

1. THE SOFTWARE PROBLEM.

- (Q1) What is software engineering? Differentiate betn industrial strength & student software.
- • Software engineering is defined as a process of analyzing user requirements & then designing, building and testing software application which will satisfy those requirement
 - IEEE, in its standard, defines software engineering as the application of a systematic, disciplined, which is a computable approach for the development operation & maintenance of software.
 - Software engineering is an engineering branch, related to the evolution of the software product using well defined scientific principles, techniques, procedures.
 - The result of software engineering is an effective & reliable software product.

<u>Industrial strength software</u>	<u>Student software</u>
• others are the users	• Developers is the user.
• Bugs are not tolerated	• Bugs are tolerable
• UI is very important	• UI is not important
• Documents are needed for the user as well as for org & proj	• Documentation is not needed
• Supports important function	• software not in critical use.
• Reliability, robustness are imp.	• Reliability, robustness are ^{not} imp
• Heavy investment is done	• No investment is done
• Testing → 5% effort is used	• 30-50% of effort is used.
• Portability is the key issue less effort, high productivity	• Portability is not a issue. • more effort less productivity

Q2) List and explain the 6 software quality attributes.

Software	functionability	usability	Maintainability
Quality			
Reliability	Efficiency	Portability	

- **functionability :-** The capability to provide functions which meet stated and implied needs when the software is used.

It is the ability of system to continue to keep operating over time.

- **Reliability :-**

It is how the user is utilizing a system effectively & the ease of which users can learn to operate or control the system. The well known principle of usability is (Keep it simple stupid). Software application should be user friendly.

- **Efficiency :-**

The capability to understand, learn & use.

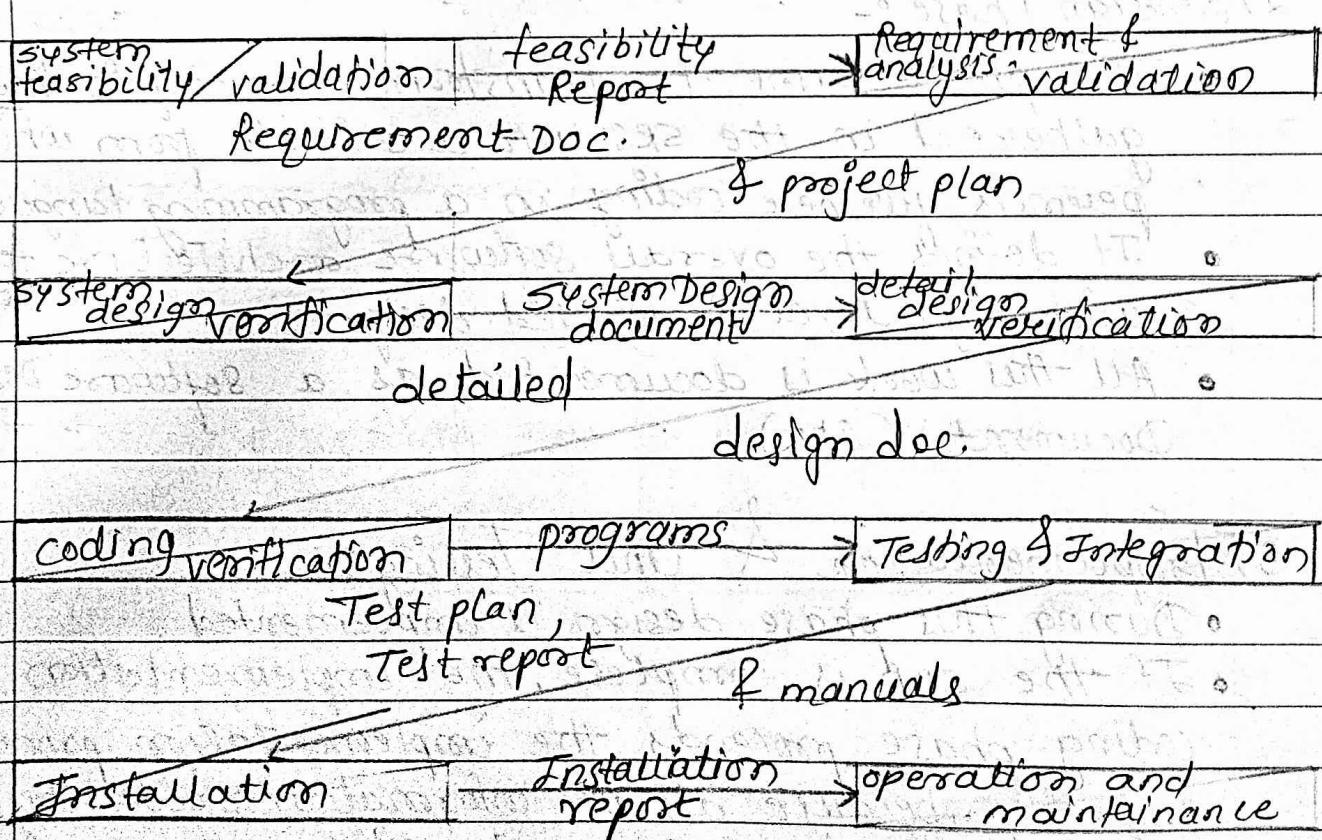
- **Maintainability :-**

It is the ability of a software application to maintain easily & support changes cost-effectively. It is capability to be modified for purpose of making corrections, improvements or adaption.

