# SOFTWARE PERFORMANCE REPORT TRACKER SYSTEM

## High Level Design

## &

## Low Level Design

## Document Control

## History

| SOFTWARE PERFORMANCE REPORT TRACKER SYSTEM |
|---|

| Date | Version | Author | Brief Description of Changes | Approver |
|---|---|---|---|---|
| | | | | |
| | | | | |

# Index

# 1. Introduction

## 1.1 Intended Audience:

The Developers working on a Software and Users that are using the software are the intended audience for the software performance report management, so that they can raise their issues and the developer of a domain can solve and manage that issue based on priority status and severity and hence help in improving the manageability of a software

## 1.2 Project Purpose:

The purpose of the project is to develop a Software Performance Report tracking system using which we can manage a software/ Product by tracking the user issues and resolving issues based on priority - high, medium and low or severity. This software makes it easier for the organization to manage and improve their product.

## 1.3 Key Project Objectives:

1. Allow users to login and get there personalized spr
2. Validates username and password
3. Allow the user to register a SPR
4. Detecting the issues in the Software Performance Report
5. Boost productivity through improve performance
6. Identify areas for improvement

## 1.4 Project scope:

1. The strategy is to identify the features required for a software performance management system and how it will solve current & future issues about the system.
2. A system that will make it easier for the user to conduct or perform the task of tracking the issue and getting it resolved.
3. The system will record user issues during tracking.
4. In the future we can add more features in our project like we add different views for different people/ if a group is working on a product so the group sprs.

## 1.5 Functional Overview:

### 1.5.1 Header files in the program:

#include<stdio.h>

#include <stdlib.h>

#include <string.h>

#include <main.h>

#include <struct.h>

#include <log.h>

### 1.5.2 Functions:

The following functions are included in the program:

**1.5.2.1 SPR_01 >>** Void get_user_data_from_csv (); This function gets all the user data from a csv file and loads it into a hashmap for authentication Making it faster to authenticate user login.

**1.5.2.2 SPR_02 >>** int authenticate (); This function lets users authenticate into their respective account.

**1.5.2.3 SPR_03 >>** Void create_spr (); This function gets the input from the user about product, creator, assignee, severity, priority, and spr description and creates a new SPR.

**1.5.2.4 SPR_04 >>**Void modify_spr (spr_id); This function gets the id of spr that he wants to modify and the changes that he wants to make in an spr and modifies the SPR.

**1.5.2.5 SPR_05 >>**Void show_spr (bool function);This function should not take bool but spr_d to search through list

# 2. Design Overview

Software Performance Report Tracker System comprises the following modules:

| | |
|---|---|
| Name of the Module | src-> modify_spr |
| Handled by | Sanjeev Yadav |
| Description | This function gets the id of spr that user wants to modify |

| | |
|---|---|
| Name of the Module | src->authenticate_user |
| Handled by | Nimish khattar |
| Description | This function lets users authenticate into their respective account |

| | |
|---|---|
| Name of the Module | src->show_spr |
| Handled by | Janaki Medisetti |
| Description | This function takes in a bool function as a parameter for filter and displays the list of spr applying that filter |

| | |
|---|---|
| Name of the Module | src->iofiles |
| Handled by | Happy Kamra |
| Description | Takes input from csv and write in csv |

| | |
|---|---|
| Name of the Module | src->create_spr |
| Handled by | Sonu |
| Description | This function gets the input from the user about product,creator,assignee,severity,priority |

**2.1 Design Objectives:**

If User is facing any issue while using SPR management System

1. Severity
2. Priority
3. Status

**SEVERITY**:

Severity means the type of issue being raised. In the severity of so many issues generated, while resolving those issues SPR will have the following stages.

A. Enhancement : In enhancement the issue raised is a suggestion for improving and Enhancing product Experience.
B. Server Defect: In the Server Defect employee login issue we resolve in this stage
C. Defect : In Defect we resolve employee login defects issue.
D. UI issue: Font and design-based issue are resolved
E. Nuisance: Small glitches

**PRIORITY:**

Priority means a resolved issue. In the Priority while resolving issues SPR will have following stages. Priority, as the name suggests, is about prioritizing a defect based on business needs and severity of the defect. Priority signifies the importance or urgency of fixing a defect.

A. Top: This has to be fixed immediately within 24 hours. This occurs when an entire Top functionality is blocked and no testing can proceed because of this.
B. High: The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed.
C. Medium: During the normal course of the development activities, defects should be resolved. It can wait until a new version is created
D. Low: The Defect is an irritant but repair can be done once the more serious Defect has been fixed.

**Status**: It shows the status of the report submitted. What is being done on that issue being raised.

Idle: This means that no one is working on the issue

Active: This means that the issue is being resolved

Inactive: this means that I have resolved the issue

**2.2 User Interface Paradigms:** The developer resolves the issues assigned to him and user can create a new SPR, they are modify the existing SPR and selected there attributes in the submenu

## 2.3 Validations:

New users should register before login or else it displays the no user found. Registered users have to login with valid credentials. Otherwise, they will get invalid username or password. After login, the user can use SPR.

