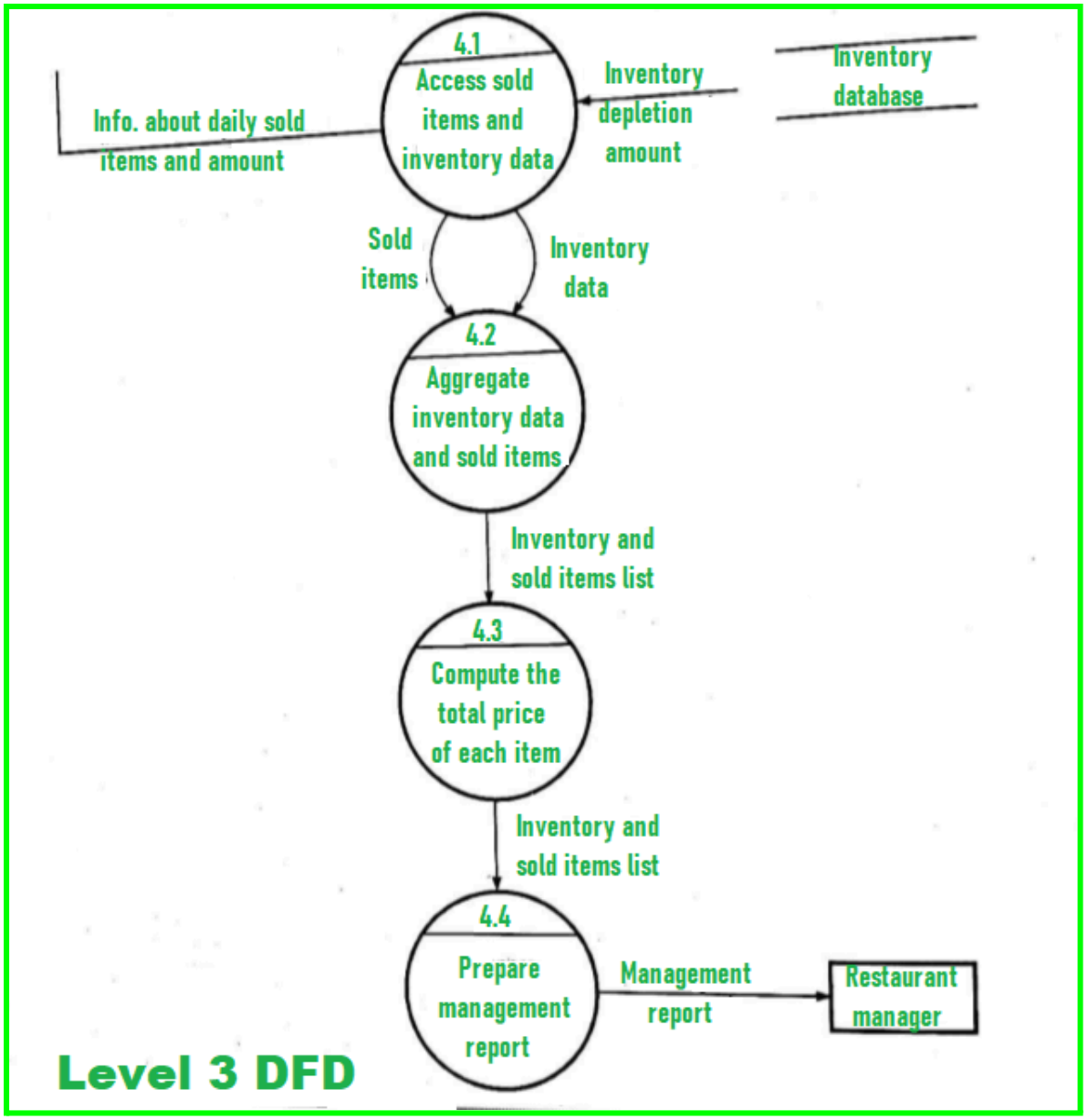




Level 1 DFD



Level 2 DFD

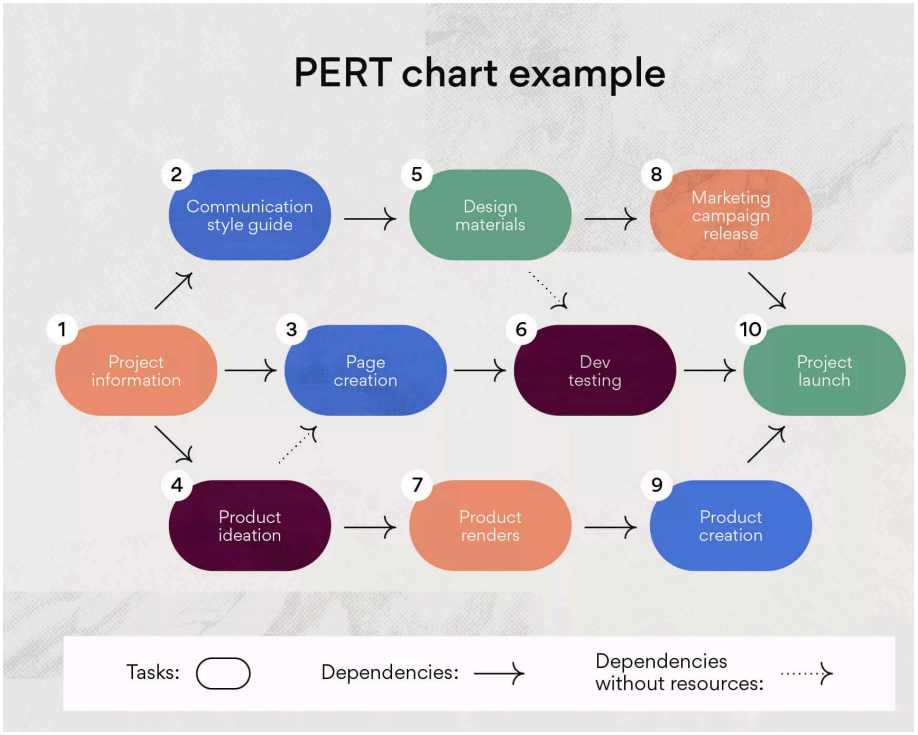


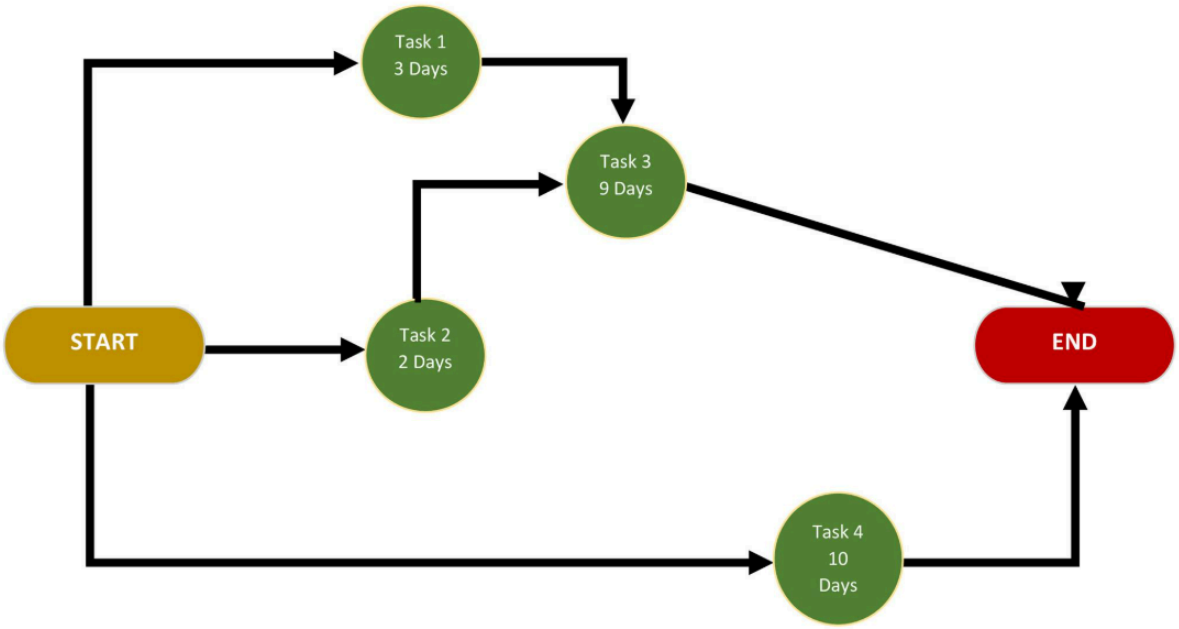
***PERT chart**

1. Program Evaluation Review Technique chart
2. PERT chart ek project management tool hota hai
3. Pert chat me project k tasks ke timeline aur dependencies ko visualize kiya jata h
4. Pert chat me project ke complete time ko manage kiya jata h
5. Pert chat ko dikhane k liye rounded rectangle, arrow, circle, dash arrow, line ka use karte h
6. Ex. jo complex projects ke tasks k dependencies aur timelines ko visualize karta h