- **Portability :-**

The capability to be adapted for different specified environment without applying actions or means other than those provided for this purpose in the product.

Q3) Explain the phases of waterfall model.



M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

- These are five phases

- 1) Requirement analysis & specification phase:-
- Aim of this phase is to understand the exact requirements of the customer & to document them properly.
 - Both the customer & software developer work together so as to document all the function performance & interfacing requirement of the software.
 - It describes the "what" of the system to be produced & not "how".
 - A large document called software requirement specification (SRS) document is created which contains a detailed description of what will the system do in a common language.

2) Design Phase :-

- This phase aims to transform the requirements gathered in the SRS into a suitable form which permits further coding in a programming language.
- It defines the overall software architecture together with high level & detailed design.
- All this work is documented as a Software Design Document (SDD).

3) Implementation & Unit Testing :-

- During this phase design is implemented.
- If the SDD is complete, the implementation or coding phase proceeds, the implementation proceeds smoothly, because all the information needed by software developers is contained in SDD.
- During testing the code is thoroughly examined & modified, small modules are tested in isolation initially.

- After that these modules are tested by writing some overhead codes to check the interaction b/w these modules & the flow of intermediate output.

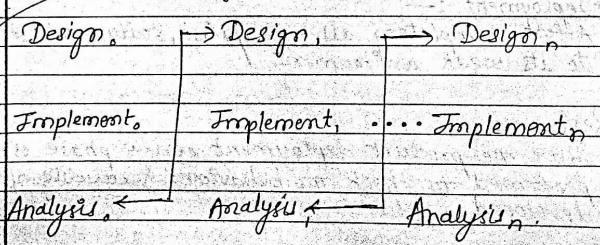
4) Integration & System Testing :-

- This phase is highly crucial as the quality of the end product is determined by the effectiveness of the testing carried out.
- The better output will lead to satisfied customers, lower maintenance costs and accurate results.
- Unit testing determines the efficiency of the individual modules.
- However in this phase, the modulus are tested for their interactions with each other & with the system.

5) Operation & maintenance phase :-

- Maintenance is the task performed by every user once the software has been delivered to the customer installed & operated.

Q4) List and explain the phases of Iterative development model with diagram.



1) Requirement Gathering & Analysis :-

- Requirements are gathered from customers & check by analyst whether requirements will fulfil or not.
- Analyst check that need will achieve within budget or not.
- After all of this, the software team skips to the next phase.

2) Design :-

- Team designs the software by the different diagrams like data flow diagram, activity diagram, class diagram, state transition diagram, etc.

3) Implementation :-

- Requirements are written in the coding language & transformed into computer programs which are called software.

4) Testing:-

- After completing the coding phase, software testing starts using different test methods, but the most common are white box, black box & grey box test method.

5) Deployment :-

- After completing all the phases, software is deployed to its weak environment.

6) Review :-

- After the product deployment review phase is performed to check the behaviour & validity of the developed product.

• And if there are any errors found then the process starts again from the requirement gathering.

7) Maintenance :-

- In the maintenance phase, after deployment of the software in the working environment there may be some bugs, some errors or new updates are req.
- Maintenance involves debugging & new addition options.

Q5) elaborate the phases & milestones of each phases of RUP model with diagram.

• Phases	Inception	elaboration	construction	Transition
Requirements	High	High	low	nil
Anal & Design	low	high	med	nil
Implementation	Nil	low	high	low
Test	Nil	low	high	med
Deployment	Nil	nil	med	high
proj Management	med	med	med	med
conf management	low	low	high	high
• milestones : lifecycle objectives	lifecycle architecture	initial operation capability	Product release	

M	T	W	T	F	S	S
Page No.:						
Date:						

YOUVA

M	T	W	T	F	S	S
Page No.:						
Date:						

YOUVA

1) Inception :-

- Communication & planning are the main ones.
- Identifies the scope of the project using a case-use model allowing managers to estimate cost & time requirements.
- Customer requirements are identified & then it becomes easy to make a plan for the project.
- The project plan, project goal, risks, use-case model, project description are made easy.
- Project is checked against the milestone criteria then the project can be either cancelled or redesigned.

2) Elaboration :-

- Planning & modelling are the main ones.
- A detailed evaluation & development plan is carried out & diminished risk.
- Revise & redefine use-case model, business case & risk.
- Again checked against milestone criteria & if couldn't pass these criteria then again project can be cancelled or redesigned.
- Executable architecture baseline.

3) Construction :-

- Project is developed & completed.
- System or source code is created & then testing is done.
- Coding takes place

4) Transition :-

- The final project is released to the public.
- Transit the project from development into production.
- Update project documentation.
- Beta testing is conducted.
- Defects are removed from the project based on feedback from the public.

Q6) Short Notes :-

① Prototype model :-

- The prototype model requires that before carrying out the development of actual software, a working prototype of the system should be built.
- A prototype is a toy implementation of the system.
- A prototype turns out to be a very crude version of the actual system, possibly exhibiting limited functional capabilities, low reliability & inefficient performance as compared to actual software.
- In many instances, the client only has a general view of what's expected from the software product.
- In such a scenario where there's an absence of detailed information regarding the input to the system, the processing needs to & the output required, the prototyping model may be employed.

* Steps of prototyping model :-

- Requirement gathering & Analyst
- Quick decision
- Build a prototype
- Assessment or user evaluation
- Prototype refinement
- Engineer product.

