**Table of contents:**

|  |  |  |
| --- | --- | --- |
| **Chapter Number** | **Title** | **Page Number** |
| 1. | Introduction | 3 |
| 2. | Problem Statement | 3 |
| i. Problem description | 4 |
| 3. | Requirements | 4 |
| i. High level requirements | 5 |
| ii. Low-level requirements | 6 |
| 4. | SWOT Analyse | 7 |
| 5. | 4 W’s and 1H | 8 |

|  |  |  |
| --- | --- | --- |
| 6. | Architecture layout | 9 |
| 7. | Design | 9 |
| 8. | Structural design | 9 |
| 9. | Behavioural design | 10 |
| 1. Flowcharts |
| 1. Use case diagram |
| 10. | Results and conclusion | 12 |
| 11. | References | 12 |

**1.** **Introduction:**

A project is composed of various different domains. Each domain has a set of functions to be carried out. In a project, all the resources available must be made available to all the domains and must be shared in a balanced way. Any imbalance in the resource distribution can cause delay in the completion of work leading to maximised cost. This problem addresses optimization of this kind of scenario.

**2.** **Problem statement:**

There is a cluster of computers. Each computer is referred as a node. You are the maintainer of the cluster. Your job is to execute various jobs that you receive for execution. You can transfer a job from a node on the cluster to some other node. It will require some network cost.

The jobs come in the batches. Each job will have its desired initial node on which it was thought to be run by its programmer. It will also require some processing power to run in a node. You have to process the jobs of the batch, i.e., you have to decide on which node the job should be executed. You can transfer the job through various nodes before assigning the final node.

Being the cluster maintainer, your aim is to optimally use the resources of the cluster. You want the load on the nodes of the cluster to be balanced. Also, you don't want to spend a huge networking cost in transferring the jobs in between the nodes.

2.1 **Problem description**:

To optimally use the resources of the cluster and want the load on the nodes of the cluster to be balanced and don't want to spend a huge networking cost in transferring the jobs in between the nodes.

**2.2** **Procedure:**

2.2.1 Identify the problem:

To optimally use the resources of the cluster and want the load on the nodes of the cluster to be balanced and don't want to spend a huge networking cost in transferring the jobs in between the nodes.

2.2.2 Understand the problem:

To optimally use the resources of the cluster and want the load on the nodes of the cluster to be balanced and don't want to spend a huge networking cost in transferring the jobs in between the nodes.

**3.** **Requirements:**

Basically, requirements are statements that indicate what a system needs to do in order to provide a capability (i.e., utility or benefit.) Requirements are generally prepared during the early stages of a project's system development life cycle (SDLC.) There are many different SDLC methodologies that are used in practice. [1]

**3.1** **High level requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Description** | **Category** | **Status** |
| HL\_01 | * Read number of computers in a cluster | Technical | Implemented |
| HL\_02 | * Read elements in the form of a matrix | Technical | Implemented |
| HL\_03 | * Able to identify the next computer (node). | Technical | Implemented |
| HL\_04 | * Display the minimum cost of the matrix | Technical | Implemented |
| HL\_05 | * Display the path | Technical | Implemented |

**3.2** **Low level requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Description | HLR\_id | Status |
| LLR\_01 | * To identify the initial computer to start with among a cluster of computers. | HL\_01 | Implemented |
| LLR\_02 | * Choose the next computer for load enhancement. | HL\_02 | Implemented |
| LLR\_03 | * Evaluate the load from initial computer to present computer. | HL\_03 | Implemented |
| LLR\_04 | * Accordingly select the next computer for load allotment. | HL\_04 | Implemented |
| LLR\_05 | * Reading the number of clusters and reading the elements of the matrix is manual and done by the user. | HL\_05 | Implemented |

**4.** **SWOT analysis:**

|  |  |
| --- | --- |
| * Strengths:  1. Efficient load distribution to each computer in the cluster. 2. Reduced response time. 3. Optimized usage of resources present. 4. Reduced networking cost | * Opportunities:  1. Displaying the cluster in matrix form. 2. Load analyses. 3. Better time management. 4. Best path allocation. |
| * Weaknesses:   1) Not suitable for a large cluster of computers.  2) Execution time dependency on the number of systems in a cluster.  3) Needs user inputs.  4) Involves Matrix calculations. | * Threat:  1. Manual load allocation on systems. |

**5.** **4 W's and 1 H:**

● Where is it used?

