./

Learning Report – Applied System Development Life Cycle and Software Testing



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be approved By** | **Remarks/Revision Details** |
| 1 |  | Name/PS No | Name/PS No | Module Owner Name | Comments |
| 2 | 15/02/21 | Sayani Basak | 99003772 |  |  |
| 3 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

Table of Contents

[Table of Figures 3](#_Toc53129062)

[Table of Tables 4](#_Toc53129063)

[ACTIVITY 1: SYSTEM/ SOFTWARE DEVELOPMENT 4](#_Toc53129064)

[**INTRODUCTION** 4](#_Toc53129065)

[**MY PRODUCT: “Name ”** 6](#_Toc53129066)

[**SWOT ANALYSIS** 7](#_Toc53129067)

[**REQUIREMENTS** 7](#_Toc53129068)

[**DESIGN** 8](#_Toc53129069)

[HIGH LEVEL DESIGN 8](#_Toc53129070)

[LOW LEVEL DESIGN 11](#_Toc53129071)

[**TEST PLANS** 13](#_Toc53129072)

[**REFERENCES** 15](#_Toc53129073)

[ACTIVITY 2: AGILE METHODOLOGY 15](#_Toc53129074)

[**THEME** 15](#_Toc53129075)

[**EPIC** 15](#_Toc53129076)

[**USER STORY** 16](#_Toc53129077)

[**REFERENCES** 17](#_Toc53129078)

[APPENDIX: 17](#_Toc53129079)

## Table of Figures

[Figure 1 CLASS DIAGRAM(HIGH LEVEL) 10](#_Toc52177314)

[Figure 2 USE CASE DIAGRAM (HIGH LEVEL) 11](#_Toc52177315)

[Figure 3 ACTIVITY DIAGRAM (HIGH LEVEL) 12](#_Toc52177316)

[Figure 4 USE CASE DIAGRAM (LOW LEVEL) 12](#_Toc52177317)

[Figure 5 ACTVITY DIAGRAM (LOW LEVEL) 13](#_Toc52177318)

[Figure 6 BLOCK DIAGRAM 13](#_Toc52177319)

[Figure 7 COMPONENT DIAGRAM (HIGH LEVEL) 22](#_Toc52177320)

[Figure 8 ACTIVITY DIAGRAM (high level) 23](#_Toc52177321)

[Figure 9 ACTIVITY DIAGRAM (LOW LEVEL) 24](https://lnttsgroup.sharepoint.com/sites/GEA/Global%20Engineering%20Academy/GEA%20Insights/Genesis/Shared%20Documents/Submission/MYSORE/2009MYSEMB/Foundation/Applied%20SDLC%20with%20Software%20Testing/99002439/FINAL.docx#_Toc52177322)

[Figure 10- ACTIVITY DIAGRAM (LOW LEVEL) 24](#_Toc52177323)

[Figure 11 TEST PLAN 25](#_Toc52177324)

[Figure 12 GIT 27](#_Toc52177325)

[Figure 13 GIT ISSUES 28](#_Toc52177326)

[Figure 14 GIT COMMITS 1 28](#_Toc52177327)

[Figure 15 GIT COMMIT 2 29](#_Toc52177328)

[Figure 16 GIT 30](#_Toc52177329)

[Figure 17 GIT MAKE 31](#_Toc52177330)

[Figure 18 GIT MAKE 2 31](#_Toc52177331)

[Figure 19 GIT BUILD 32](#_Toc52177332)

[Figure 20 GIT CODE QUALITY 32](#_Toc52177333)

## Table of Tables

[Table 1 AGING 6](#_Toc52177304)

[Table 2 GRADING COST 6](#_Toc52177305)

[Table 3 REQUIREMENTS 8](#_Toc52177306)

[Table 4 HIGH LEVEL TEST PLAN 15](#_Toc52177307)

[Table 5 LOW LEVEL TEST PLAN 16](#_Toc52177308)