* Advantages of prototype model :-

- ~~Model~~ where requirement are changing
- Reduce the risk of incorrect user requirement
- Regular visible process and management
- Support early product marketing
- Reduce maintenance cost
- Error can be detected much easier as the system is made side by side.

* Disadvantage of prototype model :-

- An unstable implemented prototype often become the final product
- Require extensive customer collaboration
- Difficult to know how long the project will be last
- Prototyping tool are expensive
- It is a time-consuming process.
- Special tools & techniques are required to build a prototype.

① Time Boxing Model :-

- In time boxing model, development is done iteratively as in the iterative enhancement model.
- However, in time boxing model, each iteration is done in a timebox of fixed duration.
- The functions to be developed is adjusted to fit the duration of timebox.
- Moreover, each timebox is divided into a sequence of fixed stages where each stage performs a clearly defined task that can be done independently.
- This model also require that the time duration of each stage is approximately equal so that pipelining concept is employed to have the reduction in development time & product releases.

* Advantages of time boxing :-

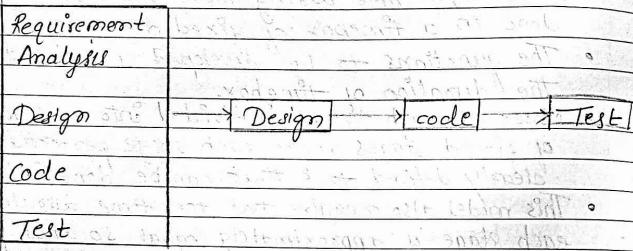
- Speeds up the development process & shortens delivery time
- well suited to development project with a number of features in short time period.

* Disadvantages of time boxing :-

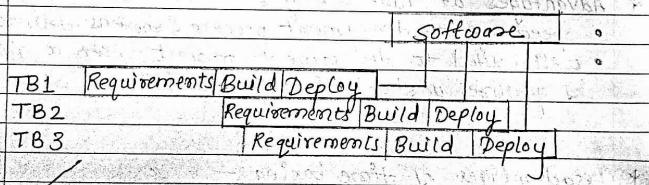
- Project management becomes more complex.
- Not suited to projects in which entire development work cannot be divided into multiple iterations of almost equal duration.

* Diagrams of prototyping & timeboxing model.

• Prototyping Model



• Timeboxing model.



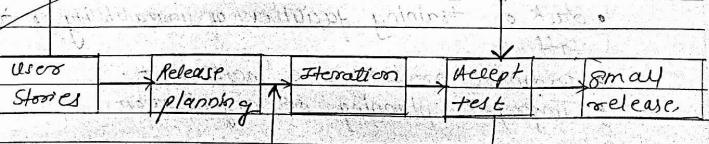
(Q7) Explain in detail Extreme programming (XP).

- XP is an agile software development framework that aims to produce higher quality software, higher quality of life for the development team. XP is the most specific of the agile framework regarding appropriate engineering practices for the software development.
- Its agile development approaches evolved in 1990's

as a reaction to documentation & bureaucracy-based processes, particularly the waterfall approach.

- Agile approaches based on some common principles:-
 - ① working software - key measure of progress in a project
 - ② for progress, rapid development & delivery of software in small increments
 - ③ Entertainment of late changes in requirements.
 - ④ Prefer face to face communication over documentation.
 - ⑤ Necessity of continuous feedback & customer involvement in developing good-quality software.
 - ⑥ Simple design that evolves & improves with time.
 - ⑦ Delusion of the delivery dates empowered by teams of talented individuals.

- Popular, well known approaches in the family of agile methods.
- To accommodate change, development process to be lightweight & quick.
- Iterative software development.
- Relies on face-to-face communication, simplicity & feedback to ensure designed changes to reflect quickly & correctly.
- Each story is written on a separate card.



OVERALL PROCESS IN XP.

CASE STUDY

- A Automated library Management System. Write a software problem, cost, schedule, quality, scale & maintenance points related to case study topics.

→ Automated library Management System—

- A library management system is a software that is designed to manage all the functions of a library.
- Used to maintain library records.
- Tracks the records of the no of books in a library, how many books are returned & how many books are published, issued, renewed or late fine charges, etc.

easy data
management

Managing book resources

library Management

No duplication
of work

Reduced
Time

* Software Problem :—

- Financial drawback
- lack of training facilities or unavailability of trained staff.
- Hardware products are insufficient.
- Improper planning or organization.

* Purpose 8 →

- To increase the efficiency of operations
- To make services more effective and accurate.
- To provide better accessibility for remote users.
- To satisfy library & patron needs that can't be met by manual methods.
- To provide easy access to other network or system resources including the internet.
- To improve the management of their physical & digital resources.

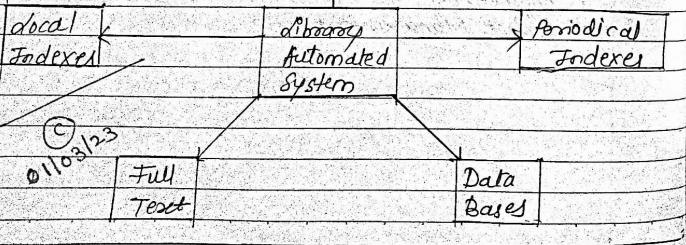
* Cost, Schedule, Quality & Scale 8—

- There are 3 stages scheduled in this system namely (1) The planning stage (2) The designing stage & (3) Operational stage.
- The actual cost of development ranges from \$25000 to \$80000 while if you choose advanced features or develop the app for more platforms then cost will automatically increase.
- Quality will mainly defined by the speed accuracy of reference & information delivery services opens access to both physical collection & online retrieval system.
- There are 2 clear dimensions in project
 - Engineering
 - Project Management.

