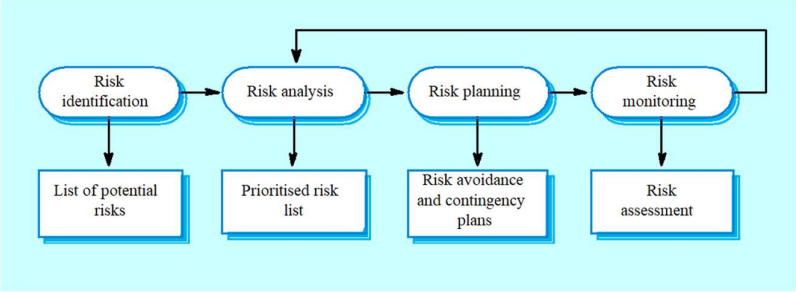
1. Assume yourself as a team leader of an ERP solution developing project for a textile company. In general, perform risk management by analysing the various types of risks that may occur in your project and categories the risks. Elaborate the risk solutions for the same.

Risk management

- Risk management is concerned with identifying risks and drawing up plans to minimise their effect on a project.
- A risk is a probability in that some adverse circumstance will occur in a project.
  - Project risks affect schedule or resources;
  - Product risks affect the quality or performance of the software being developed;
  - Business risks affect the organisation developing or procuring the software.

The risk management process



Then according to the given scenario at least five risks should be identified and according to its probability and effects it should be prioritized and then the strategy to solve the identified risk should be given. Finally risk monitoring activities should be discussed to understand whether it is less or more probable in future.

2. You have been appointed as a purchase officer of the beauty soap manufacturing company and your job is to maintain the production plant in such a way that there is no interruption in the production process for 24x7. In order to increase the production rate to match the demand in the market, your company decided to procure an equipment for its production plant. As a purchase officer you have to undergo a series of process to complete the equipment procurement process as per the company policies. The company policies can be assumed as per your wish. Demonstrate the data flow diagram in all the three levels with proper precisions.

DFD level 0 – 3 marks, DFD level 1 – 3 marks and DFD level 2 – 4 marks

Level 0 -3 Marks

Abstract view of the system,

Level -1 -3 Marks

Interaction between the modules

Level -2 – 4 Marks

Interaction within the modules and interaction between the modules.

3. Construct a structured design to provide convenient transition from data flow diagram to software architecture. Follow all the recommended steps for the architectural mapping in order to provide an appropriate program structure for the above-mentioned scenario. (refer Q.no.2)

Architectural Mapping

- Structured design provides a convenient transition from a data flow diagram to software Architecture

Types of information flow

The 2 different types of information flows:

1. **Transaction flow** - a single data item triggers information flow along one of many paths
2. **Transform flow**

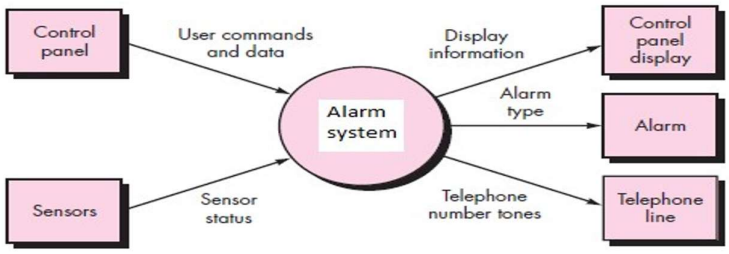
- overall data flow is sequential and flows along on a small number of straight-line paths
  - ***Incoming Flow:*** The paths that transform the external data into an internal form
  - ***Transform Center:*** The incoming data are passed through a transform center and begin to move along paths that lead it out of the software
  - ***Outgoing Flow:*** The paths that move the data out of the software

Steps Involved

- **Step 1.** Review the fundamental system model.
- **Step 2.** Review and refine data flow diagrams for the software.
- **Step 3.** Determine whether DFD has transform or transaction flow characteristics
- **Step 4.** Isolate the transform center by specifying incoming and outgoing flow boundaries
- **Step 5.** Perform "first-level factoring"
- **Step 6.** Perform "second-level factoring"
- **Step 7.** Refine the first iteration program structure using design heuristic

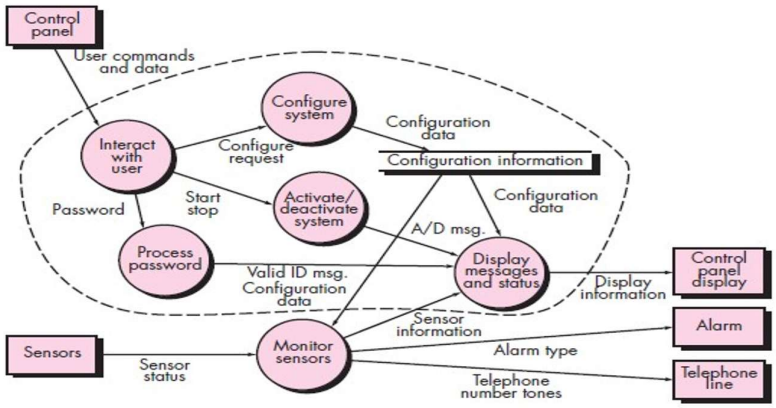
For an example:-

Step 1. Review the fundamental system model.