In system work allocation systems using resources.

● Why is is used?

To find the best way to allocate the load among computers using resources available.

● Which is the best available solution for this particular problem?

Travelling sales men approach.

●Who is going to be benefited?

Manager tries to allocate the load.

● How does it work?

It works on the principle of finding the shortest path between initial node and final node.

**Architecture:**

The contents of Architecture include:

* Design
* Structural design
* Behavioural design

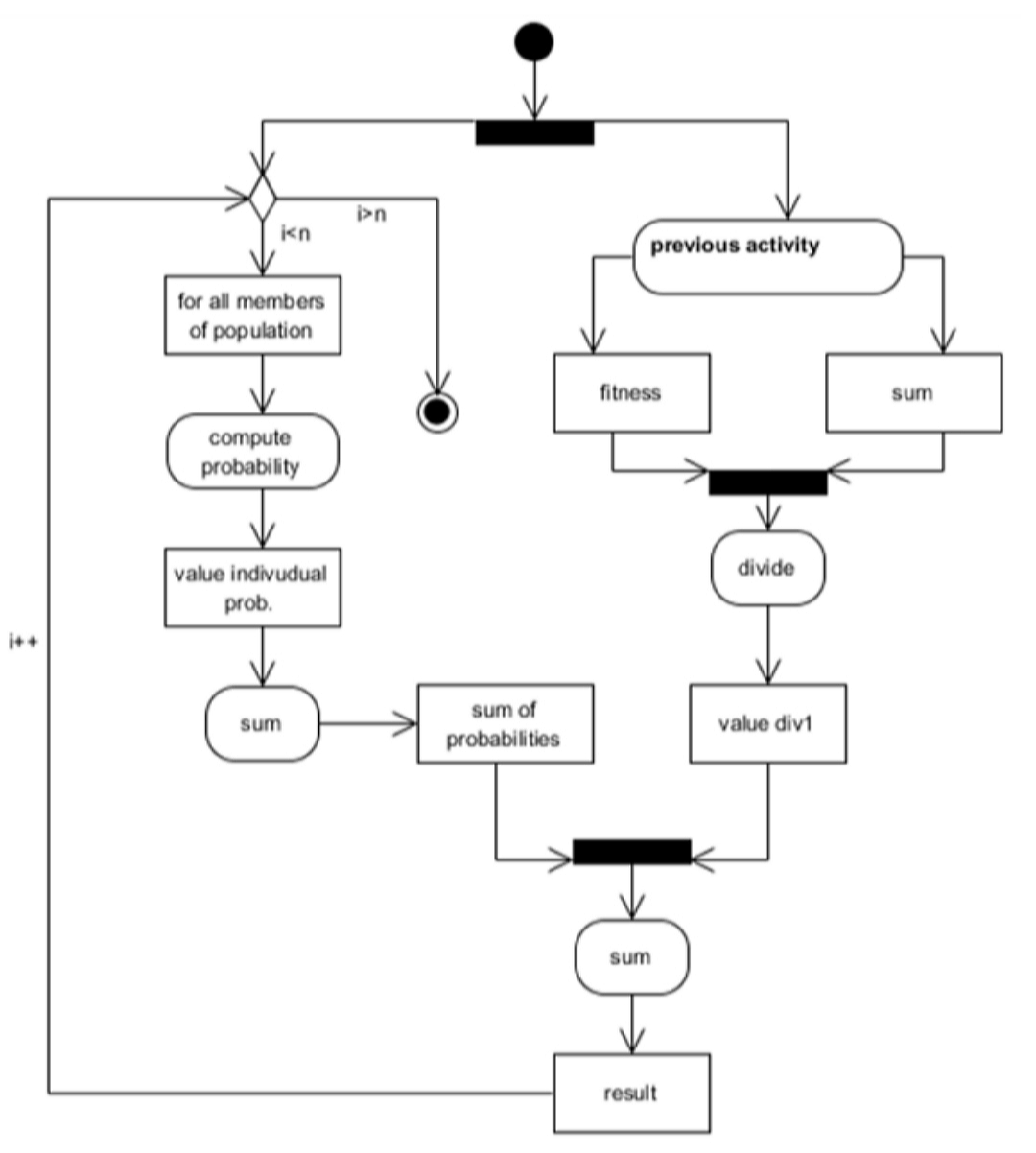
**Design:**

There are two types of designs structural and behavioural.