## 2.4 Performance:

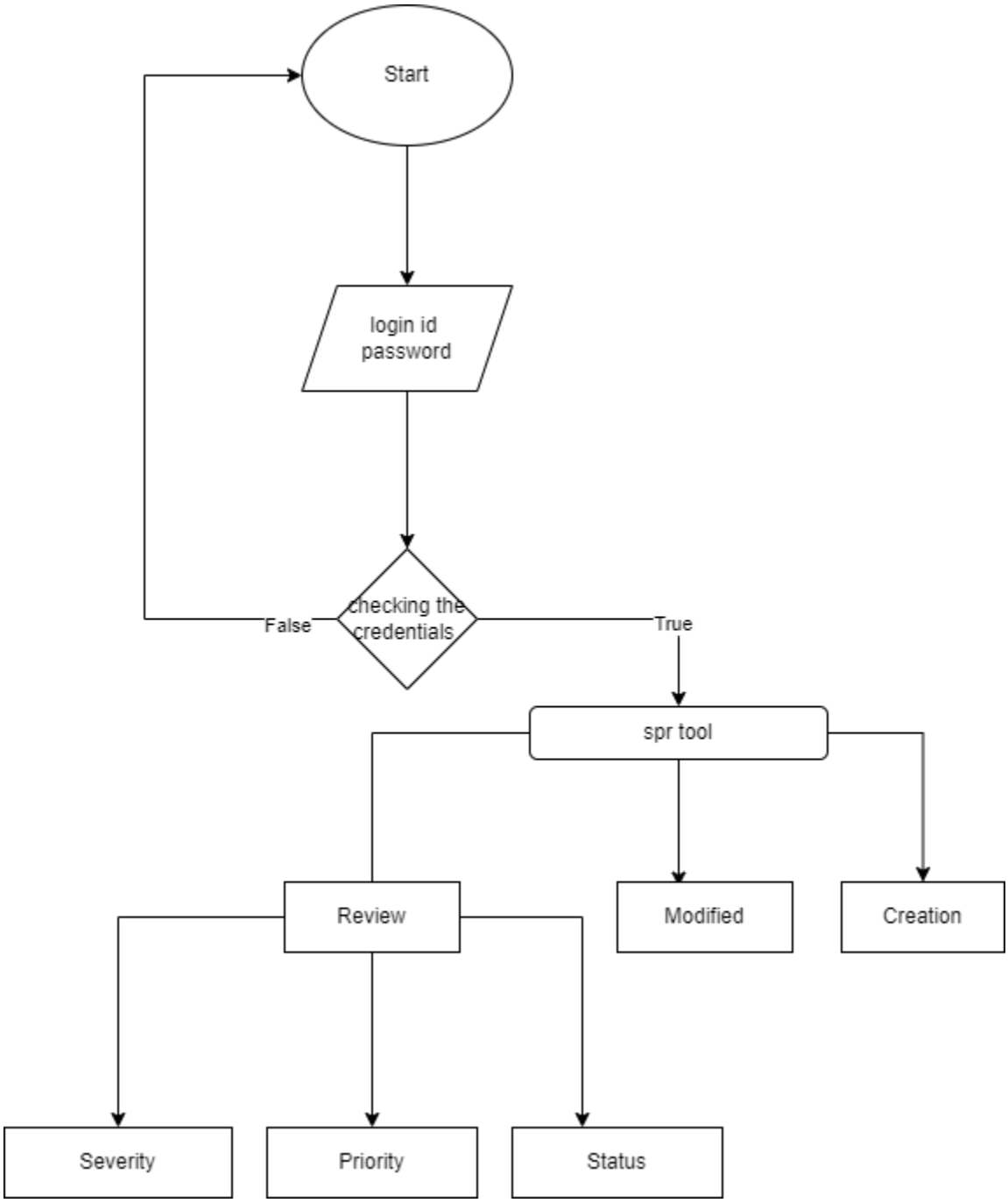
The system will work on the client terminal. The performance depends on the hardware component of the user's system.

## 2.5 Maintenance:

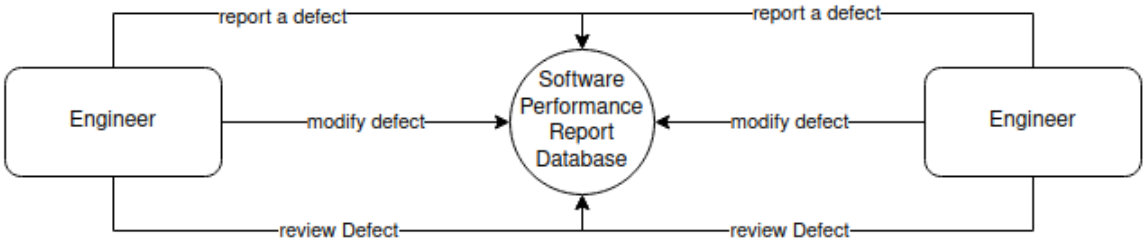
Very little maintenance should be required for this setup. An initial configuration Will be the only system required interaction after it is put together. The only other user maintenance would be any changes to settings after setup, and any specified special cases where user settings or history need to be changed. Physical maintenance on the system's parts may be required and would cause a temporary loss of data or Internet. Upgrades of hardware and software should have little effect on this project but may cause downtime.