* Maintenance :-

- The authors deals with the concept of maintenance of a library system.
- Deleted part, library blocking & multiple maintenance support, three advanced features that can be implemented in many file based library control systems, are discussed.
- Programmers are provided with a development & maintenance environment that will allow them to maximize time spent maintaining multiple versions of old code.
- The concept of deleted part involved retention of data on parts that previously existed in library control system.
- Library blocking allows 'viewability' of parts in a library system to be controlled on a part-by-part basis.
- The multiple maintenance library concept helps maintain quality software by monitoring the library system for software changes.

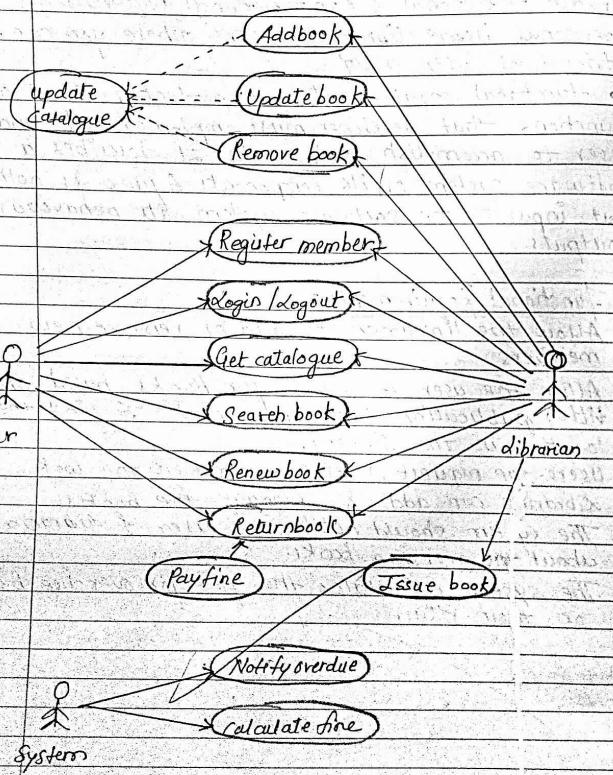
Cataloguing
Records



Assignment - 2.

- (Q1) what is functional requirement in SRS ?
 Create functional & non functional requirements for personal home library software where you can have friend as user easily.
- functional requirements are product features or functions that developer must implement to enable user to accomplish their tasks. It describes a software system or its components. A func is nothing but input to the software system, its behaviour & outputs.
 - Functional Requirements:-
 - Allow the librarian to add or remove new members
 - Allow the user to search for books based on title, publication date, author, etc & find their location in the library.
 - Users can request, receive or renew the book.
 - Librarian can add & manage the books
 - The system should notify the user & librarian about the overdue books.
 - The system calculates the fine for overdue book on their return.