[Table 6 USER STORIES 17](#_Toc52177309)

[Table 7 AGING 19](#_Toc52177310)

[Table 8 GRADING COST 19](#_Toc52177311)

[Table 9 REQUIREMENTS 21](#_Toc52177312)

[Table 10 USER STORIES 27](#_Toc52177313)

[ACTIVITY 1: SYSTEM/ SOFTWARE DEVELOPMENT](#_Toc53129064) :

INTRODUCTION:

This is a 14-digit smart scientific as well as professional calculator. Calculator is a scientific device which is used to perform various calculations like arithmetic, logical, trigonometry etc. This device is very easy to use and it is also used by many people including students. In our calculator, our team has introduced few extra features like finding volume of cone, sphere, area of square, permutation and combination etc.

MY PRODUCT: “SMART CALCULATOR”.

Research:

1. Simple Calculator: It is a basic calculator with arithmetic functions like addition, subtraction, multiplication, division.

2. Scientific Calculator: In this calculator math’s operations like trigonometry, matrix, permutation and combination etc. can be performed.

Cost and Features:

In this calculator, there will be an option of dedicated MRC which is Memory Recall and Clear as well as dedicated check keys. It will be operated from two sources which is solar power and button cell. It also includes a space for printing receipts. User can use this calculator for high level math’s functions like finding area, volume, pnc and statics problem as well. Multiview display will be also there which can help user to view several calculations on the screen at one time.

By keeping the type of customer in mind we have set the base price for this calculator to Rs 3000 to Rs 4000 depending on the product type.