5 steps of creating pert chart SLCTS

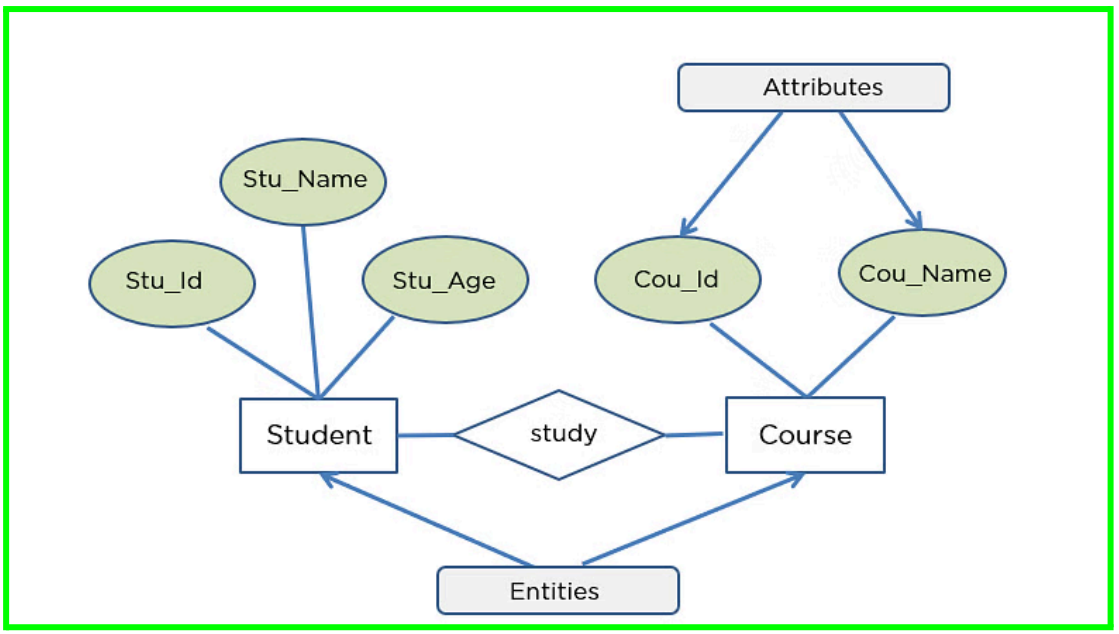
Search - Identify project tasks
List - define task dependenices
Combine - connect project tasks
Time - Estimate project timeframe
Setting -Manage task progress