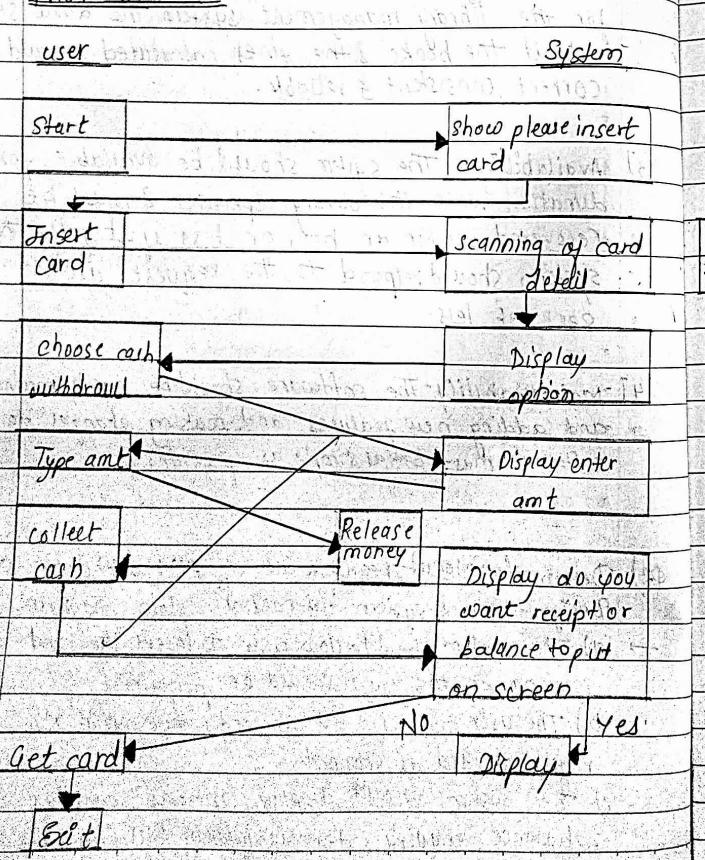
UML CASE DIAGRAM



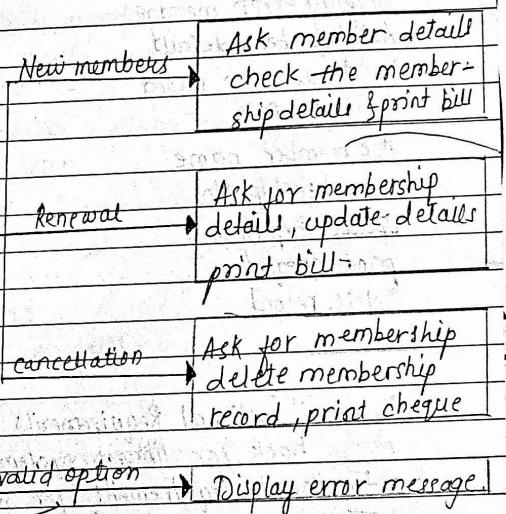
- Non-fictional Requirement
 - 1] **Usability** - It is the main non-fictional requirement for a library management system. The UI should be simple enough for everyone to understand & get the relevant information without any special training. Different language can be provided based on the requirement.
 - 2] **Accuracy** - It is another non-fictional requirement for the library management system. The data stored about the books & the files calculated should be correct, consistent & reliable.
 - 3] **Availability** - The system should be available for the duration when the library operates & must be recovered within an hour or less if it fails. The system should respond to the request within 2 sec or less.
 - 4] **Maintainability** - The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible.
- Q2) Design functional requirement of ATM withdraw cash
Design user & system interaction flow diagram.
 - a) The system should allow user to insert the card & scanning of the card should be performed.
 - b) The user is insisted to enter the pin no. after registered card is scanned.
 - c) The system should display options such as withdraw balance enquiry, deposit money, Bill pay, etc.
 - d) The user should select the option of cash withdrawal.

- e) The system should allow the user to enter the desired amount.
- f) The system should provide money.
- g) The system should ask the whether to display the amount on screen & if the user wants to print the receipt.
- h) The system should inform the user to remove card when the process is over.

FLOW DIAGRAM.



- Q3) Draw a decision tree and decision table for Library Membership Automation Software (LMS).
- The following decision tree shows graphical illustration of library membership automation software, when obtaining data from the user, the system makes a choice & then performs the corresponding actions.



- A decision table is used to represent a complex processing logic in a tabular or matrix form. The upper row of table specifies the variable or conditions to be evaluated. The lower row of the table specifies the action to be taken when the corresponding conditions are satisfied.
- A column in a table is called a rule. It implies that if a condition is true, then the corresponding action is to be executed.

	M	T	W	T	F	S	S
Page No.:							
Date:							

YOUVA

M	T	W	T	F	S	S
Page No.:						
Date:						

Conditions

Valid Selection	No	Yes	Yes	Yes
New member	-	Yes	No	No
Renewal	-	No	Yes	No
Cancellation	-	No	No	Yes

Actions

Display error message	x	-	-	-
Ask member detail	-	x	-	-
Build customer record	-	x	-	-
Generate Bill	-	x	x	-
Ask member name	-	-	x	x
membership No.				
Update Expiry date	-	-	x	-
print cheque	-	-	-	x
Delete record	-	-	-	x

- (Q) Design Functional Requirements to search book & renew book for library system.
- • Functional Requirement for searching of a book.
- The system should allow both the users of librarian to search for a book according to published date, title, author, etc.
 - The system should allow the user to login into System.
 - User should be able to see the catalogue books & choose the books upto limit set by the library.
 - System should also set the due date to return the book.
 - System should give the details about the avail of book.

- f) Now the system should tell the user to logout from the software.
- Functional Requirement for renewal of Book.
- a) The system should tell the user to login & select the renew option.
- b) System will check the book details & calculate the fine if it is overdue.
- c) The system should check the availability of the book.
- d) The system should update the details regarding the book & the user.
- e) The system should show the logout option

Q5] Short Notes On :

① Format of SRS document -

- 1) Introduction
 - i) Purpose of this document
 - ii) Scope of this document
 - iii) Overview
- 2) General Description
- 3) Functional Requirements
- 4) Interface Requirements
- 5) Performance Requirements
- 6) Design constraint
- 7) Non-functional attribute
- 8) Preliminary schedule & budget
- 9) Appendices

SRS format is complete specification & description of requirements of software that needs to be fulfilled for successful development of software system.

a) Introduction :-

→ Purpose :- Main aim of why this document is necessary & what is the purpose of this document is explained & described.

→ Scope :- Overall working & main objective of document & what value it will provide to customer is described & explained.

→ Overview :- Description of product is explained. It's simply summary or overall review of product.

• General Description :-

→ General functions of product is mentioned. It also describe features of user community.

• Functional requirements :-

→ Possible outcome of software system which include effects due to operation of program is fully explained.

• Interface requirements :-

→ How software program communicate with each other or users either in the form of any language code or message are fully described & explained.

• Performance requirements :-

→ How a software system performs desired functions under specific condition is explained.

• Design Constraints :-

→ Constraints which simply means limitation or restriction are specified & explained for design term.

• Non-Functional Attributes :-

→ Non-functional attributes are explained that are required by software system for better experience.

• Preliminary schedule & budget :-

→ Initial version & budget of project plan are explained which include overall time duration required & overall cost required for development of project.

• Appendices :-

→ Additional informed like references etc are given & explained.

b) Project Cost Management

a) Project Cost Management is the process of estimating, budgeting & controlling cost throughout the project life cycle, with the objective of keeping expenditures within the approved

b) It helps to create financial baseline against which project manager can bench-mark the current status of their project & realign the direction if needed.

