**MINI-PROJECT REPORT**

**ROCK PAPERSCISSORS**



**NAME: -** J NIKITHA

**PS NO:** 40020747

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| SERIAL No. | CONTENT | PAGE No. |
| 1 | Introduction | 3 |
| 2 | Objective | 3 |
| 3 | Features | 3 |
| 4 | Requirements | 3 |
| 5 | SWOT Analysis | 4 |
| 6 | State of Art | 5 |
| 7 | 4W and 1H | 5 |

|  |  |  |
| --- | --- | --- |
| 8 | High Level Requirements and Low-Level Requirements | 6 |

|  |  |  |
| --- | --- | --- |
| 9 | Architecture | 7 |
| 10 | Test Plan | 10 |
| 11 | Expected Outputs | 11 |

## **Introduction:**

This game have three categories one is rock(R), paper(P) and scissor(S). One choice is taken by the player and another is taken by the system such that the different choices and one with more powerful choice wins and if selected equal choice draw. Below mentioned are the

## **Objective:**

Rock-Paper-Scissors (referred to in different orders depending on region) is the most basic idea of the concept of attributable strength and weakness.

Essentially, it sets up an unbreakable loop where every entry defeats another, and is in turn defeated by yet another.

Rock beats Scissors. Scissors beats Paper. Paper beats Rock. If players choose the same weapon, neither win and the game is played again.

## **Features:**

In this rockepaperscissors game you can store the username, view the highest score secured by a user, and even reset the score. Additionally, to make the game look a little more interesting, it is divided into rounds; user must pass the first round to reach the second one.

## **Requirements**

1 VS code

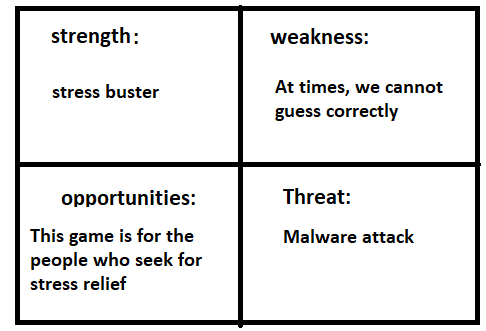
2 GitHub as version control

3 GCC in windows as well as Linux

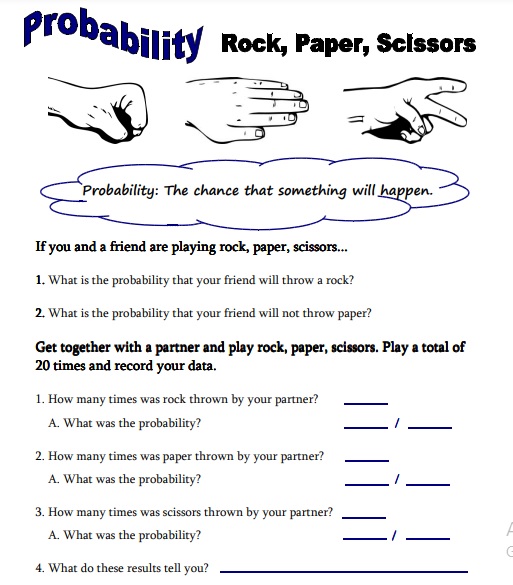
4 WSL/Dual boot/VMware

5 Mingw

# **SWOT Analysis:**



# **State of Art:**



# **4W and 1H:**

## **Who**

-One can be used as a fun game it can be upgraded to have more features.

## **What**

-This project idea is to play with computer.

## **Where**

-This project idea is available both on VSC and GitHub.

## **Why**

-this project uses simple function it is very compact to work.

## **How**

-By implementing multilevel program, it can be achieved -this project requires a lost cost of development.