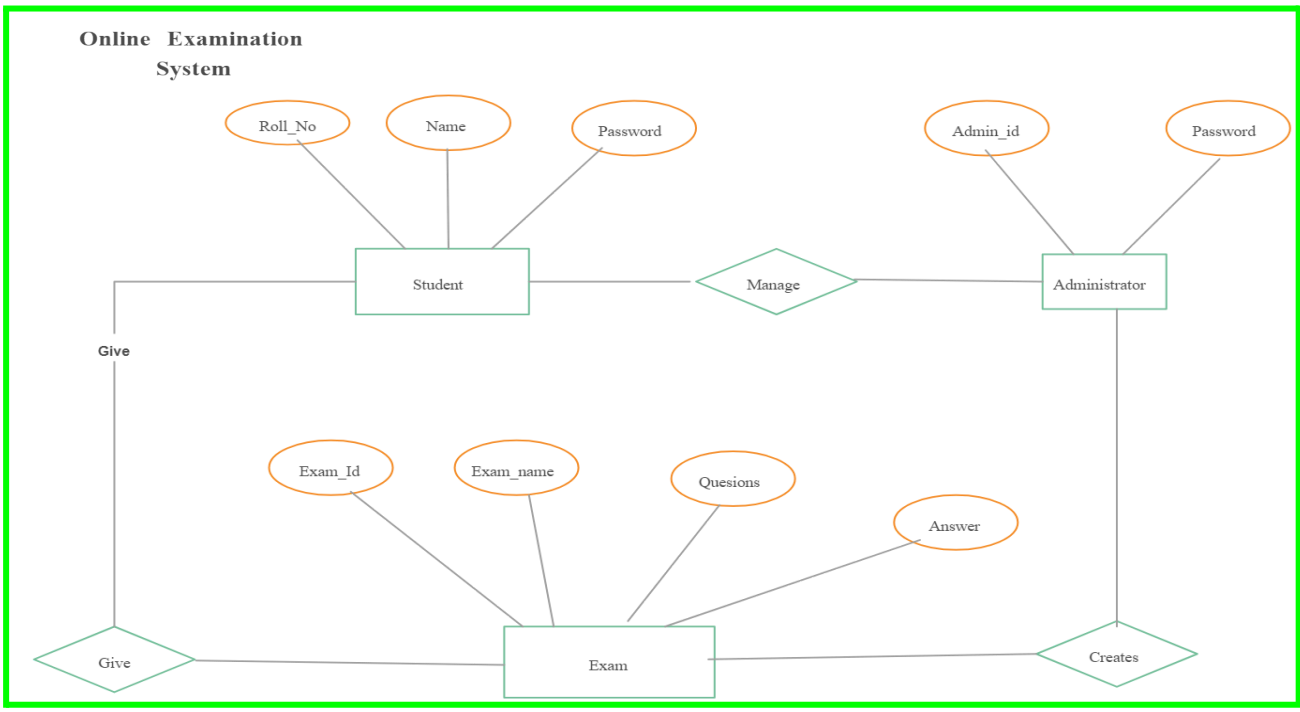




ERD

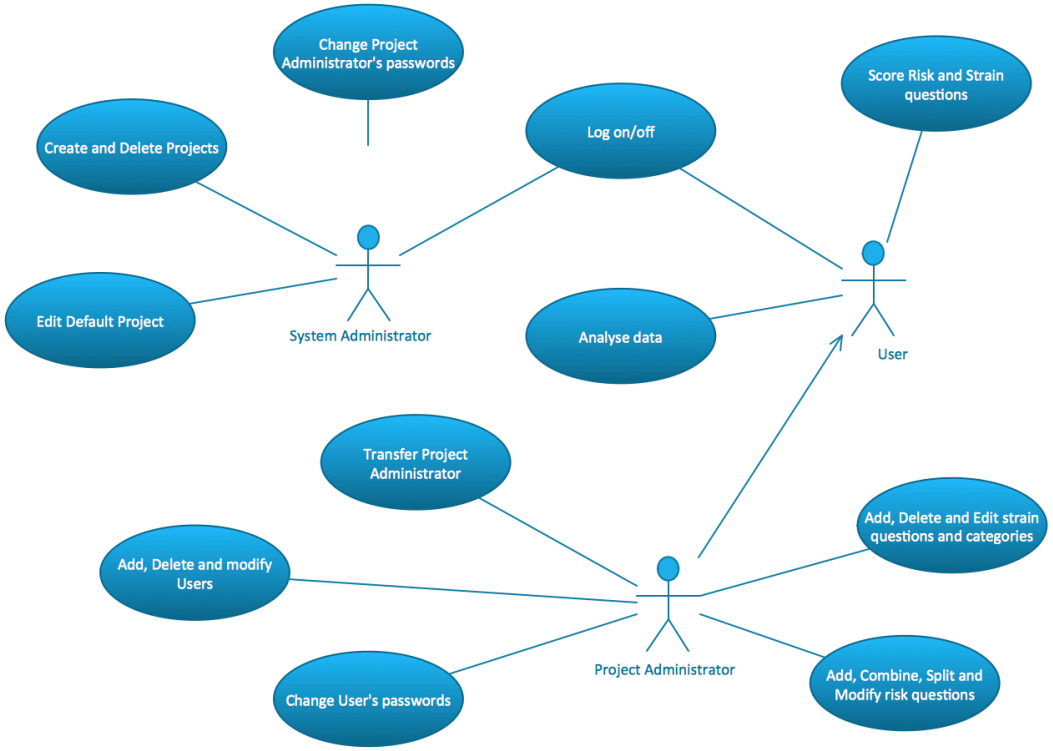
1. Entity-Relationship Diagram
2. ERD ek visual representation model hai
3. ERD database mein entities aur unke beech ke relationships ko dikhata h
4. ERD ko zyada tar database designers aur developers use karte h
5. *Ex. banking system*, jismein entities jaise *Customer*, *Account*, aur *Transaction* aur unke relationships dikhaye jaate h