SAFE HOME PROJECT

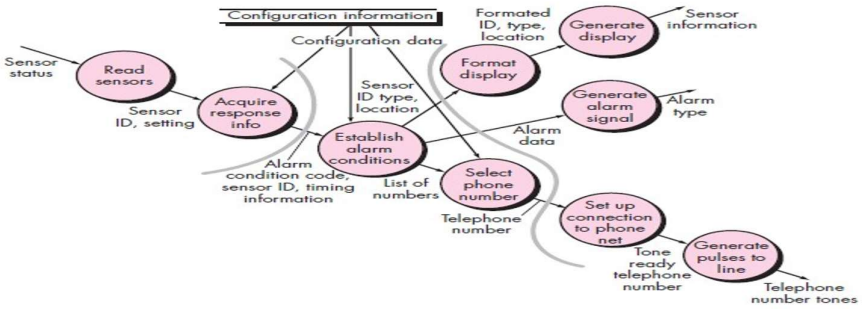
Step 2. Review and refine data flow diagrams for the software.



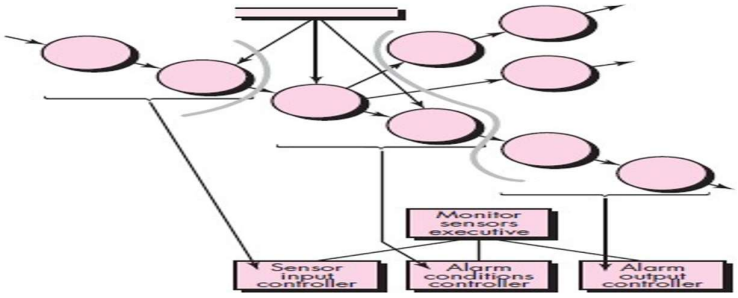
Step 3. Determine whether DFD has transform or transaction flow characteristics.

- in general---transform flow
- special case---transaction flow

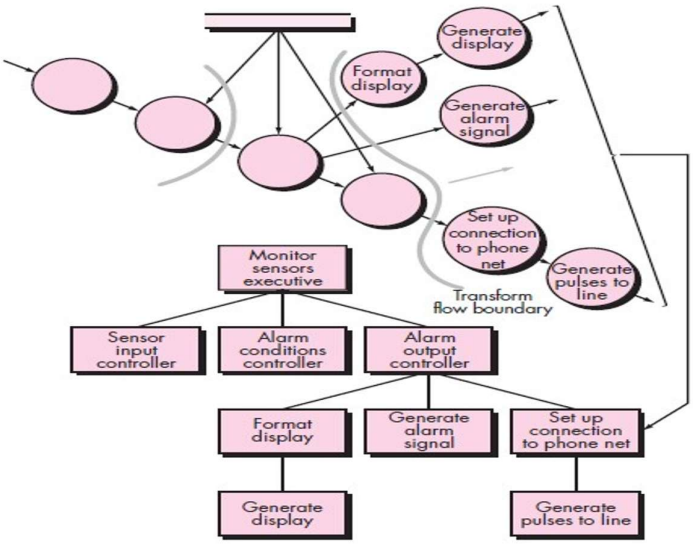
Step 4. Isolate the transform center by specifying incoming and outgoing flow boundaries



Step 5. Perform "first-level factoring"

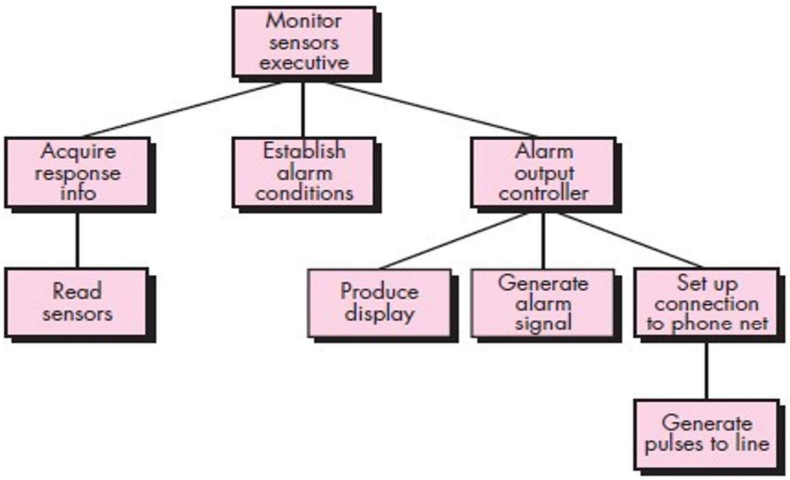


Step 6. Perform "second-level factoring"