- c) While cost management is viewed as a continuous process there are four steps
- Project Resource Planning
 - Cost estimation
 - Cost Budgeting
 - Cost control.

c) Project Charter:

- a) It is the 1st document for project & lays down the foundation for project.
- b) A project charter is a document which explains about the project at a very high level highlights the stakeholders and the approach towards project.
- c) It introduces the project manager to the project.
- d) The project sponsor owns a project charter.
- e) The project sponsor authorizes the project charter, but the project manager can create it.
- f) Project stakeholder should approve every project charter once the project charter is approved it cannot be changed throughout the project lifecycle.

d) WBS Dictionary:

- a) The work breakdown structure (WBS) dictionary is where the details of task activities & deliverables of the WBS are located.
- b) The content include whatever the milestones are related, the project scope & in some instance dates, resources, cost & quantity.
- c) The WBS dictionary allows you to define each of steps on the WBS flow to execute them to reach

- the final deliverable of the project.
- d) The document includes a work package which defines a related task & control accounts, which integrate the scope, budget, actual cost & schedule for these tasks.

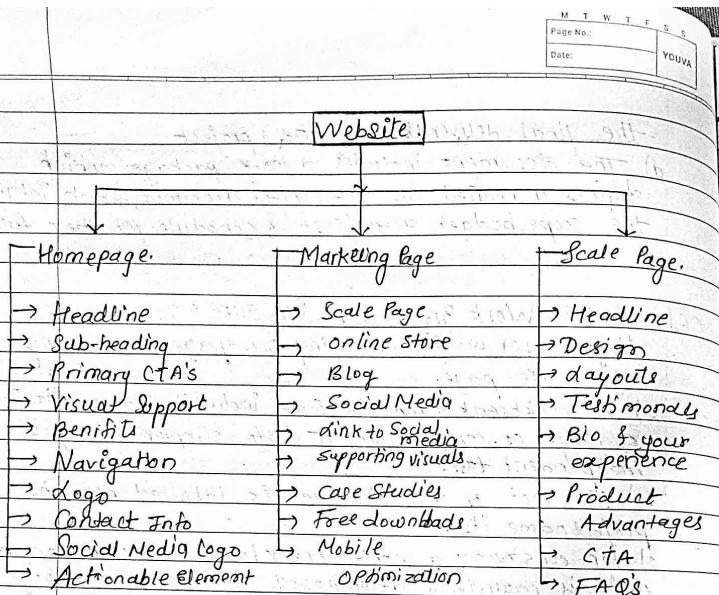
(Q6) What is Work Breakdown Structure? Draw WBS diagram for website showing homepage, marketing page, sales page.

- a) A workbreakdown structure includes dividing a large or complex project into simpler, manageable & independent tasks.
- b) The root of this structure is labelled by the project name itself.
- c) for constructing a work breakdown structure, each node is recursively decomposed into smaller sub-activities, until at the leaf level, the activities become undividable & independent.
- d) It follows a top down approach.

Step 1: Identify the major activities of the project.

Step 2: Identify the subactivity of the major activities

Step 3: Repeat till undividable simple & independent activities are created.



- Q7.1** Describe requirement gathering techniques like:
- ① Study of existing document
 - ② Interview
 - ③ Task Analysis
- Scenario
 - Form Analysis
- With help of example.
- • Requirement Gathering :→ Transportation Agency.
- Objective :→ To collect requirements from owner of transportation agency.

1) Study of Existing Documents :

- **Form Registers :**→
 - The registers hold the info about -
 - How many transportation vehicles are available?
 - How many vehicles are currently on routes?
 - On which vehicles how many items are loaded of specific comp?
- **Problems :**→
 - All info on physical med & gets damaged or unusable after some time.
 - Its hard to fill data of specific delivery with relation to each other.
 - Its difficult to find the details of specific delivery of old time. history of transportation is not maintained properly.

2) Interview :

- **Categories :**→
 - Revenue management
 - Management of vehicles
 - Relation management with customer.
- **Revenue Management :**→
 - Calculation of expenses done on maintenance of vehicle & expenses on petrol or diesel.
 - Casting of charges paid by customer according to travelling km, size of object, weight & import or export or taxes.
 - Monthly Salary payments & auditing of all expenses or taxes.

- Management of Vehicles :-
- Data about total number of vehicles on route vehicles & vehicles in parking.
- Need of categorizing the vehicle according to their capacity to transport load & their permits.

- Relation Management with Customer :-
- If there is any issue during transport the customer have to face unwanted delay.
- To inform the customer, Notification should be sent if any issue.
- Customer can't have any system where they can track real time position of their load. Tracking system which can see by customer begin.

3] Task Analysis :-

• Blackbox Testing :-

- Validating the input by communicating with owners.
- testing the outputs for specific inputs.
- checking that the storage of data is done properly on cloud.
- checking the sorting methods & ease to access data.
- check of notification system
- check of tracking system.

• from these get the further requirements & feedback from the owner.

A] Scenario Analysis :-

- ① Issue during travelling: Tire puncture, engine problem
- ② There should be system that the notification of delay can be sent to customer by driver.
- ③ The details about issue got noted & solved.

- ④ If the worker is successively absent the salary will be deducted. The salary should be deducted from monthly salary of worker & audit is maintain according to it.

B] Form Analysis :-

- Automation of vehicle categorization according to their payload & permits.
- The data storage should be replaced by cloud storage.
- The GPS input should be provided to tracking system.