# 3. Detailed System design

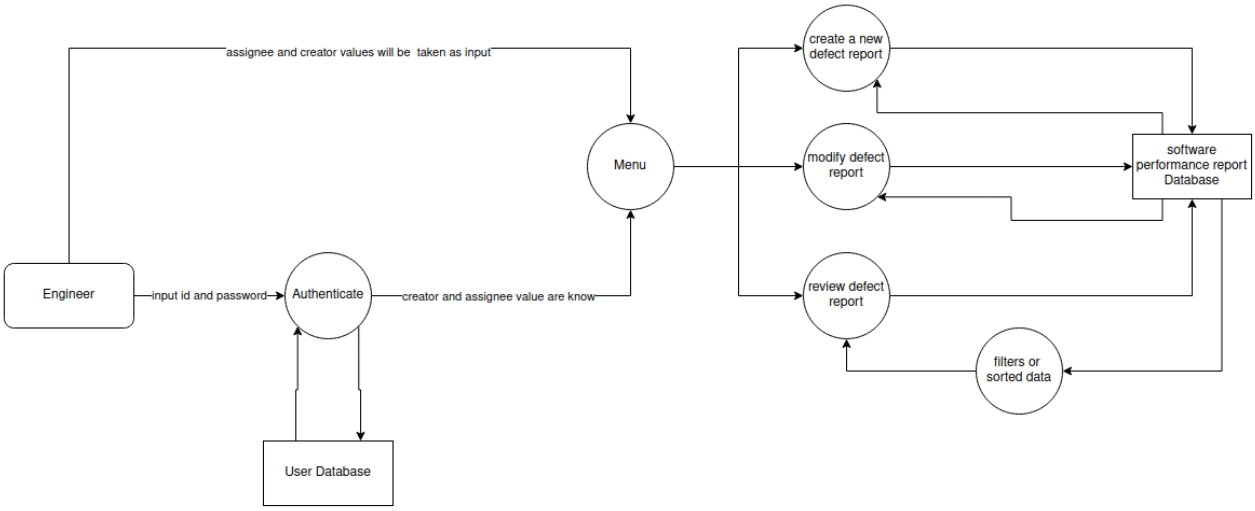
## 3.1 Flow chart:



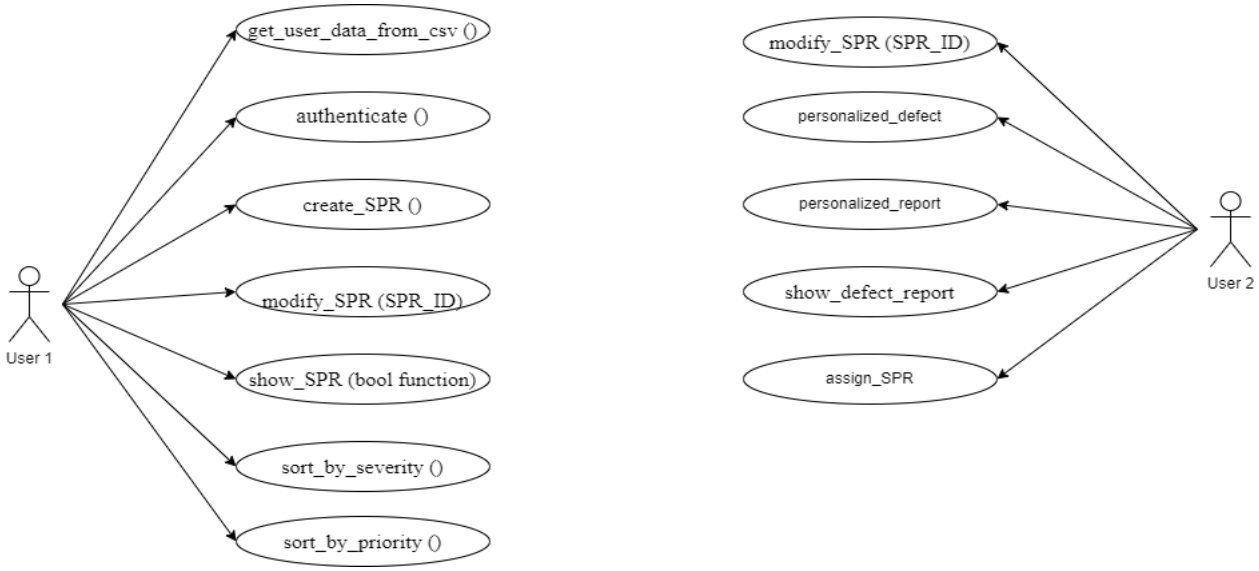
## 3.2 DFD of Level 0



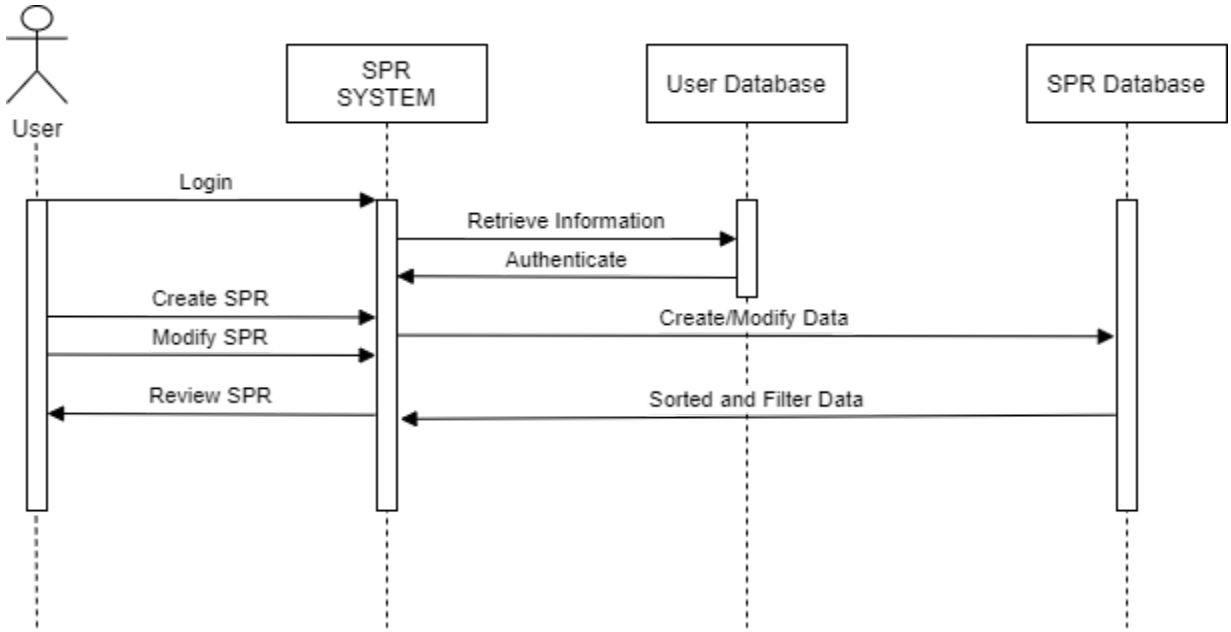

## 3.3 DFD of Level 1



## 3.1 Use Case Diagram




## 3.2 Sequence Diagram



## 5. Environment Description

**5.1 Time zone support -** IST

**5.2 Language Support -** English

**5.3 User Desktop Requirements-**

- 64-bit processor, 1GHz or faster
- At least 2 GB free hard drive space
- At least 1 GB RAM

**5.4 Server-side requirements**

- 32 64-bit processors, 1 GHz or faster
- At least 1 GB free hard drive space
- At least 1GB RAM

**5.4.1. Deployment Considerations: -**

- Easy setup: no session storage daemon, use impf's and memory caching to enhance performance.
- Local storage is used
- No network latency to consider
- To scale buy a bigger CPU, more memory, larger hard drive, or additional hardware

**Application Server Disk Space: -**

No such disk space is required as the program is fully functional on online IDE (s) as well. A local Operating System is required and one txt file to store the records of processes.

**5.4.2 Database Server Disk Space: -**

No such disk space is required as the program is fully functional on online IDE(s) as well.