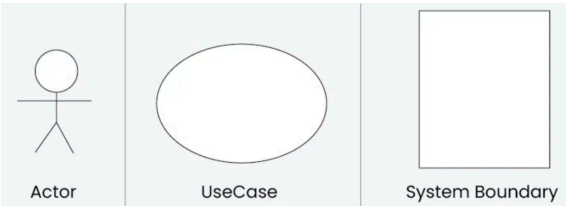
***UML**

1. (Unified Modeling Language) ek visual model h
2. uml software ke design ko diagrams ke through dikhata hai.
   uml software system ke design aur documentation ke liye use hoti hai.
3. Easy to understand from diagram, widely use
   Multiple uses like software design, system architecture, aur business planning
   Uml ko dikhane k liye oval, arrow, actor ka signs se use karte h
4. Ex. e-commerce website ke liye diagrams banana hai (jaise product dekhna, cart mein add karna, aur payment karna) aur system ke processes ko dikhaya jata hai



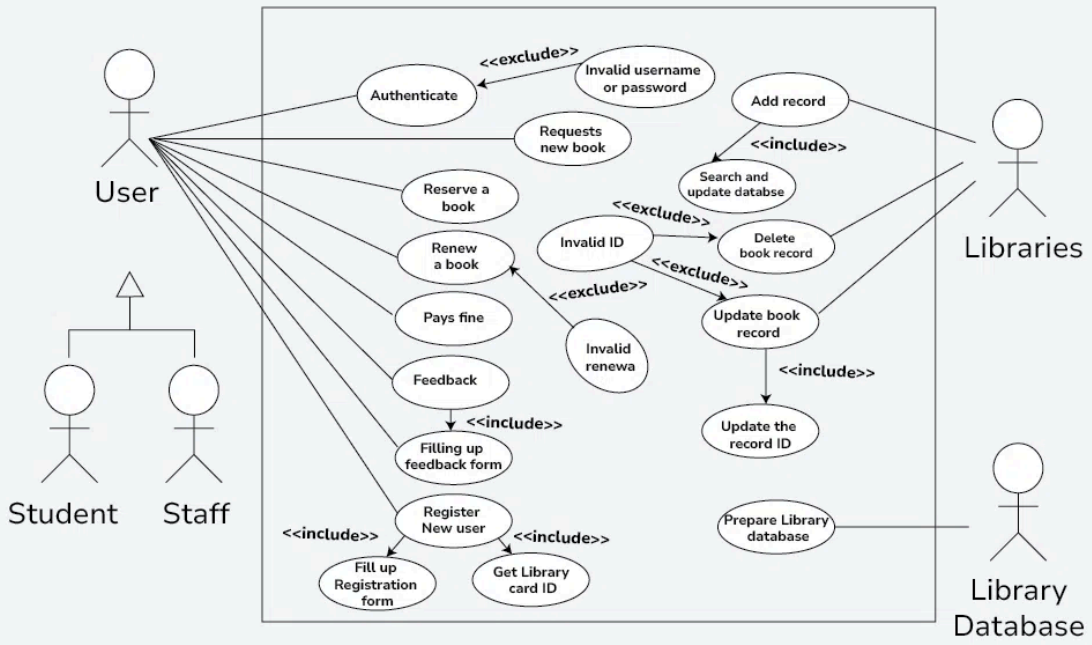
USE CASE DIAGRAM

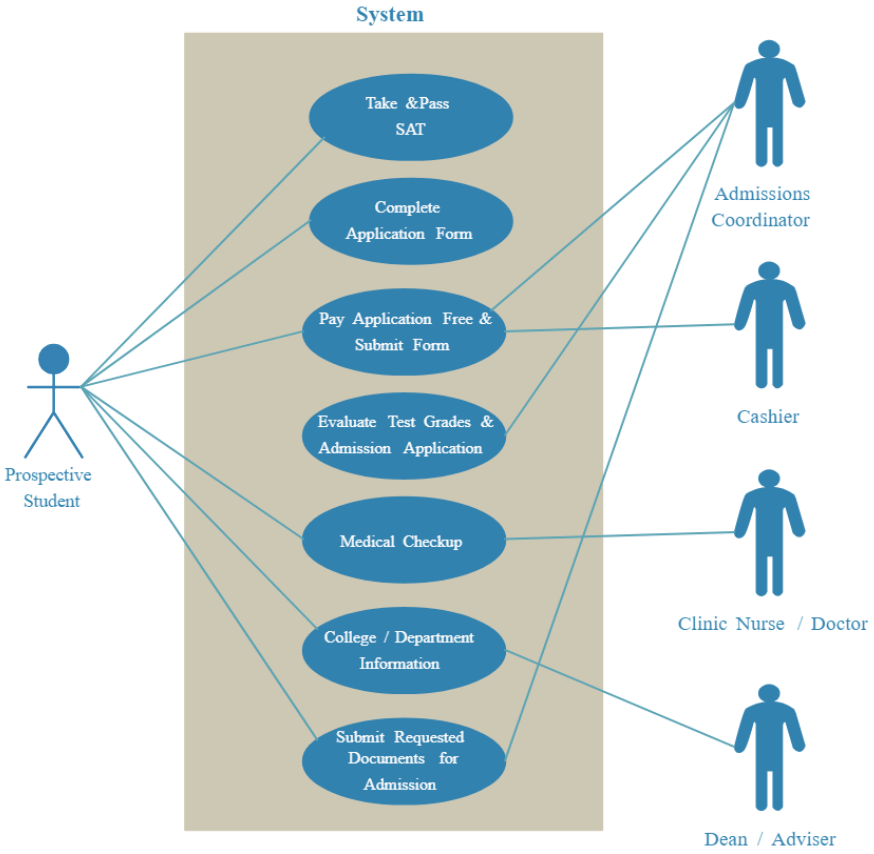
1. USD me system ke users aur unke interactions ko dikhanya jata hai.
2. Use case diagram notetions - actor, usecase, system boundry, arrow
3. UML is similar to ucd but not same
4. Usd use in software dev. When showing interaction of the system
5. Ex. Ek shopping app mein, customer 'add to cart,' 'view product,' aur 'checkout' actions perform karta hai; ye use case diagram mein dikhaye jaate hain