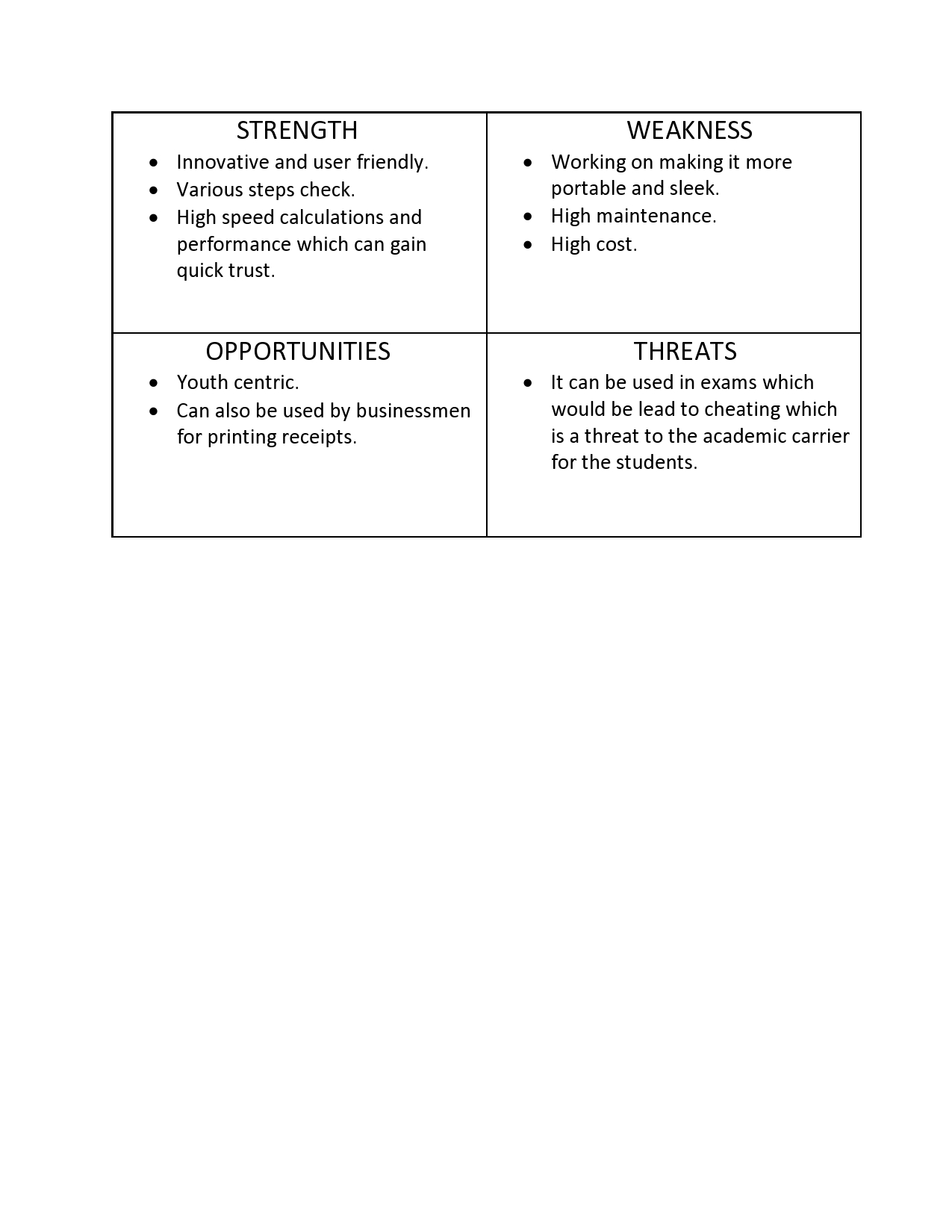


! [image](https://user-images.githubusercontent.com/78853952/107922921-b9540180-6f96-11eb-9d30-8191fb02780d.png)

## Defining Our System

1. Power ON/OFF switch.
2. Dedicated MRC switch.
3. High speed operations.
4. Space given for printing receipts.
5. 14-digit display.
6. Operated on both solar as well as battery.
7. High level math’s and science functionality.
8. Multilevel display.

SWOT ANALYSIS:



# 4W's and 1'H

## Who:

**Student and businessman.**

## What:

**Smart Scientific Calculator**

## When:

**For fast and effective way to complete calculations.**

## Where:

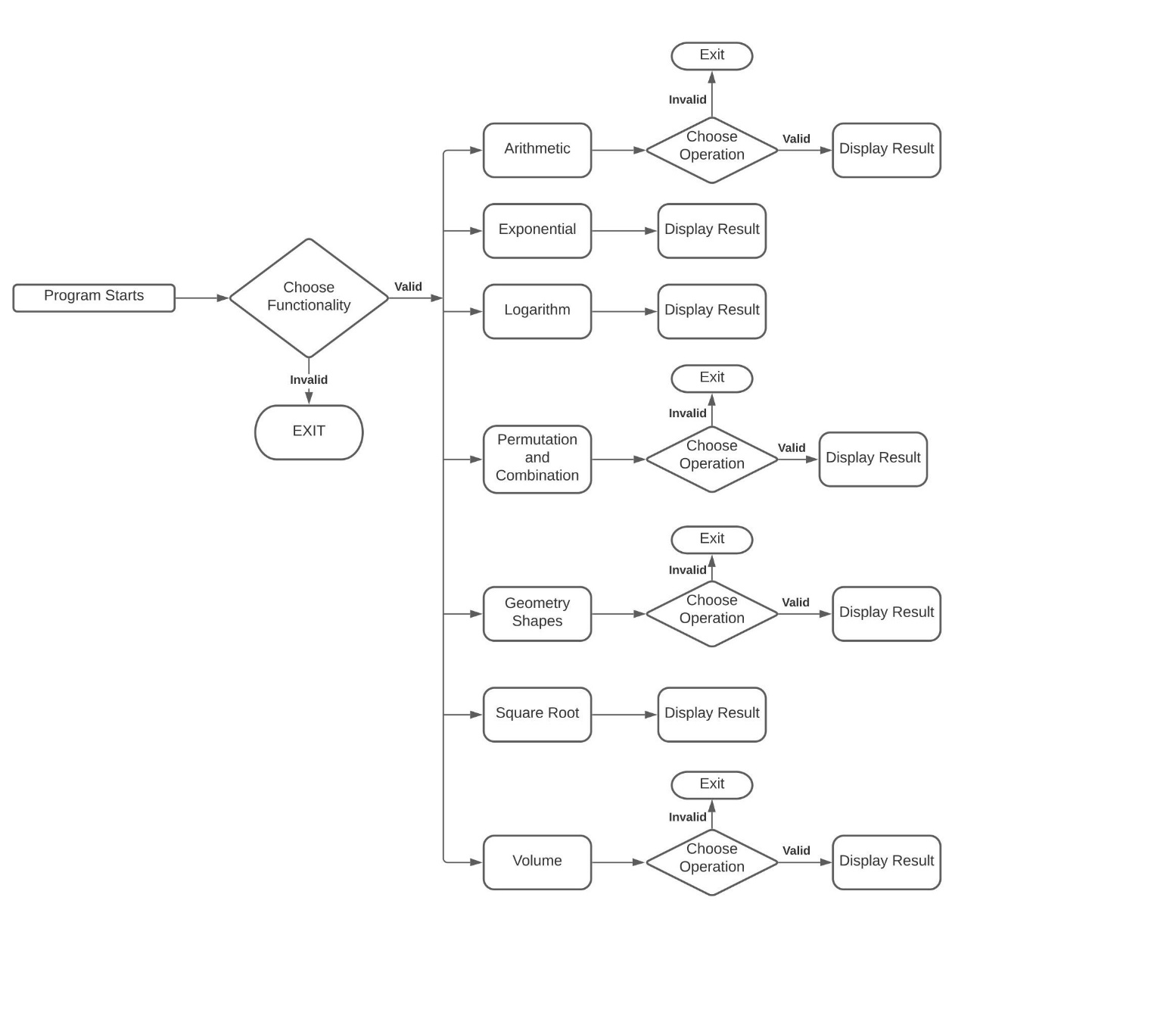
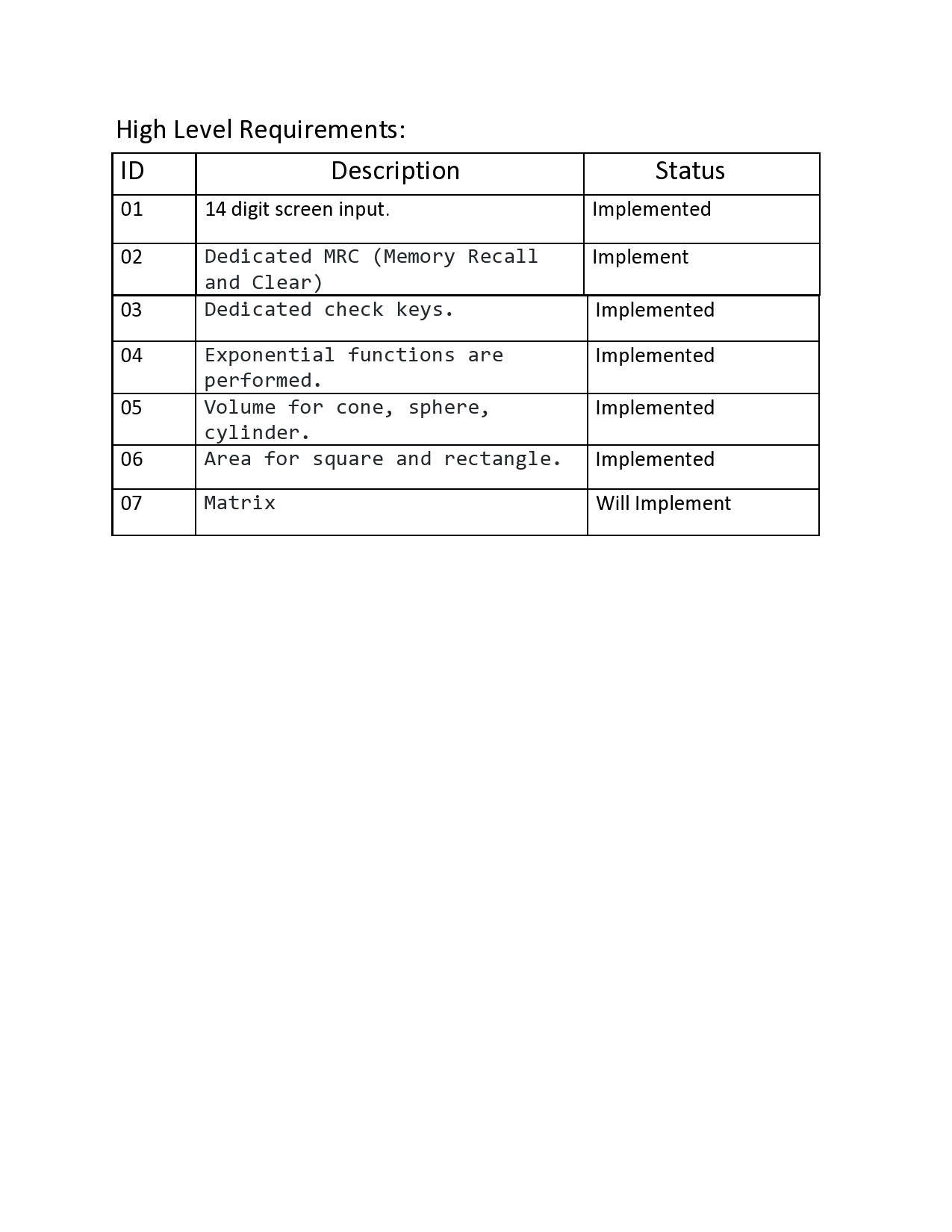
**Statics comparing the previous data with present data.**