### 5.4.3 Integration Requirements: -

a. Language:- C

b. Tools:- Valgrind, Splint

c. Compiler:- g++

d. Linux Environment

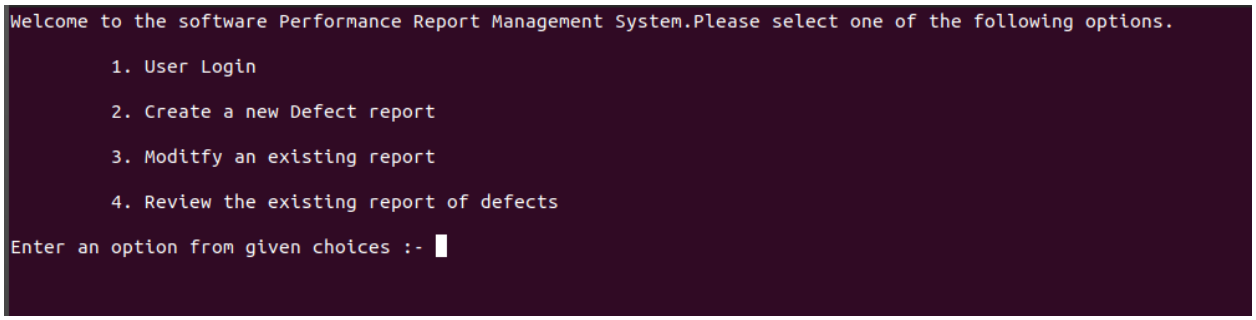
## 5.6 Configuration

5.6.1 Operating System - Linux Environment

## 6. Experimental Results

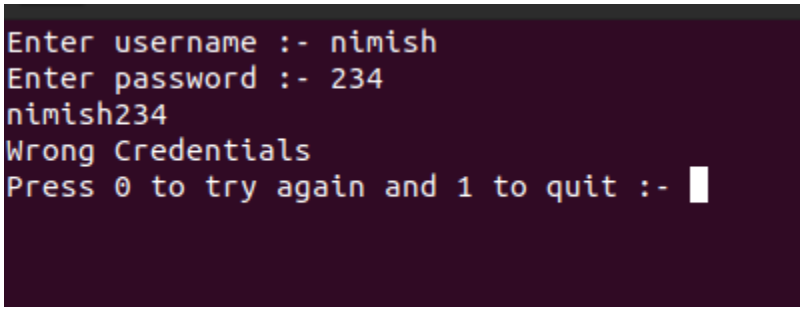