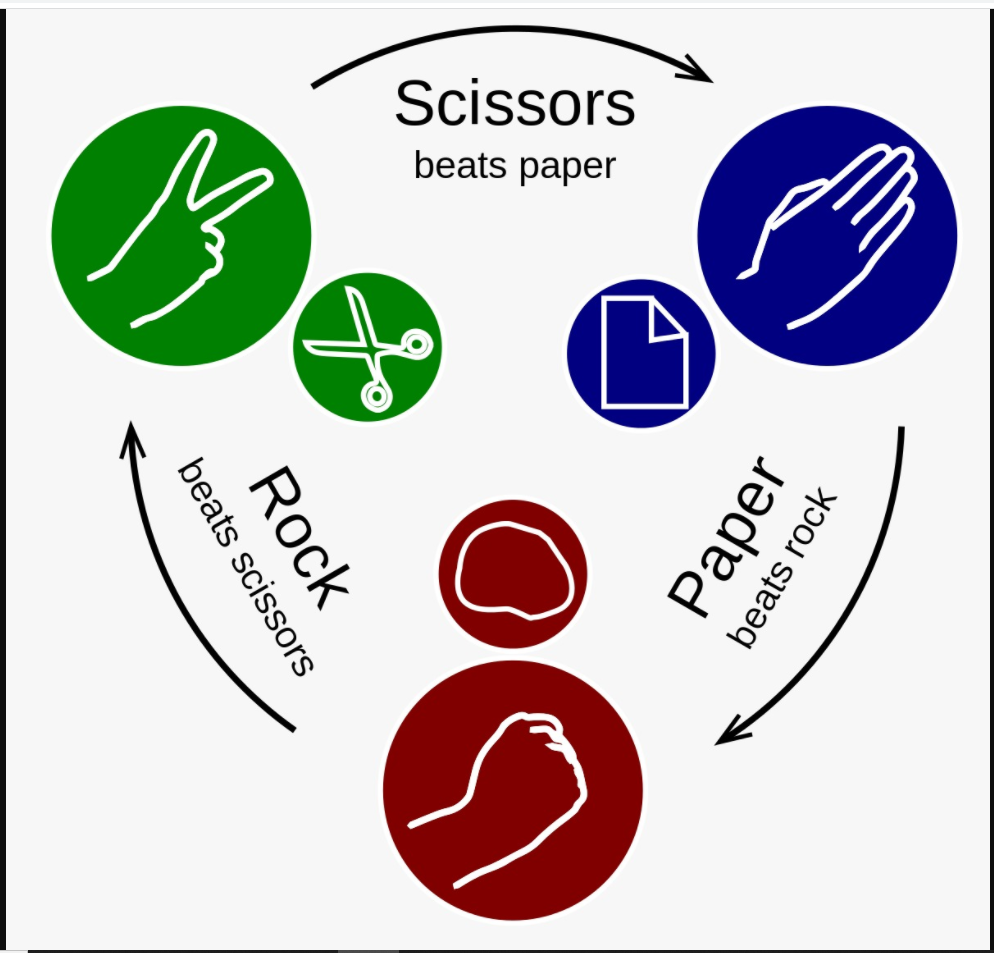
## **High Level Requirements:**

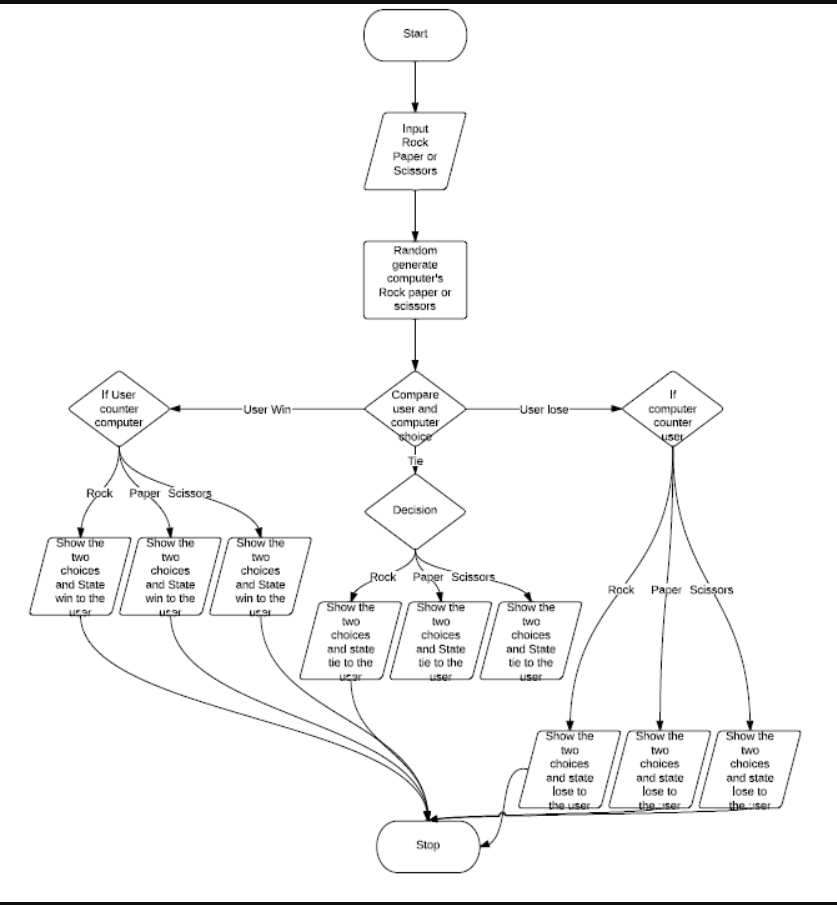
This project can be implemented by using c language in both windows and Linux os. This project requires a random integer generator function.

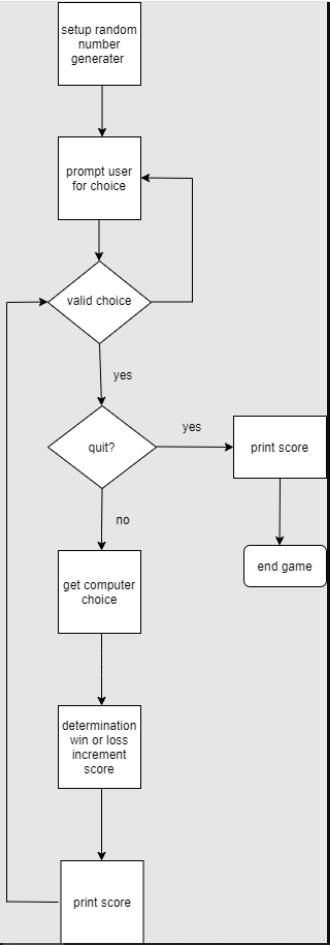
## **Low level Requirements:**

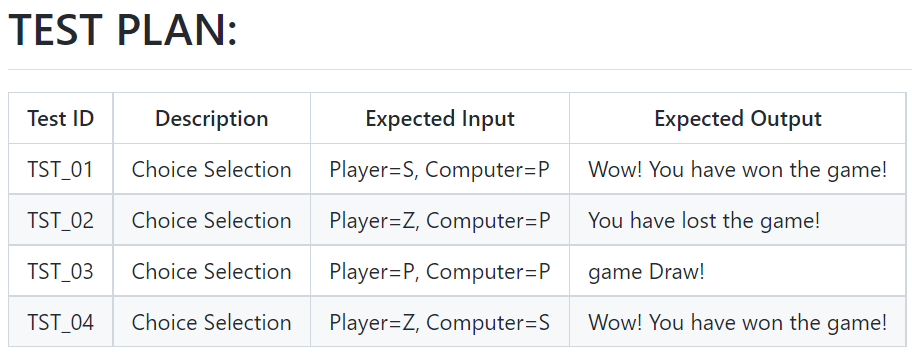
The Random number has to generate the numbers with in the specified range of number. The comparison of the character from the player input and the computer generated has to be taken care.

**Architecture:**









**Expected Output:**