## How:

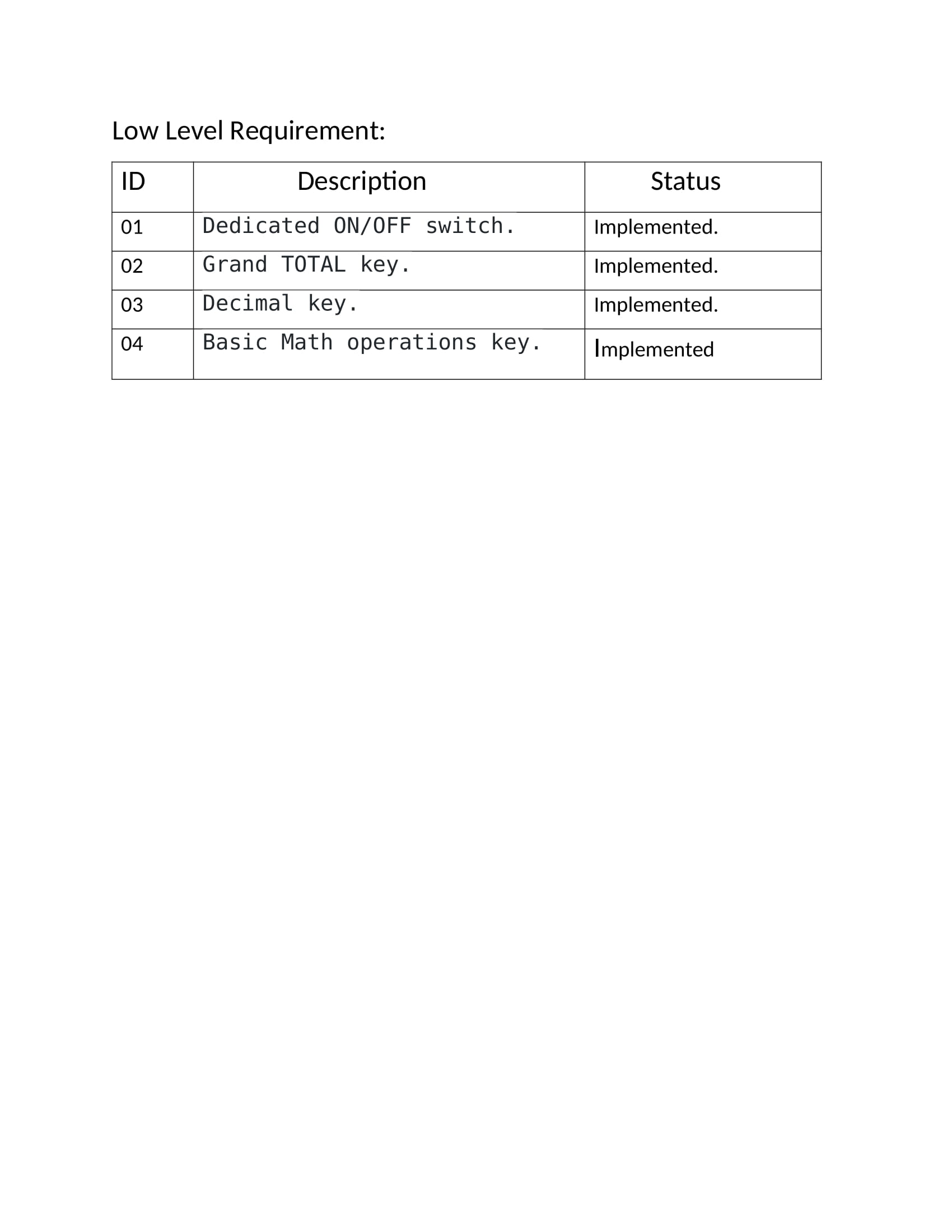
**Easy to user interface.**

# Detail requirements:

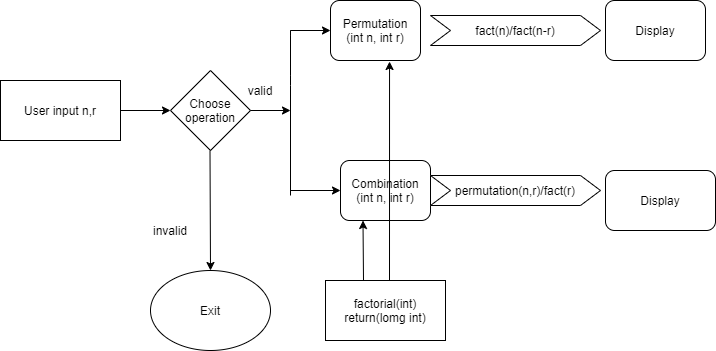
## High Level Requirements:

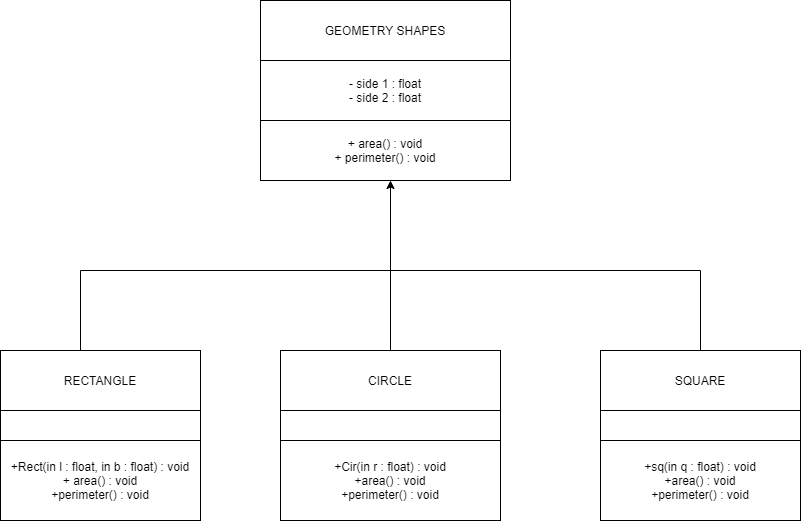


## Low level Requirements:

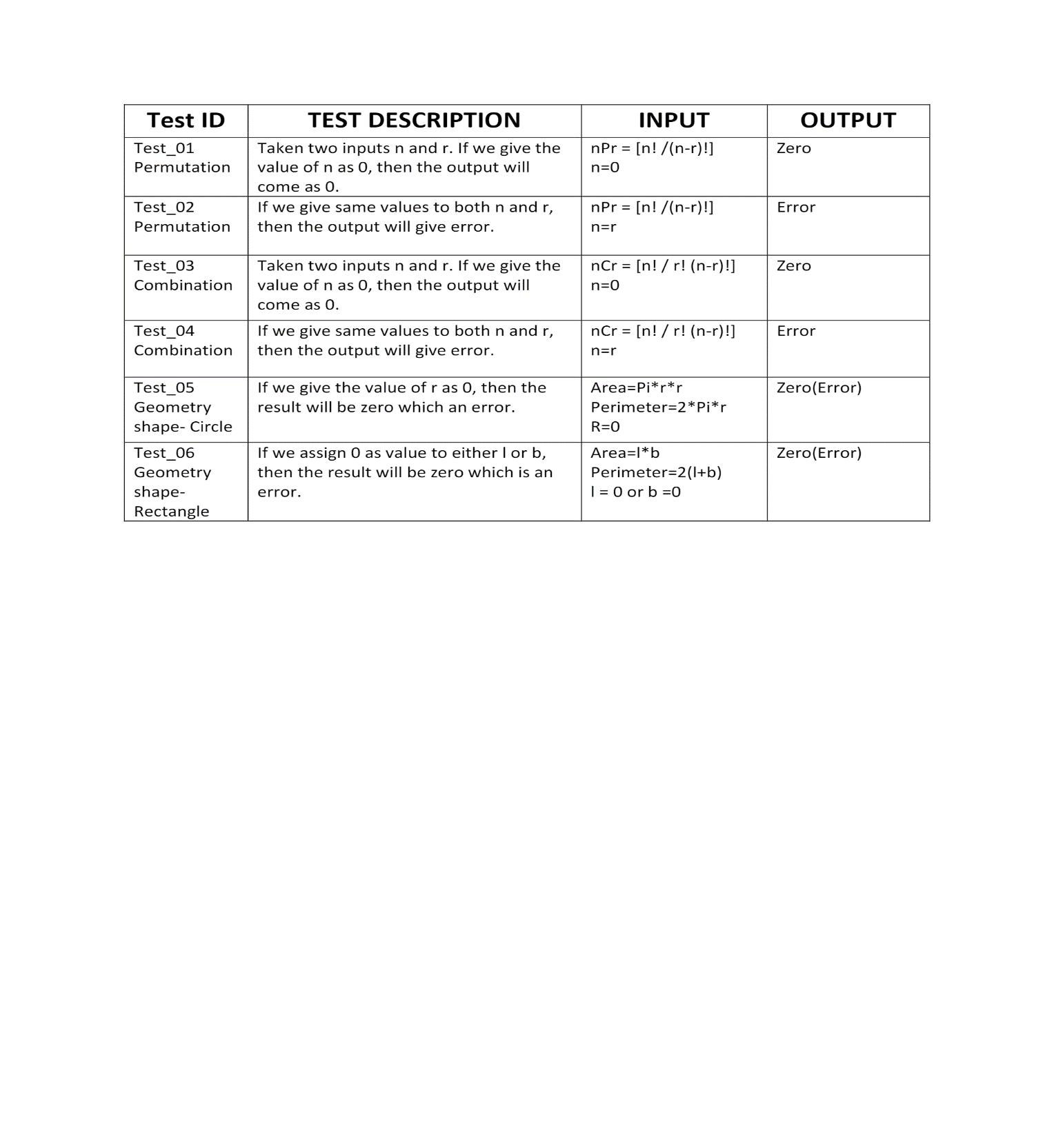


|  |
| --- |
|  |
|  |  |
|  | High level requirement: |
|  | 1. Permutation and Combination. |
|  | 2. Geometry shape. |
|  |  |
|  | Low level requirement: |
|  | 1. Permutation: Input two values a and r and both the values must be int. The result can never be in float. |
|  | The result value therefore will be in int using the permutation function. |
|  | Combination: Input two values c and d and both the values must be int.They can never be in float. |
|  | The result value therefore will be in int using the permutation function. |
|  | 2. Geometry shapes: For square input q.It can either be in int or float. Therefore, the result will be either in int or |
|  | float using the area formula (q^2). The perimeter result will also be either in int or float using the |
|  | formula (4q). |
|  | For circle input e.It can either be in int or float. The result will be either in int or |
|  | float using the area formula [(Pi\*r\*r). The perimeter result will also be either int or float using the |
|  | formula (2\*Pi\*r). |
|  | For rectangle input l and b.It can either be in int or float.The result will be either int or |
|  | float using the area formula (l\*b). The perimeter result will also be either int or float using the |
|  | formula [2(l+b)]. |
|  |  |





Test Cases:

****

**ACTIVITY 2: AGILE METHODOLOGY**

AGILE methodology is a practice that promotes **continuous iteration** of development and testing throughout the software development lifecycle of the project. In the Agile model, both development and testing activities are concurrent, unlike the Waterfall model.

The **Agile software development** methodology is one of the simplest and effective processes to turn a vision for a business need into software solutions. Agile is a term used to describe software development approaches that employ continual planning, learning, improvement, team collaboration, evolutionary development, and early delivery. It encourages flexible responses to change.

The agile software development emphasizes on four core values.

1. Individual and team interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