### 6.1 Integration testing

IT_CASE 1 : WELCOME

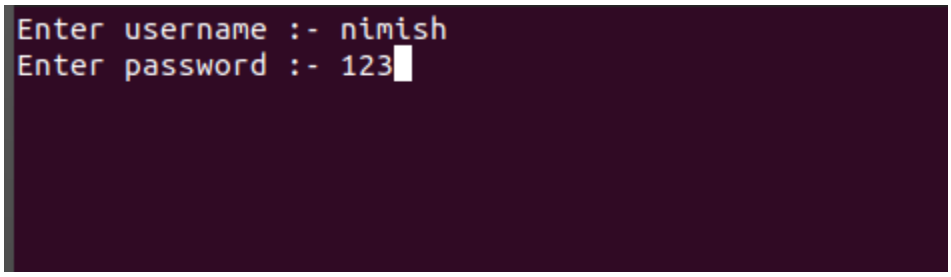



IT_CASE 2 :AUTHENTICATE

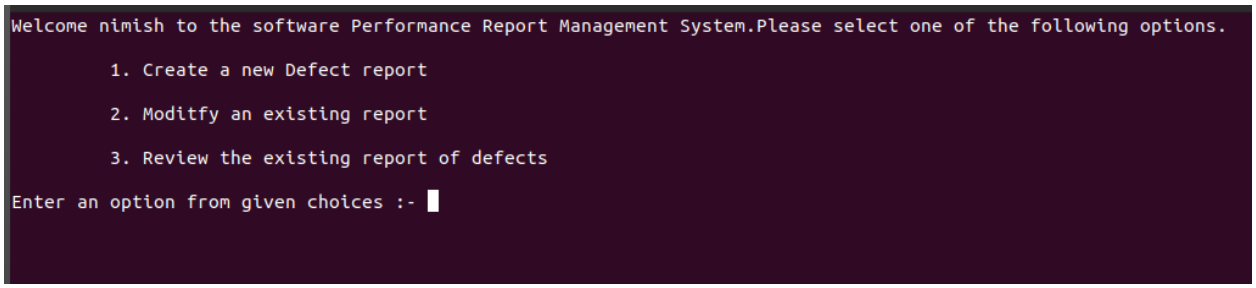
WRONG CREDENTIAL



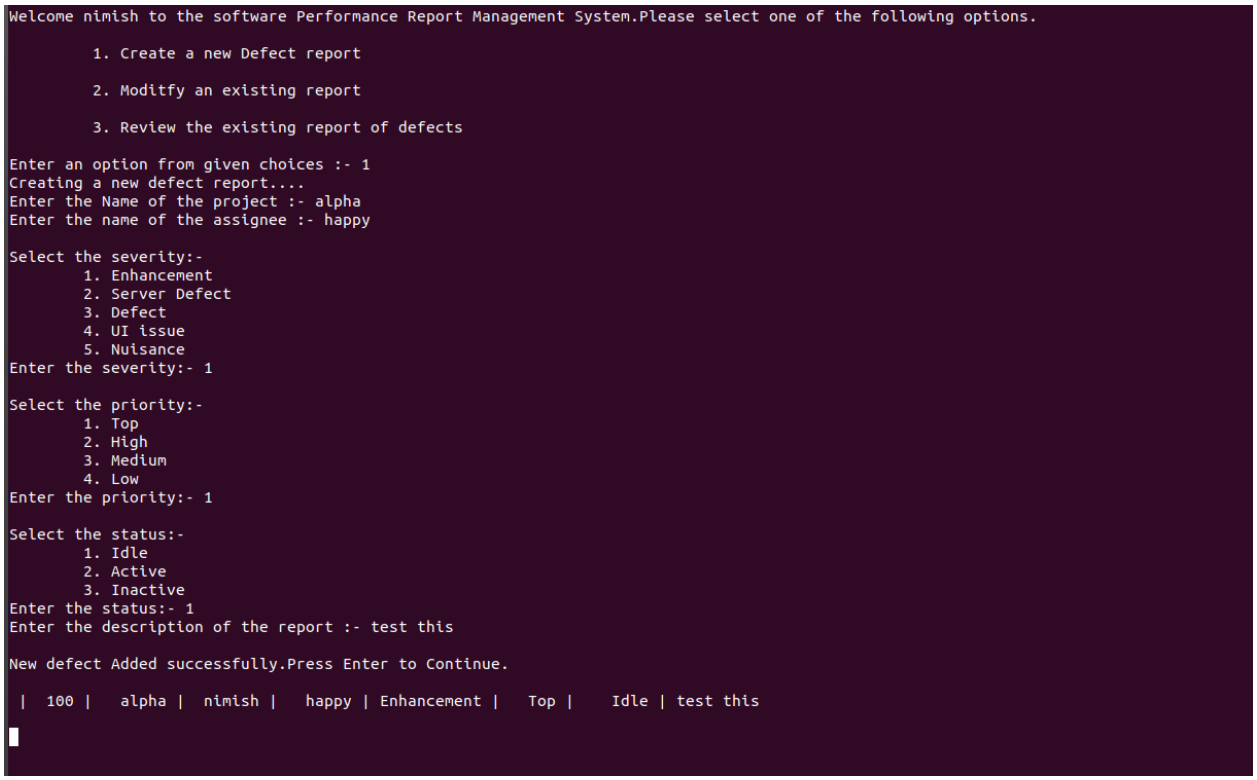
RIGHT CREDENTIAL



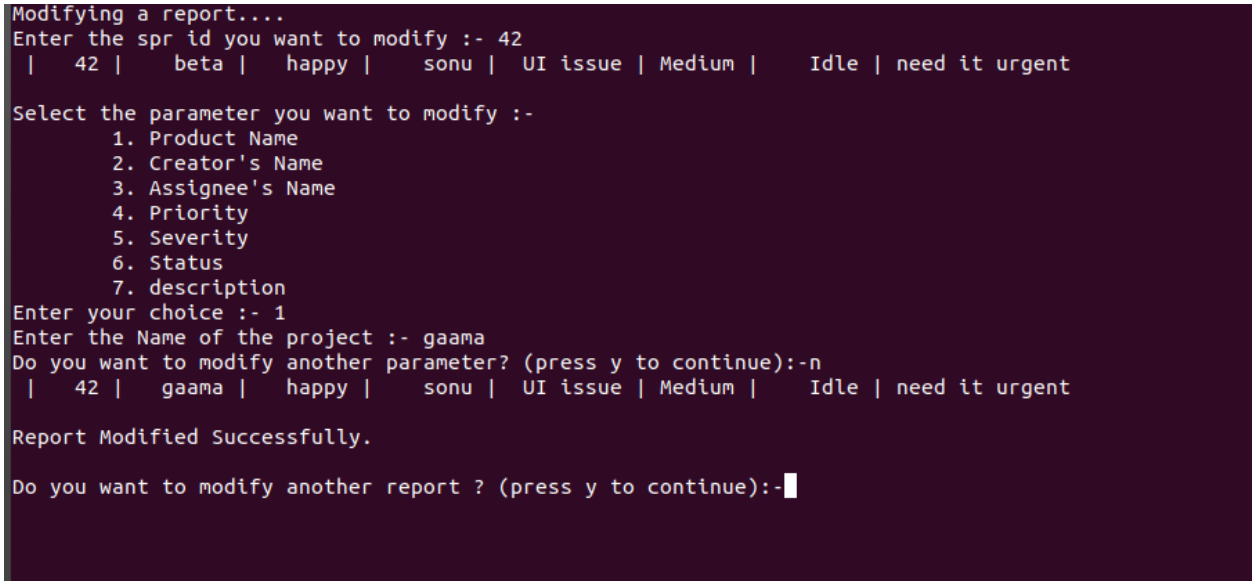

LOG IN SCREEN :



IT CASE 3 : CREATE




IT_CASE 4 : MODIFIER

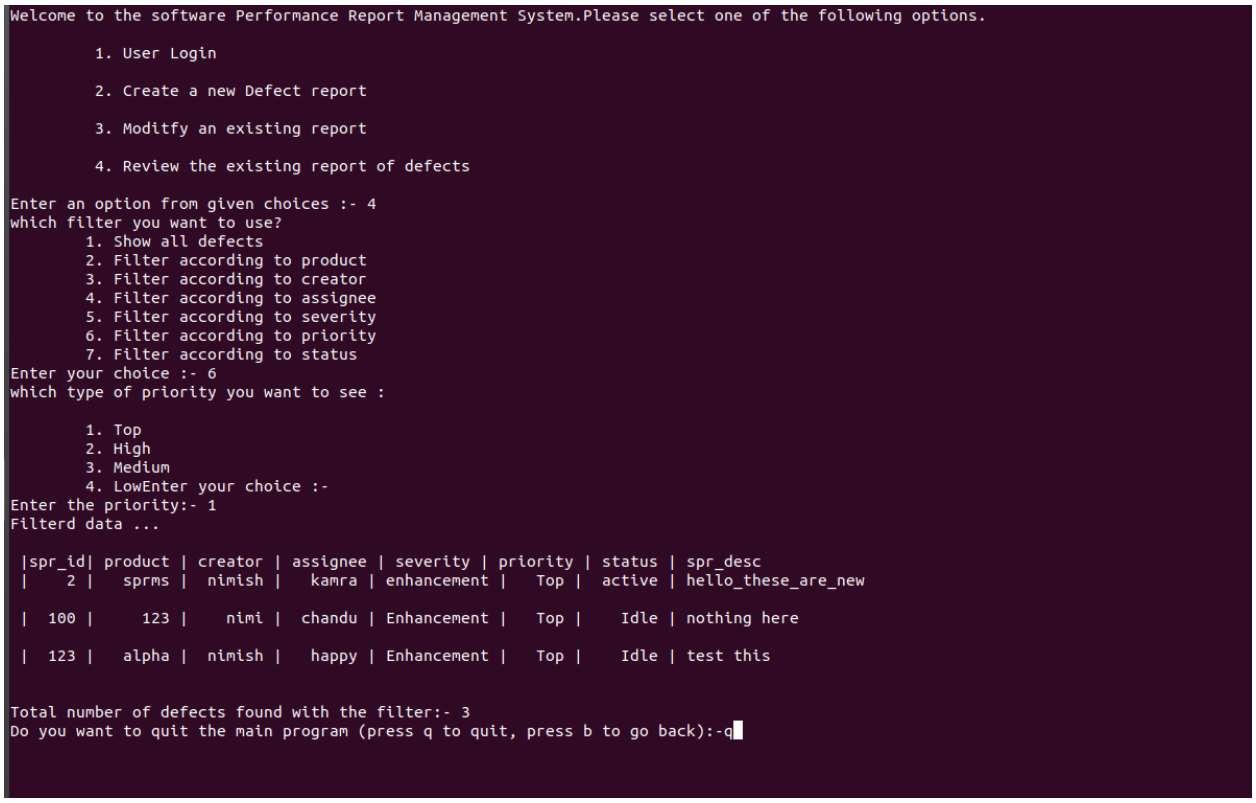


IT CASE 5: SHOWS DEFECT

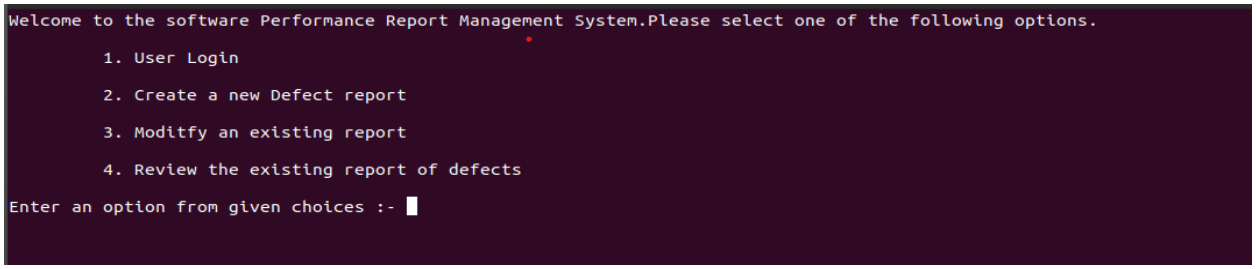
```
which filter you want to use?
       1. Show all defects
       2. Filter according to product
       3. Filter according to creator
       4. Filter according to assignee
       5. Filter according to severity
       6. Filter according to priority
       7. Filter according to status
Enter your choice :- 1
All the data is shown here:
Filterd data ...

 |spr_id| product | creator | assignee | severity | priority | status | spr_desc
 |   22 |   sprms |  nimish |   kamra | enhancement |  High |  active | hello_these_are_new

 |   12 |    spms |  nimish |  janaki | enhancement |   Low |    idle | hello_these_are_new

 |   32 |   sprms |  nimish |    sonu | enhancement | medium | inactive | hello_these_are_new

 |    2 |   sprms |  nimish |   kamra | enhancement |   Top |  active | hello_these_are_new

 |   13 |   sprms | kingpin |  janaki | enhancement |   low |    idle | hello_these_are_new

 |   31 |   sprms |    nomu |    sonu | enhancement | Medium | inactive | hello_these_are_new

 |  100 |     123 |    nimi |  chandu | Enhancement |   Top |    Idle | nothing here

 |  123 |   alpha |  nimish |   happy | Enhancement |   Top |    Idle | test this

 |   42 |    beta |  nimish |    sonu |  UI issue | Medium |  Active | need it urgent


Total number of defects found with the filter:- 9
If you want to logout press q :-
```