(1)
21/07/23

ASSIGNMENT-3.

Q1) What is Data flow diagram? Draw a DFD of a system that pays worker & Explain the pay worker system.

→ • Data flow diagram.

① A data flow diagram (DFD) is a fractional visual representation of the information flows within a system.

② It shows how data enters & leaves the system, what changes the information & where the data is stored.

③ The objective of DFD is to show the scope & boundaries of the system as whole.

④ It may be used as communication tool between a system analyst & any person who play a part in the order that acts as a starting point for redesigning the system.

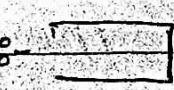
⑤ The DFD is also called as data flow graph or bubble chart.

• DFD symbols :-

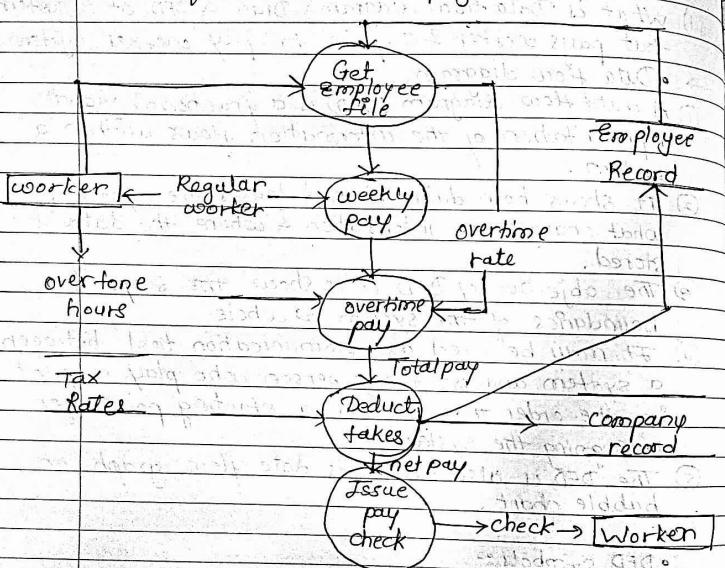
Square:-  defines source / destination of data.

Arrow:-  Identifies data flow.

Circle:-  Represents a process that transfers data flow into outgoing data.

Open Rectangle :-  It's a data store.

- DFD of a system that pays worker.

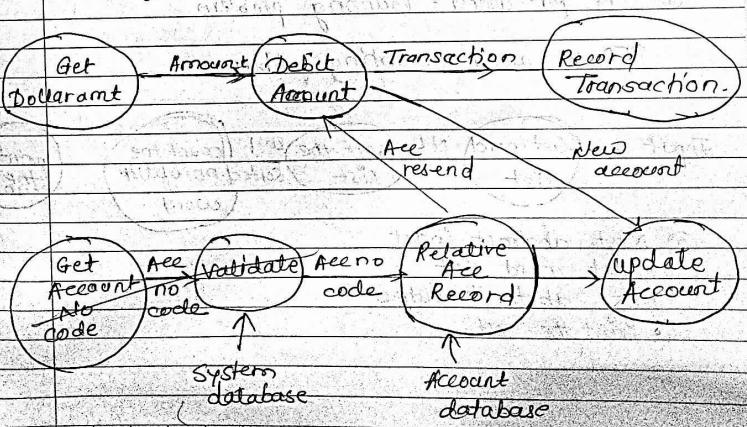


- DFD system that pay workers is shown in the above diagram.
- In this DFD, there is one basic input dataflow the weekly time sheet, which originates from the source code.
- The basic output is paycheck, the risk for which is also the worker.

- In this system the workers, the employees record is retrieved using the employees Id, which is contained in the time-sheets.
- From the employee record, the rate of payment overtime are obtained.
- These rates are the regular & overtime amount hrs, which are used to complete the payments.
- After total payment is determined taxes are deducted.
- The amount of tax deducted is recorded in the employee & company records.
- Finally the pay check is issued for the net pay the amount paid is also recorded in company records.

Q2) Give structured design methodology for ATM.

① DFD of an ATM



M T W T F S S
Page No.:
Date: YOUVA

M T W T F S S
Page No.:
Date: YOUVA

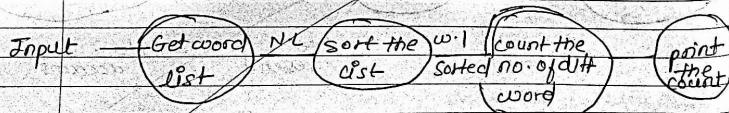
- (2) Most abstract input
 (i) Get Dollar Amount
 (ii) Get Account Number
 (iii) Record Transaction
 (iv) Update acc.
- Most abstract output
 (i) Record Transaction
 (ii) Print the last.
- DFD for transaction problem
- ```

 graph TD
 subgraph Main [Main]
 direction TB
 A[Get Acc Number] --> B[Get Dollar Amount]
 B --> C[Reactive Acc Record]
 C --> D[Debit Amnt]
 D --> E[Update Amnt]
 E --> F[Record Transaction]
 end
 G[Sort no. of diff word] --- H[Get sorted list]
 H --- I[Count no. of diff word]
 I --- J[Print the last]

```
- |                |                   |                     |            |             |                    |
|----------------|-------------------|---------------------|------------|-------------|--------------------|
| Get Acc Number | Get Dollar Amount | Reactive Acc Record | Debit Amnt | Update Amnt | Record Transaction |
|----------------|-------------------|---------------------|------------|-------------|--------------------|

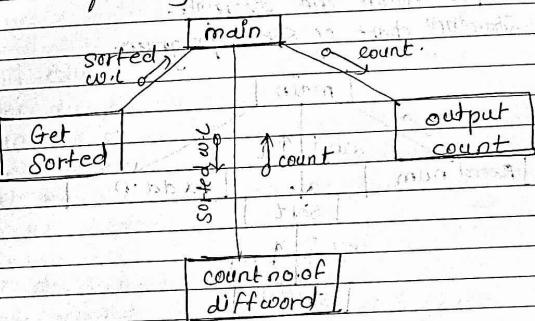
- (3) Give SDM for problem statement  
 "Determine the different no. of words in an input file."
- (4) SDN for word - counting problem

DFD for word - counting problem

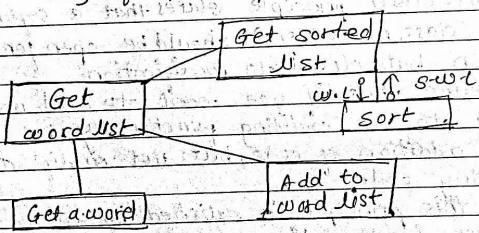


- (5) Most abstract input  
 • Get word list  
 Most abstract output  
 • Print the last.

- (3) first level factoring

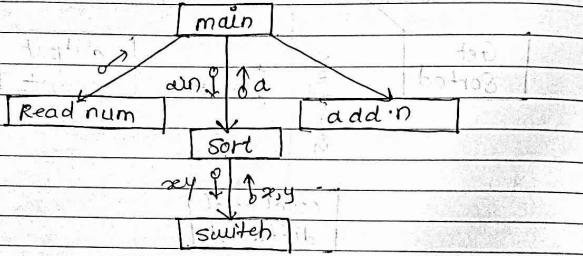


- (4) factoring of input, output, transformation module.



- Q4) Draw Structure chart for sort program & state open principle with the example.

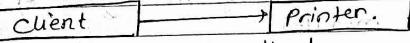
- Structure chart of sort program:



open closed principle

- ① The open-closed principle states that a software module (class, method, etc) should be open for extension but closed to modification.
- ② This means that if you want to add new features to an existing functionality, i.e. allows addition of code, but not modification of existing code.
- ③ In so this principle is satisfied by using inheritance & polymorphism.
- ④ Inheritance allows creating a new class to extend behaviour without changing the original class.
- ⑤ This can be used to support the open-closed principle.

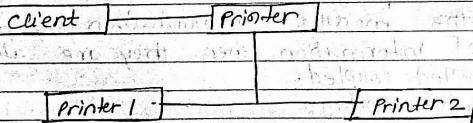
- consider example of a client object which interacts with a printer object for printer.



- client directly calls method on printer.
- If another is to be allowed
  - A new class pointer is to be allowed
  - But the client will have to be changed if it wants to use printer.

- Alternate Approach

- Make printer of a subclass of a general printer
- for modification add another subclass printer
- client doesn't need to be changed.



(Q2) What are different types of coupling & cohesion?

→ (i) Coupling :-

It is the measure of the degree of interdependence bet' the modules.

Types of coupling:-

(1) Data coupling :-

If the dependency bet' the modules is based on the fact that they communicate by passing only data then the module are said to be data coupled.

(2) Stamp coupling :-

In stamp coupling the complete structure is passed from one module to another.

(3) Control coupling :-

If the module communicate by passing control information then they are said to be controlled coupled.

(4) External coupling :-

The modules depends on other modules, external to the software being developed or to a particular type of hardware.

e.g. protocol, external file, device format, etc.

(5) Common coupling :-

The module name shared data such as global data structure.

The changes in global data mean racing back to all modules which access data to evaluate the effect of the change.

So it has got disadvantages like difficulty in reusing module, reduced ability to control data

data access & reduced maintainability.

(6) Content coupling :-

In a content coupling, one module can modify the data of another module or control flow is passed from one module to the other module.

(7) Cohesion :-

Cohesion is a measure of the degree to which the elements of the module are functionally related.

(8) It is the degree to which all elements directed towards performing a single task are contained in the component.

(9) Basically cohesion is the external that keep the module together.

(10) Type of cohesion :-

(i) Functional cohesion :-

In functional cohesion the parts of the module are grouped together.

so they are united for a single well defined purpose.

(ii) Sequential cohesion :-

A module is said to be sequential cohesion if element of a module from the component of the sequences where the output from one component of the sequence, is input to the next.

(iii) Communicational cohesion :-

A module is said to have communicational cohesion if all tasks of module refers to or update same data structure.

(4) Procedural cohesion:-

① A module is said to be procedural cohesion if the set of purpose of the modules are all parts of a procedure in which particular sequence of steps has to be carried out for achieving goal.

(5) Temporal cohesion:-  
when a module include functions that are associated by the fact all method must be executed in same time the module is said to exhibit temporal cohesion.

(6) Logical cohesion:-

- ① the ele are logically related and not functionally
- ② All the code for functions in the same component
- ③ operations are related but the functional are significantly different.

(7) Co-incidential cohesion:-  
The ele have no conceptual relationship other than location in source code.  
eg - print next line & resource the char of string in single component.

(C) 05/04/23

05/04/23