24

Step 7: Refine the first iteration program structure using design heuristics



25

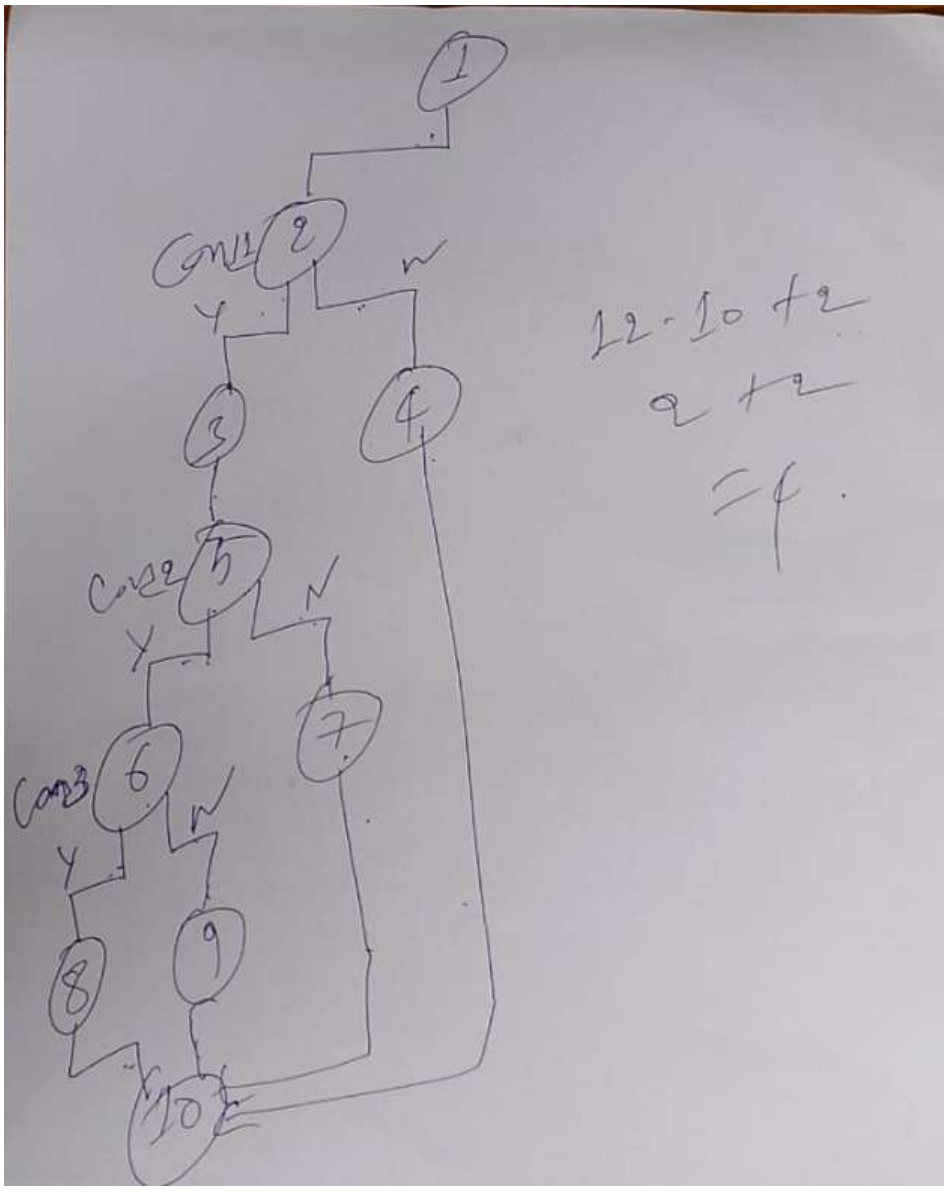
4. You are provided with a simplified JavaScript function for a web-based application that calculates and displays the yearly subscription cost for a user based on the number of users and the chosen

subscription plan. The application offers two plans: 'basic' at $100 per user per year and 'premium' at $150 per year. A discount is also applied for more than 100 users: a 10% discount for the 'basic' plan and a 15% discount for the 'premium' plan.

```
function calculateYearlySubscriptionCost(numberOfUsers, planType)
{
  let costPerUser = planType = 'basic' ? 100: 150;
  let discount = numberOfUsers > 100 ? (planType = 'basic' ? 0.1: 0.15) : 0;
  return numberOfUsers * costPerUser * (1 - discount);
 }
```

a) Calculate the cyclomatic complexity for the above-mentioned routine appropriately.

$$CC = E - N + 2$$



b) Design the test cases in such a way that it covers all the paths of the given routine. Every testcase should brief the rationale to specify its necessity.

1. Test with the 'basic' planType more than 100 users
2. Test with the 'basic' planType less than 100 users
3. Test with the 'premium' planType more than 100 users
4. Test with the 'premium' planType less than 100 users

Totally, 4 test cases to achieve full branch coverage.

5. Consider a mobile application for understanding the various events happening in our university. This application should list the events in different ways based on event date, event hosting department and event category (both technical and non-technical). It should support event registration, cancellation, payment and rating. For the given scenario develop the following

a) A technical diagram to show the attributes and methods of every entity involved.
Class diagram should be drawn with different type of attributes, methods, aggregation, composition and cardinality ratio.

b) A technical diagram to show the object interactions.
Sequence diagram:
Object interactions, looping and numbering should be shown properly.