* + 1. Structural Diagrams

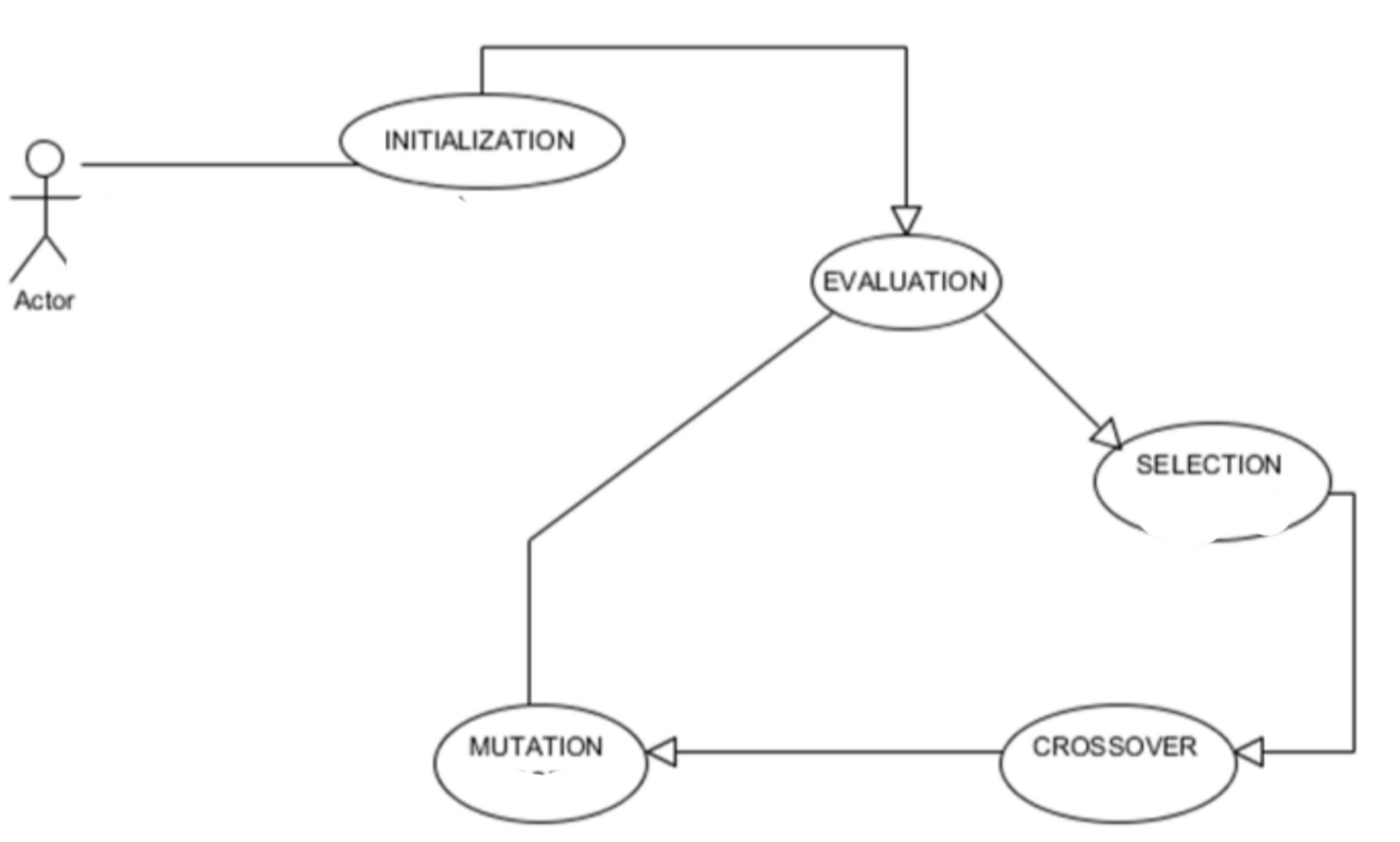
ii) Behavioural Diagrams:

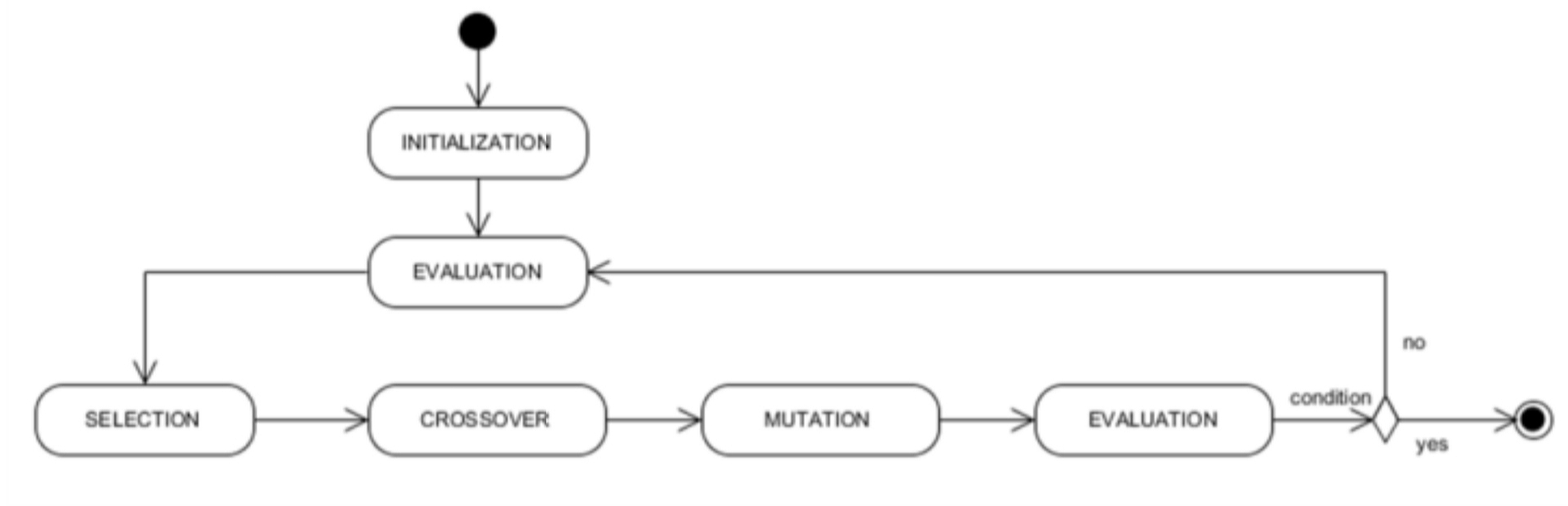
**1) Activity Diagram**



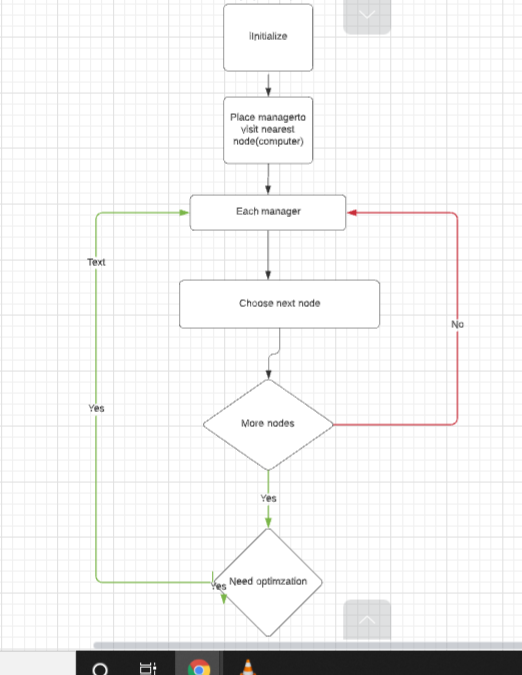
**Use case diagram:**

**i)**



**ii)**

**Flowchart:**



**Results and conclusion:**

The objective of the problem statements has been achieved. Using the traveling sales man approach the task of assigning the load to the cluster of computer made easy. The resources present are used efficiently to find best path.

**References:**

1] Solution approach: <https://www.thecrazyprogrammer.com/2017/05/TSP.html>

2] PDLC: <https://stayrelevant.globant.com/en/pdlc-best-practices/>