IT CASE 6 : MENU IS NOT WORKING CORRECTLY

BEFORE PRESSING ENTER




AFTER PRESSING ENTER



## 7. Tools Report

### 7.1 splint

splint main()

```
happy@happy-VirtualBox:~/Downloads/updated 4/src$ splint main.c
Splint 3.1.2 --- 20 Feb 2018

main.c: (in function main)
main.c:18:9: Unrecognized identifier: user
  Identifier used in code has not been declared. (Use -unrecog to inhibit
  warning)
main.c:18:15: Unrecognized identifier: hashTable
main.c:18:25: Unrecognized identifier: HASHSIZE
main.c:18:9: Statement has no effect: user * hashTable...
  Statement has no visible effect --- no values are modified. (Use -noeffect to
  inhibit warning)
main.c:19:44: Parse Error. (For help on parse errors, see splint -help
              parseerrors.)
*** Cannot continue.
happy@happy-VirtualBox:~/Downloads/updated 4/src$
```

Splint Authenticate()

```
happy@happy-VirtualBox:~/Downloads/spr_msys/src/authenticate_user$ splint mainlogin.c
Splint 3.1.2 --- 20 Feb 2018

mainlogin.c:4:18: Cannot find include file main.h on search path:
                    /usr/include/x86_64-linux-gnu;/usr/include
  Preprocessing error. (Use -preproc to inhibit warning)
mainlogin.c:5:20: Cannot find include file struct.h on search path:
                    /usr/include/x86_64-linux-gnu;/usr/include
mainlogin.c:6:17: Cannot find include file log.h on search path:
                    /usr/include/x86_64-linux-gnu;/usr/include
Preprocessing error for file: /home/happy/Downloads/spr_msys/src/authenticate_us
                    er/mainlogin.c
*** Cannot continue.
happy@happy-VirtualBox:~/Downloads/spr_msys/src/authenticate_user$
```

Splint Create()

```
happy@happy-VirtualBox:~/Downloads/spr_msys/src/create_spr$ splint maincreate.c
Splint 3.1.2 --- 20 Feb 2018

maincreate.c:4:18: Cannot find include file main.h on search path:
                     /usr/include/x86_64-linux-gnu;/usr/include
  Preprocessing error. (Use -preproc to inhibit warning)
maincreate.c:5:20: Cannot find include file struct.h on search path:
                     /usr/include/x86_64-linux-gnu;/usr/include
maincreate.c:6:17: Cannot find include file log.h on search path:
                     /usr/include/x86_64-linux-gnu;/usr/include
Preprocessing error for file: /home/happy/Downloads/spr_msys/src/create_spr/main
                     create.c
*** Cannot continue.
happy@happy-VirtualBox:~/Downloads/spr_msys/src/create_spr$
```

Splint show()

```
happy@happy-VirtualBox:~/Downloads/spr_msys/src/show_spr$ splint mainshow.c
Splint 3.1.2 --- 20 Feb 2018

mainshow.c:4:18: Cannot find include file main.h on search path:
                   /usr/include/x86_64-linux-gnu;/usr/include
  Preprocessing error. (Use -preproc to inhibit warning)
mainshow.c:5:20: Cannot find include file struct.h on search path:
                   /usr/include/x86_64-linux-gnu;/usr/include
mainshow.c:6:18: Cannot find include file log.h on search path:
                   /usr/include/x86_64-linux-gnu;/usr/include
Preprocessing error for file: /home/happy/Downloads/spr_msys/src/show_spr/mainsh
                   ow.c
*** Cannot continue.
happy@happy-VirtualBox:~/Downloads/spr_msys/src/show_spr$
```

Splint modify()

```
happy@happy-VirtualBox:~/Downloads/spr_msys/src/modify_spr$ splint mainmodify.c
Splint 3.1.2 --- 20 Feb 2018

mainmodify.c:8:18: Cannot find include file main.h on search path:
                     /usr/include/x86_64-linux-gnu;/usr/include
  Preprocessing error. (Use -preproc to inhibit warning)
mainmodify.c:9:20: Cannot find include file struct.h on search path:
                     /usr/include/x86_64-linux-gnu;/usr/include
mainmodify.c:10:17: Cannot find include file log.h on search path:
                      /usr/include/x86_64-linux-gnu;/usr/include
Preprocessing error for file: /home/happy/Downloads/spr_msys/src/modify_spr/main
                      modify.c
*** Cannot continue.
happy@happy-VirtualBox:~/Downloads/spr_msys/src/modify_spr$
```

7.2 Valgrind

```
==323627==
==323627== HEAP SUMMARY:
==323627==     in use at exit: 0 bytes in 0 blocks
==323627==   total heap usage: 139 allocs, 139 frees, 26,917 bytes allocated
==323627==
==323627== All heap blocks were freed -- no leaks are possible
==323627==
==323627== Use --track-origins=yes to see where uninitialised values come from
==323627== For lists of detected and suppressed errors, rerun with: -s
==323627== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

## 8.Reference


**Prof. Rahul Tarkunde ( Mentor)**

https://www.w3schools.com/c/

https://www.geeksforgeeks.org/c-programming-language/

https://www.tutorialspoint.com/cprogramming/index.htm