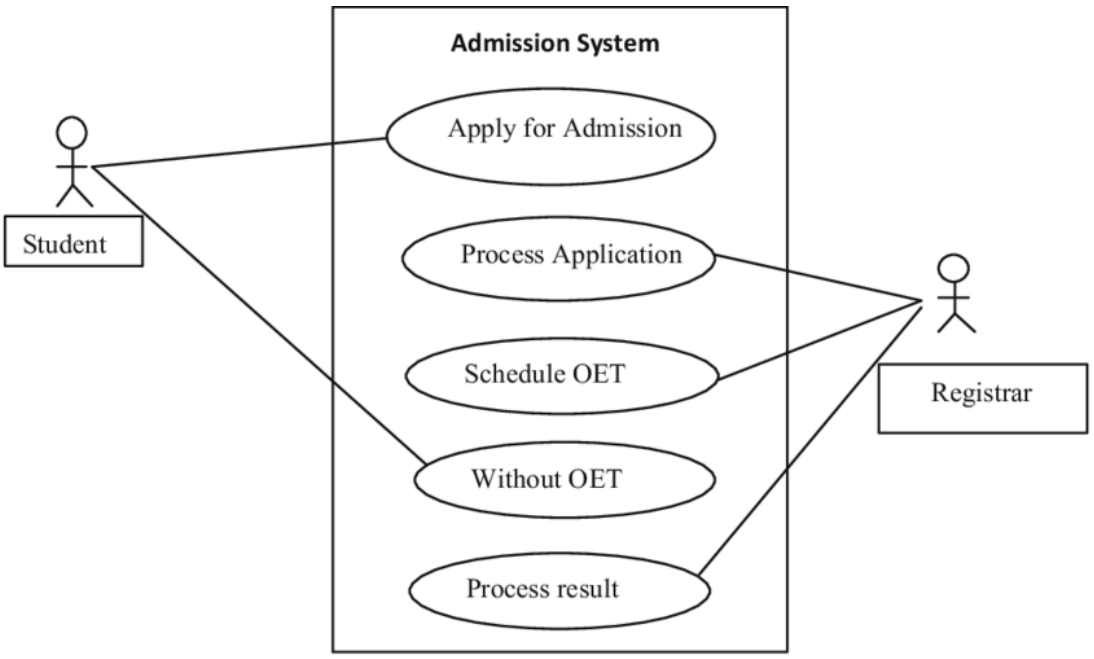


Use case diagram of Library Management System

Use Case Diagram of Library Management System



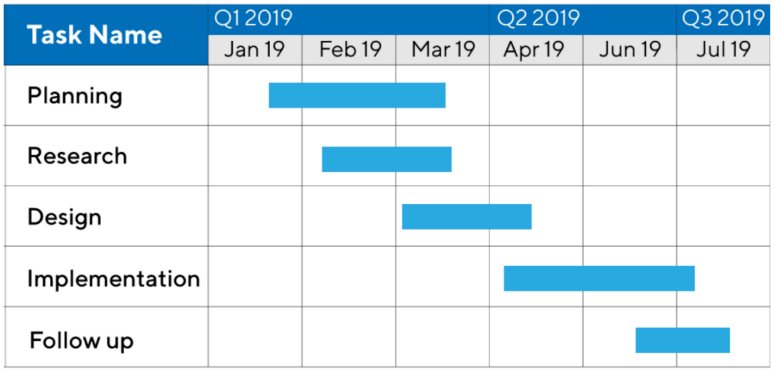




GANTT CHART

1. GANTT chart ek visual tool hai
2. jo projects k tasks aur unke complete time ko horizontally bars ki form mein dikhata h
3. Barchart :Time ko dikhane k liye bar chart ka use hota hai
   Har task ke liye ek bar hoti hai jo start aur end dates ko dikhati hai.
4. Sequence: Isme tasks ko sequence me rakha jata h
5. Gc Use in software developement when you are making calendar of the cs project

Project name:
Start date:
End date:
Present date:



ONLINE BANKING SYSTEM

Start: September 18, 2012
Finish: October 5, 2012
Report Date: October 5, 2012

Gantt Chart

WBS	Name	Work	Week 39, 2012							Week 40, 2012							Week 41, 2012																
			18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	Requirement Analysis	4d																															
2	Use Case Diag	3d																															
3	Class Diag.	2d																															
4	Sequence diag.	4d																															
5	Activity Diag.	3d																															
6	Component Diag.	2d																															
7	Test Cases	1d																															

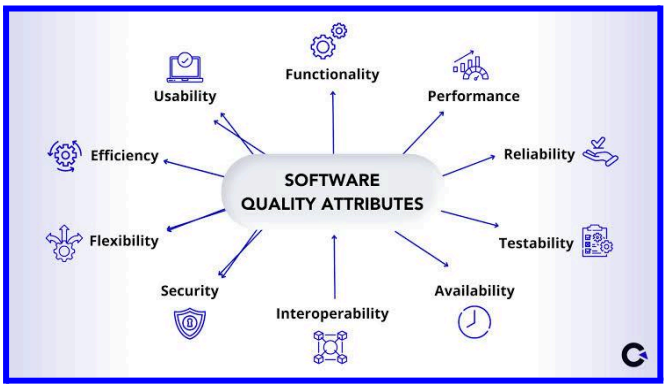
Tasks

WBS	Name	Start	Finish	Work	Priority	Complete	Cost
1	Requirement Analysis	Sep 18	Sep 21	4d		100%	
2	Use Case Diag	Sep 22	Sep 24	3d		100%	
3	Class Diag.	Sep 25	Sep 26	2d		100%	
4	Sequence diag.	Sep 27	Sep 30	4d		100%	
5	Activity Diag.	Sep 30	Oct 2	3d		100%	
6	Component Diag.	Oct 3	Oct 4	2d		100%	
7	Test Cases	Oct 5	Oct 5	1d		100%	

How will you ensure that the software developed by you meets the Quality benchmarks ? Define the term "Software Quality".

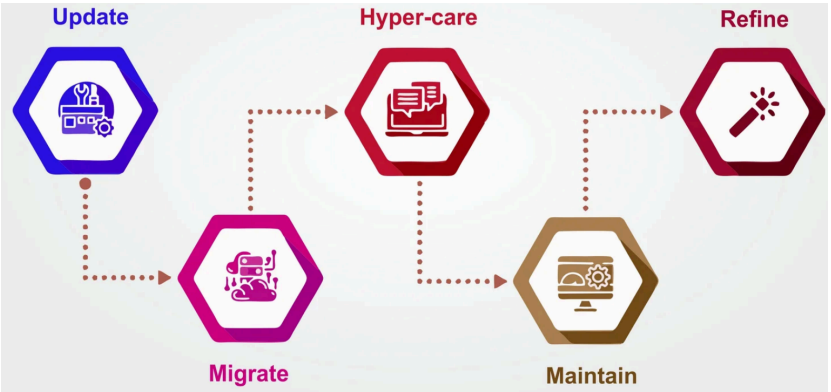
SOFTWARE QUALITY

1. software kitna accha aur reliable hai apni functionality aur use mein. software ke requirements
2. Software quality ko software k attributes se check kiya jata h
3. Sq Use in the software development when you are checking sq
4. Ex. : WINDOWS 11 ki software quality



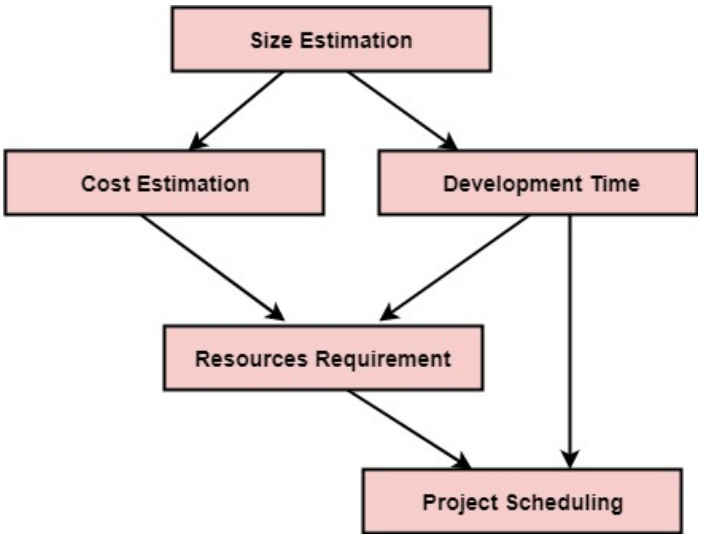
What is software maintenance ? Explain any *two* types of software maintenance in detail.

1. Software maintainance is the process of keeping software up to date and working properly
2. Software ki dekhbaal karte h
3. Software k har ek part ko maintain kiya jata h aur use sudhara jata h
4. Ex. vs code software ko maintain aur sudhara jaata h
5. types:
    - Adaptive maintenance
    - Corrective maintenance
    - Perfective maintenance
    - Preventive maintenance
    - Security maintenance



Project Planning

1. Project Planning ek process hai
2. Pp ka kaam, zaroori cheezen, aur time ka plan banate hain
3. Pp me project ko sahi tareeke se aur time par poora kiya jata h
4. Pp ko team ke leaders aur managers karte h
5. Pp use in the software development when you are making cs project
6. Ex. ek school event ka ayojan, jismein date, budget, aur zaroori cheezein tay ki jati hain taaki event sahi samay par aur achhe se ho sake



Verification

1. kisi device par user ki information ko identify kiya jata h
2. use in website and app for verification
3. EX. Jab aap OTP enter karte hain apne mobile number ki verification ke liye

4. Admin make verification method for user use

Validation

1. Kisi device par user ki infromation ko confirm kiya jata h
2. use in website and app for verification
3. Ex. jab aapka email address confirm karne ke liye link bheja jata hai
4. Admin make validation method for user use

DIFFERENCE

Validation system me user ko confirm karna hota h
Verification system me user identify karna hota h

List any *five* software testing tools indicating the testing phase in which they can be employed.

1. **Selenium**
   - **Testing Phase**: Functional Testing / Regression Testing
   - **Description**: An open-source tool for automating web applications across browsers.
2. **JMeter**
   - **Testing Phase**: Performance Testing / Load Testing
   - **Description**: Used for performance and load testing of applications.
3. **JUnit**
   - **Testing Phase**: Unit Testing
   - **Description**: A framework used for writing and running tests in Java programming.
4. **Postman**
   - **Testing Phase**: API Testing
   - **Description**: Used for testing RESTful APIs by sending requests and validating responses.
5. **QTP (QuickTest Professional)**
   - **Testing Phase**: Functional Testing
   - **Description**: A tool for automated functional and regression testing of software applications.

These tools cover different types of testing, including functional, performance, unit, and API testing.

scm

ankit5277@outlook.com
9389715277