EPIC:

An epic is a large body of work that can be broken down into a number of smaller [stories](https://www.atlassian.com/agile/project-management/user-stories), or sometimes called “Issues” in Jira. Epics often encompass multiple teams, on multiple projects, and can even be tracked on multiple boards.

Epics are almost always delivered over a set of sprints. As a team learns more about an epic through development and customer feedback, user stories will be added and removed as necessary. That’s the key with agile epics: Scope is flexible, based on customer feedback and team cadence.

USER STORIES:

A user story is the smallest unit of work in an agile framework. It’s an end goal, not a feature, expressed from the software user’s perspective.

A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer.

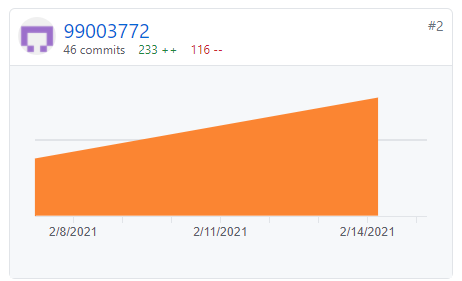
The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer. Note that "customers" don't have to be external end users in the traditional sense, they can also be internal customers or colleagues within your organization who depend on your team.

User stories are a few sentences in simple language that outline the desired outcome. They don't go into detail. Requirements are added later, once agreed upon by the team.

Stories fit neatly into agile frameworks like scrum and Kanban. In scrum, user stories are added to sprints and “burned down” over the duration of the sprint. Kanban teams pull user stories into their backlog and run them through their workflow. It’s this work on user stories that help scrum teams get better at [estimation](https://www.atlassian.com/agile/project-management/estimation) and sprint planning, leading to more accurate forecasting and greater agility. Thanks to stories, Kanban teams learn how to manage work-in-progress (WIP) and can further refine their workflows.

User stories are also the building blocks of larger agile frameworks like epics and initiatives. Epics are large work items broken down into a set of stories, and multiple epics comprise an initiative. These larger structures ensure that the day to day work of the development team (on stores) contributes to the organizational goals built into epics and initiatives.

Git commits



Git issue:

